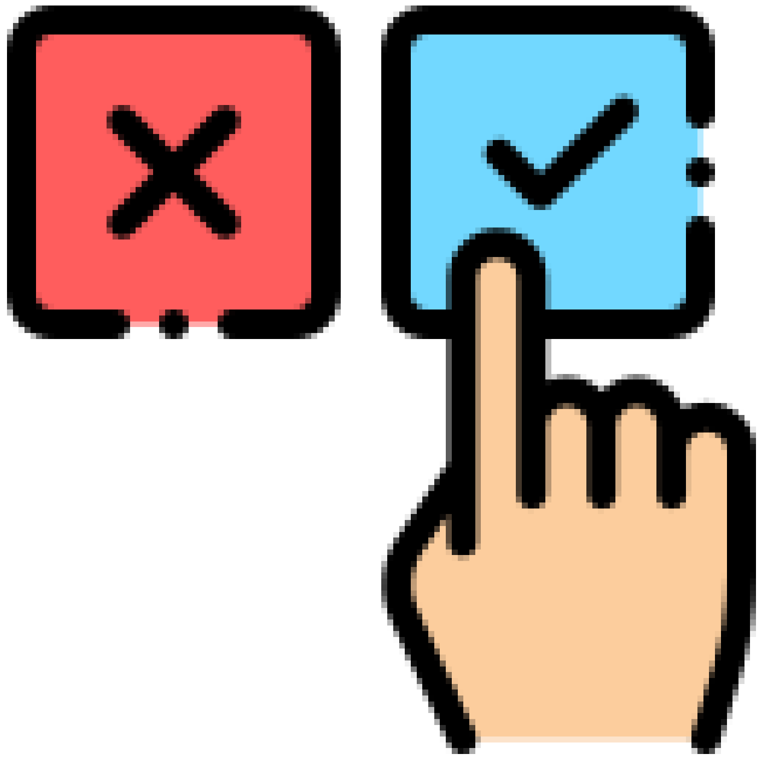
Introduction to differential privacy

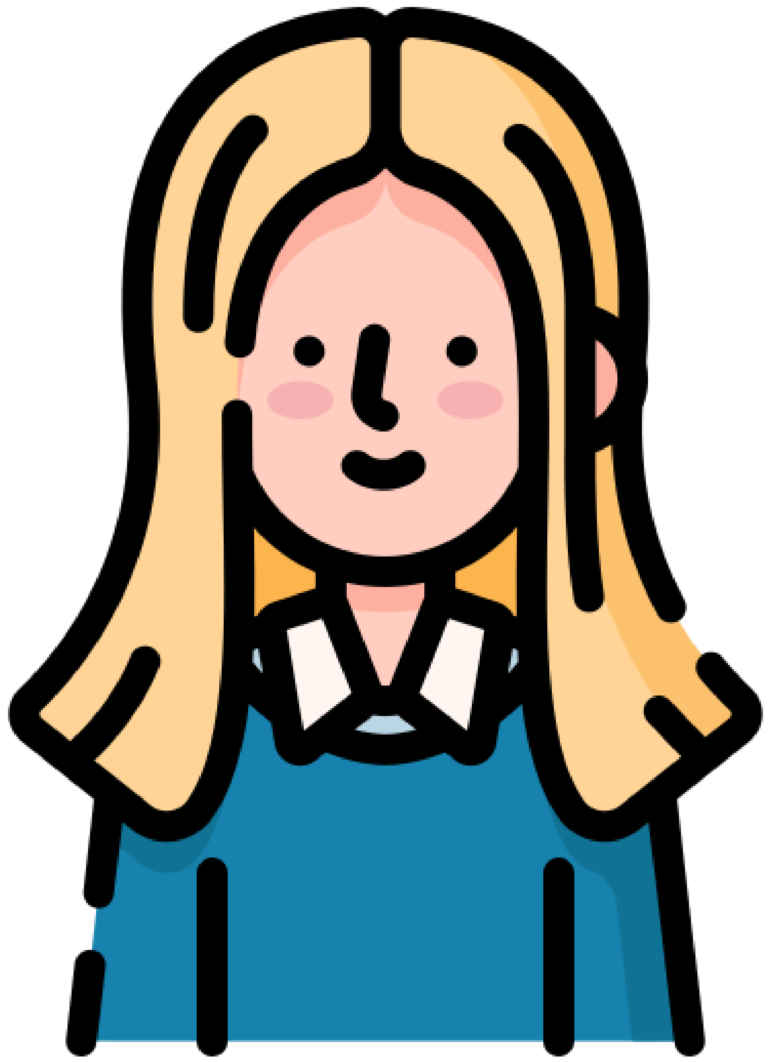
DATA PRIVACY AND ANONYMIZATION IN PYTHON

Rebeca Gonzalez
Instructor

What is differential privacy (DP)?

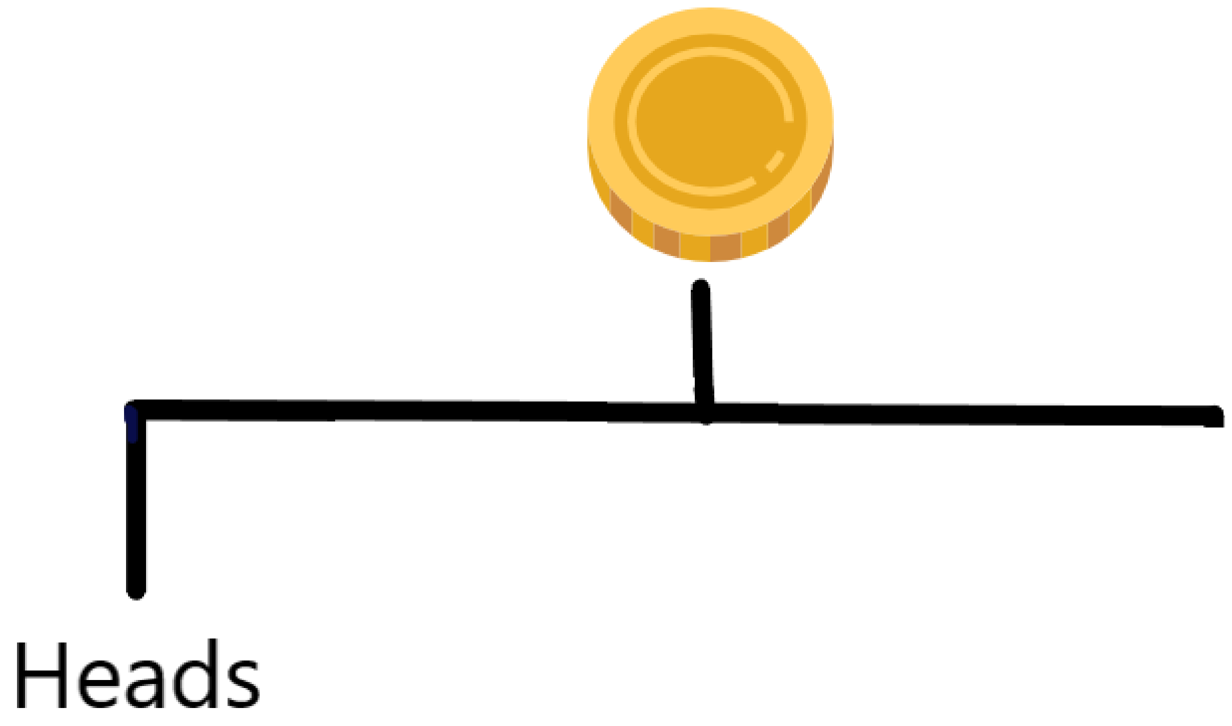
Do you dye your hair?



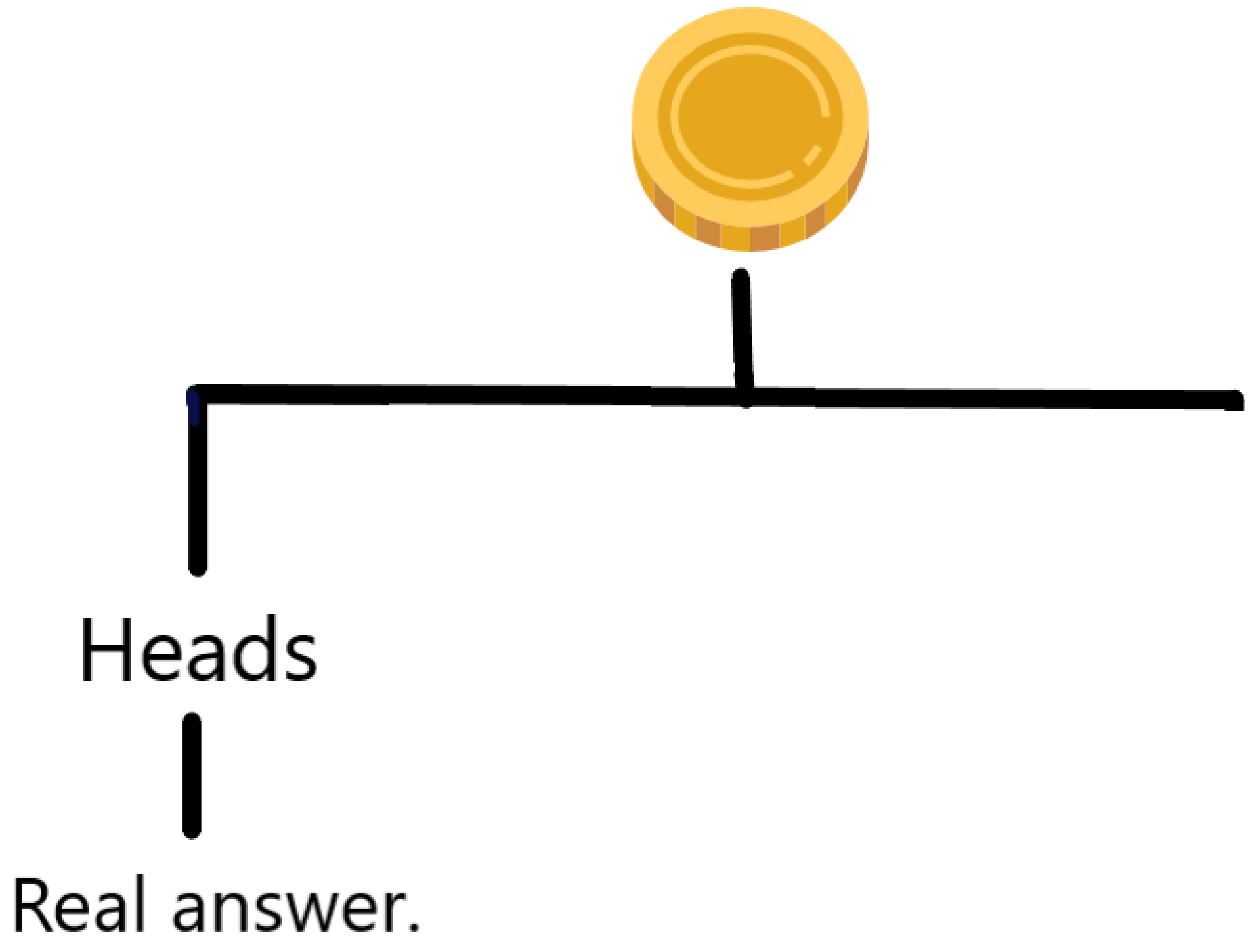


What is differential privacy (DP)?

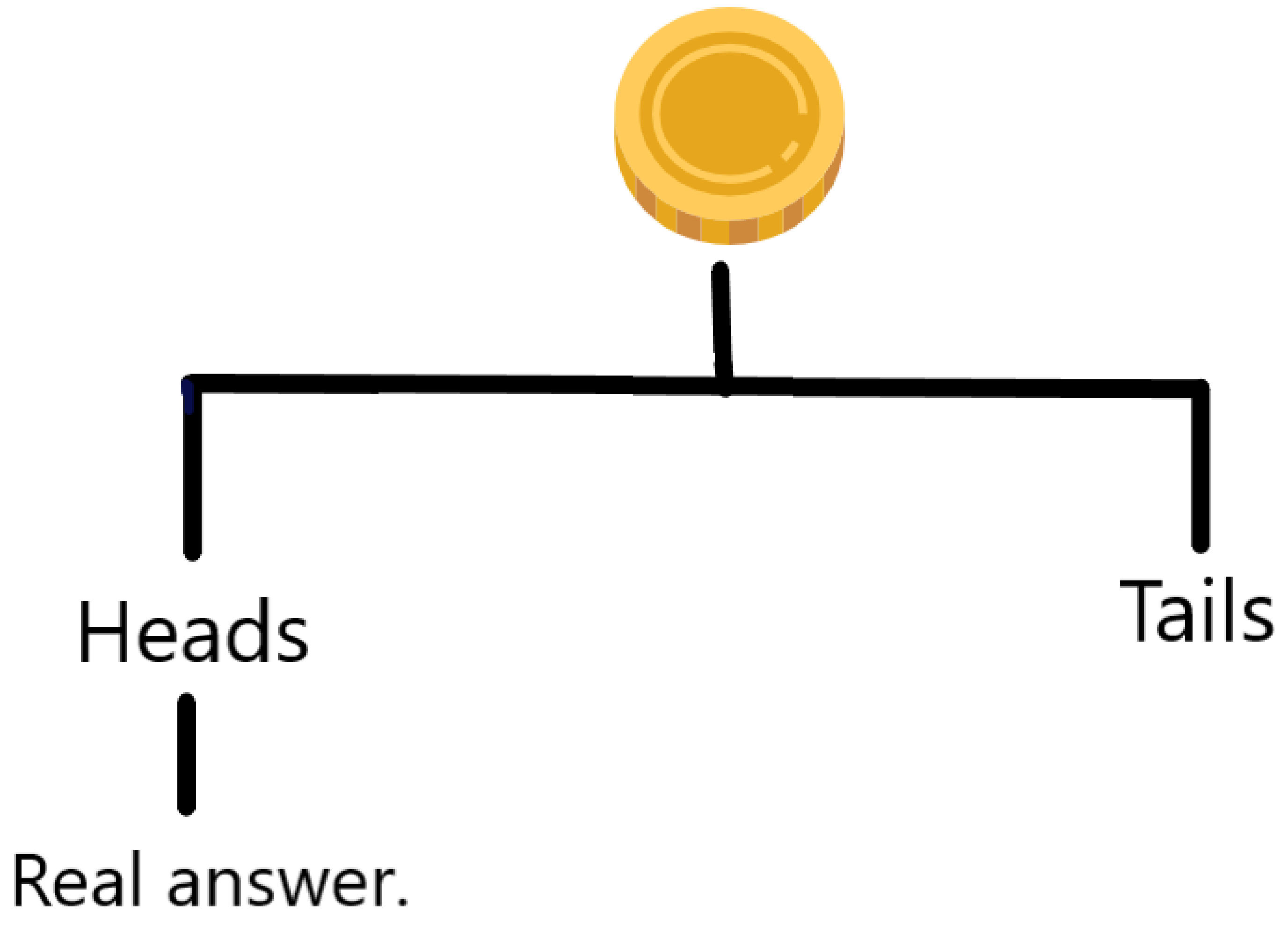
What is differential privacy (DP)?



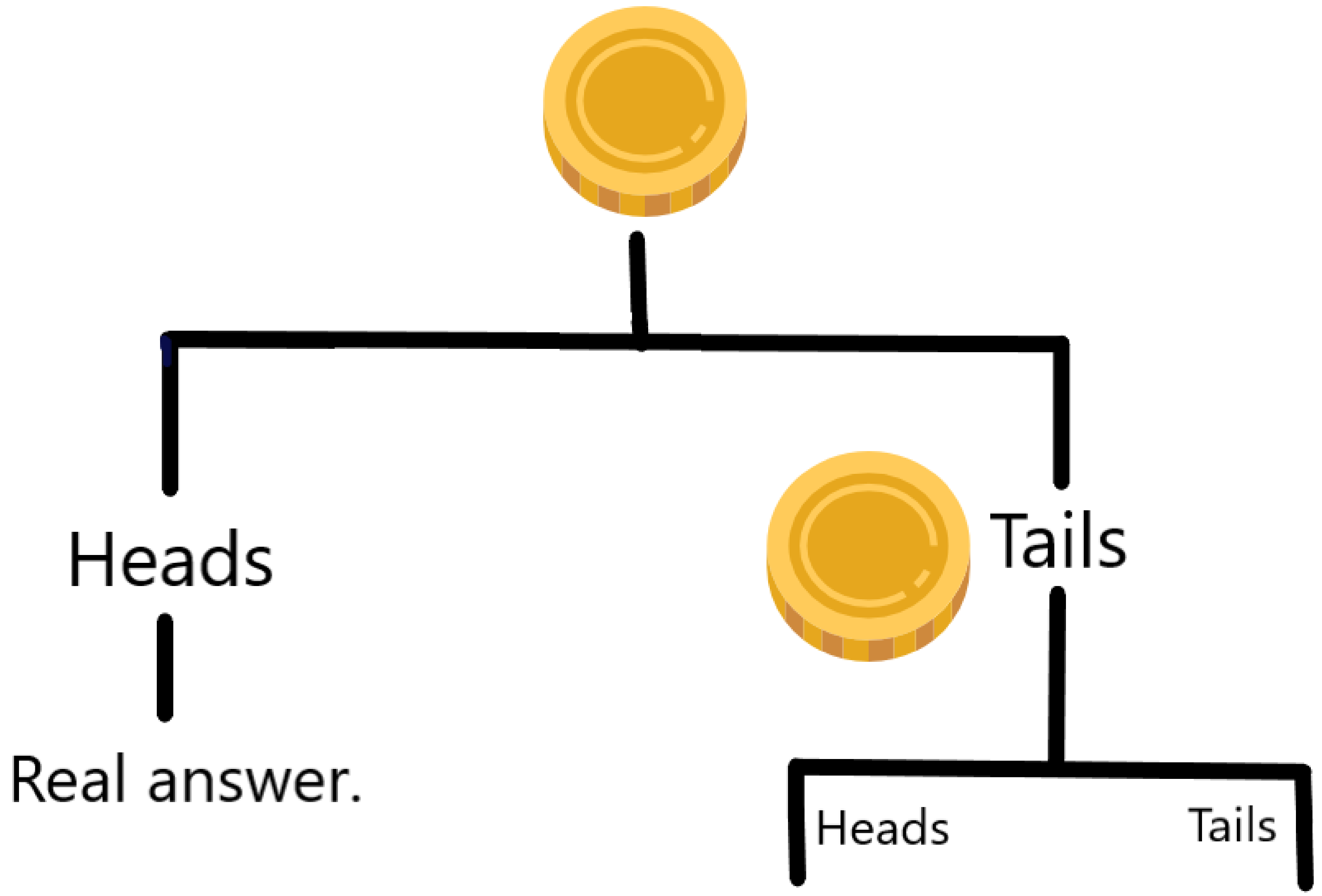
What is differential privacy (DP)?



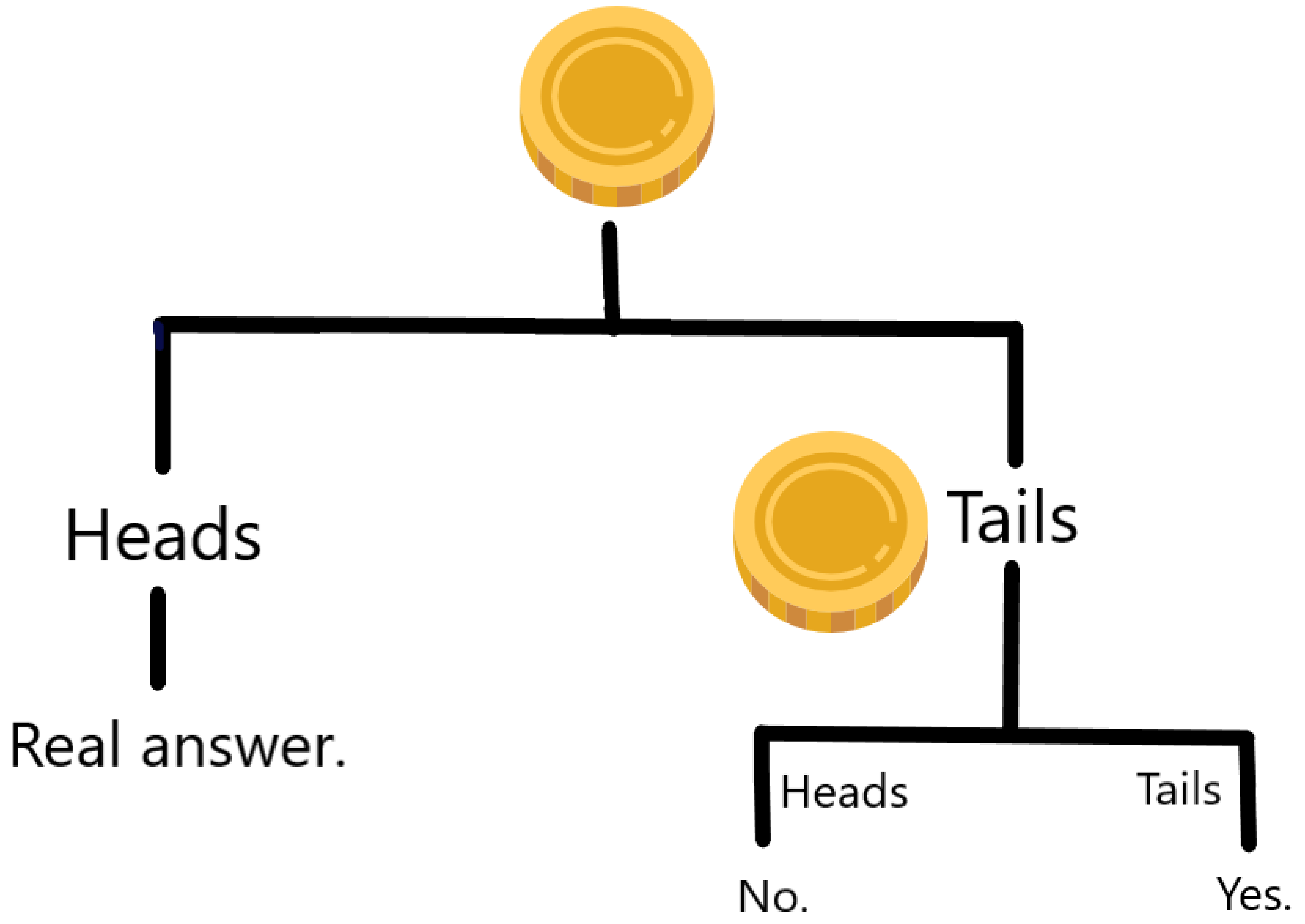
What is differential privacy (DP)?



What is differential privacy (DP)?

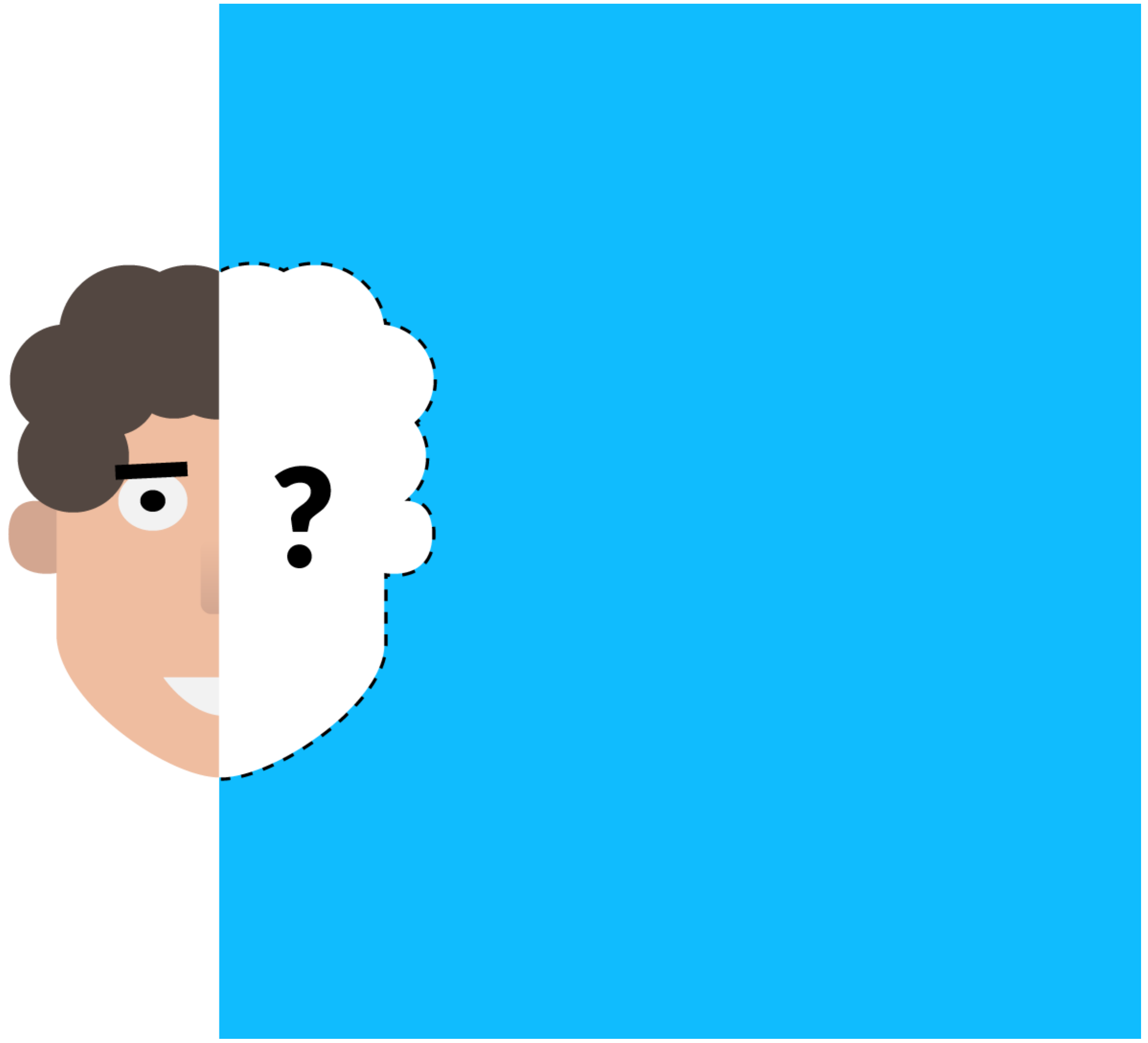


What is differential privacy (DP)?



What is differential privacy (DP)?

Differential privacy is a mathematical definition of privacy.

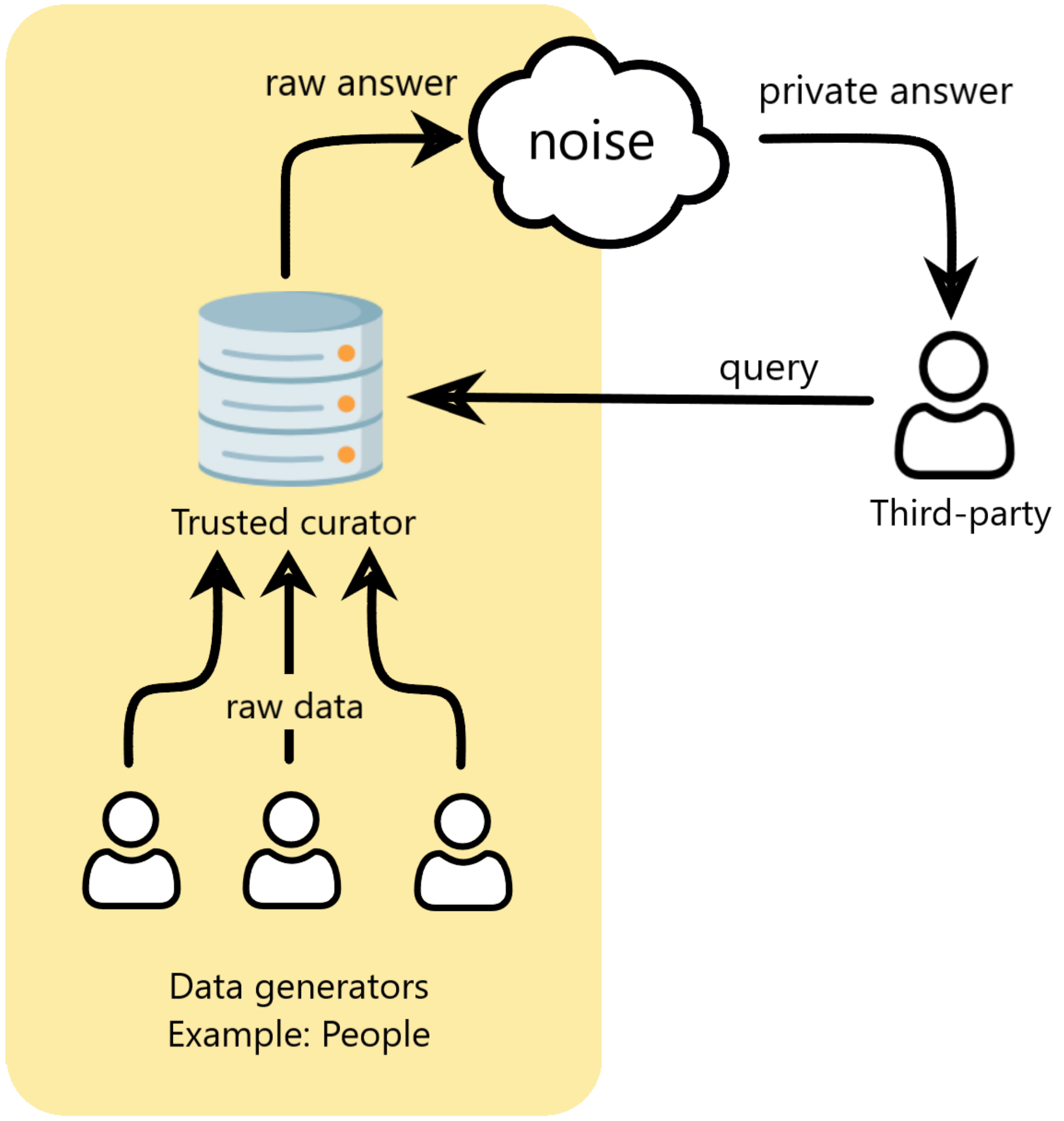


Who uses differential privacy (DP)?

Who uses differential privacy (DP)?

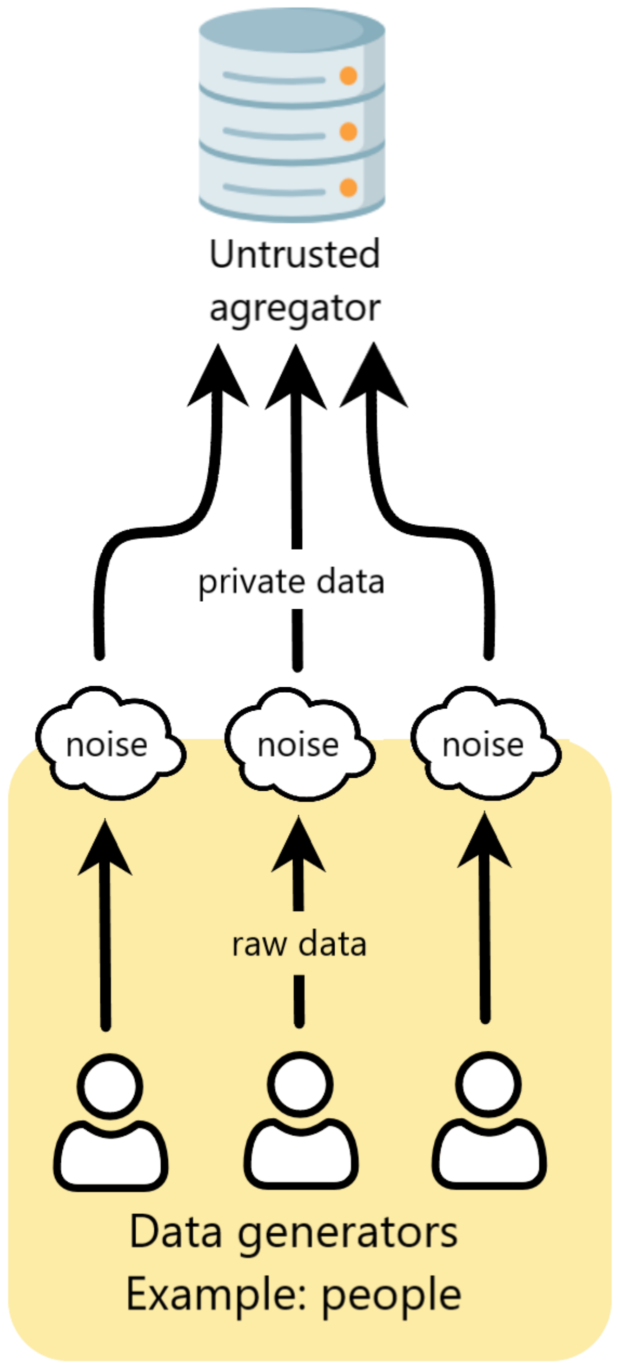
Global differential privacy

- Trusted curator protects data
- Noise added to the output



Local differential privacy

- No trusted party.
- Adding noise before sharing it.



Epsilon-differential privacy

Greek letter epsilon ϵ: How private and how noise a data release is.

- **Higher values** of ϵ indicate **more accurate and less private data**
- **Low-**ϵ systems give **highly random data**

Epsilon is exponential

For example ϵ = 1.

$$\epsilon^1 = 2.72$$

- It's almost three times more private than ϵ = 2.
  - $\epsilon^2 = 7.39$
- And over 8,000 times more private than ϵ = 10.
  - $\epsilon^10 = 22000$

K-anonymity and differential privacy

k-anonymity provides "syntactic" guarantees

- Still widely used
- Not sufficient in many cases

Differential privacy is the current de-facto privacy model

- Preferred by companies: Apple, Uber, Google
- Privacy degradation of releases can be exactly quantified

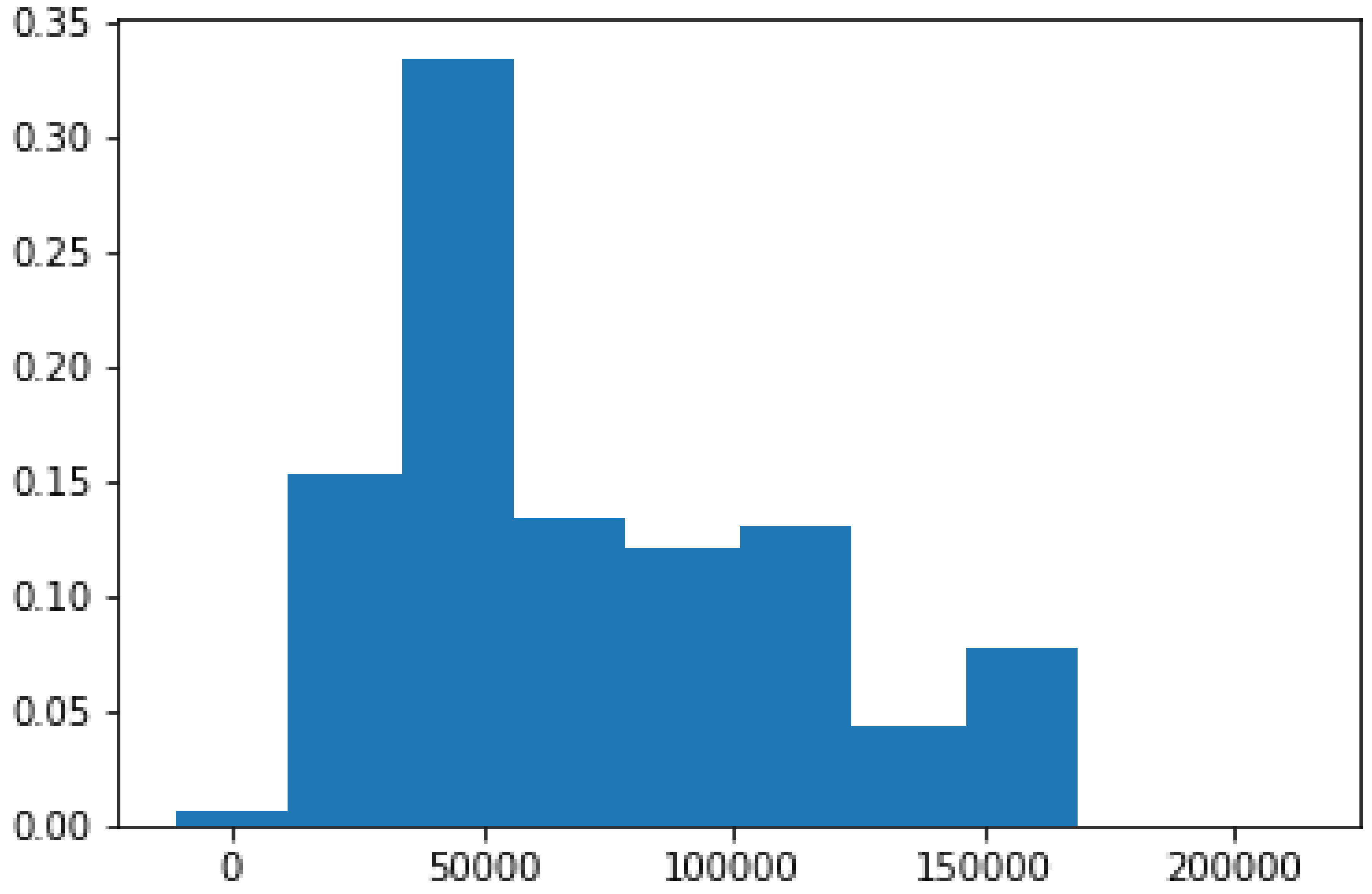
Introduction to diffprivlib

diffprivlib v0.3 from IBM

Histograms

```
# Get counts and bars for non-private histogram of salaries
counts, bins = np.histogram(salaries)

# Normalize counts to get proportions of the height
proportions = counts / counts.sum()

# Draw the histogram of proportions
plt.bar(bins[:-1], height=proportions, width=(bins[1] - bins[0]))
plt.show()
```

Histograms

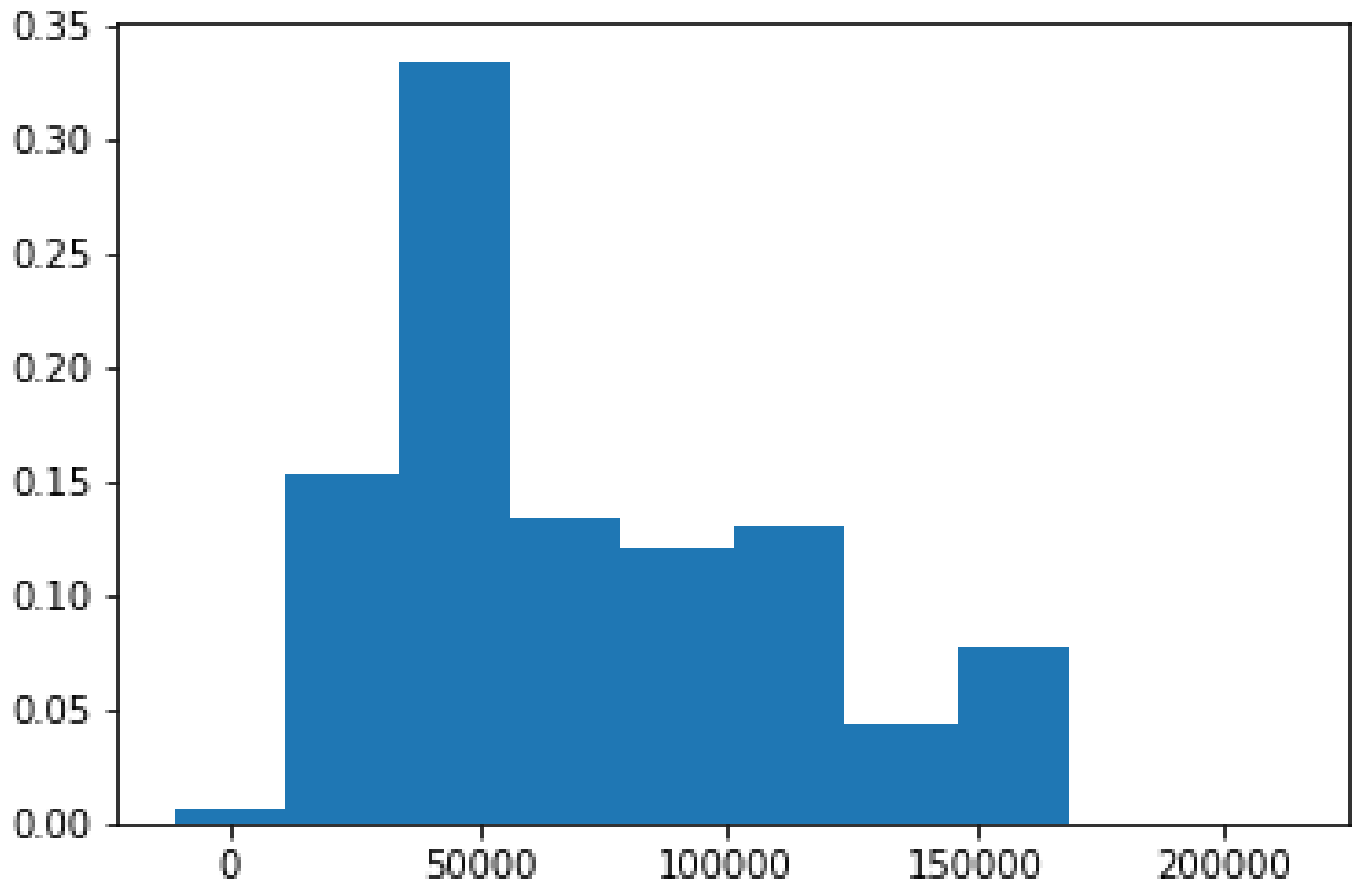


Private histogram

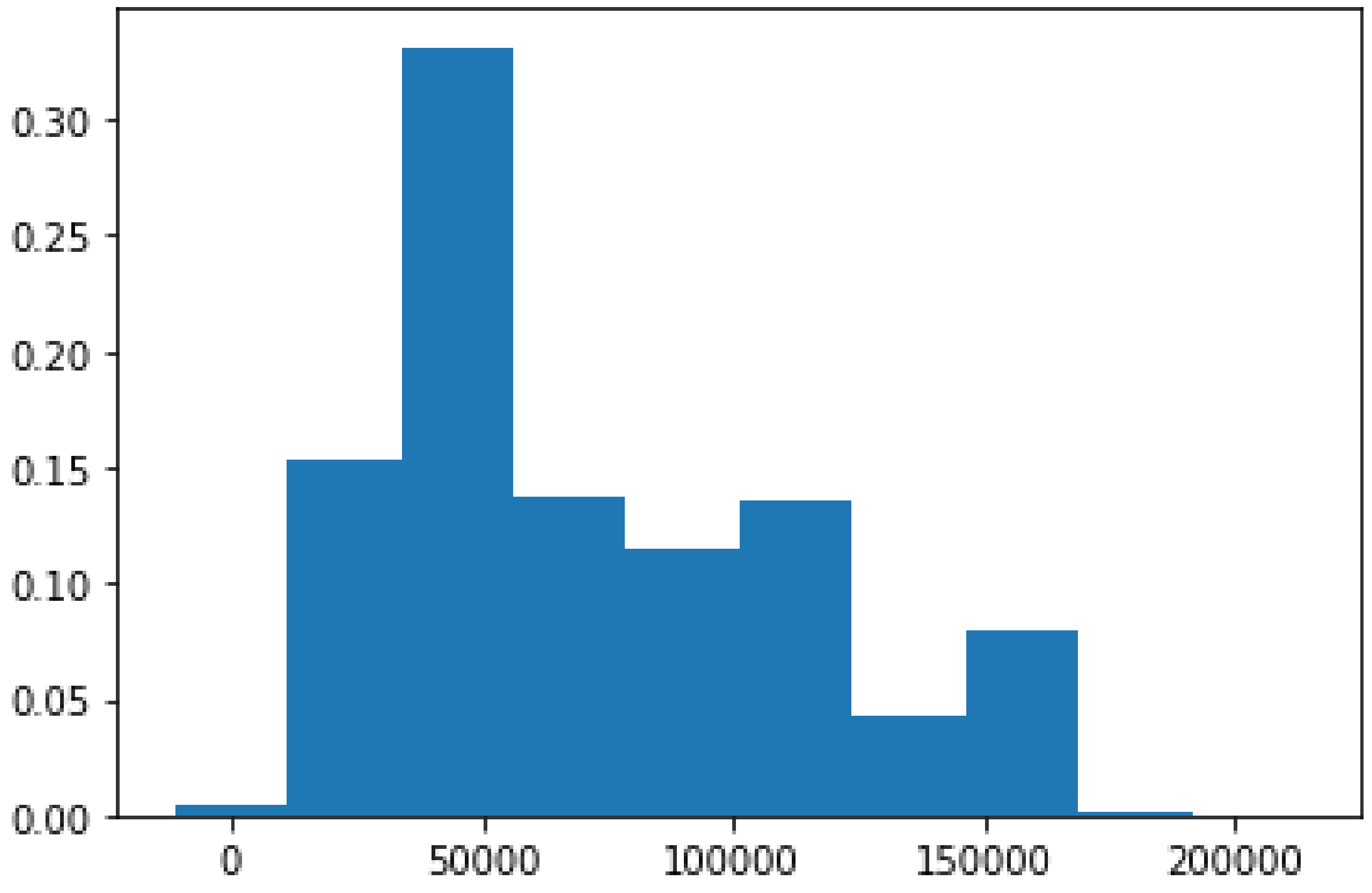
```python
import diffprivlib.tools

# Get counts and bars for private histogram of salaries with epsilon of 0.1
dp_counts, dp_bins = tools.histogram(salaries, epsilon=0.1)

# Normalize counts to get proportions
dp_proportions = dp_counts / dp_counts.sum()

# Draw the histogram of proportions and see differences
plt.bar(dp_bins[:-1], dp_proportions, width=(dp_bins[1] - dp_bins[0]))
plt.show()
```

Private histogram

Non-private histogram



Resulting private histogram



Let's practice!

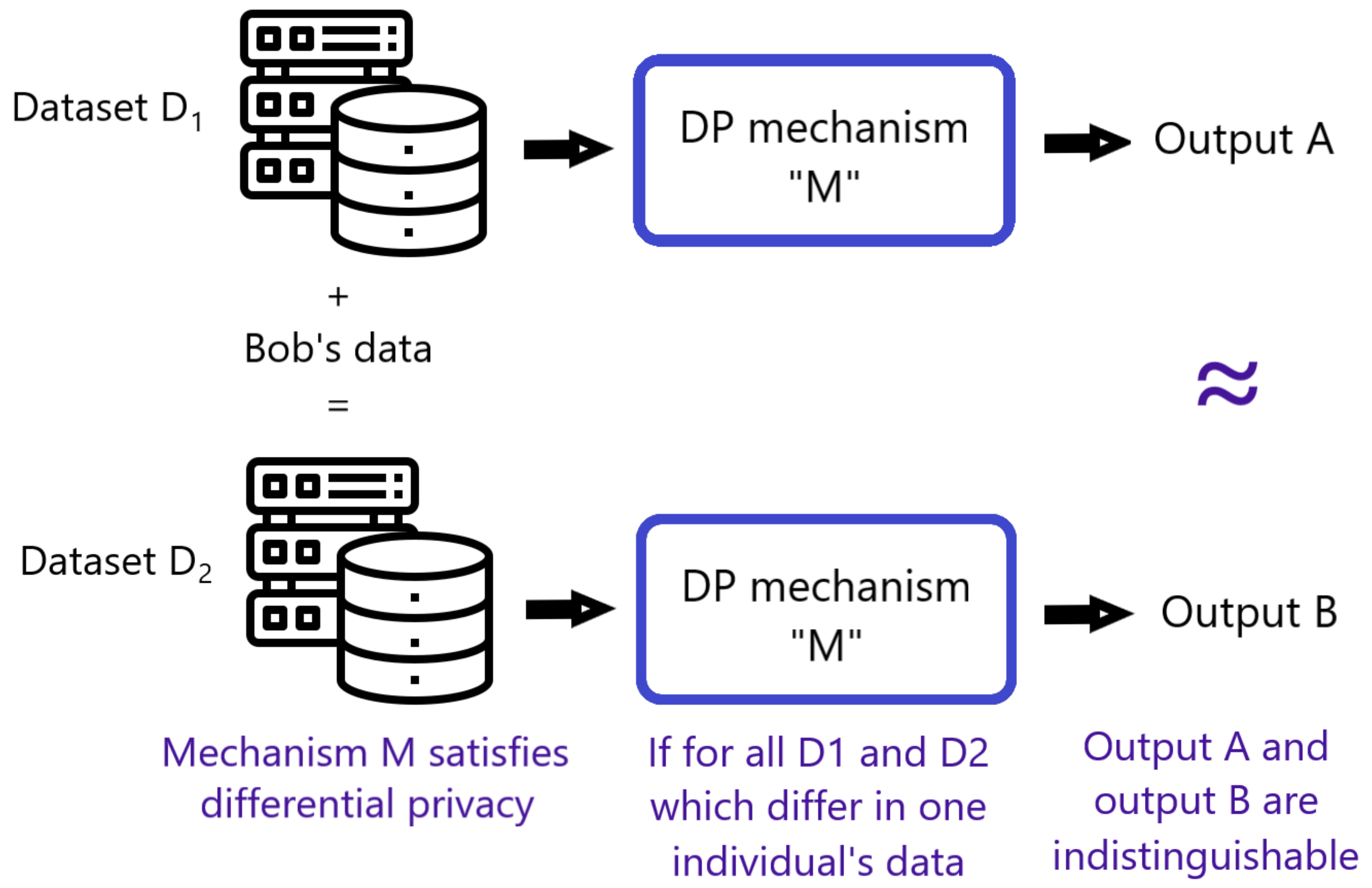
Privacy budgets

DATA PRIVACY AND ANONYMIZATION IN PYTHON

Rebeca Gonzalez
Instructor

Definition of differential privacy

- Cynthia Dwork presents differential privacy with a mathematical definition.

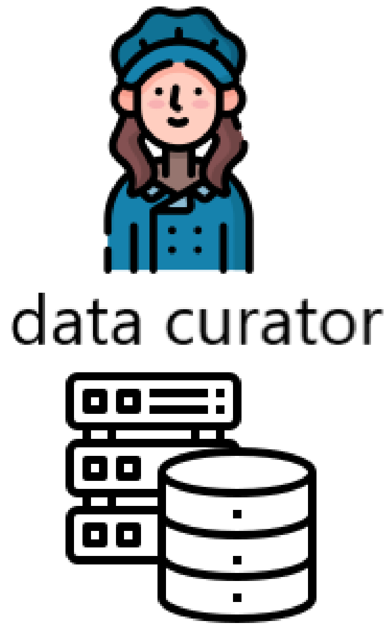

Dataset D_1
DP mechanism "M"
Output A
+
Bob's data
=
≈
Dataset D_2
DP mechanism "M"
Output B
Mechanism M satisfies differential privacy
If for all D1 and D2 which differ in one individual's data
Output A and output B are indistinguishable

- Epsilon and accuracy are the most important quantities.

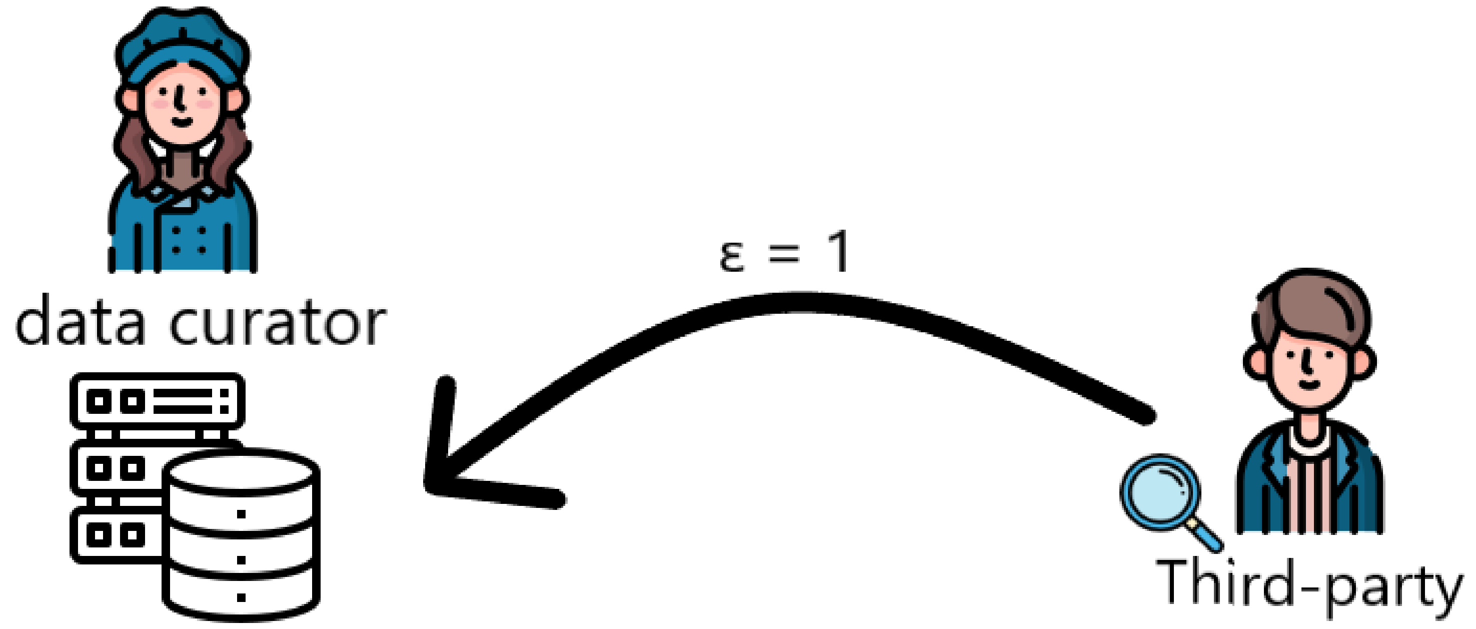
ϵ the privacy parameter

- A metric of privacy loss
- The smaller the value, the better privacy protection

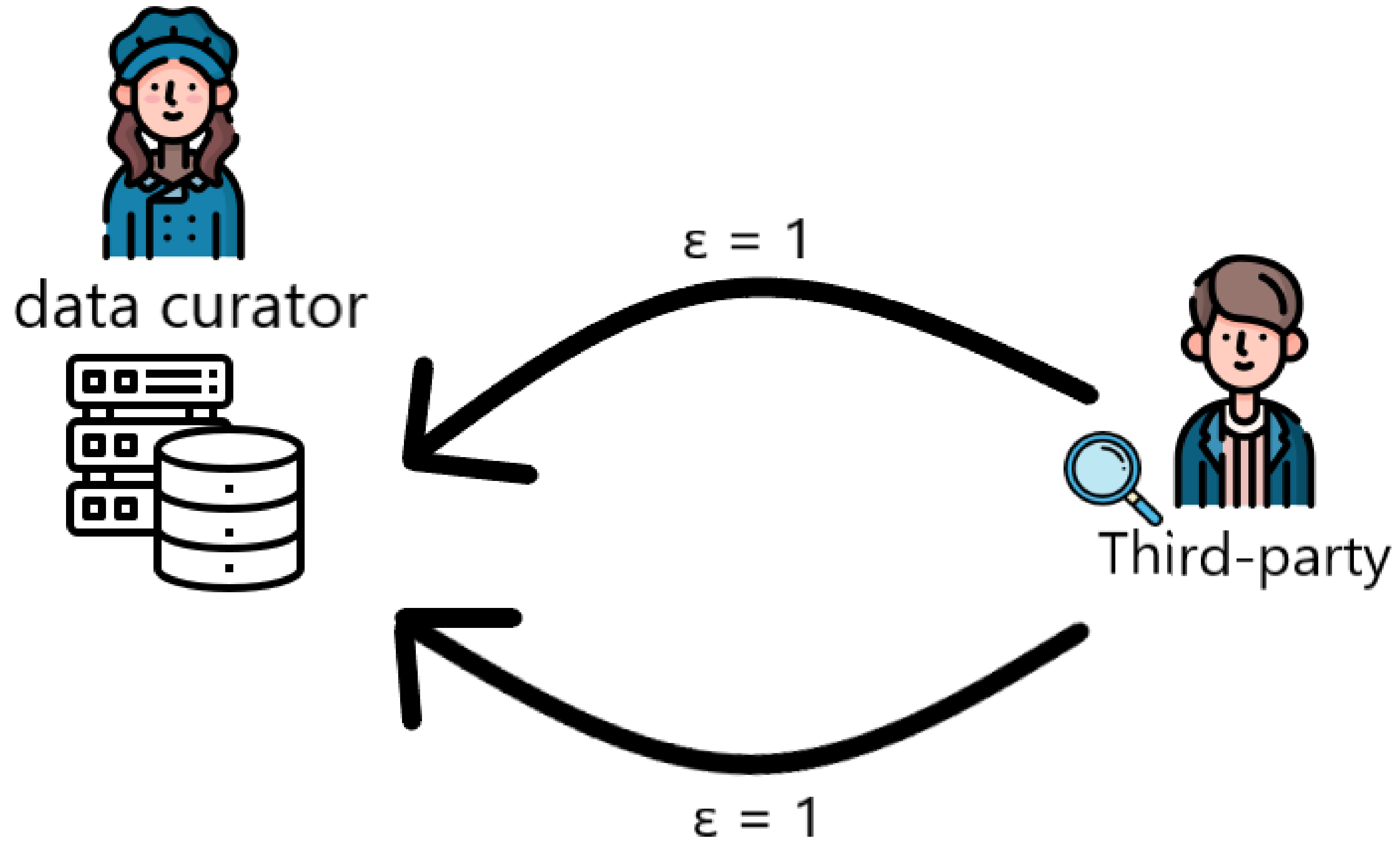
Privacy budget



Privacy budget

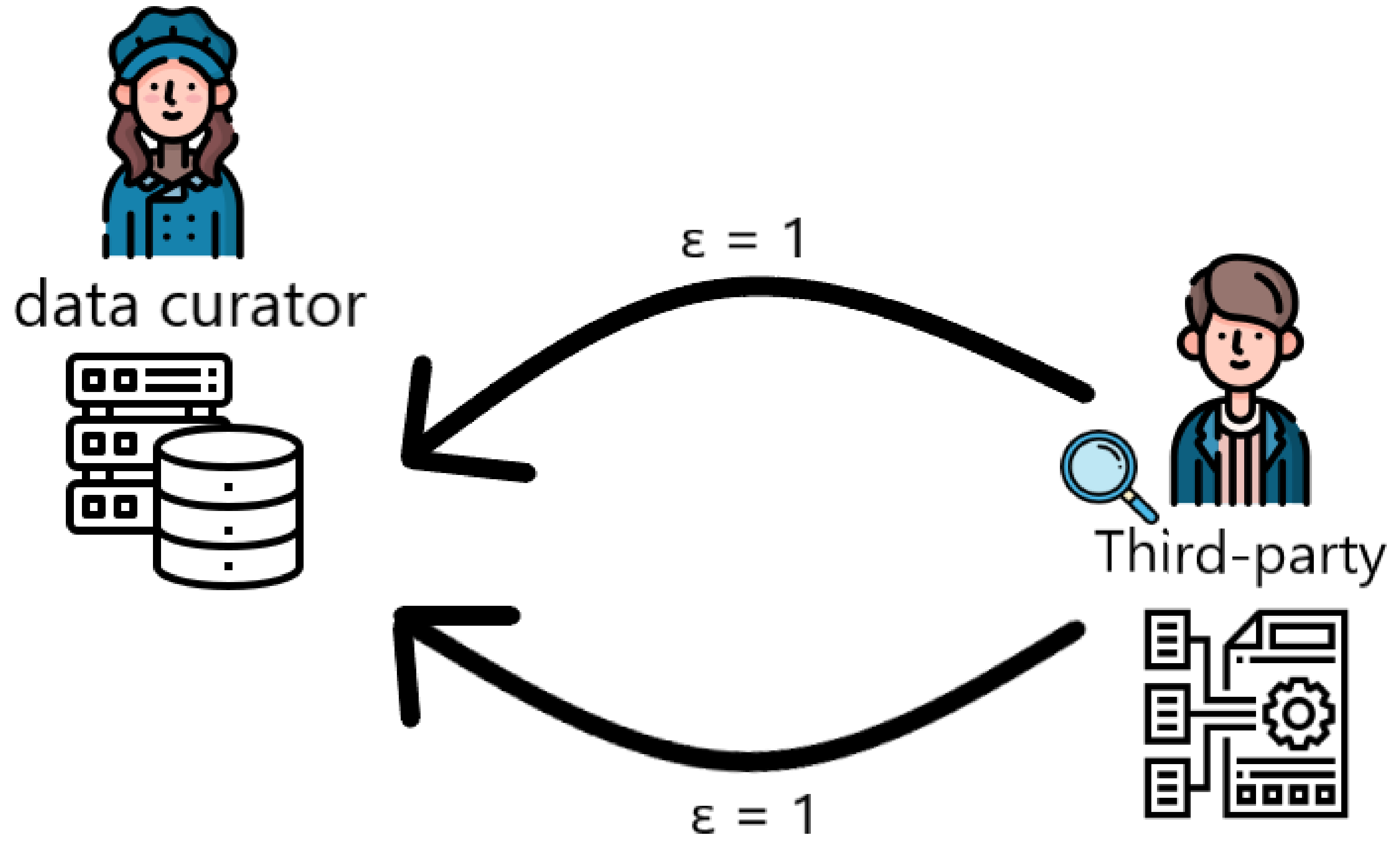


Privacy budget



Making the same private query with $\epsilon = 1$ twice, it's like making a query with privacy $\epsilon = 2$

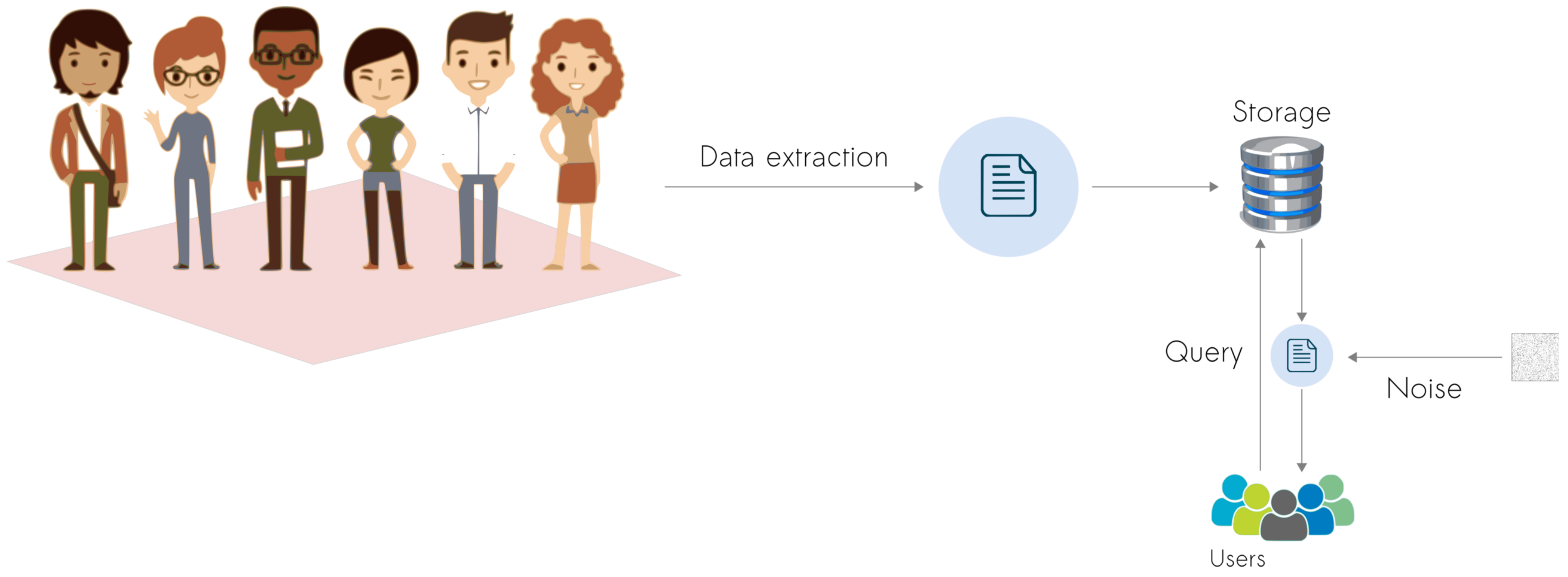
Privacy budget



Third-parties can average answers together, filtering out the noise.

Privacy budget

- Limit on the privacy loss that any individual or group is allowed to accrue
- Track the queries to the data



What's private enough?

- Its goodness depends on the query as well as the data
- Wide range of possible values for epsilon

What's private enough?

Epsilon ϵ

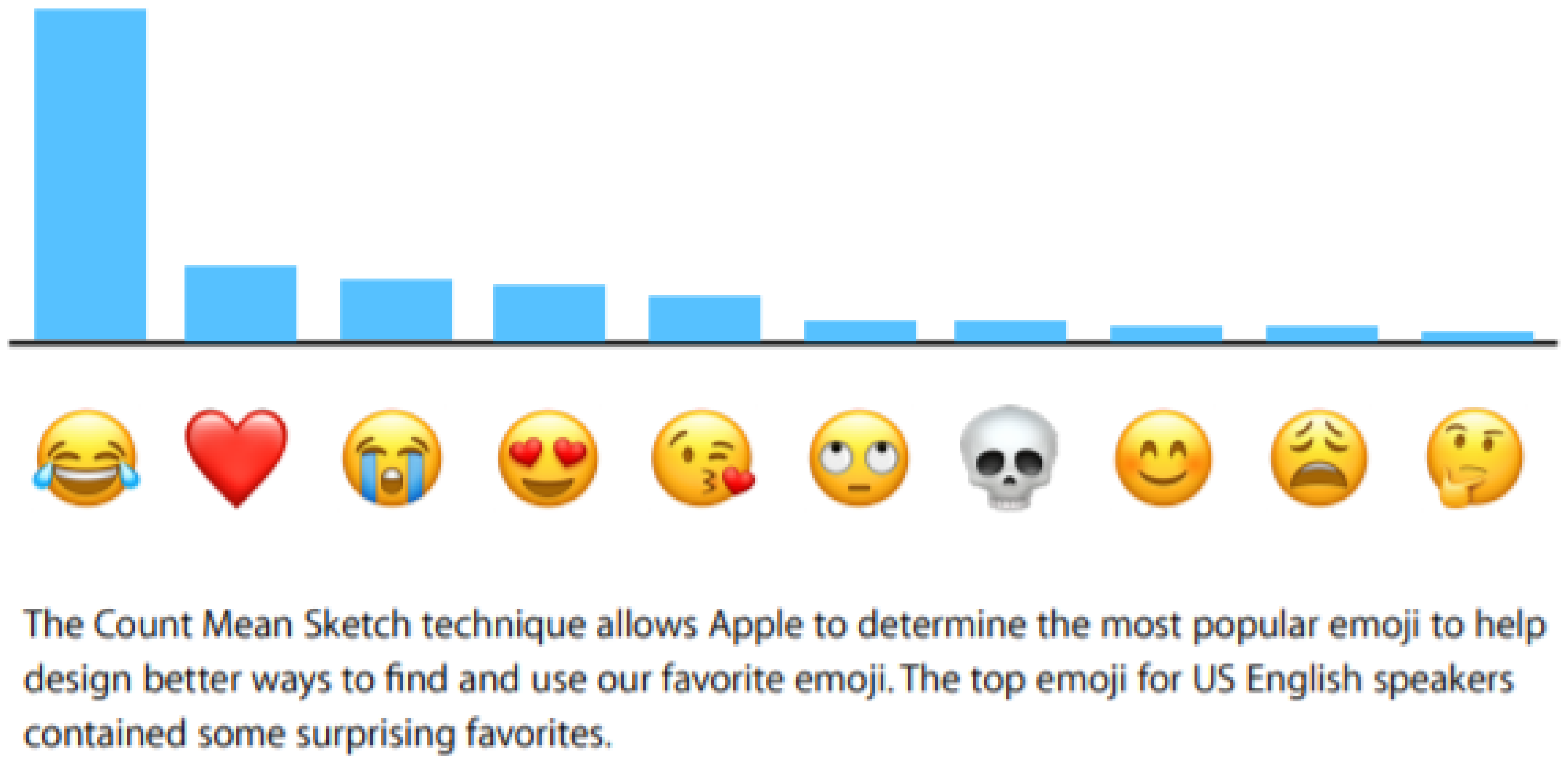
- Values between 0 and 1 are considered very good
- Values above 10 are not good
- Values between 1 and 10 are "better than nothing"

Remember that epsilon is exponential.

- A system with $\epsilon = 1$ is over 8,000 times more private than $\epsilon = 10$.

What's private enough?

- Apple was allegedly using privacy budgets of epsilon = 14 per day in 2017.
- **For emoji suggestions, Apple uses a privacy budget with epsilon of 4, and submits one contribution per day.**



The Count Mean Sketch technique allows Apple to determine the most popular emoji to help design better ways to find and use our favorite emoji. The top emoji for US English speakers contained some surprising favorites.

[1] Screenshot of the top emoji for US English speakers according to data collected by Apple.

Privacy budget: how to track it

```python
from diffprivlib import BudgetAccountant


acc = BudgetAccountant(epsilon=5)
acc
```

```
BudgetAccountant(epsilon=5)
```

Privacy budget: how to track it

```python
# Compute a private mean of the salaries using epsilon of 0.5
# Use the Budget Accountant acc and set bounds to be from 0 to 230000
dp_mean = tools.mean(salaries, epsilon=0.5, accountant=acc, bounds=(0, 230000))

# Print the resulting private mean
print("Private mean: ", dp_mean)
```

```
Private mean: 82524.72611901595
```

Privacy budget: how to track it

```python
# Total privacy spent
print("Total spent: ", acc.total())

# Privacy budget remaining
print("Remaining budget: ", acc.remaining())

# Total number of queries done so far
print("Number of queries recorded: ", len(acc))
```

```
Total spent: (epsilon=0.5, delta=0.0)
Remaining budget: (epsilon=4.5, delta=1.0)
Number of queries recorded: 1
```

Privacy budget: how to track it

```python
# Privacy budget remaining for 2 queries
print("Remaining budget for 2 queries: ", acc.remaining(2))
```

```
Remaining budget for 2 queries: (epsilon=2.25, delta=1.0)
```

Let's practice!

DATA PRIVACY AND ANONYMIZATION IN PYTHON

Differentially private machine learning models

DATA PRIVACY AND ANONYMIZATION IN PYTHON

Rebeca Gonzalez
Data Engineer

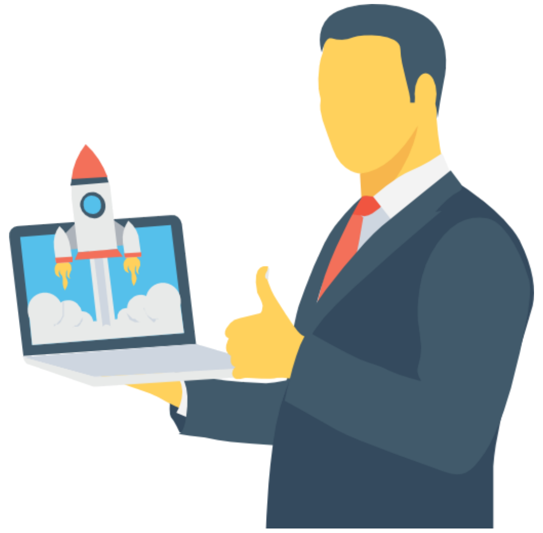
Sharing data safely

Companies with similar data, sharing information to improve products and services.

- Including Machine Learning (ML) models

Differently private machine learning models

- SaaS company that has multiple online stores as partners.
- When a new partner joins, it might take months before it can collect enough data
- With DP, the SaaS company can encourage partners to share data



Machine learning and privacy

- Datasets contain sensitive information
- Adversaries can exploit the output of machine learning algorithms

Machine learning and privacy

Differentially private machine learning models

- Follow the corresponding data distributions
- Can guarantee privacy to individuals

Differentially private classification models

```python
# Import the scikit-learn naive Bayes classifier
from sklearn.naive_bayes import GaussianNB

# Import the differentially private naive Bayes classifier
from diffprivlib.models import GaussianNB
```

Non-private classifier

```python
from sklearn.naive_bayes import GaussianNB

# Built the non-private classifier
nonprivate_clf = GaussianNB()

# Fit the model to the data
nonprivate_clf.fit(X_train, y_train)

print("The accuracy of the non-private model is ",
      nonprivate_clf.score(X_test, y_test))
```

```
The accuracy of the non-private model is  0.8333333333333334
```

Differentially private classifier

```python
from diffprivlib.models import GaussianNB as dp_GaussianNB

# Build the private classifier with empty constructor
private_clf = dp_GaussianNB()

# Fit the model to the data and see the score
private_clf.fit(X_train, y_train)
print("The accuracy of the private model is ",
      private_clf.score(X_test, y_test))
```

```
The accuracy of the private model is  0.7
PrivacyLeakWarning: Bounds have not been specified and will be calculated
  on the data provided. This will result in additional privacy leakage.
  To ensure differential privacy and no additional privacy leakage, specify bounds for each dimension.
  "privacy leakage, specify bounds for each dimension.", PrivacyLeakWarning)
```

Avoid privacy leakage

To avoid data leakage, we can replace the min and max values by passing a bounds argument. It can be:

- A tuple of the form (min, max)
  - Integers covering the min/max of the entire data
    - Example:

```
(0,100)
```

  - Arrays for min and max values of each column in the data.
    - Example:

```
([0,1,0,2],[10,80,5,70])
```

Avoid privacy leakage

```python
# Set the bounds to cover at least the min and max values
bounds = (X_train.min(axis=0) - 1, X_train.max(axis=0) + 1)

# Built the classifier with epsilon of 0.5
dp_clf = dp_GaussianNB(epsilon=0.5, bounds=bounds)

# Fit the model to the data and see the score
dp_clf.fit(X_train, y_train)
print("The accuracy of the private model is ",
      private_clf.score(X_test, y_test))
```

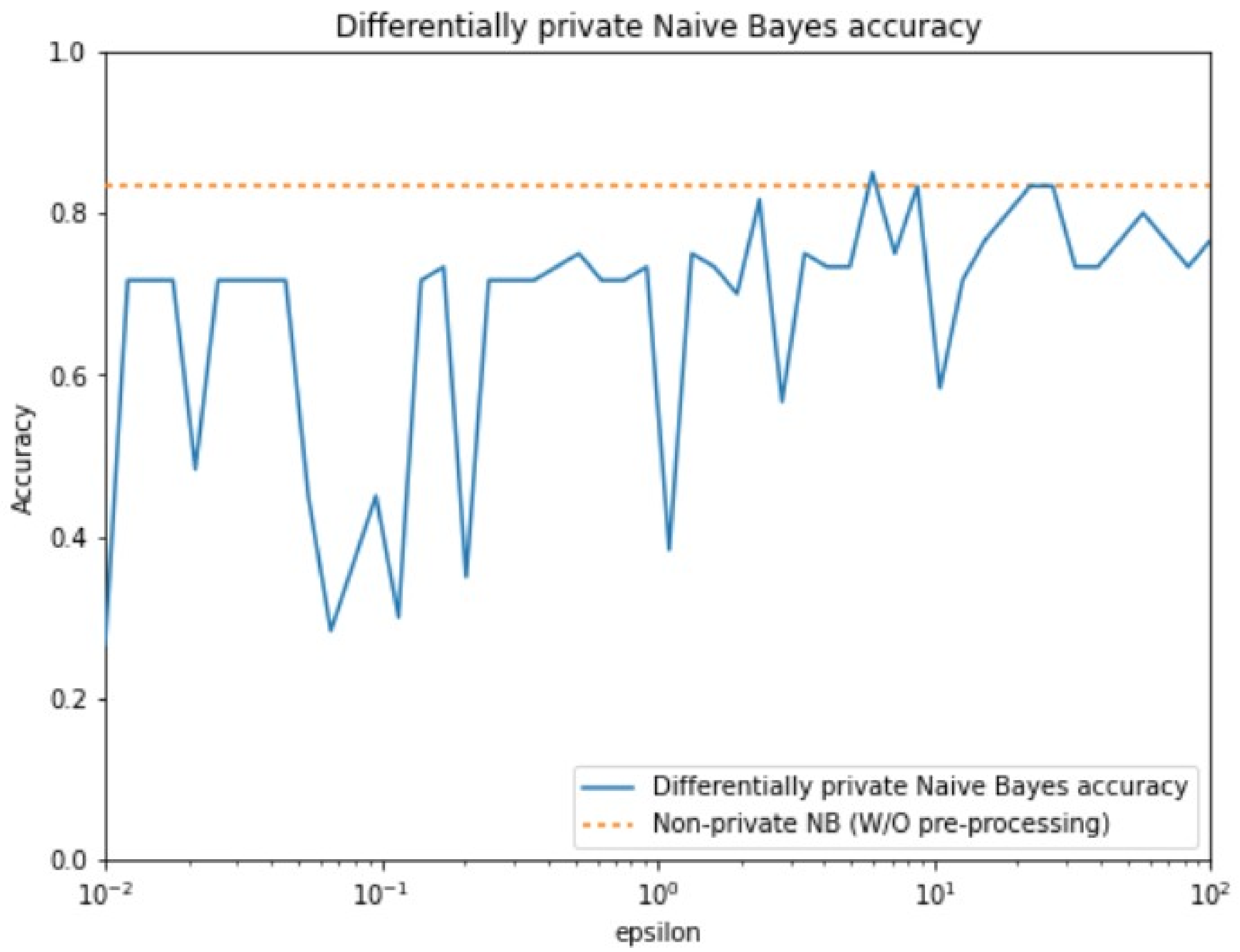
```
The accuracy of the private model is  0.807000
```

More on adding bounds

```python
# Import random module
import random
# Set the min and max of bounds in the data plus some noise
bounds = (X_train.min(axis=0) - random.sample(range(0, 30), 12),
          X_train.max(axis=0) + random.sample(range(0, 30), 12))

# Build the classifier with epsilon of 0.5
dp_clf = dp_GaussianNB(epsilon=0.5, bounds=bounds)

# Fit the model to the data and see the score
dp_clf.fit(X_train, y_train)
print("The accuracy of private classifier with bounds is ", dp_clf.score(X_test, y_test))
```

```
The accuracy of private classifier with bounds is 0.7544444444
```

Different epsilon values



Let's create privacy preserving models!

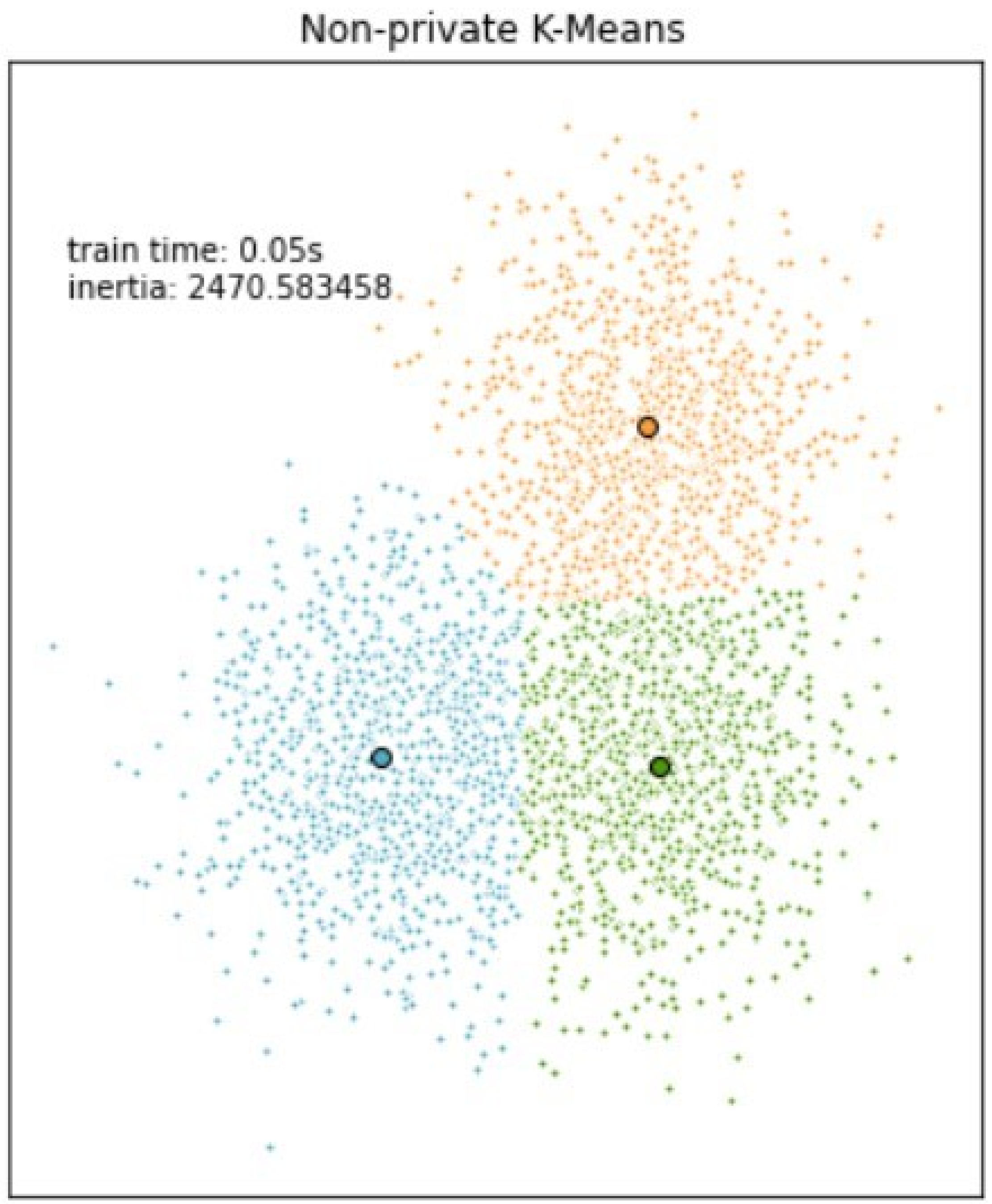
DATA PRIVACY AND ANONYMIZATION IN PYTHON

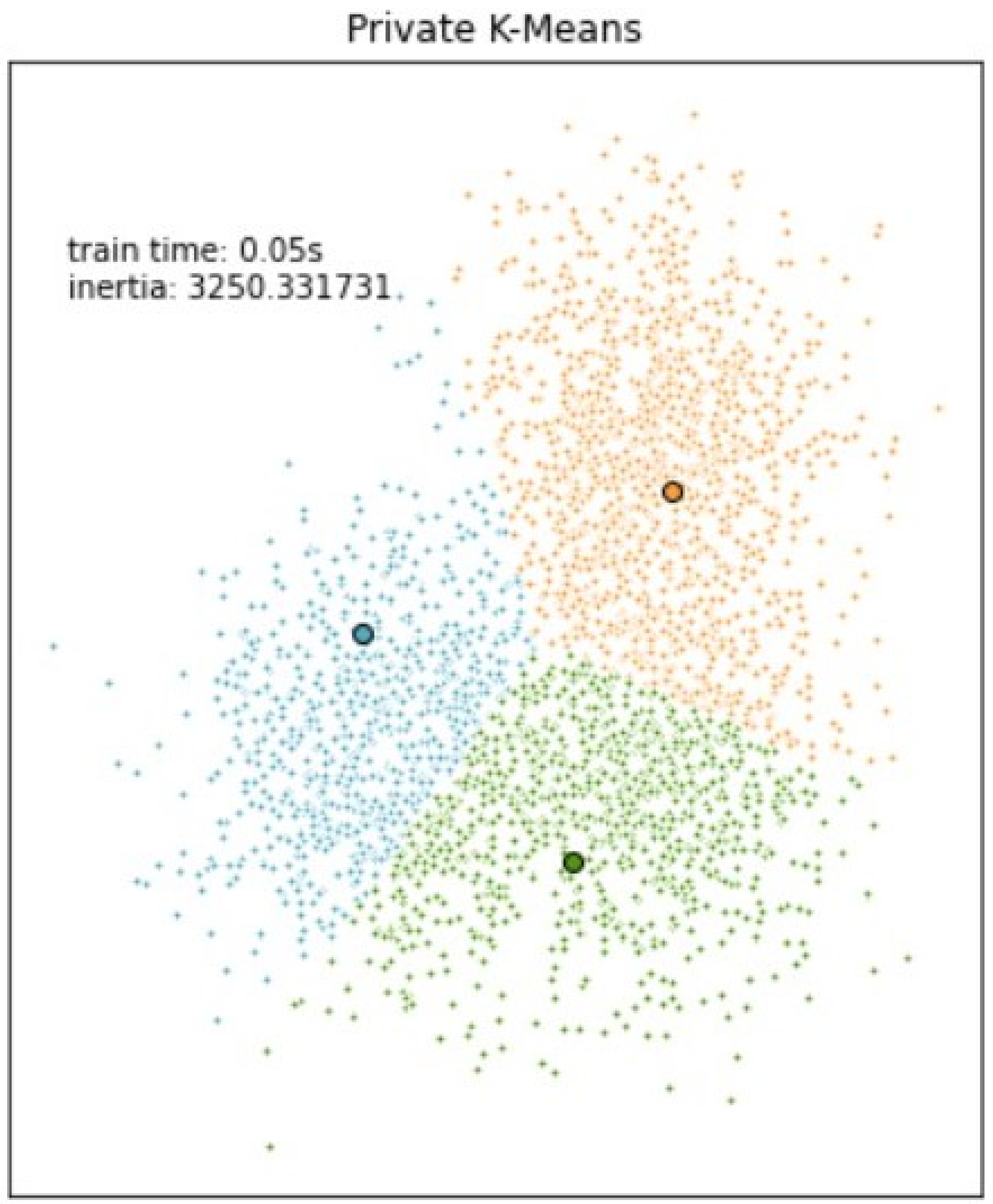
Differentially private clustering models

DATA PRIVACY AND ANONYMIZATION IN PYTHON

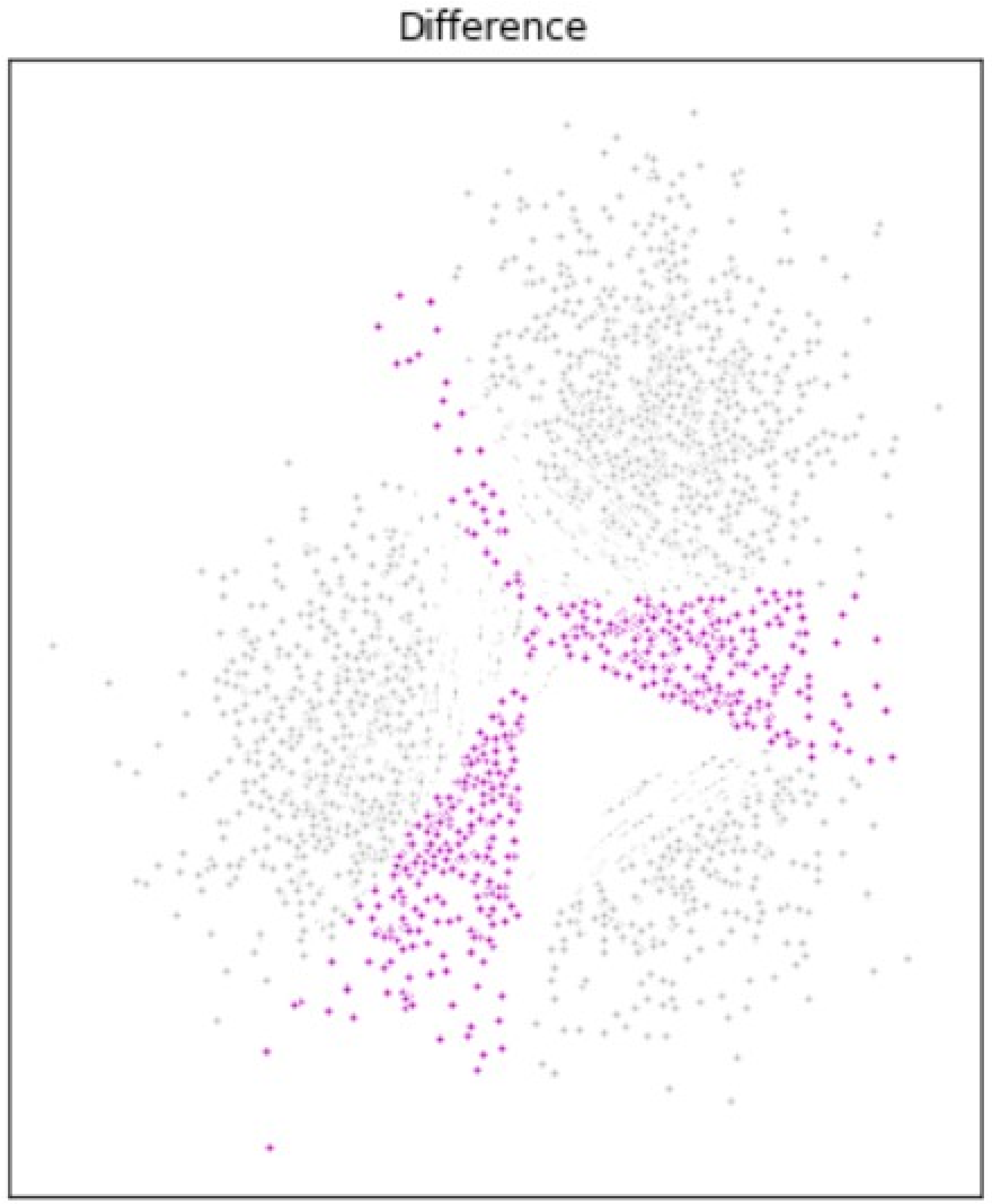
Rebeca Gonzalez
Data engineer

Comparing models





Comparing the models



- Difference between the models resulting clusters
- They have the majority of results in common

Building differently private clustering models

```python
from diffprivlib.models import KMeans

# Computing the clusters with the DP model
model = KMeans(epsilon=1, n_clusters=3)

# Run the model and obtain clusters
clusters = model.fit_predict(X)
```

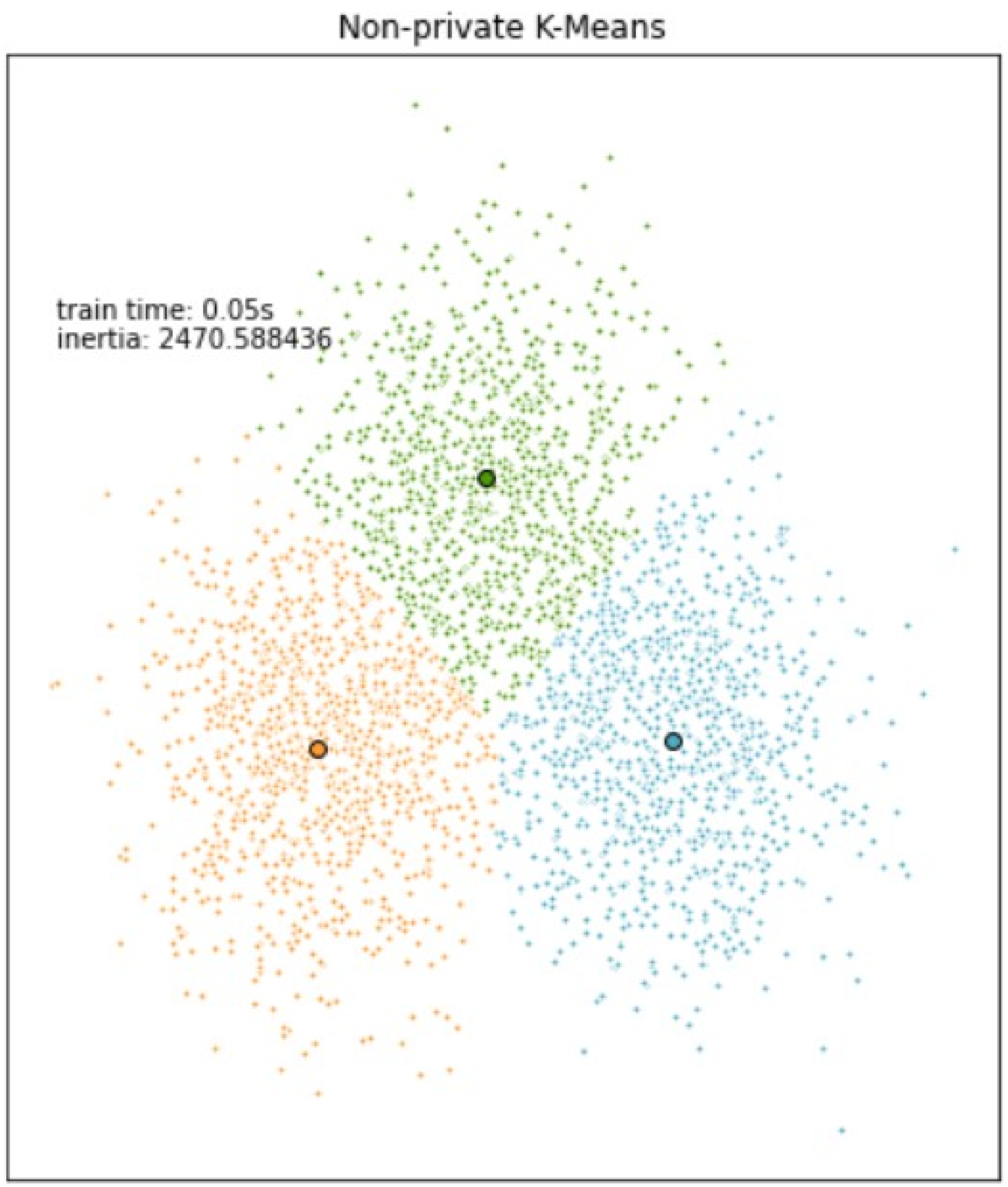
Improving DP clustering models

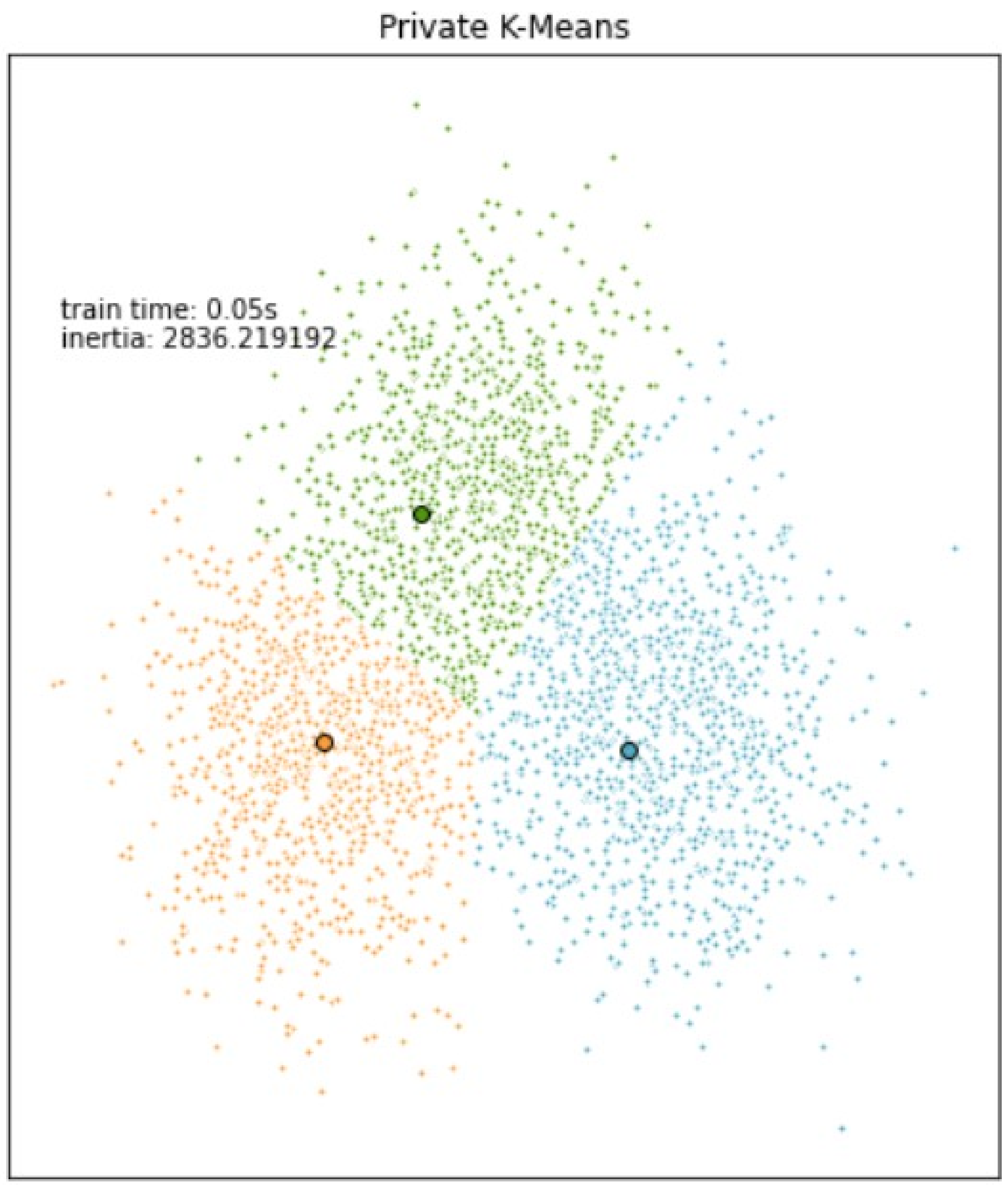
- We can pre-process data before doing clustering.
- Feature scaling such as `StandardScaler` and dimensionality reduction methods like PCA.
  - To reduce the inertia of the model
  - Get more accurate segmentation groups
- We do this with `diffprivlib` just as you would do with `sklearn` models.

Improving DP clustering models

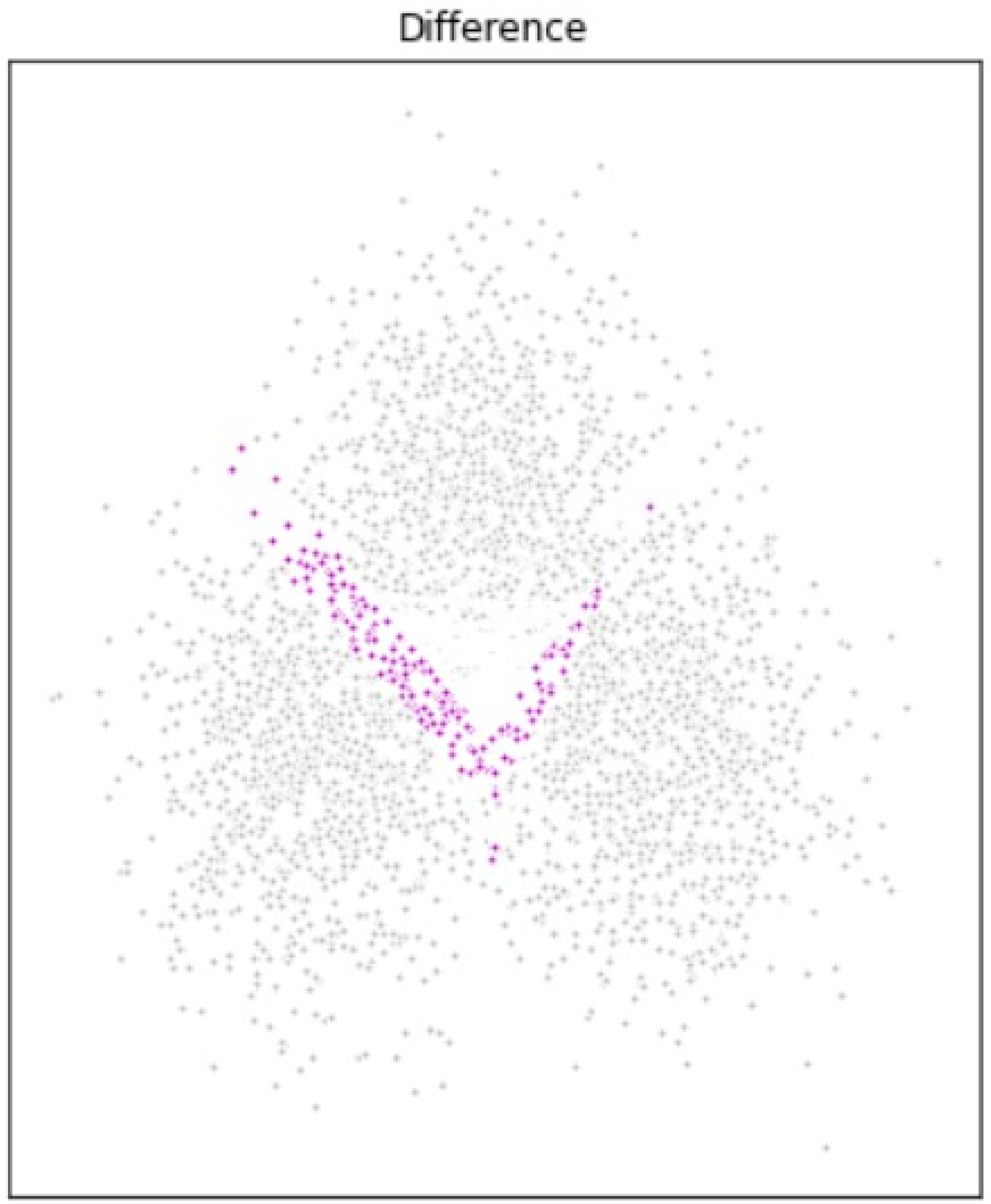
```
from sklearn.decomposition import PCA

# Initialize PCA
pca = PCA()

# Fit transform data with PCA
X = pca.fit_transform(X)

# Computing the clusters with the DP model
model = dp_Kmeans(epsilon=1, n_clusters=3)

# Run the model and obtain clusters
clusters = model.fit_predict(X)
```

Improving DP clustering models



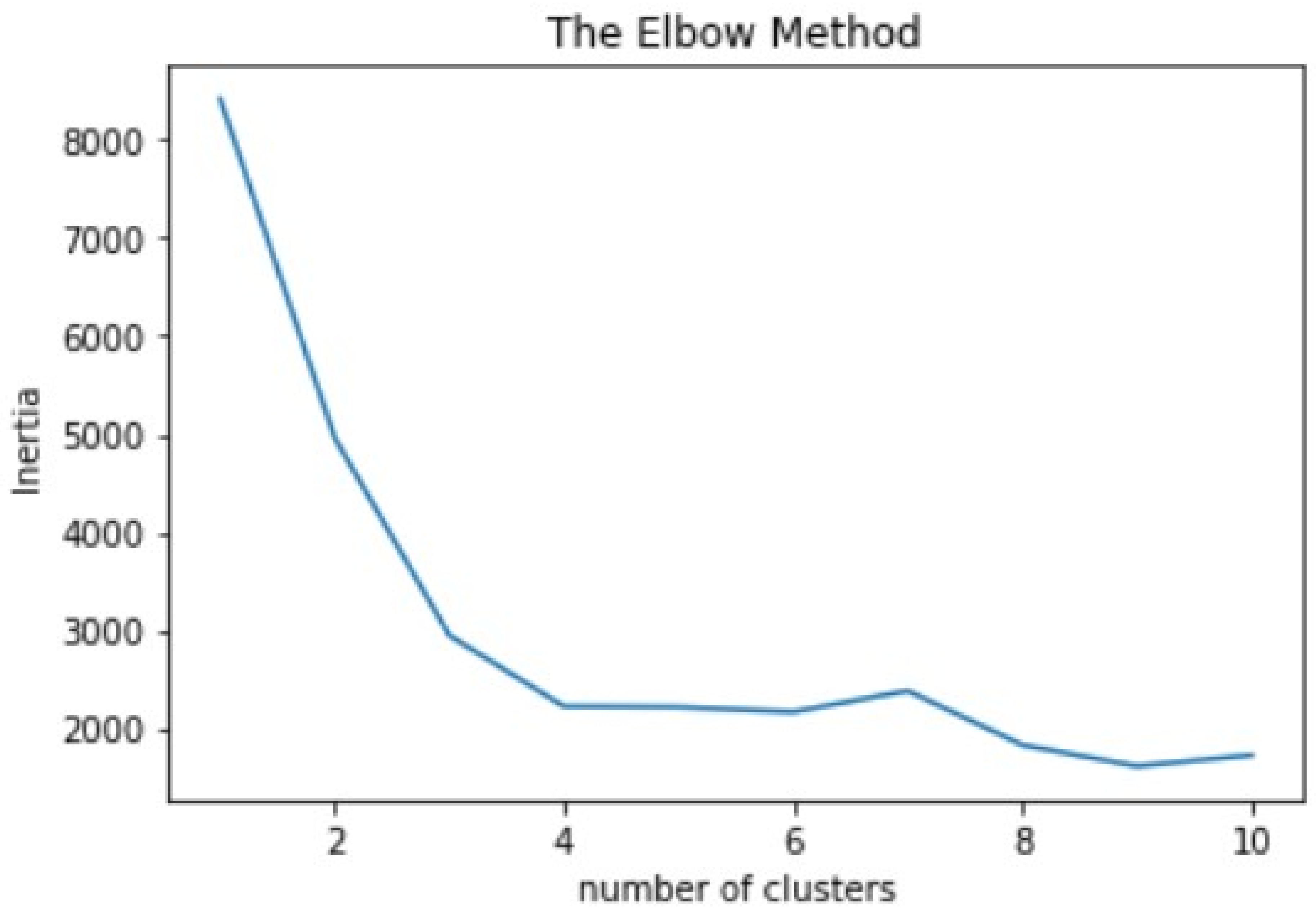


Improving DP clustering models



- Improved results by using data transformations

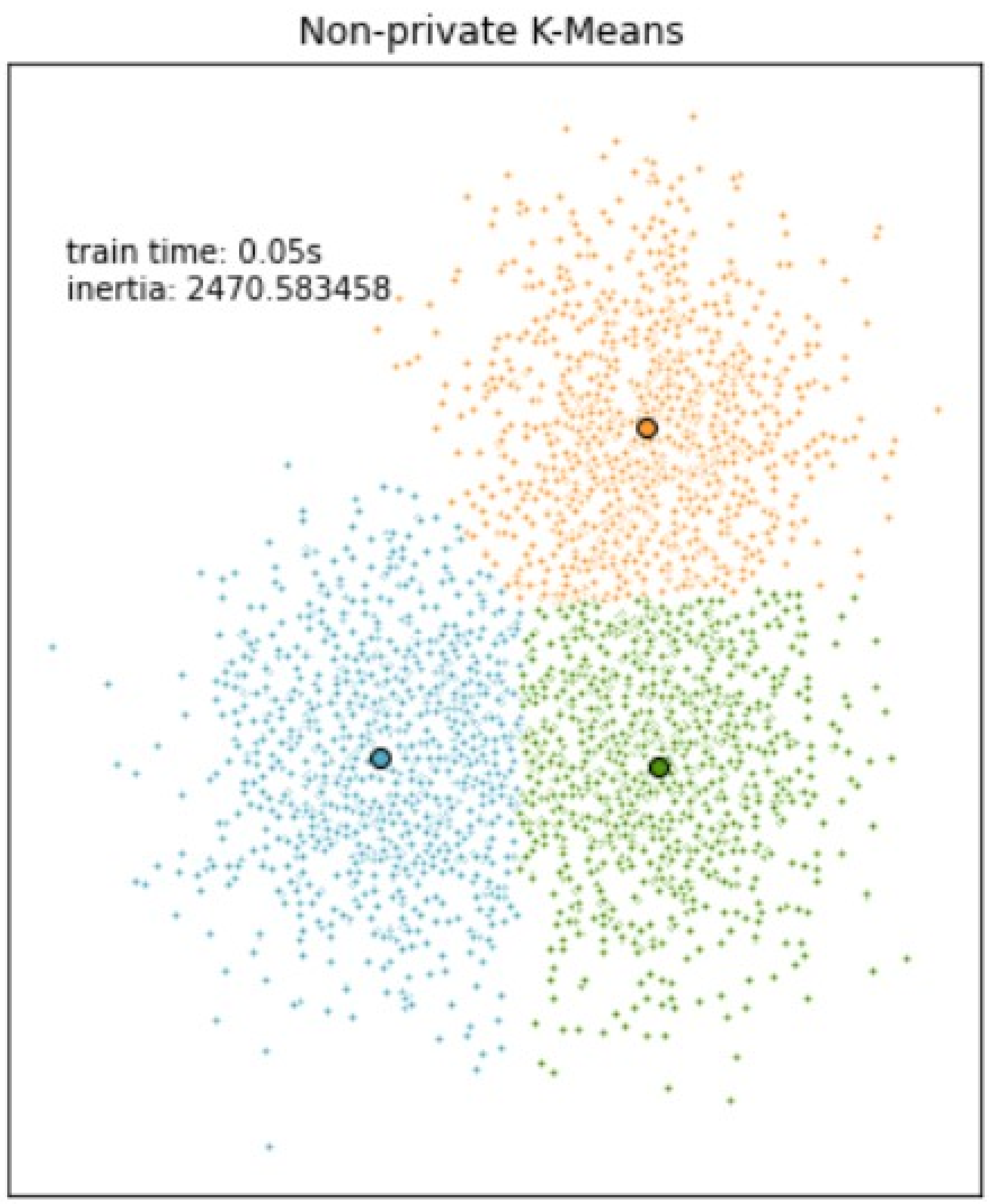
Elbow method

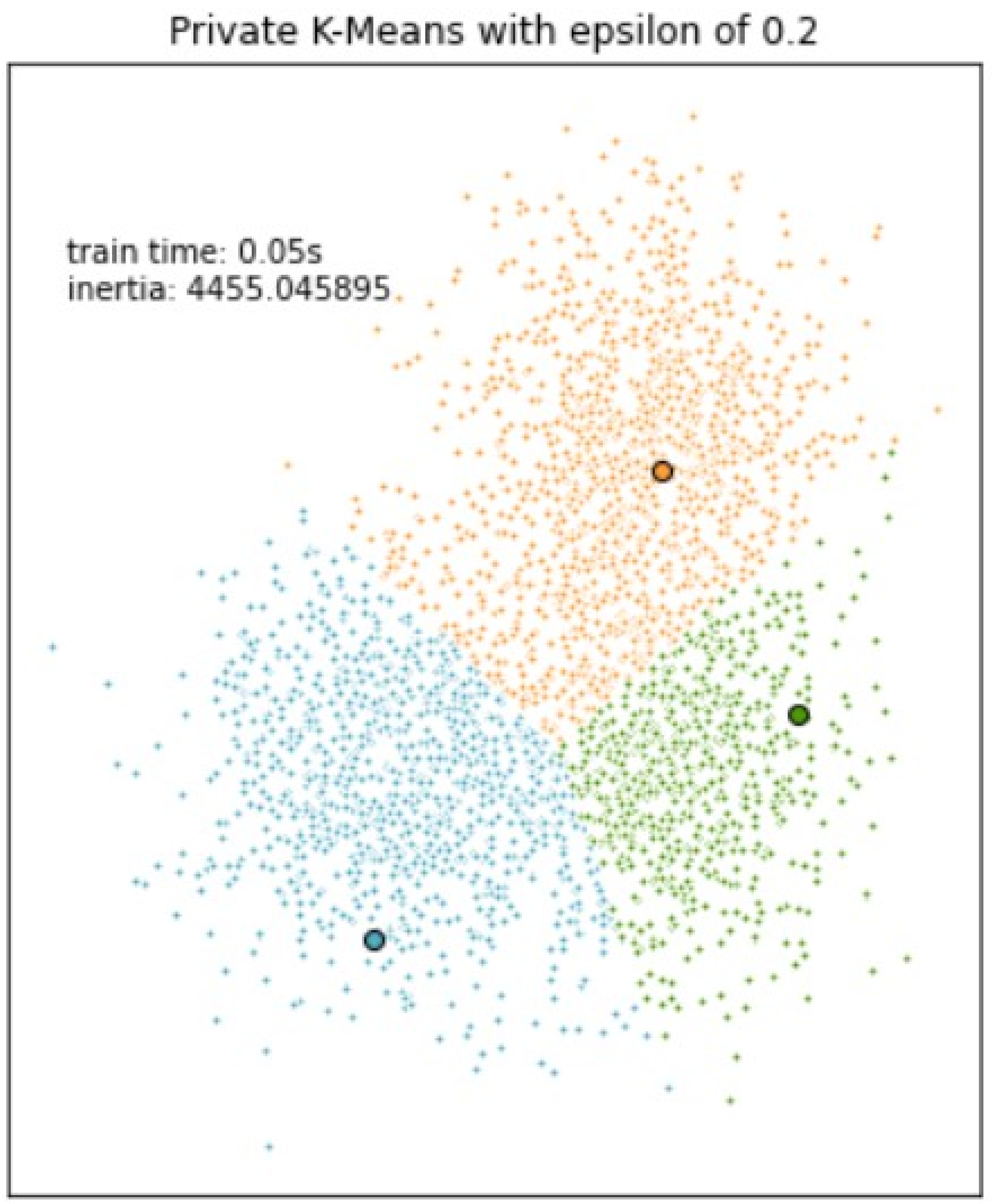


Epsilon

```python
from diffprivlib.models import KMeans as model

# Computing the clusters with the DP model
model = dp_Kmeans(epsilon=0.2, n_clusters=3)

# Run the model and obtain clusters
clusters = model.fit_predict(X)
```

Epsilon


Non-private K-Means
train time: 0.05s
inertia: 2470.583458


Private K-Means with epsilon of 0.2
train time: 0.05s
inertia: 4455.045895

Let's practice!

DATA PRIVACY AND ANONYMIZATION IN PYTHON