PrivApprox

Privacy-Preserving Stream Analytics

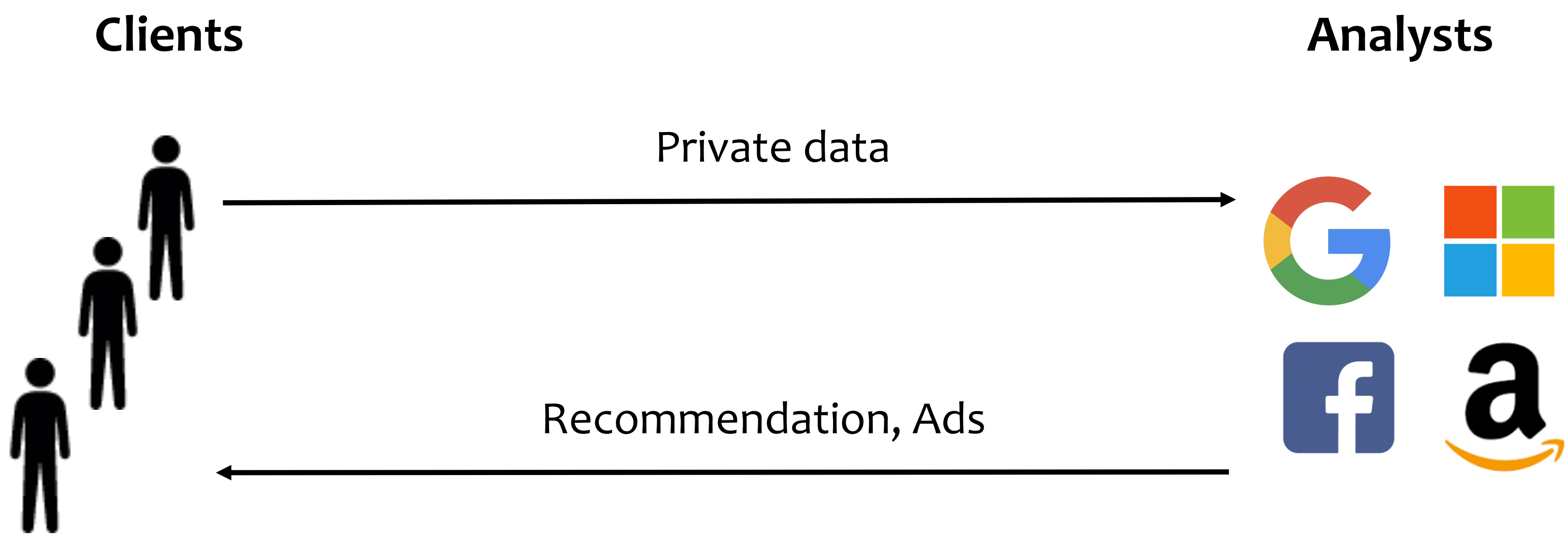
https://privapprox.github.io

Do Le Quoc, Martin Beck,
Pramod Bhatotia, Ruichuan Chen, Christof Fetzer, Thorsten Strufe

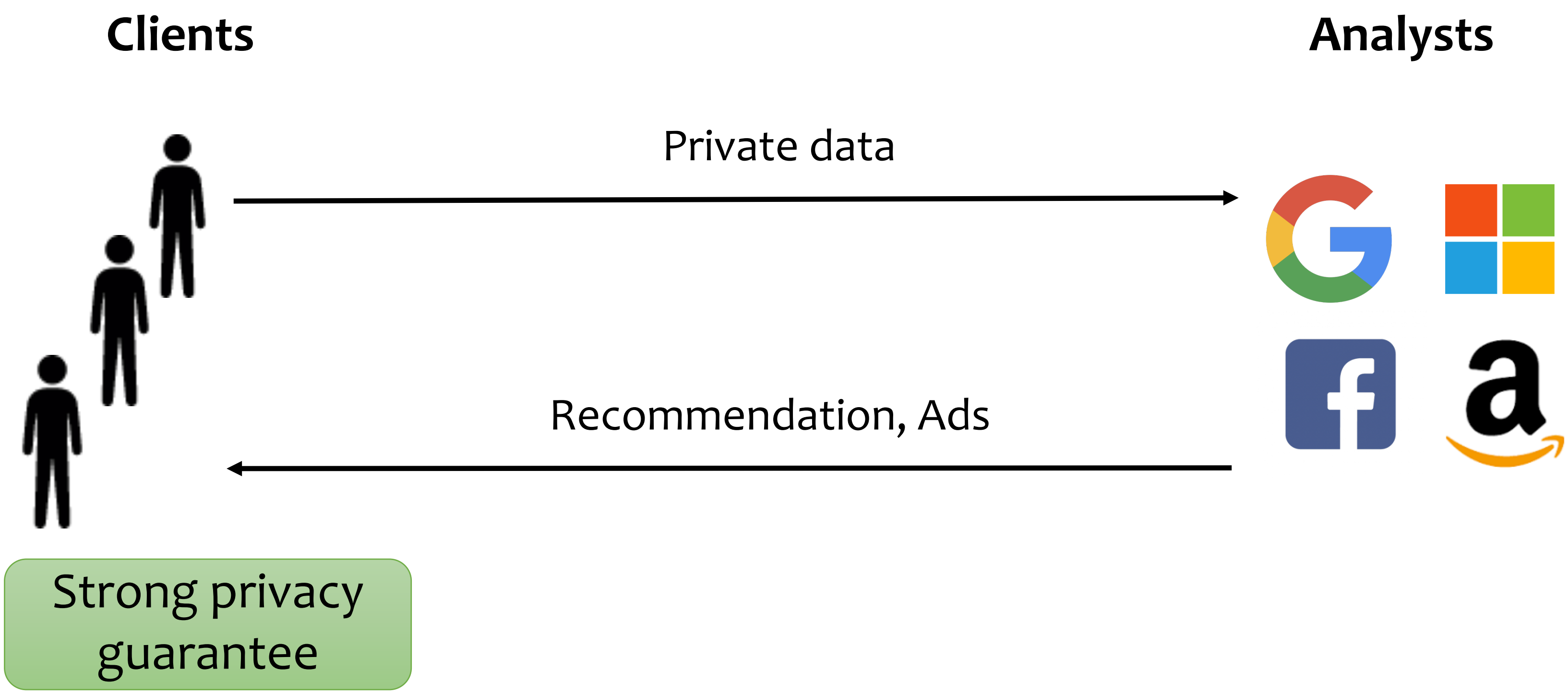
TECHNISCHE UNIVERSITÄT DRESDEN
THE UNIVERSITY of EDINBURGH
NOKIA Bell Labs

July 2017

Motivation

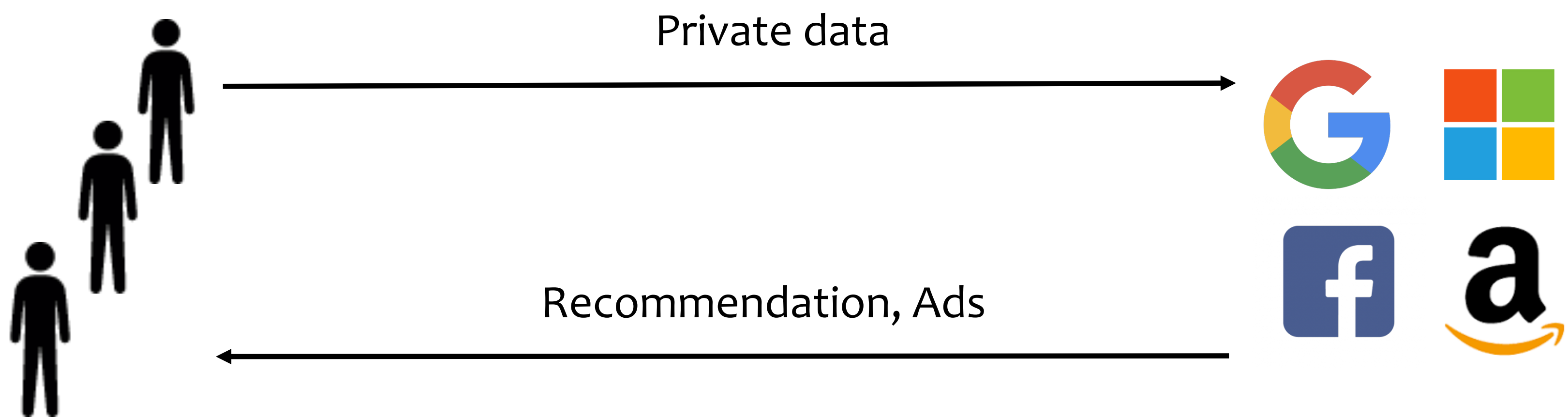


Motivation


Clients
Analysts
Private data
Recommendation, Ads
Strong privacy guarantee

Motivation

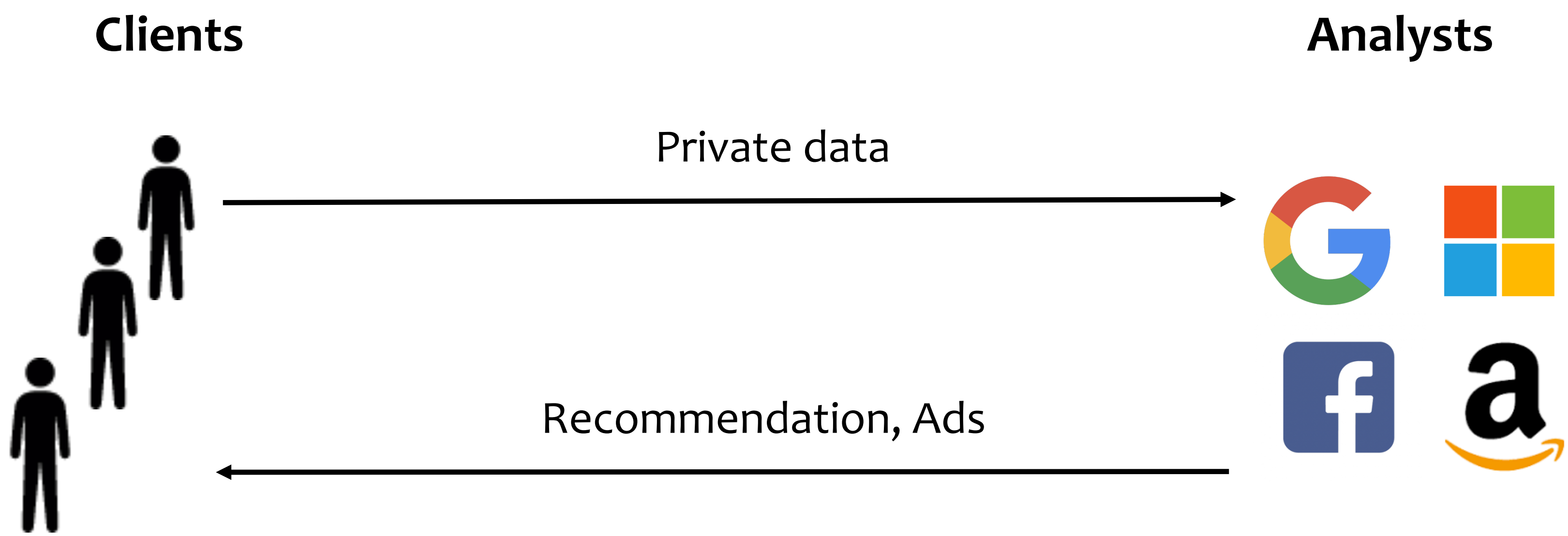
Clients **Analysts**



Strong privacy guarantee

High utility analytics in real-time

Motivation



Strong privacy guarantee

High utility analytics in real-time

How to preserve users' privacy while supporting high-utility data analytics for low-latency stream processing?

State-of-the-art systems

Clients

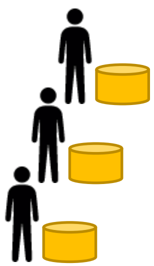
State-of-the-art systems

Clients

Personal data should be stored locally under the clients' control

State-of-the-art systems

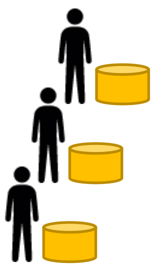
Clients



Personal data should be stored locally
under the clients' control

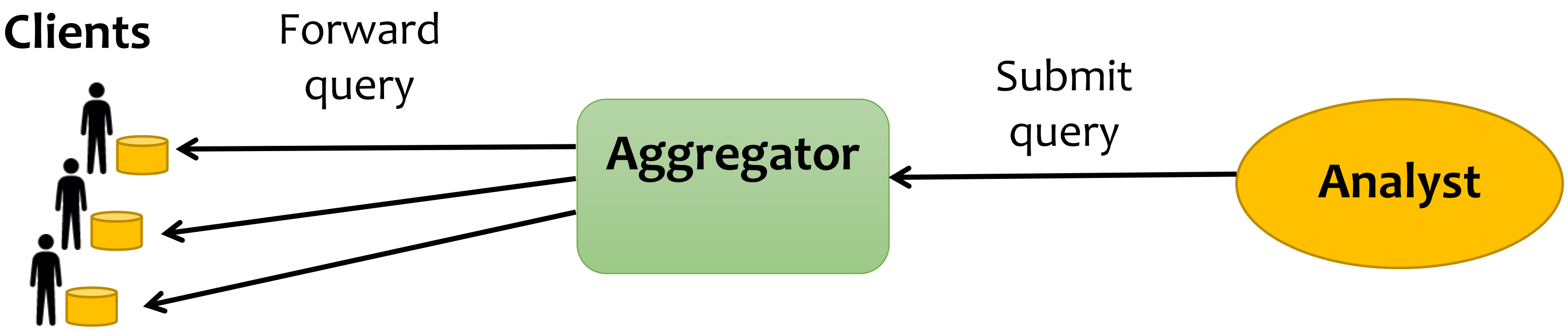
State-of-the-art systems

Clients



State-of-the-art systems

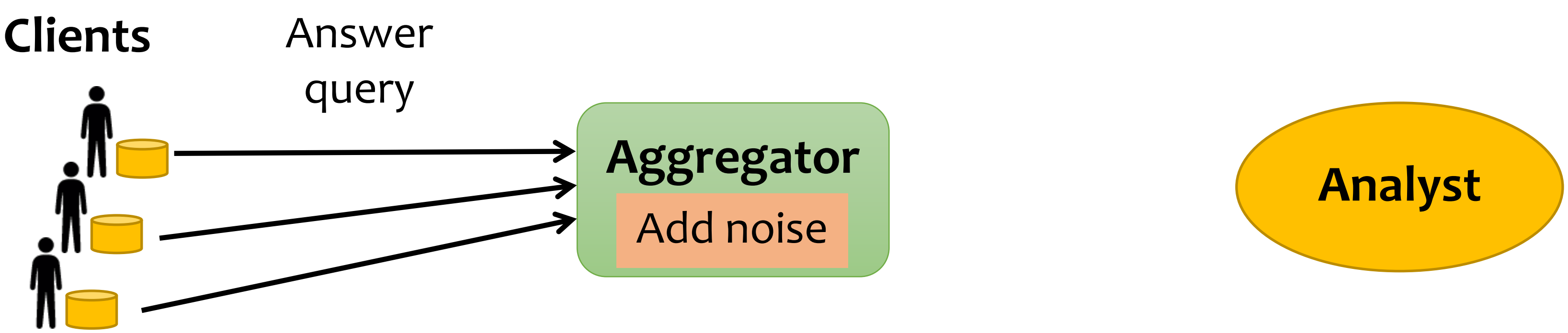
State-of-the-art systems



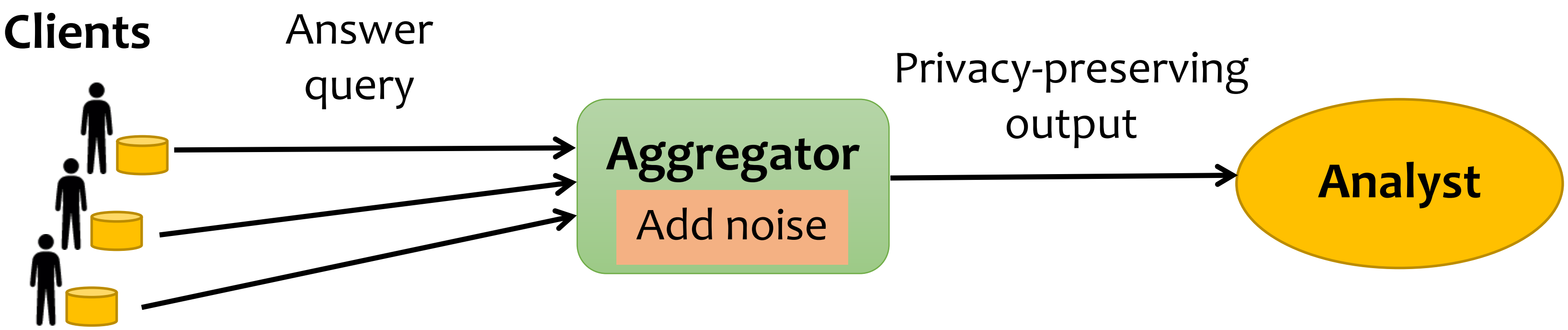
State-of-the-art systems

State-of-the-art systems

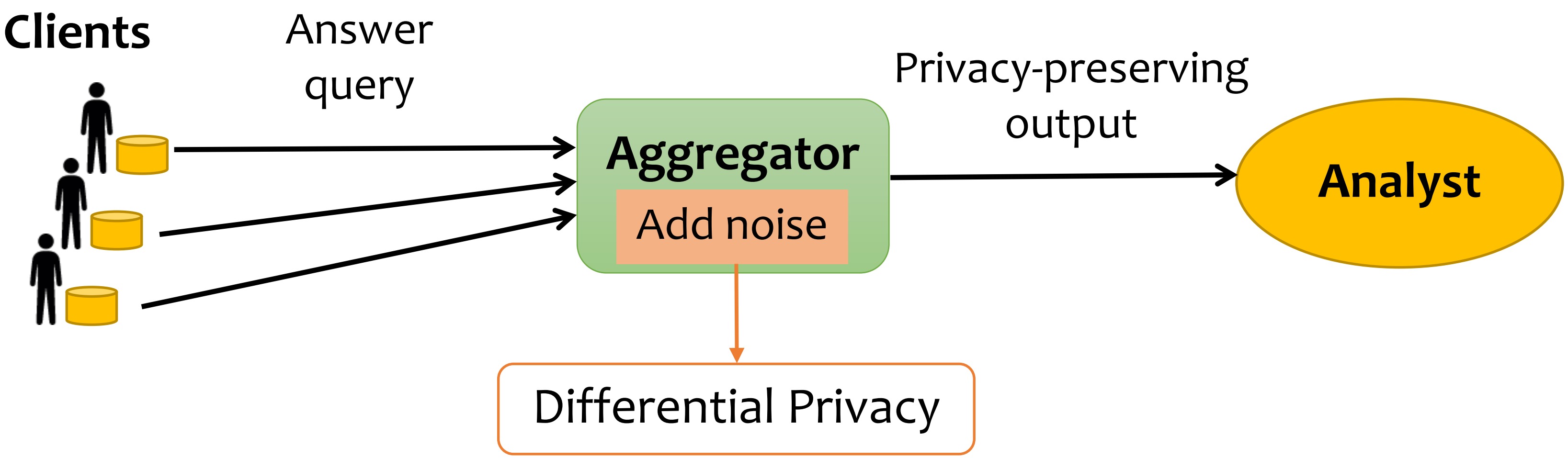
State-of-the-art systems



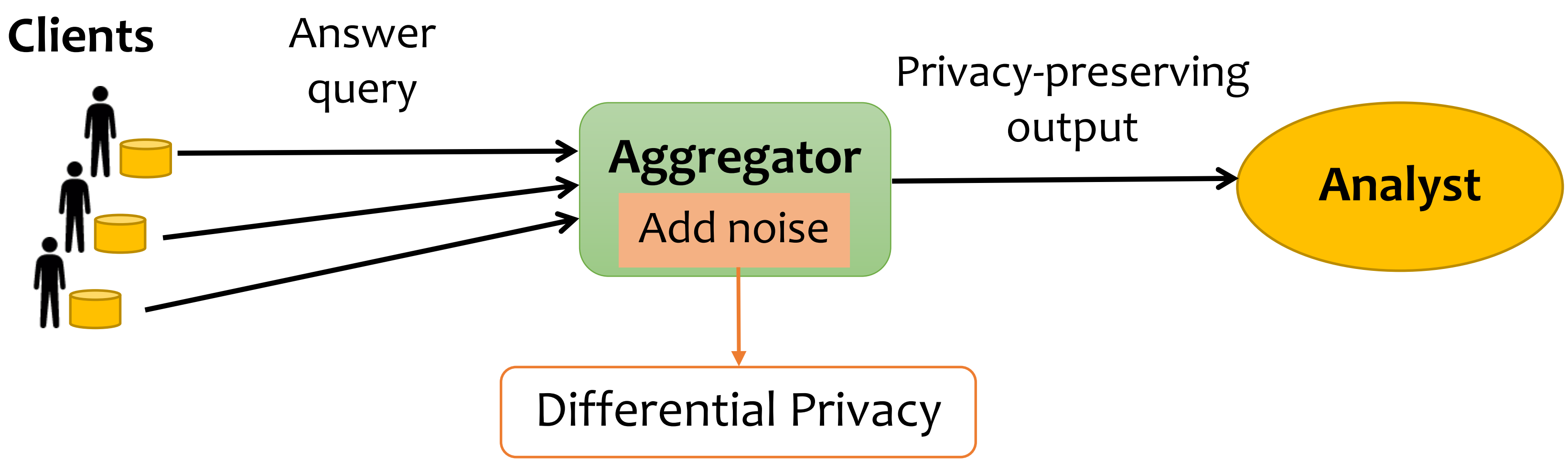
State-of-the-art systems



State-of-the-art systems

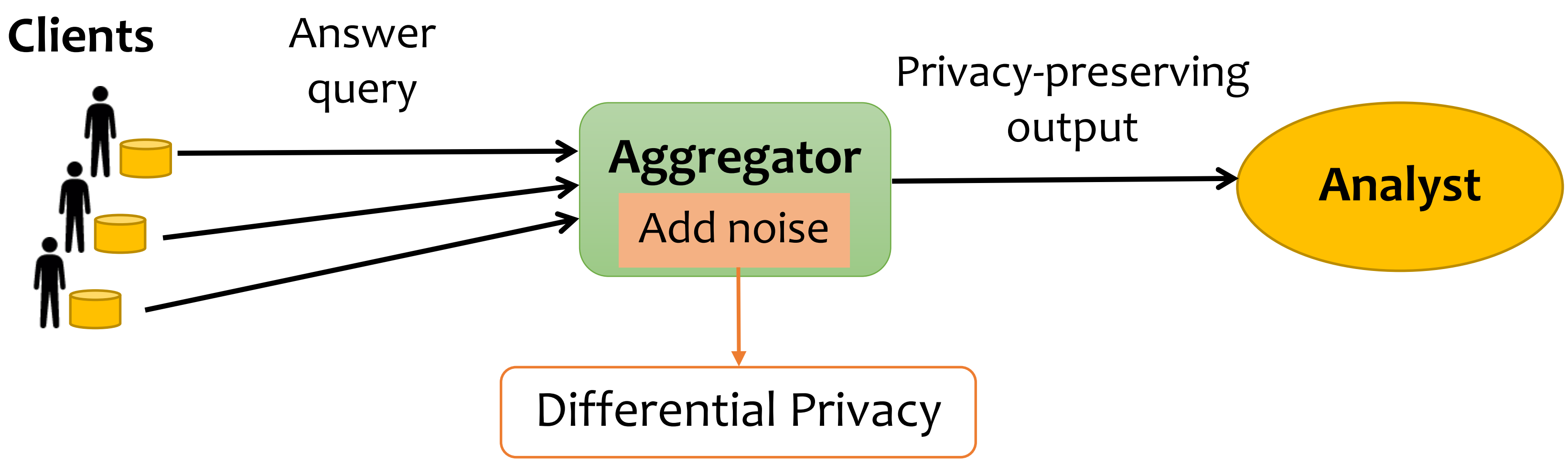


State-of-the-art systems



Limitations:

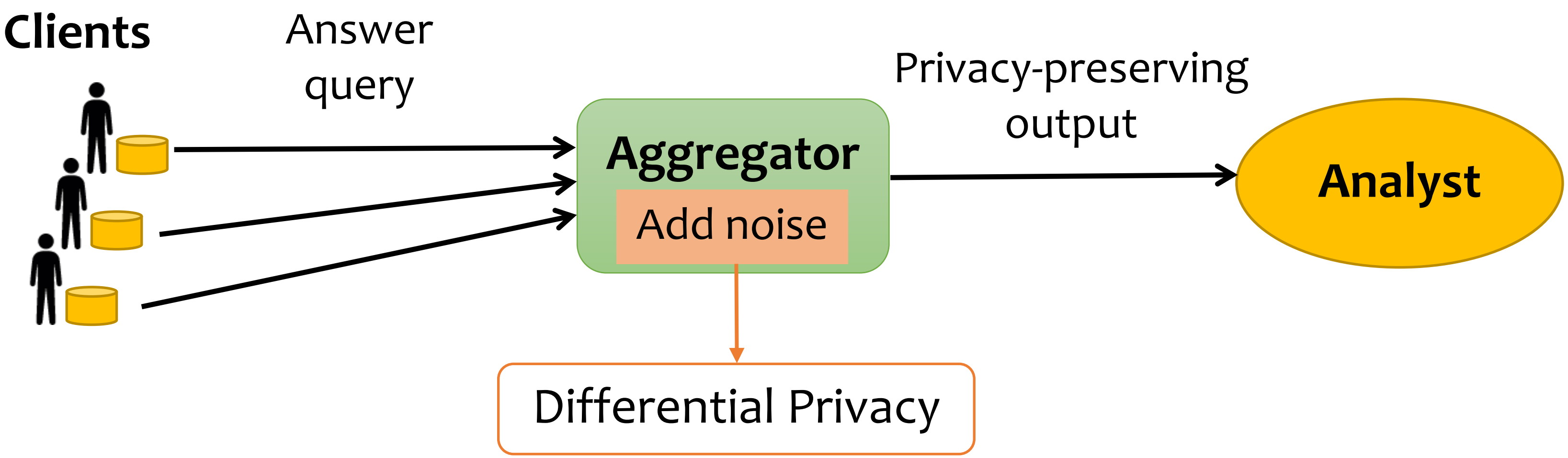
State-of-the-art systems



Limitations:

- Deal with only “single-shot” batch queries ☹

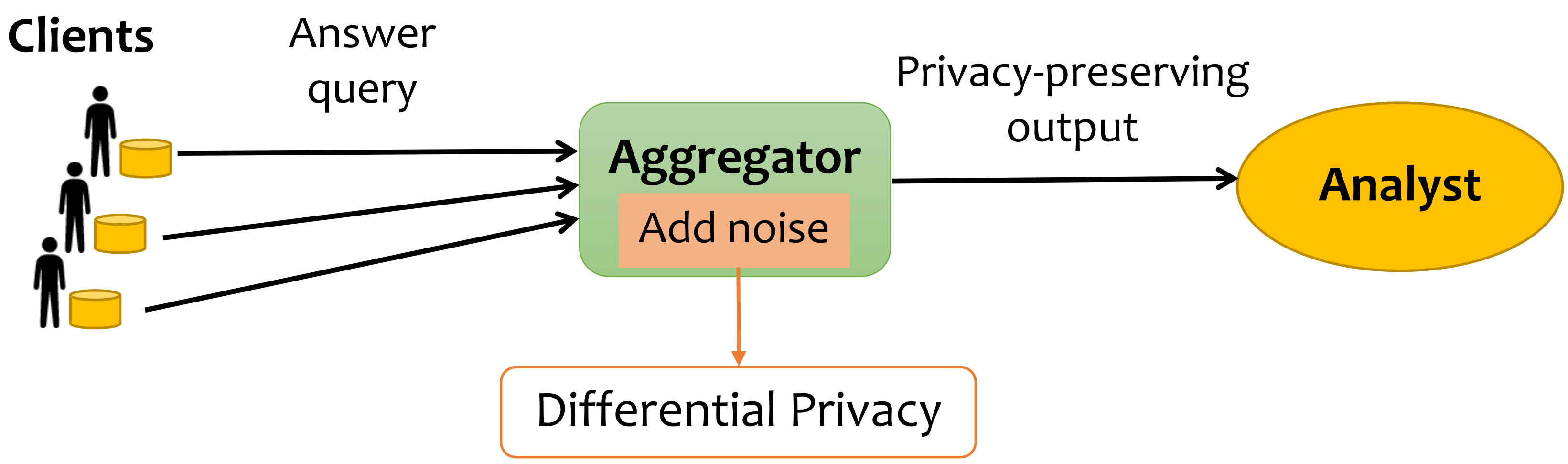
State-of-the-art systems



Limitations:

- Deal with only "single-shot" batch queries ☹
- Require synchronization between system components ☹

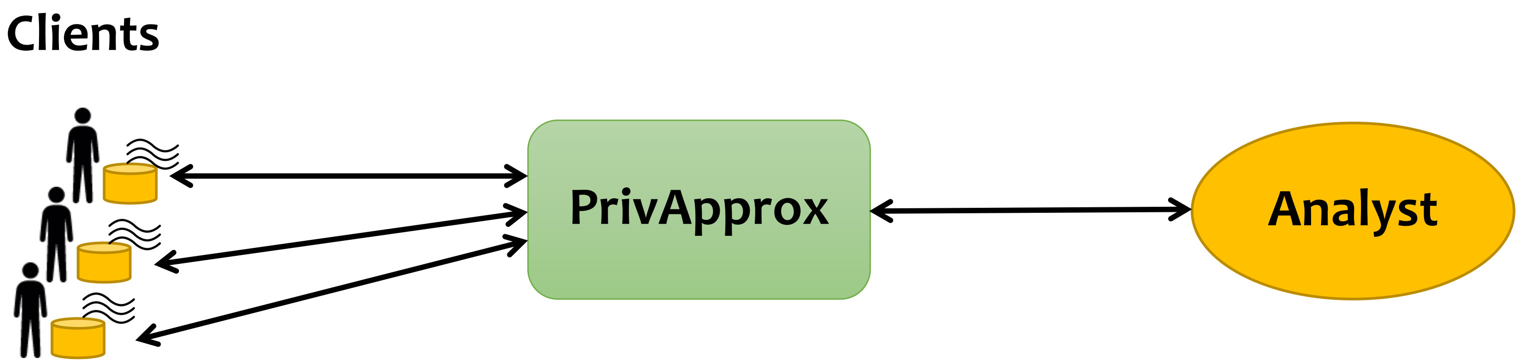
State-of-the-art systems



Limitations:

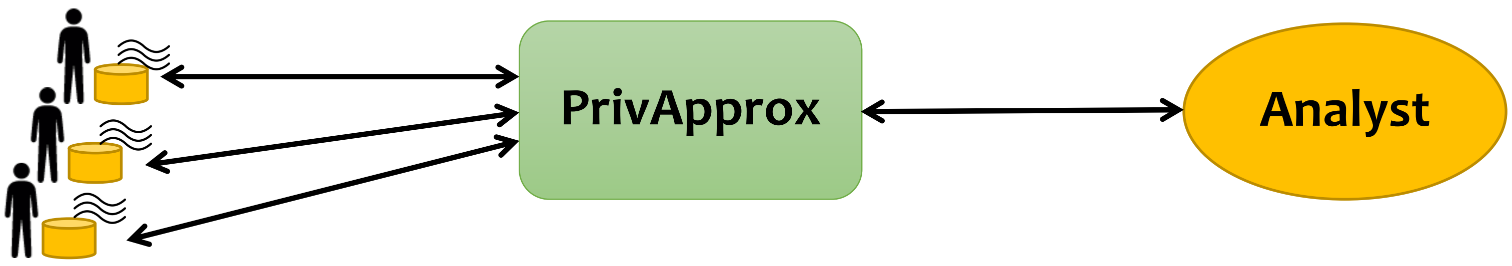
- Deal with only "single-shot" batch queries ☹
- Require synchronization between system components ☹
- Require a trusted aggregator ☹

PrivApprox



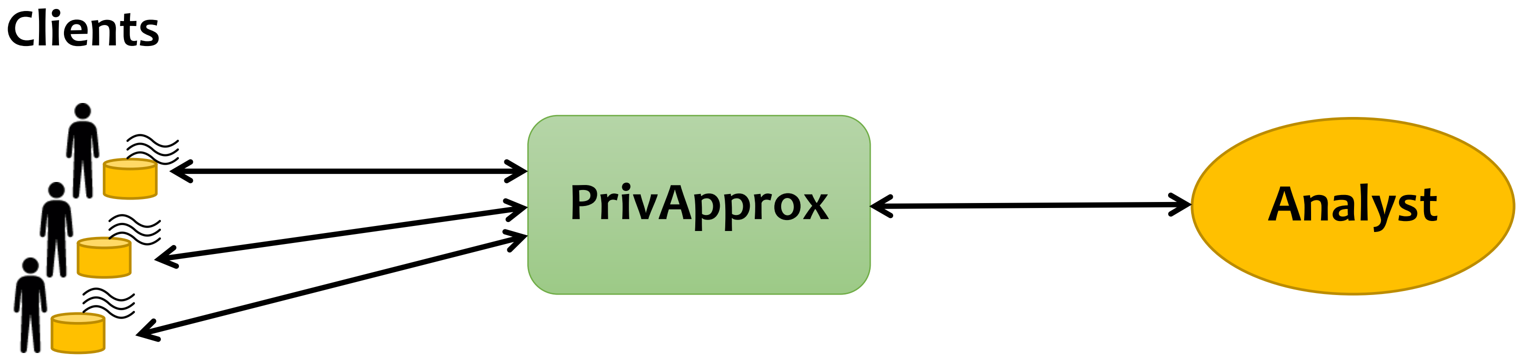
PrivApprox

Clients



PrivApprox:

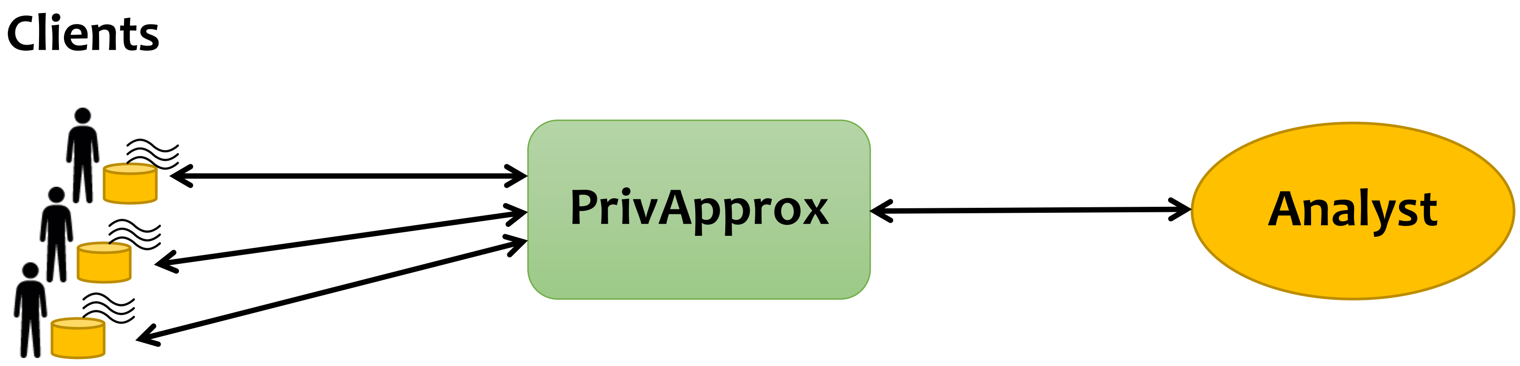
PrivApprox



PrivApprox:

- Supports **stream processing** with **low latency** ☺

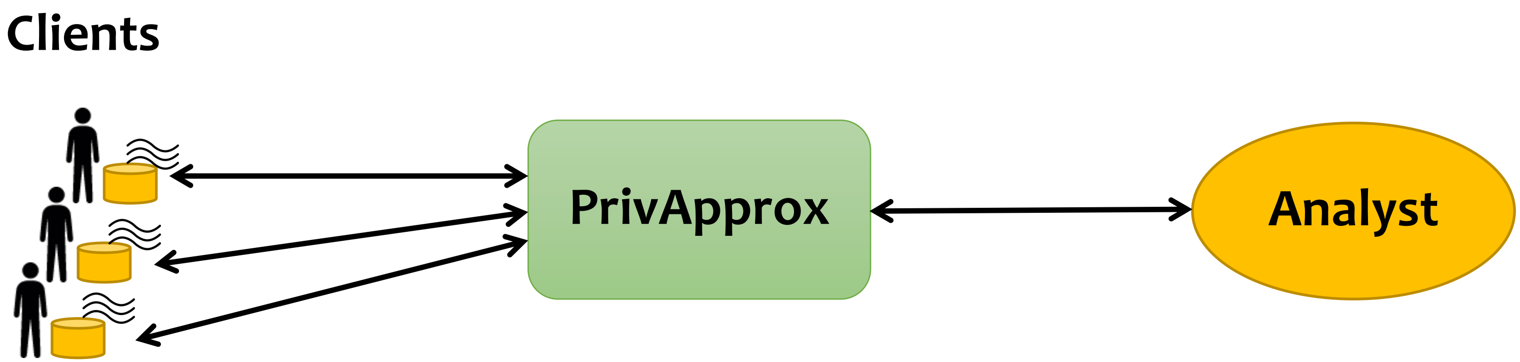
PrivApprox



PrivApprox:

- Supports **stream processing** with **low latency** ☺
- Enables a truly **synchronization-free** distributed architecture ☺

PrivApprox



PrivApprox:

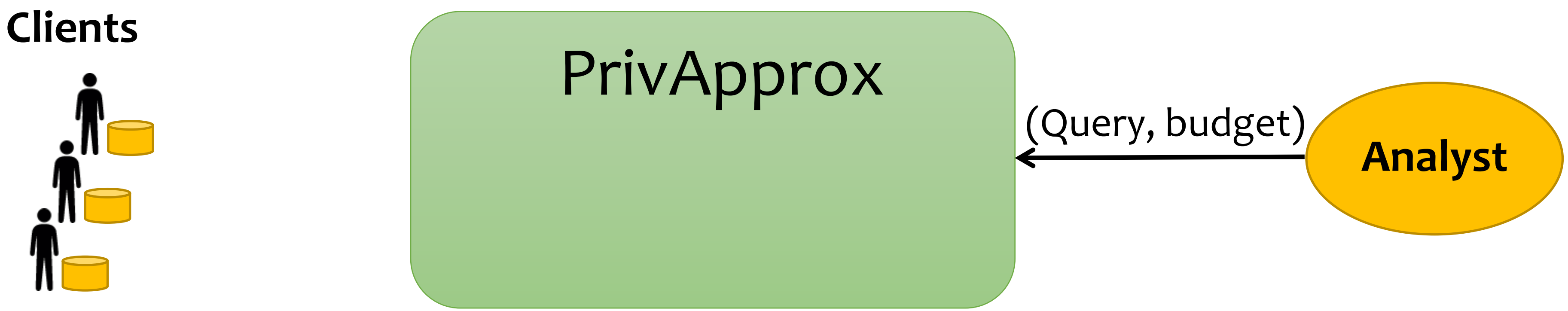
- Supports **stream processing** with **low latency** ☺
- Enables a truly **synchronization-free** distributed architecture ☺
- Requires lower trust in aggregator ☺

Outline

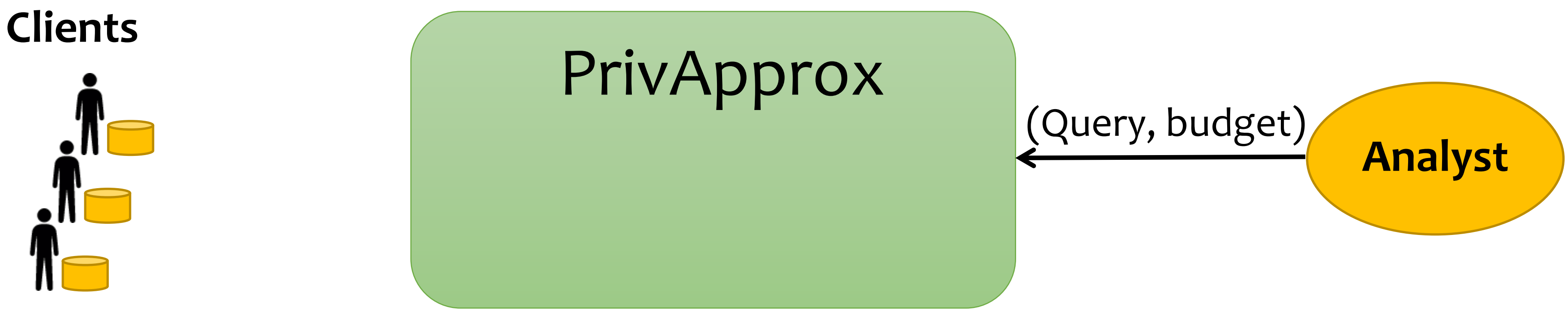
- ~~Motivation~~
- Overview
- Design
- Evaluation

System overview

System overview



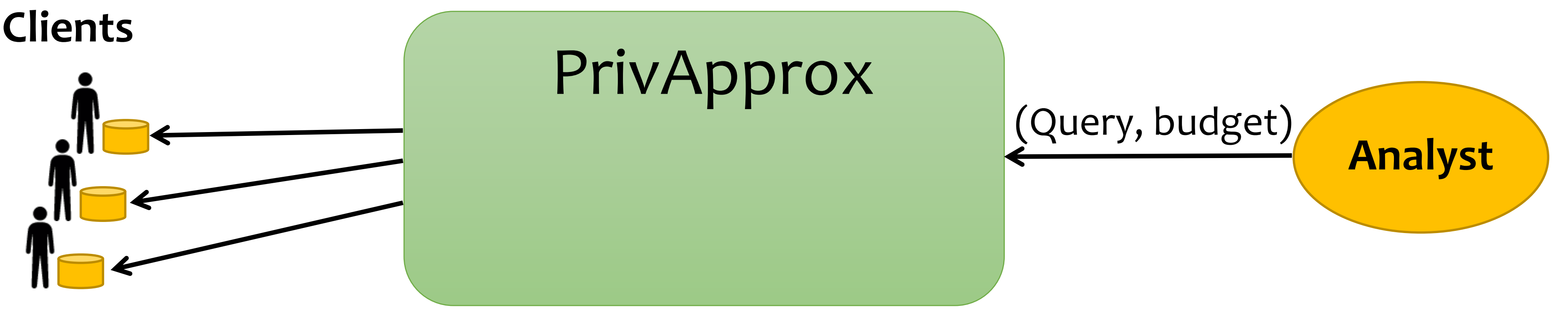
System overview



Execution budget:

- **Latency/throughput** guarantees
- Desired **computing resources** for query processing
- Desired accuracy

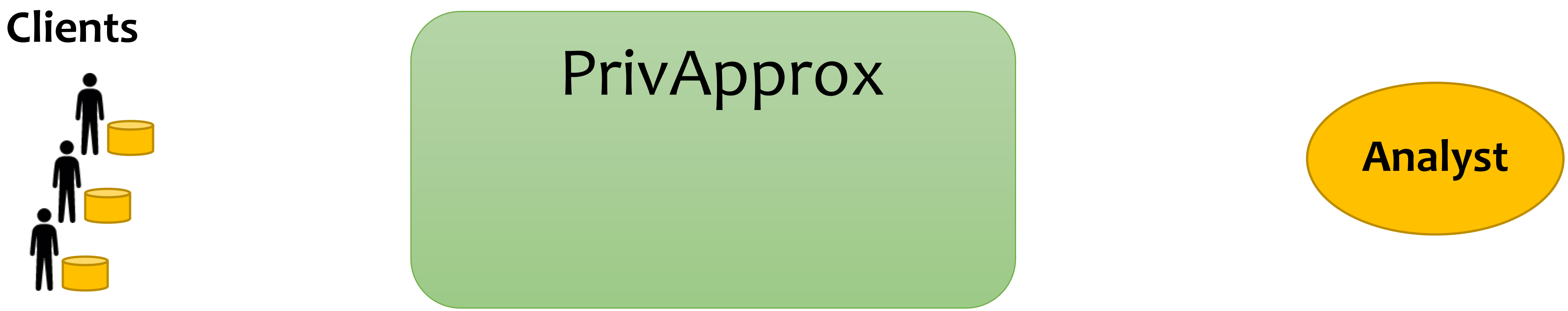
System overview



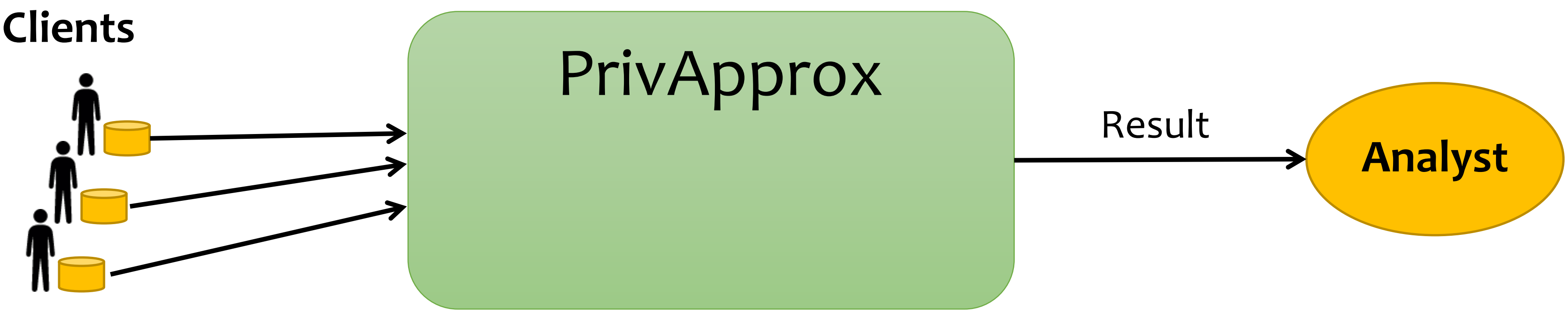
Execution budget:

- **Latency/throughput** guarantees
- Desired **computing resources** for query processing
- Desired accuracy

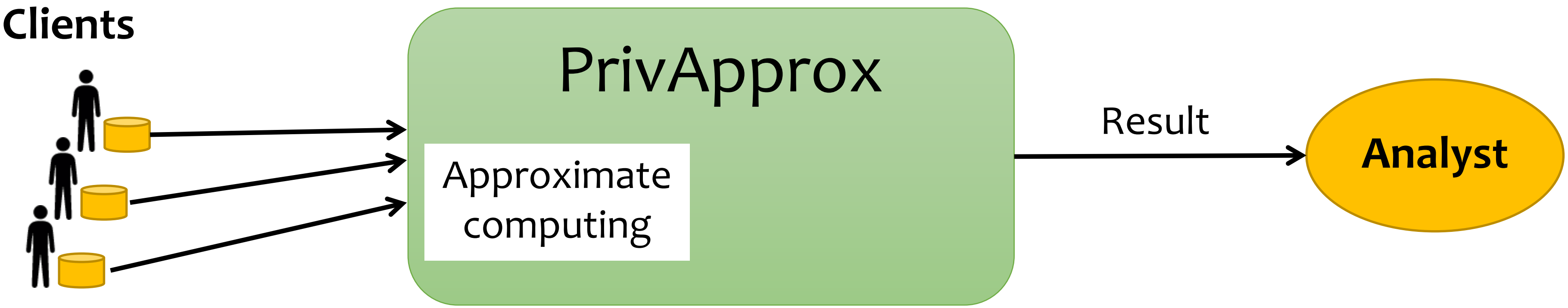
System overview



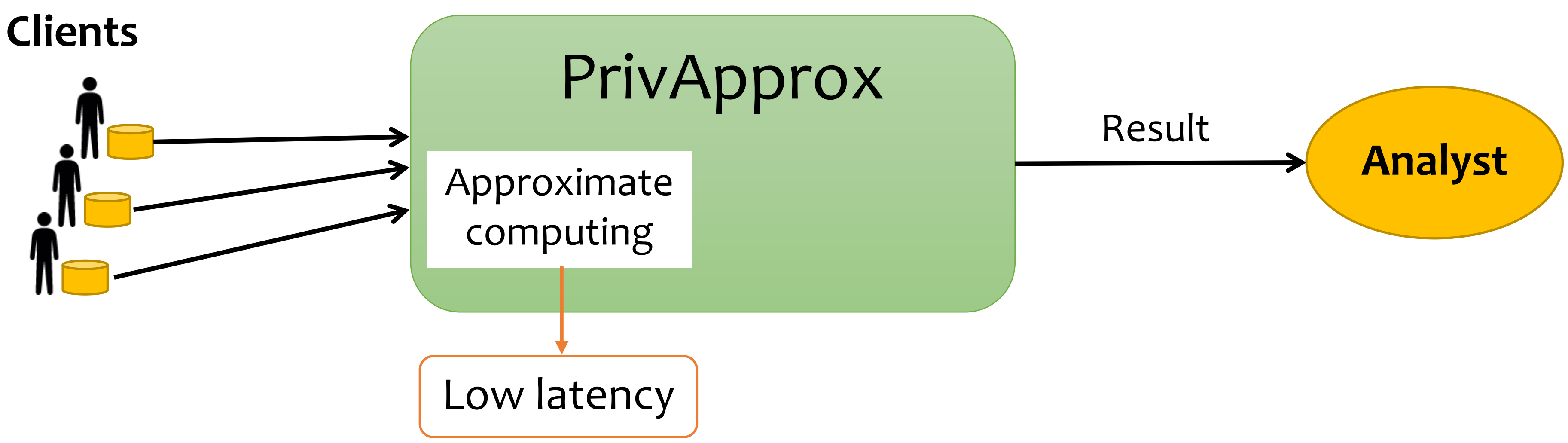
System overview



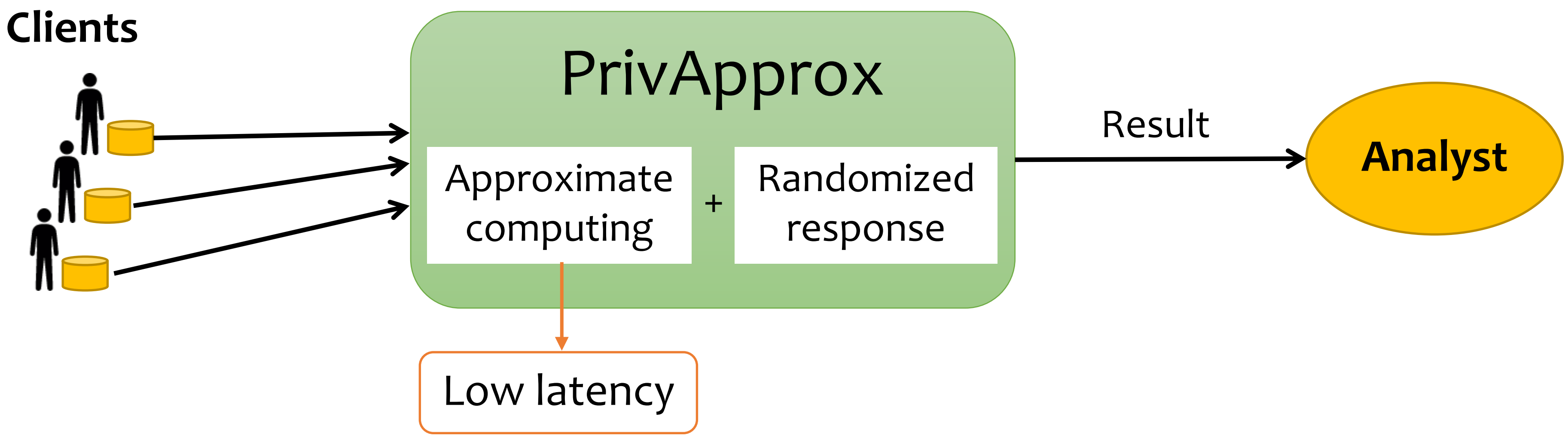
System overview



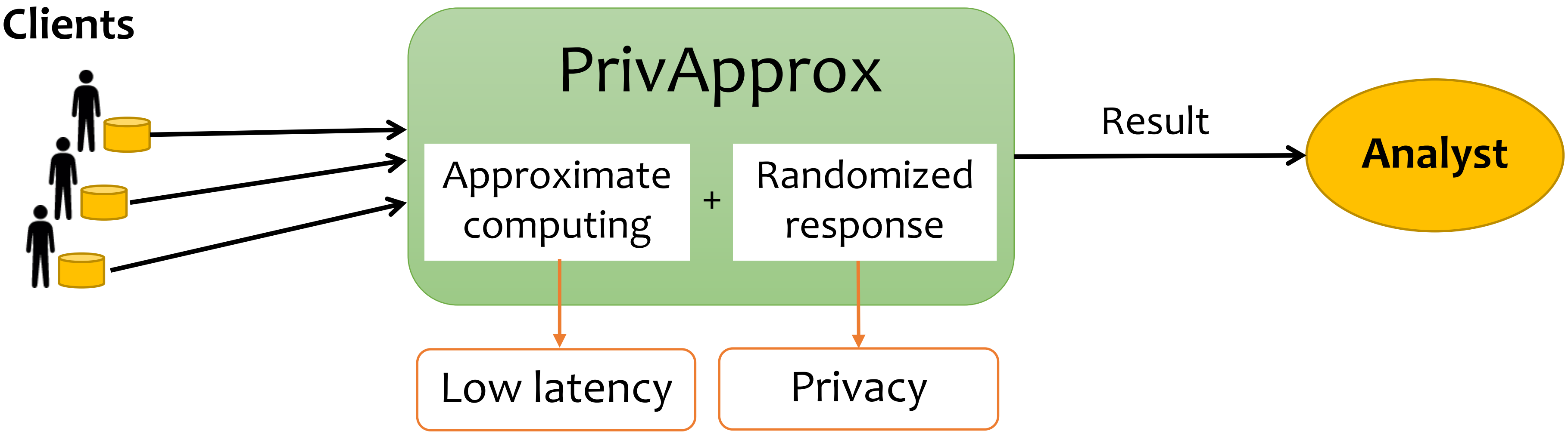
System overview



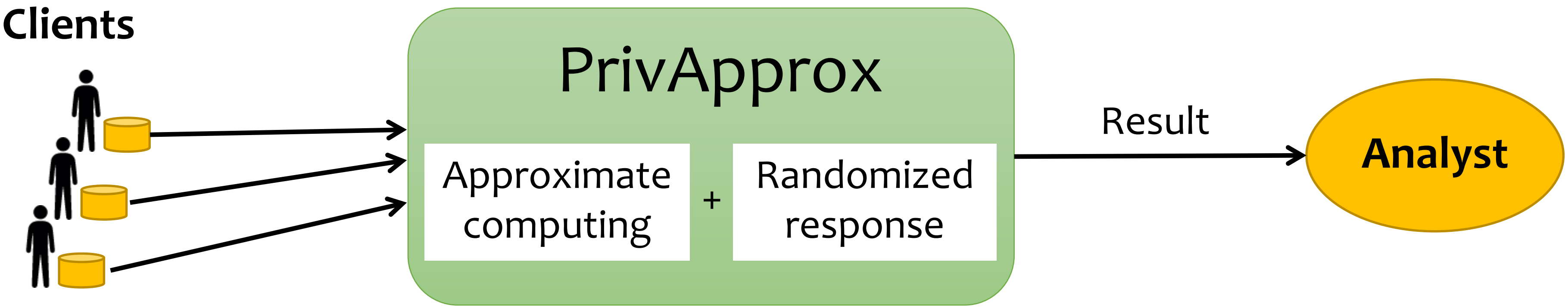
System overview



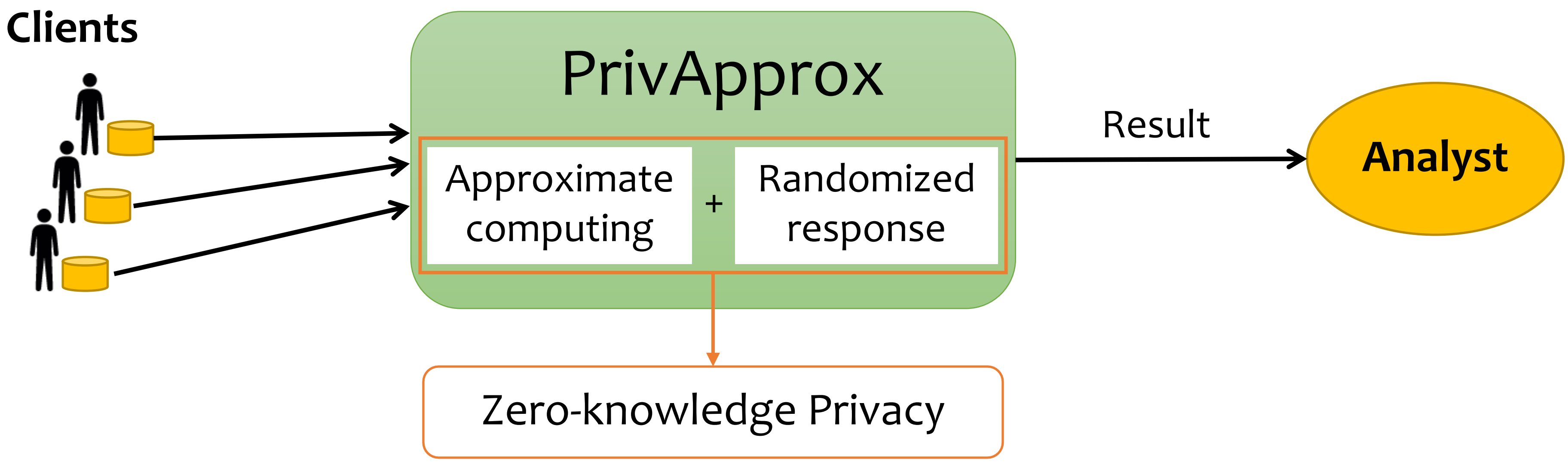
System overview



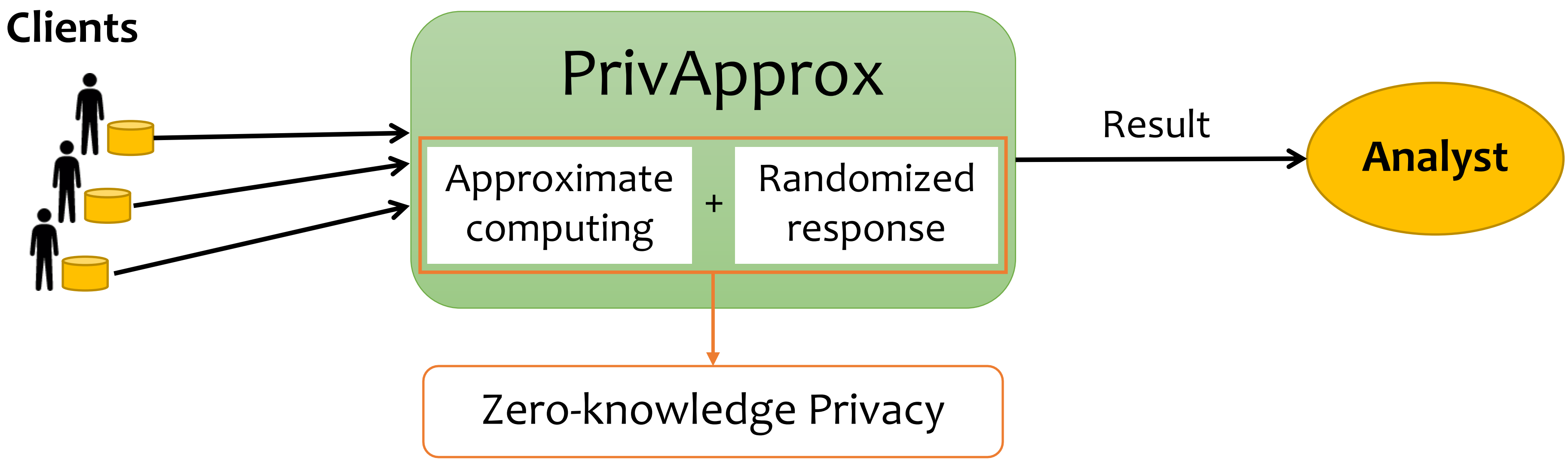
System overview



System overview



System overview

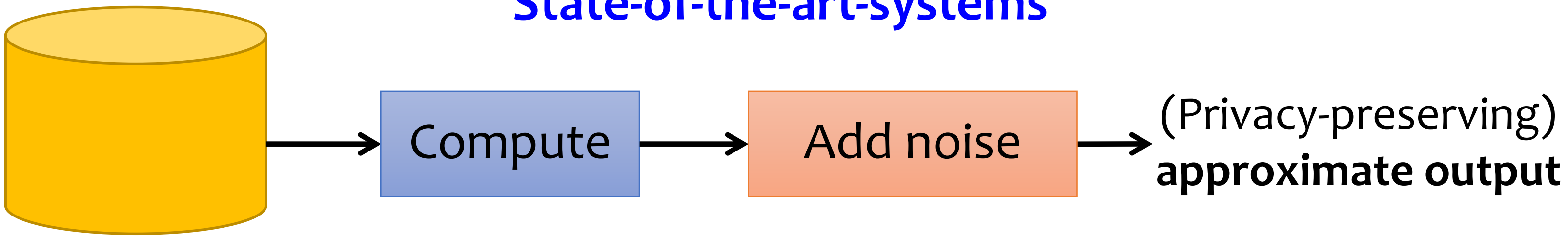


Zero-knowledge Privacy **>** Differential Privacy

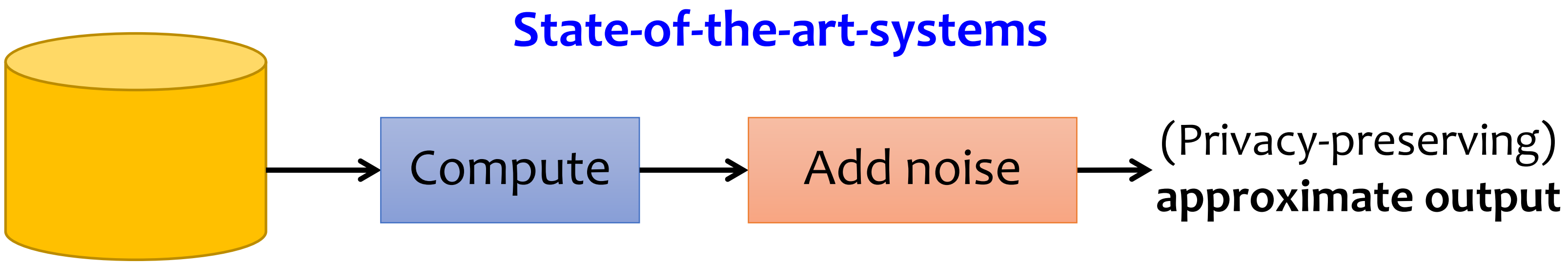
#1: Approximate computing

#1: Approximate computing

State-of-the-art-systems

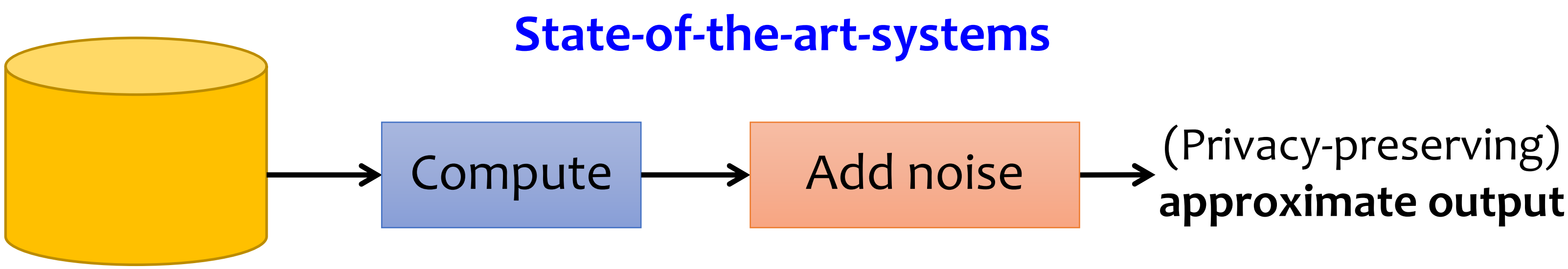


#1: Approximate computing

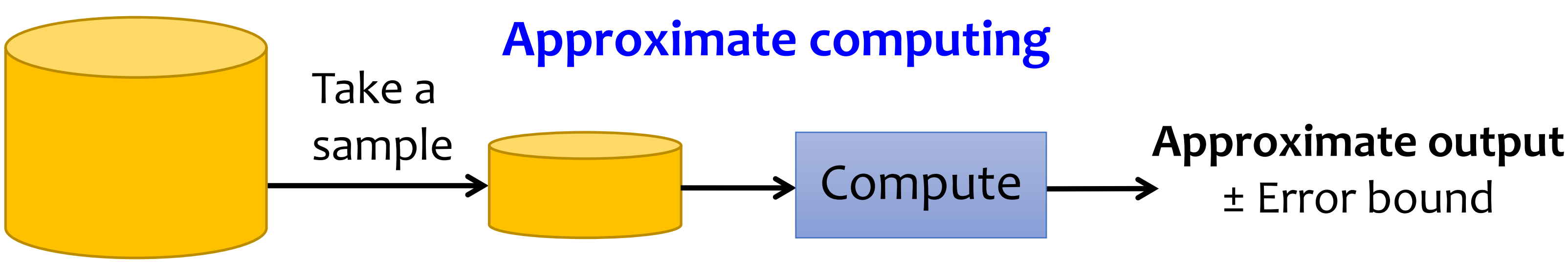


Idea: To achieve low latency, compute over a sub-set of data items instead of the entire data-set

#1: Approximate computing



Idea: To achieve low latency, compute over a sub-set of data items instead of the entire data-set



#2: Randomized response

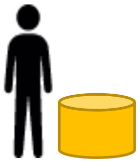
#2: Randomized response

Idea: To preserve privacy, clients may not need to provide truthful answers every time

#2: Randomized response

Idea: To preserve privacy, clients may not need to provide truthful answers every time

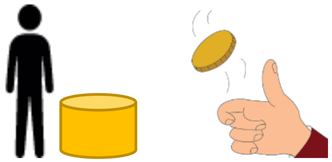
Client



#2: Randomized response

Idea: To preserve privacy, clients may not need to provide truthful answers every time

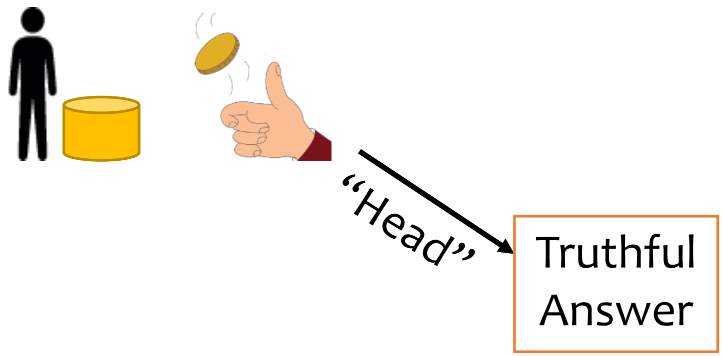
Client



#2: Randomized response

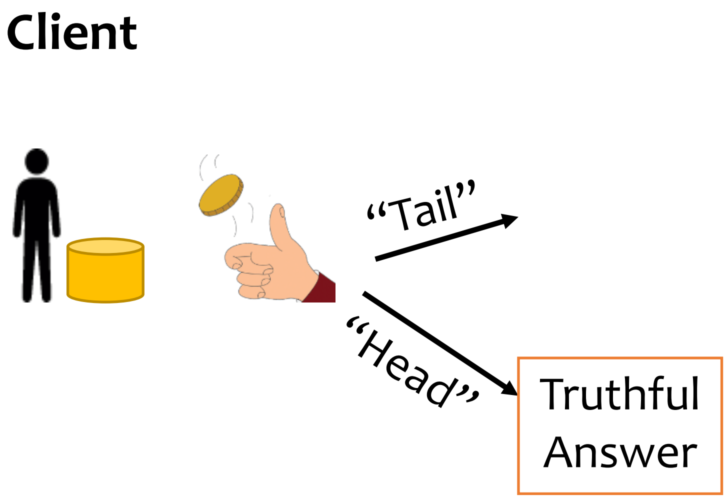
Idea: To preserve privacy, clients may not need to provide truthful answers every time

Client



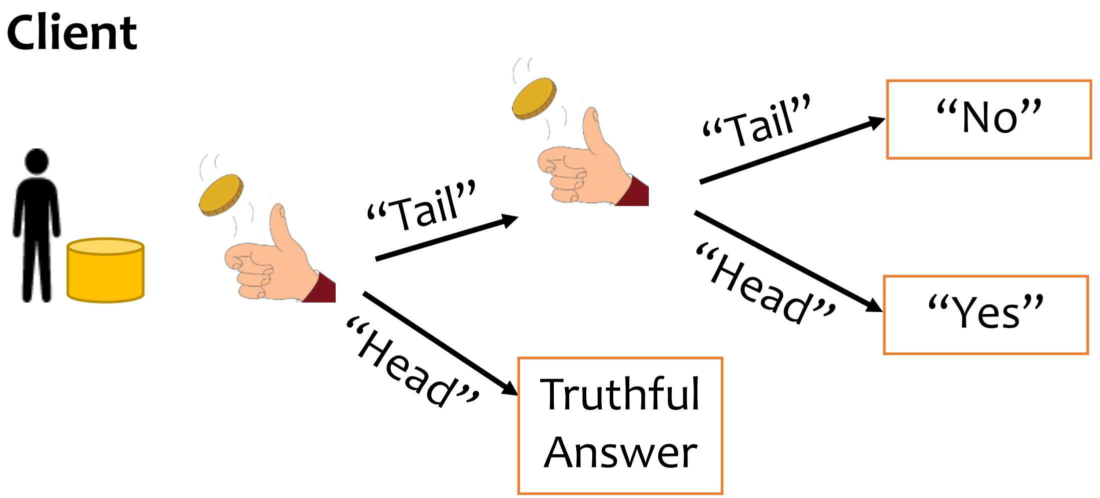
#2: Randomized response

Idea: To preserve privacy, clients may not need to provide truthful answers every time



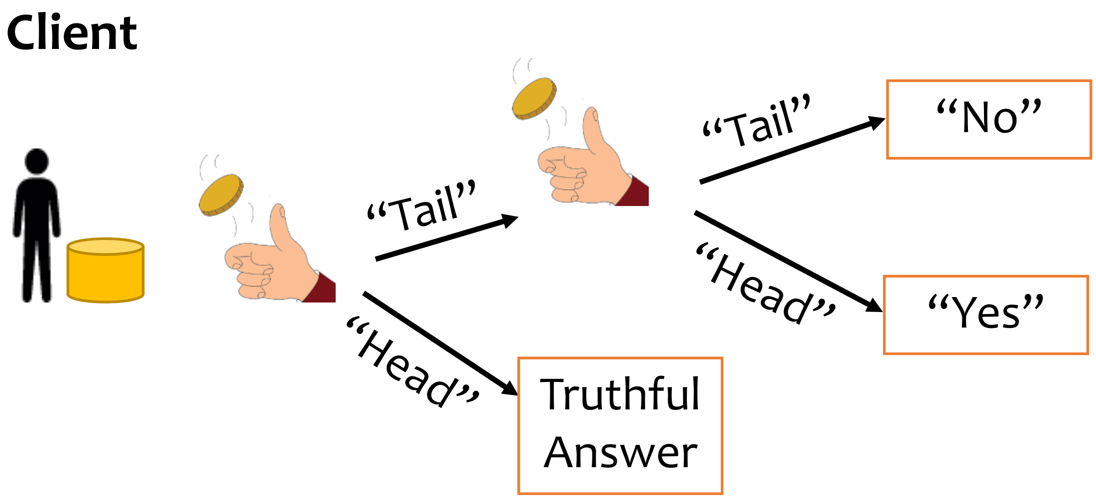
#2: Randomized response

Idea: To preserve privacy, clients may not need to provide truthful answers every time



#2: Randomized response

Idea: To preserve privacy, clients may not need to provide truthful answers every time



Provides **plausible deniability** for clients responding to sensitive queries; achieves **differential privacy** (RAPPOR [CCS'14])

Outline

- ~~Motivation~~
- ~~Overview~~
- Design
- Evaluation

Query model

Query model

Divide answer's value range into **buckets,**
enforce a **binary answer** in each bucket

Query model

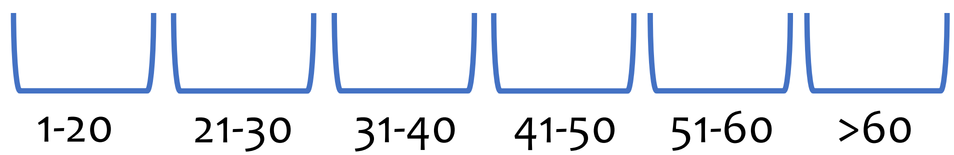
Divide answer's value range into **buckets,**
enforce a **binary answer** in each bucket

Query: SELECT age FROM clients WHERE city = 'Santa Clara'

Query model

Divide answer's value range into **buckets,**
enforce a **binary answer** in each bucket

Query: SELECT age FROM clients WHERE city = 'Santa Clara'



Query model

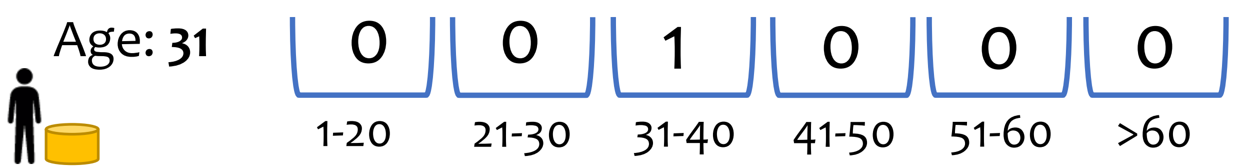
Divide answer's value range into **buckets,**
enforce a **binary answer** in each bucket

Query: SELECT age FROM clients WHERE city = 'Santa Clara'

Query model

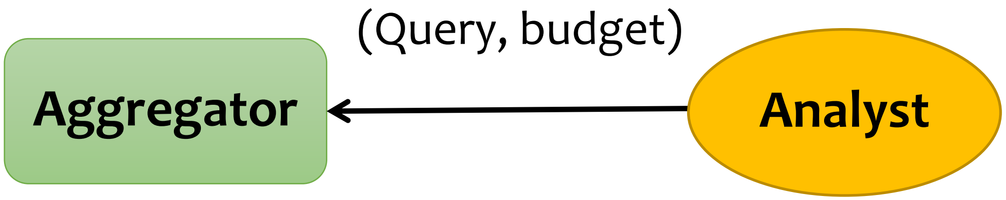
Divide answer's value range into **buckets,**
enforce a **binary answer** in each bucket

Query: SELECT age FROM clients WHERE city = 'Santa Clara'

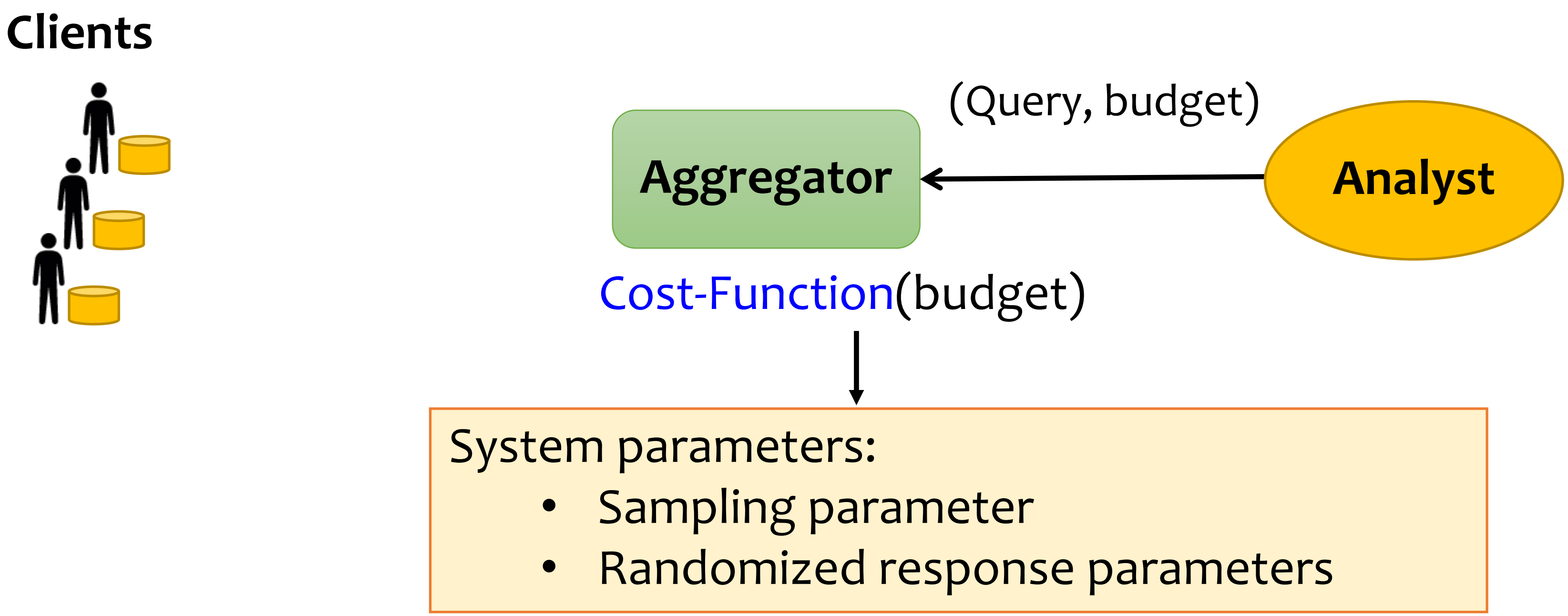


Client cannot arbitrarily manipulate answers

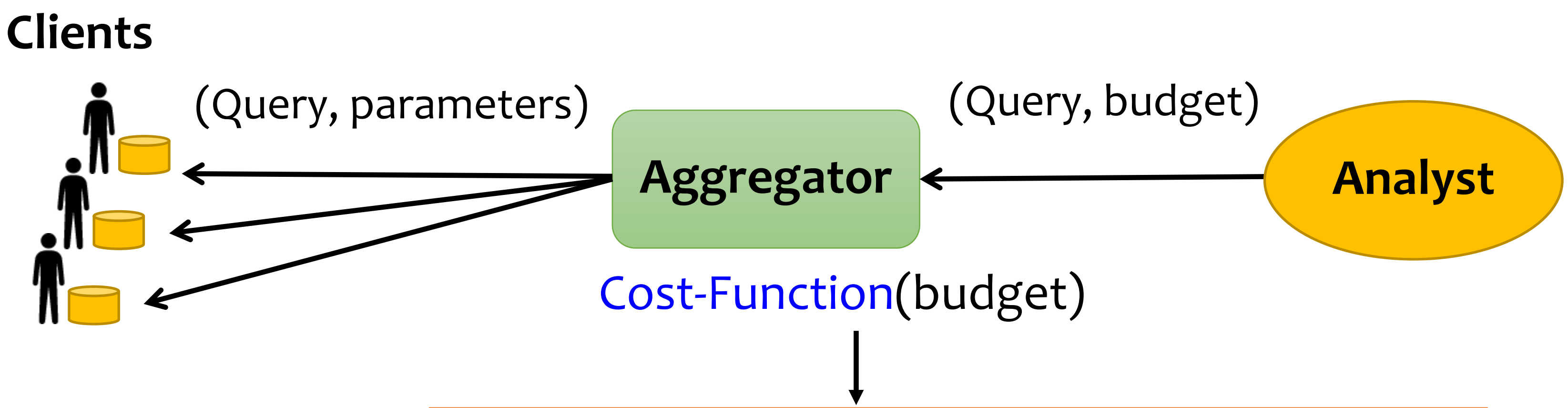
Workflow: Submit query



Workflow: Submit query



Workflow: Submit query

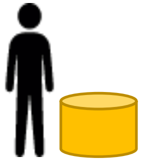


System parameters:
- Sampling parameter
- Randomized response parameters

Workflow: Answer query

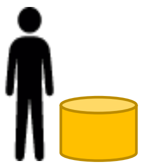
Workflow: Answer query

Client



Workflow: Answer query

Client

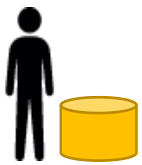


Step #1

Sampling
(Flip a coin to decide to answer query or not)

Workflow: Answer query

Client



Step #1

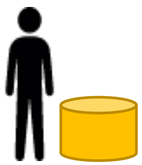
Sampling
(Flip a coin to decide to answer query or not)

Step #2

Randomized Response

Workflow: Answer query

Client



Step #1

Sampling

(Flip a coin to decide to answer query or not)

Step #2

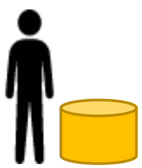
Randomized Response

Step #3

Send randomized answer

Workflow: Answer query

Client



Step #1

Sampling
(Flip a coin to decide to answer query or not)

Step #2

Randomized Response

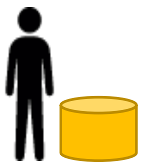
Step #3

Send randomized answer

Zero-knowledge privacy

Workflow: Answer query

Client



Step #1

Sampling
(Flip a coin to decide to answer query or not)

Step #2

Randomized Response

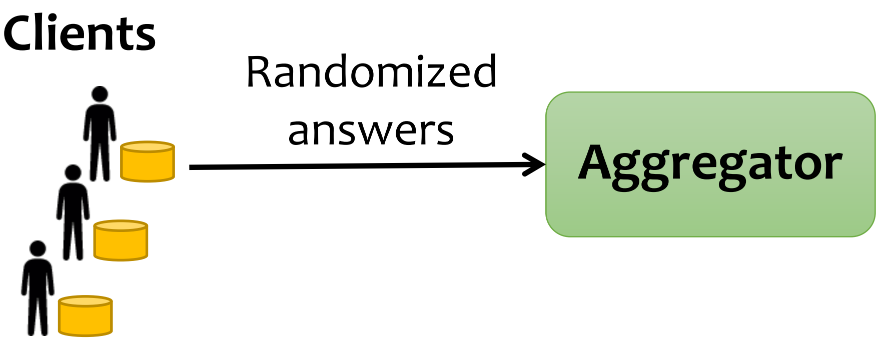
Step #3

Send randomized answer

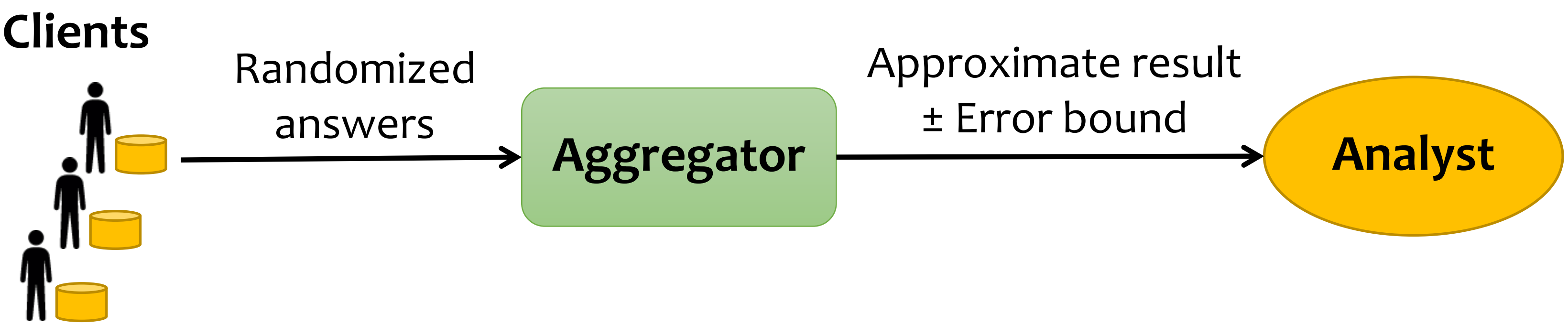
Zero-knowledge privacy

See the paper for details!

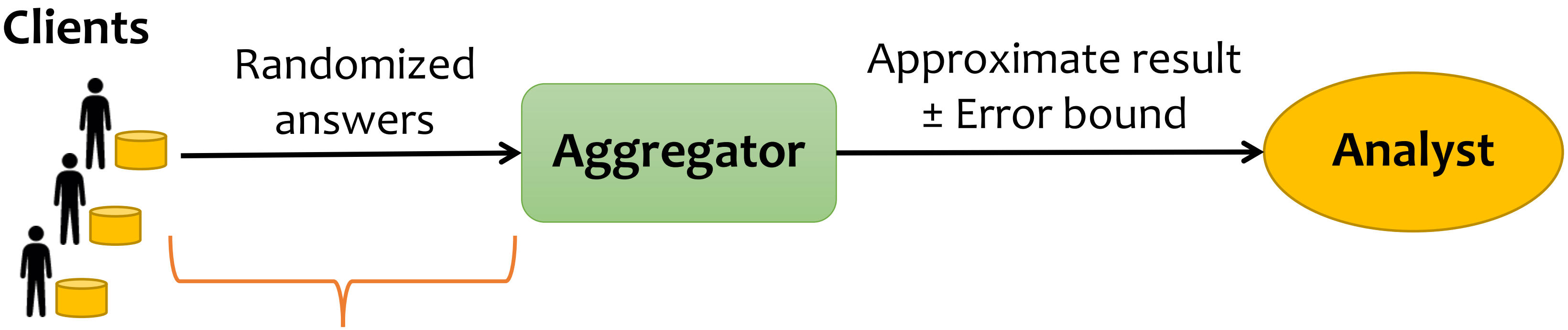
Workflow: Answer query



Workflow: Answer query



Workflow: Answer query



Lack of anonymity and unlinkability?

#3: Anonymity and unlinkability

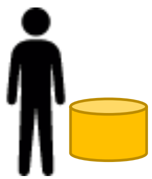
#3: Anonymity and unlinkability

Idea: XOR-based Encryption

#3: Anonymity and unlinkability

Idea: XOR-based Encryption

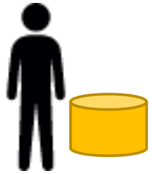
Client



#3: Anonymity and unlinkability

Idea: XOR-based Encryption

Client



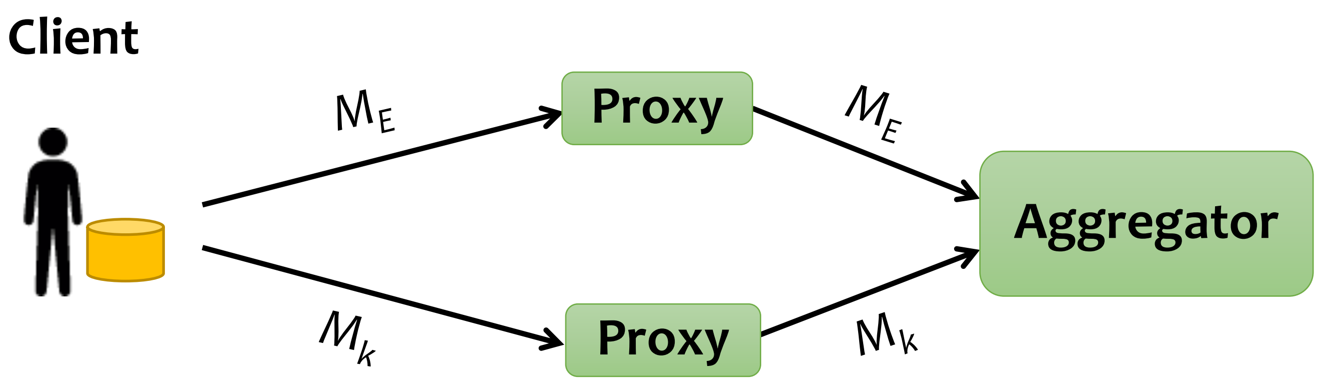
Encrypt answer *M*:

GenerateKey -> M_k

M XOR M_k -> M_E

#3: Anonymity and unlinkability

Idea: XOR-based Encryption



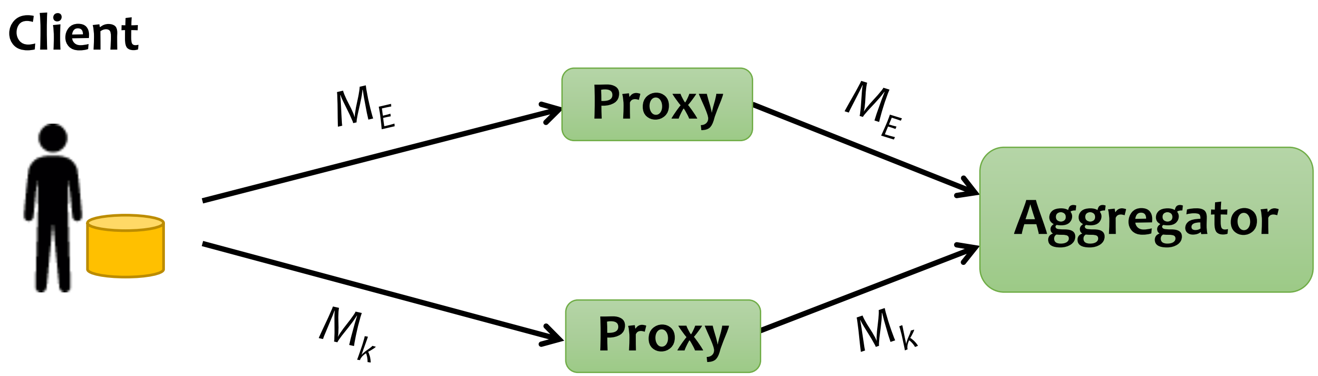
Encrypt answer *M*:

GenerateKey -> M_k

M XOR M_k -> M_E

#3: Anonymity and unlinkability

Idea: XOR-based Encryption



Encrypt answer *M*:

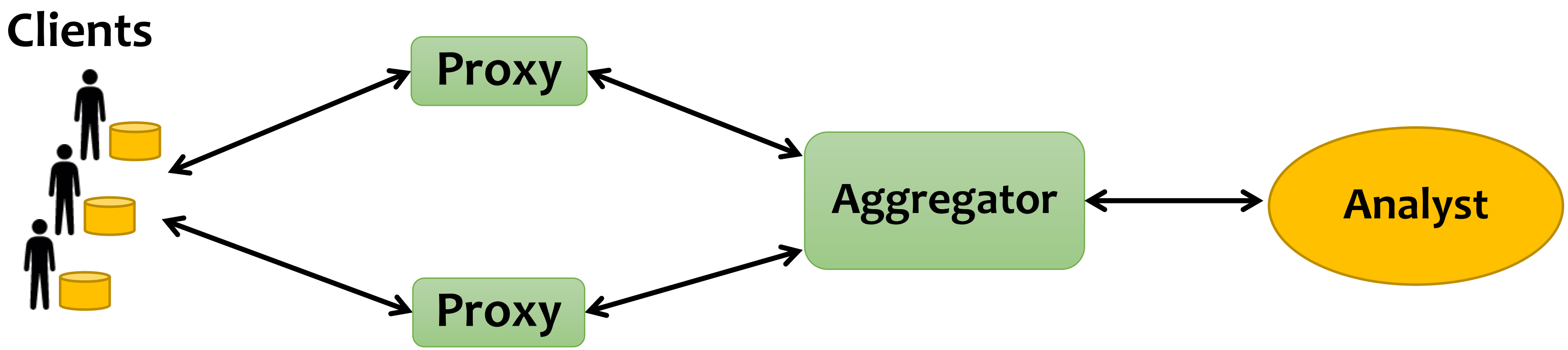
GenerateKey -> M_k

M XOR M_k -> M_E

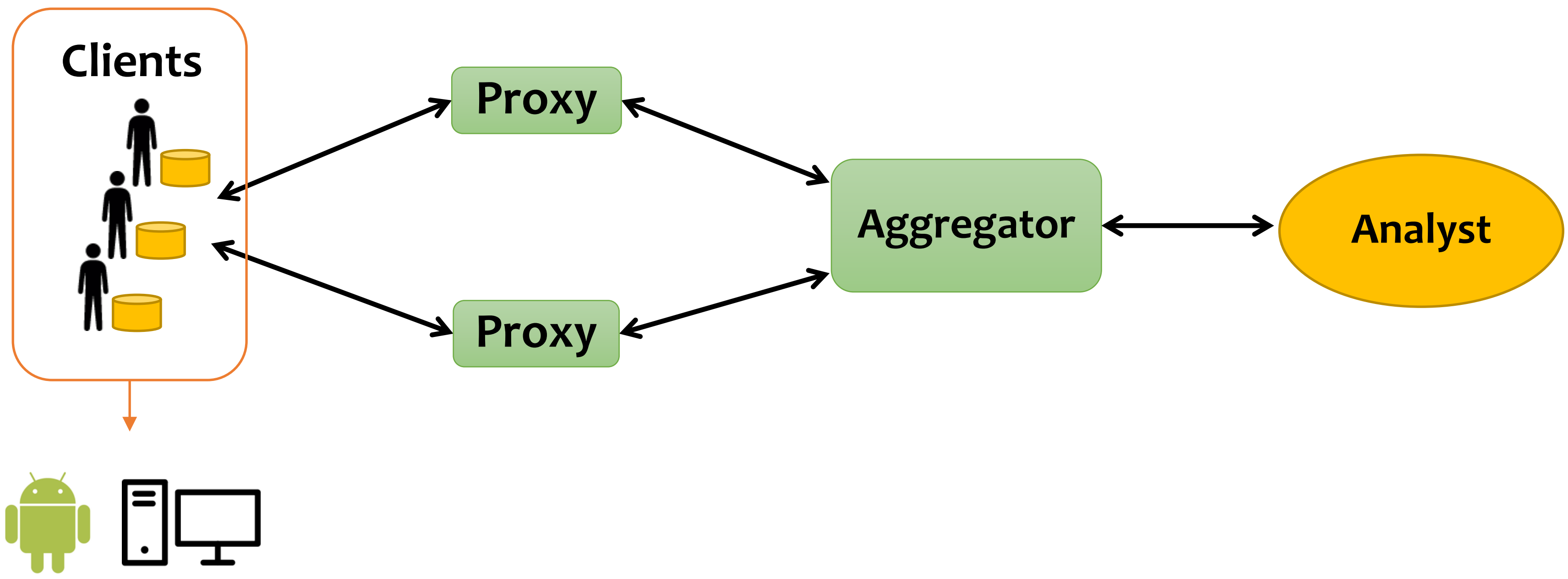
Decrypt answer M_E:

M_E XOR M_k -> M

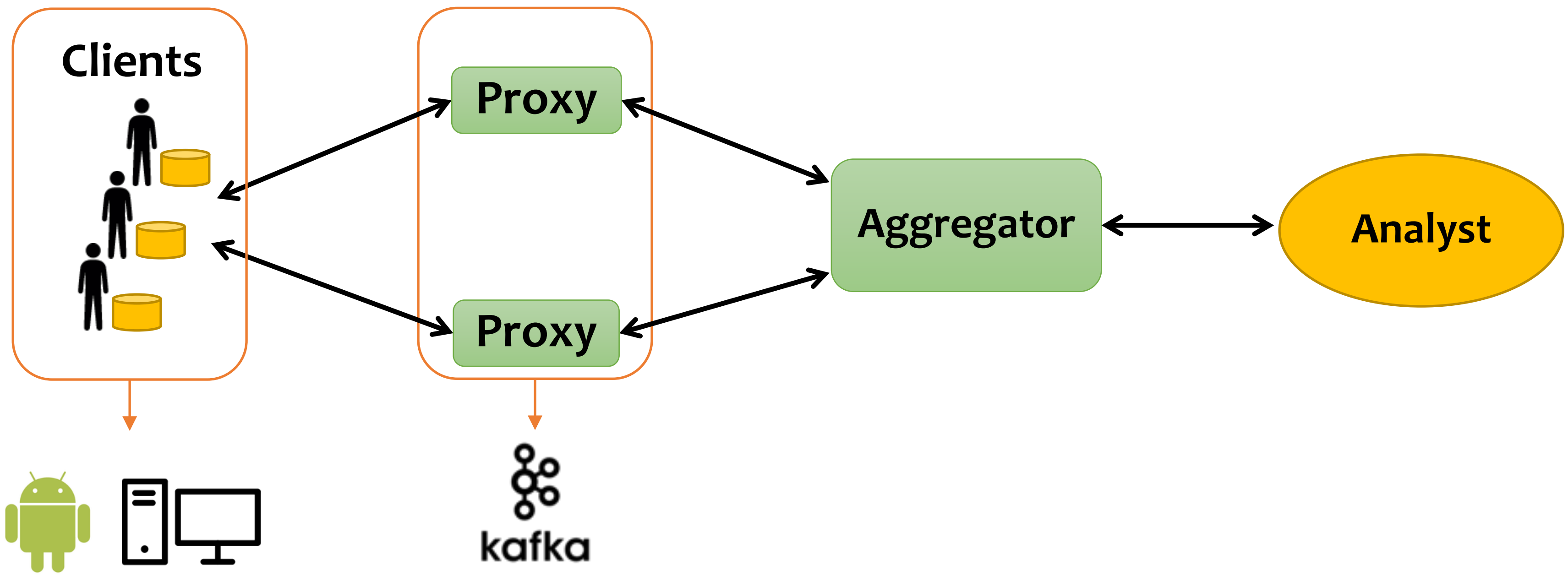
Implementation



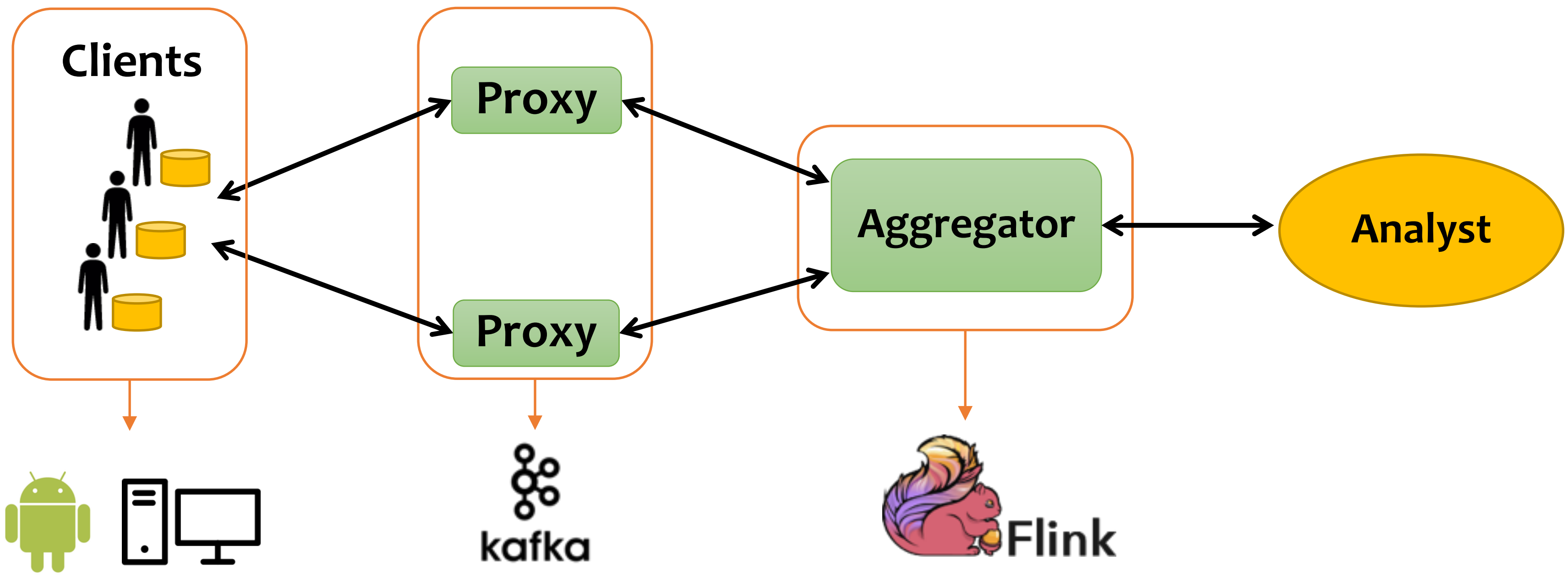
Implementation



Implementation



Implementation



Outline

- ~~Motivation~~
- ~~Overview~~
- ~~Design~~
- Evaluation

Experimental setup

- Evaluation questions
  - Utility vs privacy
  - Throughput & latency
  - Network overhead

Experimental setup

- Evaluation questions
  - Utility vs privacy
  - Throughput & latency
  - Network overhead

See the paper for more results!

Experimental setup

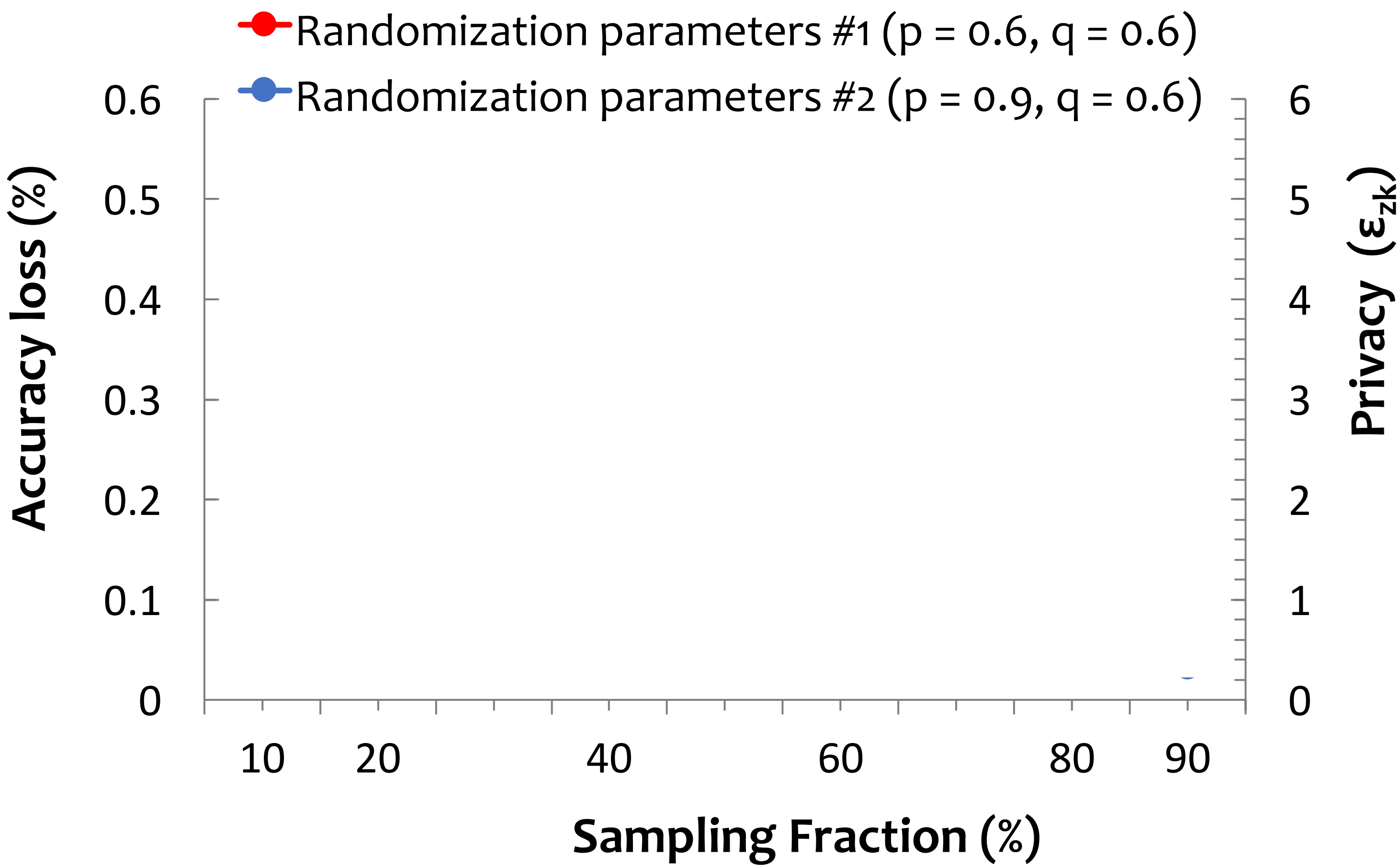
- Evaluation questions
  - Utility vs privacy
  - Throughput & latency
  - Network overhead

See the paper for more results!

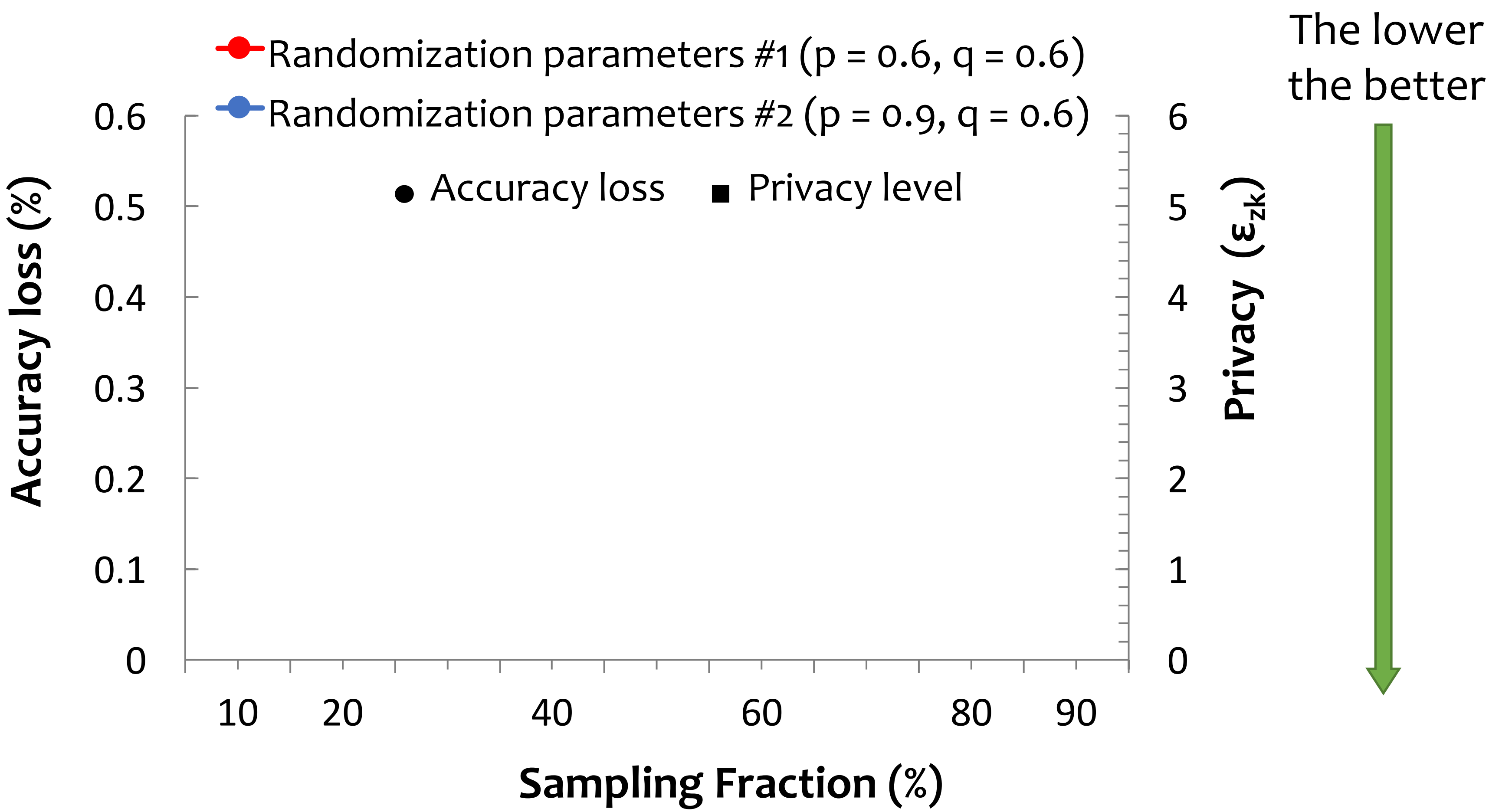
- Testbed
  - Cluster: 44 nodes
  - Dataset: NYC Taxi ride records, household electricity usage

Accuracy vs privacy

Accuracy vs privacy

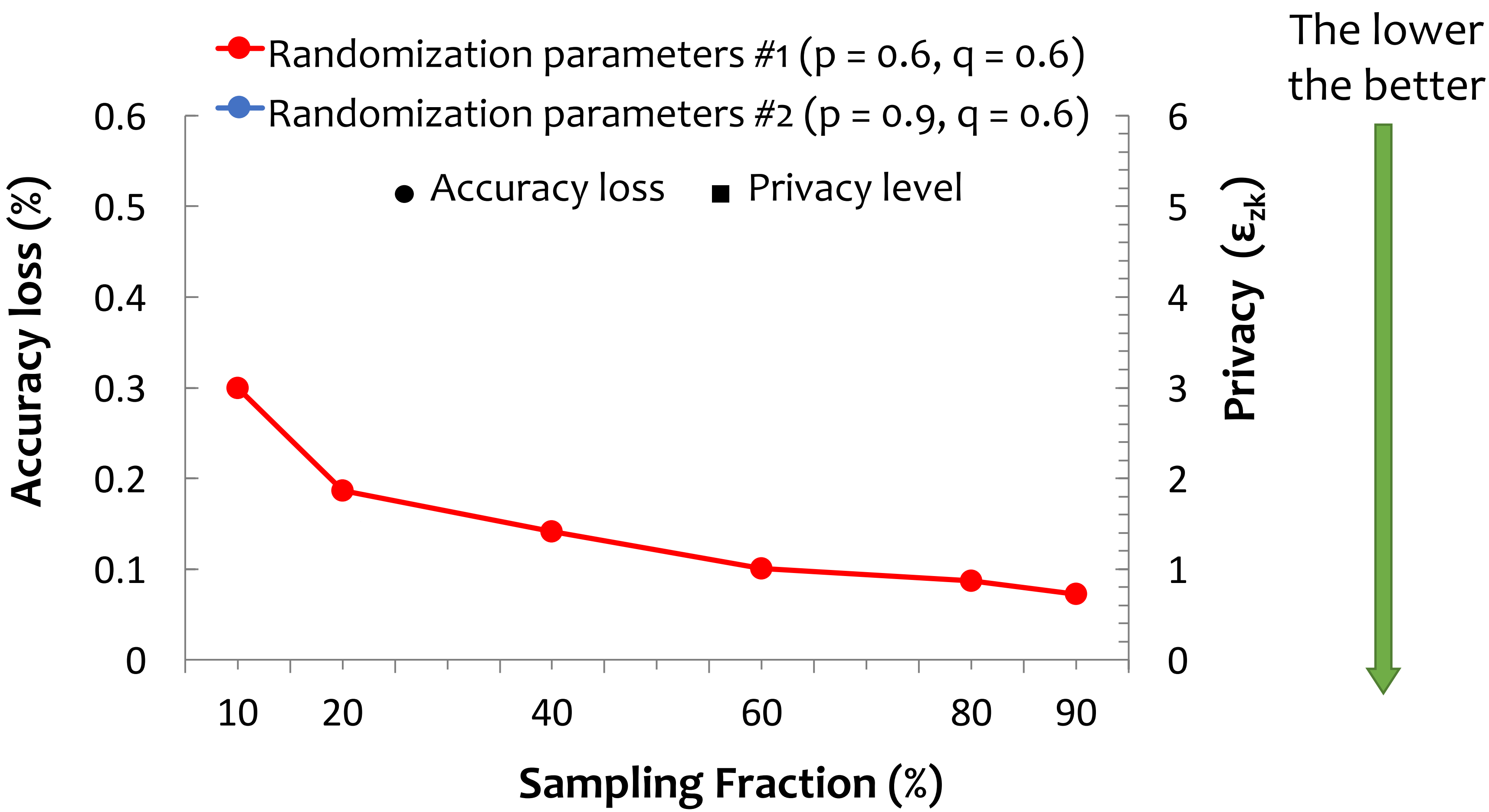


Accuracy vs privacy



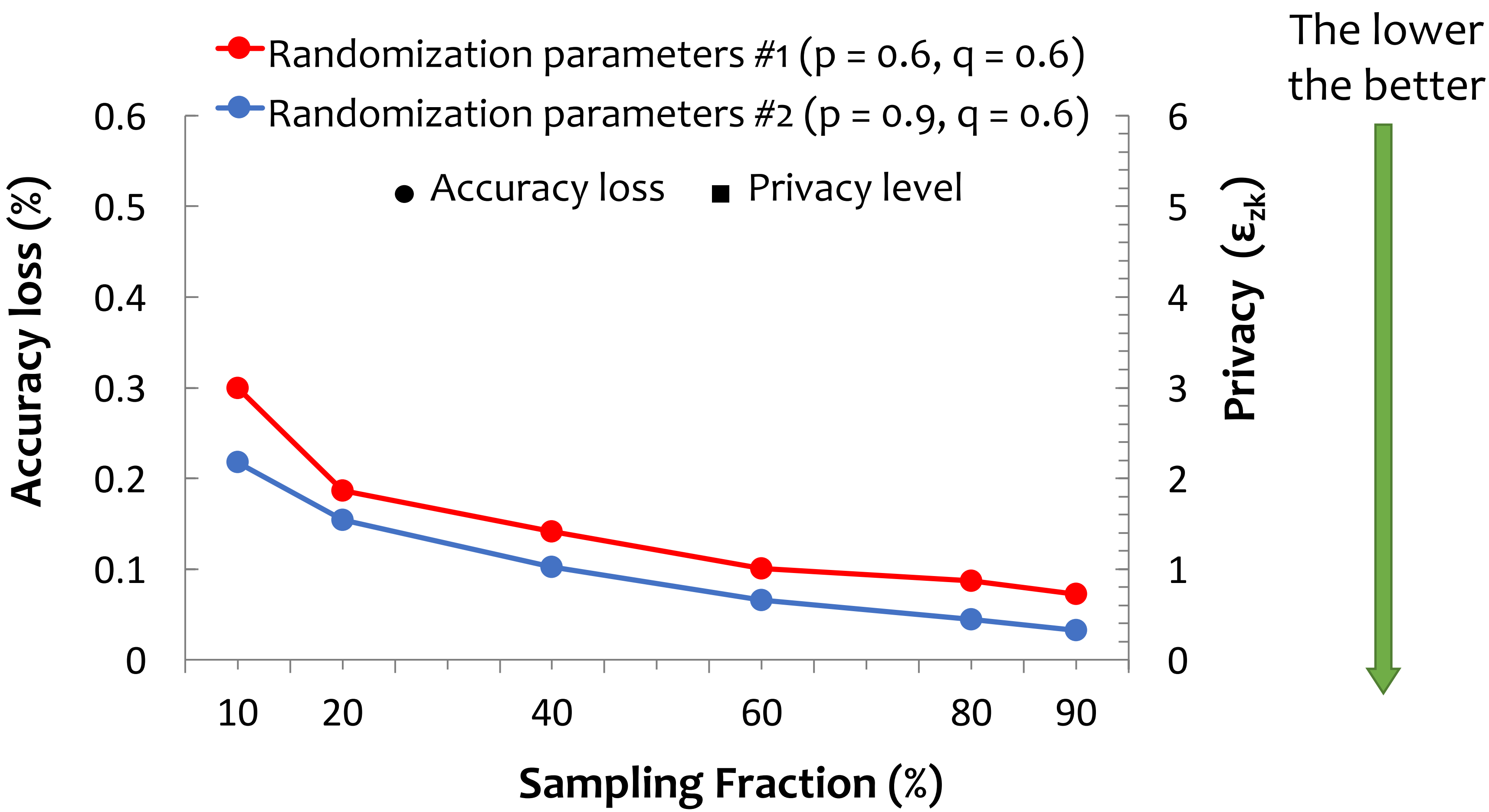
Trade-off between utility and privacy

Accuracy vs privacy



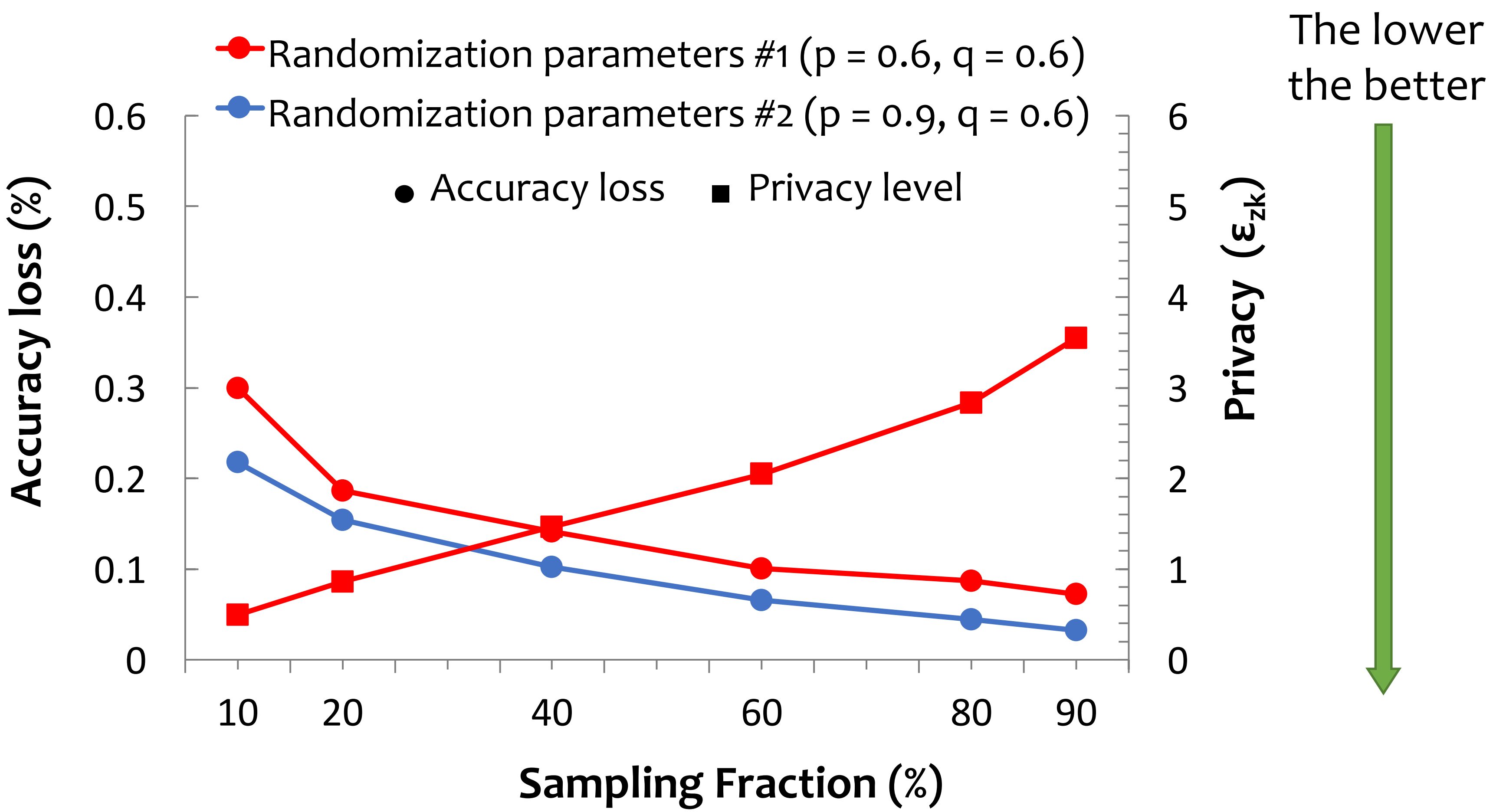
Trade-off between utility and privacy

Accuracy vs privacy



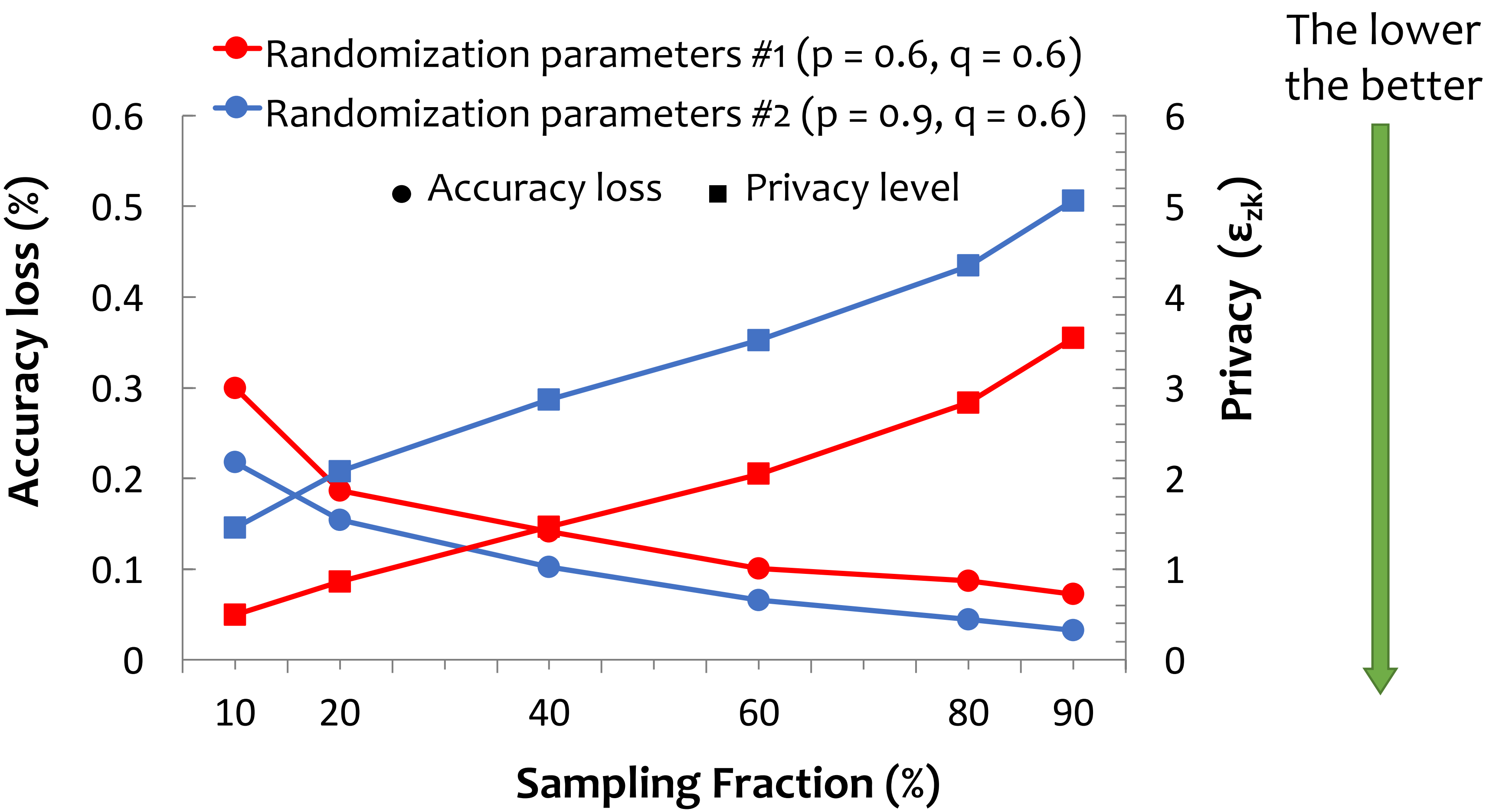
Trade-off between utility and privacy

Accuracy vs privacy



Trade-off between utility and privacy

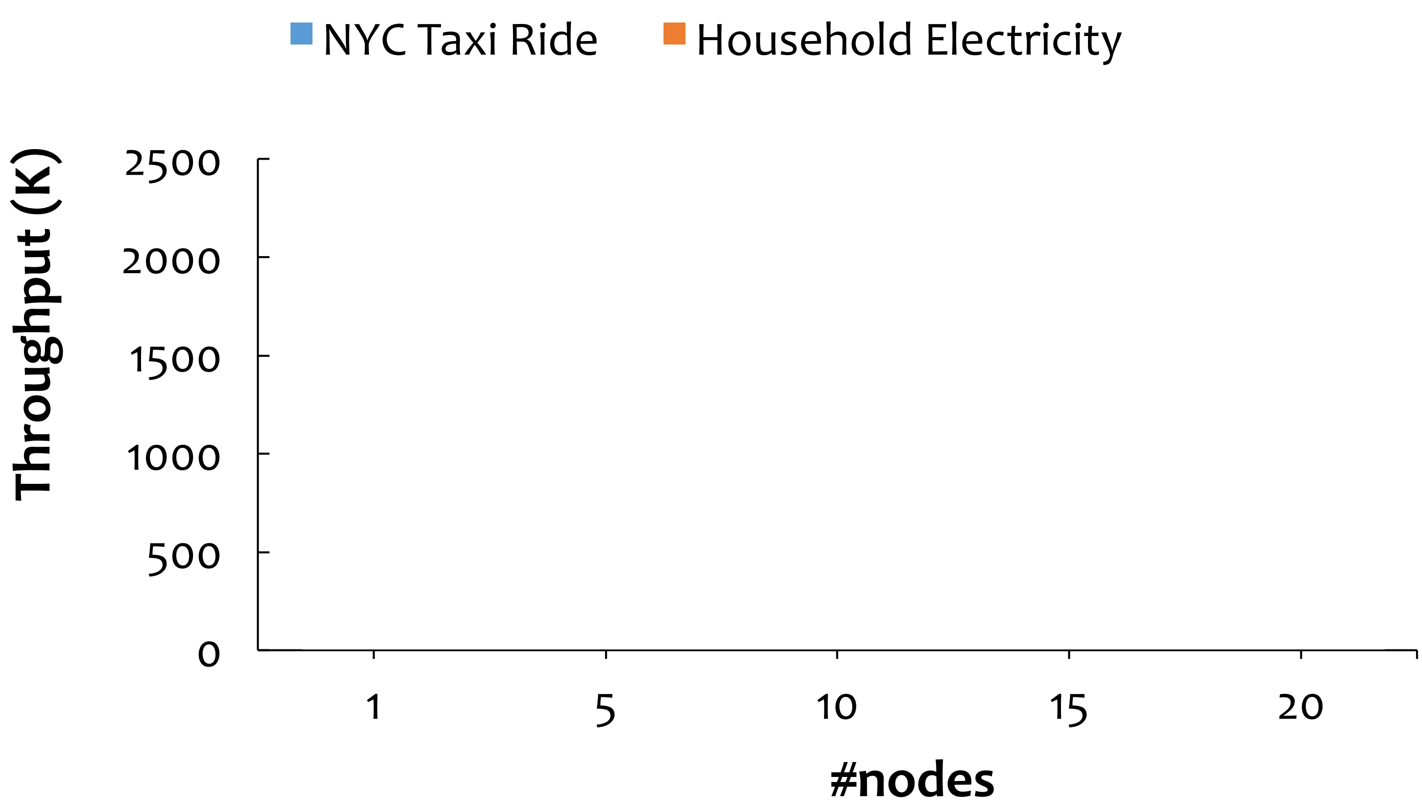
Accuracy vs privacy



Trade-off between utility and privacy

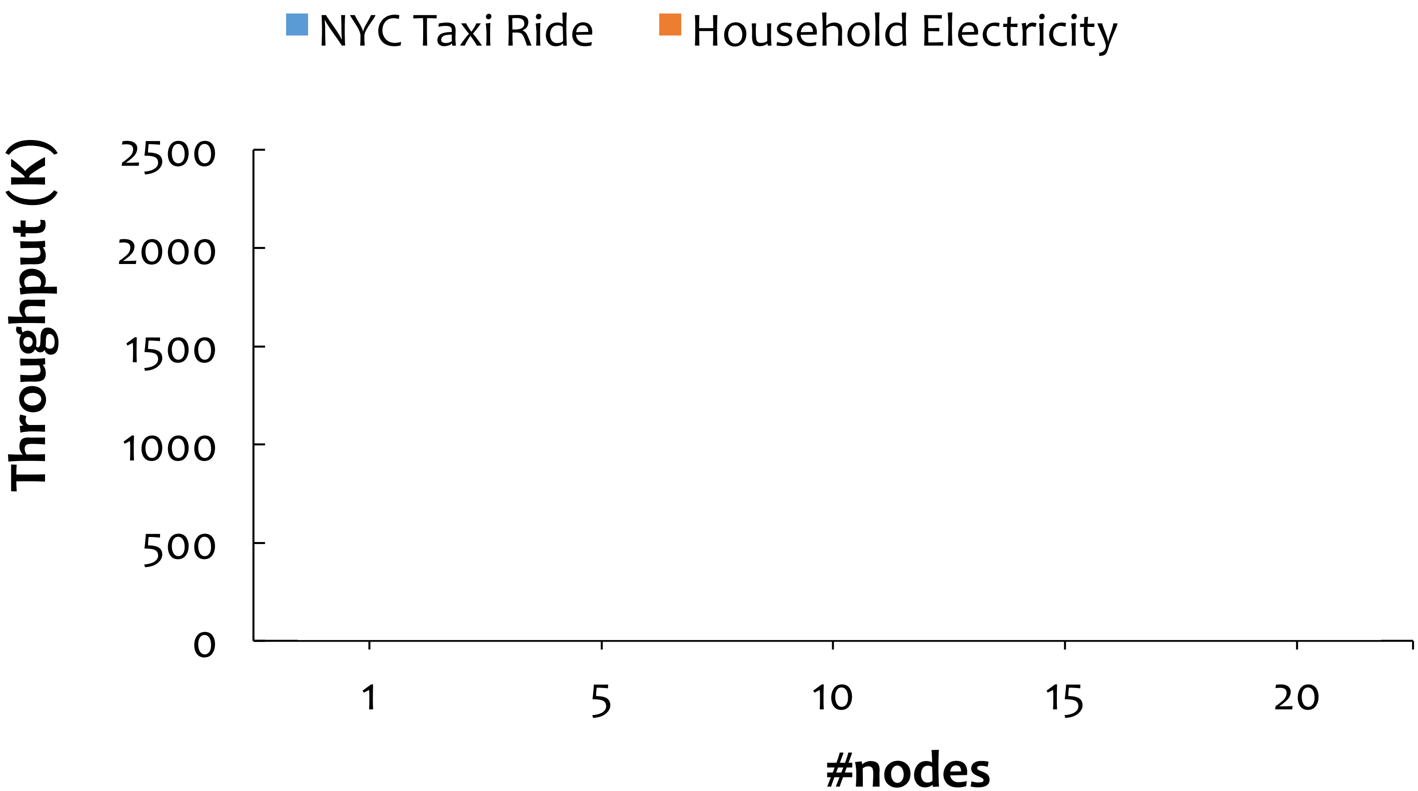
Throughput

Throughput

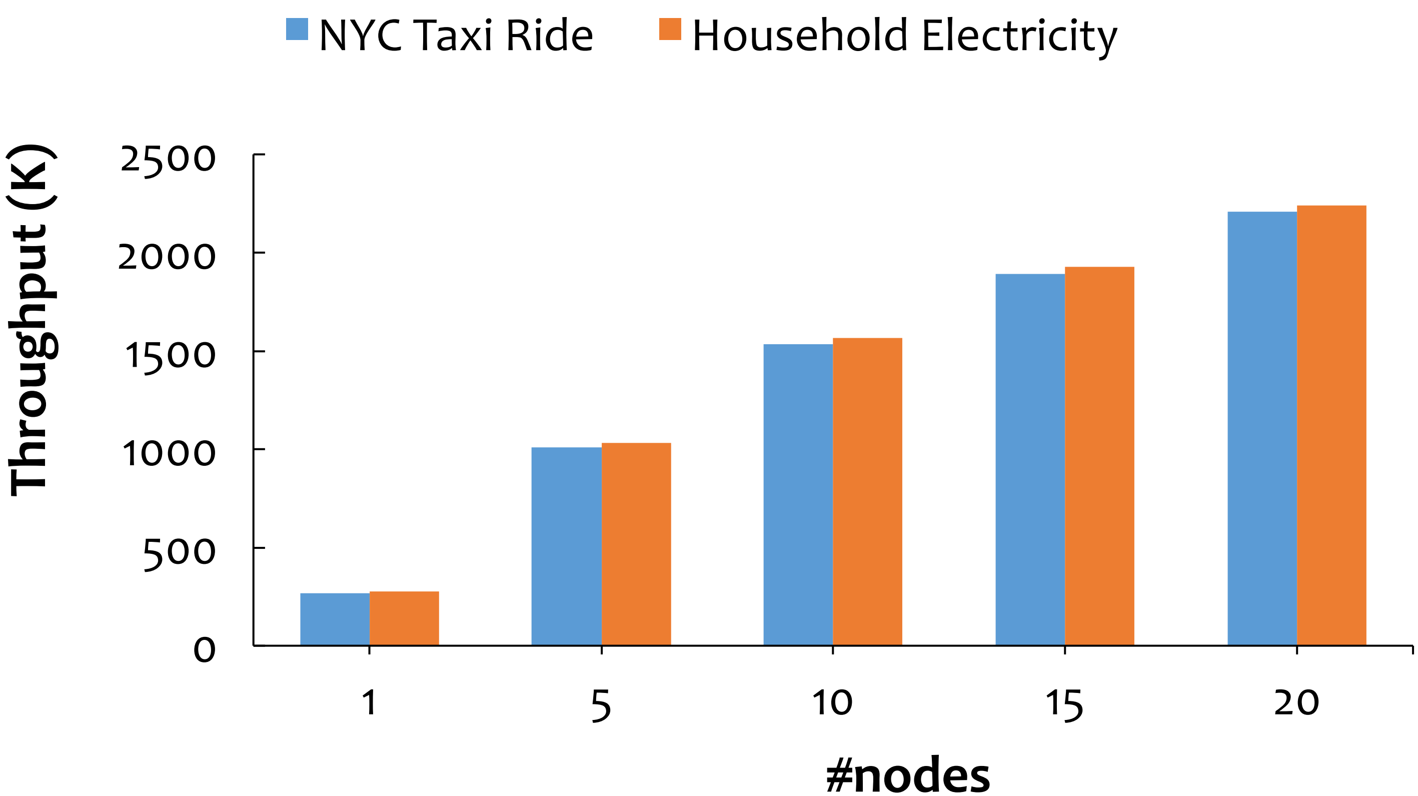

NYC Taxi Ride
Household Electricity
Throughput (K)
2500
2000
1500
1000
500
0
1
5
10
15
20
#nodes

Throughput

The higher
the better



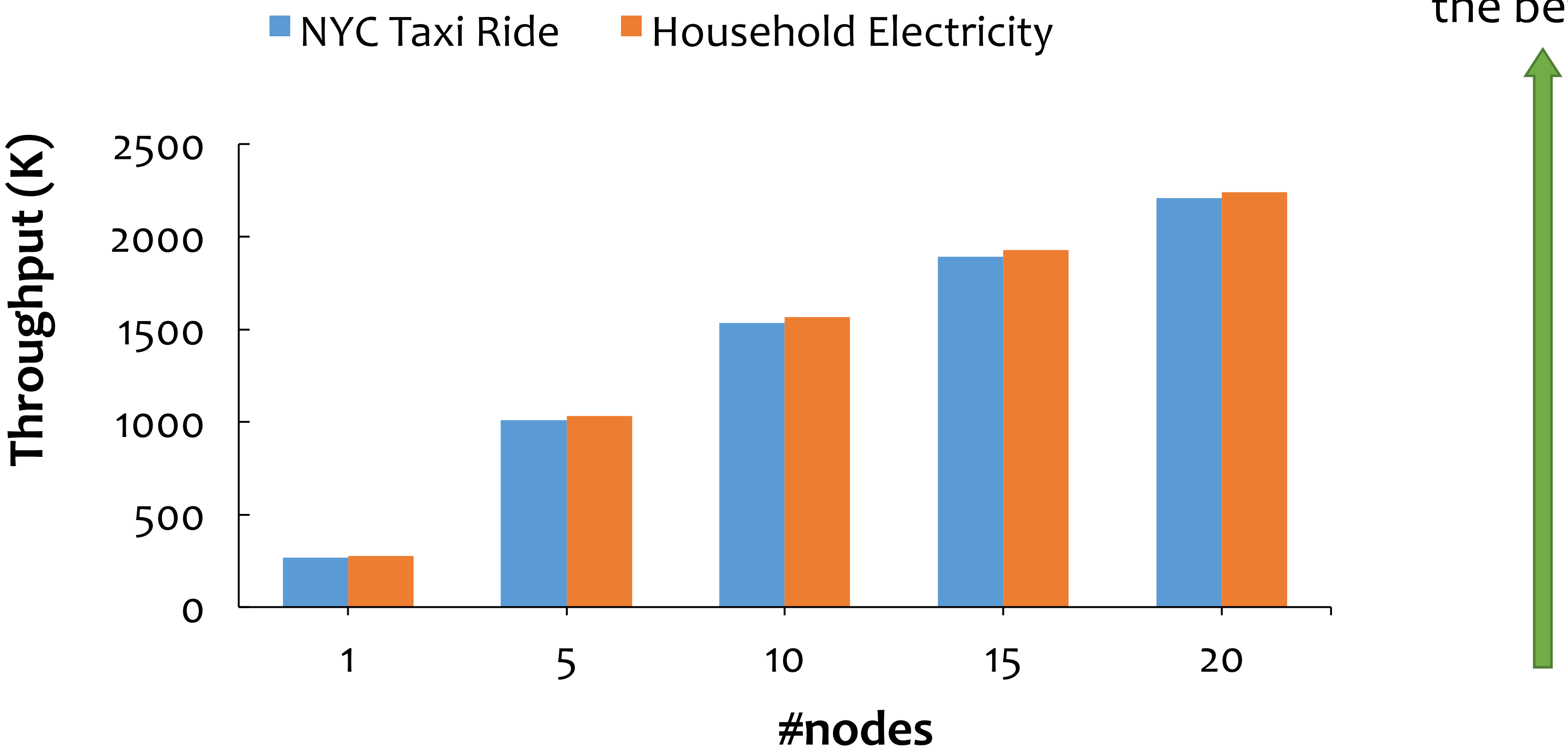
Throughput



The higher the better

Throughput

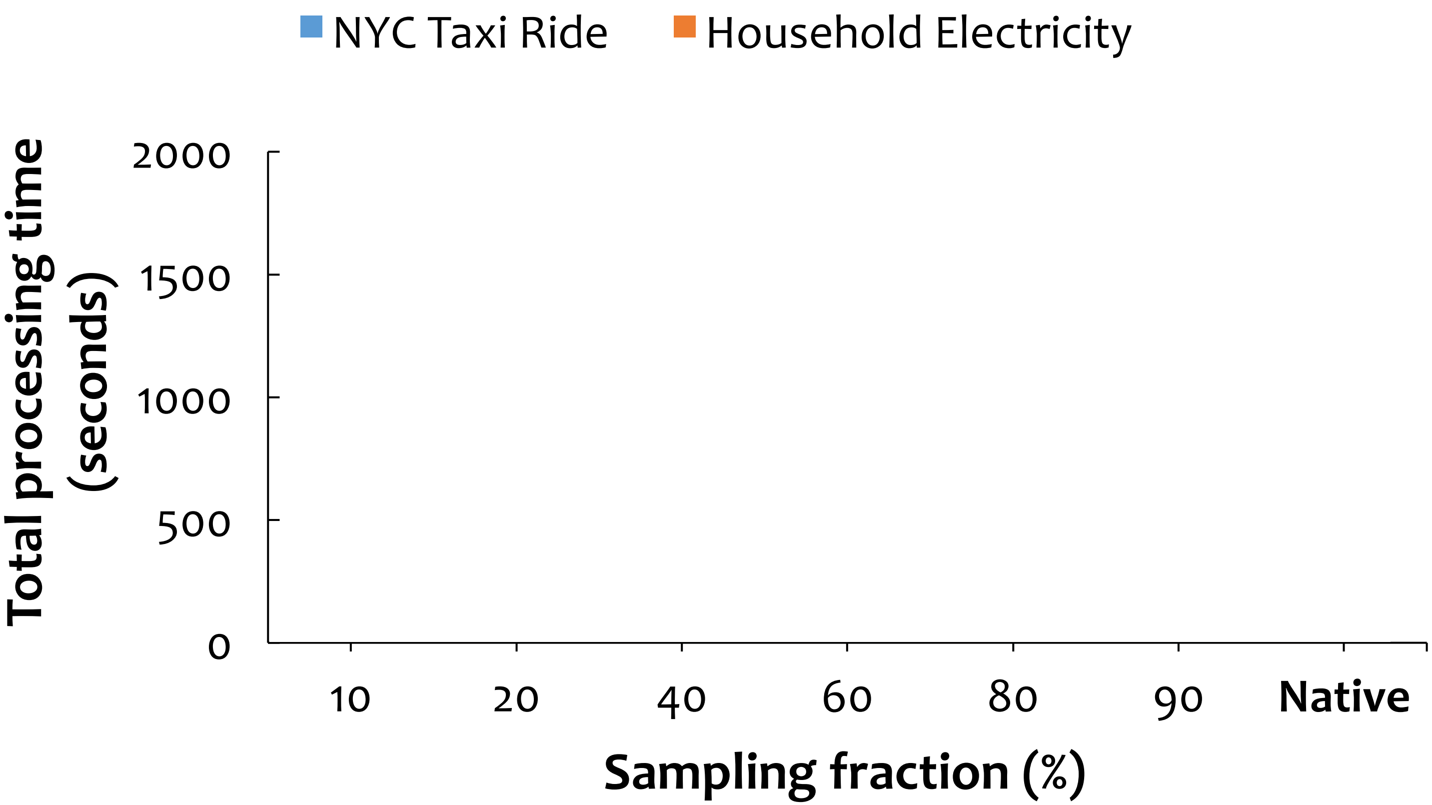
The higher the better



~8X speedup when going from one node to 20 nodes

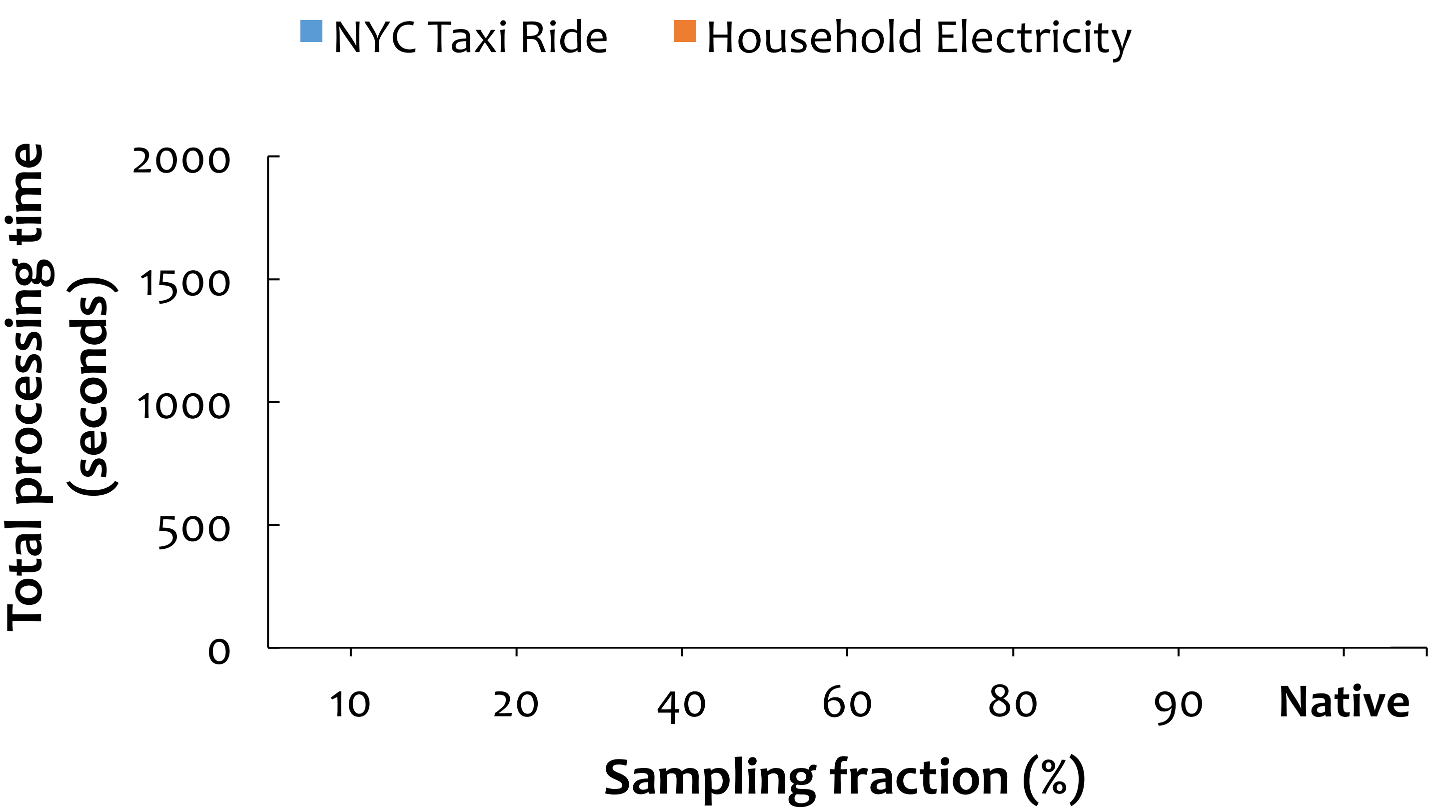
Latency

Latency



NYC Taxi Ride
Household Electricity
Total processing time (seconds)
2000
1500
1000
500
0
10
20
40
60
80
90
Native
Sampling fraction (%)

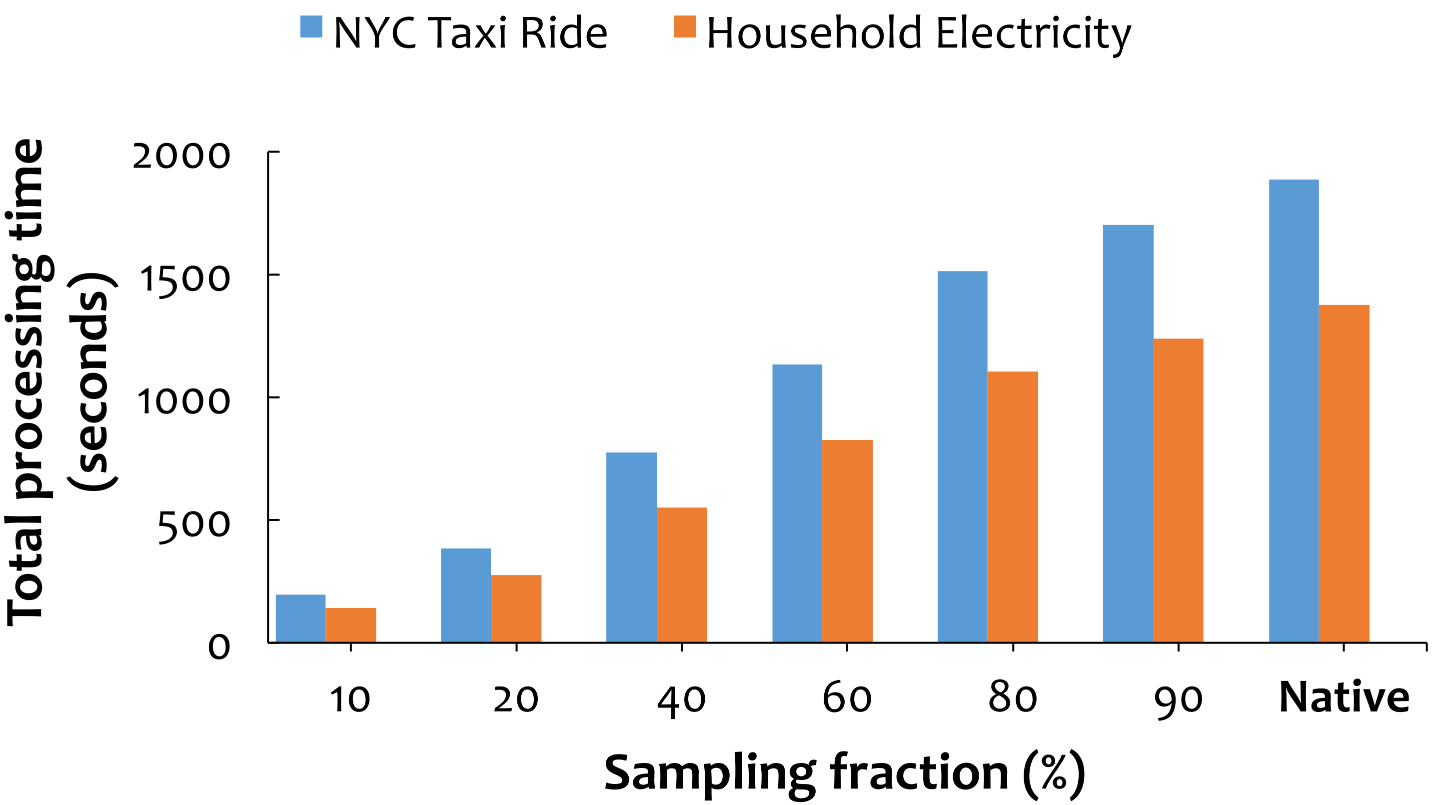
Latency



The lower the better

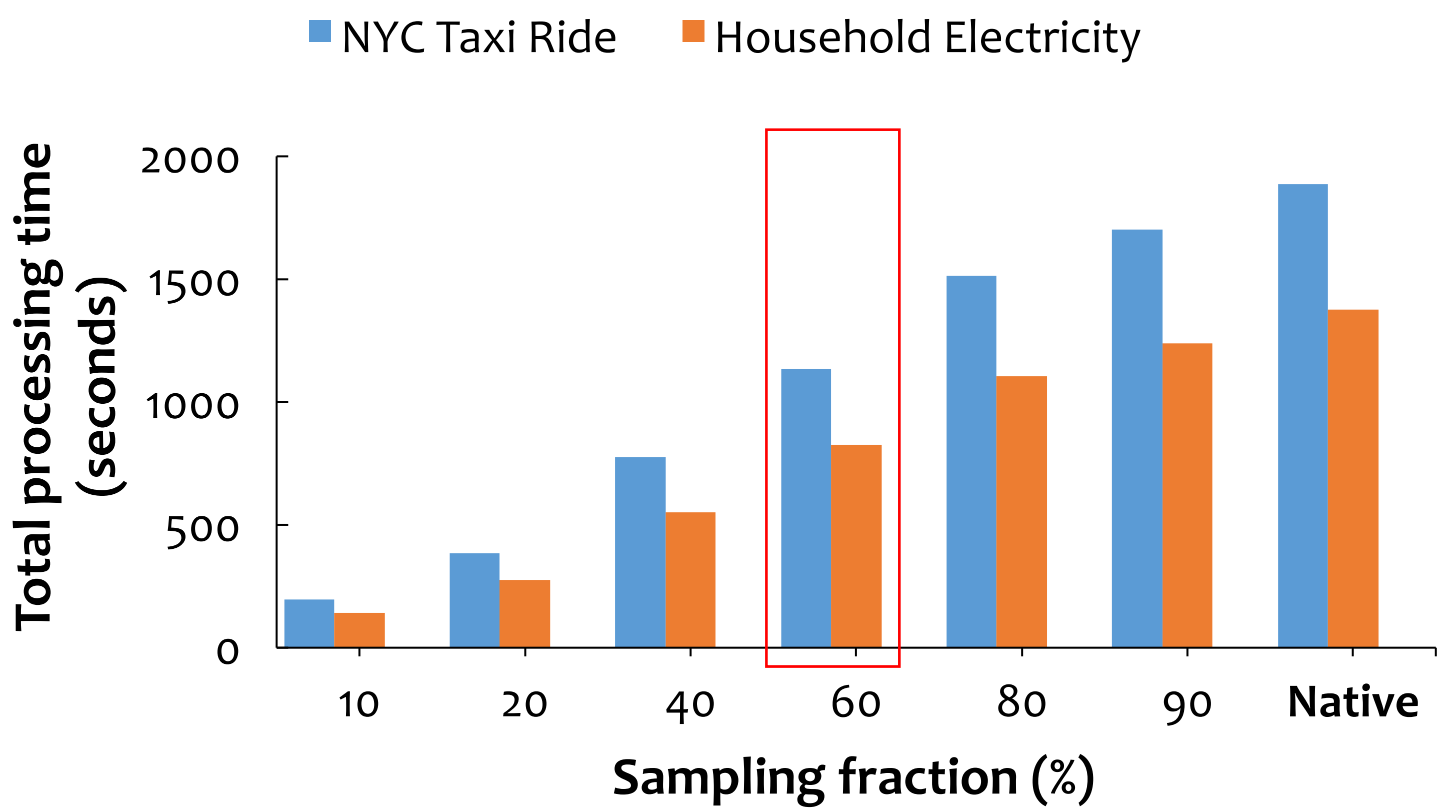
Latency

The lower the better



Latency

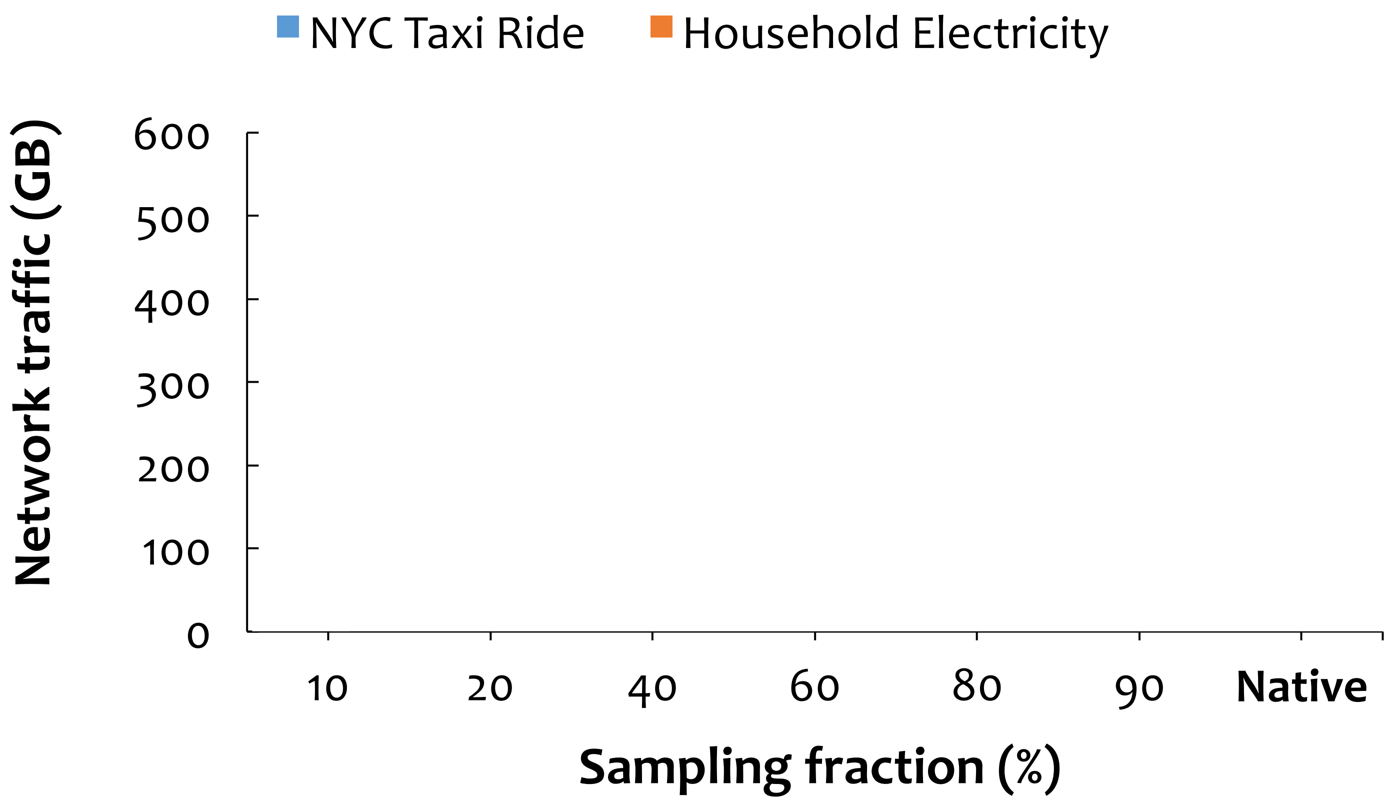
The lower the better



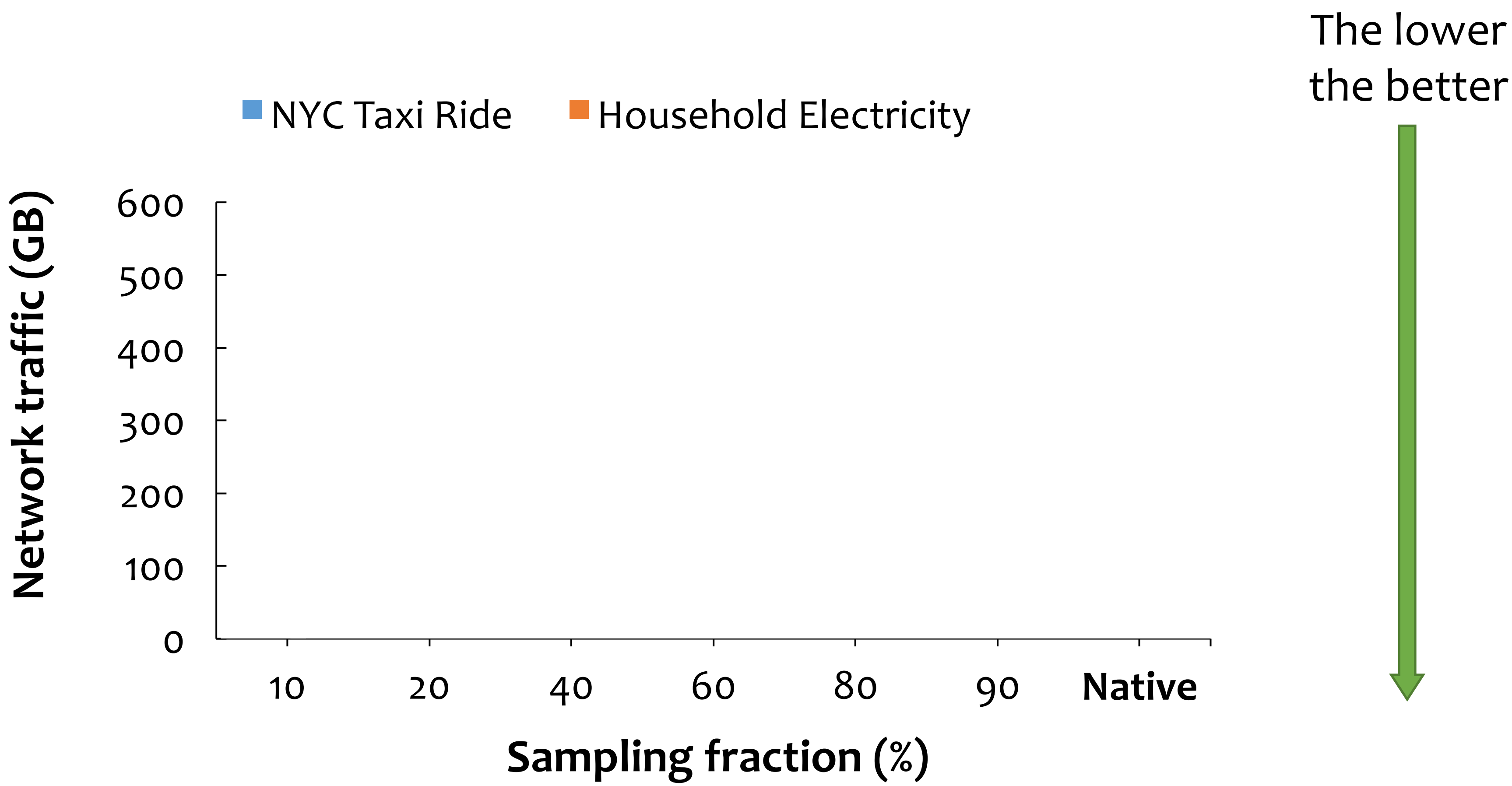
~1.66X lower than the native execution with sampling fraction of 60%

Network overhead

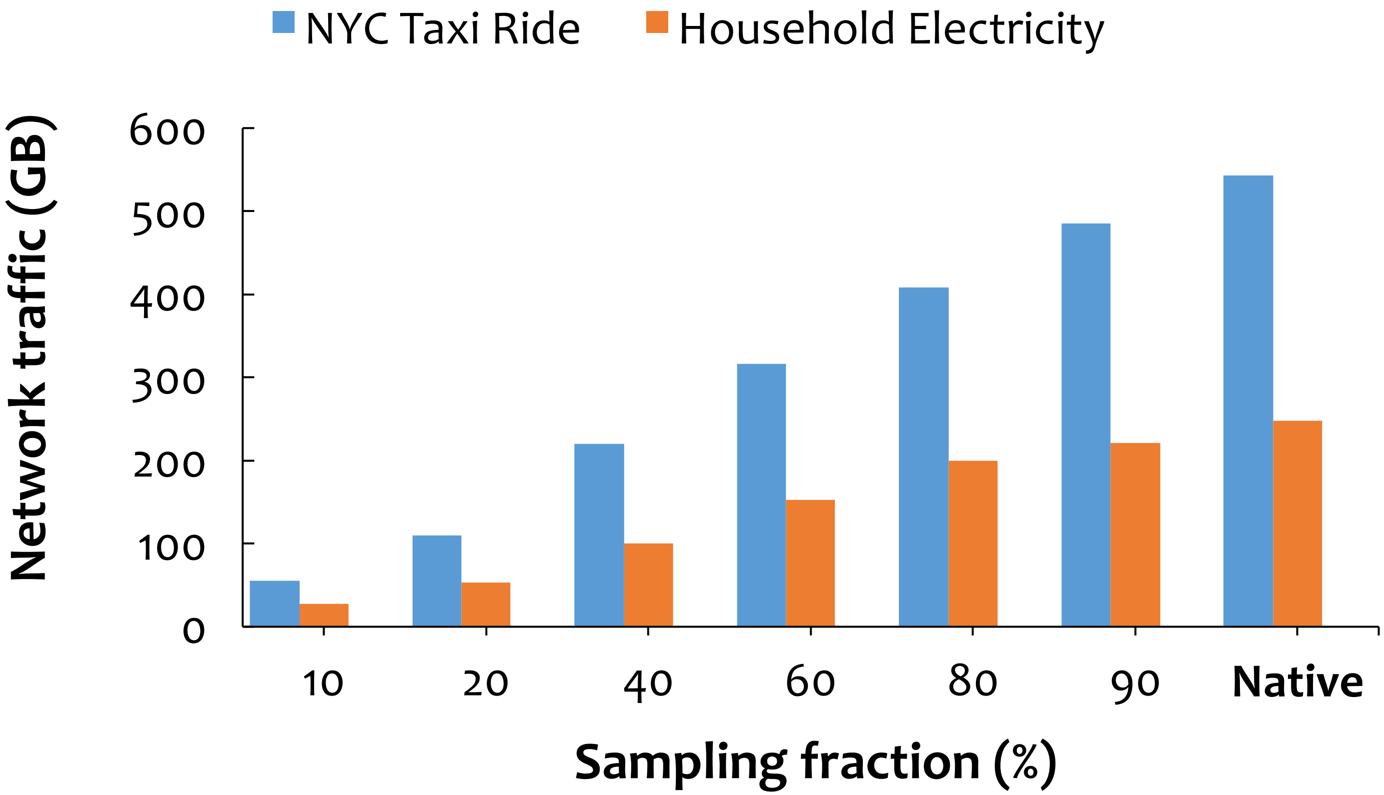
Network overhead



Network overhead

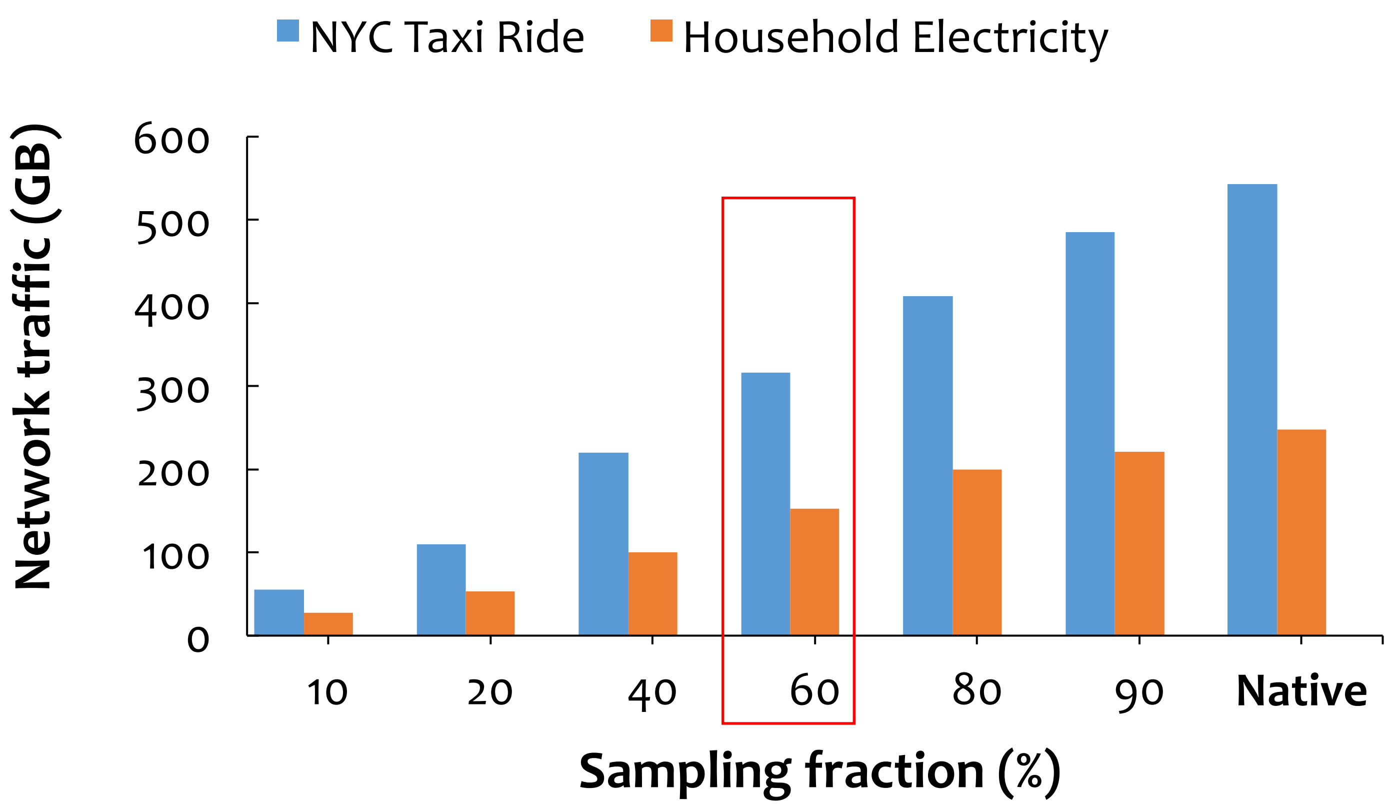


Network overhead



The lower the better

Network overhead



The lower the better

~1.6X lower than the native execution with sampling fraction of 60%

Conclusion

PrivApprox: a privacy-preserving stream analytics system over distributed datasets

Conclusion

PrivApprox: a privacy-preserving stream analytics system over distributed datasets

Privacy	Zero-knowledge privacy

Conclusion

PrivApprox: a privacy-preserving stream analytics system over distributed datasets

Privacy	Zero-knowledge privacy
Practical	Adaptive execution based on query budget

Conclusion

PrivApprox: a privacy-preserving stream analytics system over distributed datasets

Privacy	Zero-knowledge privacy
Practical	Adaptive execution based on query budget
Efficient	Randomized response & sampling techniques

Conclusion

PrivApprox: a privacy-preserving stream analytics system over distributed datasets

Privacy	Zero-knowledge privacy
Practical	Adaptive execution based on query budget
Efficient	Randomized response & sampling techniques

Thank you!

https://privapprox.github.io