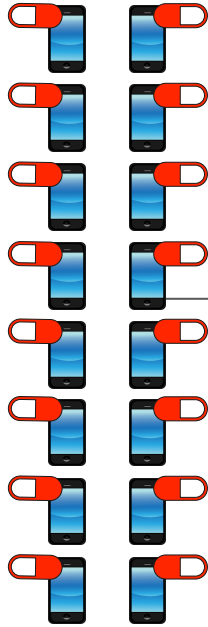


# Arboretum: A Planner for Large-Scale Federated Analytics with Differential Privacy

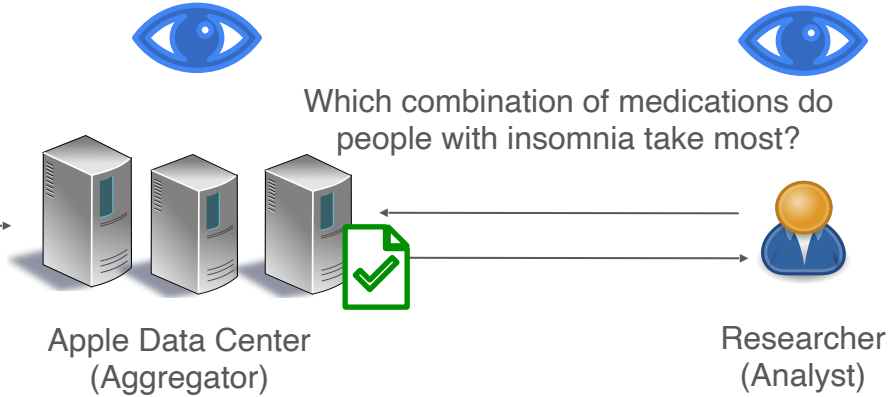
Eli Margolin, Karan Newatia, Tao Luo, Edo Roth  
University of Pennsylvania

Andreas Haeberlen  
University of Pennsylvania/Roblox

# Scenario - Health Data



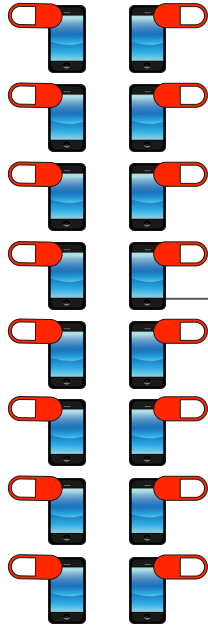
1.5 Billion iPhone Users



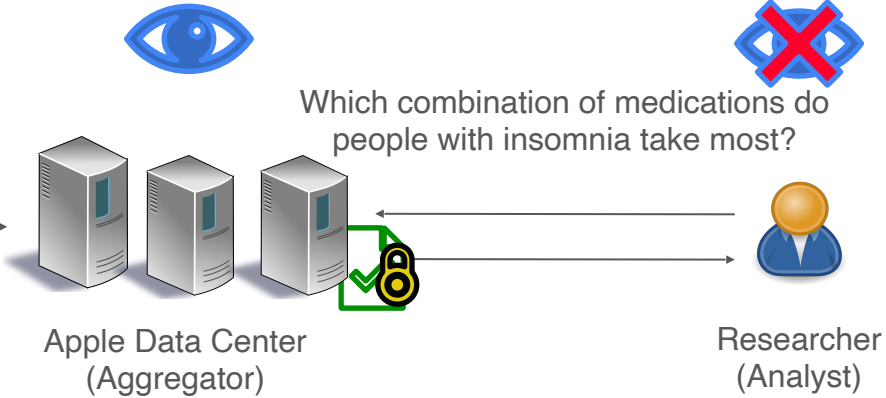
Users could just send their information to Apple to evaluate and publish a result

But then both Apple AND the researcher could learn about an individual's private data

# What about using Differential Privacy?



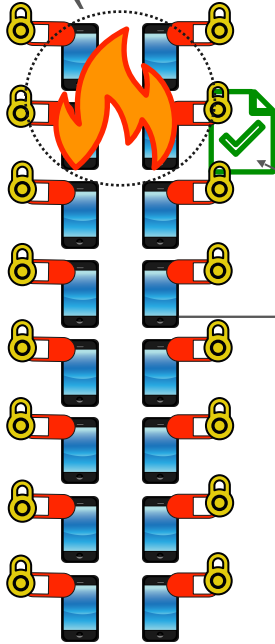
1.5 Billion iPhone Users



Users could just send their information to Apple to publish a Differentially Private result

**Protects user privacy at publication but not at computation**

# Orchard (OSDI'20)



1.5 Billion iPhone Users



Which combination of medications do people with insomnia take most?



Apple Data Center  
(Aggregator)



Researcher  
(Analyst)

Enlists a single committee of users to privately compute the result

New queries requires writing new plans by hand  
This type of query at this scale would overload the committee

## We want federated analytics

- with **privacy** at computation and publication
- for a **wide range** of queries

## And we don't want

- to have to **write new plans** for every new query
- to **overwhelm any party**

## Key Insight #1 - Offloading

All the users not doing anything in Orchard - we can offload some of the compute to them

But then analysts writing queries have to figure out the best way to do so, without needing a cryptographer...

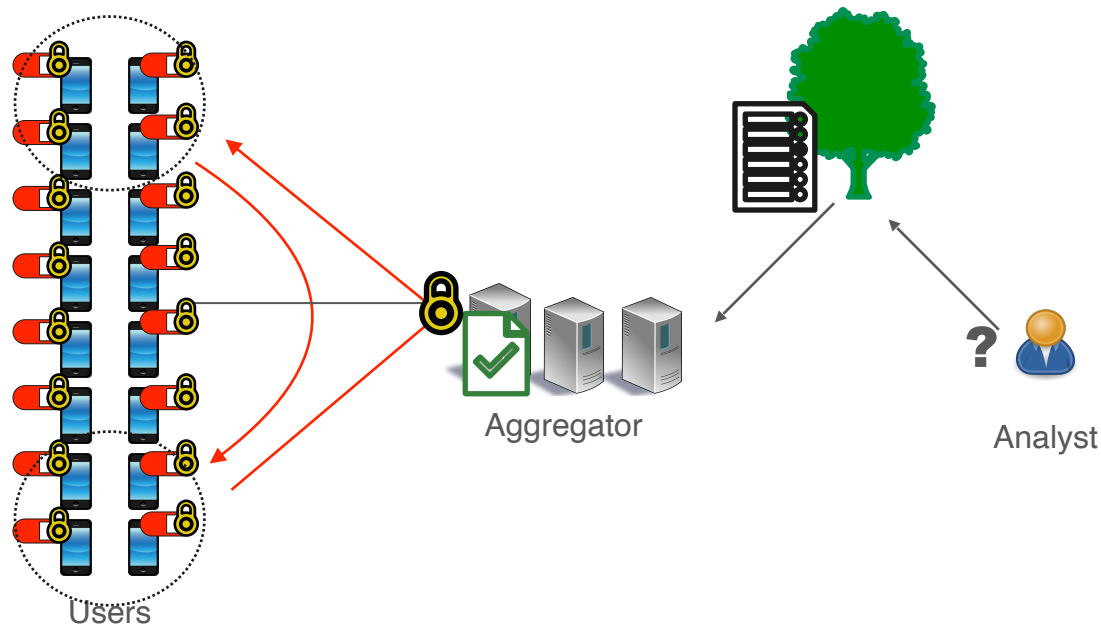
## Key Insight #2 - Automated Planning

While hand-writing plans to answer queries is hard and takes time, automatically generating them can be fast and easy

We can find good custom protocols without needing subject matter expertise

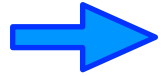
# Arboretum - A Query Planner for Private Federated Analytics

- Analyst submits a query
- Arboretum generates/  
scores potential plans
- Best plan is executed by  
**any/all** entities
- Differentially Private result  
is published





# Roadmap



## Background

Arboretum

Writing and Compiling the Query

Assigning Computation and Scoring

Plan Execution

Evaluation

Summary

# Background - The Exponential Mechanism

- A way to answer **categorical** queries with Differential Privacy
  - e.g. What is the most populous zip code
- Challenges
  - Calculate a score  $q$  for **every element in the domain**
  - Exponentiate every  $q$
  - **Both of these can be expensive**

# Background - Useful Privacy Tools

## Pros

## Cons

Multi-Party Computation  
(MPC)

Evaluate a wide variety of  
functions

Poor scalability  
Interactive

Additively Homomorphic  
Encryption  
(AHE)

Cheap

Linear operations only

Fully Homomorphic  
Encryption  
(FHE)

Supports non-linear  
operations  
Non-Interactive

Expensive

# Writing and Compiling our Query

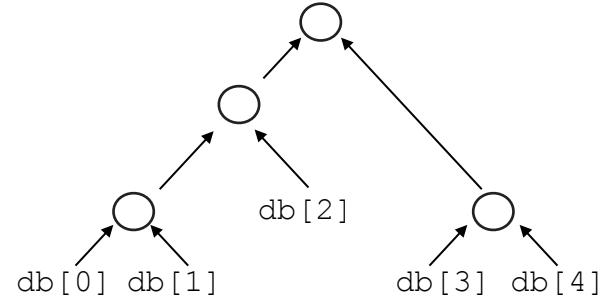
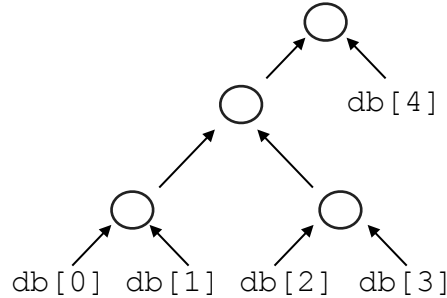
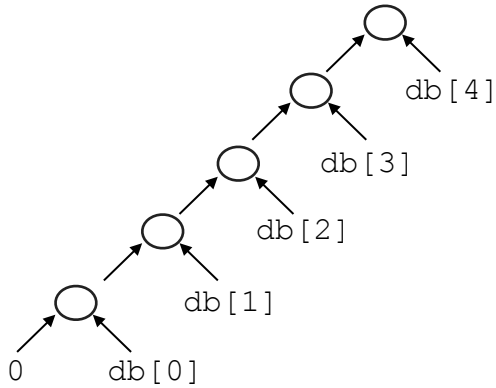
**Which combination of medications do people with insomnia take most?**

```
aggr = sum(db);  
best = em(aggr);  
output(best);
```

- Write the query in an imperative DSL
- Functions in Arboretum can be implemented in multiple ways
- Have to consider all combinations of them

# Compiling functions different ways - Toy Example

```
db = [1, ..., 5];  
aggr = sum(db);
```



**+** Good for a powerful Aggregator

**-** Parallelizes poorly

**+** Parallelize well

**-** More work for users, network costs

Each option = different distributed assignments & parallelization

# Assigning computation

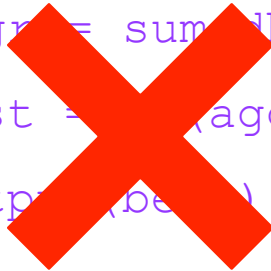
- Different entities can do different computation with different encryption
- Arboretum breaks the plan into short sequences of consecutive statements of the query
- Each one represents a single entity/cryptographic assignment

<code>aggr = sum(db);</code>	→	Aggregator w/ AHE
<code>best = em(aggr);</code>	→	MPC
<code>output(best);</code>	→	Aggregator w/ FHE

# Why is assigning hard?

## AHE

```
aggr = sum(db);  
best = argmax(aggr);  
output = best;
```



Need to be able to do more than addition for

EM

## FHE

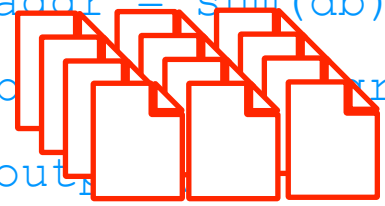
```
aggr = sum(db);  
best = argmax(aggr);  
output = best;
```



Works - but would take years of compute for even the aggregator

## MPC

```
aggr = sum(db);  
best = argmax(aggr);  
output = best;
```



Works - but committee would have to download GBs of data

# What about combining protocols?

AHE

```
aggr = sum(a);
```

FHE

```
best = enc(m(a));
```

```
output = dec(best);
```

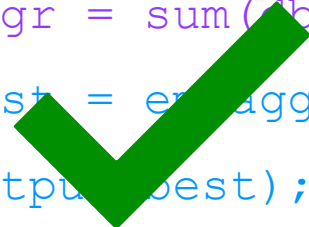


Works - but same issue as  
before **and** requires  
re-encrypting

```
aggr = sum(a);
```

```
best = enc(aggr);
```

```
output = dec(best);
```



AHE

MPC

Best - doesn't  
overload committees  
or aggregator

Now do this for every discrete operation once all the functions have been compiled down...



# Scoring

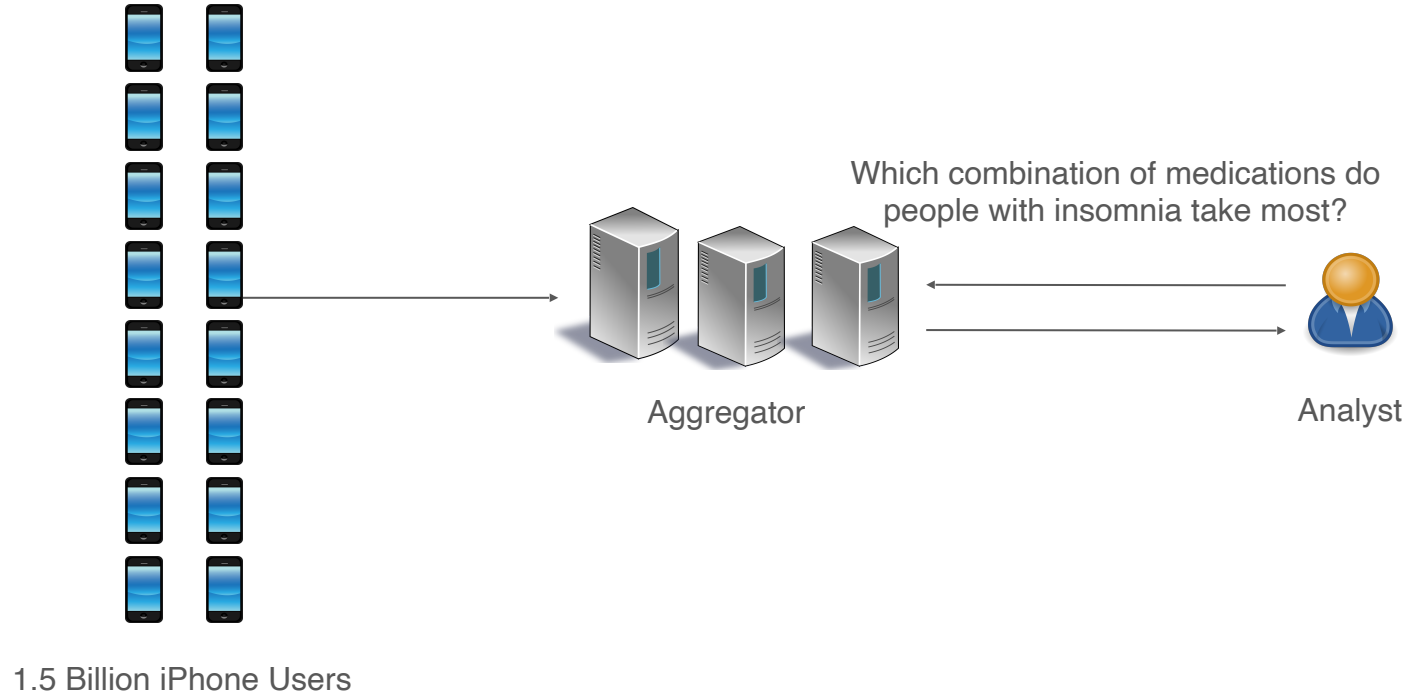
AHE `aggr = sum(db);`

~~FHE `best = em(aggr);`~~

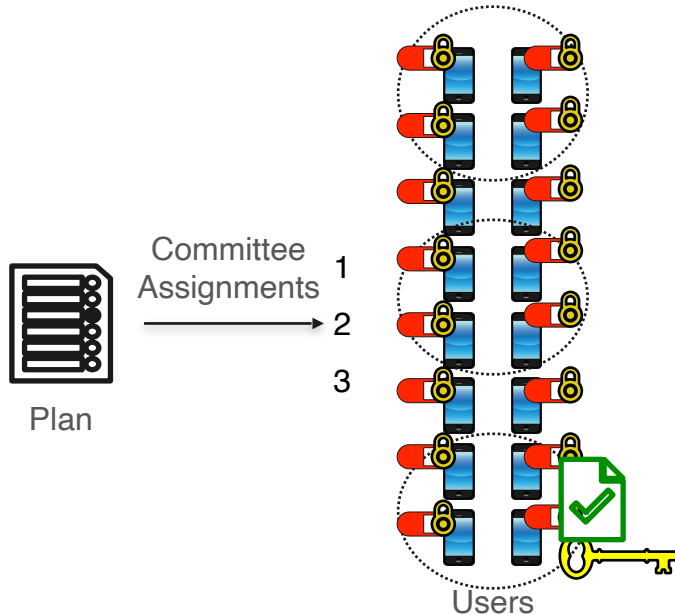
MPC `best = em(aggr);`  
`output(best);`

- Can't expect to pick the *best* plan
- Want to weed out bad plans as quickly as possible
  - Recognize quickly that starting in AHE and re-encrypting into FHE is bad
- Use a simple cost model
  - Customize with new primitives using automated system e.g. CostCO (EuroS&P'22)

# Our Most Common Medications Example...



# Plan Execution



**Set Up** - Users are assigned to committees, including one that generates an encryption key

**Input** - All users encrypt and upload their query responses

**Analysis** - Analytics happens between entities

**Release** - A final committee decrypts and publishes the Differentially Private result



# Roadmap

Background

Arboretum

Writing and Compiling the Query

Assigning Computation and Scoring

Plan Execution



Evaluation

Summary

# Evaluation

What new queries can be supported?

How expensive is the planning?

How do Arboretum's plans compare to previous work's hand tailored solutions?

How expensive are the plans on average?

How expensive are the plans for committee members?

How expensive are the plans for the aggregator?

How do Arboretum and its plans scale?

# Evaluation

What new queries can be supported?

How expensive is the planning?

How do Arboretum's plans compare to previous work's hand tailored solutions?

How expensive are the plans on average?

How expensive are the plans for committee members?

How expensive are the plans for the aggregator?

How do Arboretum and its plans scale?



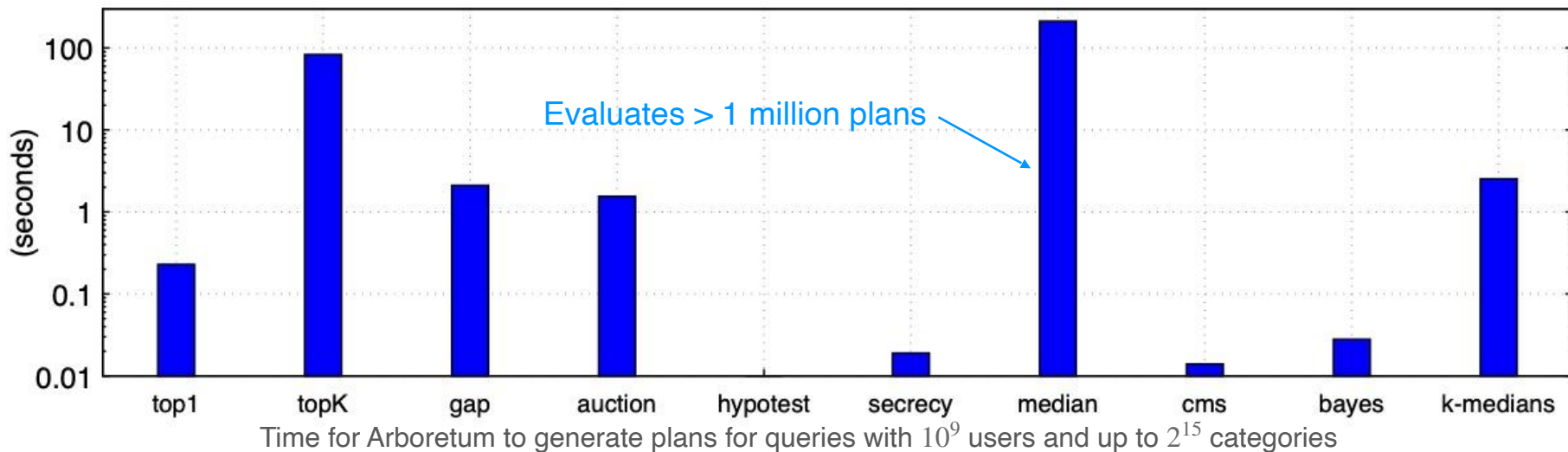
See  
paper

# Arboretum supports more queries and more categories

	<u>Query</u>	<u>Action</u>	<u>Lines of Code</u>
Newly supported classes of queries	top1	Most frequent item	3
	topK	Top-K selection	8
	gap	Gap between top 2	8
	auction	Unbounded auction	7
	hypotest	Hypothesis testing	12
	secrecy	Secrecy of the sample	16
	median	Median	39
Expanded supported # of categories from 10 to >30,000	cms	Count-mean sketch	5
	bayes	Naïve Bayes	16
	k-medians	K-Medians	30

Evaluated **new classes of queries** - but our DSL means we can support even more  
Expanded the number of categories supported by queries in existing work by **1000x**

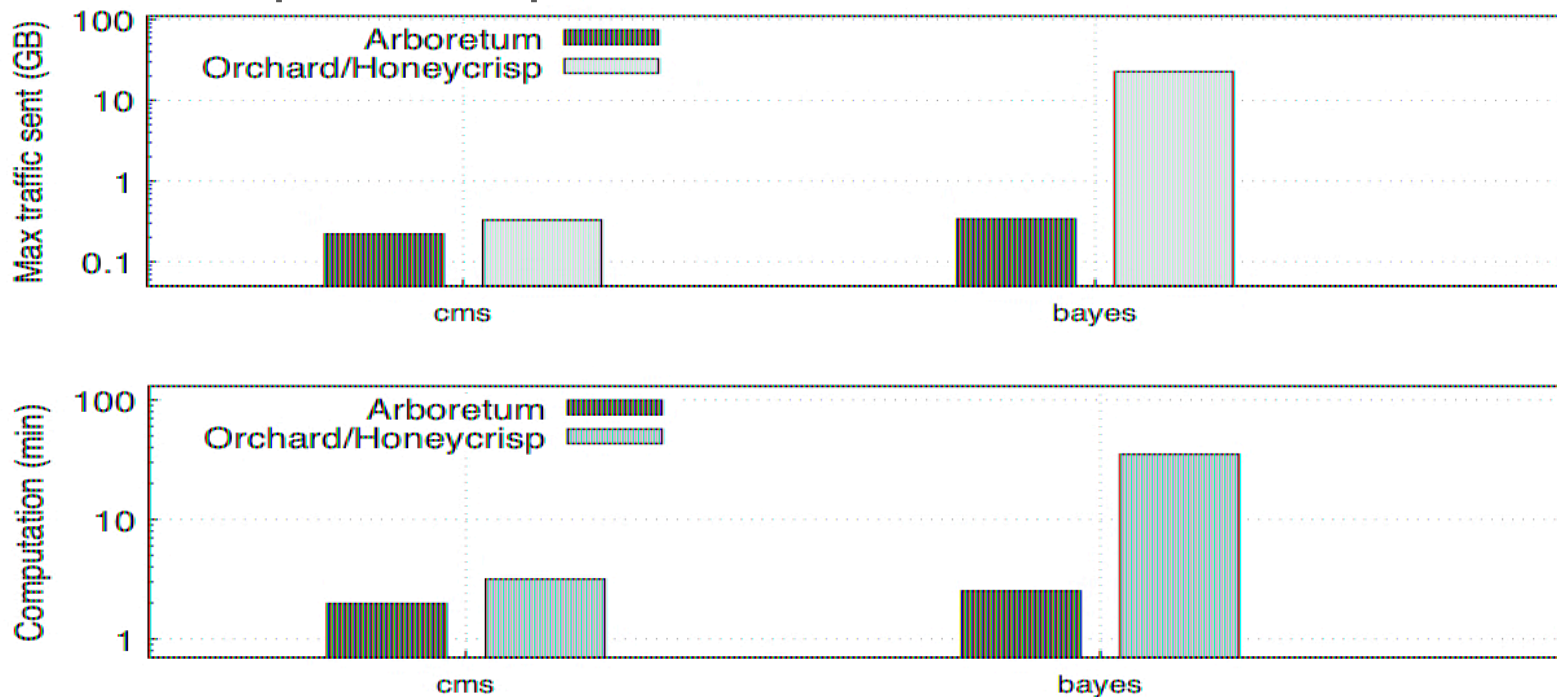
# Arboretum's Planning is Fast



Planning cost is consistently **under 4 minutes** vs doing it by hand (could take weeks!)  
Planning time is **< .1% of plan execution time**



# Arboretum's plans require less work for committee members



Maximum Traffic Sent and Compute Time for Committee Members - run for both Arboretum's and Orchard's plans

Not every user will be on a committee - this amount of work won't be done by everyone

# Summary

Thank you!

Contact: [ecmargo@seas.upenn.edu](mailto:ecmargo@seas.upenn.edu)

## **We want...**

- Federated analytics at a large scale for complex categorical queries
- With strong differential privacy guarantees

## **We don't want...**

- To plan this by hand
- Overwhelm any individual party

## **We can...**

- Have users contribute to the computation
- Explore plan space automatically

## **Arboretum!**

- Query planning for a wide range of differentially private queries
- Scales to billions of users
- Quickly finds plans that outperform hand tailored plans