A/B testing at Uber

A BYOM (Bring Your Own Metrics) Platform

Milène Darnis
Product Manager, Experimentation

Uber

1. Overview of A/B testing at Uber
2. Decoupling experimentation events from business metrics
3. Extending the platform
4. Future work
5. Conclusion

1. Overview of A/B testing at Uber
2. Decoupling experimentation events from business metrics
3. Extending the platform
4. Future work
5. Conclusion

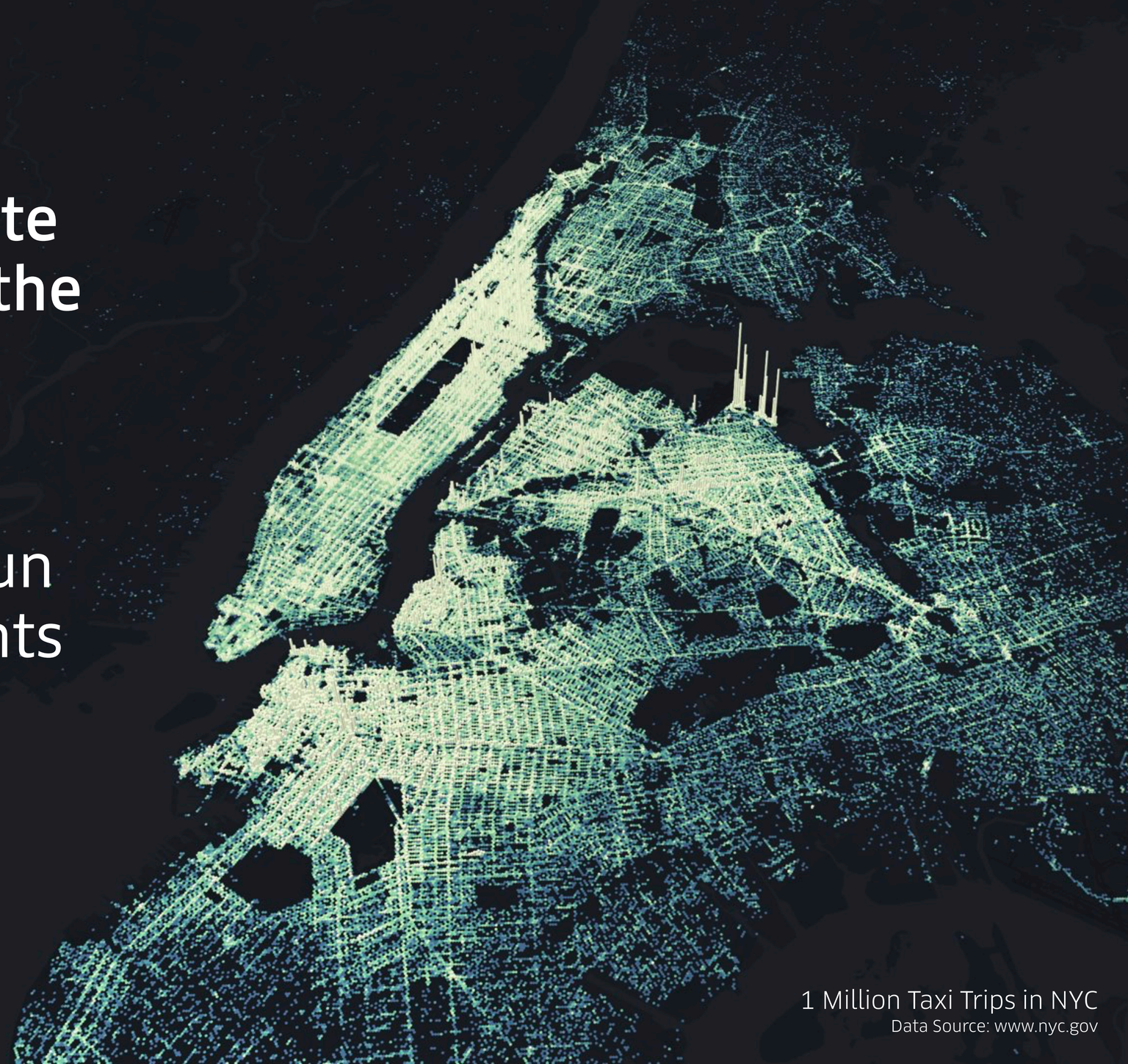
Uber’s mission is to ignite opportunity by setting the world in motion.

1 Million Taxi Trips in NYC

Data Source: www.nyc.gov

Uber’s mission is to ignite opportunity by setting the world in motion.

In order to do this, we run thousands of experiments every year.



1 Million Taxi Trips in NYC
Data Source: www.nyc.gov

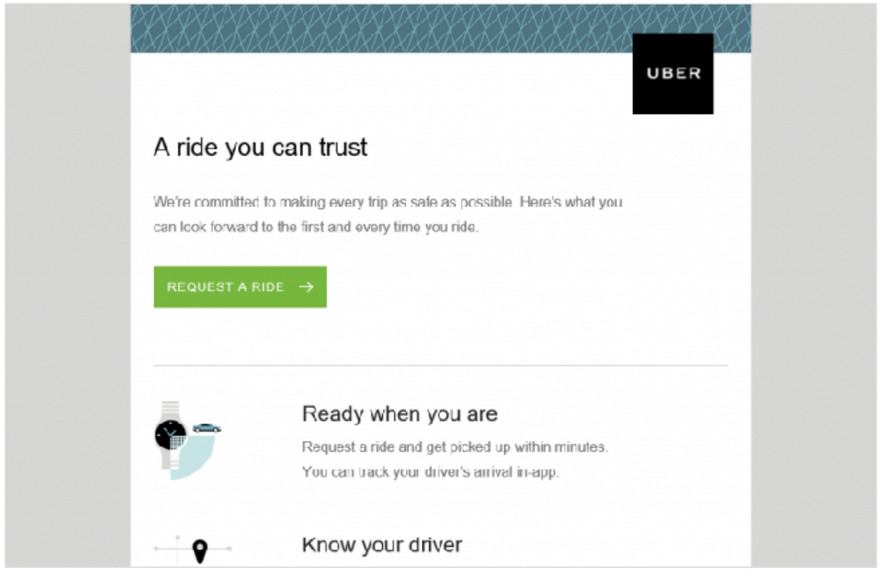
Where we experiment

Where we experiment

Backend
Python, Java, Go

Mobile
iOS, Android

Where we experiment



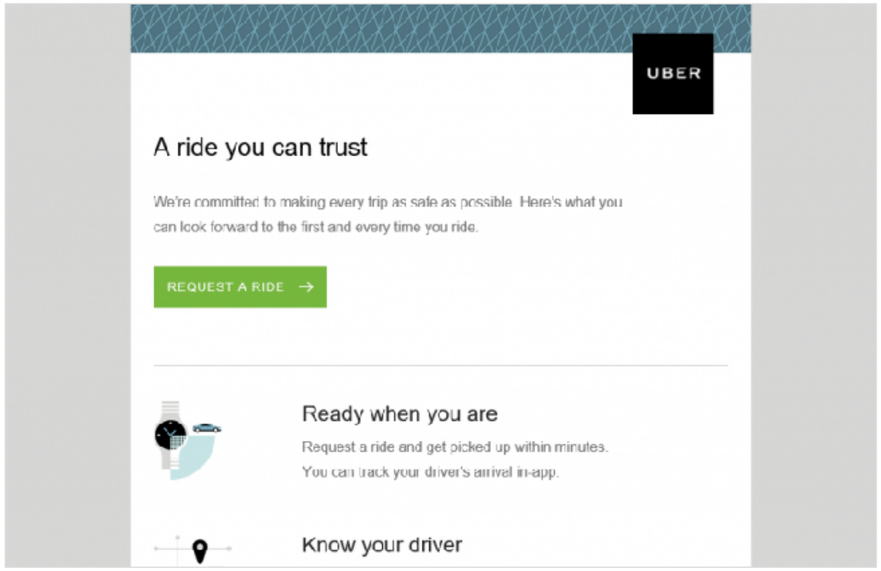
Backend

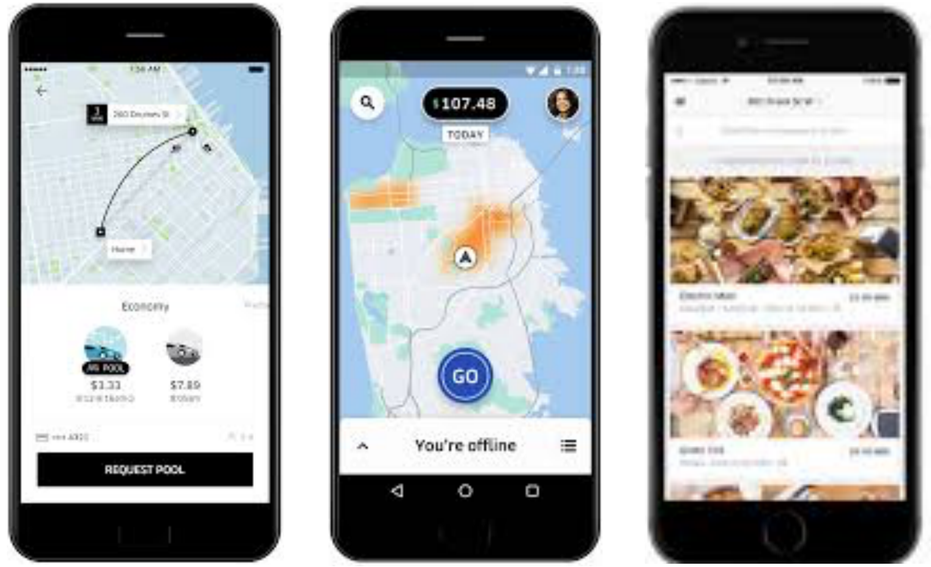
Python, Java, Go

Mobile

iOS, Android

Where we experiment

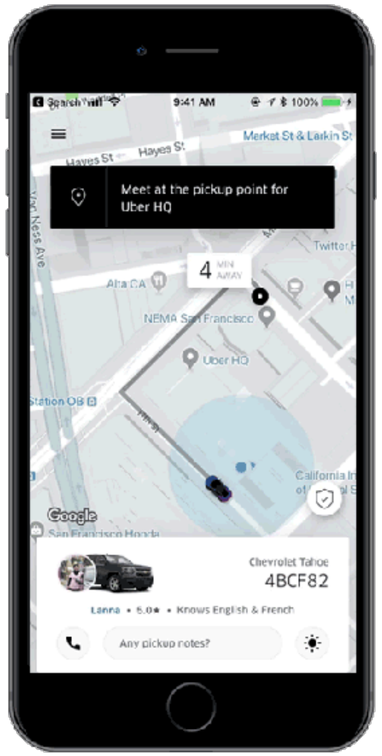




Backend
Python, Java, Go

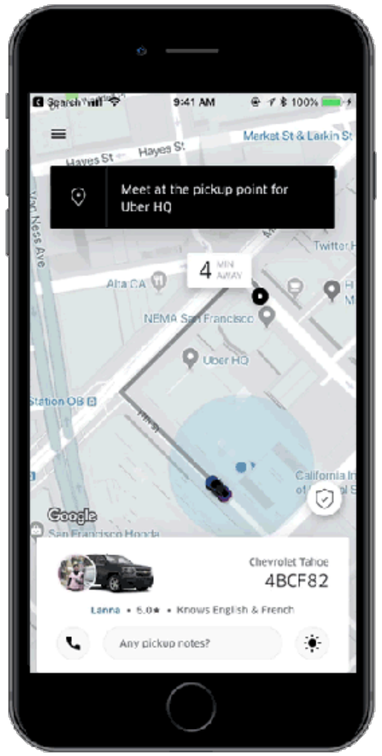
Mobile
iOS, Android

What we experiment on



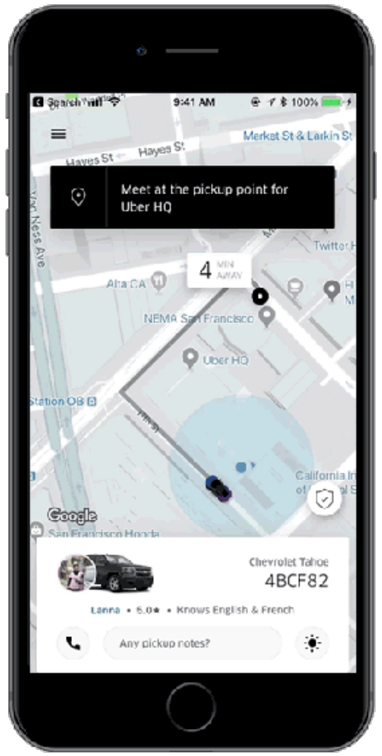
User facing features

What we experiment on



User facing features

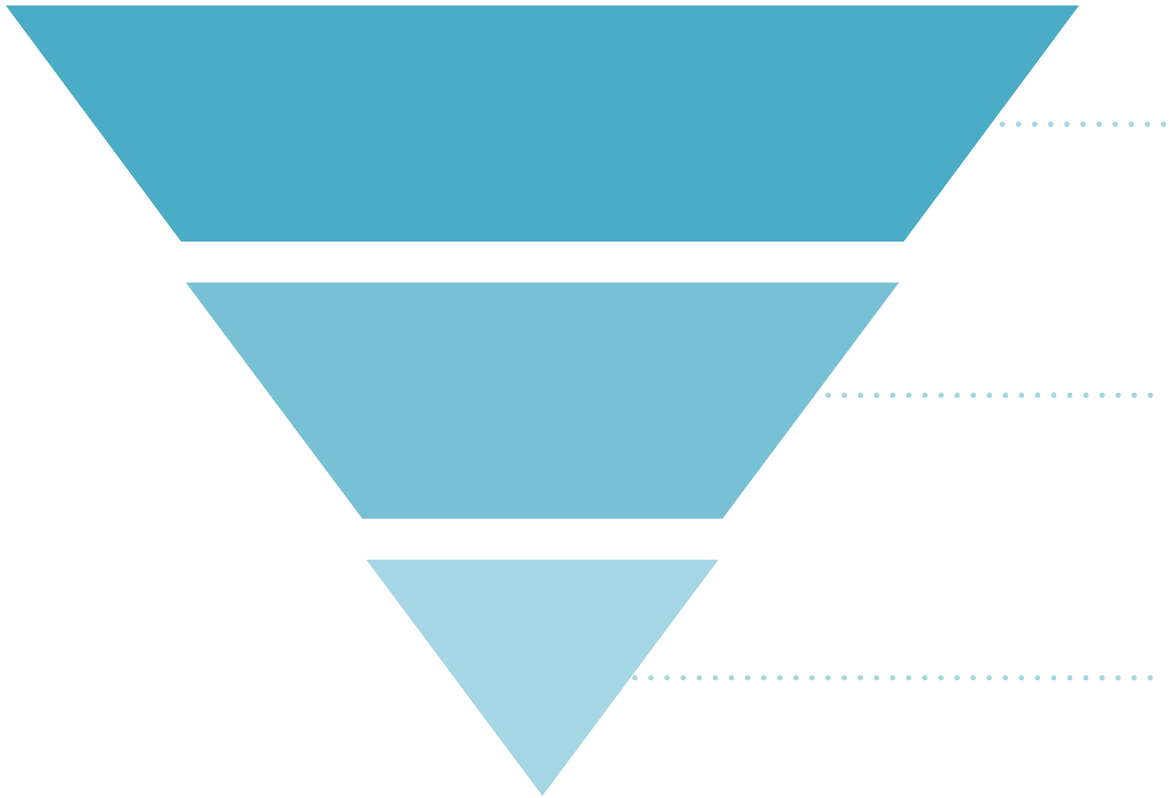
What we experiment on



User facing features

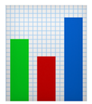
Bug fixes

Analyzing A/B tests, a seemingly simple problem...



Data logging

Metrics computation



Basic statistics

...but not at Uber's scale and pace

BEFORE

We computed all the metrics for all experiments

8

Pipelines

of 3,000+ lines of SQL

1-2

Runs per day

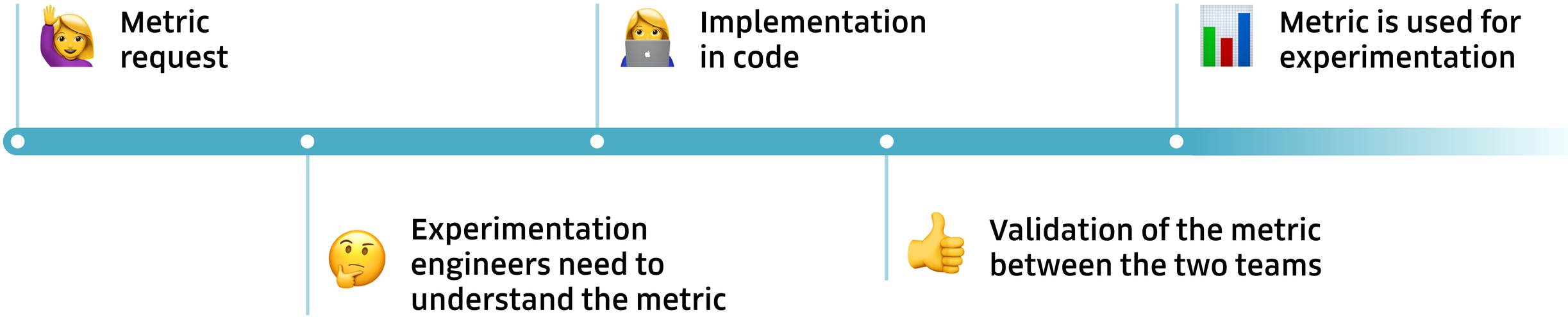
60%

Unused metrics

BEFORE To onboard new metrics

BEFORE To onboard new metrics

For each new metric:



BEFORE We had other problems too

People doing analysis themselves

Duplicate efforts across teams

Waste of time and resources

Use of slightly different methodologies

Incomprehension

BEFORE We had other problems too

People doing analysis themselves

Duplicate efforts across teams

Waste of time and resources

Use of slightly different methodologies

Incomprehension

Our team productivity suffered

Less time for more interesting problems

1 Overview of A/B testing at Uber

2 **Decoupling experimentation events from business metrics**

3 Extending the platform

4 Future work

5 Conclusion

Changing the paradigm

Experimentation team

- Experimentation data
- Who is seeing which experiment when

Other teams

- Team metrics
- Self-serve creation and edit of metrics

AFTER

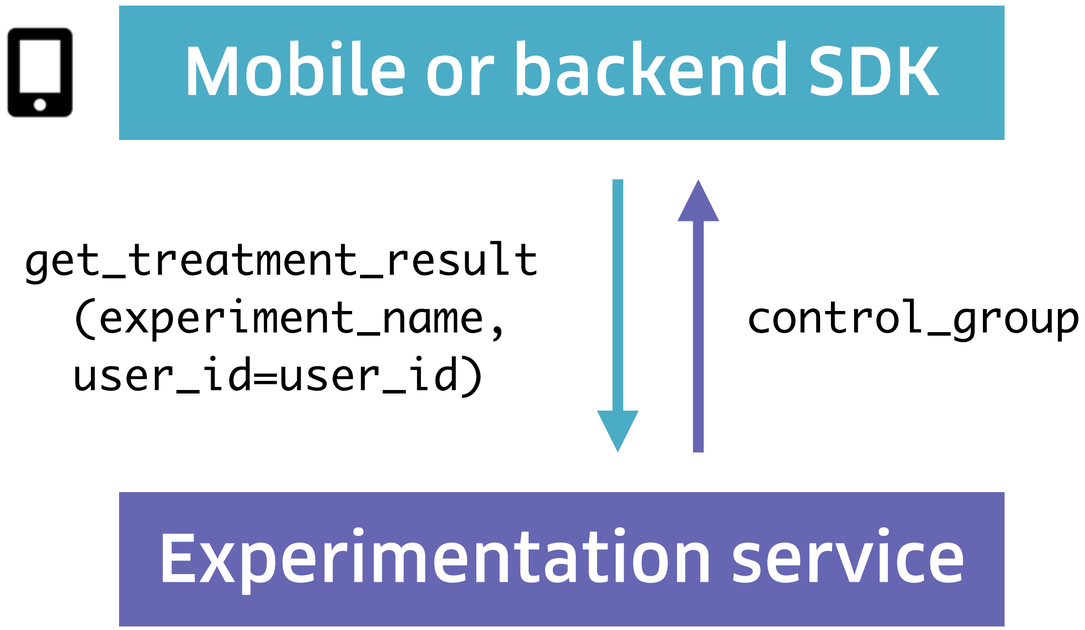
Experimentation events: the need for small data

Step 1
15B records

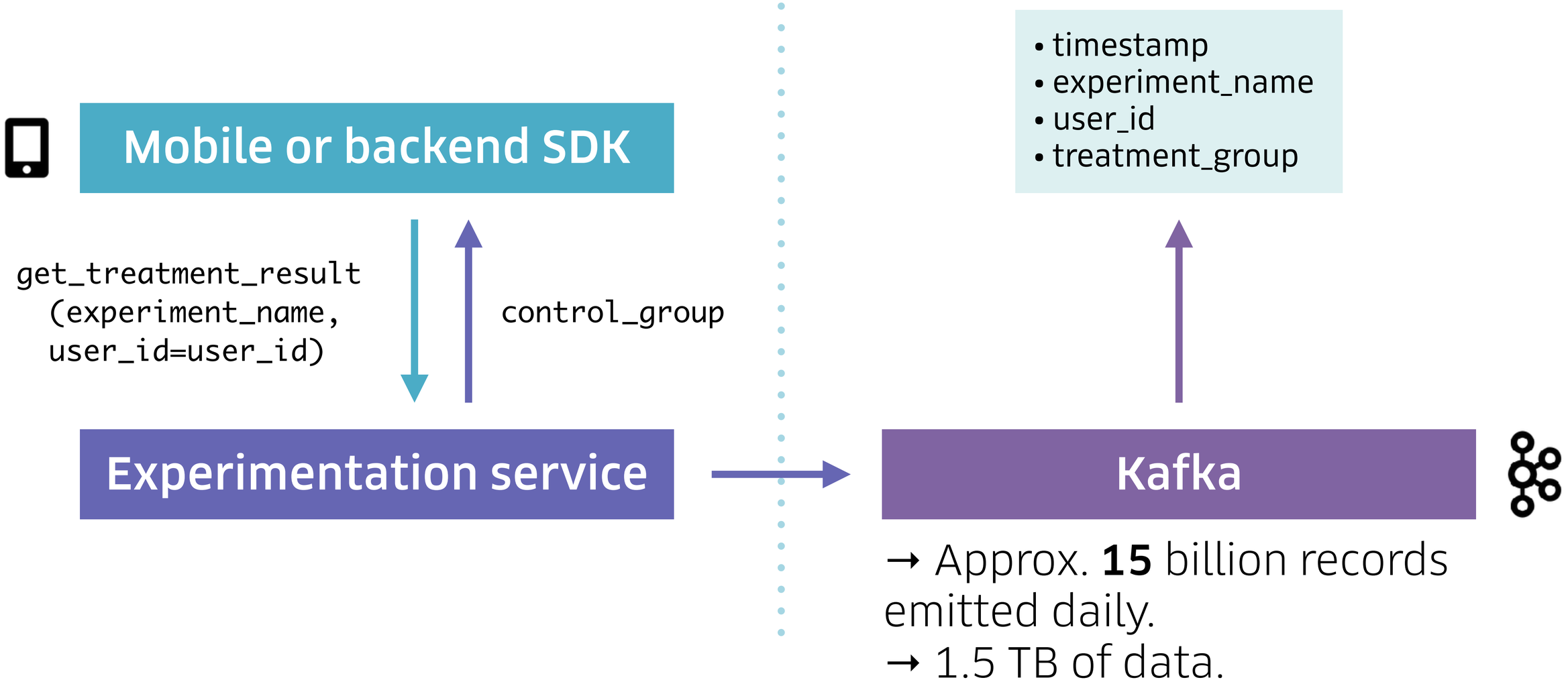
Step 2
5B records

Step 3
250M records

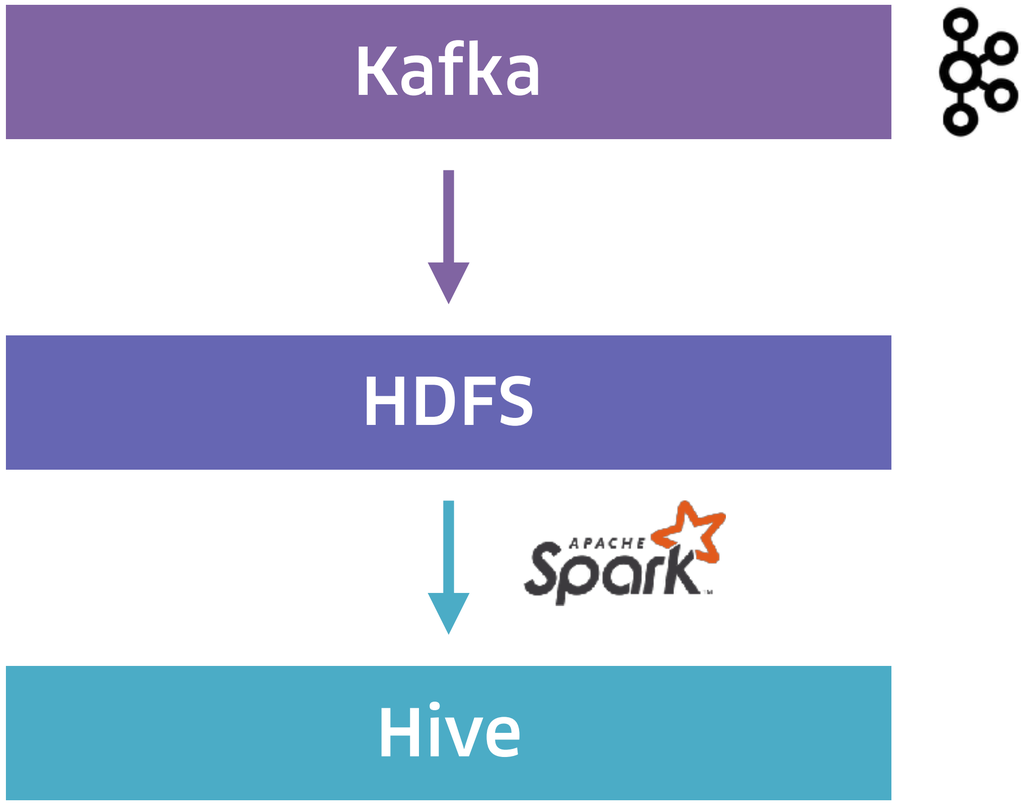
Logging of events happens automatically



Logging of events happens automatically



Step 2: Deduplication using Spark



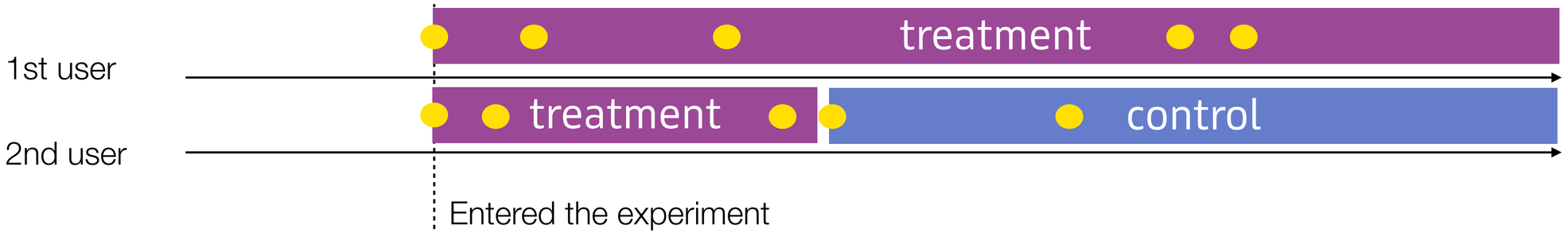
→ Down to approx. **5 billion** records daily
Partitioned by date

30 minutes
Spark Jobs average runtime

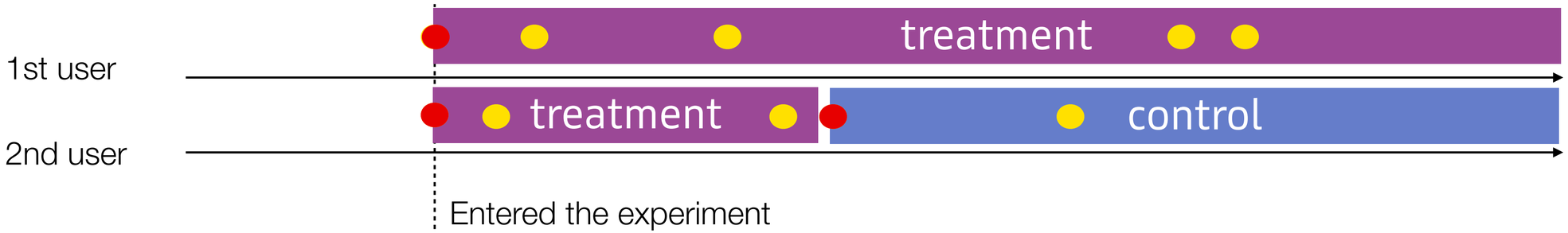
Still a lot of records

TIMESTAMP	USER_ID	EXPERIMENT_NAME	TREATMENT_GROUP
2018-09-12 10:12:45	42986	experiment_abc	treatment_1
2018-09-12 11:13:27	32989	experiment_abc	treatment_2
2018-09-12 11:17:45	98829	experiment_abc	control
2018-09-12 14:45:34	98397	experiment_abc	control
2018-09-12 14:38:28	42986	experiment_abc	treatment_1

Step 3: only keeping the relevant data

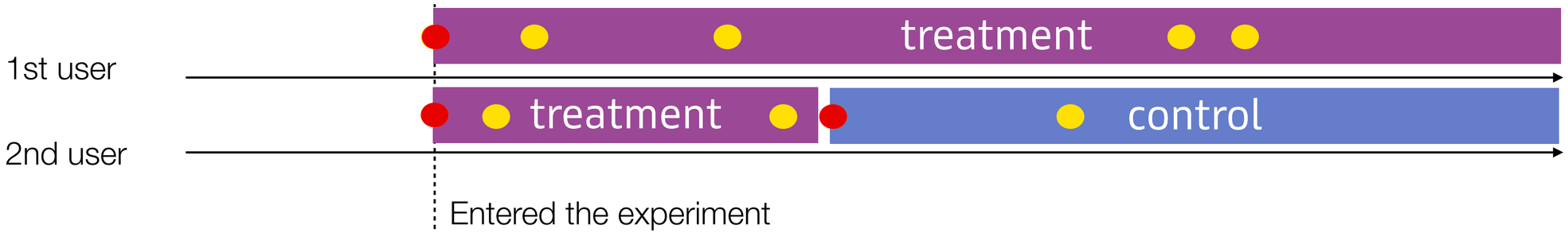


Step 3: only keeping the relevant data



ENTRY_DATE	EXIT_DATE	USER_ID	EXPERIMENT_NAME*	TREATMENT_GROUP
2018/09/01	NULL	1	experiment_abc	treatment
2018/09/01	2018/09/30	2	experiment_abc	treatment
2018/09/30	NULL	2	experiment_abc	control

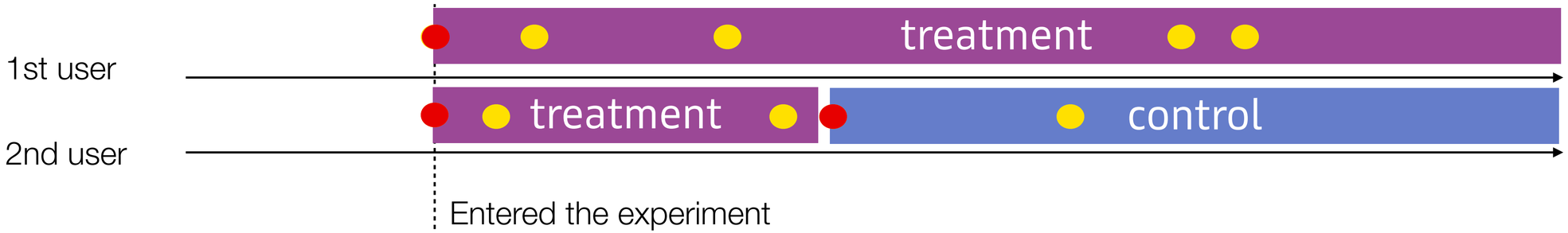
Step 3: only keeping the relevant data



ENTRY_DATE	EXIT_DATE	USER_ID	EXPERIMENT_NAME*	TREATMENT_GROUP
2018/09/01	NULL	1	experiment_abc	treatment
2018/09/01	2018/09/30	2	experiment_abc	treatment
2018/09/30	NULL	2	experiment_abc	control

Now **250 million** daily records

Step 3: only keeping the relevant data



ENTRY_DATE	EXIT_DATE	USER_ID	EXPERIMENT_NAME*	TREATMENT_GROUP
2018/09/01	NULL	1	experiment_abc	treatment
2018/09/01	2018/09/30	2	experiment_abc	treatment
2018/09/30	NULL	2	experiment_abc	control

Now **250 million** daily records

*** PARTITION COLUMN**

Takeaway #1

Whenever possible, present the data in a format that is easy to consume, not easy to compute.

Letting people define their own experimentation metrics

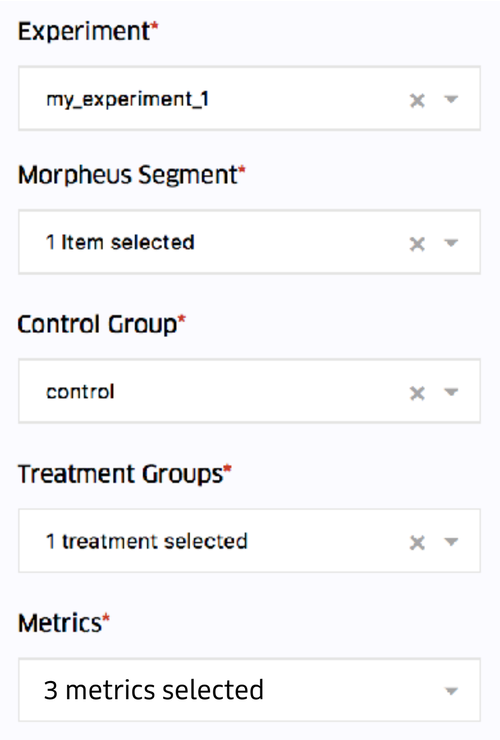
```
-------------------------------------------------------------------------
-- Start writing your metric SQL using the template included below
-------------------------------------------------------------------------

SELECT
    {{experiment_user_id}}
  , {{experiment_treatment}}
  , <your formula> AS 'metric_value'
FROM
  {{experimentCohort}}
LEFT JOIN
  <your dataset> a
ON
    AND {{experiment_user_id}} = a.user_id
    -- TODO Filter the metric table based on dates passed in from experimentat
    AND a.datestr >= {{measureStart}}
    AND a.datestr < {{measureEnd}}
GROUP BY
  {{experiment_user_id}}, {{experiment_treatment}}
```

Supported:

- Hive
- Presto
- Vertica

Tying it all together



Tying it all together

Experiment*

my_experiment_1

Morpheus Segment*

1 Item selected

Control Group*

control

Treatment Groups*

1 treatment selected

Metrics*

3 metrics selected

SQL | metric_1

```
SELECT treatment_group_key , user_uuid , denominator_value , numerator_value FROM (SELECT exp.user_uuid
     , exp.treatment_group_key
     ,                                    AS numerator_value
     ,                                    AS denominator_value

FROM (SELECT user_uuid, first_treatment_group_key AS treatment_group_key, begin_effective_timestamp FROM ex

JOIN (                                    o
  ON exp.user_uuid = o.driver_uuid
 AND o.start_timestamp_utc >= FROM_UNIXTIME(exp.begin_effective_timestamp)
 AND o.start_timestamp_utc < now()
 AND o.datestr >= '1970-01-01'

GROUP BY exp.user_uuid
       , exp.treatment_group_key) AS a
```

Tying it all together

Experiment*

my_experiment_1

Morpheus Segment*

1 Item selected

Control Group*

control

Treatment Groups*

1 treatment selected

Metrics*

3 metrics selected

SQL | metric_1

```
SELECT treatment_group_key , user_uuid , denominator_value , numerator_value FROM (SELECT exp.user_uuid
     , exp.treatment_group_key
     ,                                    AS numerator_value
     ,                                    AS denominator_value

FROM (SELECT user_uuid, first_treatment_group_key AS treatment_group_key, begin_effective_timestamp FROM e

JOIN (                                    o
  ON exp.user_uuid = o.driver_uuid
 AND o.start_timestamp_utc >= FROM_UNIXTIME(exp.begin_effective_timestamp)
 AND o.start_timestamp_utc < now()
 AND o.datestr >= '1970-01-01'

GROUP BY exp.user_uuid
       , exp.treatment_group_key) AS a
```

metric_1

Treatment	Lift Mean	-20% -10% 0%
control	– 0.587	
enabled	-8.070% 0.540	

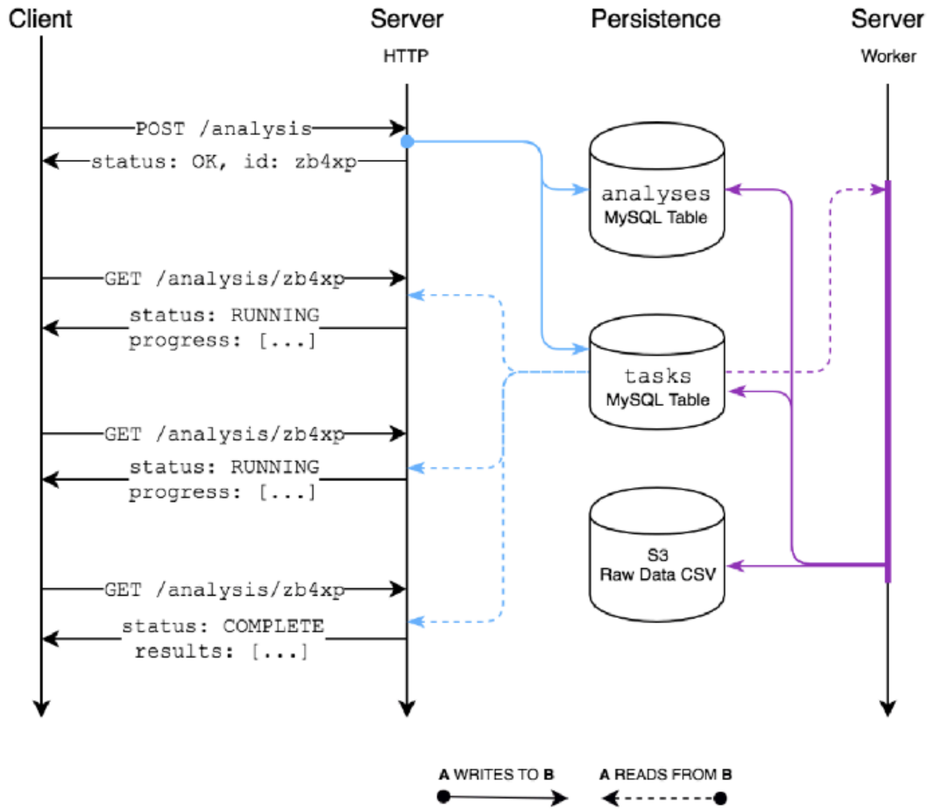
metric_2

Treatment	Lift Mean	-40% -20% 0%
control	– 0.159	
enabled	-23.485% 0.121	

metric_3

Treatment	Lift Mean	0% +20%
control	– 25.27	
enabled	+17.92% ▲ 29.80	

Everything happens asynchronously



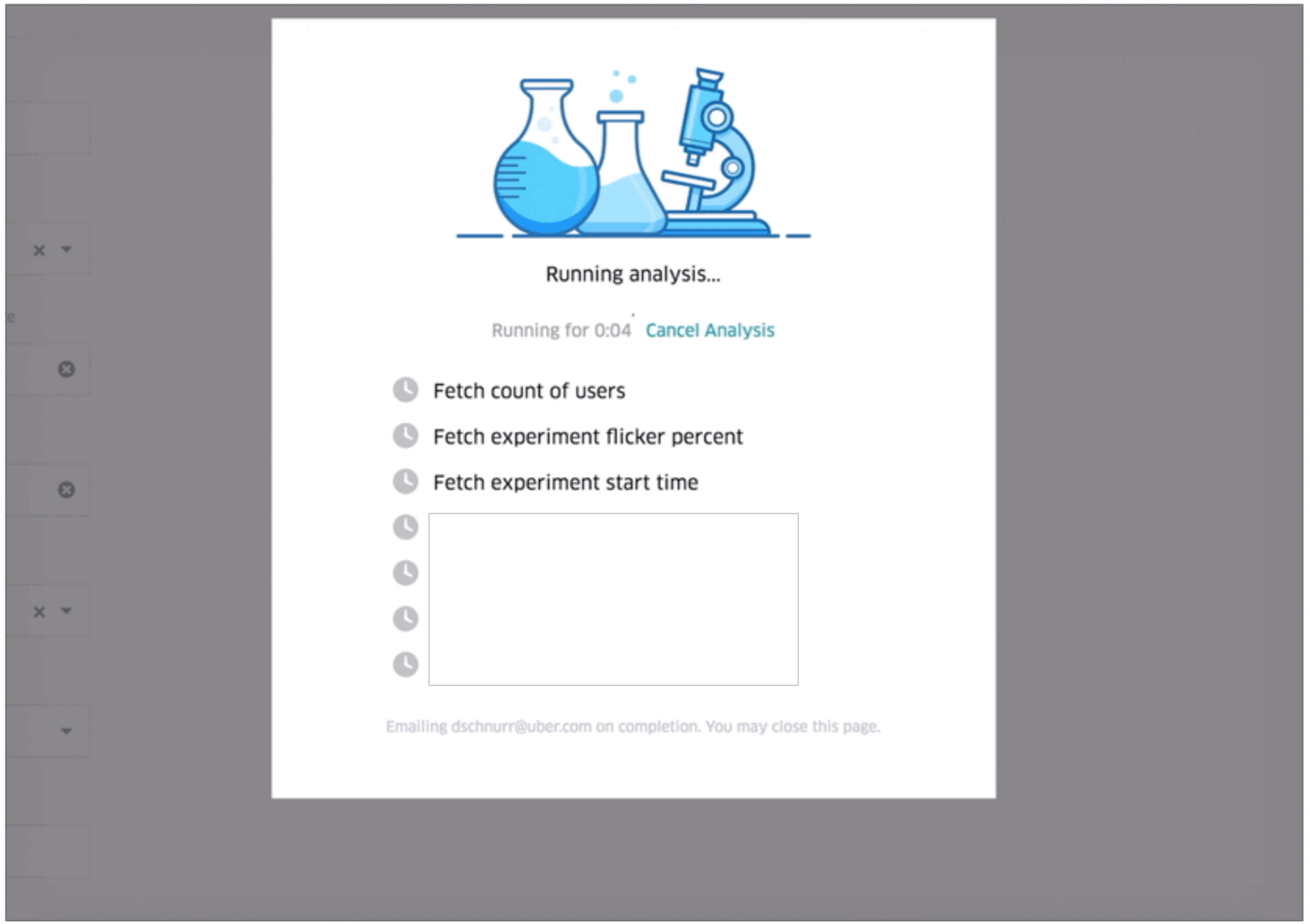
2.9 minutes
Average runtime of a metric

5.75 metrics
Average number of metrics per report

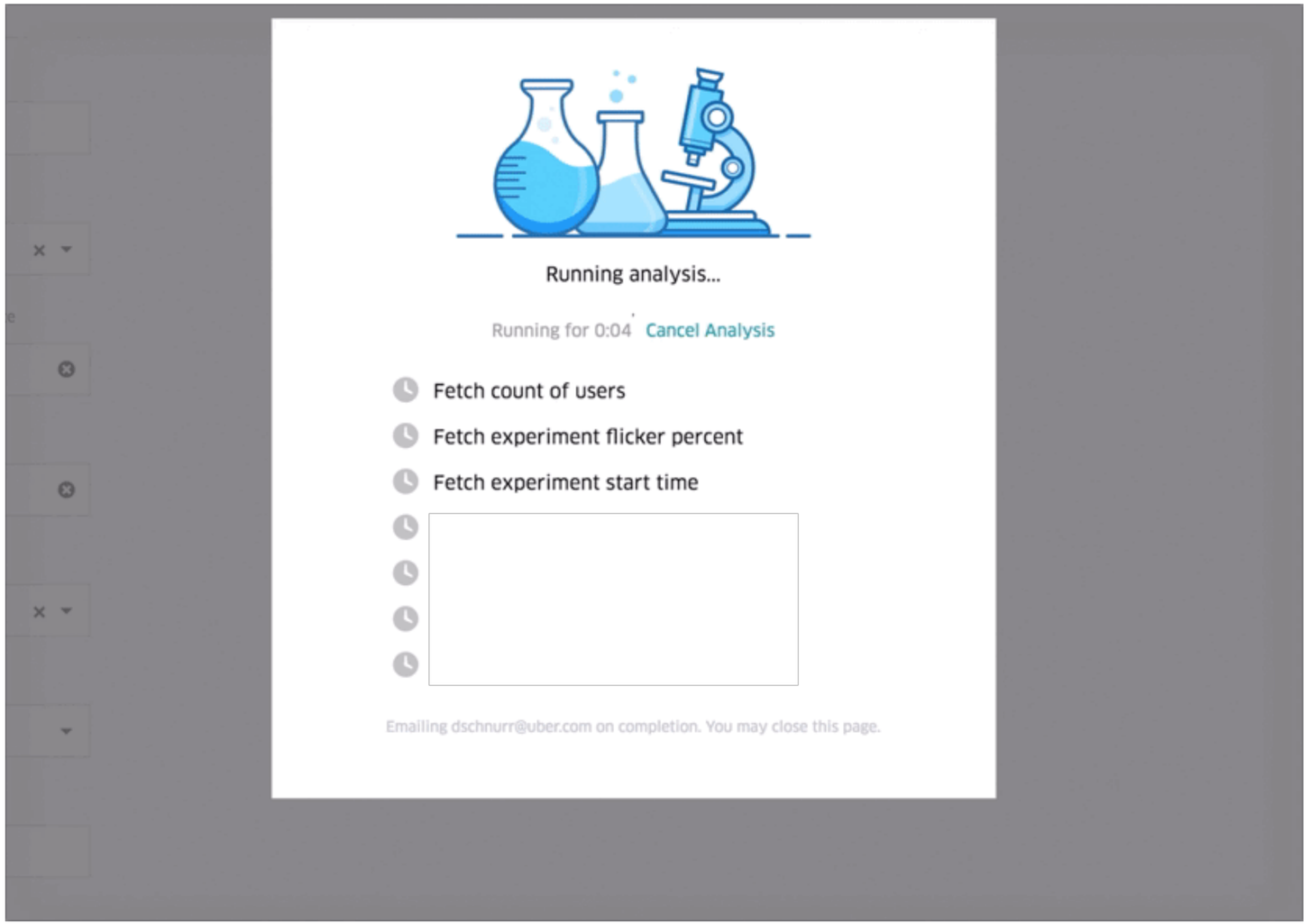
Before vs. After

	Before	After
Freshness	⭕ 24 hours	✅ 4 hours
Metrics selection	⭕ Static set	✅ BYOM
Metrics onboarding	⭕ Engineering work required	✅ No engineering work needed
Resources	⭕ Computation of irrelevant metrics	✅ Computation of what’s needed only
Speed	✅ Faster	⭕ Slower

User delight, making the wait pleasant



User delight, making the wait pleasant



Takeaway #2

Balance speed versus flexibility based on user needs.

1 Overview of A/B testing at Uber

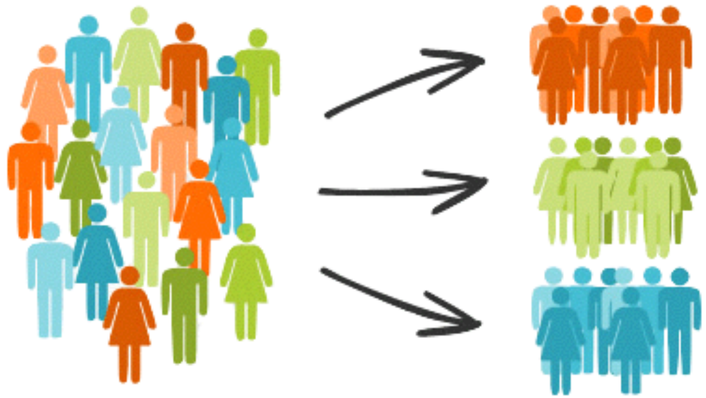
2 Decoupling experimentation events from business metrics

3 **Extending the platform**

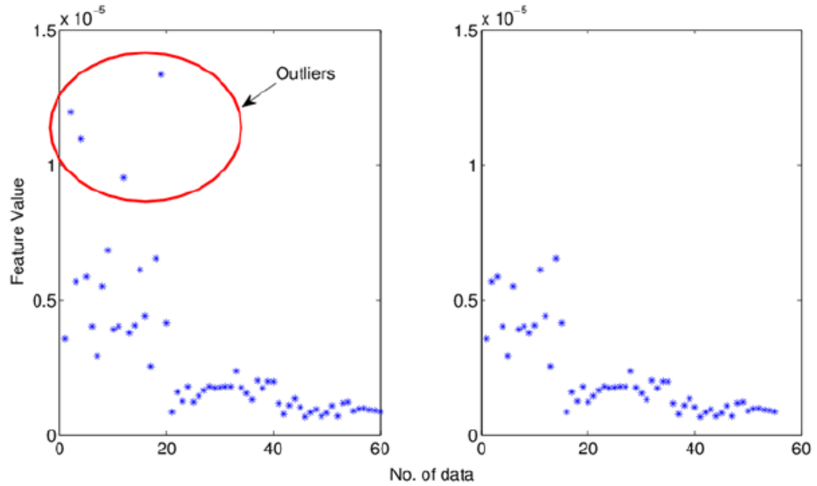
4 Future work

5 Conclusion

It’s now easy to add new features



Data Engineering



Data Science

Data engineering: slicing and dicing of results

ENTRY_DATE	EXIT_DATE	USER_ID	EXPERIMENT_NAME	TREATMENT_GROUP
2018/09/01	NULL	1	experiment_abc	treatment_1
2018/09/01	2018/09/30	2	experiment_abc	treatment_2

Data engineering: slicing and dicing of results

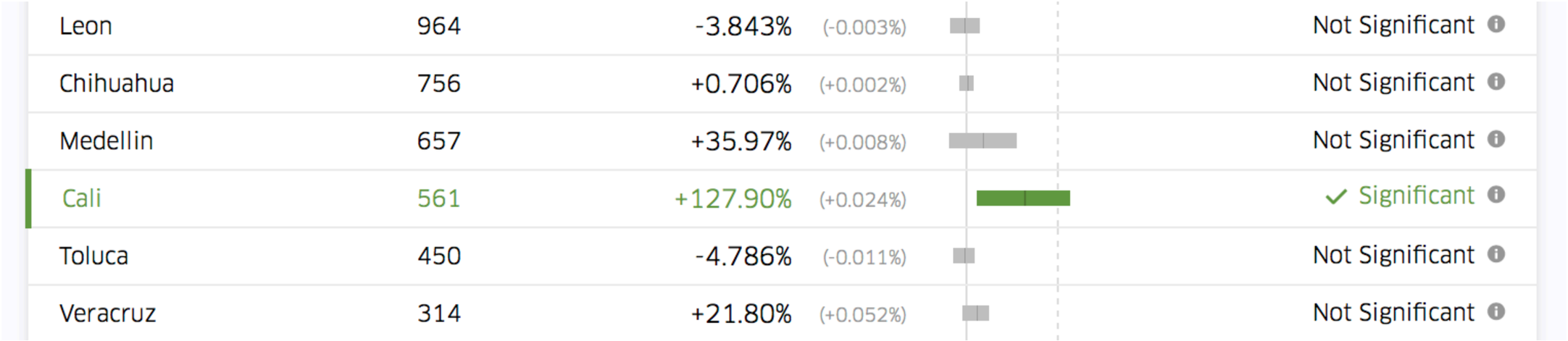
ENTRY_DATE	EXIT_DATE	USER_ID	EXPERIMENT_NAME	TREATMENT_GROUP
2018/09/01	NULL	1	experiment_abc	treatment_1
2018/09/01	2018/09/30	2	experiment_abc	treatment_2

`join on` **`user_id = user_id`** `and` **`to_date(entry_date) = date`**

DATE	USER_ID	CITY
2018/09/01	1	Veracruz
2018/09/01	2	Medellin

= Enriched experimentation logs

Segmented results for a fraction of time

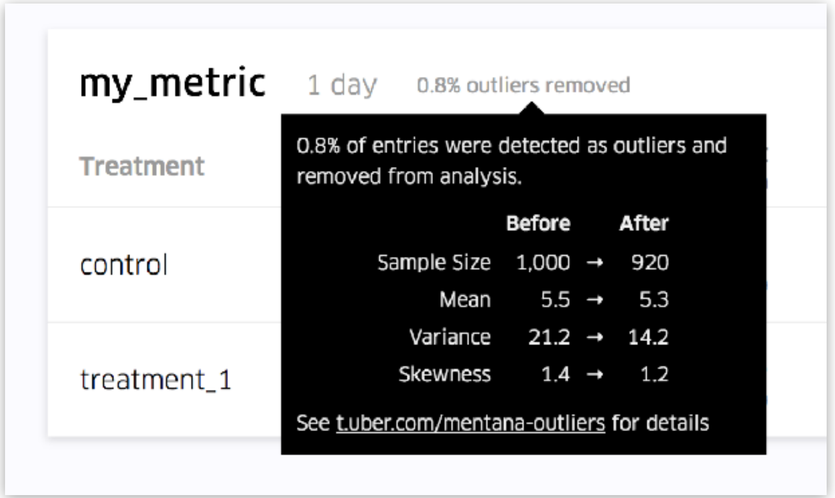


Leon	964	-3.843%	(-0.003%)	Not Significant
Chihuahua	756	+0.706%	(+0.002%)	Not Significant
Medellin	657	+35.97%	(+0.008%)	Not Significant
Cali	561	+127.90%	(+0.024%)	✓ Significant
Toluca	450	-4.786%	(-0.011%)	Not Significant
Veracruz	314	+21.80%	(+0.052%)	Not Significant

Additional run time: **+35%** on average

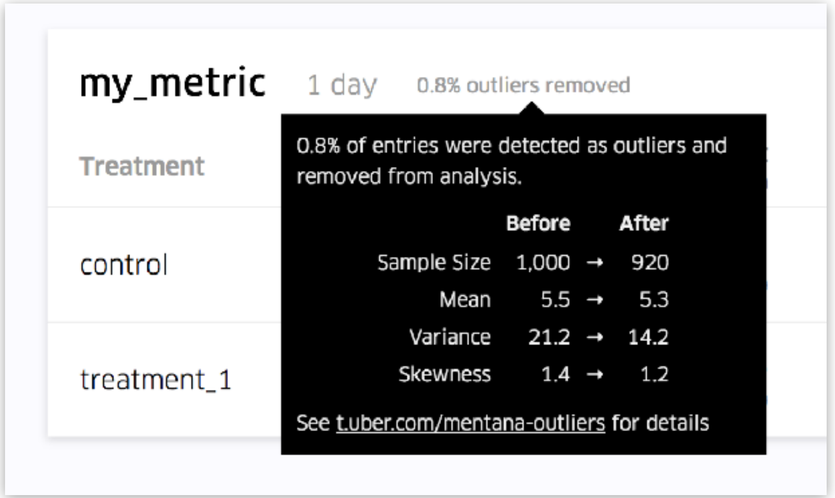
Data Science innovation

Outlier removal
Removes irregularities in the data, enables more robust results

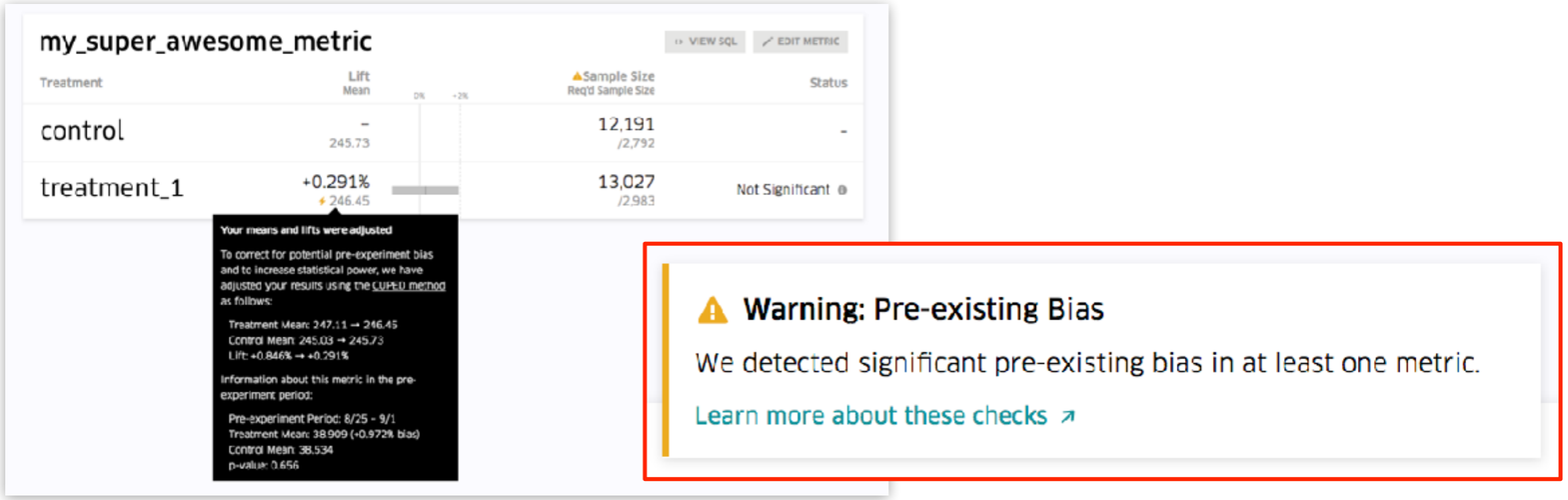


Data Science innovation

Outlier removal
Removes irregularities in the data, enables more robust results



Pre-existing bias detection and correction
Using CUPED method to adjust results and increase statistical power



Takeaway #3

Instead of trying to do it all, do what you are great at and build an infrastructure that lets others add the missing pieces.

1 Overview of A/B testing at Uber

2 Decoupling experimentation events from business metrics

3 Extending the platform

4 Future work

5 Conclusion

Metrics governance

Infinite Scroll by @artrayd

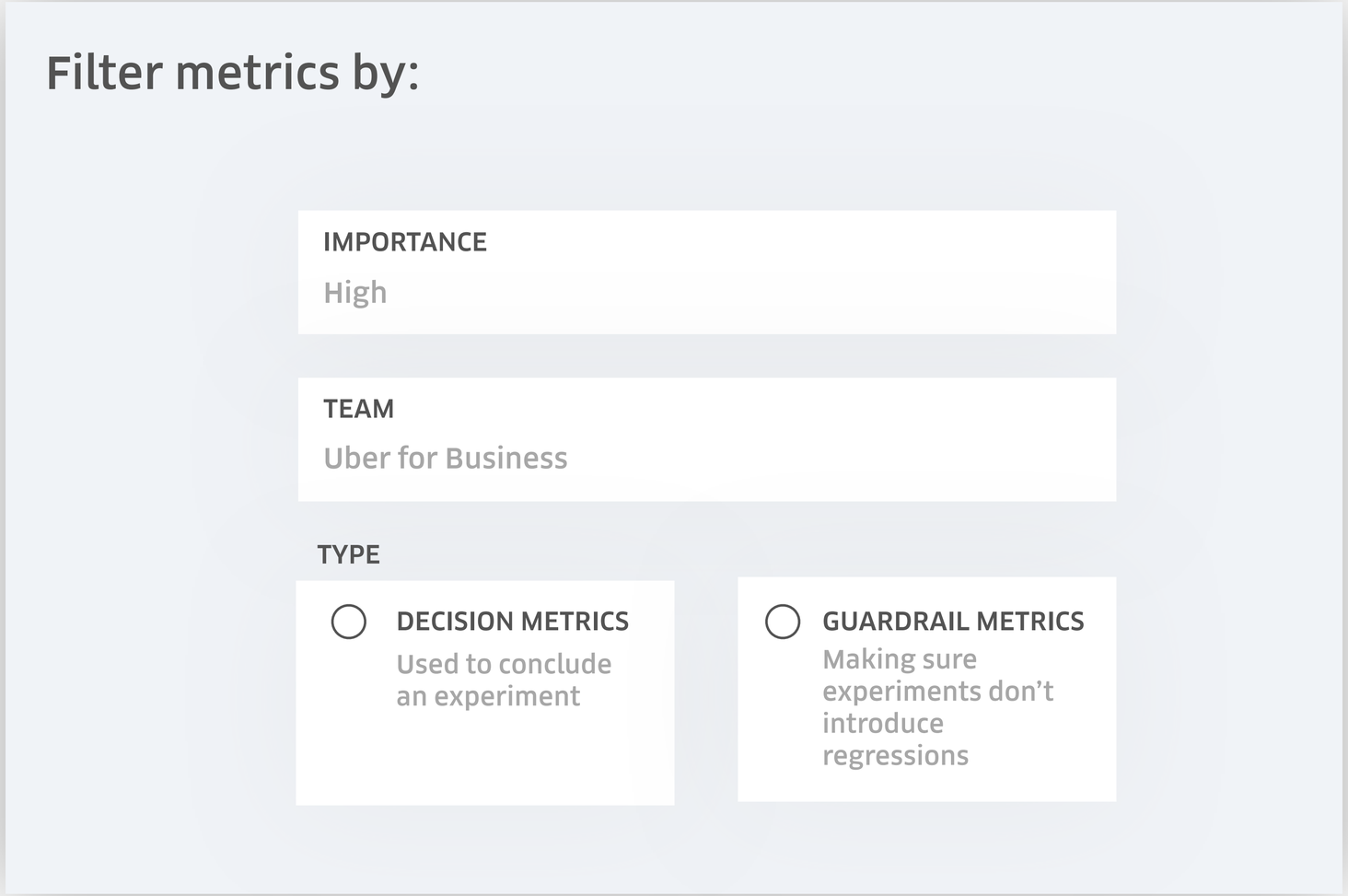
What looking for the right metric looks like

Metrics governance

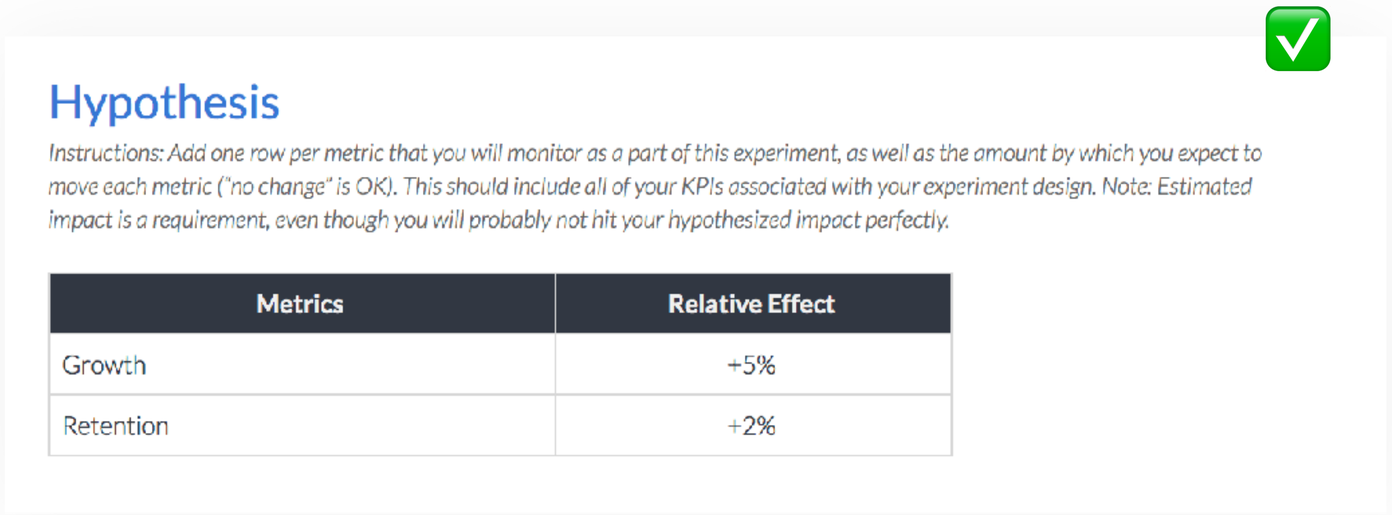
Infinite Scroll by @artrayd

What looking for the right metric looks like

A vast but organized catalog of metrics



Self-serve metrics != metrics as an afterthought



Hypothesis

Instructions: Add one row per metric that you will monitor as a part of this experiment, as well as the amount by which you expect to move each metric ("no change" is OK). This should include all of your KPIs associated with your experiment design. Note: Estimated impact is a requirement, even though you will probably not hit your hypothesized impact perfectly.

Metrics	Relative Effect
Growth	+5%
Retention	+2%

Self-serve metrics != metrics as an afterthought

✅

Hypothesis

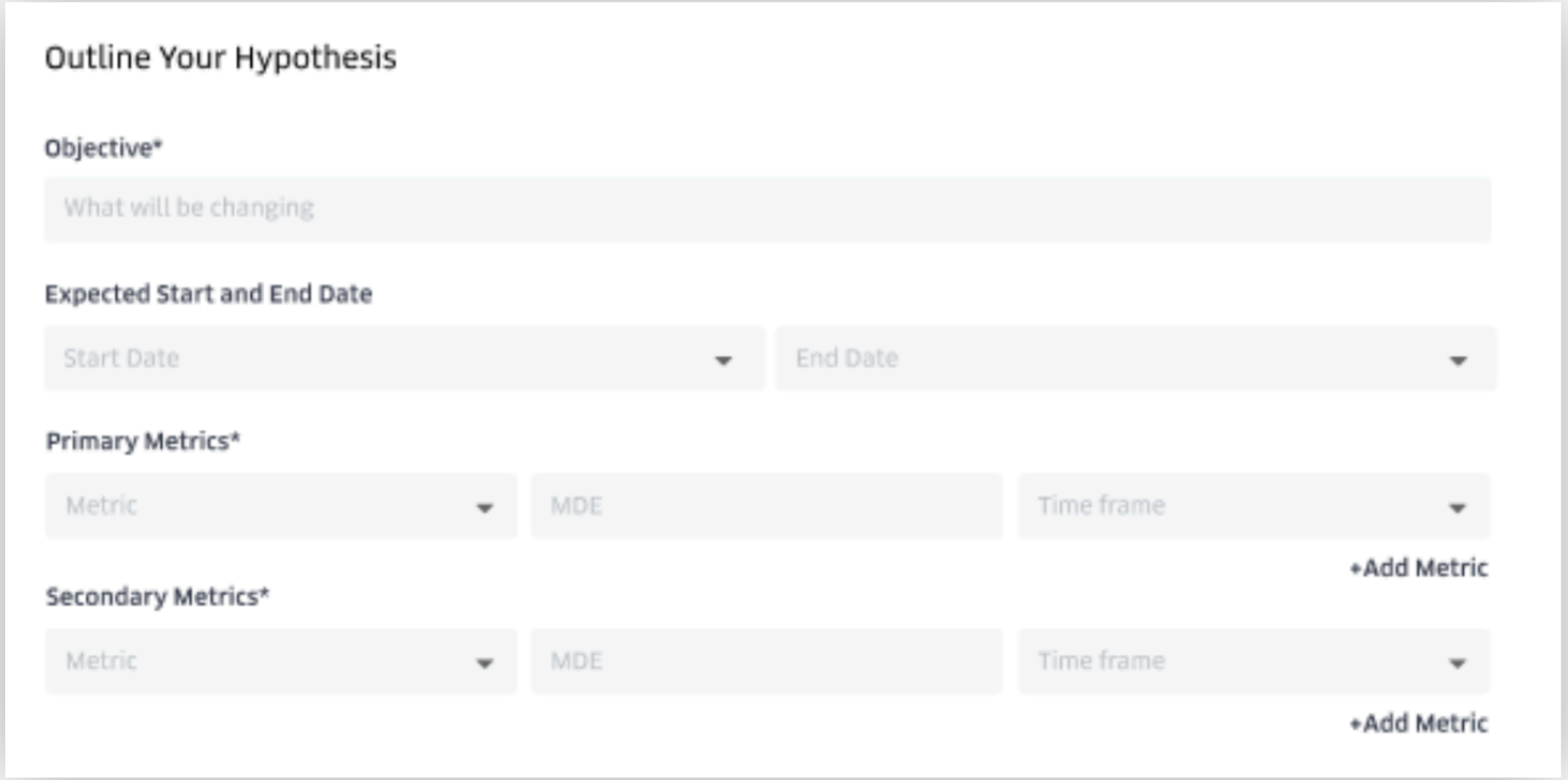
Instructions: Add one row per metric that you will monitor as a part of this experiment, as well as the amount by which you expect to move each metric ("no change" is OK). This should include all of your KPIs associated with your experiment design. Note: Estimated impact is a requirement, even though you will probably not hit your hypothesized impact perfectly.

Metrics	Relative Effect
Growth	+5%
Retention	+2%

⭕

Run experiment → Find a metric that moves in the right direction → Launch

Upcoming work



1 Overview of A/B testing at Uber

2 Decoupling experimentation events from business metrics

3 Extending the platform

4 Future work

5 Conclusion

We’re hiring!

- **Data Scientists**
  All levels
- **Software Engineers**
  All levels

>> Check uber.com/careers

Acknowledgments

Colin Reid

David Schnurr

Egor Gryaznov

Spencer Lin

Suman Bhattacharya

Tianxia Zhou

If you only remember three things

If you only remember three things

01. Easy to consume data > Easy to compute data

If you only remember three things

01. Easy to consume data > Easy to compute data

02. Speed ↔ Flexibility tradeoff

If you only remember three things

01. Easy to consume data > Easy to compute data

02. Speed ↔ Flexibility tradeoff

03. Leverage your strengths, build products that users can contribute to

Thank you!

Milène Darnis <milene@uber.com>