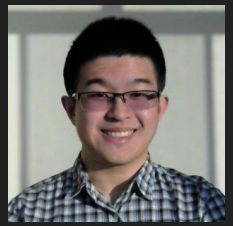
Large-scale Experimentation at Roblox

The Roblox Team

Jay Martin
Data Science Manager

Facebook, Netflix, Stitch Fix, NYU, UC Berkeley



Boris Shang
Product Manager

Facebook, Nerdwallet, Capital One, UPenn

Vincent Su
Data Engineer

Salesforce, CMU, Mizzou

Kelly Cheng
Software Engineer

eBay, 23andme, Zynga, UC Berkeley

Xiaochen Zhang
Research Scientist

Uber, CMU, Northwestern

What we’re covering today

Goal: Understand experimentation in industry

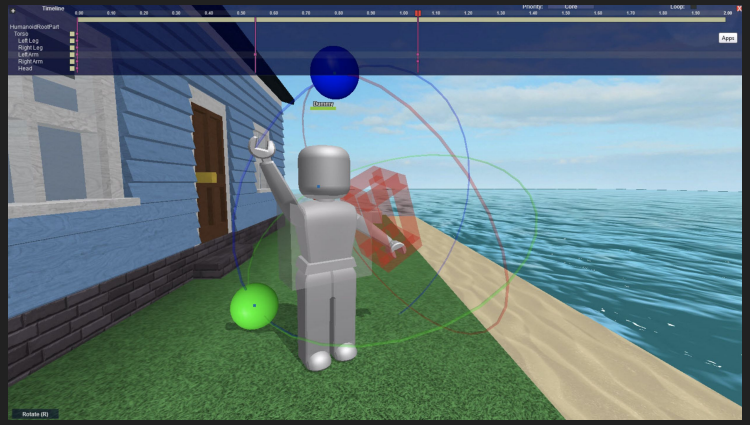
Sub-goal: Understand how Data Science works with other disciplines

- Some background
  - The team
  - Our company
  - The role of data science
- Deep dive: Experimentation

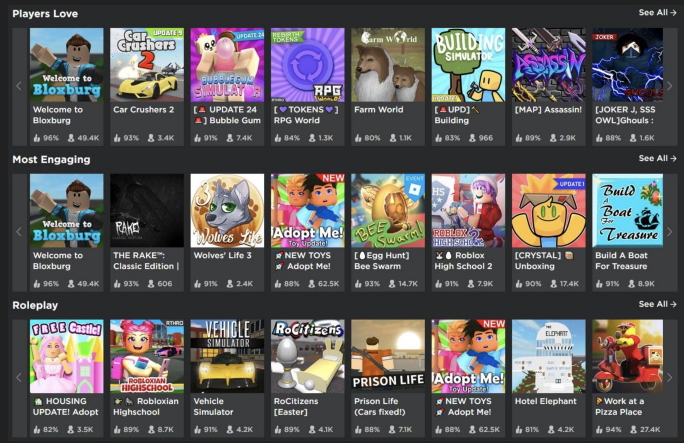
Roblox is a lot of things...

- A gaming company that doesn’t make games
- A YouTube for games
- A metaverse

Roblox as a Business



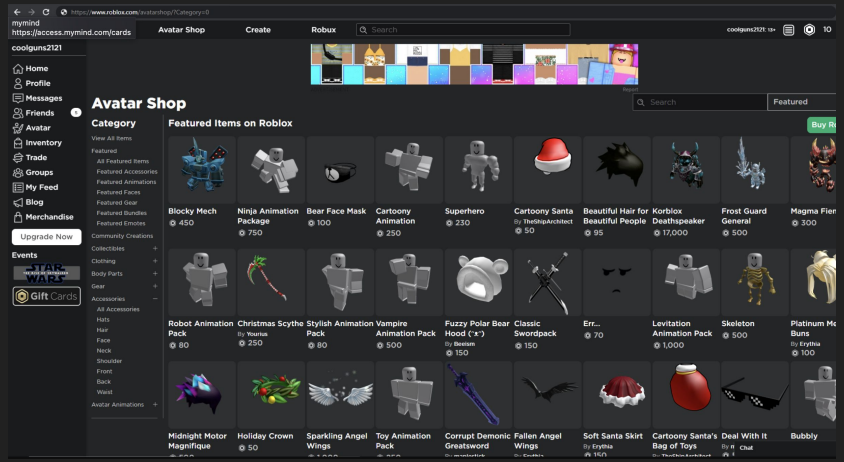
Developers supply 3D content

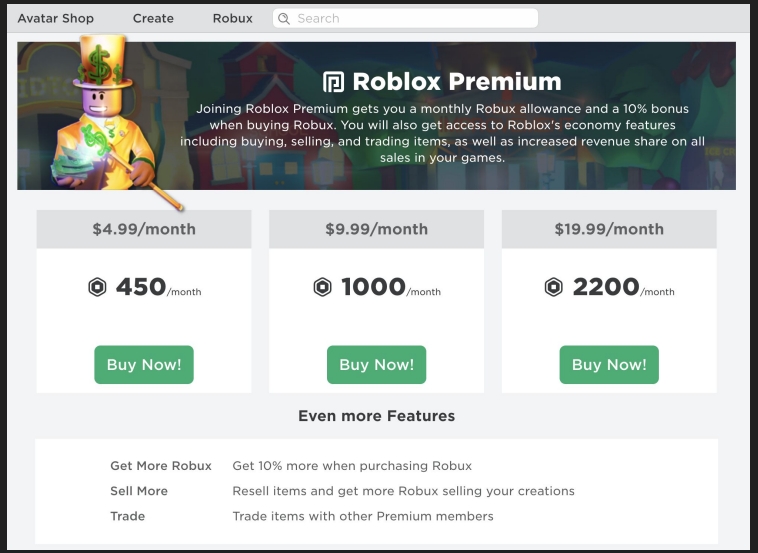


Players demand content

A “two-sided” marketplace like Uber, AirBnb, YouTube, but for games. Aka a “platform”

Roblox as a Business

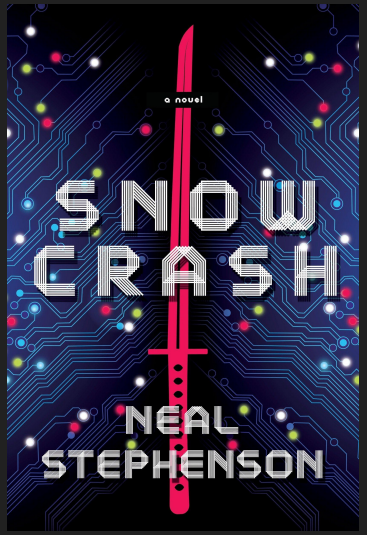




We make money from our Avatar Shop and Premium Subscriptions

Roblox as a Vision

- The metaverse is believed to be the future of the internet
- Combines VR, AR, and reality into a single, shared experience



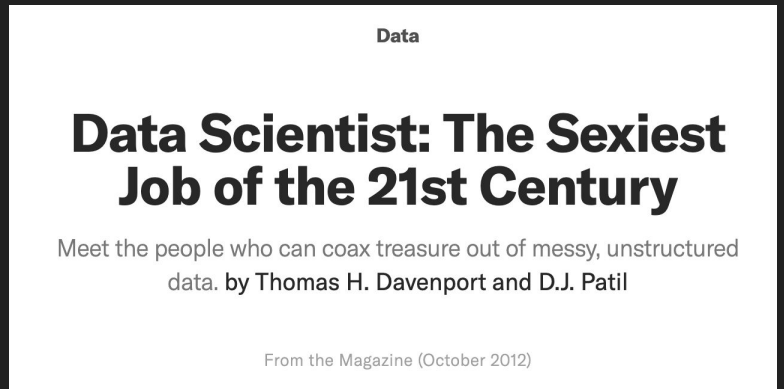
Key functions

- **Product**: Manages the development and improvement of features
- **(Data) Engineering:** Builds everything
- **User Experience “UX”**: Researches and designs products that offer the best experience for users

The Data Scientist

Data Science: Leverages big data, machine learning, and statistics to accelerate learning and automate decision making

- **Product Analyst:** Drive business strategy via user and ecosystem understanding and experiments
- **Research Scientist:** R&D novel solutions to quantitative problems
- **Machine Learning Engineer:** Implement and maintain ML models in production systems

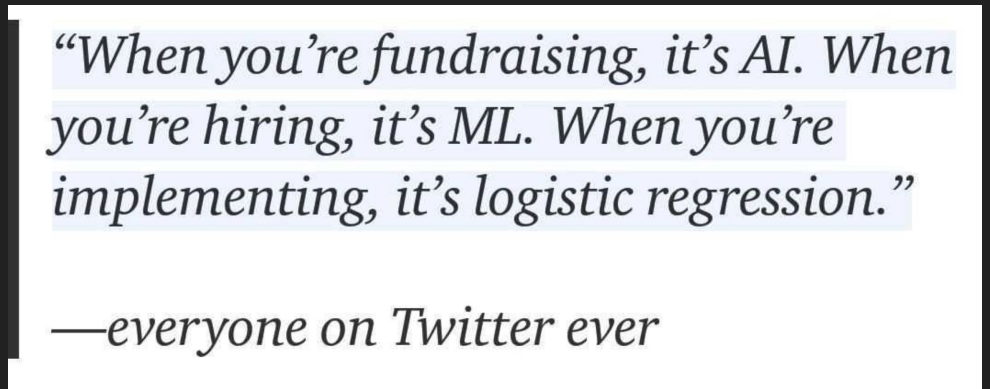


Data

Data Scientist: The Sexiest Job of the 21st Century

Meet the people who can coax treasure out of messy, unstructured data. by Thomas H. Davenport and D.J. Patil

From the Magazine (October 2012)



"When you're fundraising, it's AI. When you're hiring, it's ML. When you're implementing, it's logistic regression."

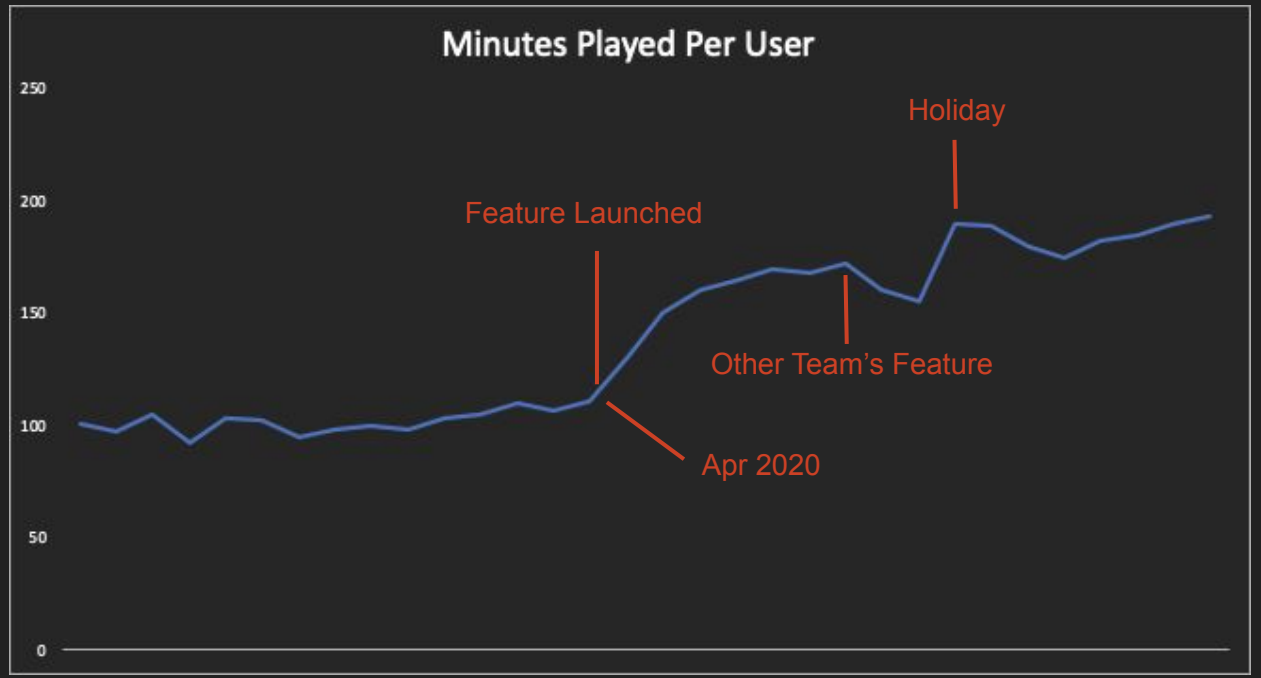
—everyone on Twitter ever

Why Run Experiments?

Which is better?

vs

Guess and Check



Run an Experiment

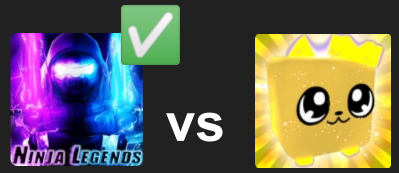
User Visits Roblox

50% of Users

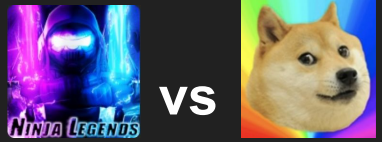
50% of Users

Continuous Improvement

Pick a Winner



New Hypothesis



Experiment Again

Experiments in the Software World

Massive scale

Hundreds of millions of users

High Concurrency

1000 experiments running at the same time

Extremely Iterative

Every experiment leads to another hypothesis

Digging in

Very simplified steps in running an experiment

- Setup - build the experiences necessary + configure exp settings
- Enroll - randomly assign users into different experiences
- ETL - data munging to get the metric results of each variant
- Interpret - run statistical tests and judge the experiment’s success

Building an Experimentation Platform

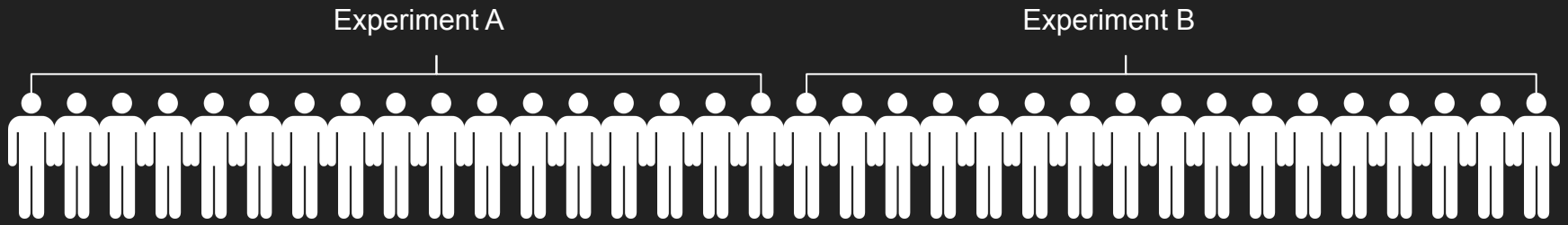
Engineering Goals

Creating a customer focused tool to reduce the effort required to start an experiment

- Support high experiment volume
  Enable parallel experimentation
- Scale with Roblox
  Avoid storing experiment state
- Reduce engineering cost per experiment
  Use experiment parameters, not variant numbers

Supporting High Experiment Volume

Option 1: Exclusive experiment allocation



Benefits:

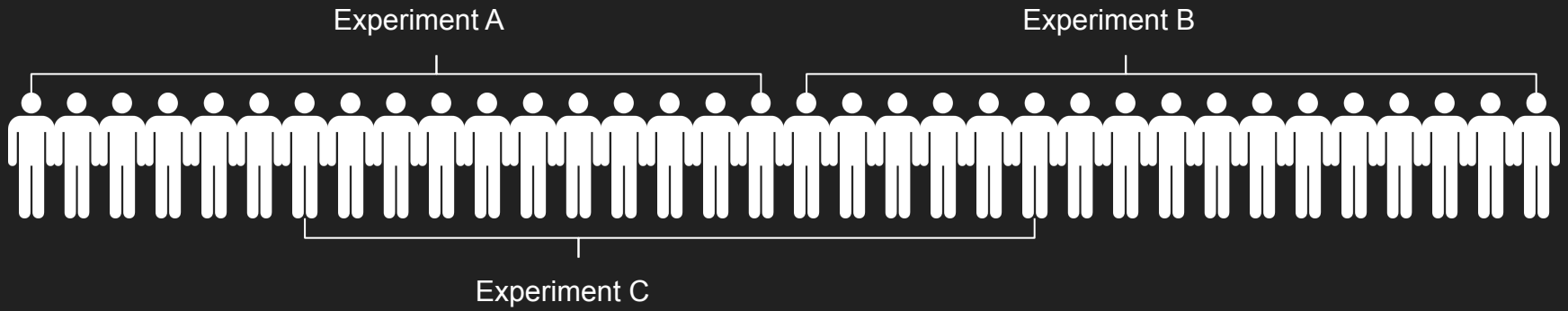
- No overlapping players

Problems:

- Low concurrency / starvation

Supporting High Experiment Volume

Option 2: Overlapping experiment allocation



Benefits:

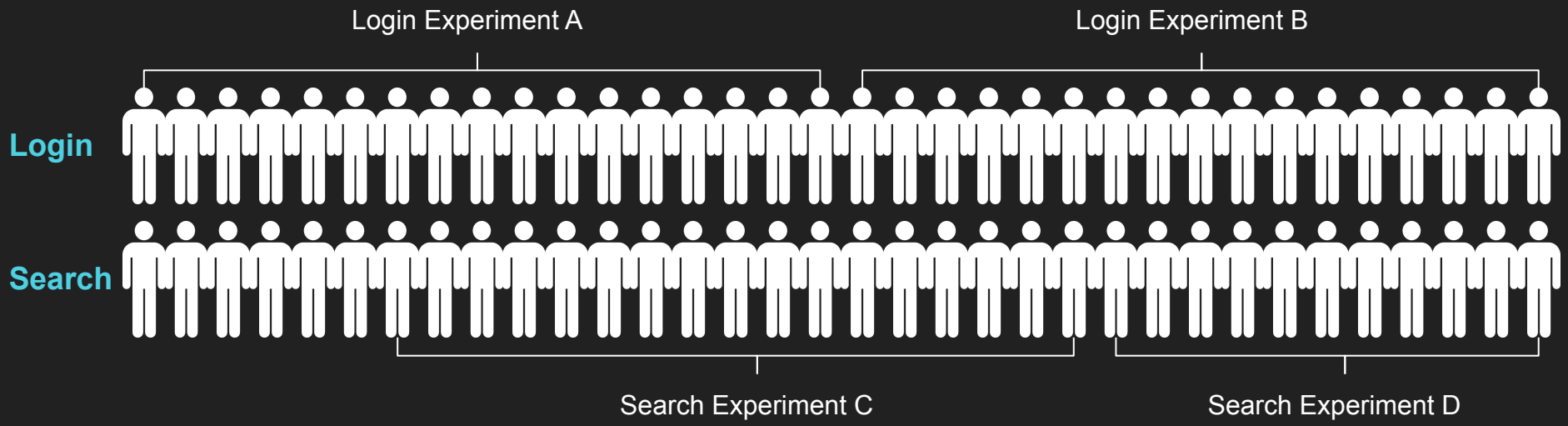
- High concurrency

Problems:

- Experiment pollution

Supporting High Experiment Volume

Option 3: Namespaced exclusive allocation



Benefits:

- High concurrency
- Limited overlapping players

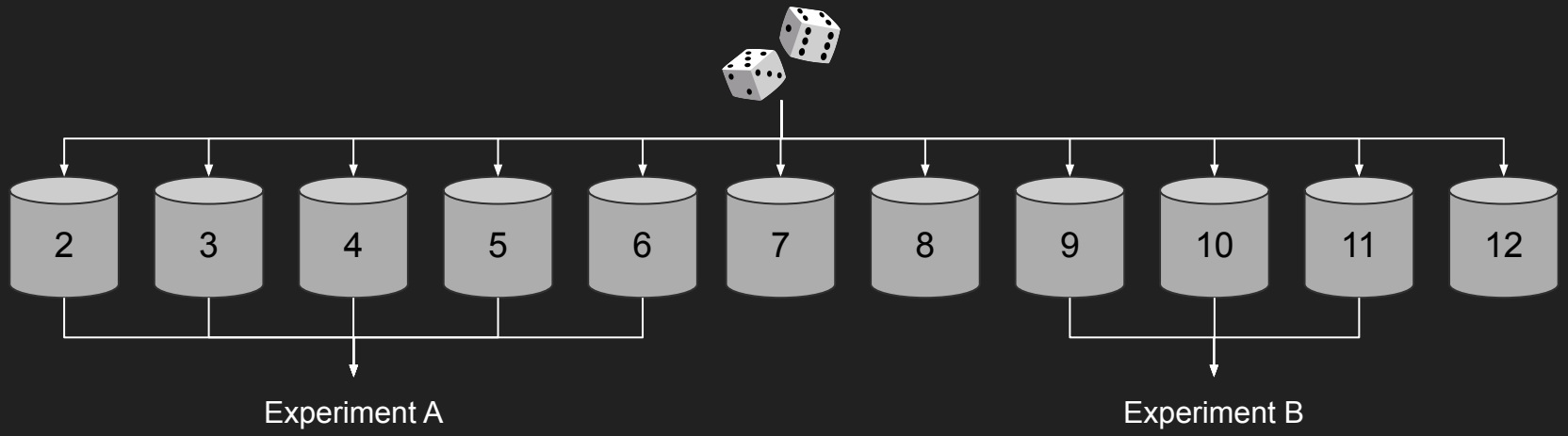
Problems:

- Complexity

Randomization for Experiments

Random numbers are used to bucket players into experiments and variants

- Random number generators are critical for experiments
- Allows us to bucket players into experiments (and variants)

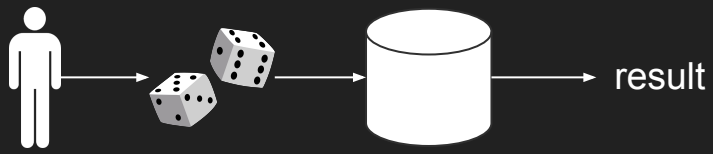


Unscalable Randomization at Roblox

Enabling high volume requests to experimentation services

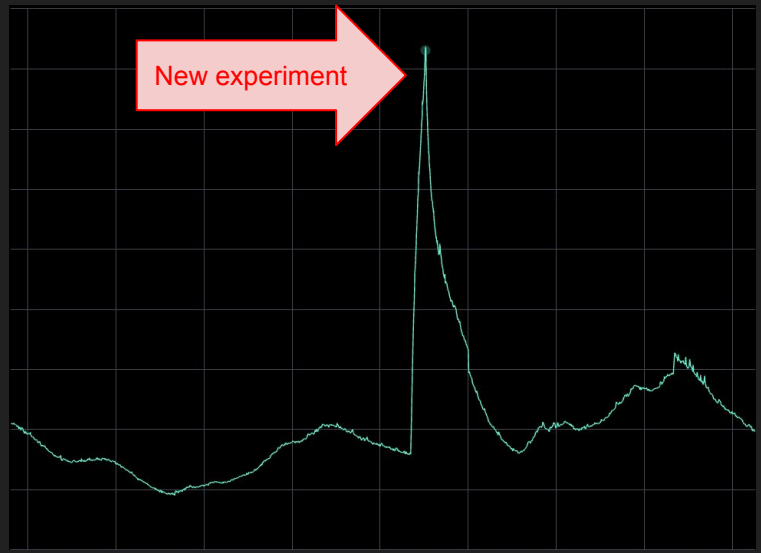
Random Number Generator

- Generate a random number
- Store number in database
- Retrieve number for user



- Database bottleneck

Database Request Volume Spikes

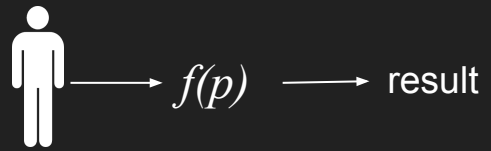


Scalable Randomization at Roblox

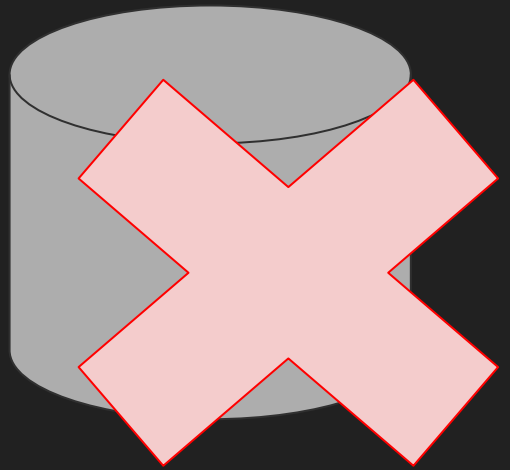
Enabling high volume requests to experimentation services

Hashing

- Compute a number for a player using a **hash function**
- The hash function generates the same result for the same input player *(p)*



- Avoids database bottleneck



Engineering Effort for Experiments

Avoid experimentation definition in application code

Typical experiment logic binds a variant id with some business logic:

```
variant = <number from exp system>

if (variant == 0):
    setButtonSize(10px)

if (variant == 1):
    setButtonSize(20px)

if (variant == 2):
    setButtonSize(40px)
```

Forces frequent code changes. Code changes are costly:

- Development time
- Code reviews
- Server deployment

Engineering Effort for Experiments

Avoid experimentation definition in application code

Deliver experiment values instead of variant ids

```
buttonSize = <value from exp system>
setButtonSize(buttonSize)

// More experiments
fontSize = <value from exp system>
setFontSize(fontSize)
```

Reduce code modification:

- Simplifies code
- Reduces deployment volume
- Improved flexibility and power:
  - Multivariate experiments
  - Dynamic variant generation

Defining Experiments

Make value generation system precise and expressive

Facebook Planout introduces a new language specifically for defining experiments

- Leverages hashing functions to simulate randomization
- Standardized JSON representation
- Provides flow control and value distribution functions

Sample Planout Script:

```
buttonSize = uniformChoice(choices=['10px', '20px'], unit=userid);
buttonColor = weightedChoices(choices=['blue', 'green'], weights=[1, 4], unit=userid);
```

Results: 4 variants

size/color	blue	green
10px	10px blue	10px green
20px	20px blue	20px green

Engineering Requirements

Implementation strategy should use layer allocation, hash randomization, and planout experiment specifications

- Support high experiment volume
  Layer based experiment allocation
- Scale with Roblox
  Hash based randomization
- Reduce engineering cost per experiment
  Parameterized experiment variables via **Planout**

Architecture & Implementation

Metadata drives the product behavior

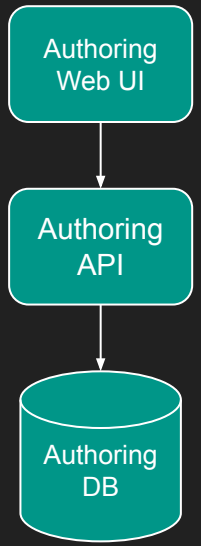
- **Experiment Authoring**
  Enables creation and storage of experiments
- **Experiment Publishing**
  Push the created experiments to production servers
- **Application Integration**
  Grant servers and mobile apps access to experiments
- **Player Behavior Tracking**
  Monitor player behavior for differences between variants

Architecture & Implementation: Authoring

Create a user-friendly interface to make experiment creation as easy as possible

- Experiment Authoring

  Enables creation and storage of experiments

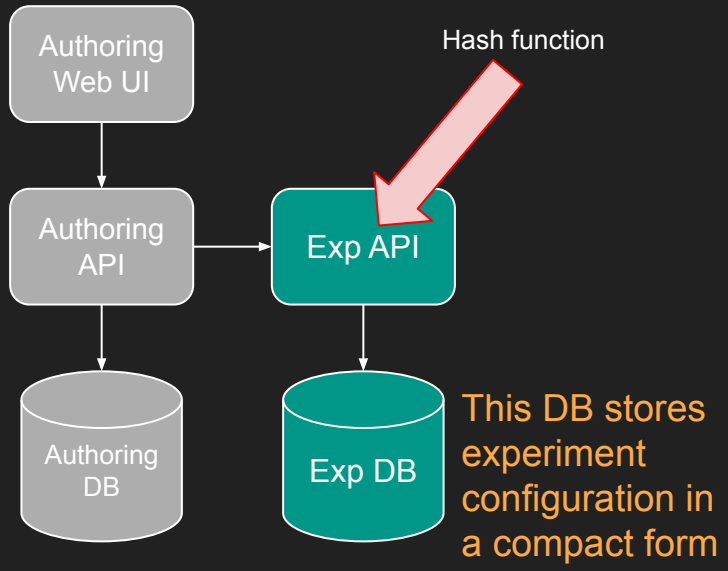


This DB stores experiment configuration in a normalized form

Architecture & Implementation: Publishing

Create a user-friendly interface to make experiment creation as easy as possible

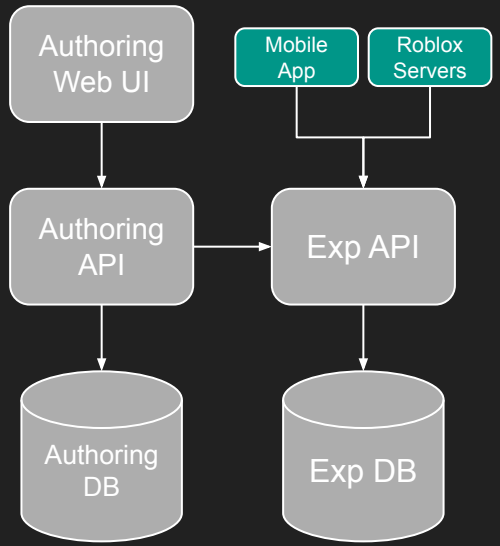
- **Experiment Authoring**
  Enables creation and storage of experiments
- **Experiment Publishing**
  Push the created experiments to production servers



Architecture & Implementation: Integration

Create a user-friendly interface to make experiment creation as easy as possible

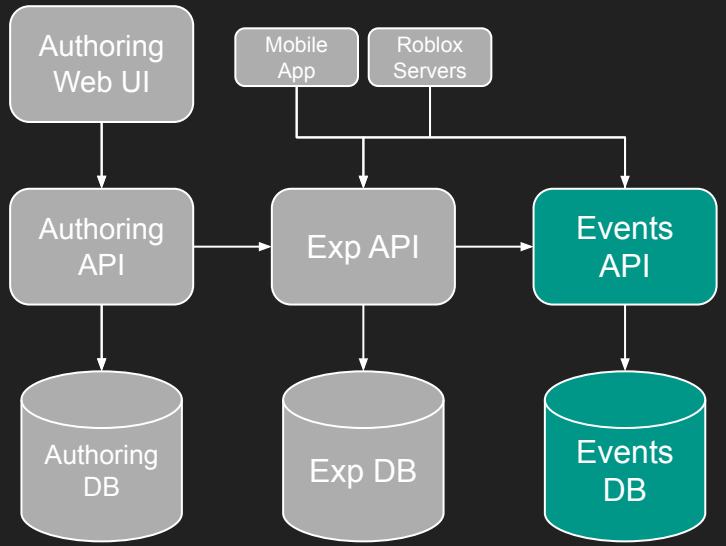
- **Experiment Authoring**
  Enables creation and storage of experiments
- **Experiment Publishing**
  Push the created experiments to production servers
- **Application Integration**
  Grant servers and mobile apps access to experiments



Architecture & Implementation: Player Behavior

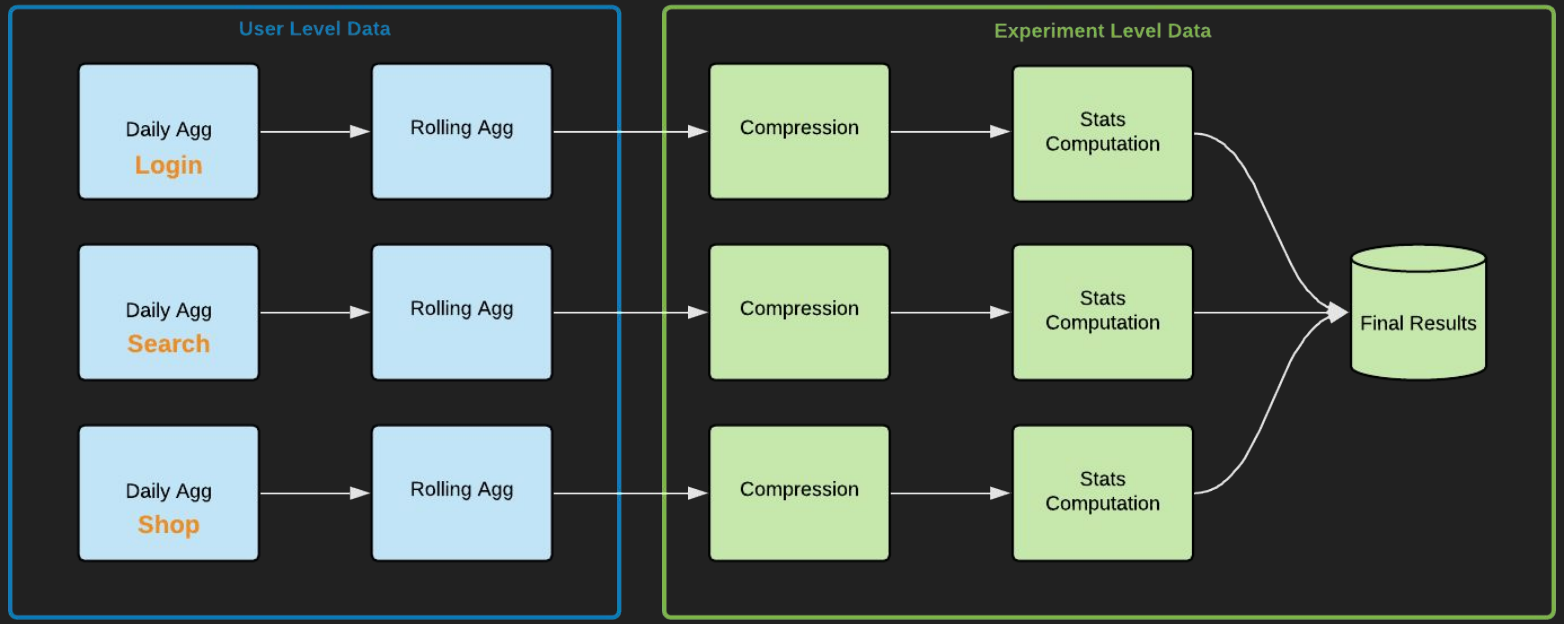
Create a user-friendly interface to make experiment creation as easy as possible

- **Experiment Authoring**
  Enables creation and storage of experiments
- **Experiment Publishing**
  Push the created experiments to production servers
- **Application Integration**
  Grant servers and mobile apps access to experiments
- **Player Behavior Tracking**
  Monitor player behavior



Automated Data Pipelines

Scheduled jobs that crunch data so you don't have to



New User Recommendation: Ninja v.s. Kitten

VS

Data Breadcrumbs (1/2)

Enrollment info

vs

date	user	experiment	variant	...
2021-01-01	meow_284	ninja_v_kitten	ninja	
2021-01-01	I_LOVE_KITTIES	ninja_v_kitten	kitten	
2021-01-02	meow_284	ninja_v_kitten	ninja	
2021-01-02	I_LOVE_KITTIES	ninja_v_kitten	kitten	
...				

Data Breadcrumbs (2/2)

Metrics info - what we seek to improve

vs

date	user	playtime_hour	...
2021-01-01	meow_284	0.5	
2021-01-01	I_LOVE_KITTIES	4.8	
2021-01-02	meow_284	0.7	
2021-01-02	I_LOVE_KITTIES	3.3	
...			

Joining Data Together

Enrollment + metrics

vs

date	user	experiment	variant	playtime_hour	...
2021-01-01	meow_284	ninja_v_kitten	ninja	0.5	
2021-01-01	I_LOVE_KITTIES	ninja_v_kitten	kitten	4.8	
2021-01-02	meow_284	ninja_v_kitten	ninja	0.7	
2021-01-02	I_LOVE_KITTIES	ninja_v_kitten	kitten	3.3	
...					

Rolling It Up Across Time

Rolling aggregation for the entire duration of each experiment

vs

user	experiment	variant	playtime_hour	...
meow_284	ninja_v_kitten	ninja	1.3	
I_LOVE_KITTIES	ninja_v_kitten	kitten	8.1	
...				

Data Compression

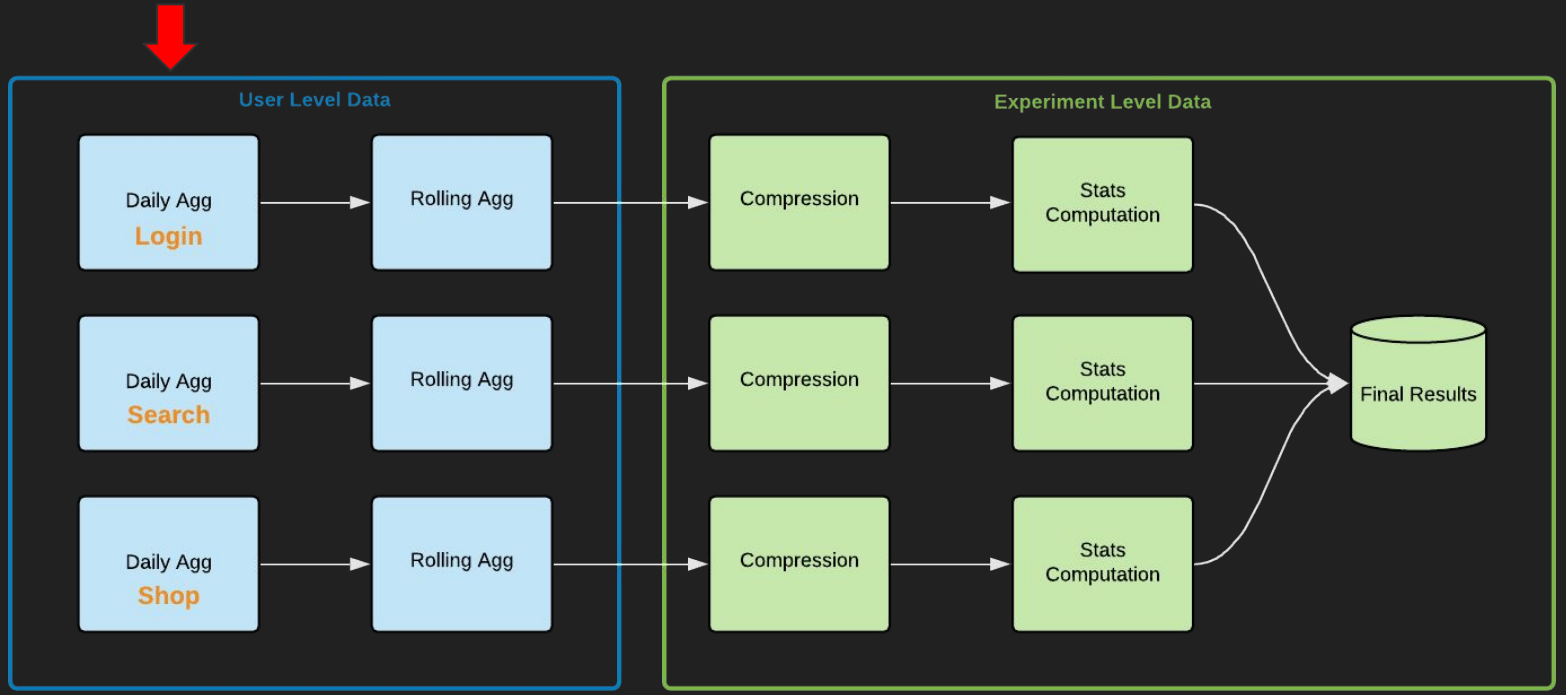
Compress data to balance between data precision and compute cost

vs

experiment	variant	metric_name	compression_method	compressed_value	count	...
ninja_v_kitten	ninja	**playtime_hour**	int_floor	20	1000	
ninja_v_kitten	ninja	**playtime_hour**	int_floor	30	2000	
...						

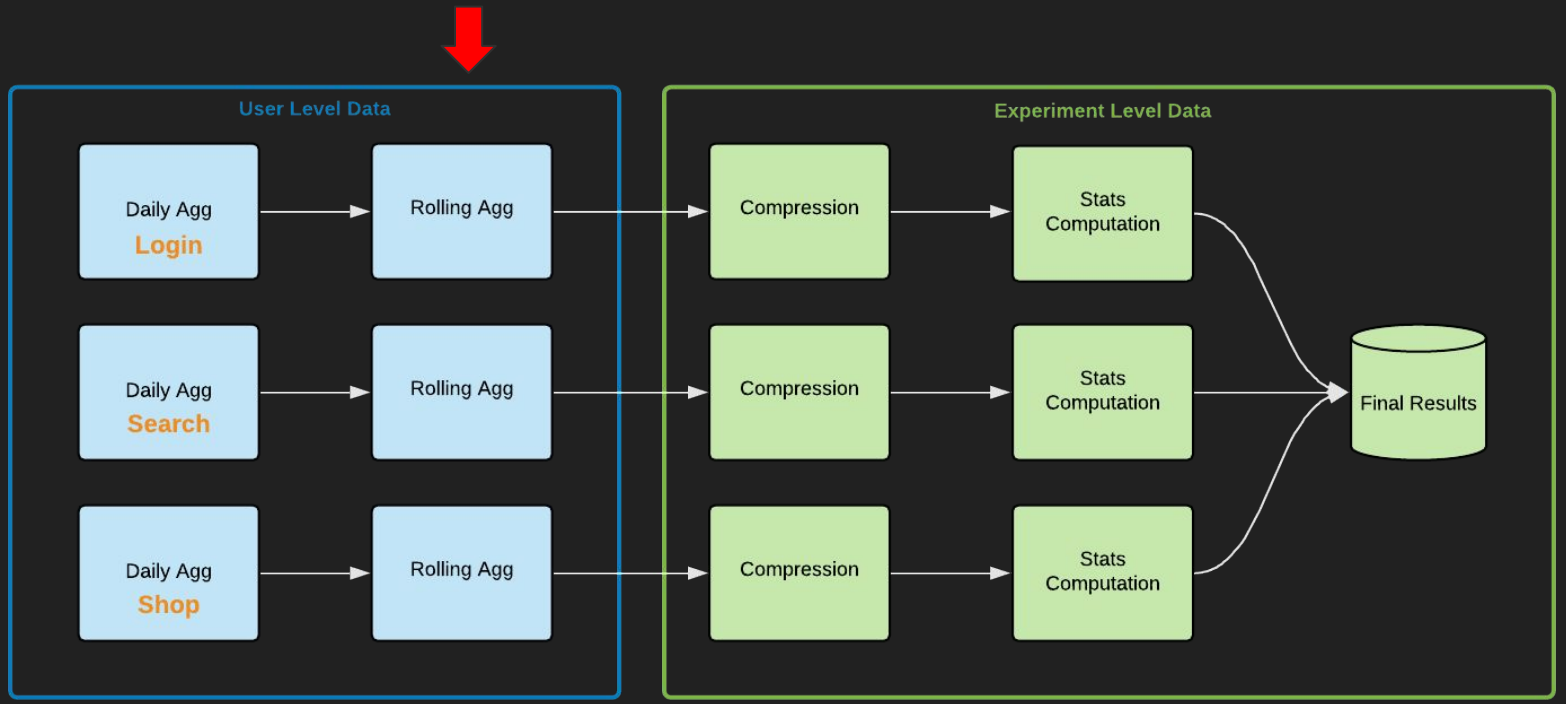
Data Aggregation

Combine data across sources to see the bigger picture



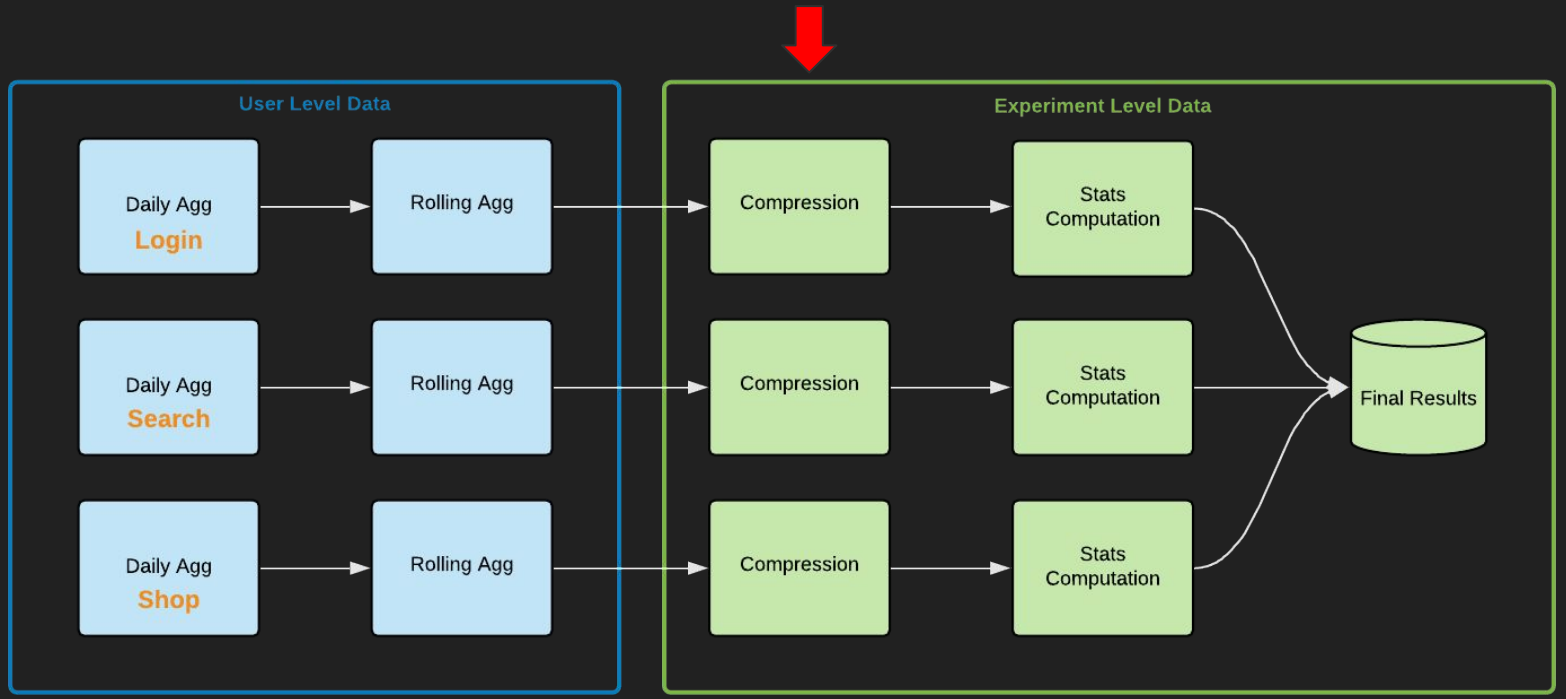
Rolling It Up Across Time

Rolling aggregation for the entire duration of each experiment



Data Compression

Compress data to balance between data precision and compute cost



Designing for Scalability and Maintainability

Building Robustness Into the System

- **Parallelization**
  Break things down into smaller parts that can run at the same time
- **Self-healing / Graceful Degradation**
  Automated retries, partial failures, etc.
- **Config-driven Code**
  Low or no code change required to add new metrics or aggregation logic
- **Alerting / Error Logging**
  Manual intervention when required and auditing for performance measurement

Learning Through Experiments

Learning Through Experiments

Variant A

Variant B

Learning Through Experiments

Variant A

Variant B

Goal metrics: *average daily playtime*

variant A > variant B => Better engagement!

by chance?

[Problem] How to define “by chance”?

10 coin flips

8 Heads 2 Tails

By chance? Or biased coin?

[Solution] Hypothesis Testing & P-values

10 coin flips

8 Heads 2 Tails

H0: Null Hypothesis p = 0.5

H1: alternative hypothesis p != 0.5

P values: how extreme the observation is?

Complications

What if we are testing

multiple groups (variants)?

multiple outcomes (metrics)?

Variant A

Variant B

Variant C

[Problem] Multi-test Correction

Multi-test is like flipping N coins simultaneously

Flipping five **fair, independent** coins together, much larger chance of getting a head!

$$50\% \rightarrow 1 - (50\%)^5 \rightarrow 97\%$$

[Solution] Automatic Multi-test Correction

- **Multi-test adjusted p value threshold**
  - Bonferroni
  - FDR, False Discovery Rate

Remember adding metric is not free!

Complications

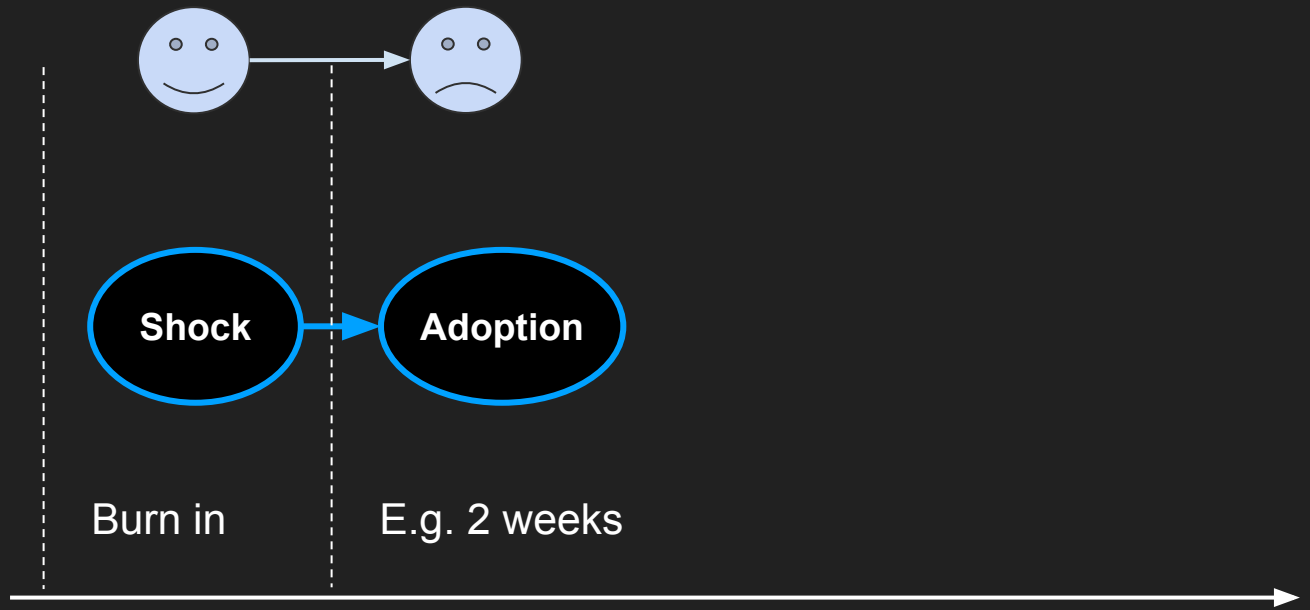
What if we are testing

- multiple groups (variants)?
- multiple outcomes (metrics)?

What if how user responds to variant change over time?

[Problem] User Behavior Changes with Time

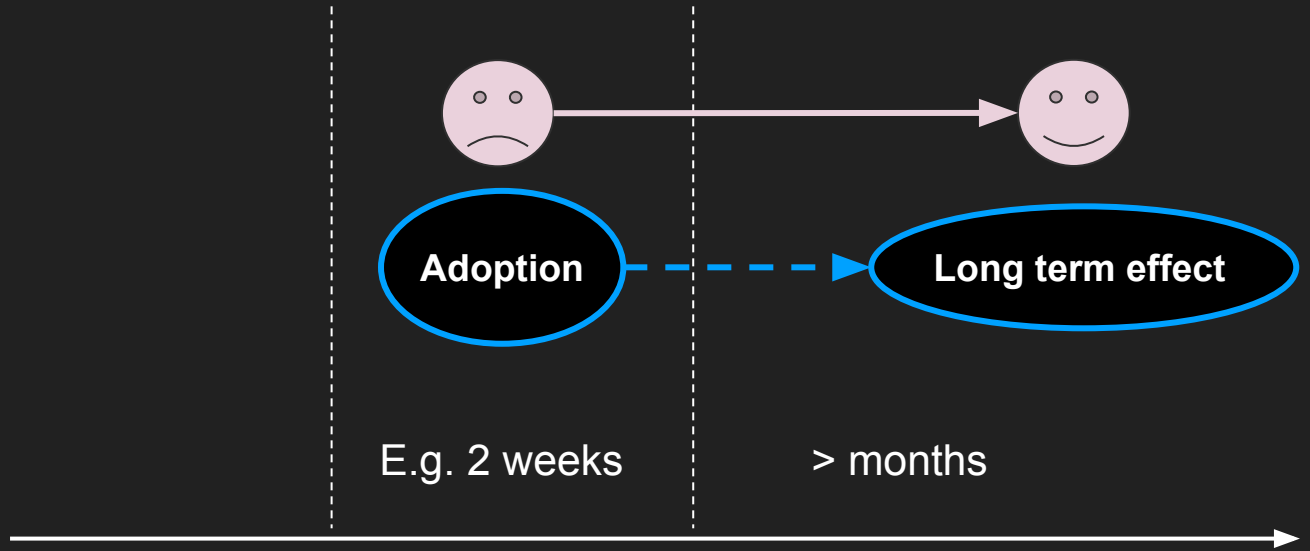
User Novelty Effect



[Problem] User Behavior Changes with Time

User Novelty Effect

Long Term Effect



[Solution] Burn-in Period & Holdout

User Novelty Effect:

Estimate it using more complex behavioral models

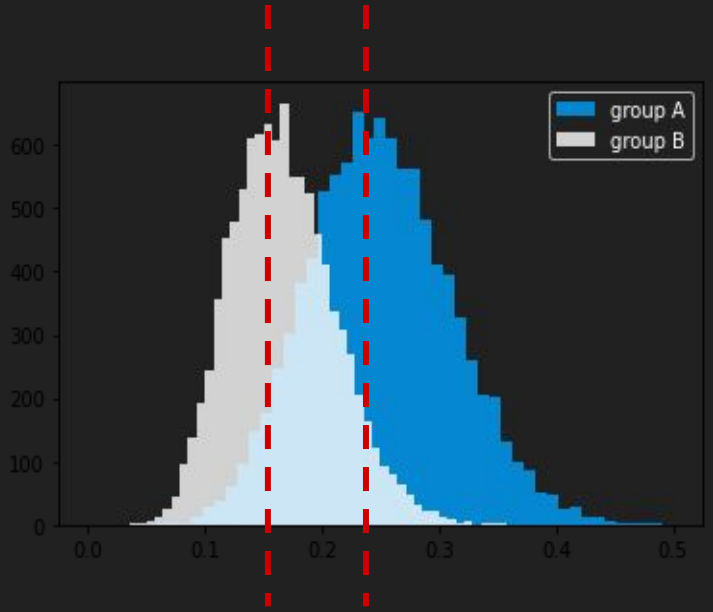
Drop the burn-in periods

Long Term Effect:

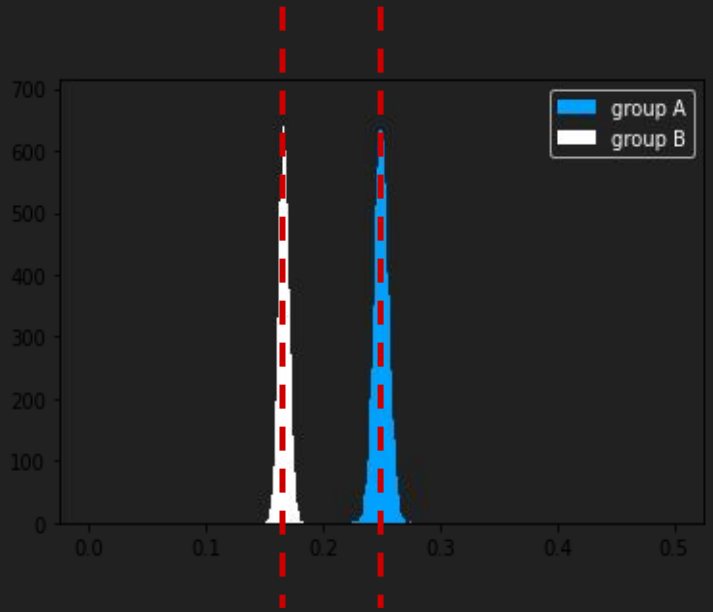
Holdout group of users

More Complex Experiment Setup

[Problem] Higher Volatility, Smaller Power of Tests



High Volatility



Low Volatility

[Solution] Automatic Variance Reduction

Block Design

Step 1: Grouping similar users into blocks

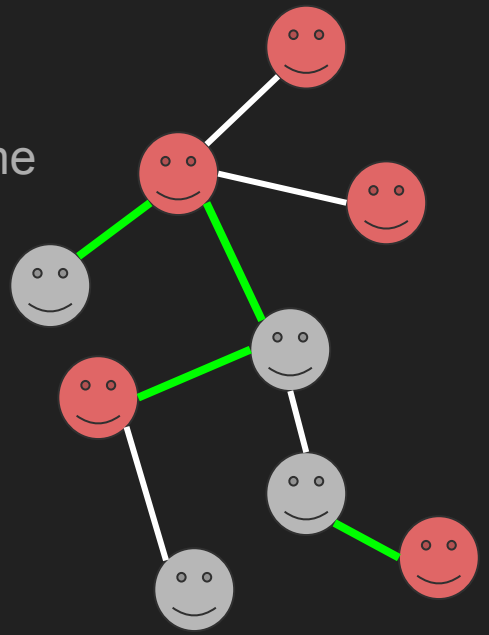
Step 2: Estimate the variants' difference within each blocks

[Problem] Network Effect

Interference between different variant groups

E.g. you (variant A, red group) are recommended a new game

and coplay with your friend (variant B, grey group)



[Solution] Network Clusters in Same Variant Group

- Multi-level design:
  - Divide the network into clusters
  - Randomly assign variant to clusters
  - Assign same variant to all users within the same clusters

- Switchback experiment

Takeaways

Data Scientists are a critical part of a well-functioning product team

Data Scientists use experiments to establish causal relationships and guide decision making

Experimentation in tech happens at massive scale and relies heavily on automated systems to solve challenges like:

- How do we randomize user experiences?
- How do we get experiment data?
- How do we correctly interpret experiment results?

Q&A

ROBLOX

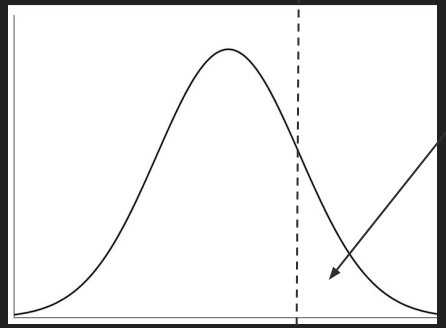
Q&A

Null Hypothesis Testing & P Values (RECAP)

Null Hypothesis Testing Framework:

H0: Null hypothesis variant A = variant B

H1: Alternative hypothesis variant A != variant B



Observed value

P value:
How likely more extreme values than observed given H0 is true?

5%

[Problem] Fail to Randomize Variants: Noncompliance

Opt-in opt-out of treatment. Treatment is not random!

E.g. experiments of new medicines on patients

treated users who opted in of the treatment is correlated with the outcome metrics

Problem categories: non-compliance

Solutions: quasi experiment techniques e.g. instrument variable