



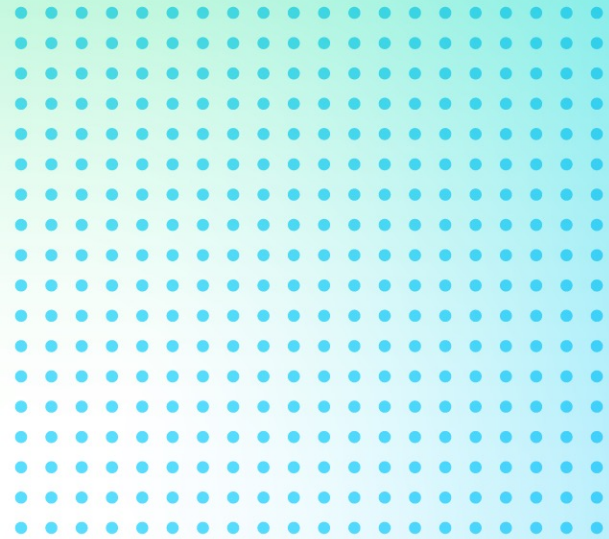
Empowering your business with Feature Experimentation



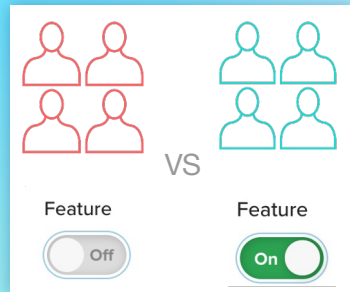
Hristo Bakalov a.k.a The Project Owner
Solution Architect, Optimizely



Kunal Shetye a.k.a The Solution Architect
Senior Solution Architect, Optimizely

- 
1. Introduction to Feature Experimentation
 2. Demo
 3. Q&A

Introduction to Feature Experimentation



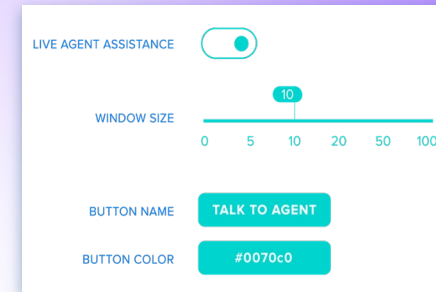
Feature Flagging

Create features in Optimizely to enable feature release and experimentation without a code deploy.



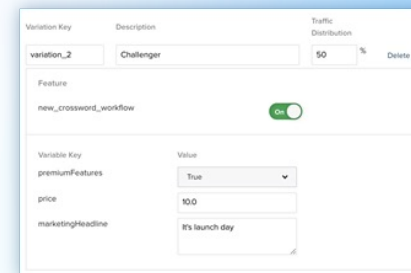
Feature Rollouts

Gradually roll out a feature to monitor impact on performance and business metrics.



Feature Variables

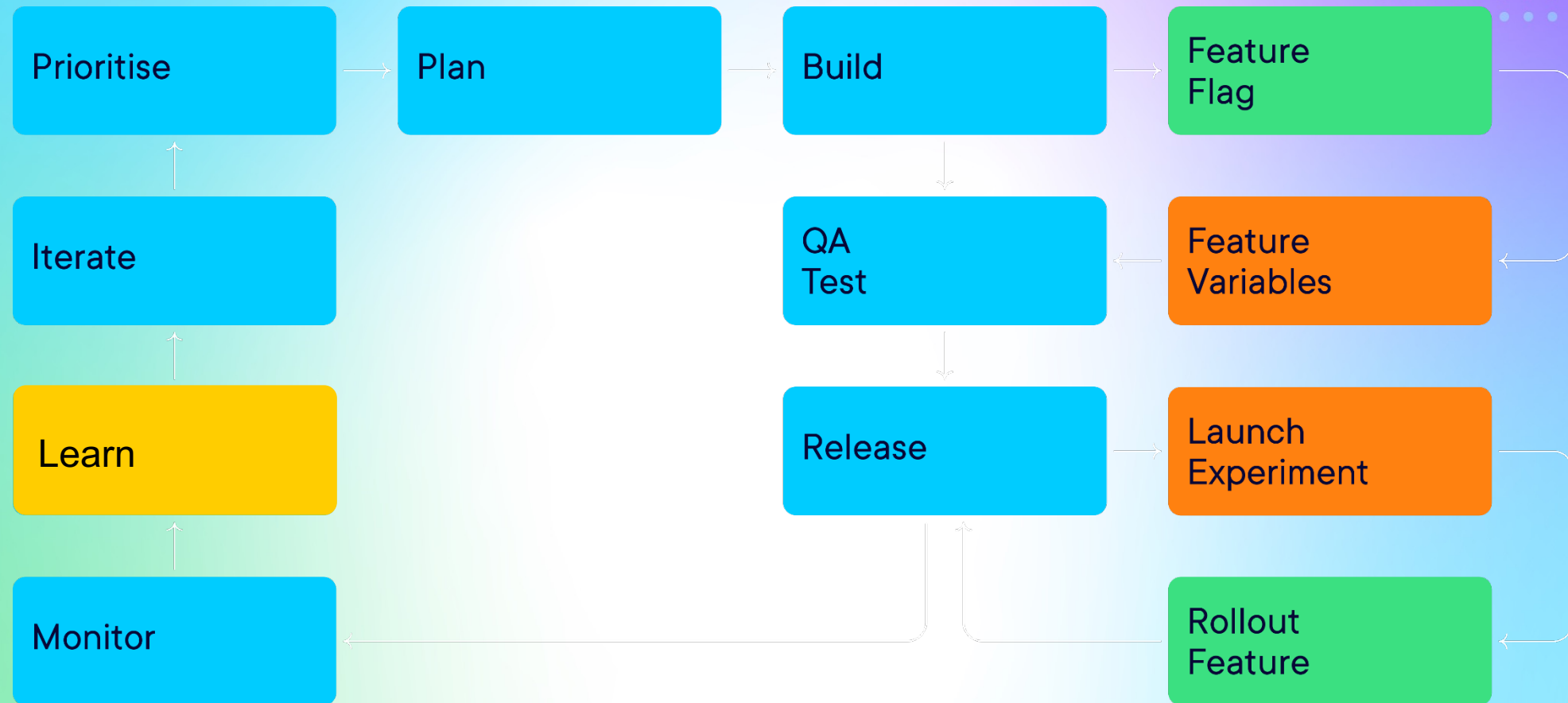
Dynamic variables that can be tested without updating code.



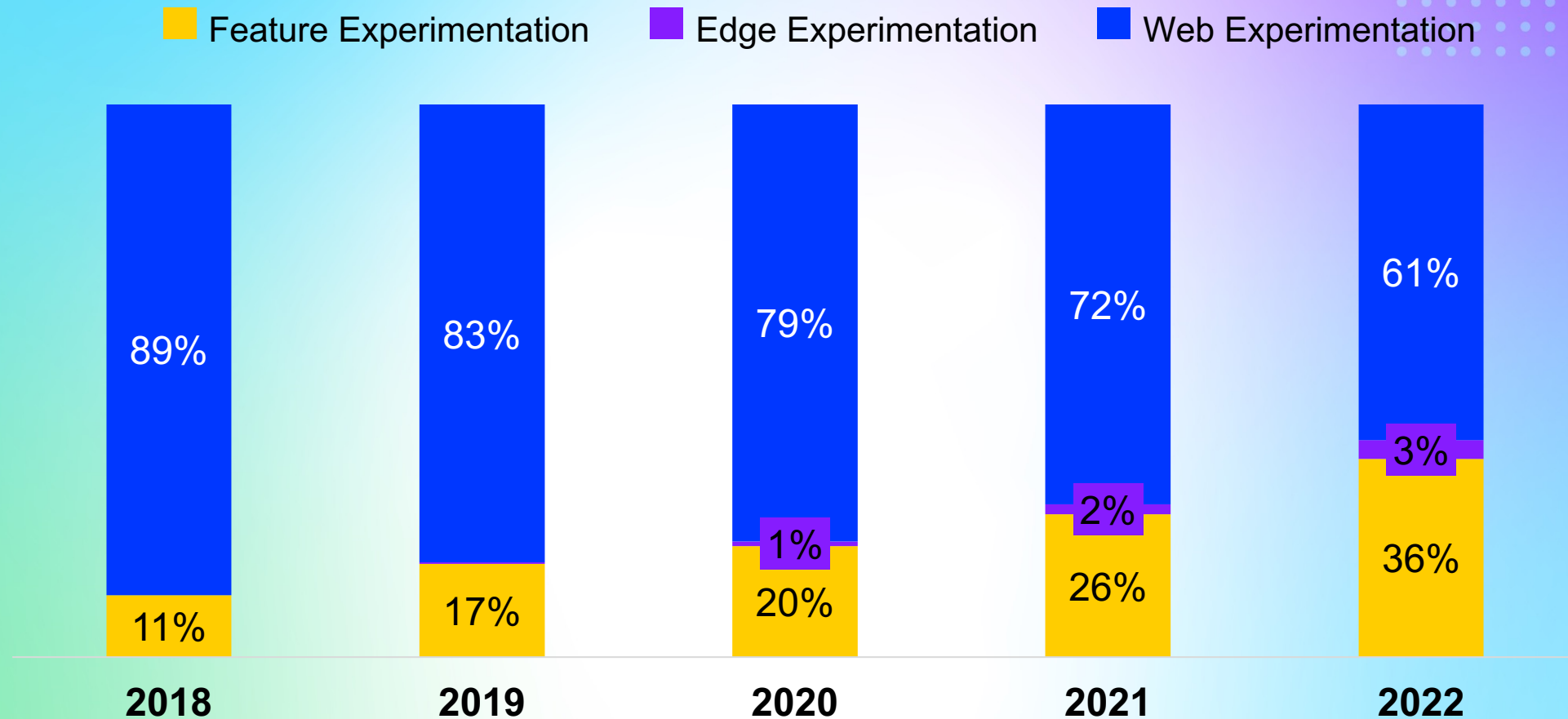
A/B/n & Feature Tests

Run experiments on existing features and elsewhere to optimize your application.

Test and Learn as part of your Software Development Cycle



Server-side experimentation is becoming increasingly adopted as maturity grows



The Plan



Ideation

● Test Ideas

Backlog of Ideas

A Central repository for all our experiment ideas

- New Features
- Design Experiments
- MVP product changes



Define
& Design

● Test Product

Minimum Viable

Lets see how many people are interested in a new feature before we do the work of integrating

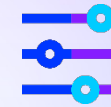


Code
& Release

● Feature Flag

Launch New Feature

Engineers look to build our experiment as a feature in the codebase wrapped in a feature flag in order to control its release.



Measure
& Iterate

● Feature Experiment

Launch New Feature as an Experiment

We want to ensure the feature does deliver against our KPIs, we also wanted to validate if being shown the price made a difference



Optimize
& Grow

● CRO & Experimentation

Experiment and Learn

Conversion rate optimization on Attract, Engage, & Convert parts of your site. Enhancing the use experience.

Sprint Planning

The screenshot displays a web application for sprint planning, organized into three main sections: a left sidebar, a central workspace, and a right-hand panel.

Left Sidebar: Contains navigation links for Home, Plan, and Requests. The Plan section is currently active.

Central Workspace:

- Top Bar:** Shows the user 'Experimentation Collab SA' and a search bar.
- Breadcrumbs:** Hypothesis | HPT-231
- Section Header:** New design for User feedback widget (Status: In Progress)
- Tabbed Interface:** Brief (selected), Variations, Experiment, Fields, History.
- Brief Content:**
 - Overview:** This project aims to develop a user feedback widget for customers.
 - Objectives:** Simplify, Enhance, Increase, Optimize, Improve.
 - Success Metrics:** Reduction, Increase, Positive feedback.
 - Goals:**
 - Enhanced User Engagement:** Encourage users to actively engage with our platform by providing feedback.
 - Real-Time Feedback Collection:** Enable immediate feedback collection to understand user needs and experiences.
 - User Experience Improvement:** Utilize feedback to enhance the overall user experience on our platforms.
 - Ease of Use:** Ensure the widget is user-friendly and accessible.
 - Success Metrics:**
 - User engagement rate with the widget.
 - Volume and quality of feedback collected.
 - Impact on overall user satisfaction and experience.

Right-Hand Panel:

- Workflow:** Experimentation Workflow (Jira)
- Task List:**
 - 1. Requirements** (Completed)
 - Add Description
 - Idea Accepted
 - Complete Experiment Template
 - Approve Experiment (Completed by Hristo Bakalov)
 - Add Description
 - Approved
 - 2. Backlog**
 - 3. Creative Design** (2 items)
 - 4. Experiment Build in Web Interface**

Create the feature flag

Mosey Credit Card

FEATURE EXPERIMENTATION

👤 Projects

🚩 Flags

📊 Reports

👥 Audiences

📈 Events

🕒 History

⚙️ Settings

📖 Tutorial

Slack Community

📖 Developer Docs

👥 Program Management

🔍 Help

Hristo B. ▾

Account Settings

Profile

Log Out

©2010–2023 Optimizely. [Privacy](#)

Flags

All Flags Exclusion Groups

Flags are decision points in your code and rules are used to run experiments and rollouts.

Find Flags

Environment
Production ▾

Enabled
Any ▾

Rules
Any ▾

Archived
No ▾

Create New Flag...

Flag	Key	Rules in Production		Enabled in Production	Created ▾	Modified	
New module	new_module	Experiment	1	Yes	Oct 17, 2023	Oct 17, 2023	⋮
Article settings	article_settings	Experiment Targeted Delivery	1 1	Yes	Aug 23, 2023	Oct 17, 2023	⋮
Flash Sale Determine impact on the revenue by using a Flash Sale	discount	Experiment	1	Yes	Jul 11, 2023	Oct 27, 2023	⋮
Checkout Flow Decide on which checkout flow works the best	checkout	Experiment	1	Yes	Jul 11, 2023	Sep 25, 2023	⋮
Inventory on PDP Flag to either show or hide inventory on the product detail page.	inventory_on_pdp	Experiment Targeted Delivery	1 1	Yes	Jun 8, 2023	Aug 25, 2023	⋮
User car recommendations	user_car_recommendations	Targeted Delivery	1	Yes	Jun 5, 2023	Jun 6, 2023	⋮
Show trade car button	show_trade_car_button	Targeted Delivery	1	Yes	Jun 5, 2023	Jun 6, 2023	⋮
Include car picture	include_car_picture	None		Yes	Jun 5, 2023	Jun 6, 2023	⋮
New Page design	new_page_design	Experiment	1	Yes	Jun 1, 2023	Jun 1, 2023	⋮

Wrap the code in the flag

The screenshot shows a VS Code editor window with the file `JS CheckoutPage.js` open. The code is wrapped in a flag, and the Explorer sidebar shows the project structure.

```
1 import { useDecision } from "@optimizely/react-sdk";
2 import { useEffect, useState } from "react";
3 import CountdownTimer from "../CountdownTimer";
4
5 export default function CheckoutPage() {
6   const [checkoutDecision] = useDecision("checkout", { autoUpdate: true });
7   const [discountDecision] = useDecision("discount", { autoUpdate: true });
8   const [price, setPrice] = useState(95.00);
9   const [shipping, setShipping] = useState(4.99);
10  const [discount, setDiscount] = useState(0.00);
11  const [total, setTotal] = useState((price + shipping).toFixed(2));
12  const [showError, setShowError] = useState(false);
13  useEffect(() => {
14    if(discountDecision.enabled){
15      console.log(total);
16      setDiscount((1.00 - discountDecision.variables.discount_percent / 100).toFixed(2));
17    }
18  }, [checkoutDecision, discountDecision]);
19
20  return (
21    <>
22      <style>
23        {
24          @layer utilities {
25            input[type="number"]::-webkit-inner-spin-button,
26            input[type="number"]::-webkit-outer-spin-button {
27              -webkit-appearance: none;
28              margin: 0;
29            }
30          }
31        }
32      </style>
33      <div className="h-screen w-screen bg-gray-100 pt-10 text-black">
34        <p className={` ${showError && checkoutDecision.enabled ? "":"hidden"} sticky bg-slate-400 top-0 mb-10 text-center text-6xl text-amb<
35        <h1 className="mb-10 text-center text-2xl font-bold">Cart Items</h1>
36        {discountDecision.enabled && (
37          <div className="flex flex-col mb-10">
```

The Explorer sidebar shows the project structure:

- FX-STORE-DEMO
 - components
 - JS CheckoutPage.js
 - JS CountdownTimer.js
 - JS Header.js
 - lib
 - node_modules
 - pages
 - JS _app.js
 - JS index.js
 - JS qr.js
 - public
 - scripts
 - styles
 - .gitignore
 - TS middleware.ts
 - JS next.config.js
 - { package-lock.json u
 - { package.json
 - JS postcss.config.js
 - README.md
 - JS tailwind.config.js
 - yarn.lock

Let's try it out together



Set feature Rollout to 15% of traffic

The screenshot shows a feature flag management interface. On the left is a sidebar with icons for various settings and a list of environments. The main area is titled 'Rules for Production' and shows a flag toggle set to 'On'. Below this, there's a section for adding rules, currently showing '1' rule with the text 'Add rules to customize delivery or run an experiment' and an 'Add Rule' button. A dropdown menu is set to 'On'. A large text area on the right says 'Add a rule to customize delivery or run an experiment'. At the bottom right are 'Revert' and 'Save' buttons.

Mosey Credit Card
user_feedback_widget
user_feedback_widget

Environments

Production	Primary environment	On
Development	Environment	Off

Flag Setup

Variables

Variations

API Values

History

Scheduled Changes BETA

Settings

Rules for Production

Flag On

1 Add rules to customize delivery or run an experiment Add Rule

For everyone in Production

On

[Learn more about interactions between flag rules.](#)

Add a rule to customize delivery or run an experiment

Revert Save

The image shows a presentation slide with a blue dotted background. At the bottom, there is a white rectangular area containing the text "Add description" in blue. Above this text, there are two small icons: a vertical bar icon and a three-dot menu icon.

Setting up an experiment

Mosey Credit Card

user_feedback_widget

user_feedback_widget

Environments

Production

Primary environment

On

Development

Environment

Off

Flag Setup

Variables

Variations

API Values

History

Scheduled Changes BETA

Settings

Rules for Production

Flag On

The following rule will be evaluated for all visitors

1

Alpha feature launch

Rollout - Everyone: On

On ...

Then, for everyone else

Off

▼

[Learn more about interactions between flag rules.](#)

Select a rule to edit

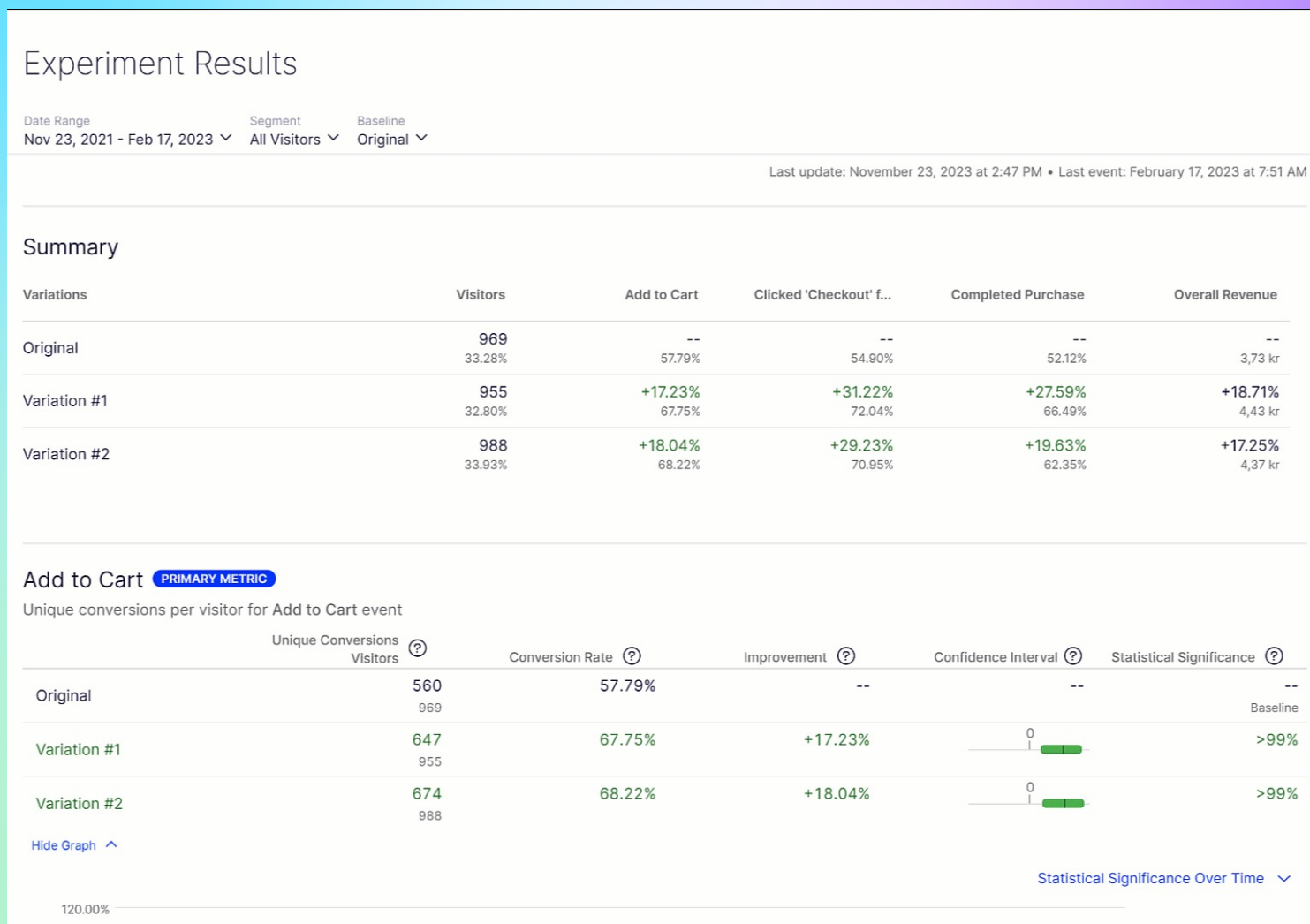
Revert

Save

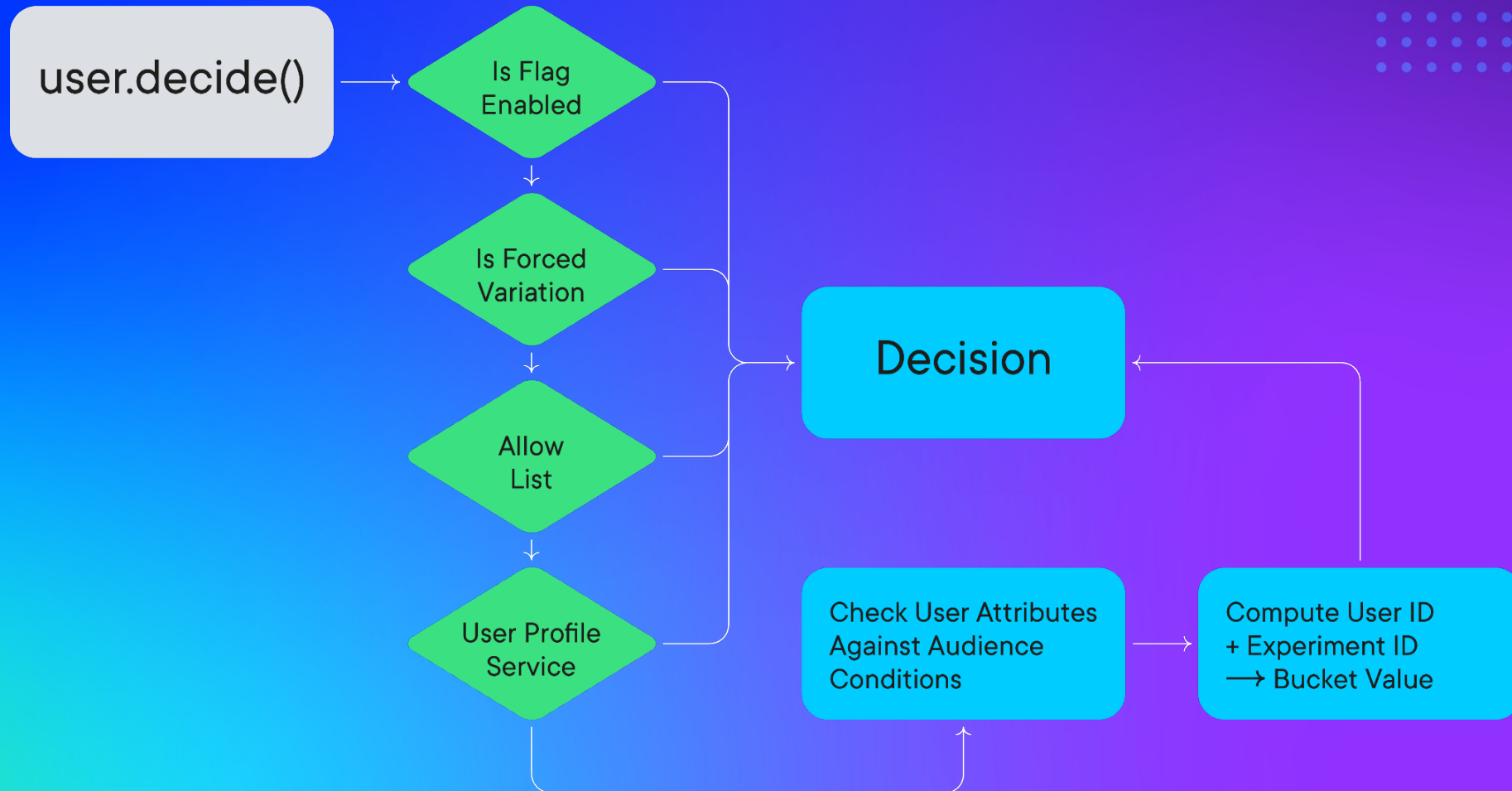
Let's try it out together again



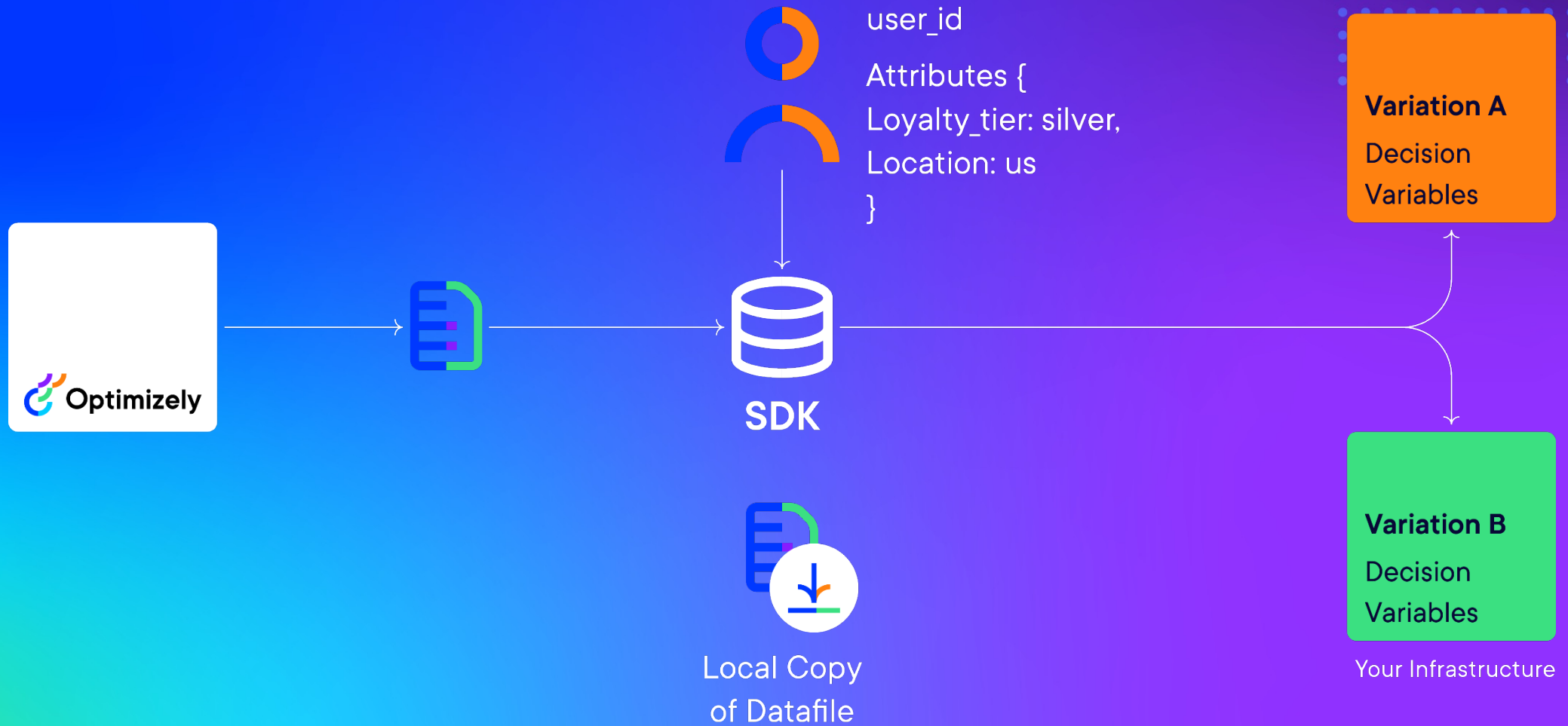
Analyzing the data



User Bucketing



Data file

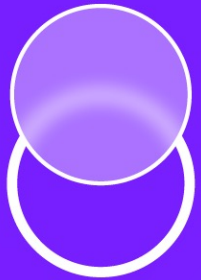


Q&A





Thank you!



For additional questions:

hristo.bakalov@optimizely.com

kunal.shetye@optimizely.com

