

Mobile Ad Integration Testing

From **Tap** to **Reward**



Sergei Suverev

Head of QA **Sweatcoin**

10+ years

in software testing

5 years

in mobile testing

Author

Mobile automation
testing course

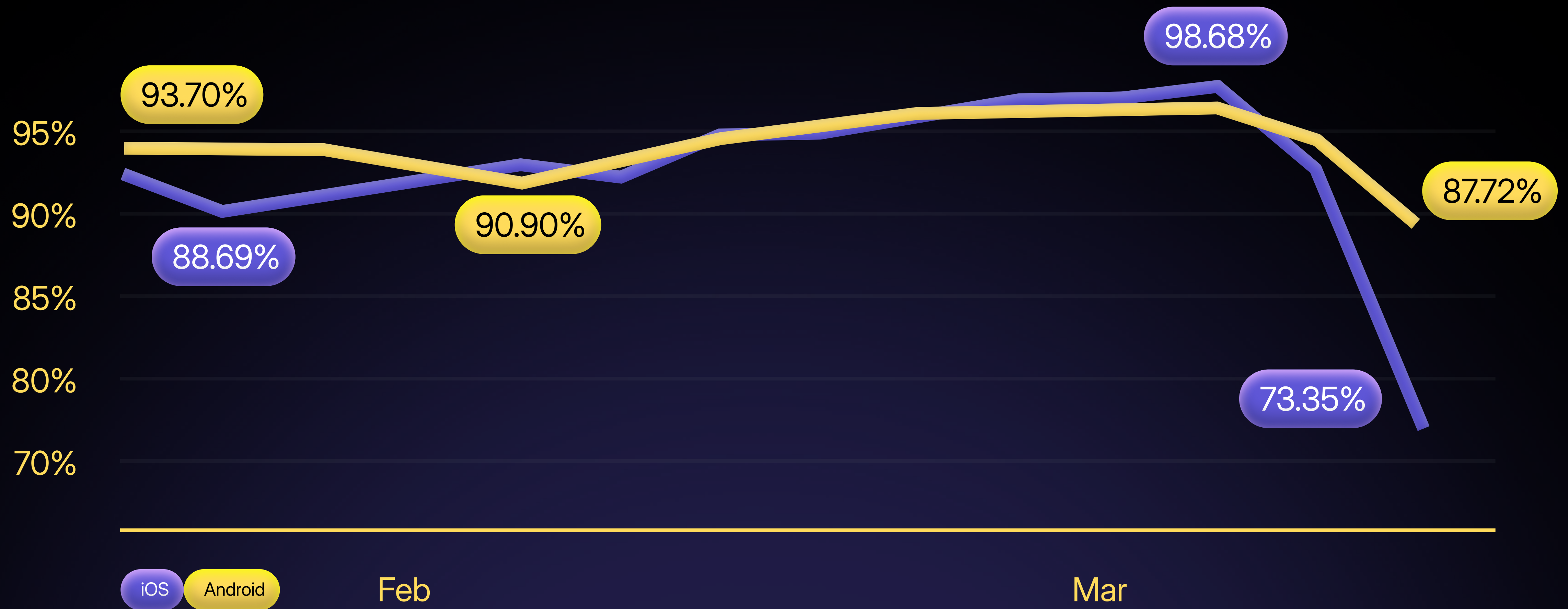


The background is a dark blue gradient. Several light blue lines and arrows are scattered across the image. One line starts from the top left, goes right, then down, then right again, ending in an arrow pointing right. Another line starts from the top center, goes right, then down, then right again, ending in an arrow pointing down. A third line starts from the top right, goes left, then down, then left again, ending in an arrow pointing left. A fourth line starts from the bottom left, goes right, then up, then right again, ending in an arrow pointing up. A fifth line starts from the bottom center, goes right, then up, then right again, ending in an arrow pointing up.

Why does **testing ad SDK**
integration matter?

Ad Conversion

Successful ad watch %



20% Ad Watch Percentage Drop



An update of the AppLovin SDK prevented ad playback if the user had not granted permissions.

Users were unable to watch ads

Ads were taking over a minute to load, leading to a poor user experience.

Why does testing ad SDK integration matter?

1

Ads unlock core features:
rewards & boost

2

High influence
on users engagement and retention

3





3

> 50%

revenue in Sweatcoin
comes from ads

How Works in an App

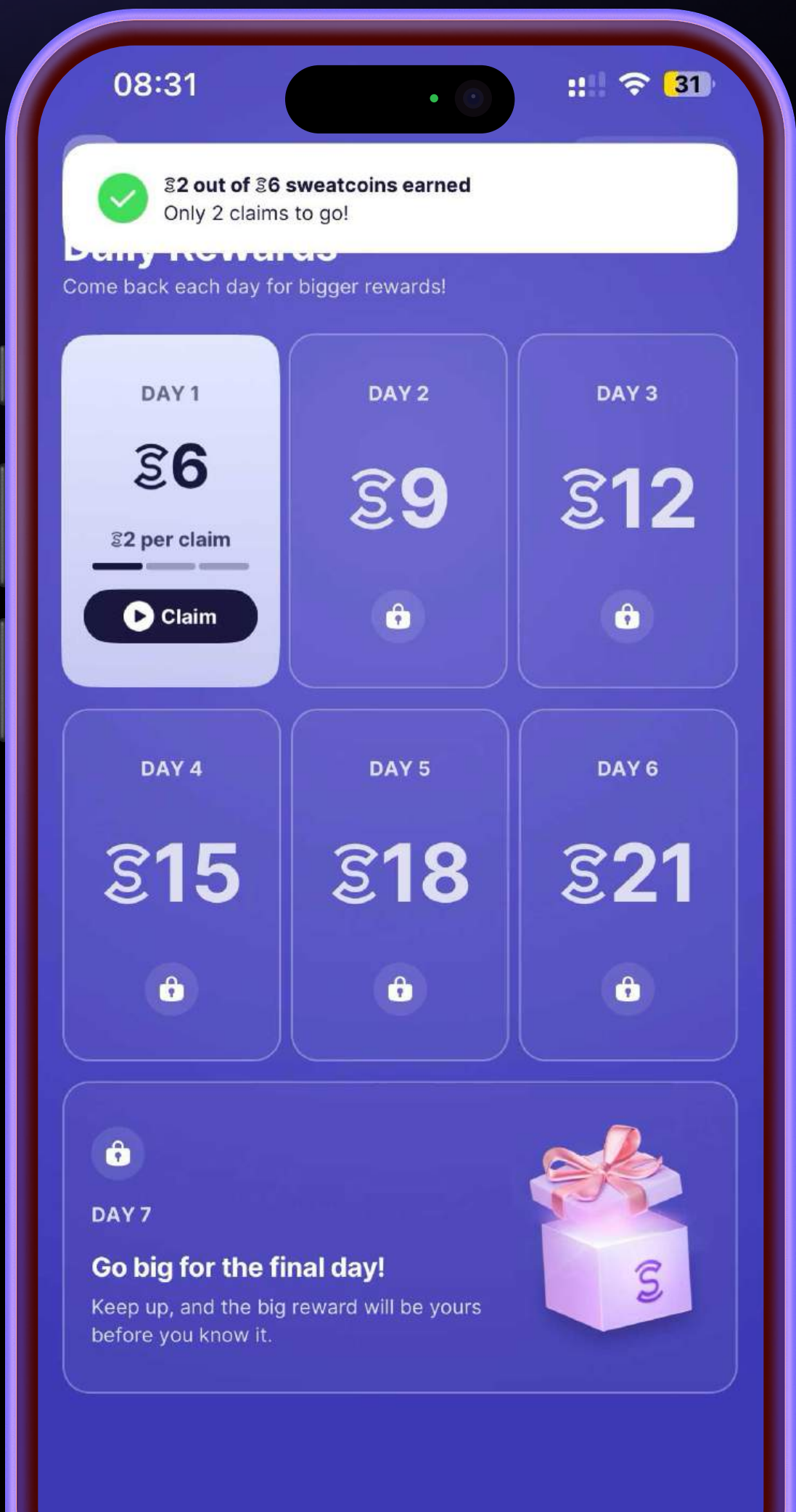
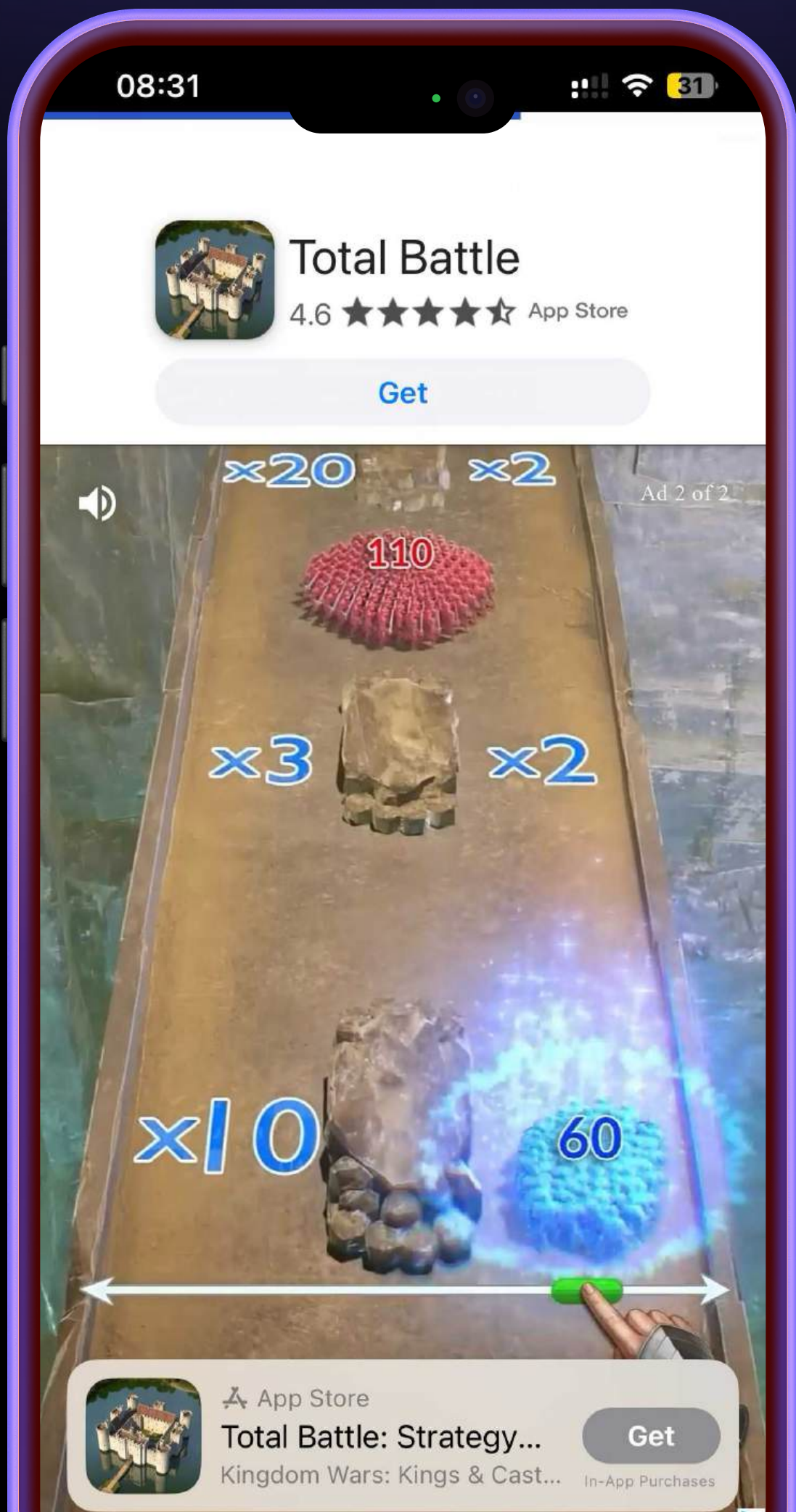
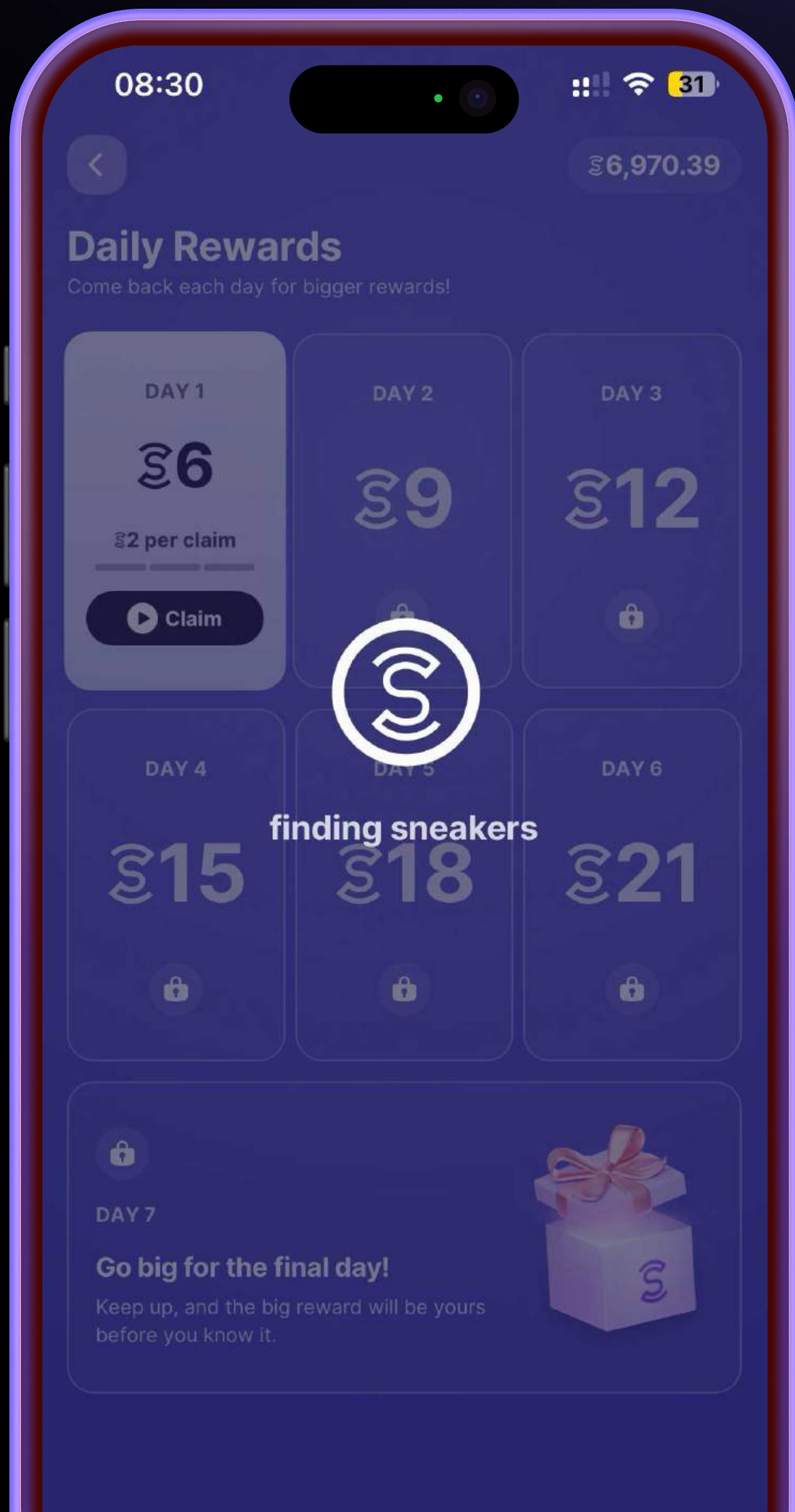
Tap



Watch



Reward



Mediation Flow

Waterfall auction

\$ 3.00 FLOOR

Request →

Partner #1

BID \$2.50

Below floor

Partner #2

BID \$3.20

Winning bid

Partner #3

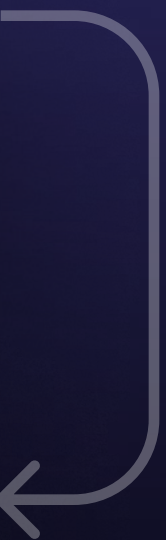
BID \$3.50

Not called

Partner #4

BID \$3.40

Not called



Header auction

\$ 3.00 FLOOR

Request

Partner #1

BID \$2.50

Below floor

Partner #2

BID \$3.20

Losing bid

Partner #3

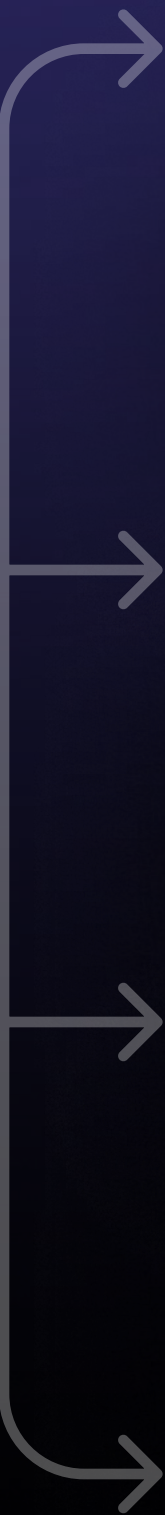
BID \$3.50

Winning bid

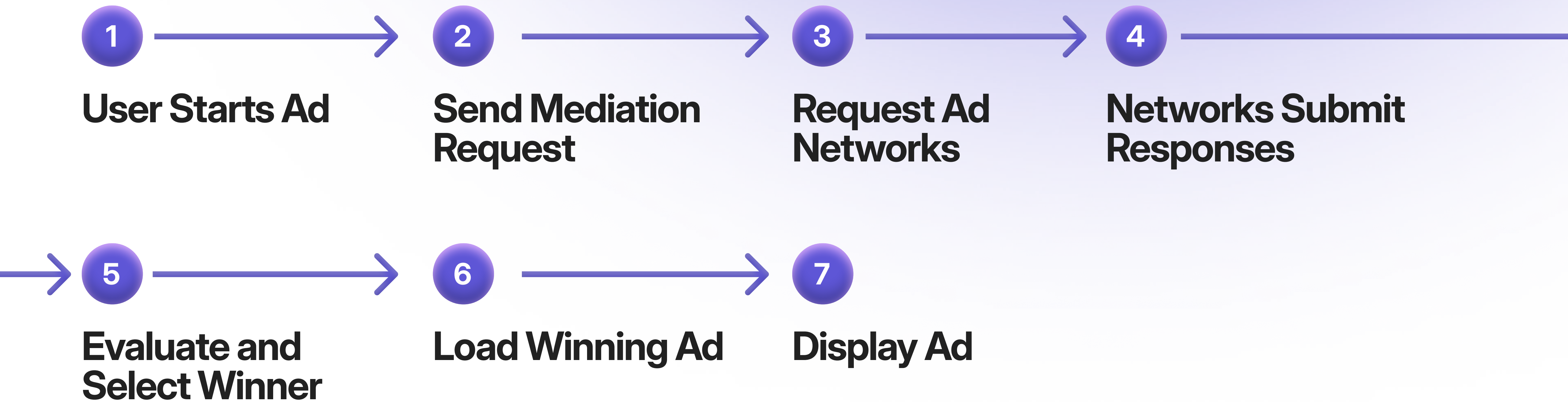
Partner #4

BID \$3.40

Losing bid



7 steps to Ad



**When
Ad breaks?**

Ad Integration
Ad Integration

Ad Integration

Integration

Ad Integration

sweatcoin

Ad Integra

Ad Integration

Ad Integration



Updating the mediation platform SDK



Updating individual ad network SDKs



Changes in SDK initialisation code



Updates to build tools and dependencies



Permissions or network changes



UI or component lifecycle changes



Updates to consent management flows



OS updates

Essential Ad-Integration Test Scenarios

1

**Full cycle feature
interaction with the ad
SDK**

2

**Validation of each ad
network individually
within this flow**

3

**Validation of negative
scenarios and fallback
behaviour**

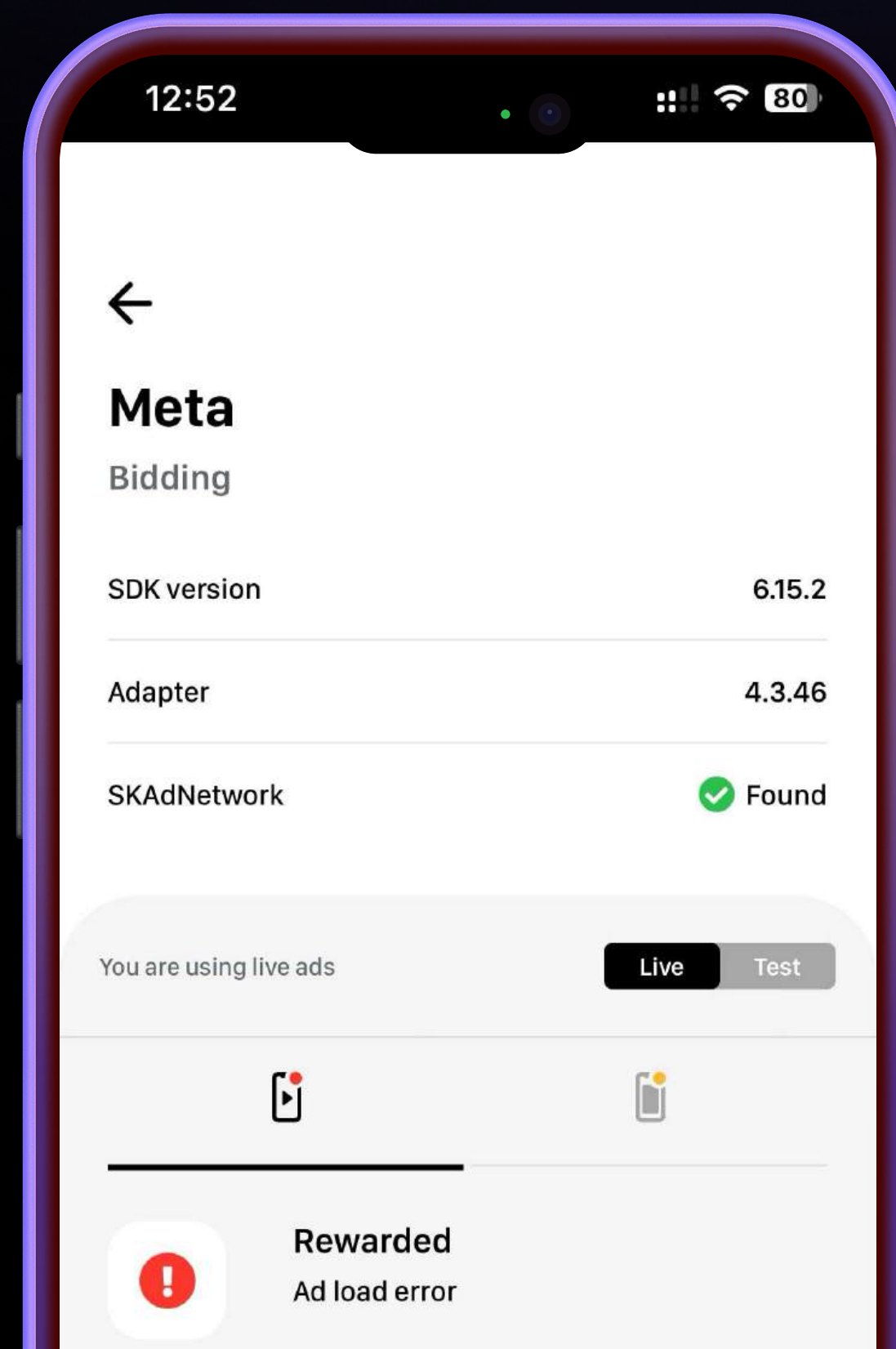
The background is a dark blue gradient. It features several abstract, light blue lines and arrows. One line starts from the top left, goes right, then down, then right again, ending in an arrow pointing right. Another line starts from the top left, goes right, then down, then right again, ending in an arrow pointing down. A third line starts from the top right, goes left, then down, then left again, ending in an arrow pointing left. A fourth line starts from the bottom left, goes right, then up, then right again, ending in an arrow pointing up. A fifth line starts from the bottom left, goes right, then up, then right again, ending in an arrow pointing right.

How to test **specific** **networks?**

How to test specific networks?

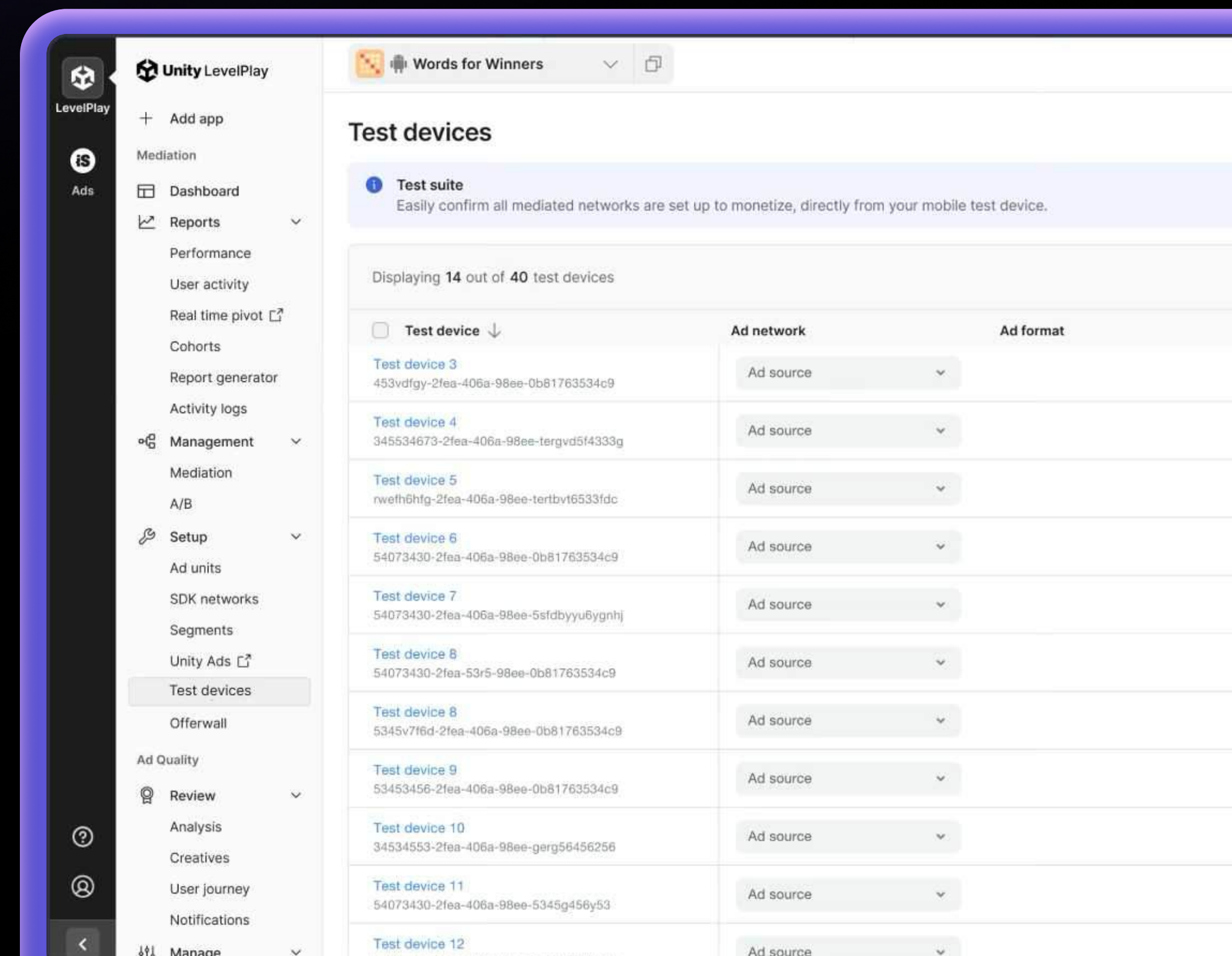
1

In-app testing using SDK tools



2

Via the mediation platform's web dashboard



What to verify?

1

**Display of
different types
of ads**

2

**Callback
correctness**

3

**Error handling
and retry logic**

4

**Fallback
mechanism**

The background is a dark blue gradient. Several light blue lines and arrows are scattered across the frame. One line starts at the top left, goes right, then down, then right again, ending in an arrow pointing right. Another line starts at the top center, goes right, then down, then right again, ending in an arrow pointing down. A third line starts at the top right, goes left, then down, then left again, ending in an arrow pointing left. A fourth line starts at the bottom left, goes right, then up, then right again, ending in an arrow pointing right. A fifth line starts at the bottom right, goes left, then up, then left again, ending in an arrow pointing up.

What else
we could do?

Separate releases for ads updates

Separate releases for ads updates

1

Isolate changes to reduce risk

2

A separate set of smoke tests for ad features and adapters

3

Simplifies rollback with a clear scope if issues arise

4

Improves cross-team coordination (QA, dev, product) on ad updates

5

Provides clear metrics and monitoring windows for ad performance

Yet another
important thing...

type: AdMediationPlatformInit.onInitFinished
payload: {"mediationPlatform":"IronSource","adNetwork": "vungle"}

type: AdMediationPlatformInit.onInitFinished

payload: {"mediationPlatform":"IronSource"}

type: AdMediationPlatformInit.on/nitStarted

payload: {"mediationPlatform":"IronSource"}

Logs -

type: RewardedVideo.onAdLoaded

payload:

{"mediationPlatform":"IronSource","adNetwork": "vungle"}

type: AdMediationPlatformInit.onInitFinished

payload: {"mediationPlatform":"IronSource"}

type: AdMediationPlatformInit.on/nitStarted

payload: {"mediationPlatform":"IronSource"}

Logs -

type: RewardedVideo.onAdLoaded

payload:

{"mediationPlatform":"IronSource","adNetwork": "vungle"}

type: AdMediationPlatformInit.onInitFinished

payload: {"mediationPlatform":"IronSource"}

type: AdMediationPlatformInit.on/nitStarted

payload: {"mediationPlatform":"IronSource"}

type: RewardedVideo.onAdLoaded

payload:

{"mediationPlatform":"IronSource","adNetwork": "vungle"}

type: AdMediationPlatformInit.onInitFinished
payload: {"mediationPlatform":"IronSource","adNetwork": "vungle"}

type: AdMediationPlatformInit.onInitFinished

payload: {"mediationPlatform":"IronSource"}

type: AdMediationPlatformInit.on/nitStarted

payload: {"mediationPlatform":"IronSource"}

Logs -

type: RewardedVideo.onAdLoaded

payload:

{"mediationPlatform":"IronSource","adNetwork": "vungle"}

type: AdMediationPlatformInit.onInitFinished

payload: {"mediationPlatform":"IronSource"}

type: AdMediationPlatformInit.on/nitStarted

payload: {"mediationPlatform":"IronSource"}

Logs -

type: RewardedVideo.onAdLoaded

payload:

{"mediationPlatform":"IronSource","adNetwork": "vungle"}

type: AdMediationPlatformInit.onInitFinished

payload: {"mediationPlatform":"IronSource"}

type: AdMediationPlatformInit.on/nitStarted

payload: {"mediationPlatform":"IronSource"}

type: RewardedVideo.onAdLoaded

payload:

{"mediationPlatform":"IronSource","adNetwork": "vungle"}

You need more logs

Why logging matters

1

Quickly pinpoint integration issues before they affect production.

2

Understand root causes: network errors, mismatched SDK versions, or misconfigurations.

3

Provide end-to-end visibility: from request initiation to ad display event.

Key metrics and events to log:



Network name



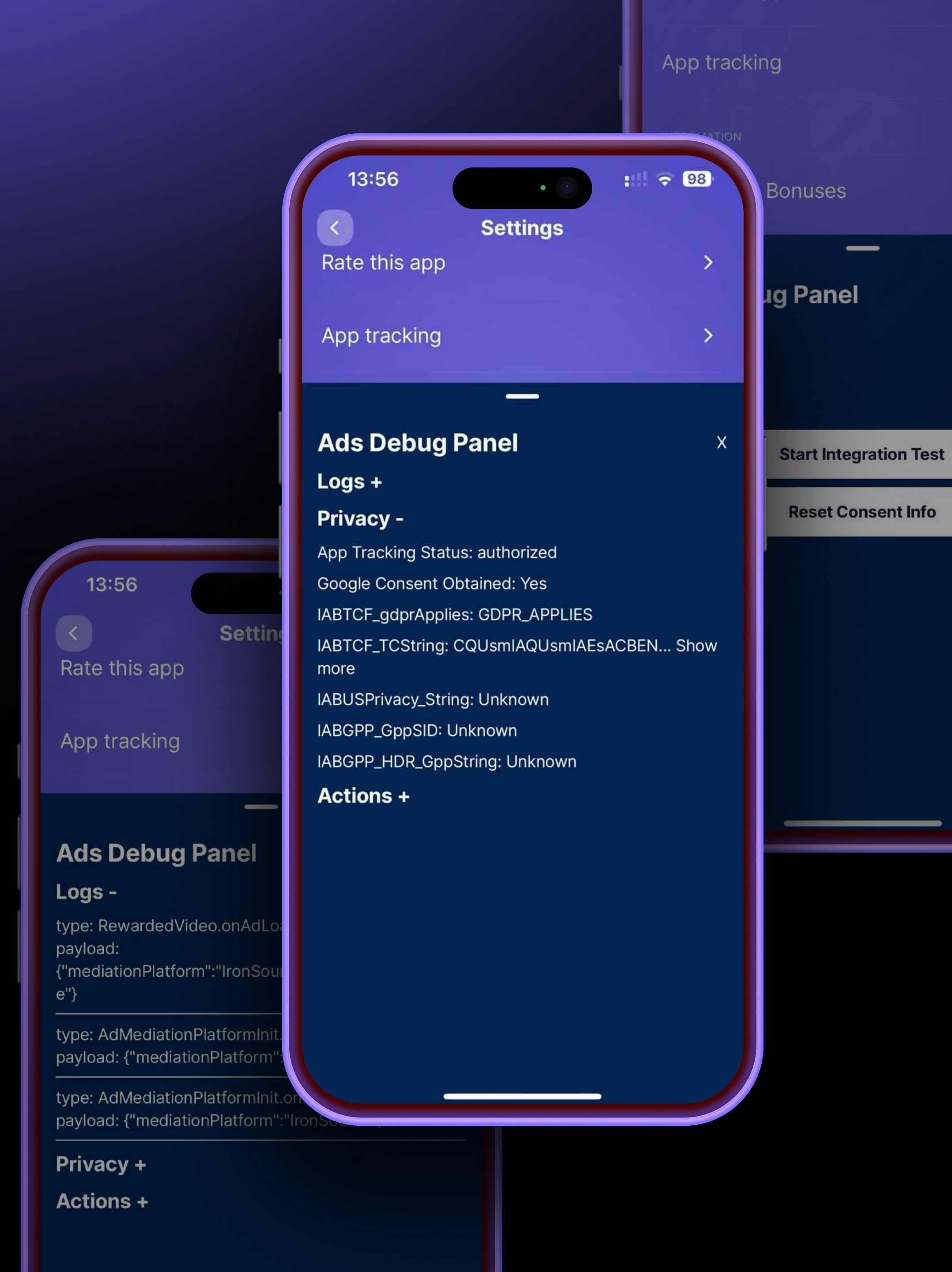
Events with error codes and messages.



Display and interaction events.

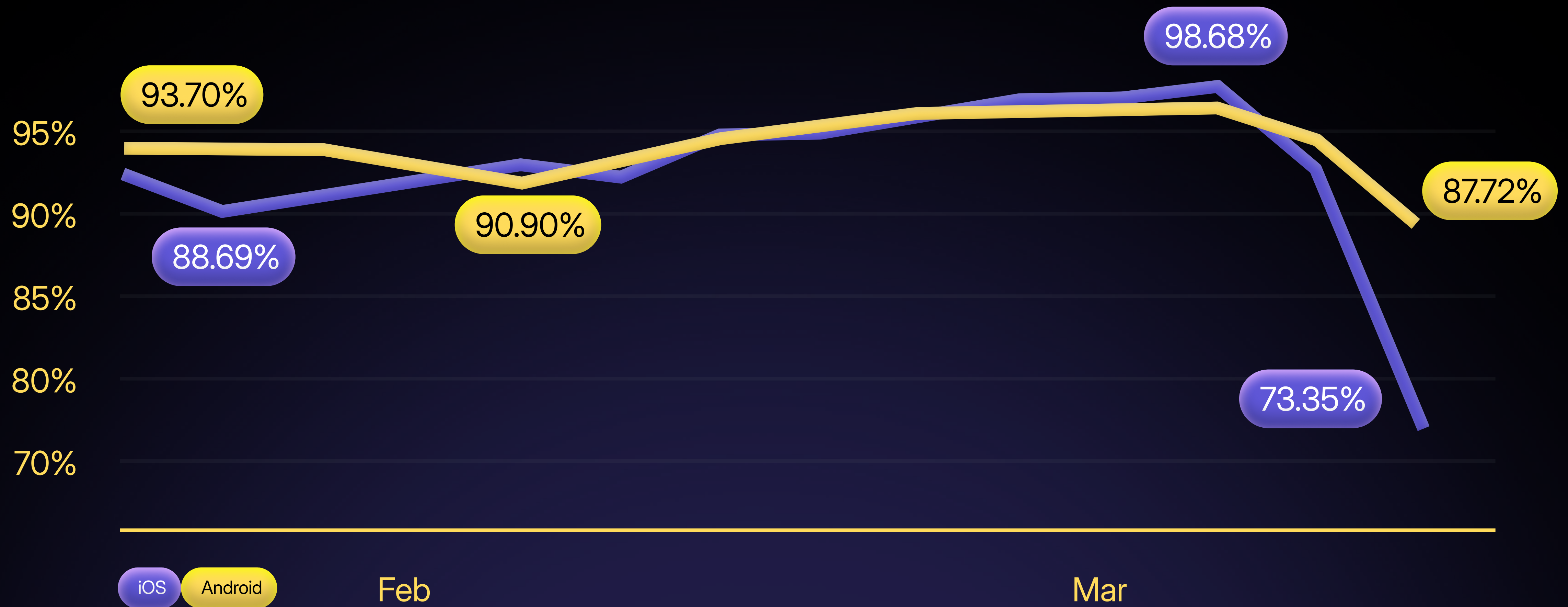


User consent statuses.



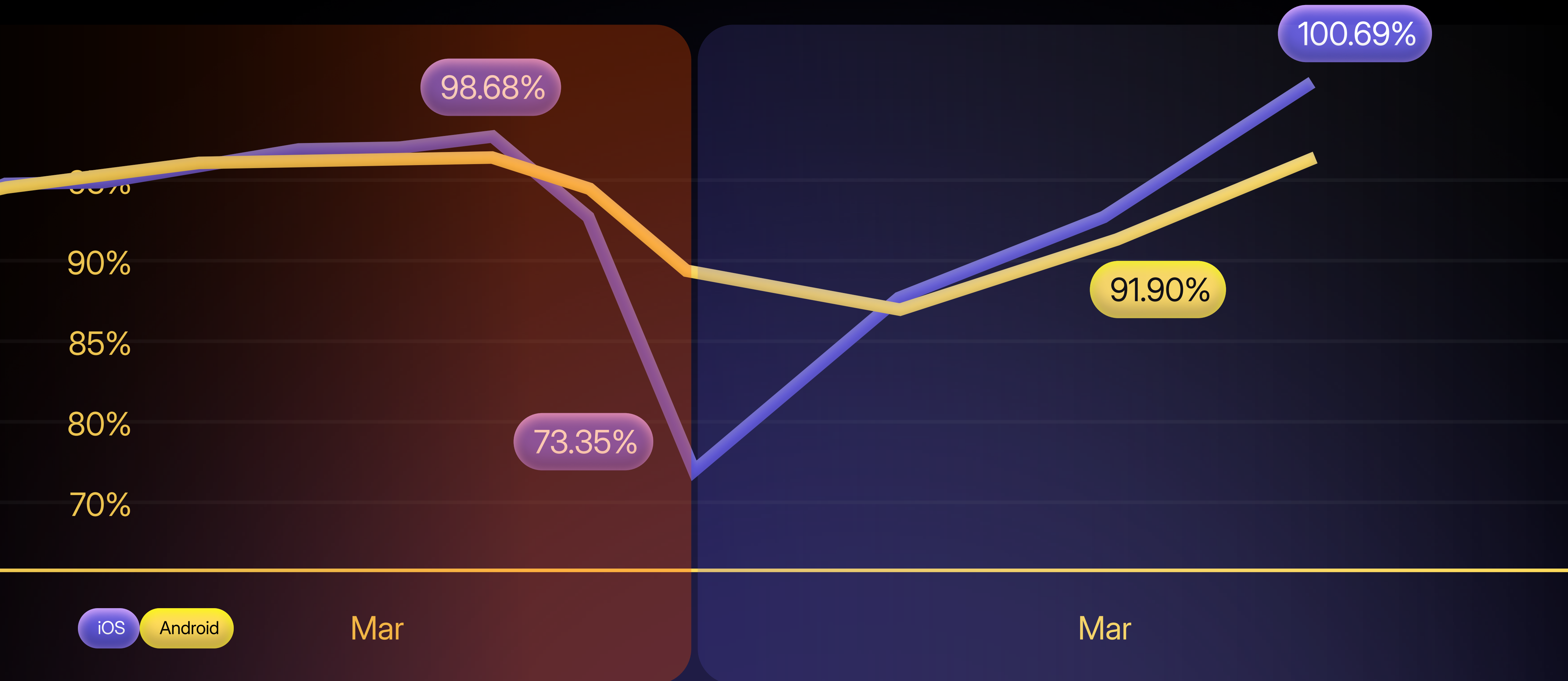
Ad Conversion

Successful ad watch %



Ad Conversion

Successful ad watch %



Key takeaways



Why It Matters

A lot of revenue from ads

One update of SDK = 20% conversion drop

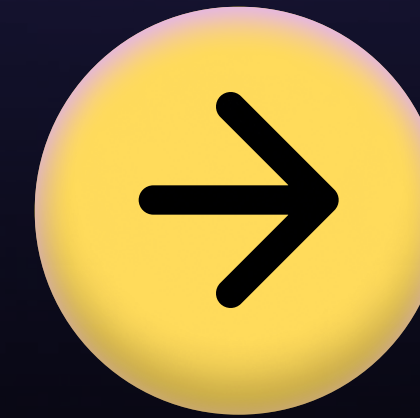


What to Test

Each ad network individually

Full flow: Tap → Watch → Reward

Negative scenarios & fallbacks



How to Do It Right

Separate releases for ad updates

Log everything: networks, events, errors

