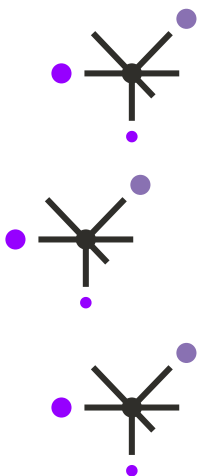


INVESTIGATION AND IMPLEMENTATION

Ad revenue kept falling. Three weeks to fix it.

A complete engagement in two reports: what was wrong, what was built, and what it earned.

Company	Nimbus, a free weather app on iOS and Android (fictional)
Question	Why is ad revenue per user falling, and should we add more ad slots?
Engagement	Initial report on day 4. Implementation report on day 21.
Data	Ad server and mediation reports, analytics, consent logs. All synthetic.
Prepared by	Alex Severianov



This is a sample. The company, people, code and numbers are invented. The problems, the changes and the reasoning come from real ad investigations and integrations I have done.

How this engagement worked

From the first question to money in the bank.

Nimbus asked one question: ad revenue per user had fallen for six months, so should they add more ad slots? This sample contains both documents they received.

+32%

ad revenue per daily user,
measured against a holdout

€24,000

more ad revenue a month at
current traffic

0

new ad slots, and no change in
crashes, start time or sessions

Days 1–4	Investigation. Ad server, mediation and consent data. Found three causes, and showed that more slots would make things worse.
Day 4	Part 1: Initial report. Findings and a plan with expected impact, effort and how each change would be measured.
Days 5–12	Implementation on iOS and Android, each change behind a remote flag.
Day 13	Release in the regular release train. Staged rollout.
Days 14–20	Measurement. 10% of users kept the old behavior as a holdout. One-change-off groups measured each change separately.
Day 21	Part 2: Implementation report. What was built, plan against result, guardrails, and what to do next.

Pages 3–8: initial report, as sent on day 4. Pages 9–15: implementation report, as sent on day 21.

PART 1 · INITIAL REPORT

00_

Executive summary

Sent on day 4

Background

Nimbus earns from ads. Revenue per 1,000 daily users fell from €15.80 to €12.00 a day between January and June (–24%), while traffic stayed stable. The team planned to add two ad slots to recover it.

Objective

Explain the decline, check whether more slots would help, and propose changes with expected impact and effort.

What was done

I broke revenue down into impressions and price per slot, compared consent signals in ad requests with consent logs, checked mediation and bidding setup for every slot, and measured how ads load and refresh on screen.

Key findings

- **Ads go out before consent is read.** Since the consent SDK update in February, 38% of requests from users who had consented carried no consent signal, and sold at a third of the price.
- **Three of seven slots have no bidding.** They carry 41% of impressions at an average €0.71 per 1,000, against €1.59 on slots with bidding.
- **Ads refresh when nobody sees them.** 46% of refreshed impressions were off screen. Viewability is 48%, which lowers what advertisers pay.
- **More slots would not fix it.** The slot added in April raised impressions +18% but revenue per user only +3%, and shortened sessions.

Recommendations

Don't add slots. Make three changes, each behind a remote flag, ship them in the next regular release, and measure them against a 10% holdout. Expected combined effect: +26–43% revenue per user.

01_

What the team believed. What the data shows.

Belief	What I found	Verdict
The ad market got weaker	Prices on slots with bidding and full consent held steady. The decline is in Nimbus's own setup.	Not the cause
Consent is handled correctly	38% of consented users' requests went out before consent was read: 27% of all requests.	Main cause
All slots compete for the best price	Three slots added since last year only use the old waterfall, with no bidding.	Leaking
More impressions mean more money	Off-screen refreshes add impressions advertisers don't value, and drag down viewability for every slot.	Partly
Two more slots will recover revenue	The last new slot added +18% impressions and +3% revenue per user.	Would hurt

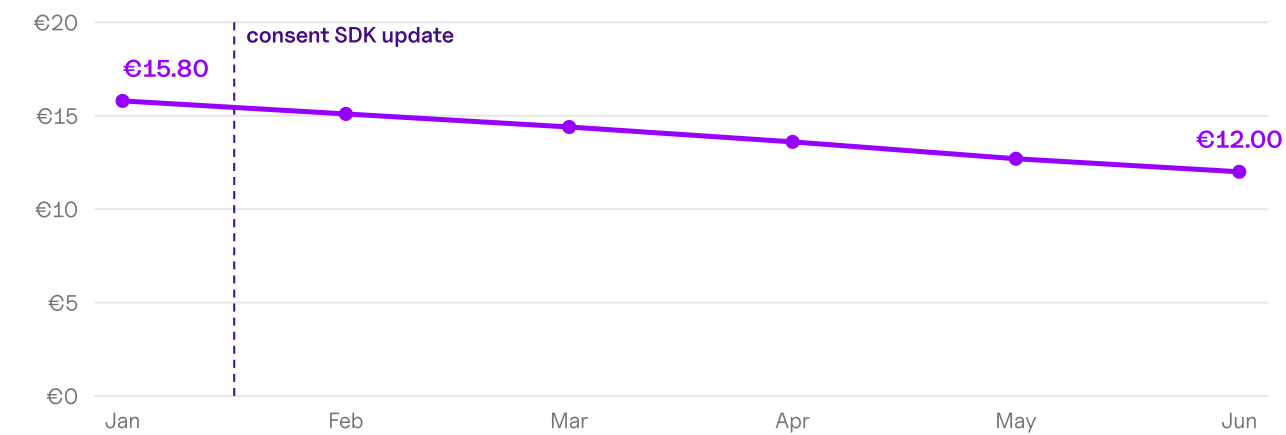
RECOMMENDATION

Fix consent timing first: the biggest effect for three days of work. Then turn on bidding everywhere and refresh only what people can see. Leave the slot count as it is.

02_

Ads were requested before consent was read.

Ad revenue per user started falling in February, the month the consent SDK was updated. The new version loads consent asynchronously. The app still started the ad SDK on launch, so the first requests of every session went out before the user’s consent was known.



Ad revenue per 1,000 daily active users per day

Requests from users who consented	Share	Price per 1,000 impressions
With consent signal: personalised ads	62%	€1.85
Without consent signal: non-personalised ads	38%	€0.62

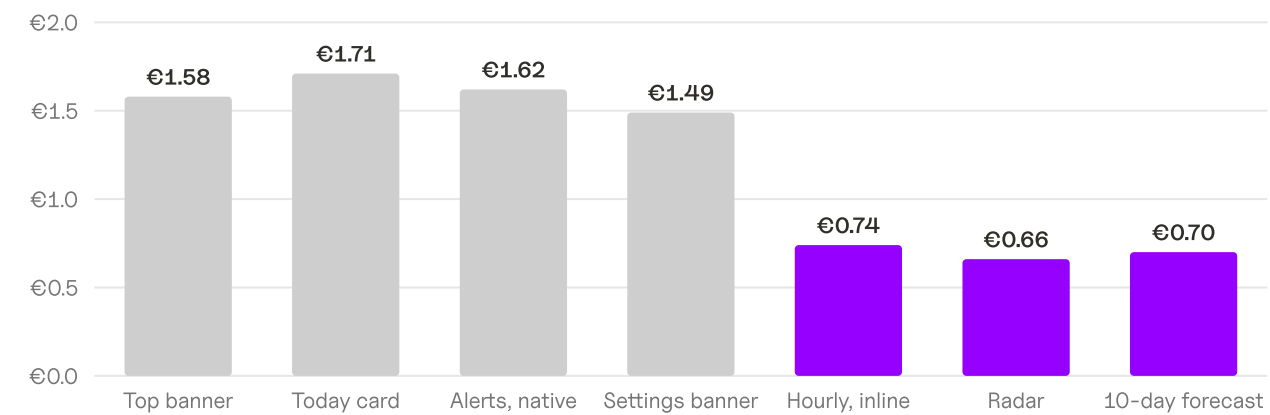
Without a consent signal, the ad network must treat the request as non-personalised. Advertisers pay 3.0× less for those. **Users had said yes. The app just didn’t pass it on in time.**

Measured by joining ad requests with consent logs by session and timestamp. A weather app is opened briefly and often, so the first seconds of a session carry a large share of all requests.

03_

Three of seven slots never compete on price.

The four original slots use bidding: several networks bid for each impression in real time. The three slots added since last year were set up with only the old waterfall, which asks networks one after another at fixed price floors.



Price per 1,000 impressions by slot · purple: no bidding

Slots	Share of impressions	Average price per 1,000
With bidding (4)	59%	€1.59
Waterfall only (3)	41%	€0.71

41% of impressions sell at less than half the price. Most of it is configuration in the mediation platform, plus two SDK adapters in the app. No new slots, no design changes.

04_

Ads refresh when nobody can see them.

Banners refresh every 15 seconds whether they are on screen or not. Ads lower on the page load when the screen opens, even though most people never scroll that far.

Loading and refresh, before	Value
Refreshed impressions that were off screen	46%
Viewability, all slots	48%
Fill rate, all slots	71%
Impressions per daily user	9.8

Advertisers and bidders measure viewability. When half of an app’s impressions are never seen, they bid less on all of them, including the ones people do see. More impressions here can mean less money.

05_

More slots would make it worse.

The 10-day forecast slot added in April is a useful test. It raised impressions per user by +18%, but revenue per user only by +3%, and median session length dropped 4%. It also arrived without bidding.

Two more slots would add impressions at the lowest price, lower viewability further, and cost user attention. The money is in selling the existing impressions properly.

06_

The plan.

Three changes, in order of value for effort. Each ships behind a remote flag in the next regular release, so it can be measured and switched off without a new build.

Change	Expected effect on revenue per user	Effort	How it will be measured
1. Consent-first ad loading Start ads after the consent decision, with a 2-second cap	+12–18%	3 days	Share of requests with a consent signal; price on affected requests
2. Bidding on all slots Mediation setup plus two SDK adapters	+8–12%	4 days	Price and fill rate on the three slots
3. Visibility-based refresh and loading Refresh only on screen; load one screen ahead	+4–8%	3 days	Viewability, price and impressions per user

EXPECTED COMBINED EFFECT

+26–43% ad revenue per daily user, measured against a 10% holdout for seven days after release. Guardrails: crash-free sessions, cold start time, session length and consent rate must not get worse.

Expected ranges come from the price gaps measured in this report, discounted for overlap between the changes. They are estimates for planning, to be replaced by measured numbers.

PART 2 · IMPLEMENTATION REPORT

00_

Executive summary

Sent on day 21

Background

The initial report found three causes behind the decline in ad revenue per user and proposed three changes instead of new ad slots. Nimbus approved the plan on day 4.

What was done

I built all three changes on iOS and Android behind remote flags, set up bidding for the three remaining slots, and shipped them in the regular release on day 13. For seven days, 10% of users kept the old behavior, and separate groups had one change switched off, so each change could be measured on its own.

Results

- **Ad revenue per daily user +32%**, €12.06 to €15.87 per 1,000 users a day. The lift held every day, between +30% and +33%.
- **About €24,000 more a month** at current traffic, with no new slots.
- **Two changes met or beat the plan.** Consent-first loading +19%, bidding +9%.
- **One delivered less.** Visibility-based refresh added +2% instead of +4–8%: prices rose, but there were fewer impressions to sell.
- **No guardrail moved.** Crashes, start time, session length and consent rate stayed the same.

Recommendations

Roll out to everyone and remove the holdout. Keep the remote flags for one more release. Add monitoring for consent signals in ad requests, and make bidding part of the checklist for every new slot.

01_

The plan, and what happened.

Change	Expected → measured	Effort	Verdict
Consent-first ad loading	+12–18% → +19%	3 days planned, 3 used	Above plan
Bidding on all slots	+8–12% → +9%	4 days planned, 3 used	On plan
Visibility-based refresh and loading	+4–8% → +2%	3 days planned, 2 used	Below plan
All three together	+26–43% → +32%	8 days	On plan

€12.1 → €15.9

ad revenue per 1,000 daily users per day

71% → 84%

fill rate, all slots

48% → 67%

viewability, all slots

Price per 1,000 impressions rose from €1.23 to €1.67, while impressions per daily user fell from 9.8 to 9.5. **Nimbus now sells fewer ads, for more money, and people see more of them.**

Each change was measured in a group where only that change was off, compared with the group where all three were on. Effects multiply rather than add, so the three individual numbers don't sum exactly to the total.

02_

Consent-first ad loading.

The ad SDK now starts after the consent decision is available. For returning users the decision is cached and arrives in milliseconds. For first-time users it waits for their answer, with a 2-second cap so a slow consent service never blocks ads completely.

```
// AppStartup.swift, before: ads start immediately
MobileAds.start()
consent.refresh()                                // async, about 0.8 s later

// after: every request carries the user's decision
Task {
    let decision = await consent.decision(timeout: .seconds(2))
    AdRequestDefaults.consent = decision
    MobileAds.start()
}
```

Requests from consented users	Before	After
Carrying a consent signal	62%	99%
Effect on revenue per user		+19%

The cost: first-time users see their first ad about 0.6 seconds later. Nobody else notices. The same change was made on Android in the application start-up code.

03_

Bidding on all slots.

Most of this change was configuration: the three waterfall-only slots joined the same bidding setup as the original four. In the app, two bidding partners needed their SDK adapters added and initialised.

```
// mediation setup, before
"hourly_inline": { "bidding": false, "waterfall": ["network_a", "network_b"] }

// after: bidders compete first, the waterfall stays as a backup
"hourly_inline": {
  "bidding": true,
  "bidders": ["bidder_1", "bidder_2", "bidder_3"],
  "waterfall": ["network_a"]
}
```

On the three slots, price per 1,000 impressions moved close to the other slots and fill rate rose, because more buyers saw every impression. Effect on revenue per user: **+9%**, inside the planned range, one day faster than planned.

The bigger lesson is process. The three slots lost money for months because nobody checked their setup when they were added. Bidding is now on the checklist for every new slot.

04.

Visibility-based refresh: less than planned.

Banners now refresh only while at least half of the ad is on screen, every 30 seconds instead of 15. Lower slots load when they come within one screen of view, instead of when the page opens.

```
// AdSlotView.swift: refresh only what people can see
refreshTimer = .every(.seconds(30)) { [weak self] in
    guard let self, self.visibleFraction >= 0.5 else { return }
    self.reload()
}

// load one screen ahead instead of on page open
slot.loadWhen(distanceToViewport: .screens(1))
```

Refreshing slots	Change
Price per 1,000 impressions	+14%
Impressions	-21%
Revenue per user, all slots	+2% planned +4–8%

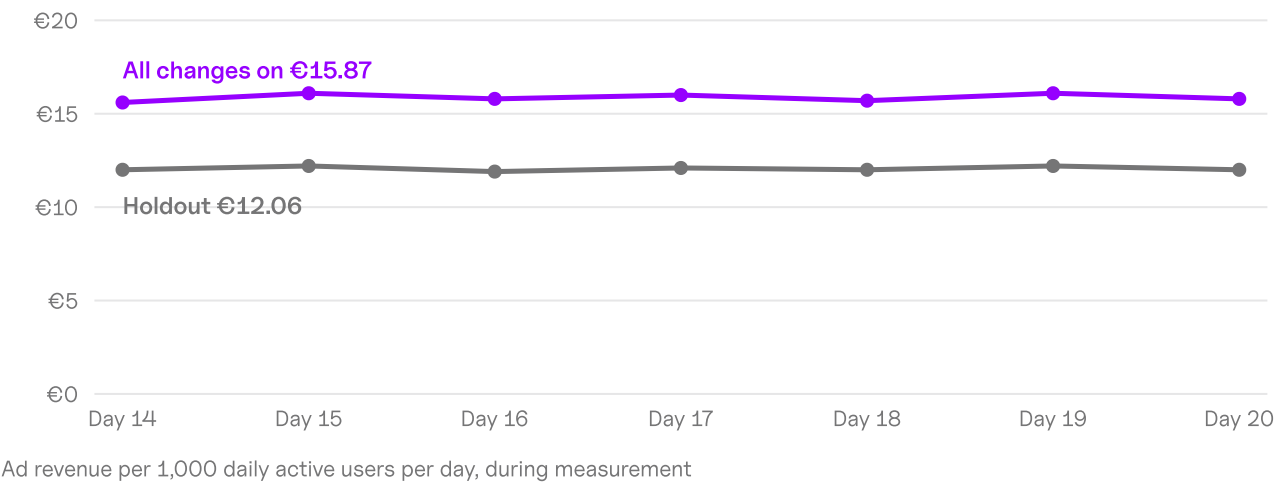
Prices rose as expected. But more of the off-screen impressions had been sold than the data in Part 1 suggested, so removing them took away more volume. The net effect is small but positive, and the app now makes about a fifth fewer ad requests, which saves users’ data and battery.

I recommend keeping it. A 20-second interval for on-screen banners is worth testing next.

05_

Shipped safely, measured honestly.

Everything went out in the regular release, behind remote flags. For seven days, 10% of users kept the old behavior. The lift was stable every day.



Guardrail	Holdout	All changes on
Crash-free sessions	99.83%	99.82%
Cold start, median	1.20 s	1.21 s
Session length, median	3.9 min	3.9 min
Consent rate	71.2%	71.0%

No guardrail moved. At current traffic, the result is about €800 more a day, €24,000 a month, on a base of about €76,000.

06_

What I would do next.

Now	Remove the holdout and roll out to everyone. Keep the flags for one more release, then remove the old code.
Monitoring	Alert when requests without a consent signal go above 3%, so an SDK update can't silently undo this.
Next test	A 20-second refresh interval for on-screen banners, measured the same way.

07_

How this engagement was done.

Data used

Ad server and mediation reports by slot, network and day, consent logs, ad request logs joined to sessions, product analytics, and crash and performance monitoring.

Definitions

Revenue per user: ad revenue per 1,000 daily active users per day. **Price:** revenue per 1,000 impressions.

Fill rate: filled requests over all requests. **Viewability:** share of impressions at least half on screen for a second.

Measurement

Random assignment by user to a holdout, a group with all changes, and one group per change with only that change off. Seven full days after rollout.

Limits

Seven days is enough for a stable lift, not for seasonal effects. Ad prices move with the market, which is why the comparison is against a holdout over the same days, not against June.

Time

Four days of investigation and the initial report. Eight days of implementation across iOS, Android and mediation setup. One week of measurement. One day to write this up.

What you get

Two reports like these, the code in your repository, reviewed and tested, and a team that knows how to keep it working.



Ads earning less than they should?

Send me your setup. I'll tell you what I would check first.

x91tech.com · Alex Severianov