Ad tech explainers · Ep 4

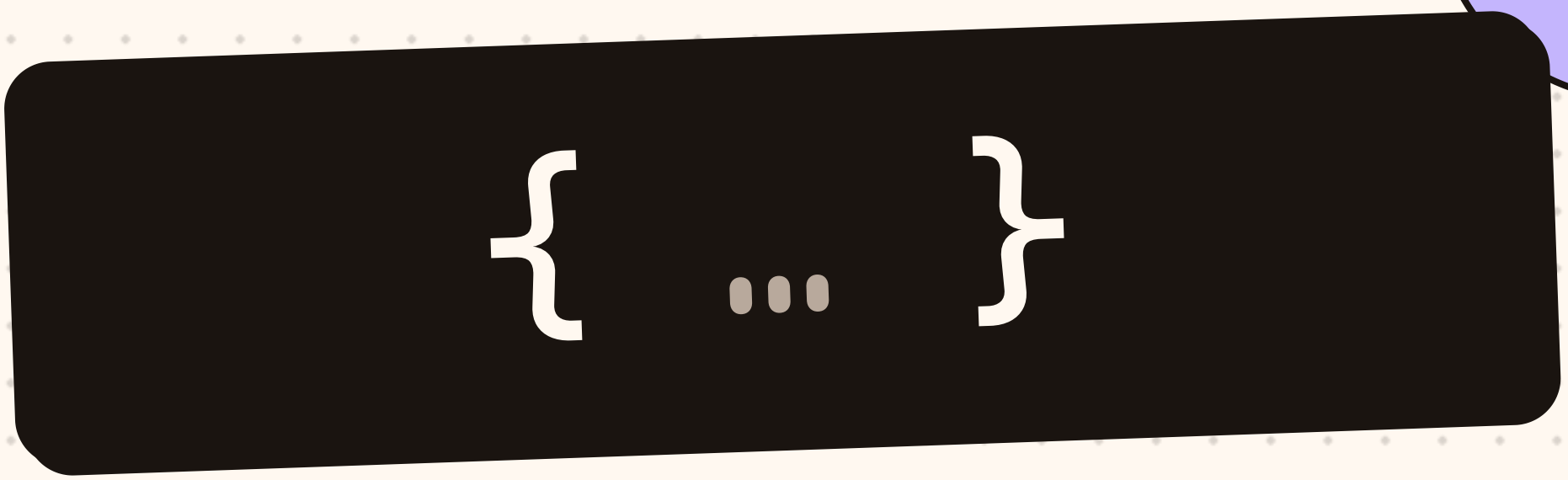
What does the buyer actually see?

Inside one OpenRTB bid request, field by field.

Swipe →

RediAds blog.rediads.com/learn

YOUR PAGE ASKS FOR AN AD



The buyer sees one message.

A bid request. Here's what's inside it, in six fields.

The term

Bid request

In OpenRTB, **one JSON message** describing one ad opportunity.

Here's one from **news.example**, field by field.

Field 1 · imp

The impression: the slot, its size and its floor.

```
"imp": {
  "w": 300, "h": 250,
  "bidfloor": 1.50
}
```

A 300×250 banner. Floor: **$1.50**.

Fields 2–4 · Context

Page, device, consent.

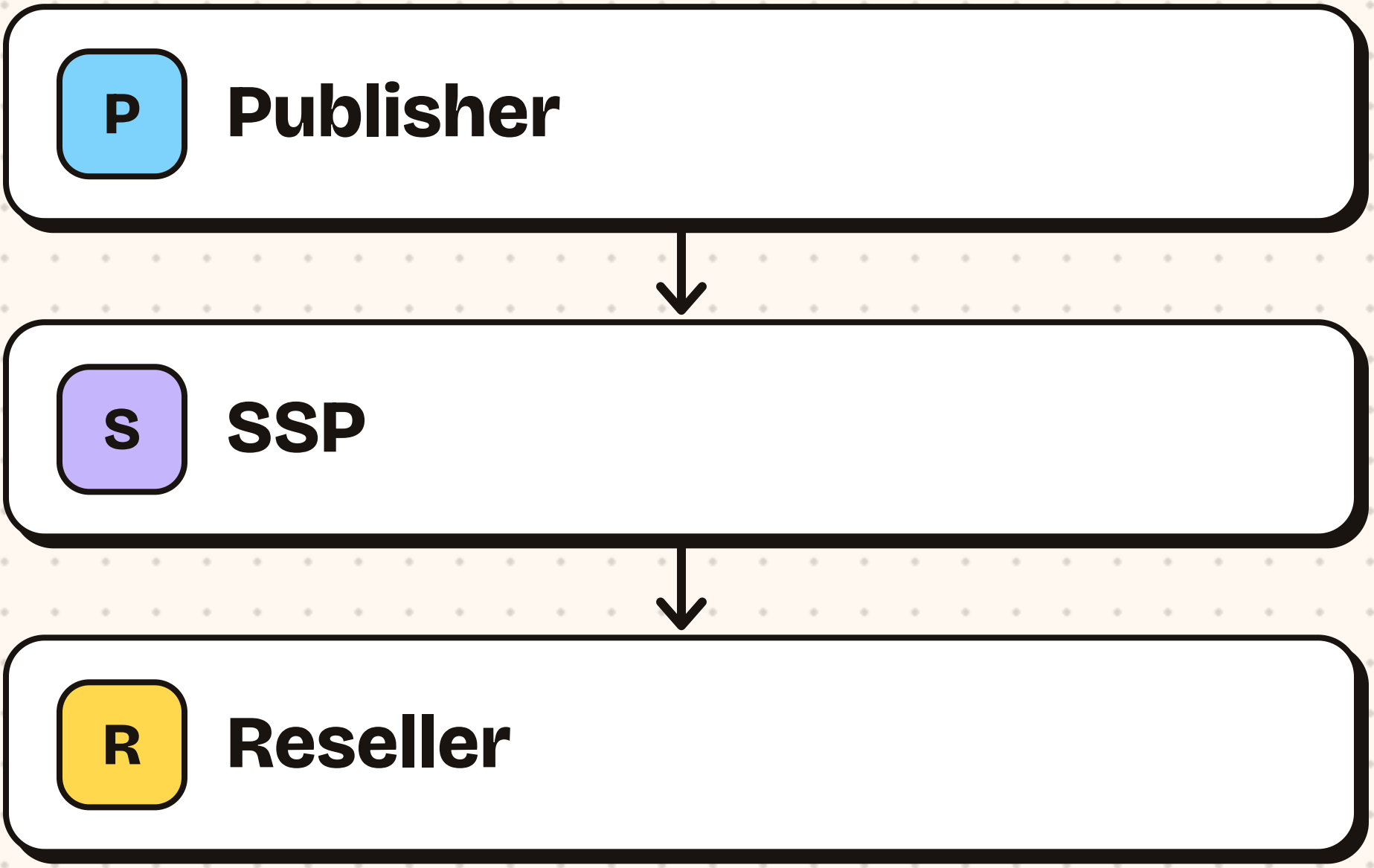
S **site**
The exact page

D **device**
The browser and country

R **user + regs**
Consent strings: what data may be used

Field 5 · source

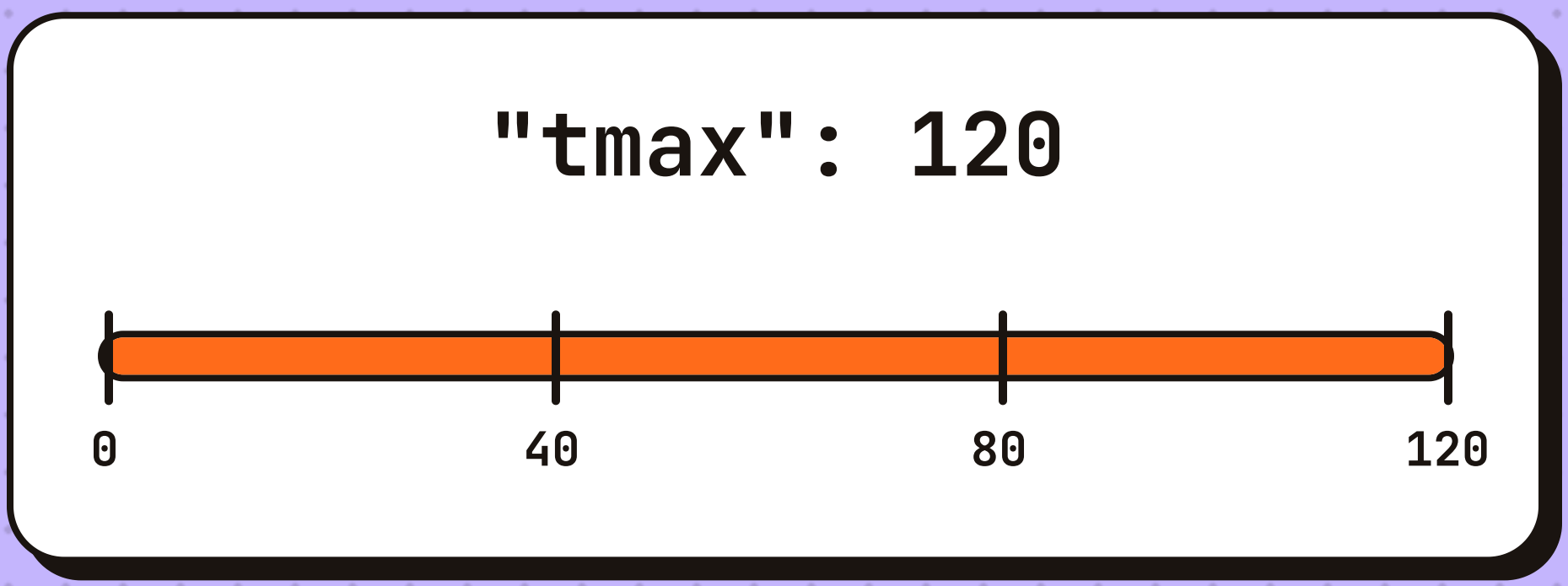
schain: every seller in the path.



CORE FIELD IN OPENRTB 2.6

Field 6 · tmax

Answer in 120 ms, or you're out.



The DSP reads it all, then decides **whether** to bid, and **how much**.

Your check today

A buyer can only price what you send.

"Which fields do you **fill**?"

- [x] Page URL
- [x] Consent string
- [x] schain

Recap

Inside one bid request:

imp

site

device

user + regs

source

tmax

NEXT · EP 5

buyers.json and DemandChain: the buyer-side mirror

Watch the 60-second video at
blog.rediads.com/learn

RediAds Follow for Ep 5

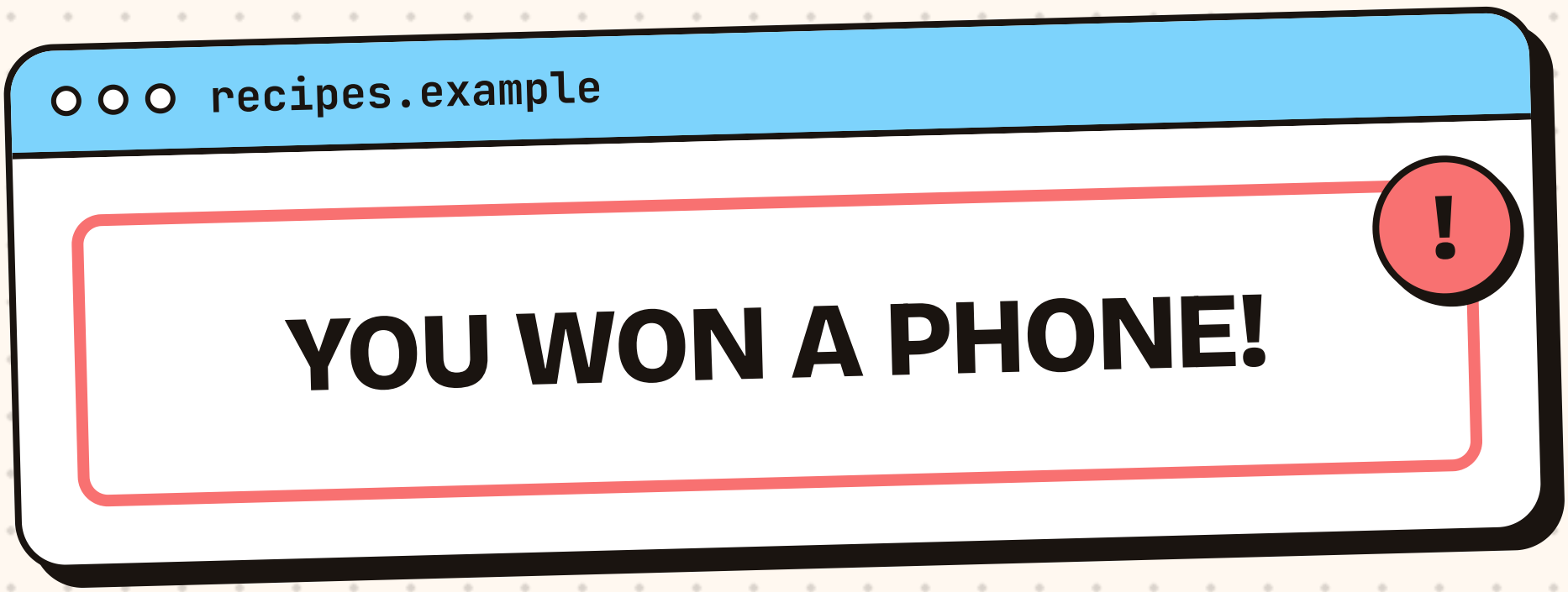
Ad tech explainers · Ep 5

Who paid for this ad?

buyers.json and DemandChain: tracing a bad ad back to the buyer behind it.

Swipe →

RECIPES.EXAMPLE · ONE AD SLOT



A scam ad lands. Who paid for it?

The term

buyers.json + DemandChain

SELL SIDE	BUY SIDE
`sellers.json`	`buyers.json`
`schain`	`dchain`

The **buyer-side mirror** of what sellers already declare.

The problem

Banned on one DSP. Back on another.

A **DSP A**
Bad advertiser banned

↓

B **DSP B**
Same advertiser, new home

Your SSP sees the DSP, **not the buyer behind it.**

The fix · Step 1

DSPs publish buyers.json

```
dsp.example/buyers.json
{ "buyer_id": "441",
  "name": "Acme Media Ltd",
  "buyer_type": "ADVERTISER" }
```

Every buyer seat, with an ID and a legal name.

The fix · Step 2

Each bid carries a dchain. Every buying platform adds a node.

A **Acme Media Ltd**
The buyer

↓

D **DSP**
Adds its domain + the buyer's seat

↓

S **Your SSP**
Traces the ad to one named buyer

`ASI + BSID` **Trace it. Block it.**

The catch

Released 2021. Uptake: slow.

Spec	v1.0 · June 2021
Who publishes	DSPs opt in
Buy-side uptake	Slow

Your check today

When a bad ad slips through:

Ask your SSP for the DSP and buyer seat.

- [x] DSP name
- [x] Buyer seat ID
- [x] Then check that DSP's buyers.json

Recap

Sellers declare. Buyers should too.

1 Mirror

2 Publish

3 Chain

4 Trace

NEXT · EP 6

ads.cert: signing the bid request

Watch the 60-second video at
blog.rediads.com/learn

Ad tech explainers · Ep 6

Signed, or just trusted?

ads.cert, the standard for signing bid requests, and its honest status.

Swipe →

Myth

ads.txt proves a bid request is genuine.

Fact

It doesn't. ads.cert is the IAB standard for signing ad traffic.

The term

ads.cert

Cryptographic signatures on ad traffic, checked by the buyer.

Did this come from that company?
Was anything altered?

The problem

ads.txt says who may sell. It can't show an edit in transit.

S **SSP**
Sends the request

↓

M **Middleman**
Edits it on the way

↓

D **DSP**
Nothing shows it

```
"format": "300x250" "VIDEO"
```

Why it stalled

Version one needed OpenRTB 3.



The industry stayed on OpenRTB 2.
Version two changed approach.

The fix · v2

Publish a key. Sign each request. Verify it.

1 **Call sign**
A public key in the company's DNS

↓

2 **SSP signs**
X-Ads-Cert-Auth header

↓

3 **DSP checks**
From that company? Unchanged?

The status

Published. Hard to find in the wild.

Tech Lab	"Ready for adoption"
Reference code	Proof of concept
Adoption	Hard to find

Your check today

Your protection today: keep all three clean.

- [x] ads.txt
- [x] sellers.json
- [x] schain

Recap

Signing works on paper. Clean files work today.

1 Gap

2 Key

3 Sign

4 Verify

NEXT · EP 7

OM SDK: how viewability gets measured in apps

Watch the 60-second video at
blog.rediads.com/learn

RediAds Follow for Ep 7

Ad tech explainers · Ep 7

Who checks your app's ads were seen?

OM SDK: how viewability gets measured in apps.

Swipe →

ONE GAME APP · FOUR VERIFICATION VENDORS

4 SDKs

ONE PER VENDOR, INSIDE ONE APP

Every vendor wanted its own code.

The term

OM SDK

The Open Measurement kit: one shared measurement layer for every vendor.

Verification firms are the companies brands hire to check ads were seen.

The problem

Before 2018, each verification firm wanted its own code in your app.

More weight

More crashes

Numbers that didn't match

The fix

IAB Tech Lab released OM instead.



STATUS · OM SDK 1.5 · IOS · ANDROID · SOME CTV

How it works · game.example

OM reports how much of the ad is on screen, and for how long.

A — **Ad + vendor script**
Each ad brings the brand's vendor script

↓

OM — **OM SDK**
Included once, in the ad SDK

↓

V — **Vendor**
Applies the viewability rule

The rule

Same rule, every vendor.

Display	`50% · 1 second`
Video	`50% · 2 seconds`

Half the pixels on screen, for that long.

Your check today

Does your ad SDK support OM?

- [x] Ad SDK supports OM
- [x] SDK version is current
- [x] Vendor scripts allowed

Without it, vendors can't verify your in-app impressions.

Recap

Open Measurement:

One kit

Every vendor

Same rule

NEXT · EP 8

SHARC: the standard meant to replace SafeFrame and MRAID

Watch the 60-second video at
blog.rediads.com/learn

Ad tech explainers · Ep 8

One ad, built twice.

SHARC, the standard meant to replace SafeFrame and MRAID, and its honest status.

Swipe →

The story

One expandable ad. Built once for apps, once for **websites**.

APP

EXPANDABLE

SUMMER SALE

WEB

EXPANDABLE

SUMMER SALE

The term

SHARC

Secure HTML Ad **Rich-media Container**.

The IAB standard meant to merge the app and web rulebooks into **one container**.

The problem

Expanding ads must talk to the page. Two rulebooks.

Apps — MRAID

Websites — SafeFrame

So creatives got **built and tested twice**.

The fix

The ad asks. The container decides.

RESIZE? ✓

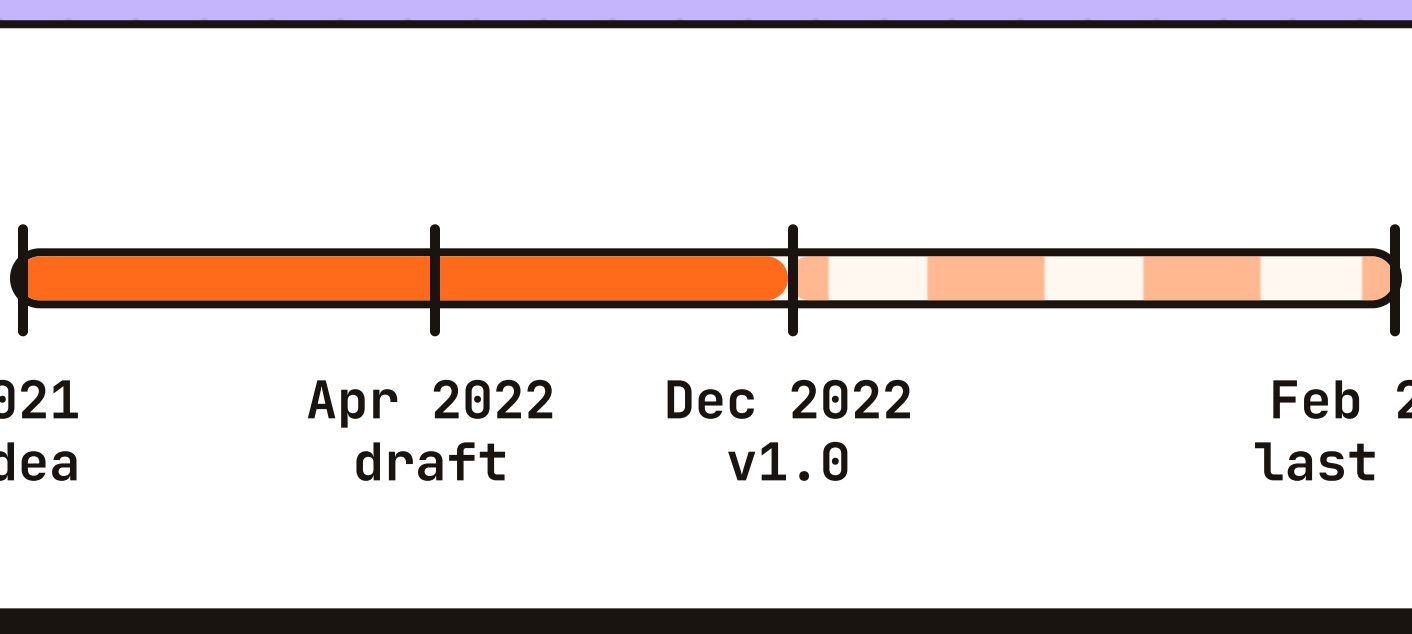
OPEN LINK? ✓

CLOSE? ✓

READ YOUR PAGE? ✗

The status

Final on paper. Hard to find in practice.



LISTED AS REPLACING BOTH

Your check today

Keep the old two working.

- [x] MRAID support in your app SDK
- [x] The SafeFrame setting on your Ad Manager creatives

Recap

One container was the plan. Two still run today.

1 Two rulebooks

2 One container

3 Ad asks

4 Still rare

NEXT · EP 9

DSA transparency in the EU bid stream

Watch the 60-second video at
blog.rediads.com/learn

RediAds Follow for Ep 9

Ad tech explainers · Ep 9

Who paid for this ad?

How the EU's DSA travels through the OpenRTB bid stream.

Swipe →

RediAds blog.rediads.com/learn

The question

In the EU, users must be told **who paid** for an ad.

?

AD

ACME SHOES

The term

DSA

The EU's Digital **Services** Act requires it.

An **OpenRTB extension** carries the answer.

The rule · Art. 26

Since Feb 2024, online platforms show users, in real time:

1. It's an ad
2. On whose behalf
3. Who paid
4. Main targeting parameters

The problem

The page doesn't know the advertiser until the ad arrives. So data goes **both ways**.

1 **Request asks**
dsarequired: 2 means no DSA info, no bid

↓

2 **Bid answers**
behalf, paid, and parameters like profiling

The fix

Flags on each side say who shows the notice.

BID REQUEST

```
"regs": { "ext": { "dsa": {
"dsarequired": 2, "pubrender": 1 } } }
```

BID RESPONSE

```
"dsa": { "behalf": "Acme Shoes",
"paid": "Acme Shoes", "adrender": 0 }
```

Illustrative values. Acme Shoes is fictional.

Your check today

Lawyer first, then your SSP.

- [x] Are we an **online platform**? (legal question)
- [x] Which `dsarequired` value is sent?
- [x] Who renders the notice?

Recap

Who paid? The bid stream now says.

1 Art. 26

2 Request

3 Response

4 Notice

NEXT · EP 10

The IAB spiders and bots list

Watch the 60-second video at
blog.rediads.com/learn

Ad tech explainers · Ep 10

A robot, billed as a reader.

The IAB/ABC Spiders & Bots List: what it catches, and what it can't.

Swipe →

RediAds blog.rediads.com/learn

Mistake

Counting a search crawler as a reader, and billing an advertiser.

VISITORS 1,000 →

VISITORS 1,001

The term

Spiders & Bots List

The IAB and ABC's **shared list** of known robots.

IAB TECH LAB · UPDATED MONTHLY · PAID LICENCE

The problem

Every auditor kept its own bot list, so counts disagreed.

Auditor A	9,400
Auditor B	10,200

Illustrative numbers.

The fix

One list, two files.

1

Valid browsers

Real user agents, the name tag every visitor sends

2

Known robots

Spiders, crawlers and other bots

Checked twice

Match a real browser. Avoid the robot list. Fail either: filtered.

`Chrome`	Counted
`Googlebot`	Robot list
`curl`	Not a browser

`FILTERED = GENERAL INVALID TRAFFIC (GIVT)`

The catch

A bot that lies about its name gets through.

Sophisticated invalid traffic (SIVT) is a different fight.

Your check today

Compare your logs.

Server logs	12,000
Analytics	10,000

Google Analytics already drops bots on this list. A big gap? **Robots are a likely reason.**

Recap

Known bots: filtered. Liars: next fight.

1 One list

2 Two files

3 Two checks

4 GIVT

NEXT · EP 11

Agentic ad protocols: ARTF and AAMP

Watch the 60-second video at **blog.rediads.com/learn**

RediAds Follow for Ep 11

Ad tech explainers · Ep 11

Chatbots haggling over every ad?

Agentic ad protocols: ARTF and AAMP, explained.

Swipe →

RediAds blog.rediads.com/learn

THE MYTH





The reality is plumbing.

The first agentic standard runs partner code inside the auction.

The term

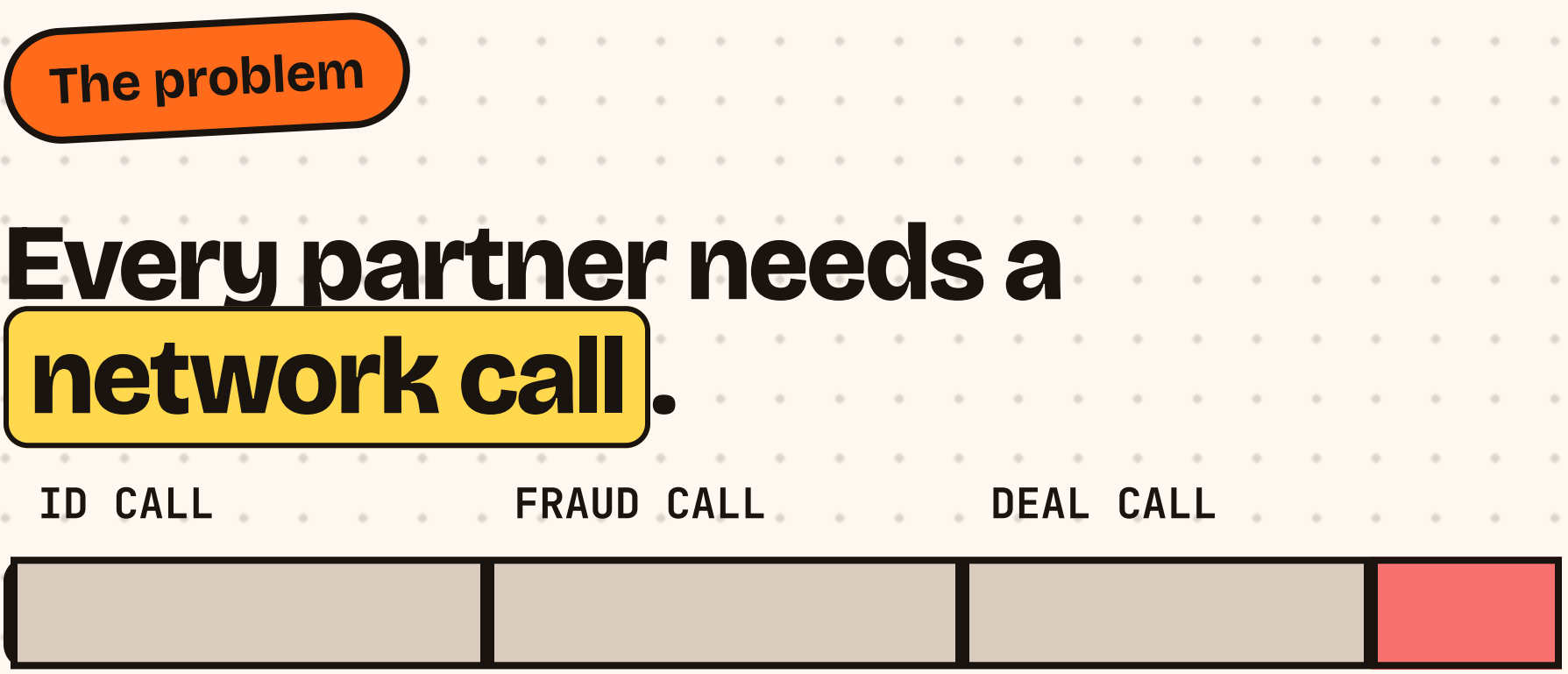
ARTF

Agentic Real Time Framework: partner code that runs **inside the auction**.

From IAB Tech Lab.

The problem

Every partner needs a network call.



Fraud filters, deal curators: each call eats into an auction of maybe **100 ms**.

The fix

Partners run as containers on the SSP's servers. They propose **mutations**.



The **SSP decides** which to apply.

The status

Final. Adoption: early.

Public comment
Nov 2025

v1.0
Final

ARTF V1.0 · FINAL · ADOPTION EARLY

One umbrella · Feb 2026

AAMP

Agentic Advertising Management Protocols: all of Tech Lab's agent work.

FOUNDATIONS

ARTF

PROTOCOLS

Agent toolkits

TRUST

Agent registry

Your check today

The question is control.

Ask your SSP:

- [x] Which partner containers run?
- [x] What can they change?
- [x] Who approves?

Recap

Agents propose. Platforms decide. You set the rules.

NEXT · EP 12

CoMP: getting paid when AI crawls your content

Watch the 60-second video at
blog.rediads.com/learn

Ad tech explainers · Ep 12

AI read it. You got $0.

CoMP: getting paid when AI crawls your content.

Swipe →

RediAds blog.rediads.com/learn

THREE AI CRAWLERS READ YOUR ARTICLE

AI BOT 1 AI BOT 2 AI BOT 3

REVENUE $0.00

Zero clicks.
Zero dollars.

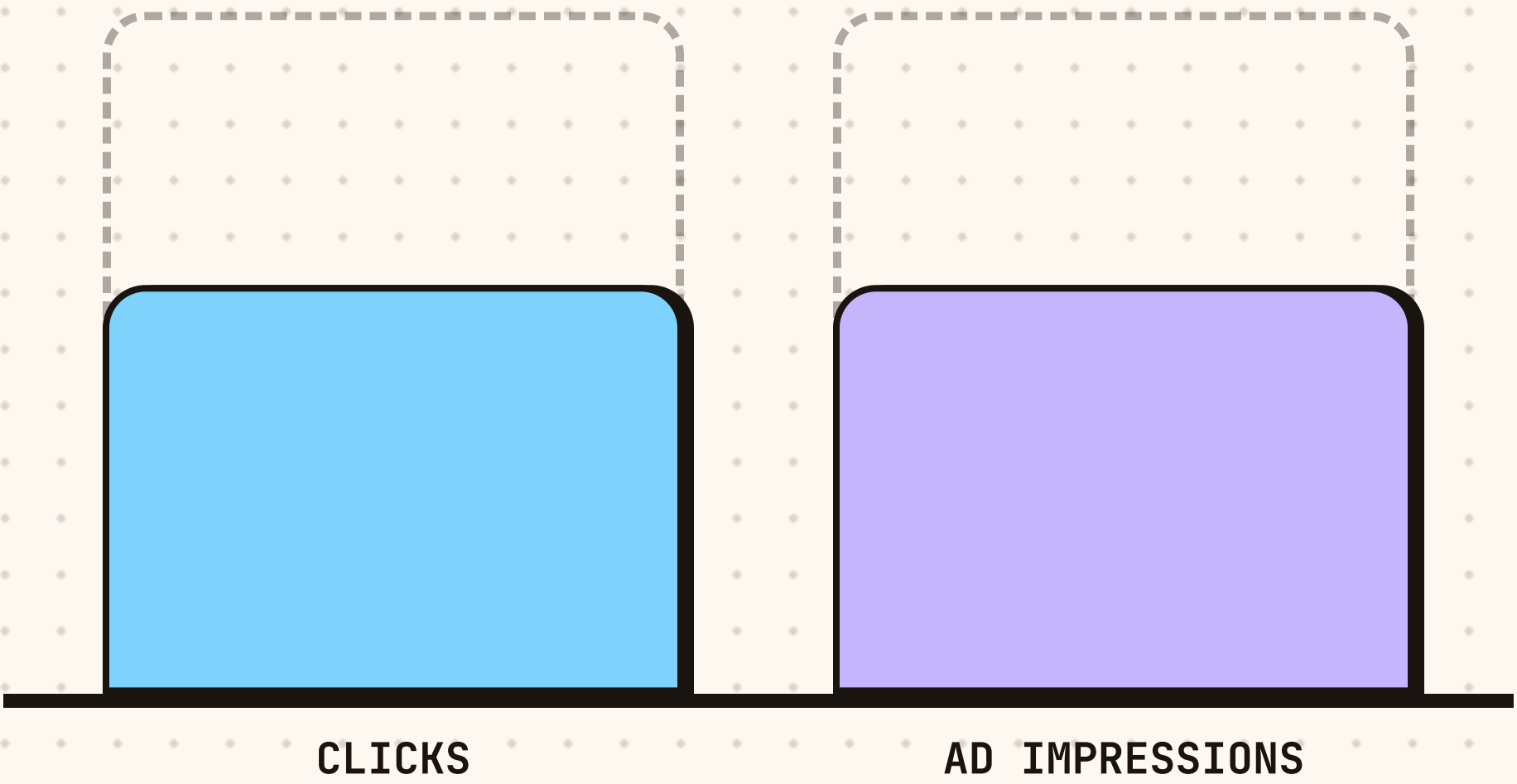
The term

CoMP

Content Monetization Protocols: Tech Lab's standard for AI systems to **license content first**.

The problem

AI answers summarize your reporting. **Fewer clicks**, fewer ads.



You could block crawlers, but not say: **here are my terms**.

Step 1 · Block at the edge

AI

AI crawler

No licence yet

C

CDN · edge

The network that delivers your site

Unlicensed bots are **refused** at your CDN.

Step 2 · Ask, license, collect

A bot with a deal states its purpose and gets a token.

No deal

Refused, pointed to your licensing page

Deal

```
access request
purpose: answer query
```

↓

ACCESS TOKEN

↓

```
machine-readable
content
```

The status

Pricing is yours.

Per crawl

Per use

By outcome

CoMP v1 · FINAL APRIL 2026

So far, we found **no major AI company** publicly using it.

Your check today

Know your crawlers.

Step one needs no standard.

- [x] List AI crawlers in your logs
- [x] Decide: allow or block
- [x] Set the rules at your CDN

Recap

Block first. License second. Get paid third.

NEXT SERIES · SEGMENT PLAYBOOKS

News publishers: AI search, fewer clicks, and yield

Watch the 60-second video at
blog.rediads.com/learn