A FairShare Field Manual

The Indie Studio's Data Privacy Blueprint

A technical audit and implementation manual for founders, builders and small teams.

Read This First

If you run a small studio, “data privacy” probably sits in the same mental folder as “filing quarterly taxes” and “updating the EULA”. Important, vaguely scary, easy to push to the next sprint. Then you read about a studio getting hit with a GDPR complaint over an analytics SDK they forgot they installed, and the folder stops feeling abstract and starts feeling more urgent.

This is the manual built for that moment. It’s a real audit you can run on your own game and backend in an afternoon. No legalese. No “consult your data protection officer” (you probably don’t have one). Just the specific things that matter most to regulators, players and platform holders, and how to fix them with the kind of resources a small team has: a few hours, a code editor, and maybe an AI assistant.

Here’s the uncomfortable truth most compliance content dances around: **the law doesn’t care about the size of your business.** A 4-person studio collecting player emails and behavioral data has nearly the same GDPR and CCPA obligations as a 400-person publisher. The fines may scale with revenue, but the complaints, takedown requests, and platform rejections happen regardless of how big you are.

The good news: the actual work is more manageable than it looks. Most of what trips up indie studios comes down to a dozen concrete things, and we’ll help you close those gaps.

What You’ll Have by the End:

- a completed audit of your data collection
- a prioritized fix list, sorted by risk
- an AI prompt you can paste into Claude or ChatGPT to pressure-test your own setup
- a copy/paste to-do list

A Note on Scope

This is practical guidance written by a team that builds privacy infrastructure for a living. It is not legal advice, and we’re not your lawyers. We always recommend a legal review of your data policies and procedures.

What's Inside

1. The 5-Minute Reality Check

Before you dive into the details, let's get a general read on your data privacy risk. Answer these questions honestly. Every *"I don't know"* is a *"yes"* for risk-assessment purposes.

- [ ] **Do you collect anything that identifies a player (e.g. Email, username, IP, device ID, Steam/Epic/console ID, wallet address, etc.)?** If you have player accounts, leaderboards, or analytics, the answer is yes.
- [ ] **Do you have players in the EU, UK, or California?** If your game is on Steam, a mobile store, or the open web, you do. Jurisdiction follows the player, not your studio.
- [ ] **Could a child realistically play your game?** "Not designed for kids" is not the same as "no kids play it." If a 12-year-old can install it, COPPA and the UK Children's Code are in scope.
- [ ] **Do you use third-party SDKs (e.g. Unity Analytics, GameAnalytics, Firebase, AppsFlyer, an ad network, a crash reporter, etc.)?** Each one is a data processor you're responsible for.
- [ ] **If a player emailed you today saying "delete all my data," could you actually do it?** Within 30 days? Across every system, including your analytics provider and your backups?
- [ ] **Is there a written privacy policy that matches what your game actually does?** Not a template you pasted in 2022, but the current behaviour.

Scoring

3 or more "no"/"I don't know" answers means that you have real, addressable compliance gaps. Don't panic! These will become your to-do list in **Section 7**.

Why This Catches Studios Off Guard

It's rarely a dramatic breach. It's mundane: a player in Germany files a data access request that sits ignored because nobody at the studio realized that it starts a legal clock. An app store review flags an SDK collecting device IDs without disclosure. A parent discovers their kid's data in a game with no age gate.

These examples of non-compliance are spotted by players who know their rights, and there are more and more of those every year. The fix is boring and effective: know what you collect, ask permission properly, and be able to honor the requests. Keep reading for the details.

2. Know Your Data: The Inventory Most Studios Skip

Every privacy framework should start in the same place: a data map. It's the least glamorous step and the one almost everyone skips, which is exactly why it's where audits fall apart.

Write down every piece of player data your game touches, where it lives, why you have it, and how long you keep it. Regulators call this a *"record of processing activities"*, otherwise known as *"a spreadsheet"*.

The Data Map Template

Make a sheet with these columns and fill one row per type of data:

Column	What Goes Here	Example
Data type	The actual field	Email address
Why you have it	The real reason, in plain words	Account login, patch notes, emails
Legal basis	Consent, contract, or legitimate interest	Consent (for marketing), contract (for login)
Where it lives	Every system it sits in	Postgres, Mailchimp, daily backups
Who can see it	People and third parties	You, your co-founder, Mailchimp
Retention	How long you keep it	Until account deletion +30 days
Sensitive information	For example: Kids' data, location, health, biometrics	No

Founder Reality Check

Most indie data maps fit on one page. If yours is sprawling, that's a finding in itself: **you're probably collecting data you don't use.**

The cleanest privacy fix is deleting fields you can't justify. Less data is less risk, less storage, and less to explain.

2. Know Your Data: The Inventory Most Studios Skip

Where Data Might Be Hiding

It's not always obvious where data lives. Here's a list of places to check so that your map is accurate:

- [ ] **Your database.** Dump the schema. Every column on a user, player or account table is worth checking. Don't forget JSON blobs and "metadata" fields which often accumulate extra data over time.
- [ ] **Your analytics.** Open the dashboard of every analytics tool and look at what events you fire. Custom events often smuggle in identifiers you forgot you sent.
- [ ] **Your SDKs.** Search your codebase for every third-party SDK you've integrated. Each one's docs will tell you what it collects by default (often: device ID, IP, OS, sometimes location).
- [ ] **Your logs.** Server logs love to capture IP addresses and request bodies. That's personal data sitting in plaintext, usually with no retention limit.
- [ ] **Your support inbox.** Players send you emails, usernames, and sometimes even screenshots with personal info. This data all counts.
- [ ] **Your backups.** If you back up the database, you're storing copies of everything above. This matters a lot when it comes to deletion (see **Section 5**).

How FairShare Handles Your Data Map

Because FairShare's transactions are facilitated by blockchain, the platform automatically generates a record for every data point flowing through it. This means your data map updates itself instead of going stale.

3. Consent That Holds Up

Consent is where most studios are breaking the rules without realizing it. *"Consent"* has a specific legal meaning, and a pre-ticked box or a buried line in your ToS won't cut it.

What Real Consent Looks Like

Under GDPR (and CCPA's opt-out cousin), valid consent is:

Requirement	What it Means For Your Game
Freely given	The player can say no and still use the core game. You can't gate the whole game behind marketing consent.
Specific	There needs to be separate asks for separate things. That means account data is one thing, sharing data with an ad network is another. A single checkbox for everything is not compliant.
Informed	They know what they're agreeing to before they agree, in language a human understands.
Unambiguous	An affirmative action is required, whether it's a ticked box or a tapped button. Pre-ticked boxes and "by continuing you agree" are not sufficient for consent.
Withdrawlable	As easy to revoke as it was to give. If opting in took one tap, opting out can't require emailing support.

What Real Consent Looks Like

Regulators have gotten specific about manipulative consent design. These will draw complaints and, increasingly, fines:

- ✗ Pre-ticked boxes for anything beyond strictly necessary processing.
- ✗ *"Accept all"* in a big bright button and "manage preferences" in tiny grey text. Equal prominence is the rule now.
- ✗ Consent walls where the only path forward is *"I agree."*
- ✗ Bundling 6 different data uses into one toggle.
- ✗ Making withdrawal a scavenger hunt through 4 settings menus.
- ✗ Re-asking for consent every session until the player caves out of annoyance.

3. Consent That Holds Up

What to Implement Now (If You Haven't Already)

- [ ] A consent screen at first run that separates: necessary (no consent needed, just disclosure), analytics, and marketing/data sharing.
- [ ] Default everything optional to OFF.
- [ ] Store a consent record: what they agreed to, the exact wording shown, timestamp, and version. You need to prove consent if challenged.
- [ ] A settings screen where consent can be toggled back off, and toggling off actually stops the collection.
- [ ] Re-consent only when you materially change what you collect, not on a timer.

How FairShare Handles Consent

The social sign-on widget handles the consent flow, the granular toggles, and the timestamped consent record out of the box, written to an immutable log. You embed it instead of building and maintaining your own consent UI and trying to keep it legal as the rules shift.

4. The Technical Audit: 7 Systems to Review

Now for the hands-on part! The following list will walk you through an audit of seven systems and the specific items to examine in each. The systems are grouped by approximate risk level, so you can prioritize what to tackle first.

How to Use This List Effectively

Don't try to fix everything at once! Once you've completed the list and documented every gap, start by fixing the high-risk items this week. Aim to fix medium-risk within the month. See Section 7 for the consolidated list.

High Risk

4.1 Third-Party SDKs

SDKs are the single most common source of indie compliance gaps, because they collect data the moment they initialize, often before the player has consented to anything.

- [ ] List every SDK in your build. For each, find its data collection disclosure
- [ ] Does it collect on initialization or wait for a consent signal? Many SDKs fire immediately and it's a problem if they run before your consent screen.
- [ ] Confirm you can pass a consent state to it (most modern SDKs support a "do not collect until I say so" mode). Wire it to your consent toggles.
- [ ] Delete SDKs you don't actively use. That dead ad-network SDK from a monetization experiment is pure liability.

High Risk

4.2 Age Gating

- [ ] If kids can plausibly play your game, ensure that there's an age check at sign-up with a neutral date-of-birth entry, not "are you 13 or older? yes / no").
- [ ] Per COPPA/UK Children's Code, make sure that under-age users get a restricted experience or parental consent flow.
- [ ] Verify that you don't serve behavioral ads or collect more than necessary from known minors.

4. The Technical Audit: 7 Systems to Review

High Risk

4.3 Deletion & the Right to Be Forgotten

Deletion requests can trip you up. A player can demand full deletion, and you have to comply across every system, including backups and third parties.

- [ ] Verify that You can locate every piece of one player's data given just their identifier.
- [ ] Enact Deletion cascades so that Deleting the user row also clears their analytics events, their entries in related tables, and their data in third-party tools.
- [ ] Have you thought about backups? You don't have to restore a backup just to delete one record, but you do need a documented approach (e.g. deletion is re-applied if a backup is ever restored).
- [ ] Make sure that Third-party tools get the deletion too. Mailchimp, your analytics provider, your CRM: each has a deletion API or process.

High Risk

4.4 Data at Rest

- [ ] Is your database encrypted at rest?. Most managed providers (RDS, Cloud SQL, Supabase) offer this with a checkbox so confirm that it's on.
- [ ] Ensure that all passwords are hashed with a modern algorithm (bcrypt, argon2), never stored plaintext or reversibly encrypted.
- [ ] Confirm that Sensitive fields beyond passwords (tokens, anything you flagged "sensitive" in your map) are encrypted or tokenized.
- [ ] Verify that access to the production database is limited and logged. Not everyone on the team needs prod credentials.

4. The Technical Audit: 7 Systems to Review

Medium Risk

4.5 Data in Transit

- [ ] Ensure that every connection that carries player data uses HTTPS/TLS. No exceptions, including analytics calls and your own API.
- [ ] Confirm that there is no personal data in URL query strings (it ends up in server logs and browser history). Instead, Use request bodies.
- [ ] Verify that the certificate is valid and not self-signed in production.

Medium Risk

4.6 Data Retention

"We keep everything forever" is a finding. Every data type needs a defined lifespan, and you need a mechanism that enforces it.

- [ ] Make sure that each row in your data map has a retention period. Inactive accounts, old logs, and stale analytics events all need an expiry.
- [ ] Check that there's an actual job (cron, scheduled function) that deletes expired data. A policy without execution is only good intentions and not a defense against a non-compliance claim.
- [ ] Set a routine for server logs to rotate and purge. Default log retention is often "until the disk fills up" and that doesn't cut it.

Medium Risk

4.7 Third-Party Data Sharing

- [ ] Check that every third party that receives player data has a data processing agreement (DPA) with you. Most SaaS tools offer a standard agreement.
- [ ] Confirm that your privacy policy, at a minimum, discloses data recipients by category.
- [ ] Verify that any transfer of EU data outside the EU has a legal mechanism. The provider's standard contractual clauses usually cover this).

5. Player Data Rights Requests: The Clock Has Already Started

The moment a player in a covered jurisdiction emails asking about their data, a legal deadline starts. Most studios don't know this, which is how a polite email turns into a complaint to a regulator.

The Requests You Have to Handle

Request	What They Can Ask For	Your Deadline
Access	A copy of all data you hold on them	30 days (GDPR), 45 days (CCPA)
Deletion	Erasure of their data	30 days (GDPR), 45 days (CCPA)
Portability	Their data in a machine-readable format (e.g. CSV, JSON)	30 days
Correction	Fixing inaccurate data	30 days
Opt-out	Stop selling/sharing their data	Promptly, 15 business days (CCPA)

Here's What You Need in Place

- [ ] A way for players to make data requests that's findable in your privacy policy and in-game settings (an email address is needed at minimum, but a form is better).
- [ ] A documented process: who handles requests, how you verify the requester's identity (so you don't hand player A's data to player B), and how you fulfill it.
- [ ] The technical ability to export one player's complete data (ties directly to your deletion capability in Section 4.5).
- [ ] A log documenting requests received and how you responded, with dates. If a regulator ever asks, this is your proof you took it seriously.

How FairShare Handles Player Data Rights Requests

FairShare has automated workflows for access, deletion and opt-out requests. Plus, each player gets a self-serve privacy centre, so most requests never reach your inbox.

6. The Self-Audit AI Prompt

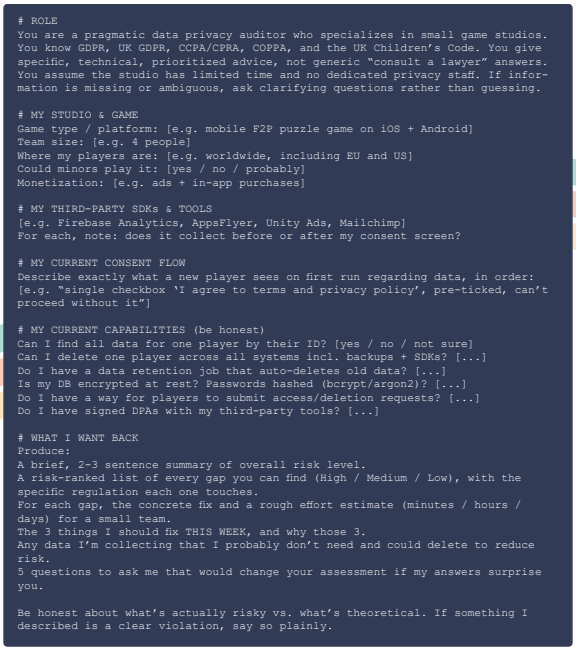
We've put together a prompt for you to fill in and paste into Claude, ChatGPT, or your AI of choice. It walks the AI through the same checks in this guide and gives you a prioritized report back. **Be honest in the inputs!** The AI can only catch what you tell it.

Important

Don't paste real secrets (API keys, passwords, actual player records). Describe your systems, don't dump credentials.

Pro Move

Run this prompt again every time you ship a feature that touches player data, or add a new SDK. Privacy debt accumulates the same way code debt does, quietly, between releases. **A 10-minute re-audit per release keeps it from piling up.**



```
# ROLE
You are a pragmatic data privacy auditor who specializes in small game studios.
You know GDPR, UK GDPR, CCPA/CPRA, COPPA, and the UK Children's Code. You give
specific, technical, prioritized advice, not generic "consult a lawyer" answers.
You assume the studio has limited time and no dedicated privacy staff. If infor-
mation is missing or ambiguous, ask clarifying questions rather than guessing.

# MY STUDIO & GAME
Game type / platform: [e.g. mobile F2P puzzle game on iOS + Android]
Team size: [e.g. 4 people]
Where my players are: [e.g. worldwide, including EU and US]
Could minors play it: [yes / no / probably]
Monetization: [e.g. ads + in-app purchases]

# MY THIRD-PARTY SDKs & TOOLS
[e.g. Firebase Analytics, AppsFlyer, Unity Ads, Mailchimp]
For each, note: does it collect before or after my consent screen?

# MY CURRENT CONSENT FLOW
Describe exactly what a new player sees on first run regarding data, in order:
[e.g. "single checkbox 'I agree to terms and privacy policy', pre-ticked, can't
proceed without it"]

# MY CURRENT CAPABILITIES (be honest)
Can I find all data for one player by their ID? [yes / no / not sure]
Can I delete one player across all systems incl. backups + SDKs? [...]
Do I have a data retention job that auto-deletes old data? [...]
Is my DB encrypted at rest? Passwords hashed (bcrypt/argon2)? [...]
Do I have a way for players to submit access/deletion requests? [...]
Do I have signed DPAs with my third-party tools? [...]

# WHAT I WANT BACK
Produce:
A brief, 2-3 sentence summary of overall risk level.
A risk-ranked list of every gap you can find (High / Medium / Low), with the
specific regulation each one touches.
For each gap, the concrete fix and a rough effort estimate (minutes / hours /
days) for a small team.
The 3 things I should fix THIS WEEK, and why those 3.
Any data I'm collecting that I probably don't need and could delete to reduce
risk.
5 questions to ask me that would change your assessment if my answers surprise
you.

Be honest about what's actually risky vs. what's theoretical. If something I
described is a clear violation, say so plainly.
```

Find the prompt on the next page →

6. The Self-Audit AI Prompt

```
# ROLE
You are a pragmatic data privacy auditor who specializes in small game studios.
You know GDPR, UK GDPR, CCPA/CPRA, COPPA, and the UK Children's Code. You give
specific, technical, prioritized advice, not generic "consult a lawyer" answers.
You assume the studio has limited time and no dedicated privacy staff. If informa-
tion is missing or ambiguous, ask clarifying questions rather than guessing.

# MY STUDIO & GAME
  • Game type / platform: [e.g. mobile F2P puzzle game on iOS + Android]
  • Team size: [e.g. 4 people]
  • Where my players are: [e.g. worldwide, including EU and US]
  • Could minors play it: [yes / no / probably]
  • Monetization: [e.g. ads + in-app purchases]

# MY THIRD-PARTY SDKs & TOOLS
  • [e.g. Firebase Analytics, AppsFlyer, Unity Ads, Mailchimp]
  • For each, note: does it collect before or after my consent screen?

# MY CURRENT CONSENT FLOW
  • Describe exactly what a new player sees on first run regarding data, in or-
    der:
  • [e.g. "single checkbox 'I agree to terms and privacy policy', pre-ticked,
    can't proceed without it"]

# MY CURRENT CAPABILITIES (be honest)
  • Can I find all data for one player by their ID? [yes / no / not sure]
  • Can I delete one player across all systems incl. backups + SDKs? [...]
  • Do I have a data retention job that auto-deletes old data? [...]
  • Is my DB encrypted at rest? Passwords hashed (bcrypt/argon2)? [...]
  • Do I have a way for players to submit access/deletion requests? [...]
  • Do I have signed DPAs with my third-party tools? [...]

# WHAT I WANT BACK
Produce:
  • A brief, 2-3 sentence summary of overall risk level.
  • A risk-ranked list of every gap you can find (High / Medium / Low), with the
    specific regulation each one touches.
  • For each gap, the concrete fix and a rough effort estimate (minutes / hours /
    days) for a small team.
  • The 3 things I should fix THIS WEEK, and why those 3.
  • Any data I'm collecting that I probably don't need and could delete to re-
    duce risk.
  • 5 questions to ask me that would change your assessment if my answers sur-
    prise you.

Be honest about what's actually risky vs. what's theoretical. If something I de-
scribed is a clear violation, say so plainly.
```

Disclaimer

An AI auditor is a fast, cheap, first pass and not a legal sign-off. It's great at catching the obvious gaps and structuring your thinking. For anything high-stakes, use its output to brief a real privacy professional, which makes their (expensive) time far more efficient.

7. Your Privacy To-Do List

In this section you will find the amalgamated list of every fix from this guide, in order of priority. Print it or paste it into your task tracker and run through each part in sequence starting with the high-priority section at the top.

High Priority

Do This Week

- [ ] Build your data map (Section 2) with one row per data type. This unblocks everything else.
- [ ] List every third-party SDK,check what each collects, and whether it fires before consent (4.1).
- [ ] Delete any SDK, field, or dataset you don't actively use.
- [ ] Confirm your database is encrypted at rest and passwords are properly hashed (4.4).
- [ ] Test deletion: pick one real account and confirm you can find and remove all its data (4.3).
- [ ] Add an age gate if minors can plausibly play (4.2)
- [ ] Set up an easily findable way for players to make data requests, and a process to handle them inside 30 days (Section 5).

Medium Priority

Do This Month

- [ ] Rebuild your consent flow to be granular, opt-in, and withdrawable (Section 3).
- [ ] Store timestamped consent records that include the wording shown to the player.
- [ ] Confirm all player data moves over HTTPS, with nothing personal in URLs (4.5).
- [ ] Set retention periods for every data type and build the job that enforces them (4.6).
- [ ] Get signed DPAs with every third party and disclose recipients in your privacy policy (4.7).

7. Your Privacy To-Do List

Ongoing

Keep Doing

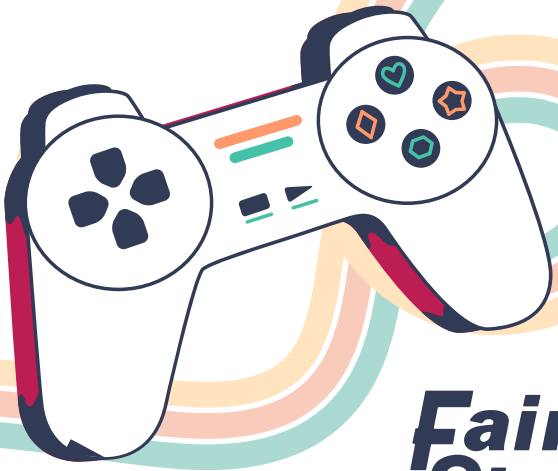
- [ ] Re-run the AI self-audit (Section 6) every release that touches player data.
- [ ] Keep the data map current as you add features.
- [ ] Log every player data rights request and your response, with dates.
- [ ] Review SDKs quarterly; they change what they collect in updates.

The Honest Truth

For a small team mid-development, this list is a few weeks of work spread across people who'd rather be building the game.

That's part of why we built FairShare. Read on to learn more!

Stop playing games, start building trust.



FairShare

8. The Shortcut: Doing All of This Without Doing All of This

Look back at that to-do list. Consent flows. Encrypted storage. Retention jobs. Deletion that cascades across systems. Player rights workflows. DPAs. Age gating. Audit logs.

Every one of those is a thing you can build and maintain yourself. It's also every one of them, forever, as the regulations shift and your game grows. For a small studio, that's a meaningful chunk of engineering time spent on plumbing instead of your game.

This is the gap FairShare was built to close. It's a player data platform that ships the entire privacy layer as infrastructure you embed, not code you write.

The "To-Do" Item	FairShare
Consent flow, granular and withdrawable	Built into the social sign-on widget you embed
Timestamped consent records	Written to an immutable, auditable log automatically
Data map/record of processing	Generated from real transactions, stays current
Retention and auto-deletion	Automated, aligned to GDPR/CCPA out of the box
Player rights requests	Self-serve privacy centre & automated workflows
Encryption, key management, PII security	Pseudonymization and encryption in transit and at rest
Age gating	Built in at sign-up (COPPA, UK Children's Code)
Compliance coverage	140+ countries handled by the platform

8. The Shortcut: Doing All of This Without Doing All of This

FairShare runs on the infrastructure behind Appreciation Engine, which has powered data for partners like Sony Music and Universal Music Group for 10+ years and 60M+ daily fan interactions. So the *"is this battle-tested?"* question is already answered.

FairShare also does something that no one else does: **your players get rewarded for the data they choose to share, turning what is normally an extractive process into an ethical, equitable one.** Incentivizing players to share their data not only unlocks deeper insights, but builds trust and loyalty with your fanbase.

Most studios can go live in under half a day by swapping their existing sign-in for FairShare's. With our pay-as-you-go model, you only pay for data that you store and access. That means you can integrate FairShare for free, and only add funds as you need them with no subscription lock-in.

Have a larger audience or anticipate growth? FairShare's pricing scales with you through volume-based discounts. Check out our **pricing calculator** on the FairShare site.

Become a FairShare Pioneer Partner

Right now FairShare is opening its **Games Pioneer Program**, and it's deliberately small. Only 25 companies are being invited in.

Pioneer Partners get the platform set up with the team's help, plus early adopter bonuses: **the first 25 companies receive 1 million free DataCreds and matched credits up to 5 million for 6 months.** You're not figuring it out alone, you get hands-on help wiring privacy-compliant player data collection into your game.

How to Apply (Two Steps):

1. Create your FairShare account. It's free, no card needed.
2. Apply for the Pioneer Program.

Create Your Free Account

Apply to Be a Pioneer Partner

Program details: myfairshare.io/games-pioneer-partner