# Data Flywheels and Public AI

Nicholas Vincent

2025-08-04

# Table of contents

# Preface

This is a "mini-book" that discusses "public AI flywheels": software meant to enable people to opt-in to contribute data towards "public AI" causes. The goal of this book is to support efforts build a transparent, people-centric data collection ecosystem that supports the evaluation and training of public-benefit AI models. If successful, public AI flywheels can create valuable data that materially improves public AI evaluation, research and development. If very successful, these flywheels might also play a role in solving thorny problems around the economics of information in a post-AI age.

More frankly, this is way to organize some design notes, practical documentation that's out of scope for a single example projec's repo, and longer abstract writing on the topic.

This document is organized as such:

- In "Part 1: Concepts", we explore the motivation and design space of public AI data flywheels.
- In "Part 2: A Case Study", we discuss one particular implementation of a Minimum Viable Product (MVP) opt-in flywheel meant to accompany a "public AI interface" (hosted interface software that hits various endpoints for "public AI models") that uses a "serverless" app + Git backend approach
  - This MVP focuses on collecting two high-signal data types: exports of "good chats" and "fail chats." This data provides immediate value for model evaluation and, at scale, can be used for fine-tuning. Importantly, collecting a list of good and bad chats is also immediately fun, so contributors can get some value before we reach a threshold of data volume needed to construct a full benchmark or dataset. We expect key ideas discussed in this doc, and concretized in this project, to generalize to other data types.
  - We also provide details on how a data retention policy for a concrete Public AI Data Flywheel might work, and more generally discuss the role of the data strategy for a "full stack" public AI application: from model endpoints to OpenWebUI interface to flywheel platform.
- The book includes some Appendicies with additional information.

# Part I

# Concepts & Rationale

# 1 Introduction

> Key insight: a public AI data flywheel is a system that enables a data collection feedback loop that embeds the principles of "public AI" – notably, transparency and accountability.

## 1.1 What is a data flywheel?

What is a data flywheel? Nvidia gives us this definition: "A data flywheel is a feedback loop where data collected from interactions or processes is used to continuously refine AI models." [1]

In general, a "data flywheel" is a system or set of systems that capture and/or incentivize data. A "flywheel" generally differs from a more general data collection system because the flywheel is embedded into some kind of application (as opposed to e.g. "standalone" data labeling tasks). So, if I just post a Google form to the Internet and say, "Hey, feel free to use this form to send me data!", that's just a form, not a "flywheel".

Generally, most data collection systems lean more towards utilizing either

- "sensor-style collection" (passive, instruments like cameras, microphones, or logging software, all of which lack an active "submit data" step) or
- "form-style collection" (active, requiring somebody to click "submit").

Historically, flywheels tend to imply a passive approach to data collection, but this is not necessarily a requirement. (More on this in a Chapter 3).

## 1.2 What is a public AI data flywheel?

First, what is "public AI"? The public AI network gives us this definition in a whitepaper from (Jackson et al. 2024): AI with

[1] That said, there is no doubt that for certain types of data, some people will need prevent their data from ending up in any public repositories in order to monetize effectively. The public AI data flywheel is only suitable for certain categories of data (in short: content that could be at home in a peer produced knowledge commons). Other types of data may be managed by complementary markets and sharing approaches.

> “Public Access – Certain capabilities are so important for participation in public life that access to them should be universal. Public AI provides affordable access to these tools so that everyone can realize their potential.” “Public Accountability – Public AI earns trust by ensuring ultimate control of development rests with the public, giving everyone a chance to participate in shaping the future.” “Permanent Public Goods – Public AI is funded and operated in a way to maintain the public goods it produces permanently, enabling innovators to safely build on a firm foundation.”

For more on the public AI concept, see also Mozilla’s work in this space and several workshop papers and preprints (RegML 2023 at NeurIPS, CodeML 2025 at ICML, workshop on Canadian Internet Policy).

Our focus in this mini-book is building “public AI” flywheels. To summarize heavily – if we try to achieve all the principles laid out in the large body work that tries to define “public AI” (and we should try!), we will face some unique challenges in the implementation of data flywheels.

In building public AI data flywheels, we are trying to create a feedback loop to improve AI by creating and collecting high-quality data (more on this in Chapter 2). However, the public AI principles mean that we likely want to start from a position of very high accessibility and very high accountability relative to other technology organizations and products. This means we need to provide an accessible explanation of exactly what happens to any data a user creates and give people real agency over the shape of the data pipeline. Ideally, public AI builders should also endeavor to make as many components of our stack as close as possible to public goods, which creates challenges around sustaining effort and funding.

Of course, it’s worth noting that some particular subset of the broad public (for instance, a particular city or state) could deliberate and make a collective decision that they prefer a more “traditional approach” to data flywheels. Very concretely, we could imagine a state conducting a referendum, and asking the public if they’d like a “public AI” product that follows industry standard practices around data and flywheels (sacrifing some degree of accessibility and/or accountability for other benefits). This might mean that the state deploys an AI chatbot with nearly the same data collection practices and privacy policies as organizations like Google or Anthropic.

In this mini-book, we are taking the stance that it’s best to start from a position of leaning heavily towards a highly accessible and accountable flywheel. We start by minimizing usage and retention of data; data that is used directly for AI research and development (R&D) should be provided via an opt-in by highly informed users.

## 1.3 Core Principles

We can translate the core principles of public AI to the data flywheel domain and arrive at roughly four requirements:

- Transparency for informed consent: Users must be fully informed about the models at play, the organizations who are building models, and the ramifications of any contributions to the flyhwheel. Ideally, users will also be informed about the training data underlying the models they use. A detailed FAQ and some kind of consent module (ideally going above and beyond standard Terms of Service[2]) are required before any data is shared. To some extent, maximally informed consent will require the active expenditure of resoures to improve the public's AI literary (i.e. we need to build AI literacy focused systems and perhaps even pay people for their attention). We need systems that really do inform people. Luckily, that's something it seems like AI can help with!
- Data Rights: A public AI data flywheel should empower users with control over their data, mirroring GDPR principles and similar regulations (this is also practically important for compliance). This includes the right to access (`$Art. 15$`), rectify (`$Art. 16$`), erase when possible (`$Art. 17$`), and port data (`$Art. 20$`). One exemplar project we might look to for inspiration around the implementation of data rights and legal terms is Mozilla's Common Voice (Ardila et al. 2019).
  - We note that data rights conflict with a "fully open" ethos; we will attempt to mitigate these tensions to the best extent possible.
  - We also note that public AI faces some unique challenges with cross-jurisdiction compliance; we discuss this at a high-level later on in **?@sec-ec**.
- Balancing reputation and pseudonymity: To the extent possible, we believe it is valuable to offer people the ability to contribute data with some kind "real account" attached, so people can earn credit and reputation. But this must be balanced with the benefits of also enabling pseudonymity or even anonymity contribution (see e.g. work from McDonald et al. (including this paper, (McDonald et al. 2019), and blog post).
  - In our MVP (discussed in the next chapter), an account with our OpenWebUI instance is required to make contributions, but users can choose to use a pseudonym (not unique; can for instance be "anonymous"). A hashed user id will be stored for internal purposes, but any public data releases will only use the pseudonym.
- Purpose Limitation & Licensing: Users should be able to specify their preferences for how their data is used (e.g., for public display? for evaluation? for future model training?). This can be captured using (new) IETF AI Use Preferences and Creative Commons Preference Signals, or other approaches that emerge. We will discuss below how this

[2] See e.g. Terms we serve with. (Rakova, Shelby, and Ma 2023)

might extend to other preference signal proposals and/or technical approaches to gating data.

  - This is critical for answering a likely FAQ around public AI data – if you succeed in creating actually useful training data or new benchmarks, won't private labs just immediately use that data as well?

# 2 Why collect data?

> Key insights: in general, we want more records that contain high-quality signals and/or observations about the world to be available to public AI organizations for training and evaluation.

If we want to build a data flywheel, it is probably useful to first specify why we want more data! This in turn can help us identify what types of data we want.

At its core, "data" is useful for AI (and for other things!) because it provides information about the world.

In general, it is intuitive that having more information will (generally) lead to better decision-making. [1]. Although there are some scenarios we might come across (or invent) where getting acquiring information is not helpful – because we might not have "room" in our memory for more data, or some records might not help us at a certain task, or data causes our model to get worse in some sense – most people benefit from having more records of high-quality observations and signals. (example: (Shen, Raji, and Chen 2024)

So let's put these more complicated cases aside for now, and make the assumption: in expectation, acquiring more high-quality data (that is "accurate", or reflects "insight") is useful. Oftentimes assessing data's quality, or its truthiness, or its insightfulness, is not at all easy! With this assumption in mind (and hearty caution about the thorniness of truth and insight), we can speak generally about the types of data we might acquire through a flywheel and that data will be useful.

## 2.1 An overly detailed accounting of all the ways we might generate LLM pre-training data

Speaking at very low-level, LLM pre-training data can come from any sensor or form that creates digital records that contain sequences of tokens. However, we generally don't want any old tokens – we want tokens that contain signals about the world and about people, and that have been organized (typically by people) in a way that captures the underlying structures of

[1] That said, there is no doubt that for certain types of data, some people will need prevent their data from ending up in any public repositories in order to monetize effectively. The public AI data flywheel is only suitable for certain categories of data (in short: content that could be at home in a peer produced knowledge commons). Other types of data may be managed by complementary markets and sharing approaches.

our world (or the structures that we people have imposed). In pre-training, it seems we can get away with mixing together many differents types of structure. For post-training, we may want specific structure (e.g. data produced by people following specific instructions).

We might further try to describe human-generated data in a very general fashion by saying: data is created when a person does something that leaves a digital trace: typing, speaking into a microphone, using other kinds of digital inputs like buttons ,controllers, etc. They might also operate a camera or other sensing instrument that captures signals from the world. We also sometimes may want to use truly "sensor-only data" (e.g., seismic readings), though those sensors are built, placed, funded, and so on by humans.

After typing, a person might use a terminal or GUI to send their inputs into some data structure – by committing code, editing a wiki, responding on a forum, and so on. Often, the person creating a record has a goal and/or a task they want to complete. This might be: ask a question, teach or correct something, build software, file a bug, summarize a meeting, translate a passage, or simply react to some information object (like/flag/skip). Critically, in practice, many high value sources of data also have some upstream social structure and corresponding incentives – institutions, communities, etc. that create meaningful incentives for people to produce records that are accurate, insightful, and so on. #todo cite key works about value of social media data, scientific data, etc.

- https://wikimediafoundation.org/news/2023/07/12/wikipedias-value-in-the-age-of-generative-ai/
- https://arxiv.org/abs/2410.08918v1

In other words, institutions and communities create incentives so that as people type (or otherwise digitize information), they don't just produce random sequences or the same common sequences repeatedly (or we might have an Internet of web pages that all say "I like good food"; don't we all…)

Moving to a more high-level overview, we might begin categorize LLM training data:

- Human-authored natural language: blogs, books, encyclopedias, news, forums, Q&A, transcripts (talks, meetings, podcasts), documentation, and manuals.
  - And now, some non-human-authored natural language (synthetic versions of any of the above).
- Code: source files, perhaps with licenses and provenance, issue threads, commit messages.
- Semi-structured text: tables, markup, configs (HTML/Markdown/LaTeX/YAML/JSON) that carry schema and relationships.
- Multimodal pairs (for VLM/ASR pretraining): image+text, audio+text, video+text, and associated captions/alignment.
  - Here, the pairing is a critical characteristic that makes this data unique. This implies somebody has looked at the each item in the pair and confirmed a connection (though paired data can be produced in an automated fashion).

- Metadata about data: records that describes characteristics of other records. language, domain/topic tags, timestamps, links, authorship/attribution, license, AI preference signals.
  - Quality signals: dedup scores, perplexity filters, toxicity/PII flags, heuristic or model-based ratings—used to weight or exclude.

Some specific tasks that might create especially useful data include:

- Asking a model a question and marking the response “good” or “fail”, optionally with a short note about *why*.
- Corrections/edits: rewriting a wrong answer; adding a missing citation; supplying a step-by-step solution.
- Pairwise preferences: “A is better than B because …” (useful for preference learning/DPO).
- Star ratings / rubrics: numeric or categorical grades on axes like factuality, helpfulness, tone, safety.
- Tagging according t os ome taxonomy: topic (“tax law”), language (“id-ID”), difficulty (“HS”), license (CC-BY-SA), and AI preference signals.
- Synthetic tasks: user-written prompts + *ideal* references (gold answers, test cases, counterexamples).
- Multimodal: an image with a caption; an audio clip with a transcript; a diagram with labeled parts.
- Programmatic contributions: code snippets with docstrings/tests; minimal reproductions of a bug.
- “Negative” structure: anti-patterns, jailbreak attempts, hallucination catalogs.

Of course, a key data for many AI systems is “implict feedback”: clicks, dwell time, scroll/hover, skips/abandonment. This data is typically collected via a “sensor” (logging software), not something users actively contribute through a form.

---

https://arxiv.org/abs/1712.00409 https://papers.neurips.cc/paper_files/paper/2022/file/7b75da9b61eda40fa35 Paper-Conference.pdf https://www.cs.ubc.ca/~hutter/earg/papers07/00585893.pdf

https://arxiv.org/abs/1812.11118

(**article?**){Belkin_2019, title={Reconciling modern machine-learning practice and the classical bias–variance trade-off}, volume={116}, ISSN={1091-6490}, url={http://dx.doi.org/10.1073/pnas.19030701 DOI={10.1073/pnas.1903070116}, number={32}, journal={Proceedings of the National Academy of Sciences}, publisher={Proceedings of the National Academy of Sciences}, author={Belkin, Mikhail and Hsu, Daniel and Ma, Siyuan and Mandal, Soumik}, year={2019}, month=jul, pages={15849–15854} }

# 3 A Democratic Data Pipeworks

## 3.1 How does data move from people to AI models — and where can we insert governance levers?

*This is a summary of a longer Data Leverage post.*

To further motivate the idea of data contribution with public AI principles, it's worth a brief discussion of what the overall "data pipeworks" of the AI industry looks like from a zoomed out view.

Key takeaways

- Modern AI can be understood as a five-stage pipeworks: (1) Knowledge & Values -> (2) Records -> (3) Datasets -> (4) Models -> (5) Deployed Systems.
- Treating AI as a cybernetic system puts feedback and control at the center. Contributors can steer outcomes by shaping data flow (more on the next chapter).
- Human factors dominate AI capabilities because they shape what gets recorded upstream. Interfaces, sensors, and incentives are therefore core AI R&D.
    - some trends may shift this – RL in real life, #todo cite experiental learning
- Properties of data create collective action problems (social dilemmas) that require markets, coalitions, and policy to fix.
- For public AI flywheels, thinking in terms of data piepworks reveals "insertion points" to add transparency, consent, rights, and preference signals so democratic inputs actually move the system.

## 3.2 Why a "pipeworks" view?

Most technical AI work zooms in on a clean optimization problem. But questions about who benefits, who participates, and how AI affects society live upstream and downstream of that problem. A "Data Pipeworks" view describes the end-to-end flow by which human activity becomes records, then datasets, then models embedded in systems that act on the world, and thereby change the future data we can collect.

This view pairs naturally with cybernetics/control: identify system state, actuators, sensors, and feedback loops; then decide which loops to strengthen or dampen.

## 3.3 Five stages of data

1. Knowledge & Values (Reality Signal): Humans (and the physical world) generate the latent “signal” AI tries to model (facts, preferences, norms). We don’t presume computability; we note its existence to emphasize sampling implications.
2. Records (Sampling Step): Interfaces and sensors transform activity into structured records (forms, clicks, edits, uploads, buttons, cameras, microphones). Design choices here shape what becomes legible to AI. Key idea: generally, any particular sampling instance either leans more towards “sensor” or “form”.
3. Datasets (Filtering & Aggregation): Organizations filter, label, merge, and license records under social, economic, and legal constraints. This determines coverage, bias, and what’s even available to learn from.
4. Models (Compression): Learning compresses datasets into input–output mappings. Modeling choices are path-dependent on Stages 1–3; data defines the feasible hypothesis space.
5. Deployed Systems (Actuation): Models are embedded in products, workflows, or infrastructure, producing value and externalities. Deployment feeds back by first, and foremost, **changing the actual world**. Deployment also alters incentives therefore affects future record creation.

Design note: small, well-placed interventions upstream can dominate large downstream tweaks.

## 3.4 Why this matters for governance and alignment

- Human factors are primary. The distributions the AI field is optimizing over are created, not discovered. Interfaces, defaults, prompts, consent flows, and incentives shape the topology of AI work.
- Social dilemmas are inevitable. Contributing high-quality records to a shared system is a collective action problem (free-riding, failure to reach critical mass). Today’s “dictator solution” (opaque scraping) collapses when people gain data agency.
- Data leverage (next chapter) is the steering wheel. Individuals and groups can alter records, licenses, and access. This allows people to steer model behavior by modulating data flow rather than model internals.
- Pluralism becomes measurable. Tracing contributions lets us quantify relative weight of individuals and communities, enabling pluralistic governance and new not

## 3.5 Where to place the levers (for public AI flywheels)

- Stage 1 to 2 (Knowledge to Records): invest in interfaces and sensors with informed consent; design contribution prompts and micro-tasks; support pseudonymity and reputation choices. Aim to raise signal quality and widen participation. Note that there will be an omni-present tension between informed consent and "frictionless" contribution. Can be resolved to some degree by building trust between public and public AI operators.
- Stage 2 to 3 (Records to Datasets): attach licenses and AI preference signals per record; validate, de-duplicate, and redact PII; publish partitioned releases. Make rights legible and keep high-trust, high-reuse bundles. Leaderboards, grants, bounties, governance hooks (votes, preferences) to sustain contributions and invite further steering.
- Stage 3 to 4 (Datasets to Models): enable data markets and coalitions, attribution, and sampling weights; build evaluation sets tied to provenance. Align training with community intent and enable bargaining. (more in this in the next section as well).
- Stage 4 to 5 (Models to Systems): publish transparent deployment notes, opt-outs, and model cards tied to data buckets. Surface externalities and set expectations for use.
- Stage 5 to 1 (Feedback loop): try to ensure that AI actually has positive benefits on the world. Improve standards of living, increase health, free-time, well-being etc. so people can become empowered active participants in whatever stage of the pipeline they please.

## 3.6 Implications for research and practice

Building flywheels are part of broader agenda to enable a data pipeworks. More in the next chapter on how data contribution through flywheels (including licensed or user-restricted contribution) interplays with data protection, data strikes, markets, etc.

## 3.7 A compact mental model

- Sensors and interfaces decide what counts.
- Filters and markets decide what persists.
- Compression decides what generalizes.
- Deployment decides what changes next.
- Governance decides who gets to steer.

Public AI flywheels turn that loop into a participatory control system: contributors see consequences, express preferences, and are (hopefully) rewarded for adding high-signal records.

#todo re-add cites from the original post!

https://www.annualreviews.org/content/journals/10.1146/annurev.soc.24.1.183

https://en.wikipedia.org/wiki/Cybernetics

https://www.vox.com/future-perfect/23787024/power-progress-book-ai-history-future-economy-daron-acemoglu-simon-johnson

https://probml.github.io/pml-book/book1.html

https://journals.openedition.org/cybergeo/1035?lang=en

https://dl.acm.org/doi/abs/10.1145/3531146.3533158

https://eckhartarnold.de/papers/2014_Social_Simulations/Whats_wrong_with_social_simulations.html

https://www.anthropic.com/news/influence-functions

https://users.ssc.wisc.edu/~oliver/PROTESTS/ArticleCopies/OliverMarwellCritMassI.pdf

https://raulcastrofernandez.com/papers/data_station_paper-11.pdf

# 4 Flywheels and Bargaining Power

Key insight: Beyond improving public AI systems, getting public AI data flywheels right can make it easier for people to use data flow as a source of (collective) bargaining power to (1) participate in markets and (2) participate in governance and alignment.

## 4.1 How can a public flywheel give people real power over AI systems?

Based on Chapter 2, we can arrive at a very obvious argument for a data flywheel: the flyhweel will produce data, and that data will make AI better!

But this isn't the only benefit of building flywheels in a "public AI" manner. Doing so can also enhance the amount of agency that people have over data flow, and make "voting with data" possible such that the public has more power to govern and align AI systems.

In short, AI is somewhat unique relative to other technologies, because of its data dependence. Data comes from people. The fact that this powerful technology has a dependency on people from around the world means that AI has a natural "governance lever".

Setting up a public AI data flywheel is thus important not only to improve AI capabilites; success of public AI data flywheels can collectively help to solve some (but not all!) of the thorny governance and alignment challenges that AI poses by fundamentally changing the data pipeworks of AI.

You can read about data leverage via this newsletter or even via this dissertation. For a short summary, of "voting with data to improve alignment", check out this post: Plural AI Data Alignment.

It's worth pulling out two distinct ways that a flywheel can interact with AI and governance:

- A flywheel with no attempt to capture contributor intent or provide data rights may still serve to increase available data, either in fully public repos or in databases accessible by public AI labs. This outcome could still make public models a bit better and help to keep public labs competitive at the margins, but it would not change the bargaining relationship between contributors and model builders.

- A well-governed flywheel that effectively manages the tension between opt-in and friction/ease-of-use can seriously reshape the broader data pipeworks/ecosystem/economy. Ideally this flywheel would also capture provenance, per-item licensing, and per-item AI-use preference (or even enforceable contracts – "you must pay some organization to use this data", or "you must follow this policy around openness, safety, alignment, etc"). Such flywheels would turn contributions into units that can be assembled, priced, withheld, or targeted, opening the door to markets and, if necessary, strikes.

## 4.2 How an opt-in flywheel enables markets

An opt-in flywheel can create the prerequisites for functioning data markets without turning the project into "just a marketplace."

Critically, on day one of the data flywheel, each contribution is a unit with provenance, license, usage preferences, and minimal schema. There is also the immediate possibility to associate contributions with reputations of contributors or collectives. This is close to something that is legible enough to transact on. While the initial goal would be to promote conscious data contribution towards public AI causes, it is possible that some data contributors could also use the legibility and the organizing effects of the flywheel to also sell some data to private actors. Indeed, a model already exists that enable people to make public contributions that benefit public interest actors while still allowing large private organizations to pay for data contractually: Wikimedia Enterprise. Wikimedia data is open to all, but Wikimedia is able to monetize "enterprise-level access". [1]

As the data flywheel "spins up", a community could form around the open data to build leaderboards, scarcity tags (rare language/domain), and quality scores. This would effectively begin to generate price signals. A bounty board ("need 5k labeled failures in X") would serve to convert demand into targeted supply. An exemplar here would be bounty boards for open source software. While the outputs of such bounty boards are code contributions that become OSS (and thus non-excludable), it's still possible to have market dynamics emerge.

Co-ops/unions/intermediaries can represent contributors, negotiate bundle terms, run audits, and set default preferences. The flywheel provides a starting shared ledger and release cadence that markets need. (Again in some cases, the intermediary may need to "move off" the flywheel and transact directly in a market).

The key idea here is that it's possible to enable market activity under two distinct sets of conditons: one in which data is kept open-but-gated-and-restricted ("markets" for bespoke Wikimedia Enterprise style packages) or by using the flywheel as a stepping stone towards a

[1] That said, there is no doubt that for certain types of data, some people will need prevent their data from ending up in any public repositories in order to monetize effectively. The public AI data flywheel is only suitable for certain categories of data (in short: content that could be at home in a peer produced knowledge commons). Other types of data may be managed by complementary markets and sharing approaches.

more “property-like” market (people organize using the flywheel community or use preference signals as exemplars, then form a data intermediary to collectively bargain directly with data users).

## 4.3 How an opt-in flywheel enables strikes (or credible refusals)

A data strike here means a coordinated, temporary withdrawal or constraint on high-signal contributions or releases, or retroactive deletion of data (which in some cases, with legal support, could trigger legally enforced retraining https://cyberscoop.com/ftc-algorithm-disgorgement-ai-regulation/ – though TBA on how this will play out in 2025 onwards).

What makes strikes possible:

- Voluntariness is preserved. Because contribution is opt-in, non-participation is a legitimate default.
- Release control. A waiting-room, processing, release pipeline provides a natural “valve” for cadence changes or strikes.
- Shared visibility. Everyone sees dependence on fresh contributions (e.g., evaluation drift). Visibility creates leverage.

There are many variants of data strikes in a flywheel ecosystem:

- Quality freeze. Contributors keep using systems but withhold labeled “good/fail” chats or corrections for a period.
- Selective embargo. A community with scarce data (language/domain) pauses releases or flips new records to “evaluation-only.”
- Preference shift. New contributions change AI-use preferences to deny training unless a stated condition is met (funding, governance, attribution).
- Rate limit. Collectives cap monthly volume to force negotiations on price or terms.

What a strike cannot do (and shouldn’t promise):

- Undo past licenses. Items released under irrevocable terms (e.g., CC0, CC-BY) remain available.
- Prevent copying entirely. Public releases can be mirrored; anti-scraping reduces risk but does not eliminate it.
- Guarantee compliance outside the ecosystem. Preference signals work when counterparties agree to honor them or when law/policy backs them.

# 5 Flywheel design space

#todo heavy rewrites in this section.

> Key insight: There is a broad spectrum of technical implementation of the flywheel, ranging from traditional database-on-a-company server to low-friction-peer-production (our preferred MVP) to radical approaches (e.g. truly federated data acess).

## 5.1 Purpose of this section

This section gives more context about the many ways we might built flywheels, and lays out alternative governance paths and a future work (in particular, a focus on futures that involve healthy data markets, data intermediaries, federated learning, etc.)

We also discuss why we think an approach that includes a minimal retention frontend + opt-in flyhweel platform can serve as a pragmatic bridge to more advanced approaches. For instance, we can use the patterns and concepts discussed here to move towards independently governed data co-ops, eventual federated learning, etc.

## 5.2 More on all the other approaches we could've taken

First, let's lay out a toy model of data "creation" and "flow" (this will come again Part 2, when we walk through the flow for a real flywheel app).

In Chapter 2 we talked about the numerous combinations of sensors, forms, task settings, social structure from institutions, communities, etc. that might exist, and in the Appendix we discuss a number of formats and types of data for LLMs in particular.

To summarize, an AI developers might collect or use some of the following kinds of data:

- Simple Signal: Binary feedback ( / ), star ratings, or flags
- Annotated Conversation: Full chat with user corrections, ratings, or notes
- Preference Pair: A/B comparisons between responses
- Examples: User-created prompts and ideal responses
- Structured Feedback: Form-based input (error type, severity, correction)

- Multimodal Bundle: Text + images + voice + metadata
- More advanced structured data …

Further, the creation of data could be prompted at several points in time:

- Proactive: User initiates contribution unprompted (e.g., "Share this chat" button)
- Reactive: System prompts based on signals (e.g., after trigger word, patterns in usage behaviour, ask "What went wrong?")
- Passive: Automatic collection with prior consent (e.g., telemetry, browser extension)
- Scheduled: Regular prompts (e.g., weekly "best conversations" review)
- Task-Based: Specific requests for data types (e.g., "Help us improve math responses")

This choice will likely impact the level of "friction" users experience, roughly:

- Zero-Friction: Purely passive
- Almost zero-friction: Purely passive with some regular re-consenting process (monthly or yearly "checkup" on sharing settings)
- Low-Friction: One-click actions with no interruption
- Medium-Friction: Multi-click actions or actions that redirect to separate interface
- High-Friction: Multi-step process, account creation, or technical skills required

Data might also be processed at one or more points in time (In practice, there is likely be some degree of "processing" at various steps, but it is important to clarify this to users):

- Pre-submission: Client-side processing before data leaves user's device
- On-submission: Real-time processing during the contribution flow
- Post-submission: Batch processing after data is received
- Pre-publication: Review and processing before making data public
- On-demand: Processing happens when data is accessed/downloaded

So, person visits an AI interface (e.g. visits a chatbot product on a website). They sit down, enter a prompt, and then react to the Output (take the information and do something with it, follow up, leave positive or negative feedback, etc.). This is our canonical object of interest: a prompt ("Input"), response ("Output"), and optional follow up data (feedback, more queries and responses, etc.).

Typically, this data must live, for some time, on the user's device. It must also be processed by an AI model ("inference"), which involves sending a payload to a hosted service or some local endpoint (if e.g. user is running open weights on their own device). It may or may not be stored on the server/system (we'll use these interchangeably for now to refer to all the devices controlled by the organization running each module) where the interface is hosted. It may or may not be stored by the server/system where the model is hosted. And finally, a flywheel may send that data to a third location.

This final data could live in a centralized database (e.g. traditional relational database), a public repository (e.g. GitHub, HuggingFace), totally local, or even in some kind of distributed network (IPFS, BitTorrent).

Finally, the resulting flywheel-produced data might be accessed in a number of ways:

- Direct Download: Raw access to complete dataset (with rate limits)
- API Access: Programmatic access with authentication and quotas
- Static Site: Read-only web interface with anti-scraping measures
- Gated Access: Application/approval process for researchers
- Hybrid Access: Public samples + gated full access, or public metadata + restricted content
- Streaming Access: Real-time feeds for continuous model training

So we have five useful questions for classifying flywheel designs:

- Where data lives: …
- When prompted: …
- When processed: …
- How accessed: …
- Friction level: …

## 5.3 Some Categories of Architectural Models

With all these design choices in mind, it will be useful to describe the general approaches we might take to build a data flywheel.

### 5.3.1 Standard "PrivateCo" Web App

An obvious option is to simply build a hosted "standard" "PrivateCo" / start-up style web app. If Netflix is successful because of its flywheel, why not just build a public AI data flywheel that looks like a private tech company's product from a technical perspective? Indeed, in some contexts it may make sense to skip building an opt-in flyhweel and simply use the data generated by users directly for training, eval, etc. In this case, there is no "third location" needed; just read data from the existing prod database. While one could argue that the Terms of Service for many existing tech products do make these products "opt in" in some sense, there are also serious downsides to the status quo. Many would argue that standard practice in tech (long, difficult to read Terms of Service and Privacy policy documents; opacity about exact details of data collection and usage; general challenges in conveying the complexity of modern data pipelines) make it hard for the standard PrivateCo Web App model to offer truly informed consent for data contribution. (For more on general issues with ToS, see e.g. Fiesler, Lampe, and Bruckman 2016 #todo add more of the "classics" of this genre.)

While some users might even prefer this approach, we believe this would **not** be a good starting place for a public AI data flywheel. We also believe it's important to communicate to users how the public AI interface differs standard practices (for instance, how does a public AI model differ in terms of data use from e.g. using ChatGPT, Gemini, or AI overviews via search).

The defining characteristics of this approach is that data is held by a private entity at all times. Under this approach, we can collect all types of signals, mix proactive and reactive data collection, use telemetry freely, process data whenever we want. It's highly likely under this approach, data from a flywheel would live in centralized, privately governed database.

Answering each of the questions posed above:

- Where data lives: private database
- When prompted: flexible
- When processed: flexible
- How accessed: flexible; likely API
- Friction level: flexible; likely low

It's also likely we would want to follow corporate practices in locking down the final data, which makes this a bad choice for maximizing publicly visible output. Put simply: while an interesting idea in theory, we probably can't run an AI product that has a prod database that is openly readable by the public.

Within the broad umbrella of taking a "Standard PrivateCo Web App" approach to data flywheels, some archetypes might include:

- Telemetry heavy approach (imagine an LLM chat app with no feedback buttons, but lots of data is collected re: dwell time, conversation length, user responses, etc.)
- Feedback heavy approach (imagine an LLM chat app where the UX is heavily focused on asking users to use thumbs up / thumbs down buttons, or presenting users with frequent A/B test responses)

### 5.3.2 Git/Wiki Platform

Another option to build a "very active flywheel" (that arguably stretches the definition because friction will be very high) is to just use peer production or version control software (a "wiki" or "git" approach) and just ask people to make their contributions using existing contribution avenues (for instance, "editing" a wiki page or making a "pull request" to a version-controlled git repository).

If we choose this approach, we do likely constrain our answers to the above questions:

- Where data lives: Public repository
- When prompted: Proactive (user initiates)
- When processed: Pre-submission (user does it) + CI/CD validation

- How accessed: Direct download via Git + web interface
- Friction level: High (technical knowledge required)

This approach has maximum transparency, built-in versioning, and low cost. But, it is likely to exclude non-technical users and has very high friction even for technical users.

Example Stack: some combo of MediaWiki, GitHub, GitLab, HuggingFace + CI/CD validation

### 5.3.3 Web service + Git Platform

The option described in Part 2 is to use a Git/Wiki approach, but use some kind of "serverless" approach (or a more traditional app; doesn't have to be serverless) with special endpoints that are triggered by users via low friction in-app actions (clicking a special button, entering special command, etc.) that writes to a Wiki / Git repo on the contributor's behalf. We could also build a system so that users can effectively commit data to the source control / wiki system automatically (e.g., "Every day, run an anonymization script on my chat history and then write the output as a new file to a shared, version-controlled server").

- Where data lives: Public repository
- When prompted: Proactive or reactive
- When processed: On-submission via serverless function
- How accessed: Git access + static site generation
- Friction level: Low (automated complexity)
- Pros: Transparency + usability, serverless scaling
- Cons: Technical issues Cold starts, API rate limits, complex error handling
- Example Stack: Vercel/Netlify + GitHub API + Hugging Face Hub

### 5.3.4 Federated Learning Model

One radically different approach might involve using federated learning.

- Where data lives: User devices (distributed)
- When prompted: Passive with consent
- Information object: Model gradients or aggregated statistics
- When processed: Pre-submission (on-device)
- How accessed: Only aggregated model updates available
- Friction level: Zero after setup
- Pros: Maximum privacy, no data transfer, infinite scale
- Cons: Complex implementation, limited debugging, device requirements

### 5.3.5 Browser Extension

We could implement a flywheel that relies on users downloading a browser extension! This only reflects a data ingestion choice: can be used with various backend choices above.

- **Where data lives:** Centralized or distributed
- **When prompted:** Proactive or passive
- **Information object:** DOM captures, interaction logs, selections
- **When processed:** Depends on backend
- **How accessed:** Depends on storage choice
- **Friction level:** Very low after installation

### 5.3.6 Export-based approach

Another idea is to build a flywheel that leverages existing export features and export mechanisms. Instead of adding feedback buttons or telemetry, flywheel designers could simply create a static site that lets users manually upload exported data from various apps. This would require manual effort (and some friction could be reduced via careful attention to UX, adding features to help standardize data, etc.) but could be powerful in jurisdictions with portability/export rights.

### 5.3.7 Other experimental approaches

Other approaches to building a flywheel might involve more radical approaches to decentralizing the actual data storage, for instance using peer to peer protocols, various crypto/web3 approaches to data sovereignty, etc.

## 5.4 Scenario Walkthroughs: A Practical Comparison

Here, we walk through two common scenarios and describe what happens (in one sentence) for each of the architectures described above.

#todo: these could be made crisper to highlight the key differences better (But also be honest about where there are similarities)

### 5.4.1 Scenario A: User marks a chat as "Good", but the flywheel needs to do some checks for personally identifying information (PII) – when does processing happen?

- **Web App:** Redirects to platform, PII scrubbed on submission, available via API after review
- **Git/Wiki:** User removes PII manually, creates PR, instantly visible on merge
- **Telemetry:** Signal sent, processed in real-time, only visible in aggregates
- **Hybrid:** Signal sent immediately, full chat processed if shared
- **Serverless+Git:** Modal appears, serverless function strips PII, PR created automatically
- **Federated:** Local processing only, contributes to next model update
- **Extension:** Captures state, removes PII client-side, sends to chosen backend
- **P2P:** Processes locally, shares with peers who validate before propagating

### 5.4.2 Scenario B: User corrects a factual error

- **Web App:** Editor interface, toxicity check on submission, published after human review
- **Git/Wiki:** User edits markdown, CI/CD checks format, visible immediately on merge
- **Telemetry:** Only captures "error" signal, no correction possible
- **Hybrid:** Error signal triggers correction UI, correction queued for review
- **Serverless+Git:** Inline correction, automated PII/toxicity checks, PR needs approval
- **Federated:** Correction processed locally, differential privacy applied
- **Extension:** Highlights error, pre-processes correction, sends to backend
- **P2P:** Broadcasts correction, network consensus before acceptance

### 5.4.3 Scenario C: Accessing the contributed data

- **Web App:** Researchers apply for API key, public sees samples on static site
- **Git/Wiki:** Anyone can clone repo, but rate-limited through CDN
- **Telemetry:** Only aggregated statistics available via public dashboard
- **Hybrid:** Public can see signals dashboard, researchers apply for conversation access
- **Serverless+Git:** Public (or gated) repo with all data, static site with search/filter
- **Federated:** No direct data access, only model checkpoints released
- **Extension:** Depends on backend choice, typically follows that model
- **P2P:** Must run client to access network, can specify data sharing preferences

## 5.5 Frontier approaches: data cooperatives, federated learning, and more

In many cases, users may want to have data governed by community organizations (e.g., organized by domain/region/language) that hold rights and decide release cadence, licensing defaults, and benefit policies.

Practically, taking a collective/intermediary focused approach has the potential to massively reduce user friction / attention costs. One visio for a low friction data intermediary approach is: users spend some time once a year choosing which intermediaries to join. Upon joining, they can choose to delegate key decision-making and participate in intermediary governance as suits their desires and needs. If joining process is good + governance is good, can achieve good outcomes!

We note that if an implementation of the flywheel is built on top of open-source software, communities can easily choose to deploy their own instance and their own data flywheel and effectively operate entirely parallel, self-governed instances. If they also choose to share opt-in data via similar licensing and preference signal approaches, such datasets could be easily merged – but with fine-grained adjustments to precise details (e.g., slight modifications on retention, access, release cadence, content moderation, adn so on.) Of course, data co-ops may choose to use quite different technical stacks. This approach is just one among many.

### 5.5.1 Transitioning for opt-in flywheel to federated learning

It may be possible to also move from an opt-in data flywheel approach to a federated learning-first approach. Here, model training occurs across user or institutional nodes; only gradients/updates (with privacy tech) are centralized. The dataset remains partitioned or local; central custodian minimized. This approach would:

- Reduces central data custody and breach surface
- Aligns with data-residency and institutional constraints
- Enables "learning from data that can't leave"

But has some major downsides / existing barriers:

- Harder reproducibility and data auditability
- Complex privacy stack (secure aggregation, DP, client attestation)
- Benchmarking must be redesigned (federated eval)

This is a bigger leap, but we believe it's important to begin to think about how the implementation of the Public AI Data Flywheels might support communities wishing to transition towards an FL approach.

One rough sketch might look like: * Build the MVP defined in Chapter 2 * Ship license + AI-preference metadata (MVP). * Maintain gated HF releases and public leaderboards/full data access. * Publish provider-payload transparency and link to provider terms (no guarantees). * Process deletions via HF mechanisms when possible; keep our mirrors in sync. * Phase 1 — Co-op pilots * Charter one or two community co-ops; define bylaws, scope, and release cadence. * Spin up many instances of interface + flywheel combos (can fork software directly, or use similar approaches) * Establish a concrete sharing / merging plan * And beyond! * Once several independent data communities, are operated, it might be possible to move from lightweight sharing and merging to more serious federation with technical guarantees. Perhaps this might start with federated evaluation and then move to federated training. Much more to do here, out of scope for this document.

# 6 Ethics and Compliance

Public AI data flywheels with face numerous ethics and compliance challenges.

This mini-book does NOT provide specific legal advice. We do discuss and link to terms of service used by various platforms.

In Part 2, we provide an example of a platform specific data policy document + terms document.

## 6.1 Ethics

### 6.1.1 Flywheel-particular challenges

There is a large literature on harms from AI and sociotechnical systems more generally. We provide a longer set of references at the end of this section.

The top ethics priority for a PAIDF is figuring out informed consent, and balancing consent and friction. One worst case scenario for a a public AI organization is that the flywheel is set up in a way that erodes user trust and ultimately hinders the broader public AI mission.

While designing ethical systems normally involves some degree of multiplicity (there is a rarely a single "most ethical solution" for a given group of people), our overall stance is that informed consent can be achieved by maximizing user information about data use and taking a fundementally opt in approach.

Beyond consent, a number of other interesting ethics challenges arise. We describe them first, and then discuss the intersection between building an ethical flywheel and a compliant flywheel.

In particular, there are three flywheel specific concerns, that primarily stem from the very general nature of modern AI data. First, it is possible that data that is contributed via the flywheel could create serious security concerns (contributing a chat that includes an injection attack). Second, data that is contributed could create privacy concerns (PII and sensitive strings, from email, names to API keys). And third, data that is contributed be seen as expressively harmful or leading to representational harms. That is, some users might produce data that is very offensive to other users. This is likely inevitable in a large enough system, and so public AI flywheel designer must plan with values conflict in mind.

In short, when we open up a form to the world, people may enter things (even in good faith) that creates security risks, violates privacy, or violates social norms. We

There are also a set of ethical risks that arise from downstream AI systems that we build/improve with flywheel data. While these are not the focus of this mini-book, it is critical to keep them in mind. A non-exhaustive list includes:

- AI-driven expressive harms: the system produces content that demeans, stereotypes, or legitimizes abuse against protected groups
- AI-driven representational harms: skewed data makes groups invisible or mischaracterized (e.g., images that underrepresent darker skin tones; code comments that assume a single gender)
- allocative harms: outputs affect access to opportunities or resources (moderation, ranking, credit scores)
- privacy harms at the model layer (distinct from data layer): re-identification, doxxing, accidental leakage of personal or sensitive data
- security harms (distinct from data layer): prompt injection and data exfiltration via model behavior; poisoning of training or eval sets
- IP and contract harms: misuse of copyrighted or licensed content; violations of platform terms

### 6.1.2 Flywheel-specific high level goals

To balance these ethical challenges, we might organize our design around high-level goals that often appear in AI regulation and ethical discussions. These might include "purpose limitation" (our flywheel should try to collect only data that is necessary for the stated task – evaluating and improving AI systems) and "proportionality" (we should weigh utility of data collection against the likelihood and severity of harm; to some extent, because the flywheel leans opt-in, some decision-making is delegated to contributors). Considering the more general set of AI harms above, we may also want to specifically acquire or filter data in a way that helps achieve fairness goals.

Typically, you will see works attempt to classify high-risk data which should be treated differently. Examples include:

- faces, voices, gait, or other biometrics
- images of minors or contexts involving schools and hospitals (Federal Trade Commission 2013; U.S. Department of Education 1974; U.S. Department of Health and Human Services 2000).
- intimate or medical contexts, support forums, addiction and mental health groups
- government IDs, financial records, geolocation trails, and precise timestamps
- credential artifacts: API keys, cookies, session tokens, SSH keys, access logs
- content from communities with clear norms against scraping or model training

- datasets whose provenance is unclear or license compatibility is uncertain

A flywheel designer likely wants to avoid collecting this kind of data, but getting 100% precision will be nearly impossible, because some of the most interesting AI outputs (especially failure cases) may involve high-stakes scenarios. A flywheel that completely bans contributions related to cybersecurity or human health risks collecting "bland" data.

### 6.1.3 Levers for solving these ethics challenges

The flywheel designer can several avenues for attempting to pre-empt some of the above challenges. In terms of informed consent, this comes down to the implementation of a usable, informative module for consent and the exact UX for opting in and out. In terms of security and privacy, this mainly comes down to implementing filtering/curation at various stages. In terms of values conflict, the designer may employ filter, but also take a normative or sociotechnical approach (leaning on peer production-style talk pages, moderation, community-generated rules, etc.). The designer has the least leverage to directly control downstream model harms, but can have some influence via further training data filtering, helping to document data produced by the flywheel, etc.

## 6.2 Compliance

In general, data protection regimes impose responsibilites on anyone operating a platform.

Most likely, any public AI data flywheel will also be connected some frontend (e.g., hosted OpenWebUI instance) and some backend (model provider). These distinct systems are likely to have their own data-related responsibilites, depending on exactly how they hold or process data.

### 6.2.1 Risks

In terms of compliance risks, some issues may emerge because of contributor mistakes: users may post personal data that evades whatever filtering/curation the designer has implemented. In way, PII, secrets, or identifiers may make it into the flywheel's data repo. Further, even when users make contributions via pseudonym, unique phrasing or context can deanonymize.Ssalted contributor hashes are still stable identifiers across contributions

In general, a major risk with an approach that creates publicly accessible data is the potential for permanence via forks and mirrors. Removed data can persist in external forks, local clones, or third-party mirrors outside this project's control. Further, while repo history can be rewritten and monthly files reissued, but downstream models may already have trained; unlearning is best-effort and not guaranteed

Another issues related to the use of various vendors. Hosting providers (e.g. Vercel and similar services, any caching databases uses, any APIs used) may retain request logs; this is outside the app's control.

In some cases, contributions that create "security-related ethical risks" (e.g. a chat in which an LLM provides instruction for conducting some kind of attack) could also create compliance risks. This creates some continuous burden on maintainers. The same is true of offensive content or privacy violations. Even with consent and public repos, some jurisdictions treat certain content types as sensitive or restricted

## 6.3 Further reading:

First: works that taxonomize harms (Shelby et al. 2023; Weidinger, Mellor, et al. 2021; Blodgett et al. 2020)

Allocative harms: outputs affect access to opportunities or resources (moderation, ranking, credit-like inferences) (Barocas and Selbst 2016; Obermeyer et al. 2019).

Works that discuss expressively harms and representative harms (Shelby et al. 2023; Weidinger, Mellor, et al. 2021; Buolamwini and Gebru 2018; Grother, Ngan, and Hanaoka 2019; Crawford and Paglen 2019; Blodgett et al. 2020).

On data that has actual security concerns (contributing a chat that includes an injection attack) (OWASP 2023; Hubinger et al. 2024; Carlini et al. 2024).

On PII and sensitive strings (from email, names to API keys) (McCallister, Grance, and Scarfone 2010; European Union 2016; Illinois General Assembly 2008; "Rosenbach v. Six Flags Entertainment Corp." 2019; Carlini et al. 2019, 2021).

- privacy harms: re-identification, doxxing, accidental leakage of personal or sensitive data (Sweeney 2000; Narayanan and Shmatikov 2008)

Further reading on:

- purpose limitation and reversability: (European Union 2016).
- proportionality: weigh utility against the likelihood and severity of harm (*ISO/IEC 23894:2023 Information Technology—Artificial Intelligence—Risk Management* 2023; NIST 2023).
- respect for context: treat data according to the social norms of its origin community (Nissenbaum 2004; Jo and Gebru 2020).
- transparency: explain collection, uses, and the limits of control in clear language (Mitchell et al. 2019; Gebru et al. 2018; Holland et al. 2018).

- accountability: assign owners, metrics, and escalation paths (NIST 2023; European Union 2024).
- fairness and non-discrimination: measure and mitigate disparate impacts (Barocas and Selbst 2016; Selbst et al. 2019; Obermeyer et al. 2019; Bender et al. 2021).
- security harms: prompt injection and data exfiltration via model behavior; poisoning of training or eval sets (OWASP 2023; Carlini et al. 2024).
- IP and contract harms: misuse of copyrighted or licensed content; violations of platform terms (U.S. Copyright Office 2024; Creative Commons 2023).

# 7 Upstream data and data contribution

Data flywheels / contribution pathways are one part of the broader "data strategy" for an AI product or organization. Another key factor is making the full public AI pipeline is transparent is telling users about upstream data. Typically, the terms of service for an application or flywheel try to tell users where the data will go; but it can also be useful to tell users about where the data/AI come from.

### 7.0.1 AI builder attribution

At a high-level: in each interaction between users and a public AI system, we want to attribute the organization who did the hard work of prepping a model.

- The custom text, branding, etc. can be org specific. Make sure all model builders are happy. Can even highlight other interfaces/endpoints, something private AI would never do.
- Important to get this right so that model developers don't "back out" of the inference MVP and just switch to their own sovereign interfaces

### 7.0.2 Data attribution

Another way that public AI platforms can differentiate themselves from private AI is by heavily emphasizing data attribution. This might involve showing users data cards, incorporating features like OlmoTrace (Liu et al. 2025), etc.

## 7.1 Why does upstream matter?

Telling users about upstream data is a key part of system-wide transparency. Transparency on both fronts (model builders, data) has the potential to provide further incentive to users to provide data in the first place (because, e.g., they specifically want to support one of the organizations providing models or data).

There are a number of other exciting conncetions between data valuation/attribution, collective action in data (algorithmic collective action, data leverage), and flywheels.

## 7.2 Further reading:

#todo

# Part II

# Case Study: Low friction peer production

# 8 The OpenWebUI Action MVP

## 8.1 Overview

Our v1 MVP is implemented as an OpenWebUI Action that enables opt-in data contribution directly from the chat interface to a HuggingFace dataset repository. This approach leverages OpenWebUI's existing infrastructure and user accounts, eliminating the need for a separate flywheel website.

## 8.2 Architecture

The flywheel consists of:

- **Frontend**: OpenWebUI instance at https://app.publicai.company/
- **Action Plugin**: Python-based OpenWebUI action that handles contributions
- **User Settings**: OpenWebUI's "User Valves" for persistent preferences
- **Data Storage**: HuggingFace dataset repository at `publicai/shared-chats`
- **Processing Pipeline**: Asynchronous scripts that process the waiting room

## 8.3 How Contribution Works

### 8.3.1 User Setup (One-time)

1. User creates an OpenWebUI account at https://app.publicai.company/
2. User opens **Settings / Valves / Functions**
3. User toggles **Sharing Enabled** to ON
4. User selects:
    - **License**: CC0-1.0, CC-BY-4.0, or CC-BY-SA-4.0
    - **AI Preference Signal**: IETF/CC preference (e.g., `train-genai=n;exceptions=cc-cr`)
    - **Pseudonym**: Use username, anonymous, or custom name
    - **Auto-feedback**: Whether to skip feedback prompts

### 8.3.2 Contributing a Chat

1. User has a conversation with any model
2. User triggers the "Public AI Data Flywheel" action
3. System shows current settings and asks for confirmation
4. Optional: User provides feedback or context
5. Action creates a contribution JSON with:
    - Conversation messages
    - Metadata (model, tokens, timestamp)
    - User's license and AI preference selections
    - Attribution (based on pseudonym setting)
    - Contributor hash (anonymized ID)
6. Contribution uploads to `_waiting_room/` in HuggingFace repo

### 8.3.3 Data Processing Pipeline

1. **Waiting Room**: Contributions land in `_waiting_room/` directory
2. **Validation**: Daily script processes pending files
3. **PII Redaction**: Automated removal of emails, SSNs, phone numbers, etc.
4. **Quarantine**: Files with >5 PII hits or errors go to `_quarantined/`
5. **Release**: Clean contributions move to monthly buckets (e.g., `data/2025-08.jsonl`)
6. **Distribution**: Published via gated HF repo and future public gallery

## 8.4 Key Features

### 8.4.1 Privacy & Attribution

- **Contributor Hash**: SHA256 hash of `salt + "openwebui:" + user_id` (16 chars)
- **Pseudonymity**: Users choose between username, anonymous, or custom pseudonym
- **Avoid unintended PII in final dataset**: Some automated redaction before release

### 8.4.2 User Control

- **Opt-in only**: Requires explicit enabling in settings
- **Per-contribution consent**: Confirmation before each share
- **Persistent preferences**: Settings saved in User Valves
- **License selection**: Per-user default (not per-contribution in v1)

### 8.4.3 Safety Features

- **Mock mode**: Test contributions without actual upload
- **PII detection**: Email, IP, SSN, IBAN, crypto wallets, phone, credit cards
- **Quarantine system**: Suspicious content held for review
- **Rate limiting**: Handled by OpenWebUI's existing limits

# 9 Data Flow and Data Retention

*Last updated: 2025-08-18*

This document describes the data retention policy for the Public AI Data Flywheel integrated with OpenWebUI at https://app.publicai.company/.

## 9.1 How Users Create Data

### 9.1.1 Using OpenWebUI (No Sharing)

When you use OpenWebUI normally: - **Chat data**: Stored in OpenWebUI database for your history - **Account data**: Email, username, preferences - **Not shared**: This data is NOT sent to the flywheel unless you explicitly opt in

### 9.1.2 Sharing via the Flywheel Action

When you enable sharing and trigger the action: 1. **One-time setup**: Enable sharing in Settings / User Valves #todo better instructions here 2. **Per-chat action**: Explicitly trigger "Public AI Data Flywheel" 3. **Confirmation required**: Review settings and confirm each share 4. **Data sent**: Chat content, metadata, your chosen license/preferences

## 9.2 Data Retention Schedule

### 9.2.1 OpenWebUI Instance Data

#todo: these are currently tentative retention schedules.

| Data Type | Retention | Deletion Method |
|---|---|---|
| Chat history | Until user deletes | User can delete anytime |
| Account data | Until account deleted | 30 days inactivity → auto-delete |
| Operational logs | 30 days | Automatic purge |
| Security logs (IP/UA) | 30 days | Automatic purge |

### 9.2.2 Flywheel Contribution Data

| Stage | Location | Retention | Access |
|---|---|---|---|
| Waiting room | `_waiting_room/` | Until processed (typically 24-48h) | Private |
| Quarantine | `_quarantined/` | Indefinite (manual review) | Private |
| Released dataset | `data/YYYY-MM.jsonl` | Permanent | Public (gated) |
| Public gallery | Future site | Permanent | Public |

## 9.3 Data Processing Pipeline

### 9.3.1 1. User Contribution

- User enables sharing in User Valves
- User triggers flywheel action in chat
- System shows license, AI preference, attribution
- User confirms or cancels

### 9.3.2 2. Waiting Room

Location: waiting_room/{contributor_hash}{timestamp}.json Contains: Full contribution JSON Duration: Until daily processing script runs Access: Repository maintainers only

### 9.3.3 3. PII Redaction

Automated detection and removal of: - Email addresses - IP addresses (IPv4/IPv6) - Social Security Numbers - IBAN accounts - Crypto wallets (ETH/BTC) - Phone numbers - Credit cards (Luhn-validated) - Name patterns ("my name is X")

### 9.3.4 4. Validation & Processing

- Files with >1 PII hits → quarantine #todo: tune the threshold
- Malformed JSON → quarantine
- Valid files → append to monthly release file

### 9.3.5 5. Public Release

- **Format**: JSONL files partitioned by month
- **Location**: HuggingFace `publicai/shared-chats`
- **Access**: Gated (request required)
- **License**: As specified by contributor
- **Permanence**: Cannot be retracted

## 9.4 User Rights & Controls

### 9.4.1 What You Control

- **Opt-in toggle**: Must explicitly enable sharing
- **License choice**: CC0, CC-BY, or CC-BY-SA
- **AI preferences**: IETF/CC signals for downstream use
- **Attribution**: Username, anonymous, or pseudonym
- **Per-chat consent**: Confirm each contribution

### 9.4.2 What You Cannot Control

- **Published data**: Once released, cannot be deleted
- **Downstream use**: Subject to license terms, enforcement may vary!
- **Provider behavior**: When chats sent to AI providers

### 9.4.3 Deletion Requests

- **Future releases**: Can exclude your data going forward
- **Past releases**: Generally permanent and irrevocable
- **Contact**: privacy@publicai.company for requests

## 9.5 Contributor Identifiers

### 9.5.1 Contributor Hash

- **Format**: 16-character hex string
- **Generation**: `SHA256(salt + "openwebui:" + user_id)[:16]`
- **Purpose**: Group anonymous contributions
- **Not stored**: Raw user_id never saved

### 9.5.2 Attribution Options

1. **Use username**: Your OpenWebUI username
2. **Anonymous**: No attribution
3. **Custom pseudonym**: Your chosen name

## 9.6 Security Measures

### 9.6.1 In Transit

- HTTPS/TLS for all connections
- API tokens for HuggingFace uploads

### 9.6.2 At Rest

- HuggingFace repository access controls
- No additional encryption on public datasets

### 9.6.3 Access Controls

- OpenWebUI: Account-based access
- HuggingFace: Gated dataset with terms acceptance
- Processing scripts: Maintainer access only

## 9.7 Third-Party Services

### 9.7.1 HuggingFace

- **Purpose**: Dataset hosting and distribution
- **Data sent**: Processed contributions (post-PII removal)
- **Their terms**: https://huggingface.co/terms-of-service

### 9.7.2 AI Providers (via OpenWebUI)

- **When**: During normal chat usage
- **What**: Prompts and system messages
- **Control**: See provider directory in Settings

## 9.8 Compliance Notes

### 9.8.1 GDPR Considerations

- **Legal basis**: Consent (explicit opt-in)
- **Data minimization**: Only necessary fields collected
- **Purpose limitation**: Only for declared research/evaluation
- **Rights**: Access, rectification, erasure (with limitations)

### 9.8.2 Limitations on Deletion

Per our Terms of Service: - Contributions are permanent once published - We can exclude from future releases - We cannot retract from existing public copies - This is clearly disclosed before contribution

## 9.9 Contact Information

- Privacy requests: privacy@publicai.company

# References

Ardila, Rosana, Megan Branson, Kelly Davis, Michael Henretty, Michael Kohler, Josh Meyer, Reuben Morais, Lindsay Saunders, Francis M Tyers, and Gregor Weber. 2019. "Common Voice: A Massively-Multilingual Speech Corpus." *arXiv Preprint arXiv:1912.06670*.

Barocas, Solon, and Andrew D. Selbst. 2016. "Big Data's Disparate Impact." *California Law Review* 104 (3): 671–732.

Bender, Emily M., Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. "On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?" In *Proceedings of the ACM Conference on Fairness, Accountability, and Transparency (FAccT)*, 610–23.

Blodgett, Su Lin, Solon Barocas, Hal Daumé III, and Hanna Wallach. 2020. "Language (Technology) Is Power: A Critical Survey of 'Bias' in NLP." In *Proceedings of ACL*, 5454–76.

Buolamwini, Joy, and Timnit Gebru. 2018. "Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification." In *Proceedings of the Conference on Fairness, Accountability and Transparency (FAT*)*, 77–91.

Carlini, Nicholas, Matthew Jagielski, Christopher A. Choquette-Choo, Daniel Paleka, Will Pearce, Hyrum Anderson, Andreas Terzis, Kurt Thomas, and Florian Tramèr. 2024. "Poisoning Web-Scale Training Datasets Is Practical." https://arxiv.org/abs/2302.10149.

Carlini, Nicholas, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. 2019. "The Secret Sharer: Measuring Unintended Memorization in Neural Networks." In *Proceedings of USENIX Security Symposium*.

Carlini, Nicholas, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, et al. 2021. "Extracting Training Data from Large Language Models." In *Proceedings of USENIX Security Symposium*.

Crawford, Kate, and Trevor Paglen. 2019. "Excavating AI: The Politics of Images in Machine Learning Training Sets." https://www.excavating.ai/.

Creative Commons. 2023. "Understanding CC Licenses and Generative AI." https://creativecommons.org/2023/08/18/understanding-cc-licenses-and-generative-ai/.

European Union. 2016. "General Data Protection Regulation (EU) 2016/679." https://eur-lex.europa.eu/eli/reg/2016/679/oj.

———. 2024. "Artificial Intelligence Act." https://eur-lex.europa.eu/.

Federal Trade Commission. 2013. "Children's Online Privacy Protection Rule (COPPA) — 16 CFR Part 312." https://www.ftc.gov/legal-library/browse/rules/childrens-online-privacy-protection-rule-coppa.

Gebru, Timnit, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, and Kate Crawford. 2018. "Datasheets for Datasets." In *arXiv:1803.09010*.

Grother, Patrick, Mei Ngan, and Kayee Hanaoka. 2019. "Face Recognition Vendor Test (FRVT) Part 3: Demographic Effects." NISTIR 8280. NIST. https://doi.org/10.6028/NIST.IR.8280.

Holland, Sarah, Ahmed Hosny, Sarah Newman, Joshua Joseph, and Kasia Chmielinski. 2018. "The Dataset Nutrition Label: A Framework to Drive Higher Data Quality Standards." https://arxiv.org/abs/1805.03677.

Hubinger, Evan, Carson Denison, Jesse Mu, Mike Lambert, Meg Tong, Monte MacDiarmid, Tamera Lanham, et al. 2024. "Sleeper Agents: Training Deceptive LLMs That Persist Through Safety Training." https://arxiv.org/abs/2401.05566.

Illinois General Assembly. 2008. "Biometric Information Privacy Act (BIPA), 740 ILCS 14." https://www.ilga.gov/legislation/ilcs/ilcs3.asp?ActID=3004.

*ISO/IEC 23894:2023 Information Technology—Artificial Intelligence—Risk Management*. 2023. ISO/IEC.

Jackson, Brandon, B Cavello, Flynn Devine, Nick Garcia, Samuel J. Klein, Alex Krasodomski, Joshua Tan, and Eleanor Tursman. 2024. "Public AI: Infrastructure for the Common Good." Public AI Network. https://doi.org/10.5281/zenodo.13914560.

Jo, Emily, and Timnit Gebru. 2020. "Lessons from Archives: Strategies for Collecting Sociocultural Data in Machine Learning." In *Proceedings of FAccT*, 306–16.

Liu, Jiacheng, Taylor Blanton, Yanai Elazar, Sewon Min, YenSung Chen, Arnavi Chheda-Kothary, Huy Tran, et al. 2025. "OLMoTrace: Tracing Language Model Outputs Back to Trillions of Training Tokens." *arXiv Preprint arXiv:2504.07096*.

McCallister, Erika, Tim Grance, and Karen Scarfone. 2010. "Guide to Protecting the Confidentiality of Personally Identifiable Information (PII)." SP 800-122. NIST.

McDonald, Nora, Benjamin Mako Hill, Rachel Greenstadt, and Andrea Forte. 2019. "Privacy, Anonymity, and Perceived Risk in Open Collaboration: A Study of Service Providers." In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 1–12.

Mitchell, Margaret, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. 2019. "Model Cards for Model Reporting." In *Proceedings of the ACM Conference on Fairness, Accountability, and Transparency (FAccT)*, 220–29.

Narayanan, Arvind, and Vitaly Shmatikov. 2008. "Robust de-Anonymization of Large Sparse Datasets." In *Proceedings of the IEEE Symposium on Security and Privacy*, 111–25.

Nissenbaum, Helen. 2004. "Privacy as Contextual Integrity." *Washington Law Review* 79 (1): 119–57.

NIST. 2023. "Artificial Intelligence Risk Management Framework (AI RMF 1.0)." NIST AI 100-1. National Institute of Standards; Technology; https://www.nist.gov/ai.

Obermeyer, Ziad, Brian Powers, Christine Vogeli, and Sendhil Mullainathan. 2019. "Dissecting Racial Bias in an Algorithm Used to Manage the Health of Populations." *Science* 366 (6464): 447–53.

OWASP. 2023. "OWASP Top 10 for Large Language Model Applications." https://owasp.

org/www-project-top-10-for-large-language-model-applications/.

Rakova, Bogdana, Renee Shelby, and Megan Ma. 2023. "Terms-We-Serve-with: Five Dimensions for Anticipating and Repairing Algorithmic Harm." *Big Data & Society* 10 (2): 20539517231211553.

"Rosenbach v. Six Flags Entertainment Corp." 2019. 2019 IL 123186, Supreme Court of Illinois.

Selbst, Andrew D., Danah Boyd, Suresh Venkatasubramanian Friedler, Suresh Venkatasubramanian, and Janet Vertesi. 2019. "Fairness and Abstraction in Sociotechnical Systems." In *Proceedings of the ACM Conference on Fairness, Accountability, and Transparency (FAccT)*, 59–68.

Shelby, Renee, Shalaleh Rismani, Kathryn Henne, AJung Moon, Negar Rostamzadeh, Paul Nicholas, N'Mah Yilla-Akbari, et al. 2023. "Sociotechnical Harms of Algorithmic Systems: Scoping a Taxonomy for Harm Reduction." In *Proceedings of the 2023 AAAI/ACM Conference on AI, Ethics, and Society*, 723–41. AIES '23. New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/3600211.3604673.

Shen, Judy Hanwen, Inioluwa Deborah Raji, and Irene Y Chen. 2024. "The Data Addition Dilemma." *arXiv Preprint arXiv:2408.04154*. https://arxiv.org/abs/2408.04154.

Sweeney, Latanya. 2000. "Simple Demographics Often Identify People Uniquely." *Carnegie Mellon University, Data Privacy Working Paper.*

U.S. Copyright Office. 2024. "Copyright and Artificial Intelligence: Policy Studies and Guidance." https://copyright.gov/ai/.

U.S. Department of Education. 1974. "Family Educational Rights and Privacy Act (FERPA)." https://www2.ed.gov/policy/gen/guid/fpco/ferpa/index.html.

U.S. Department of Health and Human Services. 2000. "HIPAA Privacy Rule — 45 CFR Parts 160 and 164." https://www.hhs.gov/hipaa/for-professionals/privacy/index.html.

Weidinger, Laura, John Mellor, et al. 2021. "Ethical and Social Risks of Harm from Language Models." *arXiv Preprint arXiv:2112.04359.*

# Part III

# Appendices

# 10 Appendix 1: LLM Data Schemas

Here, we describe many variants of LLM data. This will be relevant for when we extend the flywheel to include more types of data, and especially shift towards promoting the sharing (via opt-in flywheels, but also via new market mechanisms) of richer "content data".

- **Open Web / Crawls**
  - **WARC/WAT/WET**
    - *WARC* (container for HTTP request/response records) — spec & overview: IIPC WARC 1.1; Library of Congress format note. (IIPC Community Resources, The Library of Congress)
    - *WAT* (JSON metadata extracted from WARC) and *WET* (plain text extracted from HTML) — Common Crawl guides. (Common Crawl, Common Crawl)
  - **C4 (Colossal Clean Crawled Corpus)** — TFDS catalog & generator code. Fields are essentially clean text segments with basic metadata. (TensorFlow, GitHub)
  - **The Pile** (22-source, mixed corpus) — paper & HTML view. (arXiv, ar5iv)
- **Encyclopedic / Books**
  - **Wikipedia XML dumps** (page/revision XML; SQL tables for links) — Meta-Wiki dump format; Wikipedia database download. (Meta, Wikipedia)
  - **Project Gutenberg**
    - *Books*: plain text/HTML master formats; ePub/MOBI derived. (Project Gutenberg)
    - *Catalog schema*: daily RDF/XML (also CSV) for metadata; offline catalogs. (Project Gutenberg)
- **Scientific / Legal**
  - **arXiv** (Atom/OAI-PMH metadata; bulk & API) — OAI-PMH + API docs; bulk metadata page. (info.arxiv.org, info.arxiv.org, info.arxiv.org)
  - **JATS XML** (journal article tag suite) — NISO standards; NLM JATS site. (niso.org, jats.nlm.nih.gov)

- **Code**
  - **BigCode — The Stack / The Stack v2** (source files + license/provenance metadata; dedup variants) — HF datasets, project docs, arXiv overview. (Hugging Face, Hugging Face, BigCode, arXiv)
- **Forums / Q&A / Social**
  - **Stack Exchange dumps** (XML: Posts, Users, Comments, Votes, etc.) — SE Meta/docs & Data Explorer. (Meta Stack Exchange, data.stackexchange.com)
  - **Reddit**
    - *API JSON* schema — official API docs & help. (Reddit, Reddit Help)
    - *Pushshift* (historical dumps; research dataset) — site & paper. (pushshift.io, arXiv)
- **Instruction / Conversations (Post-training SFT)**
  - **OpenAI-style chat schema** (role-tagged: `system|user|assistant`, plus tool calls) — API reference. (OpenAI Platform)
  - **Alpaca** (JSON prompts/instructions/outputs) — Stanford post & repo; cleaned community set. (crfm.stanford.edu, GitHub, GitHub)
  - **Databricks Dolly-15k** (human-written instruction/response pairs) — repo. (GitHub)
  - **OpenAssistant OASST1** (message-tree conversations with roles) — HF dataset card. (Hugging Face)
- **Preference / Feedback (RLHF & DPO)**
  - **HH-RLHF** (Anthropic helpful/harmless, JSONL pairs: `chosen` vs `rejected`) — dataset repo readme. (GitHub)
  - **DPO format** (prompt + preferred vs dispreferred response) — DPO paper. (arXiv)
- **Multimodal (for VLMs/ASR)**
  - **LAION-5B / Re-LAION-5B** (image–text pairs with CLIP scores; links) — LAION posts. (laion.ai, laion.ai)
  - **Whisper** (weakly-supervised ASR; audio → text pairs) — paper & blog. (arXiv, OpenAI)
  - **HowTo100M** (YouTube instructional video clips + narrations) — project page & paper. (di.ens.fr, arXiv)
- **Math-reasoning (often for post-training/eval)**

  - **GSM8K** (grade-school word problems; JSON) — repo & HF dataset card. (GitHub, Hugging Face)
  - **MATH** (competition problems with step-by-step solutions) — paper & HF. (arXiv, Hugging Face)

- **Common storage containers**
  - **JSON Lines / NDJSON** — jsonlines.org; ndjson spec. (jsonlines.org, GitHub)
  - **TFRecord** — TensorFlow tutorial. (TensorFlow)
  - **Apache Parquet** — project site. (Apache Parquet)

#todo check all refs

# 11 Appendix 2 — Preference Signals for AI Data Use (CC signals + IETF AI Preferences)

#todo: improve the references here to specific lines of IETF draft and the CC Preference Signals FAQ

- **What CC signals are** A Creative Commons framework for *reciprocal* AI reuse: content stewards can allow specific machine uses if certain conditions are met (e.g., credit, contributions, openness). Overview & implementation notes. (homepage, implementation)
- **Four proposed CC signals (v0.1)**
  - **Credit (cc-cr)** — cite the dataset/collection; RAG-style outputs should link back when feasible.
  - **Credit + Direct Contribution (cc-cr-dc)** — proportional financial/in-kind support.
  - **Credit + Ecosystem Contribution (cc-cr-ec)** — contribute to broader commons.
  - **Credit + Open (cc-cr-op)** — release model/code/data to keep the chain open. Source (draft repo & posts). (GitHub, Creative Commons)
- **IETF AI Preferences (aipref) — the transport & vocabulary**
  - **Vocabulary**: a machine-readable set of *categories* (e.g., `ai-use`, `train-genai`) and *preferences* (`y` = grant, `n` = deny) with **exceptions**. Drafts. (datatracker.ietf.org, IETF, IETF AI Preferences Working Group)
  - **Attachment**: how to convey these preferences via **HTTP `Content-Usage` header** and **robots.txt** extensions. Drafts. (datatracker.ietf.org, IETF)
  - **Structured Fields**: uses RFC-standardized HTTP structured field values. (datatracker.ietf.org, datatracker.ietf.org, rfc-editor.org)
  - **Robots Exclusion Protocol** baseline. (datatracker.ietf.org, rfc-editor.org)
- **Putting them together (content-usage expression)**
  - Shape:

```
<category>=<y|n>;exceptions=<cc-signal>
```

Example in **robots.txt** (allow everything, but *AI use denied unless Credit*):

```
User-Agent: *
Content-Usage: ai-use=n;exceptions=cc-cr
Allow: /
```

Example **HTTP header** (deny *gen-AI training* unless *Credit + Ecosystem*):

```
Content-Usage: train-genai=n;exceptions=cc-cr-ec
```

(Syntax and examples from CC & IETF drafts.) (Creative Commons, IETF)

- **Operational notes (for this repo's flywheel)**
  - **Per-record fields** to store: `license` (CC0/CC-BY/CC-BY-SA) and `ai_pref` (IETF aipref value + optional CC signal), plus optional `attribution` handle. (Aligns with CC write-ups & IETF drafts.) (Creative Commons, datatracker.ietf.org)
  - **Placement**:
    - *Location-based* signals via **robots.txt** for site/paths. (datatracker.ietf.org)
    - *Unit-based* signals via **HTTP `Content-Usage`** on dataset files and API responses. (datatracker.ietf.org)
  - **Interoperability expectations**: signals are normative *preferences*; adherence relies on ecosystem norms (similar to robots.txt & CC license culture). (Creative Commons)
- **Context & momentum**
  - CC's 2025 launch posts; IETF WG activity updates (e.g., IPTC note). (Creative Commons, Creative Commons, IPTC)

---

#todo check all refs

# 12 Appendix 3: LLM Policy Docs

This Appendix contains a list of links to various live Terms of Service, Privacy Policy, and related docs for major LLM providers, including both private players like frontier labs and public AI-adjacent actors like AI2, Mozilla,