Third-Party Markets for Cultural Assets for AI Applications

Preliminary Note for Prof. Alexandra Bensamoun
[*Not for Circulation without prior permission*]

07 April, 2025
Ilan Strauss[1]

AI Applications. Assume a marketplace for the content used by AI applications in their ongoing external information retrieval and search requests. As consumers generally desire *recent* (up-to-date), *accurate*, and *high-quality* content, the price and cost of this content is likely to be of ongoing importance if AI applications are to add ongoing value to the user's service.[2]

The commercial arrangements governing text, video, and audio content markets need to be considered individually, on a case-by-case basis, given their varying market features (concentration of ownership rights, the nature of the asset, including how it is accessed and monetized, and more.). **Below we assume a third-party market for content from news and website publishers to be used by an AI application**.

1. Some Market Design Principles

Several principles of market design stand-out as being important for creating a formal commercial marketplace for cultural assets used in AI applications:

- **Clearly define what the asset is / where does value reside for the user.** Perhaps the most difficult thing is to first decide what the valuable asset is – and then how it is monetized (per click, per view, time spent). Baker v. Selden in the U.S. (1879) established that *copyright protects the expression of an idea, not the idea itself*. But in a market where a user employs an LLM application *to both find and then communicate* (i.e. express!) accurate, timely, or incisive ideas and commentary, the LLM may be adept at expressing the idea in a sufficiently unique manner. The value then may instead reside with the idea itself, *not the expression*. This "idea" may be an argument or line of reasoning that the LLM is able to hone on the basis of the content presented to it by the

[1] For more information: https://www.ssrc.org/programs/ai-disclosures-project/. Contacts: istrauss@ssrc.org and tim@oreilly.com. Thank you to Lucky Gunasekara (Miso.ai) for valuable inputs on this note, Andrew Trask (OpenMinded) for a discussion about their work in this area, and Ross Engers (AWS / Twitch) for inputs on protocol and API usage in software architectures. All views are our own.
[2] Note that some platforms, in particular Musical AI (formerly Somms.ai), are undertaking several markets for training, content licensing, and attribution. Platforms for training data payment can be fairly static if remuneration is provided on a once-off basis, acquiring the asset totally (see *Defined.ai*).

publisher.[3] Similarly, if quick access to facts is valued by the user, then a publisher providing an LLM with exactly this is providing a monetizable asset – even though facts are also not copyrightable under U.S. law.[4] Facts could be 'breaking news' released by a publisher based on its 'on the ground' coverage.

Another approach is to eschew measurement entirely, even in dynamic AI application outputs, and agree on value being co-created between an LLM application and publishers in a fixed ratio. For example, *ProRata.ai* enables generative AIs outputs to properly attribute contributing content to certain publishers and share revenues on a per-use basis *50-50*. No dynamic revenue split is calculated.

In any event, an LLM application summarizing information from a publisher in a limited manner, and then providing a monetizable click-through link to the user, is for now the immediate basis for a publisher's monetization on a query-by-query basis. This is the case for Google's search result page, which contains an AI-generated summary result on top, along with links to click on for more information. Google's incentive to provide links for clicking through to in its Search results means that it has some incentive to enforce limits on the length of its AI-generated summaries taken from publishers' sites.

Longer AI-generated summaries of publisher material may facilitate a very different model of remuneration, based on user engagement (time spent) with the material, say. Either way, some agreement would be needed on how the information is presented to the user by the AI Application, in conjunction with how access to inputs from third-parties is provided and ultimately remunerated.

- **Punish illegal alternatives**. Reducing the incentive for free alternatives to arise is vital, otherwise commercial markets that remunerate content creators in-line with the value they provide are unlikely to arise very easily. Hefty fines for AI applications found to violate robots.txt protocol – as most AI companies now do – is one way to go. This violation means that bots are effectively taking content from sites without the publishers' permission, including going behind paywalls it appears. (See attached document from *Miso.ai* with evidence on this with respect to French publishers). To enforce this one, contact Amazon AWS, Microsoft Azure, and Google Cloud and notify them of the large fines they will receive if they do not enforce their own terms of service on this issue.[5] The legal basis for such an initiative would first need to be established of course. Google Search should also be compelled to permit publishers to opt out from its AI-generated summary (at the top of its search result), but still allow for the publisher to be included in its search results page (being indexed and crawled for such a purpose only).

- **Countervailing (legal) bargaining power**. As in a standard monopsony bargaining model, markets in which the sellers of an asset are well organized are more likely to form

[3] Remunerating the publisher for this idea, without relying on a user's clicks for such payment, requires further serious consideration and cannot be ignored.
[4] *Feist Publications, Inc., v. Rural Telephone Service Co.*, 499 U.S. 340 (1991).
[5] See: https://www.wired.com/story/aws-perplexity-bot-scraping-investigation/

and be remunerated fairly. That's arguably why music has already established licensing platforms, since rights holders are more concentrated.

- **Common standards.** The market would need to agree upon conventions and rules, including ways of measuring key quantities and disclosures to undertake. A consortium to facilitate an industry adopting a particular set of standards can be helpful in this respect. *Miso.ai* is doing work on this with publishers. The challenge is to *not be too far behind the market*, such that the standard is not relevant to the products that users actually end up using; but also to be a bit *ahead of the market*, so that a single dominant entity does not decide on the standards unilaterally, once they have established themselves as the dominant force in the market.

- **Competition law considerations**. If pricing and content information from a range of publishers is combined into a single centralized platform or exchange, this could risk creating algorithmic collusion, if used by a single entity with data advantages. Permission will also be needed for publishers to combine and negotiate jointly on certain issues, after which demand and supply signals would dictate price discovery.

- **Transparency and platform independence**. Lessons from existing online platforms and algorithmically facilitated markets are that opaqueness allows for participants to be exploited by the strongest party. Risk from vertical integration on fair market outcomes is substantial. The monetizable measure in publishers' content online is likely to be highly sensitive to measurement, as with monetizing impressions in ads data – making transparent and independent measurement doubly vital.

It may still be too early to know who content rights holders should be negotiating with. Spotify arose as an application in the app store and online. It took Apple Music months to catch-up despite being a larger player in digital markets.

2. Protocols and Software Market Considerations

Protocols are low-level standards (as rules or conventions) to allow for interoperability, so that multiple things can communicate with one another, e.g., HTTP enables communication between clients (such as web browsers) and servers, defining how data, such as web pages, is exchanged. Services can talk to one another if they use the same protocol. **APIs** (application programming interface) are for the application-level. They may use one or more protocols but define the structure of the data that you are querying and the response you will get back. Any online marketplace will likely make use of both of these things, especially if new rules are needed to structure how communication takes place.

The exact protocol used, as the eventual convention, wins out based on usage and adoption in the market. So long as a dominant player in the market cannot shape this outcome in its own interests, by refusing interoperability or having a closed API, then the market outcome is likely to

be welfare enhancing.

Typically, new protocols are not created. But given that LLMs are a new unit on the internet, it makes sense that new protocols (as standards) are now arising. For example, Model Protocol (**MCP**) allows for LLMs to access external tools and systems in a standardized way for information (prompts, tools, resources). It creates a standard API for AI applications. For example, OpenMined[6] is working with the National Association of Broadcasters (NAB) to develop a protocol implemented inside a new type of web server, such that the website can allow for AIs to talk to AIs.

Stylized examples with MCP. By way of a stylized example applying MCP to a marketplace for cultural assets: assume each publisher operates its own MCP server that interfaces with their respective content repositories. These servers would expose standardized endpoints, allowing an LLM application to query and retrieve information from the publisher. Here, the LLM functions as the MCP client, capable of connecting to multiple MCP servers (i.e., multiple publishers) to access the required information. The media platform's API would be wrapped or extended by an MCP server. This MCP server would translate standardized MCP requests from the LLM into appropriate API calls to the media platform and then relay the responses back to the LLM in a standardized format.

Data layer. *One issue with this is that it is one-sided*. The publisher will want to receive information back from the LLM application on how the information they shared with it was used and monetized. This highlights the importance of databases and data layers to log metrics such as bids, sales transactions, pricing details, impressions, and clicks.

Agent-to-agent interactions. Each publisher could also use an LLM that negotiates with the user's LLM application. As before, the user's LLM application (MCP client) sends a request to the publisher's MCP server. But now, the publisher's MCP server forwards this request to their own (publisher's) internal LLM agent. The publisher's LLM agent processes the request, which may involve negotiating terms, determining pricing, or curating content dynamically. After processing, the publisher's agent sends the appropriate response back through the MCP server to the user's LLM. *A protocol for agent-to-agent communication would be needed here*.

An Exchange. *The problem is ultimately one of many publishers*. *What then*? This is why economists promote exchanges to facilitate transactions between millions of buyers and sellers *of assets of a particular type – say tokens, query responses to keywords or subjects*, or some other type of digital space-content relationship, for a given informational context. The informational context could define the scope of the exchange; just as you have different exchanges in financial markets for equities, bonds, and commodities (not all of which are fully digitized).

Without going into any technical detail, the exchange could be a middle layer between publishers and LLMs, rather than requiring every publisher to host its own MCP (Model Context

[6] Andrew Trask.

Protocol) server. This would have the benefit of consistent protocol usage, standard data formats, security processes, centralized usage metrics, and micropayment settlement. LLMs would integrate with only a few well-documented APIs: such as an ‘exchange’ API, supply API, demand API, and settlement API (to ensure settlement information is provided) – rather than dealing with hundreds or thousands of different publisher endpoints. The exchange itself would ‘speak’ MCP though, so from the LLM’s point of view, it would simply be connecting to a standard MCP endpoint. Because an exchange would orchestrate requests/responses from many different publishers, it could, in theory, handle aggregated billing, micropayments, and usage tracking.

Publishers would register their datasets or content feeds with the exchange, along with associated meta-data. LLMs could query or subscribe to particular streams or subsets of data from the exchange, simplifying discovery. An exchange could benefit from multiple concurrent pricing models too, involving some combination of a fixed monthly payment to get some content from a subset of publishers, micropayments for snippets, and payment per requests for articles retrieved. If content is monetizable per impression, click, or simply per retrieval, would need to be decided upon. In practice, one-sided databases are created every day through the scraping (crawling and often indexing) of websites by AI applications and Search engines, making the prevention of unauthorized access by AI bots to publishers' sites vital to establishing commercial exchanges.

Things to Consider. An end-to-end modular and evolving solution for creating a dynamic third-party content market for AI would need protocols – as part of a transparent and open software architecture – covering every market layer, including data provenance tracking, licensing terms, usage auditing, hallucination monitoring, royalty calculation and end-user engagement. True interoperability requires a well-defined set of mandatory fields (e.g., data schema, licensing, usage constraints, price info, model performance metrics) and a governance process that carefully reviews and standardizes new fields.

3. Designing Protocols for Markets: Ads case study

Protocols create rules. They might encode a certain way of doing business through how information is exchanged, but remain captive to the business model decided upon by market participants. They create interoperable communications that can prevent vendor lock-in and foster innovation on top *of that layer*. But this can also simply push market concentration further up the stack (e.g. applications in the world wide web). Nonetheless, the protocols and design of the digital advertising stack may have useful lessons for what the software architecture of effective third-party AI content marketplaces should – or should not – look like. *It highlights the importance of having end-to-end visibility and coverage*, including:

- *Protocols for each core segment of the market* (buying, selling, exchanging, and settlement). Otherwise exploitation and market power can easily be abused from outside of the protocol – or even through it.

- *Transparent rules and records for how the (ad) server chooses the winner, how final settlement is reported, including fees, etc, is vital*. This rests on having comprehensive previous data layers, including defining how data is stored, labeled, updated, licensed, and who can see usage metrics; after which an intermediary or "layer" can surface a transparent breakdown of costs, similar to an "itemized receipt." For real-time "bidding" for inference, final costs should be clearly defined.

- *Protocols that evolve to include all relevant information*. If the protocol only defines a few "core" fields and everything else sits in "extension" objects, you risk fragmentation, with major vendors creating closed, proprietary add-ons.

- *Trusted third-party, or mechanism*, is needed to ensure logs are correct, usage rights are respected, fees are accurate, and large participants are not exploiting information asymmetries – protections various financial markets try to have.

- *Robust privacy controls* at the protocol level – e.g., specifying methods for anonymizing or pseudonymizing data, encoding user consents, or ensuring differential privacy.

- *Build in cryptographic checks or distributed ledger technology* to track who contributed data, who processed it, and how it's used downstream (i.e., verifying model lineage, verifying training/inference boundaries). A protocol might incorporate some form of data lineage, if not through cryptographic signatures to track data sources and transformations, then through a unique ID say. This prevents a "black box" scenario where major players claim to use data in a certain way but actually do more with the data behind the scenes.

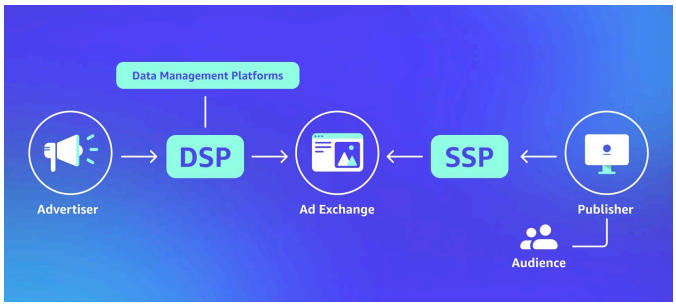
In the final analysis, a purely technical 'spec' (i.e. protocol) that doesn't account for a standardized way of disclosing and reconciling fees (and letting all parties see them) risks repeating ad tech's black box pricing problem. This may speak to the importance of additional data layers, or databases, to ensure transactions and performance metrics are transparent and recorded.

Case Study: The online advertising 'stack'. OpenRTB (Open Real-Time Bidding) is the de facto standard protocol for real-time bidding transactions in programmatic advertising online. It standardizes the *message format* in the *bid request → bid response* workflow.[7] By standardizing the format of bid requests/responses (e.g., user ID signals, device info, creative

[7] Most major demand-side platforms (DSPs), supply-side platforms (SSPs), and ad exchanges implement some version of OpenRTB. It's an open specification: any platform can implement it,creating easy interoperability between DSPs, SSPs, and exchanges.

requirements) demand-side platforms (DSPs) can bid in real time and supply-side platforms / exchanges can evaluate those bids (Figure 1).

Figure 1. The demand-side platform (DSP) allows advertisers to buy digital ad inventory across multiple exchanges in real time



Source: Amazon.

The protocol, however, does not define how publishers set up their inventory, how auctions *should* be conducted internally, or how fees and take rates should be disclosed. As a result the digital ads market has become opaque and exploitative.

A company such as Google owning the entire stack[8] integrates custom logic at each step – server-to-server calls, dynamic floor pricing, or historical data advantage – without violating OpenRTB's fundamental format requirements. OpenRTB covers just the real-time bidding handshake, while other key pieces (ad server logic, auctions, data management, analytics) remain walled off by Google's proprietary tech.

Business rules are not integrated into the protocol, and are instead left to proprietary, and platform-specific, setups. Nothing in the OpenRTB spec, for example, mandates that an exchange or intermediary must provide a transparent fee "receipt". OpenRTB does not control or regulate how final clearing prices are calculated. It merely passes bid requests and bid responses. Custom business rules or additional "wrappers" that mask certain auction dynamics or fees can be freely added as extensions to the protocol (through the '`ext`' object). This flexibility can lead to bidding logic or data usage that is opaque to outside participants.

Could protocols at each point in the process (end-to-end) address some of these concerns? Perhaps, *but probably only to the extent that the protocols standardize opaque business practices and formalize the return and exchange of various pieces of operating information from the business model layers*.[9]

[8] Google owns: (1) the dominant publisher ad server (originally DoubleClick for Publishers), (2) a major ad exchange (AdX), (3) a leading demand-side platform (DV360), and (4) major analytics and measurement tools. Because all these pieces integrate tightly, publishers and advertisers often find it more efficient to stay wholly within Google's ecosystem. This vertical integration allows them to set internal policies and fees that may not be fully transparent to external parties, regardless of the underlying OpenRTB transactions.

[9] For example:

- *Inventory and Ad Serving*. A standardized way for publishers to define and manage their inventory, *including details of each impression* (size, format, placement, user privacy settings, etc.) Clear rules for how an ad server hands off impressions to an exchange or multiple exchanges (header bidding, waterfalls, or unified auctions).

Related Readings

Kint, Jason. *Testimony before the U.S. Senate Committee on the Judiciary, Subcommittee on Antitrust, Competition Policy, and Consumer Rights, Hearing on "Big Fixes for Big Tech,"* April 1, 2025. Here.

O'Reilly, Tim. "Disclosures. I do not think that word means what you think it means. Disclosures are the language of networks and of markets". *Asimov Addendum*. Substack. March 27, 2025. Here.

O'Reilly, Tim. How to Fix "AI's Original Sin". 2024. *O'Reilly Radar*. Here.

Rosenblat, Sruly, Tim O'Reilly, and Ilan Strauss. "Beyond Public Access in LLM Pre-Training Data:Non-public book content in OpenAI's Models." *SSRC AI Disclosures Project Working Paper Series 2025-04*. April 2025. Here.

Tan, Garry. *Testimony before the U.S. Senate Committee on the Judiciary, Subcommittee on Antitrust, Competition Policy, and Consumer Rights, Hearing on "Big Fixes for Big Tech,"* April 1, 2025, Here.

- *Transaction & Auction Mechanics*. Clear definitions of how auctions are conducted (e.g., first-price vs. second-price, tie-break logic, floors, real-time dynamic floors) Built-in requirements for fee transparency – i.e., each intermediary discloses its markup or take rate in a standardized format that all parties can see.
- *Participant Identity & Data Governanc*e. Standard ways for all parties (publishers, SSPs, DSPs, data providers) to identify themselves and specify the data they are adding to the bid request or response. Unified, enforceable privacy & consent frameworks (for example, expansions on the IAB TCF but truly enforced end-to-end), so user data handling is consistent and auditable across all platforms.
- *Reporting & Auditing*. A set of standardized fields (or receipts) that detail the final settlement price, all fees taken, and which bidder won, so that buyers, publishers, and regulators can audit the transaction. Requirements for storing or logging transactions in a commonly accessible format, perhaps even a distributed ledger or tamper-proof log that third parties (auditors, regulators) can inspect.