
User-Side Realization as a Catalyst, and the Potential of LLM Agents as Powerful Catalysts

Ryoma Sato
National Institute of Informatics

rsato@nii.ac.jp

Data Processing Club Technical Report DPC-TR-2026-002
September 30, 2026

Abstract

Web platforms have historically kept users firmly locked in, making it difficult for better services to succeed even when they emerge. We argue that the recently introduced paradigm of user-side realization can serve as a catalyst that smooths the energy landscape of platform dynamics, creating a healthier environment. In particular, we argue that LLM agents enable powerful forms of user-side realization and can substantially alter these dynamics. Of course, this change is not one-sided: as the environment changes, platform operators will also change their strategies and may devise new ways to lock users in. Nevertheless, we argue that user-side realization can fundamentally change the game.

1 Introduction

The web is home to many large platforms, including video-sharing and social networking services. These platforms keep users firmly locked in through factors such as the data users have accumulated on them and the network effects arising from the presence of large numbers of other users. This makes it difficult for better services to succeed even when they emerge. Although it is difficult to define a “good service,” even if Bluesky were a “better” service than X (Twitter), or became a clearly “better” service through further improvements, it might still struggle to attract more users than X (Twitter). Users are accustomed to X, there are many interesting users on X, and they have already made all kinds of posts there and do not want to lose their existing followers. Thus, even if they are somewhat dissatisfied with X, relatively few users move to Bluesky. However, this environment may be stifling innovation by eroding incentives to develop new social networking services or improve existing ones. X, too, may become complacent despite knowing that its users are dissatisfied, or, in worse cases, take advantage of their predicament and begin prioritizing its own interests over theirs. Such an environment is unhealthy. Measures such as antitrust laws and the right to data portability under the EU General Data Protection Regulation have been introduced to address this situation. Yet, as the current state of the web shows, this environment remains unchanged.

This environment can be viewed as an energy landscape, as shown in Figure 1. Lower-energy points represent better services. Even if a service with lower energy emerges, an energy barrier separates it from the current service. A very small number of exceptionally proactive users may overcome this barrier, but switching services is difficult for many users. If the old service becomes embroiled in a scandal or becomes exceptionally difficult to use, its energy rises, and users may migrate en masse to a new, convenient service. However, such an exodus requires exceptional circumstances. Platforms are aware of this and, consciously or unconsciously, maintain high energy levels and profit from them while staying below the point at which an exodus would occur.

User-side realization [7, 12] is a promising paradigm for addressing this problem. Pretender [14] is a method for transferring preference data accumulated on an old platform to a new one when switching platforms, even if the two platforms do not support data import or export, or if no compatible contents exist. This reduces the effort required to switch platforms. User-side recommender systems [7, 9, 11, 13, 16] and user-side feed filtering and reranking [1, 2, 3, 4, 15, 17] allow users to build recommender systems that have traditionally

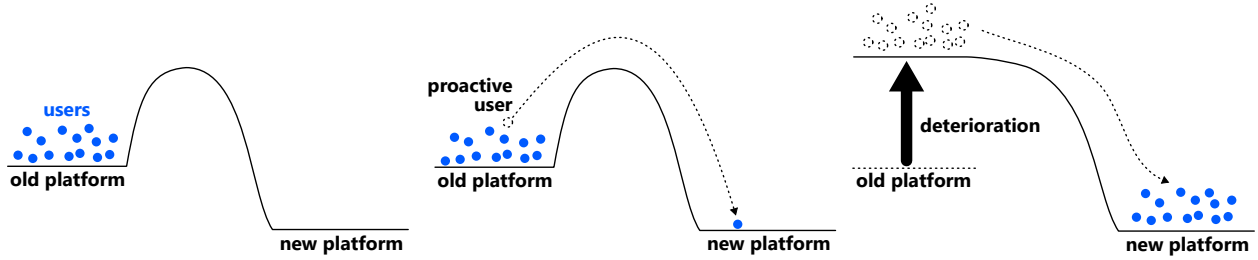


Figure 1: The energy landscape of platforms. Lower energy corresponds to a better service. Under normal conditions, even when a better service emerges, a high barrier prevents migration between platforms, except for some proactive users. Substantial deterioration of a platform may trigger large-scale migration. In practice, however, platforms take care to prevent this, so such changes rarely occur.

been controlled by platforms. These approaches are important because they enable users themselves to address the negative aspects of recommender systems, such as filter bubbles and dopamine hacking, that have been left unchecked to keep users locked into platforms, and to regain their rationality and agency.

We propose viewing such forms of user-side realization as catalysts in the energy landscape (Figure 2). User-side realization technologies lower the energy barriers that must be overcome to switch services, making it easier to move to a “good service.” This creates healthy competition and, above all, leaves platforms less room to arrogantly prioritize their own interests over those of their users. We therefore expect it to make it easier for platforms that benefit users to emerge.

In particular, user-side realization using LLMs has been actively studied in recent years [13, 16]. We argue that these technologies are powerful and have the potential to fundamentally reshape the energy landscape.

2 LLM Agents Can Be Powerful Catalysts

User-side realization can be broadly divided into wrapper and reverse methods [12]. Wrapper methods provide new services or features by communicating appropriately with the original platform while leaving it intact. Reverse methods provide new services or features by reverse-engineering the platform and rebuilding the platform itself, or part of it, from scratch.

LLM agents are powerful tools for both wrapper and reverse methods. Traditional wrapper methods have relied on APIs or reading the DOM. They have achieved nontrivial results, such as combining APIs to provide features that the APIs do not natively offer [6, 8, 10]. However, wrapper methods cannot be implemented if there are no underlying APIs or sources to read from. Wrappers that read the DOM can also stop working before one realizes it, as services change over time. LLM agents have strong natural language understanding and can interact with UIs in the same way as humans. Thus, even without an official API, they can implement wrappers by extracting information through the same UI that users use. They can also handle ambiguity in interactions, allowing them to recognize changes to a service and continue functioning.

LLM agents particularly shine in reverse methods. A weakness of reverse methods has been their high implementation cost, but LLM agents dramatically alleviate this problem. For example, Tenzen Studio created a Photoshop clone in two weeks by investigating Adobe Photoshop’s features and implementing them with GPT-6 Astra¹. As this example illustrates, advances in LLM coding agents are dramatically reducing the cost of applying reverse methods, even to large-scale software.

3 LLM Agents Do Not Solve Everything

Reverse methods using LLM agents are not a panacea. Although they can reduce coding costs, there are limits to how much they can reduce the ongoing costs of serving a system. This limits the applicability of reverse methods.

¹<https://tenzen.studio/photon>

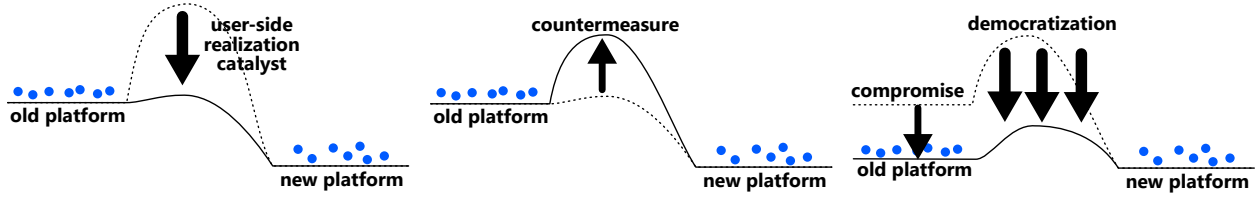


Figure 2: User-side realization as a catalyst. User-side realization lowers the barriers to switching services. Platforms may resist this movement by devising new forms of lock-in. However, as LLM agents democratize user-side realization and strengthen the ability to remove these barriers, resistance ceases to be economically viable, and effort shifts from erecting barriers to lowering energy levels. This creates the conditions for healthy competition, makes room for services that benefit users to emerge, and transforms old services into ones that put users first.

Consider Google’s web search as an extreme example. It might be possible to obtain source code equivalent to Google’s web search using an LLM coding agent, but the resulting system would not work because one would be unable to serve its enormous web index. Google’s web search is economically viable only when used by billions of people. Creating a clone for a single user would make no economic sense, and running it would be impossible in the first place.

Although Google is an extreme example, various platforms fall between these extremes in practice. Moreover, LLM agents now make it possible to adopt algorithms that are more complex but more efficient, or to move to hosting services or hardware that are less convenient but cheaper. Thus, the ongoing costs of serving systems are also decreasing in practice. A service that was once economically viable only with hundreds of users might therefore become viable as a clone created by a single user. The current situation is that LLM agents are shifting this threshold, while many services still remain above it.

4 Interactions with Platforms

User-side realization lowers switching costs by acting as a catalyst, but this movement is not one-sided. Platforms will not simply stand by; they will likely respond with further countermeasures and forms of lock-in (Figure 2, middle). Measures such as CAPTCHAs may no longer work against misbehaving AI users, but other measures are emerging: rejecting requests at the edge, as in Cloudflare’s Bot Management; charging crawlers, as in Pay Per Crawl; or directing misbehaving crawlers into honeypots to exhaust their resources, as in AI Labyrinth. Other measures, such as anubis², use proof of work to reduce incentives for crawling by consuming computational resources, even if a bot disguises itself and cleverly overcomes the countermeasures. Such measures may cancel out the catalytic effect of user-side realization.

At this stage, users and platforms enter a tug-of-war over information extraction. Platforms seek to extract various kinds of information from users, including profile information, posts, and browsing histories, and turn it into power for the platform. Users, meanwhile, extract information from platforms to choose services more freely and obtain the features they want. This may go beyond retrieving information that users once entrusted to the platform. For example, when building a user-side recommender system, users may obtain latent item data and even estimates of some of the information that other users have provided to the platform. They can do so through item information, information that the platform presents item A and item B as related, and even the underlying information that other users tend to consume item A and item B together [9]. Alternatively, as with the Photoshop clone, they may extract something equivalent to source code that the platform itself created. Because information has value, users and platforms can be expected to try to extract as much information as possible from each other through a process of using and being used, eventually reaching a kind of equilibrium. The central question is where this equilibrium lies.

²<https://github.com/TecharoHQ/anubis>

5 Further Resistance to Platforms

To further resist platforms’ pushback against LLM agents, Marro and Torr [5] take the position that interoperability between platforms and governance through legislation are necessary. Our position differs. If interoperability is necessary, then we are ultimately in the same situation as before the advent of LLM agents. If platforms were willing to support interoperability, universal interoperability should have been achievable through their cooperation in the first place, without any need for user-side realization or LLM agents. In reality, platform operators are reluctant to support interoperability, and even as legislation advances, they find other ways to lock users in. In some respects, fragmentation and lock-in have even intensified compared with when antitrust laws and the EU General Data Protection Regulation were first enacted. Of course, some advocacy for interoperability and some legislation are necessary, but we do not believe they will lead to a fundamental solution. Even with LLM agents added to the equation, the same old game of cat and mouse will likely continue.

We believe that users should take the initiative and actively use user-side realization and LLM agents to resist lock-in, rather than entrusting sovereignty to platforms or governments, as approaches based on interoperability or legislation do. Of course, we do not claim that user-side realization will solve everything. As discussed in Section 4, even when a catalyst is introduced, platform operators will likely raise barriers in response. Nevertheless, the equilibrium can be shifted. Countermeasures are not free for platform operators either. The more excessive these measures become, the more they reduce platforms’ profits, so operators would presumably prefer not to take them, and there are limits to how far they can raise the barriers. Being forced to respond changes the break-even point. Platform operators cannot keep responding indefinitely (Figure 2, right). Above all, the strength of user-side realization lies in the sheer number of users: they outnumber platform employees by orders of magnitude. There is strength in numbers. Even if each individual has little power, a guerrilla-style struggle would force platforms to compromise because they cannot respond to every action. LLM agents and LLM-based vibe coding offer substantial power in this respect. Previously, only a handful of users could write programs and carry out user-side realization. With vibe coding, virtually all users can now participate in creating programs for user-side realization. In other words, it has been democratized. This will increase the amount of catalyst introduced by orders of magnitude, and platforms will likely be unable to keep up. Now that users have gained democratic sovereignty, control of the game is about to change hands.

6 Conclusion

We have argued that the recently introduced paradigm of user-side realization can serve as a catalyst that smooths the energy landscape of platform dynamics, creating a healthier environment. User-side realization lowers the barriers to switching services. Platforms may resist this movement by devising new forms of lock-in. However, as LLM agents democratize user-side realization and strengthen the ability to remove these barriers, resistance ceases to be economically viable, and effort shifts from erecting barriers to lowering energy levels, creating the conditions for healthy competition. Through the democratization of user-side realization by LLM agents, users will take control of the game and be able to enjoy services that put them first.

References

- [1] F. Jahanbakhsh, D. Zhao, T. Piccardi, Z. Robertson, Z. Epstein, S. Koyejo, and M. S. Bernstein. Value alignment of social media ranking algorithms. In *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems, CHI*, pages 1401:1–1401:26, 2026. doi: 10.1145/3772318.3791281. URL <https://doi.org/10.1145/3772318.3791281>.
- [2] A. Kolluri, R. Su, F. Jahanbakhsh, D. Zhao, T. Piccardi, and M. S. Bernstein. Alexandria: A library of pluralistic values for realtime re-ranking of social media feeds. *Proceedings of the International AAAI Conference on Web and Social Media*, 20(1):1309–1330, 2026. doi: 10.1609/icwsm.v20i1.42697. URL <https://doi.org/10.1609/icwsm.v20i1.42697>.

-
- [3] S. Lazar, L. Thorburn, T. Jin, and L. Belli. The moral case for using language model agents for recommendation. *Inquiry*, pages 1–25, 2025. doi: 10.1080/0020174x.2025.2515579. URL <https://doi.org/10.1080/0020174x.2025.2515579>.
 - [4] O. E. Malki, M. A. L. Quéré, A. Monroy-Hernández, and M. H. Ribeiro. Bonsai: Intentional and personalized social media feeds. In *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems, CHI*, pages 1395:1–1395:24, 2026. doi: 10.1145/3772318.3791855. URL <https://doi.org/10.1145/3772318.3791855>.
 - [5] S. Marro and P. Torr. LLM agents are the antidote to walled gardens. In *Proceedings of the 43rd International Conference on Machine Learning, ICML Position Paper Track*, 2026. doi: 10.48550/arXiv.2506.23978. URL <https://arxiv.org/abs/2506.23978>.
 - [6] R. Sato. CLEAR: A fully user-side image search system. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management, CIKM*, pages 4970–4974, 2022. doi: 10.1145/3511808.3557172. URL <https://doi.org/10.1145/3511808.3557172>.
 - [7] R. Sato. Private recommender systems: How can users build their own fair recommender systems without log data? In *Proceedings of the 2022 SIAM International Conference on Data Mining, SDM*, pages 549–557, 2022. doi: 10.1137/1.9781611977172.62. URL <https://doi.org/10.1137/1.9781611977172.62>.
 - [8] R. Sato. Retrieving black-box optimal images from external databases. In *Proceedings of the 15th ACM International Conference on Web Search and Data Mining, WSDM*, pages 879–887, 2022. doi: 10.1145/3488560.3498462. URL <https://doi.org/10.1145/3488560.3498462>.
 - [9] R. Sato. Towards principled user-side recommender systems. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management, CIKM*, pages 1757–1766, 2022. doi: 10.1145/3511808.3557476. URL <https://doi.org/10.1145/3511808.3557476>.
 - [10] R. Sato. Making translators privacy-aware on the user’s side. *Transactions on Machine Learning Research*, 2024. URL <https://arxiv.org/abs/2312.04068>.
 - [11] R. Sato. Overhead-free user-side recommender systems. *arXiv preprint arXiv:2411.07589*, 2024. doi: 10.48550/arXiv.2411.07589. URL <https://arxiv.org/abs/2411.07589>.
 - [12] R. Sato. *User-Side Realization*. PhD thesis, Kyoto University, 2024. URL <https://arxiv.org/abs/2403.15757>.
 - [13] R. Sato. Agentrecommender: Llm agents enable customizable recommender systems on the user side. *arXiv preprint arXiv:2609.31166*, 2024. doi: 10.48550/arXiv.2609.31166. URL <https://arxiv.org/abs/2609.31166>.
 - [14] R. Sato. Solving the cold start problem on one’s own as an end user via preference transfer. *Transactions on Machine Learning Research, TMLR*, 2025. doi: 10.48550/arxiv.2502.12398. URL <https://arxiv.org/abs/2502.12398>.
 - [15] J. Stray, I. Baker, G. Beknazar-Yuzbashev, C. Budak, J. Kamin, K. Rutherford, M. Stalinski, T. Acosta, C. Bailey, M. S. Bernstein, M. Brandt, A. Bruckman, A. Chhabra, S. De, K. Duskin, S. Fish, B. Goldberg, A. Guess, D. Hadfield-Menell, M. Haroon, S. Hossain, M. Inzlicht, G. Jain, Y. Jiang, A. P. Landry, Y. Lelkes, H. Lu, P. Mason, J. McCoy, S. Milli, P. Resnick, E. Saltz, M. Saveski, L. Schirch, M. Spohn, S. Srinivasan, A. Tatore, L. Thorburn, J. A. Tucker, R. Willer, M. Wojcieszak, M. Wüthrich, and S. Zheng. The prosocial ranking challenge: Reducing polarization on social media without sacrificing engagement. *arXiv preprint arXiv:2603.19626*, 2026. doi: 10.48550/arXiv.2603.19626. URL <https://arxiv.org/abs/2603.19626>.
 - [16] W. Xu, Y. Shi, Z. Liang, X. Ning, K. Mei, K. Wang, X. Zhu, M. Xu, and Y. Zhang. iAgent: LLM agent as a shield between user and recommender systems. In *Findings of the Association for Computational Linguistics: ACL 2025, Findings of ACL*, pages 18056–18084, 2025. doi: 10.18653/v1/2025.findings-acl.928. URL <https://doi.org/10.18653/v1/2025.findings-acl.928>.
 - [17] H. Yuan, P. He, D. Zhang, J. Liang, and J. Zhu. Position: Recommender systems should move beyond platform-centric ranking toward personal agent-mediated recommendation. *arXiv preprint arXiv:2609.11942*, 2026. doi: 10.48550/arXiv.2609.11942. URL <https://arxiv.org/abs/2609.11942>.