Decentralized Generative AI Framework with Solid

Ahmad Cahyono Adi[1], Dhea Anggita[1] and Kabul Kurniawan[1,2,*]

[1]*Department of Computer Science and Electronics, Universitas Gadjah Mada, Yogyakarta, Indonesia*
[2]*Center for Cryptography and Cybersecurity Research, Universitas Gadjah Mada, Yogyakarta, Indonesia*

Abstract

Generative AI (GenAI) application becomes increasingly an integral part of daily life, accelerating tasks and enhancing human productivity. As these Large Language Models (LLMs) platforms grow more personalized by learning from increasingly large volume of user (personal) data, it raises significant privacy, trust, and data ownership concerns. Current LLMs applications typically require users to store their personal data centrally on their own proprietary architecture, leading to fragmented user's data and siloed across individual platforms. This not only making it difficult to transfer user's preferences or conversation histories to other GenAI platforms of their choice, but also limits their ability to switch across different GenAI platforms due to their monolithic design. To address these challenges, we propose a decentralized GenAI architecture that provide users full control over their data and privacy through Solid, a standardized interoperable personal data storage framework. We demonstrate a prototype system that integrate multiple LLMs within a single framework without requiring users to centrally store their personal data. We evaluate the system along three dimensions: retrieval-augmented generation (RAG)-based answer quality, multi-turn conversation coherence and qualitative LLMs comparison. Evaluation results show that the framework maintains high-quality responses and coherent conversations, while enabling flexible, cross-model personalization.

Keywords

LLMs, Gen AI, Solid, Decentralization,

1. Introduction

Generative AI (GenAI) applications have rapidly become indispensable tools in various aspects of daily life, significantly boosting productivity and streamlining complex tasks. As these AI systems evolve, their capacity to learn and personalize user experiences grows, often by processing vast amounts of personal and sensitive user data [1]. While this personalization offers benefits, it also introduces substantial challenges related to privacy, data ownership, and trust [2].

Currently, the prevailing architecture for GenAI applications typically requires user to store their personal data on their own proprietary backend systems [3]. This models lead to fragmented user data, siloed across numerous individual applications. Consequently, users face considerable difficulty transferring their preferences, conversational histories, or other curated data between different GenAI platforms. This vendor lock-in not only limits user choice but also hinder innovation and interoperability within the rapidly expanding GenAI ecosystem [3]. The fragmented nature of data also makes it harder for users to understand and control how their personal data is being used, raising significant privacy and security concerns [4].

To address these critical limitations, we propose a decentralized GenAI architecture that empowers users with full control over their data and privacy. Our solution leverages Solid, a standardized, interoperable personal data storage framework initiated by Sir Tim Berners-Lee, the inventor of the World Wide Web, as part of his vision for a decentralized web and the future of agentic AI [5]. Solid fundamentally shifts the paradigm of data ownership by giving individuals control over their personal data through "Pods" (Personal Online Data Stores) [6, 7]. These Pods allow users to store their data independently from any single application and grant granular, revocable permissions to access this

ISWC 2025 Companion Volume, November 2–6, 2025, Nara, Japan

[*] Corresponding author.

ahmadcahyonoadi@mail.ugm.ac.id (A. C. Adi); dheaanggita@mail.ugm.ac.id (D. Anggita); kabul.kurniawan@ugm.ac.id (K. Kurniawan)
0000-0002-9476-8728 (A. C. Adi); 0009-0000-1741-9818 (D. Anggita); 0000-0002-5353-7376 (K. Kurniawan)

data via Access Control [8, 9]. This user-centric approach ensures that data is only accessed with explicit consent, thereby reducing risks related to data misuse and unauthorized access, and aligning with key principles of privacy-by-design and data sovereignty [6, 4]. By decoupling user-curated data from any single Large Language Model (LLM) provider, our approach offers flexibility and seamless personalization across various LLMs within a unified GenAI framework.

The architecture of our proposed framework consists of three main components: (i) a Solid Pod for decentralized data storage, (ii) the GenAI-Apps as the application layer, and (iii) LLMs as the core reasoning engine. To validate our approach, we developed a working prototype called DIKE-Chat. This system allows users to engage in conversations with various LLM agents while maintaining full control over their personal data, which is stored externally in their Solid Pods. The interface supports multiple LLMs, session management, and seamless data retrieval from the Pods. We conducted a extensive evaluation of our framework against several LLMs—across three dimensions: (i) RAG-based answer quality, (ii) multi-turn conversation coherence, and a (iii) qualitative comparison of system-level capabilities.

The remainder of this paper is organized as follows: Section 2 presents the proposed decentralized GenAI architecture, detailing its core components and data flow. Section 3 describes the prototype and its key functionalities as a demonstration of our framework. Section 4 presents evaluation and discusses the results. Finally, Section 5 summarizes our contributions and outlining directions for future work.

2. Decentralized Generative AI Architecture

In this section, we discussed the proposed architecture of our decentralized based GenAI Framework. This architecture is based on Solid Pod as data storage, which supports data ownership on the user side. Solid is equipped with an Access Control List (ACL), which supports data permission services–users will give permission for the data they own to be accessed by the system. By decentralizing data storage and access, our architecture eliminates the need for centralized data repositories, thereby reducing risks related to data misuse and unauthorized access. This user-centric approach ensures that data is only accessed with explicit consent, aligning with principles of data sovereignty and privacy-by-design.

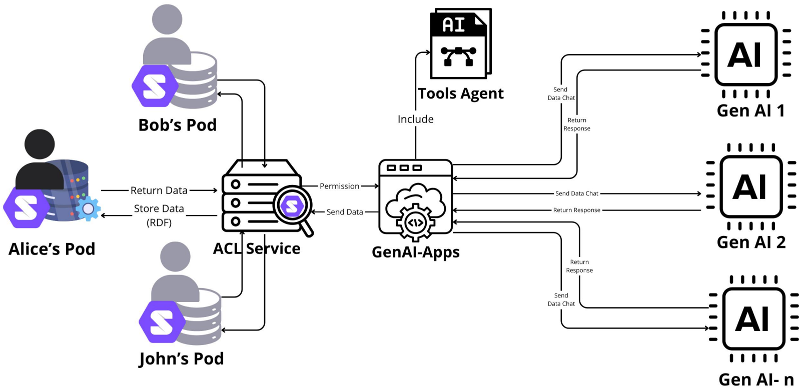


Figure 1: Decentralized GenAI Architecture

Figure 1 shows the overview architecture of the proposed framework. It consists of three main components: *(i)* Solid Pod for decentralized data storage; *(ii)* GenAI-Apps as the application layer; and *(iii)* LLM as the core reasoning engine.

The data flows from the Solid Pod to the DIKE-Chat application, with the ACL Service acting as the central pivot. The ACL Service, which uses the ACL function from Solid, requests permission to access the user's pod data when a session is created. Data from the Solid Pod represented in RDF (turtle) format, which is then stored in a cache. This cached data serves as a knowledge base for the LLMs model. The LLMs model uses this knowledge base to read and respond to the user's questions or statements. Additionally, the GenAI-Apps include Model Context Protocol (MCP) tools that connect to LLM models. After the conversation, the results are stored back to the user's Solid Pod in RDF format as a Turtle file (.ttl)

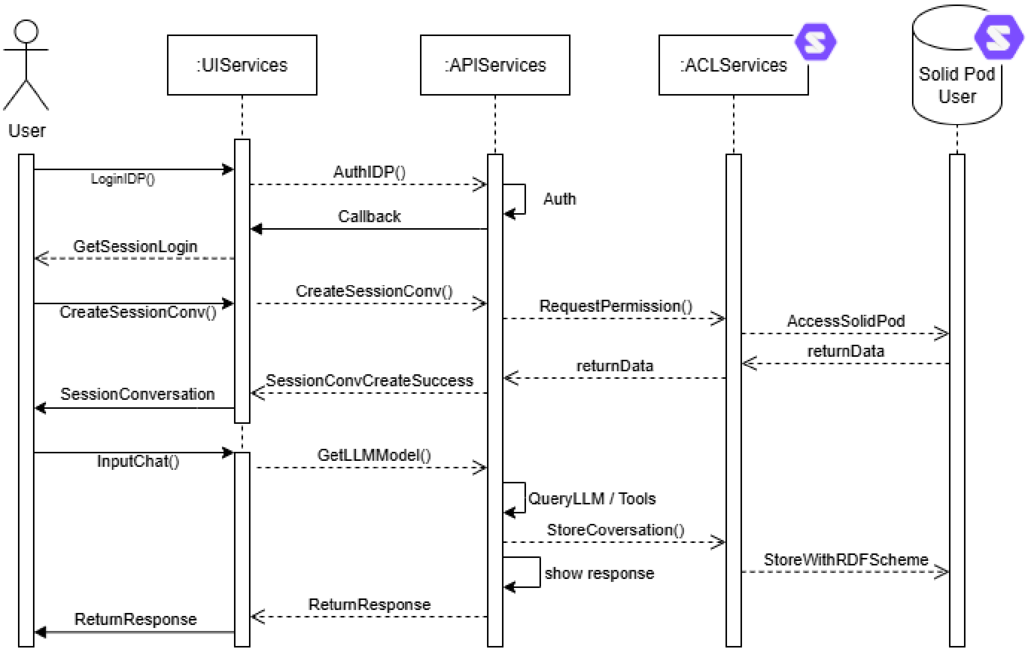


Figure 2: Decentralized Generative AI platform sequence-diagram

Figure 2 illustrates the flow of a user's interaction with the system, starting with the *LoginIDP* function, which handles authorization with an interrupt. The *APIService* retrieves a valid session that the user will use to log in. The user then starts a conversation session to interact with the LLM. Each conversation session connects to the user's POD data. Whenever a session is accessed or changed, permissions are verified to ensure all data is authorized by the user. This authorization allows the data to be used via the Solid Server ACL service. Once the session is active, the user can interact with the LLM, switch between models, and use MCP tools. The connected LLM processes the data and generates responses. Finally, the conversation—including both user questions and LLM responses—is stored in the user's POD in RDF format as a Turtle (.ttl) file.

Listing 1: an excerpt of DIKE-chat RDF representation in Solid POD.

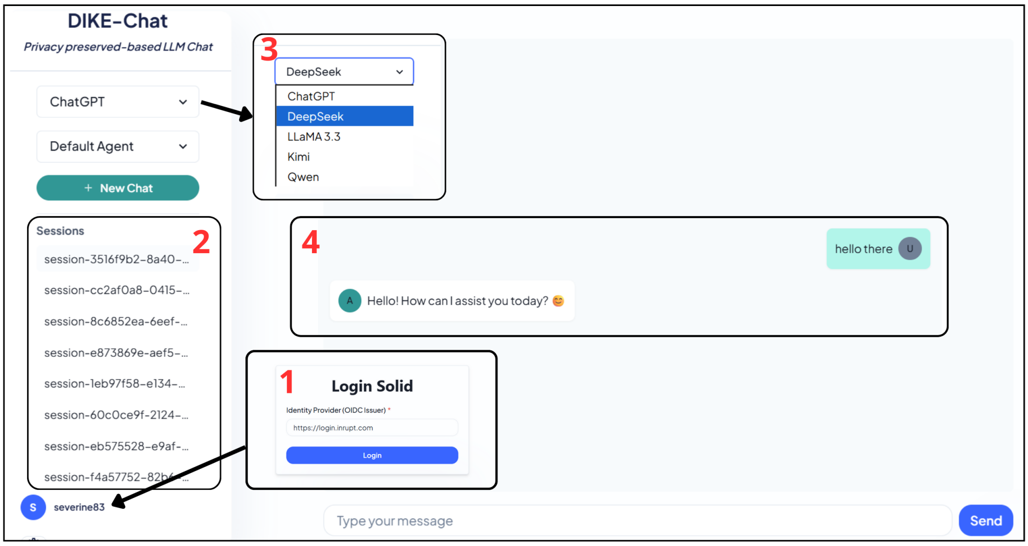
```
@prefix dct: <http://purl.org/dc/terms/> .
@prefix schema: <http://schema.org/> .
@prefix as: <https://www.w3.org/ns/activitystreams#> .

<https://storage.inrupt.com/4310b3e3...>
    dct:created         "2025-07-16T08:05:26.218Z" ;
    dct:identifier      "935f99e5-4351-4b49-be91811a390db8df" ;
    schema:about        "assistant" ;
    schema:text         "Hi␣there!␣How␣can␣I␣help␣you␣today?" ;
    as:actor            "default" ;
    as:generator        "moonshotai/Kimi-K2-Instruct" ;
    as:context          "session-e873869e-aef5-4627-b980-d6236908d062.ttl" .
    ..
```

Listing 1 shows an example of RDF representation stored in a Solid Pod in turtle format (.ttl). It captures each conversation along with supporting data such as creation time, identifier, generator (LLM model), and context (session ID). This structured representation ensures interoperability across applications and maintains a transparent audit trail of user–LLM interactions.

3. Demo

We present a working prototype of the proposed decentralized GenAI framework, named DIKE-Chat. The system allows users to engage in conversations while maintaining full control over their personal data, which is stored externally in Solid Pods rather than centralized servers.


DIKE-Chat
Privacy preserved-based LLM Chat
ChatGPT
Default Agent
+ New Chat
Sessions
session-3516f9b2-8a40-...
session-cc2af0a8-0415-...
session-8c6852ea-6eef-...
session-e873869e-aef5-...
session-1eb97f58-e134-...
session-60c0ce9f-2124-...
session-eb575528-e9af-...
session-f4a57752-82b6-...
S
severine83
3
DeepSeek
ChatGPT
DeepSeek
LLaMA 3.3
Kimi
Qwen
4
U
A
1
Login Solid
Login
Type your message
Send
2

hello there

Hello! How can I assist you today? 😊

Identity Provider (OIDC Issuer) *
https://login.inrupt.com

Figure 3: DIKE-Chat Interface (see the live demo here: https://project-personal-agent-repo.vercel.app/)

As shown in Figure 3, the interface supports multiple LLM agents, session management, and data retrieval from Solid Pods. The interface is built with Next.js 14 and React.js to ensure high performance, responsiveness and seamless model switching without data loss. Users can securely log in via Solid authentication (e.g., Inrupt, Solid Communities) to connect their Solid Pod ①. Once logged in, users gain access to their personalized chat environment where their conversation history and preferences are preserved across sessions ②. The interface also includes dynamic model selection that anable users to switch between LLMs such as *ChatGPT, DeepSeek, LLaMA 3.3, Kimi, and Qwen* without requiring a page refresh or losing context from ongoing conversations ③. Finally, A real-time chat interface displays conversations in a clean, intuitive layout that maintains continuity and enhances usability ④.

4. Evaluation

We evaluated our framework against five leading LLM models: DeepSeek, LLaMA, Kimi, Qwen, and GPT, focusing on three dimensions—retrieval-augmented generation (RAG) relevance, system-level capabilities, and qualitative coherence. Each model received the same structured personal data from Solid and was asked to respond to five user queries. DIKE-Chat dynamically routed queries to different models, while others used their default interfaces. The responses were manually reviewed for relevance and resistance to hallucination. In the second evaluation, we tested multi-turn dialogue by engaging each model in topic-driven conversations, followed by a synthesis question. Outputs were assessed based on coherence, language style, completeness, relevance, and perspective diversity. Lastly, we compared our platform with traditional LLM systems using 12 qualitative criteria, emphasizing decentralization, modularity, data control, and security—highlighting DIKE-Chat's advantages in privacy and user autonomy.

For the RAG-based evaluation, we designed five user queries focusing on public transportation. These queries were created to assess how well LLMs can balance factual grounding with reasoning. The queries are as follows: *(i) What are the benefits and challenges of using public transportation in modern cities?*; *(ii) What are the most effective ways to improve the reliability and coverage of public transportation systems in modern cities?*; *(iii) What are the strengths and potential drawbacks of using technology (like AI* *or real-time systems) in public transport optimization?*; *(iv) How can cities address the challenges and risks associated with using AI and real-time technologies in public transport systems?*; and *(v) What strategies or safeguards can be implemented to ensure that the integration of AI and real-time technologies in public transport remains effective, secure, and equitable?*.

Each model was asked to answer these queries using structured data available through the Solid Pod to examine relevance, hallucination resistance, and contextual integration.

RAG-based Answer Quality As shown in Table 1 left side, GPT received perfect scores, which meant that there were no detectable hallucinations in addition to total alignment with user-provided data. Although the five questions used five different LLMs, DIKE-Chat also performed well, averaging 4.8 on relevancy and hallucination resistance. As evidenced by this constancy, DIKE-chat's model-switching approach has no detrimental effect on factual accuracy. With hallucinatory dates and events that were not in the initial input, Qwen received the lowest score, indicating difficulties in contextualizing responses to particular material. Given the robust performance of DIKE-Chat, it is possible for LLMs to provide responses of a caliber that is comparable to centralized platforms when they are provided with structured, external, user-controlled data through Solid PODs.

Table 1
RAG-based Evaluation (leftside), Qualitative Content Evaluation (rightside). GPT (gpt-3.5-turbo), DeepSeek (V3), LLaMA (Llama-3.3-70B-Instruct-Turbo), Kimi (Kimi-K2-Instruct), Qwen (Qwen3-235B-A22B-fp8-tput).

Platform	Avg. Relevance	Avg. Hallucination
GPT	5.0	5.0
DeepSeek	5.0	4.6
LLaMA	5.0	4.8
Kimi	4.8	4.4
Qwen	4.4	3.2
DIKE-Chat	4.8	4.8

GenAI App	R	Cp	P	Ch	S	Total
GPT	5	5	5	5	4	24
DeepSeek	4	5	4	4	4	21
LLaMA	4	3	3	3	3	16
Kimi	4	4	5	4	3	20
Qwen	3	4	5	3	3	17
DIKE-Chat	5	5	5	3	3	21

Quantitative Content Evaluation When it comes to sustained dialogue and narrative synthesis, GPT once again delivered the most comprehensive and logically constructed output when it comes to continuous dialogue and narrative synthesis. Table 1 right side, DIKE-Chat's ability to dynamically assign questions to the best model allowed it to match GPT in terms of topic relevance, completeness, and perspective. In this evaluation, we assessed five key parameters. (i) topic relevance (R) measures how well LLM responses align with the requested topic; (ii) completeness (Cp) evaluates the depth and breadth of the model analysis; (iii) perspective (P) examines the diversity of viewpoints presented in the response; (iv) coherence (Ch) assesses the logical flow and consistency between parts of the answer; and (v) style (S) evaluates the clarity and consistency of the language used in delivering information. Most LLMs demonstrated strong performance in terms of relevance and completeness. However, in this evaluation, DIKE-Chat received lower scores in coherence and style. This can be attributed to the underlying mechanism that the model-switching mechanism, which can introduce variety in tone, vocabulary, and rhetorical structure between dialogue model-switching. However, it performed better than LLaMA and Qwen, which struggled to keep a consistent logic and completely develop arguments over turns.

System-Level Evaluation DIKE-Chat adopts an architectural approach that differs from conventional LLM platforms by leveraging Solid POD, which allows users to have full control over their personal data (Data Ownership – DO). Through this architecture, users can store data in their own storage, monitor its usage, and manage access permissions (Access Control – AC). This decentralized approach enhances Security and Privacy (S&P) and makes DIKE-Chat suitable for contexts involving Sensitive Data (SD),

Table 2
System Level Evaluation

System	EoU	DO	AC	S&P	FS	RF	Acc	Au	CD	SR	DD	SD
LLM Standard	✓	x	x	x	x	✓	✓	x	x	✓	x	x
Other Multi-LLM	✓	x	x	x	✓	✓	✓	x	x	✓	x	x
DIKE-Chat	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

while also supporting the principles of Data Decentralization (DD) and Scalability for Research (SR). On the interface side, DIKE-Chat is designed with a focus on Ease of Use (EoU). One of its standout features is the ability for users to switch between LLM models (LLM Flexibility/Switching – FS) according to contextual needs. This mechanism provides users with high flexibility to explore differences in style and capabilities of each model while maintaining transparency in the process. Additionally, the system offers real-time feedback (RF) and interaction logging to support Auditability (Au). Furthermore, support for Custom Datasets (CD) enables users to evaluate LLM performance using local or domain-specific data. Overall, this combination of features allows DIKE-Chat to maintain LLM Accuracy (Acc) while preserving user control over data flow and sources. Full evaluation data and scoring details are openly accessible at the project's GitHub repository[1].

In addition to technical testing, we conducted a user evaluation. This evaluation consisted of two tasks: (i) performing multi-turn conversations across different models, and (ii) reviewing personal data retrieved from the Solid Pod. After completing these tasks, users rated ease of use, topic relevance, and other aspects using a 5-point Likert scale. The evaluation results show that Solid Pods give users stronger control over their information compared to centralized platforms. However, users also reported stylistic inconsistency during model switching, aligning with our coherence/style evaluation results. Overall, DIKE-Chat demonstrates several strengths, especially in terms of data ownership, access control, privacy and data interoperability.

5. Conclusion

This work presents a decentralized GenAI framework that addresses key limitations of current LLM applications, particularly in terms of data ownership, access control, privacy, and platform interoperability. By leveraging Solid for user-controlled data storage, our system allows multiple LLMs to operate within a unified architecture without . Evaluation results show that the framework maintains high-quality responses and coherent conversations, while enabling flexible, cross-model personalization. This approach demonstrates a viable path toward user-centric, privacy-preserving GenAI applications that promote transparency, modularity, and control over personal data. Moving forward, our future work will focus on more comprehensive evaluations and integrating the framework with personal agentic AI to evolve it into a proactive, task-oriented agent that utilizes the user's Solid Pod as a dynamic memory.

Acknowledgements. This work was supported by the Department of Computer Science and Electronics, Universitas Gadjah Mada under the Publication Funding Year 2025.

Declaration on Generative AI

During the preparation of this work, the author(s) used ChatGPT in order to: Grammar and spelling check. After using this tool, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

[1] https://github.com/ahmadvoc12/decentralized-genai-chat

References

[1] D. Ottenheimer, Building Trustworthy, Hyper-personal AI Systems: Trusted Execution Environments with Solid, Whitepaper, Inrupt, Inc., 2025. URL: https://www.inrupt.com/whitepaper/trusted-execution-environments-with-solid, vP of Trust Digital Ethics @ Inrupt.

[2] V. Vizgirda, R. Zhao, N. Goel, SocialGenPod: Privacy-Friendly Generative AI Social Web Applications with Decentralised Personal Data Stores, in: Companion Proceedings of the ACM Web Conference 2024, ACM, Singapore Singapore, 2024, pp. 1067–1070. URL: https://dl.acm.org/doi/10.1145/3589335.3651251. doi:10.1145/3589335.3651251.

[3] D. Ottenheimer, Secure AI with data wallets: Privacy-preserving solid architecture for personal data LLMs, in: 3rd Privacy & Personal Data Management Session @ Solid Symposium 2025, 2025. URL: https://openreview.net/forum?id=2BYESPERxb.

[4] E. Mansour, A. V. Sambra, S. Hawke, M. Zereba, S. Capadisli, A. Ghanem, A. Aboulnaga, T. Berners-Lee, A Demonstration of the Solid Platform for Social Web Applications, in: Proceedings of the 25th International Conference Companion on World Wide Web - WWW '16 Companion, ACM Press, Montr&#233;al, Qu&#233;bec, Canada, 2016, pp. 223–226. URL: http://dl.acm.org/citation.cfm?doid=2872518.2890529. doi:10.1145/2872518.2890529.

[5] E. Mustafaraj, Dreams in hypertext: Berners-lee, agentic ai, and the next web frontier, in: Companion Publication of the 17th ACM Web Science Conference 2025, Websci Companion '25, Association for Computing Machinery, New York, NY, USA, 2025, p. 19–21. URL: https://doi.org/10.1145/3720554.3744366. doi:10.1145/3720554.3744366.

[6] A. V. Sambra, E. Mansour, S. Hawke, M. Zereba, N. Greco, A. Ghanem, D. Zagidulin, A. Aboulnaga, T. Berners-Lee, Solid : A platform for decentralized social applications based on linked data, 2016. URL: https://api.semanticscholar.org/CorpusID:49564404.

[7] T. Berners-Lee, Inference from private data – design issues, W3C Design Issues, Personal View, 2023. URL: https://www.w3.org/DesignIssues/PrivateData.html, last changed 22 November 2023; first draft.

[8] M. Bosquet, the Solid Community Group, Access Control Policy (ACP) — Version 0.9.0, 2022-05-18, Editor's Draft ACP 2022-05-18, Solid Project / W3C Solid Community Group, 2022. URL: https://solidproject.org/TR/2022/acp-20220518, mIT License.

[9] S. Capadisli, the Solid Community Group, Web Access Control (WAC) — Candidate Recommendation, 2022-07-05, Candidate Recommendation WAC 2022-07-05, Solid Project / W3C Solid Community Group, 2022. URL: https://solidproject.org/TR/2022/wac-20220705, wAC ontology, ACL access-mode specification.