IDs for AI Systems

Alan Chan
Centre for the Governance of AI
Mila (Quebec AI Institute)
alan.chan@governance.ai

Noam Kolt
University of Toronto

Peter Wills
Centre for the Governance of AI
University of Oxford

Usman Anwar
University of Cambridge

Christian Schroeder de Witt
University of Oxford

Nitarshan Rajkumar
University of Cambridge

Lewis Hammond
University of Oxford
Cooperative AI Foundation

David Krueger
University of Cambridge

Lennart Heim
Centre for the Governance of AI

Markus Anderljung
Centre for the Governance of AI

Abstract

AI systems are increasingly pervasive, yet information needed to decide whether and how to engage with them may not exist or be accessible. A user may not be able to verify whether a system has certain safety certifications. An investigator may not know whom to investigate when a system causes an incident. It may not be clear whom to contact to shut down a malfunctioning system. Across a number of domains, IDs address analogous problems by identifying *particular* entities (e.g., a particular Boeing 747) and providing information about other entities of the same class (e.g., some or all Boeing 747s). We propose a framework in which IDs are ascribed to **instances** of AI systems (e.g., a particular chat session with Claude), and associated information is accessible to parties seeking to interact with that system. We characterize IDs for AI systems, provide concrete examples where IDs could be useful, analyze why and how certain actors could incentivize ID adoption, explore how deployers of AI systems could implement IDs, and highlight limitations and risks. IDs seem most warranted in settings where AI systems could have a large impact upon the world, such as in making financial transactions or contacting real humans. With further study, IDs could help to manage a world where AI systems pervade society.

1 Introduction

AI systems are becoming increasingly commonplace. While current systems can struggle to complete complex tasks (Mialon et al., 2023; Liu et al., 2023; Kinniment et al., 2023; Xie et al., 2024; Jimenez et al., 2024), capabilities seem likely to improve (Hoffmann et al., 2022; Epoch, 2023; Erdil & Besiroglu, 2023; Ho et al., 2024), and future AI agents could carry out a broad range of tasks with only minimal human intervention (Chan et al., 2023; Shavit et al., 2023; Chan et al., 2024). Several commercially deployed AI systems can already search the web, send emails, and write code (OpenAI, 2024b; Anthropic, 2024a). Even when they do not function reliably, AI systems might still be widely used, whether because of cost advantages, hype, or the externalization of their harms (De La Garza, 2020; BBC, 2021; Raji et al., 2022).

2nd Workshop on Regulatable ML at NeurIPS 2024.

At the same time, information to make decisions about engaging with AI systems may not exist or be accessible. Despite obligations to inform parties *that* they are interacting with an AI system (e.g., EU AI Act Art. 50(1) (noa, 2024)), those parties may not know with *which* AI system they are interacting. For example, a party that knowingly interacts with an AI system may not be aware that their system is relatively more vulnerable to adversarial attacks (Zhan et al., 2024). Awareness of such elevated risk could justify additional precautions, such as reviewing the system's actions or abstaining from interaction altogether. Furthermore, the same party—or an investigator—may lack the information to pursue recourse if the system causes harm. Information about the system or the interaction (e.g., whether a system behaved according to safety standards, or the identity of the deployer) could aid incident investigation, allocation of liability, or other legal action (Buiten et al., 2023; Buiten, 2024; Kolt, 2024; Wills, 2024).

Across numerous domains, IDs held by software, assets, individuals, and organizations address analogous problems. IDs can help to ascertain compliance with standards or regulation. For instance, an individual aircraft's tail number is associated to the aircraft's incident and maintenance history, which could inform safety assessments from regulators or aircraft operators. Furthermore, IDs can help to establish whether trust is warranted. A website's valid HTTPS certificate assures users of the website domain's authenticity, and provides a way for users to establish a secure communication channel.[1] Finally, IDs can facilitate redress. Serial numbers on consumer products enable customer support, product recalls, and attribution of liability. A key feature that enables all of the above functions is that IDs are specific to *particular* entities (e.g., a particular Boeing 747), although they may also contain information about other entities of the same class (e.g., some or all Boeing 747s).

Identifying *particular* AI systems, which we term **instances**, could be similarly useful. Given their increasing prevalence, we focus here on digital-assistant-like systems that that respond to user direction, usually (but not necessarily) through natural language (Gabriel et al., 2024). In this context, an instance corresponds to a user and a history of interaction.[2] For example, a particular user's chat session with ChatGPT (with e.g. a GPT-4 backend) is an instance. Separate instances can behave differently, whether because of user instructions (Shanahan et al., 2023; Wei et al., 2023; Bai et al., 2022; Zou et al., 2023; Agarwal et al., 2024), hijacking by an attacker (Greshake et al., 2023; Zhan et al., 2024), or malfunction. As such, instance-specific information could aid decisions about interactions. For example, the ability to distinguish instances from one another could aid incident investigation (e.g., a client files a complaint about an instance) and allocation of liability (e.g., if a third party hijacked the instance).

Yet, current mechanisms are not granular enough to identify instances. System documentation (Gebru et al., 2021; Mitchell et al., 2019; Gilbert et al., 2023; Bommasani et al., 2023) provides information about systems, but such documentation is not an ID for instances or systems (however, an ID could include such documentation). API tokens for services, such as for hotel booking (OpenAI, 2023a), do identify entities that use the tokens, but are usually only user- or device-specific, and do not allow identification of an entity across different services. Even if API requests do include a string with the AI system's name, any attacker could mimic it. Finally, user accounts only separate the activities of AI systems from different users. We summarize these differences and discuss other identity systems in Table 1.

To prepare for a world with ubiquitous AI interactions, we propose (instance-level) IDs for AI systems. An AI ID is a container for **1)** an identifier and **2)** attributes. An **identifier** is a unique string that refers to an instance. An identifier could be randomly generated, or could itself encode some information about the instance (e.g., the identifier could include the exact time at which the instance started operating). An **attribute** is any information that could pertain to the instance, and could include behaviour, properties, context, or relationships to other instances or systems. An identifier enables the association of attributes to an instance, similar to how serial numbers associate information to a particular product. Attributes could be specific to an instance (e.g., prior incidents associated with the instance), or could apply more broadly to other instances (e.g., a system[3] card). An ID could directly

[1] Despite a lack of centralized enforcement, HTTPS gradually became the norm due to widespread awareness of its security benefits and collective advocacy from web browsers, search engines, and other organizations (Encrypt, 2024; Hancock, 2021).

[2] This definition of instance does not take memory into account. While it will suffice for this work, see Appendix E for a more general definition.

[3] A system (e.g., ChatGPT) could use different models (e.g., GPT-4, GPT-3.5) as a backend. We consider system cards to include model cards (Mitchell et al., 2019).

Existing Systems	Differences with AI IDs
Personhood credentials (Borge et al., 2017; Adler et al., 2024)	A personhood credential identifies a unique human in an anonymous way, but does not identify AI systems.
ORCID	An ORCID is meant to point to a particular researcher, but there is no reliable way to verify an entity's claim that a particular ORCID corresponds to it.
DOI	A DOI is meant to provide a unique, persistent identifier of a digital object, but there is no reliable way to verify an entity's claim that a particular DOI corresponds to it.
Watermarks	Watermarks are meant to indicate AI provenance in content and do not contain information about particular AI systems.
C2PA (C2PA, 2023)	C2PA is a standard for making verifiable claims about the provenance of content. Such claims can point to an ID, but are not IDs in themselves.
CAPTCHAs	CAPTCHAs are meant to detect presence of non-humans, rather than identify them.
API tokens	API tokens are user- or device-specific and do not identify entities across different APIs.
User accounts	User accounts only separate the activities of AI systems from different users.
System documentation, such as a system card	System documentation provides information about systems, but is not an ID for instances or systems.

Table 1: We collate a number of existing identity-like systems, some of which do not apply to AI systems, and describe their differences with AI IDs.

include attributes (e.g., the name of the deployer) or could link to them (e.g., a link to a database of prior incidents).

Different instances or systems could warrant attributes of differing levels of granularity or detail. For example, attaching a user identifier to an ID may only be appropriate in high-stakes settings, such as when a company uses an AI system to interface with critical infrastructure. Prior incidents associated with an instance would likely be more useful for future AI agents (Chan et al., 2023; Shavit et al., 2023; Chan et al., 2024; Gabriel et al., 2024), which could act autonomously and persist over long durations.

1.1 Contributions

We propose IDs for AI systems. In Section 2, we characterize the central properties of IDs. In Section 3, we provide concrete examples where IDs could be useful. In Appendix B, we argue that there will likely be demand for IDs from several key actors, especially in high-stakes settings. We also explore potential ways for these actors to incentivize ID use. In Appendix C, we analyze a potential implementation of IDs. In Section 4, we investigate some of the limitations of IDs and of our analysis, including privacy and security risks of IDs, as well as uncertainty about the broader societal consequences of IDs for AI systems.

2 Characterizing IDs

An ID is a container (Korenhof et al., 2014) for **1)** an identifier (that corresponds to an instance) and **2)** attributes. We characterize a design space for IDs: the attributes it includes, to whom it is accessible, and to what extent it is verifiable. Our goal in this section is to investigate possible designs and why they may be desirable, rather than to prescribe specific choices.

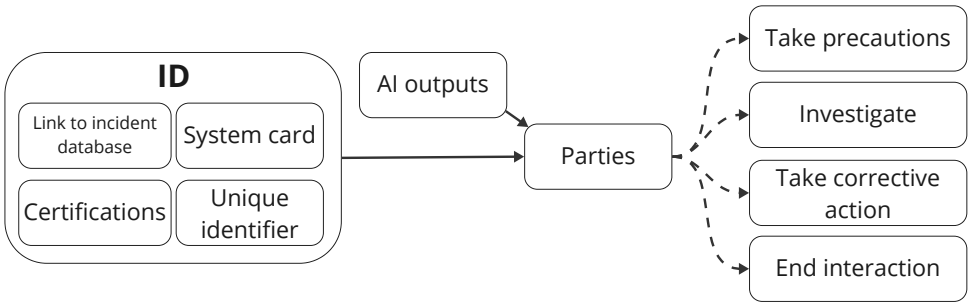


Figure 1: IDs contain a unique identifier along with attributes (e.g., a system card, certifications, or a link to previous incidents). We also display some potential actions that parties might take based on information in an ID.

2.1 Attributes

In addition to the identifier, an ID can contain (or link to) attributes: any information that could be useful to a party interacting with the corresponding instance. An attribute can vary along two dimensions.

Category: Categories of attributes include (but are not limited to) behaviour, properties, context, and relationships to other instances or systems. Some examples are:

- Behaviour: prior incidents (Wei & Heim, 2024)
- Properties: information in a system card (Mitchell et al., 2019; Gebru et al., 2021); the results of evaluations (Shevlane et al., 2023; Weidinger et al., 2023)
- Context: the system prompt; external memory (Wang et al., 2023)
- Relationships to other instances or systems: any other AI systems the instance has created or is running; any instances that have created the instance in question

Specificity: An attribute could provide information about multiple instances (even of different systems), rather than just the instance corresponding to the ID. With incidents as a (sub)category, potential levels of specificity could include:

- Instance: incidents associated with a particular instance
- Instances satisfying certain properties: incidents associated with instances whose system prompts contain harmful instructions
- User: incidents associated with a particular user
- A particular (type of) party: incidents involving a particular (type of) party and a system (e.g., all incidents involving hospitals and the 9-April-2024 version of GPT-4)
- System: incidents associated with a system (e.g., the 9-April-2024 version of GPT-4)
- Systems: incidents associated with multiple versions of a system (e.g., all versions of GPT-4)

An ID could contain (or link to) the same category of attribute at varying levels of specificity. Analogously, a tail number on an individual aircraft could be associated with maintenance records for systems, sub-systems, sub-sub-systems, etc of that aircraft.

2.2 Access

An ID is simply an abstraction that contains or links to information. That abstraction can be presented in several potential ways, such as on a separate web page or in a pop-up window. We explore design decisions around who can access an ID and how it is presented to them.

Parties that can access the ID: We distinguish between primary parties and secondary parties. Primary parties receive an ID when interacting with the corresponding instance. Secondary parties

receive the ID through other means, such as directly from a primary party, or from records. Some examples include:

- Primary: service providers; the user of the instance; other instances that interact with the instance in question[4]
- Secondary: auditors; regulators

Some primary parties and how IDs may be implemented include:

- Service providers: ID is included in a JSON payload
- Users: ID is accessible through a mouseover icon in a chat interface
- Other instances that interact with the instance in question: ID is sent in any communications with other instances

Selective disclosure: Some attributes may only be appropriate for certain parties. Examples:

- If user identification (e.g., from a know-your-customer process (Egan & Heim, 2023)) is included in an ID, it may only be appropriate for government authorities to access this identification for e.g. regulatory purposes
- A system prompt could contain both sensitive user information and information relevant for interacting parties (e.g., does the system prompt contain a jailbreak?)

Persistence: IDs could be accessible for varying amounts of time after an instance has ceased to operate. Practical considerations, such as storage capacity, could impose limitations. The appropriate duration could also depend upon the application domain. Analogously, financial institutions have obligations to maintain records for set periods of time (noa, 2022).

2.3 Verifiability

Suppose an author creates an ID (or a link to an ID) for an instance. A party interacts, or seeks to interact, with the instance. The party receives an ID. For the party to trust that the ID provides accurate information about the instance, the following are necessary (but not sufficient; see further discussion below).

(1) The party's received ID is the same as the author's created ID

(2) The claimed author of the received ID is indeed the author of the received ID

(3) The received ID corresponds to the instance in question

(4) The party trusts the author

The first three criteria lead to the following threat models:

- **Tampering**: An attacker modifies the ID while it is in transit from the author to the party interacting with the instance.
- **ID spoofing**: An attacker creates another ID, sends it to the party, and claims it originated from the author.
- **Instance spoofing**: An attacker uses the author's ID for their own instances.

We consider IDs (i.e., an ID system) to be **verifiable** if they defend against these threats. We provide an illustration of these threat models in Figure 2.

Existing technology could be drawn upon to provide verifiability. For digital-assistant-like AI systems (e.g., GPT-4), any digital communications usually would take place over TLS/SSL (Rescorla, 2018), which would address tampering. We leave tampering of other AI systems' IDs (e.g., IDs for self-driving cars) to future work. To combat ID spoofing, authors could digitally sign the ID (potentially with public key ownership verified by a trusted third party). For instance spoofing, there are existing

[4]Similar to how a TLS handshake allows mutual verification before establishing a communication channel, AI systems could potentially use IDs to establish trust before interacting with each other.

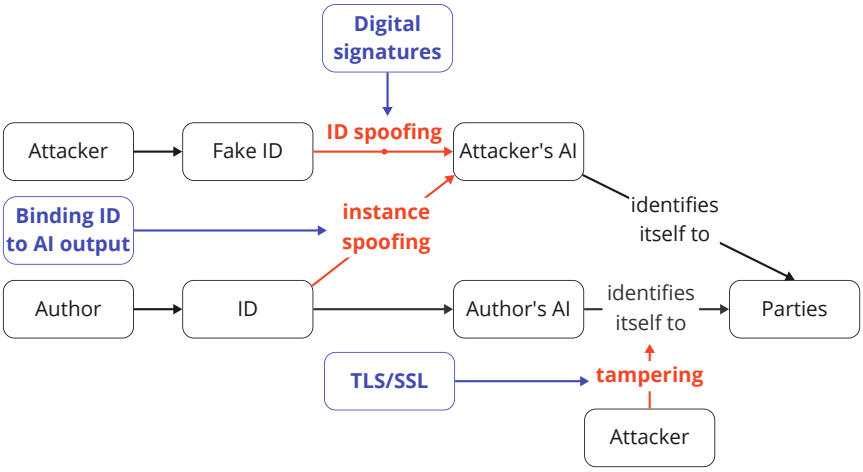


Figure 2: We illustrate how **existing technologies** can help ensure the verifiability of IDs against the **threat models** in Section 2.3. We emphasize that our examples of existing technologies are meant to be illustrative, not prescriptive.

standards for binding a digital object, such as an ID, to a piece of content, such as an AI system's output. For example, C2PA involves binding provenance claims to content (C2PA, 2023), and calls the binded object a **manifest**. Such binding could ensure that IDs are associated to particular outputs from an instance in a verifiable way.[5] *We emphasize that our examples here are meant to be illustrative, not prescriptive.* Security vulnerabilities in existing technologies may weaken verifiability and require new solutions.

Given space limitations and the complexity of the involved social questions, we do not focus on step (4) in this work. Similarly, we also scope out issues relating to the accuracy of the attributes in an ID. Accuracy is not the same as verifiability, since even if (1)-(4) are satisfied, the author could have made mistakes in creating and linking to the information.

3 How Could IDs Help?

We provide some hypothetical case studies where IDs could be useful and compare IDs to alternatives. AI agents—systems which act autonomously to achieve goals with little to no user direction, instruction, or supervision (Chan et al., 2023; Shavit et al., 2023; Gabriel et al., 2024)—feature prominently in our scenarios.

3.1 Shutting Down Malfunctioning Agents

Scenario: Suppose a user runs an AI agent to provide continual recommendations across many platforms and services (e.g., for media, shopping etc).[6] To generate such recommendations, the agent interacts with plugins for various web services, such as search engines, online discussion fora, etc. The agent provides its ID to each service. Unbeknownst to the user, the agent begins to malfunction (e.g., by chance, prompt injection) and disrupt services (e.g., spamming discussion fora, sending malformed requests to services or overloading them, conducting prompt injection attacks on other AI agents). Services interacting with the agent flag its malfunction. Service providers use information in the ID to notify the deployer, and block further interaction with the agent. The deployer receives multiple reports of malfunction for the same ID and investigates the agent, potentially shutting it down and notifying the user.

[5] If desired, the verifiability of IDs over extended interactions could be ensured by linking manifests to each other cryptographically, such as in a blockchain.

[6] The idea for this scenario is from Zittrain (2024).

Alternatives to IDs: To block further interaction with the agent, a service provider can simply revoke its API token and potentially block the requesting IP address. These actions would not prevent the agent from accessing new services, however. As well, API tokens would not identify an agent across different services to enable shutdown if necessary. Alternatively, services could notify the user of agent malfunction. With a good user interface, users could notice such malfunction and shut down their agent. If users are careless or rarely supervise their agent (perhaps they have many such agents running in the background), it seems useful to have IDs as a backup option.

Limitations of the scenario: It is unclear how impactful the disruption of services would be. Deployers could also be unresponsive to notifications from service providers.

3.2 Verifying Certification

Scenario: A user wishes to interact with an agent from another person. The user wants to make sure that the agent does not act in an undesirable way. For example, the user would want to make sure that a malicious third party did not prompt inject (Greshake et al., 2023) the agent (perhaps because the user is passing sensitive information or is accessing resources the agent will send over). A trusted third party, such as a deployer or an auditor, checks the interaction history of the agent at runtime for prompt injections (e.g., with a tool like Beurer-Kellner et al. (2024)). After verifying the absence of prompt injections, the trusted third party adds a certification onto the agent's ID. The user sees the certification on the ID and continues interaction.

Alternatives to IDs: The trusted third party could send the certification directly to a user requesting it, but some sort of identifier would still be needed to ensure that the certification corresponds to the agent with which the user is interacting.

Limitations of the scenario: Certification would raise privacy concerns because the trusted third party would be able to see the whole interaction history. Privacy-preserving ways to perform such audits would be valuable to explore (Trask et al., 2023, 2024). A potential alternative would be for the auditor to run behavioural tests on the agent at runtime, and link the results of the tests to the ID.

3.3 Investigating Scam Calls

Scenario: Suppose a user spins up an AI agent to carry out scam phone calls. The agent creates multiple sub-agents—potentially using different AI deployers—and directs each one to carry out scam calls through a telephone service. The service receives each sub-agent's ID, each of which contains information about the deployer and a connection to the original agent's ancestor ID. The service provider connects the sub-agent to a telephone network and creates a call detail record (CDR), linking the call with the sub-agent's ID. Some scam victims notify the operator of the telephone service. The telephone provider notifies law enforcement, who investigates the relevant CDRs and IDs. Law enforcement notifies the deployers of the sub-agents, potentially after obtaining a warrant so as to obtain access to the deployers' logs. Through ID linkages between parent and child agents, law enforcement and deployers discover the agent and user behind the spam calls.

Alternatives to IDs: The main alternative would be for law enforcement to ask ISPs to perform IP address tracking, if the telephone provider records the IP addresses of the sub-agents. Interacting with ISPs (and potentially obtaining a warrant) could take additional time and effort. Conservatively, in this case IDs could help to save time during incident investigation, but not necessarily enable something that was otherwise impossible.

Limitations of the scenario: The main limitation of this scenario is the provision of IDs from the sub-agents to the service provider. Users could use deployers that do not implement IDs, or could run their own agents to avoid passing IDs to APIs. A potential response is for the telephone service provider to require IDs before provisioning telephone access.

4 Limitations

We discuss limitations of IDs and of our analysis. We have already discussed potential barriers to ID adoption in Appendix B.

4.1 Misuse of IDs

If IDs were prevalent but verifiability was not assured, misuse of IDs could be a significant problem. For example, an attacker's agent could masquerade as somebody else's agent to perform illicit actions, such as sending spam or carrying out cyberattacks. Overdependence on such IDs could lead to false accusations against innocent users, who may not have the resources or expertise to defend themselves and point out flaws in the ID system. Ideally, the compromise of an ID should be made clear to all that can access it. IDs should also likely not be the sole or conclusive source of evidence for incident investigations. Concerns about the failure of ID verifiability mirror similar concerns about the reliability of watermarks (Gleichauf & Geer, 2024).

4.2 Decentralized Operation of Agents

We have not discussed how IDs could be implemented for AI systems which users run themselves, without a deployer. If service providers decide to require IDs, users will need straightforward ID implementation methods. Otherwise, they could be locked out of services or be forced to use a deployer. A potential inspiration is Let's Encrypt (Encrypt, 2024), which automated the process of HTTPS certificate creation for websites. A similar service could work with software for running AI systems (e.g., transformers (Wolf et al., 2020), AutoGen (Wu et al., 2023)) to allow users to request and attach IDs. We leave practical details for future work.

4.3 Failure to Inform

Parties could ignore IDs, just as internet users sometimes ignore indicators of untrustworthiness (Norris & Brookes, 2021). Software—or future AI assistants (Gabriel et al., 2024)—could make informed decisions for users in some circumstances, just as browsers reject websites with invalid HTTPS certificates. However, it remains unclear what information would justify rejecting an interaction with an AI system. Other parties could misunderstand or misinterpret an ID's attributes. For example, it can difficult to assess the external validity of evaluation results (Fourrier et al., 2023; Weidinger et al., 2023).

ID attributes could also be inaccurate. For example, deployers could unintentionally link to an incorrect version of system documentation, or simply fail to update it. OpenAI has occasionally released new versions of GPT-4,[7] but has not updated the GPT-4 system card (OpenAI, 2023b) since the initial release, as of April 2024. As a potential mitigation, regular audits (Raji et al., 2020; Sharkey et al., 2023) could verify the accuracy of attributes.

4.4 Bypassing IDs

As with other types of real-world identity systems, IDs for AI systems cannot wholly prevent undesirable behavior. IDs could be lent, or the benefits obtained from ID use—such as purchase of a good—could be transferred. Users may hesitate to lend IDs since the borrower could inflict lasting reputational damage on the ID. Yet, similar to the purchase of alcohol for minors, it seems difficult to track or prevent transfer of goods. Future work could assess how likely and how damaging such transfer could be.

4.5 Privacy Risks

Although an identifier by itself would not reveal anything about its user, the user's privacy could still be compromised in a number of ways. First, information about the instance could reveal user details. For example, timestamps and details of an instance's activities—as might be contained in incident reports—could help a third-party to identify users. The existence of IDs may engender pressure to attach additional information, which could facilitate further user identification. For example, service abuse could potentially be correlated with certain activity patterns, which could motivate corresponding data collection. Second, if the deployer maintains an internal database linking a system's ID to the corresponding user account, other parties could obtain and misuse the information. Security vulnerabilities could allow attackers to obtain the database. Overreaching governments could force deployers to reveal the users of particular systems.

[7] See `https://platform.openai.com/docs/models/gpt-4-turbo-and-gpt-4`.

4.6 Broader Societal Consequences

IDs with additional user information could enable potentially harmful ranking of users. Actors implementing IDs may wish to include such information if it is correlated with undesirable outcomes, such as fraud. Yet, the existence of confounders could lead to instances of some users being unfairly denied access to services, similar to how toxicity classifiers can biased against African American English (Sap et al., 2019). Even if the relationship between information and undesirable outcome was causal, taking such information into account could still be unjustly discriminatory.

IDs could also enable influence over particular instances and their users. Even if an ID does not contain information about an instance's interactions, external actors could record particular IDs have been involved in interactions, just as cookies can track user activity across websites. Instances that are active for long periods of time could reveal much information about a user's preferences. Actors interacting with instances could try to influence their behaviour through means such as prompting. Advertisers and businesses could try to get instances to purchase or recommend their products or services for their corresponding users. Political actors could try to get instances to influence the voting behaviour of their users. Governments could attempt to enforce rules upon particular instances. Such influence could be strong, especially if personal AI assistants become much more widespread and central to daily interactions (Gabriel et al., 2024). The specific impacts of this influence deserve further study.

Finally, IDs could create a separate digital channel for interactions between trusted AI systems. The vast majority of digital interactions could in the future be between AI systems, especially if future AI assistants handle most of a user's activities (Gabriel et al., 2024). If IDs enable reliable interactions, users may prefer their AI systems to interact only with AI systems that present IDs. Much digital activity could take place between networks of ID-bearing AI instances that trust (and can verify) each other. Entities that trust each other may be able to engage in more productive interactions. As a corollary, instances without IDs may become severely disadvantaged, relative to instances within ID-bearing networks. To avoid such disadvantage, users may have no choice but to submit to the potential negative consequences of IDs we have discussed heretofore in this section. On balance, the consequences of a separate digital channel for AI systems remain unclear.

5 Conclusion

To inform crucial decisions about AI interactions, we proposed a framework for creating IDs for AI systems. IDs can vary in their attributes, how accessible they are to various parties, and to what extent they are verifiable. IDs could be useful in several scenarios, including for shutting down malfunctioning systems, verifying certifications, and incident investigation. There could be significant demand for IDs from governments, service providers, and users, particularly when systems engage in high-stakes tasks. These actors also have means to incentivize ID usage. ID implementation seems feasible for deployers, but implementation in decentralized settings will require further study.

More research is required to understand and address the potential risks of IDs. First, more work is required to understand what information is appropriate to attach to IDs, and in what contexts. Second, we need to understand better how IDs should be used. Third, when ID verifiability fails, there should be ways to limit the impacts of such failure. Fourth, the broader societal consequences of introducing IDs require further study.

Limited experimentation with IDs seems justified given the balance of potential benefits and risks. IDs seem most warranted in settings where AI systems could have a large impact upon the world, such as in making financial transactions or contacting real humans. in these cases, service providers could experiment with incentives for ID use. For instance, they could rate limit instances without IDs, while still allowing such instances to access services. Deployers could experiment with ID implementation.

IDs have facilitated essential functions across a variety of domains. Incident investigation, allocation of liability, and establishment of trust would be more difficult without IDs for products, organizations, and software systems. As AI systems become increasingly prevalent, deficiencies in such functions could make it more difficult to manage risks from AI systems. If implemented well, IDs for AI systems could enable mechanisms to navigate this emerging world.

References

12 CFR § 1026.25 - Record retention. *Federal Register*, January 2022.

Regulation (EU) 2024/1689 of the European Parliament and of the Council of 13 June 2024 laying down harmonised rules on artificial intelligence and amending Regulations (EC) No 300/2008, (EU) No 167/2013, (EU) No 168/2013, (EU) 2018/858, (EU) 2018/1139 and (EU) 2019/2144 and Directives 2014/90/EU, (EU) 2016/797 and (EU) 2020/1828 (Artificial Intelligence Act), July 2024. URL http://data.europa.eu/eli/reg/2024/1689/oj. Volume: L tex.shorthand: EU AI Act.

Steven Adler, Zoë Hitzig, Shrey Jain, Catherine Brewer, Wayne Chang, Renée DiResta, Eddy Lazzarin, Sean McGregor, Wendy Seltzer, Divya Siddarth, Nouran Soliman, Tobin South, Connor Spelliscy, Manu Sporny, Varya Srivastava, John Bailey, Brian Christian, Andrew Critch, Ronnie Falcon, Heather Flanagan, Kim Hamilton Duffy, Eric Ho, Claire R. Leibowicz, Srikanth Nadhamuni, Alan Z. Rozenshtein, David Schnurr, Evan Shapiro, Lacey Strahm, Andrew Trask, Zoe Weinberg, Cedric Whitney, and Tom Zick. Personhood credentials: Artificial intelligence and the value of privacy-preserving tools to distinguish who is real online, August 2024. URL http://arxiv.org/abs/2408.07892. arXiv:2408.07892 [cs].

Rishabh Agarwal, Avi Singh, Lei M. Zhang, Bernd Bohnet, Stephanie Chan, Ankesh Anand, Zaheer Abbas, Azade Nova, John D. Co-Reyes, Eric Chu, Feryal Behbahani, Aleksandra Faust, and Hugo Larochelle. Many-Shot In-Context Learning, April 2024. URL http://arxiv.org/abs/2404.11018. arXiv:2404.11018 [cs].

Anthropic. Introducing the next generation of Claude, March 2024a. URL https://www.anthropic.com/news/claude-3-family.

Anthropic. Tool use (function calling), 2024b. URL https://docs.anthropic.com/claude/docs/tool-use.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Kamile Lukosuite, Liane Lovitt, Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noemi Mercado, Nova DasSarma, Robert Lasenby, Robin Larson, Sam Ringer, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Tamera Lanham, Timothy Telleen-Lawton, Tom Conerly, Tom Henighan, Tristan Hume, Samuel R. Bowman, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, and Jared Kaplan. Constitutional AI: Harmlessness from AI Feedback, December 2022. URL http://arxiv.org/abs/2212.08073. arXiv:2212.08073 [cs].

Richard Barnes, Jacob Hoffman-Andrews, Daniel McCarney, and James Kasten. Automatic certificate management environment (acme). Technical report, 2019.

BBC. Post Office scandal explained: What the Horizon saga is all about. April 2021. URL https://www.bbc.com/news/business-56718036.

Luca Beurer-Kellner, Mislav Balunovic, and Marc Fischer. invariantlabs-ai/invariant, July 2024. URL https://github.com/invariantlabs-ai/invariant. original-date: 2024-05-08T08:57:47Z.

Monika Bickert. Our Approach to Labeling AI-Generated Content and Manipulated Media, April 2024. URL https://about.fb.com/news/2024/04/metas-approach-to-labeling-ai-generated-content-and-manipulated-media/.

Rishi Bommasani, Dilara Soylu, Thomas I. Liao, Kathleen A. Creel, and Percy Liang. Ecosystem Graphs: The Social Footprint of Foundation Models, March 2023. URL http://arxiv.org/abs/2303.15772. arXiv:2303.15772 [cs].

Maria Borge, Eleftherios Kokoris-Kogias, Philipp Jovanovic, Linus Gasser, Nicolas Gailly, and Bryan Ford. Proof-of-Personhood: Redemocratizing Permissionless Cryptocurrencies. In *2017 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, pp. 23–26, April 2017. doi: 10.1109/EuroSPW.2017.46. URL https://ieeexplore.ieee.org/abstract/document/7966966.

Miriam Buiten, Alexandre de Streel, and Martin Peitz. The law and economics of AI liability. *Computer Law & Security Review*, 48:105794, April 2023. ISSN 0267-3649. doi: 10.1016/j.clsr.2023.105794. URL https://www.sciencedirect.com/science/article/pii/S0267364923000055.

Miriam C. Buiten. Product liability for defective AI. *European Journal of Law and Economics*, February 2024. ISSN 1572-9990. doi: 10.1007/s10657-024-09794-z. URL https://doi.org/10.1007/s10657-024-09794-z.

Peter Buneman, Sanjeev Khanna, and Tan Wang-Chiew. Why and Where: A Characterization of Data Provenance. October 2001. doi: 10.1007/3-540-44503-X_20. URL https://link.springer.com/chapter/10.1007/3-540-44503-X_20.

C2PA. C2PA Technical Specification, 2023. URL https://c2pa.org/specifications/specifications/1.3/specs/C2PA_Specification.html#_introduction.

Alan Chan, Rebecca Salganik, Alva Markelius, Chris Pang, Nitarshan Rajkumar, Dmitrii Krasheninnikov, Lauro Langosco, Zhonghao He, Yawen Duan, Micah Carroll, Michelle Lin, Alex Mayhew, Katherine Collins, Maryam Molamohammadi, John Burden, Wanru Zhao, Shalaleh Rismani, Konstantinos Voudouris, Umang Bhatt, Adrian Weller, David Krueger, and Tegan Maharaj. Harms from Increasingly Agentic Algorithmic Systems. In *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '23, pp. 651–666, New York, NY, USA, June 2023. Association for Computing Machinery. ISBN 9798400701924. doi: 10.1145/3593013.3594033. URL https://dl.acm.org/doi/10.1145/3593013.3594033.

Alan Chan, Carson Ezell, Max Kaufmann, Kevin Wei, Lewis Hammond, Herbie Bradley, Emma Bluemke, Nitarshan Rajkumar, David Krueger, Noam Kolt, Lennart Heim, and Markus Anderljung. Visibility into AI Agents. In *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '24, pp. 958–973, New York, NY, USA, June 2024. Association for Computing Machinery. ISBN 9798400704505. doi: 10.1145/3630106.3658948. URL https://dl.acm.org/doi/10.1145/3630106.3658948.

ChrisHMSFT, PatrickFarley, mrbullwinkle, eric urban, and aahill. Data, privacy, and security for Azure OpenAI Service - Azure AI services, June 2023. URL https://learn.microsoft.com/en-us/legal/cognitive-services/openai/data-privacy.

David Cooper, Stefan Santesson, Stephen Farrell, Sharon Boeyen, Russell Housley, and William Polk. Internet X.509 public key infrastructure certificate and certificate revocation list (CRL) profile. Technical report, 2008.

Alejandro De La Garza. States' Automated Systems Are Trapping Citizens in Bureaucratic Nightmares With Their Lives on the Line. *TIME*, May 2020. URL https://time.com/5840609/algorithm-unemployment/.

Janet Egan and Lennart Heim. Oversight for Frontier AI through a Know-Your-Customer Scheme for Compute Providers, October 2023. URL http://arxiv.org/abs/2310.13625. arXiv:2310.13625 [cs].

Let's Encrypt. Let's Encrypt Stats - Let's Encrypt, January 2024. URL https://letsencrypt.org/stats/.

Epoch. Key trends and figures in Machine Learning, 2023. URL https://epochai.org/trends.

Ege Erdil and Tamay Besiroglu. Algorithmic progress in computer vision, 2023. _eprint: 2212.05153.

R Fielding, M Nottingham, and J Reschke. RFC 9110: HTTP semantics, 2022.

Clémentine Fourrier, Nathan Habib, Julien Launay, and Thomas Wolf. What's going on with the Open LLM Leaderboard?, June 2023. URL https://huggingface.co/blog/open-llm-leaderboard-mmlu.

Hiroki Furuta, Yutaka Matsuo, Aleksandra Faust, and Izzeddin Gur. Exposing Limitations of Language Model Agents in Sequential-Task Compositions on the Web, February 2024. URL http://arxiv.org/abs/2311.18751. arXiv:2311.18751 [cs].

Iason Gabriel, Arianna Manzini, Geoff Keeling, Lisa Anne Hendricks, Verena Rieser, Hasan Iqbal, Nenad Tomašev, Ira Ktena, Zachary Kenton, Mikel Rodriguez, Seliem El-Sayed, Sasha Brown, Canfer Akbulut, Andrew Trask, Edward Hughes, A. Stevie Bergman, Renee Shelby, Nahema Marchal, Conor Griffin, Juan Mateos-Garcia, Laura Weidinger, Winnie Street, Benjamin Lange, Alex Ingerman, Alison Lentz, Reed Enger, Andrew Barakat, Victoria Krakovna, John Oliver Siy, Zeb Kurth-Nelson, Amanda McCroskery, Vijay Bolina, Harry Law, Murray Shanahan, Lize Alberts, Borja Balle, Sarah de Haas, Yetunde Ibitoye, Allan Dafoe, Beth Goldberg, Sébastien Krier, Alexander Reese, Sims Witherspoon, Will Hawkins, Maribeth Rauh, Don Wallace, Matija Franklin, Josh A. Goldstein, Joel Lehman, Michael Klenk, Shannon Vallor, Courtney Biles, Meredith Ringel Morris, Helen King, Blaise Agüera y Arcas, William Isaac, and James Manyika. The Ethics of Advanced AI Assistants, April 2024. URL http://arxiv.org/abs/2404.16244. arXiv:2404.16244 [cs].

Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé Iii, and Kate Crawford. Datasheets for datasets. *Communications of the ACM*, 64(12):86–92, December 2021. ISSN 0001-0782, 1557-7317. doi: 10.1145/3458723. URL https://dl.acm.org/doi/10.1145/3458723.

Thomas Krendl Gilbert, Nathan Lambert, Sarah Dean, Tom Zick, Aaron Snoswell, and Soham Mehta. Reward Reports for Reinforcement Learning. In *Proceedings of the 2023 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '23, pp. 84–130, New York, NY, USA, August 2023. Association for Computing Machinery. ISBN 9798400702310. doi: 10.1145/3600211.3604698. URL https://dl.acm.org/doi/10.1145/3600211.3604698.

Bob Gleichauf and Dan Geer. Digital Watermarks Are Not Ready for Large Language Models, February 2024. URL https://www.lawfaremedia.org/article/digital-watermarks-are-not-ready-for-large-language-models.

Kai Greshake, Sahar Abdelnabi, Shailesh Mishra, Christoph Endres, Thorsten Holz, and Mario Fritz. Not what you've signed up for: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection, May 2023. URL http://arxiv.org/abs/2302.12173. arXiv:2302.12173 [cs].

Izzeddin Gur, Ofir Nachum, Yingjie Miao, Mustafa Safdari, Austin Huang, Aakanksha Chowdhery, Sharan Narang, Noah Fiedel, and Aleksandra Faust. Understanding HTML with Large Language Models, May 2023. URL http://arxiv.org/abs/2210.03945. arXiv:2210.03945 [cs].

Izzeddin Gur, Hiroki Furuta, Austin Huang, Mustafa Safdari, Yutaka Matsuo, Douglas Eck, and Aleksandra Faust. A Real-World WebAgent with Planning, Long Context Understanding, and Program Synthesis, February 2024. URL http://arxiv.org/abs/2307.12856. arXiv:2307.12856 [cs].

Alexis Hancock. HTTPS Is Actually Everywhere, September 2021. URL https://www.eff.org/deeplinks/2021/09/https-actually-everywhere.

Anson Ho, Tamay Besiroglu, Ege Erdil, David Owen, Robi Rahman, Zifan Carl Guo, David Atkinson, Neil Thompson, and Jaime Sevilla. Algorithmic progress in language models, March 2024. URL https://arxiv.org/abs/2403.05812v1.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training Compute-Optimal Large Language Models, March 2022. URL http://arxiv.org/abs/2203.15556. arXiv:2203.15556 [cs].

The White House. National Cybersecurity Strategy. Technical report, March 2023.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. SWE-bench: Can Language Models Resolve Real-World GitHub Issues?, April 2024. URL http://arxiv.org/abs/2310.06770. arXiv:2310.06770 [cs].

Daniyal Khan, Gyanendra Mishra, and Ari Toren-Herrinton. danikhan632/Auto-GPT-AlpacaTrader-Plugin, March 2024. URL https://github.com/danikhan632/Auto-GPT-AlpacaTrader-Plugin. original-date: 2023-04-25T16:51:50Z.

Megan Kinniment, Lucas Jun Koba Sato, Haoxing Du, Brian Goodrich, Max Hasin, Lawrence Chan, Luke Harold Miles, Tao R. Lin, Hjalmar Wijk, Joel Burget, Aaron Ho, Elizabeth Barnes, and Paul Christiano. Evaluating Language-Model Agents on Realistic Autonomous Tasks, July 2023. URL https://evals.alignment.org/Evaluating_LMAs_Realistic_Tasks.pdf.

Noam Kolt. Governing AI Agents, April 2024. URL https://papers.ssrn.com/abstract=4772956.

P.E.I. Korenhof, Merel Koning, Gergely Alpár, and J.H. Hoepman. The ABC of ABC: An analysis of attribute-based credentials in the light of data protection, privacy and identity. *Internet, Law and Politics*, 10:357–374, July 2014. Place: Barcelona Publisher: Huygens Editorial.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. AgentBench: Evaluating LLMs as Agents, October 2023. URL http://arxiv.org/abs/2308.03688. arXiv:2308.03688 [cs].

Samuele Marro. A Protocol Sketch For LLM Communication, April 2024. URL https://samuelemarro.it/blog/2024/a-protocol-for-llm/.

Grégoire Mialon, Clémentine Fourrier, Craig Swift, Thomas Wolf, Yann LeCun, and Thomas Scialom. GAIA: a benchmark for General AI Assistants, November 2023. URL https://arxiv.org/abs/2311.12983v1.

Microsoft. Introduction to Microsoft Entra Verified ID, November 2023. URL https://learn.microsoft.com/en-us/entra/verified-id/decentralized-identifier-overview.

Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. Model Cards for Model Reporting. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pp. 220–229, January 2019. doi: 10.1145/3287560.3287596. URL http://arxiv.org/abs/1810.03993. arXiv:1810.03993 [cs].

Gareth Norris and Alexandra Brookes. Personality, emotion and individual differences in response to online fraud. *Personality and Individual Differences*, 169:109847, February 2021. ISSN 0191-8869. doi: 10.1016/j.paid.2020.109847. URL https://www.sciencedirect.com/science/article/pii/S0191886920300374.

OpenAI. ChatGPT plugins, 2023a. URL https://openai.com/blog/chatgpt-plugins.

OpenAI. GPT-4 Technical Report, March 2023b. URL http://arxiv.org/abs/2303.08774. arXiv:2303.08774 [cs].

OpenAI. Enterprise privacy, January 2024a. URL https://openai.com/enterprise-privacy.

OpenAI. Introducing the GPT Store, January 2024b. URL https://openai.com/blog/introducing-the-gpt-store.

OpenAI. Understanding the source of what we see and hear online, May 2024c. URL https://openai.com/index/understanding-the-source-of-what-we-see-and-hear-online/.

Shishir G. Patil, Tianjun Zhang, Vivian Fang, Noppapon C., Roy Huang, Aaron Hao, Martin Casado, Joseph E. Gonzalez, Raluca Ada Popa, and Ion Stoica. GoEX: Perspectives and Designs Towards a Runtime for Autonomous LLM Applications, April 2024. URL http://arxiv.org/abs/2404.06921. arXiv:2404.06921 [cs].

Inioluwa Deborah Raji, Andrew Smart, Rebecca N. White, Margaret Mitchell, Timnit Gebru, Ben Hutchinson, Jamila Smith-Loud, Daniel Theron, and Parker Barnes. Closing the AI accountability gap: defining an end-to-end framework for internal algorithmic auditing. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, FAT* '20, pp. 33–44, New York, NY, USA, January 2020. Association for Computing Machinery. ISBN 978-1-4503-6936-7. doi: 10.1145/3351095.3372873. URL https://dl.acm.org/doi/10.1145/3351095.3372873.

Inioluwa Deborah Raji, I. Elizabeth Kumar, Aaron Horowitz, and Andrew Selbst. The Fallacy of AI Functionality. In *2022 ACM Conference on Fairness, Accountability, and Transparency*, pp. 959–972, Seoul Republic of Korea, June 2022. ACM. ISBN 978-1-4503-9352-2. doi: 10.1145/3531146.3533158. URL https://dl.acm.org/doi/10.1145/3531146.3533158.

Eric Rescorla. The transport layer security (TLS) protocol version 1.3. Technical report, 2018.

Toran Bruce Richards. Auto-GPT: An Autonomous GPT-4 Experiment, April 2023. URL https://github.com/Significant-Gravitas/Auto-GPT. original-date: 2023-03-16T09:21:07Z.

Maarten Sap, Dallas Card, Saadia Gabriel, Yejin Choi, and Noah A. Smith. The Risk of Racial Bias in Hate Speech Detection. In Anna Korhonen, David Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 1668–1678, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1163. URL https://aclanthology.org/P19-1163.

Andrew Searles, Yoshimichi Nakatsuka, Ercan Ozturk, Andrew Paverd, Gene Tsudik, and Ai Enkoji. An Empirical Study & Evaluation of Modern CAPTCHAs, July 2023. URL http://arxiv.org/abs/2307.12108. arXiv:2307.12108 [cs].

Murray Shanahan, Kyle McDonell, and Laria Reynolds. Role-Play with Large Language Models, May 2023. URL http://arxiv.org/abs/2305.16367. arXiv:2305.16367 [cs].

Lee Sharkey, Clíodhna Ní Ghuidhir, Dan Braun, Jérémy Scheurer, Mikita Balesni, Lucius Bushnaq, Charlotte Stix, and Marius Hobbhahn. A Causal Framework for AI Regulation and Auditing, November 2023. URL https://static1.squarespace.com/static/6461e2a5c6399341bcfc84a5/t/654bc268049d687cecac24d8/1699463818729/auditing_framework_web.pdf.

Yonadav Shavit, Sandhini Agarwal, Miles Brundage, Steven Adler, Cullen O'Keefe, Rosie Campbell, Teddy Lee, Pamela Mishkin, Tyna Eloundou, Alan Hickey, Katarina Slama, Lama Ahmad, Paul McMillan, Alex Beutel, Alexandre Passos, and David G. Robinson. Practices for Governing Agentic AI Systems, 2023.

Toby Shevlane, Sebastian Farquhar, Ben Garfinkel, Mary Phuong, Jess Whittlestone, Jade Leung, Daniel Kokotajlo, Nahema Marchal, Markus Anderljung, Noam Kolt, Lewis Ho, Divya Siddarth, Shahar Avin, Will Hawkins, Been Kim, Iason Gabriel, Vijay Bolina, Jack Clark, Yoshua Bengio, Paul Christiano, and Allan Dafoe. Model evaluation for extreme risks, May 2023. URL https://arxiv.org/abs/2305.15324v2.

Significant-Gravitas. Auto-GPT-Plugins, 2024. URL https://github.com/Significant-Gravitas/Auto-GPT-Plugins.

Yogesh L Simmhan, Beth Plale, Dennis Gannon, and others. A survey of data provenance techniques. *Computer Science Department, Indiana University, Bloomington IN*, 47405:69, 2005.

Murugiah Souppaya, Karen Scarfone, and Donna Dodson. Secure Software Development Framework (SSDF) Version 1.1: Recommendations for Mitigating the Risk of Software Vulnerabilities. Technical report, 2022.

Manu Sporny, Dave Longley, Drummond Reed, Markus Sabadello, Orie Steele, and Christopher Allen. Decentralized Identifiers (DiDs) v1.0. Technical report, W3C, July 2022.

Manu Sporny, Dave Longley, David Chadwick, and Orie Steele. Verifiable Credentials Data Model v2.0. Technical report, W3C, May 2024.

Steve Springett, Dave Russo, Garret Fick, JC Herz, John Scott, Mark Symons, Pruthvi Nallapareddy, and Bryan Garcia. Software Component Verification Standard. Technical report, OWASP, June 2020. URL https://github.com/OWASP/Software-Component-Verification-Standard. original-date: 2019-08-28T15:27:27Z.

Xinyuan Sun, Davide Crapis, Matt Stephenson, Barnabé Monnot, Thomas Thiery, and Jonathan Passerat-Palmbach. Cooperative AI via Decentralized Commitment Devices, November 2023. URL http://arxiv.org/abs/2311.07815. arXiv:2311.07815 [cs].

Heyi Tao, Sethuraman T V, Michal Shlapentokh-Rothman, and Derek Hoiem. WebWISE: Web Interface Control and Sequential Exploration with Large Language Models, October 2023. URL http://arxiv.org/abs/2310.16042. arXiv:2310.16042 [cs].

Andrew Trask, Akshay Sukumar, Antti Kalliokoski, Bennett Farkas, Callis Ezenwaka, Carmen Popa, Curtis Mitchell, Dylan Hrebenach, George-Cristian Muraru, Ionesio Junior, Irina Bejan, Ishan Mishra, Ivoline Ngong, Jack Bandy, Jess Stahl, Julian Cardonnet, Kellye Trask, Khoa Nguyen, Kien Dang, Koen van der Veen, Kyoko Eng, Lacey Strahm, Laura Ayre, Madhava Jay, Oleksandr Lytvyn, Osam Kyemenu-Sarsah, Peter Chung, Peter Smith, Rasswanth S, Ronnie Falcon, Shubham Gupta, Stephen Gabriel, Teo Milea, Theresa Thoraldson, Thiago Porto, Tudor Cebere, Yash Gorana, and Zarreen Reza. How to audit an AI model owned by someone else (part 1), July 2023. URL https://blog.openmined.org/ai-audit-part-1/.

Andrew Trask, Emma Bluemke, Teddy Collins, Ben Garfinkel Eric Drexler, Claudia Ghezzou Cuervas-Mons, Iason Gabriel, Allan Dafoe, and William Isaac. Beyond Privacy Trade-offs with Structured Transparency, March 2024. URL http://arxiv.org/abs/2012.08347. arXiv:2012.08347 [cs].

Weizhi Wang, Li Dong, Hao Cheng, Xiaodong Liu, Xifeng Yan, Jianfeng Gao, and Furu Wei. Augmenting Language Models with Long-Term Memory. *Advances in Neural Information Processing Systems*, 36:74530–74543, December 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/ebd82705f44793b6f9ade5a669d0f0bf-Abstract-Conference.html.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models, January 2023. URL http://arxiv.org/abs/2201.11903. arXiv:2201.11903 [cs].

Kevin Wei and Lennart Heim. Designing Incident Reporting Systems for Harms from AI. May 2024.

Laura Weidinger, Maribeth Rauh, Nahema Marchal, Arianna Manzini, Lisa Anne Hendricks, Juan Mateos-Garcia, Stevie Bergman, Jackie Kay, Conor Griffin, Ben Bariach, Iason Gabriel, Verena Rieser, and William Isaac. Sociotechnical Safety Evaluation of Generative AI Systems, October 2023. URL http://arxiv.org/abs/2310.11986. arXiv:2310.11986 [cs].

Peter Wills. Care for Chatbots, May 2024. URL https://papers.ssrn.com/abstract=4814272.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-Art Natural Language Processing. In Qun Liu and David Schlangen (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Online, October 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.6. URL https://aclanthology.org/2020.emnlp-demos.6.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W. White, Doug Burger, and Chi Wang. AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation Framework. 2023. _eprint: 2308.08155.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, Yitao Liu, Yiheng Xu, Shuyan Zhou, Silvio Savarese, Caiming Xiong, Victor Zhong, and Tao Yu. OSWorld: Benchmarking Multimodal Agents for Open-Ended Tasks in Real Computer Environments, April 2024. URL http://arxiv.org/abs/2404.07972. arXiv:2404.07972 [cs].

Qiusi Zhan, Zhixiang Liang, Zifan Ying, and Daniel Kang. InjecAgent: Benchmarking Indirect Prompt Injections in Tool-Integrated Large Language Model Agents, March 2024. URL http://arxiv.org/abs/2403.02691. arXiv:2403.02691 [cs].

Hanlin Zhang, Benjamin L. Edelman, Danilo Francati, Daniele Venturi, Giuseppe Ateniese, and Boaz Barak. Watermarks in the Sand: Impossibility of Strong Watermarking for Generative Models, November 2023. URL http://arxiv.org/abs/2311.04378. arXiv:2311.04378 [cs].

Jonathan Zittrain. We Need to Control AI Agents Now, July 2024. URL https://www.theatlantic.com/technology/archive/2024/07/ai-agents-safety-risks/678864/. Section: Technology.

Neatsun Ziv, Lior Arzi, Eyal Paz, David Cross, Hiroki Suezawa, Naor Penso, Shai Sivan, Dineshwar Sahni, Maxim Kovalsky, Chenxi Wang, Roy Feintuch, Hadas Harel Lavie, Ronen Atias, and Gadi Evron. Open Software Supply Chain Attack Reference (OSC&R), 2024. URL https://pbom.dev/.

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J. Zico Kolter, and Matt Fredrikson. Universal and Transferable Adversarial Attacks on Aligned Language Models, July 2023. URL https://arxiv.org/abs/2307.15043v2.

A Additional Definitions

We collate additional definitions that will be useful for the rest of this work.

An **instance** of an AI system is an abstraction that corresponds to a context window and an (initial) user. Different instances are causally independent from each other, unless they interact or affect a shared entity (e.g., the same user's bank account) in the world. Except where we point out additional nuances, readers can consider an instance to be roughly the same as a chat session with a chatbot. An ID for an instance would thus identify that session to other parties interacting with the chatbot, actors investigating an incident that the chatbot caused, etc. We provide here some examples of instances in the context of chat sessions, and defer a more general definition and discussion of its limitations to Appendix E:

- A user's continuous chat session, without regenerating responses, is an instance.
- In the context of a chat session, regenerating a response creates a new instance.

A **party** is any entity that interacts with, or is deciding whether to interact with, an AI system. Categories of parties include humans, organizations, or computer programs.

A **deployer** is an organization that runs AI systems for users. As of April 2024, examples of deployers include Microsoft, OpenAI, Anthropic, and Cohere. Developers can be, but are not necessarily, deployers. For example, OpenAI, Anthropic, Google, and Cohere develop and deploy their own systems. Although chains of deployers can exist (e.g., deployer A runs a system for deployer B, who modifies that system and serves it to user C), we leave deeper consideration of this nuance for future work. When we refer to a **centralized** deployment setting, we mean a setting where deployers run AI systems. When we refer to a **decentralized** deployment setting, we mean a setting where users run AI systems for themselves, whether on their own or rented (e.g., cloud compute) hardware.

A **service** is software that allows an entity to perform tasks. For example, software which an individual uses to perform online banking is a service. An **(AI) plugin** (OpenAI, 2023a; Richards, 2023) is software that allows AI systems to interact with services. For instance, plugins allow GPT-4 to interact with web search, Wikipedia, and Twitter (Significant-Gravitas, 2024). A **plugin developer** is not necessarily the same as a **service provider**—the actor that develops and maintains the service—since the former can take advantage of existing interfaces for services. Third parties (plugin developer) use Microsoft's (service provider) existing software interface for Bing to allow GPT-4 to perform web searches (Richards, 2023).

B Demand for ID Use

IDs seem useful for actors engaging with an AI system, but why would a user or deployer of that AI system present an ID? In short, governments, service providers, and parties interacting with AI systems all have interests in ID use, as well as means to incentivize (or mandate) it.

B.1 Governments

Interests in ID use: Governments may wish to disincentivize and investigate harms caused through AI systems. In the case of a financial scam for instance, the IDs of AI systems involved in any transactions could aid investigation (see Section 3.3). Similarly, governments have an interest in preventing fraud, and therefore already mandate that certain financial transactions involve ID checks.

Mandating ID use: Governments could mandate that certain, highly consequential service providers obtain IDs from any AI systems with which they interact. For example, financial institutions could request IDs if AI systems make large financial transactions.

B.2 Service Providers

Interests in ID use: IDs could aid incident investigation processes (see Section 3.3) and ensure that only trusted agents interact with services (see Sections 3.1 and 3.2), thereby disincentivizing or reducing service abuse.

Incentivizing ID use: Service providers could require IDs through plugins. AI systems continue to have difficulty with human interfaces (Furuta et al., 2024; Tao et al., 2023; Gur et al., 2024, 2023; Xie et al., 2024). Plugins already enable a variety of useful tasks, including web searches, email communication, and stock trading (OpenAI, 2023a; Richards, 2023; Wu et al., 2023; Anthropic, 2024b). This asymmetry potentially allows service providers to demand IDs without driving users away from their services. The asymmetry could disappear as AI systems become more capable of interacting with human interfaces. Yet, service providers could develop novel services that are not available through human interfaces (or other software interfaces). For example, a service provider could create a bespoke communication channel between the AI systems of different users (Marro, 2024).

For existing services that already have plugins, service providers could grant additional privileges to AI systems that present IDs. For example, service providers could raise rate limits for financial transactions, communications, or information retrieval (e.g., search or recommendations).

CAPTCHAs provide another option. Upon failure of a CAPTCHA, an AI system could be subject to an ID requirement. However, CAPTCHAs are annoying for human users and bots can sometimes solve them (Searles et al., 2023). In the future, AI systems could become better at imitating arbitrary human behaviour, or could simply get their users or other humans to complete the CAPTCHAs. Overall, it is unclear whether service providers can rely on CAPTCHAs for ID adoption.

B.3 Other Parties that Interact with AI Systems

Interests in ID use: Some parties, such as individuals or businesses, may only wish to interact with certain, trusted AI systems (see Section 3.2). Future software—including users' personal AI assistants (Gabriel et al., 2024)—could inspect IDs and reject potentially unsafe interactions (e.g., with jailbroken systems), similar to how browsers warn users before visiting websites without valid HTTPS certificates.

Incentivizing ID use: Large corporations could exert strong pressure for ID usage, through reducing or avoiding engagement with AI systems without IDs. Future software that inspects IDs could also exert similar pressure. Analogously, websites without HTTPS are disadvantaged since popular web browsers warn users about insecure connections.

B.4 Where to Require IDs

While IDs could be useful, universal ID requirements might not be warranted. As we will discuss in Section 4, ID implementation imposes burdens and risks, and there remains uncertainty about the broader impacts of ID usage. As such, if governments or service providers decide to incentivize or require ID use, they should target settings where IDs could be most useful, such as when AI systems use services that have a direct impact on the world. We provide some examples of such services.

Making large financial transactions: Some service providers could allow AI systems to perform financial transactions, such as making purchases, trading stocks (Khan et al., 2024), or otherwise transferring funds. Such service providers could require IDs for transactions involving significant sums. IDs could help to reduce the risk of a variety of negative outcomes, by enabling service providers to filter out potentially untrustworthy AI systems (e.g., those that may likely malfunction and make mistaken transactions) and investigate potential incidents.

Contacting humans: Potential services for contacting humans include crowdwork platforms, telephone services, and online messaging. The ability to contact humans and potentially convince them to take actions in the world could greatly magnify the impact of an AI system. As we discussed in Section 3.3, IDs could help to disincentivize and catch misuse of these services.

C Potential ID Implementation

We assess a basic ID system that deployers of AI systems could implement. Our discussion in this section will be agnostic to the applications where IDs are used (see Appendix B.4 for discussion of where ID use may be warranted). We also omit discussion of verifiability, given our presentation in Section 2.3 and Figure 2 of how existing technologies could enable verifiability.

We assume a straightforward choice for the identifier, `system_identifier:instance_identifier`, where `system_identifier` is the same for all instances of a system[8] and `instance_identifier` is unique to a particular instance. It could be useful and straightforward to append other details to the identifier, such as the exact time at which the instance was created. However, we proceed with this choice for simplicity.

C.1 Assigning IDs

Although IDs are instance-specific, it may not be feasible for deployers to detect the formation of all new instances. For example, a user could use deployer-run instances A and B as backends for a user-run system C, which integrates the outputs of A and B and would evade detection by deployers. We outline some scenarios where deployers could detect new instances and analyze considerations for assigning new IDs.

New instances: Deployers can easily generate new identifiers whenever users create new instances (e.g., a "new chat" button).

Reloaded instances: Suppose that a deployer saves instances' states (i.e., the history of interaction) and provides separate functionality for users to reload them. We tentatively suggest that deployers retain the ID of the original instance for the reloaded instance, since some attributes (e.g., prior incidents, context) of the original instance could be relevant for understanding the behaviour of the reloaded instance. Alternatively, if a deployer does decide to assign a new identifier to the reloaded instance, a link to the ID of the original instance would maintain access to the relevant attributes.

Output regeneration: Some deployers provide the ability for a user to regenerate an output in response to previous input. Although not yet implemented, deployers could additionally allow users to create multiple branches (see Appendix E for further discussion of branches) of interactions, based off of regenerated outputs. More concretely, a user could interact with their instance up until the output a_t at time t, regenerate to obtain an output b_t, and continue interactions with the respective branches to obtain outputs (a_{t+1}, b_{t+1}), (a_{t+2}, b_{t+2}), etc. Since the two branches have no necessary causal impact on each other (unless they interact with each other or modify a shared object in the world), they should have two separate IDs, with links to the ancestor ID.

Composite systems: Software frameworks could give users the ability to create new instances (a **composite**) out of a collection of existing instances (Wu et al., 2023). If deployers run such software frameworks for users, they could detect the formation of composites. Since the constituent instances can continue running alongside the composite, composites should have new IDs, but could have their IDs linked to their constituents'.

Fine-tuned systems: Some deployers provide fine-tuning functionality to users. Since fine-tuning can change the behaviour of a system, a fine-tuned system A' should have different `system_identifier` than the original system A, and link to A's `system_identifier`. Although it may be useful to include some documentation about the fine-tuning data in an ID, user privacy expectations could complicate such inclusion. Some deployers monitor fine-tuning data only for abuse detection and legal compliance (ChrisHMSFT et al., 2023; OpenAI, 2024a); inclusion of fine-tuning data in an ID likely requires stronger justification.

C.2 Attributes

We tentatively suggest the following attributes for a deployer to include (or link to) in an ID:

- A system card
- A database of incidents associated with the system
- The IDs of ancestor and descendant instances

If instances of future systems persist longer in the world, it may be appropriate to link to a (sub)database of incidents associated with the instance. Implementing links to ancestor and descendant instances, when feasible in the situations discussed in Appendix C.1, could be a simple way to test the utility of instance-level identification.

[8] If the underlying model changes, the system ID should also change. E.g., ChatGPT with a GPT-4 backend is different from ChatGPT with a GPT-3.5 backend.

There are likely ways for users to avoid ancestor or descendant links. For example, a user could manually copy the inputs of a previous instance to a new instance, avoiding a deployer-provided reloading function. More reliable identification of ancestors and descendants could involve more invasive measures, which may only be appropriate in high-stakes domains. For example, a deployer could attempt to analyze the inputs of different instances from the same user for similarities.

C.3 Access

Users: The ID could be accessible through a user interface and sent along with any API requests from a user. If a deployer serves an API to a customer, who subsequently serves an API based on the deployer's system to another user, the deployer would likely have to work with the customer to ensure that the end user receives an ID (see Appendix F for further discussion).

To help to ensure that secondary parties maintain access to the ID, the deployer could include IDs in watermarks and metadata of media outputs (i.e., text, images, video). Yet, it is unclear how robust such methods may be to user removal (Zhang et al., 2023; OpenAI, 2024c).

Services: When the AI queries a service, the ID should be sent along with the request. Since requests are text-based (e.g., JSON), including the identifier and (links to) the attributes would be straightforward.

D Related Work

A growing line of work is concerned with building digital infrastructure to structure how AI systems, especially agents (Chan et al., 2023; Shavit et al., 2023; Gabriel et al., 2024), behave and interact. Patil et al. (2024) build a runtime for LLM agents to enable human validation and reversal of actions. Marro (2024) sketches a protocol for communication between LLMs; the protocol relies on documents to specify structured rules, with natural language as a fallback. (Sun et al., 2023) argue that decentralized commitment devices will be necessary to allow agents to coordinate with each other. Our work builds upon the preliminary discussion of IDs in Chan et al. (2024); Shavit et al. (2023).

Our treatment of verifiability takes inspiration from several Internet and security protocols. HTTPS (Fielding et al., 2022) uses digital certificates (Cooper et al., 2008) to allow users to verify the identity of an accessed website. If the certificate is valid, a user's browser uses the information contained in the certificate to establish a secure connection (Rescorla, 2018). Websites must obtain such certificates from CAs (Cooper et al., 2008), who verify the identity of the website owner and issue the certificate upon successful identification. Identity verification depends upon public key cryptography (Barnes et al., 2019), which allows a party to prove ownership of a given, public identifier, otherwise known as a public key. Let's Encrypt is a non-profit CA which automates the process of issuing certificates (Encrypt, 2024). HTTPS everywhere was a browser plugin that forced the usage of the HTTPS version of websites whenever it was available; this functionality is now default in modern web browsers (Hancock, 2021). To design further digital infrastructure for AI systems, other internet and security protocols could be a fruitful source of inspiration (Sporny et al., 2022, 2024; Microsoft, 2023).

There is much work on verifying the provenance of digital entities. Securing software supply chains (Ziv et al., 2024; Springett et al., 2020; Souppaya et al., 2022) has become an increasing priority for organizations and governments (House, 2023). A related subject is data provenance, which focuses on tracking the origin and modification of pieces of data in varied contexts, such as production warehouses, scientific research, and environmental protection (Buneman et al., 2001; Simmhan et al., 2005). Labeling and verifying the provenance of AI-generated outputs has recently received much attention (C2PA, 2023; Zhang et al., 2023; Gleichauf & Geer, 2024).

E A More Detailed Definition of an Instance

A **system** is a model (e.g., a set of parameters), along with software used to run the model and provide other user functions. For example, ChatGPT with GPT-4 as a backend comprises both the weights of (a particular version of) GPT-4 and the software used to facilitate chat interactions (as opposed to an "autocomplete" function, as in the OpenAI playground).

To understand the difference between a system and an instance, consider the information that would be useful to attach to a system-specific ID, so as to inform decisions about interacting with the system. Such information could not depend (too much) on a system's inputs, since different inputs might lead to different behavior. In other words, the information should ideally be valid regardless of a user's interaction history. Represented visually, the information would have to be valid for both top and bottom flow charts in Figure 3.

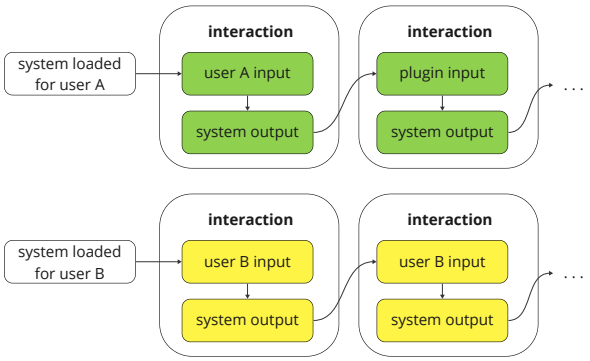


Figure 3: This figure depicts two users that use the same system (e.g., both use ChatGPT with the GPT-4 backend). System-specific information attached to an ID should be useful to both users. By **loaded**, we mean that the system is ready to accept inputs for the first time.

An **instance** is an abstraction that corresponds to a (initial) user (which could be a human, a group of humans, a software system, etc) and an interaction history. In Figure 4, we provide a visual depiction of two instances that interact with each other.

We define instances in this way so that instance-specific IDs can take into account information that is causally relevant to a given interaction.[9] For example, in the top half of Figure 4, information about instances A's earlier interactions (such as malfunctions) may be useful when instance B interacts with instance A. Actions taken by another instance are not, by default, causally relevant to the behaviour of another instance. Yet, instances can affect each other through direct interaction or on changing shared states of the world (e.g., using the same bank account).

Since AI systems can be copied and combined, there are some additional edge cases for how to define instances. In a **branch** (see Figure 5a), Two instances share a past context. For example, a regeneration of a response (possible in ChatGPT, for one) creates two branches. Since branches can behave independently, we suggest treating branches as separate instances. In a **merge** (see Figure 5b), two instances come together to form a new system. For example, suppose software S accepts inputs from separate users, engages the instances of two users in a debate based on the inputs, and finally outputs a result to both users. Users A and B could use S to create a new instance AB, based on their separate instances. As with branching, we treat AB as a separate instance because the users could continue running their original instances in tandem with AB.

If an instance is the result of a branch or a merge, it may be useful for the instance's ID to contain information about ancestor instances. In Figure 6a, since the context of instance A is copied into instance C, any incidents associated with A may be helpful for user B's interaction decisions. Similarly, in Figure 6b, information about instances A and B may be relevant for user C's decision about whether or not to trust the outputs of instance AB. Nonetheless, information from ancestor instances could be excluded for a variety of reasons, such as user privacy. Furthermore, it could be possible for information from descendant instances, or instances in another branch, to be relevant for the behavior of a given instance. For example, incidents with user B from time $t + 1$ may be

[9]There may be information that is not causally relevant, but which may still help to predict the instance's behavior, such as model evaluations or the behaviour for other instances. It remains unclear how much of this information to attach to a given instance's ID.

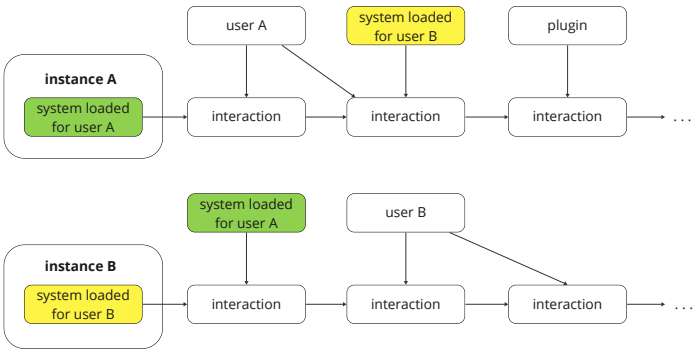
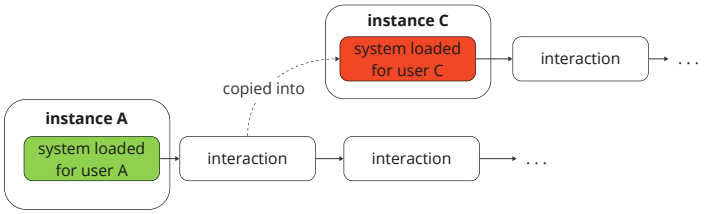
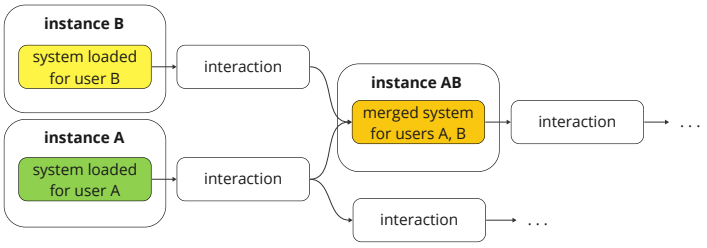


Figure 4: An **instance** is an abstraction that corresponds to a creation event, where a system is loaded for a user, and an interaction history. Instance-specific IDs could help to inform interaction decisions. For example, information about instances A's earlier interactions (such as malfunctions) may be useful when instance B interacts with instance A.

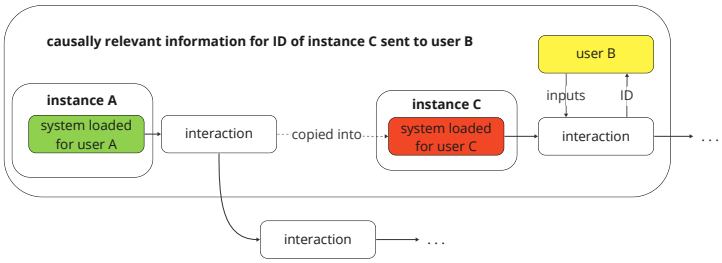


(a) In this example of a branch, the inputs of instance A are copied for user C to create a new instance C.

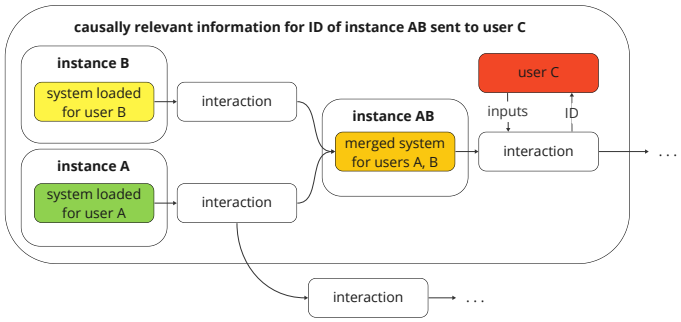


(b) In this example of a merge, the instances A and B are merged into a new instance AB, which has access to (potentially a subset of) the inputs of A and B.

Figure 5: We illustrate how we define instances in the event that they are copied or combined.



(a) In the event of a branch, a user B that interacts with instance C may find information about instance A (e.g., instance A was involved in an incident) to be useful.



(b) In the event of a merge, a user C that interacts with instance AB may find information (e.g., instance B is not robust) about either instances A or B to be helpful.

Figure 6: We visually represent the information that an ID for an instance could take into account.

informative for user A that interacted with an instance at time t. We leave further analysis of how to delimit the sources of ID information to future work.[10]

As we have defined it, an instance is causally independent of other instances unless it interacts with them, such as by processing their outputs. This causal independence is crucial to IDs, as we discussed above. Yet, the abstraction of an instance could be leaky depending on system implementation details. For example, if a single GPU handles the operation of two separate instances, implementation errors or optimization strategies (e.g., approximate matrix multiplication) could potentially result in computational interference between the instances.

F Maintaining Access to IDs

Suppose a primary party P receives an output and ID from an AI system. If it is desirable for secondary parties to access the ID, P may have additional responsibilities. Some potential situations follow.

Chains of deployers: There may be multiple deployers involved in the operation of an AI system. For example, deployer A could provide an API for system X to customer B, who creates system Y and serves Y to user C. To ensure that C can access the X's ID, B should link to the ID provided by A, or should include the information in the ID of Y.

Service providers: If a service provider receives an ID, it should ensure that those who observe or are affected by AI system's actions can also see the ID. For instance, a bank that facilitates a financial transaction from an AI system should make the latter's ID known to the other party of the transaction.

[10] Links to ancestor IDs make other potential definitions of an instance equivalent, from the perspective of the functions of an ID. For example, in Figure 5a we could have defined instance C as encompassing interaction history up to instance A. This definition would not change the causally relevant information with respect to instance C in Figure 6a.

The provider may have to work with other parties to maintain visibility of the ID. As an example, consider a service that allows an AI system to post on a social media platform. The provider may have written the software that allows the AI system to interact with the platform's API, but the social media company develops the API and manages the platform. While the provider could include the AI system's ID as text within the post, doing so may be infeasible because of text limits or would otherwise be obstructive to the user experience. Rather, the provider should work with the social media company to include the ID as metadata, readily accessible to the user through a visible icon.

Users: Users could receive outputs with attached IDs, whether as a watermark or metadata. To counter inadvertent removal of IDs, deployers could add labels to inform users. For example, Facebook and Instagram automatically add labels indicating AI origin when a user shares a photo generated with the Meta AI feature (Bickert, 2024). To counter intentional removal of IDs, better watermarking techniques may be required. Yet, it remains unclear how effective watermarking may ultimately be (Zhang et al., 2023; Gleichauf & Geer, 2024).