INFORMAL AND CURSORY NOTE REGARDING AGENTIC CONTEXT PROTOCOL (ACP) LEGAL RISKS

James Skelley | E: james@jstechlaw.com: P: +1 (650) 281-2828

The observations made herein are informal and provided merely to apprise the reader of relevant issues and should not be construed as legal advice. Accordingly, you should NOT rely upon anything presented herein without consulting with an attorney. Neither will this document establish any client, confidentiality, or other relationship between James Skelley and the reader or any other party. Finally, for clarity, the subject matter discussed in this note is in considerable flux, and so readers will appreciate that terminology and characterizations appearing herein may have been used or presented differently in the past and may be used or presented differently in the future.

I. INTRODUCTION / MOTIVATION

Attending several Agentic Context Protocol (ACP) events hosted recently (April/May 2025) in Silicon Valley (discussing, e.g. the Massachusetts Institute of Technology™'s NANDA™, Anthropic™'s MCP™, etc.), there seemed to be a general lack of awareness of the dangers latent in blithely deploying many agent AIs to perform various collective actions. Specifically, participants of these events seemed to conflate AI agents with "persons" recognized in law, thereby overlooking the legal liability that may accrue to the agents' respective operators. Ideally, ACPs would instead anticipate the operators' respective legal postures in the protocols themselves, e.g., as presently occurs in various open source package repositories where license presentation is required in order to acquire the repository.

This summary provides a very brief introduction to ACPs (Section II) and some of their potential legal issues (Section III), as well as a brief overview of relevant existing legal precedent (Section IV).

II. TECHNICAL ORIENTATION

This section briefly describes large language models (LLMs), AI agents, and ACPs.

II.A WHAT ARE LLMS?

Large Language Models (LLMs) are neural networks trained to recognize statistical patterns in human text. Where their size facilitates storage of many hierarchic correlations (between words, groups of words, and groups of texts, etc.), LLMs can imitate the responses humans might give to a natural language prompt, albeit as limited by correlations discernable from the training data. Because LLMs ultimately rely merely upon statistical correlations for their operation, their failures can be unexpected and sometimes dramatic, as when an LLM "hallucinates" statistically coherent, but physically incoherent, textual responses (e.g., queries involving causality that I've posed to some LLMs will often return incorrect responses if the query is phrased in a manner unlikely to have been encountered during training since text alone cannot fully reflect causal relations, i.e., the set of term relations in preexisting text available for training has a

lower cardinality than the set of causal object relations appearing in physical reality[1]).

II.B What is "Agentic AI"?

While the terminology is presently often in flux[2], for legal purposes, and as used herein, Agentic AIs, AI Agents, etc., generally refer to software systems availing themselves of large neural networks (often LLMs) when fulfilling various functions and which are managed by a legal entity, referred to herein as the "operator" of the agent (the term is also used herein for entities controlling servers offering agent accessible functionality). For example, as indicated in FIG. 1 below, a "restaurant reservation" AI agent may be a combination of software logic and an LLM configured to receive natural language instructions and queries from a user, interact with a service provider in accordance with the queries / instructions, and to then provide confirmation back to the user.

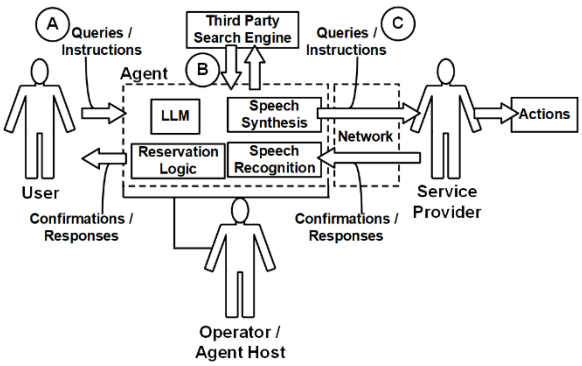

FIG. 1: Example Agent Behavior

Here, the **User** may request **A** that the **Agent** determine if a reservation at a restaurant is possible at a specified time and to schedule the reservation if so. The **Agent** may in turn, consult an internet search engine **B** to determine the restaurant's contact information using a natural language search query, then contact the maître d' (**Service Provider**) of the restaurant, engaging in a natural language voice interaction to verify and make the reservation, before reporting success or failure back to the **User**.

Importantly, note that the **Operator / Agent Host** in this diagram is the legal entity running/hosting the **Agent**, typically exercising ownership via control of the **Agent**'s executing platform. Thus, e.g., where the **Agent** is a service operated upon a cloud server computer system, then the **Operator** is the entity maintaining the **Agent** upon that server (i.e., exercising exclusionary control over the **Agent**). Alternatively, where the **User** has purchased the **Agent** for execution on, e.g., their smartphone, the **User** may be the same legal entity as the **Operator**.

II.C What are Agentic Context Protocols (ACPs)?

ACPs are protocols for informing AI agents of various resources and services available for the AI agent to fulfill its tasks. Examples include Anthropic™'s MCP™[3] and the Massachusetts Institute of Technology™'s NANDA™.[4] While several such protocols are under active development, they generally follow the client server topology of FIG. 2

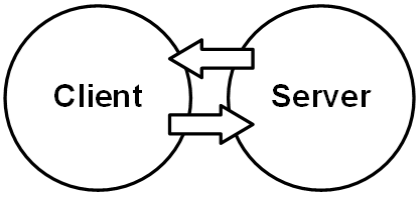

FIG. 2: Basic ACP

These protocols specify formats for informing the AI agent **Client** of a **Server** (which may or may not itself be an AI agent) and the **Server**'s resources. For example, the **Server** may be managed by a grocery chain, making available the current grocery store inventory by store location, and may publish this resource so that **Client** may avail itself of that information when responding to a natural language query (e.g., "Can I bake a cake from what's on sale at Main Street Market this week?"). In some protocols, the **Client** is explicitly advised of **Server** and its resources via a local JSON listing manually provided by the operator of **Client** (e.g., in MCP™, specifying the "mcpServers" JSON key in the client's configuration[5]).

III. Risks of ACP Legal Delegation

Naturally, combining high speed automation with hallucination prone LLMs to effect real world consequences invites the potential for legally colorful fact patterns.

Previously, many online platforms have recognized the need to establish the legal postures between parties before permitting their interaction (consider, e.g., the License.MD generated upon creation of a GitHub™ repository and which will appear prominently in any downloaded copy of the repository[6]). However, at least as indicated by the April/May events and the current protocol online documentation, many ACP protocols do not yet appear to include functionality for specifying legal postures, even though something as simple as a JSON keyword specifying terms of use would mitigate many of the issues discussed herein. Rather, ACPs instead presently appear to delegate *the entirety* of legal posture formation (whether explicitly or implicitly) to the implementors themselves. For example, the online MCP™ documentation as of 05/15/2025 recites:

> The Model Context Protocol enables powerful capabilities **through arbitrary data access and code execution paths**. With this power comes important security and trust considerations that **all implementors must carefully address** . . . Users **must explicitly consent to** and **understand all data access and operations.**" (emphasis added)[7]

As evidenced by GitHub™, these risks are not uniquely inherent to ACPs. Any time a protocol delegates responsibility for implementing core functionality to an implementor, unexpected results and inefficiencies (both technical and legal) are likely to follow. These inefficiencies in turn disrupt network effects that might otherwise facilitate adoption.

For example, computer network developers having worked with the Transmission Control Protocol (TCP), which ensures reliable and ordered packet delivery, and the User Datagram Protocol (UDP), which does not, will readily appreciate the dangers inherent to blithe utilization of the latter in disparate interacting contexts. For example, even when one developer successfully implements their own UDP solution, their manner of packet handling may not be compatible with another developer's handling solution. Because TCP instead enforces a specific handling methodology in all instances, the consistency facilitates growth along other dimensions. Thus, unsurprisingly, much of the Internet backbone eschews UDP in favor of TCP.

Similar to UDP's delegation of packet management, an ACP's delegation of legal posture formation likewise invites inefficiencies that may disrupt adoption. Just as each UDP implementation has its own ad hoc package management solution, in delegating ACPs, each pair of interacting ACP entities will have their own ad hoc legal posture. While this is true even for the above-discussed examples and topologies, such ad hoc discrepancies become *even more* pronounced as the automation and number of entities involved increases.

For example, what will be the respective legal postures when only two client agent AIs act in serial with a server as shown in FIG. 3?

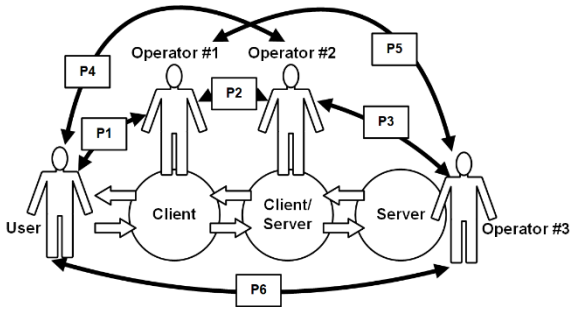

FIG. 3: Example Single Intermediary Topology

Here, **User** has made a request to **Client**, who in turn has consulted **Client/Server**, who in turn has consulted **Server**. **Client/Server** operates both as a server and as an AI agent. For example, **User** may have instructed **Client** (an event planning agent) to organize a child's birthday party (this example is adapted from Professor Ramesh Raskar's), **Client** consults **Client/Server** to commission preparation and delivery of an appropriate cake, and **Client/Server** then consults **Server** about acquiring materials and a delivery vehicle in a timely fashion for making and delivering the cake. Thus, e.g., **Client** may be an event planning smartphone application (e.g., managed by its developer **Operator #1**), **Client/Server** may be a server system operated by a party vendor **Operator**

#2, and **Server** may be a server system operated by a bakery vendor **Operator #3**. **P1-6** indicate the respective explicit (e.g., mutually agreed contractual terms) or implicit (e.g., implied contract, liability in tort, etc.) legal relationships between the entities following the respective transactions. As a fully connected network between *N* entities, the number of potential legal relationships will always be at least the binomial coefficient $\binom{N}{2}$ (naturally more if groups of entities are collectively recognized as having independent legal status).

Even casual inspection of this topology within the birthday hypothetical reveals a variety of nontrivial potential legal issues. To be clear: this note's purpose is not to suggest the law isn't able to address these issues – it generally is – rather, the point is that protocol-assisted legal posture formation would go far to mitigate or to moot many of these issues' resolution.

While it is impossible to comprehensively and dispositively address all these potential pitfalls in the space of this short note, the following provides an example listing of at least some of the potential issues:

1) Operator Tort Liability for an Agent AI's Actions
   a) Consider, e.g., if the child becomes sick eating the cake due to improper baking. Is tort liability for an Agent AI's actions imputed to the Agent's logic software designer, to the LLM creator, or to the operator entity hosting the agent? If to all of them, then how is the liability to be apportioned? If to only one of them, what foreseeability standard applies?
2) Operator Foreseeability
   a) What is the standard for foreseeability of harm by each operator on behalf of its respective client or server? Does maintaining a log of past interactions connote knowledge for purposes of future foreseeability? If so, what incentives / disincentives does this create?
   b) What server resources invite findings of operator negligence? Operator recklessness? Criminal negligence, recklessness, etc.?
   c) To what extent is an operator's foreseeability informed by what was presented to the agent during a specific request? Are operators incentivized for their agents to provide maximum context with every request to minimize their liability? Conversely, are operators incentivized for their servers to ignore as much context as possible?
3) Agency and Contractual Privity
   a) If **Server** fails to perform, is it liable to **Operator #2**, to **Operator #1**, or to the **User**? To all of them? To none of them? Why? Are there any implied contracts? Any recognition of **User**'s status as a third party beneficiary?
   b) Can any agent lower in the instruction chain be construed as acting on behalf of an upstream operator? Is this construed as "equitable control" by the upstream operator? N.b., since an AI agent is not a legal person it cannot be a *legal* agent, binding its principal, which much also be a legal person.
   c) Similarly, does contractual privity between the operators of any client and server connote agency of any intermediary? For example, if **Server**'s **Operator #3** is liable to **Client**'s **Operator #1** for its performance, does this imply that **Client/Server B**'s **Operator #2** was **Client**'s **Operator #1**'s legal agent? What representations would suffice to make them so? If so, consider the consequences for damages, e.g., is **Server**'s **Operator #3**'s liability to **Client's Operator #1** or **User** limited by **Client/Server**'s own instructions or limited by **Operator #2**'s own negligence?
   d) Is there privity between **User** and **Server**'s **Operator #3** via **P6** if **User** only issued instructions to **Client** and was unaware of whatever terms were presented to **Server** by **Client/Server** at **P3**?
4) Governing Terms
   a) If **Server** handles fulfillment of goods, which of **P1-P6** are governed by the Uniform Commercial Code (UCC)? Do warrants of merchantability and fitness for any particular purpose apply? To whom?
   b) If no jurisdiction is specified to govern **P1-P6**, and the **User** and **Operators** are each in different jurisdictions, then which laws apply for which interactions? For which harms?
   c) Since the protocol does not specify a manner of contract formation, what manner of terms presentation suffices to bind respective

operators? Note that this would appear to vary with each jurisdiction.

5) Infringement
   a) If an AI agent induces copyright, patent, or other infringement of another entity is its operator liable for the infringement? Are there any implicit representations regarding noninfringement on either side?
6) Incompatible Contract Overlap
   a) If **Server** imposes restrictions on the use of its tools/resources, must **Client/Server** affirm compliance for the upstream clients (consider, e.g., the CC-By-NC and if **User** is a commercial entity, whereas **Server** says it can only be utilized for non-commercial uses? How are these, and other restrictions, to be communicated between the agents?)?
   b) Many open source (and other) licenses are incompatible with one another (i.e., fulfilling one contract's terms obligates you to not fulfill another's). If this were a request to prepare a software program rather than plan a birthday party, then whose responsibility is it to ensure collective license compliance? Is there a separate duty by upstream parties to verify that determination?
7) AI Agent Status as Property
   a) If a request causes an agent to go beyond its offered resources under the ACP, is that considered a conversion of its system? Before blithely answering in the affirmative, consider that conversion is of property – is the agent the "property" of its operator? What if the agent is distributed across the cloud (e.g., on a blockchain) and immutable such that it loses legal status as property (see below discussion of the *Van Loon* case)? If the status is property is lost, then what foreseeability standard continues to apply for resultant torts (e.g., what are the salient temporal points to consider)?

Existing doctrines of privity, contract formation, agency, tort, etc. can answer many of the above questions, but because the ACP has not facilitated clarity in the legal postures, those answers will depend heavily upon the factual context of the transactions. *That* the answers will depend heavily upon the factual context of the transaction is thus evidence of incomplete structuring of those transactions and the appropriateness of their inclusion in the protocol rather than by ad hoc delegation. Again, even something as simple as a "terms of use" field in the protocol that must be accepted by a human operator before agents begin an exchange would do much to clarify the legal posture of the respective operators.

IV. Incidental Related Precedent

Despite attendees at the April/May events alleging that the above interactions are unprecedented, existing and growing caselaw already addresses many of the issues identified above. Indeed, improper contract formation, foreseeable harm resulting from improperly managed property, standards for establishing implied licenses and conversion, etc., are all long-established legal issues with frameworks for their resolution. Below, rather than attempt to comprehensively outline these doctrines, a few salient examples from recent ecommerce, smart contract, and related caselaw are alluded to merely for context:

IV.A Automated Contract Formation

Automated systems generally, and more recently, smart contracts specifically, have already provided fertile ground for analyzing offer, acceptance, consideration, delegation, etc. in the context of automation-assisted contract formation and execution. For example, in *Parma Tile Mosaic & Marble Co., Inc. v. Estate of Short*, 87 N.Y.2d 524 (N.Y. 1996), a fax machine's automated, non-discriminating signature attachments were found to be inadequate to serve as a signature satisfying the statute of frauds. For at least this reason, human-in-the-loop participation in agent operation (e.g., analogous to a smart contract's reference to a human oracle[8] during execution) is likely prudent for most ACPs, at least for the initial establishment of respective legal postures.

Conversely, affirmatively clicking on an "I agree" button sufficed to bind the user to a website's terms of use and consequent property transfer in *Metropolitan Regional Information Systems, Inc. v. American Home Realty Network, Inc.*, 722 F.3d 591 (4th Cir. 2013). Thus, ACP protocols incorporating human-in-the-loop functionality, where the operators are given an

opportunity to understand and affirm a given legal posture may be prudent. The Electronic Signatures in Global and National Commerce Act (ESIGN), codified in 15 U.S.C. § 7001, provides general rules for electronic record and transaction signature validity in connection with interstate or foreign commerce (most states have corresponding legislation).

Recognizing such formalities in the ACP itself may likewise be prudent since most online agreements must restrict their pathways of acceptance to ensure enforcement in any event. For example, in *In Rensel v. Centra Tech, Inc.*, No. 17-24500-CIV-KING/SIMONTON, 2018 U.S. Dist. LEXIS 100720 (S.D. Fla. June 14, 2018) the user's ability to engage with a token service *without* being confronted by the terms of service presented in the "normal" engagement pathway enabled the user to avoid the arbitration agreement contained in the terms ("The Defendants neither deny that a purchaser buying via the Smart Contract **would not have had to agree to the terms of the Token Sale Agreement to complete the transaction** nor offer **direct evidence that the Plaintiff did agree to the terms of the Token Sale Agreement**. Rather, the Defendants offer vague assertions and circumstantial and inconsistent evidence, to show that the Plaintiff "must have" entered into the Token Sale Agreement . . . because "defendant [has] offered no competent evidence to demonstrate the existence of a genuine issue of material fact concerning the existence of an arbitration agreement, [the] **motion to compel arbitration must be denied** as a matter of law without the need for a trial."" emphasis added).

IV.B Tort Foreseeability

Generally speaking, the law is already replete with frameworks for addressing foreseeability of harm in the tort and criminal liability contexts (e.g., strict liability for dog bites, liability for attractive nuisances on one's property, etc.). Absent common conventions facilitated by the ACP protocol itself, each user and operator will be trying to unilaterally limit their own liability while allocating as much as possible to their counterparty (e.g., ignoring information provided outside the ACP and trying to include as much risk-reducing information within the ACP as possible).

While not always breaking radical new legal ground, a number of cases have recently affirmed expectations or clarified postures in the automated context. For example, *Risley v. Universal Navigation Inc.*, No. 23-1340-cv, 2025 U.S. App. LEXIS 4460 (2d Cir. Feb. 26, 2025) reiterates the basic principle that platform operators in an arms-length relation to their users will not usually be liable for their actions ("In sum, we agree with the district court that it "defies logic" that a drafter of a smart contract, a computer code, could be held liable under the Exchange Act for a third-party user's misuse of the platform.").

On the other hand, again in the smart contract context, Commissioner Quintenz of the Commodity Futures Trading Commission (CFTC) has acknowledged that liability can be found based upon the intention of the developer ("I think the appropriate question is whether these code developers could reasonably foresee, **at the time they created the code**, that it would likely be used by U.S. persons in a manner violative of CFTC regulations. In this particular hypothetical, the code was specifically designed to enable the precise type of activity regulated by the CFTC, and no effort was made to preclude its availability to U.S. persons. Under these facts**, I think a strong case could be made that the code developers aided and abetted violations of CFTC regulations**. As such, the CFTC could prosecute those individuals for wrongdoing.", emphasis added).[9]

IV.C Immutable Automation / Property

Smart contracts have also provided fertile ground for confirming expectations surrounding control and exclusion sufficient for establishing property ownership.

For example, in *Van Loon v. United States Dep't of the Treasury, 122 F.4th 549*, the Fifth circuit found that the International Emergency Economic Powers Act, which was limited to "property", did not apply to a smart contract embodying an "open-source, cryptotransaction software protocol that facilitates anonymous transactions by obfuscating the origins and destinations" because the deployed smart contract was "immutable", i.e., beyond the control of its operators and creators, and therefore not "property" as the term was used in the Act (since no one has

exclusionary control over the smart contract itself once deployed; "The immutable smart contracts at issue in this appeal are **not property** because they are **not capable of being owned**." emphasis added).

Many AI agents will not meet this immutability standard since the operator will retain exclusionary control over the Agent's operation post-deployment. However, the case is mentioned here because the narrowness of this analysis is likely to inform operator liability for many AI agent actions, as well as the analysis for conversion of servers and agents by abusive clients.

V. CONCLUSION

This note simply encourages those interested in drafting ACP protocols to consider (as GitHub™ and other package repositories have already done) incorporating formal mechanisms for manifesting legal postures between the various operators so as to avoid reliance upon the (likely time-consuming and less efficient) default application of the caselaw. Providing mechanisms to mitigate or allocate liabilities *ex ante* not only reduces such inefficiencies, but is also likely to encourage adoption of the respective protocols by making more transparent the relative relations of the participating operators.

---

[1] For example, I posed the following to an LLM: Q1: "Which came first, the chicken or the egg?" This phrase was likely encountered during training and so, as expected, produced a coherent response acknowledging that the phrase was a form of causal riddle. However, simply posing Q2: "Why do eggs cause chickens?" sufficed to produce a generally incorrect response (e.g., the response asserting: "in the classical sense, eggs do not directly cause chickens" and that their doing so was "counterfactual."). Note that training on additional queries similar to Q2 may serve to "remediate" this discrepancy by forcing the system to respond as desired, i.e., to ensure that its textual correlations adhere to the physical correlations. However, such "whack-a-mole" retraining isn't always feasible. Consider a less playful example of erroneous hallucination here https://www.digitalmusicnews.com/2025/05/16/anthropic-claude-hallucination-apology/ (""Regarding citation hallucinations more generally – this is a known limitation of large language models like myself," Claude responded. "When asked to provide citations, if I don't have perfect recall of specific sources, I might generate what seem like plausible citations based on my training patterns rather than accurate bibliographic information. "For any situation requiring accurate citations, the best practice would be to use dedicated academic search tools and databases rather than relying on an AI system to recall specific publication details from memory," Claude continued.") acquired 05/16/2025.

[2] See, e.g. "AI Agents vs. Agentic AI: A Conceptual Taxonomy, Applications and Challenges" arXiv:2505.10468v1 (05/15/2025).

[3] See https://modelcontextprotocol.io/faqs ("MCP (Model Context Protocol) is a standard way for AI applications and agents to connect to and work with your data sources (e.g. local files, databases, or content repositories) and tools (e.g. GitHub, Google Maps, or Puppeteer).") acquired 05/18/2025.

[4] See https://nanda.media.mit.edu/, acquired 05/18/2025.

[5] See, e.g., https://modelcontextprotocol.io/quickstart/server#testing-your-server-with-claude-for-desktop ("You'll then add your servers in the mcpServers key.") acquired on 05/18/2025.

[6] See, e.g., https://docs.github.com/en/repositories/managing-your-repositorys-settings-and-features/customizing-your-repository/licensing-a-repository acquired 05/19/2025.

[7] https://modelcontextprotocol.io/specification/draft/index#security-and-trust-%26-safety acquired 05/19/2025.

[8] Consider, e.g., https://stellar.org/learn/smart-contract-basics-oracles acquired 05/19/2025.

[9] See https://www.cftc.gov/PressRoom/SpeechesTestimony/opaquintenz16 acquired 05/18/2025.