Applying the A2A Protocol to Unreal Strands Agents

The Agent-to-Agent (A2A) protocol is an open standard designed to allow independent agents to communicate and collaborate without needing to know each other's internal workings. In the context of Unreal Engine and Strands agents, A2A provides a structured way for NPCs and AI-driven characters to exchange information, coordinate tasks, and negotiate interactions in a shared world.

At its core, A2A revolves around a few key concepts. The Agent Card is the foundation of discovery. It is a self-describing JSON document that tells other agents who this agent is, what it can do, and how to talk to it. In Unreal, each Strands agent could expose an Agent Card that describes its role in the world—whether it is a trader, a guard, or a guide—along with the skills it supports, such as bartering, patrolling, or giving directions. This allows agents to dynamically discover each other's capabilities and decide how to interact.

Communication between agents happens through Messages and Tasks. A message is a single turn in a conversation, containing one or more parts such as text, structured data, or files. A task is a stateful unit of work that persists over time, such as navigating to a location, crafting an item, or engaging in dialogue. Tasks move through a lifecycle, from being submitted, to working, to either completion or failure. This lifecycle maps naturally to Unreal's AI state machines, where agents can be idle, working, waiting for input, or finished.

The protocol supports different transports for communication, including JSON-RPC, gRPC, and REST. All transports are functionally equivalent, but gRPC is often best suited for low-latency, real-time interactions between agents inside Unreal, while JSON-RPC or REST may be more appropriate for connecting to external services such as cloud-based AI systems. A2A also supports streaming through Server-Sent Events or gRPC streams, which is particularly useful for long-running or continuous interactions. For example, an NPC could stream updates as it narrates its patrol route or as it generates dialogue in real time.

Security is built into the protocol through standard web practices. Agents declare their authentication requirements in their Agent Cards, and clients must provide the appropriate credentials, such as API keys or OAuth tokens. While local Unreal agents may not need heavy authentication, this becomes important when connecting to external services or multiplayer environments where trust boundaries matter.

A2A also defines how agents exchange richer content. Messages can contain Parts, which may be plain text, structured JSON data, or files such as images or audio. The results of tasks are returned as Artifacts, which bundle together the outputs of an agent's work. In Unreal, this could mean an NPC returning a JSON object with navigation coordinates, a text artifact containing dialogue, or even a file artifact representing a generated map or crafted item.

The protocol provides a set of standard methods for interaction. The most im-

portant are message/send, which starts or continues a task, and tasks/get, which retrieves the current state of a task. Optional methods such as message/stream allow for real-time updates, while push notification methods let agents notify each other when long-running tasks are complete. These methods form the building blocks of agent-to-agent communication, ensuring that all agents can at least send messages, track progress, and cancel tasks when needed.

Error handling is standardized as well. If an agent requests something unsupported, the protocol defines clear error codes, such as "Task Not Found" or "Unsupported Operation." This ensures that agents can fail gracefully and adapt their behavior when interacting with others.

In practice, applying A2A to Unreal Strands agents means treating each NPC as a discoverable, self-describing service. Their Agent Cards define their roles and skills, their messages and tasks represent their interactions, and their artifacts capture the results of their actions. By adopting A2A, Unreal worlds can move beyond hardcoded interactions and toward a dynamic ecosystem where agents negotiate, collaborate, and adapt in real time. This creates richer, more emergent gameplay, where communication between agents is not just scripted but standardized and interoperable.