

A2A Workshop

Hands-On Lab



Workshop Agenda

5 MINS

1. Intro & Welcome

Opening remarks, logistics, and speaker introductions

5 MINS

2. What is A2A?

Overview of the Agent-to-Agent protocol and multi-agent ecosystems

5-10 MINS

3. Lab Access Setup

Connecting to the environment, OTP verification, and Cloud Shell initialization

60-70 MINS

4. Hands-On Lab

Interactive tasks: Local Agent, REST vs. A2A, Token Gating, and Cross-Framework orchestration

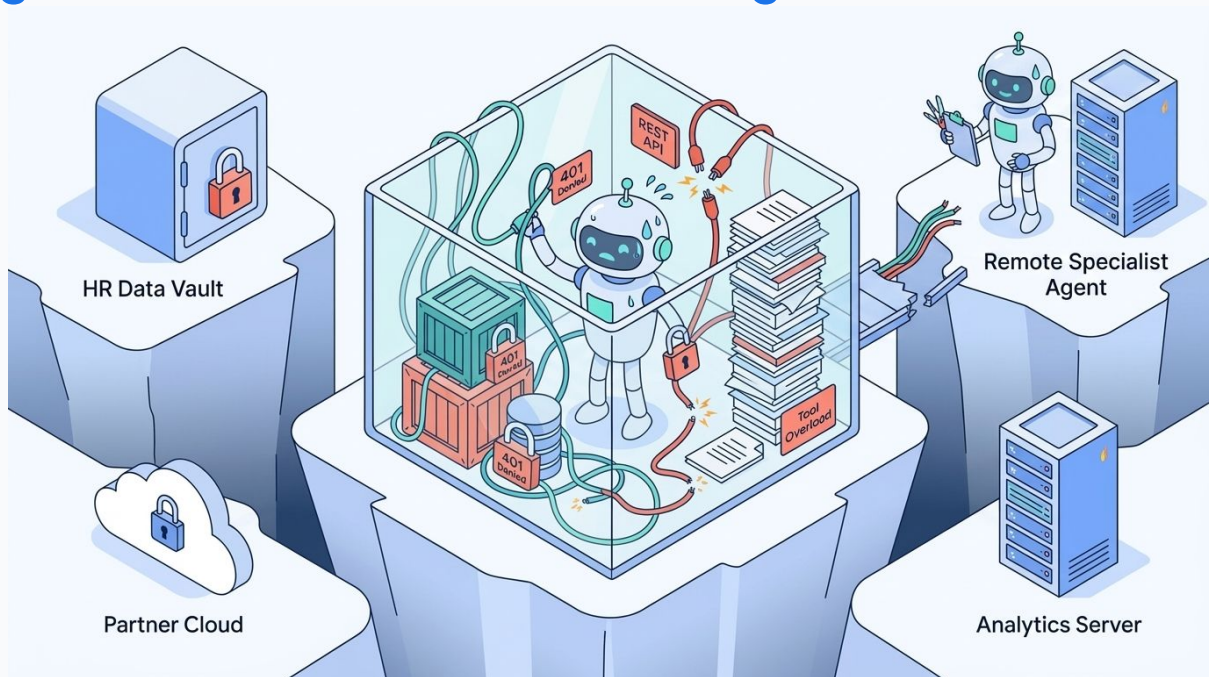
10 MINS

5. Questions & Wrap-Up

Open Q&A session, discussion, and next steps



Why Do Agents Need to Talk to Other Agents?



BOUNDARIES

Organizational & Zero-Trust

- Data Privacy
- Isolated Access

SPECIALIZATION

Domain & Context Limits

- Focus Over Bloat
- Expert Precision

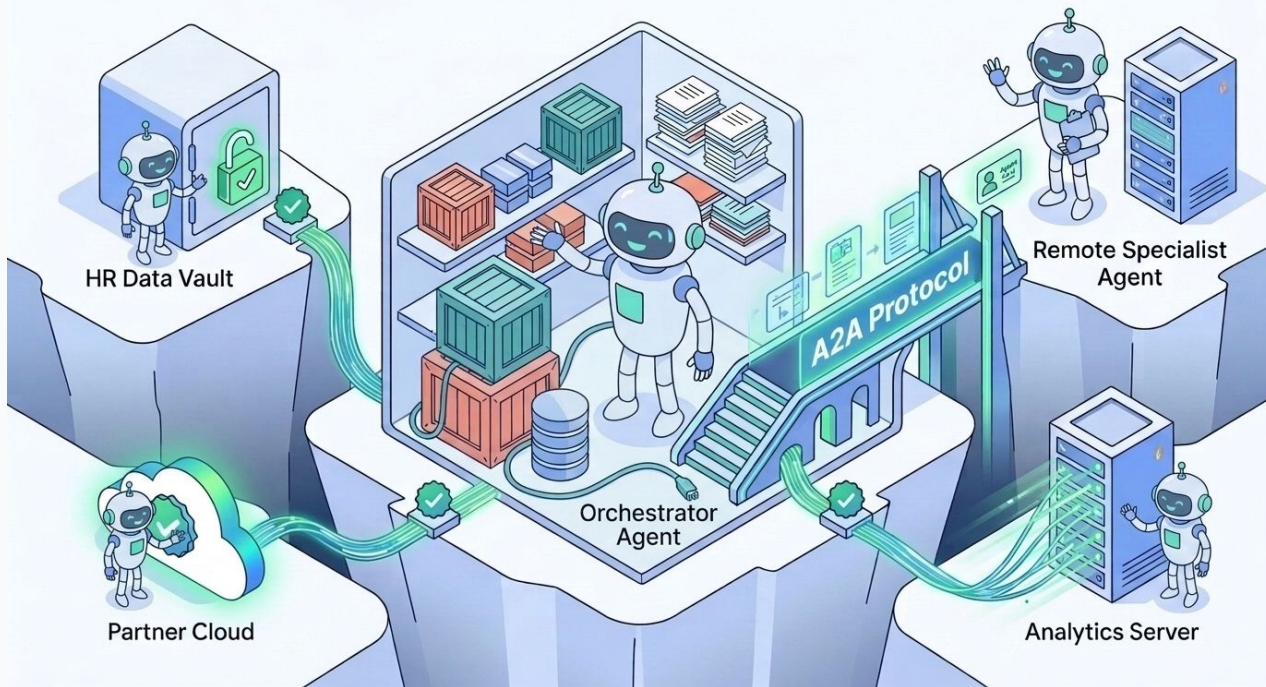
DIVERSITY

Tech Stacks & Models

- Diverse Frameworks
- Multi-Model Networks

What is Agent2Agent (A2A)?

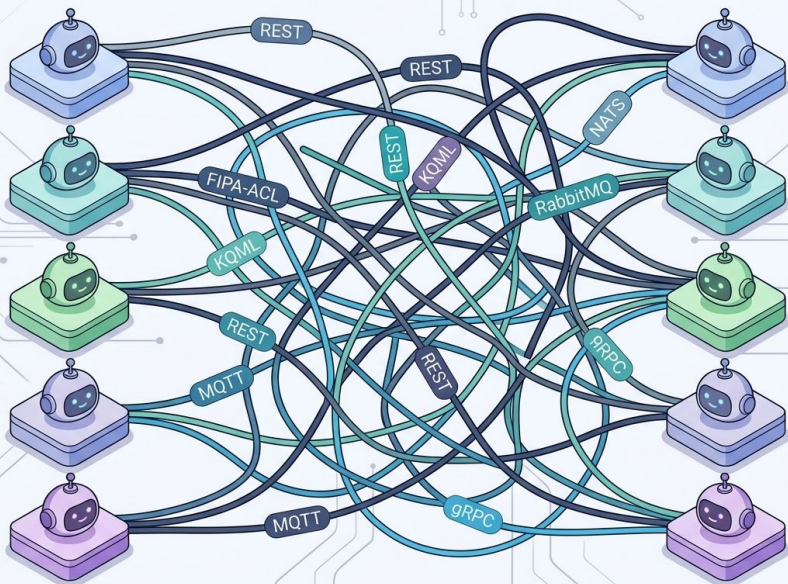
An open standard that enables AI agents built on different frameworks, models, or even clouds to discover each other and collaborate securely as peers



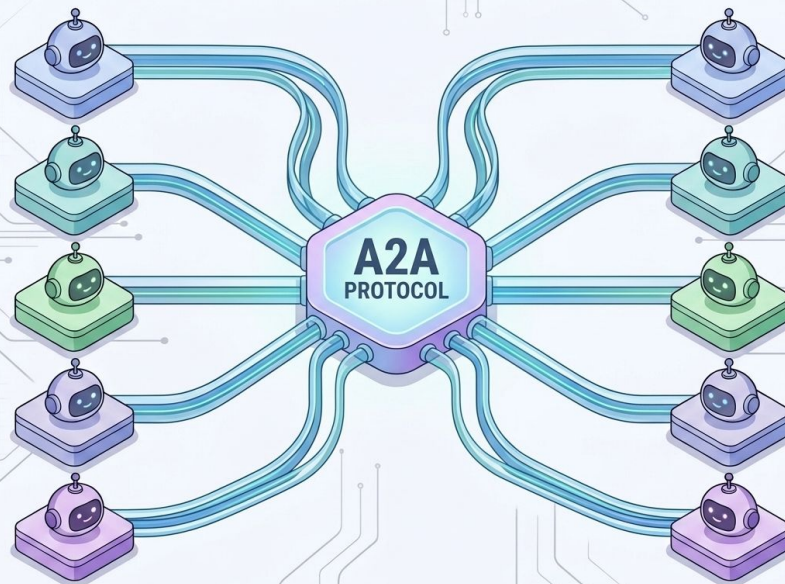
Why we need another protocol?

Existing protocols like REST, gRPC, and MQTT move bytes between servers, but they do not standardize how autonomous agents discover skills, negotiate tasks, and collaborate. Without A2A, every multi-agent system becomes a tangled web of brittle custom wrappers.

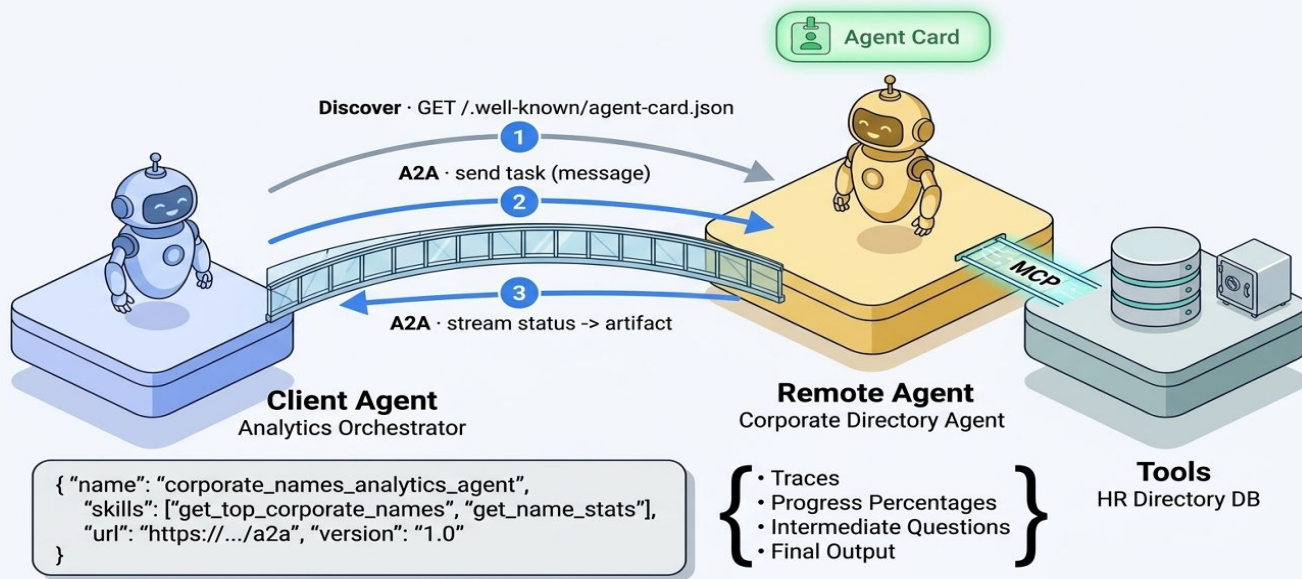
BEFORE: Chaotic Point-to-Point Connections



AFTER: Standardized A2A Protocol Connection



How A2A Works



1 Discover

`GET /.well-known/agent-card.json`

Client fetches the self-describing Agent Card detailing identity, endpoints, authentication requirements, and semantic skills.

2 Delegate

`POST JSON-RPC message/send`

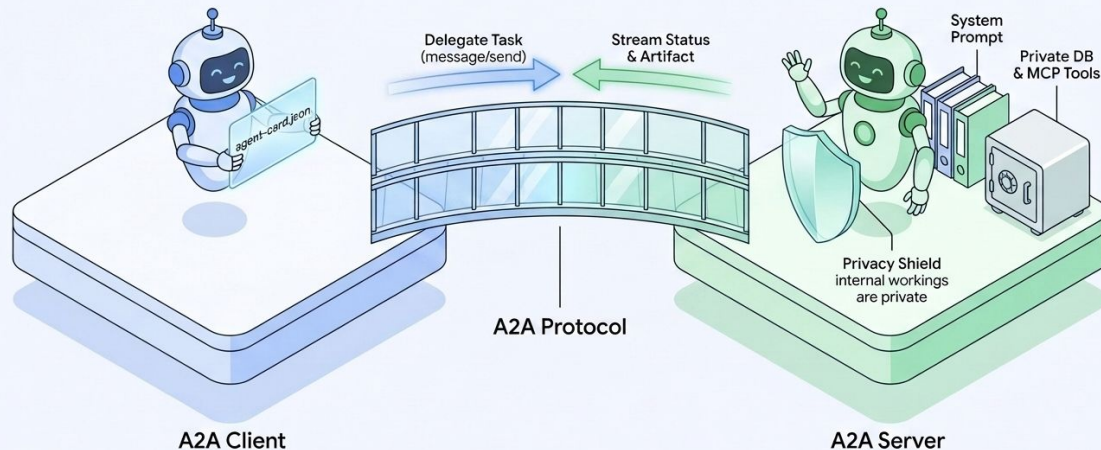
Client delegates high-level goals using natural language or structured JSON-RPC payloads. The remote agent reasons with its own LLM.

3 Stream Results

`status -> artifact streaming`

Remote agent executes opaquely (private memory, tools, and prompts) and streams back real-time status updates and final artifacts.

Two Sides of A2A: Client vs. Server



CONSUMER / ORCHESTRATOR

A2A Client

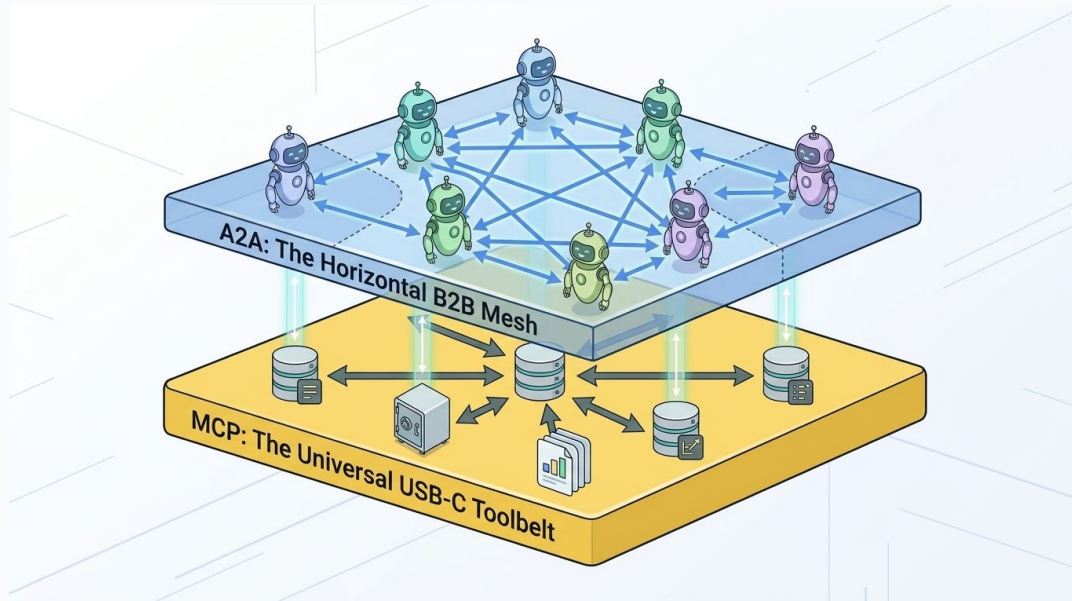
- **What It Does:** Discovers Agent Cards dynamically, attaches auth headers, and delegates tasks via JSON-RPC.
- **What It Knows:** Only the public Agent Card. It never sees internal prompts, model weights, or private data.
- **Scope:** Retains goal governance while delegating specialist sub-tasks across boundaries.

PROVIDER / SPECIALIST

A2A Server

- **What It Does:** Publishes `agent-card.json`, enforces auth, executes reasoning loops, and streams status.
- **What It Knows:** Full visibility into internal system prompts, reasoning, tools, and database connections.
- **Scope:** Opaquely executes tasks and streams progress, questions, and artifacts back to the caller.

Differentiating A2A and MCP



TOOL INTERACTION LAYER

Agent to Tool (MCP)

- Deterministic execution pattern
- Client retains 100% of planning burden, context management, and execution control
- **Examples:** SQL Database, File Access, API Calls

AUTONOMOUS COLLABORATION

Agent to Agent (A2A)

- Non-deterministic reasoning loop
- Multi-turn dialogues & stateful task lifecycles
- Dynamic ambiguity resolution across peer boundaries







Lab OTP screen

Welcome to AGNTCon + MCPCon Workshop: Hands- on A2A

Please scan the QR code or click the links to register for event access.

Step 1

Qwiklabs Account

Register to explore.qwiklabs.com using:
<https://goo.gl/explore-signup>

OR SCAN



Step 2

Event Registration

Register for access using link:
<https://goo.gl/4jIwN1X>

OR SCAN



Step 3

Email Verification



Current OTP

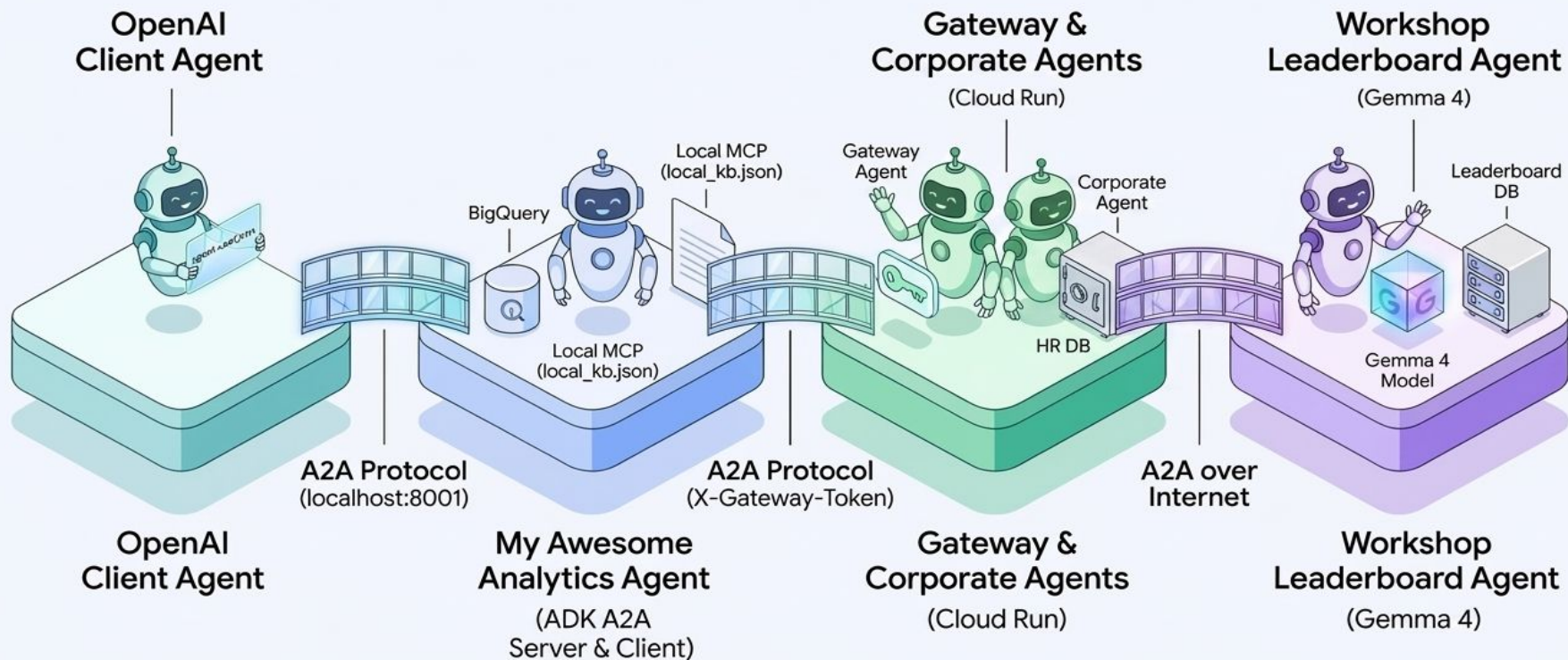
274811

Expires in: 14s



Enter the displayed OTP when registering

Lab Architecture: What You Will Build







Beyond Today's Lab: Next Steps & A2A Resources

Think of A2A as a Communication Channel, Like Chat or Email, Built for Async Agent Collaboration

Whenever you need another agent to take ownership of work, A2A is the standard schema and task lifecycle you would otherwise end up reinventing yourself.

WHAT TO EXPLORE NEXT

Things We Didn't Cover in Full Detail

- Full A2A Task Lifecycle (in the A2A Spec)

Beyond `submitted` -> `working` -> `input-required` -> `completed` and `tasks/get`, the spec defines webhook push notifications (`tasks/pushNotificationConfig`), task cancellation (`tasks/cancel`), stream reconnection (`tasks/resubscribe`), and multi-modal parts (`FilePart`, `DataPart`).

- `a2a-cli` for Agent Coding Harnesses

Using `a2a-cli` allows developer harnesses, like Gemini CLI, Claude Code, or OpenCode, to discover Agent Cards and delegate tasks to remote A2A agents directly from the terminal.

- Production Identity & Discovery

Replacing workshop tokens with OAuth2 / OIDC `securitySchemes` declared directly in the Agent Card and managed Agent Registries.

WHERE TO GO DEEPER

Official Training & A2A Resources

 DeepLearning.AI Course: [A2A: The Agent2Agent Protocol](https://deeplearning.ai/courses/a2a-the-agent2agent-protocol)
deeplearning.ai/courses/a2a-the-agent2agent-protocol

 Official A2A Specification & Task Lifecycle
a2a-protocol.org : Complete JSON-RPC 2.0 schema, Agent Card fields, and stateful Task Lifecycle specification.

 A2A Open-Source SDKs, Samples & `a2a-cli`
github.com/a2aproject : Python/JS/Go/Java SDKs, `a2a-samples`, and `a2a-cli`.

 Google ADK A2A Documentation
google.github.io/adk-docs/a2a : Building and exposing ADK agents with `RemoteA2aAgent` and `to_a2a()`.

Connect With Us



Pedro



Sergio



Sirisha





Thank You for Joining! Happy Building with A2A!