COMP 019 - Applications Development and Emerging Technologies

Emerging Technologies: MCP, A2A, Pgvector

SESSION 10: A2A PROTOCOL (EMERGING TECHNOLOGY)

ACTIVITY 10.1: INTRODUCTION TO AGENT-TO-AGENT PROTOCOL

Topic: Understanding Google's A2A protocol

Description: Learn how agents communicate with each other using A2A

INSTRUCTIONS:

Research and document A2A:
- What is A2A? (Agent-to-Agent Protocol)
- Created by Google, now Linux Foundation standard
- Purpose: Enable agent interoperability
- 150+ supporting organizations

Understand A2A vs MCP:
- MCP: Agent-to-Tool communication
- A2A: Agent-to-Agent communication
- They complement each other

A2A key concepts:
- Agent Card (/.well-known/agent.json)
- A2A Client (task requester)
- A2A Server (task executor)
- Tasks, Messages, Artifacts
- JSON-RPC over HTTP/SSE

Document A2A communication flow:
- Discovery → Capability Check → Task → Result

Deliverables: Submit A2A research document with flow diagrams	**25 Points**

ACTIVITY 10.2: BUILDING AN A2A SERVER

Topic: Creating an A2A-compliant agent server

Description: Build an agent that can receive tasks from other agents

INSTRUCTIONS:

Set up A2A development:
- pip install a2a-sdk
- Understand A2A server structure

Create Agent Card (agent.json):
- Agent name and description
- Supported skills/capabilities
- Endpoint URL

• Input/output formats

Build 'Data Analysis Agent' A2A server:

• Skill: Analyze student performance data

• Skill: Generate statistical reports

• Skill: Find trends in grades

• Connect to your PostgreSQL database

Implement A2A endpoints:

• Handle incoming task requests

• Process tasks asynchronously

• Return results in A2A format

• Support streaming for long tasks

Test with A2A client tools

Deliverables: Submit A2A server code, agent.json, and test results	**40 Points**

ACTIVITY 10.3: BUILDING AN A2A CLIENT

Topic: Creating an agent that consumes A2A services

Description: Build an agent that discovers and uses other agents' capabilities

INSTRUCTIONS:

Create A2A client agent:

• Discover agents via Agent Cards

• Parse agent capabilities

• Send task requests

• Handle responses and streaming

Build 'Personal Assistant' A2A client:

• Discovers available helper agents

• Routes tasks to appropriate agents

• Aggregates results for user

Implement orchestration:

• User asks: 'Analyze IT department performance'

• Assistant finds Data Analysis agent

• Sends task to analyze IT students

• Receives and presents results

Handle edge cases:

• Agent not available

• Task timeout

• Error responses

Create simple CLI interface for testing

Deliverables: Submit A2A client code and orchestration demo	**35 Points**

TOTAL POINTS: 100