A2A (Agent to Agent) 프로토콜: 포괄적 분석 보고서

작성자: Manus AI

작성일: 2025년 6월 26일

버전: 1.0

목차

개요

인공지능 에이전트 기술이 급속도로 발전하면서, 서로 다른 플랫폼과 벤더에서 개발된 AI 에이전트들 간의 상호운용성이 중요한 과제로 대두되고 있다. 2025년 4월, Google이 발표한 Agent2Agent(A2A) 프로토콜은 이러한 문제를 해결하기 위한 혁신적인 개방형 표준으로 주목받고 있다[1].

A2A 프로토콜은 AI 에이전트들이 기반 프레임워크나 벤더에 관계없이 서로 통신하고, 정보를 안전하게 교환하며, 작업을 조율할 수 있도록 하는 표준화된 방법을 제공한다. 이는 단순히 기술적 혁신을 넘어서, 엔터프라이즈 환경에서 AI 에이전트들이 협업하여 복잡한 워크플로를 자동화하고 전례 없는 수준의 효율성과 혁신을 추진할 수 있는 미래를 제시한다[2].

본 보고서는 A2A 프로토콜의 개념과 정의부터 시작하여, 프로토콜의 구조와 기술적 세부사항, 실제 활용 사례, 그리고 최신 트렌드와 미래 전망까지 포괄적으로 분석한다. 특히 Google과 Microsoft를 비롯한 주요 기술 기업들의 채택 현황과 50여 개 기술 파트너들의 참여 상황을 통해 A2A 프로토콜이 AI 에이전트 생태계에 미치는 영향을 심층적으로 살펴본다[3].

A2A 프로토콜 개념 및 정의

A2A 프로토콜의 기본 개념

Agent2Agent(A2A) 프로토콜은 독립적이고 잠재적으로 불투명한 AI 에이전트 시스템 간의 통신과 상호운용성을 촉진하기 위해 설계된 개방형 표준이다[4]. 에이전트들이 서로 다른 프레임워크, 언어, 또는 다른 벤더에 의해 구축될 수 있는 생태계에서, A2A는 공통 언어와 상호작용 모델을 제공한다.

A2A 프로토콜의 핵심은 에이전트들이 서로의 내부 상태, 메모리, 또는 도구에 접근할 필요 없이 사용자 목표를 달성하기 위해 안전하게 정보를 교환할 수 있도록 하는 것이다[5]. 이는 기존의 도구 중심 접근법과는 근본적으로 다른 패러다임으로, 에이전트를 단순한 '도구'로 국한하지 않고 진정한 멀티 에이전트 시나리오를 구현한다.

MCP와의 차이점 및 보완 관계

A2A 프로토콜은 Anthropic의 Model Context Protocol(MCP)과 보완적인 관계에 있다[6]. MCP가 에이전트에게 유용한 도구와 컨텍스트를 제공하는 데 중점을 둔다면, A2A는 에이전트 간의 동적이고 다중 모달 통신을 촉진한다.

구체적으로 살펴보면, MCP는 AI가 기존 도구, 웹 API, 또는 데이터에 액세스할 수 있도록 하는 프로토콜로, 구조화된 입력/출력을 통해 에이전트가 자신의 능력에 접근하는 방식을 정의한다[7]. 반면 A2A는 서로 다른 에이전트들이 동료로서 동적으로 통신하고, 작업을 위임하며, 공유 작업을 관리하는 방법을 다룬다.

이러한 차이점은 실제 활용에서 더욱 명확해진다. 예를 들어, 대출 승인 시스템에서 LoanProcessor 에이전트는 MCP를 사용하여 신용 점수 API를 호출하고, 은행 거래 내역을 검색하며, OCR을 통해 업로드된 문서를 검증할 수 있다[8]. 동시에 A2A를 사용하여 위험 평가 에이전트, 규정 준수 에이전트, 승인 에이전트와 협업하여 최종 승인 결정을 내릴 수 있다.

A2A 프로토콜의 설계 원칙

A2A 프로토콜은 다음과 같은 다섯 가지 핵심 설계 원칙을 기반으로 개발되었다[9]:

에이전트의 능력 수용(Embrace Agentic Capabilities): A2A는 에이전트가 메모리, 도구, 컨텍스트를 공유하지 않더라도 비체계적인 자연스러운 방식으로 협업할 수 있도록 지원한다. 이는 에이전트를 단순한 '도구'로 국한하지 않고 진정한 멀티 에이전트 시나리오를 구현하는 데 중점을 둔다.

기존 표준 기반 개발(Built on Open Standards): 프로토콜은 HTTP, SSE(Server-Sent Events), JSON-RPC를 포함하여 널리 사용되는 기존 표준을 기반으로 개발되었다. 이를 통해 기업에서 이미 매일 사용 중인 기존 IT 스택과 더욱 쉽게 통합할 수 있다.

기본적으로 보안 보장(Secure by Default): A2A는 출시 시점에 OpenAPI의 인증 체계와 동등한 수준에서 엔터프라이즈급 인증 및 승인을 지원하도록 설계되었다. 이는 표준 웹 보안 관행에 의존하는 엔터프라이즈 환경에 적합한 보안 통신 패턴을 제공한다.

장기 실행 작업 지원(Support Long-Running Tasks): 빠르게 처리해야 하는 작업부터 사람이 수행하면 몇 시간이나 며칠이 걸릴 수 있는 심층 연구에 이르기까지, 모든 작업을 완료하는 데 탁월한 능력을 발휘하도록 A2A를 유연하게 설계했다. 이 과정 전반에서 A2A는 사용자에게 실시간 피드백, 알림 및 상태 업데이트를 제공할 수 있다.

모달리티와 무관(Modality Agnostic): 에이전트 응용 분야는 텍스트에만 국한되지 않으므로 오디오 및 비디오 스트리밍을 포함한 다양한 모달리티를 지원하도록 A2A를 설계했다. 이는 텍스트, 오디오/비디오, 구조화된 데이터/양식, 잠재적으로 임베디드 UI 구성 요소 등 다양한 콘텐츠 유형의 교환을 지원한다.

핵심 참여자 및 생태계

A2A 프로토콜의 성공은 광범위한 업계 참여에 기반한다. Google의 주도하에 Atlassian, Box, Cohere, Intuit, Langchain, MongoDB, PayPal, Salesforce, SAP, ServiceNow, UKG, Workday 등 50여 곳의 기술 파트너가 참여하고 있다[10]. 또한 Accenture, BCG, Capgemini, Cognizant, Deloitte, HCLTech, Infosys, KPMG, McKinsey, PwC, TCS, Wipro를 포함한 선도적인 서비스 제공 업체들도 지원과 기여를 통해 프로토콜 개발에 참여하고 있다.

특히 주목할 점은 Microsoft가 2025년 5월 A2A 프로토콜 지원을 발표하며 Azure AI Foundry와 Copilot Studio에 통합을 예고한 것이다[11]. 이는 A2A 프로토콜이 단순히 Google의 독자적인 표준이 아닌, 업계 전반의 협력을 통한 진정한 개방형 표준으로 발전하고 있음을 보여준다.

프로토콜 구조 및 기술적 세부사항

핵심 아키텍처 구성 요소

A2A 프로토콜의 아키텍처는 세 가지 핵심 참여자를 중심으로 구성된다[12]. 첫째, 사용자(User)는 에이전트 지원이 필요한 요청이나 목표를 시작하는 최종 사용자(인간 또는 자동화된 서비스)이다. 둘째, A2A 클라이언트(Client Agent)는 사용자를 대신하여 원격 에이전트에게 작업이나 정보를 요청하는 애플리케이션, 서비스, 또는 다른 AI 에이전트로, A2A 프로토콜을 사용하여 통신을 시작한다. 셋째, A2A 서버(Remote Agent)는 A2A 프로토콜을 구현하는 HTTP 엔드포인트를 노출하는 AI 에이전트 또는 에이전트 시스템으로, 클라이언트로부터 요청을 받아 작업을 처리하고 결과나 상태 업데이트를 반환한다.

이러한 구조에서 원격 에이전트는 클라이언트의 관점에서 "불투명한" 시스템으로 작동한다. 즉, 클라이언트는 원격 에이전트의 내부 작동 방식, 메모리, 또는 도구를 알 필요가 없다[13]. 이는 에이전트 간의 느슨한 결합을 가능하게 하며, 각 에이전트가 독립적으로 발전하고 업데이트될 수 있도록 한다.

전송 및 데이터 형식

A2A 통신은 반드시 HTTP(S)를 통해 이루어져야 하며, A2A 서버는 AgentCard에 정의된 URL에서 서비스를 노출한다[14]. 프로토콜은 모든 요청과 응답에 대해 JSON-RPC 2.0을 페이로드 형식으로 사용한다. 클라이언트 요청과 서버 응답은 JSON-RPC 2.0 사양을 준수해야 하며, JSON-RPC 페이로드를 포함하는 HTTP 요청과 응답의 Content-Type 헤더는 반드시 `application/json`이어야 한다.

스트리밍이 사용되는 경우, 특히 `message/stream`이나 `tasks/resubscribe`와 같은 메소드에서는 서버가 HTTP `200 OK` 상태와 `text/event-stream`의 Content-Type 헤더로 응답한다[15]. 이 HTTP 응답의 본문에는 W3C에서 정의한 Server-Sent Events(SSE) 스트림이 포함되며, 각 SSE `data` 필드에는 완전한 JSON-RPC 2.0 응답 객체가 포함된다.

에이전트 발견 메커니즘

A2A 프로토콜에서 에이전트 발견은 에이전트 카드(Agent Card)를 통해 이루어진다[16]. 에이전트 카드는 A2A 서버가 게시하는 JSON 메타데이터 문서로, 일반적으로 `/.well-known/agent.json`에서 발견할 수 있다. 이 카드에는 에이전트의 신원(이름, 설명), 서비스 엔드포인트 URL, 버전, 지원되는 A2A 기능(스트리밍이나 푸시 알림 등), 제공하는 특정 스킬, 기본 입력/출력 모달리티, 그리고 인증 요구사항이 포함된다.

클라이언트는 에이전트 카드를 사용하여 에이전트를 발견하고 안전하고 효과적으로 상호작용하는 방법을 이해한다. 발견은 DNS, 레지스트리, 마켓플레이스, 또는 프라이빗 카탈로그를 통해 처리될 수 있다[17]. 에이전트 카드의 구조는 표준화되어 있어 다양한 플랫폼과 벤더 간의 일관성을 보장한다.

작업 관리 및 수명 주기

A2A 통신의 핵심은 '작업(Task)'이라는 구조화된 객체이다[18]. 작업은 에이전트가 수행하는 원자적 작업 단위를 나타내며, submitted, working, input-required, completed와 같은 수명 주기 상태를 가진다. 각 작업에는 고유한 ID가 있으며, 메시지(클라이언트와 원격 에이전트 간의 대화형 교환에 사용), 아티팩트(원격 에이전트가 생성한 불변 결과), 그리고 파트(메시지나 아티팩트 내의 자체 포함된 데이터 블록)를 포함한다.

작업의 수명 주기는 다음과 같이 진행된다. 먼저 사용자가 클라이언트 에이전트에게 요청을 보내면, 클라이언트 에이전트는 에이전트 카드를 사용하여 다른 유능한 에이전트를 찾는다[19]. 그 다음 클라이언트 에이전트는 선택된 원격 에이전트에게 JSON-RPC 2.0을 통해 HTTP로 `task/send` 요청을 보낸다. 페이로드에는 작업 ID, 세션 ID(선택사항), 그리고 하나 이상의 파트를 포함하는 메시지가 들어 있다.

원격 에이전트는 작업을 처리하고 즉시 결과(아티팩트를 통해), 중간 메시지, 또는 더 많은 정보 요청(상태를 input-required로 설정)으로 응답한다[20]. 작업이 장기 실행되는 경우, 원격 에이전트는 Server-Sent Events(SSE)를 사용하여 부분 결과를 스트리밍할 수 있다. 클라이언트가 연결을 끊으면, 클라이언트가 제공한 보안 웹훅으로 푸시 알림을 보낼 수 있다.

메시지 및 파트 구조

메시지는 클라이언트와 원격 에이전트 간의 단일 통신 턴을 나타낸다[21]. 메시지는 `role`("user"는 클라이언트가 보낸 메시지, "agent"는 서버가 보낸 메시지)을 가지며 하나 이상의 `Part` 객체를 포함하여 실제 콘텐츠를 전달한다. 메시지 객체의 `messageId` 부분은 메시지 발신자가 설정하는 각 메시지의 고유 식별자이다.

파트는 메시지나 아티팩트 내의 기본 콘텐츠 단위이다[22]. 각 파트는 특정 `type`을 가지며 다양한 종류의 데이터를 전달할 수 있다. `TextPart`는 일반 텍스트 콘텐츠를 포함하고, `FilePart`는 인라인 base64 인코딩된 바이트나 URI를 통해 참조되는 파일을 나타내며 파일명과 미디어 타입 같은 메타데이터를 포함한다. `DataPart`는 구조화된 JSON 데이터를 전달하며, 양식, 매개변수, 또는 기계가 읽을 수 있는 정보에 유용하다.

인증 및 보안

A2A는 에이전트를 표준 엔터프라이즈 애플리케이션으로 취급하며, 확립된 웹 보안 관행에 의존한다[23]. 신원 정보는 A2A JSON-RPC 페이로드 내에서 전송되지 않고 HTTP 전송 계층에서 처리된다. 프로덕션 배포에서는 반드시 HTTPS를 사용해야 하며, 강력한 암호화 스위트를 갖춘 현대적인 TLS 구성(TLS 1.2+ 권장)을 사용해야 한다.

클라이언트/사용자 신원 및 인증 프로세스는 세 단계로 이루어진다[24]. 첫째, 요구사항 발견 단계에서 클라이언트는 AgentCard의 `authentication` 필드를 통해 서버의 필수 인증 체계를 발견한다. 체계 이름은 종종 OpenAPI 인증 방법과 일치한다(예: OAuth 2.0 토큰의 경우 "Bearer", Basic Auth의 경우 "Basic", API 키의 경우 "ApiKey"). 둘째, 자격 증명 획득 단계에서 클라이언트는 필요한 인증 체계와 신원 제공자에 특정한 대역 외 프로세스를 통해 필요한 자격 증명을 획득한다. 이 프로세스는 A2A 프로토콜 자체의 범위를 벗어난다. 셋째, 자격 증명 전송 단계에서 클라이언트는 서버에 보내는 모든 A2A 요청의 적절한 HTTP 헤더에 이러한 자격 증명을 포함한다.

스트리밍 및 비동기 작업

A2A 프로토콜은 세 가지 상호작용 메커니즘을 지원한다[25]. 첫째, 요청/응답(폴링) 방식에서 클라이언트는 요청을 보내고 서버로부터 응답을 받는다. 상호작용이 상태를 가진 장기 실행 작업을 필요로 하는 경우, 서버는 처음에 `working` 상태로 응답할 수 있다. 그러면 클라이언트는 작업이 터미널 상태에 도달할 때까지 주기적으로 `tasks/get`을 호출하여 업데이트를 폴링한다.

둘째, 스트리밍(Server-Sent Events - SSE) 방식은 점진적으로 결과를 생성하거나 실시간 진행 상황 업데이트를 제공하는 작업에 사용된다[26]. 클라이언트는 `message/stream`을 사용하여 서버와의 상호작용을 시작한다. 서버는 열린 상태로 유지되는 HTTP 연결로 응답하며, 이를 통해 Server-Sent Events(SSE) 스트림을 보낸다. 이러한 이벤트는 작업 상태 변경을 위한 `TaskStatusUpdateEvent`나 새로운 또는 업데이트된 아티팩트 청크를 위한 `TaskArtifactUpdateEvent`일 수 있다. 이를 위해서는 서버가 에이전트 카드에서 `streaming` 기능을 광고해야 한다.

셋째, 푸시 알림은 매우 장기 실행되는 작업이나 지속적인 연결(SSE 같은)을 유지하는 것이 비실용적인 시나리오에 사용된다[27]. 클라이언트는 작업을 시작할 때 웹훅 URL을 제공하거나 `tasks/pushNotificationConfig/set`을 호출하여 제공할 수 있다. 작업 상태가 크게 변경되면(예: 완료, 실패, 또는 입력 필요), 서버는 이 클라이언트 제공 웹훅으로 비동기 알림(HTTP POST 요청)을 보낼 수 있다. 이를 위해서는 서버가 에이전트 카드에서 `pushNotifications` 기능을 광고해야 한다.

확장성 및 커스터마이제이션

A2A 프로토콜은 확장 메커니즘을 통해 에이전트가 핵심 A2A 사양을 넘어 추가 기능이나 데이터를 제공할 수 있도록 한다[28]. 에이전트는 AgentCard의 일부로 사용자 정의 프로토콜 확장을 선언할 수 있다. 이를 통해 특정 도메인이나 사용 사례에 맞는 특화된 기능을 구현할 수 있으며, 동시에 핵심 프로토콜의 상호운용성을 유지할 수 있다.

컨텍스트(`contextId`)는 서버가 생성하는 식별자로, 여러 관련 `Task` 객체를 논리적으로 그룹화하여 일련의 상호작용에 걸쳐 컨텍스트를 제공하는 데 사용될 수 있다[29]. 이는 복잡한 워크플로에서 관련 작업들을 추적하고 관리하는 데 유용하다.

실제 활용 사례 및 구성 예시

IT 헬프데스크 자동화 시스템

A2A 프로토콜의 가장 실용적인 활용 사례 중 하나는 엔터프라이즈 IT 헬프데스크 시스템의 자동화이다[30]. 이 시스템에서는 사용자 에이전트가 "소프트웨어 업데이트 후 노트북이 켜지지 않습니다"와 같은 요청을 받으면, 여러 원격 에이전트와의 A2A 협업을 통해 해결 워크플로를 조율한다.

구체적인 프로세스를 살펴보면, 클라이언트 에이전트는 먼저 하드웨어 진단 에이전트에게 장치 점검 작업을 전송한다[31]. 하드웨어가 정상으로 판명되면, 소프트웨어 롤백 에이전트로 에스컬레이션하여 최근 업데이트를 평가한다. 롤백이 실패하는 경우, 장치 교체 에이전트가 하드웨어 교체 프로세스를 시작한다. 각 에이전트는 독립적으로 작동하며, 에이전트 카드를 통해 자신의 기능을 노출하고, 구조화된 작업을 통해 비동기적으로 통신한다.

이 시스템의 특징은 에이전트들이 로그나 진단을 아티팩트로 스트리밍할 수 있고, 사용자 입력이 필요한 경우 흐름을 일시 중지하며, 작업 완료 시 푸시 업데이트를 전송한다는 점이다[32]. 이는 순수한 A2A 사용 사례로, 구조화된 도구나 API가 아닌 불투명하고 협업하는 에이전트들 간의 모든 조정이 이루어지며, 에이전트들이 개별적인 의사결정 로직과 네이티브 모달리티 처리를 가진 동료로 작동한다.

구매 컨시어지 시스템

Google Cloud에서 제시한 구매 컨시어지 시스템은 A2A 프로토콜의 실제 구현을 보여주는 대표적인 사례이다[33]. 이 시스템은 피자나 버거 주문을 도와주는 어시스턴트로, 사용자 요청을 적절한 전문 에

이전트에게 라우팅하는 컨시어지 에이전트와 각각 버거 주문과 피자 주문을 전문으로 하는 원격 에이전트들로 구성된다.

버거 에이전트의 에이전트 카드 구성을 살펴보면 다음과 같다:


```
{
 "name": "burger_seller_agent",
 "description": "Helps with creating burger orders",
 "url": "http://0.0.0.0:8080/",
 "version": "1.0.0",
 "capabilities": {
  "streaming": false,
  "pushNotifications": true,
  "stateTransitionHistory": false
 },
 "authentication": {
  "schemes": ["Basic"]
 },
 "defaultInputModes": ["text", "text/plain"],
 "defaultOutputModes": ["text", "text/plain"],
 "skills": [{
  "id": "create_burger_order",
  "name": "Burger Order Creation Tool",
  "description": "Helps with creating burger orders",
  "tags": ["burger order creation"],
  "examples": ["I want to order 2 classic cheeseburgers"]
 }]
}
```


이 시스템에서 작업 전송의 예시는 다음과 같다[34]:


```
{
 "jsonrpc": "2.0",
 "id": 1,
 "method": "tasks/send",
 "params": {
  "id": "de38c76d-d54c-436c-8b9f-4c2703648d64",
  "message": {
   "role": "user",
   "parts": [{
    "type": "text",
    "text": "I want to order 2 classic cheeseburgers"
   }]
  },
  "metadata": {}
 }
}
```

이 시스템은 Cloud Run에 배포되어 공개 URL을 통한 원격 에이전트 상호작용을 시뮬레이션하며, 에이전트 카드 발견부터 실제 주문 처리까지의 전체 A2A 워크플로를 보여준다[35].

금융 서비스의 대출 승인 시스템

금융 기관의 대출 승인 시스템은 A2A와 MCP가 어떻게 협력하여 복잡한 엔터프라이즈 워크플로를 자동화할 수 있는지 보여주는 훌륭한 예시이다[36]. 이 시스템에서 LoanProcessor 에이전트는 사용자로부터 대출 신청을 받으면, 먼저 MCP를 사용하여 구조화된 도구와 상호작용한다. 신용 점수 API를 호출하고, 보안 데이터 소스에서 은행 거래 내역을 검색하며, OCR을 사용하여 업로드된 문서를 검증한다.

구조화된 데이터를 수집한 후, LoanProcessor는 A2A를 사용하여 전문 에이전트들과 협업한다[37]. 위험 평가 에이전트는 신용 위험을 분석하고, 규정 준수 에이전트는 법적 요구사항을 확인하며, 승인 에이전트는 최종 승인 또는 거부 결정을 내린다. 각 에이전트는 자연어로 통신하며, 복잡한 추론과 상황별 판단을 수행한다.

이 사례는 MCP가 도구 상호운용성(에이전트가 구조화된 함수, API, 또는 도구를 호출)을 제공하고, A2A가 에이전트 상호운용성(에이전트가 자연어 또는 혼합 모달리티로 협업)을 제공하는 상호 보완적 관계를 명확히 보여준다[38].

의료 시스템의 환자 치료 조정

의료 분야에서 A2A 프로토콜은 병원 부서 간 AI 시스템의 즉석 조정을 통해 환자 치료를 혁신할 수 있는 잠재력을 보여준다[39]. 이 시스템에서는 환자 접수, 진단, 일정 관리를 담당하는 AI 에이전트들이 병원 부서 전반에 걸쳐 관리된다. A2A를 통해 이러한 시스템들이 컨텍스트를 공유하고 작업의 우선순위를 정할 수 있어, 관리적 병목 현상을 줄이고 환자 결과를 개선할 수 있다.

구체적인 시나리오를 살펴보면, 응급실 에이전트가 새로운 환자를 접수하면, 즉시 진단 에이전트와 통신하여 필요한 검사를 조율하고, 병상 관리 에이전트와 협력하여 적절한 병실을 배정하며, 의료진 스케줄링 에이전트와 조정하여 담당 의료진을 배정한다[40]. 이 모든 과정이 A2A 프로토콜을 통해 실시간으로 이루어지며, 각 에이전트는 자신의 전문 영역에서 최적의 결정을 내리면서도 전체 환자 치료 흐름을 고려한다.

공급망 관리 및 물류 최적화

물류 분야에서 A2A 프로토콜은 재고 관리와 배송 봇들의 실시간 조정을 통해 공급망 효율성을 크게 향상시킬 수 있다[41]. 한 물류 회사의 사례에서는 A2A를 통해 재고 에이전트와 배송 에이전트를 연결하여 실시간 조정을 구현했으며, 그 결과 배송 지연을 30% 감소시키고 공급망 효율성을 최적화했다고 보고되었다.

이 시스템에서 재고 관리 에이전트는 창고의 재고 수준을 모니터링하고, 수요 예측 에이전트는 향후 수요를 분석하며, 배송 최적화 에이전트는 최적의 배송 경로를 계산한다[42]. 주문이 들어오면 이 모든 에이전트들이 A2A 프로토콜을 통해 협업하여 가장 효율적인 주문 처리 방법을 결정한다. 예를 들어,

특정 상품의 재고가 부족한 경우, 재고 에이전트는 대체 창고를 찾고, 배송 에이전트는 새로운 경로를 계산하며, 고객 서비스 에이전트는 고객에게 업데이트를 제공한다.

JIRA 및 GitHub 통합 시스템

소프트웨어 개발 환경에서 A2A 프로토콜은 JIRA와 GitHub 같은 서로 다른 플랫폼의 에이전트들을 통합하여 개발 워크플로를 자동화할 수 있다[43]. 이 시스템은 JIRA 에이전트(이슈 생성, 업데이트, 상태 관리), GitHub 에이전트(코드 리뷰, PR 관리, 브랜치 작업), 그리고 두 시스템 간의 워크플로를 조율하는 통합 에이전트로 구성된다.

실제 워크플로에서는 개발자가 새로운 기능 요청을 제출하면, JIRA 에이전트가 이슈를 생성하고 우선순위를 설정한다[44]. 통합 에이전트는 이 정보를 GitHub 에이전트에게 전달하여 새로운 브랜치를 생성하고 초기 코드 구조를 설정한다. 개발이 진행되면서 GitHub 에이전트는 코드 변경사항을 모니터링하고, 테스트 결과를 JIRA 에이전트에게 전달하여 이슈 상태를 업데이트한다. 코드 리뷰가 완료되고 병합이 이루어지면, 두 에이전트가 협력하여 관련 이슈를 자동으로 닫고 릴리스 노트를 생성한다.

구현 기술 스택 및 배포 환경

A2A 프로토콜의 구현은 다양한 프로그래밍 언어와 프레임워크에서 가능하다[45]. Python 구현에서는 Flask나 FastAPI를 기반으로 하며, Python 3.12 이상이 권장된다. Java 구현도 가능하며, github.com/vishalmysore/a2ajava와 같은 순수 Java 구현이 제공된다. 또한 ADK, LangGraph, CrewAI 등 다양한 AI 에이전트 프레임워크와의 통합이 가능하다.

배포 환경 측면에서는 Google Cloud Run을 통한 공개 URL 배포가 일반적이며, 이를 통해 원격 에이전트 상호작용을 시뮬레이션할 수 있다[46]. Docker 컨테이너를 사용한 마이크로서비스 아키텍처로의 배포도 지원되며, 프로덕션 환경에서는 HTTPS가 필수적이다. 인증 방식으로는 Basic Authentication, OAuth 2.0, API Key, JWT 등 다양한 방식을 지원한다.

성능 및 확장성 고려사항

A2A 프로토콜의 실제 구현에서는 성능과 확장성이 중요한 고려사항이다[47]. 현재 멀티클라우드 환경에서 200-500ms의 지연시간이 보고되고 있으며, 이는 고빈도 거래나 응급 대응과 같이 실시간 의사결정이 중요한 산업에서는 문제가 될 수 있다. 그러나 2025년 Q3까지 100ms 미만으로 지연시간을 단축할 계획이 발표되어 있어 이러한 문제가 해결될 것으로 예상된다.

확장성 측면에서는 A2A 프로토콜이 HTTP와 JSON-RPC 2.0이라는 잘 확립된 표준을 기반으로 하기 때문에 기존 인프라와의 통합이 용이하다[48]. 또한 Server-Sent Events를 통한 스트리밍과 웹훅을 통한 푸시 알림 지원으로 다양한 사용 사례에 대응할 수 있다. 에이전트 카드를 통한 동적 발견 메커니즘은 시스템이 성장하고 변화하는 환경에서도 유연성을 제공한다.

최신 트렌드 및 미래 전망

2025년 AI 에이전트 생태계의 변화

2025년은 업계에서 "AI 에이전트의 해"로 불리며, AI 에이전트가 실험적 도구에서 엔터프라이즈 시스템의 필수 구성 요소로 진화하는 전환점이 되고 있다[49]. 이러한 변화의 중심에는 A2A 프로토콜이 있으며, 단순한 프롬프트와 응답 봇에서 사용자를 대신하여 자율적으로 행동하는 에이전트로의 패러다임 전환을 이끌고 있다. 이는 지능이 더 이상 정적 인터페이스나 단일 애플리케이션에 묶이지 않는 새로운 소프트웨어 설계 시대를 의미한다.

Microsoft의 Azure AI Foundry는 현재 70,000개 이상의 기업과 디지털 네이티브 회사에서 사용되고 있으며, 여기에는 Atomicwork, Epic, Fujitsu, Gainsight, H&R Block, LG Electronics가 포함된다[50]. 특히 주목할 점은 단 4개월 만에 10,000개 이상의 조직이 새로운 Agent Service를 채택하여 에이전트 시스템을 구축, 배포, 확장하고 있다는 것이다. 또한 Fortune 500의 90%를 포함하여 230,000개 이상의 조직이 이미 Microsoft Copilot Studio를 사용하고 있어, 에이전트 기술의 급속한 확산을 보여준다.

주요 기업들의 A2A 채택 전략

Google이 2025년 4월 A2A 프로토콜을 공식 발표한 후, Microsoft는 불과 한 달 만인 5월에 A2A 지원을 발표하며 업계의 빠른 표준 채택을 보여주었다[51]. Microsoft의 A2A 통합 계획은 포괄적이며 전략적이다. Azure AI Foundry에서는 고객들이 내부 코파일럿, 파트너 도구, 프로덕션 인프라에 걸친 복잡한 멀티 에이전트 워크플로를 구축할 수 있으면서도 거버넌스와 SLA를 유지할 수 있도록 지원한다. Copilot Studio 에이전트들은 다른 플랫폼에서 구축되거나 Microsoft 외부에서 호스팅되는 에이전트를 포함하여 외부 에이전트를 안전하게 호출할 수 있게 된다.

Microsoft의 접근 방식에서 특히 주목할 점은 보안과 거버넌스에 대한 강조이다[52]. 모든 A2A 호출은 Microsoft Entra, 상호 TLS, Azure AI Content Safety, 그리고 완전한 감사 로그를 포함한 엔터프라이즈급 보안 장치를 통과한다. 이는 에이전트 생태계가 더욱 개방적이고 분산적으로 성장하더라도 안전성, 규정 준수, 책임성이 일급 시민으로 유지된다는 것을 의미한다.

Microsoft는 또한 Semantic Kernel에 A2A 샘플 코드를 제공하여 개발자들이 쉽게 시작할 수 있도록 지원하고 있다[53]. Python으로 제공되는 이 샘플은 두 개의 로컬 에이전트가 A2A 프로토콜을 사용하여 협업하는 방법을 보여주며, 여행 일정 계획과 환율 변환을 처리하는 에이전트들이 사용자 정의 오케스트레이션 코드 없이도 원활한 상호운용성을 달성하는 것을 시연한다.

업계 인식의 근본적 변화

A2A 프로토콜의 등장과 함께 업계에서는 상호운용성에 대한 인식이 근본적으로 변화하고 있다[54]. 과거에는 상호운용성이 선택사항으로 여겨졌지만, 이제는 필수 요소로 인식되고 있다. 고객들이 이러한 시스템을 확장함에 따라, 벤더, 클라우드, 데이터 사일로에 걸쳐 작업을 조율하는 에이전트를 원하고 있으며, 락인 없이 제어, 가시성, 신뢰를 원하고 있다.

이러한 변화는 에이전트 컴퓨팅이 단순한 트렌드가 아닌 근본적인 변화라는 인식에서 비롯된다[55]. 에이전트 컴퓨팅은 소프트웨어가 구축되는 방식, 의사결정이 이루어지는 방식, 그리고 가치가 창출되는 방식을 변화시킨다. 최고의 에이전트들은 하나의 앱이나 클라우드에 존재하지 않을 것이며, 모델, 도메인, 생태계에 걸쳐 작업 흐름에서 작동할 것이다.

기술적 발전 로드맵

A2A 프로토콜의 기술적 발전은 명확한 로드맵을 따르고 있다[56]. 현재 멀티클라우드 환경에서 200-500ms의 지연시간이 보고되고 있지만, 2025년 Q3까지 100ms 미만으로 단축할 계획이다. 이는 고빈도 거래나 응급 대응과 같이 실시간 의사결정이 중요한 산업에서의 활용 가능성을 크게 높일 것이다.

2025년 Q3에는 향상된 성능을 제공하는 프로덕션 준비 버전이 출시될 예정이며, Q4에는 AWS Bedrock과 Meta의 Llama 에이전트와의 통합이 목표로 설정되어 있어 A2A의 범위가 크게 확장될 것으로 예상된다[57]. 2026년까지는 머신러닝 기반 최적화와 글로벌 거버넌스 프레임워크가 도입되어 A2A를 범용 표준으로 확고히 자리잡게 하면서도 지역별 규제 준수와 혁신 간의 균형을 맞출 계획이다.

새로운 기능 및 능력의 진화

A2A 프로토콜의 미래 발전 방향에서 주목할 만한 것은 적응형 지능(Adaptive Intelligence)의 개발이다[58]. 이는 동적으로 통신 전략을 재구성할 수 있는 에이전트를 개발하는 것으로, 상황과 요구사항에 따라 최적의 협업 방식을 선택할 수 있게 한다. 또한 윤리적 AI에 대한 강조가 증가하고 있으며, 책임감 있는 AI 개발을 위한 가이드라인이 강화되고 있다.

머신러닝 기반 최적화는 2026년까지 도입될 예정으로, 이를 통해 에이전트 간 통신의 효율성과 정확성이 크게 향상될 것으로 예상된다[59]. 글로벌 거버넌스 프레임워크의 개발도 중요한 발전 방향으로, 국제적 규제 준수를 위한 표준화된 접근 방식을 제공할 것이다.

실제 비즈니스 임팩트 측정

A2A 프로토콜의 실제 비즈니스 임팩트는 이미 여러 분야에서 측정 가능한 결과로 나타나고 있다[60]. 물류 분야에서는 실시간 조정을 통해 배송 지연이 30% 감소했다고 보고되었으며, 의료 분야에서는 병원 대기 시간을 50% 단축한 사례가 있다. 공급망 관리에서는 비용이 대폭 절감되고 효율성이 향상되었으며, 금융 규정 준수 분야에서는 오류가 최소화되고 보고 주기가 가속화되었다.

이러한 성과들은 A2A 프로토콜이 단순한 기술적 혁신을 넘어서 실질적인 비즈니스 가치를 창출하고 있음을 보여준다[61]. 특히 ExO Insight에서 제시한 바와 같이, A2A는 10배 성장을 추진하는 촉매 역할을 하며, 사일로를 해체하고 규모에 따른 효율성을 추진하고 있다.

도전 과제 및 해결 방안

A2A 프로토콜의 광범위한 채택에는 여전히 해결해야 할 도전 과제들이 있다[62]. 기술적 측면에서는 지연시간 문제가 가장 중요한 과제로, 실시간 의사결정이 중요한 산업에서는 현재의 지연시간이 문제

가 될 수 있다. 보안 표준화도 중요한 과제로, 조직 경계를 넘나드는 에이전트 간 신뢰 표준화가 필요하다. 또한 개발자를 위한 디버깅 및 최적화 도구의 개선이 필요하다.

윤리적 고려사항도 중요한 도전 과제이다[63]. 인간 감독과 자동화된 의사결정 간의 균형을 유지하는 것이 중요하며, 반복적 업무 자동화로 인한 인력 재배치 및 재교육이 필요하다. AI 협업에 대한 과도한 의존으로 인한 인간 감독 약화 우려도 해결해야 할 과제이다.

기술 융합 및 생태계 발전

A2A 프로토콜의 미래는 다른 지수적 기술들과의 융합에서 더욱 큰 잠재력을 보여준다[64]. 블록체인과의 결합을 통해 안전하고 투명한 에이전트 상호작용이 가능해지며, IoT와의 통합을 통해 실시간 데이터 공유가 최적화된다. 엣지 컴퓨팅과의 결합은 분산 환경에서의 에이전트 협업을 최적화할 것이다.

생태계 발전 측면에서는 A2A가 AI 에이전트 협업의 범용 표준으로 자리잡을 전망이다[65]. 주요 클라우드 플랫폼들의 네이티브 A2A 지원이 확산되고, 더 풍부한 SDK, 디버깅 도구, 모니터링 솔루션이 제공될 것이다. 오픈소스 커뮤니티를 통한 지속적인 프로토콜 개선도 기대된다.

산업별 전망 및 혁신 시나리오

각 산업별로 A2A 프로토콜의 적용은 고유한 혁신 시나리오를 만들어낼 것으로 예상된다[66]. 의료 분야에서는 부서 간 AI 시스템의 즉석 조정을 통해 환자 치료가 혁신될 것이며, 제조업에서는 글로벌 공급망에서 봇들의 실시간 재고 및 물류 조정이 가능해질 것이다. 금융 분야에서는 분산 원장에서 작동하는 감사 에이전트들이 투명성을 보장할 것이며, 미디어 분야에서는 스크립트부터 화면까지 콘텐츠 제작 프로세스가 간소화될 것이다.

조직 준비 전략 및 권장사항

조직들이 A2A 프로토콜의 혜택을 최대화하기 위해서는 체계적인 준비 전략이 필요하다[67]. 단기적으로는 사일로나 비효율성이 있는 워크플로를 식별하여 파일럿 프로젝트를 시작하고, Azure AI Foundry나 Google Vertex AI 등 A2A 통합 플랫폼과의 파트너십을 구축해야 한다. AI 기반 프로세스 감독을 위한 직원 교육과 A2A GitHub 워킹 그룹 리소스 활용도 중요하다.

장기적으로는 조직 전반에 걸친 A2A 기반 멀티에이전트 시스템 구축, 업계 표준 개발 및 거버넌스에의 적극적 참여, A2A를 활용한 새로운 비즈니스 모델 및 서비스 개발, 그리고 국제적 규제 환경에 맞는 A2A 기반 솔루션 구축이 필요하다[68].

결론 및 권장사항

종합 분석 및 핵심 인사이트

A2A(Agent to Agent) 프로토콜은 AI 에이전트 생태계에서 상호운용성 문제를 해결하는 혁신적인 솔루션으로 자리잡고 있다. 본 보고서의 분석을 통해 A2A 프로토콜이 단순한 기술적 표준을 넘어서 AI 에

이전트 협업의 새로운 패러다임을 제시하고 있음을 확인할 수 있었다. Google의 주도하에 시작된 이 프로토콜은 Microsoft를 비롯한 주요 기술 기업들의 빠른 채택과 50여 개 기술 파트너들의 참여를 통해 업계 표준으로 발전하고 있다.

A2A 프로토콜의 가장 중요한 특징은 에이전트들이 서로의 내부 구조나 도구를 공유하지 않고도 효과적으로 협업할 수 있도록 하는 "불투명한 협업" 모델이다. 이는 기존의 도구 중심 접근법과는 근본적으로 다른 패러다임으로, 에이전트를 진정한 협업 파트너로 취급한다. HTTP, JSON-RPC 2.0, Server-Sent Events와 같은 잘 확립된 웹 표준을 기반으로 하여 기존 인프라와의 통합이 용이하며, 엔터프라이즈급 보안과 확장성을 제공한다.

실제 활용 사례 분석에서는 IT 헬프데스크 자동화, 구매 컨시어지 시스템, 금융 서비스의 대출 승인, 의료 시스템의 환자 치료 조정, 공급망 관리 등 다양한 분야에서 A2A 프로토콜이 실질적인 비즈니스 가치를 창출하고 있음을 확인했다. 특히 물류 분야에서 30% 배송 지연 감소, 의료 분야에서 50% 대기 시간 단축 등 측정 가능한 성과가 보고되고 있어 프로토콜의 실용성을 입증하고 있다.

전략적 권장사항

단기 전략 (2025-2026)

조직들은 A2A 프로토콜 도입을 위한 단계적 접근을 취해야 한다. 먼저 현재 조직 내에서 사일로화되어 있거나 비효율성이 존재하는 워크플로를 식별하고, 이를 대상으로 한 파일럿 프로젝트를 시작해야 한다. 공급망 물류, 고객 서비스 핸드오프, 규정 준수 보고 등이 적합한 후보가 될 수 있다. Azure AI Foundry나 Google Vertex AI와 같이 A2A를 이미 통합하고 있는 플랫폼과의 파트너십을 구축하여 기술적 진입 장벽을 낮추는 것이 중요하다.

인력 개발 측면에서는 AI 기반 프로세스를 감독할 수 있는 직원 교육이 필수적이다. 이는 기술적 스킬뿐만 아니라 AI 에이전트 협업에서 인간의 역할과 책임에 대한 이해를 포함해야 한다. A2A GitHub 워킹 그룹의 오픈소스 리소스를 적극 활용하여 기존 시스템과의 정렬을 보장하고, 커뮤니티의 모범 사례를 학습해야 한다.

중기 전략 (2026-2027)

중기적으로는 파일럿 프로젝트의 성과를 바탕으로 A2A 기반 솔루션을 조직 전반으로 확장해야 한다. 이 과정에서 시간 절약이나 오류 감소와 같은 구체적인 성과 지표를 측정하여 더 광범위한 도입을 위한 비즈니스 케이스를 구축해야 한다. 또한 A2A 프로토콜과 다른 지수적 기술들(블록체인, IoT, 엣지 컴퓨팅)과의 융합 가능성을 탐색하여 혁신적인 솔루션을 개발해야 한다.

조직 문화 측면에서는 Community & Crowd와 같은 ExO 원칙을 활용하여 A2A 통합을 위한 솔루션을 크라우드소싱하고, 적응성을 촉진하는 문화를 조성해야 한다. 이는 자동화와 감독 간의 이중 초점을 유지하면서 멀티 에이전트 생태계가 가져오는 사회적 변화를 탐색하는 데 도움이 될 것이다.

장기 전략 (2027년 이후)

장기적으로는 A2A를 활용한 새로운 비즈니스 모델과 서비스를 개발하여 경쟁 우위를 확보해야 한다. 이는 기존 프로세스의 자동화를 넘어서 완전히 새로운 가치 창출 방식을 모색하는 것을 의미한다. 업계 표준 개발과 거버넌스에 적극적으로 참여하여 A2A 프로토콜의 미래 방향에 영향을 미치고, 조직의 요구사항이 반영되도록 해야 한다.

국제적 확장을 고려하는 조직들은 글로벌 거버넌스 프레임워크 개발에 참여하고, 다양한 지역의 규제 환경에 맞는 A2A 기반 솔루션을 구축해야 한다. 이는 2026년까지 도입될 예정인 글로벌 거버넌스 프레임워크와 연계하여 진행되어야 한다.

위험 관리 및 고려사항

A2A 프로토콜 도입 과정에서 조직들이 고려해야 할 주요 위험 요소들이 있다. 기술적 위험으로는 현재 200-500ms의 지연시간이 실시간 의사결정이 중요한 업무에 미치는 영향을 평가해야 한다. 2025년 Q3까지 100ms 미만으로 개선될 예정이지만, 그 이전까지는 적절한 대안을 마련해야 한다.

보안 위험 관리도 중요하다. 조직 경계를 넘나드는 에이전트 간 통신에서 데이터 보안과 프라이버시를 보장하기 위한 정책과 절차를 수립해야 한다. 특히 금융이나 의료와 같은 민감한 분야에서는 더욱 엄격한 보안 표준이 필요하다.

인력 관리 측면에서는 자동화로 인한 일자리 변화에 대비한 재교육 프로그램을 마련해야 한다. 반복적인 업무가 자동화되면서 전략적이거나 창의적인 역할로의 전환을 지원하는 것이 중요하다. 또한 AI 시스템에 대한 과도한 의존을 방지하고 인간의 감독과 판단력을 유지하는 메커니즘을 구축해야 한다.

미래 전망 및 기회

A2A 프로토콜의 미래는 매우 밝다. 2025년 Q3의 프로덕션 준비 버전 출시와 Q4의 AWS Bedrock, Meta Llama 에이전트와의 통합은 프로토콜의 범위를 크게 확장할 것이다. 2026년까지 도입될 머신러닝 기반 최적화와 글로벌 거버넌스 프레임워크는 A2A를 진정한 범용 표준으로 확립할 것이다.

특히 주목할 점은 A2A가 단순한 기술 표준을 넘어서 새로운 비즈니스 생태계를 만들어낼 잠재력이다. 에이전트들이 조직과 클라우드 경계를 넘나들며 협업하는 환경에서는 완전히 새로운 형태의 서비스와 비즈니스 모델이 등장할 수 있다. 이는 디지털 트랜스포메이션을 넘어서 "에이전트 트랜스포메이션"이라고 할 수 있는 새로운 패러다임을 제시한다.

최종 권고

A2A 프로토콜은 AI 에이전트 시대의 핵심 인프라가 될 것이다. 조직들은 이 기술을 단순한 자동화 도구로 보지 말고, 새로운 형태의 지능적 협업을 가능하게 하는 플랫폼으로 인식해야 한다. 성공적인 A2A 도입을 위해서는 기술적 준비뿐만 아니라 조직 문화, 인력 개발, 비즈니스 프로세스의 전면적인 재검토가 필요하다.

가장 중요한 것은 A2A 프로토콜이 제공하는 기회를 조기에 인식하고 적극적으로 활용하는 것이다. 2025년이 "AI 에이전트의 해"로 불리는 만큼, 이 시기에 A2A 기반 솔루션을 구축하고 경험을 축적하

는 조직들이 향후 경쟁에서 유리한 위치를 점할 것이다. A2A 프로토콜은 단순한 기술적 혁신이 아닌, 미래 비즈니스 환경에서 생존과 성장을 위한 필수 요소가 될 것이다.

참고문헌

[1] Google Developers Blog. (2025, April 9). Agent2Agent(A2A) 프로토콜 발표. https://developers.googleblog.com/ko/a2a-a-new-era-of-agent-interoperability/

[2] Google Developers Blog. (2025, April 9). Announcing the Agent2Agent Protocol (A2A). https://developers.googleblog.com/en/a2a-a-new-era-of-agent-interoperability/

[3] Google Developers Blog. (2025, April 9). Agent2Agent(A2A) 프로토콜 발표. https://developers.googleblog.com/ko/a2a-a-new-era-of-agent-interoperability/

[4] Agent2Agent Protocol Documentation. Key Concepts. https://a2aproject.github.io/A2A/topics/key-concepts/

[5] Agent2Agent Protocol Documentation. Specification. https://a2aproject.github.io/A2A/latest/specification/

[6] Google Developers Blog. (2025, April 9). Agent2Agent(A2A) 프로토콜 발표. https://developers.googleblog.com/ko/a2a-a-new-era-of-agent-interoperability/

[7] Kanaries Documentation. (2025, April 10). Google의 A2A 프로토콜을 사용한 두 개의 Python 에이전트 빌드하기. https://docs.kanaries.net/ko/articles/build-agent-with-a2a

[8] DataCamp Blog. (2025, May 6). Agent2Agent (A2A): Definition, Examples, MCP Comparison. https://www.datacamp.com/blog/a2a-agent2agent

[9] Google Developers Blog. (2025, April 9). Agent2Agent(A2A) 프로토콜 발표. https://developers.googleblog.com/ko/a2a-a-new-era-of-agent-interoperability/

[10] Google Developers Blog. (2025, April 9). Agent2Agent(A2A) 프로토콜 발표. https://developers.googleblog.com/ko/a2a-a-new-era-of-agent-interoperability/

[11] Microsoft Cloud Blog. (2025, May 7). Empowering multi-agent apps with the open Agent2Agent (A2A) protocol. https://www.microsoft.com/en-us/microsoft-cloud/blog/2025/05/07/empowering-multi-agent-apps-with-the-open-agent2agent-a2a-protocol/

[12] Agent2Agent Protocol Documentation. Key Concepts. https://a2aproject.github.io/A2A/topics/key-concepts/

[13] Agent2Agent Protocol Documentation. Key Concepts. https://a2aproject.github.io/A2A/topics/key-concepts/

[14] Agent2Agent Protocol Documentation. Specification. https://a2aproject.github.io/A2A/latest/specification/

[15] Agent2Agent Protocol Documentation. Specification. https://a2aproject.github.io/A2A/latest/specification/

[16] Agent2Agent Protocol Documentation. Key Concepts. https://a2aproject.github.io/A2A/topics/key-concepts/

[17] Agent2Agent Protocol Documentation. Specification. https://a2aproject.github.io/A2A/latest/specification/

[18] Agent2Agent Protocol Documentation. Key Concepts. https://a2aproject.github.io/A2A/topics/key-concepts/

[19] Kanaries Documentation. (2025, April 10). Google의 A2A 프로토콜을 사용한 두 개의 Python 에이전트 빌드하기. https://docs.kanaries.net/ko/articles/build-agent-with-a2a

[20] Kanaries Documentation. (2025, April 10). Google의 A2A 프로토콜을 사용한 두 개의 Python 에이전트 빌드하기. https://docs.kanaries.net/ko/articles/build-agent-with-a2a

[21] Agent2Agent Protocol Documentation. Key Concepts. https://a2aproject.github.io/A2A/topics/key-concepts/

[22] Agent2Agent Protocol Documentation. Key Concepts. https://a2aproject.github.io/A2A/topics/key-concepts/

[23] Agent2Agent Protocol Documentation. Specification. https://a2aproject.github.io/A2A/latest/specification/

[24] Agent2Agent Protocol Documentation. Specification. https://a2aproject.github.io/A2A/latest/specification/

[25] Agent2Agent Protocol Documentation. Key Concepts. https://a2aproject.github.io/A2A/topics/key-concepts/

[26] Agent2Agent Protocol Documentation. Key Concepts. https://a2aproject.github.io/A2A/topics/key-concepts/

[27] Agent2Agent Protocol Documentation. Key Concepts. https://a2aproject.github.io/A2A/topics/key-concepts/

[28] Agent2Agent Protocol Documentation. Key Concepts. https://a2aproject.github.io/A2A/topics/key-concepts/

[29] Agent2Agent Protocol Documentation. Key Concepts. https://a2aproject.github.io/A2A/topics/key-concepts/

[30] DataCamp Blog. (2025, May 6). Agent2Agent (A2A): Definition, Examples, MCP Comparison. https://www.datacamp.com/blog/a2a-agent2agent

[31] DataCamp Blog. (2025, May 6). Agent2Agent (A2A): Definition, Examples, MCP Comparison. https://www.datacamp.com/blog/a2a-agent2agent

[32] DataCamp Blog. (2025, May 6). Agent2Agent (A2A): Definition, Examples, MCP Comparison. https://www.datacamp.com/blog/a2a-agent2agent

[33] Medium - Google Cloud Community. (2025, May 15). Exploring Agent2Agent (A2A) Protocol with Purchasing Concierge Use Case on Cloud Run. https://medium.com/google-cloud/exploring-agent2agent-a2a-protocol-with-purchasing-concierge-use-case-on-cloud-run-36f4b896eadf

[34] Medium - Google Cloud Community. (2025, May 15). Exploring Agent2Agent (A2A) Protocol with Purchasing Concierge Use Case on Cloud Run. https://medium.com/google-cloud/exploring-agent2agent-a2a-protocol-with-purchasing-concierge-use-case-on-cloud-run-36f4b896eadf

[35] Medium - Google Cloud Community. (2025, May 15). Exploring Agent2Agent (A2A) Protocol with Purchasing Concierge Use Case on Cloud Run. https://medium.com/google-cloud/exploring-agent2agent-a2a-protocol-with-purchasing-concierge-use-case-on-cloud-run-36f4b896eadf

[36] DataCamp Blog. (2025, May 6). Agent2Agent (A2A): Definition, Examples, MCP Comparison. https://www.datacamp.com/blog/a2a-agent2agent

[37] DataCamp Blog. (2025, May 6). Agent2Agent (A2A): Definition, Examples, MCP Comparison. https://www.datacamp.com/blog/a2a-agent2agent

[38] DataCamp Blog. (2025, May 6). Agent2Agent (A2A): Definition, Examples, MCP Comparison. https://www.datacamp.com/blog/a2a-agent2agent

[39] ExO Insight. (2025, June 11). Revolutionizing Enterprise AI: A2A Protocol. https://publish.openexo.com/revolutionizing-enterprise-ai-a2a-protocol/

[40] ExO Insight. (2025, June 11). Revolutionizing Enterprise AI: A2A Protocol. https://publish.openexo.com/revolutionizing-enterprise-ai-a2a-protocol/

[41] ExO Insight. (2025, June 11). Revolutionizing Enterprise AI: A2A Protocol. https://publish.openexo.com/revolutionizing-enterprise-ai-a2a-protocol/

[42] ExO Insight. (2025, June 11). Revolutionizing Enterprise AI: A2A Protocol. https://publish.openexo.com/revolutionizing-enterprise-ai-a2a-protocol/

[43] Searce Blog. (2025, April 30). Building an Agentic System with Google's A2A Protocol: JIRA and GitHub Integration. https://blog.searce.com/building-an-agentic-system-with-googles-a2a-protocol-jira-and-github-integration-aedde4ca71cc

[44] Searce Blog. (2025, April 30). Building an Agentic System with Google's A2A Protocol: JIRA and GitHub Integration. https://blog.searce.com/building-an-agentic-system-with-googles-a2a-protocol-jira-and-github-integration-aedde4ca71cc

[45] Kanaries Documentation. (2025, April 10). Google의 A2A 프로토콜을 사용한 두 개의 Python 에이전트 빌드하기. https://docs.kanaries.net/ko/articles/build-agent-with-a2a

[46] Medium - Google Cloud Community. (2025, May 15). Exploring Agent2Agent (A2A) Protocol with Purchasing Concierge Use Case on Cloud Run. https://medium.com/google-cloud/exploring-agent2agent-a2a-protocol-with-purchasing-concierge-use-case-on-cloud-run-36f4b896eadf

[47] ExO Insight. (2025, June 11). Revolutionizing Enterprise AI: A2A Protocol. https://publish.openexo.com/revolutionizing-enterprise-ai-a2a-protocol/

[48] Agent2Agent Protocol Documentation. Specification. https://a2aproject.github.io/A2A/latest/specification/

[49] Microsoft Cloud Blog. (2025, May 7). Empowering multi-agent apps with the open Agent2Agent (A2A) protocol. https://www.microsoft.com/en-us/microsoft-cloud/blog/2025/05/07/empowering-multi-agent-apps-with-the-open-agent2agent-a2a-protocol/

[50] Microsoft Cloud Blog. (2025, May 7). Empowering multi-agent apps with the open Agent2Agent (A2A) protocol. https://www.microsoft.com/en-us/microsoft-cloud/blog/2025/05/07/empowering-multi-agent-apps-with-the-open-agent2agent-a2a-protocol/

[51] Microsoft Cloud Blog. (2025, May 7). Empowering multi-agent apps with the open Agent2Agent (A2A) protocol. https://www.microsoft.com/en-us/microsoft-cloud/blog/2025/05/07/empowering-multi-agent-apps-with-the-open-agent2agent-a2a-protocol/

[52] Microsoft Cloud Blog. (2025, May 7). Empowering multi-agent apps with the open Agent2Agent (A2A) protocol. https://www.microsoft.com/en-us/microsoft-cloud/blog/2025/05/07/empowering-multi-agent-apps-with-the-open-agent2agent-a2a-protocol/

[53] Microsoft Cloud Blog. (2025, May 7). Empowering multi-agent apps with the open Agent2Agent (A2A) protocol. https://www.microsoft.com/en-us/microsoft-cloud/blog/2025/05/07/empowering-multi-agent-apps-with-the-open-agent2agent-a2a-protocol/

[54] Microsoft Cloud Blog. (2025, May 7). Empowering multi-agent apps with the open Agent2Agent (A2A) protocol. https://www.microsoft.com/en-us/microsoft-cloud/blog/2025/05/07/empowering-multi-agent-apps-with-the-open-agent2agent-a2a-protocol/

[55] Microsoft Cloud Blog. (2025, May 7). Empowering multi-agent apps with the open Agent2Agent (A2A) protocol. https://www.microsoft.com/en-us/microsoft-cloud/blog/2025/05/07/empowering-multi-agent-apps-with-the-open-agent2agent-a2a-protocol/

[56] ExO Insight. (2025, June 11). Revolutionizing Enterprise AI: A2A Protocol. https://publish.openexo.com/revolutionizing-enterprise-ai-a2a-protocol/

[57] ExO Insight. (2025, June 11). Revolutionizing Enterprise AI: A2A Protocol. https://publish.openexo.com/revolutionizing-enterprise-ai-a2a-protocol/

[58] BytePlus. Skills Needed for A2A Protocol Development in 2025. https://www.byteplus.com/en/topic/551304

[59] ExO Insight. (2025, June 11). Revolutionizing Enterprise AI: A2A Protocol. https://publish.openexo.com/revolutionizing-enterprise-ai-a2a-protocol/

[60] ExO Insight. (2025, June 11). Revolutionizing Enterprise AI: A2A Protocol. https://publish.openexo.com/revolutionizing-enterprise-ai-a2a-protocol/

[61] ExO Insight. (2025, June 11). Revolutionizing Enterprise AI: A2A Protocol. https://publish.openexo.com/revolutionizing-enterprise-ai-a2a-protocol/

[62] ExO Insight. (2025, June 11). Revolutionizing Enterprise AI: A2A Protocol. https://publish.openexo.com/revolutionizing-enterprise-ai-a2a-protocol/

[63] ExO Insight. (2025, June 11). Revolutionizing Enterprise AI: A2A Protocol. https://publish.openexo.com/revolutionizing-enterprise-ai-a2a-protocol/

[64] ExO Insight. (2025, June 11). Revolutionizing Enterprise AI: A2A Protocol. https://publish.openexo.com/revolutionizing-enterprise-ai-a2a-protocol/

[65] ExO Insight. (2025, June 11). Revolutionizing Enterprise AI: A2A Protocol. https://publish.openexo.com/revolutionizing-enterprise-ai-a2a-protocol/

[66] ExO Insight. (2025, June 11). Revolutionizing Enterprise AI: A2A Protocol. https://publish.openexo.com/revolutionizing-enterprise-ai-a2a-protocol/

[67] ExO Insight. (2025, June 11). Revolutionizing Enterprise AI: A2A Protocol. https://publish.openexo.com/revolutionizing-enterprise-ai-a2a-protocol/

[68] ExO Insight. (2025, June 11). Revolutionizing Enterprise AI: A2A Protocol. https://publish.openexo.com/revolutionizing-enterprise-ai-a2a-protocol/

보고서 끝