1. Agent Builder 개요 및 배경

Vertex AI Agent Builder는 **Google Cloud Vertex AI** 플랫폼의 최신 구성 요소로서, 기업이 AI 에이전트를 손쉽게 빌드하고 배포할 수 있게 해주는 도구 모음입니다 cloud.google.com. 이는 AI 도입의 어느 단계에 있든지 기존 시스템과 워크플로를 교란시키지 않으면서 멀티 에이전트 환경을 구축하도록 지원하는 플랫폼입니다 cloud.google.com. 즉, 새로운 기술 스택으로 전환하거나 업무 프로세스를 변경하지 않아도 현재의 비즈니스 프로세스 위에 지능형 에이전트들을 추가할 수 있게 해줍니다. Google은 2025년 Cloud Next 콘퍼런스에서 이 Agent Builder를 발표하며, **Agent Development Kit (ADK)**라는 오픈소스 프레임워크를 함께 소개했습니다 developers.googleblog.com. ADK는 대규모 언어 모델(LLM)을 활용한 멀티 에이전트 시스템 개발을 단순화하고 개발자에게 더 큰 **유연성과 제어권**을 제공하기 위해 설계되었습니다developers.googleblog.com. Vertex AI Agent Builder는 이러한 ADK를 중심으로, 에이전트 샘플 모음인 **Agent Garden**, 다양한 **도구 툴킷**, 그리고 **Agent Engine**이라 불리는 관리형 런타임을 포함하여 전반적인 AI 에이전트 개발 **생태계**를 제공합니다 cloud.google.comcloud.google.com. 이를 통해 기업은 단일 목적 챗봇을 넘어 여러 에이전트가 상호 작용하며 복잡한 작업을 수행하는 **지능형 에이전트 환경**을 구축할 수 있게 되었습니다. 이러한 배경에는 챗봇과 자동화 기술이 단순 Q&A를 넘어 보다 **자율적이고 협력적인 AI**로 발전하는 산업 트렌드가 자리하고 있으며, Google의 Vertex AI Agent Builder는 그러한 멀티 에이전트 시스템을 손쉽게 현실화하기 위한 해결책으로 등장했습니다.

2. 주요 기능 및 장점

Vertex AI Agent Builder는 **다양한 핵심 기능**을 통해 개발자와 기업이 강력한 AI 에이전트를 만들고 활용할 수 있도록 해줍니다. 아래에 주요 기능과 그 장점을 정리했습니다:

- **Agent Development Kit (ADK)**: ADK는 Python 기반의 오픈소스 에이전트 개발 프레임워크로, **100줄 미만의 직관적인 코드로도** 프로덕션 수준의 에이전트를 제작할 수 있을 만큼 사용이 간편합니다cloud.google.com. 개발자는 ADK를 통해 에이전트의 **사고 과정, 추론 및 협업 방식**을 직접 정의하고 제어할 수 있으며, 결정론적 가드레일(안전 장치)과 오케스트레이션 제어를 적용하여 에이전트의 동작을 세밀하게 관리할 수 있습니다cloud.google.com. 또한 ADK에는 **양방향 오디오·비디오 스트리밍** 기능이 내장되어 있어, 사람이 대화하듯 자연스러운 상호작용(음성 대화나 화상 인터페이스까지 지원)을 구현할 수 있다는 장점이 있습니다 cloud.google.com.
- **Agent Garden**: Agent Garden는 **샘플 에이전트와 도구 모음**의 라이브러리로, 에

이전트 개발을 빠르게 시작할 수 있게 도와주는 리소스입니다cloud.google.com. 미리 구축된 다양한 **에이전트 예제, 패턴, 도구 템플릿**을 제공하여 개발 속도를 높이고, 실제 사례를 통해 모범적인 에이전트 설계를 학습할 수 있습니다. 이를 활용하면 제로부터 모든 것을 만드는 부담을 줄이고, **반복적인 작업을 자동화**하거나 일반적인 대화 시나리오를 빠르게 구성할 수 있습니다.

- **멀티 모델 및 오픈 생태계 지원**: Vertex AI Agent Builder는 특정 LLM에 종속되지 않고 **여러 언어 모델을 유연하게 활용**할 수 있다는 큰 장점이 있습니다. 예를 들어, Google의 차세대 모델인 **Gemini**를 비롯하여 Vertex AI **모델 가든**의 다양한 사전 학습 모델들을 선택하여 사용할 수 있습니다developers.googleblog.com. 또한 ADK는 **LiteLLM 통합**을 통해 Anthropic(Claude), Meta(Llama 2), AI21 등 외부 제공업체의 모델까지 손쉽게 사용할 수 있어, **최적의 모델을 과제에 맞게 선택**할 수 있습니다developers.googleblog.comdevelopers.googleblog.com. 나아가 ADK 자체가 오픈소스로 공개되어 있고 **LangChain, LangGraph, LlamaIndex, Crew**.ai 등의 인기 오픈소스 프레임워크와도 연동 가능하므로, 기존에 이러한 도구로 구축된 에이전트라도 Vertex AI 플랫폼에 쉽게 통합하거나 함께 운영할 수 있습니다 cloud.google.com. 이처럼 **개방형 생태계**를 지향함으로써 개발자는 **벤더 종속 없이** 자유롭게 에이전트를 개발하고 다양한 도구를 조합할 수 있습니다.

- **Agent2Agent (A2A) 프로토콜**: A2A는 Vertex AI Agent Builder가 제시하는 **개방형 에이전트 통신 표준**으로, 서로 다른 프레임워크나 환경에서 만들어진 에이전트들끼리도 **원활하게 소통하고 협력**할 수 있도록 해줍니다cloud.google.com. 마치 API가 애플리케이션 간 통신을 표준화하듯이, A2A는 에이전트 간의 상호작용을 표준화합니다cloud.google.com. 이를 사용하면 ADK로 만든 에이전트와 LangChain이나 다른 업체의 에이전트가 **동일한 팀**으로 연결되어 작업을 분담하거나 정보를 교환할 수 있습니다softude.com. 여러 에이전트를 **분산된 팀**처럼 활용함으로써 개별로 고립된 봇보다 **복잡한 문제를 협력적으로 해결**할 수 있으며, 기업은 특정 에이전트 프레임워크에 대한 투자 위험 없이 다양한 에이전트 기술을 함께 활용할 수 있습니다cloud.google.com. 현재 Box, Deloitte, Elastic, Salesforce, UiPath 등 **50여 개 파트너**가 이 A2A 표준을 지원하기 위한 생태계 조성에 참여하고 있을 정도로 업계의 주목을 받고 있습니다cloud.google.com.

- **다양한 **도구(Tool) 및 데이터 연동**: 에이전트에게 **외부 세계와 상호작용하는 능력**을 부여하는 것이 Vertex AI Agent Builder의 핵심 기능 중 하나입니다. ADK를 통해 에이전트는 **도구**라는 형태로 여러 가지 기능을 사용할 수 있는데, 예를 들어 **Google 검색**, **Vertex AI Search**(벡터 검색을 포함한 검색 서비스), **코드 실행 환경** 등의 **기본 내장 도구**가 제공됩니다cloud.google.com. 또한 **Model Context

Protocol(MCP)**을 지원하여 LangChain에서 제공하는 툴이나 기타 오픈소스 도구(예: Crew.ai 도구, 데이터베이스용 생성형 AI 툴킷 등)와 연동할 수 있고 cloud.google.com, 다른 에이전트를 도구처럼 사용할 수도 있습니다 developers.googleblog.com. 기업 시스템 통합을 위해, 100개+의 **사전 구축 커넥터**(예: Slack, Jira, SAP 등 연동), Apigee에 등록된 **맞춤 API**, 또는 Application Integration을 통한 **워크플로 연계** 등도 지원되어 에이전트가 기업 내부 애플리케이션과 데이터에 접근할 수 있습니다cloud.google.com. 이러한 도구 연결 기능을 활용하면, 에이전트는 **필요시 외부 정보에 접근하거나 사내 시스템에 작업을 지시**할 수 있어 응답의 정확성과 유용성이 크게 향상됩니다. 동시에 각 도구 사용에는 **비즈니스 규칙과 가드레일**을 적용할 수 있어, 에이전트가 기업 정책을 준수하면서 안전하게 동작하도록 관리할 수 있습니다cloud.google.com.

- **Vertex AI Agent Engine**: **Agent Engine**은 완전 관리형 **에이전트 실행 환경**으로, 개발한 에이전트를 손쉽게 배포하고 운영할 수 있게 해줍니다cloud.google.com. Agent Engine을 사용하면 인프라 설정이나 서버 관리에 신경 쓸 필요 없이, Google Cloud 상에서 자동으로 **확장성**, **보안**, **모니터링**이 갖춰진 상태로 에이전트가 구동됩니다cloud.google.com. 이는 많은 기업이 AI 프로젝트를 개발까지는 잘 진행하지만 **배포 단계에서 겪는 어려움**(예: 인프라 복잡성, 확장 이슈)을 크게 줄여줍니다softude.com. 선택한 ML 모델이나 에이전트 프레임워크와 관계없이 Agent Engine이 표준화된 방식으로 에이전트를 구동해주므로, 다양한 구현 방식의 에이전트도 **일관되게 배포 및 관리**할 수 있습니다cloud.google.com. 특히 Agent Engine은 **대화의 컨텍스트(문맥)를 유지**해 주는 기능이 있어, 사용자와 에이전트의 이전 대화 내용을 기억하고 이어가는 자연스러운 상호작용이 가능합니다cloud.google.com. 또한 내장된 **단기 메모리와 장기 메모리** 기능으로 세션별 기억뿐 아니라 오랜 기간의 사용자 선호나 이력도 축적하여 활용할 수 있어, 대화가 진행될수록 더욱 개인화되고 일관성 있는 응답을 제공합니다 cloud.google.com.

- **RAG (Retrieval-Augmented Generation) 통합**: Vertex AI Agent Builder는 **검색 증강 생성(RAG)** 기능을 통해 에이전트가 방대한 기업 지식이나 데이터에 기반하여 **추론 및 답변**을 할 수 있게 합니다cloud.google.com. Vertex AI Search를 즉시 활용할 수 있는 통합 옵션을 제공하여, 클릭 몇 번만으로 벡터 검색과 키워드 검색이 결합된 **하이브리드 검색** 솔루션을 에이전트에 붙일 수 있습니다 cloud.google.com. 이를 사용하면 구조화된 데이터베이스부터 비정형 문서까지 **다양한 데이터 소스**에서 관련 정보를 찾아와 답변에 활용하도록 에이전트를 구성할 수 있습니다softude.com. 예를 들어, 사내 지식관리시스템, 고객 지원 티켓, 또는 인터넷상의 **권위 있는 정보원** 등을 에이전트가 검색하여 답변에 인용하도록

할 수 있습니다cloud.google.com. Google의 방대한 웹 검색 인덱스(전 세계 검색 데이터의 99%를 커버하는 Google 검색)와 Dun & Bradstreet 등의 전문 데이터 제공업체 정보까지 연계하여, 에이전트 답변의 **사실 정확도와 신뢰도**를 높일 수 있습니다cloud.google.com. 또한 Slack, Jira, Cloud Storage, 구글 드라이브 같은 협업 도구나 파일 저장소에 연결해서 **맞춤형 RAG 엔진**을 구축할 수도 있어, 조직 내부의 사유 데이터도 활용한 고품질 답변 생성을 지원합니다 cloud.google.com.

- **Agentspace를 통한 에이전트 공유**: **Agentspace**는 기업 내에서 만든 다양한 AI 에이전트를 **조직 전체에 배포하고 공유**하기 위한 **엔터프라이즈 마켓플레이스**입니다cloud.google.com. 에이전트 빌더로 제작한 사내 AI 에이전트를 Agentspace에 게시하면, 중앙 관리 하에 해당 에이전트를 필요한 직원들이 찾아 활용할 수 있습니다cloud.google.com. 예를 들어, 영업팀을 위한 고객 질의응답 에이전트나, 개발팀을 위한 코드 조언 에이전트를 Agentspace에 올려두면 조직의 모든 구성원이 **하나의 포털**에서 자신에게 필요한 AI 도구를 쉽게 발견하고 사용할 수 있습니다 cloud.google.com. Agentspace는 **접근 권한 제어**와 **활동 모니터링** 등의 거버넌스 기능을 제공하므로, 기업은 AI 에이전트 활용을 넓히면서도 **보안과 일관성**을 유지할 수 있습니다cloud.google.com. 몇 번의 클릭으로 검증된 에이전트를 여러 팀에 배포함으로써 AI 도입 효과를 극대화하고, 조직 전반의 **AI 활용도를 증진**시키는 것이 가능해집니다.

- **엔터프라이즈급 보안 및 거버넌스**: Vertex AI Agent Builder는 기업 환경에 적합한 **강력한 보안 통제** 기능을 갖추고 있습니다. 우선, Google의 최신 LLM인 Gemini 모델이 제공하는 **내장 안전성 기능**(컨텐츠 필터링, 금지 토픽에 대한 사전 가이드라인 등)을 활용하여 에이전트의 응답 내용에 대한 **콘텐츠 수준의 제어**를 할 수 있습니다cloud.google.com. 이를 통해 에이전트가 회사 정책에 어긋나는 발언을 하거나 민감한 정보를 누설하지 않도록 **시스템 프롬프트와 필터**를 설정할 수 있습니다cloud.google.com. 또한 **ID 및 권한 제어**를 지원하여, 에이전트가 특정 **서비스 계정**의 권한으로 동작할지 혹은 각 사용자의 권한을 대행하여 동작할지를 결정하고 관리할 수 있습니다cloud.google.com. 나아가 **보안 경계(Security Boundary)** 설정을 통해 에이전트의 활동 범위를 제한함으로써, 지정된 범위 밖의 민감 정보나 시스템에 접근하지 못하게 차단할 수 있습니다cloud.google.com. 개발자는 에이전트의 입력 단계에서 민감한 내용을 걸러내거나 툴 실행 전에 매개변수를 검증하는 등, **상호작용의 모든 단계에 가드레일**을 설치하여 안전성을 확보할 수 있습니다cloud.google.com. 마지막으로, Vertex AI는 **종합적인 추적 및 모니터링** 기능을 제공하여 에이전트의 모든 행동(예: LLM의 추론 과정, 사용한 도구와 API, 의사결정 경로)을 자동으로 로깅하고 시각화합니다cloud.google.com. 이를

통해 운영자는 에이전트가 어떻게 판단하고 행동했는지 **투명하게 파악**하여 문제가 발생할 경우 원인을 조사하거나 성능을 개선하는 데 활용할 수 있습니다.

以上의 기능들 덕분에 Vertex AI Agent Builder는 **개발자에게는 강력하고 유연한 도구**를, **기업에는 신뢰성 있고 확장 가능한 AI 에이전트 솔루션**을 제공한다는 점에서 큰 장점을 갖습니다. 특히 **속도와 제어**라는 두 마리 토끼를 잡았다는 평가를 받는데softude.com, 복잡한 AI 시스템을 구현하면서도 빠른 프로토타이핑과 손쉬운 배포가 가능하고, 개방형 표준을 채택함으로써 향후 기술 변화나 다른 플랫폼과의 연계에도 유연하게 대처할 수 있는 것이 강점입니다.

3. 작동 방식 및 내부 구조

위 그림은 Vertex AI Agent Builder의 **전체 아키텍처**를 개략적으로 보여줍니다. 좌측에는 엔터프라이즈 데이터 및 다양한 도구들과의 연결이 배치되어 있는데, 이는 에이전트가 기업 내외부의 정보를 활용하고 회사의 정책과 규칙 내에서 행동할 수 있도록 **컨텍스트와 도구**를 제공하는 부분입니다. 중앙에는 **Agent Garden, ADK, Agent Engine** 등의 핵심 구성 요소가 위치하여, 에이전트의 오케스트레이션(흐름 관리), 툴 사용, 메모리 관리, 대화 세션 관리 등을 담당합니다. 우측에는 **대형 언어 모델(LLM)**들이 자리하고 있는데, Google의 Gemini와 Vertex AI Model Garden의 다양한 사전 학습 모델들뿐만 아니라 오픈소스 LLM 및 외부 에이전트 프레임워크까지 함께 작동하는 **개방형 모델 생태계**가 표현되어 있습니다. 이러한 구조를 통해 Vertex AI Agent Builder는 기업 데이터와 도구, 그리고 AI 모델을 하나의 통합된 시스템 안에서 유기적으로 결합하여 지능형 에이전트를 구현할 수 있게 합니다.

이제 이러한 내부 구조와 작동 원리를 조금 더 상세히 살펴보겠습니다. **Vertex AI Agent Builder**에서 에이전트의 동작은 기본적으로 대형 언어 모델(LLM)을 중심으로 이루어집니다. 개발자가 ADK를 사용하여 정의한 에이전트 로직(예: 에이전트의 역할 설명, 사용할 수 있는 툴 목록, 응답 방식에 대한 지침 등)이 있으며, 사용자가 질문이나 명령을 주면 에이전트는 **LLM을 통해 그 의도를 파악하고 최적의 응답이나 행동을 결정**합니다 developers.googleblog.comdevelopers.googleblog.com. 예를 들어, 개발자가 LlmAgent 객체를 생성하면서 "질문에 답하기 위해 Google 검색을 사용하라"는 지침과 함께 google_search라는 툴을 등록해두면, 에이전트는 사용자의 질의에 직접 답변할지 혹은 제공된 검색 툴을 호출하여 정보를 찾은 후 답변할지를 LLM의 추론으로 판단하게 됩니다developers.googleblog.com. 이러한 방식으로 에이전트는 **자율적으로 사고(think)**하고 필요한 경우 **도구를 액션으로 실행(act)**한 후 그 결과를 토대로 **최종 응답을 생성**하게 됩니다.

Vertex AI Agent Builder의 **내부 오케스트레이션**은 두 가지 형태를 모두 지원합니다. 하나

는 **워크플로 에이전트**를 통한 **결정론적(Deterministic) 시퀀스**입니다. 개발자가 특정 작업 절차를 Sequential 혹은 Parallel, Loop 형태의 워크플로로 정의하면 에이전트는 정해진 순서와 논리에 따라 단계를 수행합니다developers.googleblog.com. 다른 하나는 LLM의 힘을 활용한 **동적 라우팅(Dynamic Routing)**입니다. 즉, 사전에 정해놓은 흐름 없이 에이전트가 **LLM 기반의 추론**으로 현재 상황에서 어떤 하위 에이전트에게 일을 맡길지(LlmAgent 간의 이양) 또는 어떤 도구를 쓸지 등을 **실시간으로 판단**하는 것입니다 developers.googleblog.com. 이 동적 오케스트레이션 덕분에 에이전트 시스템은 예측 불가능한 사용자 요구에도 유연하게 대응할 수 있으며, 복잡한 분기 로직 없이도 상황에 맞는 행동을 실행합니다. 개발자는 필요에 따라 **엄격한 흐름 통제와 유연한 AI 추론** 중 적절한 방식을 선택하거나 결합하여 사용할 수 있습니다.

Vertex AI Agent Builder 내부에는 **멀티 에이전트 협업**을 위한 구조도 마련되어 있습니다. 단일 에이전트가 모든 일을 처리하는 대신, 각각 전문성을 갖춘 여러 하위 에이전트를 두고 **팀으로 협력**하도록 설계할 수 있습니다softude.com. 이를 위해 ADK에서는 에이전트를 계층적으로 구성하고, **요청을 자동 라우팅(auto flow)**하여 가장 적합한 에이전트가 해당 작업을 처리하도록 할 수 있습니다google.github.io. 예를 들어, 사용자 대화에서 인사나 작별 인사와 같은 부분은 **인사 담당 에이전트**가 처리하고, 날씨 문의와 같은 부분은 **날씨 정보 에이전트**가 처리하는 식으로 **역할 분담**을 시킬 수 있습니다 google.github.iogoogle.github.io. 상위 에이전트는 마치 매니저처럼 동작하여 사용자 입력을 해석한 뒤, 내부적으로 적절한 하위 에이전트에게 질문을 **위임**하고 그 응답을 종합하여 사용자에게 전달합니다google.github.io. 이러한 멀티 에이전트 팀 구조에서는 A2A 프로토콜이 커뮤니케이션을 담당하여, 서로 다른 에이전트들이 원활하게 정보 공유와 작업 협의를 수행합니다softude.com.

메모리와 컨텍스트 관리는 Agent Builder 작동 방식의 또 다른 중요한 측면입니다. 앞서 언급했듯 Agent Engine은 대화 중에 **세션 상태(Session State)**를 유지하여, 에이전트가 이전 대화 턴에서 주고받은 정보를 기억하게 합니다google.github.iogoogle.github.io. 예를 들어 사용자가 "아까 이야기했던 프로젝트의 마감일이 언제죠?"라고 물으면, 에이전트는 이전에 사용자가 언급했던 프로젝트를 기억하고 해당 마감일을 답변할 수 있습니다. 이는 단기 메모리에 해당하며, 각 대화 세션 내에서의 문맥을 유지해주어 **일관된 대화 흐름**을 가능케 합니다cloud.google.com. 더 나아가 Agent Engine은 **장기 메모리**도 지원하여, 사용자가 여러 세션에 걸쳐 상호작용하더라도 과거의 중요한 정보나 사용자의 선호를 축적했다가 활용할 수 있습니다cloud.google.com. 예를 들어 이전 대화에서 사용자 취향을 학습한 에이전트가 다음 번 대화에서도 그 취향을 기억하고 맞춤형 답변이나 추천을 제공하는 식입니다. 이러한 메모리 관리 기능은 **개인화된 사용자 경험**을 제공하고 대화의 품질을 높이는 핵심 요소입니다.

마지막으로, **추적과 평가** 메커니즘이 내부에 встро어 있어 에이전트의 작동 방식을 투명하게 모니터링할 수 있습니다. Agent Engine은 에이전트의 매 단계 행동(예: 어떤 프롬프트를 LLM에 보냈는지, LLM이 어떤 **사고의 흔적**을 남겼는지, 어떤 **도구 호출**을 실행했는지 등)을 모두 **로그로 기록하고 시각화**해줍니다cloud.google.com. 개발자는 Vertex AI의 **Web UI 대시보드**나 CLI를 통해 이 실행 추적 정보를 확인하면서 디버깅하거나 성능을 개선할 수 있습니다developers.googleblog.comdevelopers.googleblog.com. 또한 **Example Store**와 **평가 도구**를 활용하여 미리 정의한 테스트 시나리오에 대해 에이전트의 응답 품질과 과정이 기대대로 이루어지는지 **체계적으로 검증**할 수 있습니다 cloud.google.com. 이러한 평가 기능은 에이전트의 약점을 발견하고 실제 배포 전에 개선하는 데 도움을 주며, 배포 후에도 지속적인 **학습과 개선 사이클**을 운영할 수 있도록 합니다.

요약하면, Vertex AI Agent Builder의 내부 구조는 **LLM의 강력한 자연어 이해/생성 능력**, **다양한 도구와 데이터 소스 연동**, **멀티 에이전트 협업**, **메모리 기반의 문맥 유지**, 그리고 **안전장치와 모니터링**이 유기적으로 결합된 형태입니다. 이러한 구성으로 에이전트는 사람과 유사한 **유연성과 지능**을 가지면서도, 기업 환경에서 요구되는 **통제와 신뢰성**을 갖춘 형태로 작동합니다.

4. 구성 요소 설명 (대화 흐름, 인텐트, 액션, API 호출 등)

Vertex AI Agent Builder로 에이전트를 설계할 때 고려해야 하는 **핵심 구성 요소**를, 일반적인 대화형 AI 에이전트의 개념에 비추어 설명하겠습니다. 이는 전통적인 챗봇 설계 개념(예: **대화 흐름**, **의도(intent)**, **액션(action)**, **API 호출**)과 현대적인 LLM 기반 에이전트의 차이를 이해하는 데도 도움이 됩니다.

- **대화 흐름**: 대화 흐름이란 사용자와 에이전트 간의 **상호작용 진행 방식**을 의미합니다. 기존 규칙 기반 챗봇에서는 미리 정의된 시나리오에 따라 대화 흐름이 고정적으로 진행되는 반면, Vertex AI Agent Builder의 에이전트는 기본적으로 **자유로운 자연어 대화**를 지원합니다. LLM이 사용자의 입력을 해석하고 적절한 답변을 생성하기 때문에, 반드시 정해진 트리를 타지 않더라도 문맥에 맞게 **유동적인 대화**가 가능합니다. 예를 들어, 사용자가 질문을 하다가 추가 정보를 요구하면 에이전트는 이전 질문의 맥락을 이해한 채로 이에 대응할 수 있습니다. 이는 앞서 설명한 **세션 메모리**에 의해 가능하며, Agent Engine이 대화 컨텍스트를 계속 유지해 주므로 이루어집니다cloud.google.com. 물론 필요에 따라 개발자가 **대화 흐름을 가이드**할 수도 있습니다. ADK에서는 콜백 함수나 워크플로 정의를 통해 특정 입력에 특정 하위 작업을 수행하게 하거나, 단계별로 질문을 이어가는 흐름을 코딩할 수 있어 **혼합형 접근**도 가능합니다. 요컨대 Agent Builder에서는 **LLM의 자연스러운 대화 능력**을 기반으로 하면서, 원하면 개발자가 흐름을 통제하거나 지침을

부여하여 **안내된 대화**를 구현할 수 있습니다.

- **인텐트(Intent)**: 인텐트는 사용자의 질문이나 요청이 **무엇을 의도하는지**를 나타내는 개념입니다. 과거의 챗봇 플랫폼(예: Dialogflow 등)에서는 사용자의 발화를 사전에 정해둔 인텐트로 분류하고 그에 대응하는 답변이나 액션을 매핑하는 방식이 일반적이었습니다. Vertex AI Agent Builder에서는 **명시적인 인텐트 분류 단계가 없어도** LLM이 문맥 속에서 사용자의 의도를 파악하고 대응할 수 있습니다. 예를 들어 사용자가 "내 주문 배송 언제 도착하나요?"라고 물으면, 에이전트는 이를 따로 '배송 조회 인텐트'로 태깅하지 않더라도 질문만으로 맥락을 이해하고 답변을 생성합니다. 이는 대규모 언어 모델의 **자연어 이해 능력** 덕분입니다. 그러나 개발자 관점에서 여러 종류의 요청을 다루고자 할 때는, 여전히 **인텐트 개념을 활용한 설계**가 유용할 수 있습니다. Agent Builder에서는 인텐트를 하드코딩하지 않는 대신, **여러 전문 에이전트로 역할을 분리**하여 인텐트에 대응하는 방식을 취합니다 google.github.iogoogle.github.io. 예를 들어 질의응답, 상품 추천, 주문 조회처럼 카테고리가 다른 요청들을 하나의 거대 모델로 모두 처리하는 대신, 각각에 특화된 하위 에이전트를 만들고 **Auto Flow**에 따라 해당 입력을 적절한 에이전트에게 분배하도록 합니다google.github.io. 이렇게 하면 각 에이전트는 자신의 **역할 범위(사실상의 인텐트)에 집중**하여 더 정확한 답변이나 작업 수행을 할 수 있고, 시스템 전체로 보면 여러 인텐트를 포괄하는 **멀티에이전트** 솔루션이 됩니다. 요약하면 Agent Builder에서는 전통적인 인텐트-액션 매핑보다는 **LLM 기반의 자유로운 이해 + 필요시 다중 에이전트 분업으로 인텐트 대응** 방식을 채택합니다.

- **액션(Action)과 도구 사용**: 액션은 에이전트가 사용자 요청을 처리하기 위해 취하는 **구체적인 행동**입니다. 예를 들어 정보를 검색한다든지, 다른 서비스에 명령을 내린다든지, 데이터베이스를 조회/갱신한다든지 하는 모든 작업이 액션에 해당합니다. Vertex AI Agent Builder에서 액션들은 일반적으로 **"도구(Tool)"**의 형태로 구현됩니다google.github.io. 개발자는 Python 함수나 API 호출 등을 **툴로 정의**하고 이를 ADK의 에이전트에 등록해줄 수 있습니다google.github.io. 그러면 에이전트는 대화 도중 해당 툴이 필요하다고 판단될 때 그 함수를 실행하여 결과를 얻고, 이를 기반으로 응답을 구성합니다. 예를 들어, 사용자의 질문에 최신 뉴스가 필요하면 에이전트는 **Google 검색 툴**을 액션으로 실행해 뉴스를 찾아보고 답변에 반영할 수 있습니다cloud.google.com. 액션은 단순 정보 조회뿐만 아니라 **기업 시스템에서의 트랜잭션**일 수도 있습니다. 예를 들어 고객 지원 에이전트가 환불 요청을 받으면 사전에 연결된 **주문 관리 API(툴)**를 호출하여 환불 처리를 실행하는 식입니다. Vertex AI Agent Builder는 이러한 액션 수행 결과를 LLM이 이해할 수 있는 형태로 자동 변환하거나 요약하여, **대화 흐름 안에 자연스럽게 녹여** 줍니다. 요컨대, Agent Builder의 에이전트는 필요할 때 **프로그래밍적인 행위를 하**

고(툴 실행) 그 결과를 **대화에 반영**함으로써 단순히 말로 답하는 것을 넘어 **실제 작업 수행형 AI**로 기능합니다.

- **API 호출 및 통합**: 에이전트가 외부 시스템과 연동하는 가장 전형적인 방법이 **API 호출**입니다. Agent Builder는 Google Cloud의 다양한 API 서비스와 **Apigee API 관리 허브**와 긴밀히 연동되어, 기업이 이미 보유한 **REST API, GraphQL API** 등을 에이전트에게 바로 활용하게 할 수 있습니다 cloud.google.comcloud.google.com. 예를 들어 재고 조회 API, 사내 인사 시스템 API, CRM 시스템 API 등을 에이전트가 호출하도록 함으로써, 사용자와 대화 중에 **백엔드 업무를 직접 처리**하는 것이 가능합니다. 앞서 언급한 100여 개의 커넥터와 Apigee, Application Integration 연계 기능이 이러한 API 통합을 돕습니다 cloud.google.com. 개발자는 복잡한 인증이나 호출 로직을 일일이 구현하지 않고도 **템플릿화된 통합 솔루션**을 이용해 간단히 에이전트에 API 기능을 부여할 수 있습니다cloud.google.com. Agent Builder의 에이전트는 이러한 API 호출 시 **보안 주체**를 명확히 하여(예: 에이전트 자신의 서비스 계정으로 호출하거나, 현재 사용자의 권한으로 프록시 호출) **보안 및 권한 관리**도 투명하게 이루어집니다 cloud.google.com. 결과적으로, API 통합을 통해 에이전트는 대화만이 아니라 **업무 트랜잭션**까지 수행하는 **실용적인 비즈니스 도구**로 거듭나게 됩니다.

以上을 종합하면, Vertex AI Agent Builder의 구성 요소들은 **대화형 AI**를 구성하는 모든 측면을 포괄합니다. LLM 덕분에 **유연한 대화 흐름**과 **의도 파악**이 가능하고, ADK와 도구 통합을 통해 **다양한 액션과 API 호출**을 수행할 수 있으며, Agent Engine이 이 모든 과정을 **관리·조율**합니다. 개발자는 과거 챗봇 설계에서 일일이 규칙을 정하고 인텐트를 분류하던 작업 대신, **필요한 도구와 제약만 설정하면** 에이전트가 스스로 최적 경로를 찾아가도록 만들 수 있습니다. 이는 AI 에이전트 개발에 있어 생산성을 높여주고, 동시에 에이전트의 **역할과 책임을 명확히 구성**할 수 있게 해줍니다.

5. 활용 사례

Vertex AI Agent Builder는 강력하고 유연한 만큼, 다양한 분야에서 활용될 수 있습니다. 대표적인 **활용 사례**를 몇 가지 들어보면 다음과 같습니다:

- **고객 지원(Chatbot 및 콜센터 자동화)**: 고객 문의에 답변하거나 문제를 해결해주는 **AI 상담원** 에이전트를 구축할 수 있습니다. 예를 들어 은행에서는 계좌 잔액 조회나 분실 신고를 처리하는 챗봇, 온라인 쇼핑몰에서는 주문 상태를 안내하고 환불을 도와주는 챗봇을 생각할 수 있습니다. Vertex AI Agent Builder 기반의 고객 지원 에이전트는 단순히 FAQ 답변을 하는 수준을 넘어, **실시간으로 고객 데이터를 조회**하고 **필요한 조치를 수행**할 수 있다는 장점이 있습니다. 고객의 질문 의

도를 LLM이 이해하고, 필요한 경우 데이터베이스 조회 API나 주문 처리 시스템에 액션을 걸어 실제 문제를 해결해 줍니다. 또한 대화 중에 이전에 나눈 내용을 기억하여 일관된 서비스를 제공하고, 민감한 질문에 대해서는 사전에 정의된 정책에 따라 대응하거나 사람 상담원에게 이관하는 등 **컨텍스트 기반의 똑똑한 상담**이 가능합니다. 이를 통해 **대기 시간 감소**, **24/7 서비스 제공**, **운영 비용 절감** 등의 효과를 거둘 수 있으며, 고객 만족도도 높아질 것으로 기대됩니다.

- **지식 검색 및 기업 내 정보 제공**: 대기업이나 연구 기관에서는 방대한 내부 문서와 데이터가 축적되어 있는데, Agent Builder를 활용하면 **사내 지식 검색 에이전트**를 만들 수 있습니다. 예컨대 직원들이 사내 정책, IT 지원, 인사 정보, 혹은 기술 문서를 찾기 위해 일일이 문서를 열람하는 대신, 자연어로 질문하면 에이전트가 관련 정보를 찾아 요약하여 답변해주는 시스템입니다. Vertex AI의 RAG 기능과 Search 통합을 통해 이 에이전트는 **문서를 읽고 추론**하여 정확한 답을 제공합니다cloud.google.com. "지난 분기 매출 보고서에서 아시아 지역 실적은 어떻습니까?" 같은 질문을 받으면, 에이전트는 관련 보고서를 검색하고 매출 수치를 추출해 요약 답변을 해줄 수 있습니다. 또한 외부 웹과 연결하면 일반적인 지식 질문이나 시사 정보도 답변할 수 있어, **사내 정보와 공개 정보**를 아우르는 **지능형 비서** 역할을 할 수 있습니다. 이러한 에이전트는 임직원의 **업무 생산성**을 높이고, 필요한 정보를 **신속하게 얻도록** 도와줌으로써 업무 효율을 향상시킵니다.

- **업무 프로세스 자동화**: Vertex AI Agent Builder의 **멀티 에이전트 워크플로** 기능은 여러 단계에 걸친 복잡한 업무를 자동화하는 데 특히 유용합니다. 예를 들어, 한 기업에서 신규 직원 온보딩 과정을 생각해보겠습니다. 전통적으로는 인사팀이 계정 생성, 장비 지급, 교육 일정 조율 등 여러 작업을 수동 혹은 개별 시스템으로 처리해야 했습니다. Agent Builder를 활용하면, **온보딩 에이전트 팀**을 구성하여 이러한 작업을 **대화 기반으로 자동화**할 수 있습니다. 신규 입사자가 에이전트와 채팅을 통해 필요한 정보를 입력하면, 에이전트 팀은 내부적으로 역할을 분담해 한 에이전트는 계정 생성 API를 호출하고, 다른 에이전트는 노트북 발송을 창고 시스템에 지시하고, 또 다른 에이전트는 교육 일정에 등록하는 식으로 **병렬적**으로 업무를 처리합니다. 사용자는 에이전트와의 간단한 대화만으로 모든 절차가 진행되고 있음을 확인할 수 있고, 완료되면 에이전트가 요약 결과를 알려주는 식입니다. 이처럼 **사무 업무**, **IT 지원 시나리오**, **다단계 승인 절차** 등에서도 Agent Builder 기반 에이전트가 도입되어 **반복적인 작업을 자동화**하고 **오류를 감소**시킬 수 있습니다softude.com. 특히 여러 시스템 간의 중계나 복합적인 의사결정이 필요한 작업에서 멀티 에이전트의 협업 능력이 빛을 발휘하여, 사람이 일일이 조율하지 않아도 **통합된 자동화**가 가능해집니다.

- **기타 활용 분야**: 이외에도 **개인 비서/생산성 도구**(일정 관리, 이메일 응답 초안 작성, 회의 요약 등 개인 업무 지원), **데이터 분석 보조**(데이터베이스 질의와 결과 해석을 대화형으로 수행), **교육 및 트레이닝**(학습자 질문에 답하거나 맞춤형 학습 경로를 안내), **크리에이티브 콘텐츠 생성**(마케팅 카피 작성, 코딩 보조 등) 등 **다양한 도메인에서 AI 에이전트**를 구축할 수 있습니다. Vertex AI Agent Builder의 강점은 기존 프로세스에 쉽게 통합할 수 있다는 점이므로, 현재 사람이 수행하는 업무 중 규칙적이거나 AI가 도울 수 있는 부분에 에이전트를 적용하여 **업무 혁신**을 이룰 수 있습니다. 기업은 작은 **파일럿 프로젝트**로 시작하여 점차 활용 범위를 넓힘으로써, AI 에이전트가 조직 전반에 걸쳐 **생산성과 창의성**을 높이는 방향으로 활용할 수 있을 것입니다.

6. 일반 사용자 및 개발자 관점에서의 특징

일반 사용자 관점에서 볼 때, Vertex AI Agent Builder로 구축된 에이전트들은 기존 챗봇보다 한층 **뛰어난 사용자 경험**을 제공합니다. 우선 대화가 자연스럽고 맥락에 맞게 이루어지므로, 사용자는 마치 **사람과 대화**하듯이 편안하게 질문하고 요청할 수 있습니다. 에이전트가 음성이나 영상까지 지원하는 경우 목소리나 얼굴을 통해 **인간적인 인터페이스**를 제공할 수도 있습니다cloud.google.com. 또한 사용자는 자신이 필요로 하는 정보를 얻거나 작업을 처리함에 있어 **여러 부서나 시스템을 일일이 거치지 않고**, 하나의 AI 에이전트를 통해 **원스톱 서비스**를 받게 됩니다. 예컨대, 여행 계획을 세우기 위해 항공권 검색, 호텔 예약, 일정 조율을 각각 다른 사이트에서 할 필요 없이 에이전트 한 명과의 대화로 모두 해결할 수 있습니다codelabs.developers.google.comsoftude.com. 에이전트가 기업의 다양한 시스템과 연결되어 있기 때문에 가능한 일입니다. 이때 사용자에게는 복잡한 절차가 보이지 않고, **질문->응답**의 형태로 간단하게 상호작용할 뿐입니다. 더불어 이러한 에이전트는 **실시간으로 지식과 데이터를 업데이트**받기 때문에, 항상 최신 정보에 기반한 신뢰도 높은 답변을 제공합니다cloud.google.com. 예를 들어 재고가 변동되거나 새로운 정책이 생겨도 에이전트가 연결된 백엔드의 데이터를 바로 반영하므로, 사용자는 늘 정확한 정보를 얻을 수 있습니다. 마지막으로, **안전성과 프라이버시** 측면에서도 사용자는 보호를 받습니다. 기업이 에이전트에 설정한 콘텐츠 필터와 보안 경계 덕분에, 에이전트는 민감하거나 부적절한 내용은 걸러주고 사용자 권한에 따라 허용된 작업만 수행하므로, **신뢰하고 사용할 수 있는 AI 비서**로 다가옵니다cloud.google.com.

개발자 관점에서 Vertex AI Agent Builder는 **생산성과 유연성, 그리고 통합 관리** 측면에서 큰 이점을 제공합니다. 먼저, ADK를 통한 개발은 비교적 **난이도가 낮고 신속**합니다. 파이썬 언어 기반으로 친숙한 문법과 SDK를 제공하며, 100여 줄의 코드로도 작동하는 에이전트를 만들 수 있을 만큼 간결하게 설계되었습니다cloud.google.com. 이는 개발자가 반드시 머신러닝 박사일 필요 없이도(혹은 전문 AI 엔지니어가 부족한 조직에서도) **짧은**

학습곡선으로 유용한 에이전트를 만들 수 있게 함을 의미합니다softude.com. 둘째, Agent Builder는 **기존 시스템과의 통합이 용이**합니다. 오픈소스 A2A 프로토콜과 MCP 지원 덕분에, 이미 운용 중인 타사 에이전트나 데이터 파이프라인, 그리고 사내 다양한 API들과도 **호환성을 유지**하면서 연계할 수 있습니다cloud.google.comcloud.google.com. 개발자는 새로운 플랫폼 도입 시 흔히 우려하는 **종속(lock-in)** 문제없이, 필요에 따라 Vertex AI의 일부 기능만 취사선택하여 활용하거나 외부 기술과 조합할 수 있습니다. 셋째, **운영 관리가 수월**합니다. Agent Engine이 배포 후 **인프라를 자동 관리**해주므로, 개발자는 모델 튜닝이나 기능 개선 등 **핵심 로직에 집중**할 수 있고 배포/스케일링/모니터링은 플랫폼에 맡길 수 있습니다softude.comcloud.google.com. 특히 멀티 에이전트 시스템을 일일이 모니터링하고 로그를 추적하는 것은 어려운 일인데, Agent Builder는 이를 한 곳에서 **통합 모니터링 및 디버깅**할 수 있는 툴을 제공하므로 개발 및 운영(DevOps)에 드는 부담을 줄여줍니다developers.googleblog.comcloud.google.com. 넷째, **보안과 거버넌스 설정**을 코드 레벨에서 세밀하게 조정할 수 있어, 개발자는 기업의 보안팀이나 규제 요건에 부합하는 **안심할 수 있는 애플리케이션**을 구축할 수 있습니다cloud.google.com. 마지막으로, Vertex AI 플랫폼의 다른 구성 요소들과 원활히 연동되므로(예: Vertex AI Search, Example Store, 모델 모니터링 등), **완결형 AI 솔루션 개발 생태계** 안에서 모든 작업을 수행할 수 있다는 편의성도 중요합니다. 요약하면 Vertex AI Agent Builder는 개발자에게 **빠른 개발, 다양한 연동, 쉬운 배포와 관리, 그리고 강력한 통제권**을 함께 제공하는 도구이므로, **현대적인 엔터프라이즈 AI 애플리케이션 개발을 위한 종합 플랫폼**으로 자리매김하고 있습니다.

참고 자료: Vertex AI Agent Builder 공식 문서 및 제품 페이지 cloud.google.comcloud.google.com, Google Cloud 블로그developers.googleblog.com, Softude 기술 블로그softude.comsoftude.com 등.