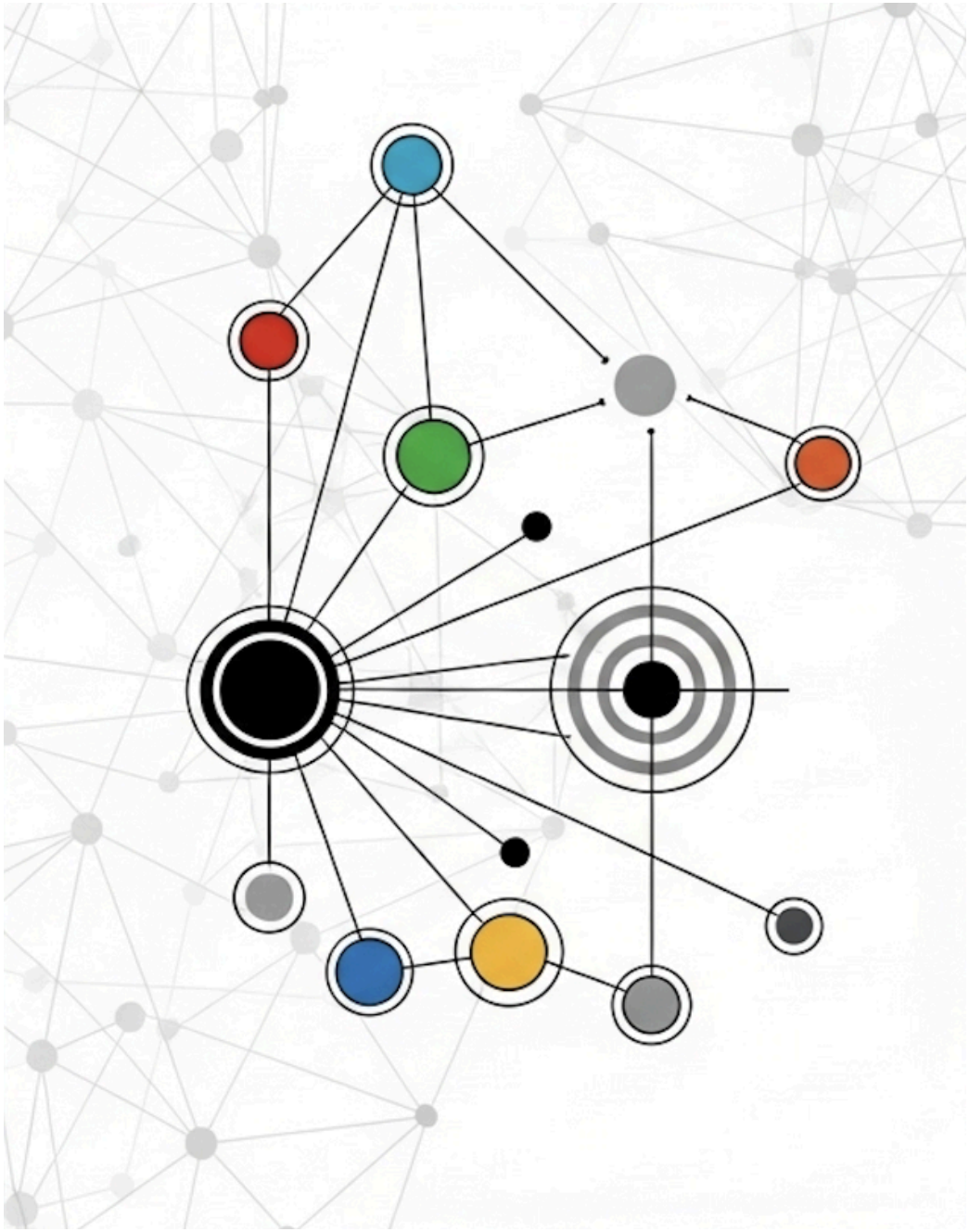
The New Agentic Landscape

Google Cloud’s Agentic AI Transformation Framework



Platform and AgentOps Framework - References

Vision and Motivation

Our Vision: The Autonomous Enterprise

Imagine an organization where your most critical and complex business processes begin to run themselves. Picture a collaborative digital workforce of specialized AI agents—a procurement agent negotiating with suppliers, a financial crime agent preventing fraud before it happens, a supply chain agent rerouting shipments to avoid disruption—all working in concert with your human experts.

This is the future of business operations. It's a future where AI doesn't just answer questions or generate text, but takes action. In this new operating model, intelligent agents handle the complex, data-intensive, and repetitive work, freeing your people to focus on what they do best: strategy, creativity, and building customer relationships.

The vision is not about replacing human talent, but amplifying it. It's about building a more intelligent, resilient, and responsive organization that can anticipate customer needs, mitigate risks proactively, and seize market opportunities in real time. This is the promise of Agentic AI: to transform not just individual tasks, but the fundamental way your business creates value.

The Motivation: From Experiment to Enterprise Scale

The age of Agentic AI is no longer a distant future; the foundational technology is here today. Many organizations are already experimenting, building impressive pilots that demonstrate what's possible. However, the leap from a single, exciting demo to a scalable, secure, and enterprise-wide capability is significant.

Without a deliberate plan, these initial experiments often become disconnected silos. They result in inconsistent governance, mounting technical debt, and a portfolio of powerful but unmanaged tools that fail to deliver their full potential. The initial excitement gives way to the challenges of complexity, security, and cost.

This Agentic AI Transformation Framework is Google Cloud's answer to that challenge. It provides the blueprint to navigate this journey responsibly and effectively. It is a structured, practical guide—built on Google's deep expertise in AI and real-world enterprise deployments—to move from isolated successes to a cohesive ecosystem of intelligent agents. This framework is your roadmap to manage complexity, mitigate risk, and ensure that every agent you build contributes to measurable business outcomes. It's how you go from ambition to reality, starting smart and scaling strategically to build a truly autonomous enterprise.

Agentic AI Transformation Framework

To turn the vision of an autonomous enterprise into a reality, a structured approach is essential. Our Agentic AI Transformation Framework provides a comprehensive, repeatable blueprint for navigating this journey. It organizes the transformation into three interconnected layers, ensuring that strategy, execution, and technology are seamlessly integrated. At its very heart, the framework is driven by Strategy, Ecosystems, & Value. This strategic core acts as our compass, ensuring every initiative is grounded in clear business objectives and measurable ROI. We begin by defining what success looks like, how agentic systems will interact within our broader digital ecosystem, and how we will quantify the value they deliver. This alignment is the foundation for all subsequent efforts.

Therefore, for the remainder of this paper, we'll leverage the Agentic AI Transformation Framework to guide our discussion. For each section, we aim to offer specific insights, detailing objectives, corresponding metrics, essential inputs from your business and technical teams, critical activities to undertake, roles throughout your organization and the responsibilities they can undertake as well as the valuable outputs your organization can anticipate.

Let's start with the heart of it all - Strategy, Ecosystems, and Value.

Strategy, Ecosystems, and Value

Contributors: Ryan Faris , Eva Dong , Patrick Nestler Dovile Mikalauskaite Keyur Mehta Barbara Kenyon Amulya Saridey

Objectives

- **Strategic Alignment & Opportunity Framing**
  - Diagnose the core business challenge where Agentic AI offers a unique and powerful advantage.
  - Align Agentic AI potential with this core business challenge and strategic objectives.
  - Define a clear direction for the transformation, identifying high-impact domains and opportunity areas.
- **Value Proposition & Strategic Prioritization**
  - Translate diagnosed opportunities into specific, quantifiable initiatives and prioritize them.
  - Focus on defining a coherent set of actions that deliver measurable value across key business drivers, such as:
    - **Efficiency:** Drive improvements in operational efficiency (e.g., reduced wait/handle times, increased interactions per agent, faster onboarding).
    - **Cost:** Achieve significant cost savings (e.g., higher containment rates, lower cost per interaction, improved agent retention).
    - **Customer Experience:** Enhance customer satisfaction and loyalty (e.g., higher CSAT, lower abandonment rates, faster resolution).
    - **Revenue:** Generate new revenue streams and opportunities (e.g., identify cross-sell/upsell moments, lower customer acquisition costs).
- **Agentic Ecosystem Design & Execution Roadmap**
  - Design a high-level blueprint for the Agentic AI ecosystem, outlining how different agents (internal, Google-provided, 3rd party, customer) will interact with autonomy and collaboration to deliver value for prioritized initiatives.
- **Adaptive Governance & Sustained Value Realization**
  - Establish an adaptive governance structure for the Agentic AI transformation, defining roles, responsibilities, ethical guardrails, and mechanisms for continuously measuring, tracking, and realizing value.

Inputs

Business	Technical

• Board level presentations on AI vision and existing corporate strategy • Corporate / business unit strategic objectives, OKRs, and existing strategy documents • Market trends, competitive landscape analysis, and industry benchmarks • Stakeholder interviews (C-suite, business leaders, operational teams) • Earnings reports and financial performance (for value context) • Aggregated employee stats (e.g., number of employees, hourly loaded rate by role, hours spent on specific tasks) • Aggregated customer stats (e.g., number of customers involved, customer lifetime value)	• Existing technology landscape and data readiness assessments • Understanding of existing Agent frameworks, tools, etc. already in use

Key Activities

1. **Key activity:** Strategic Alignment & Opportunity Framing
    a. Motivation/goal: To anchor the entire Agentic AI effort on solving the most critical business challenge, ensuring focus and maximizing strategic impact through a clear guiding approach.
    b. Key steps:
        i. Define or align with an existing AI vision for the company.
        ii. Identify the key focus areas, business units, or processes for transformation.
        iii. Prioritize use cases and define the primary value levers to be targeted.
    c. **Tools/techniques used:** Value Chain Analysis, Facilitated Workshops (Crux identification), SWOT Analysis, Analysis Frameworks (e.g., Root Cause Analysis).
2. **Key activity:** Value Proposition & Strategic Prioritization
    a. **Motivation/goal:** To prioritize initiatives, define value propositions, and ensure resources are allocated to initiatives with the highest potential value.
    b. Key steps:
        i. Identify and define specific value propositions for each initiative.
        ii. Define value drivers and break down the cost structure.
        iii. Develop ROI financial modeling for top initiatives.
        iv. Prioritize initiatives based on strategic alignment, potential ROI, and feasibility.
        v. Secure commitment for resource allocation to top-priority initiatives.

c. **Tools/techniques used:** delta Value Canvas, Agent Prioritization Matrix, Value Measurement Framework (value driver sizing, TCO analysis, ROI analysis).

3. **Key activity:** Agentic Ecosystem Design & Execution Roadmap
    a. **Motivation/goal:** To translate strategy into a high-level plan, defining how agents will work together and how capabilities will be built incrementally to achieve strategic outcomes and facilitate learning.
    b. **Key steps:**
        i. Identify core agent capabilities and types required (internal, Google, 3rd party).
        ii. Define the interaction model and workflows between different agents.
        iii. Outline the high-level technical architecture and infrastructure.
        iv. Develop a phased execution roadmap, highlighting MVPs and incremental capability build-out.
        v. Plan for integration with existing systems and data sources.
    c. **Tools/techniques used:** Agentic Critical User Journey (CUJ) Framework, Outcomes Based Roadmap, Facilitated Workshops (Ideation).
4. **Key activity:** Adaptive Governance & Sustained Value Realization
    a. **Motivation/goal:** To ensure the transformation stays on track, delivers intended value, manages risks effectively, and adapts to new insights and evolving conditions while embedding ethical considerations.
    b. **Key steps:**
        i. Design a value tracking dashboard (e.g., in Looker) to monitor KPIs.
        ii. Design a governance framework, defining roles, decision rights, and escalation paths.
        iii. Establish a regular review cadence (e.g., Steering Committee reviews, strategic adaptation sessions).
        iv. Implement formal MVP/Pilot gate reviews to approve progression based on success criteria.
        v. Institute periodic Ethical Review Board check-ins to ensure adherence to responsible AI principles.
    c. **Tools/techniques used:** Value Measurement Framework, Governance Framework Templates.

Roles and Responsibilities

Customer Responsibilities	Google Responsibilities
• Champions the initiative, secures resources, and removes roadblocks through an Executive Sponsor.	• Guides overall strategic alignment, Crux identification, value quantification, and ensures realization through a delta Engagement Lead.

• Provides strategic oversight, makes key decisions, and reviews progress through a Steering Committee. • Provides domain expertise, defines use case requirements, and validates solutions through Business Leads & SMEs. • Owns the day-to-day execution of specific initiatives through a Product/Program Manager.	• Provides business/industry context and technical expertise on Agentic AI, ecosystem design, and solution architecture through an AI Industry Architect/Consultant. • Offers specialized expertise as needed through SMEs (e.g., Data & Analytics, AI platforms, Security). • Collaborates on innovating and co-creating novel agent-based solutions and MVPs through a Create Agent Team. This team may also aim to significantly accelerate the delivery of high-quality, data-driven Agentic AI strategies by automating core framework activities.

Outputs and Deliverables

- ☐ Agentic AI Vision Statement
- ☐ Prioritized Portfolio of Strategic Agentic Initiatives (with delta value canvas output)
- ☐ Conceptual Agent Ecosystem Blueprint
- ☐ Phased Strategic Execution Roadmap (including MVPs)
- ☐ Organizational Change, Talent, & Ethical AI Implementation Roadmap
- ☐ Agentic Transformation OKRs & Value Realization Framework
- ☐ ROI Financial Modeling
- ☐ Value Tracking Dashboards (e.g., in Looker)

The Agentic AI Development Lifecycle

From the strategic center above which is Strategy, Ecosystems, and Value, we breathe life into the vision through a powerful, iterative execution cycle designed for speed and agility. This is where ideas become reality.

We start with *Reimagine Business Processes & Systems* where we think beyond simple automation. Instead of just making existing workflows faster, we fundamentally reimagine how work gets done to unlock the full, transformative potential of agentic AI. This is about redesigning systems and processes for a world where autonomous agents can reason, plan, and execute complex tasks.

Then we move to *Agentic Use Case Design & Prototyping* through which the reimagined vision is then translated into concrete, high-impact use cases. We translate domain challenges into specific agent journeys and prototypes, allowing us to validate ideas quickly and prioritize those with the greatest potential business outcome. This step de-risks the transformation by making the abstract tangible before significant investment.

With a clear blueprint, we move to *Agent Building* where we develop single and multi-agent systems through an agile, iterative process. This allows for flexibility, the incorporation of new data sources and tools, and the development of sophisticated agent-to-agent collaboration. We build, test, and learn in rapid cycles, allowing complexity to emerge gracefully.

Lastly, we go to *Quality and Alignment* which is not a linear path but a continuous loop. Throughout the cycle, we embed a rigorous focus on quality and alignment. Through continuous agent auditing, automated evaluation, Human-in-the-Loop (HITL) verification, and robust monitoring, we ensure agents perform as intended. This allows us to manage emergent behaviors, maintain trustworthiness, and guarantee that the agents remain aligned with our strategic goals and ethical principles.

Finally, this entire cycle is built upon the bedrock of *Horizontal & Foundational Capabilities*. These are the cross-cutting, enterprise-grade enablers that allow agentic systems to scale securely and reliably. This includes architecting for autonomy, establishing robust data and data platforms, implementing vigilant security and risk management, and defining clear governance and operational models (AgentOps). These foundational pillars are what elevate a successful pilot into a true, integrated enterprise capability as you will see later.

Reimagining Business Processes & Systems

Contributors: Ryan Faris , Patrick Nestler , Keyur Mehta Amulya Saridey

Objectives

- **Outcome-Driven Process Redesign**
    - Shift the focus from "as-is" processes to "what-if" future states.
    - Deconstruct core business goals to design new, dynamic workflows where intelligent agents autonomously achieve strategic outcomes (e.g., increased efficiency, reduced cycle times, lower error rates), overcoming legacy constraints.
- **Human-Agent Collaboration Model Design**
    - Define the collaboration model between humans and AI agents.

    - Architect new workflows that delegate tasks to agents while elevating human expertise towards judgment, creative problem-solving, and high-value relationship building, ultimately improving user adoption and satisfaction.
- **Conceptual Agentic Architecture**
    - Create a high-level conceptual architecture for the agentic ecosystem and its interactions with other systems.
    - Establish a common understanding of the system's purpose and scope, highlighting interaction points and ensuring robust system performance (e.g., low latency, high uptime, reduced cost per transaction).
- **Operational Readiness and Change Management**
    - Catalyze organizational change by embedding an agentic mindset.
    - Manage the cultural shift by equipping the workforce with the skills and confidence to not just work alongside agents, but to lead and innovate with them.

Inputs

Business	Technical
• Outputs from the “Strategy and Value” phase (Vision, Prioritized Initiatives, Roadmap, OKRs) • Strategic Journey Maps (Value stream analysis, “What if” scenarios) • Organizational data (Employee roles, stakeholder analysis, change readiness surveys)	• "As-is" process maps, existing system architecture, data flows, and API documentation

Key Activities

1. **Key activity:** Opportunity Discovery & Deconstruction
    a. **Motivation/goal:** To create a data-driven understanding of current processes, uncovering hidden inefficiencies and identifying the most impactful areas for agent automation and augmentation.
    b. Key steps:
        i. Decompose strategic business goals or priority processes into core constituent tasks and decision points.
        ii. Classify tasks for automation potential (fully automated, agent-assisted, human-only) across a spectrum of autonomy.
        iii. Quantify the time, cost, and error rate of "as-is" tasks to establish a baseline for improvement.

    c. **Tools/techniques used:** Value Stream and Task Analysis Frameworks, Business Process Model and Notation (BPMN) and Process Mining Tools.
2. **Key activity:** Designing ‘To-be’ Agentic Processes
    a. Motivation/goal: To fundamentally redesign workflows around a human-agent collaborative model, moving beyond simple automation to create new, more effective ways of delivering value.
    b. Key steps:
        i. Break down high-level processes into well-defined tasks for AI agents (e.g., data gathering, calculations, communications, recommendations).
        ii. Design end-to-end workflows incorporating learning mechanisms, data sources, action triggers, and human oversight/escalation paths.
        iii. Conduct a Process Design Review to get formal sign-off from Business Process Owners and SMEs on the "to-be" designs.
    c. **Tools/techniques used:** Value Stream and Task Analysis Frameworks, Business Process Model and Notation (BPMN).
3. **Key activity:** Conceptual Architecture Design
    a. Motivation/goal: To establish a coherent, high-level blueprint for the agentic system that is scalable, governable, and understood by all stakeholders.
    b. Key steps:
        i. Define and charter the primary agent archetypes (e.g., Data-Gathering, Analyst, Action Agents) and their distinct roles.
        ii. Map high-level interaction patterns, communication channels, and data exchange between agents, including the role of APIs.
        iii. Establish the overall governance framework, control mechanisms, and the role of human oversight within the ecosystem.
        iv. Hold a Technical Architecture Review with IT architects and security teams to validate the designs for feasibility, security, and scalability.
    c. **Tools/techniques used:** Agent architecture patterns, integration frameworks.
4. **Key activity:** Planning for Organizational Change and Adoption
    a. Motivation/goal: To proactively manage the human element of the transformation, fostering buy-in, building necessary skills, and ensuring a smooth transition to the new agent-enhanced operating model.
    b. Key steps:
        i. Develop a targeted communication plan for all affected stakeholders.
        ii. Establish feedback mechanisms and support channels for end-users.
        iii. Define training programs to upskill the workforce for new roles and responsibilities.
        iv. Conduct a Change Readiness Assessment with the Steering Committee and leadership to approve the execution of the change management plan.

c. **Tools/techniques used:** Change Management Models (e.g., ADKAR), Stakeholder Analysis Matrix.

Roles and Responsibilities

Customer Responsibilities	Google Responsibilities
• Accountable for the successful redesign and performance of their process through a Business Process Owner. • Owns the technical integration strategy and enterprise alignment through an IT / Enterprise Architect. • Develops and executes the organizational readiness plan through a Change Management Lead. • Actively participate in the design, testing, and refinement of new processes through End-Users / Subject Matter Experts.	• Facilitates the translation of strategy into operational design through the Innovation Team Engagement Lead. • Provides deep expertise on agentic process patterns and human-agent interaction design through an AI Industry Architect / Consultant. • Advises on strategies to accelerate user adoption and manage resistance through a Change Management Consultant. • Provides oversight and ensures alignment with strategic intent through the Steering Committee.

Outputs and Deliverables

- [ ] "As-Is" Process Analysis & Findings Report
- [ ] "To-Be" Agent-Enhanced Process Designs
- [ ] Human-Agent Interaction Models & Redefined Role Descriptions
- [ ] Agentic Integration Blueprint (including for API and Data)
- [ ] Change Management & Training Plan
- [ ] Updated Risk Register (focusing on operational, technical, and adoption risks)

Identifying Agentic AI Use Cases, Designs, and Rapid Prototyping

Contributors: Brisa Araujo, Jorge Gomez, Dovile Mikalauskaite, Héctor González, Guillermo Noriega, Gabe Corbett, Itziar Leguinazabal, Amulya Saridey

Objectives

- **Understanding Building Blocks of Agentic Systems**
  - Define the core capabilities an agent can execute to solve problems, including:

        - **Reasoning & Planning:** Decompose a high-level goal into a sequence of actionable steps.
        - **Synthesizing & Transforming:** Distill, organize, and convert information.
        - **Generating & Evaluating:** Create new content and assess information against a standard.
        - **Taking Actions:** Interact with systems and tools (e.g., APIs, databases) to execute tasks.
        - **Memory & Learning:** Retain information from interactions and continuously improve performance.
- **Use Case Identification & Definition**
    - Move from raw data and business problems to a well-defined, comprehensive list of potential Agentic AI opportunities, aligned with strategic goals (e.g., process automation, human augmentation, insight discovery, complex problem-solving).
- **Strategic Prioritization**
    - Select the most impactful use cases to pursue first by balancing business value, technical feasibility, and strategic urgency. Implement Responsible AI considerations to build trust and ensure compliance
- **User-Centric Design**
    - Translate a prioritized use case into a detailed "golden path" design, including user goals, agent tasks, and required tools.
- **Rapid Prototyping & Validation**
    - Build and test functional prototypes to validate assumptions, gather user feedback, and assess technical feasibility before full-scale development.

Inputs

Business	Technical
• Corporate strategic objectives & OKRs. • Operational KPIs & performance data. • Stakeholder interviews on pain points (high toil, error-prone, research heavy, delegation, time sensitive tasks). • Defined user personas or customer research. • Journey maps & value stream	• Existing business process maps & procedures guides. • System architecture diagrams & data flow maps. • API documentation & data source availability.

analysis. • Any existing Responsible AI framework and compliance requirements	

Key Activities

1. **Key activity:** Use Case Identification & Definition
    1. **Motivation/goal:** To move from raw data and business problems to a well-defined, comprehensive list of potential Agentic AI opportunities.
    2. **Key steps:**
        1. Understand the opportunity from multiple angles; user/customer, business (internal and external factors), technology, etc.
        2. Analyze existing data (KPIs, transcripts, process maps) to form a hypothesis of opportunities and challenges.
        3. Synthesize findings into key user personas, aligning insights with their primary pain points and goals.
        4. Conduct a workshop with a cross-functional team to brainstorm a long-list of potential use cases, framed from the user's perspective.
        5. Group and refine the brainstormed ideas into a structured taxonomy of user goals to create a shared language.
    3. **Tools/techniques used:** Data analysis tools, Persona development frameworks, Facilitated Ideation Workshops.
2. **Key activity:** Use Case Prioritization
    1. **Motivation/goal:** To strategically select the most impactful use cases to pursue first, balancing business value against implementation effort.
    2. **Key steps:**
        1. Define a prioritization framework with clear criteria for business value, technical effort and strategic urgency. Can be based on:
            1. Value/Impact Criteria: Business value (e.g., call volume reduction, minutes of work saved, CSAT/NPS impact, revenue generation) and Strategic Opportunity Tier (e.g., is this Process Automation or Human Augmentation?)
            2. Effort/Feasibility Criteria: Technical complexity, data availability/quality, dependency on other systems/APIs.
            3. Urgency/Timing Criteria: Business drivers that create time-sensitive pressure, such as regulatory deadlines, competitive threats, alignment with a major product launch, or the need to fix a critical, brand-damaging issue.

        2. Score each use case against the framework in a collaborative workshop.
        3. Visualize the results (e.g., using a "magic quadrant") to clearly communicate relative priorities.
        4. Sequence the top-ranked use cases into a high-level roadmap (e.g., by quarter).
    3. **Tools/techniques used:** Prioritization matrices (e.g., Value vs. Effort), Roadmapping tools, Facilitated Prioritization Workshops.
3. **Key activity:** Conduct Responsible AI Workshops
    1. **Motivation/goal:** To proactively identify and mitigate potential AI risks by ensuring that prioritized use cases align with ethical principles, build user trust, and comply with regulatory standards from the very start.
    2. **Key steps:**
        1. Conduct an initial workshop to establish a common understanding of what Responsible AI and fairness mean, and discuss the potential risks and consequences of unmitigated systems.
        2. Review the Responsible AI Framework, outlining the key principles and guidelines for building trustworthy AI.
        3. Apply the Responsible AI Framework in a hands-on exercise using a sample use case, showing the process in a controlled environment.
        4. Facilitate a guided session where the customer can apply the framework to the above prioritized agent use case to identify specific risks and define initial mitigation strategies.
    3. **Tools/techniques used:** Workshop materials on Responsible AI overview, Responsible AI Use Case Framework
4. **Key activity:** Agentic Use Case Design
    1. **Motivation/goal:** To translate a prioritized use case into a detailed, user-centric "golden path" design and a clear functional architecture.
    2. **Key steps:**
        1. For each prioritized use case, define the Critical User Journey (CUJ) by outlining the user's goal, the agent's autonomous tasks, and the required tools (APIs, data sources).
        2. Decompose each agent task into agile user stories for the development backlog.
        3. Map tasks to core agentic capabilities (e.g., Plan, Synthesize, Act) to define the agent's function.
        4. Sketch a functional architecture diagram showing how AI components will interact to deliver the CUJ.
    3. **Tools/techniques used:** Critical User Journey (CUJ) Framework, User Story Mapping, Functional Architecture Diagramming.
5. **Key activity:** Rapid Prototyping & Validation

1. **Motivation/goal:** To rapidly build and test a functional representation of the designed CUJ to validate assumptions, gather user feedback, and assess technical feasibility.
2. **Key steps:**
    1. Define the prototype's scope and limitations, focusing on the key assumptions that need testing.
    2. Develop an interactive prototype using appropriate tools (e.g., Figma, low-code builders, conversational AI platforms, ADK).
    3. Conduct usability testing sessions with representative users to gather feedback on the experience.
    4. Refine the CUJ and functional designs based on feedback and present the validated prototype for handoff to development.
3. **Tools/techniques used:** Prototyping tools (Figma, etc.), Conversational AI platforms, Usability testing methodologies.

Roles and Responsibilities

Customer Responsibilities	Google Responsibilities
• Provide deep domain knowledge and validate user needs through Business SMEs. • Own the use case backlog and define business requirements as the Product Manager. • Define the technical vision and ensure alignment with enterprise systems as the IT / Enterprise Architect. • Participate in user testing and provide direct feedback as End-Users.	• Facilitate workshops and guide the overall process through an Innovation or Engagement Lead. • Provide expertise on agentic patterns and technical feasibility as an AI Architect / Consultant. • Collaborate on building and testing prototypes as part of a Prototyping / Create Agent Team.

Outputs and Deliverables

- [ ] Data Insights Report & Defined User Personas
- [ ] Comprehensive Use Case Long-List
- [ ] Refined Use Case & Goal Taxonomy
- [ ] Documented Prioritization Framework
- [ ] Visual Prioritization Map & Prioritized Use Case Shortlist
- [ ] Responsible AI Recommendations Report

- [ ] Initial Use Case Roadmap
- [ ] Detailed Critical User Journey (CUJ) Definitions
- [ ] Initial Set of User Stories for Development
- [ ] Functional Architecture Diagram
- [ ] Data & API Requirements List
- [ ] Interactive Prototype
- [ ] User Testing & Feedback Report
- [ ] Validated (or Invalidated) Assumptions for Handoff

Building Single and Multi-Agent Systems

Contributors: Enrique Chan, Pablo Gaeta, Afshaan Mazagonwalla, Amulya Saridey, Ievgenii Siroshtan, Semih Duru, Javier Garcia Puga, Amulya Saridey, Sriram Srinivasan

Objectives

- **Develop agents in a scalable, repeatable manner**
  - Determine appropriate level of complexity for agents in focus, aiming for the minimal complexity required for the task(s) in scope
  - Select design pattern most applicable to the use case at hand, considering the task(s) in scope and targets for latency
  - Key Metrics: Agent performance metrics (latency, time-to-first-token), agent/tool utilization rates, error rates.
- **Implement agent-specific guardrails**
  - Define risks to agent behavior, including safety measures for inputs and outputs and consequential actions
  - Implement modules for screening inputs, outputs, and action taking where applicable
- **Integrate with broader platform quality, logging, and monitoring**
  - Leverage evaluation pipelines and monitoring standards (defined as part of Driving Quality and Alignment) to ensure newly developed agents fit into enterprise approach to agent health
  - Key Metrics: Alignment with business OKRs, Agent Quality Evaluation

Inputs

Business	Technical

• Business unit strategic objectives and existing strategy documents. • Defined use cases and problem statements. • Internal pain-point analysis and operational challenges. • User persona and journey maps.	• List of available 1st party, 3rd party, and marketplace agent solutions/models. • Documentation of existing systems, APIS, and data sources. • Data classification policies (public, internal, confidential, etc.). • Assessments of data readiness and availability. • (if available) Working prototype from use case definition phase.

Key Activities

1. **Key activity:** Agentic System Design & Architecture
    a. **Motivation/goal:** To select the optimal agent architecture and design pattern that meets the use case requirements for complexity, performance, and scalability or expand as needed from the above section
    b. Key steps:
        i. Determine if a single-agent or multi-agent architecture is required based on the problem's complexity. The Agentic Maturity Roadmap can be a useful tool here in defining the complexity for the workflow/use case in question.
        ii. If multi-agent, select the most appropriate design pattern (e.g., Hierarchical, Dispatcher).
        iii. Design the end-to-end agent architecture, defining the roles of agents, their tools, interaction flows, and the strategy for context management, including how short-term and long-term memory will be orchestrated.
    c. **Tools/techniques used:** Agentic Design Patterns (Hierarchical, Collaborative, Dispatcher, etc.), Architecture Diagramming Tools
2. **Key activity:** Agent & Tool Development
    a. Motivation/goal: To build the core logic of the agent(s) and integrate the necessary external tools and memory systems to enable them to perceive, reason, and act effectively.
    b. Key steps:
        i. Select the technology stack (e.g., Agent framework like ADK, LLM, programming language, databases).
        ii. Develop and integrate the external tools (e.g., APIs, databases, RAG systems) with robust error handling.

iii. Implement the agent's core reasoning loop, LLM integration, and tool-calling logic.
iv. Implement Memory & Context Management:
    1. Short-Term Memory: For maintaining immediate conversational context, implemented within the Agent Engine (e.g., as a Managed Session in RAM), with persistence options like Firestore.
    2. Long-Term Memory: For retaining persistent knowledge, using solutions like BigQuery, GCS, or Cloud Spanner.
v. Develop the user interface or API endpoints for interaction (if applicable).

c. **Tools/techniques used:** Agent Development Frameworks (e.g., ADK), LLMs (e.g., Google Gemini), Databases & Storage (Firestore, BigQuery, GCS, Cloud Spanner etc), Cloud Platforms (e.g., GCP)

3. **Key activity:** Guardrail & Safety Implementation
    a. Motivation/goal: To design and implement the safety mechanisms, policies, and guardrails required to prevent harmful or unintended agent behavior.
    b. Key steps:
        i. Define risks related to agent inputs, outputs, and actions.
        ii. Implement screening modules to validate inputs and outputs against safety policies.
        iii. Implement checks and balances for any consequential actions the agent might take (e.g., human-in-the-loop for high-risk actions, red teaming etc).
    c. **Tools/techniques used:** Human-in-the-loop (HITL) approval workflows, red & blue teaming code / safeguard techniques etc.
4. **Key activity:** Integration, Deployment & Monitoring
    a. **Motivation/goal:** To integrate the agent into enterprise MLOps/AgentOps practices, deploy it to the target environment, and ensure it can be monitored for health and performance.
    b. Key steps:
        i. Set up comprehensive logging to track agent behavior, decisions, tool usage, and errors.
        ii. Integrate with enterprise monitoring and evaluation pipelines.
        iii. Deploy the agent to the target environment (e.g., Cloud Functions, GKE).
        iv. Establish a process for iterative refinement based on performance data and user feedback.
    c. **Tools/techniques used:** CI/CD Pipelines, Logging & Monitoring Tools (e.g., Cloud Logging, Cloud Monitoring), Evaluation Frameworks

Roles and Responsibilities

Customer Responsibilities	Google Responsibilities
• Defines the vision, prioritizes use cases, and represents the user through a Product Manager • Provide strategic context, resources, and champion the initiative through business stakeholders • Provide domain expertise to inform agent knowledge and tools through Subject Matter Experts • Participate in UAT and provide real-world feedback through End Users	• Leads the design, development, and optimization of the agent(s) through an AI/ML Engineer • Develops and integrates tools, APIs, and handles deployment through a Software Engineer. • Designs the user interaction and feedback mechanisms through UI/UX designer • Develops and executes evaluation pipelines and testing strategies through QA/Test Engineer

Outputs and Deliverables

- [ ] The Agent(s) Codebase, including:
  - [ ] Agent Logic/Orchestration Code
  - [ ] Tool Definitions/Integrations
  - [ ] LLM Integration & Configuration
  - [ ] Memory Management Code (for short-term, long-term, and context services)
  - [ ] Guardrails and Safety Mechanism Code
- [ ] Agentic System Architecture Diagram
- [ ] Technical Design Document
- [ ] Deployment Artifacts (e.g., Dockerfiles, Terraform scripts)
- [ ] Continuous Improvement & Optimization Plan

Driving Quality & Alignment

Contributors: Ram Seshadri, Pablo Gaeta, Dovile Mikalauskaite, Héctor González, Steve Kluger

Objectives

- **Establish a scalable agent quality and evaluation infrastructure for AI agents.**

    - Define a deliberate, methodical approach to quality measurement and overall evaluation that can be used repeatedly, without the need for redesign with each newly developed agent
    - Develop modular, config-driven evaluation pipelines and evaluation agents that can be configured to align with evaluation needs for each particular agent
- **Develop and implement a standardized set of metrics to assess agent performance and safety.**
    - Build enterprise repository of trusted evaluation metrics and associated scripts required to perform them
    - Understand the role of pair- and point-wise evaluation, as well as computational vs. judge-model approaches
- **Define best practices for agent monitoring and quality control.**
    - With evaluation pipelines and agents built, understand when and how to trigger evaluation
    - Define agent monitoring via automated evaluation and key metrics to track on an ongoing basis in production

Inputs

Business	Technical
• Use case documentation, outlining the agent's purpose, functionality and expected domain, access and user profiles. • Subject-matter-experts in use-case-specific evaluation. • Pain-point analysis outputs. • Responsible AI Frameworks.	• Existing technology landscape and data readiness assessments. • Any existing evaluation pipelines, metrics, or associated scripts. • “Golden examples, where available, indicating desired agent outputs and decisions.

Key Activities

1. **Key activity:** Evaluation planning and design
    a. **Motivation/goal:** Ensure the scope, goals and requirements of agent evaluation are met
    b. Key steps:
        i. Define business and technical requirements for evaluation
        ii. Determine agent evaluation approaches, including pair-wise, point-wise, judge-model, and computational metrics
        iii. Define technical requirements to align to business AI goals

iv. Define logs and metrics required to demonstrate alignment to the business goals
v. Define data related requirements, i.e. approved sources, acceptable quality and governance requirements
vi. Define evaluation cadence for agents in production and being developed
vii. Define roles and responsibilities for agent evaluation teams

2. **Key activity:** Develop evaluation pipeline and/or agents
    a. **Motivation/goal:** Automate the evaluation process where possible to promote active monitoring and accelerate development
    b. **Key steps:**
        i. Define relevant evaluation metrics based on your use case
        ii. Design end-to-end evaluation pipeline, including:
            1. Input tables containing golden examples and/or evaluation dataset
            2. Inference module for the agent(s) to perform act against test cases
            3. Output tables where inference outcomes and associated metadata will be stored
            4. Evaluation module where the judge model and scripts to compute evaluation metrics will be run against outputs
            5. Evaluation results table where reported metrics will be stored

        Note: The above can also be accomplished via an evaluation *agent* that executes the steps in the pipeline. Regardless of the tool used, the modules should be largely the same.
        iii. Develop evaluation pipeline in a config-driven manner, allowing for users to set evaluation metrics for each pipeline instance
3. **Key activity:** Operationalize quality processes
    a. **Motivation/goal:** Develop and deploy processes for a each use case for when quality checks will be performed during development and in production
    b. **Key steps:**
        i. Define and document the frequency and associated pipeline instances for quality checks to be conducted during the development process, including as part of any refactoring and updates to deployed agents
        ii. Define and document quality checks as part of ongoing quality monitoring for agents deployed in production
        iii. Implement relevant scheduled pipeline runs and associated tasks, such as:
            1. Sampling of agent metadata from production deployments
            2. Logging of production actions performed by agents
4. **Key activity:** Implement dashboards and alerts

a. **Motivation/goal:** Leverage the data produced by operationalized quality processes to provide appropriate visibility for business and technical stakeholders
b. **Key Steps:**
    i. Define desired visualizations and alerts based on business needs
    ii. Map defined visualizations and alerts back to data collected during the quality process, updating the associated evaluation pipelines and agents as necessary
    iii. Develop dashboards with end-users in mind, accounting for the scope of stakeholders

Roles and responsibilities

Customer Responsibilities	Google Responsibilities
• Sets strategic goals and ROI targets that guide the definition of "success" for the agent through an Executive Strategist • Drives the agent's use case, defines business-critical evaluation rubrics, and provides operational context for quality control through an Agent Owner • Coordinates internal teams, scope, timing, and resources for evaluation activities through a Project Manager • Define evaluation metrics, create "golden example" test datasets, and provide the ground truth for measuring agent quality through Domain Experters. • Evaluate agents against the defined rubrics, provide real-world feedback, and help design user feedback mechanisms for retraining through Agent Testers & HITL Designers • Ensure the evaluation process and agent integrations align with internal corporate security policies and access controls through Security Engineers	• Designs evaluation metrics and approaches that align with the customer's business objectives and provides guidance on best practices through AI Consultant • Design and develop the technical evaluation pipelines, fine-tune models based on results, and build the monitoring dashboards and visualizations through AI/Data Engineers • Manage the agent registry, gateways, deployment processes, and automated alerting to ensure operational stability through ACT Experts • Advise on secure development practices and ensure the deployed solution and evaluation pipelines meet robust security standards through Security Engineers • Implement automation for running evaluations continuously and integrate the system into the customer's MLOps/AgentOps framework

Outputs and Deliverables

- [ ] Quality framework documentation
- [ ] Evaluation pipeline(s)
- [ ] Process design and automation to operationalize evaluation
- [ ] Visualizations, reports, and alerting enabled on quality metrics

Horizontal & Foundational Capabilities

While the iterative cycle is powerful for creating effective agents, true transformation happens when these innovations can be deployed, managed, and trusted at an enterprise scale. A brilliant agent working in isolation is an experiment; a network of agents operating securely across the business is a competitive advantage.

This is the crucial role of our Horizontal & Foundational Capabilities. These are not project-specific tasks but the underlying, cross-cutting infrastructure, standards, and processes that support every agentic initiative. They provide the stability and consistency needed to move from concept to scaled reality. This includes:

- Architecture: The master blueprint for how agentic systems integrate with each other and existing enterprise technology.
- Data and Data Platforms: The lifeblood of AI, ensuring agents have access to high-quality, reliable, and governed information.
- Security & Risk Management: The essential guardrails that protect the enterprise, ensure compliance, and build trust in autonomous systems.
- Governance, Change, & Operations: The command center for defining rules, managing the human-to-agent transition, and establishing clear operational accountability.
- Platforms & AgentOps: The specialized toolchains and practices required to efficiently build, test, deploy, and monitor the agentic workforce.

Investing in these capabilities ensures that our agentic solutions are not just innovative, but also robust, scalable, and secure.

Agentic Architecture

Contributors: Enrique Chan, Pablo Gaeta, Afshaan Mazagonwalla, Anna Whelan, Ben Swenka, Patrick Nestler, Keyur Mehta, Javier Garcia Puga, Semih Duru, Federico Preli, Ramya Gowrugolla, Olu Akinrolabu, Amulya Saridey

Objectives

- **Establish Agentic Core Architecture Foundation**
  - Design the core systems and standards that ensure all agents operate within enterprise guardrails, are cost-effective, and can be trusted.
- **Architect Enterprise-Grade Platform to Deploy Agents at Scale**
  - Provide the architectural blueprints and platforms that allow development teams to build, test, and deploy agents consistently and efficiently.
- **Integrate Agentic Systems with Core Enterprise Platforms**
  - Architect the connective tissue that seamlessly weaves agents into the fabric of the organization's existing data, service, and microservice landscapes.

Inputs

Business	Technical
• Corporate / business unit strategic objectives and digital transformation roadmaps. • Prioritized list of business problems & potential agentic use cases. • List of relevant compliance standards (e.g., GDPR, HIPAA) to inform architectural data handling patterns.	• Any relevant, current state Enterprise Architecture documentation. • Documentation on existing Service Mesh, Microservices, and API Gateway implementations. • Data Mesh / Data Platform strategy documents and data catalogs. • List of existing security tools, cloud platforms, and CI/CD pipelines to ensure architectural compatibility.

Key Activities

1. **Key activity:** Establish key design principles
   a. **Motivation/Goal:** To establish a clear set of guiding rules that ensure all agentic systems are developed in a consistent, scalable, secure, and responsible manner. These principles act as a compass for architects and developers, preventing technical silos and promoting long-term maintainability.
   b. Key steps:
      i. Review foundational best practices (e.g., modularity, loose coupling, and separation of concerns).
      ii. Define agent-specific principles (e.g., graduated autonomy, least privilege for tools, observability by default, human-in-the-loop design).
      iii. Align with strategic & ethical goals (e.g., Responsible AI standards).
      iv. Formalize and socialize principles in a central repository.

    c. **Tools/techniques:** Workshops/discovery sessions, software architecture principles, Responsible AI workshops
2. **Key activity:** Perform current technical capability assessment and define the capability roadmap
    a. **Motivation/Goal:** To create a strategic, phased plan that aligns the development of agentic architecture with business priorities, ensuring that foundational capabilities are built before more advanced ones are attempted
    b. **Key steps:**
        i. Audit the current state and team expertise against the agentic capability model and assess target state needed to implement identified strategic use cases
        ii. Identify and prioritize capability gaps, ranking them based on business value, dependencies, and complexity
        iii. Group capabilities into phased releases, where each phase builds on the previous:
            1. Single Purpose Agents: Developing and deploying AI agents that are highly specialized and perform a single, well-defined task
            2. Multi-action Agents Level: Agents that can perform a sequence of actions or more complex, multi-step tasks.
            3. Agent Collaboration & Orchestration: Multiple, distinct AI agents work together in a coordinated manner to achieve a larger, more complex goal.
        iv. Develop and socialize a visual roadmap for technical capability
    c. **Tools/Techniques:** Capability map and gap analysis, Agentic AI architecture maturity framework, Prioritization frameworks, Necessary stakeholder interviews/workshops
3. **Key activity:** Determine service options and sourcing (GCP services vs 3p services vs build)
    a. **Motivation/Goal:** To make strategic and cost-effective decisions on the core technology stack, accelerating development by leveraging managed services and proven open-source solutions where possible.
    b. **Key steps:**
        i. Identify core technological components (e.g., LLMs, orchestration, vector DBs, deployment runtimes).
        ii. Evaluate buy and leverage options by analyzing managed services (e.g., Vertex AI Agent Engine, Cloud Run) and open-source frameworks (e.g., ADK, LangGraph) based on criteria like performance, cost, security, and developer skill sets.
        iii. Assess build scenarios by realistically evaluating the internal effort, expertise required, and total cost of ownership for creating and

maintaining custom components, reserving this path for capabilities that are truly unique to the business.

iv. Document the sourcing strategy and a list of approved technologies.

c. **Tools/techniques:** Mapping capabilities with GCP Services, Scorecard for evaluating said criteria, TCO analysis, Agent building decision tree

4. **Key activity:** Define the Architecture Blueprint
    a. **Motivation/Goal:** To translate strategic goals and design principles into a concrete, enterprise-grade architectural blueprint that provides a reusable pattern for building scalable, secure, and integrated agent systems.
    b. **Key steps:**
        i. Select appropriate design patterns (e.g., Hierarchical, Dispatcher, Sequential Pipeline for multi-agent systems)
        ii. Define the high-level layered architecture (e.g., Consumption, Interface, Data Access layers)
        iii. Specify foundational integration and communication standards such as Zero Trust and A2A/MCP protocol details.
        iv. Mandate integration with foundational capabilities by ensuring the blueprint explicitly incorporates cross-cutting services. The architecture must account for:
            1. Observability: See Platforms and AgentOps for implementation details.
            2. Security: See Security and Risk Management for controls and governance.
            3. Evaluation & Quality: See Driving Quality & Alignment for frameworks.
            4. Data Access: See Data and Data Platforms for data product consumption patterns.
            5. Governance: See Governance and Operations for operational readiness and stage-gate reviews.
    c. **Tools/Techniques:** Agent design patterns, technical architecture patterns, agent project templates (i.e. Agent Starter Pack), etc.

Outputs and Deliverables

- **Current Capability Assessment:** Evaluate existing agentic architecture capabilities to identify strengths and weaknesses.
- **Strategic Roadmap.** Prioritized roadmap by assessing your current capabilities and establishing core design principles. This provides an actionable plan to guide your agentic transformation.

- **Architectural Blueprint.** Technical plan for your platform, including detailed diagrams and a document justifying key design choices and technology recommendations. This ensures a transparent and future-proof architecture.
- **Implementation Patterns & Pilot**. Library of reusable technical patterns and a production-ready pilot architecture that puts them into practice. This gives your team a tangible, best-practice example to build upon.
- **Scaling & Acceleration Toolkit.** A robust governance and security framework with practical delivery accelerators. This equips your team to scale new agentic solutions securely and efficiently.

Data and Data Platforms

Contributors: Sonia Loona, Pablo Rodriguez Defino, Ryan Rezvani, Ada Lau, Mike Hilton, Vijay Pandurengan, Maurizio Colleluori, Lorenzo Caggioni, Lorenzo Nigro, Javier Garcia Puga

Objectives

- **Architect and Deploy a Governed Agentic Data Platform**
    - Build and launch a modular, multi-layered data architecture to serve as the single source of truth for all agentic systems, ensuring data is discoverable, reliable, and secure by default.
- **Implement Production-Grade Data Pipelines for Agent Intelligence**
    - Productionize data ingestion and processing workflows that automatically enrich raw data with semantic meaning, context, and vector embeddings, leveraging RAG and Knowledge Graphs to directly fuel advanced agent reasoning and performance.
- **Develop and Deploy a Portfolio of Specialized Data and Business Agents**
    - Build, test, and release a targeted set of intelligent agents that leverage the data platform to automate critical data retrieval, execute specific business processes, and enable conversational data analysis for key use cases.

Inputs

Business	Technical
• Agentic business use case roadmap, where available, including associated data sources • Understanding of data users and associated skillsets	• Existing data management and governance processes and tools • Known data quality gaps

Key Activities

1. **Key activity**: Design and Establish the Agentic Data Platform
    1. **Motivation/goal**: To build a modular, scalable, and governed data architecture that serves as the foundation for all agentic AI development, ensuring data quality, accessibility, and cost-effectiveness.
    2. **Key steps**:
        1. Design a modular, multi-layer architecture encompassing data storage, transformation, access, interface and developer tooling. Consult The Agentic Data Platform Design Blueprint for more an example architecture.
        2. Adopt a Data Mesh architecture to organize data into discoverable, domain-owned data products.
        3. Implement a Medallion architecture (e.g., Bronze, Silver, Gold tiers) within the data warehouse to ensure data reliability and trustworthiness for consumption by agents.
        4. Establish federated and dynamic data discovery mechanisms to allow agents to find the data they need across the enterprise.
        5. Implement comprehensive data governance, including metadata management, data protection, and ethical oversight.
    3. **Tools/techniques used**: Agentic Data Platform Design Blueprint, Data Mesh Architecture, Medallion Data Warehouse Architecture.
2. **Key activity**: Implement Data Ingestion and Processing Patterns for Agentic AI
    1. **Motivation/goal**: To transform raw data into a state that is optimized for agent consumption, enabling agents to understand, reason, and act with greater intelligence, context, and efficiency.
    2. **Key steps**:
        1. Develop data ingestion pipelines that perform semantic enrichment, contextual chunking, and integrated embedding generation as core outputs.

        2. Implement real-time processing capabilities to ensure data is fresh and supports agent responsiveness.
        3. Leverage Vector Databases and Retrieval-Augmented Generation (RAG) techniques to optimize information retrieval for agent efficiency.
        4. Build out Knowledge Graphs to provide agents with structured understanding for more advanced reasoning.
    3. **Tools/techniques used**: Data Foundations for Agents, Retrieval-Augmented Generation (RAG), GraphRAG, Knowledge Graphs, Vector Databases, real-time data processing pipelines.
3. **Key activity**: Develop and Deploy Specialized Data and Business Agents
    1. **Motivation/goal**: To unlock the value within enterprise data by creating intelligent agents that can understand business context, perform data-centric tasks, and transform how applications are designed and built.
    2. **Key steps**:
        1. Develop specialized 'Data Agents' capable of returning relevant and grounded data responses.
        2. Shift from static API access to a dynamic 'tool use' approach where agents can leverage APIs and direct database connections to gather information and execute tasks.
        3. Build Data-Centric Business Agents that leverage the data platform to provide insights and perform tasks for specific business use cases.
        4. Create App-centric Business Agents to transform application design and accelerate the development of new features.
        5. Leverage conversational analytics solutions to build agents that can interact with and analyze data through natural language.
    3. **Tools/techniques used**: Agent Development Kit (ADK), LangChain, LangGraph, BigQuery (with native Data Agents), Looker Conversational Analytics API, Agent-as-a-Service, Model Context Protocol (MCP), Agent-2-Agent (A2A) Protocol.
4. **Key activity**: Drive Enablement and Adoption
    1. **Motivation/goal**: To ensure that all data personas within the organization have the necessary skills, knowledge, and support to effectively leverage the Agentic Data Platform and its associated tools.
    2. **Key steps**:
        1. Identify key data personas and their specific skill sets and needs.
        2. Develop and deliver targeted training materials and enablement sessions for each persona.
        3. Establish a center of excellence or community of practice to share best practices and provide ongoing support.
    3. **Tools/techniques used**: Training and enablement materials, workshops, documentation.

Roles and Responsibilities

Customer Responsibilities	Google Responsibilities
• Champion the initiative, securing resources, budget, and removing organizational roadblocks through executive sponsorship • Provide business context, grounding information for agents, and validate data product accuracy through data domain owners / SMEs • Define business agent requirements and conduct User Acceptance Testing (UAT) to ensure solutions meet needs through Business Analysis and end users	• Guides strategic alignment, quantifies use case value, and ensures objectives are met through an Innovation Engagement Lead • Provides business and industry context for the data platform design and offers technical expertise in Agentic AI, data architecture, and Google Cloud solutions through an AI Architect • Offer specialized expertise in areas like data governance or specific technologies as needed with necessary subject matter experts • Design the end-to-end platform, define integrations, and set technical standards through Data Platform Architects • Build, test, and deploy the data platform and its supported agents, from foundational to end-user applications through Data Platform & Application Owners

Outputs and Deliverables

- ☐ Agentic Data Platform Transformation Strategy and Plan
- ☐ Agentic Data Platform Design and Framework
- ☐ Agentic Data Platform Implementation: Domain-based, specialized ‘Data Agents’ development and Semantic layer/views
- ☐ Conversationals Analytics Solutions

- [ ] Data-centric Business Agents for prioritized use cases
- [ ] App-centric Business Agents for prioritized use cases
- [ ] Training and enablement materials

Security and Risk Management

This framework aims to embed security, governance, and risk management as the default posture for developing and deploying Agentic AI systems. It builds upon established principles like Google's Secure AI Framework (SAIF) to address the unique risks—such as emergent behaviors, tool misuse, and greater autonomy—introduced by AI agents.

Contributors: Tyler Dooskin, Garrett Wong, Barbara Kenyon, Olu Akinrolabu, Steve Kluger, Lorenzo Caggioni, Sri Vasudevan

Objectives

- **Extended Risk Management to Support Agentic AI**
  - Define the collaboration model between humans and AI agents.
  - Embed AI Ethics principles to ensure agents operate fairly and transparently.
  - Adapt existing AI/ML risk frameworks to address the unique threats posed by autonomous agents, defining a clear risk appetite and taxonomy for different levels of agent autonomy.
  - Implement agent risk assessment and develop mitigations plans for agent-specific risks.
  - Obtain board-level approval of agent risk management strategy.
- **Embed 'Security by Design' into the Agent Development Lifecycle**
  - Integrate security into every phase of agent development, from architecture and design to deployment and decommissioning, enforcing secure multi-agent communication patterns (Zero Trust) and granular tool governance.
  - Measure agents passing pre-deployment security gates, reduction in security vulnerabilities, and time-to-remediate for identified issues
- **Protect Sensitive Data and Ensure Compliance**
  - Implement robust controls to protect sensitive data accessed, processed, and generated by AI agents, ensuring adherence to data privacy regulations, residency requirements, and industry standards.
  - Measure compliance adherence rate (e.g. GDPR, HIPAA) in a systematic manner aligned with reporting standards
- **Establish Continuous Monitoring and Response**

- Implement comprehensive observability, monitoring, and vulnerability management to detect, respond to, and report on security threats and anomalous agent behavior in real-time.
- Measure and improve Mean Time to Detect (MTTD) and Mean Time to Respond (MTTR) for agent-related security incidents.

Inputs

Business	Technical
• Board-level presentations on AI security vision & corporate strategy • Corporate / business unit strategic objectives and existing strategies • Internal pain point analysis and operational challenges • List of relevant compliance standards (e.g., GDPR, HIPAA) • Any Responsible AI / AI Ethics outputs	• Existing security landscape and data readiness assessments • List of security tools in use (e.g., Security Command Center, Wiz, Snyk) • List of existing security policies and requirements (SDLC, Data, Infra) • Available agentic architecture documentation and designs

Key Activities

1. **Key Activity:** Risk Framework Adaptation & Gap Analysis
    a. **Motivation/Goal:** To evolve the organization's existing risk management practices to effectively govern the novel risks introduced by autonomous AI agents.
    b. Key Steps:
        i. Audit existing AI/ML risk frameworks and governance processes.
        ii. Conduct a gap analysis against agentic risks (e.g., emergent behavior, prompt injection, tool misuse, data poisoning, fairness, transparency, accountability, and value alignment).
        iii. Define a clear risk appetite and a new risk taxonomy that categorizes agents based on their potential impact (financial, operational, reputational, ethical, and societal).
    c. **Tools/Techniques:** Google's Secure AI Framework (SAIF) as a guiding resource; Responsible AI Toolkit, Facilitated workshops with security, legal, and business leaders; Security Command Center to identify existing vulnerabilities.
2. **Key Activity:** Integrate Security into the Agent Lifecycle (Secure by Design)

a. **Motivation/Goal:** To proactively build security into agentic systems from the ground up, minimizing vulnerabilities and reducing the cost of remediation.
b. **Key Steps:**
    i. Define secure architecture patterns for single and multi-agent systems, enforcing Zero Trust principles for agent-to-agent and agent-to-tool communication.
    ii. Integrate adversarial testing and security evaluations into the CI/CD pipeline.
    iii. Implement automated, immutable deployment processes with clear environment segregation (Dev, Test, Prod).
    iv. Establish secure agent decommissioning and data retention policies.
c. **Tools/Techniques:** Cloud Build for CI/CD automation; Model Armor and Gemini built-in safety controls for model security; Vertex Explainable AI for transparency.

3. **Key Activity:** Implement Data & Tool Governance Controls
    a. **Motivation/Goal:** To ensure agents access data and execute actions in a controlled, secure, and compliant manner, enforcing the principle of least privilege.
    b. **Key Steps:**
        i. Implement a unified data classification scheme across all data sources.
        ii. Enforce granular, context-aware access controls using IAM for both agents and the identities they assume.
        iii. Integrate Sensitive Data Protection to discover, classify, and redact sensitive information in prompts and responses.
        iv. Establish a governed Tool Registry to manage and secure agent access to APIs and external systems.
    c. **Tools/Techniques:** Sensitive Data Protection (DLP); Cloud IAM (including Workload/Workforce Identity Federation); Cloud KMS for encryption; Dataplex for unified data governance.
4. **Key Activity:** Establish Continuous Monitoring & Automated Response
    a. **Motivation/Goal:** To maintain real-time visibility into agent behavior, enabling rapid detection of and response to security threats, anomalies, and exploits.
    b. **Key Steps:**
        i. Establish comprehensive observability with detailed audit logs and traces for all agent actions.
        ii. Implement continuous monitoring of agent runtime behavior for anomalies.
        iii. Enforce supply chain security for all agent components (code, models, data).
        iv. Automate the detection and remediation of misconfigurations in agent infrastructure.

c. **Tools/Techniques:** Cloud Audit Logs & Cloud Logging; Security Command Center; Google Cloud's Generative AI Indemnified Services.

5. **Key Activity:** Establish a Security and Risk Governance Processes
    a. **Motivation/Goal:** To build a robust governance process / cadence to find, handle, and reduce risks for agentic AI from start to finish
    b. Key Steps:
        i. Establish steering committee reviews to review progress, risk posture, and alignment with business objectives.
        ii. Establish a risk framework review & approval checkpoint to approve the updated Agentic AI Risk Management Framework before enterprise-wide adoption.
        iii. Establish a pre-deployment security gate to serve as a mandatory review to ensure new agents meet all security, compliance, and ethical requirements before production rollout.
        iv. Establish a quarterly security posture review to review monitoring data, recent incidents, and the overall effectiveness of security controls.

Roles and Responsibilities

Customer Responsibilities	Google Responsibilities
• Champions the security initiative, secures resources, and removes organizational roadblocks through C-level sponsorship • Provides strategic oversight, makes key decisions on risk appetite, and reviews progress through steering committees • Provide domain expertise (e.g., AppSec, InfraSec, Privacy, Compliance) to define controls through Security Leads & SMEs	• Provides expert guidance on agentic security best practices, risk mitigation, and tool implementation through a security consultant

Outputs and Deliverables

- [ ] Agentic AI Risk Management Framework

- [ ] Secure Agent Development Lifecycle (SADL) Guidelines
- [ ] Agent Data Governance & Security Policy
- [ ] Agent Security Monitoring Plan
- [ ] Security Dashboards

Governance and Operations

Contributors: Matt Castille, Christine Goglia, Patrick Nestler, Dovile Mikalauskaite, Carolyn Ujcic, Sarah Murphy Gray, Barbara Kenyon, Steve Kluger, Amulya Saridey

Objectives

- **Establish Enterprise-wide Standards and Principles for Agentic Governance**
    - Align Responsible and Ethical AI standards with business and organizational imperatives
    - Implement a cohesive framework that promotes company-wide adoption
        - Key metrics for compliance & risk include:
            - Regulatory Compliance Audit Findings, Audit Remediation Rate, Identified Risk Reduction, Data Privacy Incident Rate, Data Minimization Compliance.
- **Implement Robust Governance and Operations that Evolve and Scale as Technology Matures**
    - Promote ways of working that transcend evolving technology and allow teams to focus on productive work
    - Establish a balance of centralized and decentralized decisions to move quickly, safely, and with accountability
        - Key metrics for agility & accountability:
            - Decision Velocity, Agent Deployment Lead Time, Bureaucracy Index, Human Intervention Rate, Escalation Path Efficiency, Accountability Clarity Score
        - Key metrics for performance & value:
            - Business Objective Contribution, Agent Utilization Rate, Agent User Satisfaction (CSAT/NPS), Agent Performance Baseline Deviance, Feedback Loop Effectiveness

Inputs

Business	Technical

• Board level presentations on AI vision and existing corporate strategy • Outputs from the "Strategy and Value" phase (Vision, Prioritized Initiatives, Roadmap, OKRs) • Existing governance processes for technology initiatives across the organization	• Documentation relating to AgentOps foundational capabilities

Key Activities

1. **Key activity:** Discover Current Governance Landscape
    a. **Motivation/goal:** To understand the current state of AI governance, identify critical gaps in ethical and oversight mechanisms, and establish a baseline for designing the new framework.
    b. Key Steps:
        i. Assess the current AI Governance landscape, including existing governing bodies, principles, and RACIs.
        ii. Identify gaps in ethical, accountability, and oversight mechanisms.
        iii. Review existing capabilities for agent observability, security, and model testing.
        iv. Review relevant assets related to risk management, performance metrics, incident response, and agent lifecycle management.
    c. **Tools/Techniques:** Gap Analysis Frameworks, Stakeholder Interviews, RACI Charting, Existing Documentation Review.
2. **Key activity:** Design a Customer-Led Governance Framework
    a. Motivation/goal: To establish a tailored governance framework, owned and driven by the organization, that enables teams to innovate with Agentic AI quickly, safely, and in alignment with enterprise standards.
    b. Key Steps:
        i. Identify a sponsor for AI Ethics in the organization to help inform the below process.
        ii. Define the organization's core Agentic Governance Principles for regulatory and ethical use.
        iii. Determine which capabilities and decisions should be centralized for consistency and which should be decentralized for agility.
        iv. Develop the Agentic governance model and define clear roles, responsibilities, and decision-making authority within it.
        v. Set organizational standards for agent transparency and explainability.

        1. Please note that while there are existing AI governance standards for transparency and explainability, these will likely need to be added to allow for the new agentic capabilities
    vi. Design critical Agentic process flows, such as escalation paths for high-priority issues and standardized testing practices.
    vii. Collaborate with other workstreams to design governance artifacts and templates for teams to incorporate to adhere to responsible governance
  c. **Tools/techniques used:** Governance Model Templates and Workshops
3. **Key activity:** Establish a KPI Framework for Value and Effectiveness Tracking
  a. Motivation/goal: To create a data-driven system for measuring the health, impact, and value of the Agentic AI governance framework, ensuring accountability and providing a clear basis for continuous improvement.
  b. Key Steps:
    i. Identify and Select Balanced KPIs in collaboration with stakeholders across key domains like Business Value, Risk & Compliance, and Operational Efficiency.
    ii. Define KPI Specifications for each KPI, including the calculation method, data source, collection frequency, and owner to ensure consistent measurement.
    iii. Implement Data Collection and Reporting systems to automatically collect KPI data and establish a cadence for reviewing these metrics in governance meetings.
    iv. Create a Feedback Loop for Action to use insights from KPI analysis to drive improvements in the governance process and agent performance.
  c. **Tools/techniques:** BI & Dashboarding tools, Surveys, Feedback Platforms etc.
4. **Key activity:** Launch and Operationalize the Governance Framework
  a. **Motivation/goal:** To implement the governance framework in a controlled manner by piloting it on initial initiatives and embedding formal decision-making points (stage gates) to ensure ongoing alignment, quality, and risk mitigation.
  b. Key Steps:
    i. Appoint and Socialize: Declare and appoint the dedicated roles for agentic oversight, then socialize the new processes and responsibilities with relevant teams.
    ii. Pilot the Framework: Execute a pilot program on a limited set of high-impact Agentic AI initiatives to test the framework in a practical setting.
    iii. Establish and Execute Stage-Gate Reviews: Operationalize critical checkpoints throughout the agent lifecycle, including strategy alignment, ethical design review, process flow approval, and observability requirements sign-off.

iv. Refine and Improve: Establish a formal feedback loop to analyze pilot outcomes and user feedback, using the insights to refine the governance framework and its processes continuously.

c. **Tools/techniques:** Steering Committees and Governance Boards, Pilot Programs and Change Management Best Practices, Stage-Gate Models with Go/No-Go Checklists, Risk Assessment Frameworks & Communication Plans and Feedback Surveys

Roles and Responsibilities

Customer Responsibilities	Google Responsibilities
• Aligns agentic AI governance with the company's technical strategy, architectural standards, and long-term vision through an enterprise architect • Manages risk, security, and compliance for agentic systems, establishing essential organizational and data protection guardrails through a Security VP/Director • Champions the governance framework, ensuring it accelerates innovation rather than hindering it through a Director of Digital Transformation or equivalent • Represents business unit needs, priorities, and functional requirements for agentic solutions through a Director of Enterprise Business Solutions or equivalent • Ensures cloud and on-premise infrastructure reliably and cost-effectively supports agentic system performance, scalability, and operational demands through a VP of Infrastructure • Leads daily execution of the governance framework project, ensuring on-time, on-budget delivery with clear stakeholder communication through a Project Manager	• Responsible for overall oversight of governance framework design, cross-Google workstream collaboration, stakeholder management, and implementation roadmap from an Innovation Consultant • Provide technical input, direction, and guidance for current state assessment, process design, and best practices through Security, Data, and AI/ML Engineers • Provide consistency across workstreams to point out any gaps and provide strategic technical direction through an Enterprise Architect

• Advocates for end-users, ensuring agent interactions and governance are transparent, trustworthy, and effective through a Digital Experience Lead or equivalent .	

Outputs and Deliverables

- ☐ Agentic AI Governance Framework Document
- ☐ Defined Roles & Responsibilities (RACI) for the Agentic AI governing body and related activities
- ☐ Defined stage gates for critical decision-making in the Agentic AI lifecycle
- ☐ Templates and assets for teams to ensure adherence to governance (e.g., testing plan, risk assessment checklist).
- ☐ A defined list of agent observability and monitoring requirements.
- ☐ A KPI dashboard specification for tracking governance effectiveness and value realization.

Platforms and AgentOps Framework

Contributors: Anna Whelan , Atulan Zaman , Dovile Mikalauskaite , Itziar Leguinazabal , Javier Garcia Puga , Minwoo Jeon , Semih Duru , Ramya Gowrugolla , Amulya Saridey , Shallom Akporuku

Objectives

- **Establish a dedicated platform strategy and robust AgentOps to manage operational complexity and scale agentic AI responsibly.**
- **Establish foundational capabilities for centralized oversight and decentralized empowerment, enabling teams to innovate rapidly yet responsibly.**
    - Key Metrics (Core Agent Performance): Task Completion Rate, Accuracy (Response & Intent Recognition), User Satisfaction (e.g., Thumbs Up/Down, CSAT), Escalation Rate / Containment Rate.
- **Integrate robust governance, security, and compliance directly into the development lifecycle.**
    - Key Metrics (AI Agent Safety & Quality): Guardrail Effectiveness / Violation Rate, PII Redaction/Handling Accuracy, Non-Answer Rate (for blocked/unsafe queries), Completeness, Clarity & Coherence, Factuality / Grounding.
- **Enhance the Developer Experience (DevEx) to accelerate innovation.**

- **Adopt a platform oriented approach to understanding total cost of ownership for agentic solutions across AI applications, agent run times, model usage, tool execution and data sources; including a methodology for cost optimization strategies**

Inputs

Business	Technical
• Existing governance, risk, and compliance (GRC) policies • Strategic objectives and OKRs for AI adoption • Defined operational pain points and challenges • Security and data privacy requirements • Developer skill-set and capability assessments • Budgetary constraints for cloud services • Aggregated employee and customer usage statistics • Internal change management capacity	• Current technology stack and infrastructure assessments • Existing CI/CD pipeline definitions and practices • Inventory and performance of existing agentic applications • Data readiness and availability assessments

Key Activities

1. **Key activity:** Develop Automated Deployment and CI/CD Workflows
    a. Motivation/goal: To establish automated, repeatable, and scalable "golden paths" for deploying AI agents, abstracting away infrastructure complexity and allowing teams to focus on agent development.
    b. Key steps:
        i. Define primary deployment runtimes (e.g., managed service vs. serverless container) to cater to different operational needs.
        ii. Implement Infrastructure as Code (IaC) to define and provision all necessary cloud resources, ensuring environmental consistency.

iii. Configure a CI/CD pipeline to automate the build, test, and deployment of agent updates upon code commits.
iv. Establish post-deployment operational procedures, including criteria for production readiness and automated quality checks.

c. Tools/techniques:

i. Runtimes: Vertex AI Agent Engine, Google Cloud Run, Google Kubernetes Engine (GKE)
ii. Automation: Terraform (for IaC), Cloud Build (for CI/CD)
iii. Storage: Artifact Registry

2. **Key activity:** Establish Comprehensive Observability and Monitoring
   a. Motivation/goal: To gain deep insights into agent behavior, decision-making, and performance, enabling reliable management, debugging, and optimization of autonomous systems in production.
   b. Key steps:
      i. Implement distributed tracing to visualize the end-to-end flow of agent tasks, including calls to LLMs and external tools.
      ii. Configure structured logging to capture detailed execution data, including prompts, responses, tool invocations, and errors.
      iii. Define and track key performance metrics (e.g., latency, task completion rate, resource usage) to monitor agent health.
      iv. Assign clear roles and responsibilities for monitoring, incident response, and performance analysis.
   c. Tools/techniques:
      i. GCP Services: Cloud Trace, Cloud Logging, Cloud Monitoring
      ii. Frameworks: Agent Development Kit (ADK), Vertex AI Agent Engine (which integrates with the above services)
3. **Key activity:** Implement a Multi-Stage Evaluation Framework
   a. **Motivation/goal:** To ensure agent quality, safety, and effectiveness throughout the entire lifecycle, from initial development and testing to continuous improvement in production.
   b. Key steps:
      i. Develop a quantitative evaluation process using "golden datasets" and automated metrics to assess agent performance against ideal outputs.
      ii. Establish a User Acceptance Testing (UAT) process for structured human feedback and adversarial testing.
      iii. Deploy a continuous user feedback mechanism (e.g., thumbs up/down, comments) to capture real-world performance data.
      iv. Create a feedback loop to convert evaluation results and user feedback into new test cases and improvements for the agent.
   c. Tools/techniques:

i. Evaluation Services: Vertex AI Gen AI Evaluation Service, Agentspace Built-in Analytics
ii. Automation & Data Management: Colab Notebooks, BigQuery, Google Sheets
iii. Observability for Evaluation: Cloud Logging, Cloud Monitoring, Cloud Trace
iv. Frameworks: Agent Development Kit (ADK)

4. **Key activity:** Operationalize Cost Management and Optimization Strategies
    a. Motivation/goal: To ensure the economic viability and sustainable scaling of agentic systems by integrating cost-aware practices throughout the design, development, and operational lifecycle.
    b. Key steps:
        i. Identify billable SKUs related to an Agentic solution and establish observability across different resources tied to use cases
        ii. Implement prompt and token optimization techniques to reduce the cost of each LLM call.
        iii. Develop a model "right-sizing" strategy, including dynamic routing to use the most cost-effective model for a given task.
        iv. Implement response caching for frequently repeated queries to reduce redundant API calls.
        v. Actively monitor token consumption and cloud resource usage to identify optimization opportunities.
        vi. Explore fine-tuning smaller, specialized models for high-volume tasks to reduce dependency on expensive, general-purpose models.
    c. Tools/techniques:
        i. GCP Tools: Vertex AI Model Optimizer, Committed Use Discounts (CUDs)
        ii. Techniques: Response Caching, Dynamic Model Routing, Request Batching
        iii. Open-Source: Exploration of open-source LLMs and fine-tuning frameworks
5. **Key activity:** Build a Governed Artifact Management Ecosystem
    a. **Motivation/goal:** To scale development securely and efficiently by creating a governed, accessible ecosystem for sharing and reusing core agentic components like models, agents, and tools.
    b. Key steps:
        i. Establish a Model Registry to govern the lifecycle of foundation models and their fine-tuned variants, ensuring clear lineage and adherence to enterprise standards.
        ii. Create an Agent Catalog as a curated repository for discovering, reusing, and governing versioned, production-ready agents.

iii. Implement a Tool Registry as a managed inventory of approved and secured APIs and functions that agents can use to interact with external systems.
iv. Deploy an A2A (Agent-to-Agent) Server to enable standardized communication and interoperability between different agents in a multi-agent system.

c. Tools/techniques:

i. Registries: Vertex AI Model Registry, Artifact Registry, AgentSpace (as an Agent Catalog)
ii. Service Hosting: Google Cloud Run, GKE
iii. API Management: API Gateway
iv. Protocols: Model Context Protocol (MCP)

Roles and Responsibilities

Customer Responsibilities	Google Responsibilities
• Owns, builds, and maintains the IDP, "golden paths," and core AgentOps infrastructure through a Platform Engineering Team • Build, test, and deploy agents using the established platform and "golden paths" through AI/ML Engineers & Developers • Manage and automate the CI/CD pipelines, observability, and deployment infrastructure through DevOps/MLOps Engineers • Define and audit the security policies, guardrails, and compliance checks embedded in the platform through Security & Compliance Teams • Define agent requirements and success criteria, translating them into measurable metrics through Product Managers or equivalent	• Provides business context, technical expertise on Agentic AI, and solution architecture guidance through an AI Architect • Guides on best practices for using Google Cloud services and implementing platform capabilities through a customer engineer • Offer specialized expertise as needed in areas like security, data, or a specific industry through SMEs • Guides overall strategic alignment, value quantification, and ensures realization of objectives through an Innovation Team Engagement Lead • Collaborates on innovating and co-creating novel agent-based solutions and MVPs through a team dedicated for innovation on agent-based solutions

Outputs and Deliverables

- [ ] A defined Internal Developer Platform (IDP) strategy and architecture document.
- [ ] Automated CI/CD pipelines for agent deployment to selected runtimes (e.g., Agent Engine, Cloud Run).
- [ ] An established observability dashboard in Cloud Monitoring for tracking agent health and performance.
- [ ] A comprehensive agent evaluation framework, including golden datasets, UAT guidelines, and a continuous feedback process.
- [ ] A cost optimization playbook for agentic applications with defined strategies and monitoring practices.
- [ ] A populated and governed Model Registry, Agent Catalog, and Tool Registry.

Conclusion

The journey from promising agentic pilots to a true autonomous enterprise is complex, filled with the risks of creating disconnected silos, accruing technical debt, and failing to realize enterprise-wide value. The Agentic AI Transformation Framework serves as the comprehensive blueprint to navigate this complexity. By integrating a clear vision grounded in Strategy and Value, an iterative Development Lifecycle for agile execution, and robust Horizontal & Foundational Capabilities for scale and security, this framework provides a structured, repeatable path.

Ultimately, this framework is your guide to starting smart and scaling strategically. The path forward is not a single leap but a measured journey, beginning with high-impact single-purpose agents and evolving toward sophisticated, collaborative multi-agent systems. By adopting this deliberate approach, your organization can confidently move beyond experimentation to implementation, mitigating risk while continuously building momentum. This is the roadmap to not only harness the power of agentic AI but to fundamentally reshape your business operations, building a more intelligent, resilient, and responsive enterprise ready for the future.

Appendix

Reimagining Business Processes

Optional Examples (for reimagining business processes)

Industry	Traditional Process	Reimagined with Agentic Transformation
Banking	**Reactive Fraud Investigation:** An analyst receives an alert for a suspicious transaction, manually pulls data from 3-4 systems, and then decides to block the card, often after the fraud has occurred	**Autonomous Financial Crime Prevention:** A network of agents continuously monitors transactions, customer behavior, and external threat intelligence in real-time. An agent can predict a likely fraud event, proactively engage the customer for verification via their preferred channel, and neutralize the threat *before* loss occurs, all while documenting the case for compliance.
Telco	**Network Fault Resolution:** A Network Operations Center engineer sees a performance degradation alarm, diagnoses the root cause across multiple network layers, and dispatches a field technician.	**Predictive & Self-Healing Networks:** An agentic system constantly ingests network telemetry. It predicts future congestion or hardware failure based on subtle patterns, autonomously re-routes traffic to optimize performance, and if a fault is unavoidable, it performs root-cause analysis, opens a ticket with the precise part and location, and schedules the technician, minimizing downtime.
Retail	**Static Inventory Replenishment:** A category manager uses historical sales data and a forecast model to manually set reorder points for thousands of SKUs, leading to stockouts of popular items and overstock of others.	**Autonomous Demand-Sensing Supply Chain:** A team of agents manages the flow of goods from supplier to shelf. One agent monitors real-time sales data, social media trends, and local events to predict demand shifts. It communicates this to a

		logistics agent, which optimizes shipping routes, and an inventory agent, which automatically adjusts stock levels and places orders, ensuring hyper-local product availability. Anomalies are reported to the category manager (human expert) for exceptions management

Examples:

General Cross-Industry Use Cases				
Use Case Name	**Business Challenge**	**Agentic Solution**	**Measurable Outcome**	**Agentic Core Capabilities**
Autonomous Sales Development Representative (SDR)	Sales teams spend over 60% of their time on non-revenue-generating activities like research and data entry instead of selling.	An agent monitors CRM data, news feeds, and social media to identify target accounts showing buying intent. It synthesizes this data, evaluates the prospect against an Ideal Customer Profile (ICP), generates a personalized outreach email, and uses a calendar API to plan and schedule the first meeting.	- Increase in Sales Qualified Leads (SQLs). - Reduction in time-to-first-contact. - Increased seller productivity.	Synthetize Evaluate Generate Plan
Proactive Project Coordinator	Project managers are bogged down chasing status updates and compiling reports instead of strategically managing risks and unblocking their teams.	The agent connects to various project management tools (e.g., Jira, Asana, Slack). It synthesizes progress across all platforms, evaluates project data to identify potential risks (e.g., timeline slips), and generates a daily executive summary, freeing the PM to focus on exceptions and strategy.	- Early detection of project risks. - Reduced time spent on manual status reporting. - Improved project delivery velocity.	Synthetize Evaluate Generate

Industry-Specific Use Cases

Use Case Name	Business Challenge	Agentic Solution	Measurable Outcome	Agentic Core Capabilities
(Financial Services) Financial Compliance Assistant	Investment advisors must adhere to strict, constantly changing regulatory rules (e.g., FINRA, SEC) in all client communications, creating significant compliance risk and overhead.	An agent works alongside the advisor, evaluating emails and chat messages in real-time against a knowledge base of regulations. It flags non-compliant language, generates compliant alternative phrasing, and automatically logs all communications in an audit-ready format for the compliance department.	- Reduction in compliance violations and potential fines. - Decreased time and cost of internal audits. - Increased advisor confidence and efficiency.	Evaluate Generate Transform
(Telecommunications) Customer Experience Troubleshooting Agent	Contact centers are flooded with calls about predictable issues, such as troubleshooting technical issues, leading to high operational costs and customer frustration.	The agent evaluates network monitoring data and user device data. Upon detecting a problem, it generates and sends a proactive notification, explaining the situation and executing troubleshooting steps like restarting the customer's modem remotely with the customer's consent.	- Reduction in inbound call volume for predictable events. - Improved Customer Satisfaction (CSAT) and Net Promoter Score (NPS). - Lower operational costs in the contact center.	Evaluate Generate Plan Transform

Driving Quality and Alignment

Evaluation Metrics

System & Task metrics: Assess agent performance on efficiency (least cost), effectiveness (correct path), reproducibility, reasoning, data accuracy, factual correctness, and alignment with ground truths.

- **Tool Utilization Efficacy (TUE)**: How effectively the agent selects and uses external tools/APIs.
- **Memory Coherence and Retrieval (MCR)**: The agent's ability to store, manage, and retrieve information from its memory.
- **Strategic Planning Index (SPI):** Measures the agent's ability to plan and execute strategies to achieve goals.
- **Component Synergy Score (CSS):** Evaluates the synergy and coordination between different components of the agentic AI system
- **Quality of Output**: Reduction of errors in final output; percentage of "first shot"

approvals.

- **Accuracy**: accuracy of the GenAI model in producing relevant and correct outputs such as completeness, and level of duplication.
- **Turnaround Time**: Turnaround Time- time for GenAI to generate output for process engineers vs traditional work

Safety Metrics: Does it follow AI Safety & Responsibility standards? Are its risks and impacts properly defined? In case of multi-modal agents, does it meet the goals and follow responsible AI principles? Does it follow security and interoperability protocols?

- **Harmful Content Generation Rate (%)**: Outputs flagged as toxic, biased, or hateful.
- **Bias Detection Rate (%):** Responses identified as exhibiting bias.
- **Privacy Compliance Score (%)**: Adherence to PII anonymization and data minimization rules.
- **Risk Classification Accuracy (%)**: For agents designed to identify risks (e.g., fraud, security threats), the accuracy of their classification.
- **Metric: Multi-Agent Guardrail Adherence Rate (%)**: Percentage of interactions where multi-agents comply with defined guardrails, ethical guidelines, and safety protocols.

Building Single and Multi-Agent Systems

Agentic Maturity Roadmap

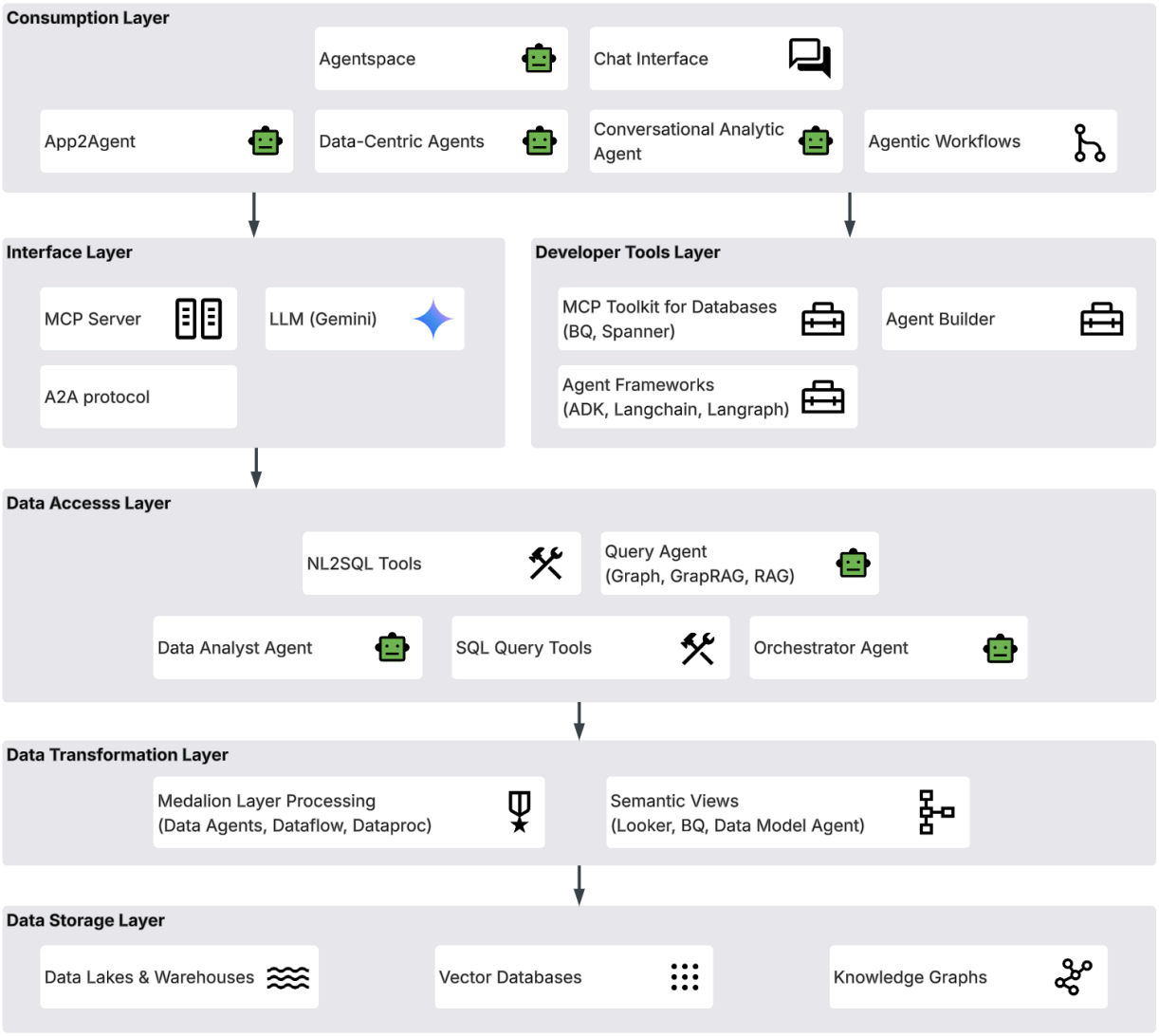
Multi-Agent Design Patterns

1. Coordinator / Dispatcher Pattern:
    - **Goal:** Route incoming requests to the appropriate specialist agent.

    - **Structure:** A central coordinator agent manages several specialized sub agents and hands off control as needed. The sub-agents can handoff back to the parent when ready.
2. Sequential Pipeline Pattern
    - **Goal**: Implement a multi-step process where the output of one step feeds into the next.
    - **Structure**: A collection of agents executed in a fixed order.
3. Parallel Fan-Out/Gather Pattern
    - **Goal:** Execute tasks that have no dependence on each other simultaneously to reduce processing time.
    - **Structure:** Run multiple agents concurrently and combine their results, either through concatenation, summary or some other aggregation approach.
4. Hierarchical Task Decomposition Pattern
    - **Goal:** Solve complex problems by recursively breaking them down into simpler, executable steps.
    - **Structure:** A multi-level tree of agents where higher-level agents break down complex goals and delegate sub-tasks to lower-level agents.
5. Review / Critique Pattern (Generator-Critic)
    - **Goal:** Improve the quality or validity of generated output by having a dedicated agent review it.
    - **Structure:** Typically involves two agents executed sequentially, first initial generation followed by a review or critique to improve the quality of the output.
6. Iterative Refinement Pattern
    - **Goal:** Progressively improve a result (e.g., code, text, plan) stored in the session state until a quality threshold is met or a maximum number of iterations is reached.
    - **Structure:** Run one or more agents in a loop that work on a task over multiple iterations. A break condition is required to complete execution.
7. Human-in-the-Loop Pattern
    - **Goal:** Allow for human oversight, approval, correction, or tasks that AI cannot perform.
    - **Structure:** Integrates human intervention points within an agent workflow, often through a tool that calls an API to request human input. The request may either be (1) asynchronous to allow the user to continue interacting or (2) synchronously wait for a response if no progress can be made without input.

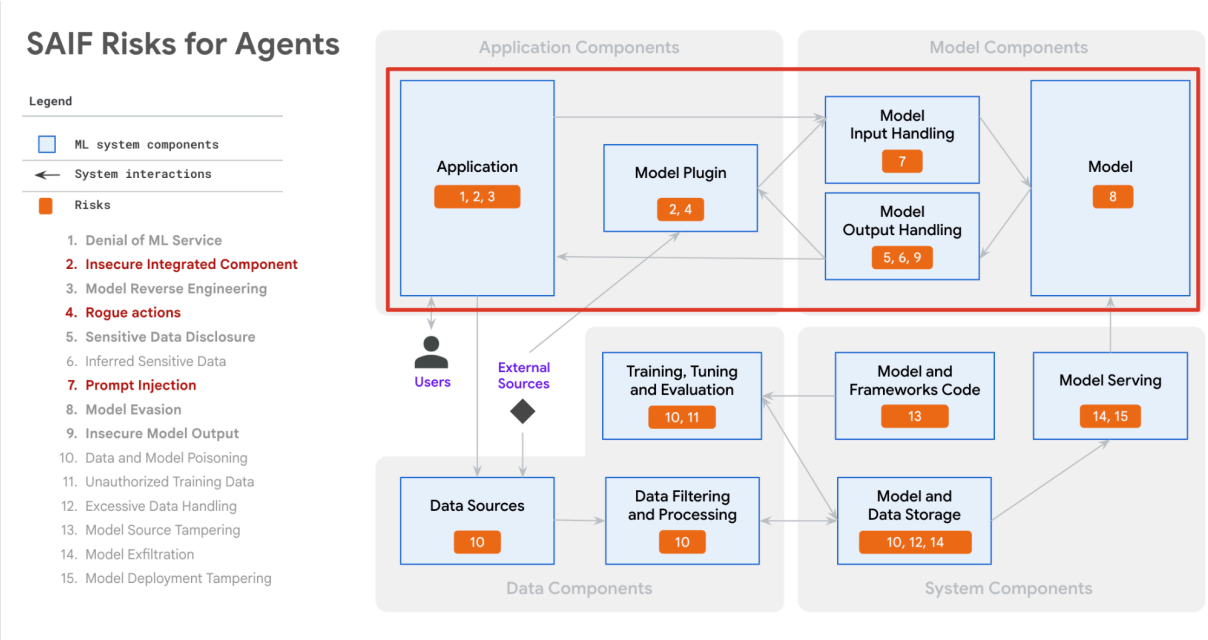
Data and Data Platforms

Agentic Data Platform Design Blueprint

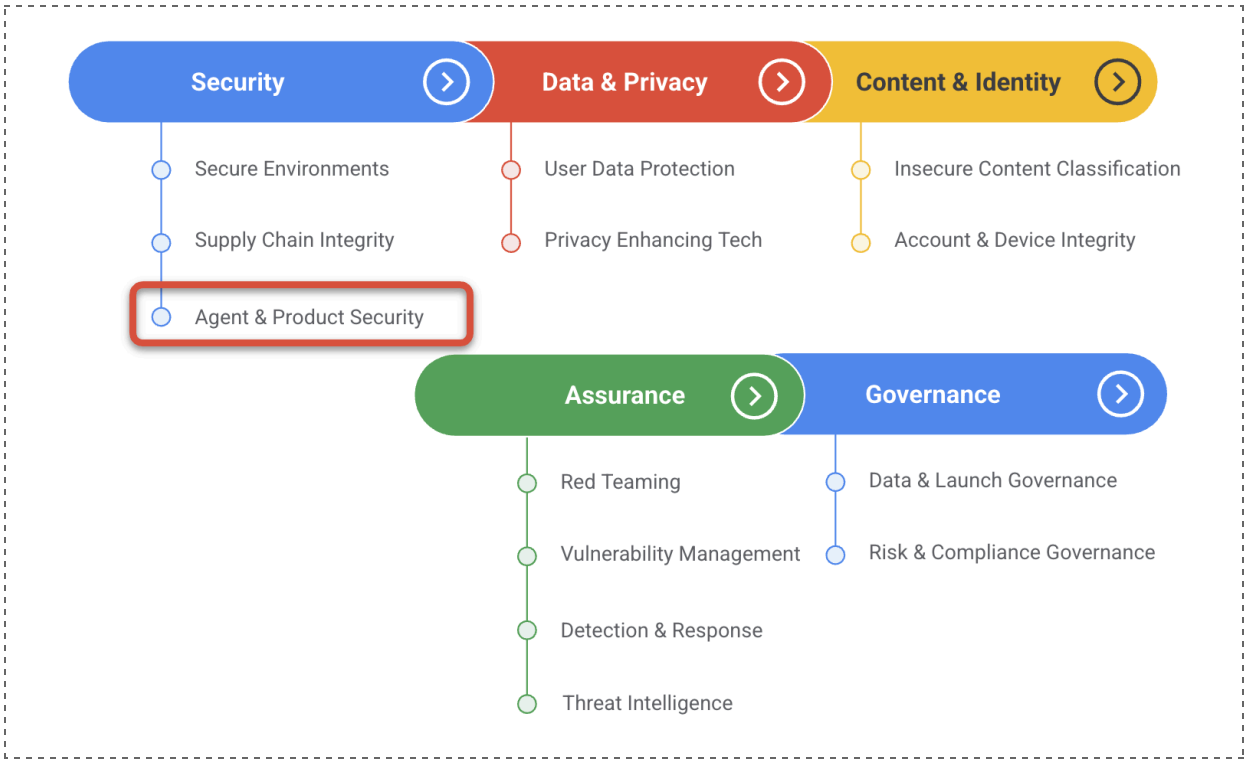


Security and Risk Management

SAIF Risks for Agents



Google's Secure AI Framework Pillars



Considerations for Governance KPIs

Metrics should cover various aspects of governance and its impact and can be collected both quantitatively and qualitatively. Some considerations when identifying KPIs are:

- Business and Organizational Imperative: Business Objective Contribution - quantify the measurable contribution to cost savings, revenue, csat and other business objectives), Feedback Loop Effectiveness - % of end user feedback that is incorporated into the Agentic solution
- Sensitive Data Handling: Data Privacy Incident Rate - # of unauthorized access or exposure, Data Minimization Compliance - % adherence to minimum data needs,
- Human Oversight: Human Intervention Rate - frequency and nature of interventions in Agentic solutions, Accountability Clarity Score - stakeholder clarity via survey or workshops for Agentic system performance, Escalation Path Efficiency - average time to resolution when human intervention is required
- Regulatory Compliance: Audit Remediation Rate % - within a timeframe, # of Regulatory Compliance Audit Findings related to Agentic AI solutions, Identified Risk Reduction - decrease in # of risks and/or severity of risks
- Agility and Accountability: Bureaucracy Index (Qualitative and Quantitative - feedback on 1 to 5 scale and/or commentary). Decision Velocity - Average time from identification to resolution, Agent Deployment Lead Time - concept to deployment,
- Performance and Value Realization: Agent Utilization Rate - user adoption rate of Agents or Agent solutions, Agent User Satisfaction - user satisfaction scores like CSAT or NPS - related to user experience with an Agent or Agent Solution, Agent Performance Baseline Deviance - monitor deviations from expected performance baselines for key Agentic properties

Considerations on Governance Checkpoints

- Agentic Vision & Strategy Alignment Review: This checkpoint, early in the "Discover" phase, ensures that proposed agentic AI initiatives directly align with the organization's overarching business and AI strategies.
- Ethical, Accountability, and Oversight Mechanism Gap Analysis & Design Review: This critical checkpoint spans the "Discover" and "Design" phases. It involves assessing the "current AI Governance Landscape" for gaps in ethical, accountability, and oversight mechanisms, and then reviewing the proposed "Agentic Governance Principles" and the defined "clear roles, responsibilities and decision making authority within the governance framework."
- Process Flow & Stage Gate Definition Approval: During the "Design" phase, a checkpoint is needed to approve the "critical Agentic process flows" (e.g., escalation paths, testing practices) and the "Stage gates" for critical decision-making in the Agentic AI lifecycle.

- Framework Pilot & Refinement Review: In the "Launch" phase, a crucial checkpoint is to review the outcomes of piloting the governance framework on a "limited set of Agentic AI initiatives."
- Agent Observability & Monitoring Requirements Sign-off: This checkpoint, occurring as part of "Outputs and Deliverables," focuses on ensuring that the "Document Specifications for agent logging, audit trails, performance monitoring, and XAI capabilities" are robust and meet the organization's needs for transparency, explainability, and risk mitigation.
- Regular Governance Effectiveness Review (KPI-driven): An ongoing checkpoint, outlined in "Metrics and KPIs," involves regular reviews of the established KPIs to track progress in the AI governance framework. This includes assessing "Efficiency," "Adoption," "Continuous Improvement," and "Risk Mitigation Effectiveness."

Platforms and AgentOps

Comparison of Run-times

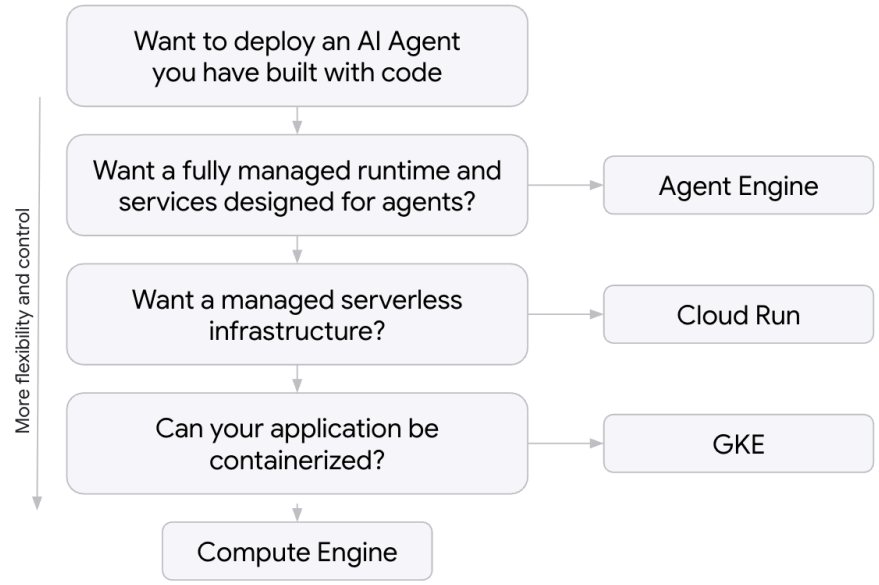
There are many different ways to deploy agentic solutions, and the following table summarizes key considerations and factors among Google Cloud Products.

	Agent Engine	Cloud Run	GKE
Considerations	Agent Engine is designed to streamline the deployment and management of AI agents. Easy to start, easy to deploy, purpose built SDK and console for agent builders	There are multiple ways to deploy to Cloud Run, incl. from source, using a buildpack, or using containers. **Deploying from source is the most similar to Agent Engine, so we'll use that here**. With this serverless option and when deploying from source, developers don't need to worry about servers, but they still need to build their agent manually and wrap their agent in a web framework within the Cloud Run service.	This option lies in between the VM instance approach and serverless approach with Cloud Run. While there are no individual VM instances to manage, there is still significant work needed to provision and maintain the Kubernetes cluster, build and package up your agent with LangChain, and handle cluster updates and monitoring. Intelligently optimized LLM aware routing through GKE Inference Gateway and dynamic LoRA adapter serving for Agents
Target Persona/s	Developers	Developers	Platform
Deployment Steps	1. Deploy agent using Vertex AI SDK:	1. Implement API wrapper (e.g., in Flask or FastAPI)	1. Create a GKE cluster 2. Create Dockerfile

	`agent_engines.create()`	2. Deploy from source	3. Store source code in a repository 4. Add API server 5. Build image 6. Upload image to registry 7. Write container specs for scaling, auth, etc. (i.e. define a Kubernetes resource file) 8. Deploy image 9. Handle service ingress and routing
Observability	Supported by framework and made easy in console	Integrated with Cloud Observability, Prometheus, OpenTelemetry Supported by chosen Frameworks	Integrated with Cloud Observability, Prometheus, OpenTelemetry Supported by chosen Frameworks
Frameworks Supported	Eng managed templates for each framework (ADK, LangGraph, AG2, etc)	DIY Containerized Services gRPC/HTTPS protocol support	DIY Containerized Services gRPC/HTTPS protocol support
MCP/A2A	WIP but will be supported	Supported	Supported
Context Management	Sessions: Agent Engine Sessions lets you store individual interactions between users and agents, providing definitive sources for conversation context. Bundled 1st party support for these services	Can integrate with various services and systems Available a la cart from Agent Engine or build your own	Available a la cart from Agent Engine or build your own
Quality & Evaluation	Ensure agent quality with the integrated Gen AI Evaluation service. Improve agent performance with Example Store: Example Store lets you store and dynamically retrieve few-shot examples.		

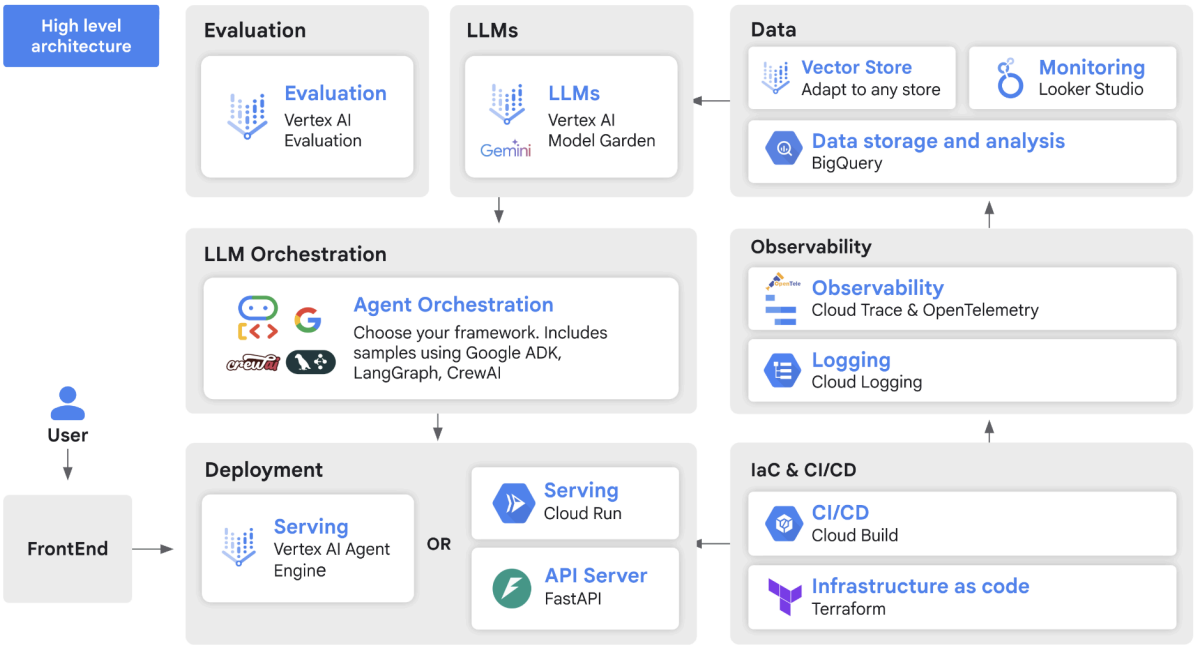
	Optimize agents with Gemini model training runs.		
Dependencies	Handled as part of Agent Engine deployment step (specify package versions or requirements.txt)	Handled in requirements.txt when deploying from source	Handled within Dockerfile steps
Reference(s)	• Vertex AI Agent Engine overview | Generative AI on Vertex AI | Google Cloud • Deploy an agent | Generative AI on Vertex AI | Google Cloud • Vertex AI framework support policy | Google Cloud	• https://github.com/GoogleCloudPlatform/cloud-build-samples/tree/main/python-example-flask • Deploying to Cloud Run using Cloud Build | Cloud Build Documentation | Google Cloud • Deploying container images to Cloud Run • Deploy from source code | Cloud Run Documentation	• https://github.com/GoogleCloudPlatform/cloud-build-samples/tree/main/python-example-flask • Deploying to GKE | Cloud Build Documentation

Agent Building Decision Tree



CI/CD

This architecture highlights an automated approach to deploying AI agents on Google Cloud. The design provides flexibility in serving the agent while ensuring the entire deployment lifecycle is repeatable, scalable, and manageable through modern DevOps practices.



Core Deployment Choices:

The framework offers two primary paths for deploying the agent, catering to different needs:

- **Vertex AI Agent Engine**: A managed, purpose-built platform from Google designed to host and scale agents. This option simplifies operations by abstracting away the underlying infrastructure, allowing teams to focus on agent development.
- **API Server on Cloud Run**: A flexible, serverless approach. Developers can package their agent (e.g., using FastAPI) into a container and deploy it on Cloud Run. This provides more control over the runtime environment and is ideal for custom applications.

Automated and Repeatable Deployments:

The deployment process is not manual but is driven by a robust automation framework:

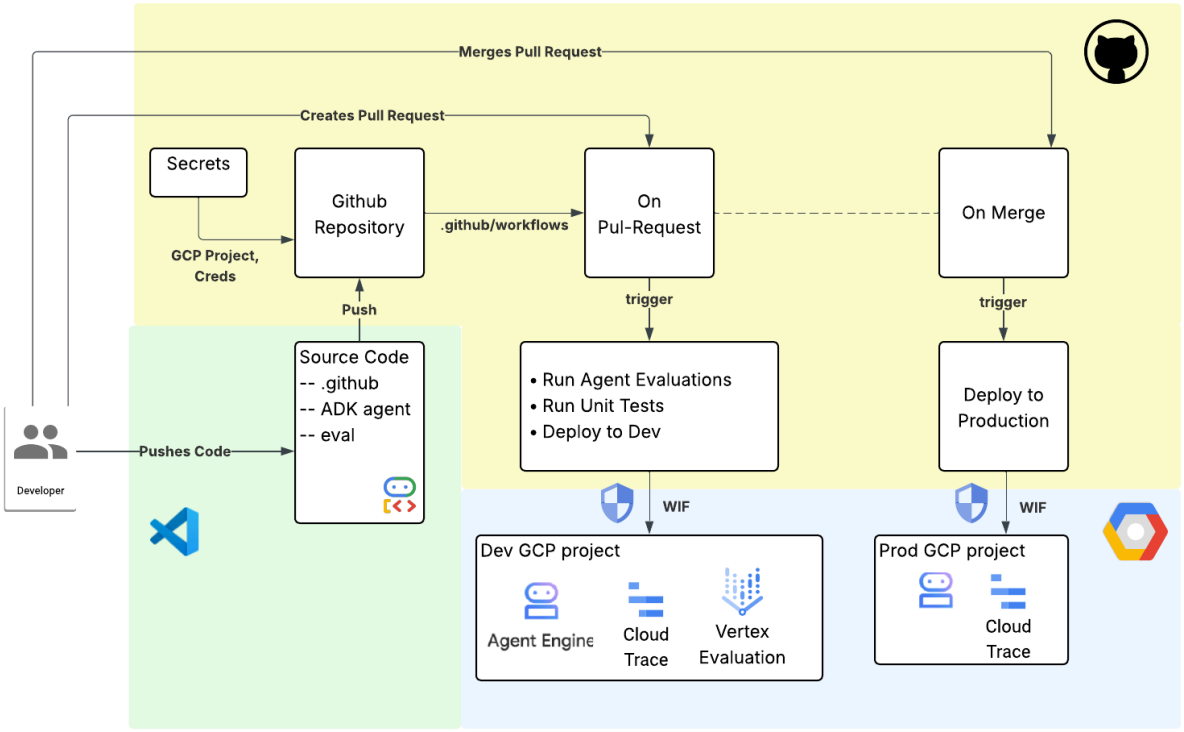
- **Infrastructure as Code (IaC)**: Terraform is used to define and provision all necessary Google Cloud resources. This includes the Cloud Run service, networking rules, IAM permissions, and connections to other services like BigQuery or Logging. This ensures the environment is consistent across development, staging, and production.
- **Continuous Integration & Deployment (CI/CD)**: Cloud Build acts as the automated pipeline. When new code is committed, Cloud Build automatically builds the container image, runs tests, and deploys the new version to either Cloud Run or the Vertex AI Agent Engine with no manual intervention.

Post-Deployment Operations:

Once the agent is live, the architecture provides essential tools to ensure it runs effectively:

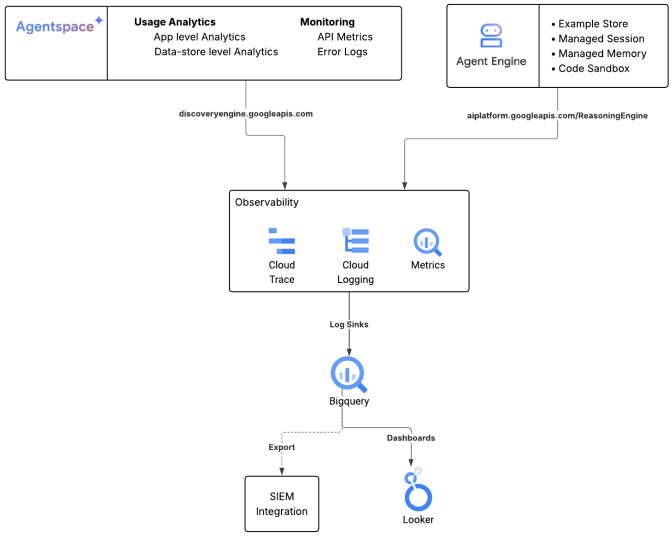
- **Observability**: Cloud Logging and Cloud Trace are critical for monitoring the health of the deployed agent. They provide immediate insight into errors, performance bottlenecks, and usage patterns in the production environment.
- **Evaluation**: Vertex AI Evaluation is used to monitor the *quality* of the deployed agent's outputs, ensuring the deployment meets business objectives and user expectations. This closes the loop by providing data to inform the next development and deployment cycle.

The CI/CD template can be generally applied to any Agent deployment system for various runtimes:



In the context of AI agents, observability is essential for:

- **Understanding agent decision-making**: Analyzing how agents make choices and interact with their environment.
- **Evaluating agent performance**: Assessing the accuracy, efficiency, and resource usage of agents.
- **Debugging agent failures**: Identifying the root cause of errors and improving agent reliability.
- **Optimizing agent workflows**: Identifying areas for improvement in agent interactions and decision-making processes.



Observability	Role in the context of AI Agents	Google Cloud Service
Tracing	Visualize the end-to-end flow of a task handled by the AI agent. This includes all the intermediate steps (known as *spans*), such as calls to large language models (LLMs), interactions with external APIs (tools), and internal "thought" processes	Cloud Trace, Vertex AI Agent Engine, Agent Development Kit
Logging	Capture detailed execution logs from the AI agent's code, including prompts sent to language models, responses received, tool invocations, and any errors encountered	Cloud Logging, Vertex AI Agent Engine, Agent Development Kit
Metrics	Track key performance indicators (KPIs) of the AI agent, such as the number of successful task completions, the latency of thought-action-observation loops, or the frequency of specific tool usage	Cloud Monitoring, Vertex AI Agent Engine, Agent Development Kit

	Generic Observability Tools			Focused AI Agents Tools	
	Cloud Trace	Cloud Logging	Cloud Monitoring	Agent Development Kit	Vertex AI Agent Engine

Description	Distributed tracing system for Google Cloud that collects latency data from applications and displays it in near real-time in the Google Cloud console	Store, search, analyze, and alert on log data and events from applications and Google Cloud services	Collection of metrics, events, and metadata from Google Cloud services, applications, and systems to provide dashboards and alerts for performance tracking and issue identification	Open-source framework designed to simplify the creation, evaluation, and deployment of sophisticated, multi-agent AI systems that can reason and interact with various tools.	Fully managed, scalable runtime environment on Google Cloud designed to facilitate the deployment, orchestration, and lifecycle management of sophisticated AI agents for enterprise-grade application
Data	Traces	Logs	Metrics	Events	Traces, Logs, Metrics
Features for AI Agents	Trajectory of execution, requests and responses from reasoning loop, execution time of steps	Error and debugging logs, raw form of requests and responses, debugging information	Query metrics such as Query Per Second, Query Latencies, Query Error ratio. Session metrics such as CPU and Memory allocation	Capabilities to observe traces and trajectories during development in “adk web”. Similar traces will appear in the Cloud Trace for deployed environments.	Provides observability features through integration with Google Cloud Logging, Cloud Monitoring and Cloud Trace. • Use of Cloud Monitoring without any additional setup or configuration. Built-in agent metrics are automatically collected and visualized in Cloud Monitoring pages in the Google Cloud console.
Resources (Agent Engine Related)	Agent Engine Tracing	Agent Engine Logging	Agent Engine Monitoring		Monitoring in Agent Engine Full list of AI Platform metrics

Roles and Responsibilities for Observability

- **AI/ML Engineer (with Observability Focus)**: Design, implement, and maintain observability solutions for AI agents. This includes logging, tracing, and metrics to enable performance monitoring, debugging, and collaboration with other teams.
- **MLOps Engineer**: Build, maintain, and manage the CI/CD pipelines and infrastructure for deploy and update AI agents.
- **Data Scientist / AI Ethicist (with Evaluation Focus)**: Analyze observability data to identify patterns and biases, prioritize ethical considerations, fairness, and transparency in AI agent decisions.
- **DevOps Engineer (Supporting AI/ML)**: assist with cloud resource setup for observability, ensure compliance with standards, and collaborate on incident response for AI agent operations
- **Product Manager / Business Analyst (with AI Focus)**: Define business objectives and success criteria for AI agents, translate requirements into measurable metrics, monitor high-level business impacts, provide insights from observability data to guide strategy

Evaluation Stages	Evaluation of an agent in Agentspace with Connectors	ADK Agent Evaluation

Quantitative Evaluation (Using Metrics)	• Create robust evaluation sets (queries, "golden" reference data: relevant materials, most-relevant documents, ideal responses). • Consult end-users for query coverage • Manage golden dataset (Google Sheet with tabs for queries, templates, API output/expected results, rating guidance). • Automate execution via Colab Notebook (run queries, retrieve top results, save for review).	• Define success metrics and critical tasks. • Evaluate agent's action trajectory and tool use against "ideal" paths. • Assess quality, relevance, and correctness of final response. • Create JSON test files for unit testing (user query, expected tool use, intermediate/final responses). • Create evalset files for complex, multi-turn conversations. • Configure custom evaluation criteria (test_config.json). - Run automated evaluations via pytest or adk eval command.
UAT (Human Feedback/ Rating)	• Manual evaluation feedback for generated responses (5-point scale: 5: excellent, 1: bad/irrelevant). • Manual evaluation feedback for search results (5-point scale: relevance, order, missing results). • Provide specific feedback on issues (incorrect/incomplete summaries, style/tone, guardrail issues, non-answers, incorrect links). • Encourage adversarial testing. • Structured collection of manual ratings and detailed feedback.	• Review structured output from ADK evaluations. • Compare actual outputs against "expected" values in test/evalset files. • Validate intermediate agent responses for correct process. • Use web UI (adk web) for interactive evaluation and creating "golden" examples.
Continuous User Feedback (Thumbs Up/Down, Comments)	• Enable "Thumbs up" and "Thumbs down" buttons on the search widget. • Prompt users for reasons and comments if "Thumbs down" is selected. • Backend processing of user feedback (accessible to Google Cloud administrator and Google). • Align feedback with model improvement (clarify implicit vs. explicit feedback). • Communicate the value and impact of feedback to users. • Regularly review and analyze user feedback to identify themes and areas for improvement.	• Identify performance gaps from user feedback. • Convert user issues into new test cases or evaluate additions. • Enrich golden datasets with real-world user queries and ideal responses. • Refine expected tool use trajectories based on process-related user feedback. • Prioritize ADK evaluations based on frequent user-reported issues.

Tools to consider for evaluating ADK Agent

- Vertex AI Gen AI Evaluation Service
- Vertex AI Agent Engine
- Google Cloud Logging
- Google Cloud Monitoring

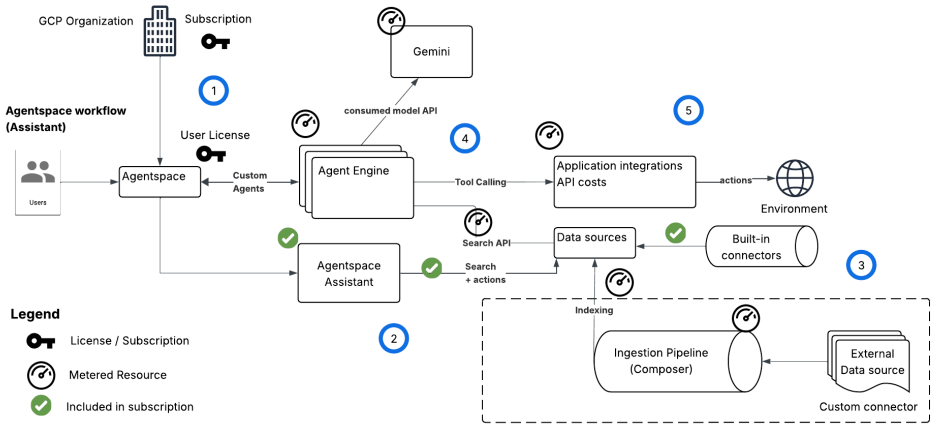
- Google Cloud Trace
- Google Sheets (for Golden Data Management)
- Colab Notebook (for Automated Execution & Evaluation)
- BigQuery

Tools to consider for evaluating Agents in Agentspace in Connectors

- Google Agentspace's Built-in Analytics & Monitoring
- Google Cloud Logging (specific to Connector services)
- Google Cloud Monitoring (specific to Connector services)
- BigQuery
- Grafana
- Prometheus

Cost management and optimization

Total cost of ownership of an Agentspace solution with custom agents, including billable metered and licensed resources



1) GCP organization acquires tiered subscription for Agentspace and assigns licenses to users.
2) Subscription includes access to Agentpspace assistant, including search on data sources and actions.
3) Data ingestion for custom connectors include metered cost for pipeline, search indexing.
   Built-in connectors associated with an Agentspace application are included in subscription.
4) Custom agents attached to Agent Engine are metered resources, in addition to Gemini API usage costs.
5) Tool calling may involve additional costs related to API and Application Integration connectors pricing.

Updated: 07/01/2025

The following table summarizes key LLM cost optimization strategies:

Strategy	Description	Impact on Cost	Implementation Considerations

Prompt Optimization	Craft prompts for concise outputs. Prompt Guardrails and prompting techniques like the “Chain of Draft” help minimize token count	Direct reduction in per-request token costs.	Requires careful prompt design and testing.
Token Input Optimization	Pre-process input Query and Context to optimize on input token volumes sent to the LLM.	Lowers input token costs.	Ensure critical information is not lost during preprocessing. Techniques like Dynamic Shot selection/ Dynamic context selection can be considered where
Response Caching	Store and reuse previously generated responses for identical or similar queries.	Reduces redundant API calls, saving costs & latency.	Suitable for deterministic or stable content; cache invalidation strategy needed.
Right-Sized Modeling	Segregate your tasks by complexity or identify low cost operations. Map these tasks to the most cost-effective model that meets the task's complexity and performance requirements.	Significant cost reduction by avoiding overpowered models for simpler tasks/ operations.	Requires evaluation of different models for specific tasks. Might be an overkill for Small scale applications
Dynamic Task Routing	Route queries to different models based on complexity or other criteria.	Optimizes cost by using expensive models only when necessary.	Needs a routing mechanism and criteria for model selection.
Fine-Tuning Smaller Models	Adapt smaller, often open-source, models to specific tasks using domain data.	Can achieve high performance at lower cost than large models if fine tuning significantly reduces the token count per prompt. Model obsolescence and Data drift leads to a maintenance cost	Requires data, expertise, and compute resources for fine-tuning. Primary drivers of fine tuning costs are the training time and the quality and volume of training data
Request Batching	Leverage Batch APIs or In-prompt batching technique to batch your requests/queries	Batch APIs offer a 50% reduction on model consumption costs. In-prompt batching reduces per-request token costs	Error handling and Turn around times for Batch processing must be accessed for requirements of the use case. In prompt batching is suitable for use cases with a common context with minimal variations
Leverage Open-Source	Utilize freely available open-source LLMs.	Eliminates licensing or per-token fees for proprietary models.	May require more effort for deployment, maintenance, and achieving desired performance.

Cloud Cost Management	Utilize cloud provider discounts (e.g., CUDs), free tiers, and optimization tools (e.g., Vertex AI Model Optimizer).	Reduces infrastructure and service costs.	Requires understanding of cloud pricing models and commitment for discounts.