Slide 1: Agentic AI – Future of Autonomous Intelligence

"Today, I'm excited to talk about the future of Agentic AI—a shift from passive models to active, autonomous agents. These systems don't just respond; they perceive, decide, and adapt. We're entering an era where AI has goals, memory, and the ability to reflect. This isn't theoretical—it's already transforming how we design intelligent systems. Let's dive into how and why this matters."

Slide 2: What is Agentic AI?

"Agentic AI goes beyond traditional machine learning. It refers to systems that not only make decisions but also plan, act, and adapt based on real-world feedback. These agents persist over time and learn through interaction, unlike static models. This evolution is critical for enabling autonomy in complex, dynamic environments. Agentic AI brings purpose, memory, and evolution to the AI lifecycle."

why now?

"Agentic AI is timely due to the rise of multi-agent frameworks like LangGraph, CrewAI, and AutoGen. We're seeing a major push for scalable autonomy in fields like healthcare, robotics, and enterprise automation. Goal-oriented LLMs are no longer experimental—they're being built into systems we use every day. The need for adaptable, decision-making agents has never been clearer. This is the right moment to invest in agentic thinking."

Slide 4: AI Agents

"In the context of generative AI, agents are systems that can take real actions—like booking flights, ordering services, or planning events. They go far beyond simple chatbots. Think of them as intelligent coworkers who can carry out tasks independently. Their actions are informed by goals and context, not just prompts. These agents are already changing workflows in both consumer and enterprise spaces."

Traditional LLM pipelines like ChatGPT are typically reactive — they answer based on a single prompt without long-term planning or external tool use. RAG pipelines enhance this with retrieval, but the flow remains stateless. Agentic AI introduces **autonomy, goal-driven behavior, tool invocation, memory, and planning loops**, allowing agents to reason, act, and reflect over time. They don't just generate responses — they complete tasks.

Slide 5: Agentic AI in the News

"We've seen a surge in media coverage highlighting how autonomous agents are redefining tech landscapes. From AI assistants managing schedules to bots making e-commerce decisions—Agentic AI is headline-worthy. This attention also reflects growing industry investment. Everyone from startups to Big Tech is racing to build the next generation of agents. The excitement is real, and it's accelerating adoption."

Slide 6: Agentic AI Evolution

"We've moved from reactive AI to proactive, long-lived agents. Early AI systems could only answer questions; now, they can plan, adapt, and persist. This evolution mirrors how human assistants operate—learning over time and improving outcomes. It's a major paradigm shift in AI design. The more they evolve, the more efficient and trusted they become."

Slide 7: Agentic AI Capabilities – Task Decomposition

"One key capability of agentic AI is decomposing complex tasks into simpler subtasks. Just like we break down big goals into steps, agents learn to do the same. This modularity makes execution efficient and scalable. It also enables collaboration between specialized agents. Task decomposition is what makes agents truly powerful problem-solvers."

Slide 8: Agentic AI Capabilities – Memory Management

"Memory is the backbone of intelligence. In Agentic AI, agents store both short-term and long-term memory to stay contextually aware. This enables them to recall past decisions, adapt to new input, and remain consistent over time. Memory also supports personalization. It's what turns a one-time assistant into a dependable partner."

Slide 9: Agentic AI Capabilities – Reflect & Adapt

"Reflection allows agents to evaluate their actions and adapt strategies. It’s like learning from mistakes—something static models cannot do. Reflection improves both accuracy and efficiency over time. This continuous feedback loop makes agents more intelligent and reliable. Ultimately, reflection turns autonomy into wisdom."

Slide 10: Agentic AI Reference Architecture

"This architecture outlines how various components—planning, memory, reflection—come together in a cohesive system. Each block plays a critical role in achieving autonomy. By decoupling components, we ensure flexibility, scalability, and composability. It’s not a monolith; it’s a modular agentic system. This is the blueprint for real-world deployment."

Slide 11: Generative AI Lifecycle

"This slide captures the lifecycle of building generative AI applications—from prompt engineering to feedback loops. Understanding this cycle helps us see where agentic capabilities fit in. Agents add autonomy to each stage, especially in reflection and planning. The goal is to go beyond isolated outputs to continuous learning systems. It’s about building systems that improve with every interaction."

Slide 12: Gen AI Architecture Patterns – APIs & Embedded Gen AI

"We distinguish between black-box LLM APIs like ChatGPT and embedded models tailored for enterprise use. While APIs are quick to deploy, embedded Gen AI ensures control, data privacy, and customization. For enterprises, balancing speed with security is key. Prompt engineering plays a huge role in both. Governance and validation must be part of the architecture."

Slide 13: Gen AI Architecture Patterns – Retrieval-Augmented Generation (RAG)

"RAG is a smart alternative to fine-tuning. It enriches prompts with real-time context retrieved from databases or documents. The pipeline works in three stages—retrieve, augment, and generate. This method grounds answers in facts and makes agents more reliable. It’s cost-effective and improves interpretability too."

Slide 14: Agentic AI Platform Reference Architecture

"This platform view shows how enterprises can orchestrate multiple agents to solve end-to-end tasks. It’s like building with Lego blocks—each agent is reusable and composable. The platform manages memory, task routing, and execution control. It opens doors to scalable AI systems. This architecture enables real, production-ready Agentic AI."

Slide 15: AI Agents Marketplace & Discovery for Multi-agent Systems

"As agents grow in number, we need a marketplace to register and discover them. This ensures reusability and clarity in large-scale systems. It also helps in choosing the best agent for a task—based on capability, cost, or reliability. Discovery is core to automation at scale. It’s how dynamic systems stay smart."

Slide 16: Complex Task Decomposition

"Complex tasks must be decomposed hierarchically into simpler subtasks. Some agents are created manually (static), while others are formed at runtime (dynamic). Dynamic composition makes agents flexible and context-aware. Task planning becomes an orchestration process. This is the foundation of adaptive AI workflows."

Slide 17: Agent Marketplace & Discovery of AI Agents

"Agent discovery requires a well-maintained registry describing what each agent can do. It’s like an app store for intelligent agents. Matching tasks to agents involves capabilities, constraints, and current state. This discovery module supports dynamic routing. Without it, agent orchestration would be guesswork."

Slide 18: Limitations of LLMs for Agent Execution

"LLMs alone aren't enough to power complex agents. They struggle with long-term planning, scalability, and system integration. Routing everything through a single LLM is inefficient. We need hybrid systems that combine LLMs with dedicated agents and learned execution plans. That's how we overcome current limits."

Slide 19: Non-determinism in Agentic AI Systems

"Agents operate in real-world environments filled with uncertainty. Some outcomes are non-deterministic, like delivery modes or credit checks. We must design agents to handle branches, exceptions, and alternate flows. This flexibility is what sets agentic systems apart. Real autonomy means handling the unknown gracefully."

Slide 20: Learning-to-Rank for Agent Discovery

"We use Learning-to-Rank (L2R) algorithms to match user prompts with relevant agents. Agent descriptions are converted to embeddings, enabling semantic search. This supports fast and accurate agent selection. It's essential for personalization and planning. L2R also helps in building a feedback loop for improving future matches."

Slide 21: MCP Demo

"Here we showcase an actual demo using MCP—our multi-agent control platform. This brings together planning, execution, and observability. It's proof that these concepts are not just theoretical—they're working in production. Watch how agents plan and execute tasks with minimal human intervention. This is Agentic AI in action."

Slide 22: Personalizing UX for Agentic AI

"Just like LLMs are fine-tuned, agents must be tailored for different user personas and contexts. Personalization improves trust, relevance, and task success. It enables agents to align tone, language, and behavior with each user. Enterprise-grade agents must be smart *and* personal. That's the key to adoption."

Slide 23: AI Agent Personalization Architecture

"This slide outlines how user data is embedded and used to personalize agent behavior. Memory modules store context, while fine-tuning adapts responses. A routing layer matches users to the right agent persona. This architecture ensures both performance and privacy. It's the last mile to making agents truly helpful."

Slide 24: User Persona Based Agent Personalization

"Here we go deeper into how user personas are formed. Aggregated histories help categorize users into segments. Each segment gets tailored agent behavior. This balances scalability with personal touch. The routing engine then ensures users get the most aligned support agent."

Slide 25: USER-LLM – User Data Embeddings

"USER-LLM captures behavioral patterns from user interactions. It transforms noisy, diverse inputs into rich embeddings. These embeddings drive personalized decisions. It's a bridge between human intent and machine understanding. It ensures agents 'remember' the user without violating privacy."

Slide 26: Reinforcement Learning Based Personalization

"RL enables agents to improve based on user feedback and sentiment. Each interaction leads to a reward or penalty, tuning the agent's behavior. Over time, agents adapt to user preferences. This is learning in action—not just training and deployment. RL personalization makes agents better after every use."

Slide 28: Agent Observability & Memory Management

"Observability is crucial for understanding how agents behave in real time. Unlike centralized systems, agents are distributed and often opaque. We need tools to inspect agent state, track execution, and debug issues. This ensures trust and accountability. Monitoring is what keeps autonomy safe."

Slide 29: Observability Challenges for Agentic AI

"Challenges arise from agent autonomy, privacy, and parallelism. We don’t always know which agents will be involved or how they behave internally. Dynamic composition makes monitoring harder. But with the right state tracking and logging, we can regain visibility. We must monitor without interfering."

Slide 30: Stateful Execution for AI Agents

"Agents must be trackable across time. Stateful execution means we can ask where the agent is in a task, and what’s happening. This includes local, composite, and historical queries. It turns agents into audit-friendly systems. Without state, agents become black boxes."

Slide 31: Conversational Memory Management using Vector DBs

"Vector DBs store and retrieve conversation memory, keeping agents context-aware. They transform raw inputs—text, audio, video—into searchable embeddings. During chat, agents query long-term memory for relevant context. This leads to more informed and coherent responses. Vector DBs are essential for scalable memory."

Slide 32: Human Memory Understanding

"We take inspiration from human memory—semantic, episodic, procedural, and emotional. Agentic systems need the same diversity. Not every memory is a fact—some are experiences, skills, or feelings. Mimicking human memory enhances AI’s ability to relate and adapt. It brings agents closer to human-level understanding."

Slide 33: Agentic Memory Management (1 & 2)

"The memory router handles whether queries go to short-term or long-term memory. STM is for dynamic lookups; LTM for learned patterns. A transformer constantly syncs STM and LTM to build structured memory graphs. Over time, agents form episodes and procedures. This allows learning across interactions."

Slide 34: Agentic AI Scenarios – RAGs & RL Agents

"We end with advanced scenarios: Agentic RAGs and RL agents. Agentic RAGs extend to SQL and structured data, enabling multimodal retrieval. RL agents benefit from LLM fine-tuning for smarter policy optimization. These hybrids show that agentic AI is not one-size-fits-all. It’s a modular ecosystem for enterprise-grade intelligence."

Slide 35: Responsible AI Agents

"Lastly, responsible AI is non-negotiable. We must build agents that are explainable, fair, and privacy-preserving. This means tackling bias, ensuring auditability, and protecting user data. Agentic AI must be safe to scale. That’s how we build trust—not just intelligence."

Slide 36: Thank You & Questions

"Thank you so much for your attention. I truly believe Agentic AI will define the next chapter of enterprise automation. I’m happy to take your questions and discuss how we can apply this within our company. Let’s build smarter, safer systems—together."

Slide 37: Explainability

"Explainability means that AI systems should not be black boxes. Every prediction must come with a justification we can understand. This is especially important in regulated industries like finance and healthcare. When agents can explain their actions, they earn trust. It’s key for user adoption and accountability."

Slide 38: Fairness & Bias

"Bias enters models through biased data. Historical inequalities, lack of representation—all of these impact outcomes. We need to design pipelines that actively mitigate these biases. Fairness isn’t just ethical; it’s critical for accuracy and equity. Agentic AI must be inclusive and just."

Slide 39: ML Privacy Risks

"Privacy risks include membership inference and property inference attacks. Even with just API access, attackers can extract sensitive data patterns. Protecting models means securing both data and predictions. Privacy-preserving techniques like differential privacy and federated learning are essential. Safety isn’t optional—it must be baked in."

Slide 40: Gen AI Privacy Risks – Novel Challenges

"LLMs introduce new privacy risks: data leaks from pretraining, chat history exposure, and unclear compliance. We need strong guidelines on how conversations are stored and used. Agents must align with user intent and privacy expectations. It’s not just about security—it’s about respecting trust."

Slide 41: Responsible Deployment of AI Agents

"Responsible deployment includes monitoring, testing, and enforcing policy compliance. Agents need defined behaviors and fallback strategies. We must evaluate their real-world performance, not just lab metrics. Think of this as DevOps for intelligent systems. Responsible AI is what enables sustainable innovation."

Slide 42: Use-Case Specific Evaluation of LLMs

"Evaluation should be aligned with the task at hand. General benchmarks are not enough—we need custom metrics per use case. For example, accuracy in summarization differs from accuracy in diagnosis. Our agents must be evaluated for what *they* are built to do. This ensures precision and reliability."

Slide 43: LLM Safety Leaderboard

"Tools like Hugging Face’s Safety Leaderboard and DecodingTrust help benchmark the trustworthiness of LLMs. They test for harmful outputs, hallucinations, and alignment. These are critical checkpoints before LLMs are deployed inside agents. We must prioritize safety metrics alongside performance."

Slide 44: Agentic RAGs

"Agentic RAGs extend traditional RAGs to structured data, like SQL. Agents act as smart retrievers that combine database queries with document search. The result is highly contextual, multi-source responses. This fusion of structured and unstructured data opens up new enterprise possibilities. It’s RAG with intelligence and agency."

Slide 45: Reinforcement Learning Agents

"Not all agents must rely on LLMs. For sequential decision-making tasks, reinforcement learning is often more suitable. RL agents learn policies based on feedback and rewards. These agents can be used for robotics, recommendations, and automation workflows. Agentic AI includes *all* intelligent agents—not just language-based ones."

Slide 46: LLM-Based Fine-Tuning of RL Agents

"LLMs can guide or fine-tune RL agents by shaping their reward functions or generating policy hints. This hybrid approach combines reasoning with learning from interaction. It’s especially useful in sparse-reward environments. The goal is not to replace RL, but to accelerate it using language models. This is a powerful synergy."

Slide 47: Responsible AI Agents (Reinforcement)

"We wrap up with a reinforcement of responsible agent design. Every agent should be explainable, fair, secure, and adaptable. From task execution to memory management to personalization—responsibility must guide every layer. It's not just about capability, but integrity. That's the future we're shaping with Agentic AI."

3. How do you ensure reliability, safety, or alignment in autonomous agents?

Answer:
We apply several layers:

- **Tool validation**: We constrain agent tools via schema checks and input sanitization.
- **Reflective loops**: Agents reflect on actions (à la ReAct or LangGraph planner-checker pattern).
- **Memory summarization**: Reduces hallucination from prompt bloat.
- **Human-in-the-loop supervision** (when needed): Ensures high-risk actions are reviewed.
  For alignment, we guide models using system prompts + fine-tuned reasoning templates per domain.

1. Goal Representation

Agents are initialized with a high-level **goal prompt** or **task description**, like:

> "Plan a 3-day New York trip with kid-friendly activities."

This goal is stored internally — often in memory, or passed along as the root of a reasoning tree.

---

2. Planning Component

Agents use a **planner** (either a logic-based loop or an LLM-based reasoning chain) to:

- Break the goal into subtasks.
- Determine the order of execution.
- Identify tools or sub-agents required.

> Example:
> "Day 1: Central Park + Zoo → Day 2: Museum → Day 3: Ferry ride"