

callibrity

Optimizing Complex Workflows with Event-Driven Multi-Agentic Approach

Mary Grygleski
Director of Emerging Technologies
@mgrygles

February 2025



Agenda

- Who's Mary?
- A Brief Intro and Background of Gen AI
- The Event-Driven World and AI ?
- Applying Event Approach to: RAG
- Applying Event Approach to Agents and Workflows
- Challenges of GenAI
- Resources

Who is Mary?







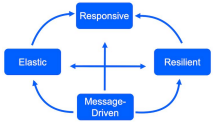
Passionate Advocate



Java Champion



PULSAR





mgrygles



mary-grygleski



mgrygles



mgrygles

JAKARTA EE

Open Liberty

MICROPROFILE

RED HAT OPENSHIFT











The GenAI Collective

Director of Emerging Technologies

- Generative AI
- Streaming
- Distributed Systems
- Reactive Systems
- IoT/MQTT
- Real-Time AI/ML

A Brief Intro and Background of Generative AI

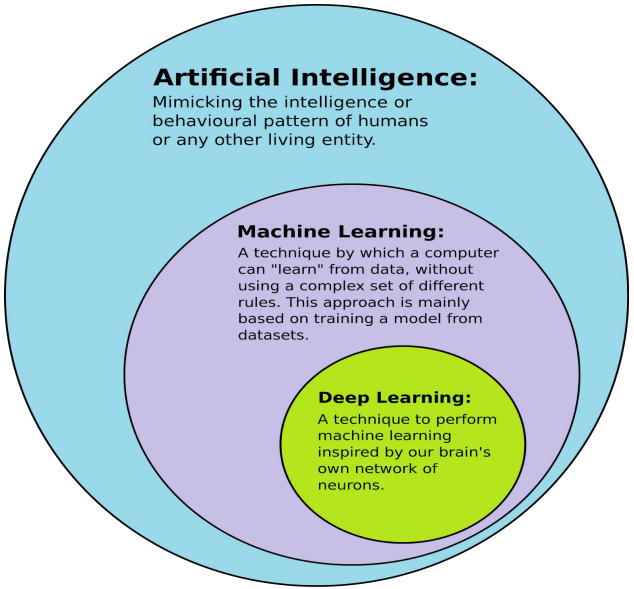
AI: Infusing into All Fabrics of our Lives

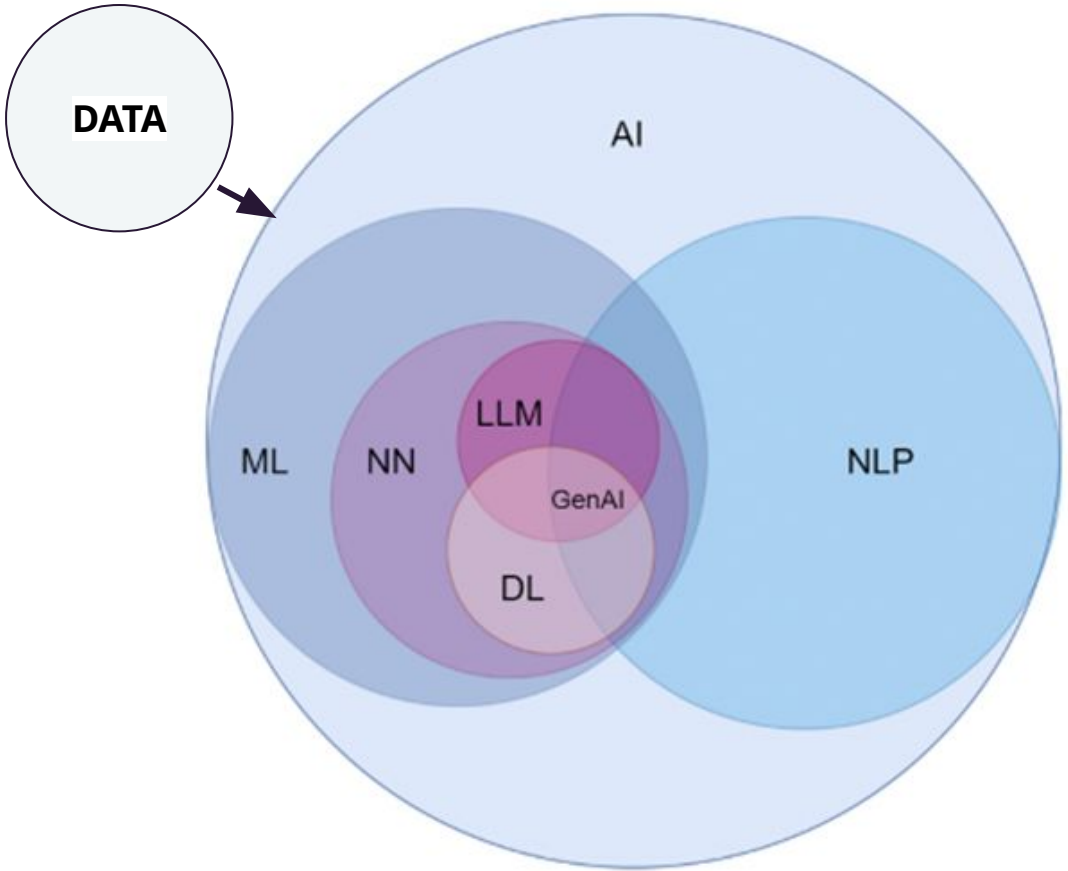


CONSUMER (e.g. Developers using GitHub Copilot)



PRODUCER (e.g. Developers building AI applications, training models…etc)


Artificial Intelligence:
Mimicking the intelligence or behavioural pattern of humans or any other living entity.
Machine Learning:
A technique by which a computer can "learn" from data, without using a complex set of different rules. This approach is mainly based on training a model from datasets.
Deep Learning:
A technique to perform machine learning inspired by our brain's own network of neurons.


DATA
AI
ML
NN
LLM
GenAI
DL
NLP

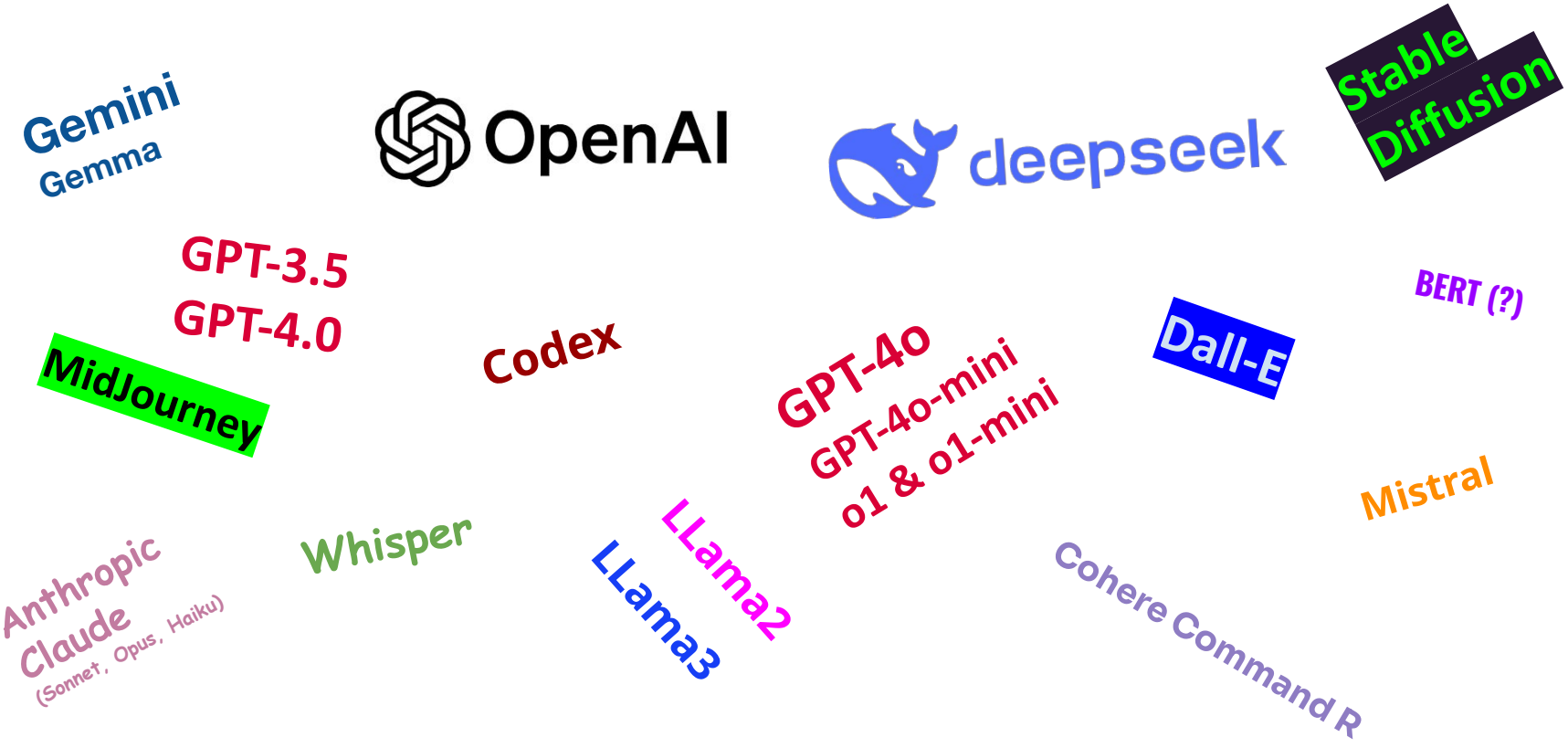
A fascinating look at the GenAI “era”



What is Generative AI (GenAI) ?

- A “disruptive” field in AI
- Has the potential to change the way we create and consume content
- Generate new contents based on prompts
- Uses a combination of machine learning and deep learning to produce contents
- Tends to be on the “creative” side: generating code, writing an article, designing new fashion, composing a new song… especially when compared with Predictive AI which tends to be more strictly about business, marketing, and weather forecasting.

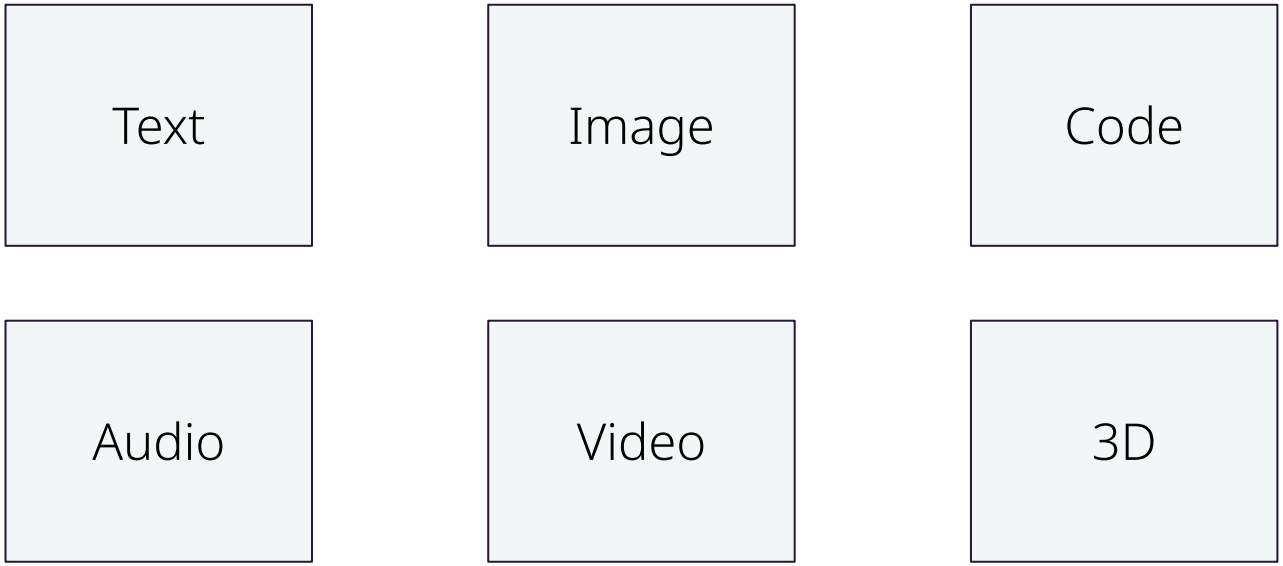
Generative Models



Generative Apps



Expanding Modality / Multi-Modal



Stepping Things Up - Beyond the Hype

- LLMs’ limitations
  - No memory
  - Probabilistic results
  - Opague - can’t add attribution to its answers
  - Outrageous cost
- The hype: “one-shot” usages (or zero-shot)
- More will be required to make it ready for prime time:
  - Scalability
  - Complex task (e.g. workflows)
  - Memory - remembering the state (particularly for enterprise-level usages)

Generative Pre-Trained Transformer (GPT)

- Takes simple prompts (in natural human language) as input
- Pattern matching (also commonly being called as “search”)
- Answers questions for the **prompts**
- Produce contents such as: a new essay, a blog post, a new computer program

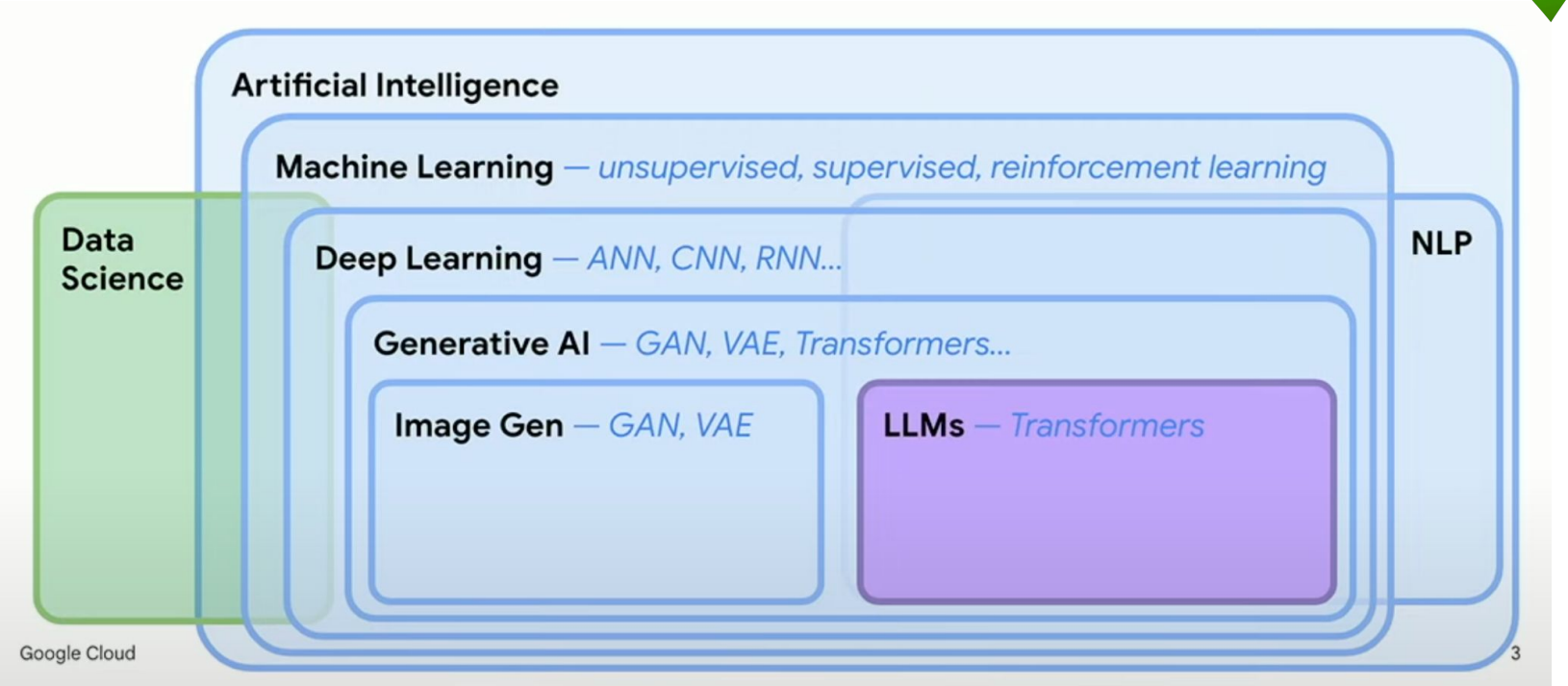
What is Natural Language Processing (NLP)?

- An interdisciplinary sub-field of linguistics of computer science
- Primary concern is to process natural language datasets (as such text corpora or speech corpora)
- Uses rule-based or probabilistic machine learning approaches
- Enables computer to learn from contents, including the contextual nuances of the language itself
- Ideally to draw insights from the documents

Large Language Models (LLMs)

- A type of Machine Learning or Deep-Learning Model
- Foundation type of model trained on a broad range of data providing general-purpose usages
- Typically the pre-training consumes a humongous amount of time and resources: $$$$$$$ GPUs
    - Language processing (utilizing NLP)
    - Visual comprehension
    - Code generation
    - Human-centered engagement
        - Answers questions (**prompts**) just like a human: analyze sentiments, chatbot conversations, etc.

How do LLMs fit into the AI/ML picture?



The Event-Driven World and AI ?

At the Heart of AI:

DATA and the

FLOW **of Data** in

many directions



Working with Data

- Architecture level: Event-Driven Architecture
- Handling Data in different States:
  - At Rest
  - In Transit
  - In Use
- Techniques:
  - Event Streaming
  - Event Messaging
  - Event Sourcing & CQRS

Generative AI Streaming usages

- GenAI App Input
  - Prompts
    - Command-line
    - Browser (different devices)
    - Mobile App
- GenAI Intermediary steps:
  - During RAG operations:
    - different sources (DB, filesystem, socket?, message queue, Kafka topic…)
    - Sending requests to LLMs
- GenAI App Output
  - Responses sent from LLMs

Benefits of the Event-Driven Approach

- Handles the dynamic nature of data and its flow patterns
- Communicates asynchronously via event streams
- Enables scalability and flexibility (independent scaling of loosely-coupled components)
- Facilitates integration of diverse AI models and services
- BUT, be aware of the challenges:
    - More difficult to debug
    - Harder to track the flow of calls and the data they carry
    - Observability needs more work

Applying Event Approach to -*Retrieval-Augmented Generation (RAG)*

What is RAG?

- A hybrid framework that integrates the 2 components of the RAG models:
  - Retrieval model
  - Generative model
- The purpose is to produce text that is not only contextually accurate but also information-rich

Role of the Retrieval Model

- In a nutshell, the retrieval model acts as a specialized 'librarian’, pulling in relevant information from a database or a corpus of documents.
- This information is then fed to the generative model, which acts as a 'writer,' crafting coherent and informative text based on the retrieved data. The two work in tandem to provide answers that are not only accurate but also contextually rich.

Role of the Generative Model

- Act as creative writers, synthesizing the retrieved information into coherent and contextually relevant text.
- Usually built upon Large Language Models (LLMs), generative models have the capability to create text that is grammatically correct, semantically meaningful, and aligned with the initial query or prompt.
- They take the raw data selected by the retrieval models and give it a narrative structure, making the information easily digestible and actionable.
- In the RAG framework, generative models serve as the final piece of the puzzle, providing the textual output we interact with.

Applying Event Approach to - *Agents and Workflows*

Fundamentals: Agents and Agentic

- Agents
  - one who is authorized to act for or in the place of another
  - a computer application designed to automate certain tasks

- **Agentic**
  - *Agentic* refers to someone or something capable of achieving outcomes independently (“functioning like an agent”) or possessing such ability, means, or power (“having agency").
  - It is especially used with a type of artificial intelligence (AI), often referred to as an AI agent, designed to execute complex tasks autonomously or with little human involvement

Gen AI Agents (within the software context)

- Software entities
  - Orchestrate complex workflows
  - Coordinate the activities of multiple agents
  - Process logic
  - Evaluate answers

Multi-Agentic Systems

- Multiple autonomous agents working together
- Each agent specializes in different task(s)
- Emergent intelligence from agent interactions
    - Divide complex tasks into manageable subtasks
    - Leverage strengths of different AI models
    - Increased robustness and fault tolerance

Fundamental Building Blocks

- Within the Java context, and leveraging on Langchain4j's approach:
    - AIServices
        - More flexible than the old "Chains" concept in Langchain
        - Declarative interface to the desired API underneath via and object (proxy)

Towards Standardization? Multi-Context Protocol (MCP)

- Originated by Anthropic
  - MCP is an open protocol that standardizes how applications provide context to LLMs.
  - MCP helps you build agents and complex workflows on top of LLMs. LLMs frequently need to integrate with data and tools
- MCP Provides:
  - A growing list of pre-built integrations that your LLM can directly plug into
  - The flexibility to switch between LLM providers and vendors
  - Best practices for securing your data within your infrastructure

Combining EDA and Multi-Agent for GenAI

- Use events to coordinate agent activities
- Event streams enable asynchronous agent communications
- Event-driven workflow orchestrate multi-step generation

Key Components:

- Event Bus: Manage event streams
- API Gateway: Handles external events
- Agents: Autonomous components w/ specific capabilities
- AI Models: Underlying generative models (e.g. GPT, DALL-E, Mistral...)

Event Types and Agent Types

Event Types:

- UserRequest: Initial generation request
- SubtaskAssigned: Agent given a special subtask
- ResultGenerated: Output from an agent/model
- FinalResponse: Completed result for user

Agent Types:

- Coordinator: Decomposes tasks, assigns to agents
- Generator: Produces content using AI models
- Evaluator: Assesses quality/relevance of generated content
- Refiner: Improves/iterates on initial outputs

An AutoGen Example with Different Types of LLM Agents, including Humans in the Loop

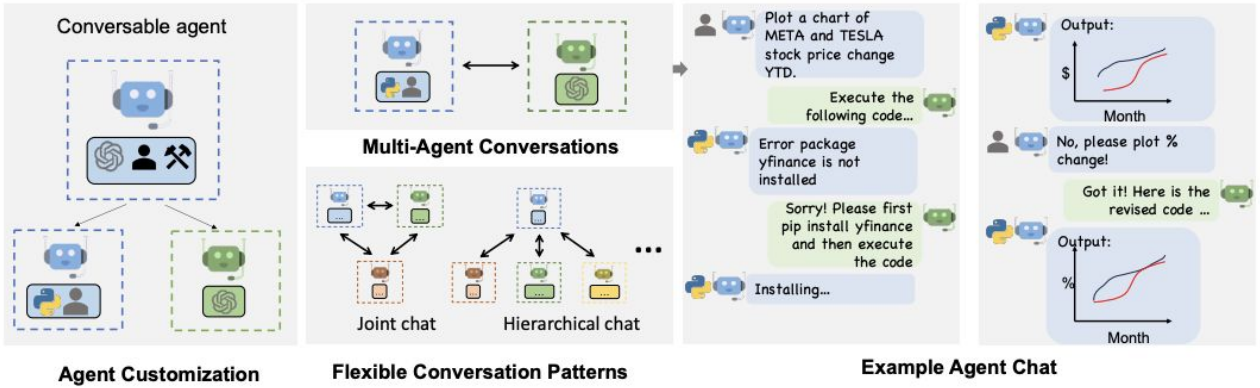


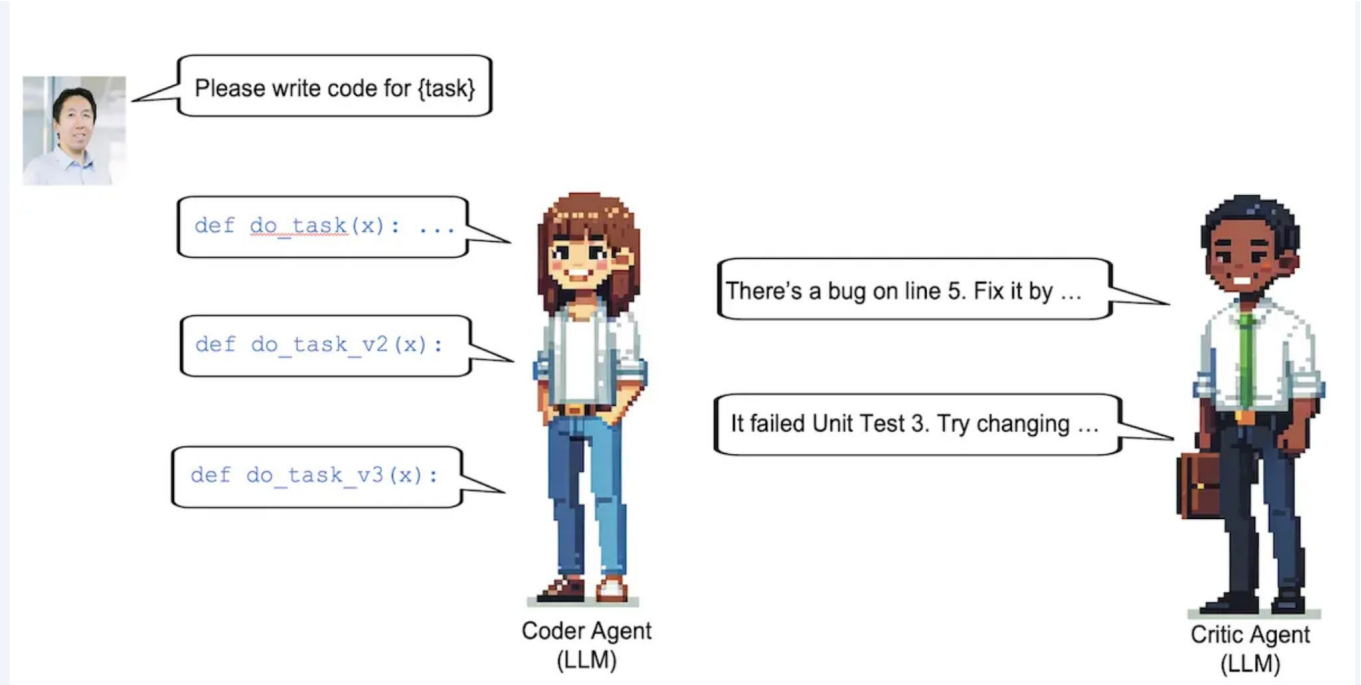
Figure 1: AutoGen enables diverse LLM-based applications using multi-agent conversations. (Left) AutoGen agents are conversable, customizable, and can be based on LLMs, tools, humans, or even a combination of them. (Top-middle) Agents can converse to solve tasks. (Right) They can form a chat, potentially with humans in the loop. (Bottom-middle) The framework supports flexible conversation patterns.

AutoGen: Enabling Next Gen LLM Apps via Multi-Agent Conversations (https://arxiv.org/abs/2308.08155)

Agentic Design Patterns

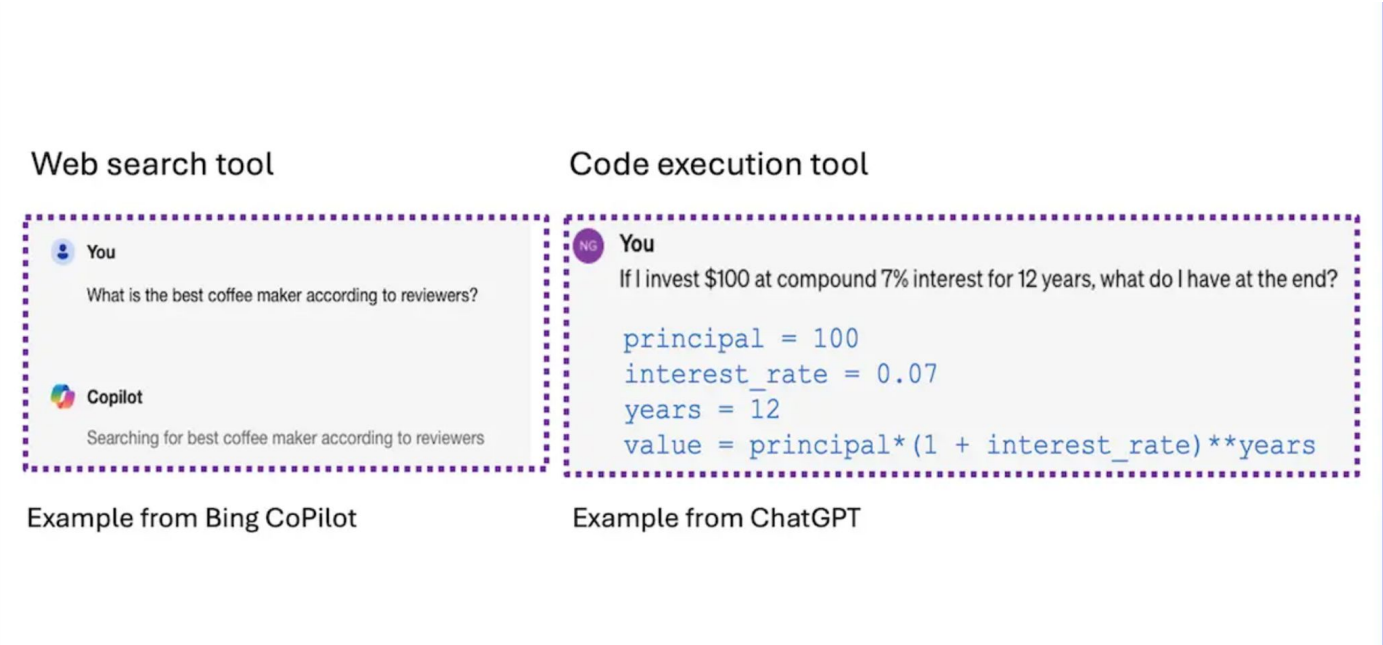
- **Reflection**
- **Tool Use**
- **Planning**
- **Multi-Agent Collaboration**

Reflection



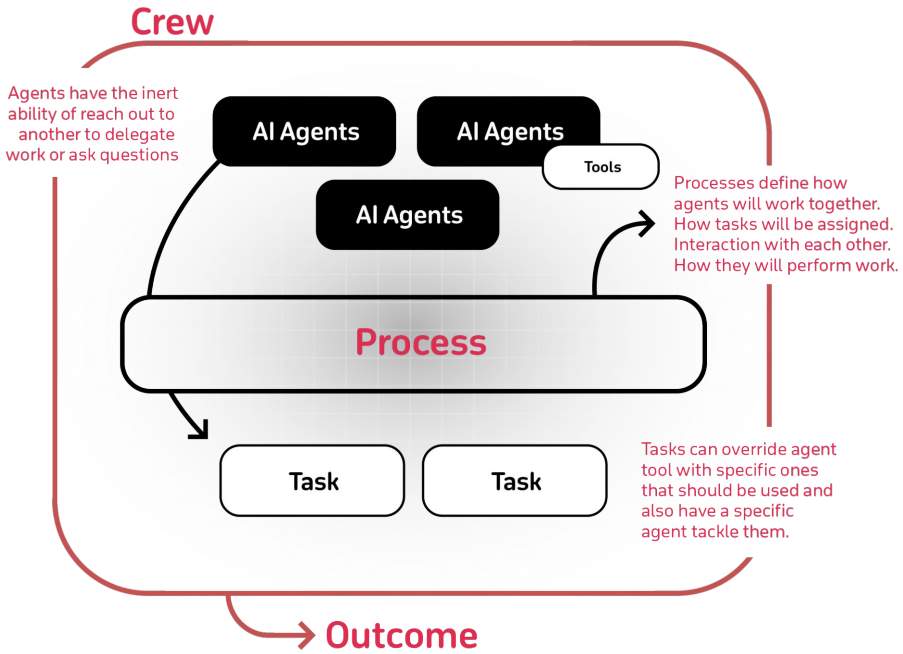
https://www.deeplearning.ai/the-batch/agentic-design-patterns-part-2-reflection

Tool Use



https://www.deeplearning.ai/the-batch/agentic-design-patterns-part-3-tool-use

Tool Use (Illustrated, from CrewAI)



Source: https://github.com/crewAIInc/crewAI/blob/main/docs/crewAI-mindmap.png

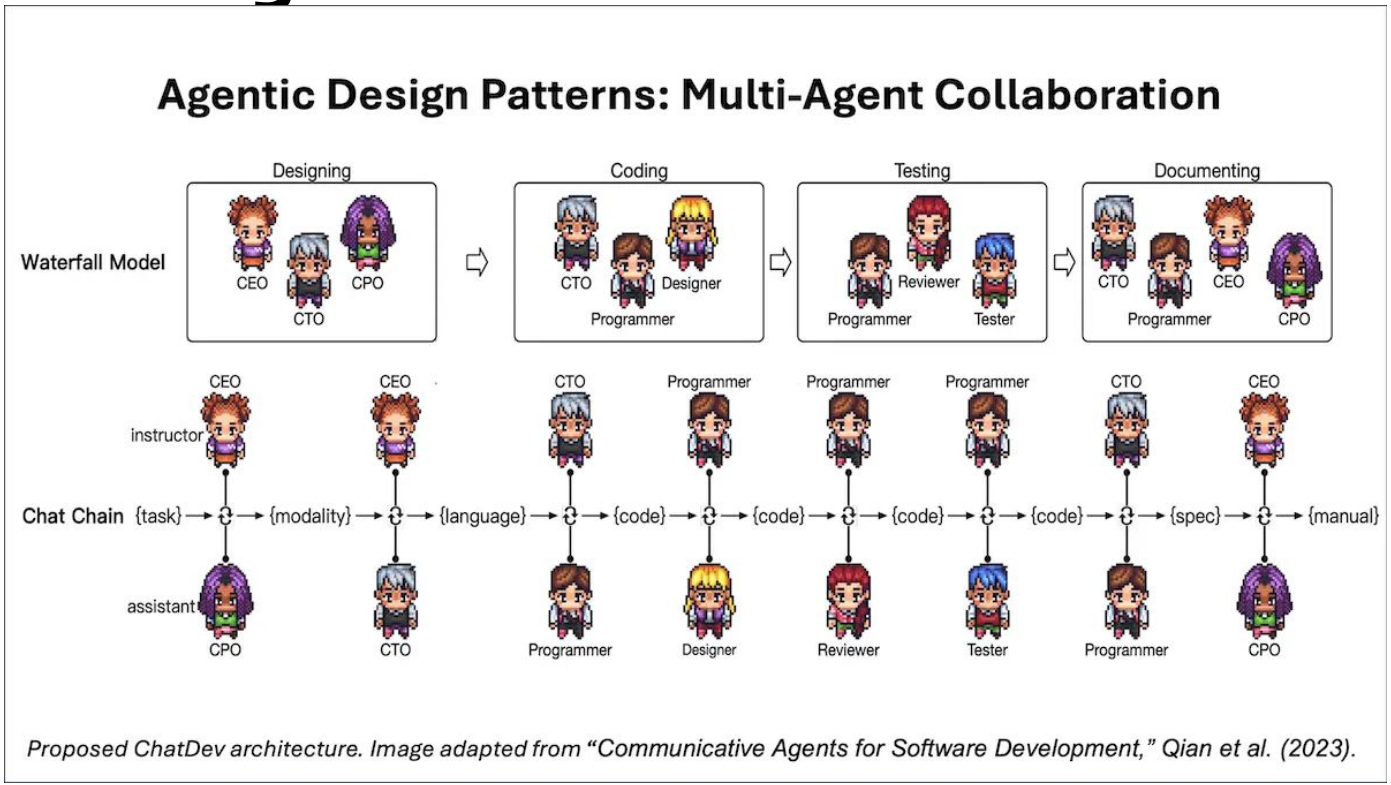
Planning



Example adapted from "HuggingGPT: Solving AI Tasks with ChatGPT and its Friends in Hugging Face," Shen et al. (2023)

https://www.deeplearning.ai/the-batch/agentic-design-patterns-part-4-planning

Multi-Agentic Collaboration



Proposed ChatDev architecture. Image adapted from "Communicative Agents for Software Development," Qian et al. (2023).

https://www.deeplearning.ai/the-batch/agentic-design-patterns-part-5-multi-agent-collaboration/

Examples of a Few Multi-Agentic Libraries

- **AutoGen**
- **CrewAI**
- **LangGraph**
- **And on the Java side: Quarkus-LangChain4j (CDI)**
    - https://docs.quarkiverse.io/quarkus-langchain4j/dev/agent-and-tools.html

Challenges

Be aware of the following issues (partial list)

- Complex system design and coordination
- Ensure consistency across distributed agents
- Handling failures and partial results
- Optimizing for latency and resource usage
- Sustainability issues
    - Energy-intensive / energy hog
    - Eco-unfriendly
- Hallucinations in GenAI with the LLMs
- Ethical concerns / Potential misuse
    - Currently no one is there to oversee its usages
- Prompt Injection



Resources

This slide deck can be accessed here:

https://bit.ly/4hFUvBH





Information

- arxiv.org Paper (AutoGen: Enabling Next-Gen LLM Apps via Multi Agent Conversations):

  https://arxiv.org/abs/2308.08155

- OSS Insight Collection (GenAI Agentic Libraries):

  https://ossinsight.io/analyze/microsoft/autogen

  https://ossinsight.io/analyze/langchain-ai/langgraph

  https://ossinsight.io/analyze/crewAIInc/crewAI

- Mary's Article on Java Advent Calendar (December 2024): https://bit.ly/3CFaSjd

Follow Mary's Stream
&
The Elusive AI Podcast

[Different topics: GenAI/ChatGPT, AI/ML, Java, Python, JS/TS, Open Source, Distributed Messaging, Event-Streaming, Cloud, DevOps, etc]

Wed|Thurs|Fri-afternoon-US/Central

https://twitch.tv/mgrygles

https://youtube.com/@marygrygleski9271

Kindly please share your feedback:
https://bit.ly/42FUIkk

Thank You

Chicago Java Users Group:
https://cjug.org



The GenAI Collective
https://www.genaicollective.ai/

AICamp https://aicamp.ai

Mary Grygleski callibrity
https://www.linkedin.com/in/mary-grygleski/
@mgrygles
https://discord.gg/RMU4Juw