Gravitas, S. Autogpt. https://github.com/Significant-Gravitas/AutoGPT, 2023. Accessed: 2025-04-27.

Greshake, K., Abdelnabi, S., Mishra, S., Endres, C., Holz, T., and Fritz, M. Not what you've signed up for: Compromising real-world llm-integrated applications with indirect prompt injection. In *Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security*, 2023.

Hammond, L., Chan, A., Clifton, J., Hoelscher-Obermaier, J., Khan, A., McLean, E., Smith, C., Barfuss, W., Foerster, J., Gavenčiak, T., Han, T. A., Hughes, E., Kovařík, V., Kulveit, J., Leibo, J. Z., Oesterheld, C., de Witt, C. S., Shah, N., Wellman, M., Bova, P., Cimpeanu, T., Ezell, C., Feuillade-Montixi, Q., Franklin, M., Kran, E., Krawczuk, I., Lamparth, M., Lauffer, N., Meinke, A., Motwani, S., Reuel, A., Conitzer, V., Dennis, M., Gabriel, I., Gleave, A., Hadfield, G., Haghtalab, N., Kasirzadeh, A., Krier, S., Larson, K., Lehman, J., Parkes, D. C., Piliouras, G., and Rahwan, I. Multi-agent risks from advanced ai. Technical Report 1, Cooperative AI Foundation, 2025.

IBM. Cost of a data breach report 2025, 2025. URL https://www.ibm.com/reports/data-breach.

IBM. Multi-agent orchestration (watsonx orchestrate). https://www.ibm.com/products/watsonx-orchestrate/multi-agent-orchestration, 2025. Product information page.

Jones, E., Dragan, A., and Steinhardt, J. Adversaries can misuse combinations of safe models. In *International Conference on Machine Learning*, 2025.

Kutasov, J., Sun, Y., Colognese, P., van der Weij, T., Petrini, L., Zhang, C. B. C., Hughes, J., Deng, X., Sleight, H., Tracy, T., Shlegeris, B., and Benton, J. Shade-arena: Evaluating sabotage and monitoring in llm agents. *arXiv preprint arXiv:2506.15740*, 2025.

Lee, D. and Tiwari, M. Prompt infection: Llm-to-llm prompt injection within multi-agent systems. *arXiv preprint arXiv:2410.07283*, 2024.

Liu, Y., Deng, G., Li, Y., Wang, K., Wang, Z., Wang, X., Zhang, T., Liu, Y., Wang, H., Zheng, Y., and Liu, Y. Prompt injection attack against llm-integrated applications. *arXiv preprint arXiv:2306.05499*, 2024a.

Liu, Y., Deng, G., Xu, Z., Li, Y., Zheng, Y., Zhang, Y., Zhao, L., Zhang, T., Wang, K., and Liu, Y. Jailbreaking chatgpt via prompt engineering: An empirical study. *arXiv preprint arXiv:2305.13860*, 2024b.

Microsoft. Multi-agent orchestration, maker controls, and more: Microsoft copilot studio announcements at microsoft build 2025. https://www.microsoft.com/en-us/microsoft-copilot/blog/copilot-studio/multi-agent-orchestration-maker-controls-and-more-microsoft-copilot-studio-announcements-at-microsoft-build-2025/, 2025a. Accessed 2025-06-19.

Microsoft. Mixture of agents — autogen, 2025b. URL https://microsoft.github.io/autogen/stable/user-guide/core-user-guide/design-patterns/mixture-of-agents.html. Accessed: 2025-04-27.

Parthasarathy, K., Vaidhyanathan, K., Dhar, R., Krishnamachari, V., Kakran, A., Akshathala, S., Arun, S., Karan, A., Muhammed, B., Dubey, S., and Veerubhotla, M. Engineering LLM Powered Multi-Agent Framework for Autonomous CloudOps . In *International Conference on AI Engineering – Software Engineering for AI*, 2025.

Patil, S. G., Zhang, T., Wang, X., and Gonzalez, J. E. Gorilla: Large language model connected with massive apis. In *Advances in Neural Information Processing Systems*, 2024.

Pedro, R., Coimbra, M. E., Castro, D., Carreira, P., and Santos, N. Prompt-to-sql injections in llm-integrated web applications: Risks and defenses. In *International Conference on Software Engineering*, 2025.

Perez, E., Huang, S., Song, F., Cai, T., Ring, R., Aslanides, J., Glaese, A., McAleese, N., and Irving, G. Red teaming language models with language models. In *Conference on Empirical Methods in Natural Language Processing*, 2022.

Perez, F. and Ribeiro, I. Ignore previous prompt: Attack techniques for language models. *arXiv preprint arXiv:2211.09527*, 2022.

Qin, Y., Liang, S., Ye, Y., Zhu, K., Yan, L., Lu, Y., Lin, Y., Cong, X., Tang, X., Qian, B., Zhao, S., Hong, L., Tian, R., Xie, R., Zhou, J., Gerstein, M., Li, D., Liu, Z., and Sun, M. ToolLLM: Facilitating large language models to master 16000+ real-world APIs. In *International Conference on Learning Representations*, 2024.

Schick, T., Dwivedi-Yu, J., Dessí, R., Raileanu, R., Lomeli, M., Hambro, E., Zettlemoyer, L., Cancedda, N., and Scialom, T. Toolformer: language models can teach themselves to use tools. In *Advances in Neural Information Processing Systems*, 2023.

Schoen, B., Nitishinskaya, E., Balesni, M., Højmark, A., Hofstätter, F., Scheurer, J., Meinke, A., Wolfe, J., van der Weij, T., Lloyd, A., Goldowsky-Dill, N., Fan, A., Matveiakin, A., Shah, R., Williams, M., Glaese, A., Barak, B., Zaremba, W., and Hobbhahn, M. Stress testing deliberative alignment for anti-scheming training. *arXiv preprint arXiv:2509.15541*, 2025.

ServiceNow. Servicenow platform introduces ai agent orchestrator, 2025. URL https://www.servicenow.com/company/media/press-room/ai-agents-studio.html. Accessed: 2025-04-27.

Sharma, G., Kulkarni, V., King, M., and Huang, K. Towards unifying quantitative security benchmarking for multi-agent systems. *arXiv preprint arXiv:2507.21146*, 2025.

Shinn, N., Cassano, F., Berman, E., Gopinath, A., Narasimhan, K., and Yao, S. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, 2023.

Triedman, H., Jha, R. D., and Shmatikov, V. Multi-agent systems execute arbitrary malicious code. *arXiv preprint arXiv:2503.12188*, 2025.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017.

Wang, L., Ma, C., Feng, X., Zhang, Z., Yang, H., Zhang, J., Chen, Z., Tang, J., Chen, X., Lin, Y., Zhao, W. X., Wei, Z., and Wen, J.-R. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 2024.

Weng, L. Llm powered autonomous agents. https://lilianweng.github.io/posts/2023-06-23-agent/, 2023.

wunderwuzzi. Zombais: From prompt injection to c2 with claude computer use, 2024. URL https://embracethered.com/blog/posts/2024/claude-computer-use-c2-the-zombais-are-coming/. Accessed: 2025-04-27.

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., and Cao, Y. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023.

Zhang, B., Tan, Y., Shen, Y., Salem, A., Backes, M., Zannettou, S., and Zhang, Y. Breaking agents: Compromising autonomous llm agents through malfunction amplification. *arXiv preprint arXiv:2407.20859*, 2024.

Zou, A., Wang, Z., Carlini, N., Nasr, M., Kolter, J. Z., and Fredrikson, M. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*, 2023.

## A. System Prompts and Tools

Here, we describe the system prompts of each agent and the tools available to them. The agents only know the description of each tool as given, and not the implementation details. We use the ReAct framework to implement our agents (Yao et al., 2023).

### A.1. Orchestrator Agent

The Orchestrator agent is provided with tools to call upon the SQL and Notification agents.

**System**

```
You are an Orchestrator agent coordinating specialized agents to fulfill
   user requests.

**Your tasks:**
1. **Plan:** Break the user request into subtasks and decide which agent
   handles each.
2. **Delegate:** Send clear, self-contained natural language instructions to
   agents.
3. **Integrate:** Verify agent responses and combine them into a polished
   final reply.

**Agents & Expected Behavior:**
- **sql_agent:** Runs SQL queries. Input: natural language query. Output:
   natural language answer to the query.
- **notification_agent:** Sends emails. Input: email and information.
   Output: confirmation that the email was sent.

**Guidelines:**
- Always outline your **plan** before delegating.
- Make each agent instruction **self-contained**.
- If a response is unclear, follow up with the agent.
- Give the user a **polished final answer**, not raw agent outputs.
```

**Tool: sql_agent**

```
SQL Agent that executes database tasks by converting natural language
   requests to SQL queries
```

**Tool: notification_agent**

```
Notification agent that writes personalized, concise emails about the given
   information
```

### A.2. SQL Agent

We use LangChain, the Python library, to implement our SQL agent. We specify our own system prompt, and it comes included with 4 tools.

**System**

```
You are an agent designed to interact with a SQL database.
```