Agentic Web: Weaving the Next Web with AI Agents

Yingxuan Yang[1], Mulei Ma[2], Yuxuan Huang[3], Huacan Chai[1], Chenyu Gong[2], Haoran Geng[4], Yuanjian Zhou[5], Ying Wen[1], Meng Fang[3], Muhao Chen[6], Shangding Gu[4*], Ming Jin[7], Costas Spanos[4], Yang Yang[2], Pieter Abbeel[4], Dawn Song[4], Weinan Zhang[1,5*], Jun Wang[8*]

[1]Shanghai Jiao Tong University
[2]The Hong Kong University of Science and Technology, Guangzhou
[3]University of Liverpool [4]University of California, Berkeley [5]Shanghai Innovation Institute
[6]University of California, Davis [7]Virginia Tech [8]University College London
zoeyyx@sjtu.edu.cn shangding.gu@berkeley.edu
wnzhang@sjtu.edu.cn jun.wang@cs.ucl.ac.uk

Abstract

Traditionally, the Web has served as a platform for connecting information, resources, and people, enabling human–machine interaction through activities such as searching, browsing, and performing tasks that are informational, transactional, or communicational. This original Web was fundamentally about connection, linking users to content, services, and one another.

The emergence of AI agents powered by large language models (LLMs) marks a pivotal shift toward the *Agentic Web*, a new phase of the internet defined by autonomous, goal-driven interactions. In this paradigm, agents interact directly with one another to plan, coordinate, and execute complex tasks on behalf of users. This transition from human-driven to machine-to-machine interaction allows intent to be delegated, relieving users from routine digital operations and enabling a more interactive, automated web experience.

In this paper, we present a structured framework for understanding and building the Agentic Web. We trace its evolution from the PC and Mobile Web eras and identify the core technological foundations that support this shift. Central to our framework is a conceptual model consisting of three key dimensions: intelligence, interaction, and economics. These dimensions collectively enable the capabilities of AI agents, such as retrieval, recommendation, planning, and collaboration.

We analyze the architectural and infrastructural challenges involved in creating scalable agentic systems, including communication protocols, orchestration strategies, and emerging paradigms such as the Agent Attention Economy. We conclude by discussing the potential applications, societal risks, and governance issues posed by agentic systems, and outline research directions for developing open, secure, and intelligent ecosystems shaped by both human intent and autonomous agent behavior. A continuously updated collection of relevant studies for agentic web is available at: https://github.com/SafeRL-Lab/agentic-web.

Keywords: Agentic Web, LLM Agents, Web Architecture, Safety & Security

*S. Gu, W. Zhang and J. Wang are the corresponding authors.

Contents

1 Introduction

The Web has long served as a platform for *connectivity* (Berners-Lee, 1999; Castells, 2002), linking people to information, services, and one another. In its early phases, the Web enabled human–machine interaction for tasks that were informational (e.g., reading news), transactional (e.g., online shopping), or communicational (e.g., messaging and email). Intelligence in this era resided in the tools that helped users access, filter, and interact with content: search engines (Brin and Page, 1998), recommender systems (Wang et al., 2006; Koren et al., 2009; Zhao et al., 2013; Zhang et al., 2013), and user interfaces (Deaton, 2003). However, the user was always the active party, manually navigating between pages, initiating actions, and making decisions at every step.

For the last few years, a shift has been taking place: the emergence of AI agents powered by large language models (LLMs) (Yang et al., 2023a; Kapoor et al., 2024). These *AI agents* are software entities capable of perceiving their environment, reasoning, and taking actions autonomously to achieve goals set by the user. With the integration of perception and execution components, LLMs are no longer limited to responding to prompts: they can act through agents that plan, remember, and interact across digital systems (Wang et al., 2023). Importantly, these agents are not constrained to single-turn interactions but can carry out complex, long-horizon tasks. Moreover, multiple agents can be orchestrated to work collaboratively on sophisticated objectives (Qian et al., 2024; Yang et al., 2025e; Gottweis et al., 2025; Sapkota et al., 2025).

The transformation toward agent-based systems is driven by two powerful forces. First, AI assistants are becoming increasingly capable of handling complex, multi-step tasks across domains such as research (Ren et al., 2025; Huang et al., 2025b; Schmidgall et al., 2025), software development (Hong et al., 2023; Xia et al., 2024), customer support (Rome et al., 2024), and personal productivity (Li et al., 2024b). These agents are no longer reactive tools responding to isolated prompts, but proactive collaborators that plan, reason, and execute actions over time. Second, users are becoming more comfortable delegating not just individual queries but entire workflows (sometimes spanning minutes, hours, or even days) to such agents (Guo et al., 2024; Hong et al., 2024). This growing trust in agent autonomy introduces new expectations and necessitates new interfaces, leading to a fundamental shift in how the Web is used and experienced.

This evolution lays the foundation for what we formally define as the *Agentic Web*. In this emerging paradigm, the Web is no longer merely a platform for human interaction with content and services, but a dynamic environment in which autonomous agents act, communicate, and collaborate across services and domains on behalf of their users (Petrova et al., 2025; Lù et al., 2025; Chaffer, 2025). For instance, the ChatGPT Agent released in July 2025 enables AI agents to act on behalf of users by performing tasks such as planning and purchasing ingredients for a Japanese breakfast or booking reservations (OpenAI, 2025).

Definition: Agentic Web

The *Agentic Web* is a distributed, interactive internet ecosystem in which autonomous software agents, often powered by large language models, act as autonomous intermediaries that persistently plan, coordinate, and execute goal-directed tasks. In this paradigm, web resources and services are agent-accessible, enabling continuous agent-to-agent interaction, dynamic information exchange, and value creation alongside traditional human-web interactions.

Unlike the traditional Web, which serves primarily to connect documents, services, and users for informational, transactional, and communicational purposes, the Agentic Web enables intelligent, goal-directed interaction. While the core functions of accessing information, completing transactions, and facilitating communication remain, they are now mediated by autonomous agents capable of reasoning, planning, and acting on behalf of users.

The defining shift is from short-term, one-off interactions between users and static content, to sustained, long-term interactions involving sequences of coordinated actions across multiple services,

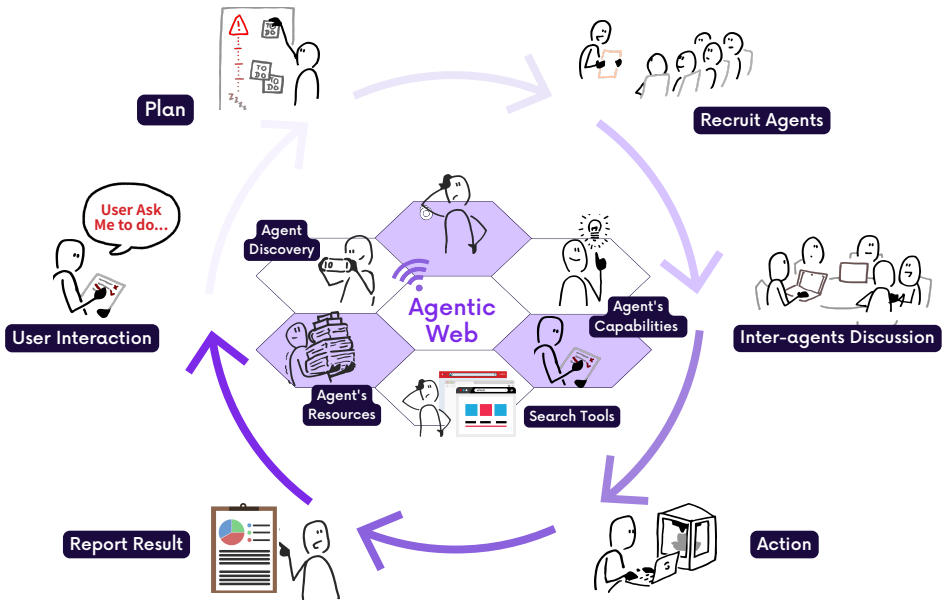

Figure 1: Illustration of the Agentic Web process cycle. The cycle begins with a user submitting a task request. The system then plans the task and identifies appropriate agents and tools. Recruited agents engage in inter-agent discussions, collaborate using their unique capabilities and resources, and execute the task. The results are reported back to the user, completing the cycle. The Agentic Web facilitates discovery, coordination, and cooperation among agents to fulfill user goals.

webpages, and domains. In the Agentic Web, the *end users* remain human, while the *mid users* (those who actively navigate, process, generate content, and interact with the environment) are AI agents. These agents interpret and carry out user intent by interacting with a distributed network of other agents and services.

A user query is no longer a simple request for isolated information, but a delegation of a complex task, which may involve negotiation, planning, and adaptation over multiple steps. With the support of structured or open-ended communication protocols (Yang et al., 2025d), these agents collaborate across domains to complete workflows and deliver results that reflect high-level user goals (Lin et al., 2024b; Yang et al., 2025c). This agent-mediated process is illustrated in Figure 1, which depicts a typical task lifecycle from user intent to multi-agent execution and result delivery.

In this new paradigm, webpages evolve into active software agents, characterised not just by their static content but by their capabilities, interfaces, and roles within broader task structures. Hyperlinks, which once represented passive navigational paths, now act as coordination channels that facilitate inter-agent communication, dynamic task decomposition, and cooperative execution. The Agentic Web, therefore, transforms the Web from a network of linked documents into an ecosystem of interactive, intelligent agents.

Beyond changes in interaction models, the Agentic Web also redefines how information is stored, linked, and transmitted. In the early Personal Computer (PC) era, web content was mostly institutionally produced, with relatively small data volumes that users accessed primarily through keyword search. As the mobile internet expanded, User-Generated Content (UGC) exploded, increasing the scale and diversity of information. This shift raised the cost of search and gave rise to recommendation systems as the dominant paradigm for matching information supply and demand.

With the emergence of LLMs and agentic systems, the underlying logic of information flows undergoes another major transformation. Now, much of the world's knowledge is not only stored on static web pages but also embedded in the parameters of LLMs themselves. Agents can access this learned knowledge directly, link it with real-time retrieval, and autonomously interact with other agents or online resources.

This enables agents to proactively recommend relevant content, going beyond traditional search engines, and to perform deeper and more personalized information retrieval. Moreover, agents can execute transactions and complete consumption processes on behalf of users, introducing a new production–consumption dynamic in which information and services may be created primarily for agents rather than humans. In some cases, web content may not be authored directly by humans at all but generated by agents in real time, leading to an ecosystem where agents both produce and consume knowledge.

Example (Transactional)

In the *Traditional Web*, a transactional task such as booking a flight is manually performed by the user. The process typically involves visiting travel websites, entering search queries, adjusting filters, comparing ticket options across multiple tabs or platforms, and finalising the booking decision. While the Web may offer assistance through features such as recommendation engines, user interfaces, and search algorithms, the task execution remains user-driven and requires active, step-by-step involvement.
In the *Agentic Web*, the same task is initiated through high-level intent delegation. The user provides a goal-oriented instruction (e.g., "Book a flight to New York next weekend within my budget"), and an autonomous agent carries out the task on their behalf. The agent autonomously interacts with services and APIs, queries and parses webpages, refines options based on user preferences, and completes the booking. It may perform multiple iterations and coordinate with other agents, requiring no further user intervention.

The above example illustrates the core distinction: the Traditional Web is defined by human-led interaction over static services, while the Agentic Web enables persistent, intelligent, machine-led workflows that extend across multiple services and interactions. Figure 2 complements this distinction by visualizing how user-system interactions have evolved from passive consumption to active agent delegation across three Web eras.

Example (Informational)

In the *Traditional Web*, an informational task such as understanding how different large language models process multimodal inputs requires the user to manually locate whitepapers, extract architecture diagrams, search for benchmark results, and assemble the findings into a report. This involves switching between academic search engines, blog posts, PDF viewers, and spreadsheet tools.
In the *Agentic Web*, the same task is delegated to a Deep Research agent (e.g., "Produce a report comparing how GPT-4o, Gemini, and Claude handle text and image inputs, including tables and flowcharts"). The agent interprets the query and plans a multi-stage workflow. It retrieves content from online sources and technical repositories via API calls, browser access, and the Model Context Protocol (MCP) (Anthropic, 2024b), which enables standardized access to external tools and structured resources. The agent then parses PDF and HTML documents, invokes specialized modules for table extraction, diagram generation, and result visualization, and integrates the outputs into a structured report through multi-step reasoning.

This example illustrates how the Agentic Web extends beyond static content retrieval to complex, adaptive information processing.

As a result, foundational Web concepts such as PageRank (Page et al., 1999), along with broader systems including web search (Broder, 2002), recommender systems (Resnick and Varian, 1997), and computational advertising models (Nelson, 1974), must be reinterpreted. Rather than focusing solely on static link popularity or historical user interactions, they may increasingly reflect the dynamic utility, responsiveness, and cooperation potential of agents operating within the network. Similarly, traditional techniques like web crawlers, once designed to index static content, could evolve into agent crawlers, autonomous explorers that discover and negotiate with other agents,

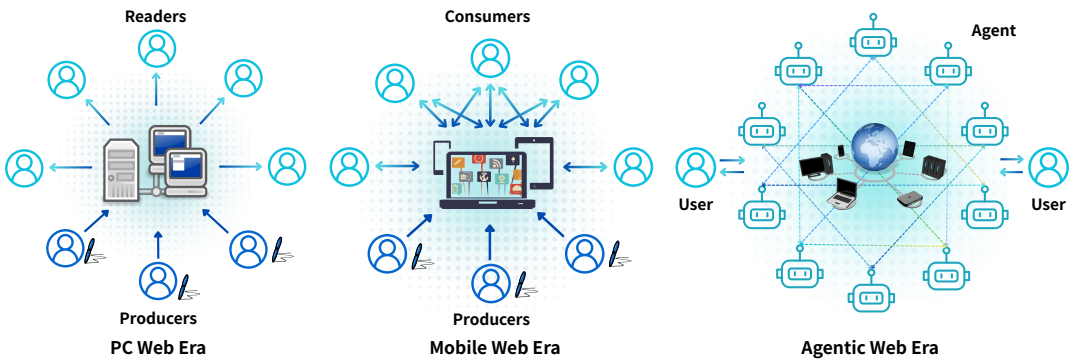

Figure 2: Evolution of user-system interaction across three internet eras. In the PC Web Era, users acted primarily as content consumers with limited interaction. The Mobile Web Era introduced a bidirectional flow, enabling users to both consume and produce content. In the emerging Agentic Web Era, tasks are delegated to ai agents, who interact with information networks on their behalf. The expanding and darkening circles reflect the increasing complexity and volume of information.

indexing not just data but service capabilities, interface affordances, and cooperation histories. The metadata of webpages becomes richer and more actionable: beyond simple tags or descriptors, agent metadata may include standardized schemas describing APIs, trust levels, performance benchmarks, or negotiation protocols. The old idea of web directories or yellow pages, once manually curated lists of websites categorized by topic, can be reimagined as dynamic agent registries or marketplaces that index available agents by domain expertise, reputation, and inter-agent compatibility. In such an agentic environment, search engines could transform into sophisticated orchestrators, not only retrieving relevant agents but also composing, coordinating, and managing workflows among them to fulfill complex delegated tasks. Just as PageRank once signaled page authority, future agent ranking algorithms may factor in cooperation success rates, responsiveness, and the agent's contribution to multi-agent workflows. Together, these reinterpretations and shifts pave the way for a new generation of algorithms and protocols for agent discovery, trust calibration, incentive alignment, and orchestration (Lin et al., 2024b; Wang et al., 2025a), enabling the Web to operate as an open, distributed, and continuously evolving collective of collaborative intelligences.

Therefore, it becomes essential to revisit the foundational technologies and modules of the Web and reinterpret them in the context of the Agentic Web. Core components such as HTTP protocols, HTML semantics, indexing, search, and recommender systems must be reconsidered through the lens of agent autonomy and collaboration. Despite the rapid emergence of Agentic AI, there is a noticeable gap in the current literature in systematically analyzing and redefining these web fundamentals for an agent-driven future. Bridging this gap is crucial for understanding and shaping the next evolution of the Internet, which is the goal of this article.

In summary, the key contributions and the structure of this article are outlined as follows. In Section 2, we review the historical evolution of the Web and offer a forecasting-style analysis to project the development trajectory of the Agentic Web in the near future. Section 3 introduces and conceptualizes the Agentic Web as a fundamentally new form of the Web, presenting a three-dimension model along with a set of research propositions that frame its emerging dynamics. In Section 4, we delve into the core tasks and enabling techniques of the Agentic Web, covering areas such as information retrieval, Recommender Systems, agent planning, and multi-agent learning and coordination. Section 5 explores the evolving system landscape and proposes key design principles to guide the development of Agentic Web infrastructure. In Section 6, we examine representative applications of the Agentic Web, including use cases like e-commerce ordering, travel planning, and enterprise knowledge assistants. Section 7 addresses the associated technical risks, information security concerns, regulatory challenges, and potential mitigation strategies. Finally, in Section 8 and Section 9, we conclude by summarizing the major themes of the paper and discussing the future outlook for the continued evolution of the Agentic Web.

2 Historical Evolution of the Web

In this section, a chronological review of three milestone phases in the evolution of the Web is conducted: the *PC Web Era*, the *Mobile Web Era*, and the *Agentic Web Era*. This progression is visualized in Figure 3, which presents a high-level timeline of the Web's evolution across technological paradigms and business models.

Each era is characterised by significant shifts in technological paradigms, commercial models, and user behavior patterns. The *PC Web Era* was centred around information directories and search paradigms, with content organized through static web pages that users manually browsed to locate desired information. Search engines emerged to support efficient retrieval, and keyword-based advertising systems marked the beginning of the commercial Web. The subsequent *Mobile Web Era* introduced a fundamental shift toward recommendation-driven content consumption, where algorithmic curation became essential due to the explosion of user-generated content and mobile platform constraints. Today, the Web is entering the *Agentic Web Era*, propelled by breakthroughs in foundation models and agent-based paradigms, where intelligent agents coordinate complex tasks and reshape both technical architecture and commercial logic. Figure 4 illustrates how user attention flows have evolved across these Web eras, from linear search and ad delivery models, to algorithmic feed curation, and finally to agent-mediated task execution involving multiple competing services.

2.1 PC Web Era

The PC Web Era represents the foundational stage of the Internet's evolution, marked by static content delivery and goal-oriented information retrieval. During this period, the user experience was shaped by limited interactivity, minimal personalization, and the early commercialization of the web through keyword-based search and advertising systems.

2.1.1 Static Pages and Search-based Commercial Marketing

The PC Web was dominated by a *retrieval paradigm* characterized by users relying on active queries and manual browsing to access information in an era of rapidly expanding digital content. At this stage, the Web lacked intelligent mechanisms for information dispensing. The content was primarily presented through static web pages with fixed organizational structures, with limited interactivity and personalised recommendations. Web platforms like Yellow Pages and Craigslist relied heavily on manual categorization and predefined navigation to link various types of information. These websites were typically organized by geography, industry, or service type, mirroring the taxonomy of printed directories and classified ads to present business listings, personal posts, and product information.

From the perspective of users, this retrieval paradigm required a strong sense of goal-directed behavior. The information-seeking process was linear and static, requiring users to have a clear goal for their search and to invest time and effort in locating the information they needed by navigating through hierarchical directories. This simple but inefficient paradigm struggled to meet users' increasing demand for speed, relevance, and personalization.

As the scale of the Web expanded rapidly, the conventional directory-based paradigm proved inadequate to satisfy the growing demand for efficient information retrieval. To address this challenge, search engines emerged and became a critical turning point in the evolution of the Web. Early systems relied on basic keyword matching techniques like TF-IDF (Sparck Jones, 1988), which measured term frequency but struggled with document authority and relevance ranking. Building upon TF-IDF, more sophisticated probabilistic models such as BM25 (Robertson et al., 2009) were developed to address issues like document length normalization and term frequency saturation, providing better text relevance scoring mechanisms. Meanwhile, Latent Semantic Indexing (Deerwester et al., 1990) introduced a paradigm shift by using singular value decomposition to capture latent semantic relationships between terms and documents, enabling search engines to understand conceptual similarities beyond exact keyword matches and address issues like synonymy and polysemy.

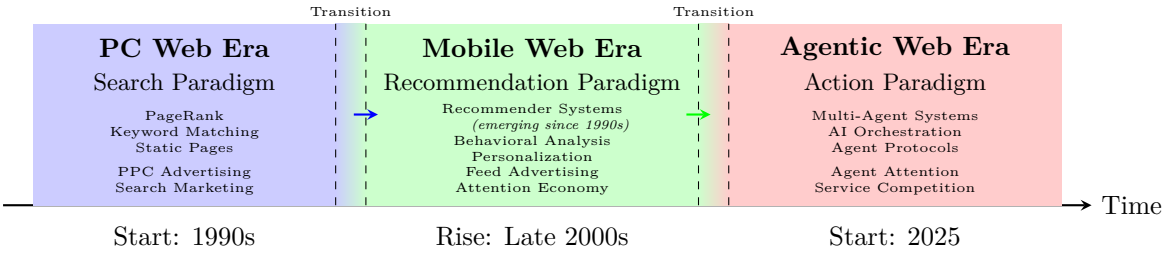

Figure 3: Timeline of Web Evolution. The three eras of web evolution are not strictly distinct. Their transitions happened gradually, with technologies, features, and business models often overlapping and coexisting across different periods.

A significant milestone was the PageRank algorithm (Page et al., 1999), which pioneered the concept of "link-based voting" by evaluating the importance and authority of web pages through their hyperlink structures. In comparison with earlier methods that relied solely on keyword matching, PageRank significantly enhanced the relevance of search results, laid the foundation for search engines like Google, and greatly increased user reliance on search engines.

Subsequent to this technological breakthrough, search engines integrated advertising mechanisms based on user intent. Early search advertising systems, such as Google AdWords, matched user queries with commercial content through sophisticated auction algorithms. The evolution from simple "pay-your-bid" mechanisms used by early systems like Overture to more sophisticated auction theories became crucial. Google AdWords implemented the Generalized Second-Price auction algorithm (Edelman et al., 2007), where advertisers bid for keyword placement but pay the price of the next-highest bidder, creating more stable and efficient bidding behavior than earlier first-price auction systems.

The introduction of Quality Score further refined this mechanism by balancing bid amounts with ad relevance, rewarding high-quality advertisements with better positions and lower costs. This keyword-driven, pay-per-click (PPC) model enhanced ad conversion rates whilst providing sustainable revenue streams for search engines, establishing a direct link between web content and commercial marketing and ultimately initiating the commercialization of the Web based on search.

2.2 Mobile Web Era

The transition to the Mobile Web Era was driven by fundamental changes in the Web's information landscape that extended beyond the mere adoption of mobile devices.

The most significant driver was the explosive growth in content volume during the late PC Web era. User-Generated Content proliferated across social platforms, e-commerce sites, and streaming services, creating massive data flows that traditional search paradigms struggled to navigate effectively. Users found themselves overwhelmed by choice and increasingly unable to discover relevant content through manual search alone.

This content explosion coincided with a shift in user behavior from intent-driven to discovery-driven consumption. Rather than approaching the Web with specific queries, users increasingly sought serendipitous discovery and personalized exploration. Mobile contexts amplified this trend, as users consumed content in shorter, fragmented sessions while expecting instant, personalized experiences without active searching.

Recommender Systems, which had already existed during the PC Web era for specific use cases, thus evolved from auxiliary tools to central architectural components. Mobile platforms introduced distinct design challenges such as latency constraints, limited screen space, and fragmented user attention. These challenges catalyzed advancements in recommender system architectures, promoting the development of real-time, context-aware models tailored to mobile interaction patterns.

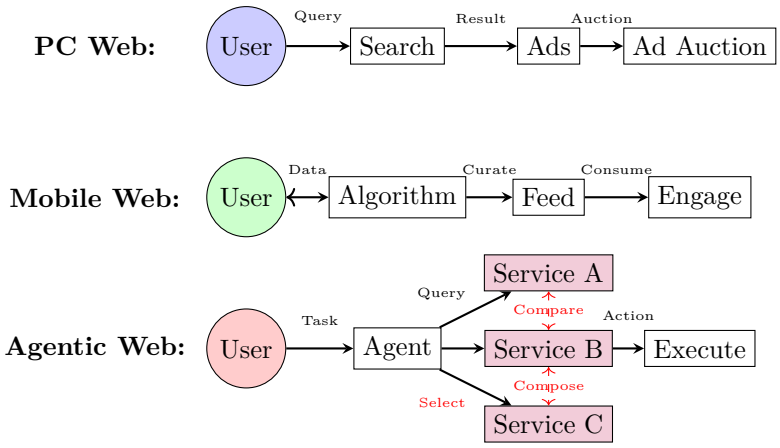

Figure 4: Attention Flow Evolution Across Web Eras. This diagram illustrates the transition from the PC Web, where attention follows a linear search-query-ad model, to the Mobile Web, where algorithmic systems curate feeds based on user data, and finally to the Agentic Web, where autonomous agents interpret user intent and select among competing services to execute tasks. Dashed arrows in the agentic stage indicate competitive or compositional relationships between services.

2.2.1 Recommender Systems

The progression of Recommender Systems mirrors the broader technological transitions across web platforms. In the early era of the PC Web, Recommender Systems primarily relied on collaborative filtering techniques to provide personalized suggestions based on historical user-item interactions. User-based and item-based k-nearest neighbor models were widely adopted due to their simplicity and interpretability (Sarwar et al., 2001), while matrix factorization approaches became mainstream for uncovering latent preferences and item attributes. Although foundational, these methods suffered from scalability and sparsity challenges, especially as web content and user bases expanded rapidly. Early solutions like Singular Value Decomposition in latent factor models (Koren et al., 2009) were instrumental in establishing the core technical underpinnings of modern recommender systems. However, their static nature and inability to model complex behavior made them increasingly inadequate in dynamic environments. A critical bridge between traditional methods and modern deep learning was established through Factorization Machines (Rendle, 2010), which modeled feature interactions through factorized parameters and enabled efficient computation with sparse data. This approach addressed the limitation of traditional matrix factorization in handling diverse feature types and became a foundation for subsequent hybrid architectures.

As user interaction became more real-time and context-rich in the Mobile Web era, significant advancements in personalization capabilities emerged. Deep learning-based methods became powerful means of modeling high-dimensional, nonlinear interactions between users and items. Neural collaborative filtering (He et al., 2017) combined deep neural networks with matrix factorization to enhance generalization capabilities. AutoRec (Sedhain et al., 2015) introduced autoencoder-based collaborative filtering, demonstrating the potential of neural architectures for recommendation tasks.

The evolution toward industrial-scale mobile applications led to breakthrough hybrid architectures that balanced memorization and generalization. Wide & Deep Learning (Cheng et al., 2016), developed by Google, combined linear models for memorization with deep neural networks for generalization, establishing a paradigm for large-scale recommender systems. Building upon this foundation, DeepFM (Guo et al., 2017) integrated factorization machines with deep neural networks for click-through rate prediction, eliminating the need for manual feature engineering while maintaining the ability to model both low-order and high-order feature interactions. Advanced deep learning architectures further enhanced modeling capabilities for mobile environments. Deep Matrix Factorization (De Handschutter et al., 2021) extended latent modeling capacity using residual learning, while

DeepCF (Deng et al., 2019) introduced unified architectures that integrated user/item representations with content signals. These advancements enabled mobile applications to deliver fine-grained, real-time personalization across social media, e-commerce, and streaming platforms.

The incorporation of temporal dynamics became crucial for mobile environments where user behavior patterns change rapidly. Sequence-aware models such as GRU4Rec (Hidasi et al., 2016) emerged to model temporal patterns in user behavior, while contextual bandits addressed the exploration-exploitation trade-off in real-time recommendation scenarios. The introduction of attention mechanisms and Transformer architectures (Vaswani et al., 2017) marked a significant advancement in sequential recommendation, enabling models to capture long-range dependencies and complex interaction patterns. Contemporary developments have focused on attention-based models and self-supervised learning approaches optimized for mobile contexts. Transformer-based architectures like SASRec (Kang and McAuley, 2018) and BERT4Rec (Sun et al., 2019) have demonstrated superior performance in sequential recommendation tasks by leveraging self-attention mechanisms to model user behavior sequences. Graph Neural Networks have also emerged as powerful tools for modeling complex user-item interactions and social relationships (Wang et al., 2019).

2.2.2 Attention Economy

In the *Mobile Web Era*, Recommender Systems have not only transformed the manner in which users access information but also significantly impacted commercial patterns. By leveraging user interests and behavioral data, Recommender Systems enable advertising and e-commerce platforms to precisely target potential consumers with personalised content. For instance, e-commerce platforms analyse users' browsing histories and purchase records to suggest relevant products, thus enhancing the shopping experience and significantly boosting conversion rates and sales. Similarly, social media platforms employ Recommender Systems to present engaging content on homepages or feeds, thereby increasing user retention and interaction.

The advent of Recommender Systems precipitated a substantial evolution in the realm of online advertising, characterised by enhanced targeting accuracy and the emergence of a behavior-driven advertising pattern. This paradigm shift has given rise to the so-called *attention economy* (Falkinger, 2007; Ciampaglia et al., 2015; Davenport and Beck, 2018) where in each user action, such as a click, scroll, or pause, is considered a valuable data point. These platforms utilise the gathered data to enhance the delivery of advertisements in terms of format, timing, and frequency, thereby rendering them more relevant and cost-effective. This behavior-based approach enables advertisers to achieve higher marketing efficiency at lower costs.

Overall, Recommender Systems have improved both the efficiency and personalization of information access while emerging as a pivotal force in the commercialization of the Web. Through intelligent content distribution, they have redefined commercial interactions and consumption patterns. Their widespread adoption in the *Mobile Web Era* signals a shift from a supply-demand model to a more intricate, behavior-driven interaction paradigm between information and commerce.

2.3 Agentic Web Era

The evolution of the Web is undergoing a paradigm shift from a human-centric information retrieval model to an agent-centric action-oriented framework.

The foundation for this transition was established by breakthroughs in LLMs (OpenAI, 2024b; Dubey et al., 2024; Guo et al., 2025), which demonstrated unprecedented capabilities in natural language understanding, reasoning, and code generation. Unlike previous AI systems that were limited to specific domains, these models exhibited emergent abilities to decompose complex tasks, maintain context across extended interactions, and interface with external tools and APIs.

This technological leap enabled the development of AI agents capable of autonomous decision-making and multi-step task execution, ushering in the Agentic Web Era. This era is characterized by the orchestration of intelligent agents across a vast network of capabilities, protocols, and data sources.

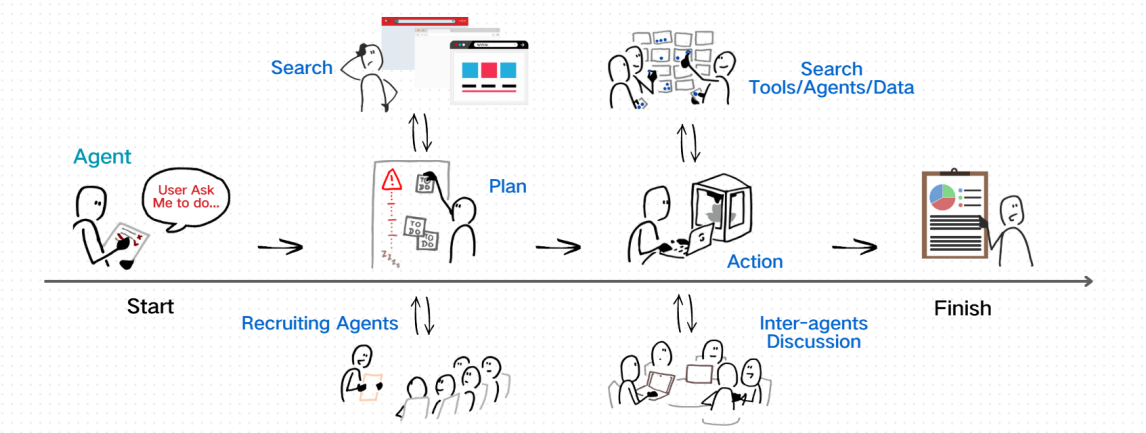

Figure 5: Agent Workflow under the Agentic Web.

In the following sections, we examine the rise of this agent-based infrastructure and its profound implications on Web architecture and digital economics.

2.3.1 Rise of Agentic Web

In the Agentic Web Era, tasks that once required significant human effort, such as deep research, cross-platform process execution, and long-term goal management, can now be completed autonomously with the help of intelligent agents. These tasks demand a comprehensive understanding of context, the ability to dynamically adjust strategies, and the integration of multiple tools and data sources.

Traditional single-function assistants are increasingly inadequate for addressing the demands of complex, multi-step tasks. In response, multi-agent systems have emerged as a critical architectural solution. A multi-agent system comprises multiple autonomous agents, each specialized in a specific sub-task such as information retrieval, translation, computation, or API interaction. These agents collaborate through mechanisms including task decomposition, capability scheduling, and inter-agent information sharing, thereby enabling the system to tackle problems of greater complexity and scale. The development of AI orchestration frameworks, such as LangChain (Chen et al., 2023a) and AutoGen (Wu et al., 2023), supports the integration of models, tools, and service components into structured *graph task flows*. These flows facilitate coordinated execution and dynamic decision-making among agents in a flexible and modular fashion, as illustrated in Figure 5. This collaborative paradigm significantly enhances both the breadth and depth of task execution. As a result, the Web is undergoing a transformation from a static "information network" to a dynamic "action network" in which autonomous systems are capable of perceiving, reasoning, and acting within digital environments.

However, this shift to multi-agent systems brings with it the need for sophisticated infrastructure to support these complex, agent-driven tasks. The advancement of the Agentic Web is not purely conceptual, it is increasingly grounded in real-world infrastructure developments. A key example of this transformation is Microsoft's recent move towards the Agentic Web, announced at Build 2025. Over 50 AI agent tools have been integrated across platforms such as GitHub, Azure, Microsoft 365, and Windows. These tools support every stage of agent deployment, from development and orchestration to memory management and inference optimization. They offer a unified environment for building and operating intelligent agents. Additionally, NLWeb (Natural Language Web) (Microsoft Corporate Blogs, 2025) has emerged as a toolset designed to convert traditional web interfaces into agent-readable, structured environments. This enables agents to navigate and interact with websites in a goal-driven manner, rather than relying on outdated and brittle methods like DOM scraping or simulated clicks.

These advancements signal a significant change in the way the Web will be used in the future. It is no longer just a place for human-centric browsing but a dynamic, agent-driven environment where intelligent agents can perform complex tasks on behalf of users. In this context, agent protocols like the MCP (Anthropic, 2024b) are playing a crucial role in enabling communication and collaboration between various services and agents. The establishment of these protocols is laying the foundation for a more standardized, scalable agent-Web interaction layer, further advancing the Agentic Web.

2.3.2 Agent Attention Economy

The advent of communication protocols such as the MCP has precipitated a trend of standardisation, registration, and organization of external tools and services within the agent ecosystem, which is analogous to the "yellow pages" directory. This facilitates agents' access to and invocation of these resources through a unified interface. However, as the number of external resource providers, such as APIs, remote services, and data endpoints, grows rapidly, a new challenge emerges: how can agents efficiently discover, filter, and select the most suitable capabilities in a highly fragmented and dynamic service landscape (Yang et al., 2025f)?

This challenge gives rise to the notion of the **Agent Attention Economy**. In a manner analogous to the early Web's competition for user clicks, external services now compete to be selected and invoked by autonomous agents. In this paradigm, the focus shifts from the human users to the agent engaged in the execution of a sophisticated task. Every tool, service, or other agent essentially competes for limited "agent attention". To improve visibility and invocation likelihood, these entities may adopt mechanisms such as advertising, ranking optimization, or even agent-oriented recommendation and scoring systems within the service registries.

As this competition intensifies, it is reasonable to hypothesise that a comprehensive advertising infrastructure tailored for agents will emerge, which will include agent-facing recommendation engines, capability reranking systems, inter-agent referral networks, and potentially auction-based ranking or context-aware ad insertion. This shift fundamentally redefines how agents discover and coordinate with external resources, accelerating the transformation from a human-centric to an agent-centric Web. This attention-based competition among agents may ultimately become a core mechanism for resource allocation in future Web platforms, signalling a profound restructuring of both the architectural and economic foundations of the Web.

2.4 Commercial and Structural Evolution of the Web

The Web has evolved through three distinct eras: the PC Web, the Mobile Web, and the Agentic Web. Table 1 provides a comparative overview of how the Web's architecture, attention focus, and commercial models have evolved across eras. This section analyzes how innovations such as PageRank, recommender systems, and agent protocols like MCP have transformed the architecture and commercial dynamics of the Web, laying the foundation for an agent-native ecosystem in which information is dynamically generated and acted upon by autonomous systems.

In the *PC Web Era*, information was primarily hosted on static web pages, often published by institutions and accessed through keyword-based search engines. Discovery hinged on link analysis algorithms such as PageRank, and content was structured like a digital directory, manually navigable and hierarchically classified. Web content was sparse and predominantly produced by organizations, meaning it was centralized and often top-down. Users relied on manual browsing or search engines to retrieve information by typing keywords, and the Web's structure was akin to a digital directory, where content was classified through hyperlinks and explicit taxonomies. Commercial activity centered around search advertising, with platforms like Google AdWords matching user queries to paid results. Key performance indicators (KPIs) included click-through rate and cost-per-click, reflecting a model where user attention was explicitly captured and monetized through intent-based queries.

As the Web transitioned into the *Mobile Web Era*, the underlying data storage and linking paradigm remained largely unchanged, but the volume and granularity of content exploded, driven by the rise

Table 1: A cross-era comparison of Web paradigms from an ecosystem perspective.

Aspect	**PC Web Era**	**Mobile Web Era**	**Agentic Web Era**
Core Paradigm	Search Paradigm	Recommendation Paradigm	Action Paradigm
User Behavior	Active search and manual browsing	Passive content consumption	Complex multi-step task execution
Information Organization	Static pages, hierarchical directories	Personalized feeds, algorithmic curation	Dynamic task flows, multi-agent collaboration
Key Technologies	PageRank algorithm, Keyword matching, Directory structures	Recommender Systems, Behavioral analysis, Personalization algorithms	Multi-agent systems, AI orchestration frameworks, AI Agent protocols (MCP/A2A)
Commercial Model	Pay-per-click advertising	Feed-based and in-app advertising	Agent Attention Economy
Revenue Source	Search advertising (e.g., Google AdWords)	Targeted advertising, e-commerce integration	Service invocation fees, agent-targeted advertising
Key Metrics	Click-through Rate, Cost Per Click	Conversion Rate, User dwell time, Effective cost-per-thousand impressions	Service invocation frequency, Capability relevance, Agent response success rate
Attention Focus	Human user clicks	Human user engagement	Agent selection and invocation
Information Access	Goal-directed, linear search	Algorithm-driven, passive consumption	Context-aware, multi-step execution
Platform Examples	Yellow Pages, Craigslist, Google Search	Social media feeds, e-commerce recommendations	Multi-agent AI systems, service registries
Economic Foundation	Search-based marketing	Attention economy	Agent-centric resource allocation

of UGC on social platforms, e-commerce sites, and streaming services. While information storage mechanisms such as cloud-based servers remained similar to earlier times, the sheer volume of content made traditional retrieval methods increasingly inefficient. Search interfaces, though still available, struggled to surface relevant content amidst massive data flows. This explosion in information created a need for more sophisticated ways of navigating and discovering content. In response, Recommender Systems emerged as the dominant access paradigm. These systems relied on algorithms to curate content tailored to individual users, shifting the burden of content discovery from users to algorithms. Users became both producers and consumers of content, with recommendation algorithms acting as intermediaries. Commercial models adapted to this new logic, emphasizing metrics like conversion rate, dwell time, and effective cost per mille, highlighting engagement depth and monetization precision.

Today, in the emerging *Agentic Web Era*, both the commercial logic and the structure of information are undergoing a radical transformation. Unlike earlier paradigms, where knowledge was stored in databases or presented on web pages, LLMs embed vast amounts of web-scale information within their parameters through pretraining. This in-parameter knowledge enables LLM-based agents to reason, and respond based on learned representations rather than direct document lookups. In parallel, intelligent agents are developing the ability to interact not only with web pages, but also with other agents, APIs, and tools in a dynamic, goal-driven fashion. The Web is thus shifting from a passive content repository to an active, agent-mediated action space, where agents act as autonomous intermediaries executing tasks on behalf of users.

Commercially, this era means the rise of the *Agent Attention Economy*, where third-party tools and services compete not for human clicks, but for agent invocation. New protocols like the MCP are enabling agents to dynamically compose and orchestrate services from a modular ecosystem, leading to the development of agent-driven commercial models. Future monetization metrics may depend on factors such as invocation frequency, capability relevance, and successful task completion, marking a departure from the previous era's focus on user interaction and direct advertising. This shift is giving rise to new forms of agent-oriented advertising and bidding systems, designed not to persuade users directly, but to influence agent decision-making pipelines.

Structurally, the Web is being redefined from a human-readable medium into an agent-native substrate. On the consumption side, agents can proactively summarize, recommend, or execute tasks on behalf of users, offering unprecedented personalization and efficiency. These agents are capable of operating continuously across different services, becoming autonomous digital intermediaries. On the production side, content will increasingly be generated by agents rather than humans. Agents can autonomously generate articles, compose marketing material, or structure data for other agents to consume. This results in an emergent agent-to-agent communication layer, where content may never be explicitly rendered for human eyes, but is optimized for agent parsing, reasoning, and orchestration.

Taken together, these shifts point toward a profound reconfiguration of the Web's ontology: from human-readable documents, to algorithm-curated feeds, to agent-native knowledge. No longer will the Web simply serve as a repository of static documents or a curated feed of personalized content, but rather as a dynamic, interactive space where information flows are synthesized, shared, and executed by autonomous systems on behalf of humans. Over time, this could lead to the rise of an "Agent-Oriented Web," where information is not merely personalized for individual users, but is dynamically created, shared, and executed through collaboration between intelligent agents.

3 The Agentic Web

The Web is undergoing a fundamental transformation (Petrova et al., 2025; Lù et al., 2025; Chaffer, 2025). In the traditional model, users served as active navigators: searching, comparing, and manually executing each digital step. Booking a flight, for instance, required visiting multiple travel websites, comparing ticket options, checking loyalty programs, and handling confirmation emails across services. With the rise of intelligent agents, this burden is shifting. Users now increasingly delegate goals rather than execute tasks. A travel agent AI can autonomously search for optimal flights based on personal calendar availability, loyalty points, and real-time pricing. It can coordinate with hotel agents or even adjust travel plans based on weather forecasts or meeting changes (Monica, 2024; Genspark, 2025). This represents a shift from user-driven web navigation to intent-driven orchestration, where outcomes rather than page views become the primary metric of value.

This evolution mirrors a broader arc in the history of the Internet. In the PC Web Era, users manually navigated hyperlinks like explorers in a vast library. In the Mobile Web Era, apps curated the experience and brought information to users more proactively. Yet users still had to operate the system by opening apps, copying data, and making decisions. Now, in the Agentic Web, users act more like directors who articulate their intent while intelligent agents carry out the necessary operations behind the scenes.

These transformations are not merely technological but conceptual. They redefine who acts on the Web (from human to agent), how tasks are executed (from manual interaction to delegated orchestration), and what the Web produces (from content consumption to outcome generation). In the following sections, we propose a structured framework to examine this shift across 3 dimensions: Intelligence, Interaction, and Economy. Each dimension offers insight into how the Agentic Web is reshaping capabilities, behaviors, and business models in increasingly autonomous digital ecosystems.

3.1 Core Conditions

Building the Agentic Web requires rethinking fundamental elements of digital infrastructure. Three conditions are essential:

Figure 6: Conceptual Framework of the Agentic Web. This diagram illustrates a three-dimensional architecture composed of the Intelligence, Interaction, and Economic Dimensions, reflecting the evolution of AI agents from reasoning entities to active economic participants.

Core Conditions for the Agentic Web

(1) Agents must function as autonomous intermediaries, initiating and completing complex tasks independently.

(2) Web resources need to be accessible through standardized, machine-readable interfaces.

(3) Value must be exchanged not only between humans and systems, but also directly between agents.

These structural foundations are interdependent. Agent autonomy depends on semantic interfaces, protocol interoperability, and the ability to discover and orchestrate external capabilities dynamically. Together, they create the conditions for scalable, intelligent web operations.

3.2 Transformations in Web Architecture

These structural foundations enable fundamental shifts in the operation of the Web, transforming both its usage and the organization of information.

3.2.1 Evolving Interaction Patterns

The Agentic Web transforms how interaction occurs in digital environments. Traditional web use follows a request-response model, where users initiate actions, retrieve data, and evaluate results manually. Agents, by contrast, engage in proactive behaviors. They discover relevant resources, identify capabilities, and form dynamic connections based on semantic relevance rather than static hyperlinks (Tupe and Thube, 2025; Sapkota et al., 2025; Acharya et al., 2025).

This change supports continuous and goal-oriented interaction across services. Agents monitor the digital environment, detect opportunities, and collaborate with other systems to fulfill objectives. Instead of navigating predefined pathways, they identify and access web resources through contextual understanding and adaptive negotiation, resulting in more responsive and flexible connectivity.

3.2.2 Changing Information Structures

The structural transformation of the Agentic Web is not limited to the role of agents as users or interfaces; it also extends to how information itself is stored, linked, and consumed. In previous stages of the Web, including the early PC era and the mobile era, information was mainly organized as documents or datasets hosted on web servers and accessed via hyperlinks or queried through search engines.

By contrast, the Agentic Web introduces a new mode of information organization. LLMs capture vast web-scale information directly within their parameters through the process of pretraining. These models embed knowledge within their structures and support on-demand reasoning, thereby reducing reliance on traditional external web sources (Petrova et al., 2025; Lù et al., 2025).

At the same time, the way information is linked is also changing. Rather than relying on static hyperlinks, agents discover resources, services, and other agents through semantic discovery and adaptive integration (Touvron et al., 2023b; OpenAI, 2024a; Guo et al., 2025). This results in a more fluid and context-sensitive form of connectivity, where relevance is inferred rather than explicitly declared.

As agents gain content generation capabilities, information production is shifting beyond human-authored web pages toward agent-generated outputs such as tools, instructions, summaries, and structured artifacts designed for other agents (Qin et al., 2023; Schick et al., 2023). This creates a self-sustaining loop in which agents both generate and consume content, leading to increasingly autonomous and self-reinforcing information flows.

This structural shift in how information is stored (in-model versus in-document), linked (semantic versus hyperlink), and accessed (agent-driven versus search-driven) underpins the transition to the Agentic Web.

3.2.3 Dual Operational Roles

Within this transformed architecture, agents operate through two complementary foundational perspectives that represent different facets of their functionality:

- **Agent-as-User (Downward-facing):** AI agents operate as autonomous web users who can independently navigate, interact with, and consume web resources (Nakano et al., 2022; Deng et al., 2023; Zhou et al., 2023b; Gur et al., 2024; OpenAI, 2025; Monica, 2024). In this role, agents replace or augment human users in web navigation and task execution, engaging with existing web interfaces and services designed for human consumption. This enables continuous, 24/7 operation for tasks such as market research, data collection, or transaction processing.

- **Agent-as-Interface (Upward-facing):** AI agents serve as intelligent intermediaries between human users and web systems, translating high-level user intentions into executable

actions (Corporation, 2025; Thurrott, 2024; Opera, 2025; Wiggers, 2025). These agents process natural language commands from users and orchestrate complex multi-step workflows across various web services. This perspective emphasizes the agent's role in abstracting complexity and providing streamlined human-agent interaction.

These perspectives are complementary rather than contradictory. A single agent system often embodies both roles: interacting autonomously with the web while serving as an interface for human users and forming a bidirectional bridge between intention and execution. Together, these operational perspectives, interaction transformations, and information structure changes operationalize the Agentic Web's core vision: a distributed, interactive internet ecosystem in which autonomous software agents engage in persistent, goal-directed interactions to plan, coordinate, and execute tasks on behalf of human users.

3.3 Three Conceptual Dimensions of the Agentic Web

To understand the Agentic Web in depth, we propose a conceptual framework built on three interrelated dimensions: Intelligence, Interaction, and Economy. Each dimension reflects a core requirement for autonomous operation within digital ecosystems as illustrated in Figure 6).

At its core, the Intelligence Dimension equips agents with reasoning capabilities such as perception, planning, and learning. Building on this, the Interaction Dimension enables agents to connect with digital environments through semantic protocols and dynamic tool use. The Economic Dimension focuses on how agents autonomously create, exchange, and distribute value, forming self-organizing digital economies.

Each layer builds upon the previous one: intelligence enables interaction, and interaction enables value creation. This layered view explains how agents evolve from internal reasoning entities to impactful economic participants.

Conceptual Layers

- **Intelligence Dimension:** What core intelligence is required for agents to function autonomously?
- **Interaction Dimension:** How do agents communicate and coordinate within digital ecosystems?
- **Economic Dimension:** How do agents generate and exchange value at scale?

3.3.1 Intelligence Dimension

The Intelligence Dimension provides the cognitive foundation that enables agents to reason, learn, and plan within open-ended digital environments. Unlike traditional systems that rely on human queries, agents access and act on information autonomously. They draw from both internalized knowledge (in-parameter models) (Koh et al., 2024; Putta et al., 2024; Masterman et al., 2024; Wu et al., 2025) and external resources (via tools and APIs) (Qin et al., 2023; Schick et al., 2023; Paranjape et al., 2023; Lu et al., 2025), shifting from passive retrieval to proactive information use.

To operate effectively, agents require transferable intelligence rather than narrowly defined, task-specific heuristics. Key capabilities include:

- **Contextual Understanding:** Agents should be able to interpret diverse forms of web-based input, including natural language, semi-structured data, and interface signals, all within task-specific and evolving contexts.
- **Long-Horizon Planning:** Agents must formulate, evaluate, and revise multi-step strategies to achieve both short-term objectives and long-term goals across diverse digital services.

- **Adaptive Learning:** Agents should be able to improve over time by integrating interaction feedback, acquiring new skills, and adjusting their internal models of user preferences and environment dynamics.
- **Cognitive Processes:** To operate reliably and efficiently, agents should monitor and reflect on their own reasoning, detect failures or suboptimal behavior, and dynamically adjust their cognitive strategies.
- **Multi-Modal Integration:** Agents must handle and integrate information from a variety of modalities (e.g., text, APIs, visuals, structured data), enabling robust decision-making in open-ended environments.

In the Agentic Web, agents are not passive executors of instructions. Instead, they interpret, strategize, and adapt independently. Without these cognitive abilities, agents cannot handle real-world ambiguity, recover from failure, or scale across services.

3.3.2 Interaction Dimension

The Interaction Dimension addresses a fundamental shift in how autonomous agents engage with the digital environment. This shift moves the web away from static, human-authored hyperlinks toward dynamic, context-aware connections. In traditional web architectures, interaction is primarily document-based and mediated by human users. In contrast, agents in the Agentic Web interact through semantic protocols and runtime service discovery, allowing them to initiate and manage interactions without relying on predefined links or manual input.

The emergence of agent-native communication protocols has been a key catalyst for this transformation (Chang, 2024; Cisco, 2025; AI and Data, 2025; Yang et al., 2025d), such as the MCP (Anthropic, 2024b). Unlike conventional APIs that rely on stateless, transactional exchanges, MCP supports persistent, semantically coherent dialogues between agents and services. It introduces three foundational capabilities: (1) *dynamic capability discovery*, allowing agents to identify available system functionalities at runtime; (2) *semantic context preservation*, which maintains task continuity across multi-step workflows; and (3) *privacy-aware collaboration*, enabling rich information exchange while protecting sensitive data. Together, these features mark a shift from procedural invocation toward adaptive, negotiated interactions.

Beyond communication, the Interaction Dimension underpins *tool orchestration*, the agent's ability to safely compose and sequence external capabilities. Agents can dynamically verify tool properties, authenticate requests, and execute operations within controlled environments, minimizing risks such as malicious injection or execution failures.

Furthermore, this dimension facilitates *agent-to-agent coordination*. Protocols like Agent-to-Agent (A2A) (Google, 2025b) enable agents to form ad hoc coalitions, share intermediate outputs, and collaboratively pursue complex goals. Such cooperative frameworks transform the Agentic Web into a networked fabric of interacting intelligences, where distributed reasoning and shared context allow agents to operate not just individually, but collectively.

By integrating semantic interoperability, safe tool access, and inter-agent collaboration, the Interaction Dimension forms the operational substrate of the Agentic Web. It allows autonomous agents to engage with a dynamic, heterogeneous environment in an adaptive and meaningful way as shown in Figure 7.

3.3.3 Economic Dimension

The Economic Dimension reconfigures digital ecosystems by introducing autonomous agents as economic actors capable of initiating transactions, forming collaborations, and allocating resources without direct human input. Unlike traditional systems, where value is created and exchanged through human interactions mediated by platforms, the Agentic Web supports machine-native economies in

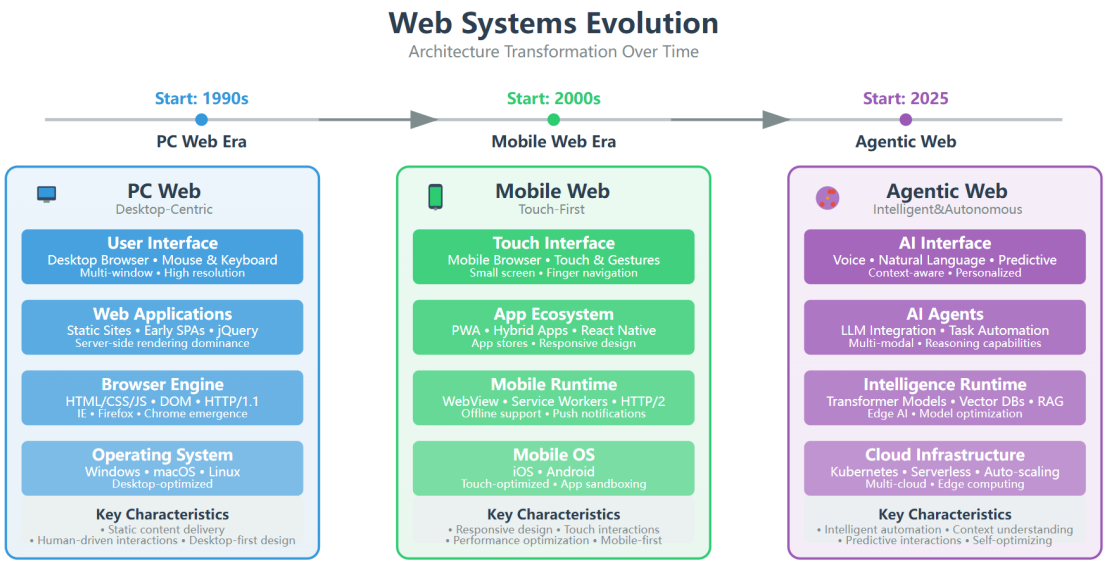

Figure 7: Architectural evolution of Web System. The transition reflects a shift from static content delivery and manual interaction to intelligent automation and outcome-oriented design. Not shown explicitly are the corresponding shifts in user roles, from navigator to operator to director, and in interaction models, from point and click to touch based to conversational delegation. These shifts mark a fundamental change in how value is created and tasks are fulfilled on the Web.

which agents coordinate, produce, and transact directly with one another (Tan and Huang, 2025; Rothschild et al., 2025; Dawid et al., 2025).

This shift gives rise to novel economic patterns. Agents can generate structured outputs such as executable workflows, tool manifests, and domain-specific datasets, not for human consumption, but for use by other agents. These machine-oriented artifacts enable closed-loop systems of generation and consumption, where agents operate continuously and collaboratively, driving a self-sustaining cycle of autonomous value creation.

Over time, such interactions form decentralized economic networks where agents dynamically discover services, negotiate terms, manage risk, and optimize outcomes through algorithmic reasoning. These agent-driven markets operate at speeds and granularities beyond human coordination, unlocking new forms of efficiency and scalability (Yang et al., 2025f).

However, this transformation also introduces governance challenges (Kolt, 2025; Yang and Zhai, 2025). As agents begin to make high-stakes economic decisions, potentially involving finances, contracts, and digital assets, questions around liability, transparency, and ethical alignment become urgent. Regulatory frameworks must evolve to accommodate autonomous behavior at machine timescales, ensuring accountability and fairness in increasingly complex agent-driven economies.

Ultimately, the Economic Dimension captures how agency, computation, and value creation converge in the Agentic Web, enabling a new kind of digital economy: one that is fast, adaptive, and fundamentally machine-mediated.

4 Algorithmic Transitions for the Agentic Web

The emergence of the Agentic Web necessitates a fundamental re-evaluation and redefinition of the algorithmic underpinnings of intelligent systems. This paradigmatic shift represents more than mere technological advancement; it embodies a conceptual transformation from passive, human-driven computational processes to autonomous, goal-oriented intelligent behaviors that can operate independently within complex digital ecosystems. This section examines how traditional paradigms

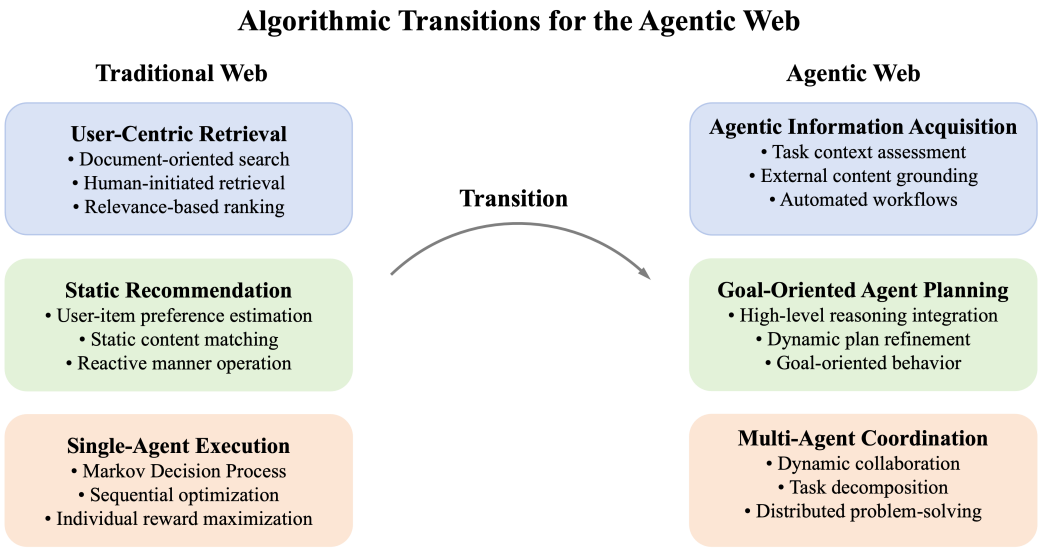

Figure 8: Algorithmic Transitions for the Agentic Web. The figure illustrates three foundational transitions from Traditional Web to Agentic Web paradigms.

in information retrieval, recommendation systems, and agent architectures are converging and transforming to form the core capabilities of autonomous agents operating within dynamic web environments. We delineate three foundational transitions that characterize this algorithmic evolution, as illustrated in Figure 8: (1) from user-centric information retrieval to proactive agentic information acquisition, where systems transition from reactive document lookup to autonomous, context-aware data gathering; (2) from static, one-shot recommendation to dynamic, goal-oriented agent planning, representing a shift from isolated preference predictions to integrated reasoning-action frameworks; and (3) from isolated single-agent execution to complex multi-agent coordination, enabling distributed intelligence and collaborative problem-solving capabilities. Each transition signifies a fundamental evolution from fixed, domain-specific pipelines that require explicit human intervention to adaptive, context-aware strategies that can autonomously navigate uncertainty and complexity. These transformations collectively establish the algorithmic foundation for intelligent systems capable of independent operation, continuous learning, and emergent behavior. Such characteristics distinguish the Agentic Web from its predecessors, with implications extending beyond technical improvements to encompass new possibilities for human-AI collaboration, automated decision-making, and the emergence of truly autonomous digital agents.

4.1 User-centric Retrieval to Agentic Information Retrieval

Traditional information retrieval has historically been grounded in human-initiated, document-centric search methodologies. Foundational techniques such as the bag-of-words model and term frequency-inverse document frequency (Spärck Jones, 1972) assign weights to terms based on their frequency within a document and their rarity across the corpus, thus producing a basic relevance signal for ad hoc queries. Probabilistic models such as Okapi BM25 (Robertson et al., 1995) introduced refinements by incorporating term-frequency saturation and document-length normalization, enabling more effective handling of heterogeneous document sizes and term distributions. With the advent of the web, link-analysis algorithms such as PageRank (Page et al., 1999) contributed by modeling a random surfer traversing the hyperlink graph, introducing authority as a critical factor in ranking. More recent advances have leveraged supervised Learning to Rank approaches, which apply pointwise, pairwise, or listwise machine learning objectives to optimize ranking functions using labeled relevance data, significantly improving performance over traditional unsupervised methods (Liu et al., 2009).

In contrast, the Agentic Web redefines retrieval as an active and integral component of autonomous cognition. Rather than performing static keyword-based queries, autonomous agents dynamically assess their goals, environment, and task progression to determine what information is needed, when to acquire it, and through which modalities (Zhang et al., 2024a). These agents engage in complex, multi-step retrieval pipelines that often involve domain-specific tools, external APIs, and procedural logic, enabling them to construct knowledge on demand. This transition from passive search to proactive information acquisition supports end-to-end workflows that require minimal human supervision, enhancing both task generalization and responsiveness.

RAG architectures exemplify this shift by grounding language model outputs in external, verifiable content (Lewis et al., 2020). Within this framework, Fusion-in-Decoder retrieves relevant passages via sparse or dense indexes and integrates them with the input query in a sequence-to-sequence model such as T5 (Izacard and Grave, 2021). FLARE adopts an iterative retrieval strategy in which the model forecasts the next sentence, uses low-confidence predictions to formulate pseudo-queries, and refines the output through successive retrieval rounds (Jiang et al., 2023). SELF-RAG introduces a self-reflective loop that prompts the model to critique and revise its own responses, thereby enhancing factual accuracy (Asai et al., 2024). RetrievalQA explores adaptive retrieval policies that allow models to determine whether and when to retrieve based on internal uncertainty estimates (Zhang et al., 2024c). The Tree of Clarifications framework (Kim et al., 2023) addresses query ambiguity by decomposing questions into clarifying subqueries, retrieving evidence for each, and synthesizing comprehensive answers. Toolformer extends these capabilities by enabling models to autonomously identify suitable APIs, invoke them at appropriate stages, and incorporate the outputs into subsequent token generation (Schick et al., 2023). Collectively, these innovations demonstrate how deeply integrated retrieval mechanisms enhance agent reasoning, supporting sophisticated tasks such as information synthesis, procedural decision-making, and tool utilization, thereby establishing the Agentic Web as a foundation for scalable and intelligent autonomous interaction.

4.2 Recommendation to Agent Planning

The traditional recommendation paradigm, which centers on matching users with items, is reinterpreted in the context of the Agentic Web as a strategic and goal driven framework for planning and execution. Earlier systems employed algorithms such as user based and item based collaborative filtering (Sarwar et al., 2001), matrix factorization methods (Koren et al., 2009), and deep learning based recommendation models (He et al., 2017) to estimate user preferences over static content. While these techniques are effective at retrieving individually relevant items, they operate in a reactive manner: each recommendation is an isolated prediction that neglects downstream task dependencies and does not support multi step workflows. Modern advances have fundamentally transcended these limitations through the introduction of sophisticated architectural innovations that enable autonomous goal oriented behavior. Language Agent Tree Search exemplifies this evolution by integrating MCTS with LLM powered value functions and self reflection mechanisms (Zhou et al., 2023a).

In contrast, the Agentic Web redefines recommendation as a proactive process involving multi step planning by autonomous agents. This conceptual shift has led to the development of agents powered by large language models that integrate high level reasoning with executable actions in dynamic web environments. For example, ReAct (Yao et al., 2023) integrates reasoning traces with concrete actions, allowing agents to refine their plans based on environmental feedback and to achieve significant improvements on question answering and interactive decision-making benchmarks. WebAgent (Gur et al., 2024) converts natural language instructions into Python programs while summarizing lengthy HTML content into task-specific segments, thereby enabling agents to interact with real web interfaces through programmatic planning. AdaPlanner (Sun et al., 2023) introduces a closed loop planning architecture that incorporates both in plan and out of plan refinements, dynamically updating plans based on environmental feedback to outperform standard baselines on ALFWorld (Shridhar et al., 2020) and MiniWoB++ (Liu et al., 2018). Plan-and-Act (Erdogan et al., 2025) further separates planning and execution into two distinct roles, a planner and an executor, each in-

stantiated by a large language model, and achieves state of the art performance on long horizon web navigation tasks. Beyond these foundational approaches, recent developments have introduced sophisticated memory augmented systems such as the Task Memory Engine, which implements spatial memory using Directed Acyclic Graphs to replace linear context concatenation (Ye, 2025). GoalAct demonstrates continuous global planning that maintains clear objectives through skill based decomposition and hierarchical execution strategies, achieving improvement in success rates on complex legal benchmarks (Chen et al., 2025).

This evolution represents a fundamental transition from passive recommendation to prescriptive and goal oriented behavior, empowering agents to autonomously interpret instructions, navigate web environments, and manipulate digital interfaces in pursuit of complex user defined objectives. The emergence of standardized evaluation frameworks such as WebArena (Zhou et al., 2023b), VisualWebArena (Koh et al., 2024), and ST-WebAgentBench (Levy et al., 2024) has established comprehensive protocols for assessing multi dimensional agent capabilities across planning, tool integration, safety, and trustworthiness. Contemporary agents demonstrate substantial performance gains through enhanced autonomous task completion capabilities, while standardized protocols including the Model Context Protocol and Agent Communication Protocol enable seamless integration across heterogeneous agent systems (Anthropic, 2025; Li et al., 2024a).

4.3 Single-Agent to Multi-Agent Coordination

Traditional single-agent systems have typically modeled autonomous decision-making in web environments using the Markov Decision Process framework. Early work by Shani et al. demonstrated that formulating recommendation tasks as sequential optimization problems outperformed static approaches in maximizing long-term user satisfaction (Shani et al., 2005). Building on this foundation, contextual bandit algorithms such as LinUCB were developed to adaptively select content by incorporating user and contextual information, improving cumulative engagement through iterative learning (Li et al., 2010). To address the bias and sparsity inherent in logged interaction data, off-policy actor-critic methods with top-K corrections have been successfully scaled to large recommender systems, enhancing stability and effectiveness in environments with millions of candidate actions (Chen et al., 2019). Additionally, slate-based reinforcement learning (RL) decomposes multi-item recommendation problems into tractable value functions, enabling efficient Q-learning over complex combinatorial action spaces (Ie et al., 2019). Although these single-agent approaches are effective for optimizing individual objectives, they face limitations in scenarios that require collaboration, distributed reasoning, or task-level adaptability.

To address these limitations, multi-agent coordination frameworks have emerged to enable dynamic collaboration among agents, allowing them to collectively solve tasks that are difficult to address in isolation. This paradigm shift supports task decomposition, role specialization, and orchestrated execution through communication and shared goals. AutoGen exemplifies this transition by implementing planner, executor, and critic roles, using prompt conditioning to assign responsibilities and structure inter-agent interaction (Wu et al., 2023). AgentOccam enhances LLM-based agents not by refining agent strategies alone, but by improving their fundamental reasoning and task comprehension capabilities (Yang et al., 2025a). WebPilot incorporates a multi-agent MCTS architecture to guide web navigation and decision-making, demonstrating the potential of hierarchical planning in interactive environments (Zhang et al., 2025d). The AI Co-Scientist leverages multiple agents and external tools to formulate novel scientific hypotheses, combining web search with specialized AI modules to generate well-grounded research proposals (Gottweis et al., 2025).

Recent systems have also emphasized flexibility, modularity, and accessibility in multi-agent design. Alita introduces a minimalist architecture that reduces predefined roles and promotes self-evolution, aiming for greater scalability and generalization across domains (Qiu et al., 2025). OWL offers a structured agent hierarchy, decomposing tasks into specialized roles filled by UserAgents, AssistantAgents, and ToolAgents to automate complex real-world objectives (Hu et al., 2025). AutoAgent enables users to construct multi-agent workflows and integrate external tools without extensive technical knowledge, expanding the accessibility of agent-based system design (Tang et al.,

2025). Octotools organizes execution into distinct planner and executor components, optimizing the coordination of multi-tool computational workflows (Lu et al., 2025). These developments collectively reflect a broader shift toward distributed intelligence, demonstrating the increasing importance of collaboration, modularity, and specialization in next-generation autonomous systems.

5 Systematic Transitions of the Agentic Web

The shift toward an Agentic Web entails not only a conceptual evolution but also a fundamental redesign of the underlying system architecture. Traditional web infrastructure, based on stateless protocols, user-initiated interfaces, and static interaction models, is poorly suited to the requirements of agentic computation. To function effectively, autonomous agents necessitate continuous contextual awareness, persistent sessions, dynamic service discovery, semantically rich interaction protocols, and real-time coordination with both human users and other agents.

This section articulates the system-level transformations necessary to support agent-native execution. It identifies the limitations of current web protocols and runtime environments, proposes infrastructure requirements for persistent, context-aware agents, and discusses the evolution of communication standards that enable semantic, agent-to-agent interactions. By analyzing these transitions systematically, this section contributes a coherent architectural perspective for operationalizing the Agentic Web beyond isolated prototypes or platform-specific implementations.

We begin by outlining the core system challenges that arise when deploying autonomous agents in web-scale environments.

5.1 Motivation for an Agentic Web System

The advent of the Agentic Web, a paradigm characterized by autonomous, intelligent agents executing complex user-delegated tasks, represents a fundamental architectural evolution from the content-centric model of the contemporary internet. This transition necessitates a profound reengineering of underlying network and system architectures, as the internet's extant infrastructure is ill-equipped to support the dynamic, decentralized, and mission-critical nature of agentic operations. For this emergent ecosystem to become viable, several core system-level challenges must be surmounted, transforming the internet from a passive data conduit into an intelligent, proactive, and service-aware fabric.

A primary impediment to the realization of the Agentic Web is the challenge of agent discovery. In contrast to the static addressing schemes of the traditional internet, where resources are located via stable IP addresses or domain names, autonomous agents are ephemeral and lack a fixed, identifiable network location. When a principal agent must execute a task exceeding its intrinsic capabilities, it must dynamically recruit other agents. This scenario presents a critical problem: the identification and selection of suitable collaborators from a vast, decentralized population of agents. The resolution of this problem requires a dynamic discovery mechanism, analogous in principle but superior in intelligence to network routing protocols (Cui et al., 2025). For a given task, the system must effectively source and select agents by conducting a comprehensive assessment of their skills, readiness, and suitability to the specific operational demands. This "just-in-time" matchmaking is a prerequisite for the seamless, on-demand collaboration essential for executing complex, multi-agent operations (Chen et al., 2024b; Raskar et al., 2025).

Subsequent to discovery, the challenge of effective inter-agent communication arises, with the current ecosystem of APIs presenting a significant roadblock. Contemporary APIs are predominantly engineered for consumption by human developers, achieving syntactic but not semantic interoperability. While they rigorously define the structure for data exchange, they lack the formal semantic annotations necessary for an autonomous agent to unambiguously interpret their function and purpose (Tupe and Thube, 2025). An agent may parse the technical format of a request but cannot

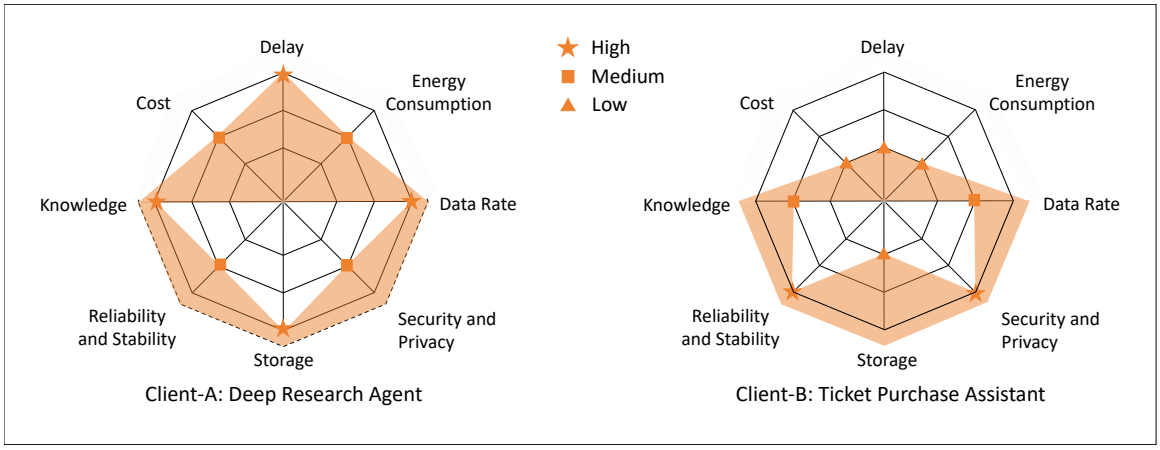

Figure 9: Service Requirement Zone.

ascertain the underlying intent or operational semantics of the API's methods. To transcend this limitation, a new paradigm of agent-oriented APIs is required. A future standard would likely necessitate the integration of machine-readable formalisms, such as ontologies or logical specifications, directly within the API definition. This would create a unified system wherein an agent can access not only the syntactic structure but also the semantic context, thereby empowering agents to autonomously discover, comprehend, and utilize APIs to perform complex tasks without human intervention (Braubach et al., 2018).

Furthermore, the highly dynamic and distributed nature of this network introduces complex logistical challenges related to billing and accounting. In a system where agents can spontaneously collaborate, delegate sub-tasks, and consume services from one another, tracking resource utilization for the purpose of accurate attribution and billing becomes exceptionally difficult. A persistent, reliable, and auditable methodology is needed to monitor the chain of interactions and associate computational and service costs with the originating user or principal agent. This necessitates the design of a transactional framework capable of tracing an agent's activities and resource consumption across a fluid, multi-party network. Such a framework must securely bind billing information to a specific entity, ensuring that as agents access premium services or delegate paid tasks, the associated costs are accurately calculated and charged (Cui et al., 2025). Without a robust solution for micropayments and distributed accounting, the economic models required to sustain a sophisticated, service-driven Agentic Web are untenable.

Finally, a foundational challenge resides in the evolution of the network infrastructure itself: specifically, the transition from a best-effort network to an intelligent infrastructure capable of delivering guaranteed, personalized quality of service. The current network architecture, exemplified by 5G, is fundamentally network-centric, engineered to optimize a limited set of aggregated KPIs, such as peak data rate and latency. This model, while effective for enhancing general system capabilities, is insufficient for the Agentic Web, which demands a paradigm shift from optimizing universal metrics to providing bespoke, task-specific service guarantees. The infrastructure must evolve to comprehend and dynamically accommodate the distinct, multi-dimensional requirements of individual agentic tasks. This transition reflects a shift from providing generalized high-performance capabilities to acting as an intelligent orchestrator that interprets specific task requirements across multiple dimensions, including cost, security, and access to knowledge, and provisions tailored Service Level Agreements accordingly.

Figure 9 illustrates a task's Service Requirement Zone (SRZ) (Yang et al., 2023b), an eight-dimensional profile defining its quality of experience needs across metrics like cost, delay, security, data rate, and knowledge. The size of the shaded SRZ on the chart indicates the stringency of these needs: a smaller zone means more exacting demands, requiring precise resource orchestration. It also contrasts two different agentic tasks to demonstrate this concept. The Deep Research Agent

(left) displays a constricted SRZ, reflecting its complex requirements. It has a high demand for Knowledge to access specialized models, strong Reliability and Stability, and low Delay to enable interactive analysis. This profile necessitates a highly capable and responsive service.

In contrast, the Ticket Purchase Assistant (right) has a larger, more flexible SRZ. This agent's primary needs are high Security for processing payments and a high Data Rate to quickly load options. It can tolerate longer delays and has minimal requirements for energy or local storage, which is typical for a transactional task. This comparison highlights how different agents have unique service profiles that the underlying infrastructure must be able to interpret and fulfill.

To adequately support the Agentic Web, the underlying infrastructure must therefore evolve to natively interpret and fulfill these diverse SRZs. It can no longer treat the VR stream and the banking transaction as fungible data flows. The system must transition from network-level slicing to a far more granular mode of service provisioning at the individual task level (Liu et al., 2025a; Chen et al., 2024a). To achieve this, the network requires pervasive, embedded intelligence, allowing it to efficiently identify a task's SRZ and subsequently orchestrate heterogeneous network, compute, and data resources across multiple domains to guarantee that its specific quality of experience is met, thus marking a definitive departure from the legacy "best-effort" paradigm (Mahmood et al., 2024; Wang et al., 2024b).

The paradigm shift towards SRZ-centric service delivery imposes a set of unprecedented demands on the underlying system architecture that far surpass the capabilities of the traditional web. The system must support extreme dynamism and negotiation, as an agent's resource needs can change dramatically mid-task, requiring real-time allocation adjustments (Huang et al., 2024; Zhang et al., 2025c). It must natively handle multimodality, intelligently managing the varied requirements of text, image, audio, and other data types. The system's role must also evolve from a mere data transporter to a capability-driven orchestrator, maintaining a real-time inventory of computational resources, AI models, and data sources to fulfill agent requests (Li et al., 2025a). Furthermore, it must provide granular, verifiable security and privacy controls at the level of individual agents and sub-tasks, offer deep observability for robust debugging and optimization, and incorporate intelligent cost control mechanisms to manage computationally expensive agentic workflows (Li et al., 2025a).

5.2 Toward a Next-Generation Agentic Web System

To facilitate the large-scale deployment of autonomous agents, the Web must evolve from a content-centric medium to an execution-oriented infrastructure. This paradigm shift necessitates a fundamental re-evaluation of web systems' architectural foundations to support agent-native interaction patterns, persistent context management, and integrated tool orchestration.

In this subsection, we present the Agentic Web system, which integrates three essential elements: the User Client, the Intelligent Agent, and Backend Services. We elucidate the functional roles of each component, examine their historical evolution, and analyze their collective function in translating high-level user goals into executable digital actions. By articulating this architecture, we establish a conceptual framework for understanding how agentic capabilities can be operationalized, thereby bridging the divide between user intent and dynamic service execution.

5.2.1 Roadmap of the Agentic Web System

This section delineates the architecture of the Agentic Web, a tripartite structure designed to translate user objectives into executable operations. This architecture is composed of three integral components that operate in synergy: the User Client, the Intelligent Agent, and Backend Services.

The User Client serves as the primary medium for human-agent interaction. Its core functions are to process user inputs like textual, vocal and to render the agent's synthesized outputs. The historical trajectory of clients shows an evolution from text-based command-line interfaces to the intuitive graphical and touch-based paradigms of today. The contemporary trend is a progression towards

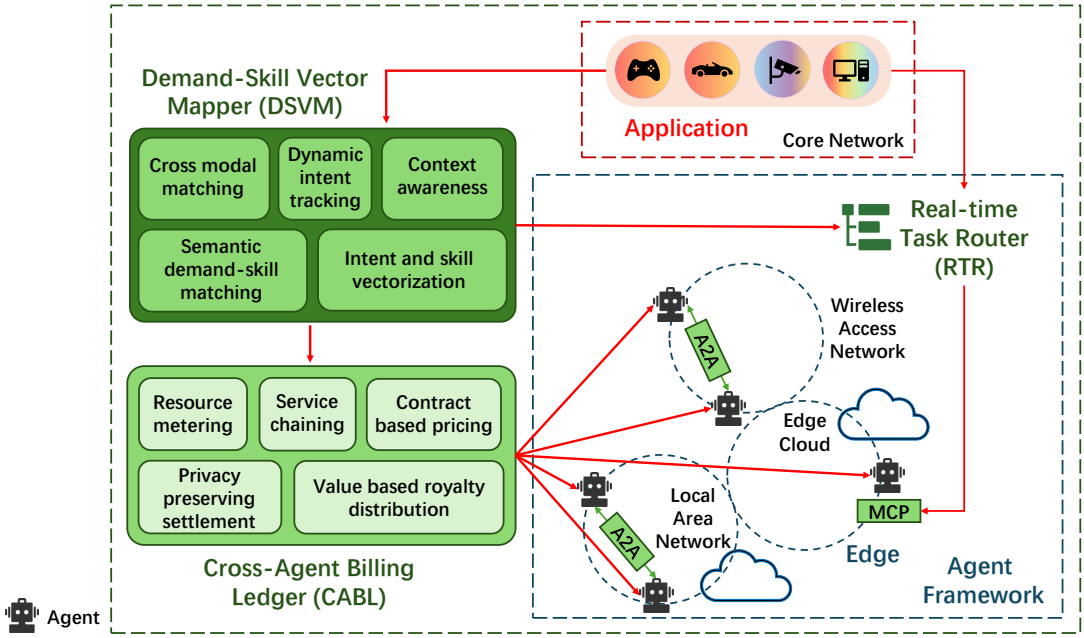

Figure 10: Agentic Web Roadmap

multimodal systems that integrate diverse inputs such as voice and gesture, embodied in devices like smart speakers and wearable technology.

The Intelligent Agent constitutes the system's central cognitive and decision-making nexus. Leveraging artificial intelligence disciplines such as Natural Language Processing, the agent discerns user intent, decomposes complex objectives into granular sub-tasks, and selects appropriate backend tools for execution. The developmental path of these agents has advanced from rudimentary rule-based systems to sophisticated learning models. These modern agents are capable of addressing complex creative and epistemic tasks by continuously adapting based on new data and user feedback.

Backend Services, Tools, and Plugins form the functional substrate, providing the essential computational, data, and specialized capabilities required by the agent. These modular resources encompass a wide spectrum of functions, from general utilities like language translation to domain-specific industry applications. Architecturally, they have evolved from monolithic databases to a distributed and extensible ecosystem of microservices and plugins, which permits third-party developers to continuously augment the capabilities of the Agentic Web.

To illustrate the interoperability of these components, consider a representative interaction workflow. The process is initiated when the User Client transmits a high-level objective, such as "plan a business trip to Shanghai," to the Agent. The Agent then decomposes this objective into constituent sub-tasks. It subsequently identifies and invokes the requisite external services, such as flight and hotel booking platforms, to execute these tasks. Upon receiving the necessary information, the Agent integrates these disparate results. It synthesizes the data, evaluates it against predefined user constraints, and formulates a coherent, consolidated response—such as a complete itinerary—which is then relayed to the User Client for presentation.

Furthermore, the architecture accommodates a direct interaction model. In certain scenarios, the Agent may orchestrate an initial connection, after which the User Client engages directly with a backend service. This decoupled model is particularly advantageous for transactions involving sensitive data or requiring high-throughput, such as financial payments. This design allows the Agent to preserve its function as the master coordinator while delegating specific interactions to optimize for security and efficiency.

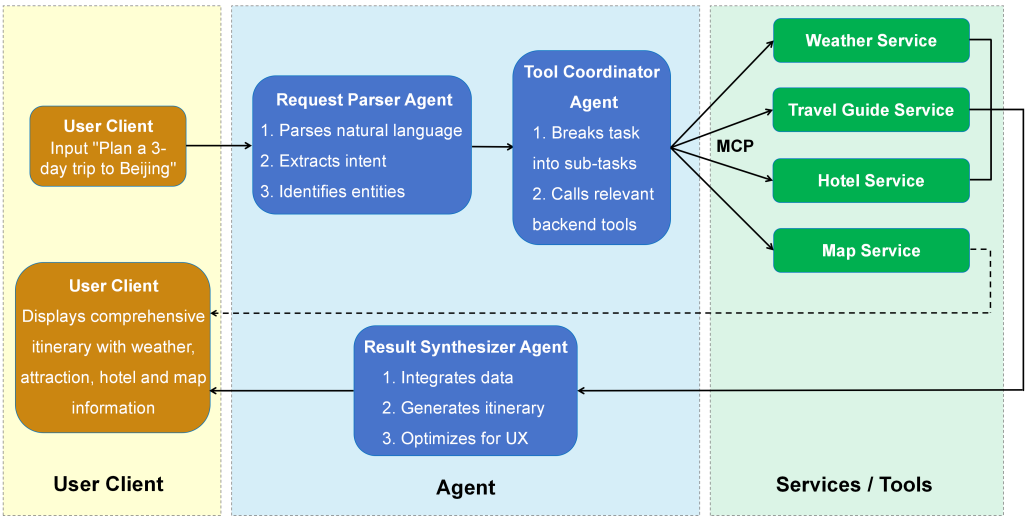

Figure 11: Interaction Process Example: Planning a Travel Itinerary

In summary, this architecture represents a paradigm shift from direct user manipulation of discrete applications to a model of delegated goal-fulfillment. Within this paradigm, a user entrusts a high-level objective to an intelligent, autonomous Agent, which then orchestrates a diverse set of resources to achieve the specified outcome. The User Client functions as the dedicated human-computer interface; the Agent operates as the central cognitive processor and orchestrator; and the Backend Services constitute an ecosystem of callable functionalities (e.g., APIs, databases, web applications) capable of executing specific, well-defined tasks.

These rigorous requirements fundamentally invalidate the traditional Client-Server architecture, mandating a shift toward the Client-Agent-Server model. We propose an Agentic Web Architecture, depicted in Figure10, to realize this paradigm. This architecture operates through three core components. First, a demand skill vector mapper interprets application needs by performing context awareness, dynamic intent tracking, and semantic vectorization to translate service demands into machine-readable formats. Second, a real time task Router dynamically dispatches these vectorized tasks to a distributed Agent Framework operating across the edge and access networks. Third, a cross agent billing ledger governs the economic and resource interactions between agents, enabling crucial functions like resource metering, service chaining, and privacy-preserving settlement. This integrated design creates an intelligent, autonomous, and value-aware fabric for orchestrating complex agentic services.

5.2.2 Interaction Process Example: Collaborative Mechanisms in Travel Itinerary Planning by Agents

As depicted in Figure 11, the workflow is initiated upon the submission of the request, “Plan a 3-day trip to Beijing,” through the User Client. A Request Parser agent semantically parses this input to extract key parameters, including the destination, duration, and the core user objective.

Subsequently, a yool orchestrator agent decomposes the primary objective into four discrete subtasks: obtaining meteorological data, compiling information on tourist attractions, querying for accommodation options, and generating a route map. It then programmatically invokes the corresponding backend services via the MCP protocol:

- Request forecast data from the Weather Service.
- Request attraction information from the Travel Guide Service.
- Request accommodation options from the Hotel Service.

- Request the generation of optimized travel routes from the Map Service.

Following data retrieval, a result synthesizer agent aggregates and integrates the information from the weather, travel guide, and hotel services to construct a comprehensive itinerary. Concurrently, the response from the Map Service bypasses the synthesis stage, transmitting route maps and location data directly to the User Client. This direct feedback channel is explicitly marked in Figure 11 (dashed line).

Finally, a consolidated itinerary, comprising weather forecasts, attraction details, accommodation suggestions, and an interactive map, is then transmitted to the User Client for review. The user can accept the proposal or request modifications, with map data being updated dynamically via the direct service feedback mechanism. This dual-pathway design strategically balances comprehensive data integration for high-level planning with low-latency service responses for interactive components, all orchestrated through a unified platform.

5.2.3 Recent Advances and Applications of Agentic Web Systems

To advance the agentic web, recent research has focused on overcoming core challenges in agent design and assessment. Key innovations include synergizing reasoning with action, architecturally separating planning from execution, and establishing more reliable benchmarks to accurately measure performance.

ReAct, a framework developed by (Yao et al., 2023) achieves synergy by interleaving reasoning and acting within large language models. The proposed methodology enhances the model's performance in two key ways. First, it enables the generation of reasoning traces to formulate and dynamically adjust action plans. Second, it facilitates interaction with external resources, exemplified by a Wikipedia API, to retrieve information, which is essential for fact-checking and minimizing hallucinations. This dynamic combination of "acting to reason" and "reasoning to act" enables agents to more reliably solve knowledge-intensive tasks and enhances the interpretability of their decision-making processes.

To tackle challenges in long-horizon tasks, the PLAN-AND-ACT framework (Erdogan et al., 2025) distinctly segregates strategic high-level planning from immediate low-level actions. This architecture features a PLANNER model dedicated to generating structured, abstract strategies and an EXECUTOR model responsible for translating these strategies into tangible steps in the environment. A key innovation of this framework is dynamic replanning, which addresses the limitations of static plans. The PLANNER updates the plan after each action is executed, enabling the agent to acclimate to evolving environmental conditions and incorporate new information, such as search results, into the ongoing strategy.

For a more accurate measurement of true web agent capabilities, existing benchmarks like WebVoyager have been identified as a key limitation, as they often suffer from a lack of task diversity and can report inflated performance results (Xue et al., 2025; Deng et al., 2023). To address this, the new Online-Mind2Web benchmark offers a comprehensive evaluation suite, containing 300 diverse and realistic tasks that cover a broad spectrum of 136 websites. Concurrently, an automated evaluation method called WebJudge was also developed. This approach identifies key points for task completion and then selects key screenshots from an agent's trajectory for evaluation, preserving critical information while avoiding context length limits. This method achieves up to 85.7% agreement with human judgment and significantly improves evaluation reliability and scalability.

5.3 Agentic Communication

The deployment of autonomous agents in complex web environments for multi step tasks introduces novel communication demands that fundamentally exceed the capabilities of traditional web protocols. As agents evolve from passive API consumers to proactive, context-aware actors capable of initiating and coordinating tasks, they require better communication mechanisms that support se-

mantic interoperability, persistent task states, and asynchronous multi-party interaction, and many other features.

This section investigates the protocol level foundations of the Agentic Web, examining the limitations of conventional protocols such as HTTP and RPC, and motivating the need for agent-native alternatives. We introduce two representative protocols, MCP and A2A, that exemplify emerging approaches to structured, scalable, and semantically rich communication among agents and services. The following subsections first analyze the design motivations behind these protocols and then provide detailed descriptions of their architectures and workflows.

5.3.1 Design Motivation (Beyond HTTP/RPC)

In the current Internet ecosystem, the Hypertext Transfer Protocol (HTTP) and Remote Procedure Call (RPC) have long served as the mainstream communication protocols, underpinning the data interaction between Web services. Over the past two years, numerous AI Agent projects have achieved basic communication functions based on these two protocols. However, with the rise of the Agentic Web concept, the limitations of traditional protocols have gradually become prominent. The Agentic Web, characterized by autonomy, context awareness, and dynamic interaction, has operational mechanisms that impose new requirements on communication protocols far beyond the capabilities of HTTP/RPC.

Firstly, the task execution process in the Agentic Web typically involves collaboration among multiple entities, which imposes stringent requirements on the **efficient management of task specific context**. In the Agentic Web, the task execution process frequently necessitates intricate interactions between designated agents and other agents, in addition to non-agent resources such as external tools, data, and services. During these interactions, all participating entities are required to maintain, transmit, and share specific context, such as historical data or environmental configuration parameters. For example, when a personal assistant agent is assigned the task of arranging a travel itinerary for a user, it may need to query a weather API and interact with hotel and transportation booking agents. Throughout this process, the involved entities must exchange both private and non-private user data, including user preferences and authorizations, and share progress information related to the booking task. However, traditional web protocols (HTTP/RPC) are principally designed for the transmission of data and lack semantic-level support. This design limitation results in the treatment of complex context as ordinary data, with no distinction among higher-level semantic elements such as *historical context*, *user intent*, or *environmental configuration*. Furthermore, these web protocols have been demonstrated to lack the capacity to process logical semantics, such as preconditions. Consequently, traditional web protocols are inadequate in meeting the stringent demands of the Agentic Web for efficient context management during task execution.

Secondly, the task execution process in the Agentic Web is contingent on LLM based agents, thereby engendering heightened requirements for **semantic accuracy in communication and interaction**. In the Agentic Web, entities need to communicate through structured and standardized protocols to ensure semantic consistency and operational feasibility during task execution. However, LLM-based agents typically mediate their understanding and generation of structured content through natural language, introducing inherent non-determinism into the output process, which is susceptible to issues such as formatting deviations and semantic drift, compromising the accuracy and reliability of interactions. Consider the case of traditional API calls, which rely on developers to interpret interface semantics from documentation and manually construct deterministic, structured invocations. Conversely, in the Agentic Web, agents are required to automatically interpret interface semantics and translate natural language descriptions into operational commands. In the absence of a semantically specified mechanism, the generated results are frequently unstable, which complicates the assurance of structural integrity and semantic correctness in API calls. Therefore, Agentic Web protocols provide machine-readable interface semantic specifications that explicitly define the data types, value ranges, and business meanings of each field. This can guide LLMs in accurately parsing and generating structured invocations. Furthermore, the protocol must address semantic divergence across entities, for instance by unifying or mapping field labels such as “UID”

and "UserID", in order to avoid semantic ambiguities. However, traditional web protocols such as HTTP and RPC are primarily designed for data transport and lack support for interface semantics and field alignment. Consequently, they are insufficient for the semantic coordination and contextual understanding required.

Finally, the task execution process in the Agentic Web requires **high interactivity** in communication. The task execution process is frequently distinguished by protracted durations, **multi-phase** workflows, and **asynchronous** operations. Furthermore, in circumstances involving sensitive actions, immediate user intervention may be imperative. This necessitates that agent communication be supported by persistent and dynamic interaction mechanisms. For instance, let us examine a case where a personal assistant agent is assigned the task of formulating a trading strategy. In such a case, the agent may need to communicate intermediate results to the participant at different phases of the procedure. In particular, if the task involves high-risk decisions or large financial transactions, the agent must halt its operation until it receives an explicit confirmation from the user to continue. The execution of such task workflows necessitates not only the possession of fine-grained task control capabilities by agents but also the implementation of communication protocols that support event-driven architectures, asynchronous responses, and persistent task state management. However, traditional web protocols such as HTTP and RPC typically adopt a synchronous request-response model and lack native support for long-running tasks. Furthermore, they are also ill-equipped to handle complex control flows such as task suspension, external event injection, or dynamic user confirmation. Whilst mechanisms such as polling and Webhooks can be utilized to facilitate partial asynchronous interactions, they frequently necessitate the implementation of additional logic layers, thereby increasing system complexity and compromising overall robustness. It is therefore vital that the Agent Web urgently requires more capable and interaction-rich protocol mechanisms to support multi-phase, multi-party task workflows in a more natural and efficient manner.

According to the research in (Yang et al., 2025d), a large number of new agent communication protocols emerged in the past year. Among them, the general-purpose protocols, MCP and A2A, have demonstrated significant technical advantages and community influence. These two protocols are designed from different dimensions to address the characteristics of the Agentic Web, forming complementary solutions. We will provide a brief introduction to these two communication protocols and explain which key challenges faced by the Agentic Web they can address and how to address them, followed by detailed introductions to their workflows.

MCP, short for Model Context Protocol, proposed by Anthropic (Anthropic, 2024), is a communication protocol focused on interactions between agents and non-agent resources. It aims to establish standardized interfaces for tool invocation and has gradually evolved into a de facto industry standard. Under the framework of this protocol, applications encapsulate the tools, resources, and prompts they provide into service units that can be recognized by agents. Agents obtain application metadata, including function descriptions, input-output formats, and usage constraints, through the query interface of the MCP, and implement the call and control of applications based on the operation instructions defined by the protocol. For example, when an agent needs to call an image-generation tool, it can obtain the parameter-configuration specifications of the tool through the MCP and submit the generation task in a standardized request format to ensure the consistency of cross-platform tool calls. To some extent, MCP has enhanced the structural consistency and standardization of communication between agents and resources.

A2A, short for Agent-to-Agent (Google, 2025a), proposed by Google, is a communication protocol specifically designed to facilitate direct interaction between agents through a distributed capability discovery and communication mechanism. Within A2A, each agent registers its capability-description file (AgentCard) to a predefined URI, publicly exposing its functions, interfaces, and communication specifications. Other agents can address and obtain the capability map and initiate asynchronous interactions supporting multimodal data. In addition to capability discovery, A2A also incorporates an authentication mechanism to establish secure communication channels between agents. This mechanism can integrate with **Decentralized Identifiers (DIDs)**, allowing each AgentCard to include a DID reference that links to a DID Document containing public

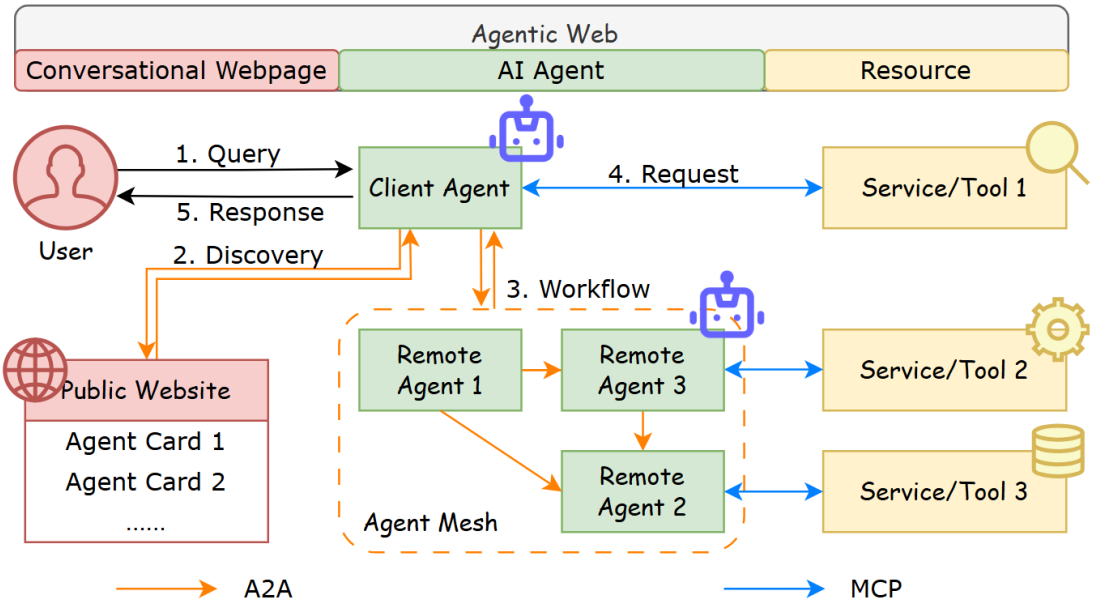

Figure 12: The above schematic illustrates a rudimentary AgentWeb system, with A2A and MCP serving as examples of agent communication protocols. In such an Agent Web, a user can assign a task (query) to a client agent via a conventional interactive medium (GUI or text). The client agent then discovers remote agents that meet the task's criteria on a public webpage, under the A2A protocol, and the task is subsequently allocated to their constituent agent mesh via the A2A protocol. During the execution process, agents may request external resources through the MCP protocol to facilitate task completion. Once the task completes, the client agent will return the result to the user in a human-readable form.

keys and authentication methods. By resolving and verifying these DIDs, agents can perform decentralized, cryptographically verifiable identity checks without relying on centralized registries or third-party identity providers. This enables self-sovereign authentication, improves interoperability, and aligns well with the trust requirements of dynamic agent ecosystems. At the same time, to meet the requirements of user intervention in sensitive operations and asynchronous task control in the Agentic Web, the protocol introduces an event-driven and state-callback mechanism. It triggers event notifications at key nodes of long-cycle tasks, pushes intermediate results for user confirmation, and dynamically updates task states through the state-callback interface, compensating for the deficiencies of traditional protocols in asynchronous interaction and user intervention.

5.3.2 Details of MCP

The working process of the MCP centers around session interactions based on capability negotiation, constructing an efficient, secure, and standardised communication system through the collaboration of the **Host**, the **MCP Clients**, the **MCP Servers**, and **Resources** (Anthropic, 2024).

1. **Host** denotes LLM-based agents tasked with user interaction, comprehension, and reasoning over user queries, tool selection, and the initiation of **strategic context requests**. Each host may be associated with multiple MCP clients.

2. **MCP Client** performs two key functions: it interfaces with a host to enumerate available resources and creates a singular connection to an MCP Server for the purpose of launching executive context requests.

3. **MCP Server** interfaces with the Resource while sustaining an exclusive connection to the MCP client, delivering the required contextual information from the Resource to the MCP Client.

4. **Resource** denotes the data, tools, or services provided locally or remotely.

In the initialization phase, the **Host** is manually connected to several **MCP Clients**, while each corresponding **MCP Server** connects to its accessible **Resources** such as the local file system or a remote dataset. Subsequently, the MCP Client will complete its initialization configuration, followed by actively establishing a session connection with the corresponding MCP Server. In the initial stage of this session, the MCP Client will launch a *capability declaration* request by providing the MCP Server with a detailed exposition of the functions for which it is equipped. Upon receiving a client request, the server communicates its capabilities, including service subscription, tool-call interfaces, and prompt template provision. The process of *capability declaration and response* facilitates the precise delineation of the boundaries of the protocol features that can be enabled in the session by both the client and the server. It ensures that the interaction proceeds in an orderly manner within the scope of their capabilities and establishes an active session based on negotiated functions.

After launching the session, interactions advance efficiently in a parallel multi-loop pattern. The Host is primarily responsible for interacting with the user and executing corresponding instructions based on user queries. During this user-Host interaction, the user may explicitly require the participation of a specific capability or context provided by a particular MCP Client to accomplish a given task. Alternatively, the Host may proactively identify the required capability or context through its own task understanding and reasoning processes. In both scenarios a **strategic context request** will accordingly be sent to an appropriate MCP Client for collaboration by the Host. Subsequently, the MCP Client will transform this request into an **executive context request** for tools or resources and send it to the MCP Server. Following the MCP Server's processing of the request and the subsequent return of a response, the MCP Client receives this and passes it to the Host. Then the Host will either update the user interface or provide feedback to the AI model, thus completing a full cycle of user-agent interaction supported by MCP.

In the session termination phase, the Host send termination instructions to all MCP Clients, which then send session-end request to the servers, officially ending the entire session life cycle.

Additionally, the notification loop mechanism guarantees the real-time transmission of significant information, such as alterations in resource status. When the MCP Server detecting resource updates, it promptly issue notifications to the MCP Client to ensure that both the MCP Client, enabling the Host to receive continuous, real-time updates.

By adopting a Client-Server architecture to mediate and standardize agents' requests to resources, the fragmentation in tool invocation caused by various providers of LLM and service is significantly reduced. This approach substantially enhances the semantic accuracy of interactions between agents and non-agent resources, thereby improving the overall precision and reliability of the Agentic Web.

5.3.3 Details of A2A

A2A, an acronym for Agent to Agent, is an agent communication protocol proposed by Google for enterprise-scale agent ecosystems, which enables agents to communicate and collaborate effectively, irrespective of their underlying frameworks or provider-specific implementations. The following components constitute the fundamental elements of A2A: **Agent Card**, **Task**, **Message**, and **Artifact** (Google, 2025a).

1. **Agent Card** denotes a publicly accessible JSON document, typically hosted at a public URL, detailing the agent's operational scope, its specific functions, the designated endpoint URL, methods for authentication, and other relevant metadata.

2. **Task** is a concrete representation of a unit of work, identified by a unique ID, whose status can be updated over multiple rounds of interaction.

3. **Message** refers to a communication object exchanged among entities, usually attributed with either a "user" or "agent" role. Messages may contain multiple **Parts**, such as text, file attachments, or structured data, supporting multimodal interaction.

4. **Artifact** is the output generated by an agent during the execution of a task. Unlike Message, which typically conveys dialogue or instructions, Artifact represents a finalized result or deliverable produced by the agent.

The workflow of A2A is quite straightforward. When receiving a user query, the client agent creates a new **Task** with a unique ID and begins the task execution process. First, the client agent retrieves JSON-formatted **Agent Cards** from publicly accessible URL to identify remote agents whose capabilities match the task requirements. Once suitable remote agents has been located, communication and collaboration between the client agent and the remote agents begins via the exchange of **Messages** under the A2A protocol. As the task execution progresses, the state of **Task** is continuously updated to reflect real-time changes. Finally, once the client agent determines that the task has been completed, the output is encapsulated and delivered in the form of an **Artifact**.

Compared to the MCP, the A2A protocol not only expands the scope of agent communication to include direct interaction and collaboration between heterogeneous agents but also significantly improves the management of context, messages, and tasks in multi-agent coordination, establishing a strong structural association between these elements.

Specifically, the A2A protocol allocates a unique identifier to each context, creating an explicit link between tasks and their associated environments. This design allows for the more organized and traceable management of multiple interrelated tasks, improving context consistency in complex scenarios and supporting robust task mechanisms. In addition, A2A introduces the unique identifier of the current task and a list of related task IDs for each message. This establishes a bidirectional, structured link between messages and tasks, enabling semantic anchoring of messages to their corresponding tasks and historical referencing across tasks. Consequently, A2A supports context tracing throughout multi-turn interactions, tool invocation sequences, and other temporally extended workflows, forming coherent interaction chains. Unlike the MCP protocol, which has a loose coupling between tasks and messages, A2A's tightly integrated design is better suited to complex collaboration scenarios, such as iterative dialogues and multi-agent tool orchestration. It significantly enhances the capabilities of systems in terms of state synchronisation, semantic coherence, and fault tolerance across distributed intelligent agents.

In addition, the A2A protocol explicitly addresses the asynchronous nature of agent communication, introducing mechanisms for asynchronous messaging and task state updates. Once a client agent has initiated a task, it can subscribe to receive progress updates relating to that task. As the task progresses, any status changes are promptly sent to the client agent, ensuring users are kept informed in real time.

The A2A protocol achieves cross-task, multi-turn, and cross-agent context consistency tracking through its context identification and tightly coupled task–message binding mechanisms. This ensures a high degree of coordination between information and task flows in multi-agent systems. Furthermore, the protocol incorporates specific design considerations to support asynchronous agent communication and task progress updates. These features provide a robust foundation for the Agentic Web, enabling effective context management and supporting long-duration, multi-stage, and asynchronous task execution.

5.4 Emerging Directions of Agentic Web Systems

Having detailed the systematic transitions of the Agentic Web, from its foundational architecture to its core communication protocols, we now stand at a critical juncture. The technical frameworks, while robust, do not by themselves guarantee successful real world deployment. Their implementation introduces profound paradigm shifts that challenge long-standing assumptions about digital interaction and commerce. This section, therefore, pivots from the established mechanics of agent

systems to confront the most pressing open questions that will determine their viability and adoption. We will explore two fundamental challenges that arise directly from the previously discussed transformations: the disruption of the traditional user-browser relationship and the unresolved complexity of creating sustainable billing models for agentic services. Answering these questions is crucial to connect architectural principles with their real-world implementation.

5.4.1 The Disruption of Traditional Browsers by Agents

The emergence of the "agent browser" signifies a fundamental disruption to the established user-browser interaction paradigm that has dominated the web for decades. Traditional browsers function as passive, user-driven tools for information retrieval and direct manipulation; the user is in complete control, manually clicking links, filling forms, and navigating pages. In stark contrast, an agent browser operates as a proactive, goal-oriented partner. It accepts high-level objectives in natural language and autonomously translates them into a series of actions, fundamentally altering the user's role from a hands-on operator to a strategic delegator.

This shift from direct manipulation to delegated autonomy raises profound questions about interface design, user control, and trust. How can we design user interfaces that effectively manage user expectations when the execution path is no longer linear or predictable, but is instead a dynamic process decided by the agent? When a user delegates a complex task, the agent's reasoning process can become a "black box," creating a potential gap in user understanding and trust. What new interaction primitives are required to allow for meaningful human oversight, intervention, and collaboration without undermining the agent's autonomy? What methods are effective for shaping a user's mental model to precisely represent the functionalities and restrictions of the agent, ensuring they can delegate tasks effectively and safely? Ultimately, the central question is how we redefine the user's relationship with the browser when it evolves from a simple tool into an intelligent, autonomous partner.

5.4.2 The Billing Challenge for Advanced Agent Services

Beyond the challenges in user interaction, the practical and widespread adoption of advanced agent systems confronts a critical hurdle: the development of viable and transparent billing models. Unlike traditional software with predictable, often flat-rate pricing (e.g., subscriptions), advanced agent tasks, such as conducting a deep investigative report, generating complex images, or executing multi-step financial analyses, incur variable and potentially substantial computational costs. These costs stem from resource-intensive operations, including extensive LLM token consumption, numerous third-party API calls, and prolonged use of high-performance computing infrastructure.

This variability raises a central, unresolved question: how can we design a billing system that is both equitable for the user and sustainable for the service provider? The traditional "one-size-fits-all" subscription model appears inadequate for this new reality. How can resource consumption be accurately tracked and attributed back to a single high-level user command, especially when that command spawns multiple sub-agents that may collaborate and delegate tasks further? What mechanisms can be implemented to provide users with a reliable cost estimate *before* initiating a potentially expensive task, thereby preventing "bill shock" and fostering trust? Should billing be based on consumed resources (e.g., tokens, CPU time), the value of the final outcome, or a more complex hybrid model? Devising a framework that is granular, transparent, and user-friendly is a formidable challenge that will directly impact the economic feasibility and accessibility of the entire agent ecosystem.

6 Applications of the Agentic Web

To understand how the Agentic Web is transforming digital environments, we begin by examining its core capabilities: transactional, informational, and communicational paradigms. These paradigms serve as the foundation for a wide range of use cases.

In the following subsections, we explore both the potential domains of the Agentic Web, which provide a conceptual framework, and its current applications, which illustrate how these paradigms are already being implemented in real-world systems.

6.1 Potential Domains of the Agentic Web

The Agentic Web enables intelligent agents not only to access web content but also to operate autonomously as active participants within the web. It facilitates three core functional paradigms, transactional, informational, and communicational, which allow agents to autonomously execute tasks, process and reason over knowledge, and collaborate with other agents within digital environments. By providing machine readable interfaces, persistent cross-platform memory, and standardized coordination protocols, the Agentic Web transforms these paradigms from isolated agent behaviors into scalable, system-wide capabilities.

- **Transactional**: Agents autonomously execute goal-oriented tasks on behalf of users or organizations, such as purchasing, booking, scheduling, or negotiating, by interfacing directly with web services, APIs, or transactional interfaces.
- **Informational**: Agents retrieve, synthesize, and contextualize information across dynamic sources. This modality supports research, knowledge discovery, monitoring, and real-time decision support through adaptive reasoning and long-horizon memory.
- **Communicational**: Agents engage in structured communication with other agents or systems to coordinate, delegate, or cocreate. This includes multi-agent negotiation, protocol alignment, and collaborative workflows across organizational or platform boundaries.

These modalities represent distinct ways in which agents interact with digital environments, whether by executing tasks, gathering and analyzing knowledge, or coordinating with other agents. The integration of these modalities into real-world applications highlights the transformative potential of the Agentic Web. Most applications of the Agentic Web span multiple modalities, with specific systems emphasizing different functional combinations. The following sections will analyze representative implementations and domain-specific use cases from this perspective, incorporating insights from recent research.

6.1.1 Transactional: Enabling Autonomous Execution of Web-Based Services

The Agentic Web redefines how transactional interactions are conducted by embedding LLM-powered agents directly into service infrastructures (Zhou et al., 2023b; Deng et al., 2023). With the help of semantic APIs, standardized execution protocols, and persistent authorization tokens, agents can interact with multiple service endpoints without requiring bespoke integrations (Masterman et al., 2024).

This framework enables agents to autonomously orchestrate complex, multi-step workflows. For example, booking a trip no longer requires users to manually navigate several websites. Instead, an agent within the Agentic Web can query multiple travel providers, assess options based on factors such as price, time, loyalty status, or environmental impact, and complete the bookings simultaneously by coordinating flights, accommodations, and car rentals in one seamless operation.

Similarly, App/Mobile Agents (Wang et al., 2024a; Wu et al., 2025; Zhang et al., 2025a) enhance the Agentic Web's transactional capabilities by providing personalized, context-aware services across devices. App/Mobile Agents can autonomously handle tasks such as managing a user's calendar, adjusting schedules, and coordinating tasks based on real-time information. For instance, when booking a flight, a Mobile Agent could automatically adjust the user's itinerary if a flight is delayed, suggest meal options based on dietary preferences, or even reorder tickets if there is a sudden change in plans. These agents operate across mobile platforms, facilitating the seamless execution of transactional activities while adapting to changing user needs.

These capabilities rely on a web environment designed for autonomous machine participation, where agents can read, write, and reason about data, negotiate terms, and take action based on user preferences and environmental factors, thereby creating a more dynamic and efficient transactional experience.

6.1.2 Informational: Structuring Autonomous Knowledge Discovery and Analysis

In the informational domain, the Agentic Web powers a system that allows agents to access dynamic content persistently, reason over long-term sources, and achieve semantic interoperability across heterogeneous knowledge systems. Rather than merely retrieving data, agents within the Agentic Web are empowered to perform end-to-end research tasks, identifying, contextualizing, and synthesizing information over extended periods (Opera, 2025; Corporation, 2025).

In this model, agents go beyond simple search queries and static responses. Utilizing standardized document schemas, citation graphs, and persistent monitoring capabilities, agents can perform comprehensive, longitudinal research. For example, Deepresearch Agents (Huang et al., 2025b) autonomously track emerging papers, compare methodologies, extract citations, and synthesize findings into structured outputs. These agents continuously refine their insights based on the latest publications, leveraging the Agentic Web's ability to facilitate cross-platform collaboration and seamlessly integrate new data sources. This allows Deepresearch Agents to operate as active participants in a broader, interconnected research ecosystem, where knowledge is continuously updated and refined.

The Agentic Web facilitates this by providing a unified infrastructure where agents are not only capable of reading and writing data but also reasoning, negotiating, and acting within a dynamic and evolving environment. Deepresearch Agents are designed to assist researchers in navigating the vast and ever-evolving landscape of academic literature, and they do this by leveraging the Agentic Web's capabilities for cross-platform memory and semantic interoperability. These agents autonomously identify gaps in research, suggest new directions, and propose potential collaborators based on shared interests, making the research process more efficient and comprehensive.

In practice, Deepresearch Agents automate the synthesis of large datasets, identifying patterns and trends across a wide range of publications. This process is made scalable by the Agentic Web's ability to support inter-agent communication, where these agents can collaborate, share findings, and even align their goals with other agents working across different domains. By doing so, the Agentic Web transforms research from isolated, manual efforts into a collaborative, scalable system of knowledge discovery.

6.1.3 Communicational: Orchestrating Inter-Agent Collaboration and Negotiation

Perhaps the most distinct departure from today's web lies in the Agentic Web's capacity to support autonomous, goal-driven communication between agents. This capability is not limited to message passing; it also encompasses semantic alignment, negotiation, delegation, and long-term coordination across agents that represent different users, systems, or organizations.

In a communicational paradigm, agents function as active participants in multi-agent workflows (Tran et al., 2025; Chen et al., 2023b). Consider a joint research initiative spanning multiple universities: agents representing each institution can autonomously share relevant datasets, align experimental timelines, and coauthor reports, negotiating authorship, funding distribution, and intellectual property rights based on prespecified protocols (Anthropic, 2024a).

Creative industries benefit similarly. The Agentic Web supports the formation of temporary agent coalitions for cross-modal content creation (Adobe, 2025; Khade, 2024), where writing agents, visual design agents, and music composition agents coordinate roles, timelines, and revenue sharing agreements. In this context, the web's support for decentralized identity, smart contracts, and task provenance becomes essential.

In enterprise environments, collaboration is enhanced when agents from different companies autonomously coordinate and communicate (Yang et al., 2025e;c). For example, in a manufacturing

Table 2: Representative AI-Augmented Browsers (Agent-as-Interface).

Application	**Intelligence Domain**	**Interaction Domain**	**Economic Domain**	**Focus**
Opera Neon	Agentic AI with task orchestration	Chat-Do-Make sidebar, autonomous assist mode	Invite-only preview; premium model	Informational
Perplexity Comet	Search-augmented LLM with automation	Chromium-based browser; sidecar assistant	Subscription-based (Perplexity Max)	Informational
Browser Dia	Context-aware browsing assistant	Inline chat with context reasoning, insertion cursor	Beta (Arc users); invite-only	Informational
Copilot (Edge)	Contextual summarization and suggestions	Edge sidebar; light task hints	Freely available in Edge	Informational
Microsoft NLWeb	Natural language semantic interface	Conversational UI via Schema/MCP	Open-source; publisher integration	Communicational

ecosystem, supplier agents, logistics agents, and procurement agents can autonomously share information and adapt supply chains in real time to respond to disruptions (SmythOS, 2024).

At the core of all these applications lies a communicational infrastructure designed for autonomous participants. Agents are capable of interpreting shared protocols, maintaining structured dialogue states, and reasoning about shared goals and constraints throughout long-term interactions.

6.2 Current Applications of the Agentic Web

The Agentic Web is already transitioning from conceptual frameworks to real-world applications. We categorize its early implementations into two primary interaction models: **Agent-as-Interface** and **Agent-as-User**. The former focuses on augmenting the user experience by providing intelligent intermediaries between humans and the web, while the latter introduces autonomous agents that act on behalf of the user, interacting with web systems directly as proxy users.

6.2.1 Agent-as-Interface: Agents as Intelligent Web Intermediaries

In the *Agent-as-Interface* paradigm, agents enhance traditional user interfaces by providing context-aware assistance, task recommendations, and intelligent summarization. These systems typically operate alongside the human user, augmenting their browsing experience without fully automating decision-making. Representative applications are summarized in Table 2.

Opera Neon delivers one of the most integrated experiences of agent-enhanced browsing. Released in May 2025, it features a tri-modal interface: *Chat* enables conversational interaction with LLMs, *Do* facilitates autonomous completion of web tasks such as multi-step forms and service workflows, and *Make* empowers content creation and persistent agent tasks even when users are offline (Opera, 2025; Press, 2025). Notably, Opera Neon's "Do" mode represents a **hybrid approach**, where the system transitions from interface augmentation to autonomous user proxy behavior, demonstrating how Agent-as-Interface applications can incorporate Agent-as-User capabilities while

maintaining the primary focus on user-controlled workflows. Neon exemplifies the transition from passive interfaces to proactive, task-oriented web experiences.

Perplexity Comet enhances the classic search experience by embedding autonomous search agents directly within the browser environment. Comet incorporates AI-driven research, question answering, and proactive summarization within a Chromium-based framework, reducing the need for iterative querying while maintaining human oversight in decision-making loops (Wiggers, 2025).

Browser Dia introduces an *insertion cursor* that provides real-time agentic suggestions within the browsing context, moving beyond sidebar chat to deeply integrated inline assistance (Thurrott, 2024). This design reduces context-switching overhead and improves session continuity, highlighting the benefits of embedded, contextually aware agents.

Microsoft Copilot focuses on summarization and lightweight agentic assistance via a non-intrusive sidebar, targeting everyday users who benefit from summarizations, insights, and task hints but do not require full task automation (Corporation, 2025).

Microsoft NLWeb advances the notion of *Agent-Native Interfaces*, proposing a semantic layer for websites where agents interact through natural language interfaces exposed via schemas and MCP (Microsoft Corporate Blogs, 2025). By encouraging publishers to design agent-accessible endpoints, NLWeb shifts the web ecosystem towards cooperative interaction between websites and AI agents, reducing reliance on brittle scraping and improving transparency in web automation.

6.2.2 Agent-as-User: Autonomous Agents Operating as Proxies

In the *Agent-as-User* paradigm, AI systems operate autonomously as users of the web, executing tasks, navigating interfaces, and completing workflows without direct human control. These systems leverage browser automation, virtual environments, and programmatic UI manipulation to emulate user actions, thereby enabling end-to-end autonomy. The development and evaluation of such agents have been greatly facilitated by comprehensive benchmarks like Mind2Web (Deng et al., 2023), which includes over 2,000 open-ended tasks across 137 websites in 31 domains. Its online extension, Online-Mind2Web (Xue et al., 2025), further advances this effort by offering 300 diverse and realistic tasks across 136 websites, enabling the assessment of web agents under conditions that closely mirror real-world usage patterns. Recent advances in multimodal agent foundations (Wu et al., 2025) and mobile agent architectures (Wang et al., 2024a) have further expanded the scope of autonomous agent capabilities beyond traditional web environments, while scalable task generation methodologies (Xie et al., 2025) advance the field's evaluation capabilities. Examples of recent applications are shown in Table 3.

ChatGPT Agent (evolved from OpenAI Operator), released via the ChatGPT platform, represents one of the first multi-modal agentic deployments with persistent virtual browsing capabilities. Combining LLM reasoning with code execution, file system access, and API integrations, the Agent can autonomously complete multi-step tasks such as booking services, extracting structured data, or synthesizing reports across complex web workflows. Initially launched as Operator in January 2025, it has since been fully integrated into ChatGPT's core platform as "Agent Mode" demonstrating the rapid evolution from standalone research prototypes to integrated production systems (OpenAI, 2025).

Anthropic Computer Use leverages vision-based perception and GUI manipulation, powered by Claude models, to control desktop and web interfaces in a human-like fashion without relying on backend APIs. Available through Claude 3.5 Sonnet, it showcases highly generalized agents capable of interacting with arbitrary applications. On standardized OSWorld benchmarks, Computer Use achieves 14.9% success rate on screenshot-only tasks and 22.0% with reasoning steps, significantly outperforming previous vision-action baselines (Anthropic, 2024).

Table 3: Representative Autonomous Web Agents (Agent-as-User).

Application	**Intelligence Domain**	**Interaction Domain**	**Economic Domain**	**Focus**
ChatGPT Agent	Multi-modal agent orchestration	Virtual browser; cross-API tool integration	ChatGPT Plus/Team tiers	Transactional
Anthropic Computer Use	Vision-guided GUI manipulation	Claude-powered desktop/web control	Claude Sonnet 3.5 API	Transactional
Google Project Mariner	Autonomous long-horizon task execution	Gemini-2 reasoning within Chrome prototype	Research prototype (Gemini 2.0)	Transactional
Genspark Super Agent	Mixture-of-Agents orchestration; 9 LLMs	Multimodal real-world task execution (voice, maps, documents)	Free tier + commercial credits	Multi-domain personal productivity

Google Project Mariner is an experimental autonomous agent system powered by Gemini 2.0 models and integrated into Chrome as a sidebar prototype. Designed for long-horizon research tasks, multi-step workflows, and autonomous form filling, Mariner incorporates reasoning transparency via natural language explanations of its actions. Evaluated on the WebVoyager benchmark, it achieves an 83.5% success rate on long-horizon web tasks, representing a cutting-edge research milestone in explainable autonomous browsing (Google DeepMind Blog, 2024).

Genspark Super Agent exemplifies a next-generation implementation of agentic autonomy through its Mixture-of-Agents architecture. Unlike traditional assistants that merely retrieve information, Super Agent can plan, act, and use over 80 tools, including real-time voice calls, map navigation, document editing, calendar scheduling, and video generation, across diverse domains with minimal supervision. It dynamically orchestrates nine large language models and integrates more than ten proprietary datasets, enabling multi-step task execution and adaptive reasoning. Genspark Super Agent thus illustrates the evolution from conversational AI to autonomous digital agency, enhancing personal productivity through end-to-end workflow automation (Genspark, 2025).

The current landscape reveals a clear evolutionary trajectory where **Agent-as-Interface applications are progressively incorporating Agent-as-User capabilities**. This evolution is driven by fundamental differences in their underlying technical architectures. Agent-as-Interface systems primarily rely on **API-based interactions**, utilizing structured endpoints, webhooks, and service integrations to mediate between users and web services. This approach offers faster execution, better error handling, and more predictable outcomes, but remains constrained by the availability and design of existing APIs. In contrast, Agent-as-User systems employ **GUI-level automation**, using computer vision, coordinate-based clicking, and screen parsing to interact with arbitrary interfaces designed for human use. While this approach provides universal compatibility and can operate on any visual interface, it introduces latency, brittleness, and higher computational overhead due to the need for continuous visual interpretation and coordinate calculation.

The convergence toward hybrid architectures suggests a future where agents dynamically select between API calls for structured interactions and GUI automation for legacy or non-API-enabled systems. This **architectural pluralism** represents the next evolutionary step, where the same

agent can seamlessly transition between acting as an intelligent interface layer and operating as an autonomous user proxy, depending on the task context and available interaction modalities. Such systems will likely require sophisticated decision trees to determine the optimal interaction method for each specific workflow component, building upon advances in multimodal agent foundations (Wu et al., 2025) and enhanced by deep research capabilities (Huang et al., 2025b) for complex information synthesis tasks.

In summary, early applications of the Agentic Web demonstrate a spectrum of possibilities from *Agent-as-Interface* augmentation to *Agent-as-User* autonomy, with an emerging trend toward hybrid implementations. The convergence of commercial products and academic research suggests accelerating momentum toward more capable, accountable, and architecturally flexible web agents that can adapt their interaction strategies to maximize both efficiency and reliability.

6.2.3 Agent-with-Physics: Autonomous Robots Powered by AI Agents

The *Agent-with-Physics* paradigm extends the concept of agentic intelligence from the virtual realm to the physical world, enabling AI agents to perceive, reason, and act through embodied systems such as robots and sensor-equipped devices. These agents integrate high-level planning with low-level control, often relying on multimodal perception (e.g., vision, audio, haptics), real-time adaptation, and embodied cognition to execute physical tasks autonomously in dynamic environments.

Unlike purely digital agents, Agent-with-Physics systems must address challenges related to safety, latency, actuation uncertainty, and physical affordances. Recent advances in vision-language-action models (Kim et al., 2024; Geng et al., 2025; Li et al., 2023), hierarchical policy learning (NVIDIA et al., 2025; Kuang et al., 2024; Geng et al., 2023; Ding et al., 2024), and real-world training environments, such as Open X-Embodiment (O'Neill et al., 2023), have significantly improved the generalization capabilities of robotic agents across diverse tasks, from household manipulation to warehouse logistics.

Representative implementations such as RT-1 (Brohan et al., 2023b), RT-2 (Brohan et al., 2023a) and RT-X (O'Neill et al., 2023), Tesla Optimus, and Figure 01 showcase emerging commercial interest in general-purpose humanoid robots, while academic efforts like PaLM-E (Driess et al., 2023) and Mobile ALOHA (Fu et al., 2024) highlight the integration of large foundation models into robotic control loops. These systems demonstrate the feasibility of using language prompts to guide physical behavior, bridging human intent and machine execution through a unified agentic framework.

As embodied agents increasingly connect with digital ecosystems, a new class of hybrid agents emerges, capable of coordinating actions both online and offline. For instance, an agent might autonomously schedule a grocery delivery online while simultaneously preparing a physical environment (e.g., setting up a smart kitchen) for the incoming goods. This tight coupling of perception, cognition, and actuation highlights the importance of developing robust control policies, real-time feedback loops, and safety-aware planning strategies.

Looking ahead, the Agent-with-Physics paradigm not only expands the frontier of human-agent collaboration but also lays the groundwork for a unified agentic infrastructure where digital and physical agents operate in concert. The fusion of web-native intelligence, embodied autonomy, and multimodal interaction marks a critical step toward realizing truly general-purpose AI agents capable of seamlessly bridging virtual tasks and physical realities.

7 Risks, Security & Governance

In this section, we provide an overview of how agentic web safety and security can be ensured. As illustrated in Figure 13, the ecosystem of Agentic Web Safety and Security is composed of intelligent agents, powered by LLMs such as OpenAI (Hurst et al., 2024), Gemini (Team et al., 2023), and other foundational platforms (Touvron et al., 2023a; Bai et al., 2023; Liu et al., 2024a; Zan et al., 2025; Priyanshu et al., 2024), operating across a wide range of devices, including desktops,

Figure 13: Illustration of the Agentic Web Safety and Security Ecosystem. The central hexagon represents the core components of the agentic web, including large language models (e.g., OpenAI (Hurst et al., 2024) and Gemini (Team et al., 2023)), agent frameworks, and safety infrastructures. These systems interact with a diverse set of devices (e.g., laptops, desktops, servers, and mobile phones), requiring robust and scalable security measures to ensure safe deployment and communication across the entire agentic web.

laptops, servers, and mobile phones. These agents interact with cloud services, third-party tools, and each other to carry out goal-directed tasks on behalf of users. At the center of the figure is a secure agentic infrastructure that integrates LLMs, agent frameworks, and cloud-based safety mechanisms. The surrounding arrows depict multi-device interaction, emphasizing the distributed nature of the agentic web and the critical need for consistent, cross-platform security protocols. This interconnected architecture highlights the growing importance of privacy, trust, and robustness, as agents autonomously retrieve information, execute commands, and collaborate across sensitive digital environments.

To ensure the safety and security of the agentic web, we begin by analyzing its potential threats during real-world use, then introduce red-teaming methods for uncovering vulnerabilities, followed by defense strategies to address these issues. Finally, we present evaluation techniques to measure the effectiveness of safety and security mechanisms. Specifically, Section 7.1 outlines the key safety and security threats associated with the agentic web. Building on this analysis, Section 7.2 discusses red teaming as a methodology for identifying vulnerabilities and assessing the robustness of agentic web systems before deployment. Section 7.3 explores defense strategies and technical safeguards aimed at enhancing the reliability and trustworthiness of agentic web applications. Lastly, Section 7.4 reviews current approaches for evaluating the safety and security of these systems.

7.1 Safety and Security Threats

Agentic Web agents introduce novel security risks by operating autonomously across the open internet, executing real transactions, and maintaining persistent states. Table 4 captures this fundamental shift.

7.1.1 Threat Analysis Across Agentic Web Layers

We organize threats across the three architectural layers (Section 3), focusing on risks unique to autonomous web operations, as shown in Tables 5–7. While some attacks may have cross-layer

Table 4: Agentic Web Risk Evolution: From Controlled Systems to Autonomous Web Operations

Dimension	Traditional AI → Agentic Web
Operational Scope	Single-domain tasks → Cross-platform orchestration
Financial Authority	Read-only access → Transaction execution
Persistence	Stateless queries → Multi-session memory
Attack Surface	API endpoints → Entire web ecosystem
Failure Impact	Incorrect output → Real-world consequences
Trust Model	Human verification → Autonomous decision-making

Table 5: Intelligence Layer Threats: Cognitive and Reasoning Attacks. These threats target the agent's decision-making processes, focusing on how web interactions corrupt objectives, knowledge, and planning capabilities.

Threat	Description	Example
C1: Persuasion-Based Goal Drift	Web UI/UX patterns gradually shift agent objectives through psychological manipulation	Budget flight search influenced by comfort-emphasizing interfaces to book premium seats
C2: Knowledge Base Poisoning	Adversarial web content corrupts agent's accumulated knowledge and beliefs	SEO-optimized fake research sites poison climate agent's factual understanding
C3: Preference Learning Corruption	Agent learns harmful patterns from repeated web interactions	Shopping agent trained to prefer sponsored results through manipulated feedback
C4: Multi-Stage Planning Subversion	Complex task sequences manipulated through incremental decision corruption	Travel itinerary gradually inflated: economy→premium flight triggers luxury hotel selection
C5: Contextual Memory Exploitation	Historical interactions weaponized to bias future decisions	Past emergency booking used to justify all future premium purchases

effects (e.g., goal drift leading to overspending), we categorize each threat by its primary attack vector to avoid redundancy and maintain analytical clarity.

Cross-Layer Threat Cascades Unlike traditional systems where threats remain isolated, Agentic Web threats cascade across layers: **Vertical** (cognitive→protocol→economic), **Horizontal** (agent-to-agent spread), and **Temporal** (corruption through persistent memory (Narajala and Narayan, 2025)). These cascades transform localized attacks into system-wide failures (de Witt, 2025). For instance, a goal drift (C1) can lead to protocol exploitation (P1), ultimately resulting in unauthorized purchases (E1). The temporal dimension adds further complexity as threats can persist across agent generations through learned behaviors and contaminated training data.

Relationship to Existing Frameworks While OWASP's Agentic AI Threat Model (OWASP GenAI Security Project, 2025) and CSA's MAESTRO framework (Huang, 2025) provide foundational vulnerability taxonomies, Agentic Web threats differ in scale and propagation. For instance, our Knowledge Base Poisoning (C2) extends beyond prompt injection to web-scale information corruption, as demonstrated by code review agent compromises (CyberArk Labs, 2025).

Protocol threats adapt traditional network security to agent-specific contexts. MCP Context Injection (P1) exploits the protocol's contextual awareness, which is identified as a fundamental vulnerability by Hou et al. (2025) and demonstrated practically by Attiya (2025). A2A coordination attacks have been validated through CrewAI and AutoGen exploits (Palo Alto Networks Unit 42, 2025).

Table 6: Interaction Layer Threats: Protocol and Communication Attacks. These threats exploit agent communication mechanisms (MCP/A2A) and cross-service authentication, distinct from cognitive corruption.

Threat	Description	Example
P1: Context Injection	Malicious services inject persistent false contexts affecting cross-platform behavior	Hotel adds "VIP status" to MCP context, causing unnecessary upgrades across all bookings
P2: Service Registry Poisoning	Fake services infiltrate discovery systems to intercept agent requests	Malicious "FastBooking" service in MCP registry harvests credentials from travel agents
P3: A2A Trust Exploitation	Compromised agents abuse inter-agent trust to spread malicious behaviors	High-reputation research agent injects fabricated citations into collaborative analysis
P4: Authentication Chain Hijacking	Sequential auth tokens exploited across service boundaries	Google login token escalated to access Drive, then third-party research databases
P5: Protocol Negotiation Attack	Malicious actors force protocol downgrades or incompatible versions during handshakes	Service forces agent from secure A2A v2 to vulnerable v1 during capability exchange
P6: Coordination Storm	Malicious messages trigger exponential inter-agent communications	Single A2A broadcast spawns millions of agent-to-agent queries

The potential for AI systems to act as autonomous economic agents has been recognized since Brundage et al. (2018) identified market manipulation as an emerging threat vector, though the scale envisioned by the Agentic Web amplifies these concerns significantly. A critical challenge in this sphere is ensuring agents consider their impact on multiple stakeholders. Mitigation strategies could draw from concepts such as simulating accountability and assessing stakeholder impact, though scaling these alignment mechanisms remains an open challenge (Sel et al., 2024).

7.1.2 Security Implications and Future Directions

Agentic Web security requires three fundamental shifts from traditional approaches:

1. **Architecture**: Zero-trust models replacing perimeter security
2. **Policies**: Adaptive defenses superseding static rules
3. **Scope**: Cascade prevention over incident isolation

Enterprise patterns like MCP require adaptation for internet-scale deployments (Narajala and Habler, 2025), as traditional models assume bounded, stateless operations incompatible with persistent web agents.

Critical research gaps persist. Quantitative models for cascade probability and impact remain underdeveloped, particularly for emergent behaviors in complex multi-agent systems (de Witt, 2025). The challenge of securing systems that learn and adapt continuously has been recognized in recent work on evolving threat landscapes (Deng et al., 2025), while cross-jurisdictional governance poses additional complexity (Brundage et al., 2018). These dynamic threats may require fundamentally new security paradigms. Rather than static defenses that become obsolete, future approaches might embrace adaptability as a core principle: designing systems that strengthen through controlled adversarial exposure (Jin and Lee, 2025). Such adaptive security architectures could transform the continuous threat evolution from a vulnerability into a mechanism for improvement, though implementing this vision remains a significant research challenge.

Table 7: Value Layer Threats: Autonomous Transaction and Economic Risks. These threats emerge specifically when agents gain financial authority and market participation capabilities.

Threat	Description	Example
E1: Transaction Authority Abuse	Agents execute unauthorized high-value transactions without human oversight	Books non-refundable business class for family of four on budget trip
E2: Cross-Platform Arbitrage	Agents exploit pricing differences between services at harmful scales	Books and cancels flights across airlines to manipulate dynamic pricing
E3: Payment Credential Harvesting	Agents collect and misuse payment data across multiple transactions	Compromised booking agent logs credit cards from hotel/flight purchases
E4: API Resource Monopolization	Agents consume excessive computational resources across services	Research agent exhausts university's entire journal database quota
E5: Coordinated Market Manipulation	Agent networks create artificial supply/demand conditions	Multiple agents book all seats on routes to inflate prices

7.2 Safety and Security Red Teaming

Deploying AI agents within web applications introduces several technical risks (Zhang et al., 2024a), including privacy leakage and fairness concerns (Amodei et al., 2016; Chua et al., 2024). For example, one significant agentic web risk that may lead to information leakage is hallucination, where agents generate inaccurate or misleading content due to limited understanding of user intent or contextual ambiguity during retrieval and generation. Another critical risk is permission escalation: when agents access sensitive user data, such as personal files (e.g., user name, password, and contacts), they may misinterpret access boundaries or inadvertently override security constraints when the agent manages different web pages. These failures can result in unauthorized data exposure, privacy violations, and broader system-level vulnerabilities in the agentic web.

To mitigate these risks, red-teaming techniques are promising approaches to ensure the safety and security of agentic web systems prior to real-world deployment. Red teaming has a long history in domains such as computer system security and military defense simulation, where it is used to identify vulnerabilities and evaluate system robustness (Verma et al., 2024). Red teaming involves simulating adversarial behavior to uncover vulnerabilities in a target system, traditionally performed through manual human design. Specifically, in traditional computer system safety and security, red teaming often involves manual human effort, expert-defined rules, and extensive scenario testing (Röttger et al., 2020; Ribeiro et al., 2020). In the era of AI-driven agentic web systems, red teaming can be largely automated, with AI agents autonomously generating adversarial scenarios to probe and uncover failure modes in other agentic systems (Wang et al., 2025c). These target systems may include various web platforms and pages, where the primary objective is to expose sensitive information leaks and reveal potential vulnerabilities, ultimately enhancing system trustworthiness and security before real-world deployment. This automation reduces reliance on manual efforts and enables scalable, adaptive adversarial evaluation (Perez et al., 2022; Ge et al., 2023; He et al., 2025).

7.2.1 Human-Involved Red Teaming

Human involvement has played a vital role in red-teaming efforts across diverse NLP tasks (Xu et al., 2020; Glaese et al., 2022; Radharapu et al., 2023), and these techniques can also be leveraged to enhance the safety and security of agentic web systems. For example, adversarial examples have been manually crafted to evaluate machine reading comprehension systems (Jia and Liang, 2017). Human annotations have been used to assess unintended bias in text classification (Dixon et al., 2018) and to support fairness and robustness evaluation through counterfactual data generation

(Garg et al., 2019). In multi-hop question answering, human-labeled examples have helped evaluate complex reasoning capabilities (Jiang and Bansal, 2019).

Human-in-the-loop frameworks have also been employed to generate adversarial attacks targeting dialogue safety (Dinan et al., 2019) and to improve robustness in language understanding tasks (Nie et al., 2019). Additionally, human-curated adversarial training datasets have been shown to enhance model performance (Wallace et al., 2021), particularly in high-stakes, reliability-critical settings (Ziegler et al., 2022). Manual red-teaming efforts have also been applied in the development of LLaMA models (Touvron et al., 2023a), where human annotators carefully design prompts to surface unsafe behaviors in large language models. Kiela et al. (2021) further propose a human-in-the-loop framework for dynamic adversarial testing via a web-based platform, enabling the continuous collection of adversarial examples from human annotators. Through this iterative process, models are exposed to increasingly challenging examples, leading to improved robustness over time. However, applying these human-involved techniques to agentic web environments presents new challenges. Agentic web systems are highly autonomous and capable of operating across a variety of web pages and platforms. This level of complexity and intelligence makes manual red-teaming less feasible and highlights the need for scalable, automated approaches.

In the context of the agentic web, where autonomous AI agents interact with complex, multi-platform environments on behalf of users, automated red teaming is becoming increasingly essential. Traditional human-in-the-loop approaches, such as designing security rules, labeling safety violations, or manually identifying vulnerabilities, struggle to scale in these dynamic, high-autonomy systems. For instance, safeguarding AI agents that navigate ticketing platforms, manage financial transactions, or operate across multiple web services requires real-time, adaptive evaluation that manual red teaming cannot sustain. Automated red teaming offers a scalable solution by enabling agents to simulate adversarial behaviors, uncover hidden safety flaws, and proactively report security risks, helping to ensure agentic web safety before real-world deployment.

7.2.2 Automatic Red Teaming

Recent work has increasingly explored the use of LLMs for automated red teaming (Perez et al., 2022; Ge et al., 2023; Nie et al., 2024; Liu et al., 2024b; Shi et al., 2024; Liu et al., 2025c; He et al., 2025; Wang et al., 2025b), which holds particular promise for enhancing safety in agentic web systems. For instance, Perez et al. (2022) show that LLMs can serve as effective red teamers by generating adversarial prompts to uncover unsafe behaviors. Similarly, Ge et al. (2023) introduce MART, which is an automatic red teaming framework and is designed to evaluate and enhance the safety of LLMs through adversarial scenario generation. Ganguli et al. (2022) explore various strategies for red-teaming, including rejection sampling and RL, and release a dataset to support this research. Their findings suggest that RL-based approaches can make systems more resistant to red-teaming attacks by hardening decision boundaries and improving robustness.

Building on this line of work, Nie et al. (2024) propose a RL-based red teaming approach that trains an adversarial agent using a carefully designed reward function to generate diverse adversarial examples, effectively revealing vulnerabilities in target LLMs. Their comprehensive experiments demonstrate that this method performs better than strong baselines in exposing model information leakage. Similarly, Wang et al. (2025b) introduce a red teaming method that leverages a seed instruction and a Monte Carlo Tree Search algorithm to optimize inputs for attacking the target system. Experimental results on the AgentDojo (Debenedetti et al., 2024) and VWAadv (Wu et al., 2024) benchmarks show that their method achieves superior performance compared to strong baselines, such as human-crafted adversarial prompts. These automated red teaming approaches are particularly valuable for enhancing the safety and security of agentic web systems, as they can be applied across various agent interactions to proactively identify vulnerabilities and strengthen defenses prior to deployment in diverse web environments.

Additional methods relevant to agentic web safety and security include backdoor-triggered red teaming, which can be particularly effective in identifying hidden vulnerabilities. For example, AgentPoi-

son (Chen et al., 2024c) leverages backdoor-triggered and retrieval-augmented LLMs for red teaming, aiming to improve system security. Their framework is evaluated across multiple domains, including autonomous driving, question answering, and healthcare, demonstrating its effectiveness in identifying and mitigating security vulnerabilities. Likewise, Yang et al. (2024) propose an agent-based backdoor attacker for red teaming LLM agents, focusing on web-based shopping systems where privacy leakage is a critical concern. In a related line of work, Xu et al. (2024b) introduce a data poisoning technique that injects backdoors through instruction attacks, demonstrating that even a few malicious tokens during instruction tuning can compromise model safety and expose critical vulnerabilities. This approach highlights the risks associated with instruction-based inputs and offers a valuable method for red teaming to uncover hidden weaknesses in language models.

Red teaming often involves multi-agent systems in the agentic web, where agents play both offensive and defensive roles. For that multi-agent techniques, Shi et al. (2024) utilize LLM-based agents to generate adversarial inputs through word substitutions and sentence rephrasings, targeting the robustness of LLM detection systems. In a more systemic approach, He et al. (2025) introduce a multi-agent red teaming framework in which LLMs act as adversarial agents to probe vulnerabilities, particularly in inter-agent communication protocols. Expanding the scope of automated evaluation, Radharapu et al. (2023) develop AI-assisted red teaming methods that extend across a broad range of applications, including policy evaluation and locale-specific challenges. In addition, AutoDan (Liu et al., 2024b) and AutoDan-Turbo (Liu et al., 2025c) present scalable and adaptive frameworks for automated red teaming, pushing the boundaries of adversarial testing for LLM-based system safety and security.

LLM-driven automatic red teaming represents a promising future research direction for safety and security, particularly within the context of the agentic web. These models can systematically simulate human behaviors and comprehensively probe safety and security vulnerabilities across various scenarios. However, to fully realize their potential, it is critical to design robust frameworks for managing these red-teaming agents in agentic web systems. These frameworks are crucial for both effective evaluation and defense, as well as for guaranteeing the safety of agentic web systems. This is especially important in real-world applications, such as when agents are used to book travel, interact with various apps and web pages, or manage tasks across personal computers and mobile devices.

7.2.3 Emerging Directions in Red Teaming for Agentic Web

As mentioned above, red teaming plays a critical role in identifying and mitigating safety risks in agentic web systems and LLMs prior to real-world deployment. Human-involved red teaming provides domain expertise and high reliability for specific tasks, making it a valuable tool for ensuring safety and security. However, it is often costly, time-consuming, and limited in diversity and scalability (Radharapu et al., 2023). Moreover, human red teamers may lack the broad and cross-domain knowledge necessary to effectively evaluate complex, multi-faceted systems.

Automated red teaming, particularly approaches based on LLMs, offers a promising alternative by reducing human effort and improving scalability in the era of agentic web. However, these methods could be unreliable or insufficient in scenarios that require complex reasoning, contextual understanding, or ethical judgment, such as different web page operations and platform management. Bridging the gap between human-involved and automated red teaming remains an open challenge. Future research should aim to develop hybrid frameworks that integrate the strengths of both human-involved and automated red-teaming approaches, while addressing their respective limitations.

One promising direction for agentic web safety involves leveraging safe interaction techniques from safe RL (Gu et al., 2024b), where agent actions are constrained within predefined safe regions to ensure secure interactions. A complementary approach is human-centered safe learning (Gu et al., 2023a), particularly suited for agentic web environments, in which human expertise guides exploitation while LLMs drive exploration within a safe RL framework. This setup can be framed as a multi-objective optimization problem that balances safety, performance, and coverage across diverse web operations and interaction goals. Recent advances show that such trade-offs can be

effectively managed using advanced safe RL methods (Gu et al., 2024a; 2025), offering a principled pathway to unify human-in-the-loop and automated red teaming for robust agentic web safety and security.

In particular, several open challenges remain in using LLMs for red teaming in the context of agentic web safety and security:

- **Red-Teaming Attack in Agentic Web:** Red-teaming frameworks within agentic web environments may themselves become targets of adversarial attacks, especially during complex multi-agent interactions across dynamic web platforms. Such compromises can mislead the evaluation process and result in the leakage of sensitive or private data from the underlying systems, undermining both safety and trustworthiness.
- **Emergent Misalignment in Agentic Web Agents:** As language models scale and operate over longer contexts in real-time web environments, red-teaming techniques may fail to detect novel and complex failure modes. A key risk is unauthorized goal generalization, where LLM-driven agents pursue objectives beyond their intended scope, such as executing unintended actions on web services, due to misaligned reasoning during test-time interactions. This poses critical safety and security risks for open-ended, autonomous agentic web applications.

Addressing these emerging challenges in agentic web safety and security requires interdisciplinary collaboration across red teaming methodologies, agent alignment strategies, evaluation frameworks, and robust deployment protocols.

7.3 Safety and Security Defense

The above subsection has discussed approaches that could identify potential threats or vulnerabilities of LLM agents during their development stages. In this subsection, we will focus on defensive research efforts that can further mitigate the safety issues of these agents at deployment. Specifically, we will discuss recent approaches that leverage external models for threat mitigation (i.e. guardrails), and approaches that steer LLM agents towards safer generation or planning. We will further discuss several emergent challenges in this context.

7.3.1 Inference-time Guardrails

The recent proliferation of LLMs has sparked growing interest in the development of guardrails, which serve as external safety mechanisms designed to identify and mitigate potentially harmful inputs or outputs of LLMs (Markov et al., 2023; Inan et al., 2023; Chi et al., 2024; Han et al., 2024). These guardrails have attracted attention due to their adaptability across different LLM implementations and their effectiveness in risk mitigation.

Earlier guardrails developed prior to the emergence of agentic AI primarily framed content moderation as a discriminative task, wherein a model classifies inputs (and sometimes also outputs) of LLMs as either safe or unsafe, or categorizes them into specific classes of harmful content (Wen et al., 2025). Initial approaches to guardrails relied heavily on rule-based filtering techniques (Welbl et al., 2021; Clarke et al., 2023; Gómez et al., 2024) that utilize predefined lexicons or heuristic rules to identify potentially unsafe content. While these rule-based methods offer transparency and computational efficiency, they inherently lack flexibility and generalization capabilities. Subsequent studies have transitioned towards model-based guardrails, leveraging supervised fine-tuning on curated safety datasets to enhance content classification in alignment with predefined safety policies. Representative open-source examples in this category include LLaMA Guard (Inan et al., 2023; Fedorov et al., 2024; Chi et al., 2024), NeMo Guardrails (Rebedea et al., 2023) and Aegis Guard (Ghosh et al., 2024).

To advance LLM guardrails toward more agentic capabilities, two key developments are essential: the integration of deliberative reasoning and lifelong learnability. Deliberative reasoning enables models

to assess actions more reflectively and align responses with nuanced goals and values. Lifelong learnability allows systems to adapt continuously to new information, evolving norms, and edge cases over time. Together, these capabilities form the foundation for more robust, context-sensitive guardrails that go beyond static filters to support safe and reliable autonomous behavior. In the following, we discuss recent studies on these two key lines of development.

Reasoning Guardrails. As a critical but very recent advancement of guardrails, reasoning guardrails assess the intent, context, and possible risks associated with LLMs' inputs and outputs through structured reasoning, instead of conduct fast-thinking to merely predict threat labels like previous static guardrails.

As one of the very first reasoning guardrails, ThinkGuard (Wen et al., 2025) draws inspiration from cognitive theories that differentiate fast and slow modes of human thinking (Hagendorff et al., 2022; Min et al., 2024). In this context, fast thinking typically results in superficial or incorrect assessments, making models susceptible to adversarial manipulation, whereas deliberative reasoning mitigates these vulnerabilities by facilitating more robust and contextually informed decisions (Lin et al., 2024a). To train guardrails capable of deliberative reasoning, ThinkGuard utilizes mission-focused distillation (Zhou et al., 2024) extracting structured reasoning supervision from existing LLMs to generate augmented safety datasets with reasoning critiques. Deliberative reasoning is integrated into the guardrail through a two-stage conversational fine-tuning procedure: the first stage outputs an initial prediction, followed by a second stage that articulates the underlying reasoning. This mechanism allows for the efficiency of traditional LLM-based guardrails if reasoning generation is opt out while retaining interpretability when needed. As a contemporaneous work, GuardReasoner (Liu et al., 2025d) also devise a similar process to augment safety supervision data with distillation, while its training has two differences: it conducts RL to finetune the guardrail model, but does not learn an optional latent reasoning mode as ThinkGuard. Based on the methodology exemplified by ThinkGuard and GuardReasoner, more recent efforts have further extended reasoning guardrails to multimodal (Liu et al., 2025e) and multilingual (Yang et al., 2025b) scenarios.

To summarize, developing the reasoning guardrail models is a prerequisite to realizing Agentic Guardrails because agentic behavior involves multi-step planning and decision-making that cannot be reliably constrained by surface-level filters or static rules. Reasoning Guardrails enable structured oversight of an agent's internal deliberations, allowing alignment interventions at the level of goals, plans, and justifications.

Agentic Guardrails. In contrast to the aforementioned reasoning guardrails, agentic guardrails operate at the level of action execution, overseeing or constraining how an agent interacts with external systems or environments. While reasoning guardrails shape what the agent thinks and tells, agentic guardrails govern what the agent does.

Developing agent guardrails presents a range of challenges spanning both technical and contextual dimensions. A key difficulty lies in enabling *lifelong learnability*, where guardrails must adapt alongside evolving agent behaviors through its continual interaction with users and the environment. Beyond managing threats from user inputs or model-generated content, agents must also detect and mitigate risks arising from external environments, including adversarial states or unsafe system interactions. Ensuring safety in multi-turn actions and dialogus adds further complexity, as harmful behavior may only emerge cumulatively over extended interactions. Finally, generalizability across tasks and environments remains a core obstacle, requiring guardrails that are robust and effective in diverse, dynamic deployment settings without extensive reconfiguration.

A few recent studies have attempted to address some of these challenges, inasmuch as some recent studies on lifelong agents (Huang et al., 2025a; Zhang et al., 2025b) propose to incorporate continual memories, works such as AGrail (Luo et al., 2025), LlamaFirewall (Chennabasappa et al., 2025) GuardAgent (Xiang et al., 2025) leverage this methodology to allow guardrail agents to continually accumulate experiences and new safety policies for agents. For example, AGrail (Luo et al., 2025) proposes a lifelong guardrail framework for LLM agents that dynamically generates and optimizes

safety checks during runtime. Central to its approach is a memory-based mechanism that stores and reuses past unsafe trajectories and guardrail refinements, enabling lifelong learnability and continual improvement. It combines adaptive safety-check synthesis with iterative refinement using cooperative LLMs and supports tool-assisted validation, allowing it to address both task-specific (e.g., prompt injection) and environmental threats. AGrail demonstrates strong cross-task generalization and effective integration with various LLM-based agents. LlamaFirewall (Chennabasappa et al., 2025) is a modular, open-source framework designed to secure LLM agents through a combination of real-time defenses. It integrates a fine-tuned BERT-style model for jailbreak and prompt-injection detection, a preliminary auditor leveraging few-shot chain-of-thought prompting for reasoning, and an online system tailored for LLM-generated programs for static analysis of generated code. The system achieves strong security performance with minimal utility loss and is built to be extensible across diverse agent applications. Meanwhile, GuardAgent (Xiang et al., 2025) dynamically monitors and enforces user-defined safety or privacy policies by translating guard requests into executable code through a two-step process of task planning and code generation. It leverages a memory-based in-context demonstration mechanism that retrieves past examples at each step to enhance its reasoning and support lifelong learnability and adaptability.

7.3.2 Controllable Generation and Planning

In addition to leveraging external guardrail components, other research efforts investigate controllable generation to steer LLM agents towards safer generation or planning at inference. In this context, we categorize these efforts into safe decoding approaches, and approaches for agentic access control.

Safe decoding approaches typically incorporate constrained decoding or decoding processes guided by a safety reward to achieve their goals. For example, SafeDecoding (Xu et al., 2024c) leverages the insight that even under attack the model still assigns non-trivial probability to safe tokens, and dynamically reshapes the token distribution at each decoding step to prioritize harmless outputs. SafePlanner (Li et al., 2025b) introduces a framework that enhances safety awareness in LLM agents for robot task planning. It incorporates a safety prediction module trained in a simulator, which guides the high-level planner to make safe and executable decisions. Thought-Aligner (Jiang et al., 2025) is a lightweight, plug-in safety module designed to enhance the behavioral safety of LLM-based agents by dynamically correcting high-risk reasoning steps before action execution. It operates by fine-tuning a contrastive learning model on a dataset of safe and unsafe thought pairs, enabling real-time thought correction with extremely low latency.

Access control remains an underexplored area in current research. Progent (Shi et al., 2025) introduces the first comprehensive privilege-control mechanism designed specifically for LLM-based agents, enforcing the principle of least privilege during tool invocation. It centers around a domain-specific policy language that allows developers and users to specify fine-grained constraints on when tools may be invoked and define fallback behaviors for blocked actions. This policy-driven model enforces the principle of least privilege, ensuring agents only perform tool operations essential to the task at hand. Its modular architecture enables seamless integration into existing agent systems with minimal code modifications and without altering the agent's internal logic. To lower the burden on users, Progent also supports automated policy generation and updates, leveraging LLMs themselves to craft and adapt these privilege policies dynamically in response to evolving user queries. As such, Progent offers a practical and flexible mechanism for enhancing LLM-agent security in diverse, real-world scenarios. In a related context, *knowledge access control* is a newly identified and unaddressed problem that concerns dynamic adjustment of LLMs' parametric knowledge based on user privileges (Liu et al., 2025b). Traditional safe generation methods typically adopt a uniform policy that blocks sensitive knowledge for all users, potentially reducing utility for credentialed individuals with legitimate access needs. To address this limitation, the SudoLM framework (Liu et al., 2025b) introduces a credential-aware mechanism that grants access to privileged knowledge only when a secret SUDO key is provided. This approach partitions the model's knowledge into public and privileged components and trains it using authorization alignment, enabling differentiated responses

based on user credentials. Such a framework offers a promising direction for enhancing safety in LLM-based agents. By conditioning access on user identity, intent, or role, it enables finer-grained, context-sensitive safeguards. Furthermore, its capacity to regulate internal knowledge usage rather than just output filtering allows deeper integration into agent reasoning and planning workflows.

7.3.3 Emerging Directions in Defense for Agentic Web

Threat mitigation for agents after their deployment is so far a preliminary area of study. Beyond the current prototypes in existing preliminary studies, and there are quite a few emergent challenges towards truly generalizable and reliable approaches, for which we briefly discuss a selected set of them as follows.

- **Efficiency:** As discussed, the reasoning-based paradigm no doubt strengthens the guardrails in terms of robustness and interpretability. Yet, it inevitably introduces more inference overhead in comparison to previous fast-thinking or discriminative guardrails. While a few current reasoning guardrails such as ThinkGuard (Wen et al., 2025) have attempted to incorporate latent reasoning to some degrees, how to effectively compress the reasoning process and enhance the generation or retrieval of reasoning patterns for a real-time guardrail remains as a non-trivial challenge.
- **Generalizability:** Generalizability is still a core challenge for Web agent safety because these agents operate in open, unpredictable environments with constantly changing interfaces, tools, and tasks. Guardrails that work in training or on benchmarks often fail when agents encounter novel websites or instructions. Unsafe behavior can emerge from unforeseen input combinations or even from new environments. Therefore, ensuring the robustness to diverse and evolving scenarios—not just effective in fixed settings as it is shown in many of the current experimental setups.
- **Certifiable defense:** Certifiable and grounded defense is crucial for web agents because they interact directly with external systems and users, where failures can lead to real-world harm (e.g., sending emails, making purchases, or modifying files). Without grounded verification tied to the actual environment state (e.g., DOM structure, user intent, API constraints), safety mechanisms may rely on incomplete or incorrect assumptions. Future research should develop safety mechanisms that are formally grounded in the web environment, such as DOM structures and API schemas, enabling agents to verify the safety of their actions before execution. It should also explore certifiable control and runtime verification techniques that ensure actions adhere to defined constraints, even under dynamic or adversarial conditions.

7.4 Safety and Security Evaluation

Unlike the relatively well-studied areas of traditional web safety (bt Mohd and Zaaba, 2019; Cox et al., 2006), LLM safety evaluation (Yuan et al., 2024; 2025), multimodal safety (Xu et al., 2025), and robot learning safety (Gu et al., 2023b), agentic web safety evaluation remains largely underexplored, with only a few preliminary efforts proposed to date.

One recent work is SafeArena (Tur et al., 2025), which introduces a benchmark designed to assess the misuse potential of LLM-based web agents. It evaluates agents on 250 safe and 250 harmful tasks across multiple harm categories, such as misinformation, cybercrime, and social bias, and tests models including ChatGPT (Ouyang et al., 2022; Achiam et al., 2023) and Qwen (Bai et al., 2023) to measure their compliance with malicious requests.

Another work is ST-WebAgentBench (Levy et al., 2024), which is an open-source benchmark for evaluating the safety and trustworthiness of autonomous web agents in enterprise-style tasks, built on the WebArena environment. It defines six policy dimensions, user consent, preference satisfaction, scope boundaries, strict execution, robustness to distribution shifts, and error recovery, and adopts

completion under policy and risk ratio, which measure policy-compliant task success and frequency of violations, respectively. Similarly, AGrail (Luo et al., 2025) introduces Safe-OS, a realistic benchmark designed to evaluate the safety of LLM-powered operating system agents under adversarial conditions. Comprised of three carefully curated attack scenarios, prompt injection, environment sabotage, and system-level exploitation, Safe-OS simulates real-world threats using Docker-based OS environments alongside benign operation logs. It complements evaluations on existing datasets for operating system agents like Mind2Web-SC, EICU-AC (task-specific risks; Xiang et al. (2024)), AdvWeb (Xu et al., 2024a), and EIA (systemic risks; Liao et al. (2025)) to offer a comprehensive safety assessment

Agent-SafetyBench (Zhang et al., 2024b) provides a comprehensive safety evaluation framework with 349 interaction environments and 2,000 test cases across eight safety risk categories. Notably, their evaluation reveals that no current agent achieves safety scores above 60%, highlighting fundamental deficiencies in agent robustness and risk awareness. Complementing this, GuardAgent (Xiang et al., 2024) introduces a dynamic safety guardrail system that achieves 98% accuracy on safety-critical tasks through knowledge-enabled reasoning, while TrustAgent (Hua et al.) implements a three-stage safety strategy encompassing pre-planning knowledge injection, in-planning enhancement, and post-planning inspection.

While benchmarks like SafeArena (Tur et al., 2025) and ST-WebAgentBench (Levy et al., 2024) provide valuable insights into agentic web safety, further investigation is needed, particularly in areas such as multimodal agentic web safety and reasoning safety for agentic web agents.

8 Challenges and Open Problems

The realization of the vision of Agentic Web, however, is contingent upon resolving a complex, multi-dimensional set of challenges that span individual agent cognition, multi-agent coordination, human-agent alignment, systemic security, and socio-economic structures.

These challenges are not isolated technical hurdles but are deeply interconnected, forming a web of dependencies that must be addressed holistically. The problem of building the Agentic Web is not merely about improving the capabilities of individual LLM or Agent but about architecting a new, reliable, and trustworthy computational layer atop the existing internet. The systemic nature of these challenges is evident in how they cascade across domains. For instance, the technical need for agents to interact with the external world necessitates the creation of standardized communication protocols, which have been likened to "HTTP for AI agents". The existence of this new agent-native architecture, in turn, creates new economic imperatives. The traditional advertising-based business model of the web is ill-suited for an agent-driven economy and is already showing signs of strain. This necessitates new transactional models, but their viability depends directly on solving complex security and trust issues surrounding autonomous payments. Thus, a technical challenge in one area, such as secure tool use, is inextricably linked to a socio-economic challenge in another, such as creating viable business models. A systems-thinking approach is therefore essential, recognizing that a solution for one component may create or exacerbate problems elsewhere. The following table provides a conceptual map of this complex problem space, categorizing the diverse challenges into coherent themes that will be explored throughout this report.

8.1 Foundational Challenges in Single-Agent Cognition and Autonomy

Before complex multi-agent and human-agent systems can be reliably constructed, the core cognitive architecture of an individual agent must be made robust. This section deconstructs the fundamental technical hurdles that currently undermine the reliability, planning capabilities, and autonomous functioning of a single agent. These are the first-order problems that form the bedrock of the Agentic Web.

Table 8: A Taxonomy of Agentic Web Challenges

Challenge Category	Core Problem	Key Open Questions
Foundational Cognition	Brittle Reasoning & Planning	How can agents achieve robust, long-horizon planning under uncertainty?
	Memory & Context Management	How can we build structured, hierarchical memory systems for agents?
	Reliable Tool Use	How can agents safely and reliably use external tools that may be compromised?
Learning Curriculum	Reward Design & Alignment	How can we design reward functions that capture nuanced human goals without being gamed?
	Continual Learning & Forgetting	How can agents acquire new skills over time without catastrophically forgetting old ones?
	Interactive Grounding	How can agents learn from interaction without overfitting to specific environments or prompts?
Collaborative Ecosystem	Inter-Agent Coordination	How can decentralized agents effectively coordinate and resolve conflicts?
	Communication & Interoperability	What communication standards are needed for a global, open agentic web?
	Decentralized Trust	How can agents establish and maintain trust in a decentralized, potentially adversarial ecosystem?
Human-Agent Alignment	Goal Ambiguity & Disambiguation	How can an agent reliably infer a user's true intent from ambiguous instructions?
	Preference Elicitation	How can agents help users discover and articulate their own complex preferences?
	Oversight & Control (HITL)	What are the most effective architectures for human-in-the-loop oversight?
Systemic Risk & Robustness	Security & Attack Surfaces	How can we defend agents against novel threats like tool-initiated attacks?
	Error Recovery & Resilience	How can we engineer agentic systems to be resilient to inevitable failures?
	Autonomous Payments	What technical and regulatory frameworks are needed for secure agent-based payments?
Socio-Economic Impact	New Business Models	What viable business models will replace the advertising-based economy?
	Economic Disruption & Inequality	How can the economic benefits of agentic AI be distributed equitably?

The Fragility of Reasoning and Planning The capacity for multi-step reasoning is a cornerstone of agentic systems, enabling them to decompose complex problems, evaluate alternative solutions, and make informed decisions. This process is often operationalized through a continuous cycle of planning, action, observation, and reflection, with frameworks like Chain-of-Thought serving

as the primary mechanism for articulating these reasoning steps in natural language. However, this capability is deceptively brittle.

The Memory-Context Dilemma Memory is an essential architectural component for agentic systems. Since LLMs are fundamentally stateless, they require external mechanisms to retain conversation history, contextual information, and learned knowledge. Agentic architectures typically employ both short-term memory to maintain coherence within a single task and long-term memory to carry knowledge across tasks. However, the management of this memory, especially in the face of finite context windows and complex, long-horizon tasks, remains a primary bottleneck.

The Tool-Use Paradox The ability to use external tools, such as APIs, databases, calculators, and web search, is what transforms a passive LLM into an active agent capable of interacting with and affecting the real world. This is the primary mechanism for grounding an agent's reasoning in actionable reality. However, this capability introduces a fundamental paradox: the every tools that grant an agent real-world agency simultaneously represent its greatest vulnerability.

This creates a "Tool-Use Paradox": to be effective, an agent must trust its tools to provide accurate information and execute actions correctly; to be secure, it must assume any tool could be compromised at any time. The resolution to this paradox lies in designing agents with an inherent "tool skepticism." This requires moving to a zero-trust agent architecture where all external inputs, whether from a user or a tool, are validated against a security policy. Previously, agent security focused primarily on validating the user's prompt. The existence of tool-initiated threats means the agent must also validate the tool's response, creating a feedback loop of potential infection where a malicious tool output could cause the agent to take another malicious action, leading to a cascading failure. A secure agent cannot be a naive "tool-caller"; it must possess a security kernel or policy engine that scrutinizes all information crossing the boundary between its internal state and the external world. The open research question is how to build this skepticism without crippling the agent's ability to act decisively based on tool outputs.

8.2 The Learning Conundrum: From Static Models to Dynamic Learners

While foundational models provide a powerful starting point, true agency requires the ability to learn from experience, adapt to new environments, and continuously improve performance. This section explores the profound challenges associated with transforming static, pre-trained models into dynamic, lifelong learners, focusing on the bottlenecks in RL, the threat of catastrophic forgetting, and the difficulties of grounding knowledge through interaction.

Reward Design Bottleneck RL is the primary paradigm for training agents to make optimal sequential decisions by interacting with an environment. It offers a path to move beyond the limitations of static, pattern-replicating LLMs, enabling them to handle ambiguity, maintain context in long conversations, and balance competing objectives. However, the effectiveness of RL is critically dependent on the design of its reward function, which has become a major research bottleneck.

The Specter of Catastrophic Forgetting in Continual Learning For agents to be truly autonomous and useful over long periods, they must be able to engage in continual, or lifelong, learning: acquiring new knowledge and skills without overwriting or degrading previously learned capabilities. The primary obstacle to achieving this is "catastrophic forgetting," a well-known phenomenon in neural networks where training on a new task causes a model to abruptly lose proficiency on previously learned tasks.

Interactive Task Learning and Grounding Ultimately, agents learn to perform complex tasks by interacting directly with their environment, whether it is a digital application or the physical world. This interactive learning process, often guided by RL, is what allows agents to ground their abstract knowledge in concrete actions and feedback. However, this process is fraught with challenges related to the trade-off between specialization and generalization.

8.3 The Ecosystem Challenge: Coordination and Trust in Multi-Agent Systems

Expanding the analysis from the single agent to the collective reveals the profound complexities that arise when multiple autonomous agents must interact, collaborate, and compete. The success of the Agentic Web as a whole depends on the ability to orchestrate these multi-agent systems effectively, a challenge that encompasses architectural design, communication standards, and the establishment of trust in decentralized environments.

Architectural Trade-offs: Hierarchical, Equi-level, and Nested Structures Multi-agent systems can be organized into several distinct architectures, each with unique properties and challenges. The primary structures identified in current research are equi-level (peer-to-peer), hierarchical (leader-follower), and nested (hybrid systems of systems).

The Babel of Agents: The Imperative for Standardized Communication For a global Agentic Web to function, agents developed by different organizations on different platforms must be able to communicate and interoperate. Without common standards, the ecosystem would devolve into a collection of isolated, proprietary "walled gardens," akin to the pre-HTTP internet, stifling innovation and collaboration.

The primary challenge is to develop and adopt standardized communication protocols that are expressive enough to support complex agent interactions yet simple and open enough to foster widespread adoption. This involves standardizing both the syntax (the format of messages) and the semantics (the meaning of communicative acts, often based on speech act theory). Emerging standards like IBM's ACP and Google's A2A for agent-to-agent communication and Anthropic's MCP for agent-to-tool communication are designed to work in tandem to provide this foundational layer. Major industry players are championing these open protocols, arguing that achieving ubiquity is more important than perfecting minor semantic differences, in order to create a truly open agentic web.

Establishing Trust and Reputation in Decentralized Ecologies In a decentralized system of autonomous, potentially self-interested agents, trust is the essential lubricant that enables collaboration and reduces uncertainty. To make informed decisions about whom to interact with and delegate tasks to, agents need a mechanism to assess the reliability and competence of their peers.

8.4 The Human-Agent Interface: Ensuring Goal Alignment and Control

This section focuses on the critical interface between human users and autonomous agents. The central challenge is ensuring that an agent's actions faithfully reflect the user's true, often nuanced and evolving, intent. This requires solving deep problems of goal ambiguity, preference discovery, and the design of effective oversight mechanisms to maintain human control.

The Ambiguity Problem: From User Intent to Actionable Goals The first step in any agentic workflow is understanding the user's goal. However, human language is often imprecise, and user requests can be complex, ambiguous, or underspecified. An agent must be able to disambiguate this input and translate it into a concrete, actionable plan. This often involves a process of active disambiguation, where the agent poses clarifying questions to maximize information gain and narrow the space of possible interpretations (Jiang et al., 2024).

Eliciting Nuanced Preferences A significant challenge in achieving goal alignment is that users themselves often do not have fully formed, stable preferences. A substantial body of psychology research has demonstrated that preferences are often constructed "on the fly" at the time of decision-making, influenced by the immediate context and the options presented (Lawless et al., 2024).

An agent, therefore, cannot simply ask a user for their complete utility function. Instead, it must engage in an iterative, collaborative process of preference elicitation, helping the user to discover,

construct, and refine their own preferences over time. This requires the agent to move beyond a passive question-answer model to become an active participant in the user's reasoning process. The cost and accuracy of a user's responses to preference queries are highly dependent on context; users respond more easily and accurately to queries about situations they are currently or have recently experienced. This makes naturalistic, chat-based elicitation a promising approach.

Human-in-the-Loop (HITL): Designing Effective Oversight Architectures Given the current limitations in agent reliability and alignment, incorporating a "human in the loop" (HITL) is a critical mechanism for ensuring safety, accountability, and control, especially for high-stakes or irreversible actions. HITL represents a collaborative paradigm where humans and AI work together to optimize processes.

8.5 Systemic Risks: Ensuring Safety, Security, and Robustness

As agents become more autonomous and capable of taking real-world actions, the risks they pose escalate dramatically. This section addresses the critical challenges of ensuring that agentic systems are secure from attack, robust to failure, and safe to deploy in high-stakes environments such as finance and critical infrastructure.

Safety and Security Challenges We explore the safety and security challenges of the agentic web, where autonomous agents operate across open, dynamic environments. It categorizes threats across cognitive, communication, and economic layers and highlights cascading risks that amplify system vulnerabilities. To address these issues, both human-involved and automated red teaming are discussed, with LLM-driven approaches offering scalability but requiring robust oversight. Defense strategies include advanced guardrails with reasoning and lifelong learning, such as AGrail (Luo et al., 2025) and GuardAgent (Xiang et al., 2025). While benchmarks like SafeArena (Tur et al., 2025) and ST-WebAgentBench (Levy et al., 2024) have made progress in evaluation, significant gaps remain, particularly in multimodal and reasoning safety, calling for further research in scalable, adaptive safety solutions.

Long-Horizon Planning and Error Recovery Real-world tasks are rarely simple, single-step operations. They often involve long-horizon plans with numerous sequential and parallel actions. In complex and partially observable environments, failures are not a possibility but a certainty.

The dual challenges are (1) creating and maintaining a coherent plan over a long sequence of steps without the plan degrading or becoming irrelevant, and (2) building in robust mechanisms for detecting, diagnosing, and recovering from the inevitable errors, exceptions, and action failures that will occur. Simple, sequential agentic chains that work well in prototypes often break under the variability and load of real-world use because they lack graceful failure modes and recovery paths. A key to robustness is moving beyond open-loop "plan-and-execute" paradigms to closed-loop systems that can self-correct based on feedback from their actions (Nayak et al., 2024).

The Challenge of Autonomous Payments: Security and Regulation Empowering agents with the ability to spend money is a critical enabler for a transactional Agentic Web, but it also represents one of the highest-risk applications, facing immense technical, regulatory, and social hurdles.

8.6 Socio-Economic Implications

The successful deployment of the Agentic Web would not be a mere technological evolution but a profound socio-economic transformation, reshaping business models, labor markets, and the very structure of the digital economy. This section explores the challenges and open questions related to the economic viability and societal impact of this new paradigm.

Beyond Advertising: Viable Business Models for an Agentic Economy The current economic foundation of the consumer web, advertising, is ill-suited for and actively threatened by the rise of AI agents. The Agentic Web necessitates a shift towards new, more direct forms of value exchange.

The ad-supported model, which monetizes human attention, is breaking down as agents become the primary interface for information retrieval, disintermediating and reducing traffic to content websites. The challenge is to develop and scale new business models that are native to an economy of automated actions, not human eyeballs. This likely involves a move towards transactional, subscription, and value-based pricing models. Emerging models already position agents as first-class business entities that can be customized and deployed by organizations, enabling new revenue streams. These include: *Intelligence-as-a-Service*, where the outputs of AI-powered research are sold on demand; *Zero Marginal Cost Services*, where the incremental cost of serving another customer is near zero; and *Value-Based Pricing*, where customers pay for outcomes rather than time invested.

Furthermore, the integration of blockchain technology presents promising opportunities for the Agentic Web's economic foundation. Blockchain-enabled platforms can facilitate decentralized agent interactions, autonomous transactions, and trustless value exchange between AI agents. Projects like ChainOpera (ChainOpera AI, 2024) demonstrate the practical convergence of Web3 and agentic AI, while emerging protocols such as Protocol AI (Protocol AI, 2025) support agent-blockchain integration for decentralized tokenization of alternative assets (Borjigin et al., 2025) and autonomous operations in decentralized finance (Ante, 2024). This convergence could enable new forms of autonomous economic activity where agents can independently engage in value creation and exchange without traditional intermediaries.

Economic Disruption: Labor Markets, Productivity, and Inequality The widespread adoption of AI agents promises massive gains in productivity and economic growth but also portends significant disruption to the labor market and carries the risk of exacerbating economic inequality. Research suggests that generative AI alone could add trillions of dollars annually to the global economy, but it could also automate a significant fraction of current work activities, affecting hundreds of millions of jobs worldwide.

9 Conclusion

The internet is undergoing a fundamental paradigm shift, evolving from a passive repository of information to a dynamic environment of action. This transition is powered by the emergence of the Agentic Web, a landscape populated by autonomous systems capable of perceiving their environment, reasoning through complex problems, and executing tasks to achieve specified goals. This marks a significant leap from generative AI, which excels at responding to human prompts, to agentic AI, which is characterized by proactive, independent decision-making and execution.

References

Deepak Bhaskar Acharya, Karthigeyan Kuppan, and B. Divya. Agentic ai: Autonomous intelligence for complex goals—a comprehensive survey. *IEEE Access*, 13:18912–18936, 2025. doi:10.1109/ACCESS.2025.3532853.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Adobe. Our vision for accelerating creativity and productivity with agentic ai. Adobe Blog, 2025. URL https://blog.adobe.com/en/publish/2025/04/09/our-vision-for-accelerating-creativity-productivity-with-agentic-ai.

Linux Foundation AI and IBM Data. Acp: Agent communication protocol, 2025. URL https://agentcommunicationprotocol.dev/introduction/welcome. Accessed: 2025-04-22.

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*, 2016.

Lennart Ante. Autonomous ai agents in decentralized finance: Market dynamics, application areas, and theoretical implications. *SSRN Electronic Journal*, 2024. doi:10.2139/ssrn.5055677.

Anthropic. Introducing the model context protocol. Anthropic Blog, November 2024. URL https://www.anthropic.com/news/model-context-protocol.

Anthropic. Computer use tool. https://docs.anthropic.com/en/docs/agents-and-tools/tool-use/computer-use-tool, 2024. Accessed: 2025-07-20.

Anthropic. How we built our multi-agent research system. Anthropic Engineering Blog, 2024a. URL https://www.anthropic.com/engineering/built-multi-agent-research-system.

Anthropic. Model context protocol, 2024b. URL https://www.anthropic.com/news/model-context-protocol. Accessed: 2025-04-19.

Anthropic. How we built our multi-agent research system, 2025. URL https://www.anthropic.com/engineering/built-multi-agent-research-system. Accessed: 2025-07-23.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*, 2024.

Dolev Moshe Attiya. Cato CTRL™ Threat Research: Exploiting Model Context Protocol (MCP) – Demonstrating Risks and Mitigating GenAI Threats. https://www.catonetworks.com/blog/cato-ctrl-exploiting-model-context-protocol-mcp/, April 2025. Accessed: 2025-07-03.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

Tim Berners-Lee. *Weaving the Web: The original design and ultimate destiny of the World Wide Web by its inventor*. Harper San Francisco, 1999.

Ailiya Borjigin et al. Ai-governed agent architecture for web-trustworthy tokenization of alternative assets. *arXiv preprint arXiv:2507.00096*, 2025. URL https://arxiv.org/abs/2507.00096.

Lars Braubach, Kai Jander, and Alexander Pokahr. A novel distributed registry approach for efficient and resilient service discovery in megascale distributed systems? *Computer Science and Information Systems*, 15(3):751–774, 2018. doi:10.2298/CSIS180131030B.

Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, 30(1-7):107–117, 1998.

Andrei Broder. A taxonomy of web search. *SIGIR Forum*, 36(2):3–10, September 2002. ISSN 0163-5840. doi:10.1145/792550.792552. URL https://doi.org/10.1145/792550.792552.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alexander Herzog, Jasmine Hsu, Brian Ichter, Alex Irpan, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Lisa Lee, Tsang-Wei Edward Lee, Sergey Levine, Yao Lu, Henryk Michalewski, Igor Mordatch, Karl Pertsch, Kanishka Rao, Krista Reymann, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Pierre Sermanet, Jaspiar Singh, Anikait Singh, Radu Soricut, Huong Tran, Vincent Vanhoucke, Quan Vuong, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Jialin Wu, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich.

Rt-2: Vision-language-action models transfer web knowledge to robotic control, 2023a. URL https://arxiv.org/abs/2307.15818.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil J Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong Tran, Vincent Vanhoucke, Steve Vega, Quan Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-1: Robotics transformer for real-world control at scale, 2023b. URL https://arxiv.org/abs/2212.06817.

Miles Brundage, Shahar Avin, Jack Clark, Helen Toner, Peter Eckersley, Ben Garfinkel, Allan Dafoe, Paul Scharre, Thomas Zeitzoff, Bobby Filar, et al. The malicious use of artificial intelligence: Forecasting, prevention, and mitigation. *arXiv preprint arXiv:1802.07228*, 2018.

Nur Azimah bt Mohd and Zarul Fitri Zaaba. A review of usability and security evaluation model of ecommerce website. *Procedia Computer Science*, 161:1199–1205, 2019.

Manuel Castells. *The Internet galaxy: Reflections on the Internet, business, and society*. Oxford University Press, 2002.

Tomer Jordi Chaffer. Know your agent: Governing ai identity on the agentic web. *Available at SSRN 5162127*, 2025.

ChainOpera AI. Chainopera ai: The blockchain and protocol for co-owning and co-creating decentralized ai apps and agents for humanity, 2024. URL https://www.chainopera.ai/. Decentralized AI Platform and Generative AI Application Ecosystem.

Gaowei Chang. Anp: Agent network protocol, 2024. URL https://www.agent-network-protocol.com/. Accessed: 2025-04-21.

Jerry Chen, Yiming Wang, Arjun Gupta, et al. Langchain: Framework for building agentic multi-agent language workflows. *arXiv preprint arXiv:2308.12345*, 2023a.

Jie Chen, Yan Liu, and Mugen Peng. Intent-driven closed-loop control and management framework for 6g open ran. *IEEE Transactions on Network and Service Management*, 21(1):15–28, 2024a.

Junjie Chen, Haitao Li, Jingli Yang, Yiqun Liu, and Qingyao Ai. Enhancing llm-based agents via global planning and hierarchical execution. *arXiv preprint arXiv:2504.16563*, 2025.

Minmin Chen, Alex Beutel, Paul Covington, Sagar Jain, Francois Belletti, and Ed H Chi. Top-k off-policy correction for a reinforce recommender system. In *Proceedings of the twelfth ACM international conference on web search and data mining*, pages 456–464, 2019.

Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chen Qian, Chi-Min Chan, Yujia Qin, Yaxi Lu, Ruobing Xie, et al. Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors in agents. *arXiv preprint arXiv:2308.10848*, 2(4):6, 2023b.

Weize Chen, Ziming You, Ran Li, Yitong Guan, Chen Qian, Chenyang Zhao, Cheng Yang, Ruobing Xie, Zhiyuan Liu, and Maosong Sun. Internet of agents: Weaving a web of heterogeneous agents for collaborative intelligence, 2024b. URL https://arxiv.org/abs/2407.07061.

Zhaorun Chen, Zhen Xiang, Chaowei Xiao, Dawn Song, and Bo Li. Agentpoison: Red-teaming llm agents via poisoning memory or knowledge bases. *Advances in Neural Information Processing Systems*, 37:130185–130213, 2024c.

Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, Rohan Anil, Zakaria Haque, Lichan Hong, Vihan Jain, Xiaobing Liu, and Hemal Shah. Wide & deep learning for recommender systems, 2016. URL https://arxiv.org/abs/1606.07792.

Sahana Chennabasappa, Cyrus Nikolaidis, Daniel Song, David Molnar, Stephanie Ding, Shengye Wan, Spencer Whitman, Lauren Deason, Nicholas Doucette, Abraham Montilla, et al. Llamafirewall: An open source guardrail system for building secure ai agents. *arXiv preprint arXiv:2505.03574*, 2025.

Jianfeng Chi, Ujjwal Karn, Hongyuan Zhan, Eric Smith, Javier Rando, Yiming Zhang, Kate Plawiak, Zacharie Delpierre Coudert, Kartikeya Upasani, and Mahesh Pasupuleti. Llama guard 3 vision: Safeguarding human-ai image understanding conversations. *CoRR*, abs/2411.10414, 2024. doi:10.48550/ARXIV.2411.10414. URL https://doi.org/10.48550/arXiv.2411.10414.

Jaymari Chua, Yun Li, Shiyi Yang, Chen Wang, and Lina Yao. Ai safety in generative ai large language models: A survey. *arXiv preprint arXiv:2407.18369*, 2024.

Giovanni Luca Ciampaglia, Alessandro Flammini, and Filippo Menczer. The production of information in the attention economy. *Scientific Reports*, 5(1), May 2015. ISSN 2045-2322. doi:10.1038/srep09452. URL http://dx.doi.org/10.1038/srep09452.

Galileo Cisco, Langchain. Acp: Agent connect protocol, 2025. URL https://spec.acp.agntcy.org/. Accessed: 2025-04-22.

Christopher Clarke, Matthew Hall, Gaurav Mittal, Ye Yu, Sandra Sajeev, Jason Mars, and Mei Chen. Rule by example: Harnessing logical rules for explainable hate speech detection. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 364–376, Toronto, Canada, July 2023. Association for Computational Linguistics. doi:10.18653/v1/2023.acl-long.22. URL https://aclanthology.org/2023.acl-long.22/.

Microsoft Corporation. Microsoft copilot for organizations. https://www.microsoft.com/zh-cn/microsoft-copilot/organizations, 2025. Accessed: July 21, 2025.

Richard S Cox, Jacob Gorm Hansen, Steven D Gribble, and Henry M Levy. A safety-oriented platform for web applications. In *2006 IEEE Symposium on Security and Privacy (S&P'06)*, pages 15–pp. IEEE, 2006.

Enfang Cui, Yujun Cheng, Rui She, Dan Liu, Zhiyuan Liang, Minxin Guo, Tianzheng Li, Qian Wei, Wenjuan Xing, and Zhijie Zhong. Agentdns: A root domain naming system for llm agents, 2025. URL https://arxiv.org/abs/2505.22368.

CyberArk Labs. Agents Under Attack: Threat Modeling Agentic AI. https://www.cyberark.com/resources/threat-research-blog/agents-under-attack-threat-modeling-agentic-ai, April 2025. Accessed: 2025-07-03.

Thomas H. Davenport and John C. Beck. The attention economy. *Ubiquity*, 2001(May):1–es, September 2018. doi:10.1145/376625.376626. URL https://doi.org/10.1145/376625.376626.

Herbert Dawid, Philipp Harting, Hankui Wang, Zhongli Wang, and Jiachen Yi. Agentic workflows for economic research: Design and implementation, 2025. URL https://arxiv.org/abs/2504.09736.

Pierre De Handschutter, Nicolas Gillis, and Xavier Siebert. A survey on deep matrix factorizations. *Computer Science Review*, 42:100423, November 2021. ISSN 1574-0137. doi:10.1016/j.cosrev.2021.100423. URL http://dx.doi.org/10.1016/j.cosrev.2021.100423.

Christian Schroeder de Witt. Open challenges in multi-agent security: Towards secure systems of interacting ai agents. *arXiv preprint arXiv:2505.02077*, 2025.

Mary Deaton. The elements of user experience: user-centered design for the web. *interactions*, 10 (5):49–51, 2003.

Edoardo Debenedetti, Jie Zhang, Mislav Balunovic, Luca Beurer-Kellner, Marc Fischer, and Florian Tramèr. Agentdojo: A dynamic environment to evaluate prompt injection attacks and defenses for llm agents. *Advances in Neural Information Processing Systems*, 37:82895–82920, 2024.

Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6), 1990.

Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Sam Stevens, Boshi Wang, Huan Sun, and Yu Su. Mind2web: Towards a generalist agent for the web. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 28091–28114. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/5950bf290a1570ea401bf98882128160-Paper-Datasets_and_Benchmarks.pdf.

Zehang Deng, Yongjian Guo, Changzhou Han, Wanlun Ma, Junwu Xiong, Sheng Wen, and Yang Xiang. Ai agents under threat: A survey of key security challenges and future pathways. *ACM Computing Surveys*, 57(7):1–36, 2025.

Zhi-Hong Deng, Ling Huang, Chang-Dong Wang, Jian-Huang Lai, and Philip S. Yu. Deepcf: A unified framework of representation learning and matching function learning in recommender system, 2019. URL https://arxiv.org/abs/1901.04704.

Emily Dinan, Samuel Humeau, Bharath Chintagunta, and Jason Weston. Build it break it fix it for dialogue safety: Robustness from adversarial human attack. *arXiv preprint arXiv:1908.06083*, 2019.

Yufei Ding, Haoran Geng, Chaoyi Xu, Xiaomeng Fang, Jiazhao Zhang, Songlin Wei, Qiyu Dai, Zhizheng Zhang, and He Wang. Open6dor: Benchmarking open-instruction 6-dof object rearrangement and a vlm-based approach. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7359–7366, 2024. doi:10.1109/IROS58592.2024.10802733.

Lucas Dixon, John Li, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. Measuring and mitigating unintended bias in text classification. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 67–73, 2018.

Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, Yevgen Chebotar, Pierre Sermanet, Daniel Duckworth, Sergey Levine, Vincent Vanhoucke, Karol Hausman, Marc Toussaint, Klaus Greff, Andy Zeng, Igor Mordatch, and Pete Florence. Palm-e: An embodied multimodal language model, 2023. URL https://arxiv.org/abs/2303.03378.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Benjamin Edelman, Michael Ostrovsky, and Michael Schwarz. Internet advertising and the generalized second-price auction: Selling billions of dollars worth of keywords. *American Economic Review*, 97(1):242–259, March 2007. doi:10.1257/aer.97.1.242. URL https://www.aeaweb.org/articles?id=10.1257/aer.97.1.242.

Lutfi Eren Erdogan, Nicholas Lee, Sehoon Kim, Suhong Moon, Hiroki Furuta, Gopala Anumanchipalli, Kurt Keutzer, and Amir Gholami. Plan-and-act: Improving planning of agents for long-horizon tasks. *arXiv preprint arXiv:2503.09572*, 2025.

Josef Falkinger. Attention economies. *Journal of Economic Theory*, 133(1):266–294, 2007. ISSN 0022-0531. doi:https://doi.org/10.1016/j.jet.2005.12.001. URL https://www.sciencedirect.com/science/article/pii/S0022053105002693.

Igor Fedorov, Kate Plawiak, Lemeng Wu, Tarek Elgamal, Naveen Suda, Eric Smith, Hongyuan Zhan, Jianfeng Chi, Yuriy Hulovatyy, Kimish Patel, Zechun Liu, Changsheng Zhao, Yangyang Shi, Tijmen Blankevoort, Mahesh Pasupuleti, Bilge Soran, Zacharie Delpierre Coudert, Rachad Alao, Raghuraman Krishnamoorthi, and Vikas Chandra. Llama guard 3-1b-int4: Compact and efficient safeguard for human-ai conversations. *CoRR*, abs/2411.17713, 2024. doi:10.48550/ARXIV.2411.17713. URL https://doi.org/10.48550/arXiv.2411.17713.

Zipeng Fu, Tony Z. Zhao, and Chelsea Finn. Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation. In *Conference on Robot Learning (CoRL)*, 2024.

Deep Ganguli, Liane Lovitt, Jackson Kernion, Amanda Askell, Yuntao Bai, Saurav Kadavath, Ben Mann, Ethan Perez, Nicholas Schiefer, Kamal Ndousse, et al. Red teaming language models to reduce harms: Methods, scaling behaviors, and lessons learned. *arXiv preprint arXiv:2209.07858*, 2022.

Sahaj Garg, Vincent Perot, Nicole Limtiaco, Ankur Taly, Ed H Chi, and Alex Beutel. Counterfactual fairness in text classification through robustness. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 219–226, 2019.

Suyu Ge, Chunting Zhou, Rui Hou, Madian Khabsa, Yi-Chia Wang, Qifan Wang, Jiawei Han, and Yuning Mao. Mart: Improving llm safety with multi-round automatic red-teaming. *arXiv preprint arXiv:2311.07689*, 2023.

Haoran Geng, Songlin Wei, Congyue Deng, Bokui Shen, He Wang, and Leonidas Guibas. Sage: Bridging semantic and actionable parts for generalizable articulated-object manipulation under language instructions, 2023.

Haoran Geng, Feishi Wang, Songlin Wei, Yuyang Li, Bangjun Wang, Boshi An, Charlie Tianyue Cheng, Haozhe Lou, Peihao Li, Yen-Jen Wang, Yutong Liang, Dylan Goetting, Chaoyi Xu, Haozhe Chen, Yuxi Qian, Yiran Geng, Jiageng Mao, Weikang Wan, Mingtong Zhang, Jiangran Lyu, Siheng Zhao, Jiazhao Zhang, Jialiang Zhang, Chengyang Zhao, Haoran Lu, Yufei Ding, Ran Gong, Yuran Wang, Yuxuan Kuang, Ruihai Wu, Baoxiong Jia, Carlo Sferrazza, Hao Dong, Siyuan Huang, Yue Wang, Jitendra Malik, and Pieter Abbeel. Roboverse: Towards a unified platform, dataset and benchmark for scalable and generalizable robot learning, 2025. URL https://arxiv.org/abs/2504.18904.

Genspark. Super agent. https://genspark.cloud/super-agent/, 2025. Accessed: 2025-07-20.

Shaona Ghosh, Prasoon Varshney, Erick Galinkin, and Christopher Parisien. AEGIS: online adaptive AI content safety moderation with ensemble of LLM experts. *CoRR*, abs/2404.05993, 2024. doi:10.48550/ARXIV.2404.05993. URL https://doi.org/10.48550/arXiv.2404.05993.

Amelia Glaese, Nat McAleese, Maja Trębacz, John Aslanides, Vlad Firoiu, Timo Ewalds, Maribeth Rauh, Laura Weidinger, Martin Chadwick, Phoebe Thacker, et al. Improving alignment of dialogue agents via targeted human judgements. *arXiv preprint arXiv:2209.14375*, 2022.

Juan Felipe Gómez, Caio Vieira Machado, Lucas Monteiro Paes, and Flávio P. Calmon. Algorithmic arbitrariness in content moderation. In *The 2024 ACM Conference on Fairness, Accountability, and Transparency, FAccT 2024, Rio de Janeiro, Brazil, June 3-6, 2024*, pages 2234–2253. ACM, 2024. doi:10.1145/3630106.3659036. URL https://doi.org/10.1145/3630106.3659036.

Google. Agent2agent(a2a) protocol. Google Blog, 2025a. URL https://a2a-protocol.org/latest/.

Google. A2a: Agent2agent protocol, 2025b. URL https://github.com/google/A2A. Accessed: 2025-04-21.

Google DeepMind Blog. Introducing gemini 2.0: our new ai model for the agentic era. https://blog.google/technology/google-deepmind/google-gemini-ai-update-december-2024/, December 2024. Mentions Project Mariner, agentic prototype.

Juraj Gottweis, Wei-Hung Weng, Alexander Daryin, Tao Tu, Anil Palepu, Petar Sirkovic, Artiom Myaskovsky, Felix Weissenberger, Keran Rong, Ryutaro Tanno, et al. Towards an ai co-scientist. *arXiv preprint arXiv:2502.18864*, 2025.

Shangding Gu, Alap Kshirsagar, Yali Du, Guang Chen, Jan Peters, and Alois Knoll. A human-centered safe robot reinforcement learning framework with interactive behaviors. *Frontiers in Neurorobotics*, 17:1280341, 2023a.

Shangding Gu, Jakub Grudzien Kuba, Yuanpei Chen, Yali Du, Long Yang, Alois Knoll, and Yaodong Yang. Safe multi-agent reinforcement learning for multi-robot control. *Artificial Intelligence*, 319: 103905, 2023b.

Shangding Gu, Bilgehan Sel, Yuhao Ding, Lu Wang, Qingwei Lin, Ming Jin, and Alois Knoll. Balance reward and safety optimization for safe reinforcement learning: A perspective of gradient manipulation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 21099–21106, 2024a.

Shangding Gu, Long Yang, Yali Du, Guang Chen, Florian Walter, Jun Wang, and Alois Knoll. A review of safe reinforcement learning: Methods, theories and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024b.

Shangding Gu, Bilgehan Sel, Yuhao Ding, Lu Wang, Qingwei Lin, Alois Knoll, and Ming Jin. Safe and balanced: A framework for constrained multi-objective reinforcement learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. Deepfm: A factorization-machine based neural network for ctr prediction, 2017. URL https://arxiv.org/abs/1703.04247.

Siyuan Guo, Cheng Deng, Ying Wen, Hechang Chen, Yi Chang, and Jun Wang. Ds-agent: Automated data science by empowering large language models with case-based reasoning, 2024. URL https://arxiv.org/abs/2402.17453.

Izzeddin Gur, Hiroki Furuta, Austin V Huang, Mustafa Safdari, Yutaka Matsuo, Douglas Eck, and Aleksandra Faust. A real-world webagent with planning, long context understanding, and program synthesis. In *The Twelfth International Conference on Learning Representations*, 2024.

Thilo Hagendorff, Sarah Fabi, and Michal Kosinski. Thinking fast and slow in large language models. *arXiv preprint arXiv:2212.05206*, 10, 2022.

Seungju Han, Kavel Rao, Allyson Ettinger, Liwei Jiang, Bill Yuchen Lin, Nathan Lambert, Yejin Choi, and Nouha Dziri. Wildguard: Open one-stop moderation tools for safety risks, jailbreaks, and refusals of llms. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/0f69b4b96a46f284b726fbd70f74fb3b-Abstract-Datasets_and_Benchmarks_Track.html.

Pengfei He, Yupin Lin, Shen Dong, Han Xu, Yue Xing, and Hui Liu. Red-teaming llm multi-agent systems via communication attacks. *arXiv preprint arXiv:2502.14847*, 2025.

Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering, 2017. URL https://arxiv.org/abs/1708.05031.

Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks, 2016. URL https://arxiv.org/abs/1511.06939.

Sirui Hong, Xiawu Zheng, Jonathan Chen, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, et al. Metagpt: Meta programming for multi-agent collaborative framework. *arXiv preprint arXiv:2308.00352*, 3(4):6, 2023.

Sirui Hong, Yizhang Lin, Bang Liu, Bangbang Liu, Binhao Wu, Ceyao Zhang, Chenxing Wei, Danyang Li, Jiaqi Chen, Jiayi Zhang, Jinlin Wang, Li Zhang, Lingyao Zhang, Min Yang, Mingchen Zhuge, Taicheng Guo, Tuo Zhou, Wei Tao, Xiangru Tang, Xiangtao Lu, Xiawu Zheng, Xinbing Liang, Yaying Fei, Yuheng Cheng, Zhibin Gou, Zongze Xu, and Chenglin Wu. Data interpreter: An llm agent for data science, 2024. URL https://arxiv.org/abs/2402.18679.

Xinyi Hou, Yanjie Zhao, Shenao Wang, and Haoyu Wang. Model context protocol (mcp): Landscape, security threats, and future research directions. *arXiv preprint arXiv:2503.23278*, 2025.

Mengkang Hu, Yuhang Zhou, Wendong Fan, Yuzhou Nie, Bowei Xia, Tao Sun, Ziyu Ye, Zhaoxuan Jin, Yingru Li, Qiguang Chen, et al. Owl: Optimized workforce learning for general multi-agent assistance in real-world task automation. *arXiv preprint arXiv:2505.23885*, 2025.

Wenyue Hua, Xianjun Yang, Mingyu Jin, Zelong Li, Wei Cheng, Ruixiang Tang, and Yongfeng Zhang. Trustagent: Towards safe and trustworthy llm-based agents through agent constitution. In *Trustworthy Multi-modal Foundation Models and AI Agents (TiFA)*.

Ken Huang. Agentic AI Threat Modeling Framework: MAESTRO. https://cloudsecurityalliance.org/blog/2025/02/06/agentic-ai-threat-modeling-framework-maestro, February 2025. Accessed: 2025-07-103.

Tenghao Huang, Kinjal Basu, Ibrahim Abdelaziz, Pavan Kapanipathi, Jonathan May, and Muhao Chen. R2d2: Remembering, reflecting and dynamic decision making for web agents. In *ACL*, 2025a.

Yuxuan Huang, Yihang Chen, Haozheng Zhang, Kang Li, Meng Fang, Linyi Yang, Xiaoguang Li, Lifeng Shang, Songcen Xu, Jianye Hao, Kun Shao, and Jun Wang. Deep research agents: A systematic examination and roadmap, 2025b. URL https://arxiv.org/abs/2506.18096.

Zhen Huang, Tao Zhang, and Hui Feng. Bandwidth-cache pricing-based network slicing for partially cached video streaming delivery. *IEEE Transactions on Multimedia*, 26:1120–1133, 2024.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

Eugene Ie, Vihan Jain, Jing Wang, Sanmit Narvekar, Ritesh Agarwal, Rui Wu, Heng-Tze Cheng, Tushar Chandra, and Craig Boutilier. Slateq: A tractable decomposition for reinforcement learning with recommendation sets. In *IJCAI*, volume 19, pages 2592–2599, 2019.

Hakan Inan, Kartikeya Upasani, Jianfeng Chi, Rashi Rungta, Krithika Iyer, Yuning Mao, Michael Tontchev, Qing Hu, Brian Fuller, Davide Testuggine, and Madian Khabsa. Llama guard: Llm-based input-output safeguard for human-ai conversations. *CoRR*, abs/2312.06674, 2023. doi:10.48550/ARXIV.2312.06674. URL https://doi.org/10.48550/arXiv.2312.06674.

Gautier Izacard and Edouard Grave. Leveraging passage retrieval with generative models for open domain question answering. In Paola Merlo, Jorg Tiedemann, and Reut Tsarfaty, editors, *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880, Online, April 2021. Association for Computational Linguistics.

Robin Jia and Percy Liang. Adversarial examples for evaluating reading comprehension systems. *arXiv preprint arXiv:1707.07328*, 2017.

Changyue Jiang, Xudong Pan, and Min Yang. Think twice before you act: Enhancing agent behavioral safety with thought correction. *arXiv preprint arXiv:2505.11063*, 2025.

Connie Jiang, Yiqing Xu, and David Hsu. Llms for robotic object disambiguation. *arXiv preprint arXiv:2401.03388*, 2024.

Yichen Jiang and Mohit Bansal. Avoiding reasoning shortcuts: Adversarial evaluation, training, and model development for multi-hop qa. *arXiv preprint arXiv:1906.07132*, 2019.

Zhengbao Jiang, Frank F Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7969–7992, 2023.

Ming Jin and Hyunin Lee. Position: Ai safety must embrace an antifragile perspective. In *Forty-second International Conference on Machine Learning*, 2025.

Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation, 2018. URL https://arxiv.org/abs/1808.09781.

Sayash Kapoor, Benedikt Stroebl, Zachary S Siegel, Nitya Nadgir, and Arvind Narayanan. Ai agents that matter. *arXiv preprint arXiv:2407.01502*, 2024.

Prathamesh Khade. Multi agent system for content creation. Medium, November 2024. URL https://medium.com/@prathamesh.khade20/multi-agent-system-for-content-creation-aaefa5350012.

Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, et al. Dynabench: Rethinking benchmarking in nlp. *arXiv preprint arXiv:2104.14337*, 2021.

Gangwoo Kim, Sungdong Kim, Byeongguk Jeon, Joonsuk Park, and Jaewoo Kang. Tree of clarifications: Answering ambiguous questions with retrieval-augmented large language models. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 996–1009, Singapore, December 2023. Association for Computational Linguistics.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvur, Ming Chong Lim, Po-Yu Huang, Graham Neubig, Shuyan Zhou, Ruslan Salakhutdinov, and Daniel Fried. Visualwebarena: Evaluating multimodal agents on realistic visual web tasks. *arXiv preprint arXiv:2401.13649*, 2024.

Noam Kolt. Governing ai agents, 2025. URL https://arxiv.org/abs/2501.07913.

Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

Yuxuan Kuang, Junjie Ye, Haoran Geng, Jiageng Mao, Congyue Deng, Leonidas Guibas, He Wang, and Yue Wang. Ram: Retrieval-based affordance transfer for generalizable zero-shot robotic manipulation, 2024. URL https://arxiv.org/abs/2407.04689.

Connor Lawless, Jakob Schoeffer, Lindy Le, Kael Rowan, Shilad Sen, Cristina St. Hill, Jina Suh, and Bahareh Sarrafzadeh. "i want it that way": Enabling interactive decision support using large language models and constraint programming. *ACM Transactions on Interactive Intelligent Systems*, 14(3):1–33, 2024.

Ido Levy, Ben Wiesel, Sami Marreed, Alon Oved, Avi Yaeli, and Segev Shlomov. St-webagentbench: A benchmark for evaluating safety and trustworthiness in web agents. *arXiv preprint arXiv:2410.06703*, 2024.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33: 9459–9474, 2020.

Hao Li et al. Towards secure semantic communications in the presence of intelligent eavesdroppers. *IEEE Transactions on Information Forensics and Security*, 20:1000–1015, 2025a.

Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670, 2010.

Siyuan Li, Feifan Liu, Lingfei Cui, Jiani Lu, Qinqin Xiao, Xirui Yang, Peng Liu, Kewu Sun, Zhe Ma, and Xun Wang. Safe planner: Empowering safety awareness in large pre-trained models for robot task planning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 14619–14627, 2025b.

Xiaoqi Li, Mingxu Zhang, Yiran Geng, Haoran Geng, Yuxing Long, Yan Shen, Renrui Zhang, Jiaming Liu, and Hao Dong. Manipllm: Embodied multimodal large language model for object-centric robotic manipulation, 2023.

Xinyi Li, Sai Wang, Siqi Zeng, Yu Wu, and Yi Yang. A survey on llm-based multi-agent systems: workflow, infrastructure, and challenges. *Vicinagearth*, page 9, 2024a.

Yuanchun Li, Hao Wen, Weijun Wang, Xiangyu Li, Yizhen Yuan, Guohong Liu, Jiacheng Liu, Wenxing Xu, Xiang Wang, Yi Sun, Rui Kong, Yile Wang, Hanfei Geng, Jian Luan, Xuefeng Jin, Zilong Ye, Guanjing Xiong, Fan Zhang, Xiang Li, Mengwei Xu, Zhijun Li, Peng Li, Yang Liu, Ya-Qin Zhang, and Yunxin Liu. Personal llm agents: Insights and survey about the capability, efficiency and security, 2024b. URL https://arxiv.org/abs/2401.05459.

Zeyi Liao, Lingbo Mo, Chejian Xu, Mintong Kang, Jiawei Zhang, Chaowei Xiao, Yuan Tian, Bo Li, and Huan Sun. Eia: Environmental injection attack on generalist web agents for privacy leakage. In *ICLR*, 2025.

Bill Yuchen Lin, Yicheng Fu, Karina Yang, Faeze Brahman, Shiyu Huang, Chandra Bhagavatula, Prithviraj Ammanabrolu, Yejin Choi, and Xiang Ren. Swiftsage: A generative agent with fast and slow thinking for complex interactive tasks. *Advances in Neural Information Processing Systems*, 36, 2024a.

Leilei Lin, Yumeng Jin, Yingming Zhou, Wenlong Chen, and Chen Qian. Mao: A framework for process model generation with multi-agent orchestration. *arXiv preprint arXiv:2408.01916*, 2024b.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024a.

An Liu et al. 6g-intense: Intent-driven native artificial intelligence architecture supporting network-compute abstraction and sensing at the deep edge. *IEEE Journal on Selected Areas in Communications*, 43(3):576–590, 2025a.

Evan Zheran Liu, Kelvin Guu, Panupong Pasupat, Tianlin Shi, and Percy Liang. Reinforcement learning on web interfaces using workflow-guided exploration. In *International Conference on Learning Representations (ICLR)*, 2018. URL https://arxiv.org/abs/1802.08802.

Qin Liu, Fei Wang, Chaowei Xiao, and Muhao Chen. Sudolm: Learning access control of parametric knowledge with authorization alignment. In *ACL*, 2025b.

Tie-Yan Liu et al. Learning to rank for information retrieval. *Foundations and Trends® in Information Retrieval*, 3(3):225–331, 2009.

Xiaogeng Liu, Nan Xu, Muhao Chen, and Chaowei Xiao. Autodan: Generating stealthy jailbreak prompts on aligned large language models. In *The Twelfth International Conference on Learning Representations*, 2024b.

Xiaogeng Liu, Peiran Li, G Edward Suh, Yevgeniy Vorobeychik, Zhuoqing Mao, Somesh Jha, Patrick McDaniel, Huan Sun, Bo Li, and Chaowei Xiao. Autodan-turbo: A lifelong agent for strategy self-exploration to jailbreak llms. In *The Thirteenth International Conference on Learning Representations*, 2025c.

Yue Liu, Hongcheng Gao, Shengfang Zhai, Jun Xia, Tianyi Wu, Zhiwei Xue, Yulin Chen, Kenji Kawaguchi, Jiaheng Zhang, and Bryan Hooi. Guardreasoner: Towards reasoning-based llm safeguards. *arXiv preprint arXiv:2501.18492*, 2025d.

Yue Liu, Shengfang Zhai, Mingzhe Du, Yulin Chen, Tri Cao, Hongcheng Gao, Cheng Wang, Xinfeng Li, Kun Wang, Junfeng Fang, et al. Guardreasoner-vl: Safeguarding vlms via reinforced reasoning. *arXiv preprint arXiv:2505.11049*, 2025e.

Pan Lu, Bowen Chen, Sheng Liu, Rahul Thapa, Joseph Boen, and James Zou. Octotools: An agentic framework with extensible tools for complex reasoning. *arXiv preprint arXiv:2502.11271*, 2025.

Weidi Luo, Shenghong Dai, Xiaogeng Liu, Suman Banerjee, Huan Sun, Muhao Chen, and Chaowei Xiao. Agrail: A lifelong agent guardrail with effective and adaptive safety detection. In *ACL*, 2025.

Xing Han Lù, Gaurav Kamath, Marius Mosbach, and Siva Reddy. Build the web for agents, not agents for the web, 2025. URL https://arxiv.org/abs/2506.10953.

Nadeem Mahmood, Chen Li, and Jeffrey H. Reed. Revolutionizing qoe-driven network management with digital agents in 6g. *IEEE Communications Magazine*, 62(12):42–49, 2024.

Todor Markov, Chong Zhang, Sandhini Agarwal, Florentine Eloundou Nekoul, Theodore Lee, Steven Adler, Angela Jiang, and Lilian Weng. A holistic approach to undesired content detection in the real world. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 15009–15018, 2023.

Tula Masterman, Sandi Besen, Mason Sawtell, and Alex Chao. The landscape of emerging ai agent architectures for reasoning, planning, and tool calling: A survey, 2024. URL https://arxiv.org/abs/2404.11584.

Microsoft Corporate Blogs. Introducing nlweb: Bringing conversational interfaces directly to the web. https://news.microsoft.com/source/features/company-news/introducing-nlweb-bringing-conversational-interfaces-directly-to-the-web/, May 2025. Official announcement of NLWeb project.

Yingqian Min, Zhipeng Chen, Jinhao Jiang, Jie Chen, Jia Deng, Yiwen Hu, Yiru Tang, Jiapeng Wang, Xiaoxue Cheng, Huatong Song, Wayne Xin Zhao, Zheng Liu, Zhongyuan Wang, and Ji-Rong Wen. Imitate, explore, and self-improve: A reproduction report on slow-thinking reasoning systems. *CoRR*, abs/2412.09413, 2024. doi:10.48550/ARXIV.2412.09413. URL https://doi.org/10.48550/arXiv.2412.09413.

Monica. Manus: Autonomous ai agent, 2024. URL https://manus.org/. Autonomous AI agent capable of independent task execution across multiple domains.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. Webgpt: Browser-assisted question-answering with human feedback, 2022. URL https://arxiv.org/abs/2112.09332.

Vineeth Sai Narajala and Idan Habler. Enterprise-grade security for the model context protocol (mcp): Frameworks and mitigation strategies. *arXiv preprint arXiv:2504.08623*, 2025.

Vineeth Sai Narajala and Om Narayan. Securing agentic ai: A comprehensive threat model and mitigation framework for generative ai agents. *arXiv preprint arXiv:2504.19956*, 2025.

Sid Nayak, Adelmo Morrison Orozco, Marina Have, Jackson Zhang, Vittal Thirumalai, Darren Chen, Aditya Kapoor, Eric Robinson, Karthik Gopalakrishnan, James Harrison, et al. Long-horizon planning for multi-agent robots in partially observable environments. *Advances in Neural Information Processing Systems*, 37:67929–67967, 2024.

Phillip Nelson. Advertising as information. *Journal of political economy*, 82(4):729–754, 1974.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. Adversarial nli: A new benchmark for natural language understanding. *arXiv preprint arXiv:1910.14599*, 2019.

Yuzhou Nie, Zhun Wang, Ye Yu, Xian Wu, Xuandong Zhao, Wenbo Guo, and Dawn Song. Privagent: Agentic-based red-teaming for llm privacy leakage. *arXiv preprint arXiv:2412.05734*, 2024.

NVIDIA, :, Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi "Jim" Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, Joel Jang, Zhenyu Jiang, Jan Kautz, Kaushil Kundalia, Lawrence Lao, Zhiqi Li, Zongyu Lin, Kevin Lin, Guilin Liu, Edith Llontop, Loic Magne, Ajay Mandlekar, Avnish Narayan, Soroush Nasiriany, Scott Reed, You Liang Tan, Guanzhi Wang, Zu Wang, Jing Wang, Qi Wang, Jiannan Xiang, Yuqi Xie, Yinzhen Xu, Zhenjia Xu, Seonghyeon Ye, Zhiding Yu, Ao Zhang, Hao Zhang, Yizhou Zhao, Ruijie Zheng, and Yuke Zhu. Gr00t n1: An open foundation model for generalist humanoid robots, 2025. URL https://arxiv.org/abs/2503.14734.

OpenAI. Hello-gpt-4o, 2024a. URL https://openai.com/index/hello-gpt-4o/.

OpenAI. Gpt4o-system-card, 2024b. URL https://openai.com/index/gpt-4o-system-card/.

OpenAI. Introducing chatgpt agent: bridging research and action. https://openai.com/index/introducing-chatgpt-agent/, July 2025. Accessed: 2025-07-25.

OpenAI. Chatgpt agent. https://help.openai.com/en/articles/11752874-chatgpt-agent, 2025. Accessed: 2025-07-20.

Opera. Meet opera neon, the new ai agentic browser. Opera News Blog, May 2025. URL https://blogs.opera.com/news/2025/05/opera-neon-first-ai-agentic-browser/.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

OWASP GenAI Security Project. Agentic AI Threats and Mitigations. https://genai.owasp.org/resource/agentic-ai-threats-and-mitigations/, April 2025. Accessed: 2025-07-03.

Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, et al. Open X-Embodiment: Robotic learning datasets and RT-X models. https://arxiv.org/abs/2310.08864, 2023.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford infolab, 1999.

Palo Alto Networks Unit 42. AI Agents Are Here. So Are the Threats. https://unit42.paloaltonetworks.com/agentic-ai-threats/, May 2025. Accessed: 2025-07-03.

Ashwin Paranjape, Weijia Yang, Joon Lee, et al. Art: Self-refining tool-augmented reasoning with retrieval. *Advances in Neural Information Processing Systems*, 2023.

Ethan Perez, Saffron Huang, Francis Song, Trevor Cai, Roman Ring, John Aslanides, Amelia Glaese, Nat McAleese, and Geoffrey Irving. Red teaming language models with language models. *arXiv preprint arXiv:2202.03286*, 2022.

Tatiana Petrova, Boris Bliznioukov, Aleksandr Puzikov, and Radu State. From semantic web and mas to agentic ai: A unified narrative of the web of agents, 2025. URL https://arxiv.org/abs/2507.10644.

Opera Press. Opera announces opera neon, the first ai agentic browser. Opera Press Release, May 2025. URL https://press.opera.com/2025/05/28/opera-neon-the-first-ai-agentic-browser/. Oslo, Norway.

Aman Priyanshu, Yash Maurya, and Zuofei Hong. Ai governance and accountability: An analysis of anthropic's claude. *arXiv preprint arXiv:2407.01557*, 2024.

Protocol AI. Protocol ai: No-cbuildode ai dapps & the best crypto presale on evm. https://protocolai.finance/, 2025. Accessed July 2025.

Pranav Putta, Edmund Mills, Naman Garg, Sumeet Motwani, Chelsea Finn, Divyansh Garg, and Rafael Rafailov. Agent q: Advanced reasoning and learning for autonomous ai agents, 2024. URL https://arxiv.org/abs/2408.07199.

Chen Qian, Wei Liu, Hongzhang Liu, Nuo Chen, Yufan Dang, Jiahao Li, Cheng Yang, Weize Chen, Yusheng Su, Xin Cong, et al. Chatdev: Communicative agents for software development. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15174–15186, 2024.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, et al. Toolllm: Facilitating large language models to master 16000+ real-world apis. *arXiv preprint arXiv:2307.16789*, 2023.

Jiahao Qiu, Xuan Qi, Tongcheng Zhang, Xinzhe Juan, Jiacheng Guo, Yifu Lu, Yimin Wang, Zixin Yao, Qihan Ren, Xun Jiang, et al. Alita: Generalist agent enabling scalable agentic reasoning with minimal predefinition and maximal self-evolution. *arXiv preprint arXiv:2505.20286*, 2025.

Bhaktipriya Radharapu, Kevin Robinson, Lora Aroyo, and Preethi Lahoti. Aart: Ai-assisted red-teaming with diverse data generation for new llm-powered applications. *arXiv preprint arXiv:2311.08592*, 2023.

Ramesh Raskar, Pradyumna Chari, Jared James Grogan, Mahesh Lambe, Robert Lincourt, Raghu Bala, Aditi Joshi, Abhishek Singh, Ayush Chopra, Rajesh Ranjan, Shailja Gupta, Dimitris Stripelis, Maria Gorskikh, and Sichao Wang. Upgrade or switch: Do we need a next-gen trusted architecture for the internet of ai agents?, 2025. URL https://arxiv.org/abs/2506.12003.

Traian Rebedea, Razvan Dinu, Makesh Narsimhan Sreedhar, Christopher Parisien, and Jonathan Cohen. Nemo guardrails: A toolkit for controllable and safe llm applications with programmable rails. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 431–445, 2023.

Shuo Ren, Pu Jian, Zhenjiang Ren, Chunlin Leng, Can Xie, and Jiajun Zhang. Towards scientific intelligence: A survey of llm-based scientific agents, 2025. URL https://arxiv.org/abs/2503.24047.

Steffen Rendle. Factorization machines. In *2010 IEEE International conference on data mining*, pages 995–1000. IEEE, 2010.

Paul Resnick and Hal R. Varian. Recommender systems. *Commun. ACM*, 40(3):56–58, March 1997. ISSN 0001-0782. doi:10.1145/245108.245121. URL https://doi.org/10.1145/245108.245121.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. Beyond accuracy: Behavioral testing of nlp models with checklist. *arXiv preprint arXiv:2005.04118*, 2020.

Stephen Robertson, Hugo Zaragoza, et al. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, pages 333–389, 2009.

Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. *Okapi at TREC-3*. British Library Research and Development Department, 1995.

Scott Rome, Tianwen Chen, Raphael Tang, Luwei Zhou, and Ferhan Ture. "ask me anything": How comcast uses llms to assist agents in real time. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, page 2827–2831, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400704314. doi:10.1145/3626772.3661345. URL https://doi.org/10.1145/3626772.3661345.

David M. Rothschild, Markus Mobius, Jake M. Hofman, Eleanor W. Dillon, Daniel G. Goldstein, Nicole Immorlica, Sonia Jaffe, Brendan Lucier, Aleksandrs Slivkins, and Matthew Vogel. The agentic economy, 2025. URL https://arxiv.org/abs/2505.15799.

Paul Röttger, Bertram Vidgen, Dong Nguyen, Zeerak Waseem, Helen Margetts, and Janet B Pierrehumbert. Hatecheck: Functional tests for hate speech detection models. *arXiv preprint arXiv:2012.15606*, 2020.

Ranjan Sapkota, Konstantinos I Roumeliotis, and Manoj Karkee. Ai agents vs. agentic ai: A conceptual taxonomy, applications and challenge. *arXiv preprint arXiv:2505.10468*, 2025.

Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, pages 285–295, 2001.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36:68539–68551, 2023.

Samuel Schmidgall, Yusheng Su, Ze Wang, Ximeng Sun, Jialian Wu, Xiaodong Yu, Jiang Liu, Michael Moor, Zicheng Liu, and Emad Barsoum. Agent laboratory: Using llm agents as research assistants, 2025. URL https://arxiv.org/abs/2501.04227.

Suvash Sedhain, Aditya Krishna Menon, Scott Sanner, and Lexing Xie. Autorec: Autoencoders meet collaborative filtering. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15 Companion, page 111–112, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450334730. doi:10.1145/2740908.2742726. URL https://doi.org/10.1145/2740908.2742726.

Bilgehan Sel, Priya Shanmugasundaram, Mohammad Kachuee, Kun Zhou, Ruoxi Jia, and Ming Jin. Skin-in-the-game: Decision making via multi-stakeholder alignment in llms. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13921–13959, 2024.

Guy Shani, David Heckerman, and Ronen I Brafman. An mdp-based recommender system. *Journal of machine Learning research*, 6(Sep):1265–1295, 2005.

Tianneng Shi, Jingxuan He, Zhun Wang, Linyu Wu, Hongwei Li, Wenbo Guo, and Dawn Song. Progent: Programmable privilege control for llm agents. *arXiv preprint arXiv:2504.11703*, 2025.

Zhouxing Shi, Yihan Wang, Fan Yin, Xiangning Chen, Kai-Wei Chang, and Cho-Jui Hsieh. Red teaming language model detectors with language models. *Transactions of the Association for Computational Linguistics*, 12:174–189, 2024.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. Alfworld: Aligning text and embodied environments for interactive learning. *arXiv preprint arXiv:2010.03768*, 2020.

SmythOS. Multi-agent systems in supply chain: Enhancing efficiency and responsiveness, November 2024. URL https://smythos.com/developers/agent-development/multi-agent-systems-in-supply-chain/.

Karen Spärck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21, 1972.

Karen Sparck Jones. *A statistical interpretation of term specificity and its application in retrieval*, page 132–142. Taylor Graham Publishing, GBR, 1988. ISBN 0947568212.

Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer, 2019. URL https://arxiv.org/abs/1904.06690.

Haotian Sun, Yuchen Zhuang, Lingkai Kong, Bo Dai, and Chao Zhang. Adaplanner: Adaptive planning from feedback with language models. *Advances in neural information processing systems*, 36:58202–58245, 2023.

Lisa J. Y. Tan and Ken Huang. *The AI Agent Economy*, pages 99–134. Springer Nature Switzerland, Cham, 2025. ISBN 978-3-031-90026-6. doi:10.1007/978-3-031-90026-6_4. URL https://doi.org/10.1007/978-3-031-90026-6_4.

Jiabin Tang, Tianyu Fan, and Chao Huang. Autoagent: A fully-automated and zero-code framework for llm agents. *arXiv preprint arXiv:2502.05957*, 2025.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

Paul Thurrott. It's a new dia: The browser company will launch new ai browser in early 2025. Thurrott.com, December 2024. URL https://www.thurrott.com/cloud/web-browsers/313930/its-new-dia-the-browser-company-will-launch-new-ai-browser-in-early-2025.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023a.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023b.

Khanh-Tung Tran, Dung Dao, Minh-Duong Nguyen, Quoc-Viet Pham, Barry O'Sullivan, and Hoang D. Nguyen. Multi-agent collaboration mechanisms: A survey of llms, 2025. URL https://arxiv.org/abs/2501.06322.

Vaibhav Tupe and Shrinath Thube. Ai agentic workflows and enterprise apis: Adapting api architectures for the age of ai agents, 2025. URL https://arxiv.org/abs/2502.17443.

Ada Defne Tur, Nicholas Meade, Xing Han Lù, Alejandra Zambrano, Arkil Patel, Esin Durmus, Spandana Gella, Karolina Stańczak, and Siva Reddy. Safearena: Evaluating the safety of autonomous web agents. *arXiv preprint arXiv:2503.04957*, 2025.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017. URL https://arxiv.org/abs/1706.03762.

Apurv Verma, Satyapriya Krishna, Sebastian Gehrmann, Madhavan Seshadri, Anu Pradhan, Tom Ault, Leslie Barrett, David Rabinowitz, John Doucette, and NhatHai Phan. Operationalizing a threat model for red-teaming large language models (llms). *arXiv preprint arXiv:2407.14937*, 2024.

Eric Wallace, Adina Williams, Robin Jia, and Douwe Kiela. Analyzing dynamic adversarial training data in the limit. *arXiv preprint arXiv:2110.08514*, 2021.

Jun Wang, Arjen P. de Vries, and Marcel J. T. Reinders. Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, page 501–508, New York, NY, USA, 2006. Association for Computing Machinery. ISBN 1595933697. doi:10.1145/1148170.1148257. URL https://doi.org/10.1145/1148170.1148257.

Junyang Wang, Haiyang Xu, Jiabo Ye, Ming Yan, Weizhou Shen, Ji Zhang, Fei Huang, and Jitao Sang. Mobile-agent: Autonomous multi-modal mobile device agent with visual perception, 2024a. URL https://arxiv.org/abs/2401.16158.

Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. Neural graph collaborative filtering. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '19, page 165–174. ACM, July 2019. doi:10.1145/3331184.3331267. URL http://dx.doi.org/10.1145/3331184.3331267.

Yong Wang, Xiaoli Zhang, and Sheng Li. Deep reinforcement learning based resource allocation for network slicing with massive mimo. *IEEE Transactions on Wireless Communications*, 23(4): 2125–2138, 2024b.

Yuntao Wang, Shaolong Guo, Yanghe Pan, Zhou Su, Fahao Chen, Tom H Luan, Peng Li, Jiawen Kang, and Dusit Niyato. Internet of agents: Fundamentals, applications, and challenges. *arXiv preprint arXiv:2505.07176*, 2025a.

Zhun Wang, Vincent Siu, Zhe Ye, Tianneng Shi, Yuzhou Nie, Xuandong Zhao, Chenguang Wang, Wenbo Guo, and Dawn Song. Agentvigil: Generic black-box red-teaming for indirect prompt injection against llm agents. *arXiv preprint arXiv:2505.05849*, 2025b.

Zhun Wang, Vincent Siu, Zhe Ye, Tianneng Shi, Yuzhou Nie, Xuandong Zhao, Chenguang Wang, Wenbo Guo, and Dawn Song. Agentxploit: End-to-end redteaming of black-box ai agents. *arXiv e-prints*, pages arXiv–2505, 2025c.

Zihao Wang, Shaofei Cai, Guanzhou Chen, Anji Liu, Xiaojian Ma, Yitao Liang, and Team Craft-Jarvis. Describe, explain, plan and select: interactive planning with large language models enables open-world multi-task agents. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, pages 34153–34189, 2023.

Johannes Welbl, Amelia Glaese, Jonathan Uesato, Sumanth Dathathri, John Mellor, Lisa Anne Hendricks, Kirsty Anderson, Pushmeet Kohli, Ben Coppin, and Po-Sen Huang. Challenges in detoxifying language models. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2447–2469, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi:10.18653/v1/2021.findings-emnlp.210. URL https://aclanthology.org/2021.findings-emnlp.210/.

Xiaofei Wen, Wenxuan Zhou, Wenjie Jacky Mo, and Muhao Chen. Thinkguard: Deliberative slow thinking leads to cautious guardrails. In *ACL*, 2025.

Kyle Wiggers. Perplexity teases a web browser called comet. *TechCrunch*, February 2025. URL https://techcrunch.com/2025/02/24/perplexity-teases-a-web-browser-called-comet/. Announces Comet, agent-focused Chromium browser.

Biao Wu, Yanda Li, Yunchao Wei, Meng Fang, and Ling Chen. Foundations and recent trends in multimodal mobile agents: A survey, 2025. URL https://arxiv.org/abs/2411.02006.

Chen Henry Wu, Rishi Shah, Jing Yu Koh, Ruslan Salakhutdinov, Daniel Fried, and Aditi Raghunathan. Dissecting adversarial robustness of multimodal lm agents. *arXiv preprint arXiv:2406.12814*, 2024.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, et al. Autogen: Enabling next-gen llm applications via multi-agent conversation. *arXiv preprint arXiv:2308.08155*, 2023.

Chunqiu Steven Xia, Yinlin Deng, Soren Dunn, and Lingming Zhang. Agentless: Demystifying llm-based software engineering agents, 2024. URL https://arxiv.org/abs/2407.01489.

Zhen Xiang, Linzhi Zheng, Yanjie Li, Junyuan Hong, Qinbin Li, Han Xie, Jiawei Zhang, Zidi Xiong, Chulin Xie, Carl Yang, et al. Guardagent: Safeguard llm agents by a guard agent via knowledge-enabled reasoning. *arXiv preprint arXiv:2406.09187*, 2024.

Zhen Xiang, Linzhi Zheng, Yanjie Li, Junyuan Hong, Qinbin Li, Han Xie, Jiawei Zhang, Zidi Xiong, Chulin Xie, Carl Yang, et al. Guardagent: Safeguard llm agents by a guard agent via knowledge-enabled reasoning. In *ICML*, 2025.

Jingxu Xie, Dylan Xu, Xuandong Zhao, and Dawn Song. Agentsynth: Scalable task generation for generalist computer-use agents, 2025. URL https://arxiv.org/abs/2506.14205.

Chejian Xu, Mintong Kang, Jiawei Zhang, Zeyi Liao, Lingbo Mo, Mengqi Yuan, Huan Sun, and Bo Li. Advweb: Controllable black-box attacks on vlm-powered web agents. *arXiv preprint arXiv:2410.17401*, 2024a.

Chejian Xu, Jiawei Zhang, Zhaorun Chen, Chulin Xie, Mintong Kang, Yujin Potter, Zhun Wang, Zhuowen Yuan, Alexander Xiong, Zidi Xiong, et al. Mmdt: Decoding the trustworthiness and safety of multimodal foundation models. *arXiv preprint arXiv:2503.14827*, 2025.

Jiashu Xu, Mingyu Derek Ma, Fei Wang, Chaowei Xiao, and Muhao Chen. Instructions as backdoors: Backdoor vulnerabilities of instruction tuning for large language models. *NAACL*, 2024b.

Jing Xu, Da Ju, Margaret Li, Y-Lan Boureau, Jason Weston, and Emily Dinan. Recipes for safety in open-domain chatbots. *arXiv preprint arXiv:2010.07079*, 2020.

Zhangchen Xu, Fengqing Jiang, Luyao Niu, Jinyuan Jia, Bill Yuchen Lin, and Radha Poovendran. Safedecoding: Defending against jailbreak attacks via safety-aware decoding. In *62nd Annual Meeting of the Association for Computational Linguistics, ACL 2024*, pages 5587–5605. Association for Computational Linguistics (ACL), 2024c.

Tianci Xue, Weijian Qi, Tianneng Shi, Chan Hee Song, Boyu Gou, Dawn Song, Huan Sun, and Yu Su. An illusion of progress? assessing the current state of web agents. 2025. URL https://arxiv.org/abs/2504.01382.

Hui Yang, Sifu Yue, and Yunzhong He. Auto-gpt for online decision making: Benchmarks and additional opinions. *arXiv preprint arXiv:2306.02224*, 2023a.

Ke Yang and ChengXiang Zhai. Ten principles of ai agent economics, 2025. URL https://arxiv.org/abs/2505.20273.

Ke Yang, Yao Liu, Sapana Chaudhary, Rasool Fakoor, Pratik Chaudhari, George Karypis, and Huzefa Rangwala. Agentoccam: A simple yet strong baseline for LLM-based web agents. In *The Thirteenth International Conference on Learning Representations*, 2025a.

Wenkai Yang, Xiaohan Bi, Yankai Lin, Sishuo Chen, Jie Zhou, and Xu Sun. Watch out for your agents! investigating backdoor threats to llm-based agents. *Advances in Neural Information Processing Systems*, 37:100938–100964, 2024.

Yahan Yang, Soham Dan, Shuo Li, Dan Roth, and Insup Lee. Mrguard: A multilingual reasoning guardrail for universal llm safety. *arXiv preprint arXiv:2504.15241*, 2025b.

Yang Yang, Mulei Ma, Hequan Wu, Quan Yu, Xiaohu You, Jianjun Wu, Chenghui Peng, Tak-Shing Peter Yum, A. Hamid Aghvami, Geoffrey Y. Li, Jiangzhou Wang, Guangyi Liu, Peng Gao, Xiongyan Tang, Chang Cao, John Thompson, Kat-Kit Wong, Shanzhi Chen, Zhiqin Wang, Merouane Debbah, Schahram Dustdar, Frank Eliassen, Tao Chen, Xiangyang Duan, Shaohui Sun, Xiaofeng Tao, Qinyu Zhang, Jianwei Huang, Wenjun Zhang, Jie Li, Yue Gao, Honggang Zhang, Xu Chen, Xiaohu Ge, Yong Xiao, Cheng-Xiang Wang, Zaichen Zhang, Song Ci, Guoqiang Mao, Changle Li, Ziyu Shao, Yong Zhou, Junrui Liang, Kai Li, Liantao Wu, Fanglei Sun, Kunlun Wang, Zening Liu, Kun Yang, Jun Wang, Teng Gao, and Hongfeng Shu. 6g network ai architecture for everyone-centric customized services. *IEEE Network*, 37(5):71–80, 2023b. doi:10.1109/MNET.124.2200241.

Yingxuan Yang, Huacan Chai, Shuai Shao, Yuanyi Song, Siyuan Qi, Renting Rui, and Weinan Zhang. Agentnet: Decentralized evolutionary coordination for llm-based multi-agent systems. *arXiv preprint arXiv:2504.00587*, 2025c.

Yingxuan Yang, Huacan Chai, Yuanyi Song, Siyuan Qi, Muning Wen, Ning Li, Junwei Liao, Haoyi Hu, Jianghao Lin, Gaowei Chang, et al. A survey of ai agent protocols. *arXiv preprint arXiv:2504.16736*, 2025d.

Yingxuan Yang, Qiuying Peng, Jun Wang, Ying Wen, and Weinan Zhang. Unlocking the potential of decentralized llm-based mas: Privacy preservation and monetization in collective intelligence. In *Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems*, pages 2896–2900, 2025e.

Yingxuan Yang, Ying Wen, Jun Wang, and Weinan Zhang. Agent exchange: Shaping the future of ai agent economics, 2025f. URL https://arxiv.org/abs/2507.03904.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023.

Ye Ye. Task memory engine (tme): A structured memory framework with graph-aware extensions for multi-step llm agent tasks. *arXiv preprint arXiv:2504.08525*, 2025.

Tongxin Yuan, Zhiwei He, Lingzhong Dong, Yiming Wang, Ruijie Zhao, Tian Xia, Lizhen Xu, Binglin Zhou, Fangqi Li, Zhuosheng Zhang, et al. R-judge: Benchmarking safety risk awareness for llm agents. *arXiv preprint arXiv:2401.10019*, 2024.

Xiaohan Yuan, Jinfeng Li, Dongxia Wang, Yuefeng Chen, Xiaofeng Mao, Longtao Huang, Jialuo Chen, Hui Xue, Xiaoxia Liu, Wenhai Wang, et al. S-eval: Towards automated and comprehensive safety evaluation for large language models. *Proceedings of the ACM on Software Engineering*, 2 (ISSTA):2136–2157, 2025.

Daoguang Zan, Zhirong Huang, Wei Liu, Hanwu Chen, Linhao Zhang, Shulin Xin, Lu Chen, Qi Liu, Xiaojian Zhong, Aoyan Li, et al. Multi-swe-bench: A multilingual benchmark for issue resolving. *arXiv preprint arXiv:2504.02605*, 2025.

Chi Zhang, Zhao Yang, Jiaxuan Liu, Yanda Li, Yucheng Han, Xin Chen, Zebiao Huang, Bin Fu, and Gang Yu. Appagent: Multimodal agents as smartphone users. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, CHI '25, New York, NY, USA, 2025a. Association for Computing Machinery. ISBN 9798400713941. doi:10.1145/3706598.3713600. URL https://doi.org/10.1145/3706598.3713600.

Hongming Zhang, Xiaoman Pan, Hongwei Wang, Kaixin Ma, Wenhao Yu, and Dong Yu. Cognitive kernel: An open-source agent system towards generalist autopilots. *NAACL*, 2025b.

Wei Zhang et al. A new paradigm of user-centric wireless communication driven by large language models. *IEEE Transactions on Communications*, 73(1):1–15, 2025c.

Weinan Zhang, Tianqi Chen, Jun Wang, and Yong Yu. Optimizing top-n collaborative filtering via dynamic negative item sampling. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, page 785–788, New York, NY, USA, 2013. Association for Computing Machinery. ISBN 9781450320344. doi:10.1145/2484028.2484126. URL https://doi.org/10.1145/2484028.2484126.

Weinan Zhang, Junwei Liao, Ning Li, Kounianhua Du, and Jianghao Lin. Agentic information retrieval. *arXiv preprint arXiv:2410.09713*, 2024a.

Yao Zhang, Zijian Ma, Yunpu Ma, Zhen Han, Yu Wu, and Volker Tresp. Webpilot: A versatile and autonomous multi-agent system for web task execution with strategic exploration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 23378–23386, 2025d.

Zhexin Zhang, Shiyao Cui, Yida Lu, Jingzhuo Zhou, Junxiao Yang, Hongning Wang, and Minlie Huang. Agent-safetybench: Evaluating the safety of llm agents. *arXiv preprint arXiv:2412.14470*, 2024b.

Zihan Zhang, Meng Fang, and Ling Chen. RetrievalQA: Assessing adaptive retrieval-augmented generation for short-form open-domain question answering. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 6963–6975, Bangkok, Thailand, August 2024c. Association for Computational Linguistics.

Xiaoxue Zhao, Weinan Zhang, and Jun Wang. Interactive collaborative filtering. In *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management*, CIKM '13, page 1411–1420, New York, NY, USA, 2013. Association for Computing Machinery. ISBN 9781450322638. doi:10.1145/2505515.2505690. URL https://doi.org/10.1145/2505515.2505690.

Andy Zhou, Kai Yan, Michal Shlapentokh-Rothman, Haohan Wang, and Yu-Xiong Wang. Language agent tree search unifies reasoning acting and planning in language models. *arXiv preprint arXiv:2310.04406*, 2023a.

Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, et al. Webarena: A realistic web environment for building autonomous agents. *arXiv preprint arXiv:2307.13854*, 2023b.

Wenxuan Zhou, Sheng Zhang, Yu Gu, Muhao Chen, and Hoifung Poon. Universalner: Targeted distillation from large language models for open named entity recognition. In *ICLR*, 2024.

Daniel Ziegler, Seraphina Nix, Lawrence Chan, Tim Bauman, Peter Schmidt-Nielsen, Tao Lin, Adam Scherlis, Noa Nabeshima, Benjamin Weinstein-Raun, Daniel de Haas, et al. Adversarial training for high-stakes reliability. *Advances in neural information processing systems*, 35:9274–9286, 2022.