

This is the first article in our four-part series on the Future of Application Design. Each article explores a crucial aspect of how AI agents are transforming application development:

1. The History of Application Design
- . We trace the historical progression of application design through major technological breakthroughs.
  1. State of Agentic Applications
- . We explore the existing ecosystem of Web2 and Web3 agent-based applications.
  1. The Agentic Application Tech Stack
- . We examine the technical architecture and components powering agent-based applications.
  1. Design Challenges in Agentic Applications
- . We investigate how blockchain and trusted execution environments (TEEs) address key challenges in agent-based systems.

## The Evolution of Application Design

Application design has continuously evolved through technological advances, changing user needs, and emerging design philosophies. Each era has introduced new paradigms that reshape how we interact with software, from command-line interfaces to AI-powered agents.

### Early Days: Command-Line Interfaces (CLIs) (1960s-1970s)

[

image (5)

1300×1026 83.1 KB

](<https://collective.flashbots.net/uploads/default/original/2X/b/b1a4832f3285950830f25bfc4a86b5aef821374d.jpeg>)

Figure 1. VT100 terminal.

In computing's early days, application design prioritized raw functionality over user experience. Applications were purely text-based, operating through command-line interfaces (CLIs). Users interacted with computers by typing specific commands, with the design heavily constrained by the limited hardware capabilities and engineering possibilities of the era.

### The Rise of Graphical User Interfaces (GUIs) (1970s-1980s)

[

image (6)

1920×387 28.1 KB

](<https://collective.flashbots.net/uploads/default/original/2X/2/24f51edb8a549e5c7e02e7f99154bbd237ed1272.png>)

Figure 2. History of GUIs.

A pivotal moment in human-computer interaction arrived with Xerox PARC's development of the graphical user interface (GUI). The Xerox Alto, introduced in 1973, pioneered the fundamental concepts we still use today: windows, icons, menus, and pointer-based interaction.

Apple's release of the Macintosh in 1984 brought GUI to the mainstream. Their Human Interface Guidelines established design principles that would influence decades of software development. Microsoft's release of Windows in 1985 further cemented GUI-based design as the standard, democratizing visual computing for millions of users.

### The Web Era (1990s)

[

image (7)

271×186 6.3 KB

](<https://collective.flashbots.net/uploads/default/original/2X/b/b202227fb1b1f5530fd8d54c2c577e016fa51cc2.png>)

Figure 3. Basic Webpage Using HTML and CSS.

The rise of the Internet transformed application design fundamentally. Early web applications began as static HTML pages, essentially digital versions of print documents. However, the introduction of CSS in 1996 marked a crucial turning point, separating content structure (HTML) from visual presentation (CSS).

This separation enabled the creation of comprehensive design systems and style guides, allowing developers to maintain consistent branding and user experiences across websites. Web applications evolved from simple document viewers to interactive platforms, setting the stage for the modern web.

## Mobile Revolution (2000s)

[

image (8)

1000×600 189 KB

](<https://collective.flashbots.net/uploads/default/original/2X/b/bbf402c46ff3e24b7a6e00ec9f8b9b7b80f28333.png>)

Figure 4. Mobil App Development Process.

The launch of the iPhone in 2007 ushered in the mobile computing era, introducing touch-based interfaces and creating new paradigms for user interaction. Mobile-first design principles emerged, emphasizing responsive layouts and touch-optimized interfaces.

App stores revolutionized software distribution, with Apple's App Store and Google Play lowering the distribution barrier allowing developers to reach a mass consumer market. This era saw the development of sophisticated layout systems and design patterns specifically tailored for varying screen sizes and touch interaction.

## Cloud and SaaS Transformation (2000s-2010s)

[

image (9)

1600×1000 162 KB

](<https://collective.flashbots.net/uploads/default/original/2X/d/d83294095dbef923888daa1f5e1b259f8ad72bc7.jpeg>)

Figure 5. A Brief History of Application Development.

Salesforce pioneered the Software-as-a-Service (SaaS) model in 1999, demonstrating that complex applications could be delivered entirely through web browsers. This shift fundamentally changed how applications were designed, developed, and maintained.

Amazon Web Services (AWS) launched in 2006 offering cloud computing infrastructure to applications. Cloud services shifted applications further toward modularity through microservice architecture. Applications could now be built as collections of smaller, independent services enabling dynamic scaling and efficient load management.

SaaS enabled applications to leverage cloud services for data and state storage, leading to continuous deployment and updates. Features could be rolled out gradually, improving user experience. Dependencies on external services required application designs to account for potential failures, leading to the adoption of circuit breakers and retry mechanisms.

## Web2 (2000s and 2020s)

[

image (10)

1200×420 17.8 KB

](<https://collective.flashbots.net/uploads/default/original/2X/3/333b9af77148a32bafff5caca97c784805c02426.png>)

Figure 6. Web 2.0 Applications.

Web2 technological innovations reshaped the landscape of web applications. The introduction of Asynchronous Javascript and XML (AJAX) in 2005 allowed for dynamic content updates instead of full-page reloads.

The rise of social media platforms allowed users to become content creators. It required applications to create custom interfaces, social features that connected a user profile to their social graph, and real-time features like push notifications

and WebSocket communications.

Web2 also laid the groundwork for current web applications with composable event-driven architecture, state management patterns, OAuth authentication, and rich media handling.

## Web3 and Blockchain (2020s)

[

image (11)

800×682 132 KB

](https://collective.flashbots.net/uploads/default/original/2X/8/8802d71beee8f1645402040af06a09fdd1976c6.png)

Figure 7. Decentralized Applications Architecture.

Web3 and blockchain transformed application design by decentralizing user control and incorporating token incentives. These applications give users greater control over their data and digital identities.

Token incentives accelerate user adoption and help developers validate product-market fit. The open-source nature of many Web3 projects enables community contributions, fostering innovation and user-centric design.

## AI and Agents (2020s - current)

[

image (12)

1145×720 341 KB

](https://collective.flashbots.net/uploads/default/original/2X/e/e0ed134c50f15c5d3d2407255c8fbe812161a80b.png)

Figure 8. Agentic AI Workflows.

The launch of OpenAI's ChatGPT in 2022, alongside other large language models like Claude and Gemini, created a framework for applications to deliver personalized experiences, interactive support, and conversational interfaces. Agents analyze complex processes and optimize tasks, enhancing efficiency and user experience.

AI-enabled features have introduced new architectural patterns in application design. Applications now incorporate an AI service layer managing model interactions, a context manager tracking user behavior and intentions, and learning mechanisms collecting interaction data to improve model performance.

## Modern Application Design (current)

Modern application design continues evolving with new technologies while maintaining core principles:

- User-centered design emphasizes understanding needs and behaviors to create intuitive experiences
- Responsive design adapts interfaces across devices
- Artificial intelligence powers personalized experiences and predictive features
- Modular design systems ensure consistent, scalable approaches across platforms

The evolution of application design reflects our changing relationship with technology. From command-line interfaces to AI agents, each advancement has pushed applications to become more intuitive, powerful, and deeply integrated into users' daily lives. The future of application design lies in creating platforms that are increasingly intelligent and personalized while remaining accessible and user-centric at their core.

## Resources

1. Raymond, E. and Landley, R. [The Art of Unix Usability](#).
2. [History of GUIs](#)
3. [w3.org. A Brief History of CSS Until 2016](#).
4. Abhijeet Nirmal. [An Insight to Mobile Development Process](#).
5. Muji, [A Brief History of Application Development](#)

6. Kinza Yasar, [What is Web 2.0?](#)
7. Nikita Savchenko, [Decentralized Applications Architecture: Back End, Security, and Design Patterns](#).
8. Vectorize, [Designing Agentic AI Systems. Part 1: Agent Architectures](#)
9. Vectorize, [Designing Agentic AI Systems, Part 2: Modularity](#)