

Introduction to the Indivitech System

The Era of Programmers

As the IT era began in earnest, programmers emerged as a vital new profession. At times, programming was regarded as a highly desirable career; at other times, the intense workload caused it to be viewed as one of the most demanding occupations. Nevertheless, throughout the IT era, programmers have been a core driving force of technology and a key source of business competitiveness.

Global companies such as Google and Microsoft grew through the technology, ideas, and efforts of countless programmers.

The economic rewards received by programmers have varied widely. Developers working for large corporations have sometimes earned substantial compensation, while many programmers who supported open-source software contributed remarkable technologies to the world without receiving equally large financial returns. Even so, it is undeniable that the efforts of countless programmers have helped the world advance at extraordinary speed.

Then came AI. The people who created AI were themselves programmers, engineers, and researchers. Many of the algorithms and technologies used by AI were also strengthened by the contributions of developers who supported open technologies and open-source communities.

Programming, however, has always been demanding work. Writing complex code while maintaining intense concentration for long periods can require enormous mental and physical effort.

One of the long-standing goals of programmers has therefore been to automate difficult and repetitive work. Programming languages and development tools have gradually evolved toward making development easier and more productive.

In Android development, Kotlin has become widely used; Apple promotes Swift for iPhone development; and Microsoft has developed productive platforms around C# and .NET. These developments share a common goal: to make software development faster and easier.



Figure 1. Human-centered development environments evolved and helped lead to the emergence of AI.

AI Is Creating a New Development Method

Since the emergence of AI, the way programmers develop software has begun to change. In the past, programmers wrote code directly in development tools. Today, developers can describe the code they need to AI, receive AI-generated source code, place that code into a development tool, compile it into an executable file, or deploy it to a platform and run it.

We are moving from an era in which people perform all coding themselves toward an era in which AI can perform a significant portion of the coding. As a result, the act of writing code itself can become less difficult and less labor-intensive.

This change requires us to reconsider the traditional distinction between 'easy' and 'difficult' programming languages.

We created programming languages and tools that were easier for humans to use because humans were doing the coding. When AI performs much of the coding, the situation changes.

C is one of the oldest programming languages still in widespread use and is particularly well suited to system-level integration. Because C is compiled into native machine code and can interact closely with computer systems, it can provide efficient resource use, relatively low memory overhead, and high execution performance.

The difficulty is that C programming can require substantial technical knowledge. This is one reason higher-level languages and development environments were created: even when they introduce additional abstraction or resource overhead, they can make development easier and improve human productivity.

C provides fine-grained control over performance and system resources, which makes it strong in system programming and high-performance applications. However, it can demand significant knowledge and effort from human developers. This led to the growth of high-level and scripting languages designed to improve productivity. If AI performs much of the coding, however, the question of whether a language is easy or difficult for a human may become less important than whether it provides the right performance, stability, and platform compatibility for the task.

In such an environment, the future of development tools may no longer be defined primarily by making programming languages easier for humans to type.

So what kind of development tool will emerge?

We believe integrated development tools will increasingly analyze, combine, and connect multiple pieces of AI-generated code automatically. The traditional coding editor may become less central, while buttons, configuration controls, and fields for expressing development requirements become more important. This could produce a new class of development tools very different from what we have used in the past.

With the emergence of AI, software development is beginning to move from a purely human-centered process toward a new AI-assisted development model.



Figure 2. The transition toward an era in which AI performs much of the programming.

Indivitech Is a Development Tool — More Precisely, a Tool for Collaboration with AI

AI is rapidly changing the way work is performed in fields such as design, document editing, multimedia production, and software development. Programming includes not only writing code but also device setup, platform deployment, server operation, and other technical processes. These processes may also become increasingly automated. As a result, the developer's role can expand from writing code to system design, requirements definition, verification, and operational management.

One reason programming has remained partly dependent on human specialists is that software development includes device setup and platform deployment tasks that AI alone has not traditionally handled. As these barriers are progressively automated, the number of developers whose primary role is manual coding may decrease substantially.

Indivitech was created in response to this direction. It is a development tool designed around collaboration between people and AI, and its core architecture is called Cloud Blockchain. The status and scope of related patent rights should be confirmed through the relevant official patent records.

Indivitech is designed to reduce the amount of programming knowledge required for development. A developer does not necessarily have to write the code directly. Instead, the developer defines what the system should do, while AI performs much of the code generation and modification.

The Indivitech development system is connected to the Indivitech platform, where independent server processes can run for individual projects.

Many modern IT systems are based on HTTP and HTTPS. Apache is a representative web-server daemon that provides services using these protocols. Technologies such as HTML, PHP, and JSP operate on Internet-based systems using standardized communication protocols such as HTTP and HTTPS. On top of this foundation, applications can be developed for Android phones, iPhones, and Windows PCs.

Building systems on common protocols makes reuse and upgrades easier. It also allows work created by one developer to be transferred to another developer and extended over time. This has been particularly important in the era of human-written software.

Indivitech also uses standardized HTTP/HTTPS-based communication, but this is only part of the architecture. HTML is used to present screens and provide the UI/UX.

The core server-side logic is developed in C. We call this CP, or Cloud Process.

Rather than implementing every project function only as script code, project-specific functionality can be implemented in C and run as small executable server processes or daemons. As a result, every system does not have to depend on exactly the same server logic; different projects can connect to server processes designed for their individual purposes.

What does this mean? It means that systems do not have to be uniform. They can differ according to their purpose. We describe this concept as an individualized system. AI makes this approach more practical because AI can assist with creating and modifying the diverse code required by each project.

Indivitech can be viewed as one example of the new development tools emerging in the AI era. Similar tools are likely to continue appearing as more development work becomes automated. Over time, the role of developers and the subjects emphasized in computer science may also evolve toward new forms of system design, AI collaboration, verification, and operation.


indivitech
 AI × People × Technology
 A New Standard for Development

Anyone Can Develop with AI

From Ideas to Real Applications — All in One Platform

No Coding. More Possibilities!


Idea & Requirements
 Just describe what you want to build.


AI Generates Code
 AI automatically creates the optimal code.


Compile & Build
 Put the AI-generated code into the tool and build it.

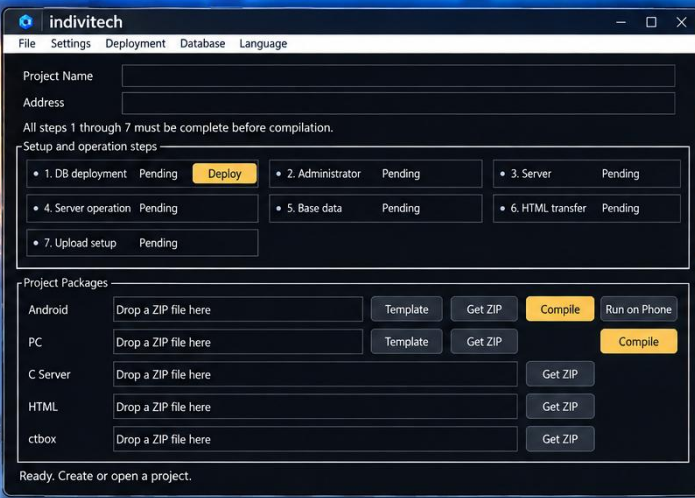

Deploy to Multiple Platforms
 Deploy to Android, iOS, PC, and servers with ease.


Operate & Scale
 Run, manage, and expand with independent servers.

The Past: Human Coding
 Complex setup, coding, debugging...
 Time-consuming and difficult.
 

Collaboration with AI
 Just tell AI what you need, and AI writes the code for you.
 

The Future: Beyond Coding
 Focus on ideas and creating greater value. AI handles the coding.
 



One Project, Multiple Platforms


Cloud Blockchain
 (Independent Server Structure)
 Connects various servers for a more diverse world

Core Technology

-  **HTTP / HTTPS**
Unified transmission protocol (partial)
-  **C Language based CP**
(Cloud Process)
-  **Individualized System**
Connects various types of servers
-  **HTML for UI/UX**
Supports flexible and powerful interfaces

Key Benefits of indivitech

-  No IT knowledge required
-  No programming languages needed
-  AI handles code writing
-  All-in-one development for Android, iOS, PC, Servers
-  Independent servers and flexible architecture
-  Easy reuse, upgrade, and expansion

A More Creative World with AI!

indivitech Technology for a Better Tomorrow
 Beyond Human Limits, Toward a Smarter World
AI Creates Possibilities for Everyone

Figure 3. Indivitech — a development tool designed for collaboration between people and AI.