



APUSIC  
固若长城  
睿比世界

## Product Whitepaper

Apusic Application Server V10

## Copyright Notice

All intellectual property rights including software copyrights involved in this document have been legally registered and duly owned by Kingdee Apusic Cloud Computing Co., Ltd. Such rights are protected by Copyright Law of the People's Republic of China, Regulations on Protection of Computer Software, Regulations on Intellectual Property Protection, relevant international copyright conventions, laws, regulations and other intellectual property legislations. Unauthorized use is strictly prohibited.

## Disclaimer

All copyright information contained herein is legally owned by Kingdee Apusic Cloud Computing Co., Ltd. and protected by applicable laws. The Company shall bear no liability for any information unrelated to Kingdee Apusic contained in this document. You may view, copy and print this document only within the scope permitted by Copyright Law of the People's Republic of China. No entity or individual may use, modify or republish any part or content of this document without written authorization from Kingdee Apusic Cloud Computing Co., Ltd. Any violation shall constitute infringement, and the Company reserves the right to pursue legal liabilities accordingly.

Updates to this document shall not be notified separately. You may consult the Company regarding issues related to this document. All rights not expressly granted by the Company are reserved.

## Trademark Notice

 is a registered trademark filed with the National Intellectual Property Administration of PRC by Kingdee Apusic Cloud Computing Co., Ltd. Shenzhen. Exclusive rights to the registered trademark are legally owned by Kingdee Apusic and protected by law. Without written consent of Kingdee Apusic, no entity or individual shall use, copy, modify, disseminate, transcribe, or bundle and sell any part of this trademark with other products in any manner or for any reason. Kingdee Apusic shall pursue legal liabilities against any infringement of its trademark rights. All other trademarks or registered trademarks mentioned in this document belong to their respective owners.

# Contents

- 1 Introduction
- 2 Product Overview
  - 2.1 The Role of Kingdee Apusic Application Server in Enterprise Applications
  - 2.2 Kingdee Apusic Application Server and Java Cloud Native
  - 2.3 Kingdee Apusic Application Server and AI
- 3 Product Architecture
  - 3.1 Product Architecture
  - 3.2 High-Availability Cluster Architecture
  - 3.3 Technical Specifications and Standards Conformance
  - 3.4 Runtime Environment
    - 3.4.1 Supported Software and Hardware Environments
    - 3.4.2 Deployment Environment Requirements
- 4 Features
  - 4.1 Web Container
  - 4.2 EJB Container
  - 4.3 Web Service Container
  - 4.4 Transaction Service
  - 4.5 CDI Service
  - 4.6 Message Service
  - 4.7 JavaMail Service
  - 4.8 Data Source
  - 4.9 JNDI Resources
  - 4.10 JCA Connectors
  - 4.11 Thread Pool
  - 4.12 Cluster Management
  - 4.13 Application Management
  - 4.14 Security Service
  - 4.15 JMX Service
  - 4.16 Observability
  - 4.17 Log Management
  - 4.18 Application Diagnostics
  - 4.19 Operations Management

- 4.20 CLI Command-Line Tool
- 4.21 Copilot Assistant
- 5 Product Editions
  - 5.1 Enterprise Edition
  - 5.2 Standard Edition
  - 5.3 Intelligent Security Edition
- 6 Product Benefits
  - 6.1 Fully Independent Intellectual Property Rights
  - 6.2 Outstanding Product Performance
  - 6.3 Enterprise-Grade Application Security Protection
  - 6.4 Support for Mainstream Development Frameworks
  - 6.5 Application Server Migration Tool
  - 6.6 High-Quality Service Assurance
- 7 Application Scenarios
  - 7.1 Government System Applications
  - 7.2 Core Enterprise Business
  - 7.3 Financial Business Applications
  - 7.4 Java Cloud-Native Applications
- 8 Customers and Cases
- 9 Appendix: Supported Jakarta EE Specifications
- 10 Appendix: Glossary

# 1 Introduction

In today's digital era, the survival philosophy of enterprises and organizations has shifted from "big and comprehensive" to "fast and new." Application architectures have evolved continuously from traditional monolithic applications to SOA, and further to microservices and cloud-native architectures. Today, cloud native — characterized by containerization, automated continuous delivery, and elastic scalability — has become the mainstream technology paradigm, while the explosive growth of artificial intelligence is giving rise to new application forms such as agentic AI applications that use large language models as their reasoning core. All of this imposes unprecedented new requirements on application servers.

In the cloud-native era, applications need the ability to scale elastically and rapidly, to adopt stateless design, and to deliver consistent container images across environments; they also need to integrate deeply with container orchestration platforms such as Kubernetes to achieve declarative deployment and management.

Under a microservices architecture, an application is split into multiple autonomous, independent services, each with its own lifecycle. Services communicate through HTTP RESTful APIs or message queues, requiring the application server to provide a high-performance web container that handles high-concurrency requests, and comprehensive service governance capabilities that support service registration and discovery, load balancing, circuit breaking, and rate limiting.

In the era of AI-native applications, agentic applications driven by large language models are becoming a new trend. Such applications must handle massive volumes of concurrent HTTP requests and responses, support complex asynchronous task scheduling and execution, and provide reliable long-running capability. As the hosting platform for AI applications, an application server's stability and performance directly determine the quality of AI services.

The Kingdee Apusic Application Server (Apusic Application Server, hereinafter referred to as "AAS") was born precisely to meet these challenges. As a domestic application server with fully independent intellectual property rights, AAS fully embraces cloud-native technologies, supports Docker containerized deployment and Kubernetes orchestration, and supports JDK 21 virtual threads to enhance high-concurrency processing. In terms of microservices support, AAS provides mature capabilities such as a high-performance web container, distributed session management, and high-availability clustering. In terms of AI-native application support, AAS, with its outstanding concurrency performance and comprehensive resource management, provides a solid technical foundation for agentic applications.

For more than two decades, Kingdee Apusic Application Server has, through arduous refinement and bold innovation, extensively absorbed and satisfied the urgent needs of domestic software developers and key application customers. It has become the preferred domestic middleware product for a broad community of developers and architects, helping domestic e-government, e-commerce, and enterprise applications achieve continuous success.

This document is intended for technical decision-makers and enterprise architects. By reading the beginning and the end of this document, you can gain a macro-level understanding of Kingdee Apusic Application Server; to fully understand the product's capabilities, reading the entire document is recommended.

## 2 Product Overview

Kingdee Apusic Application Server is an enterprise-grade Java application server that fully implements the Jakarta EE specifications, the latest web services standards, and mainstream interoperability standards.

The Jakarta EE specifications provide a new set of standard APIs for creating distributed applications. These applications can access various services and enable dynamically designed visual web content as well as interoperability among heterogeneous systems. Kingdee Apusic Application Server fully supports the Jakarta EE specifications and has passed the relevant specification certifications; the completeness of its specification implementation reaches the advanced level of comparable international products.

Kingdee Apusic Application Server fully meets and partially exceeds the requirements of the specification APIs. It provides capabilities such as data persistence, web services, high availability, clustering and dual-machine hot standby, message transmission and routing, and cross-platform support. Together with a series of foundational middleware products — including the Kingdee Apusic Load Balancer, Kingdee Apusic Distributed Cache, and Kingdee Apusic Distributed Message Queue — Kingdee Apusic Application Server provides robust and reliable foundational service support that meets the needs of large-scale distributed applications. It is the best implementation of independent middleware conforming to international standards, and a model of "international standards, Chinese wisdom."

Kingdee Apusic Application Server allows applications to be deployed in highly available and highly scalable environments. Enterprises can configure clusters of Kingdee Apusic Application Server instances in heterogeneous environments to distribute load and to respond promptly when hardware or network failures occur. Extensible security features protect access to applications, safeguard enterprise data, and prevent malicious attacks. The management tools provided by Kingdee Apusic Application Server allow system administrators to monitor and optimize application performance.

In short, through these features and tools, Kingdee Apusic Application Server provides users with a robust runtime environment for securely deploying critical applications with massive data.

### 2.1 The Role of Kingdee Apusic Application Server in Enterprise Applications

Kingdee Apusic Application Server plays a vital role in enterprise application architecture, serving as the bridge connecting enterprise business applications with the underlying infrastructure.

In the field of enterprise application development, AAS provides development teams with development frameworks and runtime environments conforming to the Java EE and Jakarta EE standards. Business applications developed on standard APIs enjoy good portability — when migrating an application between application servers from different

vendors, no large-scale code modification is required. Capabilities provided by AAS such as declarative transaction management, security control, and dependency injection greatly simplify the development complexity of enterprise applications, enabling developers to focus on implementing business logic rather than low-level technical details.

In the field of enterprise IT operations, AAS provides centralized management and monitoring capabilities. Through a visual console, operations personnel can uniformly manage application instances deployed across multiple servers and monitor application status, performance metrics, and resource usage. High-availability features such as cluster management, load balancing, and failover reduce operational complexity and improve system reliability. The patch management and hot deployment capabilities of AAS support version upgrades and issue fixes without shutting down applications.

In the field of enterprise IT governance, AAS serves as a foundational support platform that helps enterprises achieve standardization and controllability of their technology stack. AAS owns fully independent intellectual property rights, enabling enterprises to control the pace of technology evolution, avoid dependence on a single foreign vendor, and reduce supply chain risk. AAS has passed the national standard certification for application server middleware, meeting the compliance requirements of industries with high security demands such as government and finance.

## 2.2 Kingdee Apusic Application Server and Java Cloud Native

Cloud native has become the mainstream technology paradigm for enterprise digital transformation. Kingdee Apusic Application Server fully embraces cloud-native technologies, providing solid support for enterprise cloud-native applications.

In terms of containerization, AAS provides standard Docker images, supporting the packaging of applications and their runtime environments into immutable container images that can be deployed and run consistently in any container runtime environment. Kubernetes has become the de facto standard for container orchestration; AAS supports the Kubernetes health check interface and can integrate deeply with the Horizontal Pod Autoscaler (HPA) to automatically adjust the number of instances according to business load.

In terms of microservices architecture, AAS provides comprehensive service governance capabilities. A built-in circuit-breaking and rate-limiting mechanism protects system stability and prevents the spread of cascading failures; distributed session management supports session sharing across instances, ensuring that user sessions are not lost in failure scenarios; the VIP thread pool reserves dedicated resources for critical services, ensuring that important business is not affected by other requests. Through integration with microservices frameworks such as Spring Cloud, AAS can work with configuration centers such as Nacos and Apollo Config, as well as APM tools such as SkyWalking and Pinpoint, to build a complete microservices governance system.

In terms of performance optimization, AAS supports JDK 21 virtual thread technology. Virtual threads greatly reduce the memory and scheduling overhead of threads, enabling applications to handle massive numbers of concurrent connections; in a standard test scenario, a single instance can support one million online users. Combined with

technologies such as HTTP/2 multiplexing, AAS achieves high-performance, low-latency request processing in cloud-native environments.

## 2.3 Kingdee Apusic Application Server and AI

Artificial intelligence is evolving from an auxiliary tool into a component of enterprise applications, and the way applications are built and their runtime characteristics are changing accordingly. Kingdee Apusic Application Server provides high-performance, highly reliable runtime support for AI-native applications.

In AI application architectures, agentic AI applications are becoming a new application form. Such applications use large language models as their reasoning core, with an AI Agent framework taking on orchestration responsibilities such as tool invocation, task planning, and memory management, transforming the model's reasoning capability into executable and observable business actions. Compared with traditional request-response applications, their prominent characteristics are long-running operation, high-frequency external calls, and bursty concurrency — precisely the scenarios where an application server delivers the most value. With its mature concurrency handling and resource scheduling capabilities, AAS provides a stable hosting platform for AI Agent applications.

In high-concurrency scenarios, the AAS web container can handle large volumes of concurrent AI application requests, and the HTTP/2 multiplexing feature is well suited to high-frequency communication between AI applications and external model services. AAS supports JDK 21 virtual threads; combined with thread pool management and JVM tuning capabilities, it ensures that inference calls under high concurrency obtain stable computing resources and avoids response latency caused by resource contention.

In terms of reliability, the high-availability clustering capability of AAS safeguards the continuous availability of AI services: failed instances are automatically removed and new instances automatically join, so that the failure of some nodes does not affect overall service capability; distributed session management supports state retention for AI applications; crash detection and automatic restart mechanisms enable rapid failure recovery; and the monitoring and alerting system collects AI application runtime metrics in real time and issues timely alerts on anomalies, providing operational assurance for the stable operation of AI services.

AAS also integrates with mainstream AI development frameworks. Frameworks such as Spring AI provide a unified abstraction for model invocation; as their runtime support, AAS provides enterprise-grade HTTP connection pooling, session management, and security control, enabling AI applications to directly reuse the mature engineering capabilities and existing operations systems of the Java ecosystem.

## 3 Product Architecture

Kingdee Apusic Application Server is a standardized, secure, efficient, and feature-rich enterprise application server. It fully supports the Jakarta EE technical specifications and provides a web container, EJB container, and web service container that conform to those specifications, supporting the latest technical specifications such as WebSocket, Servlet, and HTTP. As a reliable foundational platform for enterprise-grade application operation, it provides critical support for the convenient development, flexible deployment, reliable operation, efficient management and control, and rapid integration of enterprise-grade applications.

### 3.1 Product Architecture

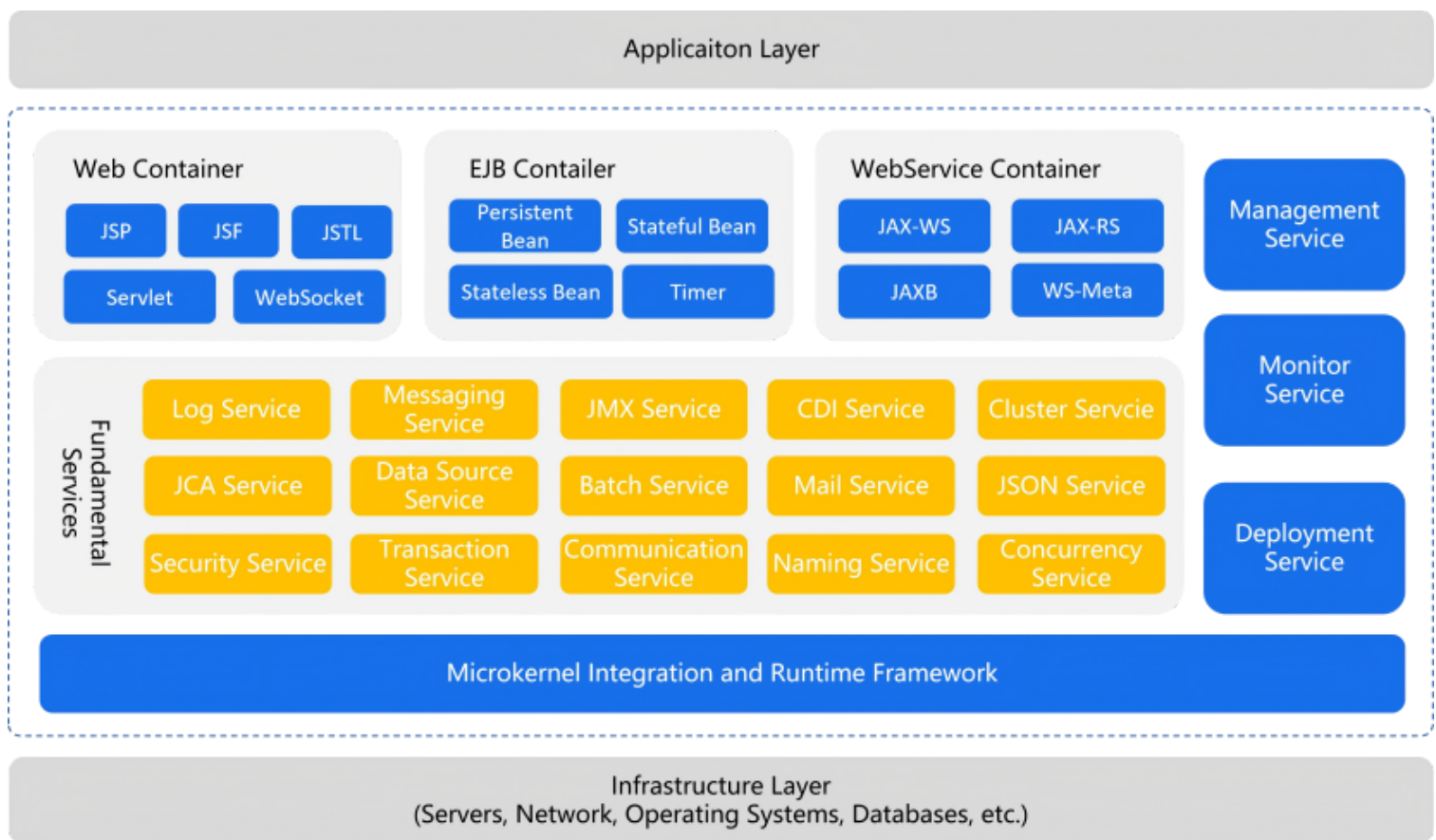


Figure 1 Product architecture of Kingdee Apusic Application Server

Kingdee Apusic Application Server adopts a microkernel architecture, in which the microkernel provides foundational capabilities such as class loading services, thread management, and memory management. Functional modules (web container, EJB container, security service, message service, data source management, and so on) run on the microkernel framework in a pluggable manner, and modules can be upgraded or replaced without restarting the server.

AAS provides a unified web management console for configuration management of application server instances and cluster components, as well as full-lifecycle management, monitoring, diagnostics, and operations of deployed business applications.

## 3.2 High-Availability Cluster Architecture

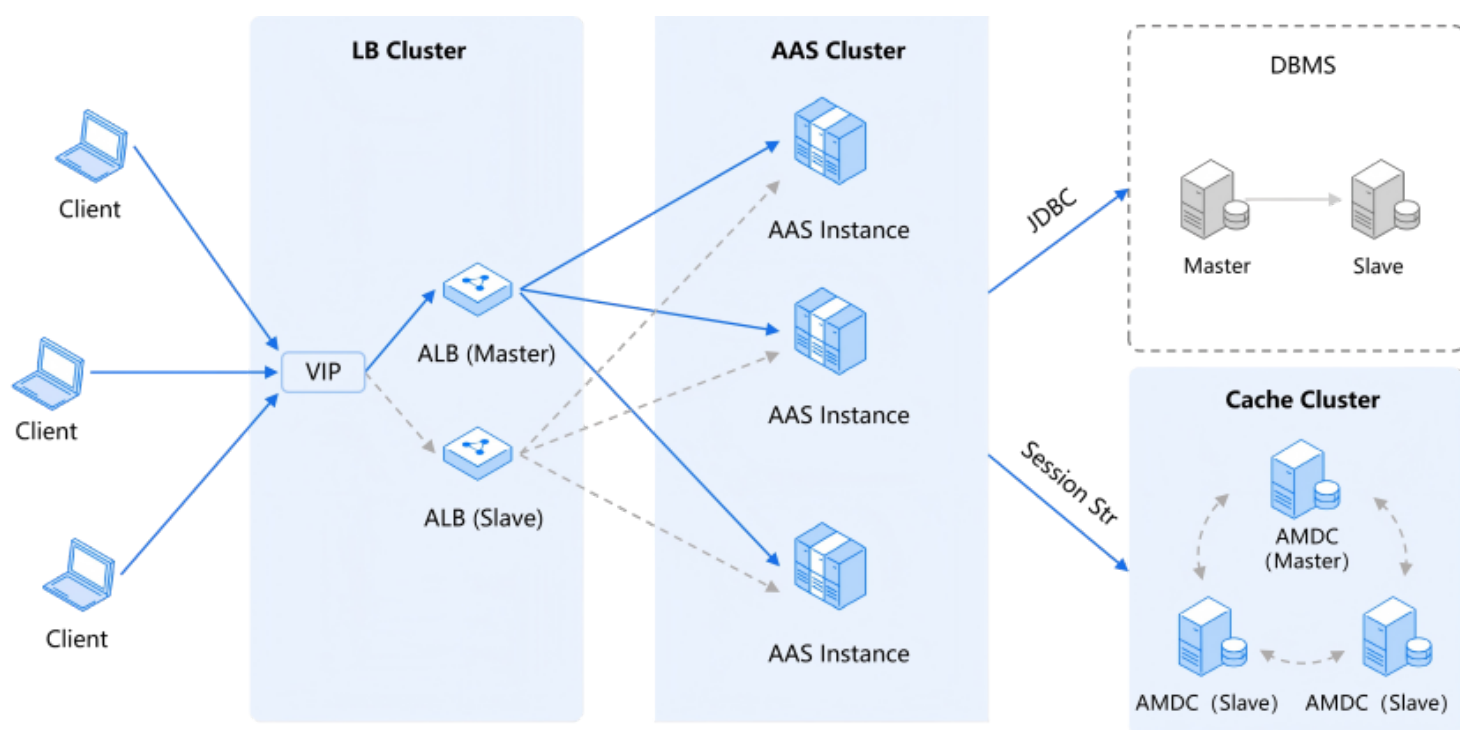


Figure 2 Cluster architecture of Kingdee Apusic Application Server

Kingdee Apusic Application Server supports a high-availability cluster architecture that ensures business continuity of business application systems. An application server cluster consists of multiple instances that share applications and configuration, and each instance node can handle requests independently.

A Kingdee Apusic Application Server cluster is typically combined with an Apusic Load Balancer cluster and a Kingdee Apusic Distributed Cache cluster to build a typical enterprise-grade application cluster architecture.

### Load Balancer Cluster

ALB (Apusic Load Balancer) supports active-standby (or dual-active) mode and, as the traffic entry point, performs load balancing and distribution of requests. ALB supports virtual IP configuration to achieve high availability of the load balancer cluster, provides SSL/TLS offloading, and supports national cryptographic algorithm certification.

### Application Server Cluster

The AAS application server cluster consists of multiple instances that share applications and configuration and independently handle requests distributed by the load balancer. It features built-in instance health detection,

supports crash alerting and automatic restart, and supports session affinity policies to ensure continuity of user access.

## Distributed Cache Cluster

AMDC (Apusic In-Memory Data Cache) supports sentinel and cluster modes, enabling session sharing and state synchronization for application servers. It also provides caching of hot business data to reduce the load on databases.

## 3.3 Technical Specifications and Standards Conformance

Kingdee Apusic Application Server is a commercial application middleware product that fully conforms to industry technical specifications and standards.

### Conformance to Java EE / Jakarta EE Technical Specifications

Jakarta EE (formerly Java EE) is a Java technical specification for enterprise-grade applications, maintained by the Eclipse Foundation. The specification defines web components (Servlet, JSP), enterprise components (EJB, JPA, JMS), web services (REST, SOAP), as well as technical standards for security and management, and is the foundation for applying Java technology in the enterprise domain.

Specification-conformant applications offer good portability. Applications developed can run on any specification-conformant application server, and migration between products from different vendors requires no large-scale code modification. The specifications also foster ecosystem development: developers can use standard APIs, and framework and tool vendors can provide products based on the specifications.

Kingdee Apusic Application Server has passed the Jakarta EE 8, Jakarta EE 9, Jakarta EE 9.1, Jakarta EE 10, and Jakarta EE 11 specification certifications, and its technical capabilities are internationally recognized.

### Conformance to the GB/T 26232-2025 Technical Standard

GB/T 26232-2025, Information Technology — Middleware — Technical Requirements for Application Server Middleware, is a national standard issued by the Standardization Administration of China. It specifies technical requirements and test methods for application server middleware across aspects such as functionality, performance, security, reliability, management, and interfaces. This standard is the foundational standard for the sale and application of application server products in the domestic market; passing its conformity test certification proves that a product's technical indicators meet national requirements.

Kingdee Apusic Application Server has passed the conformity test certification of GB/T 26232-2025, Information Technology — Middleware — Technical Requirements for Application Server Middleware.

### Conformance to the GB 18030-2022 Technical Standard

GB 18030-2022, Information Technology — Chinese Coded Character Set, is a foundational standard in the field of Chinese information technology that specifies how Chinese characters are encoded and represented in information systems. The standard is divided into implementation levels 1, 2, and 3, with level 3 being the highest; it requires support for all Chinese characters and the scripts of China's ethnic minorities, and covers extended characters such as rare surname characters and place-name characters.

Kingdee Apusic Application Server supports GB 18030-2022 implementation level 3, providing the foundational guarantee for the correct storage, processing, and display of Chinese for applications such as government systems and financial systems that process large volumes of Chinese data. Kingdee Apusic Application Server has passed the GB 18030-2022, Information Technology — Chinese Coded Character Set, implementation level 3 certification.

### 3.4 Runtime Environment

Kingdee Apusic Application Server offers extensive software and hardware environment adaptability, supporting mainstream domestic and international operating systems, chip architectures, and database environments. Its depth and breadth of adaptation to domestic environments are at a leading domestic level, providing stable and reliable runtime support for enterprise-grade applications.

#### 3.4.1 Supported Software and Hardware Environments

Table 1 List of supported software and hardware environments

| Category         | Name                                                                                                                                                                                                                               |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Operating system | Supports popular domestic and international UNIX/Linux systems, including:<br>HP-UX<br>IBM AIX<br>Red Hat Enterprise Linux<br>CentOS Stream/Rocky Linux<br>SUSE Enterprise Linux<br>Ubuntu Server<br>Debian<br>Oracle Linux<br>... |
|                  | Windows operating systems<br>Windows Server 2016/2019/2022/2025<br>...                                                                                                                                                             |
|                  | Supports mainstream domestic operating systems, including:<br>Kylin OS<br>UnionTech OS<br>openEuler                                                                                                                                |

|          |                                                                                                                                                                                                                                                  |
|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|          | Primware Linux<br>CAS Fangde<br>ZTE NewStart OS<br>Ningsi Linux<br>Deepin<br>...                                                                                                                                                                 |
| Database | Supports mainstream databases, including:<br>Dameng<br>KingbaseES<br>GBase<br>ShenTong<br>Highgo<br>Huawei GaussDB<br>ZTE GoldenDB<br>Alibaba Cloud OceanBase<br>TiDB<br>MySQL<br>PostgreSQL<br>Oracle<br>Microsoft SQL Server<br>IBM DB2<br>... |
| JDK      | Supports mainstream JDK distributions and versions 8, 11, 17, 21, and 25, including:<br>Oracle JDK<br>Open JDK<br>IBM JDK<br>Apusic JDK<br>BiSheng JDK<br>...                                                                                    |
| Chip     | Supports mainstream chip architectures, including:<br>X86/X64: Intel X86/X64, AMD X86/X64, Hygon series<br>ARM64: Kunpeng 920/930 series, Phytium series<br>LoongArch: Loongson (Loongson 3A5000/3A6000) series<br>SW64: Sunway<br>RISC-V, etc.  |
| Server   | Supports mainstream server hardware, including:<br>Inspur, Great Wall, Sugon, Huawei TaiShan, Lenovo, Tsinghua Tongfang, Powerleader, Digital China, SUMA...                                                                                     |
| Browser  | Chrome 85.0+, Firefox 80.0+<br>Supports various domestic browsers, such as Haitai Red Lotus, 360 Browser, and QiAnXin                                                                                                                            |

Trusted Browser

### 3.4.2 Deployment Environment Requirements

Single-instance deployment is suitable for development and test environments or small production systems. Kingdee Apusic Application Server consumes relatively few resources at runtime and can run in an environment with a 4-core CPU, 4 GB of memory, and 10 GB of disk space.

For production environments, the recommended configuration is advised for better performance: an 8-core CPU, 16 GB of memory, and more than 100 GB of disk space. The Java runtime environment supports JDK 8 and above, with JDK 17 and above recommended.

Table 2 AAS deployment environment resource requirements

| System component | Minimum configuration     | Recommended configuration |
|------------------|---------------------------|---------------------------|
| CPU              | 4 cores                   | 8 cores and above         |
| Memory           | 4 GB                      | 16 GB and above           |
| Disk space       | 10 GB                     | 100 GB and above          |
| Network          | Gigabit network and above | Gigabit network and above |

## 4 Features

### 4.1 Web Container

The AAS web container supports the Servlet 4.0/6.0 specifications. Servlet 4.0 introduces HTTP/2 protocol support, enabling a single TCP connection to process multiple concurrent requests in parallel, and provides Server Push, by which the server can proactively push resources to the client.

AAS supports the JSP 2.3/3.1 specifications, and the Expression Language EL 3.0/5.0 simplifies page data access. It supports WebSocket 1.1/2.1 full-duplex communication.

AAS includes a built-in security filter chain: the CORS Filter uniformly configures cross-origin access policies; the XSS Filter automatically detects and blocks cross-site scripting attacks; the CSRF Filter prevents cross-site request forgery attacks. These filters can be enabled through configuration without modifying application code.

### 4.2 EJB Container

The EJB container is responsible for lifecycle management, state management, and JTA transactions of EJB bean instances, and supports publishing EJBs as web services. It supports EJB 3.2 and dependency injection, and optimizes invocation performance using EJB object pooling and object caching technologies.

The AAS EJB container supports stateless and stateful session beans, suitable for concurrent request handling and user session maintenance scenarios, and supports message-driven beans integrated with the JMS message service.

The EJB container supports local and remote invocation, with component location achieved through JNDI lookup. It supports container-managed transactions and bean-managed transactions, and supports distributed transaction processing to ensure data consistency across databases and services.

### 4.3 Web Service Container

The AAS web service container supports the JAX-RS and JAX-WS specifications, providing capabilities for developing and deploying web services.

The JAX-RS 2.1/3.1 specifications support RESTful web service development, defining resource classes and methods through annotations and automatically mapping Java objects to JSON or XML format. The JAX-WS 2.2 specification supports SOAP-based web service development, with support for WSDL file generation and client code generation.

The AAS web service container provides WS-Security support, enabling digital signatures, encryption, and identity authentication for SOAP messages. It supports MTOM (Message Transmission Optimization Mechanism) to optimize the transmission of large attachments.

## 4.4 Transaction Service

The AAS transaction service provides JTA transaction management capabilities, ensuring data consistency across components and databases. AAS supports container-managed transactions, in which the container automatically handles transaction boundaries: developers declare transaction attributes through annotations, and the container automatically begins and commits transactions before and after method invocations. It also supports bean-managed transactions, in which developers control transaction boundaries precisely through code.

AAS supports distributed transaction processing, coordinating multiple resource managers through the two-phase commit protocol to ensure data consistency across databases and services. It supports transaction timeout configuration and transaction suspension and resumption.

## 4.5 CDI Service

AAS supports the CDI specification, providing dependency injection and lifecycle management capabilities. CDI uses type-safe dependency injection, injecting components into target objects through annotations, and supports constructor injection, field injection, and method parameter injection. It supports the `@Named` annotation to register a bean as a named bean accessible through EL expressions, so that it can be accessed in JSF and JSP pages.

CDI provides a variety of scope annotations such as `@ApplicationScoped`, `@RequestScoped`, `@SessionScoped`, and `@ConversationScoped` to manage the lifecycle of bean instances. CDI supports an event mechanism, whereby components can listen for and handle events through the `@Observes` annotation.

## 4.6 Message Service

AAS supports the JMS 2.0 specification, providing JMS connection factories and destination queue/topic configuration, and supports both point-to-point and publish/subscribe messaging models.

## 4.7 JavaMail Service

AAS provides a JavaMail service that supports sending and receiving email through the standard JavaMail API. AAS supports mail protocols such as SMTP, POP3, and IMAP, and allows configuration of parameters such as mail server address, port, and authentication method.

JavaMail sessions can be registered through JNDI, and applications use them by looking up the JNDI name; when the mail server configuration changes, no application code modification is required.

## 4.8 Data Source

The AAS JDBC connection pool manages database connections using a FIFO queue: a connection is taken from the head of the queue when acquired, and placed at the tail of the queue when returned. Both acquiring and returning a connection update the connection timestamp, so that all connections are used evenly and their activity is maintained.

AAS provides a rich set of connection pool tuning parameters: minimum connections, maximum connections, connection timeout, idle timeout, connection validation, and so on. It supports JNDI data source configuration and encrypted storage of data source passwords. It supports multiple data source configuration, allowing a single application to access multiple different databases simultaneously.

AAS provides connection leak detection, automatically reclaiming connections that have not been released for a long time. It provides slow SQL monitoring, recording SQL statements whose execution time exceeds a threshold along with their execution duration.

## 4.9 JNDI Resources

AAS provides a JNDI service for binding objects to a naming directory and obtaining objects through JNDI name lookup.

The AAS JNDI service supports the registration and lookup of enterprise-grade objects such as EJB, DataSource, JMS connection factories, and JavaMail sessions. Through JNDI, application code is decoupled from concrete resources, so that no application code modification is required when configuration changes.

## 4.10 JCA Connectors

AAS supports the JCA specification, providing enterprise information system (EIS) integration capabilities.

JCA connectors enable integration with enterprise legacy systems such as ERP and CRM, connecting to enterprise information systems through resource adapters. The JCA specification defines three container services: connection management, transaction management, and security management.

The AAS JCA implementation supports connection pool management, improving performance through EIS connection reuse. It supports local transactions and distributed transactions, guaranteeing data consistency across system operations. Security management provides credential mapping and permission control.

## 4.11 Thread Pool

AAS provides a thread pool feature that limits the number of execution threads in the system and optimizes resource utilization.

A thread pool reuses the same thread across multiple tasks, spreading the overhead of thread creation across multiple tasks. Because threads already exist when a request arrives, the latency caused by thread creation is

eliminated. By appropriately adjusting the number of threads in the pool, new requests wait for an available thread once the request count exceeds the threshold, preventing resource exhaustion.

AAS provides configurable parameters such as the core thread count, maximum thread count, and queue length. It provides a VIP thread pool that allocates dedicated thread resources to important requests, ensuring that critical business is not affected by other requests.

## **4.12 Cluster Management**

An AAS cluster consists of multiple instances that share applications and configuration and independently handle requests. AAS provides a graphical cluster interface that makes it easy for users to perform unified configuration management of clusters, cluster instances, cluster front-end load balancers, and distributed cache clusters.

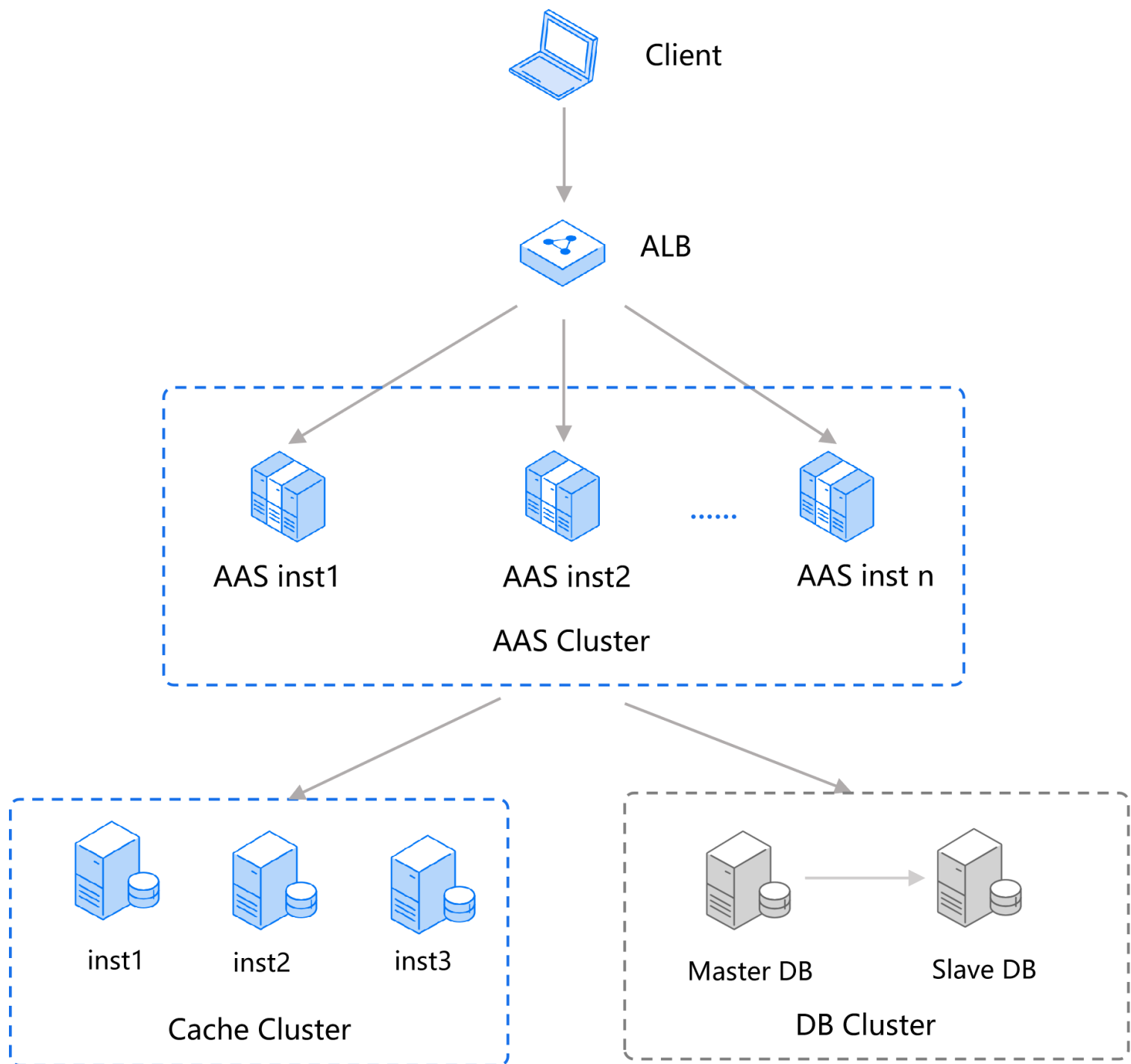


Figure 3 Cluster management interface of Kingdee Apusic Application Server

AAS supports ALB, Nginx, and Apache HTTP Server as load balancers, and supports session affinity policies.

AAS supports the use of AMDC and Redis cache clusters to implement distributed centralized storage management. AAS distributed session management stores user session data in an independent cache cluster rather than in application server memory, so that when an instance fails, user session data is not lost and other instances can continue to handle that user's requests.

## 4.13 Application Management

AAS provides visual application management, enabling full-lifecycle management of applications deployed in various formats such as WAR, JAR, and EAR packages. It supports one-click deployment and automatic extraction and installation, supports application version management with the ability to view version information of deployed applications, and supports application upgrades and rollbacks.

It supports application start/stop management, allowing start and stop operations to be performed on a single application or on multiple applications at once. It supports application health status monitoring, allowing users to view metrics such as application running status, number of sessions, and response time.

## 4.14 Security Service

The AAS security mechanism covers aspects such as network communication, identity authentication, access control, and security auditing.

National cryptographic algorithm support meets domestic compliance requirements. The SM2 algorithm is used for digital signatures and key exchange, the SM3 algorithm for message integrity verification, and the SM4 algorithm for data encryption; together they form a complete domestic cryptographic system.

The SSL/TLS protocol encrypts network communication. TLS 1.3 significantly reduces handshake latency compared with earlier versions. JKS and PEM are common certificate storage formats; one-way authentication verifies only the server's identity, while two-way authentication verifies the identities of both the client and the server.

Three-role separation management achieves the separation of and mutual constraint among permissions. The system administrator is responsible for operational duties such as application deployment, resource management, and configuration management; the security and confidentiality officer is responsible for security configurations such as user accounts, role permissions, and password policies; and the security auditor is responsible for viewing and auditing operation logs and security logs.

Password policies force users to use complex passwords and change them regularly, reducing the risk of passwords being cracked. Weak password detection prohibits the use of common weak passwords. Login failure lockout prevents brute-force attacks. OTP two-factor authentication adds a one-time password verification on top of the password, greatly improving account security.

Slow attack protection detects and blocks attacks that exploit HTTP protocol flaws, such as Slowloris and Slow POST.

## 4.15 JMX Service

AAS provides JMX management capabilities for monitoring and managing server runtime status.

JMX MBeans are the core components managed by AAS, including server MBeans, application MBeans, data source MBeans, and thread pool MBeans. Through MBeans, runtime information such as the server's CPU usage, memory usage, thread count, and connection pool status can be obtained.

The AAS JMX service supports remote JMX management: JMX clients (such as JConsole and VisualVM) can connect remotely to application server instances for monitoring and management, including modifying configuration parameters, triggering garbage collection, and obtaining thread stacks. JMX MBeans can be converted into SNMP OIDs by an SNMP adapter, achieving integration with enterprise monitoring platforms.

## 4.16 Observability

AAS features a unified observability system that supports comprehensive capabilities such as monitoring metrics, log analysis, performance profiling, and distributed tracing, helping development and operations personnel diagnose, analyze, and locate application faults.

The AAS monitoring management system organizes monitored objects in a tree structure, covering runtime metrics such as JVM memory, threads, GC, HTTP connections, database connection pools, and message queues. Monitoring data can be integrated with open-source monitoring platforms such as Prometheus and Zabbix, supports proactively sending alert events to the monitoring center, and supports multiple notification channels such as email, SNMP, and JMS. Full GC monitoring records the trigger reason, duration, and memory changes before and after each garbage collection.

Threshold alerting triggers a notification when a metric exceeds a configured threshold. AAS provides a comprehensive performance diagnostic toolchain, supporting capabilities such as call chain analysis and flame graphs.

## 4.17 Log Management

AAS log configuration supports setting different log levels by package name and supports log rotation (by size or by time), with rotated historical logs able to be compressed and archived. It supports log pushing, outputting logs to message queues such as Kafka.

The AAS access audit log records detailed information about each HTTP request: request time, client IP, request method, request URL, response status code, response bytes, response time, and so on.

## 4.18 Application Diagnostics

AAS call chain analysis displays the complete invocation path of a request from receipt to completion and the time consumed at each node, quickly locating the source of latency. Performance collection generates flame graphs that visually display the proportion of CPU time consumed by each method, helping to quickly identify performance hotspots. Slow request tracing records HTTP requests whose processing time exceeds a threshold, including information such as request parameters, processing duration, and thread stack. Busy thread monitoring identifies threads with high CPU usage, and blocked thread monitoring identifies threads waiting for lock resources; thread

information can be downloaded for further analysis, and problematic threads can be manually stopped when necessary.

AAS supports one-click generation of a system status snapshot when a problem occurs, including process information, thread stacks, heap memory distribution, configuration files, and log files.

## 4.19 Operations Management

AAS supports patch upload and installation, version description viewing, and rollback operations. It supports server lifecycle management and graceful shutdown, as well as script recording and batch execution.

The Migration Tool supports migrating from mainstream application servers such as Apache Tomcat and Oracle WebLogic to Apusic: it automatically scans the source application's code and configuration, identifies the features and APIs in use, and assesses migration difficulty; it converts deployment descriptors and configuration files; and it ultimately generates a migration report and operation recommendations, reducing migration cost and risk.

The SNMP monitoring agent supports integration with enterprise monitoring platforms, and SNMP traps can be configured for proactive event notification.

It supports threshold alerting for metrics such as CPU, memory, and disk.

The monitoring playback feature supports replaying historical monitoring data for a specified time range.

## 4.20 CLI Command-Line Tool

The AAS CLI (Command Line Interface) is a command-line management tool provided by AAS, intended for operations engineers and AI Agents.

The CLI supports server lifecycle management operations, including server start, stop, restart, and status query. It supports application deployment and undeployment, and the creation, modification, and deletion of resources such as data sources, JMS, and thread pools. It supports configuration management operations, allowing server configuration parameters to be viewed and modified. It supports batch operations on multiple instance nodes, enabling the same operation to be performed on multiple AAS server instances at once, which is suitable for cluster environment management.

The CLI also supports integration with AI Agents, enabling them to perform automated operations management of AAS through the CLI, covering scenarios such as automatic scaling, self-healing, and intelligent inspection.

## 4.21 Copilot Assistant

The Kingdee Apusic Application Server web management console integrates the Kingdee Apusic Copilot assistant service, which supports querying AAS product feature descriptions and configuration methods through natural

language interaction.

The Copilot assistant is built on AAS product documentation and a knowledge base, and adopts retrieval-augmented generation (RAG) technology: when a user asks a question, relevant document fragments are first retrieved from the knowledge base and then handed to a large language model to organize an answer based on them, so that answers are grounded in the original documentation rather than freely generated by the model, reducing the risk of inaccurate content. The knowledge base can be updated in sync with product version iterations, so that new features are covered without retraining the model.

Developers and operations personnel can use Copilot to quickly obtain information such as AAS usage guidance, troubleshooting suggestions, and best practice recommendations. Copilot supports multi-turn conversations and can understand user intent based on context, providing coherent technical support services.

In addition to the web management console, the Copilot assistant can also be used as a standalone tool.

## 5 Product Editions

Kingdee Apusic Application Server is available in Enterprise Edition, Standard Edition, and Intelligent Security Edition, to meet the needs of different business scenarios.

### 5.1 Enterprise Edition

The Kingdee Apusic Application Server Enterprise Edition fully supports the Jakarta EE specifications and provides centralized management and control. It supports configuring server clusters to achieve load balancing and failover, supports elastic scaling and intelligent routing of cluster instances, and supports a variety of instance detection methods. It provides a class loading conflict detection tool, snapshot functionality, monitoring configuration, and other features.

This edition mainly targets large-scale distributed applications and scenarios with high requirements for reliability and scalability, such as large enterprise systems and government, finance, transportation, energy, and telecommunications systems.

### 5.2 Standard Edition

The Kingdee Apusic Application Server Standard Edition is a Jakarta EE–conformant web application server with fast startup and a lightweight footprint, able to quickly deploy and run applications in resource-constrained environments, and supporting the rapid development of Java microservices applications such as those based on Spring Boot.

This edition targets Java distributed web application scenarios, as well as cloud container scenarios that require low resource usage and fast startup.

### 5.3 Intelligent Security Edition

The Kingdee Apusic Application Server Intelligent Security Edition builds on the Enterprise Edition by adding a real-time application security protection module, providing precise and comprehensive security detection and defense for enterprise applications.

The Intelligent Security Edition precisely analyzes the behavior of user input within an application and, based on the analysis results, distinguishes legitimate behavior from attack behavior to implement efficient defense. It plays a key role in security scenarios that are difficult to handle with traditional perimeter-based firewalls. It can effectively defend against common OWASP security issues, protect the operational security of application systems, and enhance the data security of application systems. It is suitable for scenarios that require more advanced application security defense capabilities.

## 6 Product Benefits

### 6.1 Fully Independent Intellectual Property Rights

Since its first line of code in 1999, Kingdee Apusic Application Server has been entirely independently developed, with a self-developed core code rate of 100% and intellectual property rights to all core technologies. Over more than two decades, the product has undergone the complete specification evolution from J2EE 1.3 to Jakarta EE 11, and is at a leading domestic level in its understanding and depth of implementation of the Java EE specifications.

Fully independent intellectual property rights mean that users can fully control their technology stack, avoid dependence on a single foreign vendor, and fundamentally eliminate supply chain risk. Kingdee Apusic Application Server has passed source code security audits by authoritative national testing institutions, ensuring no backdoors and no medium- or high-risk vulnerabilities, providing a credible basis for selection in industries with high security requirements such as government and finance.

### 6.2 Outstanding Product Performance

With its performance optimization scheme, Kingdee Apusic Application Server can support one million online users on a single instance in a standard test scenario, handling ultra-large-scale concurrent access and fully meeting the needs of high-traffic business systems such as large e-commerce platforms and government portals, as well as industry-critical business applications.

### 6.3 Enterprise-Grade Application Security Protection

National cryptographic algorithm support is the foundation for meeting cryptographic compliance requirements. The three national cryptographic algorithms SM2/SM3/SM4 cover security scenarios such as digital signatures, message integrity, and data encryption, and Kingdee Apusic Application Server supports invoking national cryptographic functions through the standard Java Security API, meeting the cryptographic compliance needs of industries with high security requirements such as government and finance.

Classified protection compliance support enables the system to meet the requirements of the multi-level protection scheme. Classified protection is a fundamental system of information security in China, and critical information systems are required to meet the corresponding security level requirements. The three-role separation and security auditing mechanisms of Kingdee Apusic Application Server can meet the requirements of Classified Protection Level 3.

### 6.4 Support for Mainstream Development Frameworks

The product is fully compatible with mainstream Java development frameworks, covering the main technology stacks of enterprise application development: the web layer supports Spring MVC, Spring Boot, and Struts2; the persistence layer supports Hibernate, MyBatis, and Spring Data JPA; microservices and service governance support Spring Cloud and Apache Dubbo; and it is compatible with common components such as Spring Security, Apache Shiro, and Quartz. An enterprise's existing technology stack can be deployed and run directly, without refactoring business code in order to change the application server; the core frameworks have completed compatibility verification, reducing selection and migration costs.

The product fully implements standard components such as JPA, JSF, CDI, RESTful, and Web Service defined in the Jakarta EE specifications. Applications can either use these standard capabilities directly or use them together with frameworks such as Spring in the same application, covering common development needs such as dependency injection, object persistence, page rendering, and service publishing.

Class conflicts between a framework and the server are a typical problem in application deployment. The product supports custom class loading strategies that can adjust the loading order of application and server class libraries; it provides class loading conflict detection and generates a detection report to help developers quickly locate dependency conflicts. The product also supports deploying applications developed on the OSGi framework, allowing browsing, dynamic dependency adjustment, reloading, and starting at runtime.

## 6.5 Application Server Migration Tool

Kingdee Apusic Application Server comes with an application server migration tool. Aimed at Xinchuang (domestic IT innovation) middleware replacement scenarios, it organizes migration implementation into a standardized four-phase pipeline — "Assessment & Analysis — Migration Preparation — Migration Execution — Verification & Acceptance" — with the entire process performed through a GUI-based web console. The Migration Tool supports migrating to AAS from mainstream application servers such as Apache Tomcat 9/10/11 (including Spring Boot embedded containers) and Oracle WebLogic. Migration implementation is non-intrusive to the application source environment: the source side is scanned in read-only mode and no Agent needs to be installed on the target side; remote (SSH) mode is supported, which is suitable for intranet migration from a source environment to a Xinchuang target environment. Configuration differences are automatically converted by a rules engine with a before-and-after comparison generated, and uncovered items can be adjusted manually. On exceptions, migration automatically pauses and preserves the scene, supporting resume-from-failure and one-click rollback; each migration automatically generates a standard migration report that can be exported as PDF/DOCX for Xinchuang acceptance and results archiving. The Migration Tool follows the relevant technical standard of T/CESA 9172-2021, Guide for Application Server Middleware Migration.

## 6.6 High-Quality Service Assurance

Kingdee Apusic has established a technical service network covering the whole country, with 6 major regions and 50 branch offices across more than 200 cities and regions. Wherever customers are, they can obtain local technical

support services.

Service channels include hotlines, online customer service, technical communities, knowledge bases, and regional centers. The 7×24 hotline ensures that customers can obtain timely help when they encounter urgent problems. Technical communities provide a platform for exchanging experience with other users, and the knowledge base has accumulated a large number of technical documents and answers to frequently asked questions.

## 7 Application Scenarios

### 7.1 Government System Applications

Government systems are the core supporting platform for the operation of state administrative organs, carrying critical business such as administrative approval, public services, and data sharing. Government systems are usually deployed on government extranets or the internet, in complex environments with stringent security and compliance requirements.

Government systems must meet the requirements of the multi-level protection scheme and undergo cryptographic application compliance assessment; national cryptographic algorithm support and classified protection certification are hard conditions for selection. At the same time, government systems are increasingly deployed on domestic chips and domestic operating systems, and the application server must be fully adapted to the Xinchuang environment. In addition, government systems involve large volumes of citizen data and government data, making security auditing and access control capabilities indispensable.

Kingdee Apusic Application Server fully meets the needs of government systems. The product supports the SM2/SM3/SM4 national cryptographic algorithms and can invoke national cryptographic encryption capabilities through the standard Java Security API, meeting cryptographic application compliance requirements; it has passed Classified Protection Level 3 certification, and its three-role separation management mechanism meets the requirement of checks and balances of power; and its complete operation log and security audit functions support the tracing of security incidents. The product has completed adaptation verification with domestic chips such as Kunpeng, Loongson, Phytium, Hygon, and Sunway, and domestic operating systems such as Kylin OS and Deepin, and supports security mechanisms such as TLS 1.3 encrypted transmission, OTP two-factor authentication, and fine-grained access control.

### 7.2 Core Enterprise Business

Core enterprise business systems include enterprise-grade applications such as ERP, MES, and CRM, and are the hub of enterprise business operations. Such systems usually need to run uninterrupted 7×24, and business interruption directly causes economic loss, so they have extremely high requirements for availability and performance.

Core enterprise business systems need failover capability — a single point of failure must not cause business interruption; they need to scale out quickly to cope with traffic growth during peak periods; core enterprise business often involves large numbers of concurrent users and complex business logic, requiring the application server to have outstanding concurrency handling capability; and they also need to interconnect with multiple surrounding systems, requiring comprehensive service integration capability.

Kingdee Apusic Application Server provides solid support for core enterprise business systems. The product supports multi-instance cluster deployment, with instances sharing applications and configuration; it has a built-in ALB load balancer and supports session affinity to ensure continuity of user access; and crash detection and automatic restart mechanisms ensure rapid failure recovery. AAS can support one million online users on a single instance in a standard test scenario and supports the latest web technology specifications such as HTTP/2 multiplexing. The cluster supports dynamic scaling, allowing the number of instances to be adjusted flexibly according to business load. It supports mainstream integration technologies such as Web Service (SOAP/REST), JMS message queues, and JDBC data sources, and the EJB container supports distributed transaction processing to ensure cross-system data consistency.

## 7.3 Financial Business Applications

Financial systems include bank core systems, securities trading systems, and insurance business systems, and are a critical part of the national financial infrastructure. Financial systems have the most stringent requirements for security and reliability among all industries, and any security incident could trigger systemic financial risk.

Financial systems must meet the regulatory requirements of the financial industry, support national cryptographic algorithms and cryptographic compliance, and provide capabilities such as multi-factor authentication, security auditing, and data encryption; they usually require availability above 99.99%, requiring comprehensive high-availability and disaster recovery capabilities; scenarios such as securities trading need to respond to massive concurrent requests at the millisecond level, imposing extremely high requirements on latency and throughput; and they also need to pass industry security assessments and meet the requirements of regulatory bodies such as the National Financial Regulatory Administration and the China Securities Regulatory Commission.

Kingdee Apusic Application Server provides a highly secure and highly reliable runtime environment for financial systems. The product has passed Classified Protection Level 3 certification and fully supports the SM2/SM3/SM4 national cryptographic algorithms, supports TLS 1.3 encrypted transmission and two-way certificate authentication, and its OTP two-factor authentication and fine-grained access control meet the multi-factor authentication requirements of the financial industry. The product supports multi-active cluster deployment, and distributed session storage ensures that sessions are not lost in failure scenarios; its complete monitoring and alerting system supports early detection and early handling of problems. The product supports high-concurrency, low-latency request processing, and JVM tuning and thread pool management support financial-grade performance requirements. It has passed relevant security assessments for the financial industry and supports cryptographic application compliance testing and audit log output, meeting regulatory reporting requirements.

## 7.4 Java Cloud-Native Applications

Cloud native has become the mainstream technology paradigm for enterprise digital transformation. Microservices architecture splits a monolithic application into multiple autonomous services, achieving rapid iteration and elastic

scaling. Containerization and Kubernetes orchestration further improve deployment flexibility and resource utilization efficiency.

Cloud-native scenarios require the application server to be deployable as container images, supporting Docker and Kubernetes environments; to support cloud-native features such as health checks, horizontal scaling, and environment variable configuration; to support microservices governance capabilities such as service registration and discovery, load balancing, circuit breaking, and rate limiting; and to pursue faster startup and lower resource consumption.

Kingdee Apusic Application Server fully embraces cloud-native technologies. The product provides standard Docker images, supporting rapid deployment and migration in container environments; supports the Kubernetes health check interface and can integrate with the Horizontal Pod Autoscaler (HPA), and supports ConfigMap and Secret configuration management; it has a built-in circuit-breaking and rate-limiting mechanism to protect system stability, distributed session management to support session sharing across instances, and a VIP thread pool to guarantee resources for critical services; it supports JDK 21 virtual threads to improve concurrency processing efficiency, supports configuration centers such as Apollo, Nacos, and ZooKeeper for dynamic configuration updates, and supports APM tools such as SkyWalking and Pinpoint for distributed tracing.

## 8 Customers and Cases

The customers of Kingdee Apusic Application Server cover the Xinchuang "2+8+N" industries such as government, finance, telecommunications, transportation, healthcare, and energy, with more than 10,000 customers and more than 200 partners. The product is highly recognized by the market and has won a number of first places:

- The application server product has won first place in user satisfaction for many consecutive years
- First in the most widely used industries and the number of installations among domestic middleware products
- First in the number of users among national central ministries and commissions
- First in the number of users among local governments
- First in the number of users among Fortune 500 enterprises

A partial list of customers of Kingdee Apusic Application Server is as follows:

### Central Ministries and Commissions

General Office of the CPC Central Committee, Cryptographic Bureau of the General Office of the CPC Central Committee, Organization Department of the CPC Central Committee, Central Commission for Discipline Inspection, International Department of the CPC Central Committee, National Committee of the Chinese People's Political Consultative Conference, National People's Congress, National Supervisory Commission, Supreme People's Court, Supreme People's Procuratorate, United Front Work Department of the CPC Central Committee, Overseas Chinese Affairs Office of the State Council, Central and State Organs Work Committee, National Development and Reform Commission, Ministry of Foreign Affairs, Ministry of Science and Technology, Ministry of Human Resources and Social Security, Ministry of Ecology and Environment, Ministry of Natural Resources, Ministry of Transport, Ministry of Emergency Management, Ministry of Agriculture and Rural Affairs, Ministry of Education, Ministry of Industry and Information Technology, Ministry of Finance, Ministry of Civil Affairs, Ministry of Water Resources, Ministry of Housing and Urban-Rural Development, National Audit Office, People's Bank of China, General Administration of Customs, National Radio and Television Administration, State Administration for Market Regulation, National Bureau of Statistics, State-owned Assets Supervision and Administration Commission of the State Council, Chinese Academy of Sciences, China Meteorological Administration, National Energy Administration, China Geological Survey, National Public Complaints and Proposals Administration, National Archives Administration, National Administration of State Secrets Protection, National Food and Strategic Reserves Administration, Civil Aviation Administration of China, China National Intellectual Property Administration, National Administration of Traditional Chinese Medicine, National Commission for Disaster Reduction, and others.

### Local Governments

Beijing Municipal Committee of the CPC, Beijing Municipal Government, Political and Legal Affairs Commission of the Beijing Municipal Committee, Beijing Municipal Bureau of Civil Affairs, Organization Department of the Beijing

Municipal Committee, Beijing Municipal Emergency Management Bureau, Beijing Municipal Transport Comprehensive Law Enforcement Corps, Beijing Municipal Bureau of Culture and Tourism, Beijing Women's Federation, Beijing Water Authority, Shanghai Municipal Government, Organization Department of the Shanghai Municipal Committee, Shanghai Municipal Bureau of Civil Affairs, Shanghai Municipal Commission of Economy and Informatization, Shanghai Municipal Archives, Shanghai Municipal Administration for Market Regulation, Overseas Chinese Affairs Office of Shanghai Municipality, Shanghai Municipal Conscription Office, Commission for Discipline Inspection of Guangdong Province, Guangdong Provincial Public Security Department, Guangdong Provincial Supervisory Commission, Guangdong Provincial Department of Civil Affairs, Guangdong Provincial Administration for Market Regulation, Guangdong Provincial Department of Transport, Guangdong Provincial Department of Industry and Information Technology, Shenzhen Municipal Government, State-owned Assets Supervision and Administration Commission of Shenzhen Municipality, Shenzhen Tax Service, Shenzhen Municipal Supervisory Commission, Shenzhen Municipal Archives Bureau, Jiangsu Maritime Safety Administration, Heilongjiang Provincial Government, Organization Department of the Jilin Provincial Committee, Liaoning Provincial Government, Shaanxi Provincial Committee of the CPC, Shaanxi Provincial Audit Office, Gansu Provincial Government, Xinjiang Uygur Autonomous Region Government, Inner Mongolia Autonomous Region Government, Hubei Provincial Government, Hebei Provincial Government, Hunan Provincial Government, Henan Provincial Government, Organization Department of the Guizhou Provincial Committee, Sichuan Provincial Government, Yunnan Provincial Government, Guangzhou Municipal Government, Nanning Municipal Government, Suzhou Industrial Park, Hainan Provincial Government, Xinjiang Production and Construction Corps, and others.

## Industry Customers

CASIC, Aisino, Aviation Industry Corporation of China, China South Industries Group, Eastern Theater Command, Naval University of Engineering, Beijing Aerospace Intelligent Manufacturing, Xi'an National Civil Aerospace Industrial Base, China Academy of Engineering Physics, Beijing Institute of Spacecraft Environment Engineering, Shenyang Aircraft Design and Research Institute, Institute of Information Engineering of the Chinese Academy of Sciences, China Construction Bank, City Commercial Bank, China Everbright Bank, Export-Import Bank of China, Huaxia Bank, China Merchants Securities, Sunshine Insurance, China Pacific Insurance, Central China Securities, China Cinda Asset Management, China Unicom System Integration Company, China Telecom System Integration Company, China Unicom Guangzhou Branch, China Telecom (Guangdong, Liaoning, Hunan, Hubei), China Mobile (Guizhou, Sichuan, Hebei, Shenzhen), Beijing Guodiantong, Hunan China Tobacco, Yueyang Tobacco, Changde Tobacco, Chenzhou Tobacco, Hengyang Tobacco, Shaoyang Tobacco, and others.

## Group Enterprises

China Railway Construction Corporation, China State Shipbuilding Corporation, CRRC Corporation, COFCO Group, China Tourism Group, China National Postal and Telecommunications Appliances Group, China National Cereals, Oils and Foodstuffs Corporation Oils & Fats, China National Offshore Oil Corporation, China National Construction Machinery Corporation, China United Cement Group, China Aviation Supplies Group, China Metallurgical Geology Bureau, China Minmetals, China National Coal Group, State Grid Corporation of China, China Huadian Corporation,

Xidian Group, CHN Energy, Guangdong Power Grid, Tonghua Mining, Sichuan Coal Industry Group, Tonghua Iron and Steel, Beijing Capital International Airport, China Eastern Airlines Group, Shanghai Airlines, Shenzhen Airlines, Beijing Public Transport Group, Shanghai Pudong Public Transport Group, Chongqing Rail Transit, Guangdong Province Shipping Group, Wenzhou Port, Dalian Port, Guangzhou Port, Haier Group, China Merchants Group, Vanke Group, Lingnan Group, New Hope Group, ENN Group, Rongcheng Iron and Steel, Bright Food, Liaoning Fangda Group, Anhui State Farms Group, Beidahuang Grain Group, Northern International Steel Electronic Trading Center, Guangdong Wens Group, Guangdong Salt Industry Group, Nissan (China) Investment Co., Ltd., Xinxing Cathay International Group, and others.

## 9 Appendix: Supported Jakarta EE Specifications

Table 3 Supported Jakarta EE specifications

| Specification                                | Version   |
|----------------------------------------------|-----------|
| Servlet                                      | 4.0 / 6.0 |
| JavaServer Pages (JSP)                       | 2.3 / 3.1 |
| Expression Language (EL)                     | 3.0 / 5.0 |
| Java API for WebSocket                       | 1.1 / 2.1 |
| JavaServer Faces (JSF)                       | 2.3 / 4.0 |
| JavaServer Pages Standard Tag Library (JSTL) | 1.2       |
| JAX-RS (RESTful Web Services)                | 2.1 / 3.1 |
| JAX-WS (XML-Based Web Services)              | 2.2       |
| JAX-RPC                                      | 1.1       |
| Enterprise JavaBeans (EJB)                   | 3.2       |
| Java Persistence (JPA)                       | 2.2 / 3.2 |
| Java Message Service (JMS)                   | 2.0       |
| Contexts and Dependency Injection (CDI)      | 2.0 / 4.0 |
| Bean Validation                              | 2.0 / 3.0 |
| JavaMail                                     | 1.6       |
| Java EE Connector Architecture (JCA)         | 1.7       |
| Batch Applications                           | 1.0       |
| Concurrency Utilities                        | 1.0       |
| Java EE Security / Jakarta Security          | 1.0 / 3.0 |
| JASPIC                                       | 1.1       |
| JACC                                         | 1.5       |
| JSON-P (JSON Processing)                     | 1.1       |

|                       |     |
|-----------------------|-----|
| JSON-B (JSON Binding) | 1.0 |
| JAXB                  | 2.2 |

# 10 Appendix: Glossary

Table 4 Glossary

| Term       | Full Name                                       | Description                                                                 |
|------------|-------------------------------------------------|-----------------------------------------------------------------------------|
| AAS        | Apusic Application Server                       | Kingdee Apusic Application Server                                           |
| ALB        | Apusic Load Balancer                            | Kingdee Apusic Load Balancer                                                |
| AMDC       | Apusic In-Memory Data Cache                     | Kingdee Apusic Distributed Cache                                            |
| Jakarta EE | Jakarta Enterprise Edition                      | Open-source enterprise Java specification set led by the Eclipse Foundation |
| Servlet    | Java Servlet                                    | Specification for server-side programs written in Java                      |
| JSP        | JavaServer Pages                                | Java technology for developing dynamic web pages                            |
| EJB        | Enterprise JavaBeans                            | Enterprise component model of the Java EE platform                          |
| JCA        | Java EE Connector Architecture                  | Java connector architecture                                                 |
| JPA        | Java Persistence API                            | Java persistence API                                                        |
| JMS        | Java Message Service                            | Java messaging service specification                                        |
| JDBC       | Java Database Connectivity                      | Java database connectivity specification                                    |
| JNDI       | Java Naming and Directory Interface             | Java naming and directory interface                                         |
| JTA        | Java Transaction API                            | Java transaction processing API                                             |
| JMX        | Java Management Extensions                      | Java management extensions                                                  |
| JAX-RS     | Java API for RESTful Web Services               | RESTful web services specification                                          |
| CDI        | Contexts and Dependency Injection               | Contexts and dependency injection                                           |
| SSL/TLS    | Secure Sockets Layer / Transport Layer Security | Secure sockets layer / transport layer security protocol                    |

|             |                                    |                                                                                                                              |
|-------------|------------------------------------|------------------------------------------------------------------------------------------------------------------------------|
| OTP         | One-Time Password                  | One-time password                                                                                                            |
| Kubernetes  | Kubernetes                         | Container orchestration platform                                                                                             |
| CLI         | Command Line Interface             | Command-line interface                                                                                                       |
| JDK         | Java Development Kit               | Java development kit                                                                                                         |
| JVM         | Java Virtual Machine               | Java virtual machine                                                                                                         |
| MBean       | Managed Bean                       | Component managed by JMX                                                                                                     |
| SNMP        | Simple Network Management Protocol | Simple network management protocol                                                                                           |
| HPA         | Horizontal Pod Autoscaler          | Kubernetes horizontal pod autoscaler                                                                                         |
| APM         | Application Performance Management | Application performance management                                                                                           |
| RAG         | Retrieval-Augmented Generation     | Retrieval-augmented generation                                                                                               |
| LLM         | Large Language Model               | Large language model                                                                                                         |
| AI Agent    | Artificial Intelligence Agent      | An agent: an AI program that autonomously plans, invokes tools, and executes tasks                                           |
| SM2/SM3/SM4 | ShangMi 2/3/4                      | Domestic commercial cryptographic algorithms, used respectively for digital signatures, message digests, and data encryption |

UNIFIED SERVICE HOTLINE

4008-555-800

Kingdee Apusic Cloud Computing Co., Ltd. (referred to as “Kingdee Apusic” ) was founded in 2000. Its predecessor was “Kingdee Middleware Company.” It is a basic software platform provider under Kingdee Group, a national standards-setting enterprise for cloud computing, and a core software enterprise in the national IT application innovation industry. Kingdee Apusic is an enterprise undertaking the National 863 Key R&D Program and the National Major Science and Technology Project for Core Electronic Components, High-end General-purpose Chips and Fundamental Software, and is also a basic platform provider for the national key projects of “Two Networks, One Website, Four Databases, and Twelve Golden Projects.” Its products are widely used in key industries such as government, military, finance, and energy, and it has cumulatively served more than 100,000 customers.



National standard setter for cloud computing  
Basic software enterprise under Kingdee Group  
Core enterprise for IT application innovation  
Website: [www.apusic.com](http://www.apusic.com)

