AiX Architecture

v2026.3

DDX Business Solutions Limited

Customer Reference

Table of Contents

1. Architecture

1.1 Overview

■# AiX Platform Architecture

Overview

AiX is a modular enterprise AI platform designed to support business applications, AI-assisted workflows, knowledge processing, document processing, and automation across cloud, on-premises, hybrid, and local deployment environments.

The platform separates application services, AI orchestration, task execution, AI model services, data persistence, authentication, and platform management into independent service components.

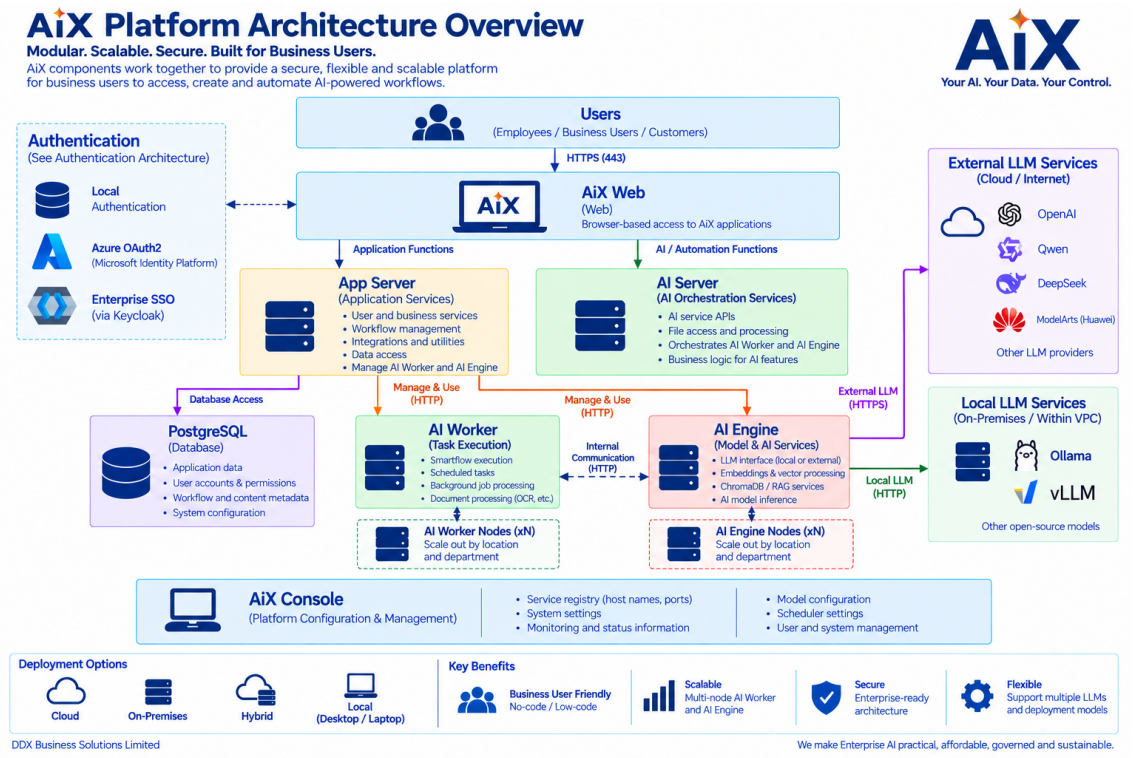
This modular architecture allows AiX to operate as a compact single-host deployment or scale across multiple infrastructure nodes according to workload, organizational, security, and availability requirements.

AiX supports both external and locally hosted AI models while allowing application data, user access controls, workflow execution, and platform configuration to remain within the designated deployment environment.

Platform Architecture Overview

The following diagram illustrates the primary AiX platform components and their logical relationships.

<figure markdown="span">



AiX platform architecture overview

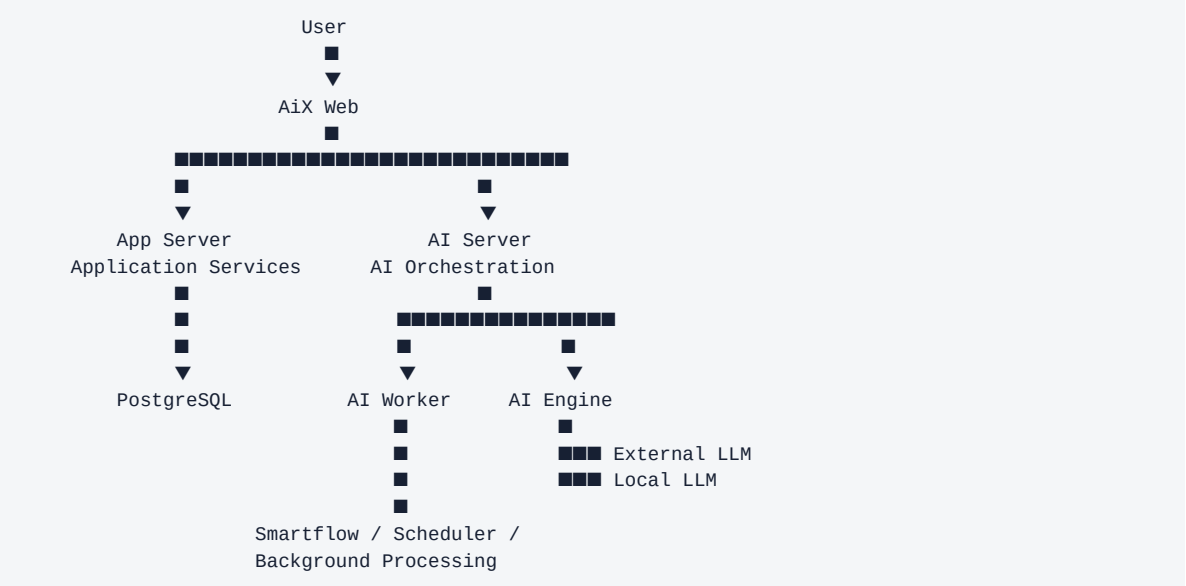
<figcaption>Figure: AiX Platform Architecture Overview</figcaption>

</figure>

Users access the platform through **AiX Web**. Depending on the requested function, AiX Web communicates directly with either the **App Server** or the **AI Server**.

The two backend paths provide a logical separation between application functions and AI or automation functions:

text



The **App Server**, **AI Server**, **AI Worker**, and **AI Engine** operate as separate services with defined responsibilities. This allows individual components to be maintained and scaled independently.

Core Platform Components

AiX Web

AiX Web provides the browser-based user interface for the AiX platform.

Users connect to AiX Web through HTTPS. AiX Web then communicates with the appropriate backend service depending on the requested operation:

- **App Server** for application and business functions
- **AI Server** for AI and automation functions

AiX Web therefore does not depend on a single backend gateway for all platform operations.

App Server

The **App Server** provides the core application service layer.

Its responsibilities include:

- User and business services
- Application and workflow management
- Integrations and utilities

- Application data services
- User and permission-related functions
- Shared application services required by AI Worker and AI Engine

The App Server communicates with PostgreSQL for persistent application data.

AI Worker and AI Engine may also interact with the App Server when task execution or AI processing requires application-level information or services.

PostgreSQL

PostgreSQL provides the primary relational database for the AiX platform.

It stores persistent platform information such as:

- Application data
- User and permission information
- Workflow and content metadata
- System configuration
- Other application-level metadata

PostgreSQL is accessed through the appropriate backend services and is not directly exposed to AiX Web or end users.

AI Server

The **AI Server** provides the orchestration and API layer for AI and automation-related operations.

Its responsibilities include:

- AI service APIs
- File access and processing coordination
- AI and automation business logic
- AI Worker orchestration
- AI Engine orchestration

When AiX Web requests an AI or automation operation, the AI Server coordinates the appropriate processing service.

The AI Server does **not** directly invoke local or external LLM services. Model access is handled through the AI Engine.

AI Worker

The **AI Worker** provides workflow, task, document, and background processing capabilities.

Its responsibilities include:

- Smartflow execution

- Scheduled task execution
- Background job processing
- Document processing
- OCR and other processing workloads

The **Scheduler** operates as part of the AI Worker and initiates scheduled Smartflows and background tasks according to configured schedules.

During execution, AI Worker may interact with the App Server for application services and with AI Engine where AI or model capabilities are required.

AI Engine

The **AI Engine** provides the model and AI service abstraction layer.

Its responsibilities include:

- AI model inference
- Local and external LLM integration
- Embedding generation
- Vector processing
- ChromaDB and RAG-related services

AI Engine is the primary AiX component responsible for communication with underlying AI model services.

This provides a consistent model interface to the rest of the AiX platform and prevents application and orchestration components from being directly coupled to individual model providers.

AI Model Integration

AiX supports both **external** and **locally hosted** AI model services.

External LLM Services

External model services may include:

- OpenAI
- Qwen
- DeepSeek
- Huawei ModelArts
- Other supported or customer-approved model providers

The AI Engine communicates with approved external model services using the corresponding provider APIs, normally over HTTPS.

Local LLM Services

For environments requiring private, on-premises, or offline model execution, AiX supports locally hosted model-serving technologies such as:

- Ollama
- vLLM

Local model services can operate within the same infrastructure environment as AiX or on dedicated AI infrastructure.

The logical model path is therefore:

text

```
AiX Web
    ■
    ▼
AI Server
    ■
    ▼
AI Engine
    ■
    ■■■■■■■ External LLM Services
    ■       OpenAI / Qwen / DeepSeek / ModelArts
    ■
    ■■■■■■■ Local LLM Services
            Ollama / vLLM
```

The AI Server coordinates the AI request, while the AI Engine performs the actual model-service interaction.

Scale-Out Architecture

AI Worker and AI Engine support multi-node deployment.

A basic deployment may use one AI Worker and one AI Engine:

text

```
AI Server
   ■
   ■■■ AI Worker
   ■
   ■■■ AI Engine
```

Larger deployments can introduce additional processing nodes:

text

```
AI Server
   ■
   ■■■ AI Worker
   ■      ■■■ Worker Node 1
   ■      ■■■ Worker Node 2
   ■      ■■■ Worker Node n
   ■
   ■■■ AI Engine
          ■■■ Engine Node 1
          ■■■ Engine Node 2
          ■■■ Engine Node n
```

AI Worker and AI Engine nodes can be assigned and routed according to **location** and **department**.

This supports scenarios such as:

- Additional processing capacity

- Workload distribution
- Department-specific processing
- Location-specific processing
- Dedicated AI model infrastructure
- Data locality requirements
- Controlled access to specific AI resources

AI processing capacity can therefore scale independently from the core application services.

AiX Console

The **AiX Console** provides centralized platform configuration and management information used by AiX services.

Configuration may include:

- Service registry information
- Service host names and ports
- System settings
- Model configuration
- Scheduler settings
- Monitoring and service status information
- User and system management settings

This allows service locations and infrastructure configuration to be managed centrally rather than hard-coded into individual AiX services.

Authentication

Authentication is provided as a separate platform capability.

AiX supports:

- Local authentication
- Microsoft Azure OAuth2 authentication
- Enterprise Single Sign-On through Keycloak

For enterprise SSO, Keycloak can operate as the **AiX SSO Bridge/Gateway**, providing a standardized integration layer between AiX and the customer's enterprise identity infrastructure.

Authentication establishes the user's identity, while AiX maintains application-level authorization such as user status, roles, permissions, and access to platform capabilities.

For additional information, see:

- Authentication Support
- SSO Integration Architecture

Deployment Architecture

AiX components are containerized and can be arranged according to the size and infrastructure requirements of the deployment.

The logical architecture remains consistent regardless of whether the services operate on one host or across multiple hosts.

Single-Host Deployment

For smaller deployments, multiple AiX services can operate as Docker containers on the same server or virtual machine.

For example:

text

```
■■■■■■■■■■■■■■■■■■■■■■ AiX Host / VM ■■■■■■■■■■■■■■■■■■■■■■■■■
■                                                              ■
■  Docker Network                                             ■
■                                                              ■
■  ■■■■■■■■■■■■■■■     ■■■■■■■■■■■■■■■     ■■■■■■■■■■■■■■■■■  ■
■  ■App Server ■     ■ AI Server ■     ■ PostgreSQL  ■  ■
■  ■■■■■■■■■■■■■■■     ■■■■■■■■■■■■■■■     ■■■■■■■■■■■■■■■■■  ■
■                                                              ■
■  ■■■■■■■■■■■■■■■     ■■■■■■■■■■■■■■■     ■■■■■■■■■■■■■■■■■  ■
■  ■ AI Worker ■     ■ AI Engine ■     ■ AiX Console ■  ■
■  ■■■■■■■■■■■■■■■     ■■■■■■■■■■■■■■■     ■■■■■■■■■■■■■■■■■  ■
■                                                              ■
■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
```

Containers on the same Docker network can communicate using **Docker service names and internal service ports**.

For example:

text

```
http://appserver:5080
http://aiserver:5085
http://aiengine:5084
http://aiworker:5087
http://console:5088
```

Internal container-to-container traffic does not need to be exposed publicly.

Where services communicate only within the same trusted Docker host and private Docker network, internal HTTP communication may be used while HTTPS/TLS is applied at externally exposed boundaries.

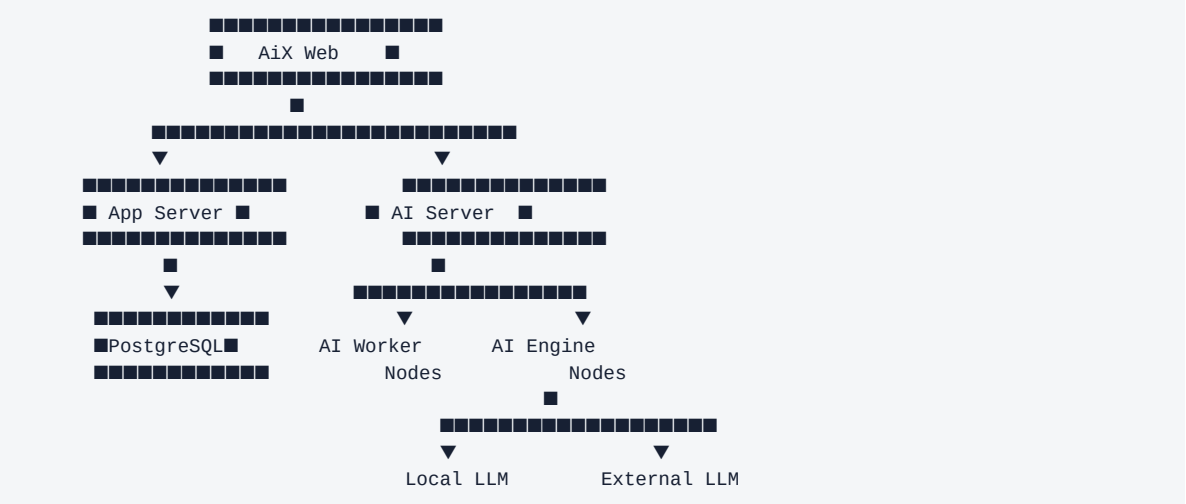
This reduces unnecessary certificate management between containers while maintaining network isolation and service-level authentication.

Multi-Host Deployment

For larger or scale-out deployments, AiX components can operate across separate servers, virtual machines, or infrastructure nodes.

For example:

text



When communication crosses host, VM, subnet, or other network boundaries, the communication should be protected according to the customer's network and security architecture.

Controls may include:

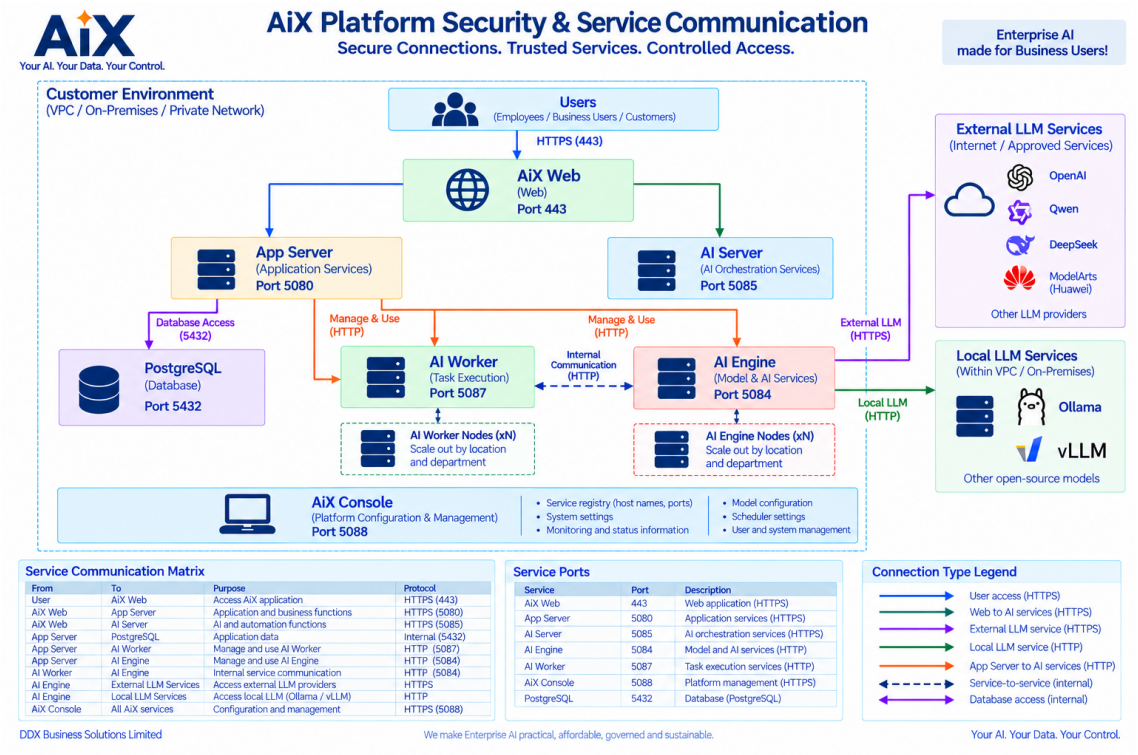
- Private VPC or internal network routing
- Network segmentation
- Firewall and security-group rules
- TLS where required across network boundaries
- Restricted service exposure
- Service-to-service authentication
- Approved outbound connectivity

The same logical AiX service architecture is therefore maintained while the physical deployment topology can evolve independently.

Security and Service Communication

The following diagram illustrates the principal service communication paths and standard service ports within the AiX platform.

<figure markdown="span">



AiX platform security and service communication

<figcaption>Figure: AiX Platform Security & Service Communication</figcaption>

</figure>

The communication architecture distinguishes between:

- User-facing HTTPS access
- Web-to-backend API communication
- Application database access
- AI orchestration
- Task and workflow execution
- Model-service communication
- Internal service-to-service communication
- External model-provider communication

Service Communication

The primary logical communication paths are:

| From | To | Purpose | | ---------- | --------------------- | --- | | User | AiX Web | Access AiX applications | | AiX Web | App Server | Application and business functions | | AiX Web | AI Server | AI and automation functions | | App Server | PostgreSQL | Application data access | | App Server | AI Worker | Application services supporting task execution | | App Server | AI Engine | Application services supporting AI processing | | AI Server | AI Worker | Orchestrate Smartflows, scheduled tasks and background processing | | AI Server | AI Engine | Orchestrate AI and model operations | | AI Worker | AI Engine | Use AI/model capabilities during task execution | | AI Engine | External LLM Services | Access approved external model providers | | AI Engine |

Local LLM Services | Access locally hosted model services | | AiX Console | AiX Services | Platform configuration and service information |

The actual network protocol used for an internal connection depends on the deployment topology and security boundary.

Standard Service Ports

The standard AiX service ports are:

| Service | Port | Purpose | | ---------- | ---: | ---------- | | AiX Web | 443 | Web application / HTTPS | | App Server | 5080 | Application services | | AI Server | 5085 | AI orchestration services | | AI Engine | 5084 | Model and AI services | | AI Worker | 5087 | Task execution services | | AiX Console | 5088 | Platform configuration and management | | AiX SSO Bridge | 5090 | Enterprise SSO integration | | PostgreSQL | 5432 | Database service |

These ports identify the standard service endpoints. They do not imply that every port must be exposed outside the host or deployment environment.

In a containerized deployment, only services that require access from outside the internal Docker network should normally be published or exposed through the appropriate network boundary.

Service-to-Service Authentication

AiX uses a **platform credential** for trusted service-to-service authentication.

This is separate from end-user authentication:

text

```
User Authentication
        ■
        ■■■ Local
        ■■■ Azure OAuth2
        ■■■ Enterprise SSO
        ■
        ▼
      AiX User
   Roles & Permissions


Service Authentication
        ■
        ▼
Platform Credential
        ■
        ▼
Trusted AiX Service
```

The platform credential allows a receiving AiX service to verify that a request originates from an authorized platform service.

Network location alone should therefore not be treated as the only mechanism for establishing service trust.

Data Protection

AiX uses an application encryption key to protect sensitive stored information where application-level encryption is required.

Protected information may include items such as:

- Credentials
- Authentication tokens
- Sensitive configuration values
- Other protected application secrets

This is separate from encryption in transit.

HTTPS/TLS protects communication in transit where required, while the AiX encryption mechanism protects designated sensitive information handled or stored by the application.

Infrastructure-level protections such as encrypted disks, encrypted database storage, private networks, and cloud security controls may provide additional layers of protection depending on the deployment environment.

Network Exposure

AiX follows the principle that services should only be exposed to the networks that require access to them.

A typical deployment therefore separates:

text

```
External / User Access
        ■
        ■ HTTPS
        ▼
      AiX Web
        ■
        ▼
■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
 Internal / Trusted Environment
■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
 App Server
 AI Server
 AI Worker
 AI Engine
 AiX Console
 PostgreSQL
 Local LLM Services
```

Backend services, databases, workers, engines, and locally hosted model services should normally remain within the trusted deployment network unless there is a defined requirement to expose them through another controlled network boundary.

External LLM connectivity is initiated by **AI Engine** to approved model endpoints.

Deployment Models

AiX supports several deployment models:

- **Cloud**

- **On-Premises**
- **Hybrid**
- **Local**

The deployment model does not change the logical responsibilities of the AiX components.

A small deployment may consolidate services onto a single Docker host, while larger deployments can separate application, database, AI processing, and model infrastructure across multiple nodes.

This allows the platform to evolve from a compact deployment into a distributed architecture without changing the fundamental application design.

Architecture Principles

The AiX platform architecture follows several core principles:

Modular — Application, orchestration, processing, model, database, authentication, and management responsibilities are separated into defined services.

Scalable — AI Worker and AI Engine can scale independently through multi-node deployment and location or department-based routing.

Flexible — AiX supports cloud, on-premises, hybrid, and local deployment models together with external and locally hosted LLM services.

Secure by Design — External exposure, internal service communication, user authentication, service authentication, data protection, and network controls are treated as separate security layers.

Infrastructure Independent — The same logical architecture can operate on a single Docker host or across multiple infrastructure nodes.

Model Independent — AI Engine provides an abstraction layer that allows AiX to work with different external and local model services without directly coupling the application layer to a particular LLM provider.

Business-Oriented — The underlying service architecture supports AiX applications, Smartflows, Knowledge Bases, Assistants, scheduled automation, document processing, and other AI-enabled capabilities while abstracting the complexity of the underlying infrastructure.

Download Documentation

Prefer an offline or printable copy of this section?

Download Architecture (PDF)

The PDF contains the complete Architecture documentation for this release and is generated from the same documentation source as this website.

1.2 Authentication Support

Overview

AiX supports multiple authentication methods to accommodate different deployment environments, security requirements, and enterprise identity infrastructures.

Authentication is managed through the **AiX Auth Service**, which provides a consistent authentication and user management layer for the AiX platform.

AiX currently supports three primary authentication methods:

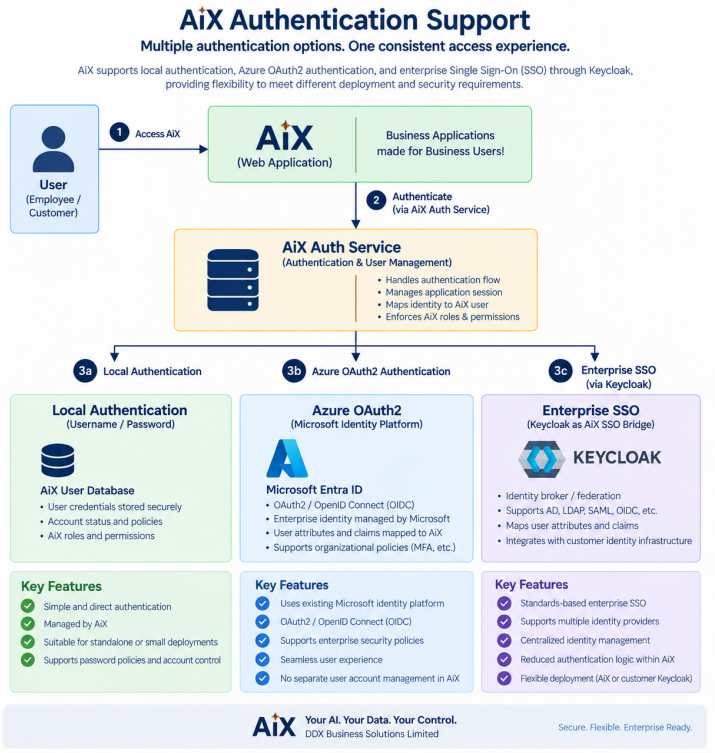
- **Local Authentication** – Username and password authentication managed directly by AiX.
- **Azure OAuth2 Authentication** – Authentication using Microsoft identity services through OAuth2 / OpenID Connect (OIDC).
- **Enterprise Single Sign-On (SSO)** – Enterprise identity integration through Keycloak, which acts as the **AiX SSO Bridge/Gateway**.

Regardless of the authentication method used, AiX maintains application-level user information, roles, permissions, and access controls.

Authentication Architecture

The following diagram illustrates the authentication methods supported by AiX and how they integrate through the AiX Auth Service.

<figure markdown="span">



AiX authentication support

<figcaption>Figure: AiX Authentication Support</figcaption> </figure>

The **AiX Auth Service** provides the common authentication layer between the AiX application and the configured authentication method.

This architecture allows the authentication mechanism to vary according to the deployment environment without requiring the AiX application itself to implement separate authentication flows for each identity platform.

Local Authentication

Local authentication provides direct username and password authentication using credentials managed by AiX.

The AiX Auth Service validates the supplied credentials against the AiX user database and establishes an application session when authentication is successful.

Local authentication can be used where an external identity provider is not available or where a deployment requires AiX-managed user authentication.

AiX remains responsible for the associated user account, account status, roles, permissions, and applicable authentication policies.

Azure OAuth2 Authentication

AiX supports authentication using the **Microsoft identity platform** through OAuth2 / OpenID Connect (OIDC).

With this authentication method, the user's identity is authenticated by Microsoft rather than through a locally managed AiX password.

The authentication result and associated identity information are returned to the AiX Auth Service, which maps the authenticated identity to the corresponding AiX user and establishes the AiX application session.

This approach allows organizations already using Microsoft identity services to use their existing enterprise identities and authentication policies when accessing AiX.

Enterprise Single Sign-On

For enterprise environments requiring integration with different or existing identity infrastructures, AiX supports Single Sign-On through **Keycloak**.

Keycloak acts as the **AiX SSO Bridge/Gateway**, providing an identity brokering and federation layer between AiX and the customer's enterprise identity infrastructure.

Depending on the customer environment, Keycloak can integrate with identity technologies such as:

- Microsoft Active Directory / LDAP
- Microsoft Entra ID
- SAML 2.0 identity providers
- OpenID Connect (OIDC) identity providers
- Existing customer-managed Keycloak environments
- Other identity providers supported through Keycloak

This architecture provides AiX with a standardized enterprise SSO integration while allowing the customer's existing identity infrastructure to remain responsible for user authentication.

For additional information about the enterprise SSO architecture, see AiX SSO Integration Architecture.

Authentication and Authorization

AiX separates **authentication** from **application authorization**.

Authentication determines and verifies the identity of the user. Depending on the configured authentication method, this may be performed by AiX, Microsoft identity services, or the customer's enterprise identity provider through Keycloak.

After authentication, the AiX Auth Service maps the authenticated identity to an AiX user.

AiX then applies application-level authorization, including:

- User account status
- Roles and permissions
- Module access
- Application access
- Other AiX-specific access controls

This separation allows organizations to retain control of their enterprise authentication policies while AiX independently manages application-level authorization.

Authentication Method Selection

The authentication method used for an AiX deployment depends on the customer's infrastructure and security requirements.

| Authentication Method | Authentication Provider | Typical Use | | ---------- | ---------- | ---------- | | **Local Authentication** | AiX | AiX-managed accounts or environments without an external identity provider | | **Azure OAuth2** | Microsoft identity platform | Organizations using Microsoft identities and OAuth2 / OIDC | | **Enterprise SSO** | Customer IdP through Keycloak | Enterprise environments requiring federation with existing identity infrastructure |

The supported authentication architecture allows AiX to operate in standalone, cloud, on-premises, and enterprise environments without coupling the AiX application directly to a specific identity technology.

Enterprise SSO Integration

Enterprise SSO is implemented as an additional integration layer rather than embedding customer-specific identity integrations directly within the AiX application.

The logical architecture is:

text

```
AiX
 ■
 ▼
AiX Auth Service
 ■
 ▼
```

```
Keycloak
(AiX SSO Bridge/Gateway)
 ■
 ▼
Customer Enterprise Identity Provider
```

This approach isolates AiX from differences between customer identity environments and provides a consistent integration point for enterprise authentication.

The detailed architecture, authentication flow, identity federation model, and Keycloak deployment options are described in AiX SSO Integration Architecture.

1.3 Authentication & SSO

Overview

AiX supports enterprise Single Sign-On (SSO) through a dedicated identity integration layer based on **Keycloak**.

Keycloak acts as the **AiX SSO bridge/gateway** between AiX and the customer's enterprise identity infrastructure. This approach provides AiX with a consistent authentication interface while allowing different customer environments to use their existing identity providers and authentication technologies.

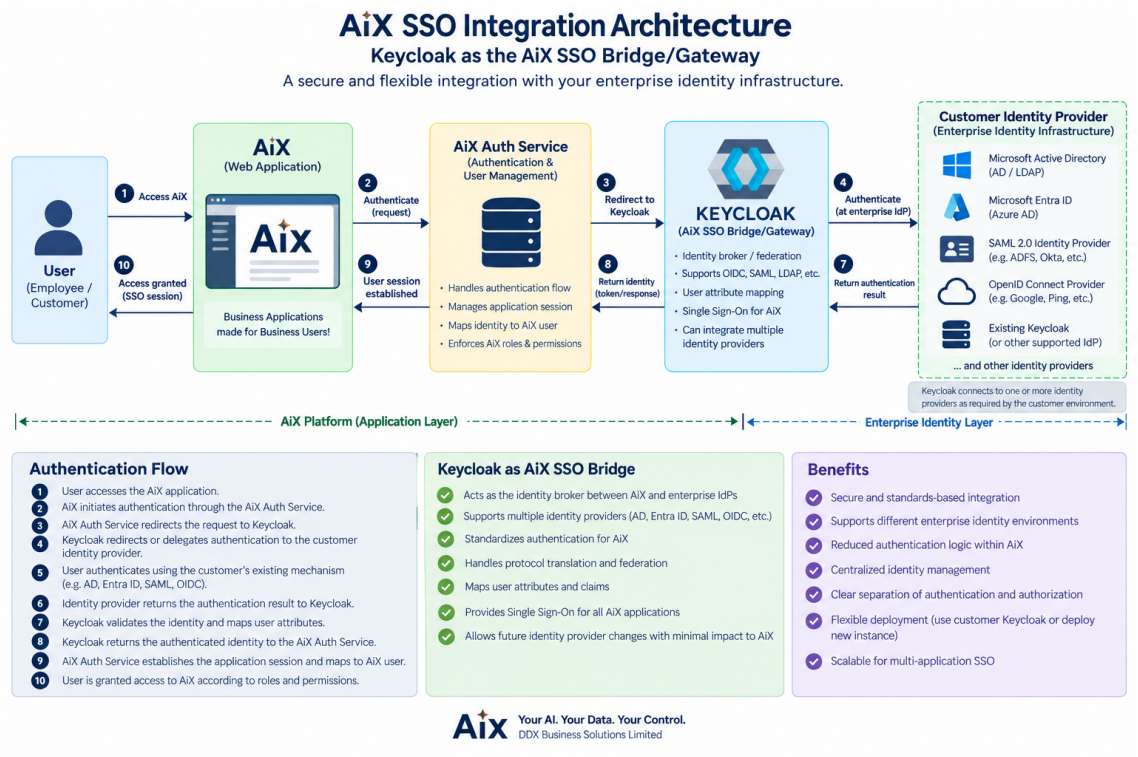
AiX integrates with Keycloak using standard authentication protocols such as **OpenID Connect (OIDC)**. Keycloak manages the connection to the customer's identity provider using the appropriate supported protocol or federation mechanism.

This architecture separates AiX application authentication from customer-specific identity infrastructure and reduces the need for AiX to implement separate authentication logic for each identity platform.

SSO Integration Architecture

The following diagram illustrates the logical integration between AiX, the AiX Auth Service, Keycloak, and the customer's enterprise identity infrastructure.

<figure markdown="span">



AiX SSO integration architecture

<figcaption>Figure: AiX SSO Integration Architecture</figcaption> </figure>

The architecture consists of four primary logical components:

- **AiX Application** – The application accessed by the user.
- **AiX Auth Service** – Manages the AiX authentication flow, application sessions, user identity mapping, and application-level authorization.
- **Keycloak (AiX SSO Bridge/Gateway)** – Provides the standardized identity integration layer between AiX and external enterprise identity providers.
- **Customer Identity Provider** – The customer's existing enterprise identity infrastructure responsible for authenticating the user.

Depending on the customer environment, the identity provider may include technologies such as:

- Microsoft Active Directory / LDAP
- Microsoft Entra ID
- SAML 2.0 identity providers
- OpenID Connect (OIDC) identity providers
- An existing customer-managed Keycloak environment
- Other identity platforms supported through Keycloak

Authentication Flow

When a user accesses AiX using enterprise SSO, the authentication process follows a standardized flow:

1. The user accesses the AiX application.

2. AiX initiates authentication through the AiX Auth Service.
3. The AiX Auth Service redirects the authentication request to Keycloak.
4. Keycloak identifies the configured customer identity provider and redirects or delegates authentication accordingly.
5. The user authenticates using the customer's existing authentication mechanism.
6. The customer identity provider returns the authentication result to Keycloak.
7. Keycloak validates the identity and maps the required user attributes and claims.
8. Keycloak returns the authenticated identity to the AiX Auth Service.
9. The AiX Auth Service maps the authenticated identity to the corresponding AiX user and establishes the application session.
10. The user is granted access to AiX according to the roles and permissions configured within AiX.

Authentication and Authorization Separation

The AiX SSO architecture separates **enterprise authentication** from **application authorization**.

The customer's identity provider is responsible for authenticating the user and establishing the user's enterprise identity.

Keycloak provides the identity brokering and federation layer between the customer's identity environment and AiX. It normalizes the authentication process so that AiX can use a consistent integration regardless of the underlying enterprise identity provider.

The AiX Auth Service remains responsible for AiX-specific application access, including:

- Mapping the authenticated identity to an AiX user
- User account status
- AiX roles and permissions
- Module and application access
- Application session management

This separation allows enterprise authentication policies to remain under the control of the customer's identity infrastructure while AiX maintains control over application-level authorization.

Keycloak as the AiX SSO Bridge/Gateway

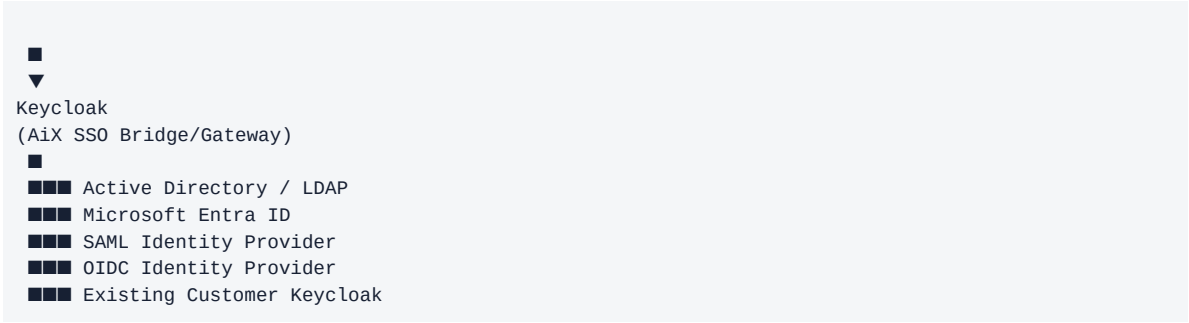
Keycloak provides an abstraction layer between AiX and the customer's identity infrastructure.

Instead of implementing separate authentication integrations within AiX for each identity technology, AiX communicates through a standardized Keycloak integration.

For example:

text

```
AiX
 ■
 ▼
AiX Auth Service
```



If the customer's identity infrastructure changes in the future, the corresponding integration can generally be handled at the Keycloak layer without requiring the AiX application authentication architecture to be redesigned.

Deployment Models

The Keycloak integration layer supports different deployment models depending on the customer's existing infrastructure.

AiX-Provided Keycloak

Where the customer does not already operate a suitable Keycloak environment, a dedicated Keycloak instance can be deployed as part of the AiX SSO integration architecture.

The deployed Keycloak instance acts as the AiX SSO bridge between AiX and the customer's identity infrastructure.

Customer-Managed Keycloak

Where the customer already operates Keycloak, AiX may integrate with the customer's existing Keycloak environment, subject to the customer's architecture, security policies, and configuration requirements.

In this deployment model, an additional AiX-specific Keycloak instance may not be required.

Architecture Benefits

The AiX SSO bridge/gateway architecture provides:

- Standards-based enterprise SSO integration
- Separation between AiX and customer-specific identity infrastructure
- Support for different enterprise identity technologies
- Centralized identity federation and protocol handling
- Reduced authentication-specific logic within AiX
- Consistent authentication integration across customer environments
- Flexibility to accommodate future identity infrastructure changes
- Clear separation between enterprise authentication and AiX authorization

The architecture therefore allows AiX to maintain a consistent authentication model while integrating with the identity infrastructure and security policies already established within the customer's enterprise environment.