DigitalPersona® Altus

Solution Guide

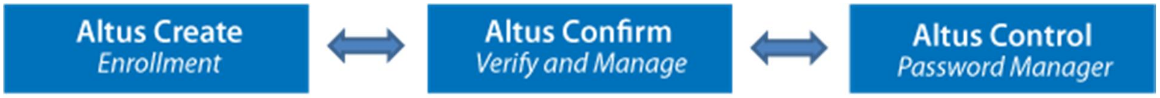
Contents

DigitalPersona Altus Solution

Financial Institutions are looking for ways to reach currently unbanked consumers. Retailers are searching for ways to reduce theft and fraud. Government agencies want to improve service levels and reduce costs. All of these needs require the affordable and positive identification you only get from biometrics. DigitalPersona has developed a secure, affordable and easy-to-deploy solution called DigitalPersona Altus to meet these identity assurance needs.

The DigitalPersona Altus solution provides a modular framework that delivers identity assurance through a strong multi-factor authentication client and server in a Windows platform. The solution leverages fingerprint biometrics, smart cards, Bluetooth and other secure, yet affordable technologies. The DigitalPersona Altus solution enables service providers to create assured identities and subsequently authenticate employees and customers in real-time over the Intranet or VPN.

The solution features a three-part framework that includes Altus Create for enrollment, Altus Confirm for verification and credential management and Altus Control for password management.



MODULAR SOLUTION — CREATE-CONFIRM-CONTROL

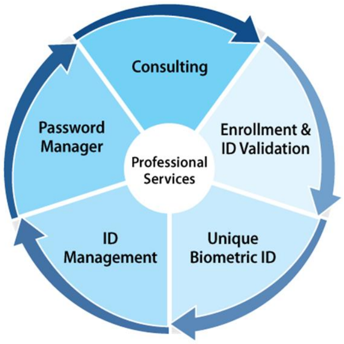
The **Altus Create** module establishes a non-repudiable biometric database for secure enrollment and access to applications. The database contains digital identities that bind key elements, including biometrics, biographic data and breeder documents (such as birth certificates).

Non-repudiation means that the validity of the biometric identities stored cannot be successfully challenged. For example, if the authenticity of a biometric enrollment is being challenged, then the authenticity is being "repudiated". Non-repudiation has the following characteristics:

- A service that provides proof of the integrity and origin of data.
- An authentication that can be asserted to be genuine with high assurance.

The **Altus Confirm** module verifies a user and provides an assured identity. The user’s credentials are managed during the entire life cycle, from enrollment to de-provisioning. When a user wants to access a bank account, government services, corporate networks or any other asset, they simply provide their fingerprint or other enrolled biometric. This biometric is verified against identities stored in the database.

The **Altus Control** module provides access to computers and applications creating a single sign-on (SSO) environment using biometrics. This provides a seamless integration with core applications and prevents unauthorized access. Altus Control also offers audit trails and a variety of management reports.



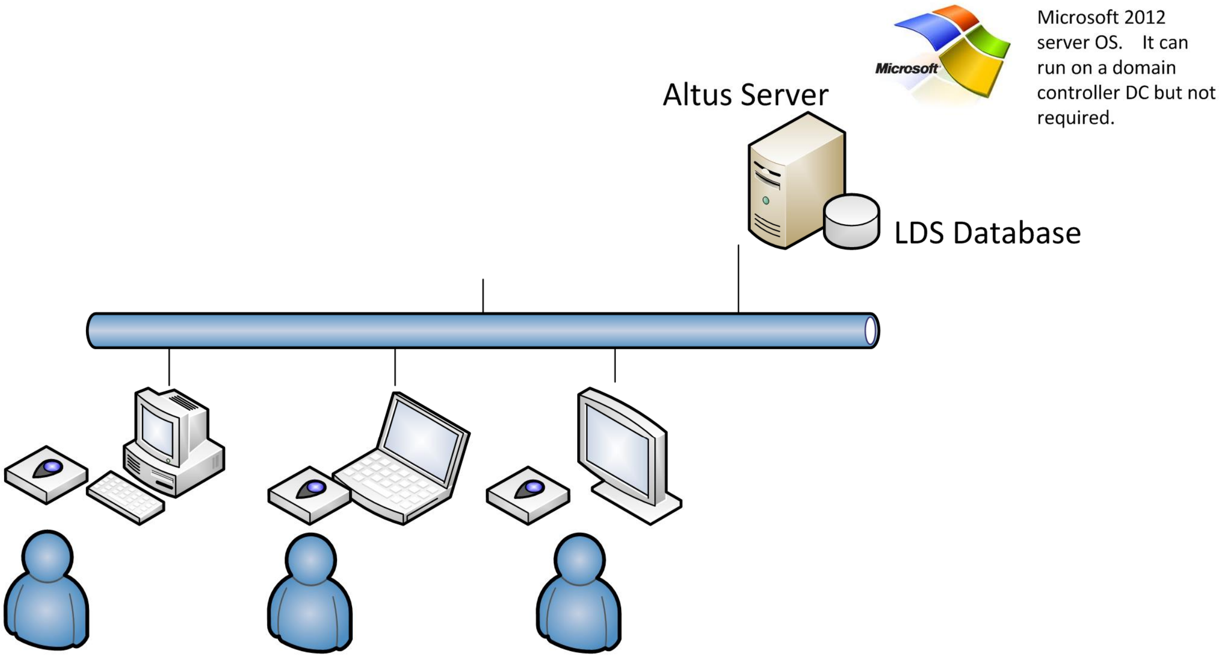
DigitalPersona Altus Solution

EXPERT SERVICES — ASSESS-DESIGN-DEPLOY-SUPPORT

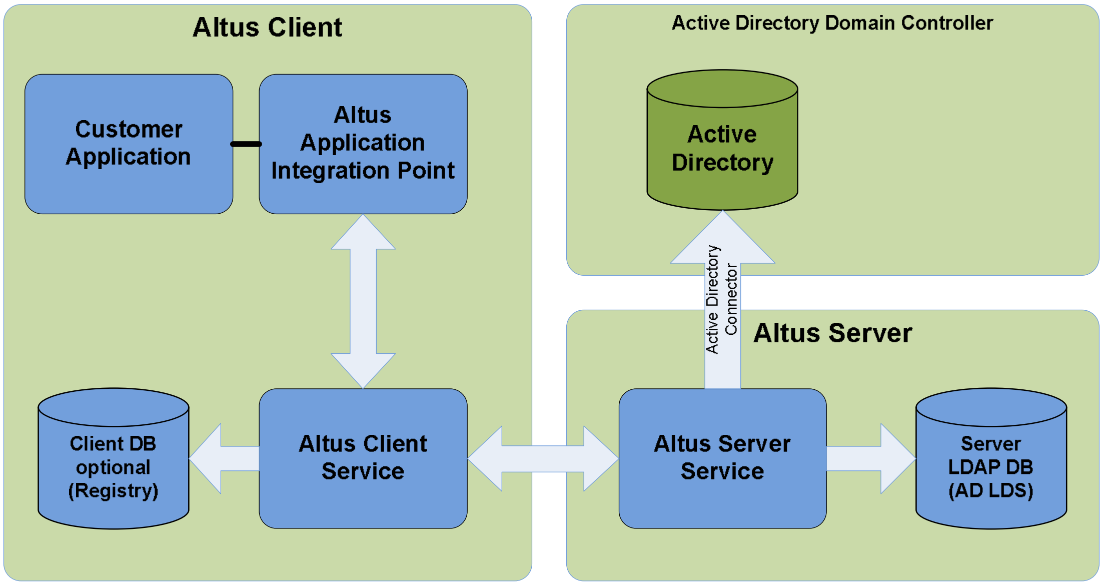
The DigitalPersona Altus solution also includes two critical services. With **Altus Consult**, experienced identity professionals evaluate a customer's security requirements, recommend best practices and configure highly effective solutions that align with the customer's business model. With **Altus Services**, our experts assist with deployment, training and ongoing optimization of the customer's DigitalPersona Altus solution. These services are valuable in the design of the customer's solutions, as well as the integration of best practices for enrollment and creation of non-repudiable identities. Services may include policy definition, customized workflows, software development and hardware options.

- See more at: www.digitalpersona.com/altus

DigitalPersona Altus System Architecture


Microsoft
Microsoft 2012 server OS. It can run on a domain controller DC but not required.
Altus Server
LDS Database

Altus architecture is a complete Windows-based client-server solution platform for Identity Assurance. The Altus Server is based on Microsoft Windows 2012 Server and is responsible for managing the Identities in a single database. Employee Module (Employee with AD account) and Customer Module (Employee without AD account or Customers) Module are supported.


Altus Client
Customer Application
Altus Application Integration Point
Client DB optional (Registry)
Altus Client Service
Active Directory Domain Controller
Active Directory
Active Directory Connector
Altus Server
Altus Server Service
Server LDAP DB (AD LDS)

Altus Architecture Overview

Altus Client Application

Any application which is running on a Client Machine and powered with the Altus Authentication API is considered an Altus Client Application, which uses the rich variety of authentication methods and policies provided by DigitalPersona Altus.

Altus Client Service

The Altus Client Service provides authentication capabilities and enforces authentication policies for the Altus Client Application via the Altus Authentication API

Altus Client Database

The Altus Client Database is used by the Altus Client Service to store and retrieve user sensitive information such as user credentials, policies, secrets, etc. The Altus Client Database is based on the Microsoft Registry and is used mostly for caching purposes, as the primary storage of user information is the Altus Server Database. The Altus Client Server uses the Altus Client Database for user data only when the Altus Server Service is unavailable, for example, due to network issues.

Altus Server Service

The Altus Server Service is used by the Altus Client Service to perform operations such as user authentication, retrieval of user policies and other user data such as email address, etc.

Altus Server Database

The Altus Server Database is used by the Altus Server Service to store/retrieve user specific data. It could be security sensitive data such as user credentials and secrets or public user data such as user name, e-mail, etc. The Altus Server Database is based on Microsoft Active Directory Lightweight Directory Services (AD LDS). There are two types of Altus users: a) **Altus Users** (these are either employee without an AD account or general users (e.g. bank customers) or b) **Altus AD Users** (typically employee).

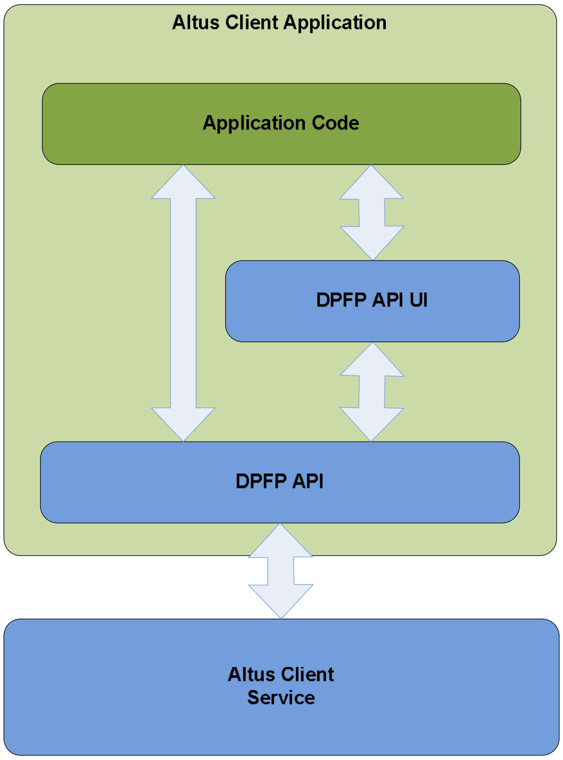
Altus Users are typically users that are also in another database (e.g. core banking software database) and Altus not necessarily need to include personal information about these users.

Altus AD Users are users from Active Directory Domains and DigitalPersona is using the Altus Server Database to store only security sensitive information for such users like user credentials and secrets. All public information about them, such as user names, email addresses, phone numbers, etc. are stored in Active Directory and the Altus Server Service uses the Active Directory Connector to retrieve the information.

Active Directory

This is a standard Microsoft Active Directory forest where information about Active Directory users is stored.

Altus Client Application Architecture



Altus Client Application Architecture

Application code implements application-specific business logic and uses the Altus API (DPFP API) only when user authentication is required. The Altus API allows any application to authenticate the user via the multi-factor authentication provided by the Altus framework, and to receive some application specific secret (if necessary). An application secret can be a user password, an encryption key or any other user-specific security-sensitive data.

The main purpose of the Altus Client APIs is user authentication and retrieving application specific secrets. But it also provides a set of auxiliary functions such as querying a user, retrieving user public data, authentication policies, saving application secrets, etc.

The Altus framework provides two APIs for access to Altus features:

- DPFP API UI and
- DPFP API.

The application can choose to use either one or both APIs.

DPFP API UI

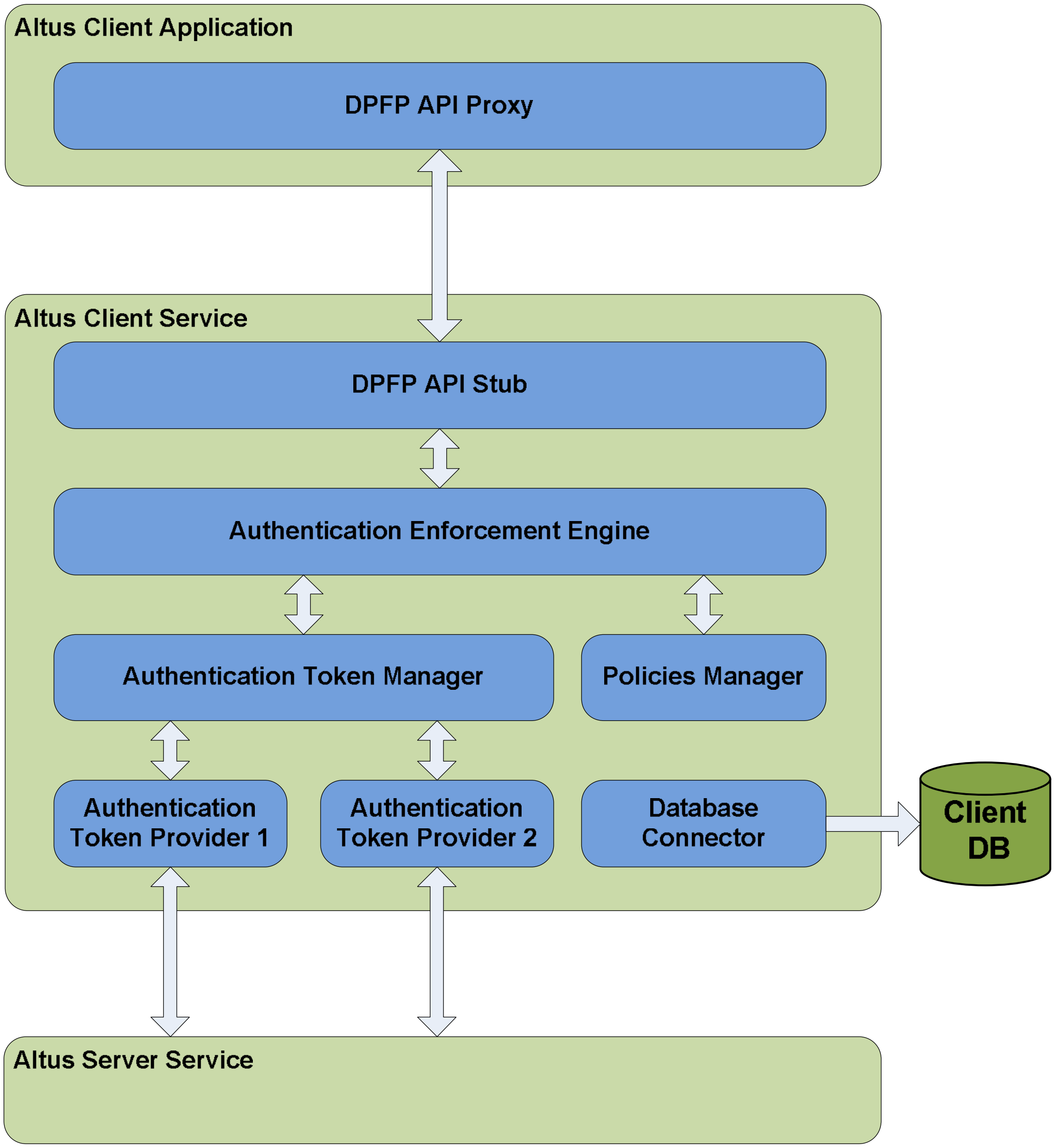
DPFP API UI is a high level API which provides an elaborate GUI for user authentication. It does not perform the actual authentication, but internally uses DPFP API for authentication.

DPFP API

DPFP API is a low level GUI-less API for user authentication. It does not perform the actual authentication, but is just a proxy which redirects all requests using Local RPC (LRPC) to the Altus Client Service.

These API are used by the Professional Services Team to implement dedicated services to enable application to use the Altus Identity Assurance capabilities. As an example, a core banking software could integrate fingerprint authentication directly within the application workflows.

Altus Client Service Architecture



Altus Client Service Architecture

There are a number of components in the Altus Client Service, which are described in detail on the following pages.

DPFP API Stub

DPFP API Stub is responsible for receiving requests from the Altus Client Application.

DPFP API Stub is implemented as an RPC Service listening on the Local RPC (LRPC) protocol. The RPC service is configured to use privacy and integrity on the communication channel to ensure that transmitted data cannot be exposed for middle-man and cannot be altered.

After DPFP API Stub receives an authentication request from the Altus Client Application, it redirects the requests to the Authentication Enforcement Engine.

Authentication Enforcement Engine

The goal of the Authentication Enforcement Engine (AEE) is to perform user authentication, ensure user authentication satisfies the authentication policy and then to release application-specific secrets if necessary.

AEE does not perform user authentication itself, instead it redirects any authentication request to the Authentication Token Manager (see description below).

AEE uses the Policies Manager to retrieve user authentication polices.

Policies Manager

The goal of the Policies Manager component is to collect authentication policies from different sources, create the authentication RSoP (Resultant Set of Policy) report and provide it to the requesting party.

NOTE: The only available source for authentication policies in the current version of Altus is the GPO.

Authentication Token Manager

A core feature of Altus is a pluggable architecture for Authentication Token Providers (see description below). The goal of the Authentication Token Manager is to enumerate the Authentication Token Providers available on the system and load them into the Altus Client Service.

When the Authentication Token Manager receives an authentication request from the Authentication Enforcement Engine, the request is redirected to the corresponding Authentication Token Provider.

Authentication Token Providers

Every authentication method (credential), i.e. fingerprint, smart card, PIN, etc., needs to implement its own Authentication Token Provider.

The purpose of an Authentication Token Provider is to perform the enrollment and authentication of a specific credential (fingerprint for example).

The Authentication Token Manager (described above) will discover all installed Authentication Token Providers and load them into the Altus Client Service.

The maximum number of authentication methods/credentials, i.e. Authentication Token Providers is 64, the number of bits in the policy item. The pluggable architecture allows DigitalPersona (and also third party developers) to add new authentication methods (credentials) to the Altus framework.

The Authentication Token Provider does not necessarily perform authentication itself, but may direct the authentication call to the Altus Server Service for authentication on the server.

The usual workflow is that the Authentication Token Provider directs an authentication request to the Altus Server Service, and if the Server is not reachable, it gets the necessary information from the Client DB using the Database Connector (see description below) and attempts to perform authentication locally.

Database Connector

The Database Connector is an auxiliary component which is used by all other Altus Client Service components to access (read/write) data stored in the Client DB.

Client Database

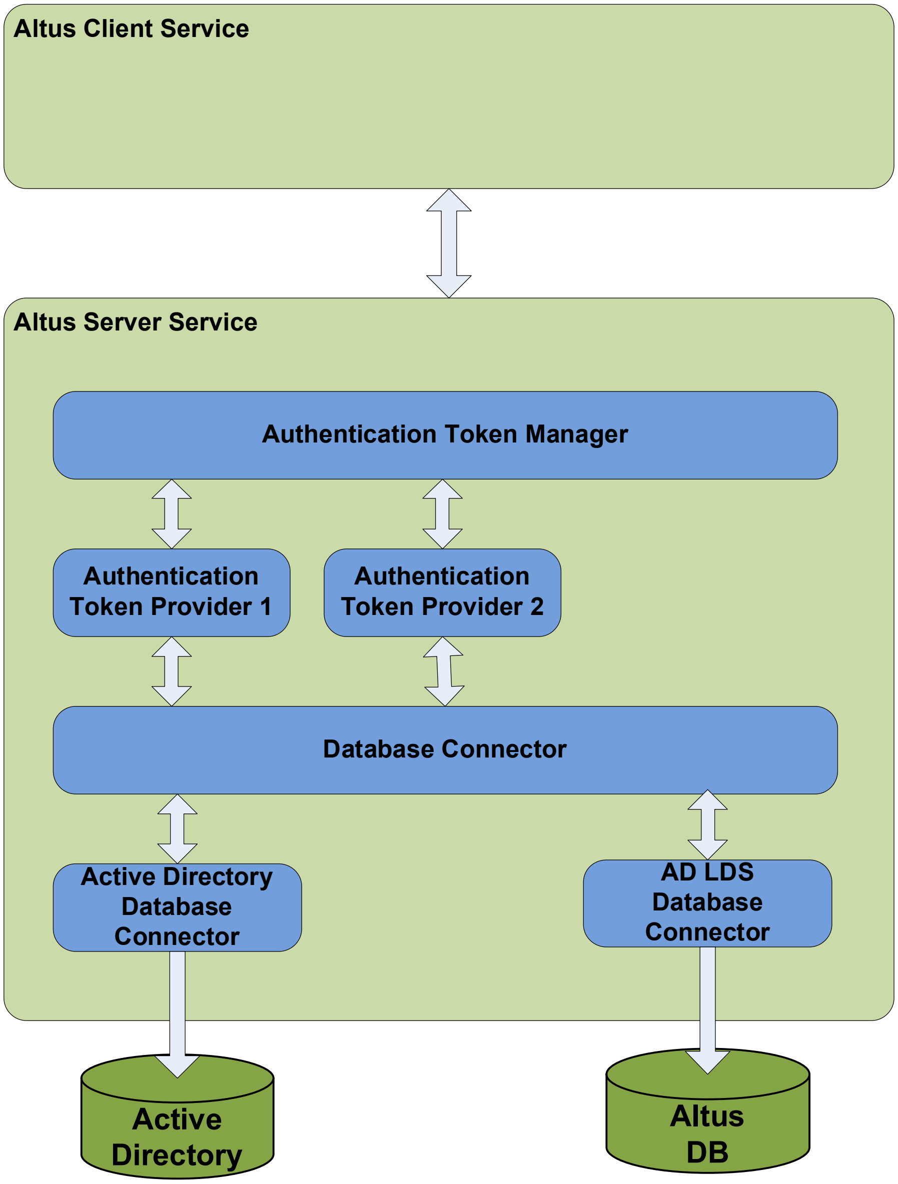
The Client Database is a database based on the Windows registry and used by the Altus Client Service to store information about users, such as their user credentials, relevant policies, public information, etc.

As mentioned above, the Client Database is mostly used for caching purposes in case the Altus Server Service is not accessible.

The user data in the Client database is encrypted for security purposes using an Altus Client Service 2,048-bit RSA encryption key. The only allowed access to the Client Database is by the DigitalPersona Altus Client Service.

Altus Server Service Architecture

The following diagram illustrates the architecture of the Altus Server Service, whose components are described in detail on the following pages.


Altus Client Service
Altus Server Service
Authentication Token Manager
Authentication Token Provider 1
Authentication Token Provider 2
Database Connector
Active Directory Database Connector
AD LDS Database Connector
Active Directory
Altus DB

Altus Server Service Architecture

Authentication Token Manager

The Server implementation of the Authentication Token Manager is generally the same as the Client implementation. The main goal of the Authentication Token Manager is to enumerate the Authentication Token Providers installed on the Altus Server and load them into the Altus Server Service.

The Authentication Token Manager redirects all authentication requests to the appropriate Authentication Token Provider.

Authentication Token Providers

The goal of the Authentication Token Provider is to perform the enrollment and authentication of a specific credential (fingerprint for example) on the Altus Server. Every credential (authentication method) must implement its own Authentication Token Provider.

The pluggable architecture for Authentication Token Providers allows DigitalPersona to easily implement and add new authentication methods to the Altus Server.

The Authentication Token Provider uses the Database Connector to store and retrieve necessary data to/from the Altus Database.

Database Connector

The Database Connector allows other components (mostly the Authentication Token Providers) to store and retrieve data in the Altus Database.

There are two types of users in the Altus Framework: Altus Users and Altus AD Users.

- Altus Users - The Database Connector uses the AD LDS Database Connector to store/retrieve data for Altus Users.
- Altus AD Users - The Database Connector uses the AD LDS Database connector to store/retrieve Altus-specific information about Altus AD Users, and the Active Directory Database Connector to retrieve general user information.

AD LDS Database Connector

The AD LDS Database Connector is an auxiliary component that allows other Altus Server components to communicate with the Altus Server Database.

Altus uses the Active Directory Service Interfaces (ADSI) to communicate with AD LDS.

Active Directory Connector

The Active Directory Database Connector is an auxiliary component that allows other Altus Server components to communicate with Microsoft Active Directory.

Altus uses the Active Directory Service Interfaces (ADSI) to communicate with Active Directory.

The Active Directory Connector can only retrieve data from Active Directory and cannot write to it.

Altus Server Database

The Altus Server Database is a database where all Altus user-specific information, such asuser credentials, encryption keys, secrets and public information is stored. The Altus Server Database is implemented based on Microsoft’s AD LDS.

User security-sensitive information, such as user credentials, secrets, encryption keys, etc., are stored encrypted with Altus Server Service’s 2,048-bit RSA encryption key. Only the Altus Server Service has access to this information.

Altus Users - All information about Altus Users is stored in the Altus Server Database.

Altus AD Users - Only Altus- specific information such as user credentials, secrets, etc., for Altus AD Users is stored in the Altus Server Database, all generic information is retrieved from Active Directory.

The AD LDS schema is extended to support Altus-specific data in a user record.

Altus also includes a flexible design which supports extension of the AD LDS schema in a customer environment to support customer needs. For example suppose a customer requires storing a user's License ID information during user enrollment. The AD LDS schema can simply be extended to support a new attribute (License-Id for example) for the user object and the GUI can be easily modified to handle it.

The Standard DigitalPersona schema extension for AD LDS is detailed in the TFS document "Altus AD LDS Schema Extension."

Active Directory

Active Directory is a Microsoft Active Directory Forest where all information about AD users is stored.

NOTE: Altus Server has no rights to write to AD and can only read information from it.

Policies

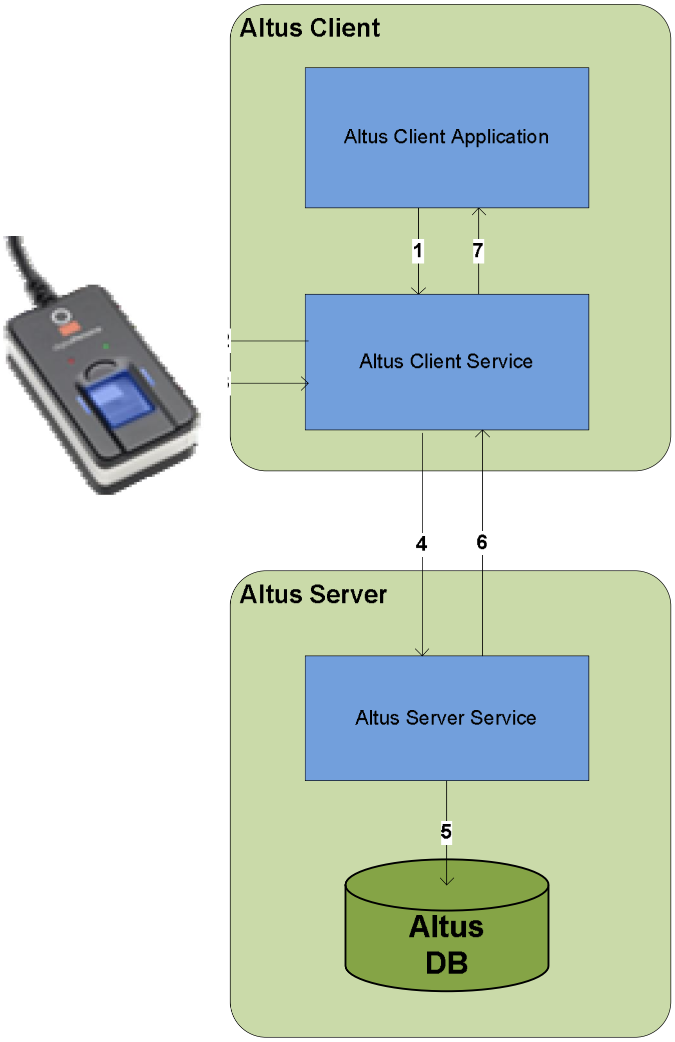
As mentioned above, DigitalPersona uses Active Directory Group Policy Objects (GPOs) to handle Altus 1.0 policies. Altus Policies can be set using the standard GPO Editor. Any Domain Administrator or Delegated Administrator can set Altus policies.

Security

Details of the Altus security architecture is provided in the ***Altus 1.0 Security Guide***.

Authentication Data Flow

The following illustration shows an example of Authentication Data flow using Fingerprint Authentication.



Authentication Data Flow

1. The Altus Client Application requests fingerprint authentication. The request is directed to the Altus Client Service. The Altus Client Application uses DPFP API UI to display a GUI prompting the user to touch/swipe the fingerprint device.

2. The Altus Client Service establishes a connection with the fingerprint device(s) attached to the PC and waits for feedback.
3. When a user touches/swipes the fingerprint reader, it sends a fingerprint image back to the Altus Client Service, which processes the fingerprint image and creates the fingerprint feature set.
4. The Altus Client Service sends the created fingerprint feature set to the Altus Server Service for fingerprint matching.
5. The Altus Server Service receives the fingerprint authentication request from the Altus Client Service. It retrieves the user fingerprint templates stored in Altus Database during enrollment.

6. The Altus Server Service compares the fingerprint feature set received from the Altus Client Service with the fingerprint templates stored in the Altus Database and returns the match result back to the Altus Client Service.
7. The Altus Client Service returns the match result received from the Altus Server Service to the Altus Client Application.

Altus Servers Deployment and Publication

Two Altus components must be deployed for a successful Altus Server deployment: 1) Microsoft AD LDS; and 2) the Altus Server.

AD LDS Deployment and Maintenance

AD LDS is part of Microsoft Windows Server 2012. For an overview on AD LDS, see http://msdn.microsoft.com/en-us/library/windows/desktop/aa705884(v=vs.85).aspx.

A customer can deploy as many Altus Servers as they want in an environment. However, it is important to first configure the AD LDS database on any unique instance of Altus Server before configuring replication between it and any additional Altus Servers.

The customer should backup their AD LDS database(s) regularly. For instructions on backing up and restoring an AD LDS database, see

http://msdn.microsoft.com/en-us/library/windows/desktop/aa705895(v=vs.85).aspx.

Altus Server Installation and Maintenance

As mentioned above, a customer can deploy as many Altus Servers as they want in their environment.

Every instance of Altus Server provides identical service. This means it doesn’t matter with which particular Altus Server the Altus Client will communicate. The Altus Database should be replicated between instances based on AD LDS replication as described above.

Each Altus Server registers itself in Active Directory, creating a Service Connection Point object. It also registers its own DNS address, the Site it belongs to, etc.

The customer does not need any specific maintenance for Altus Server except the AD LDS database backup described above.

Altus Server Discovery

When Altus Client Service starts, it sends a search request to the Active Directory Global Catalog (GC) to find all Altus Servers in the Site it belongs to. If it doesn’t find an Altus Server in the same Site, it sends a request to find all Altus Servers in the same Active Directory Forest.

When the Altus Client Service receives the list of available Altus Servers, it chooses one randomly to make sure the load is balanced equally among all available Altus Servers.

For failover, if the Altus Server the Altus Client connects to is down, the Altus Client immediately re-starts Server discovery and randomly connects to another Altus Server so that the end user will always have a live connection with the Altus Server.