passlogix®

white paper

v-go®

v-GO Sign-On Platform Technical Architecture

This paper provides IT professionals with a guide to the v-GO Sign-On architecture used in the v-GO suite of authentication and password management solutions.

Its main purpose is to give IT management and staff an overview of the overall functionality of the v-GO family of products, and how they interact with other access control systems and each other.

A Sign-On Platform Defined

By Passlogix's use of the term "platform," we mean that the products provide not only a set of base functionality, but also serve as a foundation upon which a complete enterprise identity management infrastructure can be either integrated or built, incorporating complementary technologies such as authentication management and user provisioning.

The fundamental notion underlying a sign-on platform is that authentication and other enterprise infrastructure can benefit from a reliable set of basic sign-on services and functionality. Combining sign-on services with other infrastructure components and services can provide powerful new applications or custom solutions, such as:

- Seamlessly bridging strong authentication to application logon.
- Creating a closed-loop credential provisioning cycle.
- Integrating internal v-GO event reporting into security forensic systems.
- Integrating knowledge-based authentication with corporate data repositories.

Platform Architecture

The v-GO Sign-On Platform is comprised of six products, all-centered on the core component, v-GO Single Sign-On (v-GO SSO). v-GO SSO provides the core enterprise with single sign-on functionality – automatically logging users on to applications.

Additional components expand functionality to allow users locked out of their Windows account to gain access, manage kiosk sessions when there is no user login to Windows, add support for all types of strong authentication devices and provisioning systems, and automate the provisioning of smart cards and other strong authentication devices.

The products are built using standard Microsoft development tools and run in a Microsoft-centric environment with Windows-based clients and Windows servers. Wherever possible, the products rely upon existing external infrastructure, such as directory services, which may themselves run on Windows, Linux, or other operating system. The products are designed to interoperate with virtually all applications, e.g. SAP R/3, which again, may be running on Windows or non-Windows operating systems.
Standards are used wherever possible, such as for cryptographic services (3DES or AES), directory services (LDAP), smart card interfaces (MS-CAPI & PKCS #11), provisioning integration (SPML), and configuration data (XML).

The platform separates the data processing functions from storage and administration functions. Almost all data processing occurs at the client – logon recognition, encryption, key management, provisioning instruction processing, etc. The client component is referred to as the "v-GO Agent."

Windows and LDAP user authentication are leveraged by the v-GO Agent to eliminate the need for v-GO to directly authenticate users. v-GO utilizes the MS CAPI interfaces that are on every workstation for encrypting all data as well as for smart card authentication.

User data – user IDs and passwords -- are stored in a centralized repository, typically a directory. The v-GO Platform uses multiple central storage repositories for storing all data created managed and accessed. These repositories include Active Directory, generic LDAP directories, SQL databases, and a file share system.

Using the directory or database infrastructure already in place reduces implementation effort and cost. . The task of developing a failover and recovery plan for is minimized, because it rides on the plans already in place for the network. Of course, v-GO can also be configured to use a dedicated directory or database.

User settings and policies – e.g. password policies -- are also stored in the centralized repository. All client configuration settings retrieved from the directory are written to the registry under HKLM and HKCU, while client behaviors and policies can be maintained through the repository in the form of group policies.

The platform is administered via several consoles that read and write data to the central repository. Each time the v-GO Agent synchronizes its data with the directory, it retrieves the latest user data and administrative settings and policies. Each component of the platform adds additional components to the v-GO Agent as well as settings in the directory. In addition, three of the products have server components that perform limited special purpose processing.

All audit data is generated by the v-GO Agent. Audit data is configured to be written to either an MS event viewer table or an XML file to ensure that it can be easily accessed and compiled. To assist with the ease of deployment, the administrative console can develop a customized msi package that can be pushed to user desktops via the existing mechanism in the network.

Major Functions

The v-GO Sign On Platform handles all the day-to-day tasks associated with granting users access to the applications, including sign on, application password change, session management, Windows account password reset, strong authentication, application password provisioning, and strong authentication provisioning.

A. Application sign-on

The v-GO Agent runs in the system tray of each client waiting for logon prompts. When it detects a prompt, whether from a Windows, Web, mainframe, or telnet application, it responds with the correct username and password ("credentials") on behalf of the user.

B. Application password change

When applications request that the password be changed, the v-GO Agent detects this change request and responds automatically. It either automatically generates a new password or prompts the user to pick one, and submits that new password to the application on behalf of the user. v-GO verifies that any new password conforms to the applicable password policy.

C. Windows password reset

For end-users that lock themselves out of their Windows domain account, v-GO permits end-users to pick a new Windows password or unlock their locked Windows account in order to gain access to the network. The end-users must successfully complete a challenge/response sequence (correctly answer a series of questions) to gain access to their Windows account. The reset function can be accessed from either the locked out computer via a chained GINA extension or another computer with browser-based access to the network.

D. Session management and application termination

Computers are sometimes deployed to function as kiosks that can be shared among many users, typically in hospitals, manufacturing plants, warehouses, or other "stand up" environments. In those situations, the kiosk is typically logged in with a generic account ID. A new user logs in on top of that session to establish his or her unique identity. When a user leaves the kiosk, any open applications must be automatically terminated according to specified procedures. v-GO performs these functions.

E. Strong authentication

v-GO accepts user authentication by ID and password, smart card, one-time password token, biometric device, and proximity card. It permits the end-user to alternate between these authenticators on the fly, and to fall back to an ID and password if the smart card or token is lost. It also permits the administrator to restrict access to specific applications to only those users who have logged in using a specific authenticator, e.g. smart card.

F. Application password provisioning

v-GO can accept provisioning instructions to add, modify, or delete credentials from an end-user, eliminating the need for the end-user to manually enter the data. These instructions can originate from a commercial vendor's provisioning system, a custom developed provisioning system, a custom program, or interactive user input via a web-based console.

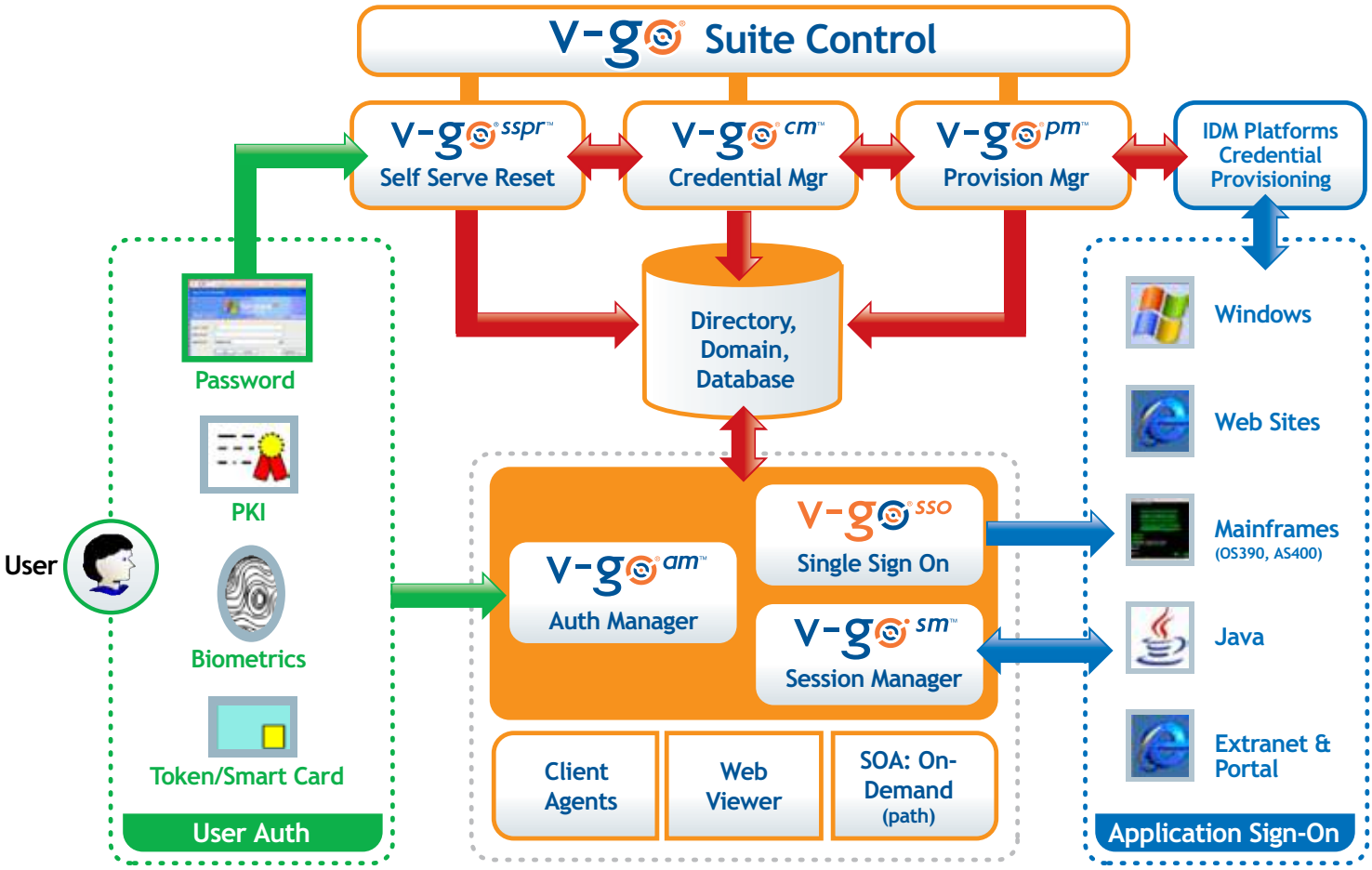
G. Strong authentication provisioning

Also referred to as "credential management", v-GO can automate the provisioning and administration of smart cards and other strong authentication devices. For smart cards, v-GO can initialize the card, interact with the corporate PKI to place down the certificates on the card, personalize the card for the end-user and track the inventory of cards issued, valid, invalid and revoked. These functions can be performed by the administrator or the end-user via a web-based interface.

Major Components

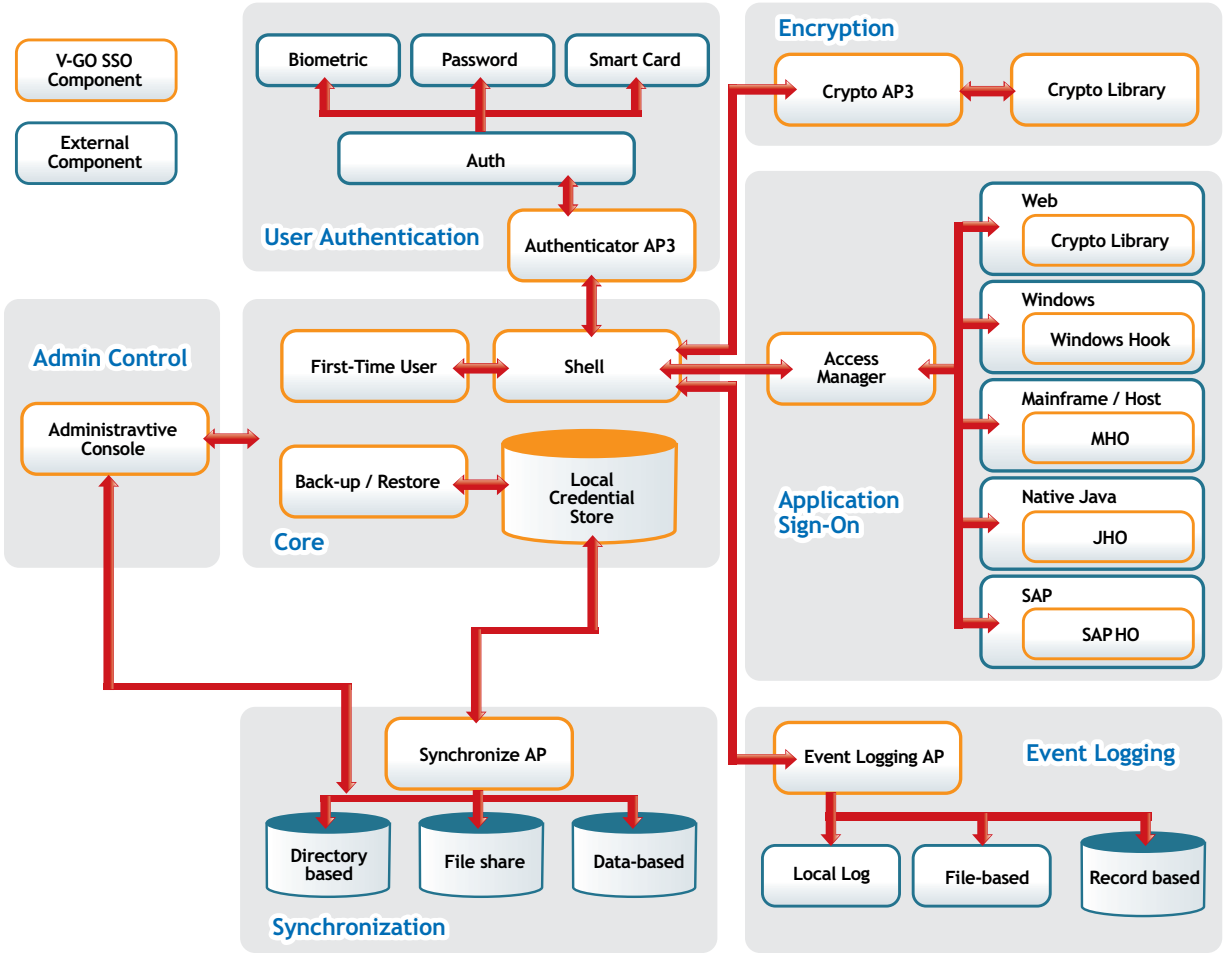
The major platform components and their functions are:

— v-GO Single Sign-On — single sign-on.
— v-GO Self-Service Password Reset — Windows account unlock.
— v-GO Session Manager — session management for kiosks.
— v-GO Authentication Manager — strong authentication services.
— v-GO Provisioning Manager — interoperability with provisioning systems.
— v-GO Credential Manager — strong authentication provisioning.



v-GO Single Sign On

v-GO SSO's architecture consists of seven areas: (1) Authentication; (2) Encryption; (3) Intelligent Client Response; (4) Core (including Storage); (5) Credential Synchronization; (6) Event Logging; and (7) Miscellaneous components.



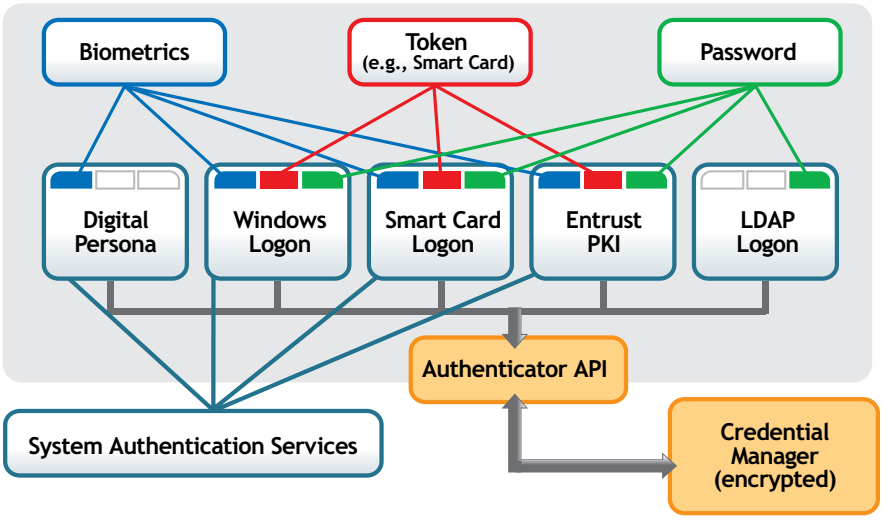
User Authentication

User Authentication is how the system validates users so they gain access to v-GO SSO. It consists of three layers: (1) the authenticator itself; (2) the authentication service; and (3) the v-GO SSO Authenticator API. Once the system validates the user, it passes the user's validation information to the v-GO Shell.

v-GO SSO ships with several authenticators and several authentication services that support a variety of authenticators. v-GO SSO Authenticator API supports any authenticator and authentication service to provide access to user credentials.

Authentication to v-GO SSO involves three steps: (1) the user provides credentials to the authenticator; (2) the authenticator validates the user with the authentication service; and (3) the authentication service passes to the v-GO SSO Authenticator API information confirming validation and unlocking the user's encryption keys.

An authenticator allows users to prove their identity, whether through a password, biometric, or token (for example, smart card). The authenticator takes the user's proof and passes it to the authentication service. v-GO SSO ships with a set of authenticators, including Windows authentication, Smartcard, LDAP, RSA and Entrust.



The authentication service validates the credentials provided by the authenticator against either its own store, or a system authentication Service such as a Windows domain or a PKI. If validated, it passes the validation to the Authenticator API.

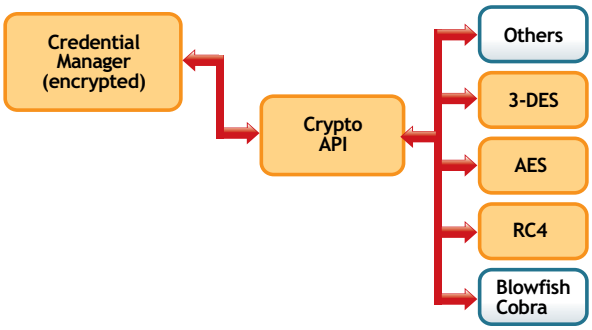
An authentication service can support "disconnected" mode if it meets the requirements of the v-GO SSO Authenticator API. This allows users to access their credentials even when the system authentication services are not available.

v-GO SSO ships with six authentication services: Windows (Domain) Smart Card Logon (pass phrase and certificate based), LDAP, Entrust PKI, and enhanced versions of the Windows and LDAP authenticators that support pass phrase challenge.

The v-GO SSO Authenticator API is a set of plug-in interfaces used to integrate the authentication user interface with the main v-GO SSO client. It serves as a conduit between the authentication service and v-GO SSO. Third-party authentication services can integrate with v-GO SSO by utilizing the Authenticator API. For more information on the Authenticator API, contact Passlogix.

Encryption

v-GO SSO supports virtually any symmetric algorithm for encrypting user credentials. This enables companies to meet their security/audit requirements or comply with local government regulations. By default, v-GO SSO uses the Triple-DES symmetric key encryption algorithm supplied by MS CAPI that is FIPS 140-2 certified.



v-GO SSO uses the encryption algorithm to secure all user credentials locally on the desktop and to remote directories or network drives. v-GO SSO also includes AES, RC4 Blowfish and Cobra as administratively selectable algorithms. Additionally, v-GO SSO also includes non MS-CAPI based version of Triple-DES and AES for backward compatibility with previous versions. The v-GO SSO Encryption API enables substitution of practically any other symmetric encryption algorithm.

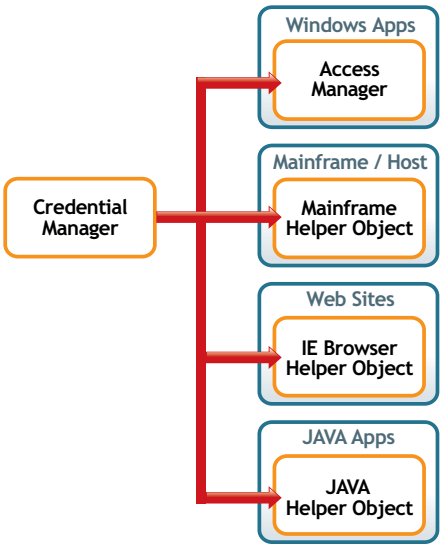
v-GO SSO uses cryptography to confirm user authentication and to secure storage of user credential data. Upon first-time use, v-GO SSO generates and maintains a cryptographically unique "primary authentication key." The primary authentication key is Authenticator-independent and requires successful completion of the authentication process before it can be used. Upon successful authentication, this key becomes available internally to v-GO, where it is used to decrypt and access user credentials. Each credential is only decrypted on an as-needed basis and is never stored or cached in the clear.

For Random Number Generation, such as for generating the "primary authentication key" or for generating new password for applications based on password policies, v-GO SSO uses Microsoft CAPI, the Intel Hardware RNG, and RSA CSPs.

Intelligent Client Response

When an application presents a request for credentials, v-GO SSO intelligently and securely detects this event, determines the appropriate action, and responds with the correct credentials. v-GO SSO detects requests for credentials in a variety of ways, depending on application type (Web, Windows, Mainframe/Host). Details for each application type appear later in this section. v-GO SSO determines whether the event is a logon or password change.

If it is a logon request and it has the required information, v-GO SSO Access Manager retrieves the credentials from the Local Credential Store and submits them to the application in the most effective and secure way possible for that application. If v-GO does not have the user credentials, it prompts the user for those credentials. If the user provides credentials to v-GO SSO, v-GO Shell stores the credentials in the Local Credential Storage and v-GO Access Manager submits them to the application.



For a password change, v-GO SSO can be configured to either generate a new password or prompt the user to enter a new password. In either case, v-GO SSO then submits, in the most secure way possible for that application, the old password (if required), new password, and new password again (if required). If the password change is not successful, v-GO SSO will attempt the process again until it is successful.

v-GO SSO has separate internal components, one each to respond to Windows applications, mainframe applications, non-Java web-based applications and Java applications.

Windows Applications

All credential requests in Windows have specific attributes including application name, window name, and the control ID of the input field. v-GO SSO looks for the specific attributes of each application's logon and password-change dialogs and responds accordingly. The attributes are stored in the basic **applist.ini** and administrative **entlist.ini** configuration files. See the v-GO SSO Administrative Console help system for additional information, including deployment.

The v-GO hook (**vgohook.dll**) component captures standard, OS-level Windows messages and sends them to the v-GO Shell and Access Manager components. When a specified application creates a dialog, v-GO SSO looks at the window title. If v-GO SSO recognizes the window title, it searches for the appropriate control ID(s).

v-GO SSO submits credentials to most Windows applications via secure, standard, OS-level Windows messages. Thus, keyboard-sniffing utilities cannot intercept the credentials. Furthermore, since v-GO SSO does not use scripts or keystrokes, users cannot confuse the response by selecting and working in another application.

Mainframe/Host Applications

All requests for credentials in Mainframe/Host/telnet applications have specific attributes such as window title and various blocks of text (at specific row-column coordinates for block screen 3270/5250-based mainframe applications) and username/password field text. v-GO SSO looks for and matches on a specific set of text strings on each application's logon and password-change screens and responds accordingly. The attributes are stored in the administrative **entlist.ini** configuration file. See the v-GO SSO Administrative Console help system for additional information, including deployment.

The v-GO SSO Mainframe Helper Object monitors emulators, looking for the defined matches. When a new screen is presented, v-GO SSO reviews the text for matching fields. If all strings match, v-GO SSO uses the Mainframe Helper Object to submit user credentials. v-GO SSO submits credentials to most emulators via HLLAPI. By avoiding sending keystrokes, keyboard-sniffing utilities cannot intercept these credentials.

Because v-GO SSO does not use scripts or keystrokes for these emulators, users cannot confuse the response by selecting and working in another application. v-GO SSO also supports non-HLLAPI-capable emulators that have a scripting language capable of presenting a visible or hidden pop-up Windows dialog box, responding to them as it would a Windows application. For more information, contact Passlogix.

Web Applications

All credential requests in Web applications are either in web forms or pop-up dialog boxes. The v-GO SSO Browser Helper Object (BHO) and Event Manager respond to the specific events of a web dialog popping up or of a web page rendering. Internet Explorer embedded within Lotus Notes and Firefox are also supported. Since v-GO SSO does not use scripts or keystrokes for IE, users cannot confuse the response by selecting and working in another application.

Pop-up dialogs have specific attributes and fields, e.g., realm, site. v-GO identifies and matches on these specific attributes for each application's logon and password-change requests, and responds accordingly. The attributes are stored in the basic applist.ini and administrative **entlist.ini** files. When a new pop-up dialog is created, v-GO reviews the dialog, requests credentials from v-GO Shell, then submits them to the pop-up dialog.

Forms have specific attributes including URL (including domain), frame name, form name, specific blocks of text on the page, username/password field text, and password fields (HTML <Input type=password>). v-GO SSO looks for the specific attributes of each application's logon and password-change screens and responds accordingly.

The attributes are stored in the basic **applist.ini** and administrative **entlist.ini** files. When a new page is fully rendered, the BHO reviews the page for matching criteria. If at least a password field is present, the BHO requests credentials from v-GO Shell, then injects them into the appropriate fields in the form.

v-GO SSO responds to login and password change requests for virtually all AWT and Swing Java Applications and Applets built on the Sun Java Runtime Engine 1.1.8 or higher. v-GO SSO's Java Helper Object (JHO) hooks into the Java Runtime via standard runtime interfaces. The JHO looks for and responds to logon and password change events in the same manner as described above.

Core (including Storage)

The Core connects all of v-GO SSO. It consists of the First-Time Use, v-GO Shell, and Local Credential Storage components.

The First-Time Use component generates creation of user-specific keys and can be set to prompt the user to bulk-load passwords.

v-GO Shell receives user validation from the Authenticator API. It encrypts and decrypts data from Local Credential Storage through the Encryption components. It can then supply the credentials to the Intelligent Client Response components, notify the Event Logging API, and trigger Credential Synchronization as needed.

Local Credential Storage is a permission-protected folder containing a set of encrypted files with the user's credentials. v-GO SSO never stores unencrypted credentials on disk or in memory. v-GO SSO stores credentials locally, encrypted specifically for each user, in individual files. These files are located in a specific directory within the application data directory of the user profile.

Within these files are the encrypted records for each set of user credentials, user settings, and additional configuration information. The file can be secured from other users by properly configuring Windows security on NTFS partitions. If Windows "Roaming Profiles" are enabled, users can log on to Windows from any computer within a domain to access their credential file.

For example, the Windows 2000 variable for the user profile directory is %**UserProfile**%, which defaults to **C:\Documents and Settings**\%**UserName**%. The directory secrets is located in %**UserProfile**%**Application Data\Passlogix** so the path for user "user1" might be **C:\Documents and Settings\user1\Application Data\Passlogix\secrets**.

Because the credential file is stored locally, users can access their applications even when there's a disruption in the centralized server infrastructure. Storing files locally also speeds up access to credentials by minimizing server latency.

Credential Synchronization and Client Administration

While v-GO SSO stores user credentials and settings locally, it can synchronize the credentials and settings with remote network shares, directories, and devices. Synchronization can be of the entire credential file; v-GO SSO uses the newer of the local and remote files, and overwrites the older ones, via silent backup and restore functionality.

Or, v-GO can synchronize by individual credential record within the credential file using the newer of the local and remote record for each record and overwriting the older ones. The synchronization is triggered by a change to the Local Credential Storage or settings.

v-GO SSO supports multiple directory services including Sun Directory, Novell eDirectory, IBM Directory, Oracle Directory 9i and 10g, Microsoft Active Directory and ADAM, Oracle 9i and 10g, Microsoft SQLServer, and IBM DB2. In addition, v-GO SSO supports a record level File System Sync, Windows "Roaming Profiles," file-level synchronization and backup.

v-GO SSO also provides a standard API for record-level synchronization of user credentials with any external application or device. Synchronization can be extended to any storage mechanism via the Synchronization API.

The synchronization is based on record date/time and other authentication information. During synchronization, v-GO SSO takes credentials from the local credential file and credentials from remote storage, and merges them by date and time.

If a set of credentials exists in one place but not the other, v-GO SSO copies those credentials to the location where they are missing, either locally or remote storage. When a user deletes a set of credentials, v-GO SSO places that credential set's unique identification (UID) in the UID list.

Stored remotely, the UID list contains all deleted credential sets, each one containing a unique identifier. v-GO SSO overwrites the older of the two local and remote records, replaces those files with the newer ones, and then uses the local credentials.

The administrator can configure this synchronization process to occur as often as every credential change. This record-level synchronization allows users to log on from multiple computers simultaneously. v-GO SSO determines and uses the latest record for each set of credentials.

Client administration is fully supported via the same Synchronization component. The administrator dynamically delivers updated settings and configuration data to the v-GO client through the central storage mechanism.

The administrator can administer all functions of v-GO SSO from a central directory, database, or file server. Upon startup, v-GO SSO retrieves the latest copies of the first-time use settings, application configurations, and administrative override settings, overwriting older versions. See the *v-GO SSO Administration Guide* for more information on these files including a complete list of their variables and values.

Event Logging

When notified by v-GO Shell, v-GO SSO can log all SSO system events, including credential use, credential changes, global credential events, v-GO SSO events, and v-GO SSO feature use. v-GO SSO can also log specified application fields.

Events can be logged locally or to any external destination through the Event Logging API. These destinations can include an SNMP service, a Windows server (for viewing via the Windows Event log), or even a local XML log file for simplified parsing and reporting.

v-GO SSO can report events locally and remotely. Specifically, v-GO SSO can log:

- Credential use events: logons, manual password changes, automatic password changes.
- Credential changes: add credentials, delete credentials, change credentials, copy credentials.
- Global credential events: backup, restore, synchronize.
- v-GO SSO events: startup and shutdown.
- v-GO SSO feature use: Logon Manager, Settings, Help, About.
- Administrator-specified fields: Domain, Windows username, system username, application name, application username, application third and forth fields, date, time.
- Events to any desired destination: Local XML storage, SNMP service, Windows Event log, directory server, Tivoli.

Miscellaneous Components

v-GO SSO also contains the following miscellaneous modules:

- For users who do not perform any Credential Synchronization, the Backup/ Restore component enables archiving and restoration of user credentials.
- For environments that require usage of v-GO within a Citrix or Terminal Services environment, additional components are supplied to allow v-GO to interact appropriately within each session.
- v-GO ships within a Windows Installer package that supports the flexibility of that technology for easier deployment and customization.

v-GO Self Service Password Reset

v-GO Self-Service Password Reset (SSPR) provides a mechanism for users to reset their Windows passwords or unlock their Windows accounts -- after exceeding the number of allowed attempts -- without a call to the IT help desk. The user can become authorized to reset his own password by answering a set of questions that are scored through an extremely flexible confidence-based scoring system.

In addition to questions and responses stored in v-GO SSPR directly, v-GO SSPR can retrieve user data (e.g., social security number, birthday) through an API from HR databases and other systems. The user's answers to the verification questions are checked against the data in these systems, and v-GO SSPR scores his performance.

When the user answers enough questions to reach a verification threshold, he is permitted to reset his password. Answering questions correctly adds to his score. Wrong answers subtract from his score but do not necessarily disqualify him. This way, lapses in memory and typing errors do not drive the user to the IT help desk.

v-GO SSPR supports three types of questions:

Required Questions:

These are questions are that all users should have factual answers for. These should have a pretty high and equal weighting for both correct and incorrect answers. Your It is strongly recommended that your selection of Required questions should have answers that come from as many different sources as possible. For example, in some states, a driver's license may display the social security number and date of birth.

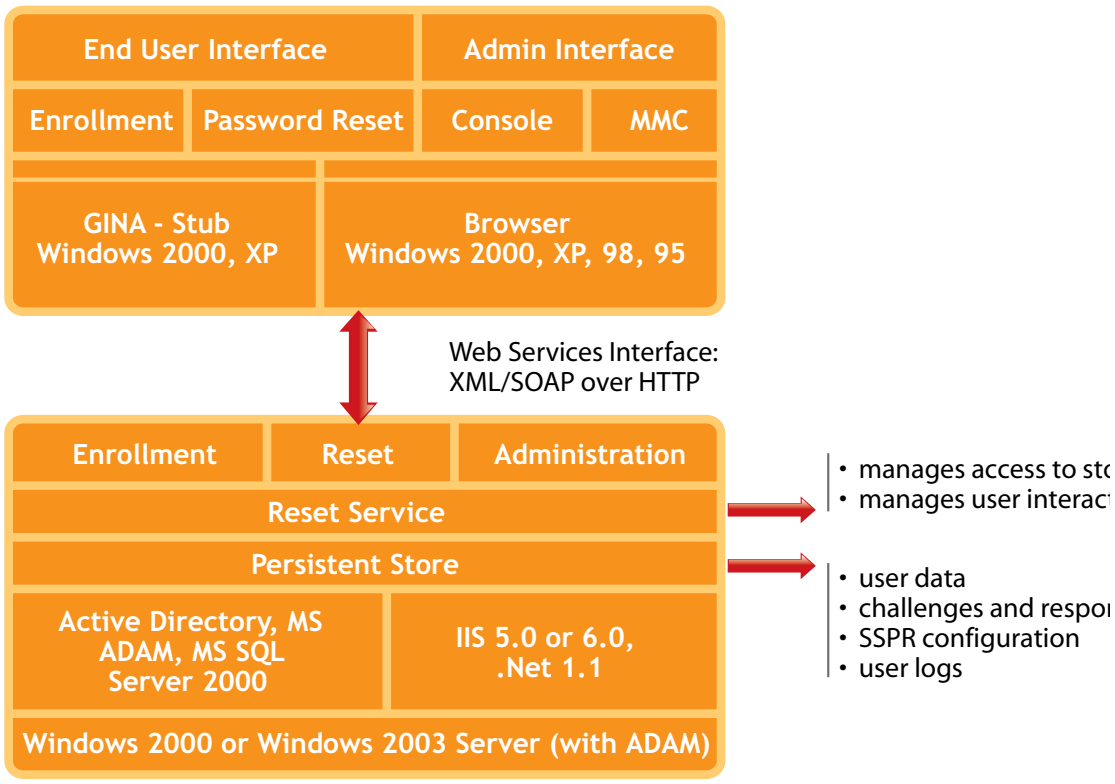
Eliminators:

These questions are called eliminators because the authorized end user is very unlikely to answer them incorrectly. The answers are personal, and therefore have low or no point-value for correct answers and high negative point-value if answered incorrectly.

Optional questions:

Optional questions are questions that are directed to certain groups in an organization, because they may not apply to all enrollees. Often these questions are preference questions which could change over time and are given less weighting for both correct and incorrect answers. Access to these and all other questions types can be control through administration console.

The diagram below provides a pictorial representation of the architecture.



Back End Services / Server

This server is responsible for managing all the questions and answer data in the repository, providing a challenge session when a user has forgotten their password, resetting the users password in the Active Directory and auditing all the event that have occurred. The server communicates with the SSPR Client via HTTP or HTTPS.

As an option, a Password Reset button can be installed on the user's Windows Desktop logon screen. By clicking this button, the user can initiate reset at their own desktop without being logged in.

The user's client is installed as a chained GINA stub and is only activated on pushing the button; it does not replace the Microsoft GINA. The GINA interface enables users to reset without being logged in. When users click the "initiate reset" button, the request is sent to the reset server.

v-GO SSPR can optionally force users to enroll. If users are not enrolled, a certain number of cancelled enrollments are allowed, after which users cannot logon again without first enrolling.

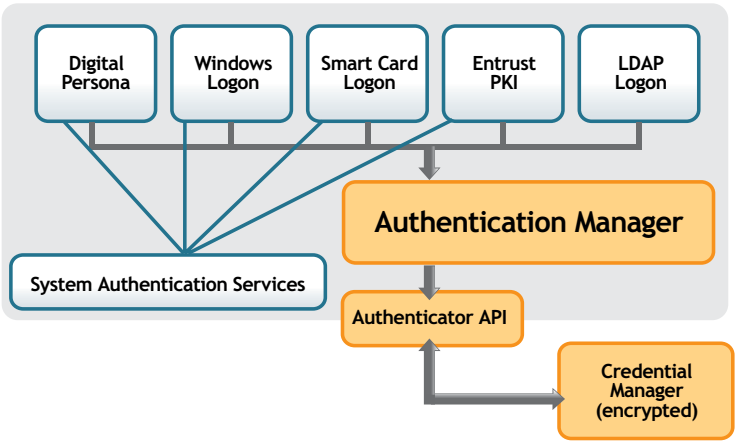
External data validation API

By default, SSPR requires that all the end-user enroll by answering the challenge questions. However, SSPR can also work with external data sources. External data sources allow organizations to write an interface to extract data from various sources (i.e. HR database) which contain the correct answers and can be used to validate an end-users response without forcing the end-user to enroll.

For example, lets say one of the reset questions is "What is your Social Security Number?" By default, when a user enrolls, the enrollment interview asks them to supply their social security number. Then when a user resets their password, they are asked to enter their social security number.

With an external validator in place, an administrator can direct SSPR to an external data source containing a pre-defined list of social security numbers. The user enters the answer to that question when attempting to reset their password, and SSPR validates it against the number in the external data source. If all system questions are answered by an external validator, users can skip the enrollment process.

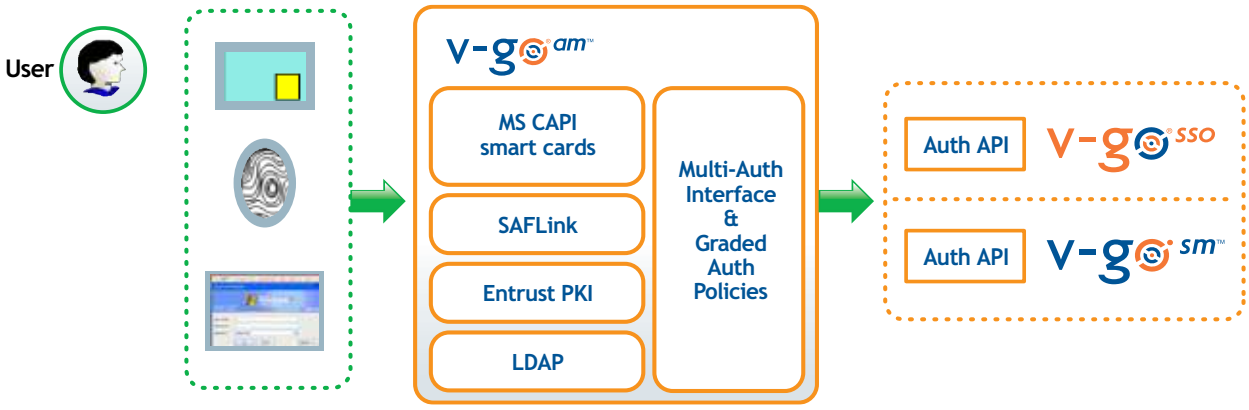
v-GO Authentication Manager



The interface between v-GO SSO and any single user authentication device (e.g. smart card) or service (e.g. Windows logon) is in a v-GO internal component called an **authenticator**. v-GO SSO ships with two authenticators: Windows authenticator and LDAP authenticator – and works with only one authenticator at a time.

v-GO Authentication Manager (v-GO AM) adds many more authenticators, including one for smart card, Xyloc proximity badge, Entrust PKI, Digital Persona biometric devices, and SafLink biometric middleware. Others are planned over time.

v-GO AM also contains a meta-authenticator named **multi-auth**. This meta-authenticator manages end-users switching between different authenticators on the fly without having to reconfigure v-GO SSO. The diagram below illustrates this architecture.



v-GO AM can restrict single sign-on to certain applications to only those instances when the user has logged on with a sufficiently strong authenticator. v-GO AM assigns a maximum strength value to each authenticator, called a **grade**.

The administrator can assign a grade to each authenticator. For example, Windows id/password has a grade of 1, a Xyloc proximity badge has a grade of 2, and a smart card has a grade of 3.

Within v-GO SSO are **application templates**. These are the signatures that v-GO SSO uses to identify and respond to applications. Each template contains the window title name, control ID values, and other variables that control v-GO SSO's behavior when responding to that application.

The application template contains a value that defines the minimum authenticator grade required before v-GO SSO will respond to the application. For example, the SAP R/3 template may specify a minimum grade of 2.

Before v-GO SSO responds to an application logon request, it verifies that the maximum grade of the authenticator used to initially sign-on to Windows and v-GO SSO is equal to or greater than the minimum grade specified by the application template. If not, v-GO AM prompts the user to authenticate with another authenticator whose grade is sufficient for v-GO SSO to complete the sign-on.

If the user does not fulfill this authentication request, v-GO SSO does not sign-on to the application. So, in the example above, users who log on with a Windows ID and password cannot sign on to SAP R/3 until they present a Xyloc badge or smart card.

v-GO Provisioning Manager

v-GO Provisioning Manager ("v-GO PM") is a gateway that resides between v-GO SSO and data sources for users' application IDs and passwords. v-GO PM is implemented as a .Net application that runs under Microsoft IIS and an add-on module to the v-GO SSO Agent.

v-GO PM implements the Simple Provisioning Markup Language (SPML) standard. SPML has 4 instructions that can process, add credentials, modify credentials, delete credentials, and add new users. These instructions can include one or more of the following data elements: application name, SSO user name (often the same as the Active Directory user name), application user ID, application password, application third field (e.g. domain or database name), and application fourth field.

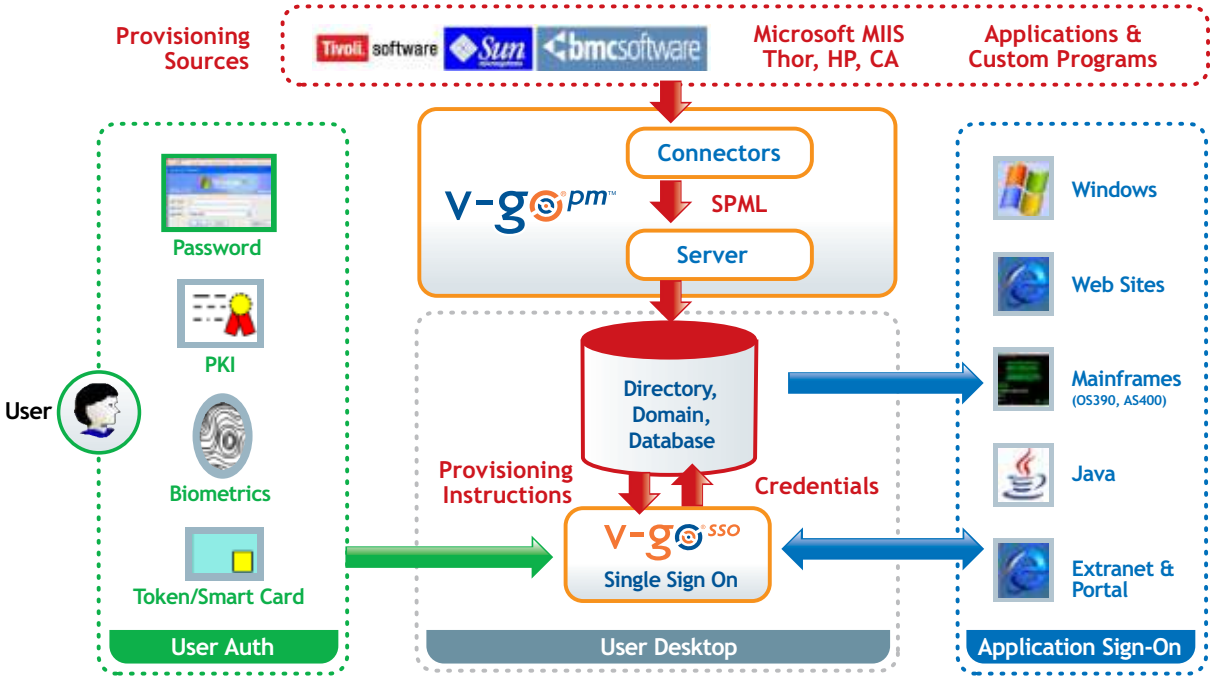
v-GO PM can receive instructions and data from a variety of sources. Encrypted and stored in the appropriate user containers, these instructions can be retrieved and followed by v-GO SSO. v-GO PM can write instructions to a subset of the repositories supported by v-GO SSO: Microsoft Active Directory and ADAM, IBM Tivoli Directory, Oracle Internet Directory, and Sun JAVA Directory.

No databases are supported. v-GO SSO retrieves the instructions -- e.g., Add, Lotus Notes, John Doe, johndoe, password -- and takes the appropriate action, such as creating a new credential in the v-GO SSO Local Credential Store for Lotus Notes.

v-GO PM exposes a server side programming interface, available as both a .Net and Java API and a command line interface ("CLI"). Administrators can use this programming interface to develop custom provisioning programs or connectors to data sources. These data sources can push instructions and data into v-GO PM.

v-GO PM comes with a Web-based console -- implemented using the CLI/API -- that permits an administrator to interactively provision v-GO SSO. It also provides a series of connector" written using the API. These connectors enable direct integration with several commercially available identity management systems including IBM Tivoli Identity Manager, Sun Identity Manager, Oracle Identity Manager, and BMC Control-SA.

These connectors are workflow steps that are installed in the relevant application provisioning workflow of the IDM system. They intercept the provisioning action and relevant data from the workflow, and route a copy of that data to v-GO PM.



v-GO Session Manager

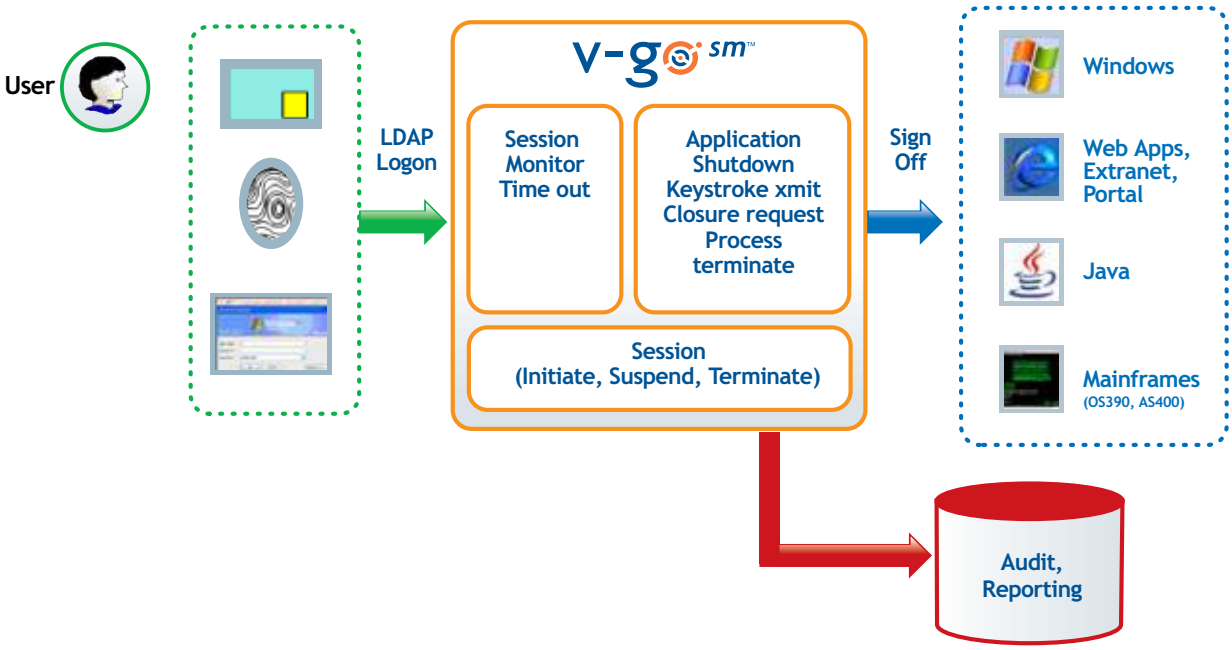
v-GO Session Manager (v-GO SM) is a client component that executes along side of v-GO SSO. It enables user sessions and to automatically terminate open applications upon session end.

v-GO SM has three session states: active, suspended, and no session. An active session is one where the user is actively working on the kiosk. A suspended session is one where a user is logged in but not actively working on the kiosk. A no session is when there is no logged in user. A terminated session is the same as no session [missing a word here?' …. a previously active session was terminated.)

v-GO SM executes different actions when the certain events – such as session state change – occur. Major events are session begin, session end, session suspend, session unsuspend, authenticator removal, authenticator timeout, and external application call received. Major actions are terminate applications and run program.

To determine which applications to terminate, v-GO SM implements a black list / white list concept. If a black list is specified, v-GO SM terminates applications named on the black list. If a white list is specified, v-GO SM terminates all applications except those named on the white list.

When a black list is specified, each application is terminated in a fashion dictated by an application template – the same concept used for application sign-on. Among other things, the template specifies whether the application should be terminated with a Windows Close command (similar to a user pressing Alt-F4), a process-kill (similar to a user ending a process in Windows Task Manager), or with a specific series of keystrokes (for example, to close any open database records before shutting down the application).



v-GO SM has three major components: GINA stub, Desktop Manager, and Session Agent. The v-GO SM GINA stub chains to the Microsoft GINA to enable user logon. It also prevents end users from shutting down v-GO SM and circumventing the user logon requirement. Note: v-GO SM does not replace the Microsoft GINA.

The Desktop Manager is responsible for granting users access to the kiosk desktop once authentication has been completed. It hides and unhides the Microsoft desktop with the v-GO SM screen saver.

The Session Agent monitors session activity or inactivity, lock and unlocks access to the kiosk, displays a screen-saver that informs other users about the session state, and executes actions upon specified events.

v-GO Credential Manager

v-GO Credential Manager ("v-GO CM") enablesadministrators with a means to centrally manage smart cards or other authentication devices and end-users with a means to self-administer many aspects of those devices.

v-GO CM consists of a server component and client-side authentication middleware. The server is implemented as a .Net based application that runs on Microsoft Windows 2000 and 2003.

Active directory, Microsoft SQLServer, or Microsoft SQLServer Express are used for storage of configuration data and attributes, and SQLServer for storage of event and audit data. v-GO CM uses the Microsoft PKI for certificate services, but can also work with other PKI vendors. v-GO CM does not rely on Microsoft's Certificate Lifecycle Manager.

v-GO CM currently administers smart cards. When a smart card is issued to auser, the software pairs the card's unique identification number with the user and puts the card into an active state. It also interacts with the PKI to request and retrieve the use certificates and place them on the card.

The user can select a unique PIN code for the card, and unblock the card after 3 bad PIN attempts. The administrator can revoke a card when an employee leaves the company, and the card can be recycled for a new user.

v-GO CM uses a secure infrastructure to maintain the smart card and user information. The smart card data – including physical access rights, logical access rights, and certificates -- is encrypted to prevent security breaches. Sensitive information, such as transport/master keys and unblock PIN, is encrypted and stored in the directory.

v-GO CM uses a Key Encryption Key (KEK) architecture requiring the presence of a security officer to start or restart the system. A hardware cryptographic module -- theHardware Security Module (HSM) -- will be supported in the future.

Smart Card Issuance Process

To issue smart cards to new users, the administrator provisions the certificates for the users from the v-GO CM administration console. From their desktops, users make a secure connection to the v-GO CM web portal.

The v-GO CM portal prompts users for their current network security credentials. Once users are verified, they can conduct the certificate issuance process.

Through the secure connection to the v-GO CM server, all cryptographic operations are performed on the smart card. So there is never a need for the user's credentials to be revealed outside the smart card.

Once issued, the card can be used for certificate-based network logon as well as to unlock the v-GO SSO credential store. The end-user does not need to come to the administrator's office to pick up and initialize the smart card.

Lost Card Scenario

If a user loses his card and needs a replacement, v-GO CM knows which key pairs to load back from the archive into the replacement card. The user can request a new card from his desktop through the v-GO CM secure portal, in the very same way as the initial issuance of the card took place.

###

For more information on how you can eliminate the need for users to remember passwords to their enterprise applications, contact Passlogix today:

Passlogix, Inc.
160 Pearl Street, #400
New York, NY 10005
Phone 212-825-9100
1 (866) PASSLOGIX Toll free
1 (866) 727-7564
Fax 212-825-0326
www.passlogix.com

v-go

passlogix

white paper

160 Pearl Street, 4th floor, New York, NY 10005
Tel: 212.825.9100 x 2
or 866.727.7564 x 2
Fax: 212.825.0326
Web: www.passlogix.com
Email: sales@passlogix.com