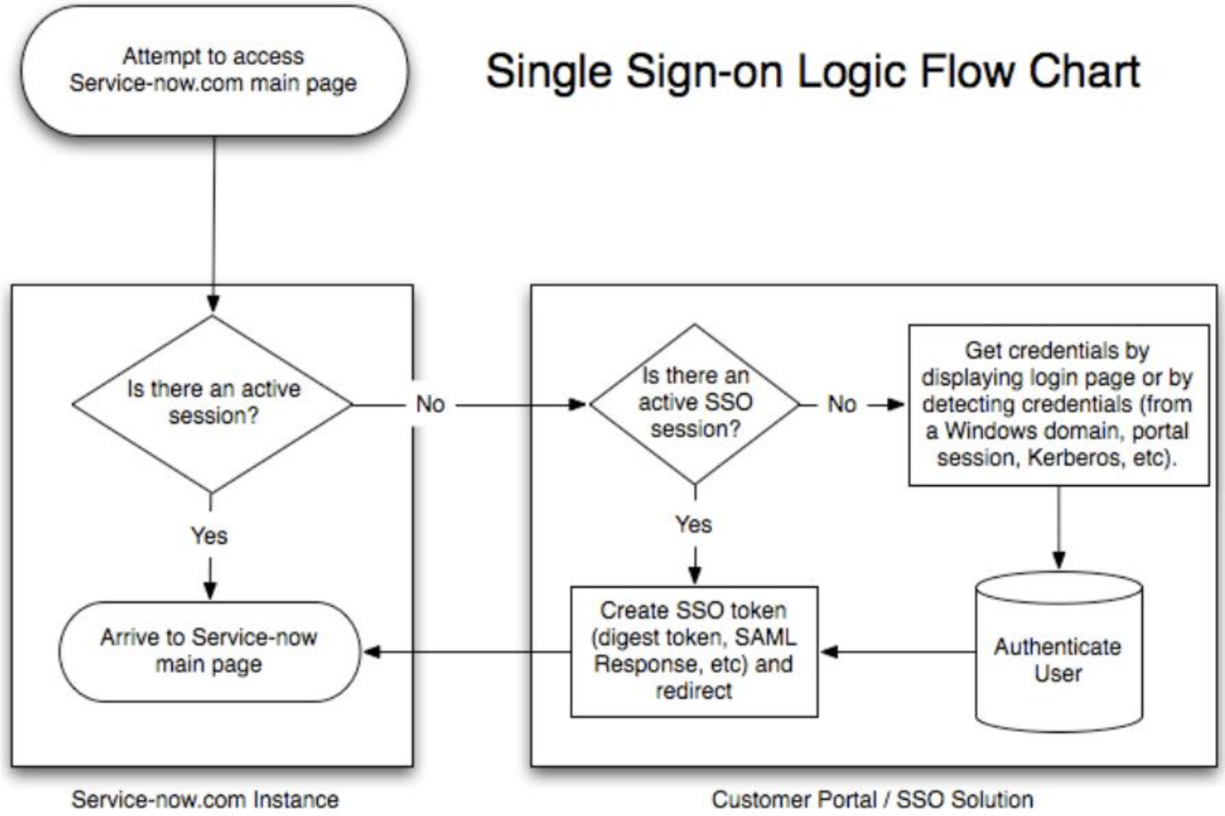
(1) The **Security Assertion Markup Language** (**SAML**) is an XML-based standard for exchanging authentication and authorization data between security domains.


Single Sign-on Logic Flow Chart
Attempt to access Service-now.com main page
Is there an active session?
No
Yes
Arrive to Service-now main page
Is there an active SSO session?
No
Yes
Get credentials by displaying login page or by detecting credentials (from a Windows domain, portal session, Kerberos, etc).
Authenticate User
Create SSO token (digest token, SAML Response, etc) and redirect
Service-now.com Instance
Customer Portal / SSO Solution

(2) The **Multifactor authentication**, also known as two-step verification, is a security requirement that asserts a user enter more than one set of credentials to authenticate to an instance.

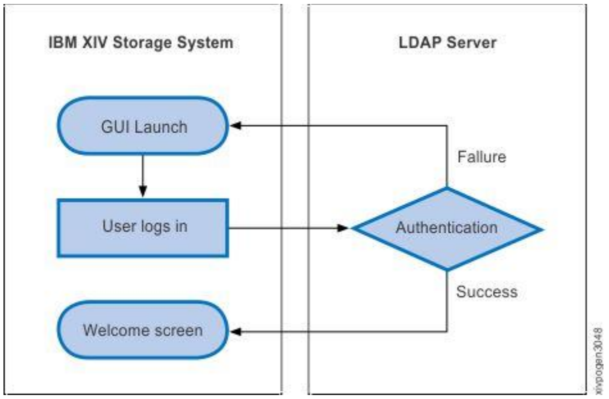
The basic level of authentication to an instance is local database authentication—the user enters a username and password combination. Multifactor authentication, in con trast, gives administrators and users the ability to require a second level of authentication—the user must enter a passcode or token in addition to the password. A mobile application on a user mobile device generates the passcode.

Users can require multifactor authentication for their own login credentials.

Administrators can require multifactor authentication for any user login credentials.

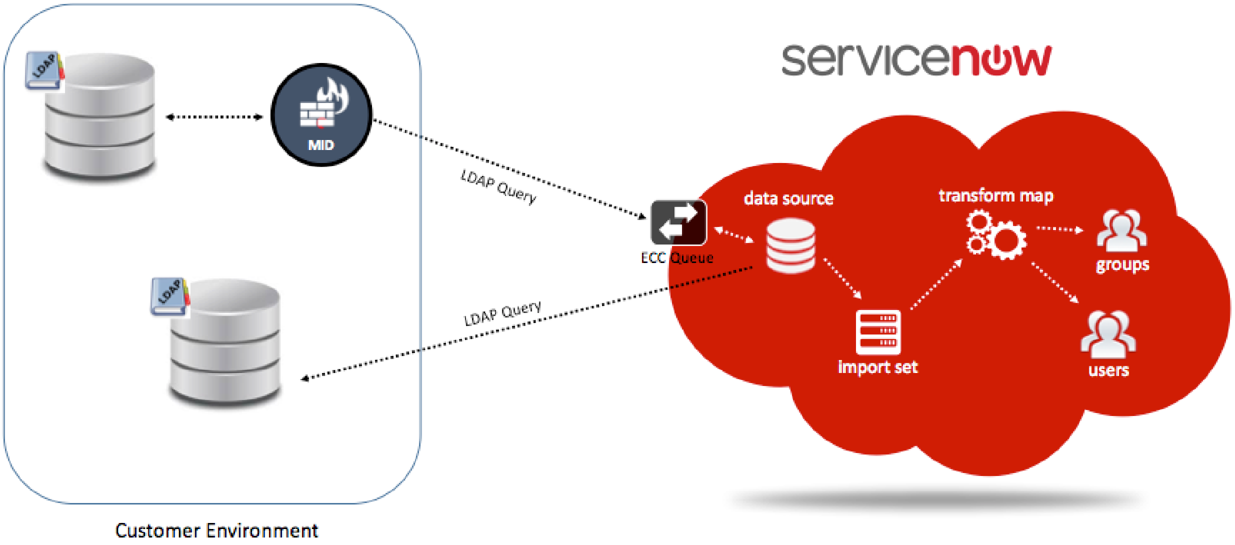
(3) If you want to use **Local Database Authentication** when Multi-SSO is active, you must set the glide.authentication.external.disable_local_login property to **false**.

(4) The **LDAP** (**Lightweight Directory Access Protocol)** is used to streamline the user login process and to automate administrative tasks such as creating users and assigning them roles. An LDAP integration allows the system to use your existing LDAP server as the master source of user data. Typically, an LDAP integration is also part of a single sign-on implementation.



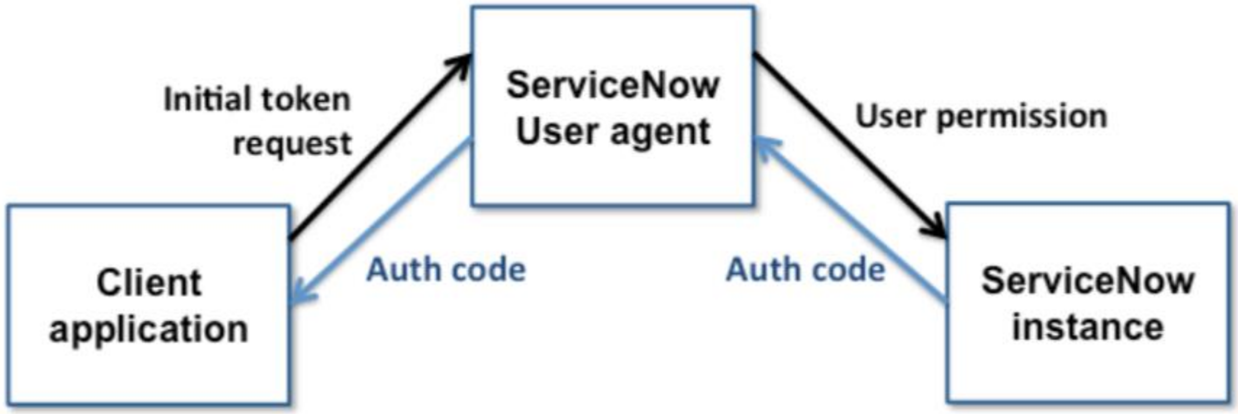
The integration uses the LDAP service account credentials to retrieve the user distinguished name (DN) from the LDAP server. Given the DN value for the user, the integration then rebinds with LDAP with the user's DN and password. The password that the user enters is contained entirely in the HTTPS session. The integration never stores LDAP passwords.

The integration uses a read-only connection that never writes to the LDAP directory. The integration only queries for information, and then updates its internal database accordingly.



(5) (5) The **OAuth** (Open Authorization) is an open standard for token-based authentication and authorization on the Internet.

OAuth allows an end user's account information to be used by third-party services, such as Facebook, without exposing the user's password. OAuth acts as an intermediary on behalf of the end user, providing the service with an access token that authorizes specific account information to be shared. The process for obtaining the token is called a flow.

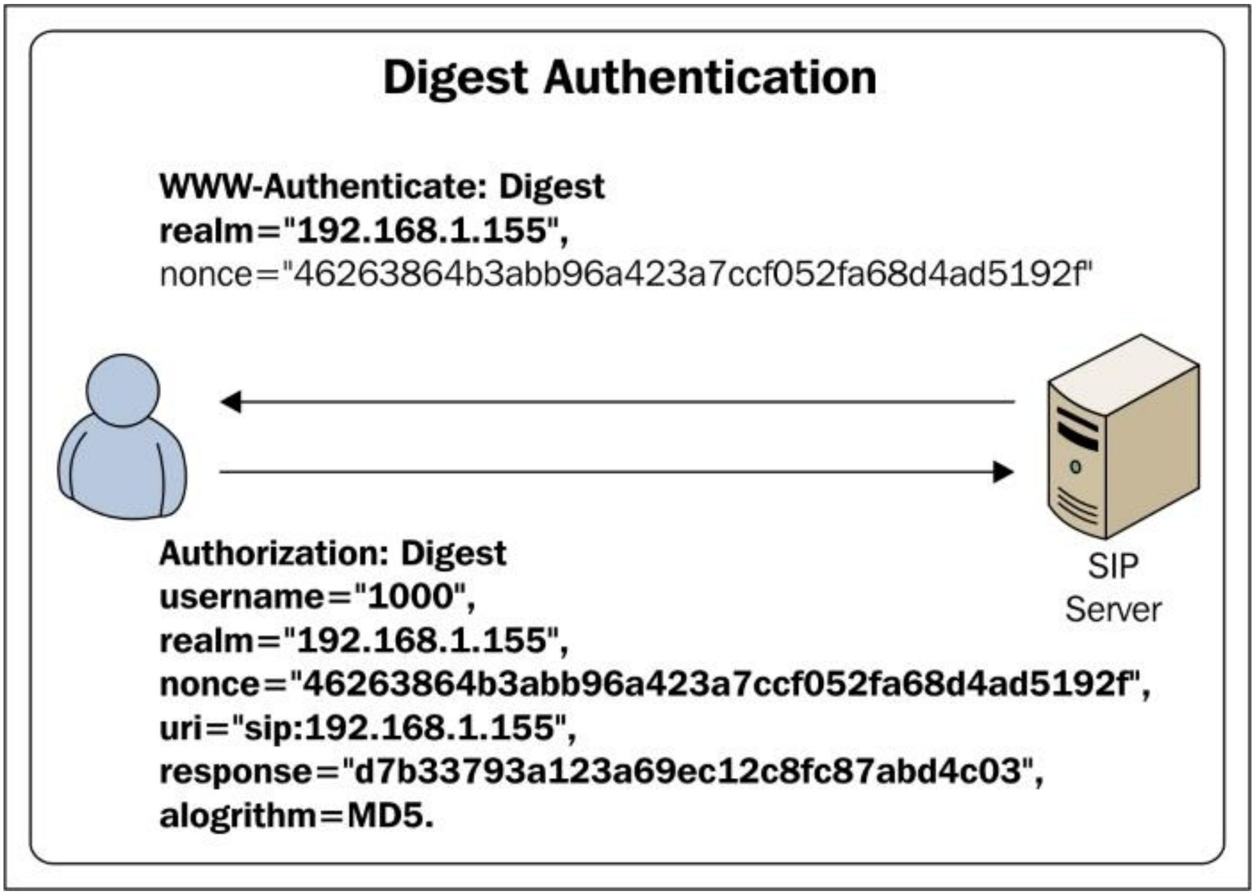


OAuth is an open standard for access delegation, commonly used as a way for Internet users to grant websites or applications access to their information on other websites but without giving them the passwords. This mechanism is used by companies such as Amazon, Google, Facebook, Microsoft and Twitter to permit the users to share information about their accounts with third party applications or websites.

(6) The **digest token** authentication passes user credentials and a digest token within an unencrypted HTTP header.

The instance reads the HTTP header value and compares its computed hash value of the digest token. If the computed hash value matches the digest token value, then the instance searches for a matching value in the User table. If there is a matching value in the User table, the instance considers the user pre-authenticated and logs the user in.

Digest token authentication is more secure than simple unencrypted HTTP headers because any accidental or intentional change to the unencrypted HTTP header produces a different hash value. If the hash value fails to match, the instance denies the user access to the requested instance. This prevents users from attempting to login with another user's credentials.



```
WWW-Authenticate: Digest
realm="192.168.1.155",
nonce="46263864b3abb96a423a7ccf052fa68d4ad5192f"
```

```
Authorization: Digest
username="1000",
realm="192.168.1.155",
nonce="46263864b3abb96a423a7ccf052fa68d4ad5192f",
uri="sip:192.168.1.155",
response="d7b33793a123a69ec12c8fc87abd4c03",
alogrithm=MD5.
```