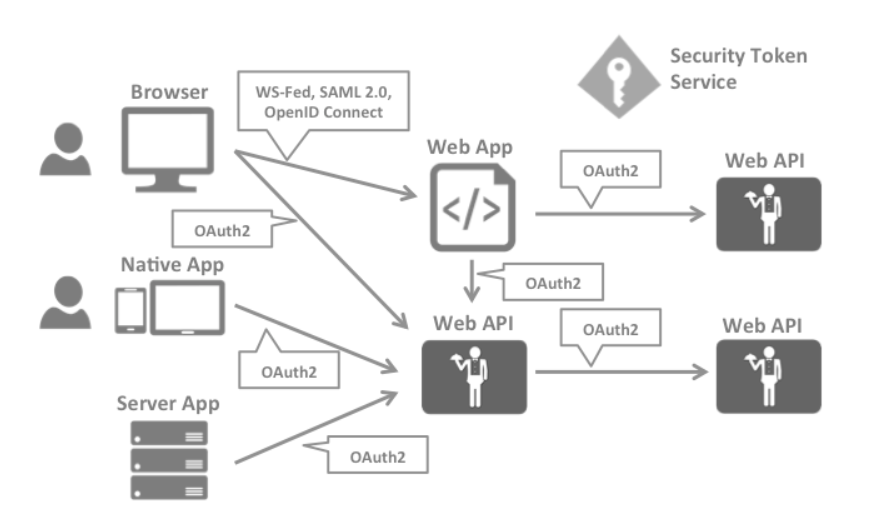
0.1.6：

1.CORS default true

2.sessions: Add read:self and cancel:self actions and enable them by default (in new project scopes) for all sessions. This allows a user to read or cancel any session that is associated with their user ID. read and cancel actions are still available that allow performing these actions on sessions that are associated with other users

0.2.0

1. New OIDC(OpenID Connect) auth method type added with support for create, read, update, delete, and list (see new cli oidc subcommands available on CRUDL operations for examples), as well as the ability to authenticate against it via the SDK, CLI, admin UI, and desktop client. 

2. Permissions in new scope default roles have been updated to include support for list, read:self, and delete:self on auth-token resources. This allows a user to list and manage their own authentication tokens. (As is the case with other resources, list will still be limited to returning tokens on which the user has authorization to perform actions, so granting this capability does not automatically give user the ability to list other users' authentication tokens.)

3. cli: Support for reading and deleting the user's own token. New logout command deletes the current token in Boundary

0.3.0

1. The new Managed Groups feature allows groups of accounts to be created based on an authenticating user's JWT or User Info data. This data uses the same filtering syntax found elsewhere in Boundary to provide a rich way to specify the criteria for group membership. Once defined, authenticated users are added to or removed from these groups as appropriateds each time they authenticate. These groups are treated like other role principals and can be added to roles to provide grants to users.

0.4.0

1. Credential Stores: This release introduces Credential Stores, with the first implementation targeting Vault. A credential store can be created that accepts a Vault periodic token (which it will keep refreshed) and connection information allowing it to make requests to Vault.

2. Credential Libraries: This release introduces Credential Libraries, with the first implementation targeting Vault. Credential libraries describe how to make a request to fetch a credential from the credential store. The first credential library is the generic type that takes in a user-defined request body to send to Vault and thus can work for any type of Vault secrets engine. When a credential library is used to fetch a credential, if the credential contains a lease, Boundary will keep the credential refreshed, and revoke the credential when the session that requested it is finished.

3. Credential Brokering: Credential libraries can be attached to targets; when a session is authorized against that target, a credential will be fetched from the library that is then relayed to the client. The client can then use this information to make a connection, allowing them to gain the benefit of dynamic credential generation from Vault, but without needing their own Vault login/token (see NOTE below).

NOTE: When using credential brokering, remember that if the user can connect directly to the end resource, they can use the brokered username and password via that direct connection to skip Boundary. This isn't any different from normal Boundary behavior (if a user can directly connect, they can bypass Boundary) but it's worth repeating.

0.5.0

1. Credential Libraries: renamed to credential sources.

2.