Few questions we need to address with CA Embedded Entitlements Manager (CA EEM).

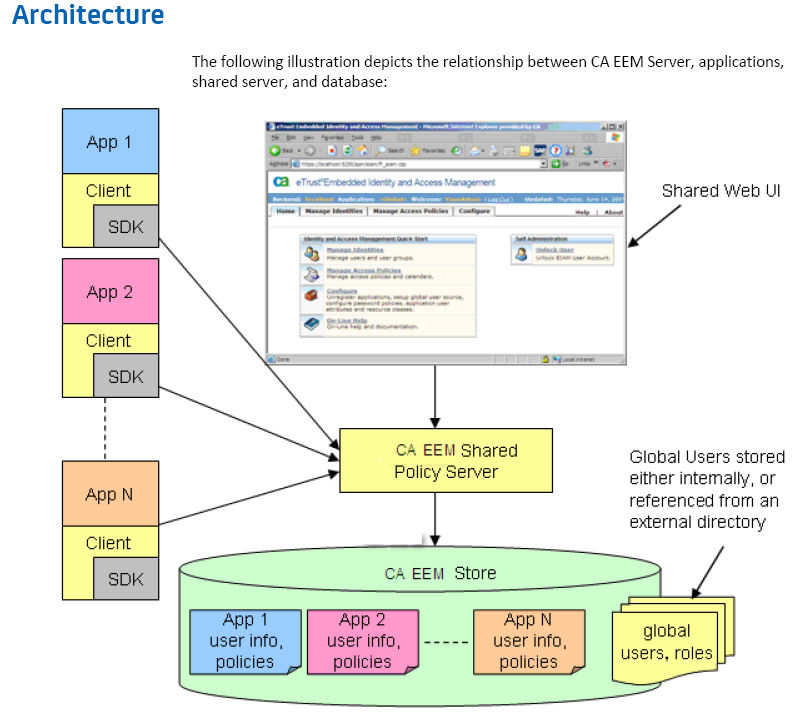
1. What forest and domain are we going to use for the LDAP connector ?
   1. Based on this, we will need the username / domain password for authentication are we going to use a service account for this or do we need to request a special ID / Access for this ?
   2. We need to know the Base DN
   3. We will need to make sure the LDAP ports between the server and the Active Directory Domain Controller are open. Possible Ports:

|  |  |  |
| --- | --- | --- |
| Protocol and Port | AD and AD DS Usage | Type of traffic |
| TCP and UDP 389 | Directory, Replication, User and Computer Authentication, Group Policy, Trusts | LDAP |
| TCP 636 | Directory, Replication, User and Computer Authentication, Group Policy, Trusts | LDAP SSL |
| TCP 3268 | Directory, Replication, User and Computer Authentication, Group Policy, Trusts | LDAP GC |
| TCP 3269 | Directory, Replication, User and Computer Authentication, Group Policy, Trusts | LDAP GC SSL |
| TCP and UDP 88 | User and Computer Authentication, Forest Level Trusts | Kerberos |
| TCP and UDP 53 | User and Computer Authentication, Name Resolution, Trusts | DNS |
| TCP and UDP 445 | Replication, User and Computer Authentication, Group Policy, Trusts | SMB,CIFS,SMB2, DFSN, LSARPC, NbtSS, NetLogonR, SamR, SrvSvc |
| TCP 25 | Replication | SMTP |
| TCP 135 | Replication | RPC, EPM |
| TCP Dynamic | Replication, User and Computer Authentication, Group Policy, Trusts | RPC, DCOM, EPM, DRSUAPI, NetLogonR, SamR, FRS |
| TCP 5722 | File Replication | RPC, DFSR (SYSVOL) |
| UDP 123 | Windows Time, Trusts | Windows Time |
| TCP and UDP 464 | Replication, User and Computer Authentication, Trusts | Kerberos change/set password |
| UDP Dynamic | Group Policy | DCOM, RPC, EPM |
| UDP 138 | DFS, Group Policy | DFSN, NetLogon, NetBIOS Datagram Service |
| TCP 9389 | AD DS Web Services | SOAP |
| UDP 67 and UDP 2535 | DHCP  Note DHCP is not a core AD DS service but it is often present in many AD DS deployments. | DHCP, MADCAP |
| UDP 137 | User and Computer Authentication, | NetLogon, NetBIOS Name Resolution |
| TCP 139 | User and Computer Authentication, Replication | DFSN, NetBIOS Session Service, NetLogon |

1. We need to setup an Elevated Privileged users group
   1. We need to know what the user ID’s should be in this group (I believe we only need one)



* 1. How are we going to manager global users and global groups ?
     1. Passwords and password policies.



Policy Server

The CA EEM Policy Server is shared by all the applications and is used to:

* Authenticate users
* Deliver audit events to audit collection tools
* Store application-specific information
* Set application-level user attributes
* Configure external user store
* Set calendars

CA EEM performs policy evaluation by calling the Safe::Context::authorize method. This method is invoked with the following parameters:

* Identity to check
* Resource class name
* Resource name
* Action requested
* Queue of named attributes

Defining Identity and Access Requirements

1. Can we use the “safex” command to migrate identities from the 4.x environment to R11.3x
2. We need to talk about

Accessing

If the application includes an internal access control method and you want to implement it using CA EEM, automation can be used to create an CA EEM representation of the application's policies, depending on the complexity and structure of your current ACLs or policies. Alternatively, you can also create the corresponding CA EEM policies through the Web UI.

If the application uses calendars along with access policies, you may can automation for

converting them to CA EEM calendars.

1. We need to talk about

Modifying StoredObjects

The StoredObject class provides the interface to the repository on the CA EEM backend

server. The following CA EEM objects are derived from the StoredObject class:

* ApplicationInstance
* GlobalUser
* GlobalUserGroup
* User
* UserGroup
* Policy
* Calendar
* AppObject

You can modify StoredObjects as follows, based on the environment:

1. Registering an Application
2. Attaching to a Backend Server
3. Creating an Application Instance
4. Defining User Attributes
5. Create Global Users
6. Create Application Specific Users
7. Associate Global User with Application-Specific Details
8. Create Global User Groups
9. Retrieve a Global User Group
10. Retrieve a User Group
11. Policy Evaluation

Policies are evaluated in the following process:

1. Check for explicit denies:
   1. Match for explicit denies.
   2. Evaluate matched policy filters.
   3. In case of explicit deny, stop checking, and return a denied recommendation specifying the policy.

2. Check for explicit grants:

1. Match for explicit grants policies.
   1. Evaluate matched policy filters.
   2. In case of explicit grants, stop checking, and return a granted recommendation
   3. specifying the policy.

3. Check for delegated authority:

1. Match/evaluate the delegated authority. For each delegator, find a grant with
2. no explicit deny.
3. For each delegator, repeat step 1and search for explicit grants.
4. If a grant was returned by delegation, return a granted recommendation
5. specifying the policy and the delegator chain.

4. Calculate obligations for this access check:

1. Add the following attributes to the ones passed in the authorization call:

* PolicyName, the name of the obligation policy that caused the response
* DelegationChain, the name of the delegation chain returned

1. b. Match and evaluate each SafeObligation as follows:

* ResourceClass set to SafeObligation.
* Resource name set to {action} + "/" + {original resource class} + "/"

1. {original resource name}.

* Action set to FulfillOnGrant (if the authorization results in a grant), or

FulfillOnDeny (if the authorization results in a deny).

1. Do the following for each matching or evaluating SafeObligation policy:

* Append each obligation to the authorization results.
* Calculate the values of the obligation attributes and append them to the authorization results.

Note: Applications must handle the obligations returned from an authorization check. The application should not grant or deny access until and unless the obligations could not be performed.

5. Return a denied recommendation, in case of no matches:

Note: All the policies are not evaluated for every request. Since CA EEM supports explicit policies (explicit grant or explicit deny), policy evaluation is performed only at the first instance.