Basic IdM objects

identity, role, contract

Identity

Identity is the most basic object in CzechIdM. From the user's point of view, they must have an identity to login to the IdM.

Identity is used for:

- Assigning permissions - this means binding to a role. This relation is implemented by IdmIdentityRole.
- Only Identity can approve user tasks in workflow processes.
- Only Identity can be notified by e-mail, sms, ... (see Notification).
- Assigning contracts.
- Account assignment - this means linking to an IdM account. This relation is implemented by AccIdentityAccount. Account Management is responsible for administering IdM accounts.
- Identity can be propagate to the end systems, ie Provisioning.
- Identity is possible Synchronization from the end system.

Structure of the Identity object:

- The basic static fields of Identity object are here IdmIdentity.
- Dynamic fields - Object Identity, supports EAV attributes that allow the dynamic addition of additional fields.

Role

Roles in CzechIdM are mainly used to assign permissions to the user. By assigning a specific role, the user gets permission to access application agendas. Permissions obtained from role also provide access to specific agenda data (rights on data).

For example, we can have a "Worker" role that allows the user to see only their own profile. We can also have the role a "Leader". This role allows a leader to gain access to the data of his or her subordinates and grants him or her permission to edit it.

In addition to controlling permissions within CzechIdM, it allows the role to assign and manage an account on an end system. An example is the "LDAP" role, the assignment of which results in an account being created on the LDAP end system.

Role is used for:

- Assigns permissions on agendas and their data in CzechIdM. A detailed description of the permissions, including the configuration example, is here.
- Assigns an account to the end system. More specifically, it defines the link between the role and the system. Allows overloading mapped system attributes. An example of how to create link a

role-system and how overload the default attributes is here.
- The role can be assigned manually (user request) or automatically. Automatic role assignation is based on tree structure (usually on organizational structures). This means that if the identity is assigned to the tree structure to which the automatic role is attached, this role will be assigned to identity.

Structure of the Role object:

- The basic static fields of object Role are here IdmRole.
- Dynamic fields - Object Role, supports EAV attributes that allow the dynamic addition of additional fields.

Contractual relationship

This binding object defines a contractual identity relationship. A typical example is when the user has a contract in the company. This contractual relationship defines its name (position name), position in the organizational structure, and the validity of this contract. The relationship also determines who is the leader and subordinate. The leaders for the contract may result from the position in the organizational structure. Leaders can also be manually defined directly for the contract.

An important one is that all assigned roles of an identity are always assigned through a contractual relationship. In general, each identity has at least one contractual relationship. This feature allows for effective identity control (after the expiration of the contractual relationship, appropriate permissions will also be removed).

Example: We have a "john.doe" user who has two jobs in the company (two contracts). The first working relationship is a "doorman", this job has no subordinates. The second working relationship is the "chief caretaker" who has 5 subordinates. As a result, the identity of "john.doe" ends up having two contractual relationships created. The first one was called "doorman", the second was called "chief caretaker". Both are then linked to the tree of the organizational structure. This will result from both the leaders and the subordinates. For each contract, appropriate roles are created to grant permissions to the user. For example, if the second job ends, only the permissions that belong to that relationship are removed.

Contract relation is used for:

- Define identity contractual relationships. See more here.
- Assigns (controls) identity rights. All assigned roles are linked through a contract.
- Defines leaders and subordinates by assigning identity (contract) to a specific tree node (typically a tree of organizations).
- Assigns permissions based on tree node binding. If the tree node has automatic roles defined, these roles will be assigned after the creation of the contract. See more here.
- Defines leaders by direct link to the contract. This allows you to manually define a leaders.

Structure of the Contract relation object:

- The basic static fields of object Relation are here IdmIdentityContract.

- Dynamic fields - Object Relation, supports EAV attributes that allow the dynamic addition of additional fields.