Contractual relationship (CR)

contract, identity

They define the link between the identity and the tree structure. In the application, we advance the logic according to which every identity has at least one CR. This is why there is one CR which is formed automatically to every identity after its creation according to the configuration of the default organizational structure. If **default element of the structure is configured**, then this one is used when creating the default CR ⇒ **the identity is "positioned" on the default position of the organizational structure in question**. If there is no selected default element of the structure, the identity is "positioned" on the position named as **"Default"** without being included in the organizational structure.

The CR plays a significant, if not the main role when assigning role to the identity - **the role is always assigned to the CR, not directly to the identity**. This is a way of ensuring that the authorization evaluation will always pass through one way through the CR where a tree (organizational) structure can figure ⇒ the authorizations can be linked through these structures / positions in the organization.

Another intended functional feature is that when the CR ceases to exist / is invalidated, all the roles ensuing from this CR will cease to exist as well. For the periodic review of invalid CR, a task which can be scheduled has been created `IdentityContractExpirationTaskExecutor`.

Through CR, users are searched in the elements agenda of the tree structure / organizational structure, who are "positioned" on the selected element. In the agenda, only the users related to a certain type of the structure are displayed ⇒theyhavean CR with a selectedtype.

Prime contract position

CR can be flagged as "main". Can be flagged more than one main CR or non. Prime contract is computed by CR priority:

1. main
2. valid (valid by from-till and not disabled)
3. with working position with default tree type
4. with working position with any tree type
5. with undefined valid from
6. other with lowest valid from

Search managers by CR

Managers could be found:

- through tree structers - identity with CR on tree node above.
- through the direct relation on CR - guarantees.

Searching managers and subordines could by overriden in custom module by implementing `SubordinatesCriteriaBuilder` interface.

CR state, validity

When CR validity ends, then all roles assigned to given CR is removed. Its not possible to assign roles to invalid contracts. Invalid contract is defined by:

- `validFrom` and `validTill` attributes
- and `state` attribute:
  - `DISABLED` - contract is invalid
  - `EXCLUDED` - contract is excluded but remains valid (if contract is valid by `validFrom` and `validTill` attributes). Roles assigned for this contract are not removed - accounts on target systems remains untouch. Roles assigned for this contract are not added for logged identity.
  - otherwise is contract validity controlled only by `validFrom` and `validTill` attributes ⇒ state can be `null`.

HR processes

HR processes depends on CR state and validity:

- When contract is invalid, then all assigned roles for this contract are removed (automatic and manually assigned roles too) by configured HR processes. When contract will be valid again, then all automatic roles are assigned again.
- When the last contract is removed or all contracts are invalid or excluded, then identity is disabled by configured HR processes too. When contract is valid again or new valid contract is added, then identity is activated again.

This automatic processes can be configured two ways, by:

- processors - executes process immediately, when identity contract is changed (active operation).
- long running tasks - long running tasks are scheduled, mainly over night. So contract change is persisted only and HR processes are executed separately.

Choosing the way is configurable - processor can be disabled, long running tasks can be scheduled or not.

Other contract positions

Other positions can be configured for the contract. Other contract positions are used just for assigning of automatic roles by the tree structure. Filtering and evaluating managers and subordinates are not supported by other contract positions.

Tree structures indexing

To make queries in an efficient manner, a separate library on the tree structure has been created

ForestIndex which builds an index next to the tree structure with the following advantages:

- the possibility to ask about the children of any element of the tree with one query (all the children in the downward direction)
- the possibility to ask about all the parents of any element of the tree with one question

The documentation and an an example of getting involved in the project can be found here.

Searching through index is linked to:

- Searching identities according to the organizational structure (through CR)
- Display of the identity position in the organizational structure

To rebuild the index, the task `RebuildTreeNodeIndexTaskExecutor` where you need to enter the **code** of the structure which should be re-indexed.

Automatically assigned roles

role, contract

The intro is described in the admin section here.

Heredity of assigned roles

If the role is assigned to an organizational structure component, the following behaviour may occur:

```
  A
  |
  B
 / \
C   D
   / \
  E   F
```

- The role is assigned to a user who is linked precisely to this specific organizational structure component
  - If the role is linked to "B", the role will be automatically assigned to every user whose contract is assigned to "B"
- The role is assigned to all users who are linked to this organizational structure component and to the whole subtree (from the root to the leaves)
  - Heredity exists in the whole subtree without depth restrictions
  - If the role is linked to "B", the role will be automatically assigned to every user whose contract is assigned to "B", but also when it is in "C", "D", "E", "F".
- The role is assigned to all users who are linked to this structural organization component and to all the managers (i.e. from the node to the root).
  - This behaviour enables assigning according to the scenario "the manager has everything which the subordinate does". Therefore, by linking the role to "B", it is automatically assigned to the users with a contract in "A".
  - There is at least one customer where this is used
  - We are not going to implement this now, but it is mentioned here because of potential

consideration of possibilities on if and how we could solve this

Audit

All changes in assigning roles to the organizational structure will be audited. The minimum indicated in the audit log will be:

- Information that a change in roles has been made based on an application of an automatic rule and which one
- References to the process from which the change has emerged in order to identify that it has occurred within synchronization or saving from the web...

Change of user's roles

An update (adding and removing) of automatically assigned roles within an identity occurs at least in the following cases:

- When a change in the organizational placement is saved.
  - A relation (or relations) of an identity is linked to the organizational structure. If a role (or roles) is assigned to a component in the organizational structure, the user should receive these roles automatically.
- When a change is saved in the "role-organizational structure" link
  - If the role is linked to an organizational structure component, it is necessary to recalculate the roles on identities which are linked by a relation to the given organization structure component
- Automatic assigning of roles to the user does not require an approval and is realized immediately.

Implementation details

- Configuration and assigning of automatic roles is implemented according to the description above
- Approving of the configuration of automatic roles (adding, removing) is configurable, The entire implementation is within processors which can be turned off or replace the workflow definition (see application configuration. Default workflow definition are ready for approving, where the adding and removing of the configuration of an automatic role is approved by the guarantee of the role (the first one wins if there are more), if the guarantee itself is not initiator of the operation (the it does not have to approved and is carried out immediately).
- Editing of an automatic role is not implemented ⇒ the algorithm "drop and create". It can be added in the future.
- Approving of the assigning of a role to an identity according to the configuration of the automatic role is not implemented - it depends on the currently developed functionality of role queries - will be implemented eventually.
- audit track will be implemented in the future
- If an automatic role is deleted, all assigned roles following this role are deleted.
- If a new automatic role is added, it is assigned to all the current identities with an existing CR, which should get the role. The realization is carried out via a Long running task.

- For roles becoming effective at a future time, the account management will be performed when the validity is met.
- Working with CR validity:
  - When changing the contract validity, the changes are projected in the validity of automatically assigned roles
  - Automatic roles with future validity are ready for contracts
  - Automatic roles are not assigned to invalid contracts (in the past or disabled). When changing the validity to the past, the roles are removed directly by event processor. Removing of expired roles and roles with expired contracts is carried out by the Long running tasks (`IdentityRoleExpirationTaskExecutor`, `IdentityContractExpirationTaskExecutor`). Long running tasks are tied to the validity of contracts ⇒ the attribute disabled too. Expired roles and roles with expired, disabled or excluded contracts are not applied at login.
- Recalculating of automatic roles when changing the tree structures (moving the components) have not been addressed yet (coming soon).