

Permissions

Permissions in "Platform stack" is one of the most advanced permissions systems around, allowing you to define very fine grained rights for your Editor-, Anonymous-, Member-, ... Users.

Overview

In the Permission system by default a user does not have access to anything, to give them access, they need to inherit Roles, typically assigned to the User Group they belong to.

Model

Roles

First part of the Permission model is the Roles, and they consist of the following parts:

```
RoleLimitation *- RoleAssignment >- Role -< Policy -*< Limitation
```

- A role assignment can optionally have a limitation, role limitation examples: [SubTreeLimitation](#) or [SectionLimitation](#)
- A role can have several assignments, role example: Editor, Member, ProSubscriber, ..
- A role consists of several policies, policy example: content/read/*, content/edit/* (where * refers to full access, aka no limitation)
- A policy consists optionally of several limitations, limitation example: [ContentTypeLimitation](#), [SectionLimitation](#), [OwnerLimitation](#), ...

Users

Second second part of the model is Users and User Groups:

```
User -*< UserGroup
```

- A user can be member of several user groups, user group examples: Administrator Users, Member Users, ProSubscriber Users

Role assignments

Last part on the permission model is the fact that RoleAssignments can be assigned to both User **or** UserGroup:

```
User - RoleAssignment - UserGroup
```

- A role assignment can be assigned to either a user *or* user group

Best Practice

Best practice is to avoid assigning Roles to Users directly, and instead make sure you model your content (*types, structure, sections, ..*) in a way that can be reflected in generic roles. Besides being much easier to manage and keep on top of security wise, this also makes sure your system performs best. *The more Role assignments and complex policies you add for a given user, the more complex the search/load queries powering the whole CMS will be, as they always take Permissions into account.*

Extensibility

Two parts of the Permissions system is extensible from a programmatic perspective: Policies and Limitations

- **Policies:** Custom policies can be added for use in your own code, custom policy example comment/create
- **Limitations:** You can extend existing policies, and hence extend the permissions of the CMS, example could be adding [SubscriptionLimitation](#) to content/read policy

Further reading: [Limitations reference](#)