

## Base Distribution - Story # 55814

Epic # 55813 (Closed): Access Control Lists

<b>Status:</b>	Closed	<b>Priority:</b>	Should have
<b>Author:</b>	Bastian Waidelich	<b>Category:</b>	
<b>Created:</b>	2014-02-09	<b>Assigned To:</b>	
<b>Updated:</b>	2014-03-24	<b>Due date:</b>	
<b>Subject:</b>	Access restriction for modules		
<b>Description</b>			
<p>Currently it's possible to bind a Neos module to a resource via Settings.yaml:</p> <pre>1TYPO3: 2  Neos: 3    modules: 4      management: 5        label: 'Management' 6        # ... 7        resource: 'TYPO3_Neos_Backend_Module_Management' 8      submodules: 9        workspaces: 10       label: 'Workspaces' 11       # ... 12       resource: 'TYPO3_Neos_Backend_Module_Management_Workspaces'</pre>			
<p>This already allows the integrator to hide modules from certain roles. But the current implementation lacks some features:</p> <ul style="list-style-type: none"><li>- The access restriction is not configurable over the browser</li><li>- Only the complete module can be bound to resources, a more fine-grained ACL would be useful (e.g. allow role x to view the user module but only to manage roles y and z).</li></ul>			
<b>TYPO3 CMS</b>			
<p>In TYPO3 CMS the access control for modules is quite basic:</p> <p>ACL_Modules.png</p> <p>+ Easy to use</p> <ul style="list-style-type: none"><li>- Not fine grained (only complete modules can be included/excluded)</li><li>- The parent module has to be included manually (allowing only "Web&gt;Page" does not work, "Web" needs to be included as well)</li></ul>			

### History

#### #1 - 2014-02-11 20:40 - Rens Admiraal

I discussed this with Christopher on the codesprint, we came to the conclusion that the current role implementation is insufficient (as said with 'not configurable in browser').

If we would somehow try to use the current security framework it would also be really heavy, and regenerating the proxy classes would be a big issue (which would be required if we change the content of the Policy.yaml).

Idea: add something on top which can be used to define functional roles / policies. This way the policies in Policy.yaml define the basic roles known in

the package, more functional roles like "limit access to resources with extension .swf" could then be defined differently. And for those 'functional roles' counts: they can be site specific.

**#2 - 2014-03-24 17:59 - Bastian Waidelich**

- Status changed from New to Closed

Moved to <https://jira.typo3.org/browse/NEOS-27>

## Files

---

ACL_Modules.png	36.8 kB	2014-02-09	Bastian Waidelich
-----------------	---------	------------	-------------------