Overview

After the user is successfully authenticated, authorization is handled by ACLs.

Some ACL checks are performed by C1 Core. For this to work a module must register ACL information within C1 Core. See ACLs on how to do this.

Authorization checks within modules

A lot of the authorization checks are done within C1 Core. But not everything can be checked there. The following must be done within modules.

RPC requests

There are four possible permission types that can be assigned to a RPC method:

1. `read`
2. `write`
3. `event`
4. `isAdmin`

The per RPC method permission is given by the group ACLs assigned to the user in C1 Core. This permission is checked within C1 Core, so there is no need to check this within the module.

For this to work, the module must pass ACL information to C1 Core upon module registration. This ACL information is a JSON defined in the settings (setting `aclInfo`, `acl_info`, `moduleAclInfo` or `module_acl_info`).

REST requests

REST permission types are assigned automatically based on the HTTP method. The association is as follows:

HTTP method	Permission type
`DELETE`	`write`
`GET`	`read`

HTTP method	Permission type
`PATCH`	`write`
`POST`	`write`
`PUT`	`write`

The permission is checked within C1 Core, so these checks are not required within the module.

In addition all requests that require the user to have the administrator role for the module, must be placed under the path `/admin/...`.

All public requests, i. e. requests that don't require authentication, must be placed under the path `/public/...`.

Passing of principal and ACL to modules using libc1-module

With every method call the parameter `resulting_principal` is passed to `rpc_method_callback`. It has the following properties:

Property	Type	Description
`sp`	`String`	The ID of the system provider. This is always set except for `type` `7`.
`sd`	`String`	The ID of the system distributor. This is always set except for `type` `7`, also for system provider users. For module access a system distributor must always be specified.
`bp`	`String`	The ID of the business partner. This is always set except for `type` `7`, also for system provider and system distributor users. For module access a business partner must always be specified.
`id`	`String`	The users's, edge client's or module's ID.
`type`	`Int`	`1` for super users (unused currently), `2` for system provider users, `3` for system distributor users, `4` for business partner users, `5` for end users, `6` for edge clients and `7` for modules.

Alternatively the property `resultingPrincipal` of the verified metadata `Struct` can be used.

The ACL is passed to `rpc_method_callback` as the parameter `user_acl`.

Additional security settings

There are a few generic security settings that completely deny access when set to `false`:

Property	Type	Default	Description
`allow_business_partner_user_access`	`Boolean`	`true`	When set to `false`, business partner users are denied access.
`allow_end_user_access`	`Boolean`	`false`	When set to `false`, end users are denied access.
`allow_home_client_access`	`Boolean`	`false`	When set to `false`, edge clients are denied access.
`system_provider_module`	`Boolean`	`false`	When set to `true`, only system provider users or modules with at least system provider level are granted access.

Passing of principal and ACL to modules using C1 Module Proxy

When using C1 Module Proxy, the property `resultingPrincipal` is appended to the verified metadata `Struct`. It has the following properties:

Property	Type	Description
`sp`	`String`	The ID of the system provider. This is always set except for `type` `7`.
`sd`	`String`	The ID of the system distributor. This is always set except for `type` `7`, also for system provider users. For module access a system distributor must always be specified.
`bp`	`String`	The ID of the business partner. This is always set except for `type` `7`, also for system provider and system distributor users. For module access a business partner must always be specified.
`id`	`String`	The users's, edge client's or module's ID.
`type`	`Int`	`1` for super users (unused currently), `2` for system provider users, `3` for system distributor users, `4` for business partner users, `5` for end users,

Property	Type	Description
		`6` for edge clients, `7` for modules and `8` for events from the event broker.

Additional security settings

There are a few generic security settings that completely deny access when set to `false`:

Property	Type	Default	Description
`allowBusinessPartnerUserAccess`	`Boolean`	`true`	When set to `false`, business partner users are denied access.
`allowEndUserAccess`	`Boolean`	`false`	When set to `false`, end users are denied access.
`allowEdgeClientAccess`	`Boolean`	`false`	When set to `false`, edge clients are denied access.
`systemProviderModule`	`Boolean`	`false`	When set to `true`, only system provider users or modules with at least system provider level are granted access.

Passing of principal and ACL to modules using nodejs-c1-module

When using nodejs-c1-module, the property `resultingPrincipal` is passed as a parameter to the callback methods. It has the following properties:

Property	Type	Description
`sp`	`String`	The ID of the system provider. This is always set except for `type` `7`.
`sd`	`String`	The ID of the system distributor. This is always set except for `type` `7`, also for system provider users. For module access a system distributor must always be specified.
`bp`	`String`	The ID of the business partner. This is always set except for `type` `7`, also for system provider and system distributor users. For module access a business partner must always be specified.

Property	Type	Description
`id`	`String`	The users's, edge client's or module's ID.
`type`	`String`	`su` for super users (unused currently), `sp` for system provider users, `sd` for system distributor users, `bp` for business partner users, `eu` for end users, `ec` for edge clients, `m` for modules and `e` for events from the event broker.
`rawType`	`Int`	`1` for super users (unused currently), `2` for system provider users, `3` for system distributor users, `4` for business partner users, `5` for end users, `6` for edge clients, `7` for modules and `8` for events from the event broker.

Additional security settings

There are a few generic security settings that completely deny access when set to `false`:

Property	Type	Default	Description
`allowBusinessPartnerUserAccess`	`Boolean`	`true`	When set to `false`, business partner users are denied access.
`allowEndUserAccess`	`Boolean`	`false`	When set to `false`, end users are denied access.
`allowEdgeClientAccess`	`Boolean`	`false`	When set to `false`, edge clients are denied access.
`systemProviderModule`	`Boolean`	`false`	When set to `true`, only system provider users or modules with at least system provider level are granted access.

Mandatory checks to do within a module

Checks depending on user type

CALLER IS A USER (TYPES `SU`, `SP`, `SD`, `BP`, `EU`/RAW TYPES 1 TO 5)

1. An end-user (`type` `eu`) must only be allowed access to its own data.
2. All other users (`type` `su`, `sp`, `sd` and `bp`) must only be allowed access to data of the accessed business partner.

CALLER IS AN EDGE CLIENT (TYPE `EC`/RAW TYPE 6)

1. An edge client must only be allowed access to data of itself or an associated user. See below on how to determine the associated users.

CALLER IS A MODULE (TYPE `M`/RAW TYPE 7)

1. When the module is associated to a principal (`sp`, `sd` or `bp` are not `0`), it must only be allowed access to this principal.
2. When the module is not associated to a principal it is trusted and can be allowed full access.

CALLER IS THE EVENT BROKER (TYPE `E`/RAW TYPE 8)

Events are a special case. For events it depends on the source if you can implicitly trust the event or not. Depending on the source, the rules above apply. So:

1. For events coming from edge clients, the rules of type `ec` apply (see above).
2. For events from modules the rules of type `m` apply (see above).
3. For events from users the rules of types `su`, `sp`, `sd`, `bp` and `eu` apply (see above).

Checks depending on asset access

When `assetAccess` is defined in the ACL and the user does not have access to all assets used by the module, the module must:

1. either deny access to the user completely
2. or only allow access to the assets allowed by the ACL.

Checks depending on role access

When `roleAccess` is defined in the ACL and the user does not have access to all roles used by the module, the module must:

1. either deny access to the user completely
2. or only allow access to the roles allowed by the ACL.

> **Warning**
>
> The checks above are mandatory. Not doing them introduces a huge security risk.

Determining associated users

In a `moduleMethod` call

Only module clients (not modules) receive `moduleMethod` calls. I. e. the method name is set to `moduleMethod` - in modules the RPC method is called without the `moduleMethod` wrapper. In this case, the users associated to an edge client are placed in parameter with index `6`. See the section about inter-module RPC usage for more information.

In all other RPC method calls

For all other modules, the users associated to an edge client can be read from the property `homeClientUsers` of the verified metadata.