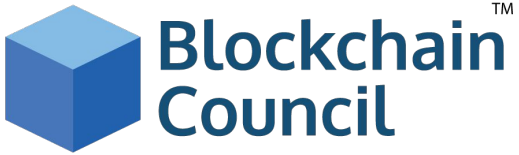

Certified Hyperledger Developer

Hyperledger Composer Access Control Language

Access Control Language

- Hyperledger Composer uses an access control language (ACL) which provides declarative access control over the elements of the domain model.
- ACL rules are used to determine which users/roles are permitted to create, read, update or delete elements in a business network's domain model.
- Hyperledger Composer offers two types of control: access control for resources within a business network (business access control) and access control for network administrative changes (network access control).
- Business access control and network access control are both defined in the access control file (.acl) for a business network.

Types of ACL rules

- **Simple ACL rule:** Simple rules are used to control access to a namespace or asset by a participant type or participant instance.
    - For example, a simple ACL rule can state that any instance of a particular type can perform what type of operations on instances of a particular asset.
- **Conditional ACL rules** : It can introduce variable bindings for the participant and the resource being accessed, and a Boolean JavaScript expression, which, when true, can either ALLOW or DENY access to the resource by the participant.

How to write a ACL rule?

- An ACL rule definition requires these arguments:
    - **Resource** defines the things that the ACL rule applies to. This can be a class, all classes within a namespace, or all classes under a namespace. It can also be an instance of a class.
    - **Operation** identifies the action that the rule governs. Four actions are supported: CREATE, READ, UPDATE, and DELETE.
    - **Participant** defines the person or entity that has submitted a transaction for processing.
    - **Transaction** defines the transaction that the participant must have submitted in order to perform the specified operation against the specified resource.
    - **Condition** is a Boolean JavaScript expression over bound variables. Any JavaScript expression that is legal within an if(...) expression may be used here.
    - **Action** identifies the action of the rule. It must be one of: ALLOW, DENY.

Sample ACL rule

```
rule SimpleRule {
    description: "Description of the ACL rule"
    participant: "org.example.SampleParticipant"
    operation: ALL
    resource: "org.example.SampleAsset"
    action: ALLOW
}
```

What access control affects?

- Composer Network
- Composer Identity
- Composer Participants

How to grant Network Access Control?

- Access control for a business network is defined by an ordered set of ACL rules.
- The rules are evaluated in order, and the first rule whose condition matches determines whether access is granted or denied. If no rule match then access is denied.
- ACL rules are defined in a file called permissions.acl in the root of the business network. If this file is missing from the business network then all access is permitted.
- Network access is granted using the system namespace.
- The system namespace is always **org.hyperledger.composer.system.Network** for network access, and **org.hyperledger.composer.system** for all access.

Access Control Rule for Network Access

- The following access control rules gives the NetworkControl participant the authority to use all operations with network commands:

```
rule NetworkControlPermission {
  description:  "NetworkControl can access network commands"
  participant: "org.example.basic.NetworkControl"
  operation: ALL
  resource: "org.hyperledger.composer.system.Network"
  action: ALLOW
}
```

Access Control Rule for System Access

- The following access control rule will give all participants access to all operations and commands in the business network, including network access and business access.

```
rule AllAccess {
  description: "AllAccess - grant everything to everybody"
  participant: "org.hyperledger.composer.system.Participant"
  operation: ALL
  resource: "org.hyperledger.composer.system.**"
  action: ALLOW
}
```

THANK YOU!

Any questions?

You can mail us at

hello@blockchain-council.org