

Eluvio Content Fabric V2 Spec

2022

0 Definitions

Node A server which stores and serves parts.

Provider An individual or organization which owns, secures, and operates nodes.

Tenant An individual or organization which owns content.

Content A piece of data which is owned by a tenant.

Space A group of providers, provider nodes, and tenants, where providers agree to run nodes which store content owned by a tenant according to a common set of rules.

Part A part is a sequence of bytes stored in the space, referenced by its hash.

Content Object Version A collection of parts created by a tenant, referenced by its hash.

Content Object A collection of versions.

Library A 'folder' of content objects owned by a tenant with a permission structure that determines who within a tenancy is able to create/modify/delete content objects and content object versions.

KMS A tenant-owned server which holds keys for encrypting/decrypting content which the tenant stores in the space.

Blockchain A distributed ledger responsible for orchestrating cooperation between providers, the exchange of value between providers and tenants, and governance that determines the rules of a space.

The following entities are defined by fixed length identifiers as follows:

Entity	Identifier	Type
Node	ID_{node}	Fixed length byte string
Space	ID_{space}	Fixed length byte string
Content Object	ID_{conq}	Fixed length byte string
Content Object Version	$ID_{version}$	Fixed length byte string
KMS	ID_{kms}	Fixed length byte string
Library	ID_{lib}	Fixed length byte string

1 Space

The space functions as the top level governance structure of the fabric that orchestrates how providers cooperate to serve tenant data. Governance is TBD.

1.1 Space rules

Provider Bond An amount, $BOND_{prov}$, of currency each provider must lock up in order to participate within the space. Funds can be slashed from here if a provider misbehaves.

Tenant Bond An amount, $BOND_{ten}$, of currency each tenant must lock up in order to participate within the space. Funds can be slashed from here if a tenant misbehaves.

SLAs Specifications for availability requirements provider nodes must have.

Partition number The partitioning constant for part storage

2 Provider

A provider is a logical group of nodes within a space, and a permission structure for keys. When a provider is created, it is bootstrapped with an admin key k_{admin} that has total control of the provider.

2.1 Provider Permissions

Provider keys have the following permission levels, from most to least privileged

1. $PERM_{root}$ can do everything
2. $PERM_{admin}$ can add/remove nodes, bill tenants
3. $PERM_{node}$ can co-author versions with tenants and mark nodes as no longer pending

2.2 Blockchain actions

In addition to setting permissions on keys, we have the following actions

CreateProvider($k_{\text{origin}} = k_{\text{root}}, \text{ID}_{\text{space}}, \text{ID}_{\text{prov}}$) Creates the provider

- TODO: check governance to see whether origin can create a provider
- Creates ID_{prov} and sets its space to ID_{space}
- Sets k_{root} as the creator of ID_{prov}
- Sets k_{root} as a key for ID_{prov} with level $\text{PERM}_{\text{root}}$
- Bonds $\text{BOND}_{\text{prov}}$ from k_{root} to the space under ID_{prov}

AddNode($k_{\text{origin}}, \text{ID}_{\text{prov}}, \text{ID}_{\text{node}}, k_{\text{node}}, \text{LOC}_{\text{node}}$) adds a node

- Checks that k_{origin} has permission $\text{PERM}_{\text{admin}}$ or above for ID_{prov} .
- Creates a node ID_{node} with locator LOC_{node}
- Registers k_{node} to ID_{prov} with permission level $\text{PERM}_{\text{node}}$ ¹
- Marks the node as pending while it syncs up parts with the rest of the space

ConfirmNode($k_{\text{origin}}, \text{ID}_{\text{prov}}, \text{ID}_{\text{node}}$) marks a node as no longer pending

- Checks that k_{origin} has permissions $\text{PERM}_{\text{node}}$ or above for ID_{prov}
- Sets ID_{node} to no longer pending

RemoveNode($k_{\text{origin}}, \text{ID}_{\text{prov}}, \text{ID}_{\text{node}}$) removes a node

- Checks that k_{origin} has permissions $\text{PERM}_{\text{admin}}$ or above for ID_{prov}
- Removes all ID_{node} information from the space and provider

BillTenant TODO

3 Tenant

A tenant is an owner of content, responsible for providing a service which manages keys which encrypt content to providers.

3.1 Tenant Permissions

Tenant keys have the following permission levels, from most to least privileged

1. $\text{PERM}_{\text{root}}$ can add/remove admins
2. $\text{PERM}_{\text{admin}}$ can add/remove kmses, add/remove users to/from libraries
3. PERM_{kms} can co-author content object versions with provider nodes

¹Should this error if the key already exists within the permissions scheme?

3.2 Blockchain Actions

CreateTenant($k_{\text{origin}} = k_{\text{root}}, \text{ID}_{\text{space}}, \text{ID}_{\text{tenant}}$) creates a tenancy

- TODO: check governance to see whether origin can create a tenant
- Creates $\text{ID}_{\text{tenant}}$ and sets its space to ID_{space}
- Sets k_{root} as the creator of $\text{ID}_{\text{tenant}}$
- Sets k_{root} as a key for $\text{ID}_{\text{tenant}}$ with level $\text{PERM}_{\text{root}}$
- Bonds BOND_{ten} from k_{root} to the space under $\text{ID}_{\text{tenant}}$

AddKMS($k_{\text{origin}}, \text{ID}_{\text{tenant}}, \text{ID}_{\text{kms}}, k_{\text{kms}}, \text{LOC}_{\text{kms}}$) creates a kms

- Checks that k_{origin} has permission $\text{PERM}_{\text{admin}}$ or above for $\text{ID}_{\text{tenant}}$.
- Creates a node ID_{kms} with locator LOC_{kms}
- Registers k_{kms} to $\text{ID}_{\text{tenant}}$ with permission level PERM_{kms}

RemoveKMS($k_{\text{origin}}, \text{ID}_{\text{tenant}}, \text{ID}_{\text{kms}}$) removes a node

- Checks that k_{origin} has permissions $\text{PERM}_{\text{admin}}$ or above for $\text{ID}_{\text{tenant}}$
- Removes all ID_{kms} information from the space and tenancy

TODO: Remove Tenant, Top up billing balance