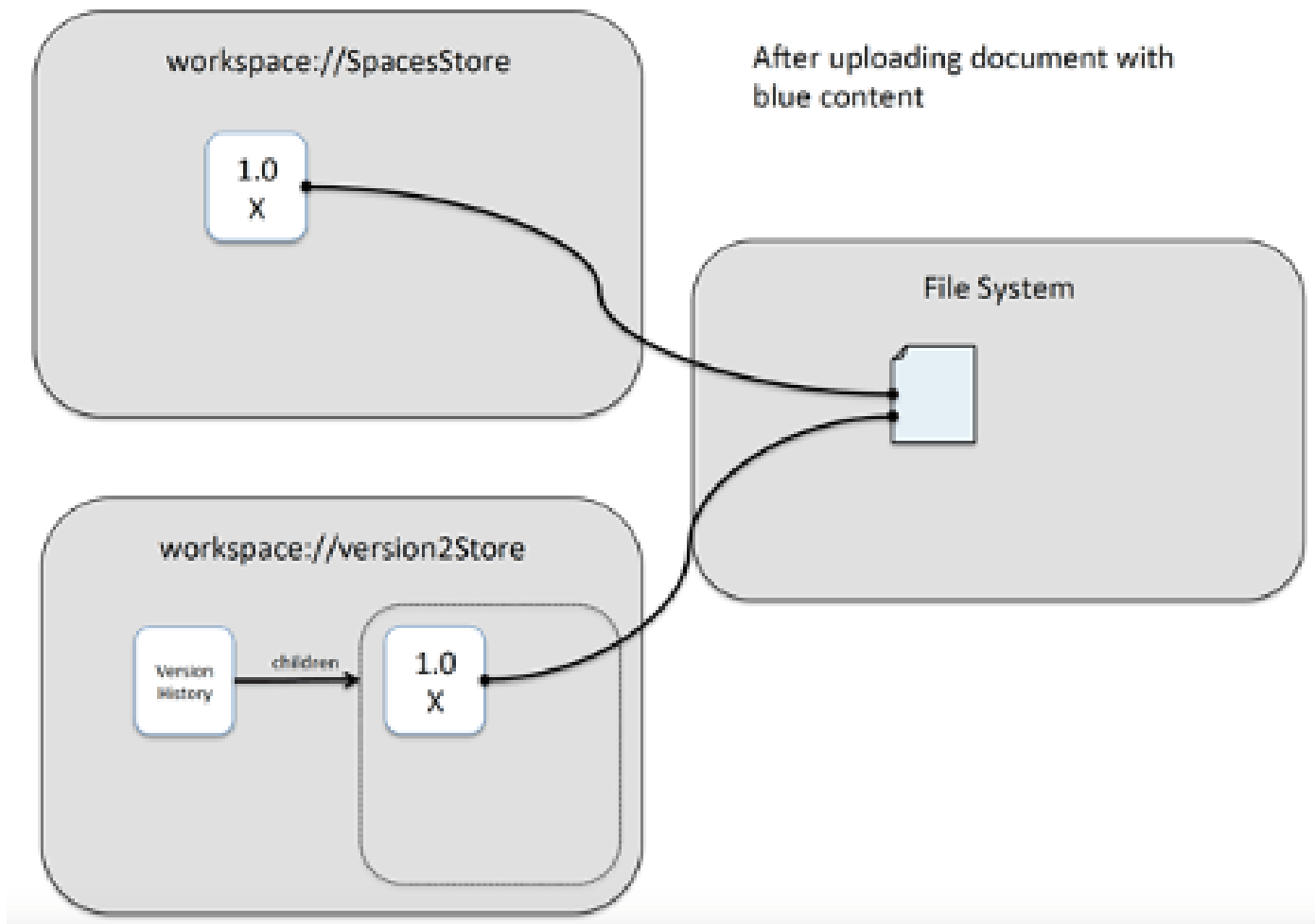


Alfresco Versioning allows you to track content history. When an Alfresco content node is versionable (has the versionable aspect), the version history is started. The first version of the content is the content that exists at the time of versioning.

Nodes that have the cm:versionable aspect are known as **versionable** nodes and they contain references on multiple contentStores (version2Store and spacesStore).



We can assume that a versionable node has “at least” 2 copies of its binary content, one in workspace://SpacesStore and other in workspace://version2Store. If we think about storage space, a versionable node that carries a pdf file of 5MB will actually be occupying “at least” 10 MB (2x 5MB) from the moment of its upload.

Inside the Alfresco database this is also true, meaning that a versionable node is actually stored as “at least” 2 nodes, each with its own metadata fields and with references in tables alf_node and alf_node_properties.

Important Considerations

- Upload in Alfresco Share will **automatically** add the versionable aspect
- Creation of a document via CMIS (5.0/5.1/5.2) **automatically** adds the versionable aspect
- Creation of a document via REST API **automatically** adds the versionable aspect
- Associations to versionable nodes are stored referencing the live node (not the version)
- Versions cannot have different security.
- Versions cannot be searched.
- Versions cannot have their own associations.

Versionable node initial properties

When a node is uploaded and gets the versionable aspect, it inherits the following properties

cm:autoVersion => true – Indicates that the cm:versionable aspect is applied and initial snap shot of the node was taken.

cm:initialVersion => true – Indicates that whenever a change is made to the “content of a versionable node” a new version will be created.

cm:autoVersionOnUpdateProps => false – Indicates that version number will not be incremented when properties are updated on a node.

The versionable aspect contains information about the current version that the node relates to and is required by the version service when working with a node.

Default configuration for versioning

The Alfresco default configuration for the repository versioning is the following

- **version.store.enableAutoVersioning=true**
- **version.store.initialVersion=true**
- **version.store.enableAutoVersionOnUpdateProps=false**


Long Term Impact of uncontrolled Version Policies

Long living Alfresco repositories that have been up and running with the default versioning configuration (or with uncontrolled version policies) are sometimes impacted by having a huge number of un-necessary versions.

This contributes to having a bigger database and also bigger storage necessities, resulting on a slower repository and increased complexity while executing normal database maintenance tasks such as:

- **Index Defragmentation / Index Optimisation**
- **Statistics Rebuild**
- **Caches Rebuild**

During my consulting practice, I've found several customers that were simply unable to execute those operations due to the time necessary for them to complete. This is common on databases that have billions of rows in the main alfresco tables (alf_node, alf_node_properties, alf_node_assoc, alf_prop_value). A deeper analysis to customers that struggle with huge databases normally shows that they have more versions than the actual nodes managed on workspace://spacesStore. Most of those customers do not even make use of any versions of their documents.

 Home Database Size Large Folders Large Transactions Activities Feed Workflows Archived Nodes List Nodes... Locked Resources Authorities Self Memory JMX Properties Export Report	
<h3>Nodes by Store</h3>	
Store	Nodes Count
workspace://version2Store	112647040
workspace://SpacesStore	63608722
archive://SpacesStore	336320
system://system	14
user://alfrescoUserStore	8
workspace://lightWeightVersionStore	1

In the example above, we can see that there are twice as much versions than live nodes, showing that the repository may have been accumulating un-necessary versions.

Reducing the size of the Alfresco database can have big impact on stability and transactional throughput. One mechanism to reduce the size of the database by pruning un-used versions but one must understand the Alfresco node lifecycle.