Grouping Objects

KMIP Face to Face 2020
Context

• We have the **Object Group** attribute
• It is a TextString
• It has no defined core semantics

A Managed Object MAY be part of a group of objects. An object MAY belong to more than one group of objects. To assign an object to a group of objects, the object group name SHOULD be set into this attribute.
Context

• KMIP-1.x
  – All Objects can have 0..n Object Group attributes

• KMIP-2.0
  – Query can specific list of Object Group values
  – Object Defaults – reference Object Groups
  – Constraints – reference Object Groups
  – Right – reference Object Groups
History

- Some servers support grouping of objects
  - Server-specific semantics enforced
- Some servers have no concept of grouping of objects
  - It is just an arbitrary attribute with no special handling
- Some servers limit behaviour
  - Only one Object Group
  - Object Group is mandatory
  - Object Group has a default value
  - Group members must be homogeneous
  - Object Groups must be pre-defined
History (cont)

• Semantics too loose for the concept of “group of objects”
• Adding an object to a group
  – May have to pre-exist
• Removing an object from a group
  – May not be permitted
• No standard way to manage groups
• Interoperability issues
Group Automation

• **Automation** requires
  – creation of groups
  – destruction of groups
  – enumeration of groups

• Add a **System Object** of type **Group**

• Replace **Object Group** with **Group**
  – Reference or **NameReference**

• Add **Link Type**\(^*\) of **Group Link**\(^*\)
  – Or new **Group Link** attribute
Solution 1

• Attributes for **Group System Object**
  – *Unique Identifier*
  – *Short Unique Identifier*
  – *Object Class*
  – *Subset of Date attributes*
  – *Comment & Description*
  – *Vendor Attributes*
  – Name
  – Object Defaults*
  – Rights*
Example – Create Group

<BatchItem>
  <Operation ... value="CreateGroup"/>
  <RequestPayload>
    <Name type="TextString" value="Hot Keys"/>
  </RequestPayload>
</BatchItem>
Example – Destroy Group

<BatchItem>
  <Operation ... value="Destroy"/>
  <RequestPayload>
    <UniqueIdentifier type="TextString"
      value="0123456789..."/>
  </RequestPayload>
</BatchItem>
Example – Enumerate Groups

<BatchItem>
  <Operation type="Enumeration" value="Locate"/>
  <RequestPayload>
    <ObjectClassMask type="Integer" value="SystemObjects"/>
    <Attributes>
      <ObjectType type="Enumeration" value="Group"/>
    </Attributes>
  </RequestPayload>
</BatchItem>
Historical Oddity

• KMIP 1.1 added Locate

When the Object Group attribute and the Object Group Member flag are specified in the request, and the value specified for Object Group Member is ‘Group Member Fresh’, matching candidate objects SHALL be fresh objects from the object group. If there are no more fresh objects in the group, the server MAY choose to generate a new object on-the-fly, based on server policy. If the value specified for Object Group Member is ‘Group Member Default’, the server locates the default object as defined by server policy.
Solution 2

- Delete **Object Group Member** concept
  - Side-effect on Locate
  - Variable side-effect
- The first part (find Fresh objects only) does not require Object Group Member
- The second part (**create on-the-fly**) should be handled differently
Conclusion

• Enable group life-cycle to be managed
• Interoperable