

# Provisioning Models

The [access control models](#) described in this guide all assume some mechanism to communicate Grouper group and membership changes to target services or an intermediary like an LDAP based enterprise directory service. Provisioning may be set up to keep various groups in sync with target systems, translate a group membership to an eduPersonEntitlement value, or create and keep remote identity records up to date.

[Grouper provisioning](#) mechanisms broadly fall into several categories:

1. **"Direct from Grouper to target service"** covers Grouper specific components and plugins for various targets such as [AD/LDAP](#), [Duo](#), etc. Grouper contains a change log and provisioning engine for loosely coupled connections to external systems.
  - a. Grouper has a [built-in provisioning framework](#).
  - b. Interfaces could be implemented for custom provisioners (provisioning interface or change log consumer)
2. **"Message queue based delivery"** relies on a message queue infrastructure to communicate changes to appropriate provisioning components. In this model the logic for communicating with the external system would not be executed / managed / monitored / audited inside of Grouper
3. **External systems** can use web services, LDAP, or SAML to pull data from Grouper into their data repository.
4. **A non-Grouper provisioning engine** can get data from Grouper and provision it to the target system. Grouper can provision the middleware provisioning system (e.g. MidPoint) directly, or with a message queue, or it can pull from LDAP / SQL / WS.

Group and membership changes are provisioned to target services with two main strategies:

1. **Full-sync batch scheduled provisioning** looks at the source and the target and fully synchronizes the data
2. **Incremental near real-time provisioning** looks at the change log to send focused events to the target.

It is best to do both full and incremental provisioning if possible. The full and incremental sync should not run at the same time (they should wait until the other is done).

Whether you use one or the other, or both models, largely depends on your specific situation and provisioning targets.

Previous: [Access Control Models](#)

Next: [Operational Considerations](#)