

Includes SQL Server Code Named 'Denali'

Permission Syntax

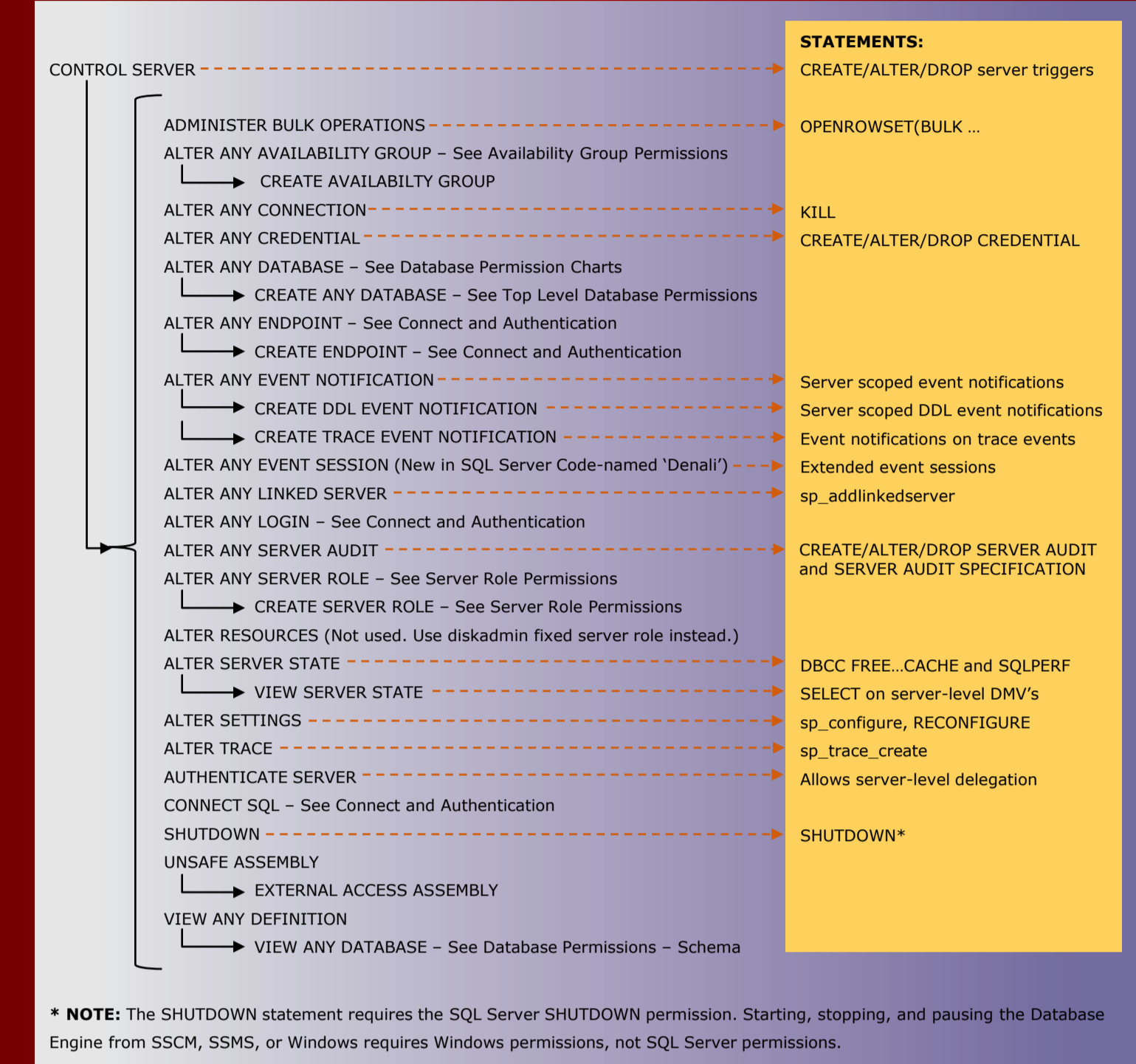
Most permission statements have the format :
AUTHORIZATION PERMISSION ON SECURABLE::NAME TO PRINCIPAL
 AUTHORIZATION must be GRANT, REVOKE or DENY.
 PERMISSION is listed in the charts below.
 ON SECURABLE::NAME is the server, server object, database, or database object and its name. Some permissions do not require ON SECURABLE::NAME.
 PRINCIPAL is the login, user, or role which receives or loses the permission. Grant permissions to roles whenever possible.
 Sample grant statement: GRANT UPDATE ON OBJECT::Production.Parts TO PartsTeam
 Denying a permission at any level, overrides a related grant.
 To remove a previously granted permission, use REVOKE; not DENY.

How to Read this Chart

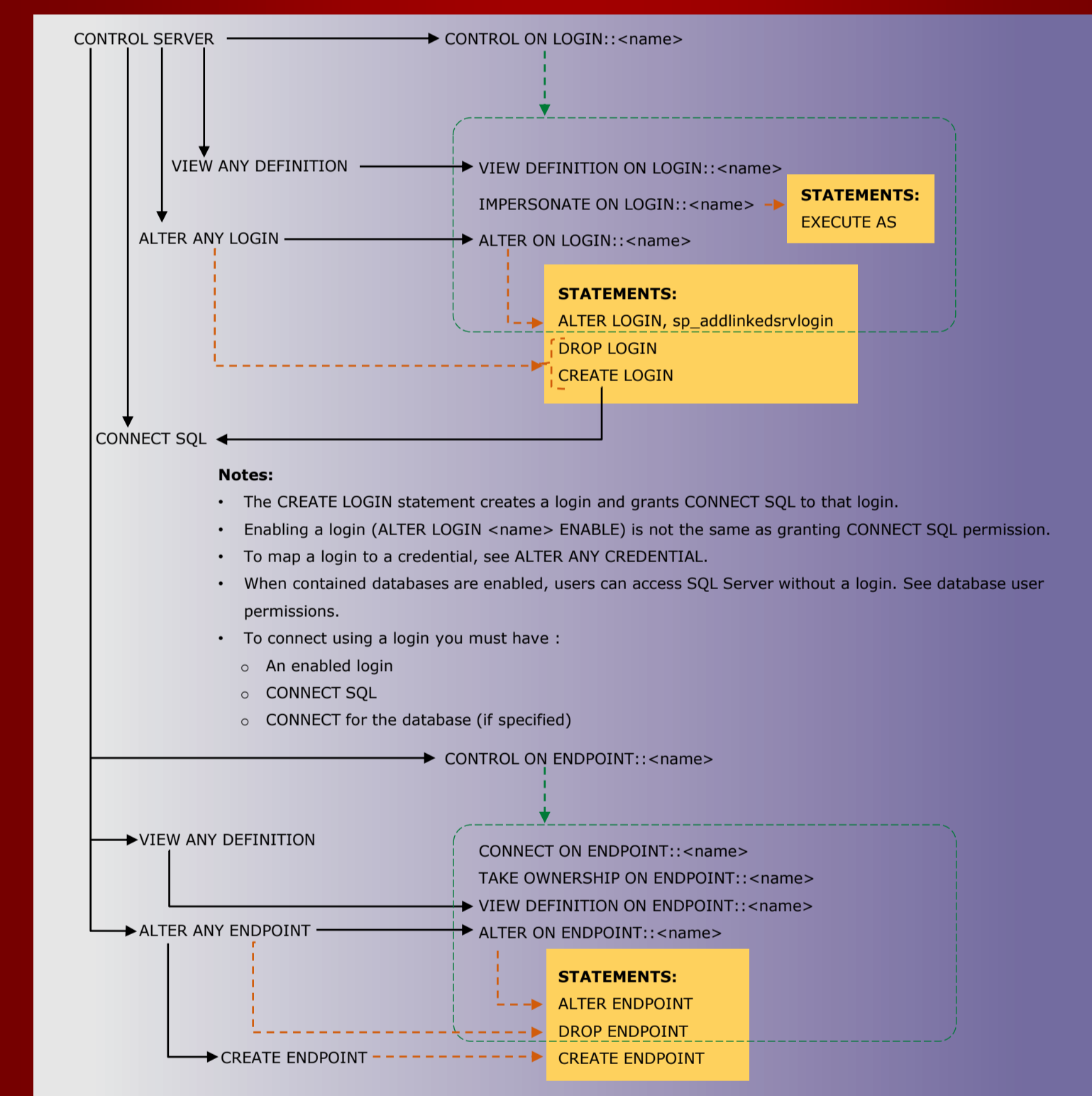
Most of the more granular permissions are included in more than one higher level scope permission. So permissions can be inherited from more than one type of higher scope.
 Black, green, and blue arrows and boxes point to subordinate permissions that are included in the scope of higher level permission.
 Brown arrows and boxes indicate some of the statements that can use the permission.

Server Level Permissions

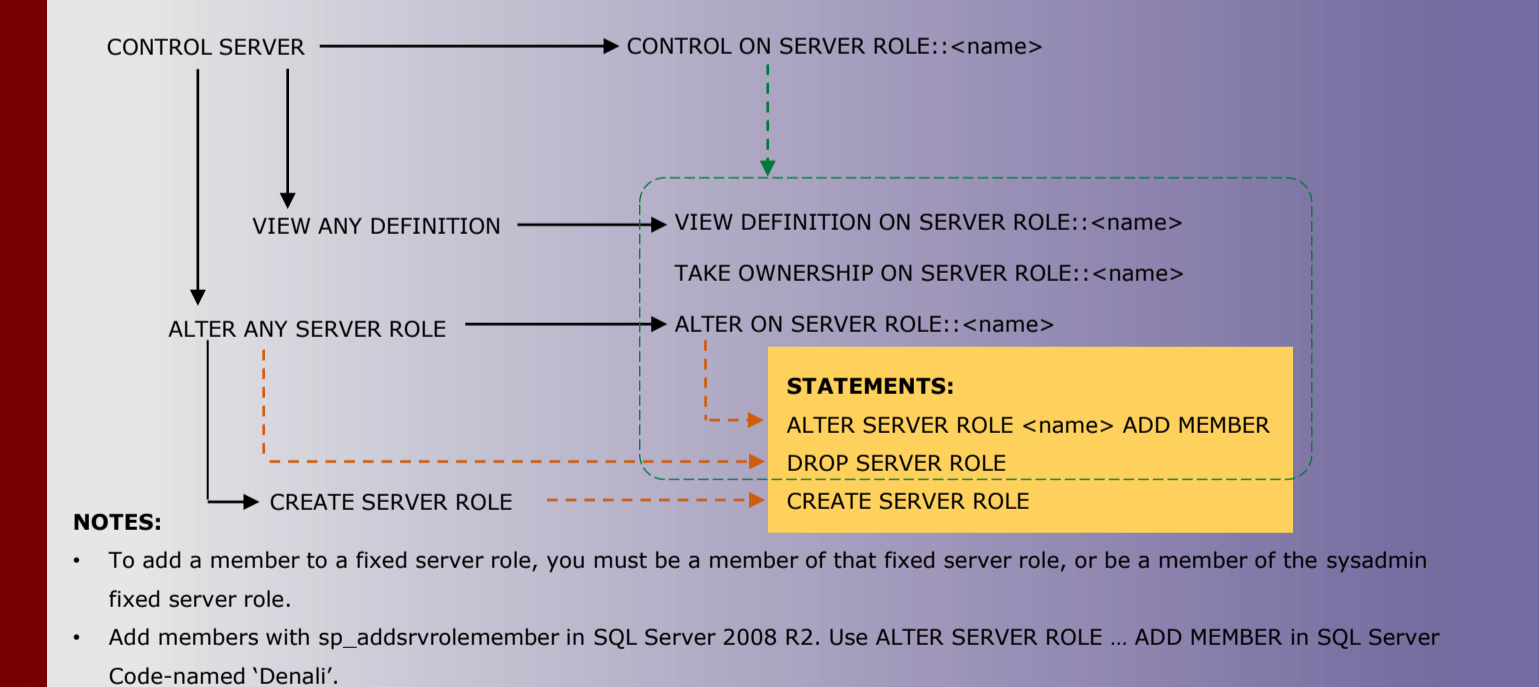
Top Level Server Permissions



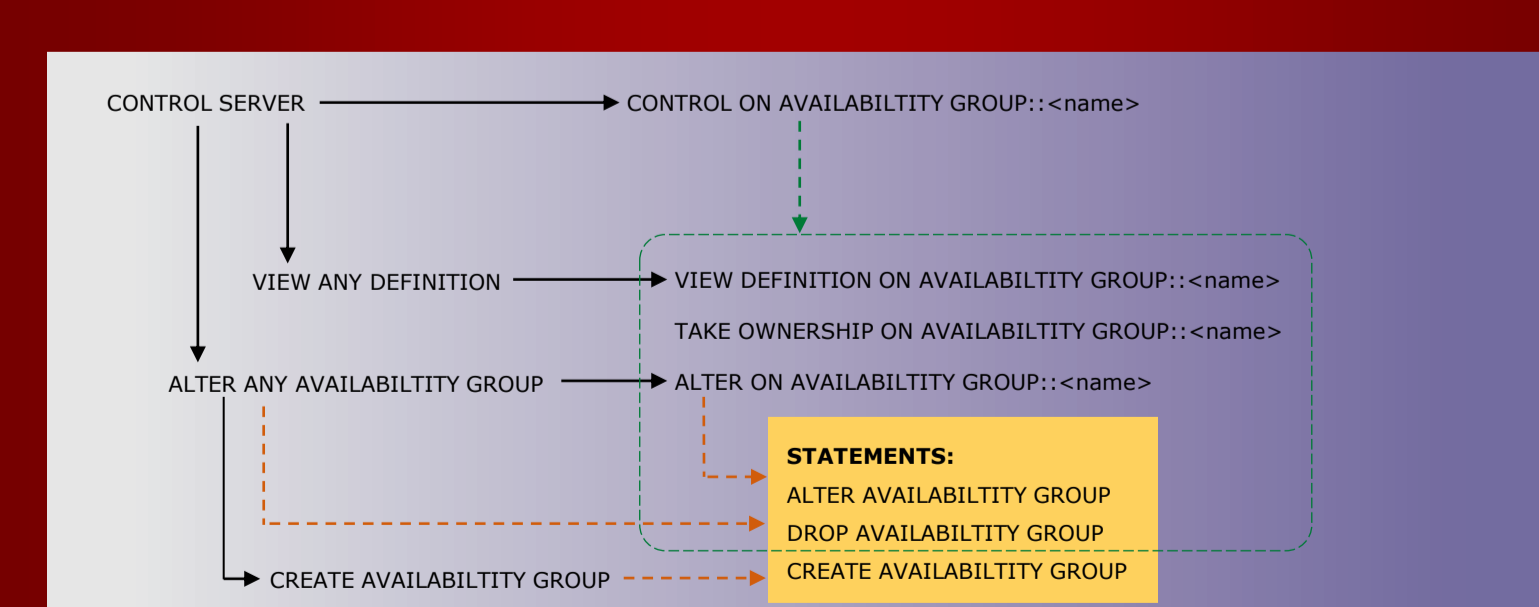
Connect and Authentication - Server Permissions



Server Role Permissions (New in SQL Server Code-named 'Denali')

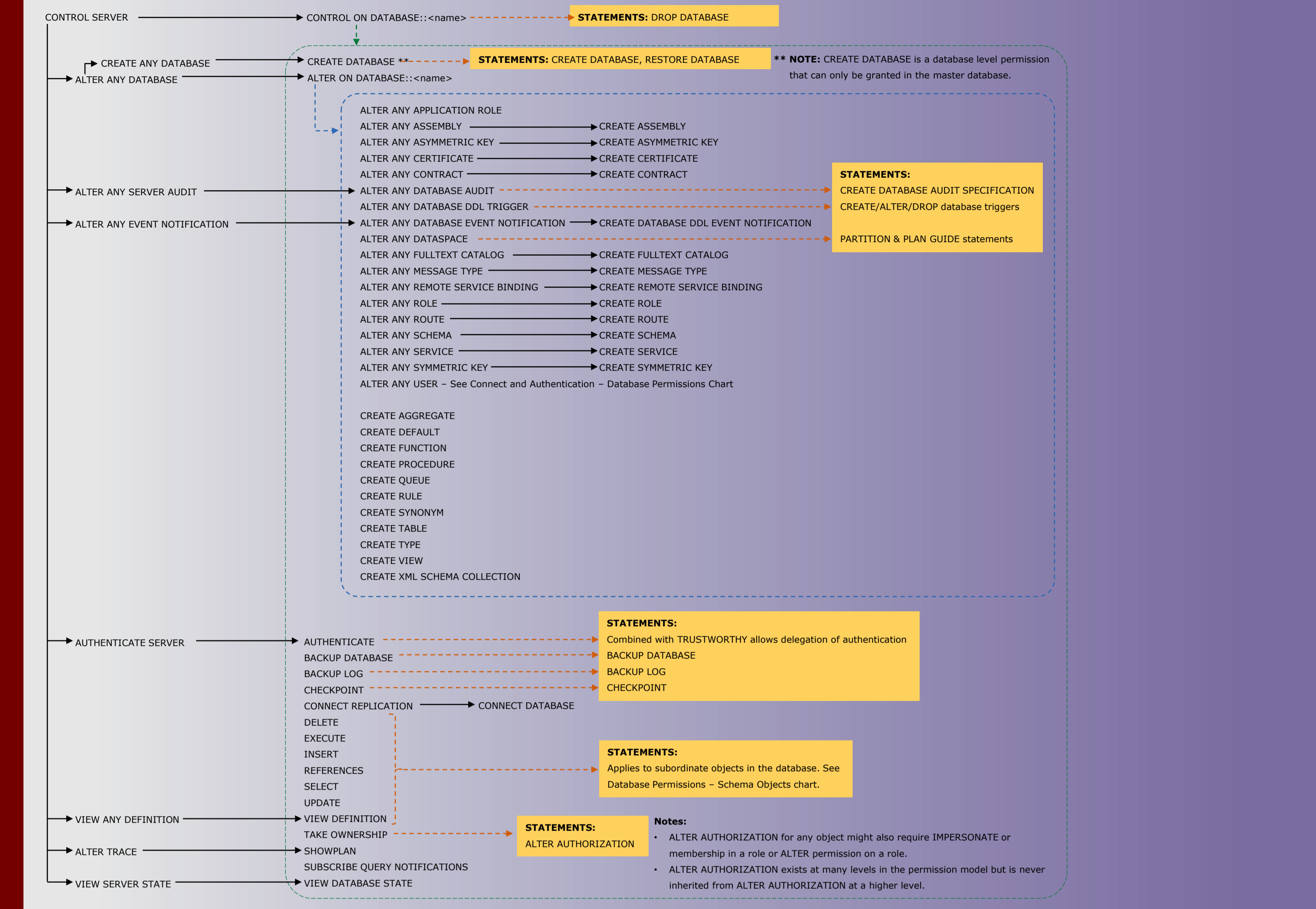


Availability Group Permissions (New in SQL Server Code-named 'Denali')

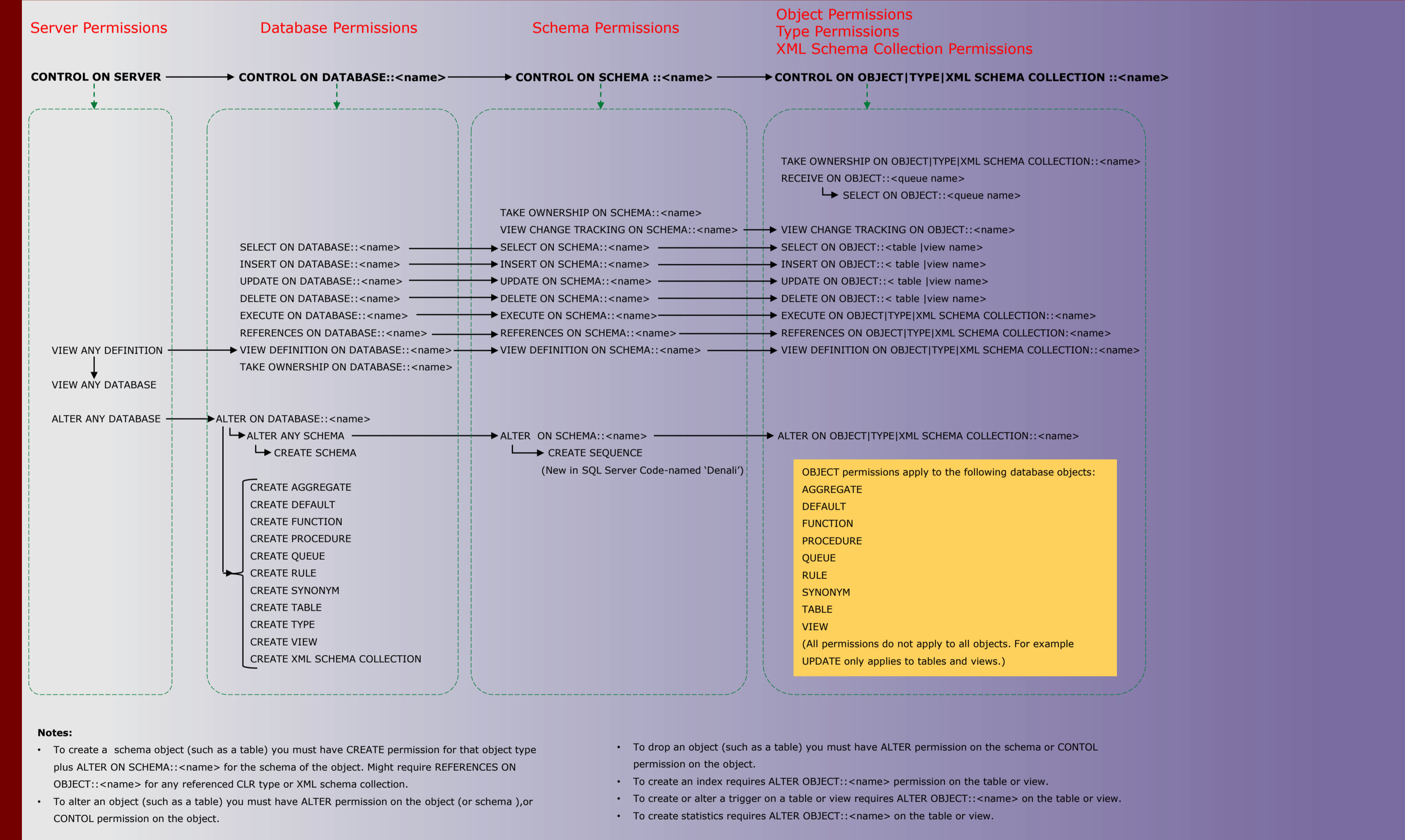


Database Level Permissions

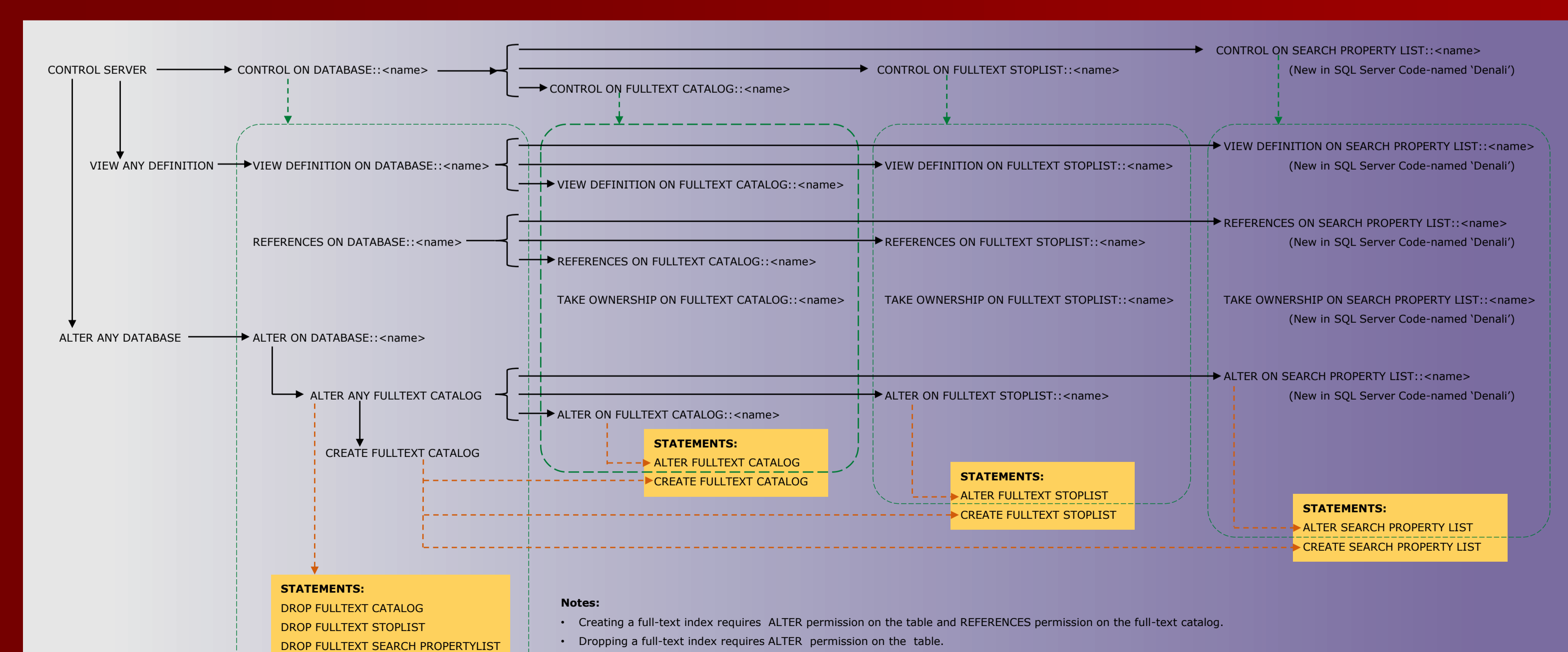
Top Level Database Permissions



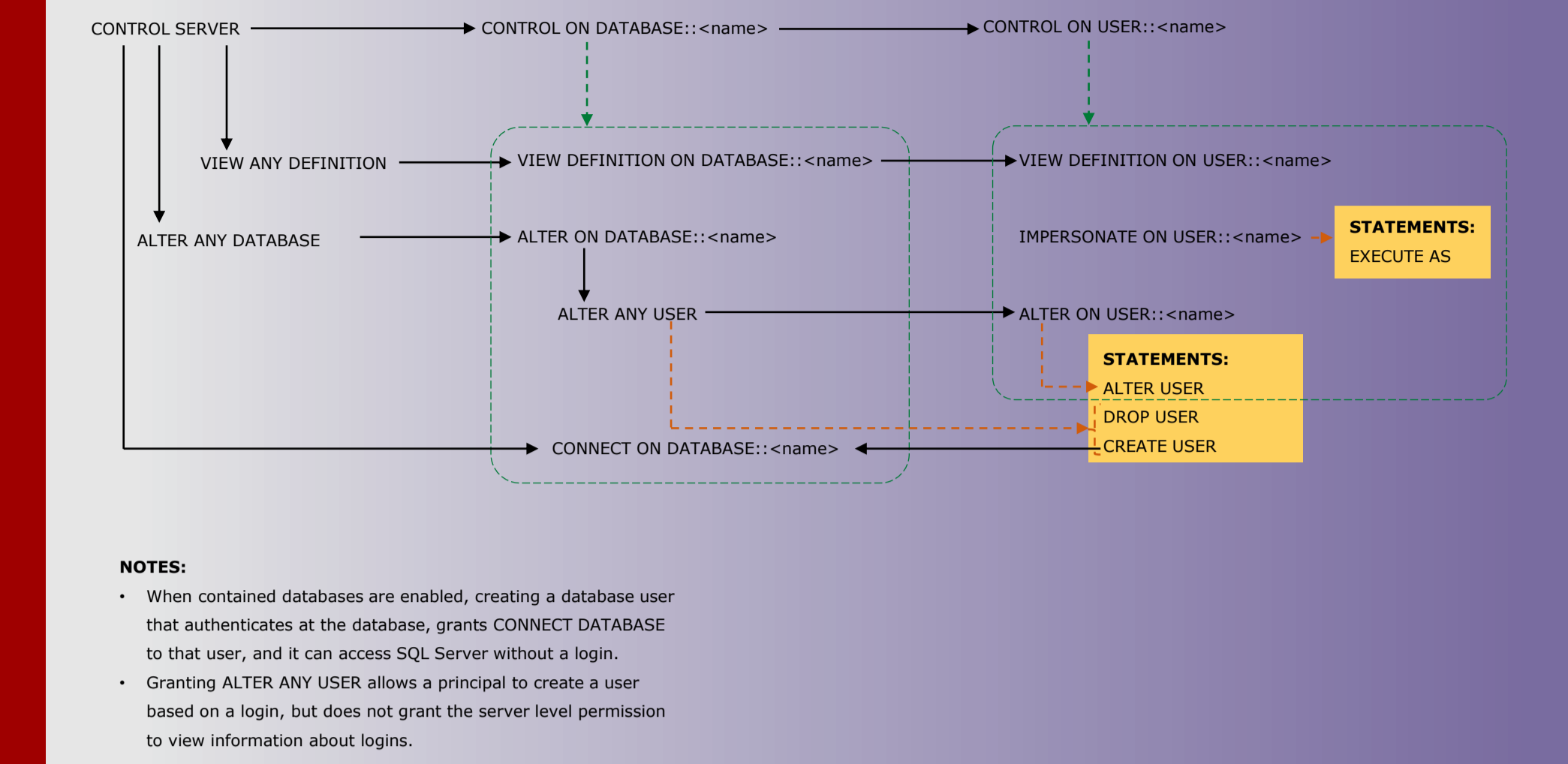
Database Permissions - Schema Objects



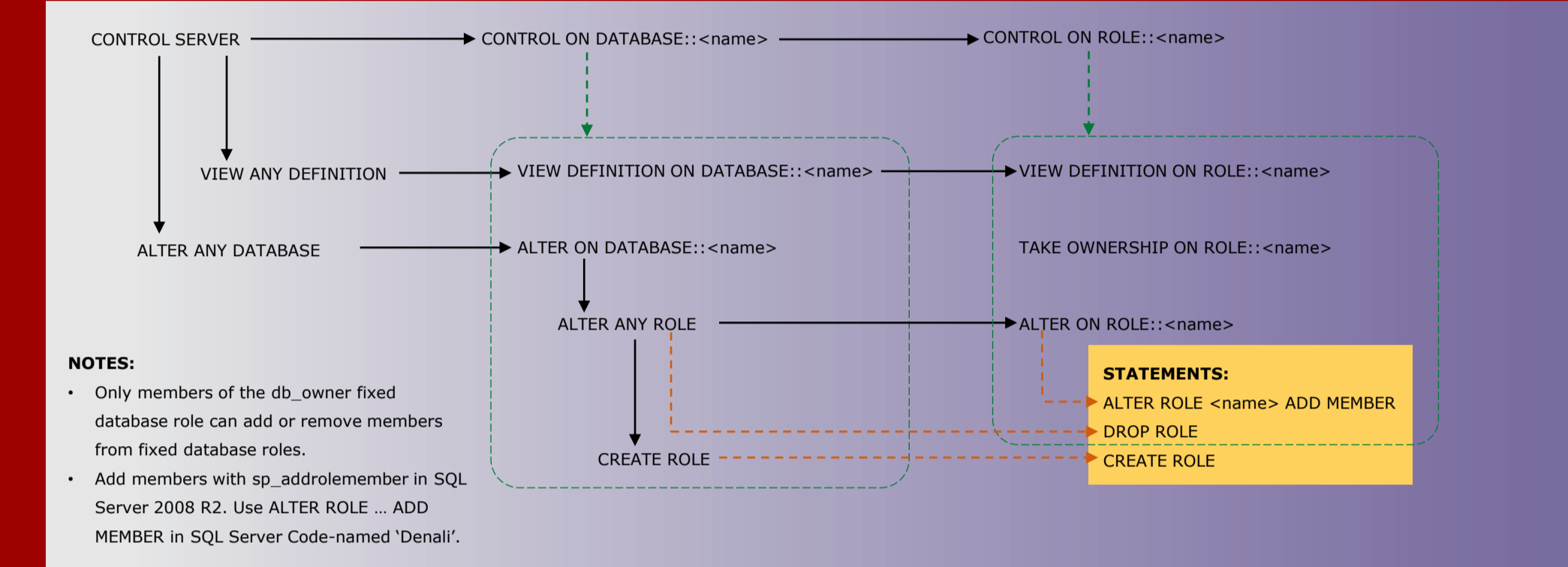
Full-text Permissions



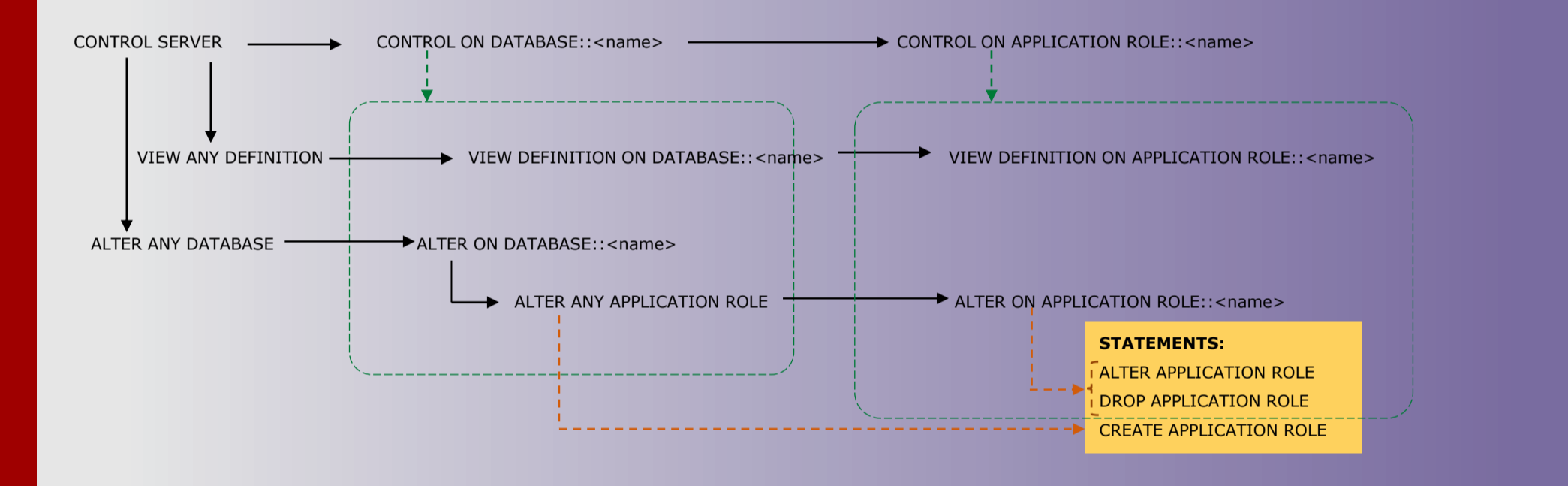
Connect and Authentication - Database Permissions



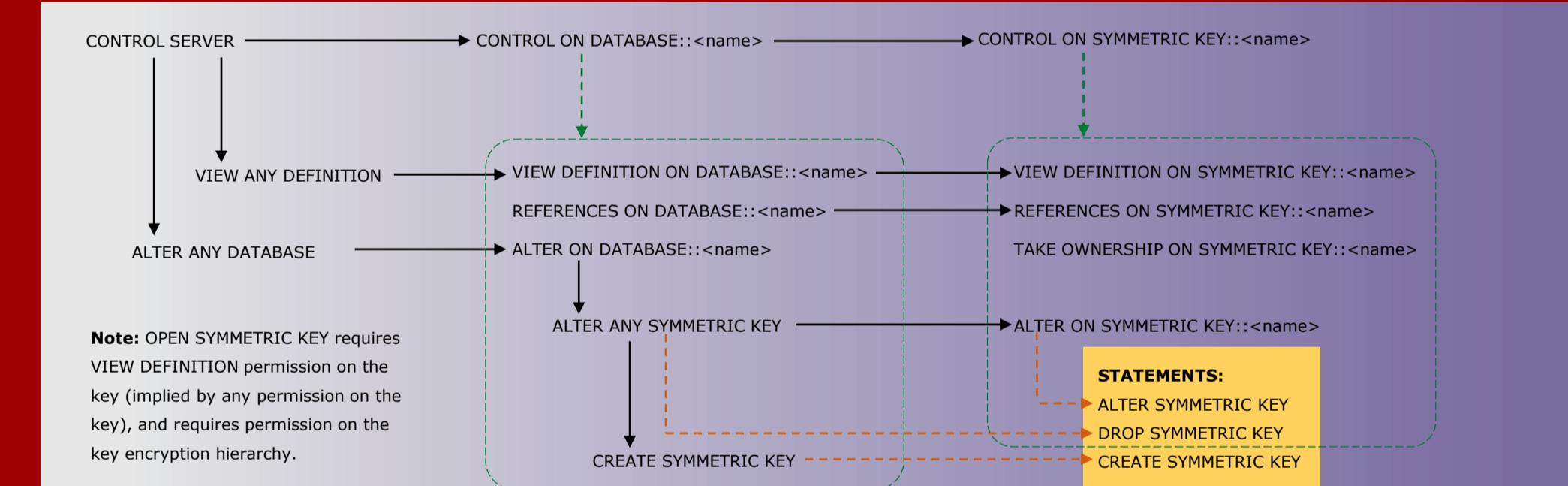
Database Role Permissions



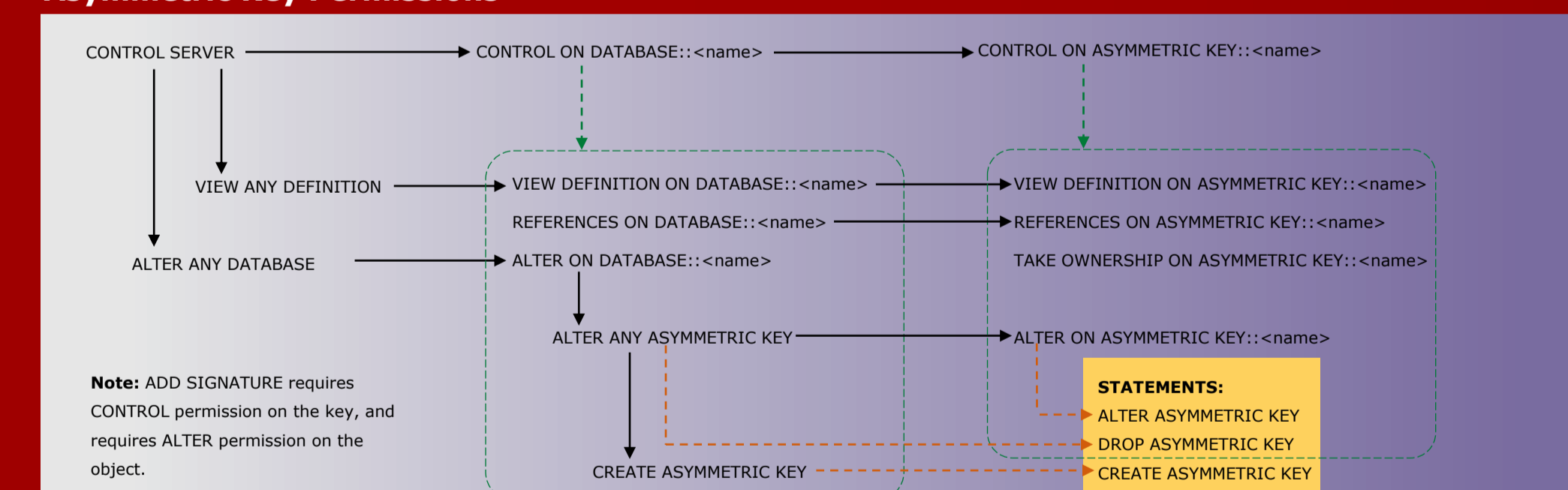
Application Role Permissions



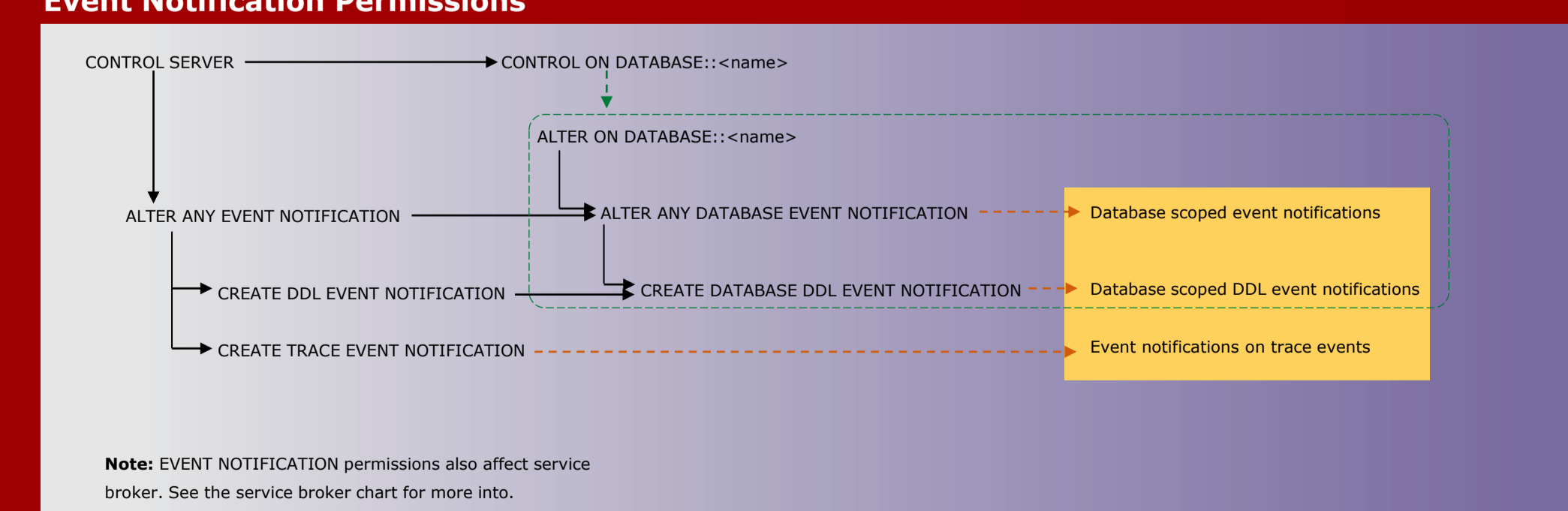
Symmetric Key Permissions



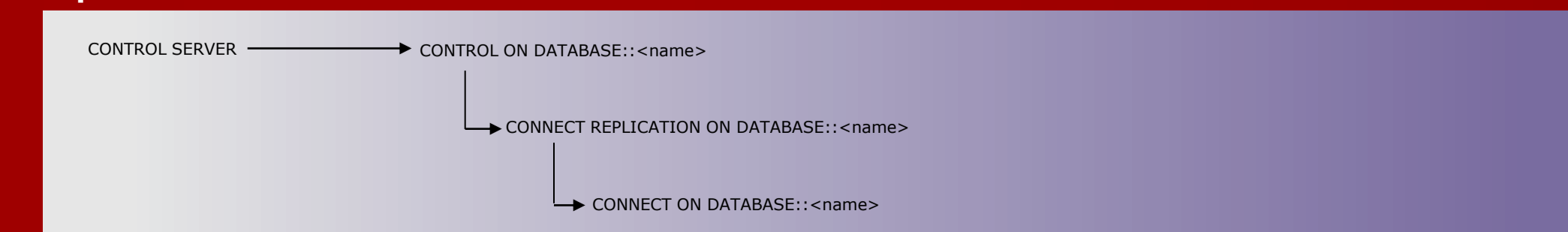
Asymmetric Key Permissions



Event Notification Permissions



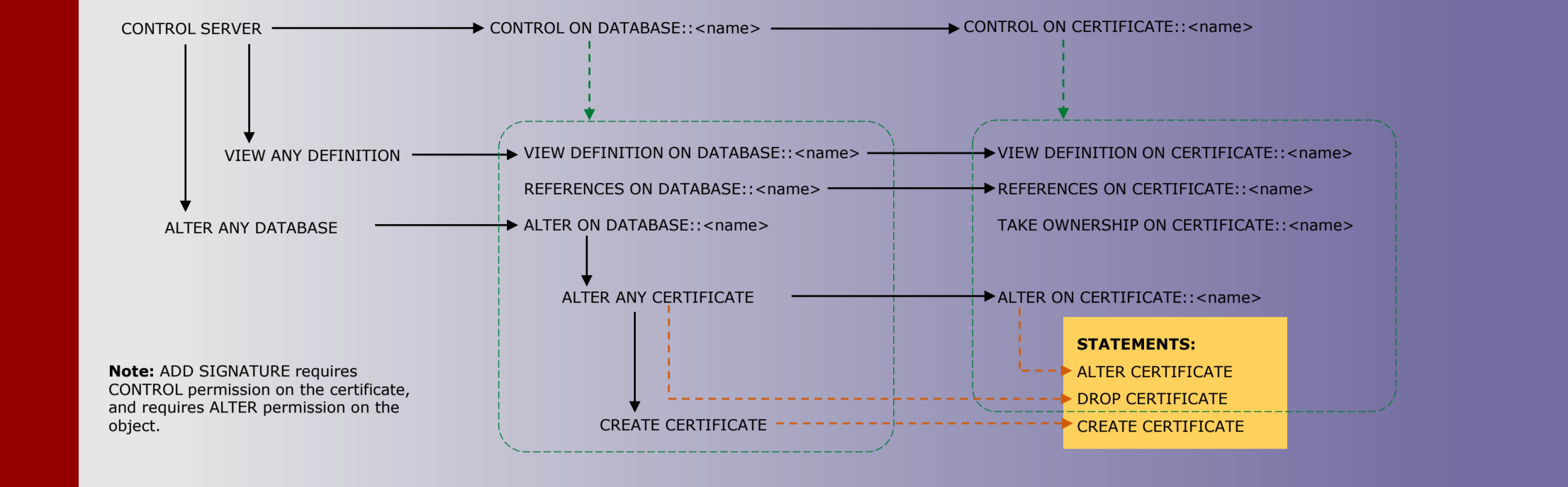
Replication Permissions



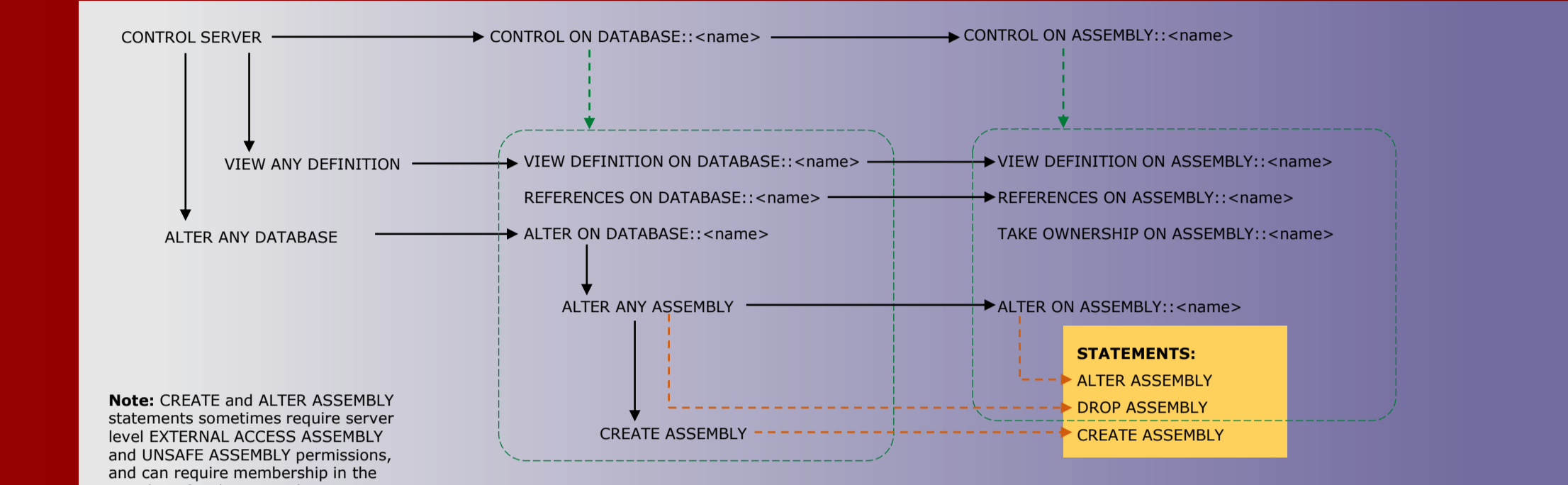
NOTES:

- The CONTROL SERVER permission has all permissions on the instance of SQL Server.
- The CONTROL DATABASE permission has all permissions on the database.
- Permissions do not imply role memberships and role memberships do not grant permissions. (E.g. CONTROL SERVER does not imply membership in the sysadmin fixed server role. Membership in the db_owner role does not grant the CONTROL DATABASE permission.) However, it is sometimes possible to impersonate between roles and equivalent permissions.
- Granting any permission on a securable allows VIEW DEFINITION on that securable. It is an implied permissions and it cannot be revoked, but it can be explicitly denied by using the DENY VIEW DEFINITION statement.

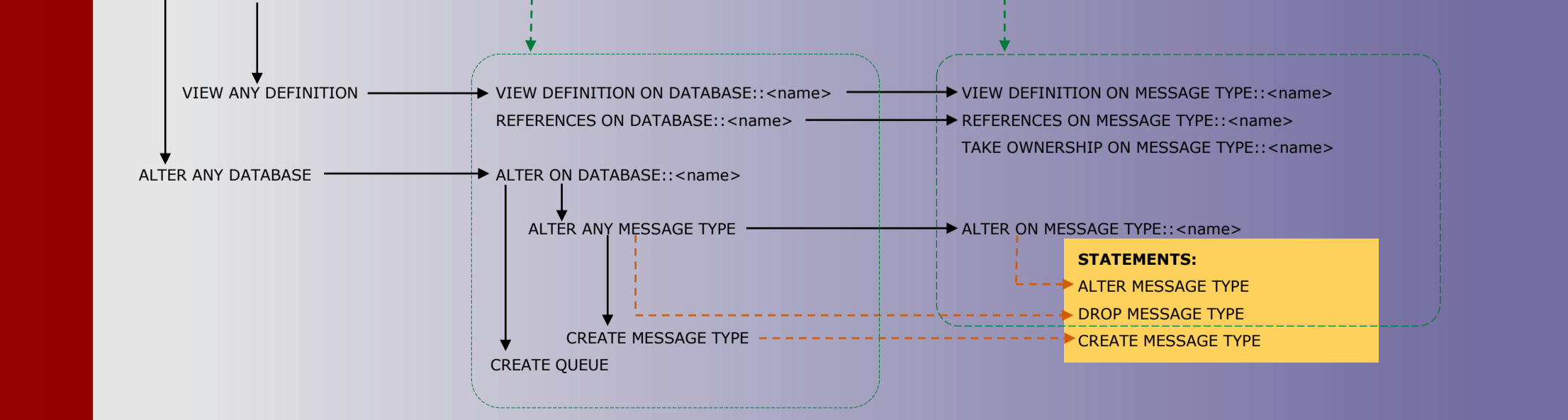
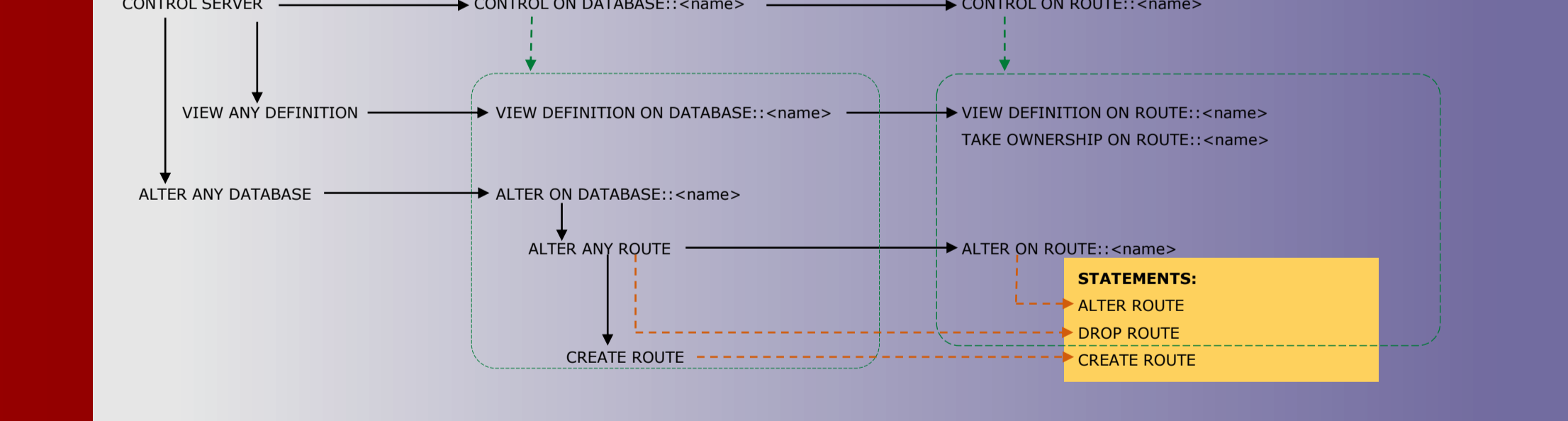
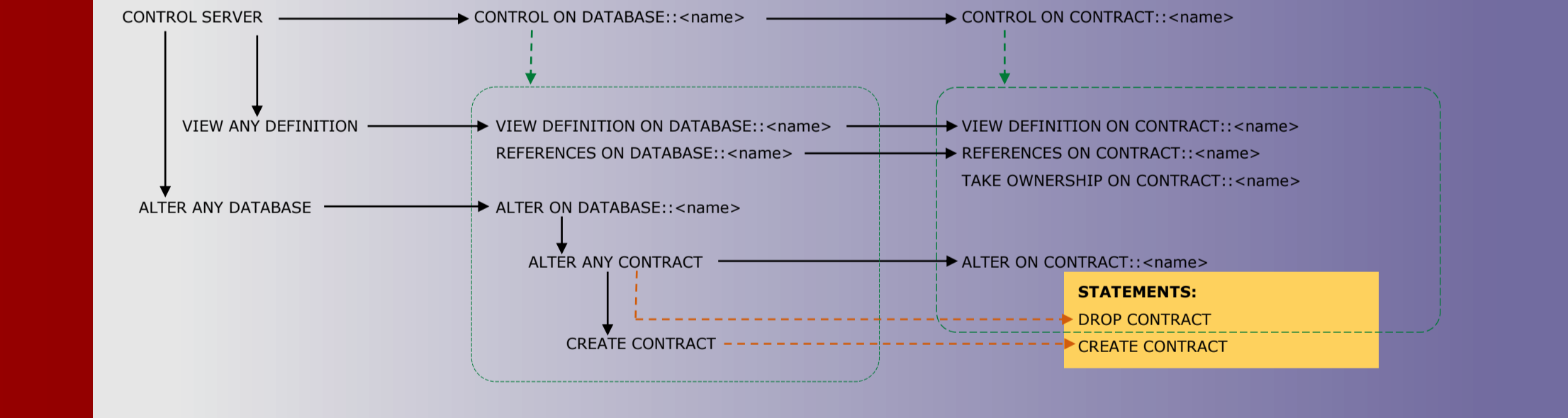
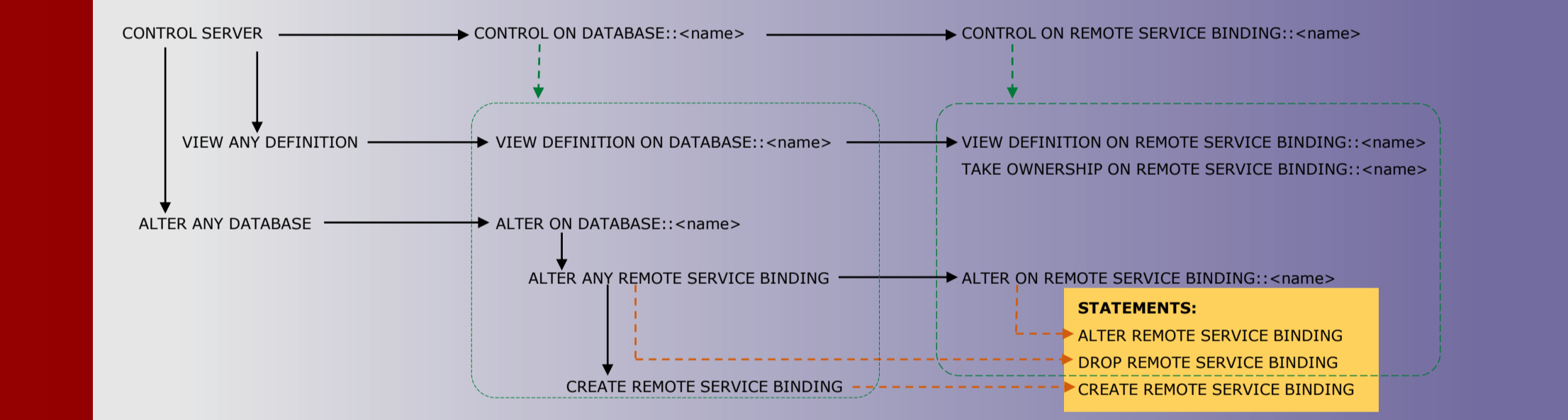
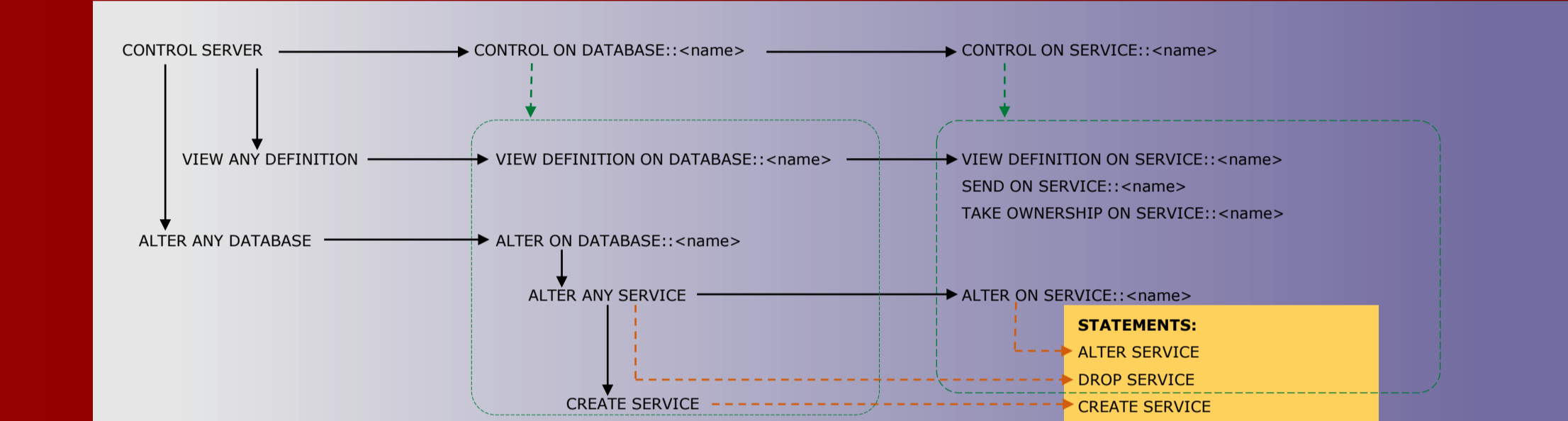
Certificate Permissions



Assembly Permissions



Service Broker Permissions



- Notes:**
- The user executing the CREATE CONTRACT statement must have REFERENCES permission on all message types specified.
 - The user executing the CREATE SERVICE statement must have REFERENCES permission on the queue and all contracts specified.
 - To execute the CREATE or ALTER REMOTE SERVICE BINDING the user must have impersonate permission for the principal specified in the statement.
 - When the CREATE or ALTER MESSAGE TYPE statement specifies a schema collection, the user executing the statement must have REFERENCES permission on the schema collection specified.
 - See the ALTER ANY EVENT NOTIFICATION chart for more permissions related to Service Broker.
 - See the SCHEMA OBJECTS chart for QUEUE permissions.
 - The ALTER CONTRACT permission exists but at the time there is no ALTER CONTRACT statement.



August 25, 2011
 © 2011 Microsoft Corporation. All rights reserved.
 Questions and comments to Rick.Byham@Microsoft.com