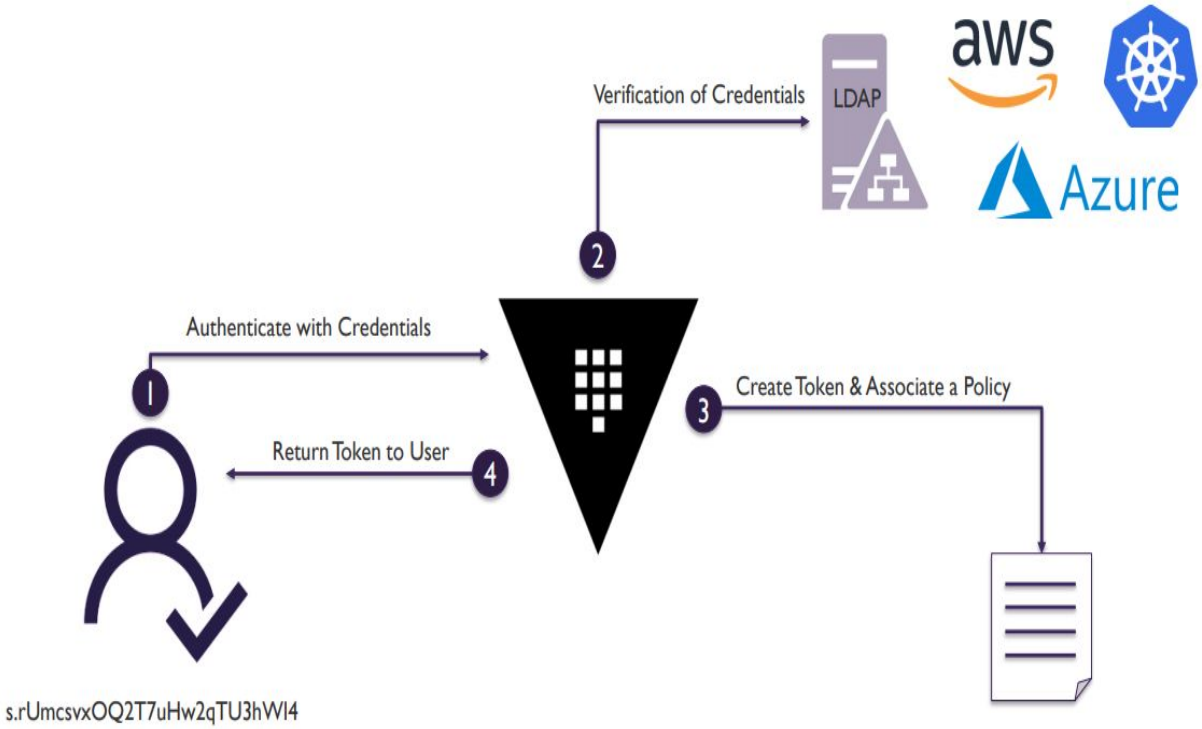
Policies

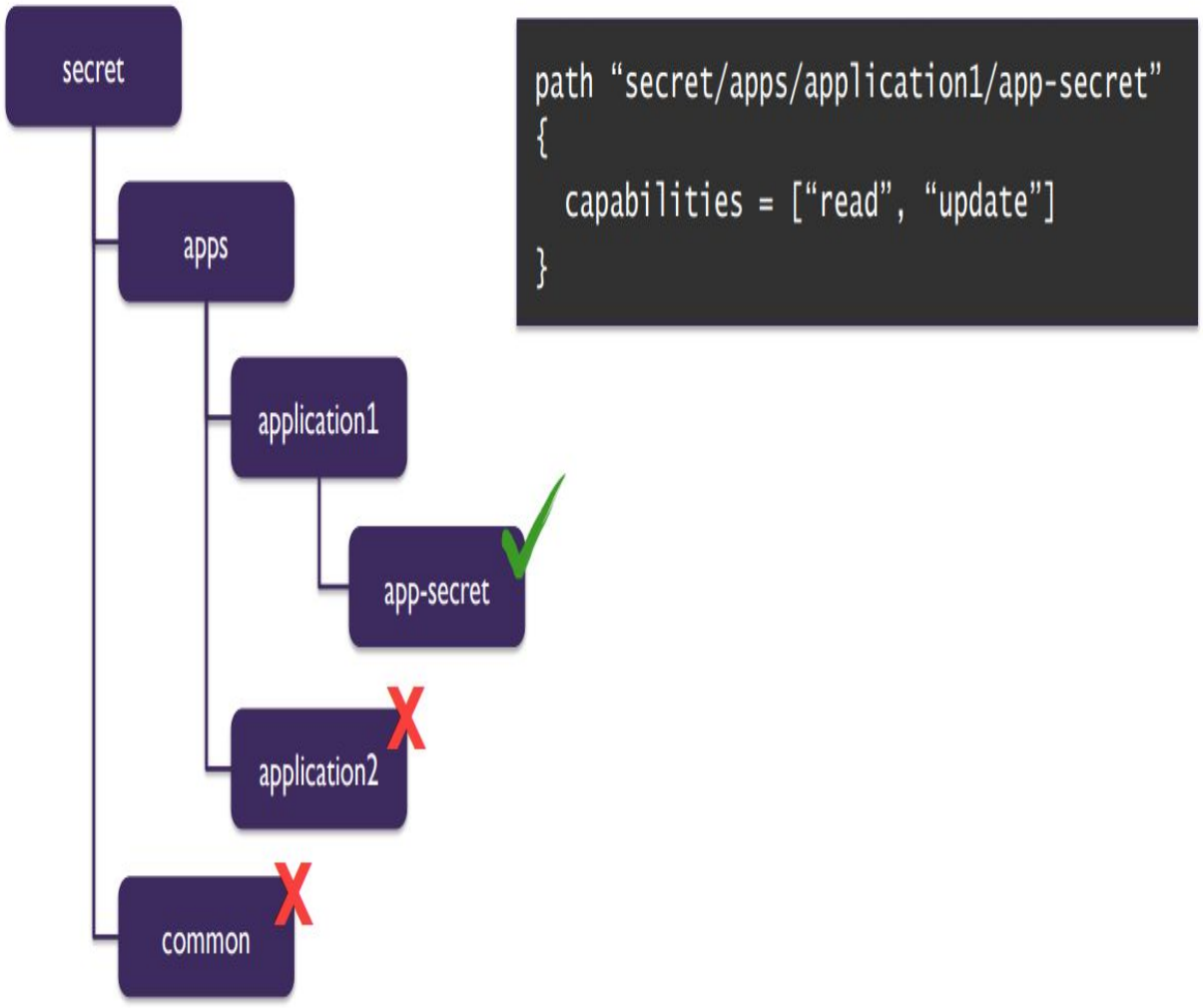
What are Vault Policies?

- Provides a way to permit or deny access to certain paths or actions within Vault (RBAC)
- Provides authorization using a declarative policy written in HCL or JSON
- Permissions include:

  List, Deny, Sudo, Create, Read, Update, Delete
- Policies are deny by default
- Simplest form of policy will include a path (secret/data) and the permitted capabilities (create, read, update, etc.)
- More complicated policies can include variable replacements and/or parameters
- Deny always takes precedence over other capabilities

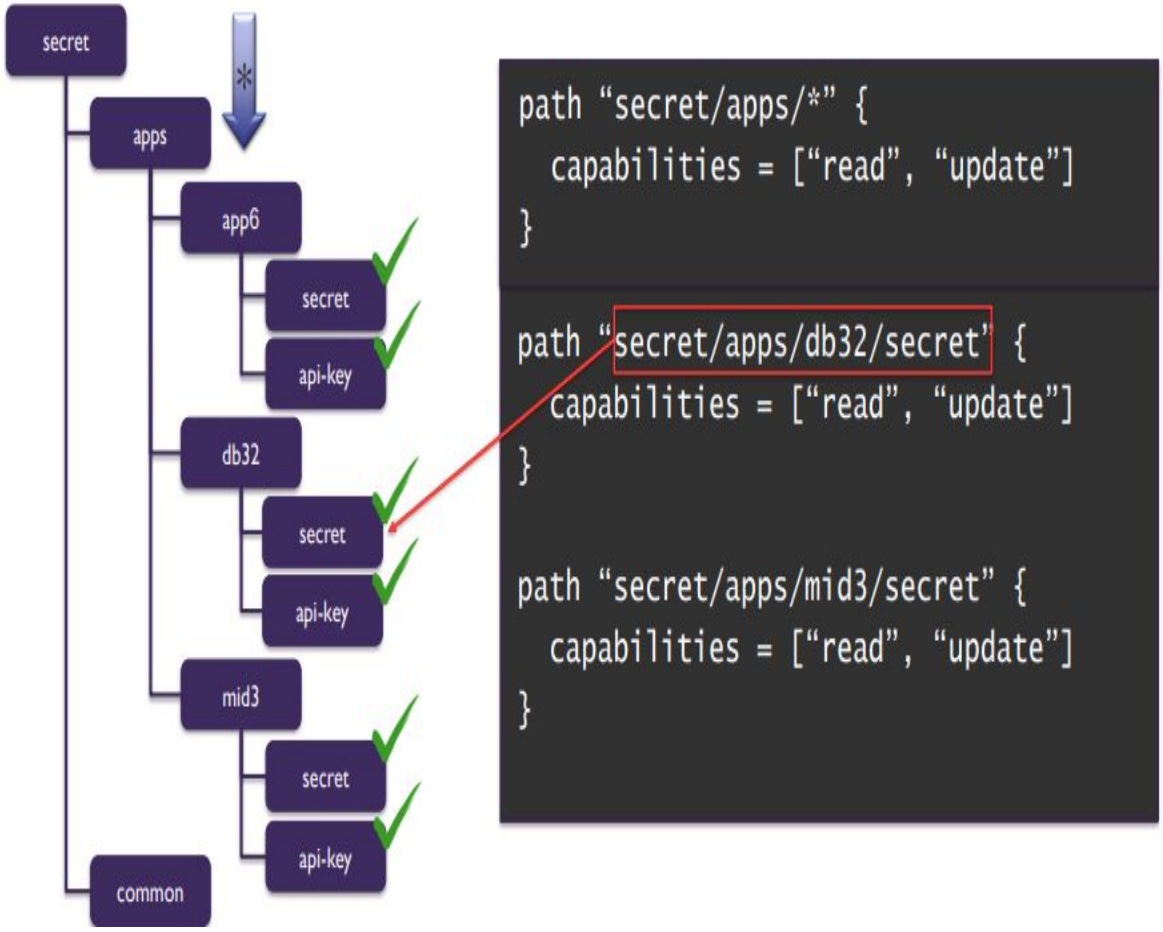


Simple Policy Example:

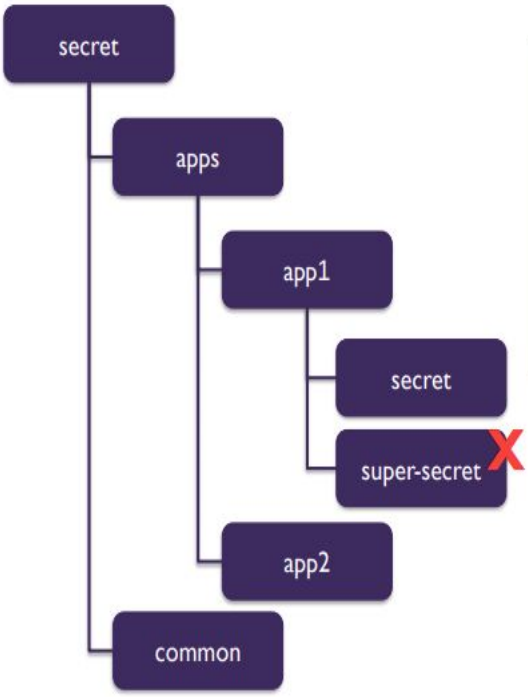


Using the * and +:

- The glob (*) is a wildcard and can only be used at the end of a path
- Can be used to signify anything “after” a path or as part of a pattern
  – secret/apps/application1/* - allows anything below application1
  – secret/path/db-* - would match secret/path/db-2 but not secret/path/db2
- The plus (+) denotes any number of characters within a single path segment (secret/+/db - matches secret/db2/db or secret/app/db)
- Can be used in multiple path segments (i.e., secret/+/+/db)



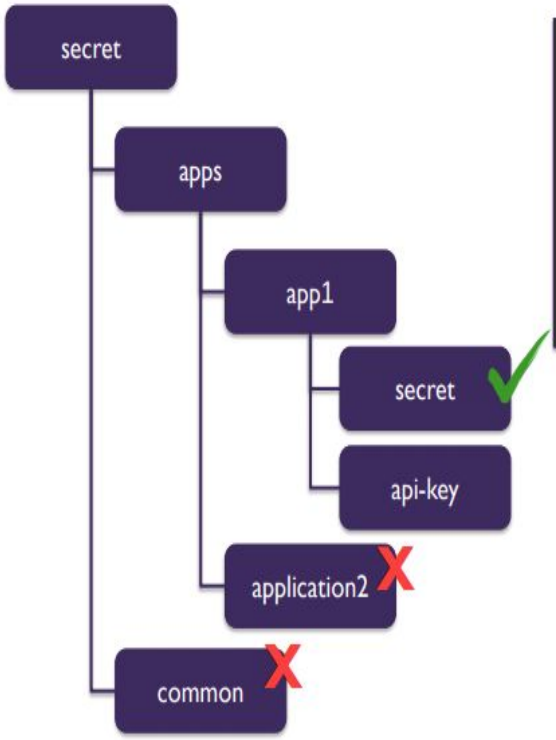
Using Deny:



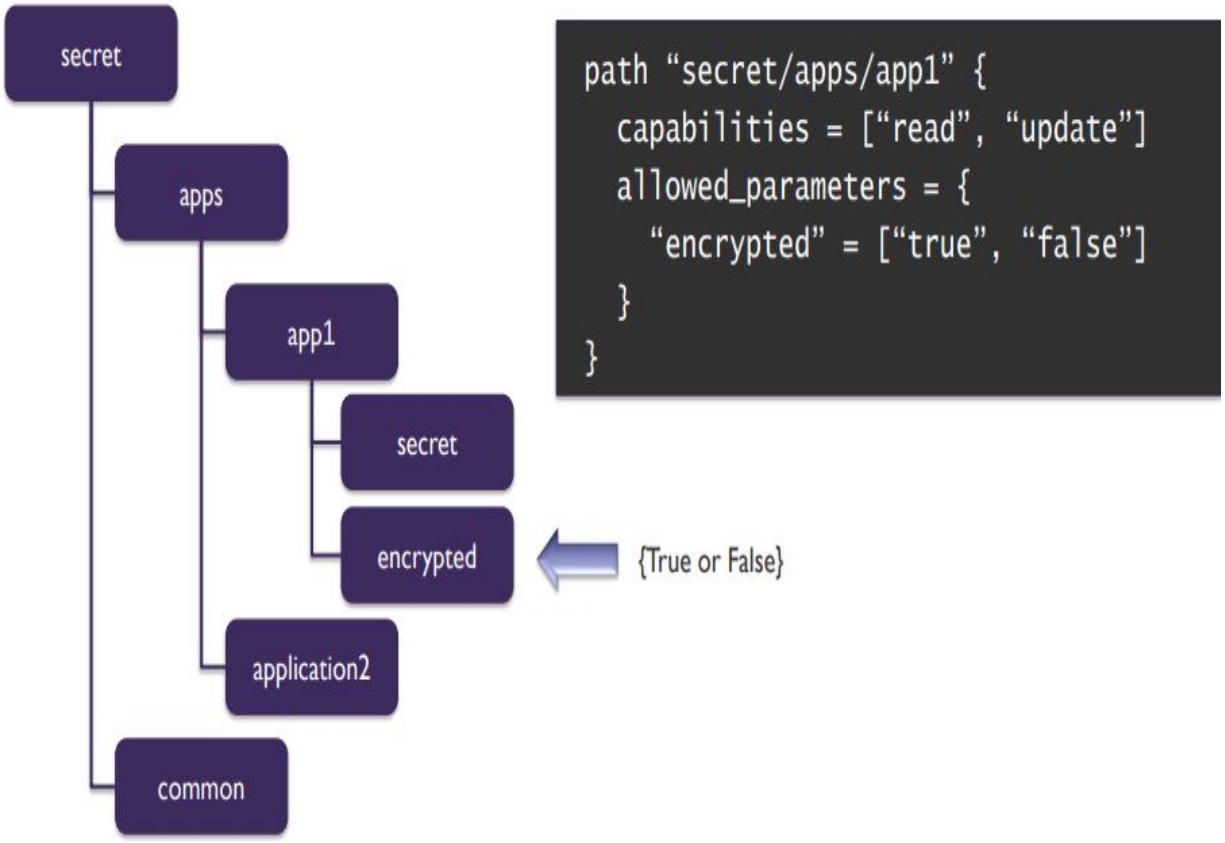

```
path "secret/*" {
  capabilities = ["read", "update"]
}

path "secret/apps/app1/super-secret" {
  capabilities = ["deny"]
}
```

Detailed Policy Example:



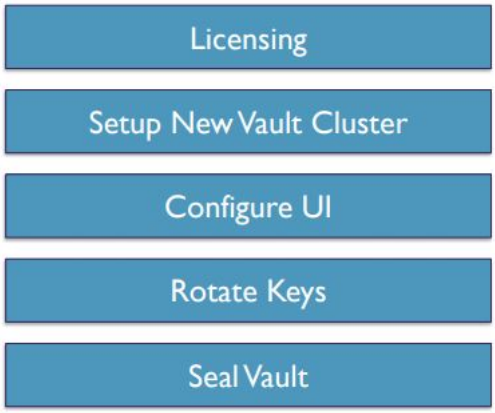

```
path "secret/apps/app1/*" {
  capabilities = ["read", "update"]
  allowed_parameters = {
    "secret" = []
  }
}
```



Administrative Policies:

- Permissions for Vault backend functions live at the sys/ path
- Users/admins will need policies that define what they can do within Vault to administer Vault itself
  - – Unsealing
  - – Changing policies
  - – Adding secret backends
  - – Configuring database configurations

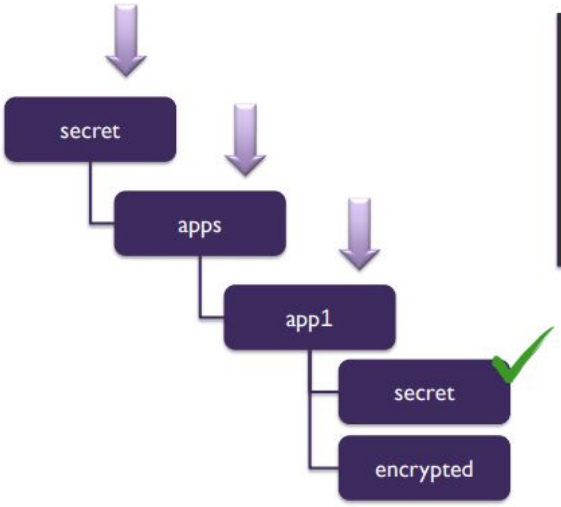
Admin Policy Example:




```
# Configure License
path "sys/license" {
  capabilities = ["read", "list", "create", "update", "delete"]
}
# Initialize Vault
path "sys/init" {
  capabilities = ["read", "update", "create"]
}
# Configure UI in Vault
path "sys/config/ui" {
  capabilities = ["read", "list", "update", "delete", "sudo"]
}
# Allow rekey of unseal keys for Vault
path "sys/rekey/*" {
  capabilities = ["read", "list", "update", "delete"]
}
# Allows rotation of master key
path "sys/rotate" {
  capabilities = ["update", "sudo"]
}
# Allows Vault seal
path "sys/seal" {
  capabilities = ["sudo"]
}
```

Policies for the UI:

- If you're using the UI, you'll likely need to add additional LIST permissions
- Remember that LIST doesn't allow the user to READ secrets
- Without LIST, the user cannot browse to the desired path/secret
- The lack of LIST permissions is sort of security by obscurity




```
path "secret/apps/app1/secret" {
  capabilities = ["read", "update"]
}
path "secret/*" {
  capabilities = ["list"]
}
```