

Mapping to Swift Pattern Matching

1:1 mapping from table to
code

```
switch (role, resource, action, isOwner) {  
    // Admin can do anything  
    case (.admin, _, _, _):  
        return .allow  
  
    // User access to public resources  
    case (.user, .publicResource, .read, _):  
        return .allow  
  
    // User access to shared resources  
    case (.user, .sharedResource, .read, _):  
        return .allow  
  
    // User access to owned resources (full control)  
    case (.user, _, .read, true):  
        return .allow  
    case (.user, _, .write, true):  
        return .allow  
    case (.user, _, .delete, true):  
        return .allow  
  
    // Guest access  
    case (.guest, .publicResource, .read, _):  
        return .allow  
  
    default:  
        return .deny
```

Mapping to Swift Compiler Enforces Completeness

```
) -> AccessResult {
    switch (role, resource, action, isOwner) {
        // Admin can do anything
        case (.admin, _, _, _):
            return .allow

        // All other cases denied (explicit enumeration for exhaustiveness)
        case (.user, .publicResource, .write, _),
               (.user, .publicResource, .delete, _),
               (.user, .sharedResource, .write, _),
               (.user, .sharedResource, .delete, _),
               (.user, .privateResource, .read, false),
               (.user, .privateResource, .write, false),
               (.user, .privateResource, .delete, false),
               (.guest, .publicResource, .write, _),
               (.guest, .publicResource, .delete, _),
               (.guest, .sharedResource, _, _),
               (.guest, .privateResource, _, _):
            return .deny
    }
}
```