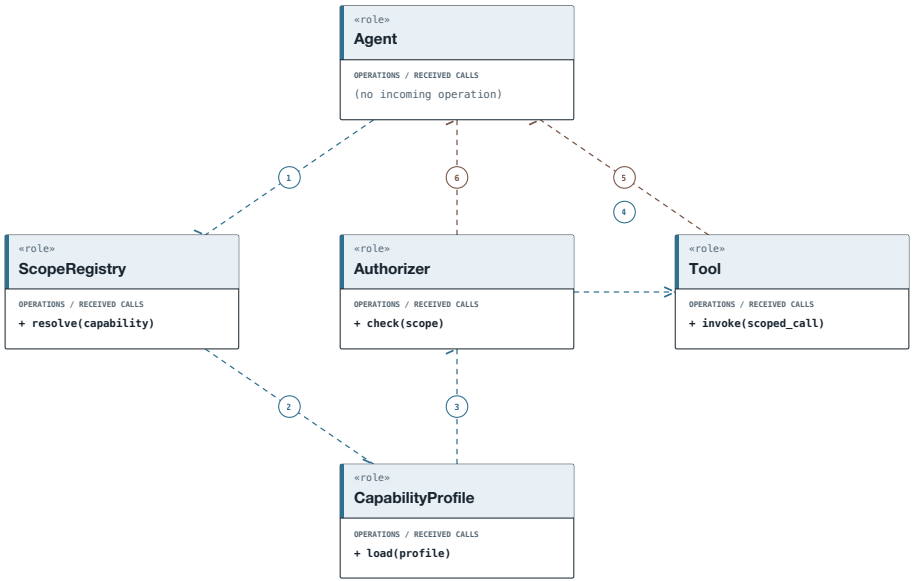
# P25 Scope-based authorization and profiles

**Primitive / UML CLASS VIEW / ANALYTICAL MODEL**

Bind capabilities to explicit scopes.

**UML CLASS VIEW**

dashed arrow: call dependency · brown: return/failure · analytical, not an observed trace




**POINTCUT / PATTERN INTERFACE**

| | |
|---|---|
| CONTEXT | A caller invokes a method or service with an operator, session, or agent identity. |
| TRIGGER | Capability is checked against a requested scope. |
| STRUCTURE | Scope identity, profile, grant and revocation state, and resolution function. |
| EFFECT | The call is permitted only if the principal currently holds a covering scope. |

**CONTRACT / REQUIRED BEHAVIOR**

| | |
|---|---|
| REQUIRES | Principal identity and requested capability are explicit. |
| ENSURES | Authorization is evaluated at the call boundary and records the covering scope. |
| INVARIANT | Scopes are not inherited beyond declared containment or lifetime. |
| VIOLATION | Missing, expired, revoked, or cross-scope capabilities are denied explicitly. |

**RELATIONSHIPS / ANALYTICAL CALLS**

number · caller → callee · message

1 Agent → ScopeRegistry resolve(capability)
2 ScopeRegistry → CapabilityProfile load(profile)
3 CapabilityProfile → Authorizer check(scope)
4 Authorizer → Tool invoke(scoped_call)
5 Tool ↩ Agent return(result)
6 Authorizer ↩ Agent deny(failure)

**VARIATION / TEST / NEGATIVE CONTROL**

| | |
|---|---|
| VARIATION | Method scopes, roles, session grants, profiles, and delegated identities. |
| TEST_OBLIGATION | Test widening, revocation, expiration, child agent, and cross-tenant access. |
| NEGATIVE_CONTROL | A boolean admin flag is not a scope model. |