Relationships for UMA WG Workshop

Eve Maler, Steve Venema, Scott Fehrman

9 Dec 2021

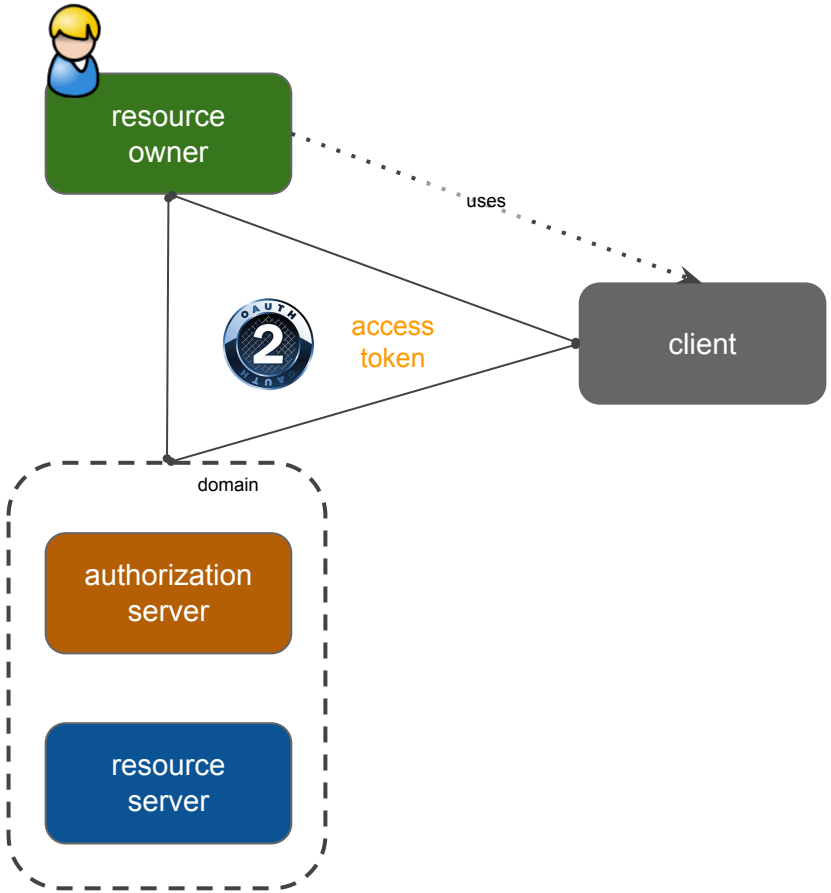
OAuth and UMA basics

OAuth and UMA

“ALICE-TO-SELF” SHARING

OAuth enables **constrained delegation** of access to **apps** on request

Alice can **agree** to app connections and also **revoke** them

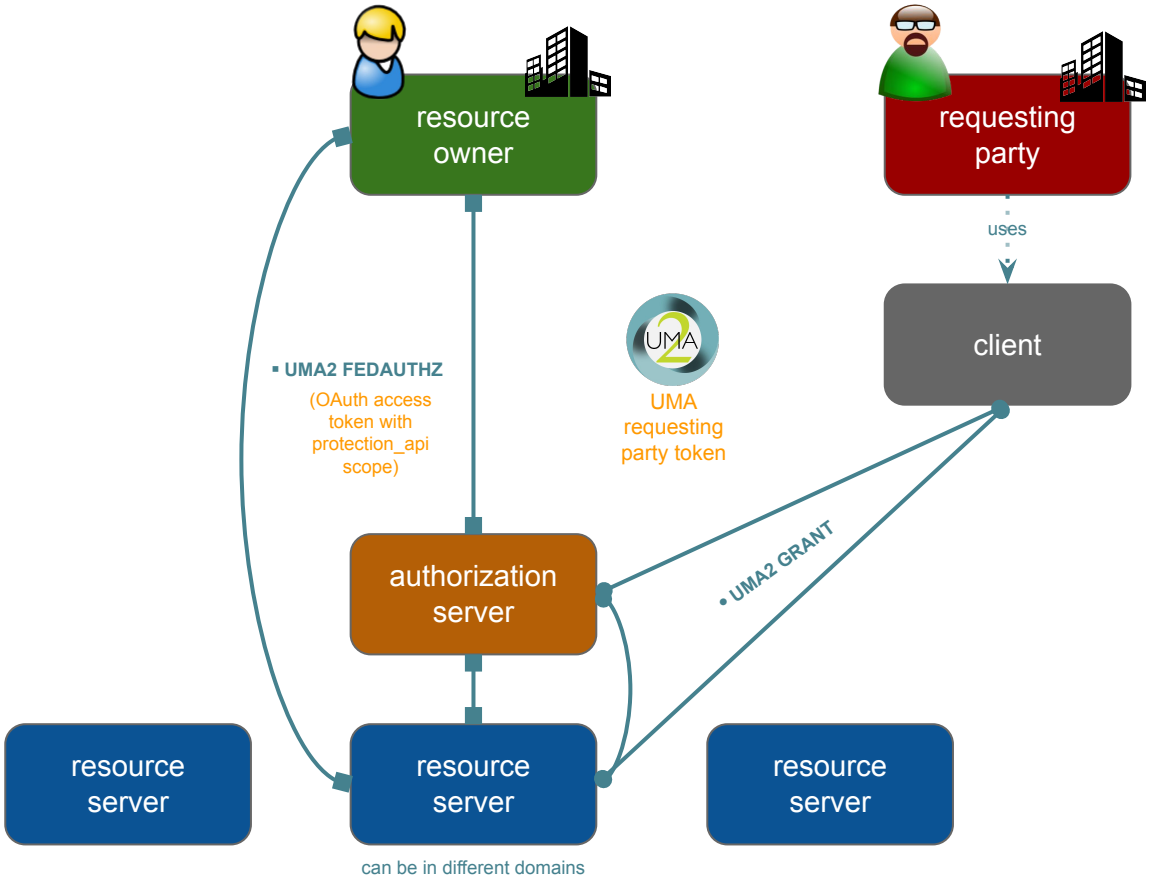


OAuth and UMA

“ALICE-TO-BOB” SHARING

UMA adds **control** of **cross-party sharing**, letting Alice be **absent** when Bob uses a client to attempt access

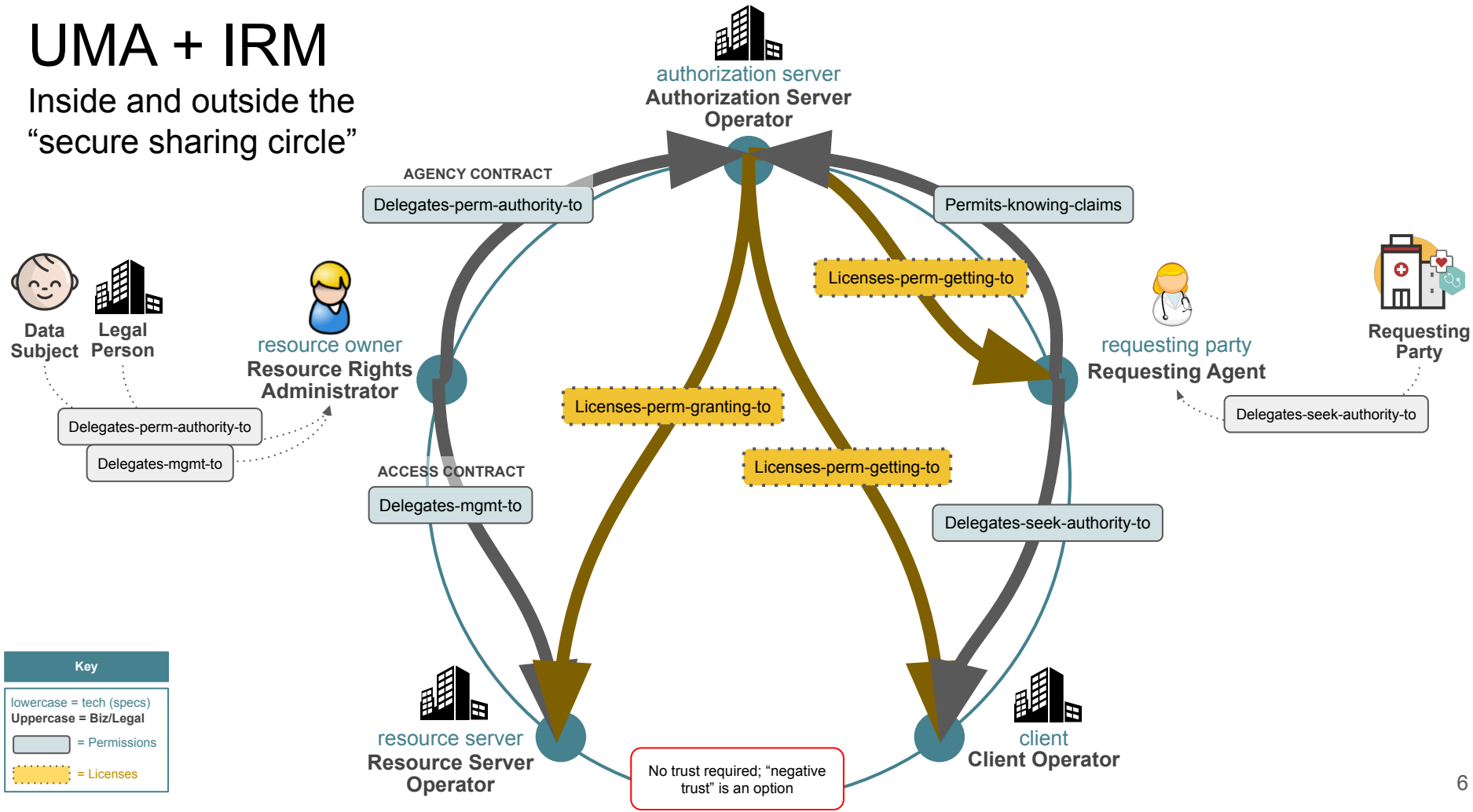
Alice **controls trust** between resource hosts and authorization services – enabling a **wide ecosystem** of resource hosts, so Alice can manage sharing **across** them



Relationship basics

UMA + IRM

Inside and outside the “secure sharing circle”



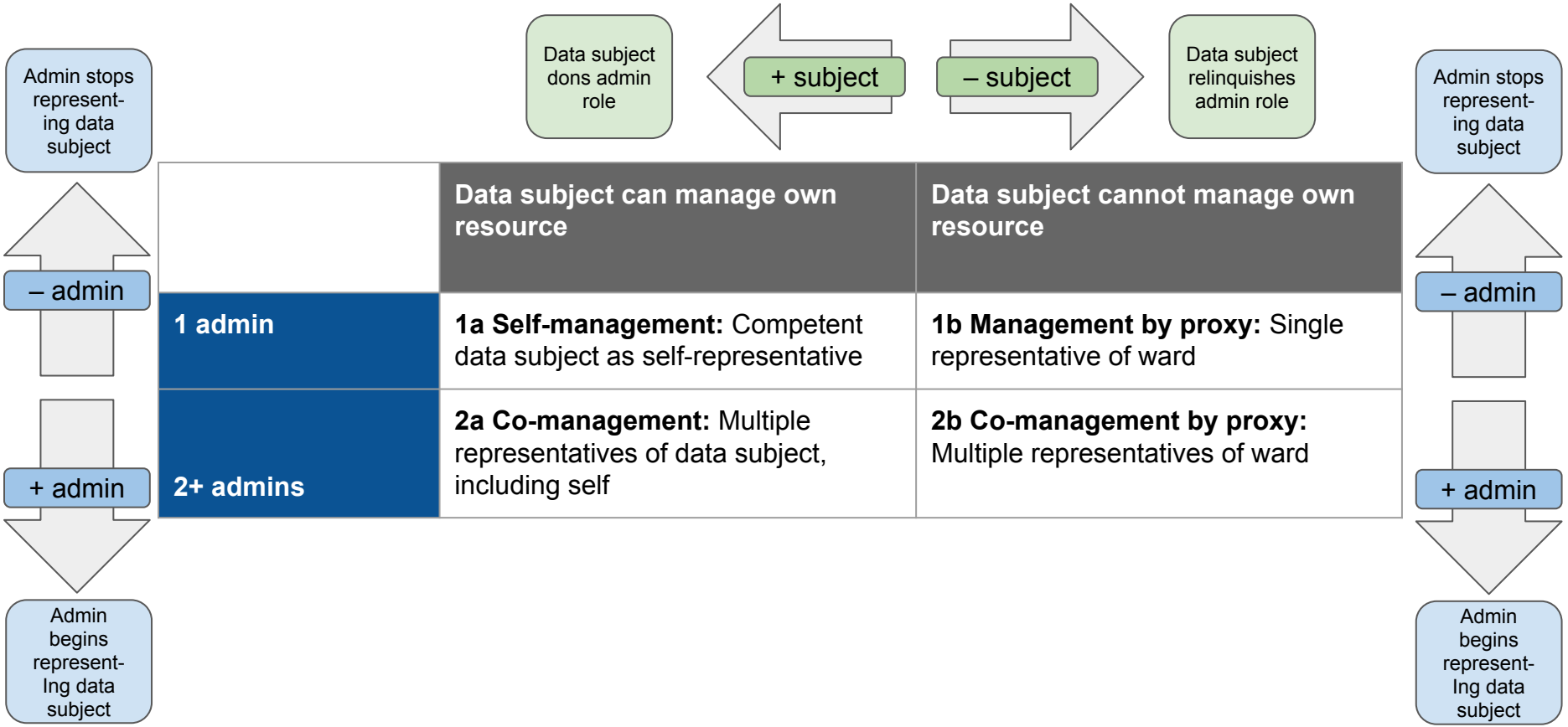
Key steady-state design patterns

Framework for use cases and flows

	Data subject can manage own resources	Data subject cannot manage own resources
1 admin	**Self-management:** Competent data subject as self-representative	**Management by proxy:** Single representative of ward
2+ admins	**Co-management:** Multiple representatives of data subject, including self	**Co-management by proxy:** Multiple representatives of ward

Design patterns of key state changes

Framework for use cases and flows



	Data subject can manage own resource	Data subject cannot manage own resource
1 admin	**1a Self-management:** Competent data subject as self-representative	**1b Management by proxy:** Single representative of ward
2+ admins	**2a Co-management:** Multiple representatives of data subject, including self	**2b Co-management by proxy:** Multiple representatives of ward