Kantara Workshop: Making the World Safe for User-Managed Access

Eve Maler
Kantara UMA Work Group Chair
4 May 2010

About Kantara Initiative

http://kantarainitiative.org

- **Participation:** Open global identity, web, and developer community of individuals and organizations, such as:
  - Deployers, operators, Web 2.0 service providers, eGovernment agencies, IT vendors, consumer electronics vendors
  - Developers and members of the open source, legal, and privacy communities
- **Goal:** Harmonize identity community activities to help ensure secure, identity-based, online interactions
  - While preventing misuse of personal information so that networks will become privacy protecting and more natively trustworthy environments.
- **Work:** 18 Work Groups and Discussion Groups (including UMA) and two certification oversight bodies (Assurance and Interop)

How to participate

- It's absolutely free to participate in any group
  - You can also support the overall goals of the Initiative with an individual or organizational Membership
- Today is a public workshop
  - You are invited to become an UMA WG participant ("UMAnitarian"!) to contribute actively to our work
  - To become a participant right now, visit **kantarainitiative.org**, select the User-Managed Access Group, and click on Join This Group
    - We operate under reciprocal royalty-free rules

Suggested agenda for today

- Introductions
- UMA in a nutshell
- The landscape and requirements
- Scenarios, use cases, and user experiences
- The UMA protocol
- Policies, claims, and agreements

Introductions

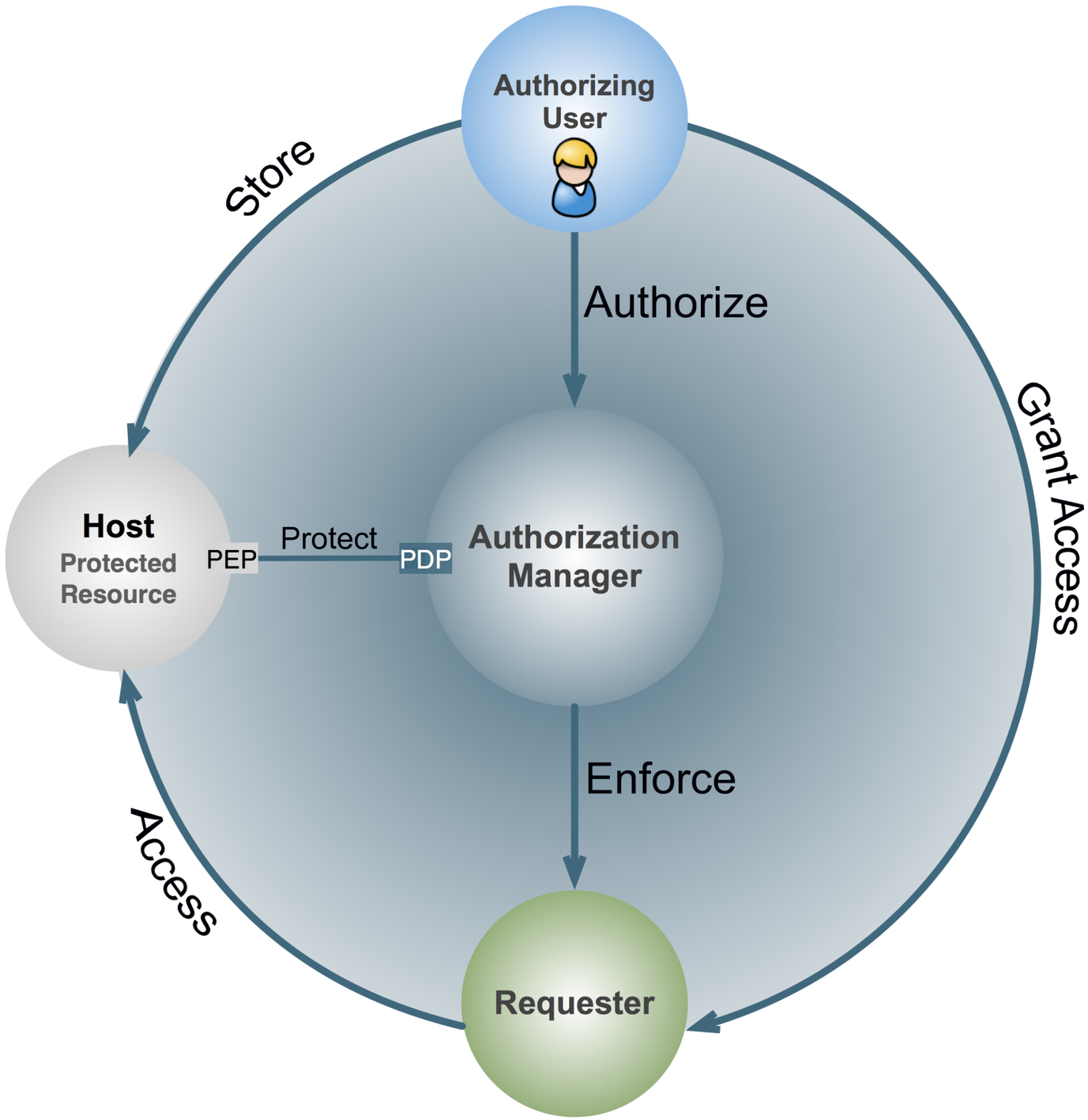
UMA in a nutshell

UMA is...

- A web protocol that lets you control authorization of data sharing and service access made on your behalf
- A Work Group of the Kantara Initiative that is free for anyone to **join** and contribute to
- A set of draft specifications that is free for anyone to implement
- Heading towards multiple implementation efforts
- Going to be contributed to the IETF
- Striving to be simple, OAuth-based, identifier-agnostic, RESTful, modular, generative, and developed rapidly

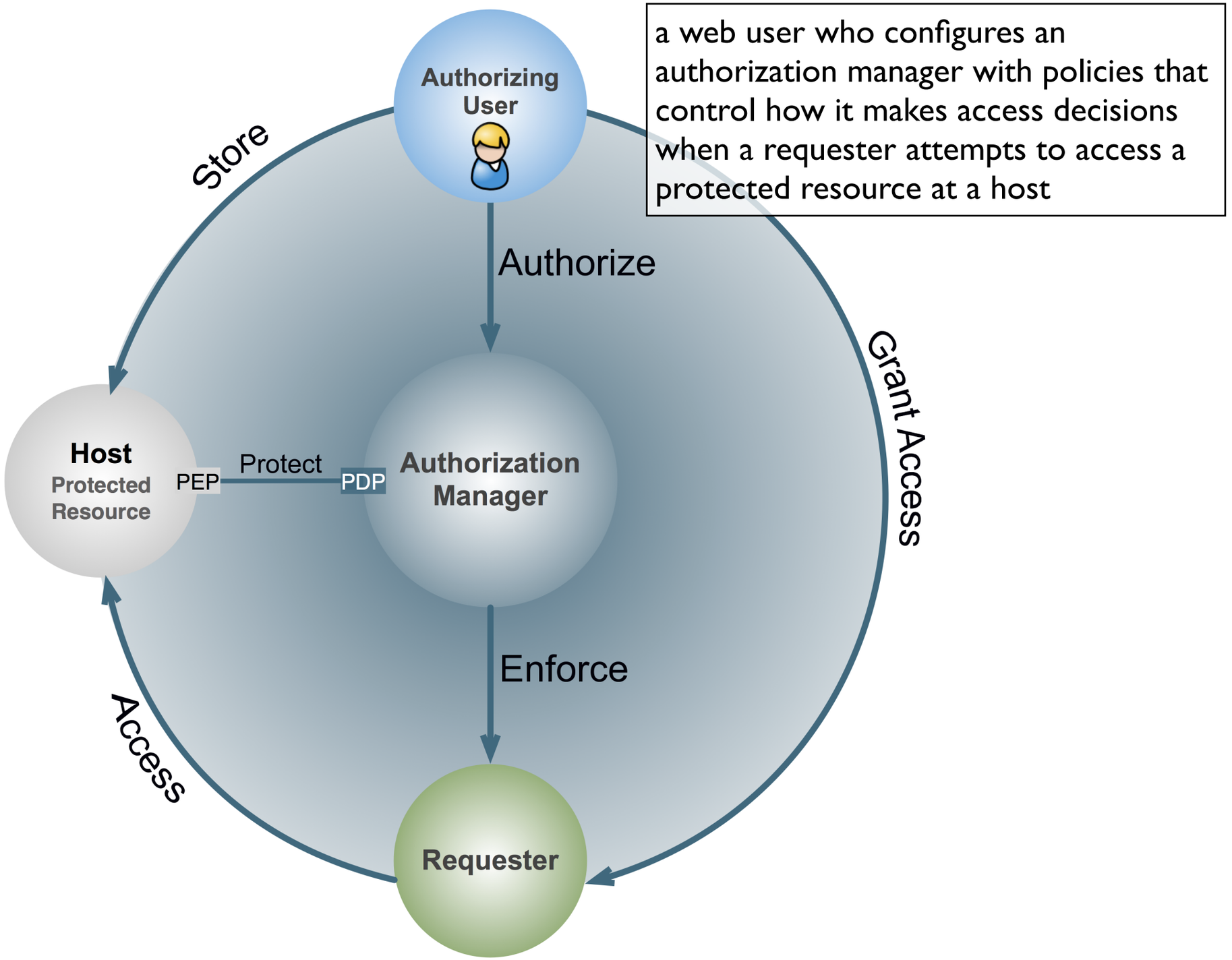
The players

(definitions come from core protocol spec)



The players

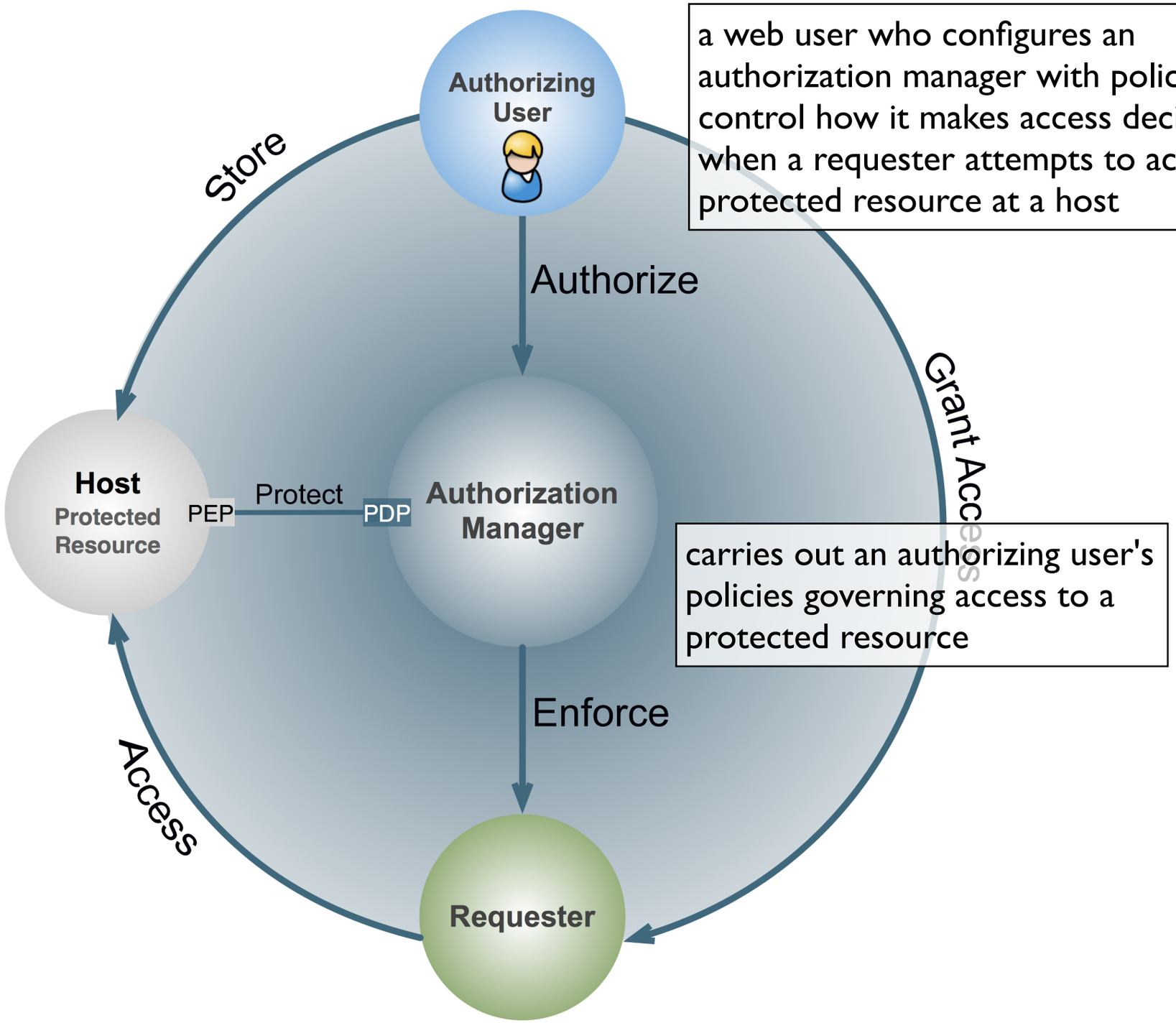
(definitions come from core protocol spec)



a web user who configures an authorization manager with policies that control how it makes access decisions when a requester attempts to access a protected resource at a host

The players

(definitions come from core protocol spec)

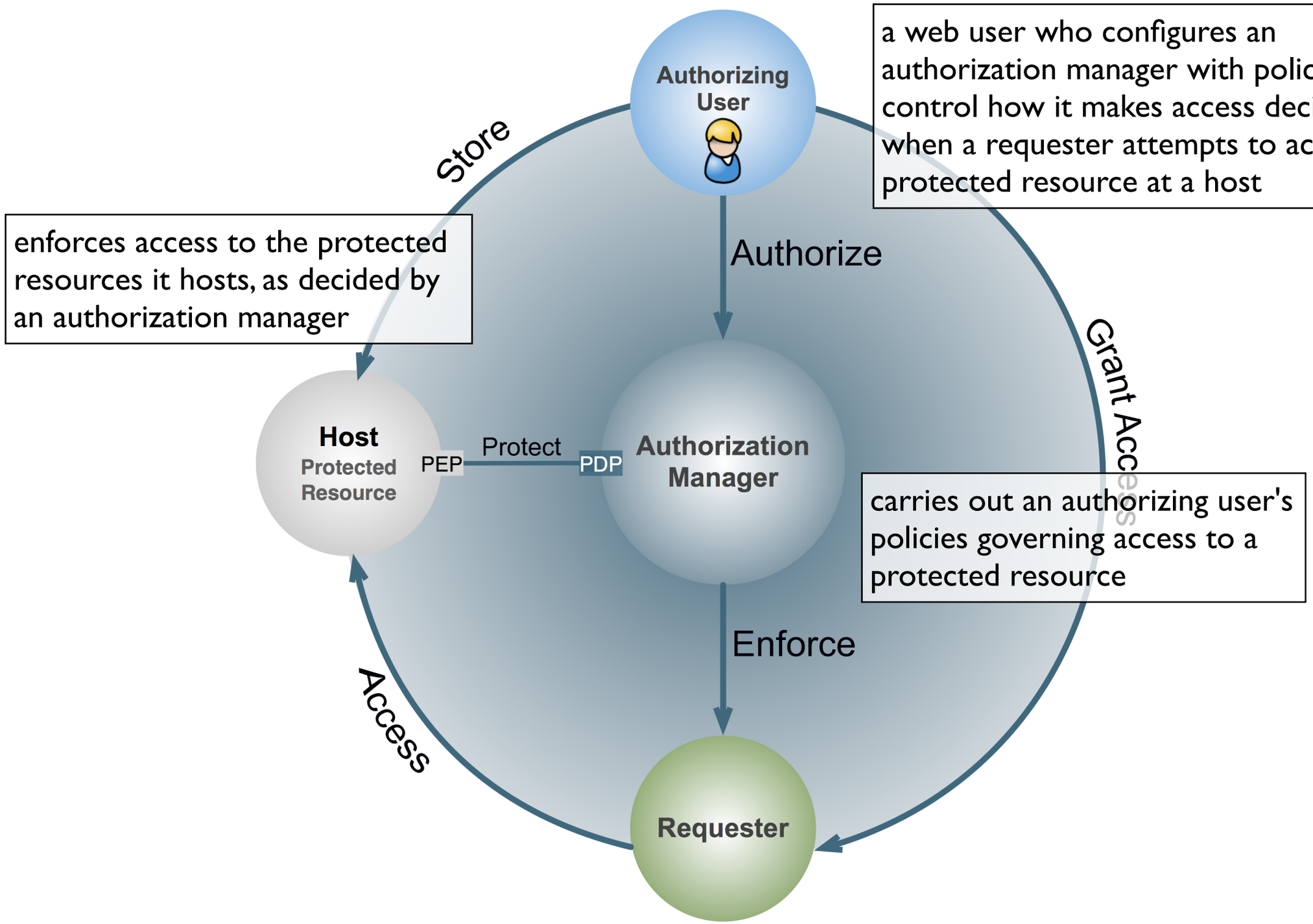


a web user who configures an authorization manager with policies that control how it makes access decisions when a requester attempts to access a protected resource at a host

carries out an authorizing user's policies governing access to a protected resource

The players

(definitions come from core protocol spec)



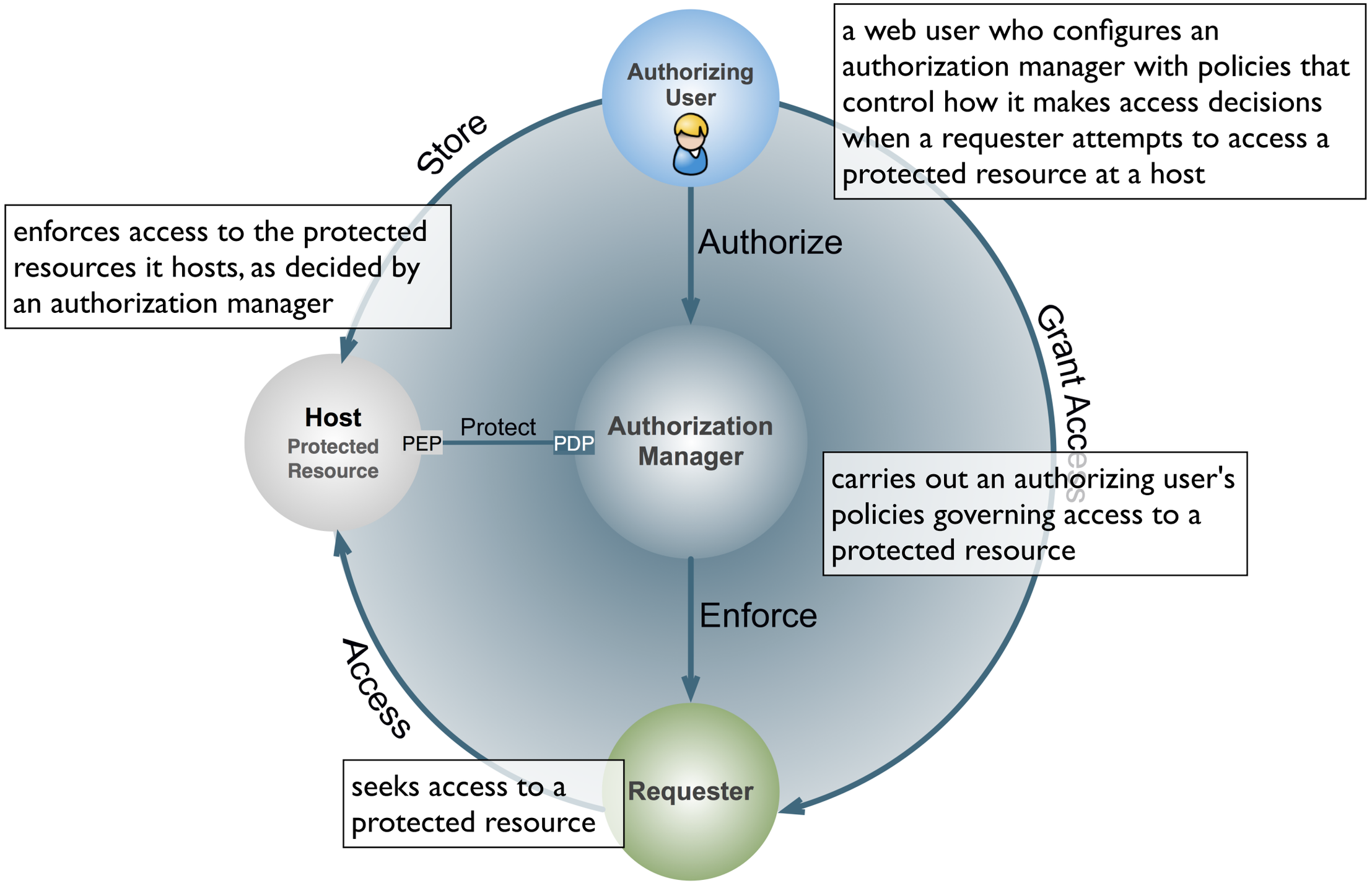
a web user who configures an authorization manager with policies that control how it makes access decisions when a requester attempts to access a protected resource at a host

enforces access to the protected resources it hosts, as decided by an authorization manager

carries out an authorizing user's policies governing access to a protected resource

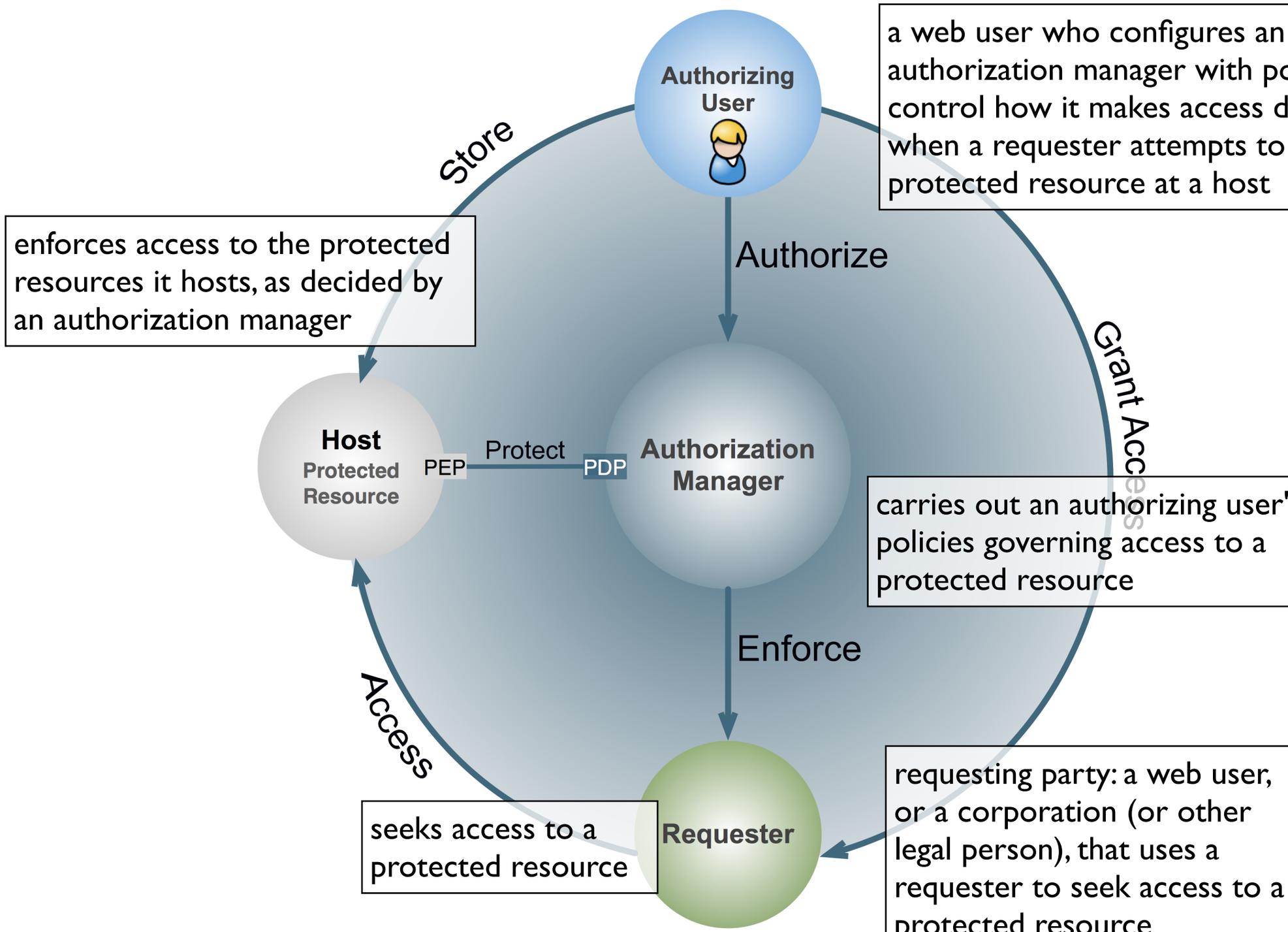
The players

(definitions come from core protocol spec)



The players

(definitions come from core protocol spec)



a web user who configures an authorization manager with policies that control how it makes access decisions when a requester attempts to access a protected resource at a host

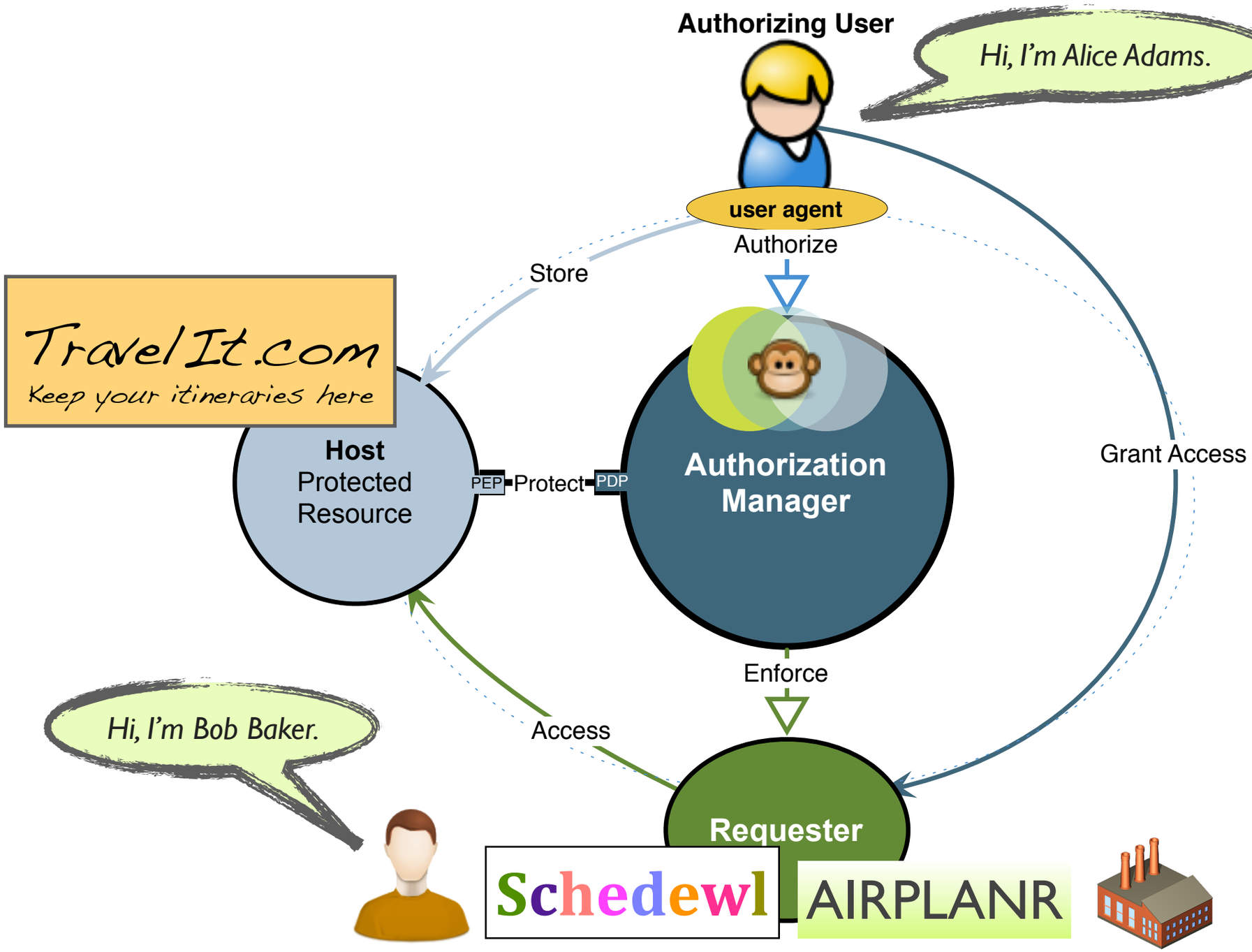
enforces access to the protected resources it hosts, as decided by an authorization manager

carries out an authorizing user's policies governing access to a protected resource

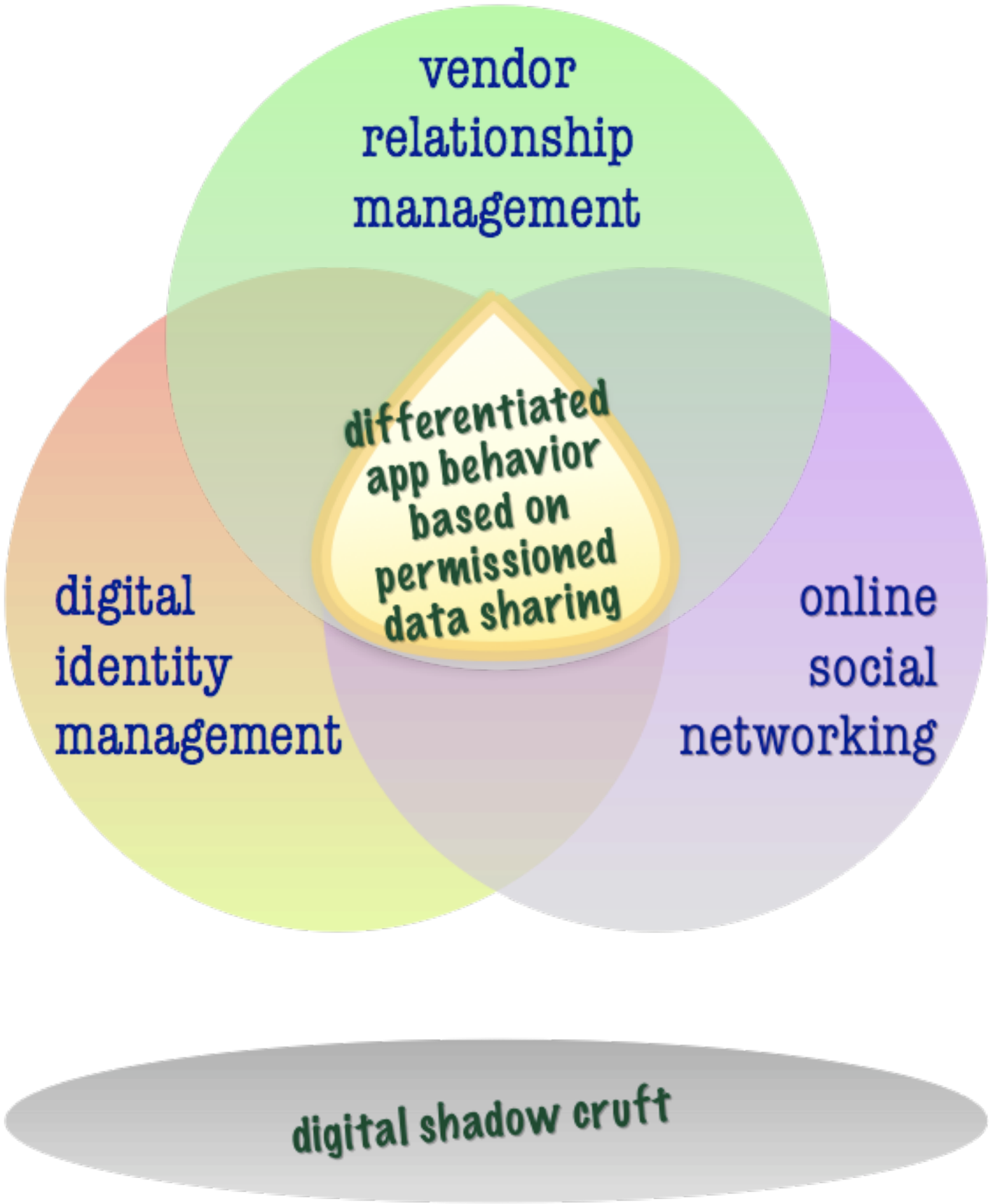
seeks access to a protected resource

requesting party: a web user, or a corporation (or other legal person), that uses a requester to seek access to a protected resource

Some starter scenarios



Landscape and requirements

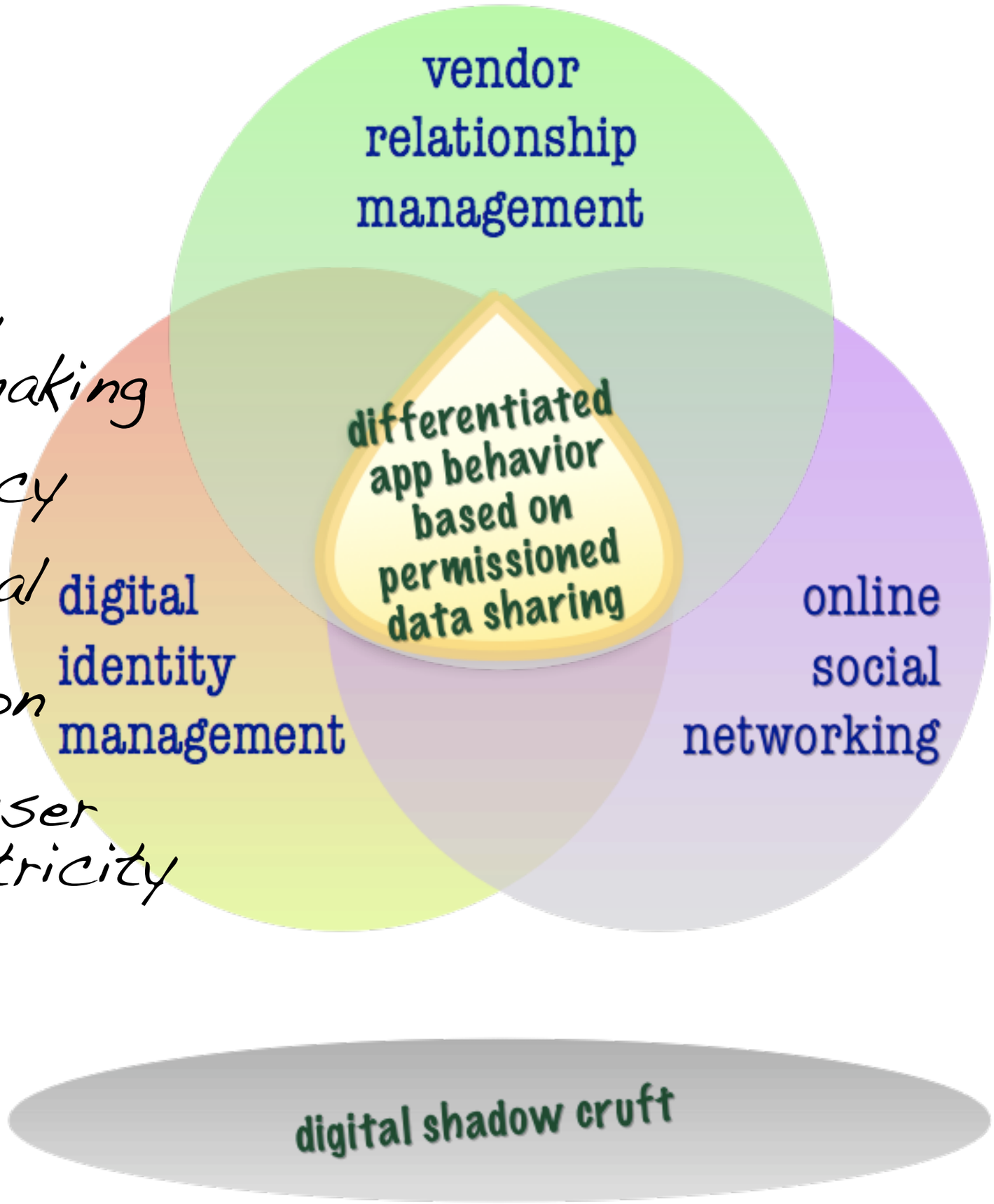

vendor
relationship
management
digital
identity
management
online
social
networking
differentiated
app behavior
based on
permissioned
data sharing
digital shadow cruft

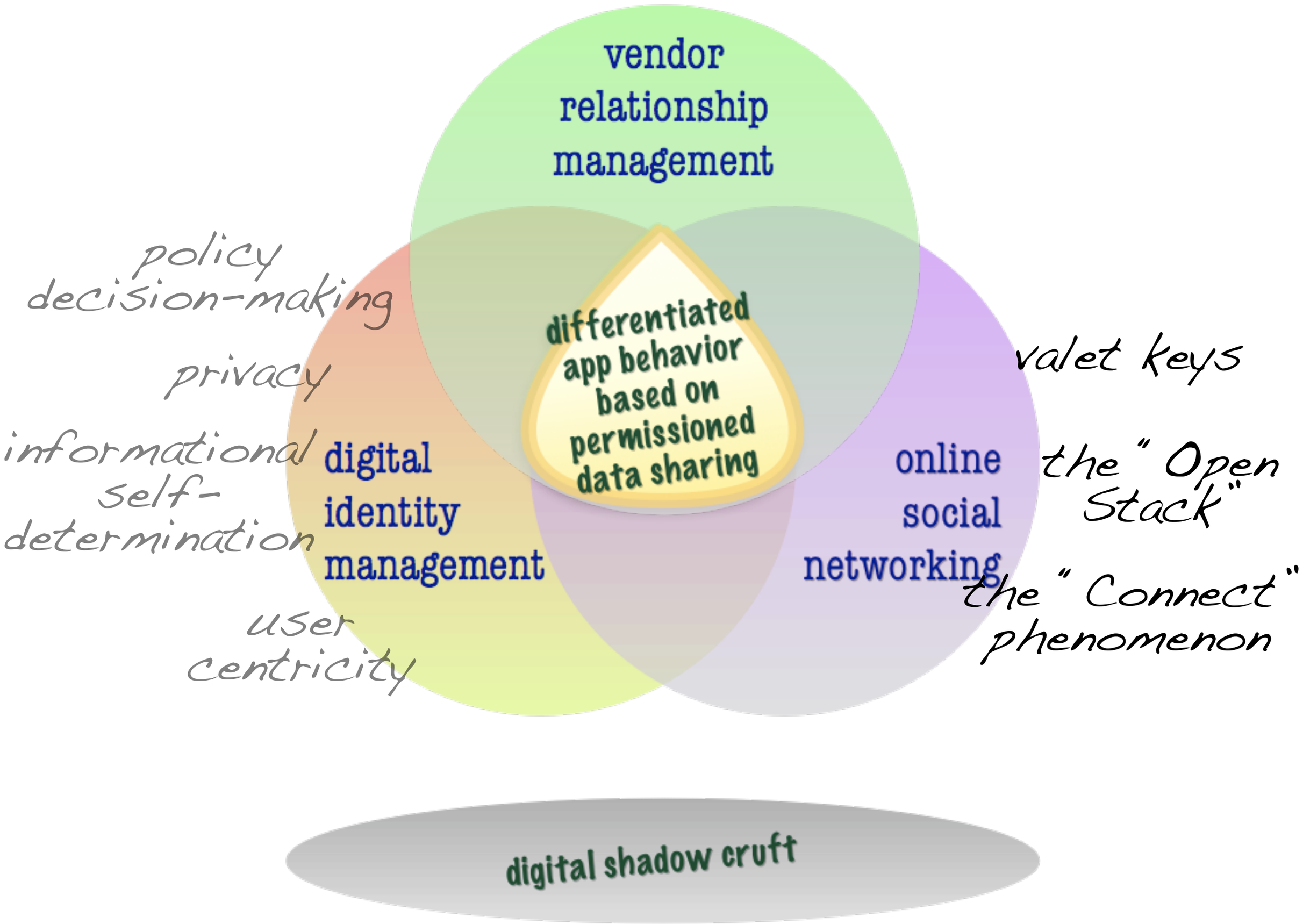
policy decision-making

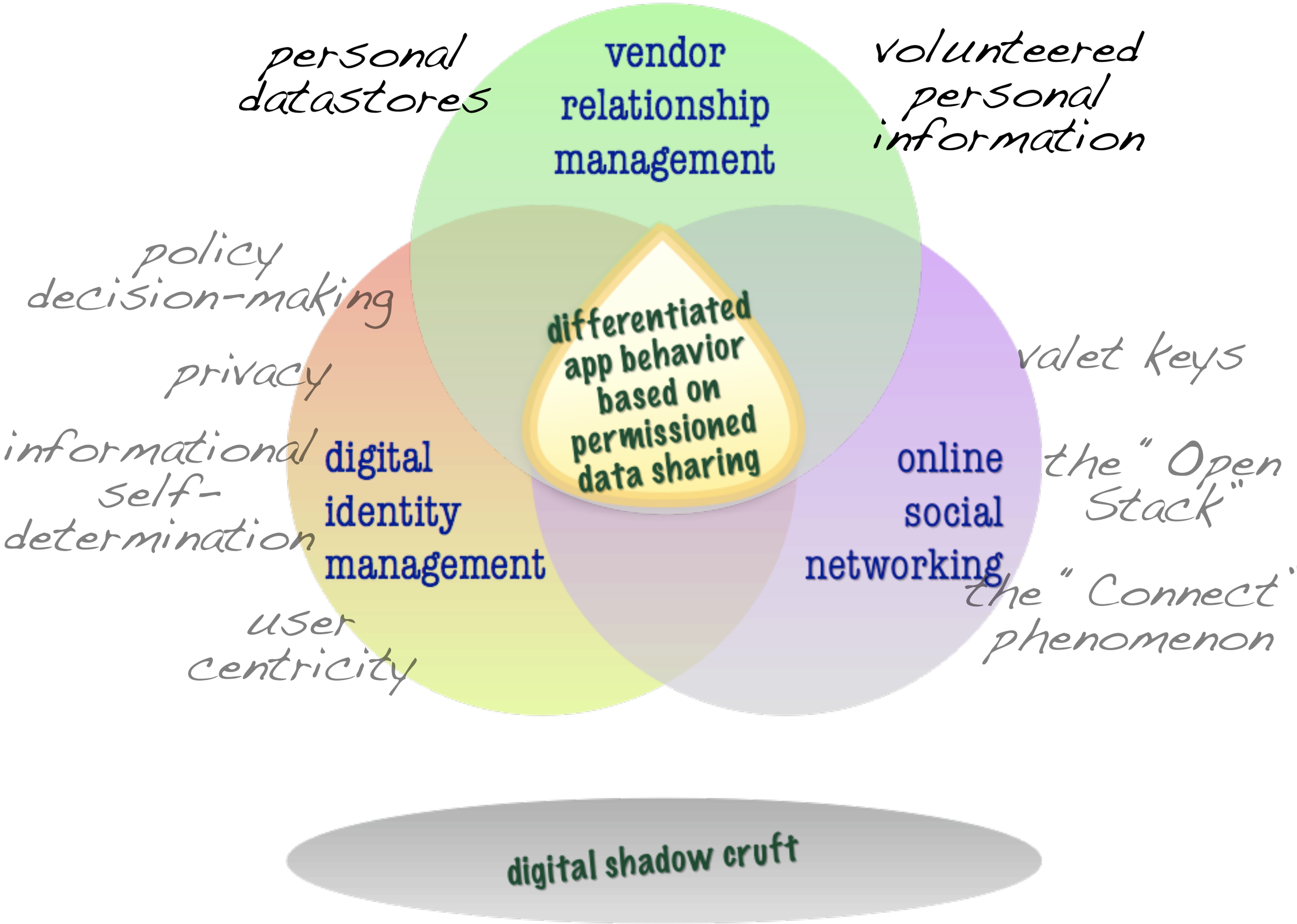
privacy

informational self-determination

user centricity




vendor relationship management
differentiated app behavior based on permissioned data sharing
digital identity management
online social networking
digital shadow cruft
policy decision-making
privacy
informational self-determination
user centricity
valet keys
the "Open Stack"
the "Connect" phenomenon


personal datastores
vendor relationship management
volunteered personal information
policy decision-making
privacy
informational self-determination
digital identity management
differentiated app behavior based on permissioned data sharing
valet keys
the "Open Stack"
online social networking
the "Connect" phenomenon
user centricity
digital shadow cruft

personal datastores

vendor relationship management

volunteered personal information

policy decision-making

privacy

differentiated app behavior based on permissioned data sharing

valet keys

informational self-determination

digital identity management

online social networking

the "Open Stack"

the "Connect" phenomenon

user centricity

intentional vs. behavioral data

digital shadow cruft

digital footprint dashboard

Provisioning data “by hand and by value” is broken

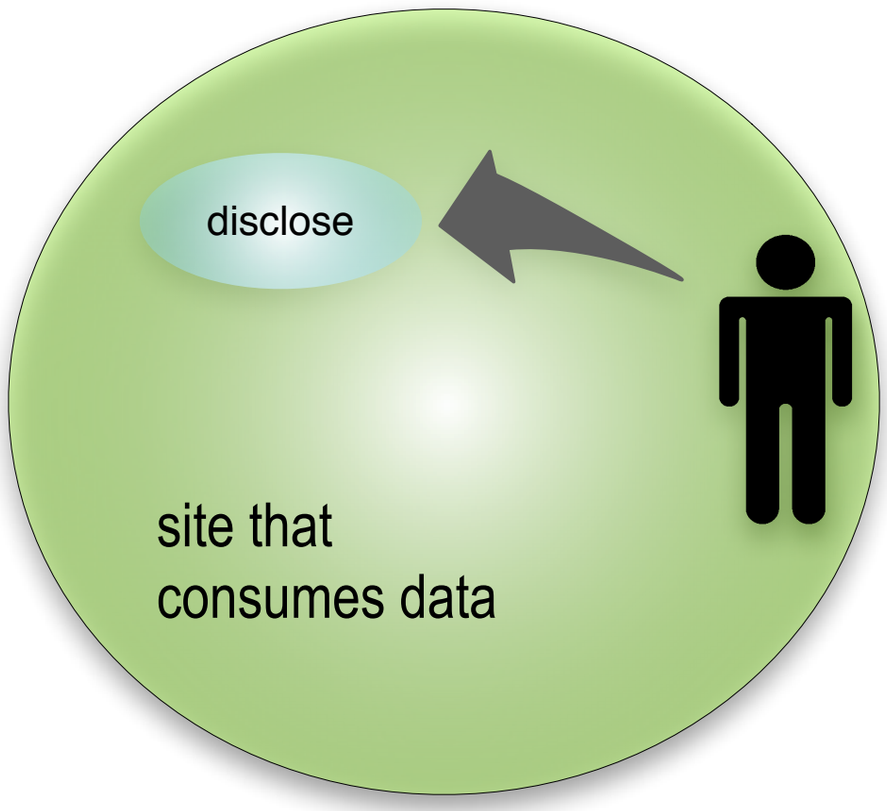
Name	
Street Address	
City	
State	Enter Text
Zip/Postal	
Province	
Country	Enter Text
Phone	
Email	
Preferred Communication	◯ Postal Mail ◯ Phone ◯ E-mail

Comparing models

Classic web form model

Comparing models

Classic web form model

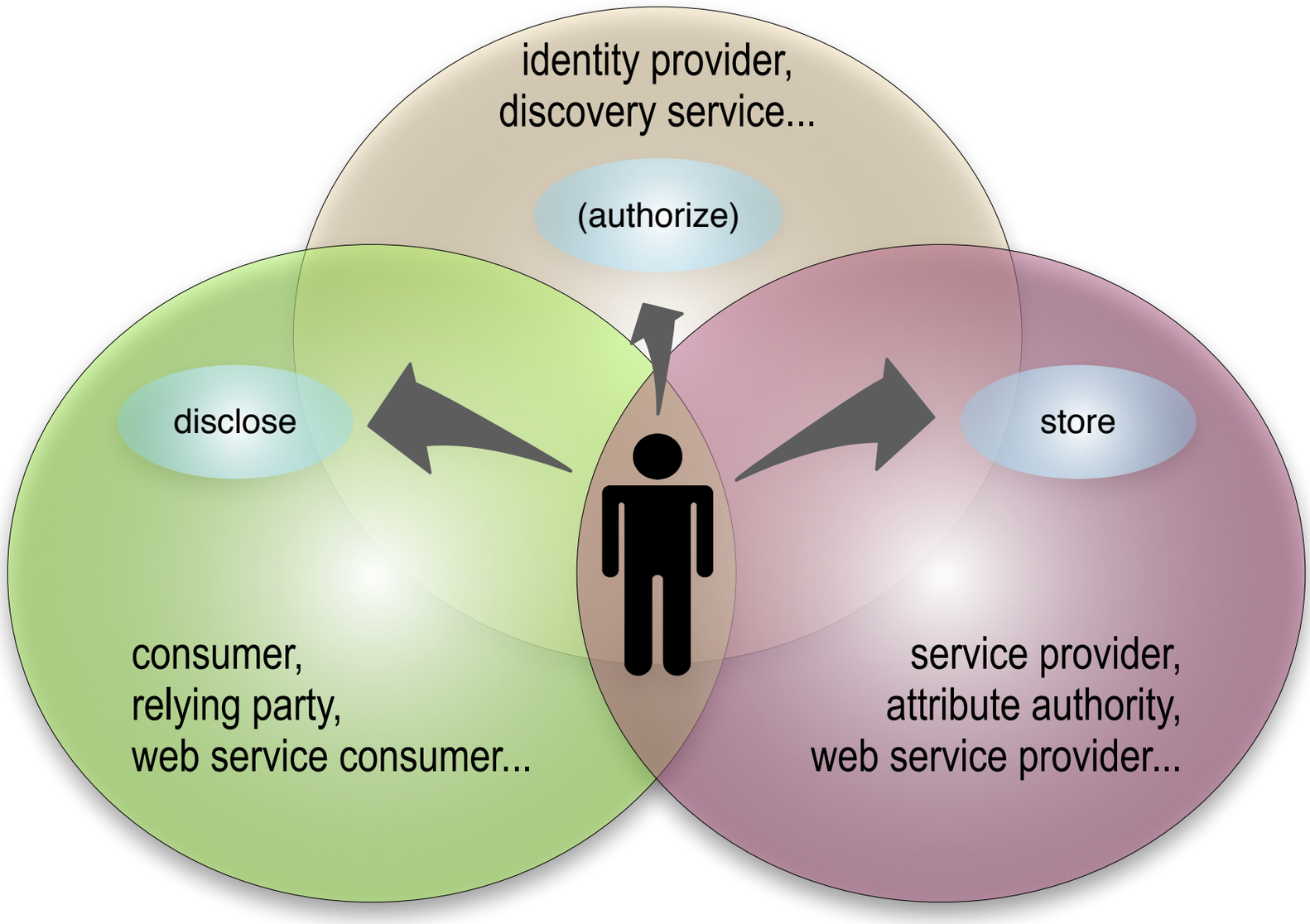


Comparing models

Classic federated identity model

Comparing models

Classic federated identity model

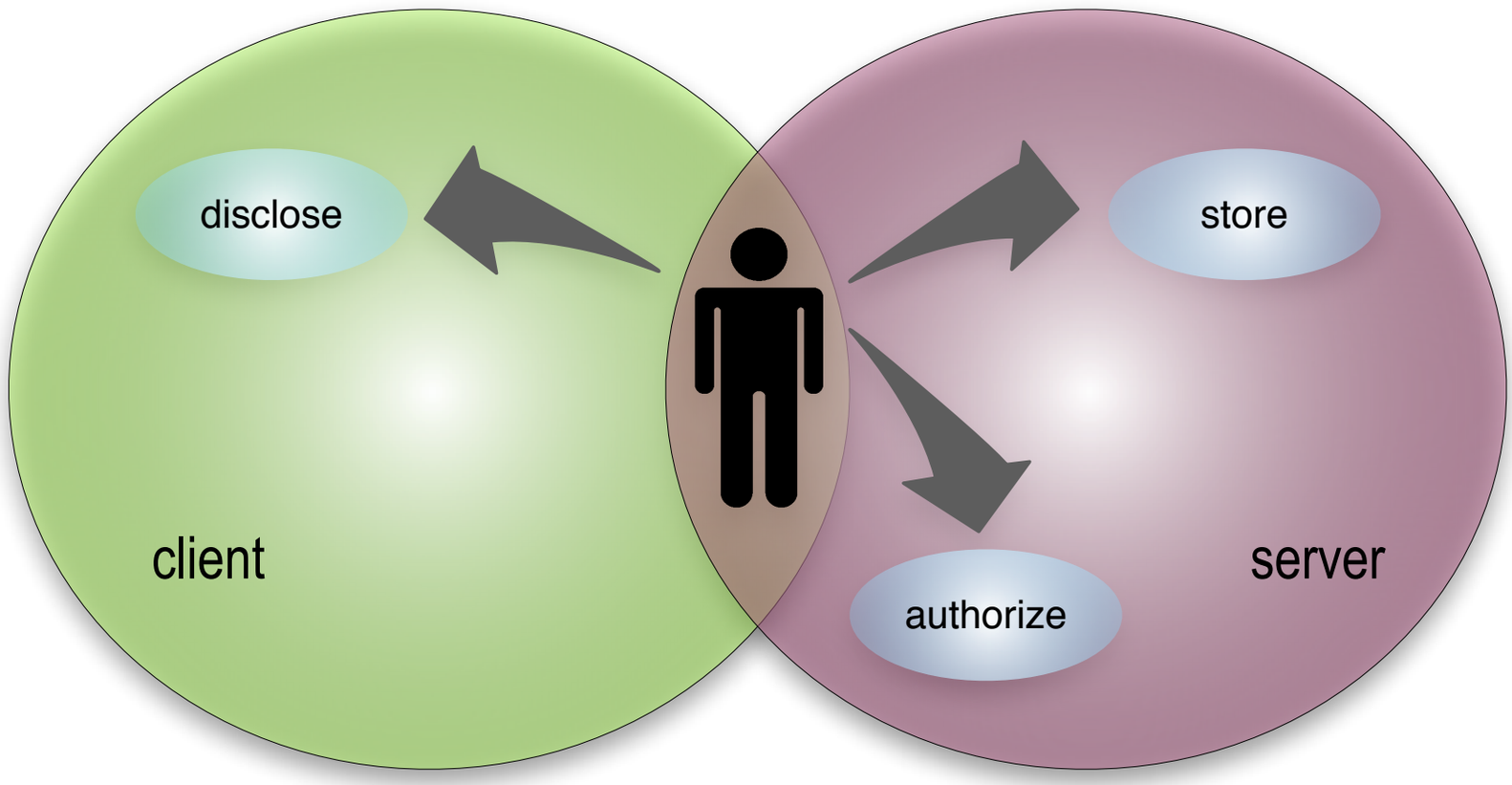


Comparing models

OAuth-style model

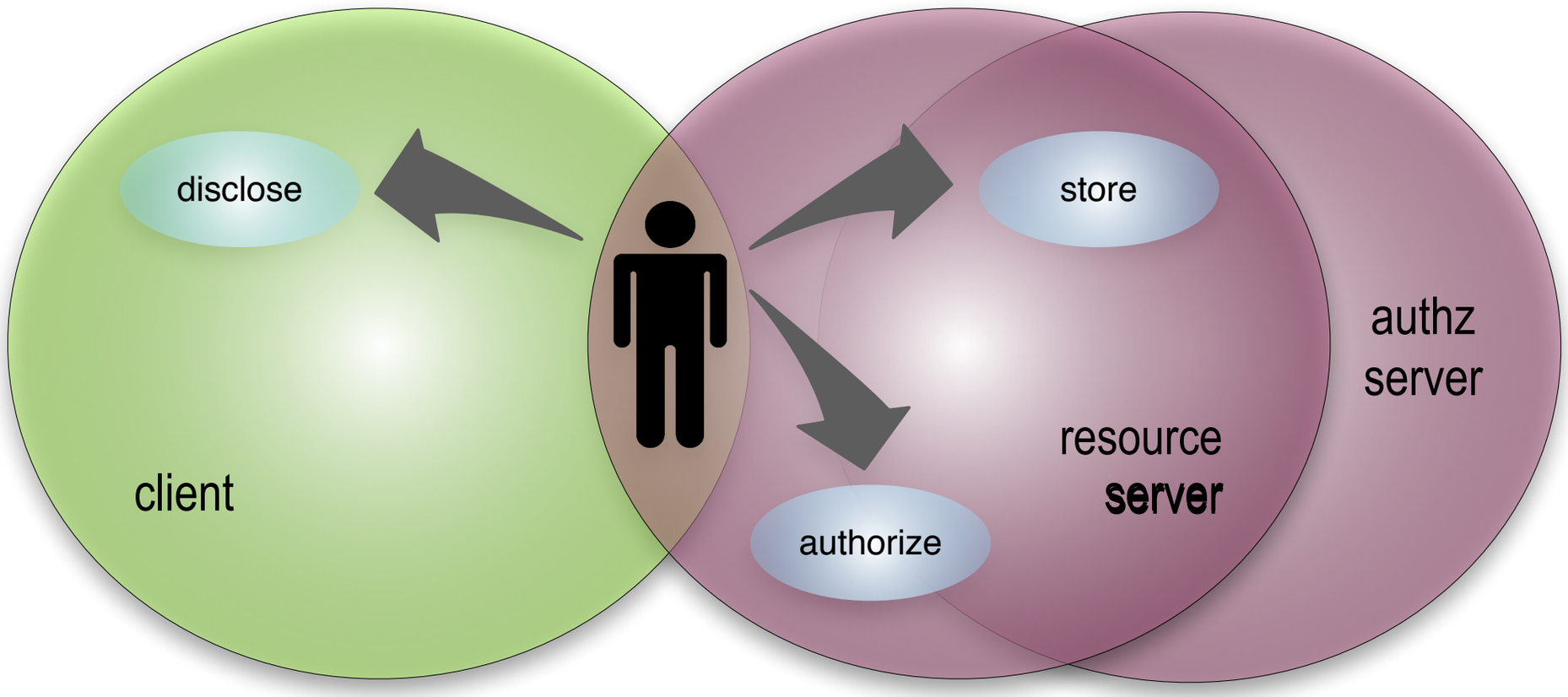
Comparing models

OAuth-style model



Comparing models

OAuth-style model

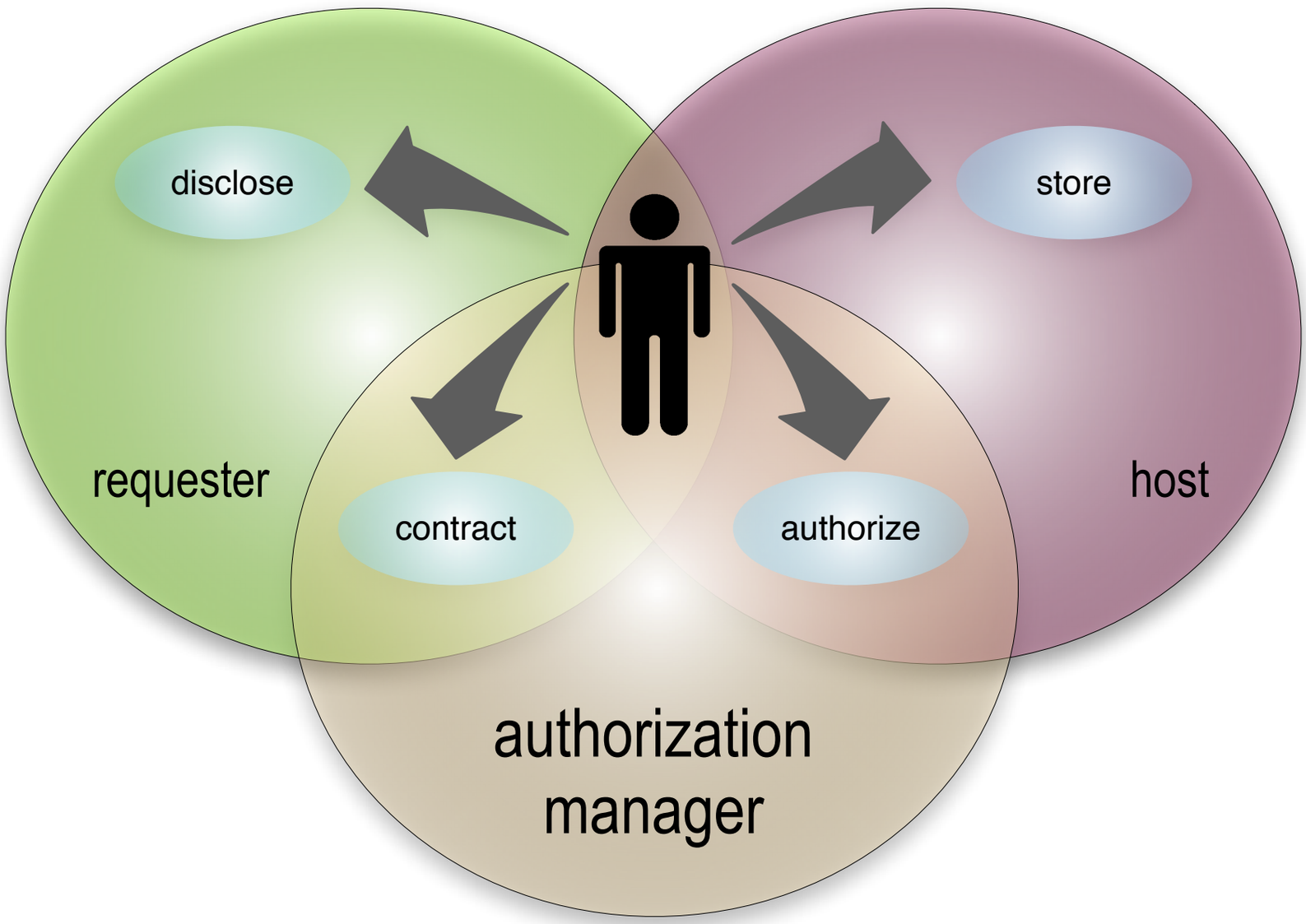


Comparing models

UMA model

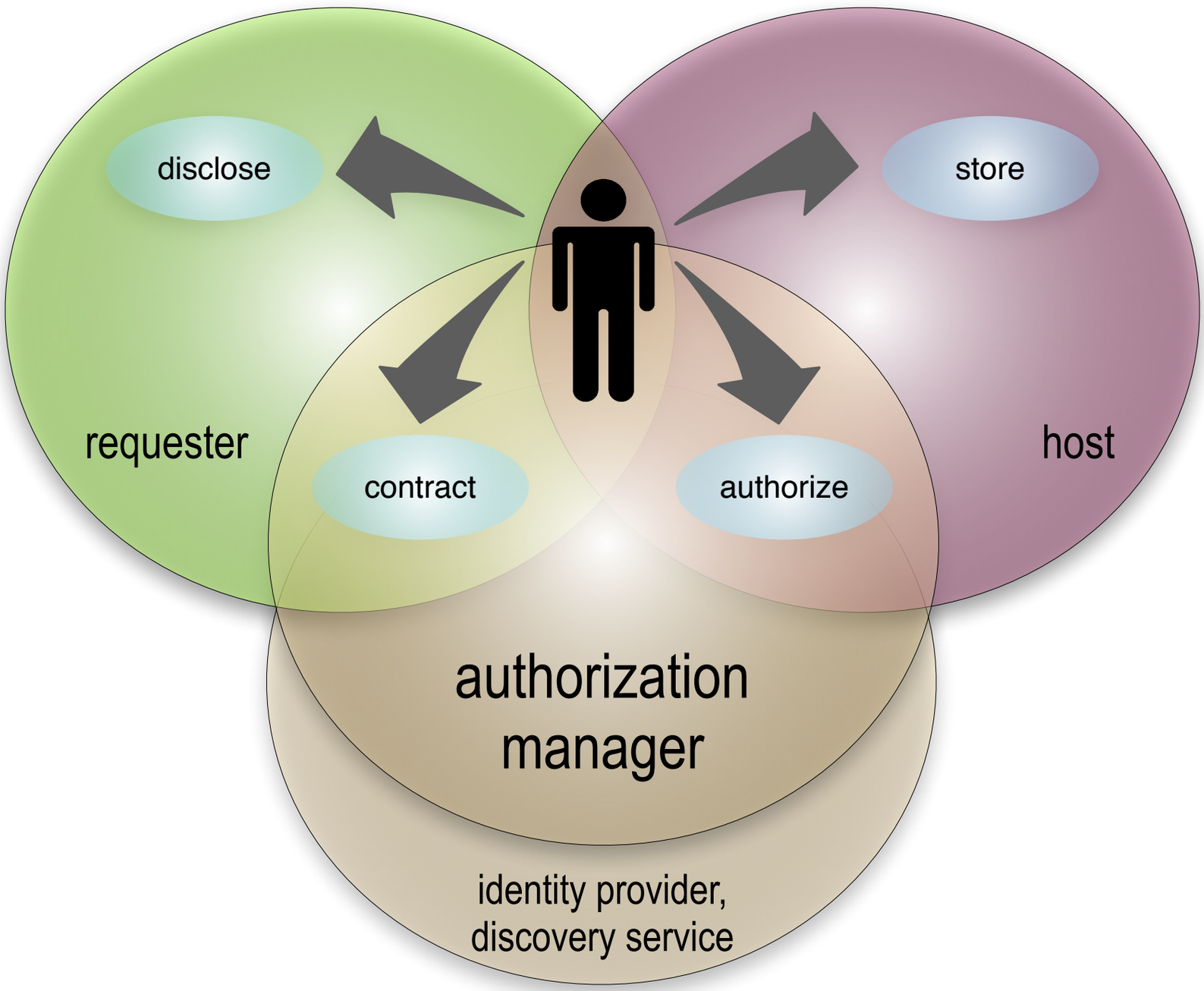
Comparing models

UMA model



Comparing models

UMA model

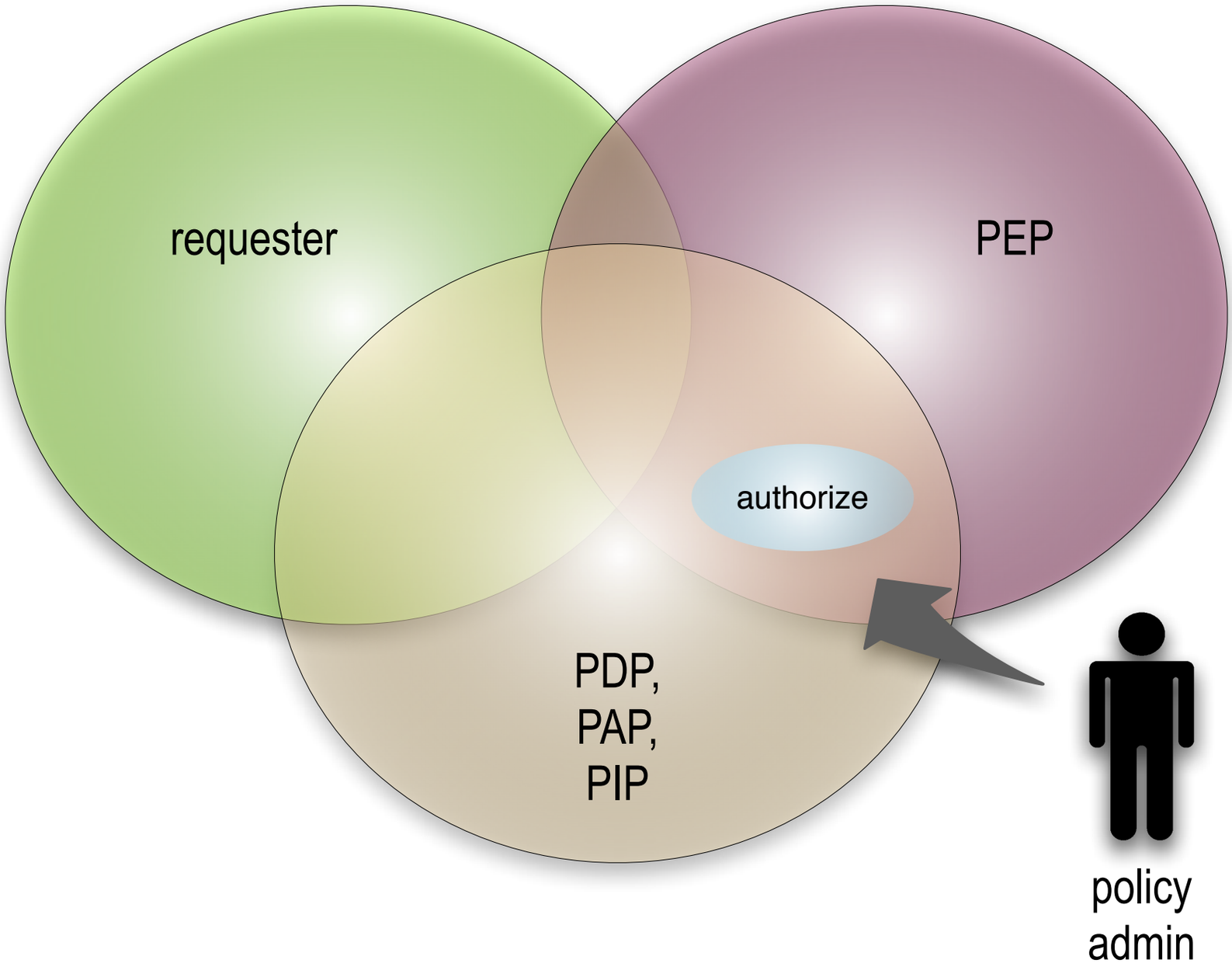


Comparing models

Classic access management model

Comparing models

Classic access management model



Design principles and requirements

(see also requirements doc)

- **Original DPs:** simple; OAuth; ID-agnostic; RESTful; modular; generative; fast
- **Emergent DPs:** cryptography; privacy; complexity; authentication; user experience
- **Original reqs:** access relationship service; user-driven policies and terms; user management of relationship; auditing; direct access; multiple hosts
- **Emergent reqs:** host/AM separation; resource orientation; correlation of authorizing user by multiple hosts; representation-agnostic AM

User experiences, scenarios, and use cases

User experiences imagined – and real

- Newcastle University SMART project (presented by Maciej Machulak)
- "Vision wireframes" (thanks to Domenico Catalano)

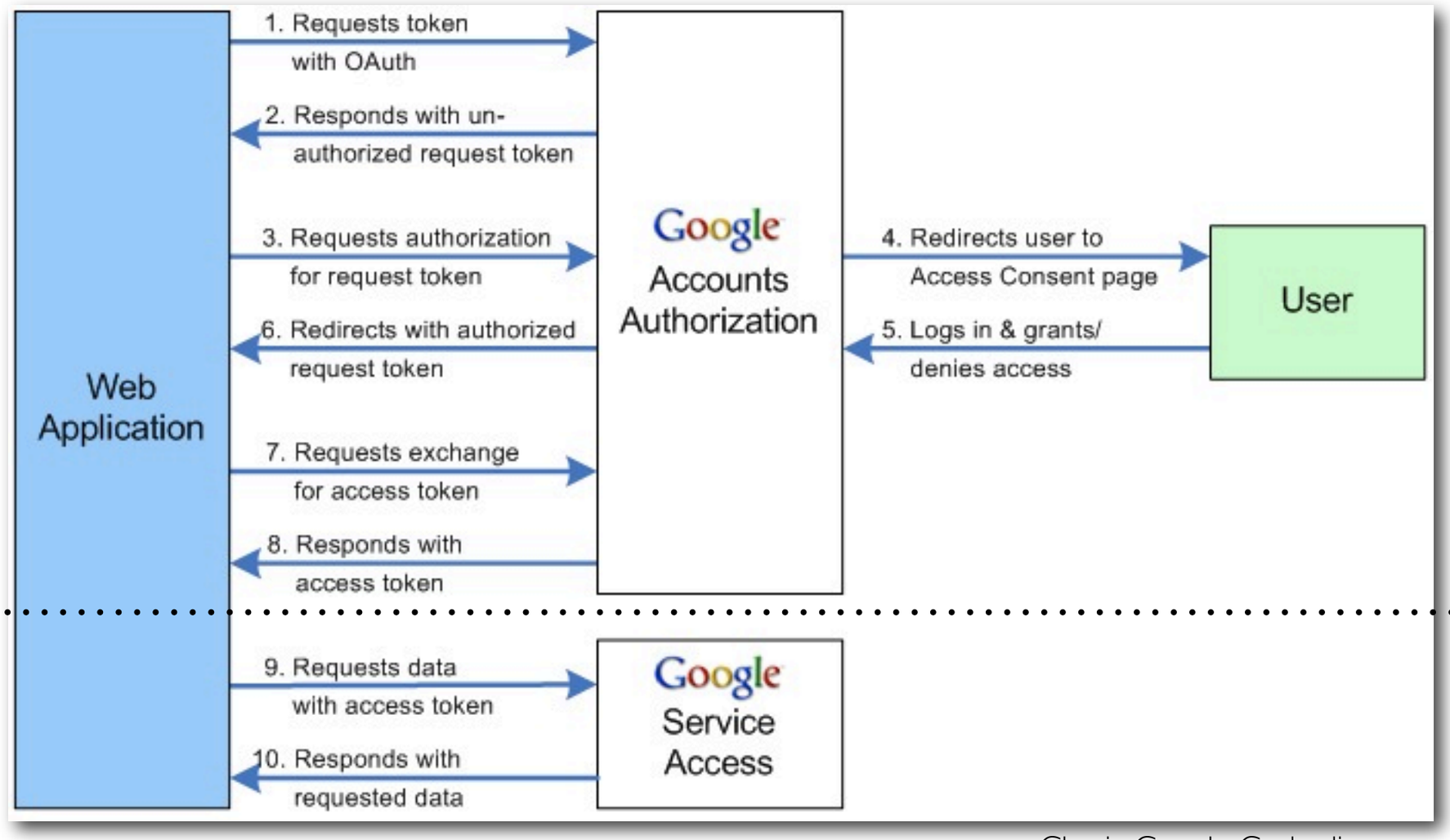
A sampling of scenarios

(see also scenarios and use cases doc)

- Sharing a Calendar with Vendors (Accepted)
- Packaging Resources for E-Commerce Vendors (Accepted)
- Online Personal Loan request scenario (Accepted)
- Distributed Services (Pending)
- Managing Information in Which Employers and Employees Both Have a Stake (Pending)
- Delegating Access Management to Custodians (Pending)
- Moving Resources Between Hosts (Pending)
- Controlling Access to Health Data (Pending)

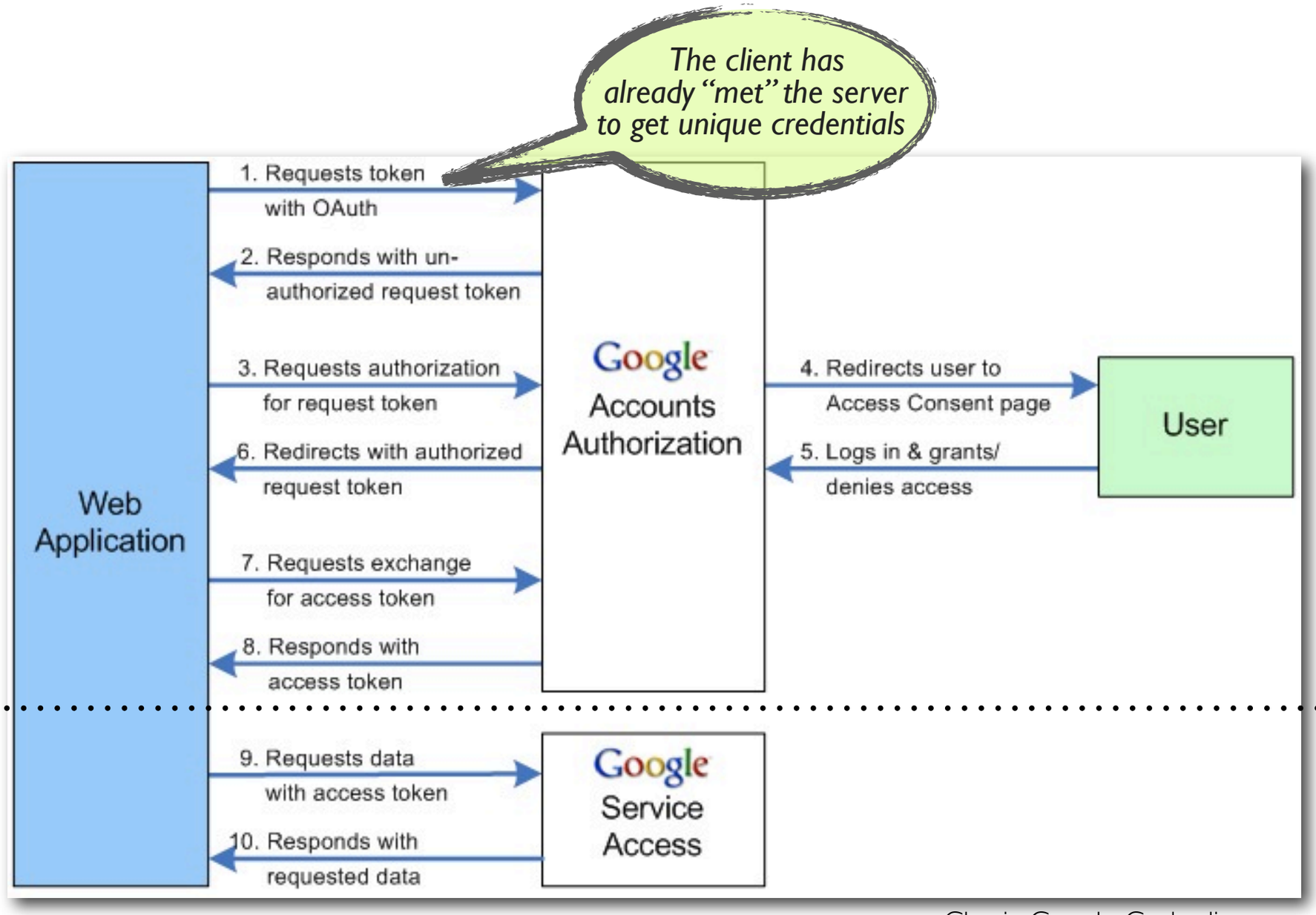
The protocol

First, a little bit about OAuth



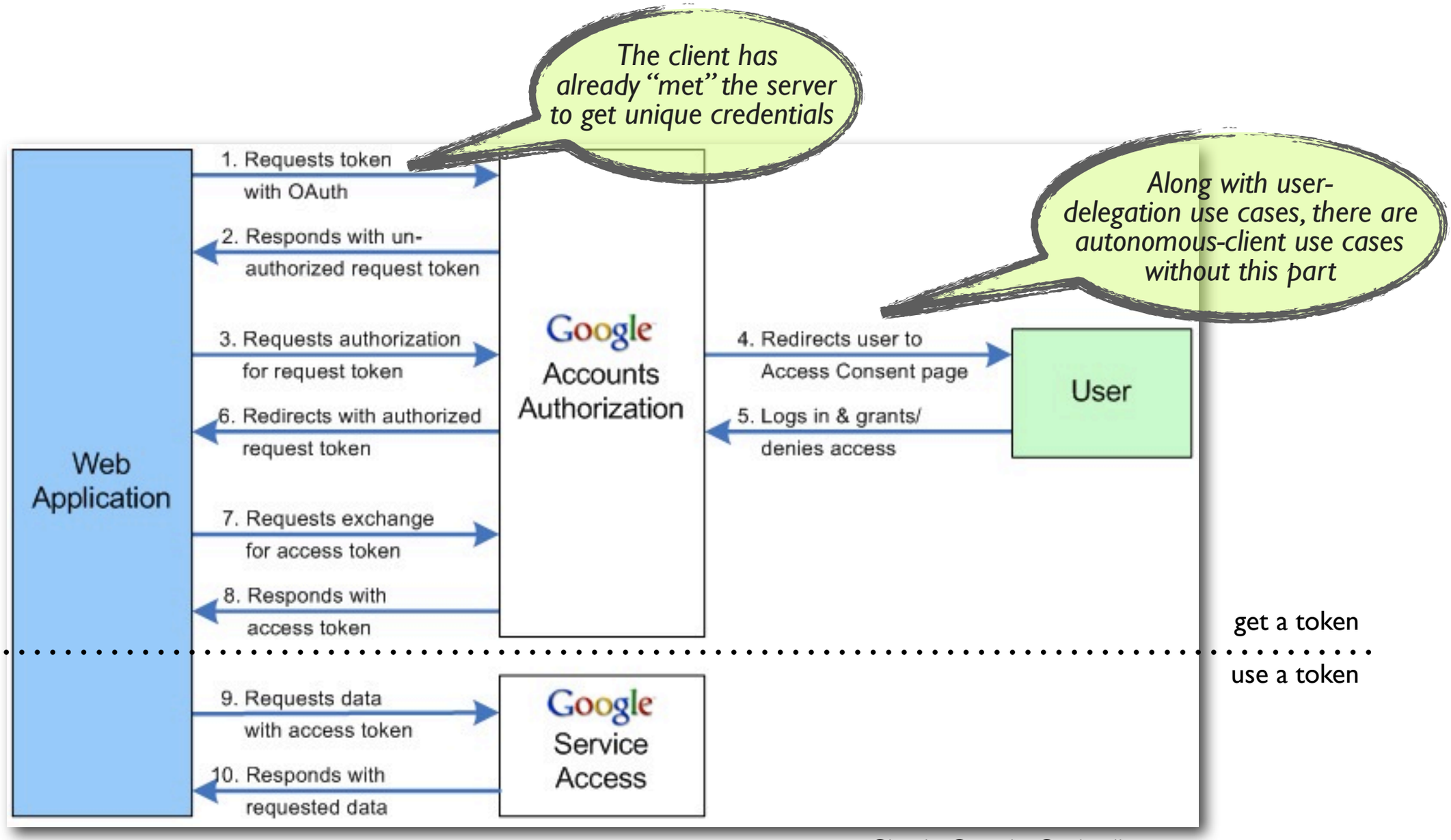
Classic Google Code diagram

First, a little bit about OAuth



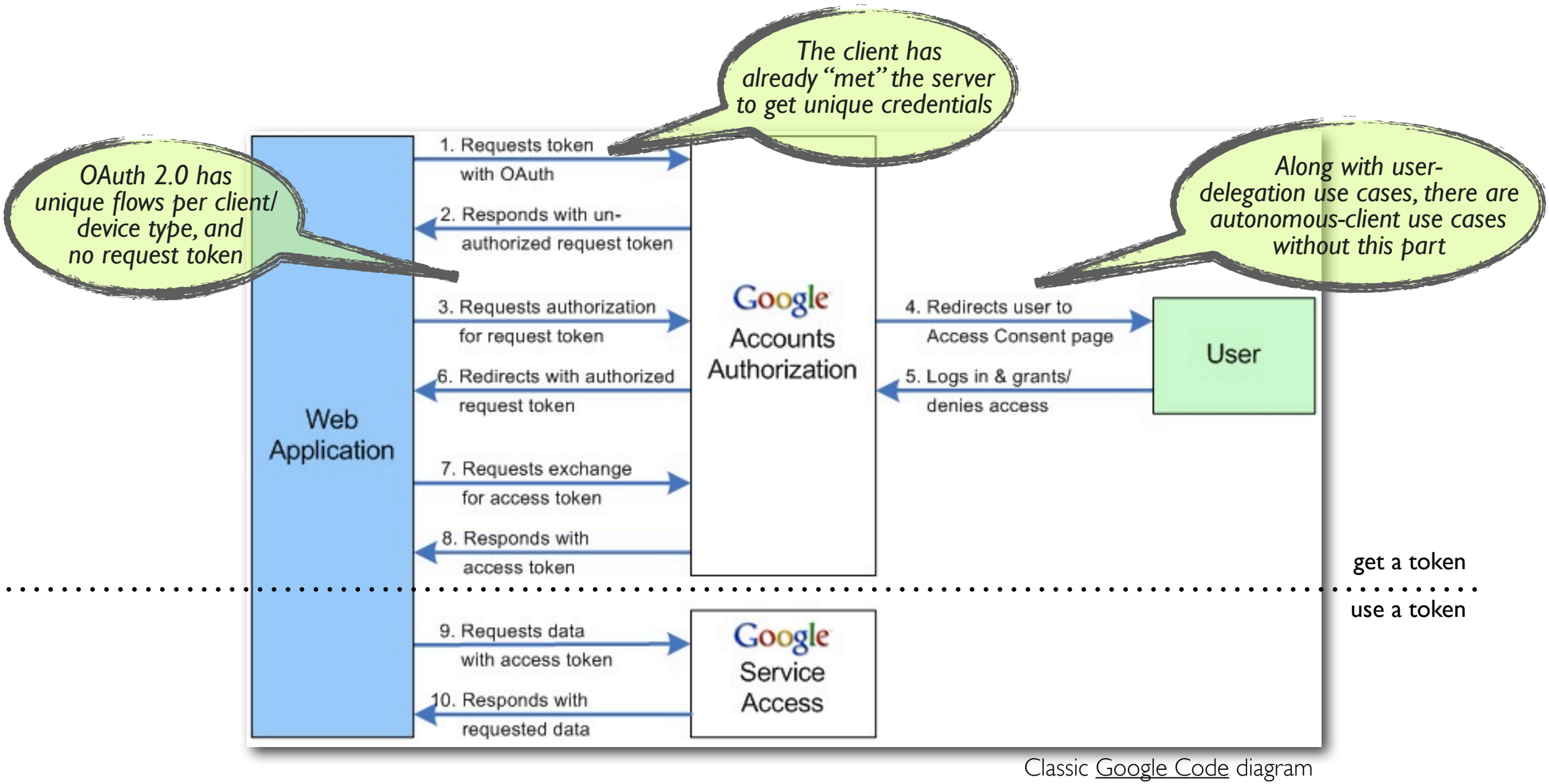
Classic Google Code diagram

First, a little bit about OAuth



Classic Google Code diagram

First, a little bit about OAuth



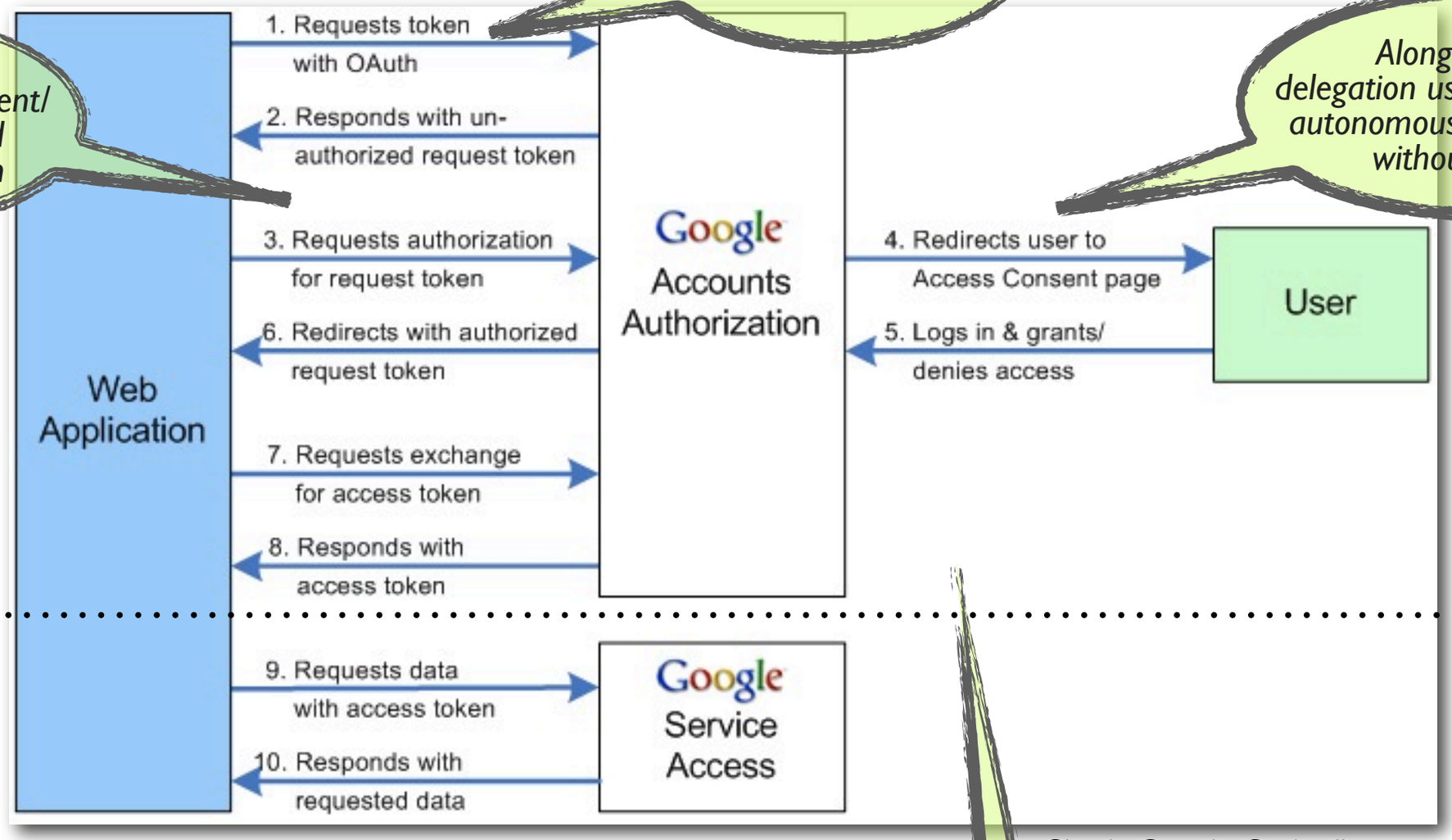
Classic Google Code diagram

First, a little bit about OAuth

The client has already "met" the server to get unique credentials

OAuth 2.0 has unique flows per client/ device type, and no request token

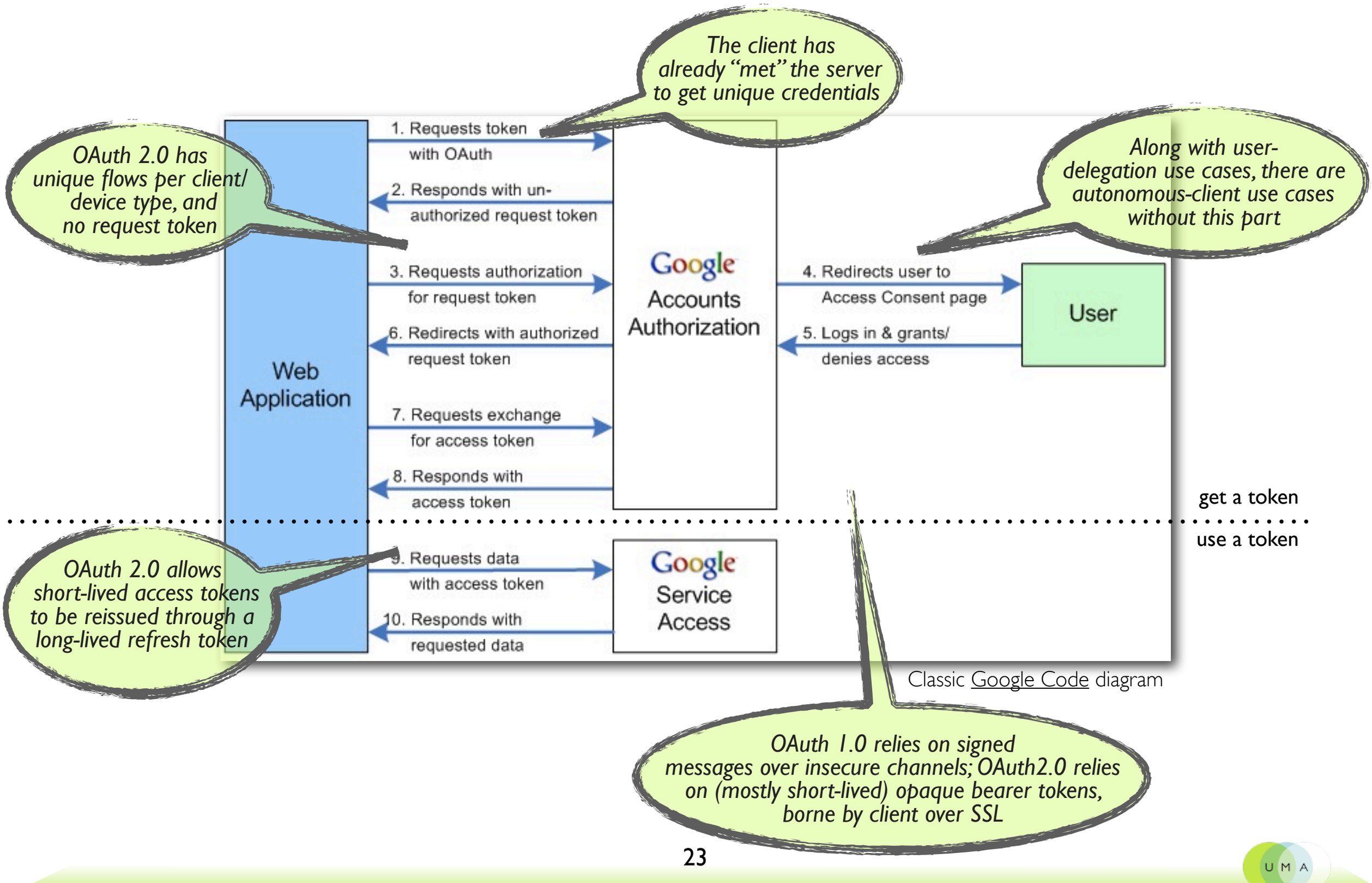
Along with user-delegation use cases, there are autonomous-client use cases without this part



Classic Google Code diagram

OAuth 1.0 relies on signed messages over insecure channels; OAuth2.0 relies on (mostly short-lived) opaque bearer tokens, borne by client over SSL

First, a little bit about OAuth



Classic Google Code diagram

UMA's history with OAuth

we're right about here

ProtectServe

1.0

UMA

1.0

UMA

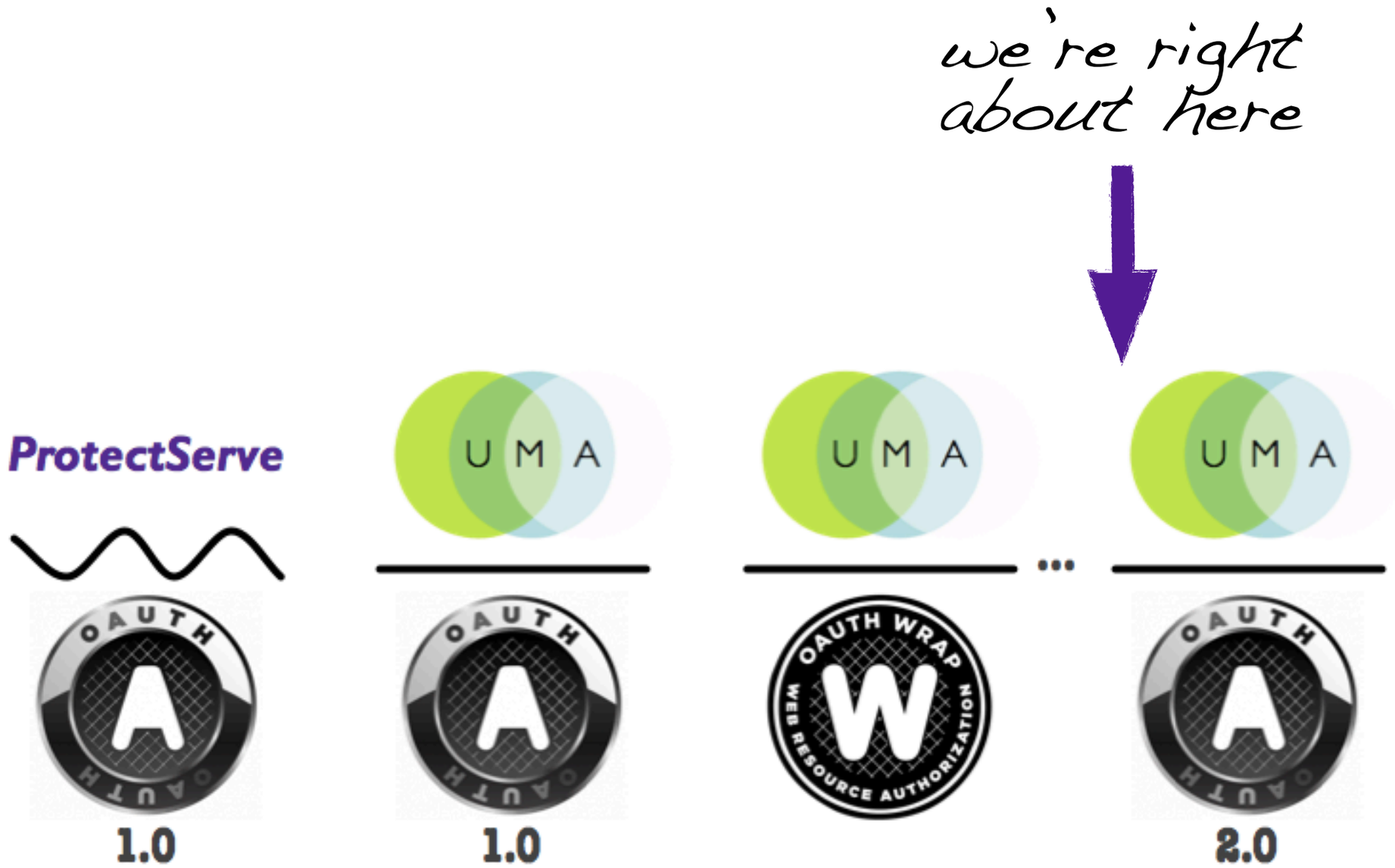
OAUTH WRAP
WEB RESOURCE AUTHORIZATION

...

UMA

2.0

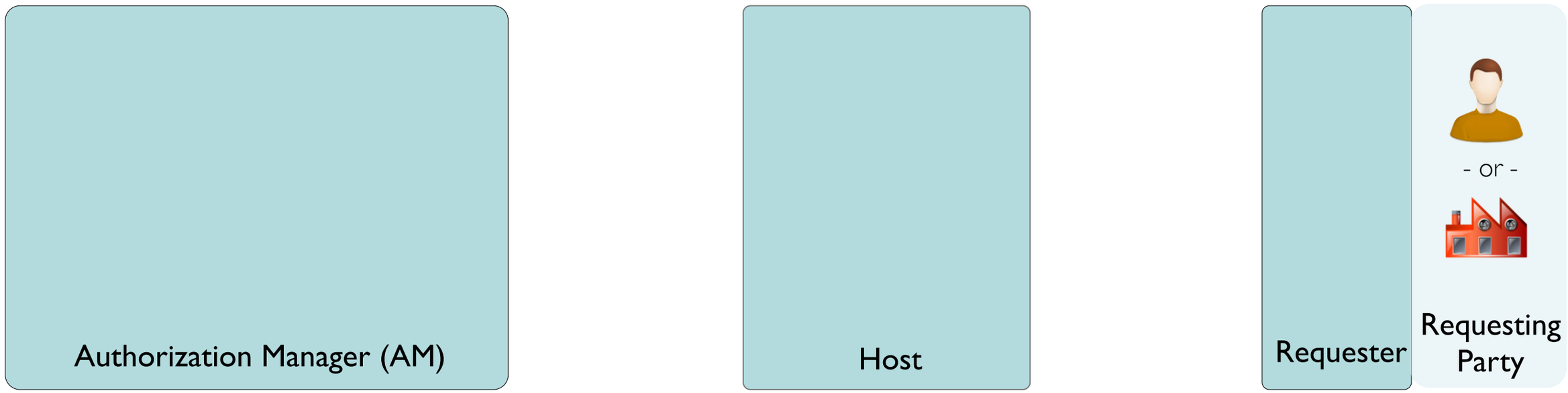
UMA's history with OAuth



Comparing OAuth2 and UMA

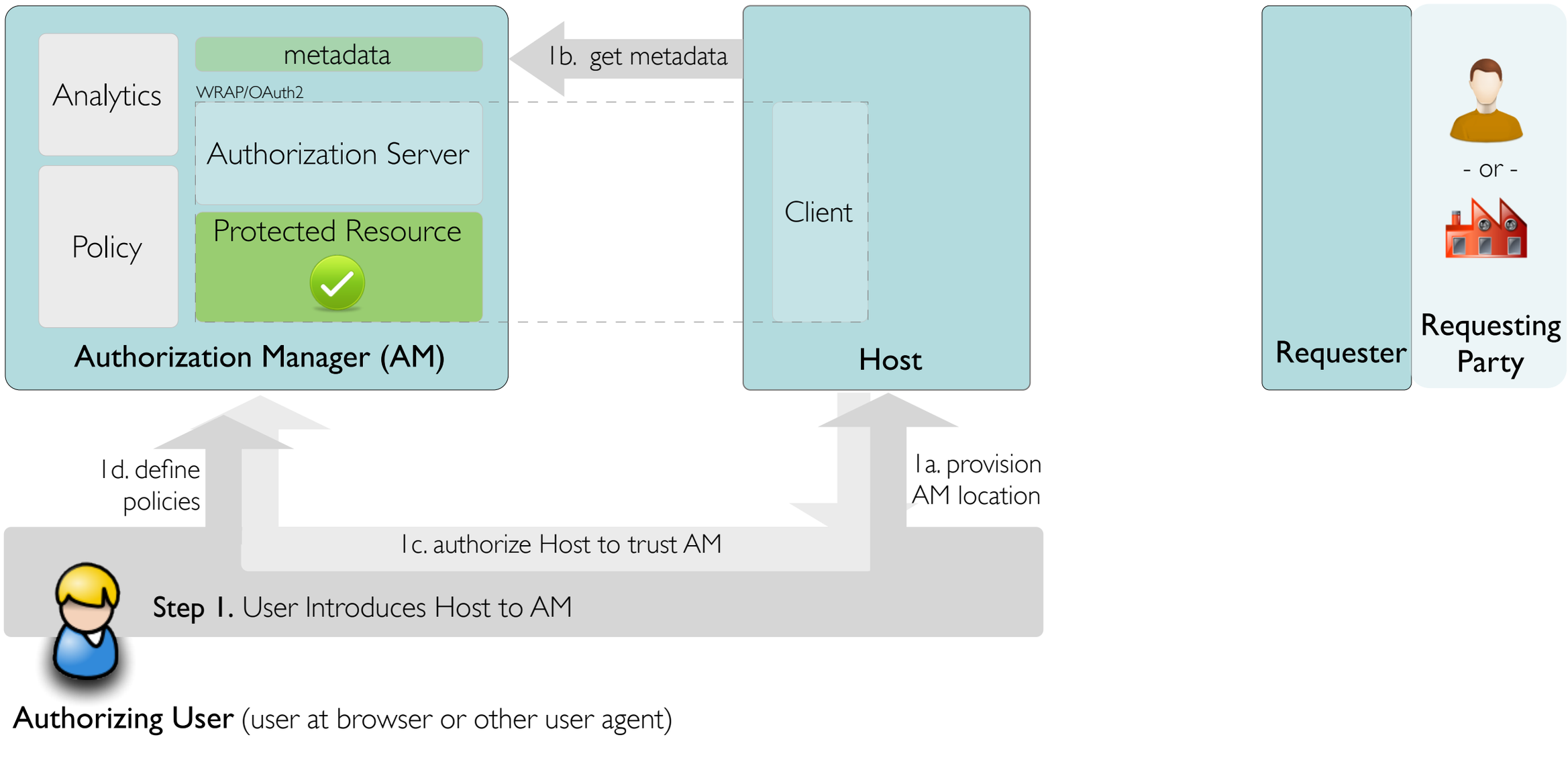
• Resource owner	➡ Authorizing user
• Resource server	➡ Host
• Authorization server	➡ Authorization manager
• Client	➡ Requester
• The two servers meet out of band	➡ Host and AM can meet dynamically (using OAuth!)
• Authz is binary: you get a token or you don't	➡ Authz can depend on requester "claims"
• Token validation process is unspecified	➡ Host can ask AM to validate token at run time

The UMA protocol in a nutshell: trust a token, get a token, use a token

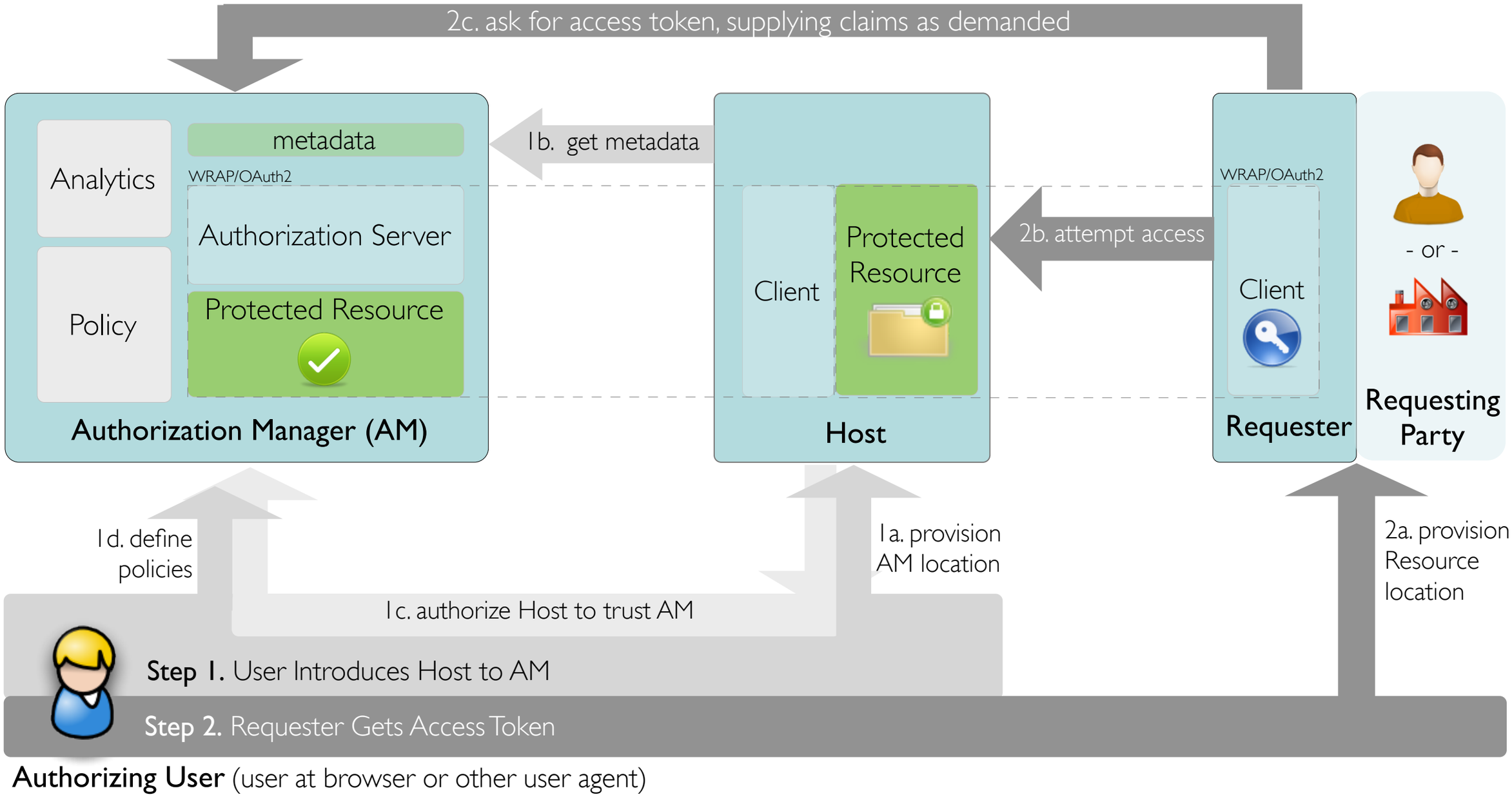


Authorizing User (user at browser or other user agent)

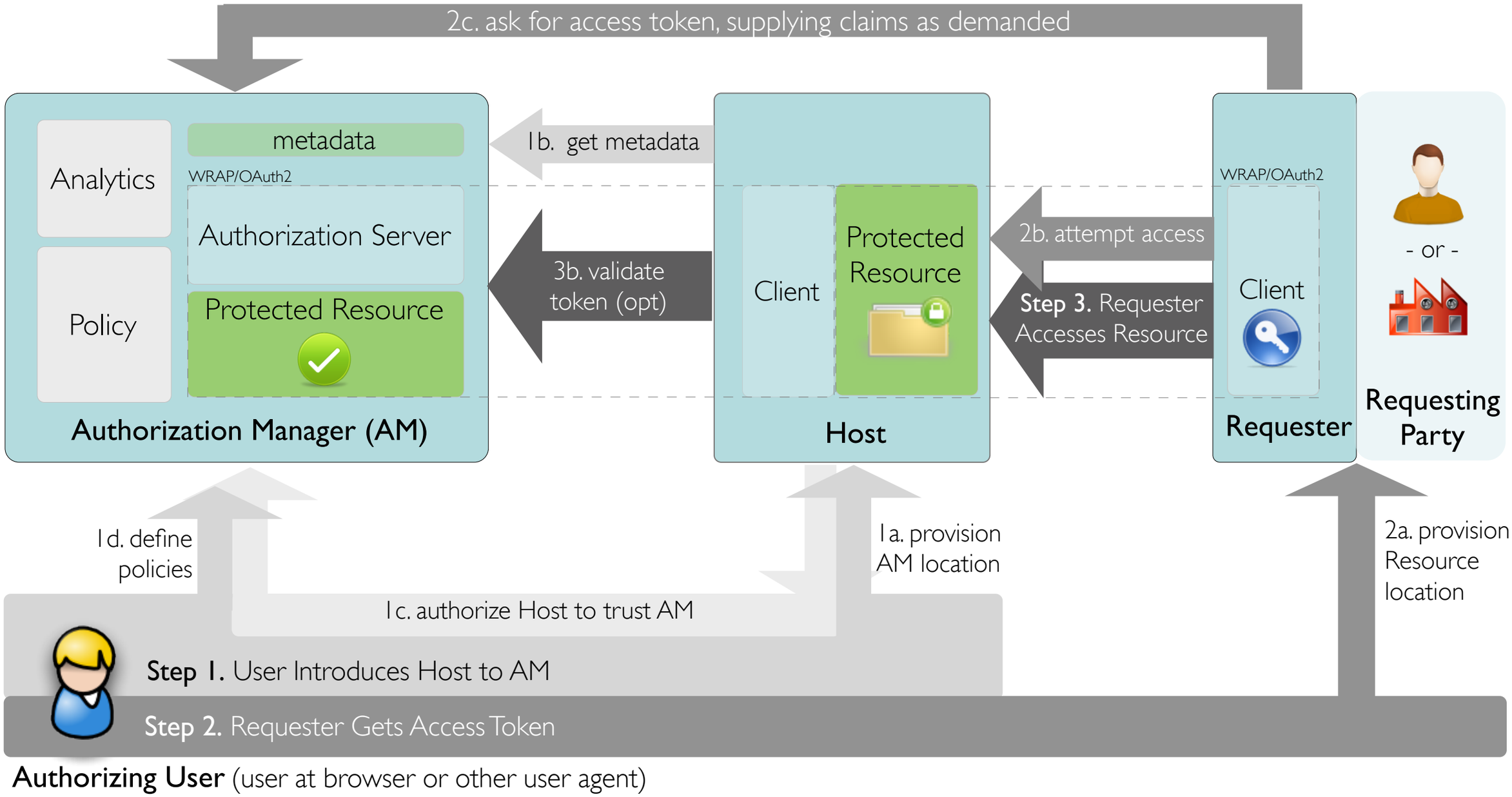
The UMA protocol in a nutshell: trust a token, get a token, use a token



The UMA protocol in a nutshell: trust a token, get a token, use a token



The UMA protocol in a nutshell: trust a token, get a token, use a token



Some current protocol issues

(see also protocol issues list)

- Spec modularity
- Flows for getting a token: user delegation? autonomous client? others?
- Token validation models
- Claims about requesting-party identity
- Inter-entity info discovery
- Resource basket optimizations

Policies, claims, and agreements

Policies can be unilateral or can require claims

- Unilateral:
  - "Allow access for a week"
- Claims-requiring:
  - "Allow access to anyone who agrees to my licensing terms" (*promissory* statement)
  - "Allow access to someone who can prove themselves to be bob@gmail.com" (*affirmative* statement)
  - "Allow access to anyone who says they're 18 or older" (*affirmative* statement)

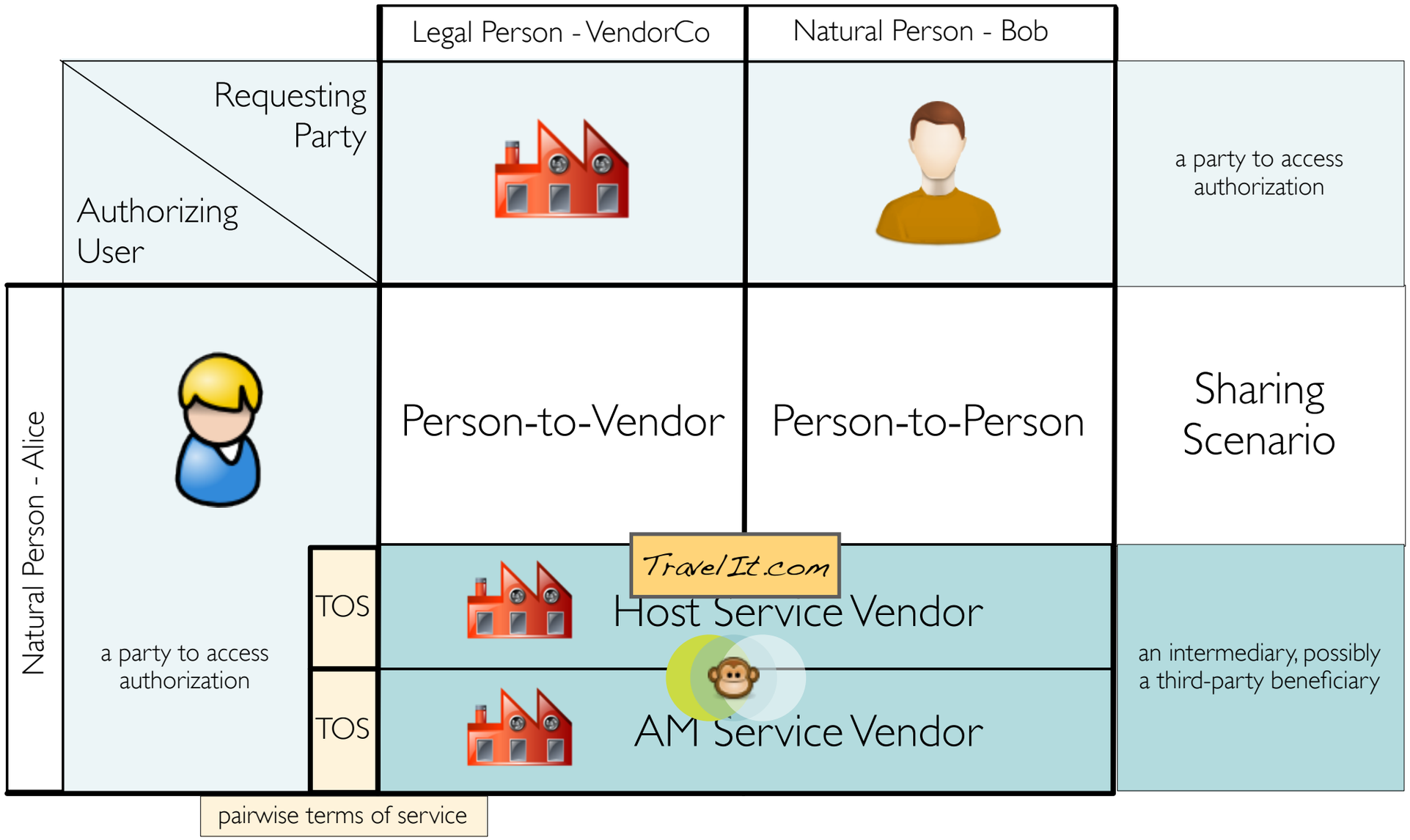
Claims are about requesting parties (not requesters)

AIRPLANR

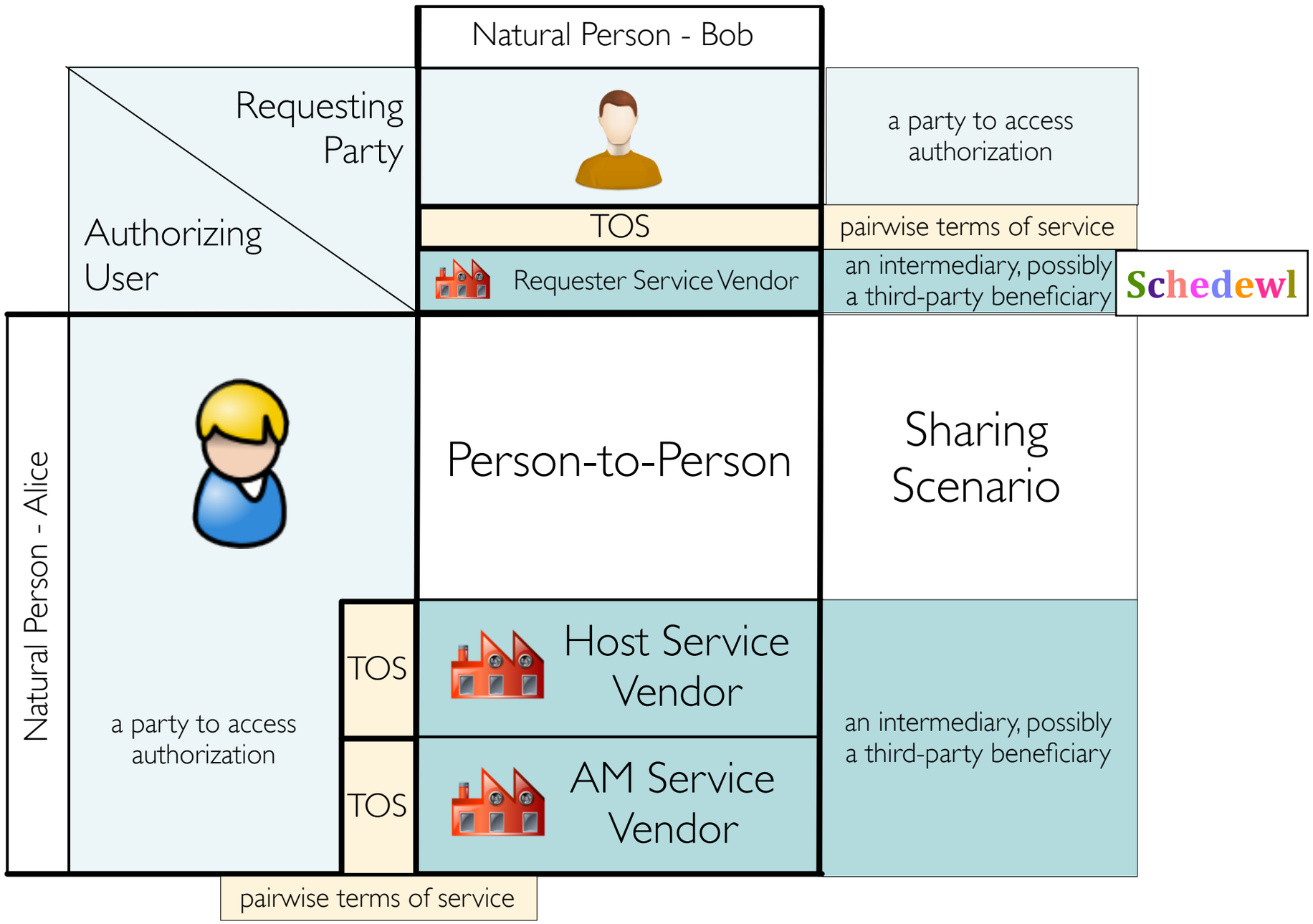
Hi, I'm Bob Baker.

Requesting Party / Authorizing User	Legal Person - VendorCo	Natural Person - Bob	a party to access authorization
Natural Person - Alice	*Hi, I'm Alice Adams.* a party to access authorization	Person-to-Vendor	Person-to-Person	Sharing Scenario

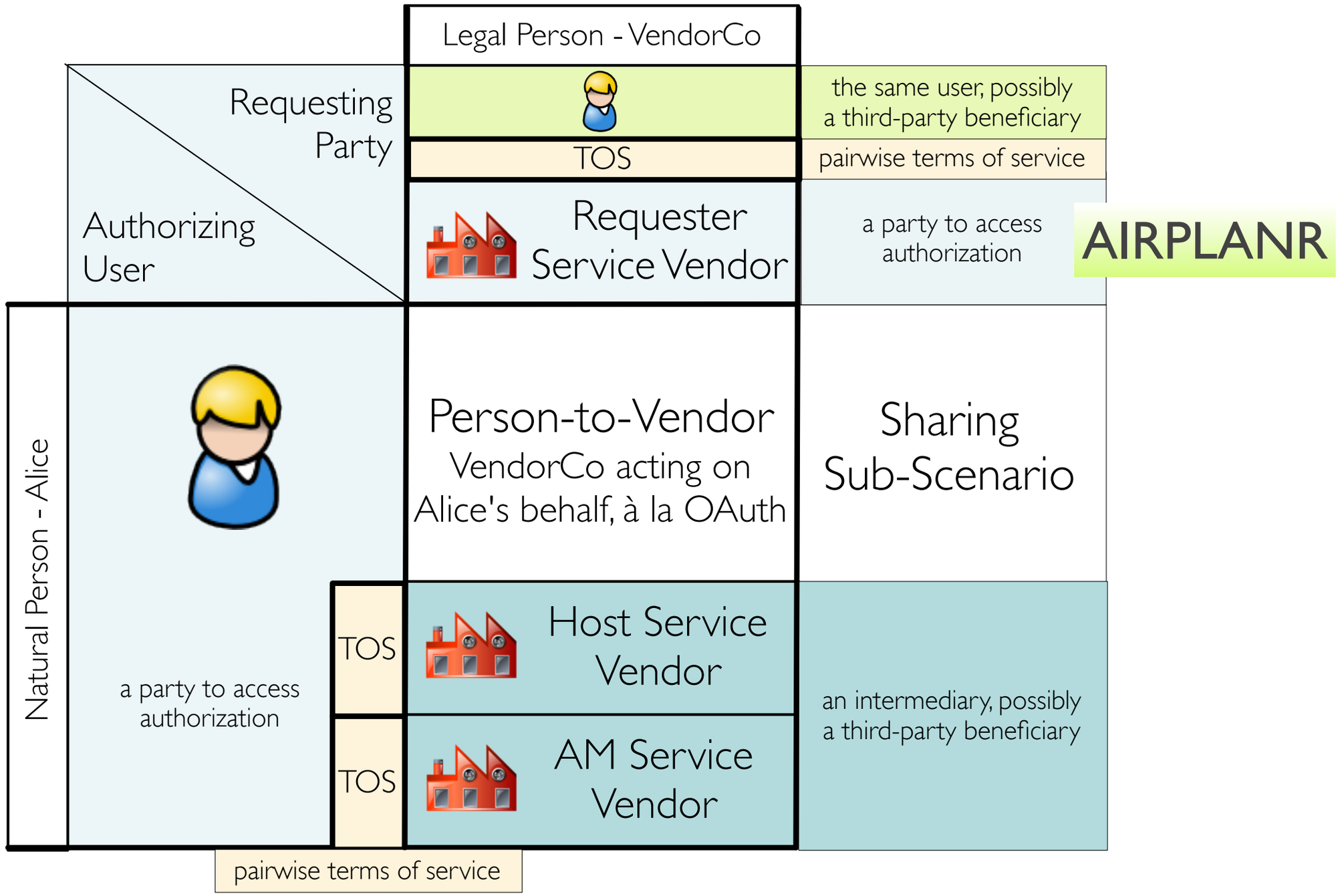
Alice uses some intermediary services



Bob uses intermediary services too



When the requesting party is a company...



Or the company might be acting on its own behalf