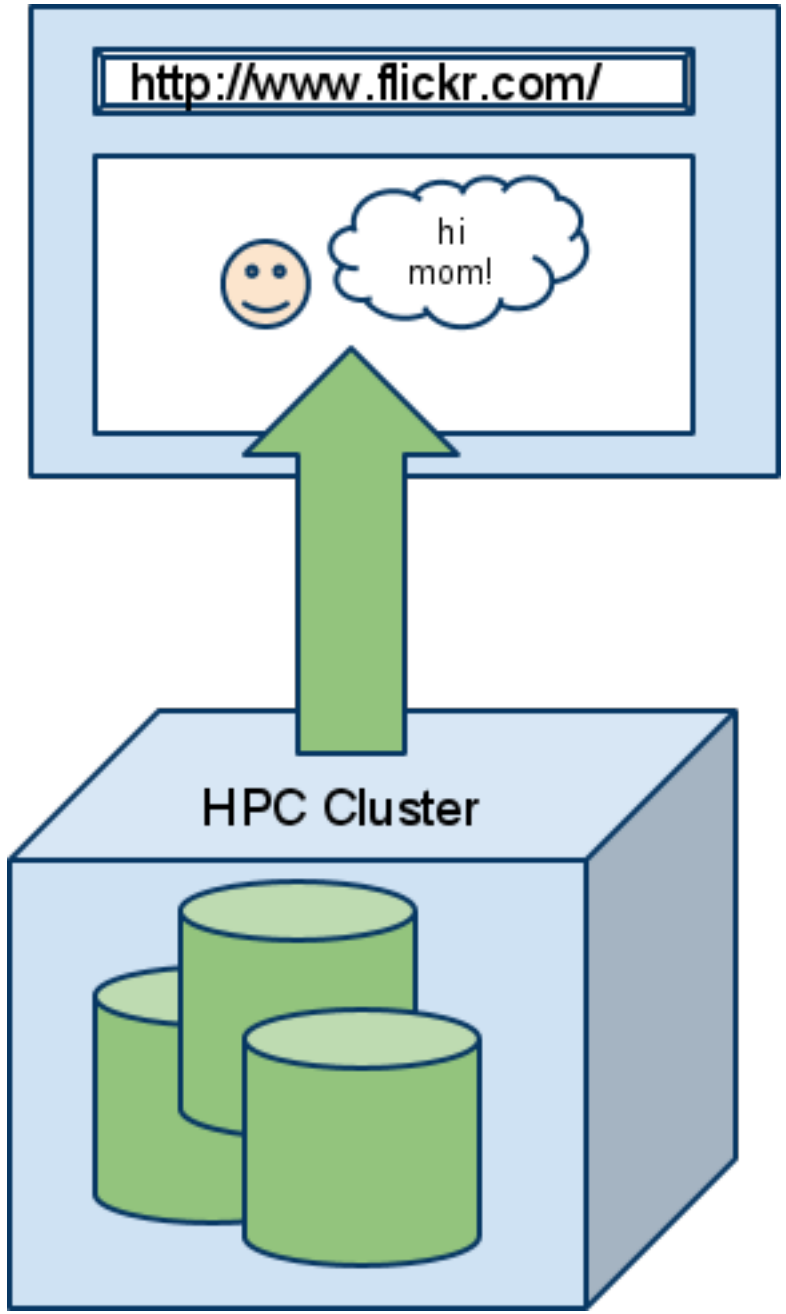
Capability Transfer for Service Collaboration

R. Alexander Milowski
&
Henry Thompson

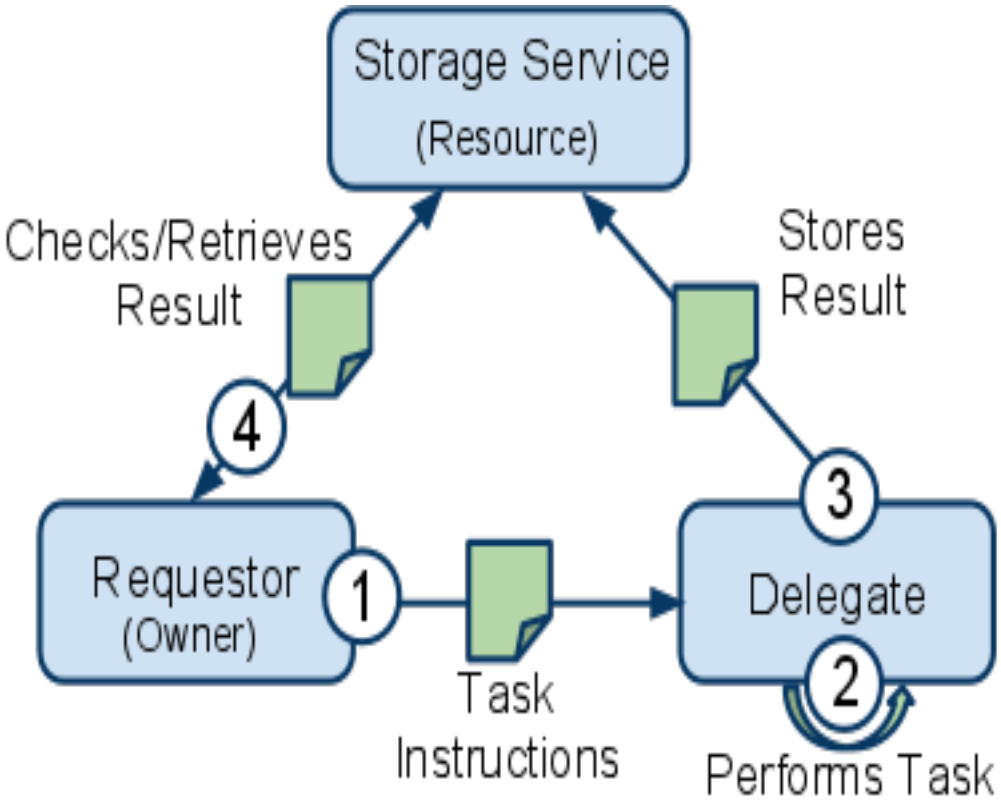
A Motivating Story

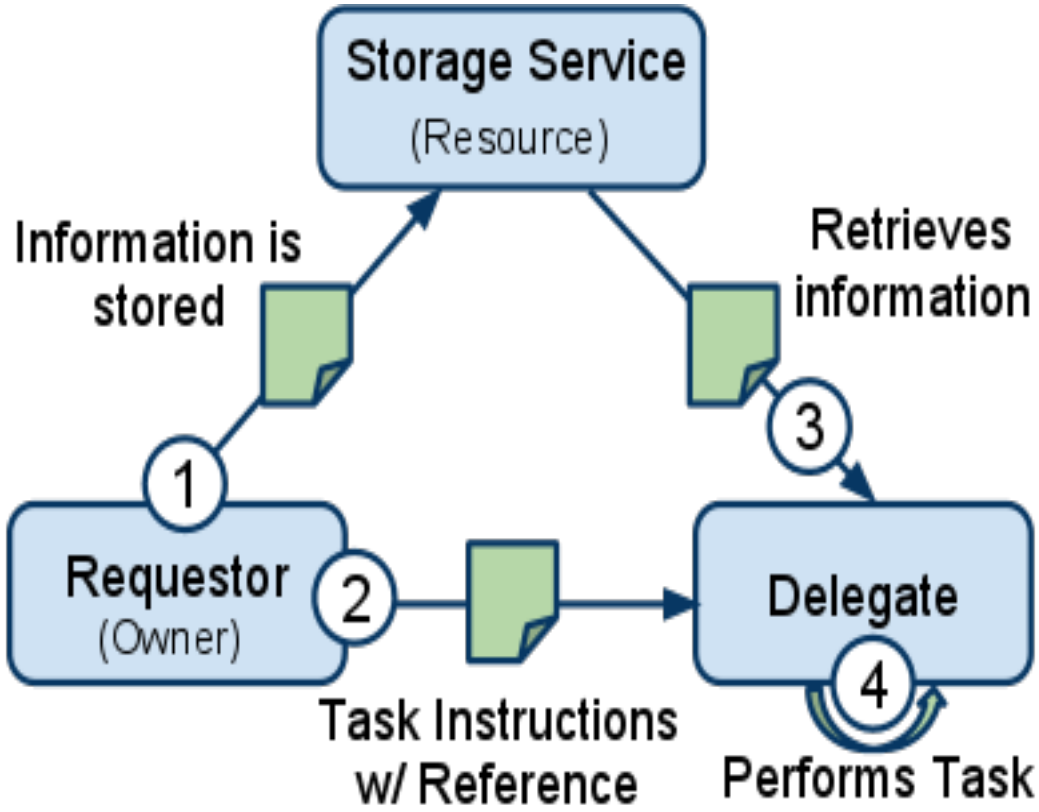
- Gabrielle Allen gave a presentation here at our e-Science Institute about exploiting Web 2.0 for scientific simulations.
- Problem: where can images from ongoing simulations be stored for real-time collaboration by researchers?
- Idea: Use Flickr!
- Issue: we don't want to give the HPC cluster our Flickr credentials!



Basic Uses Cases

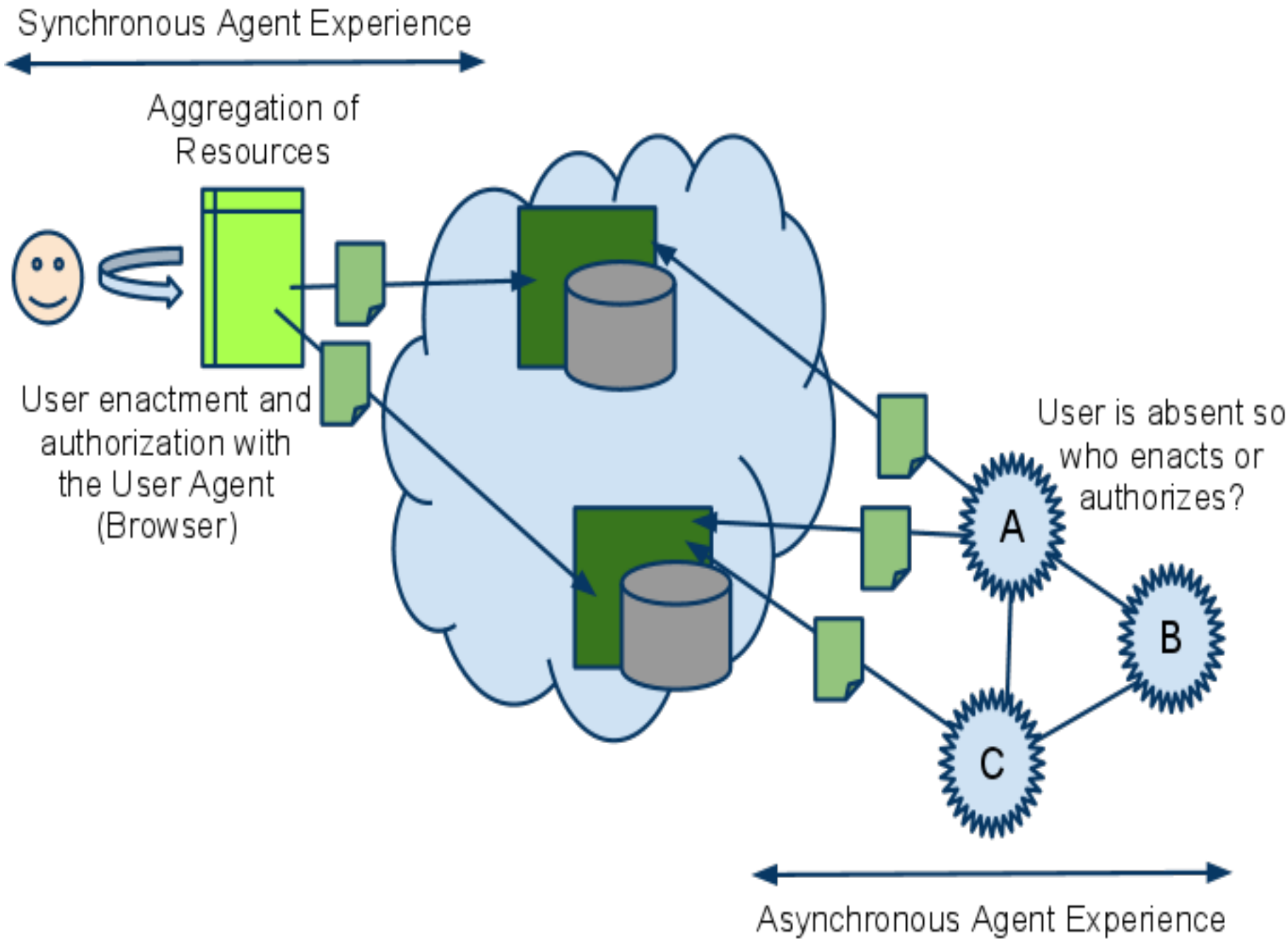
Storage Triangle





Access Triangle

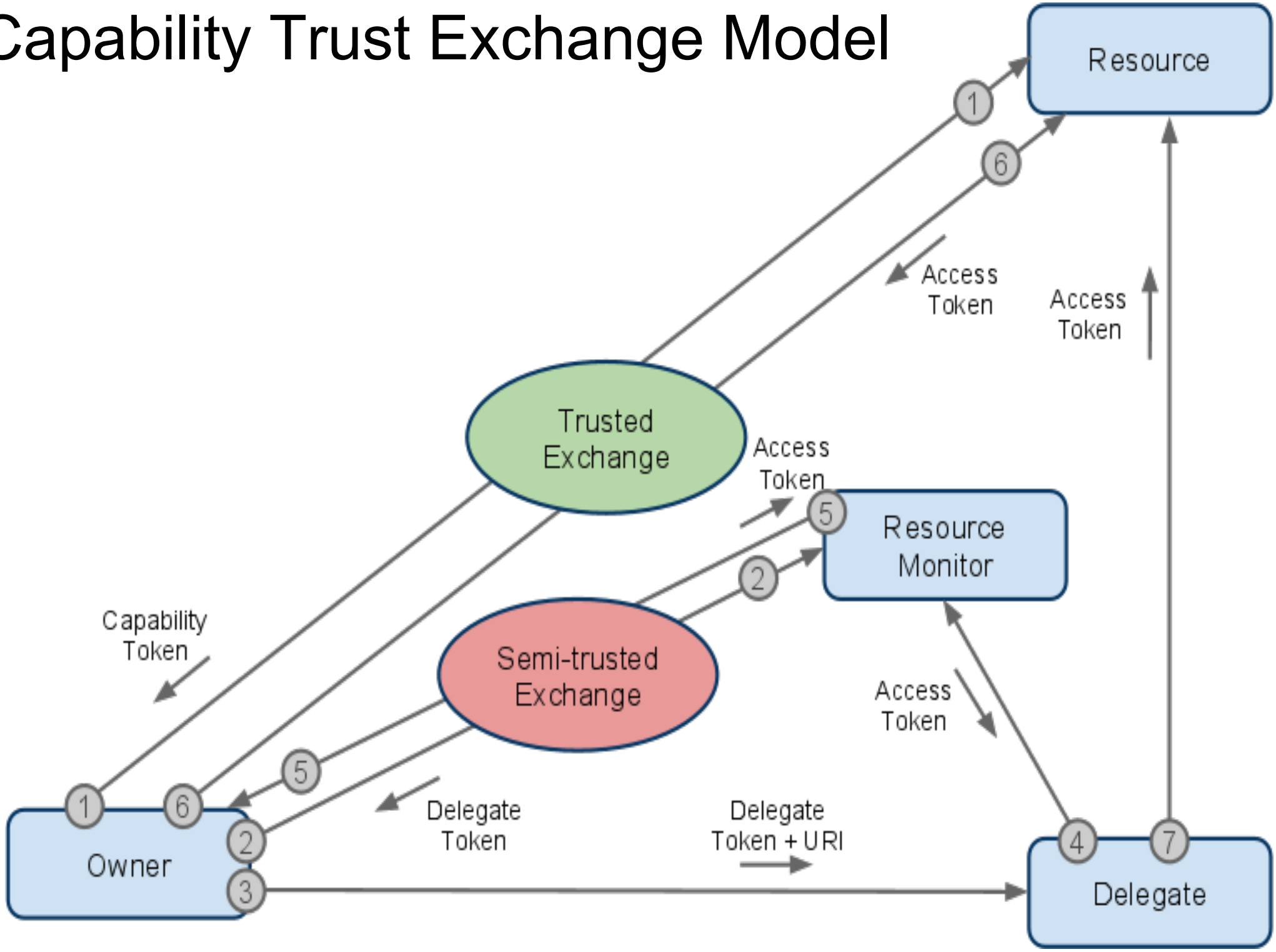
Agents Duality



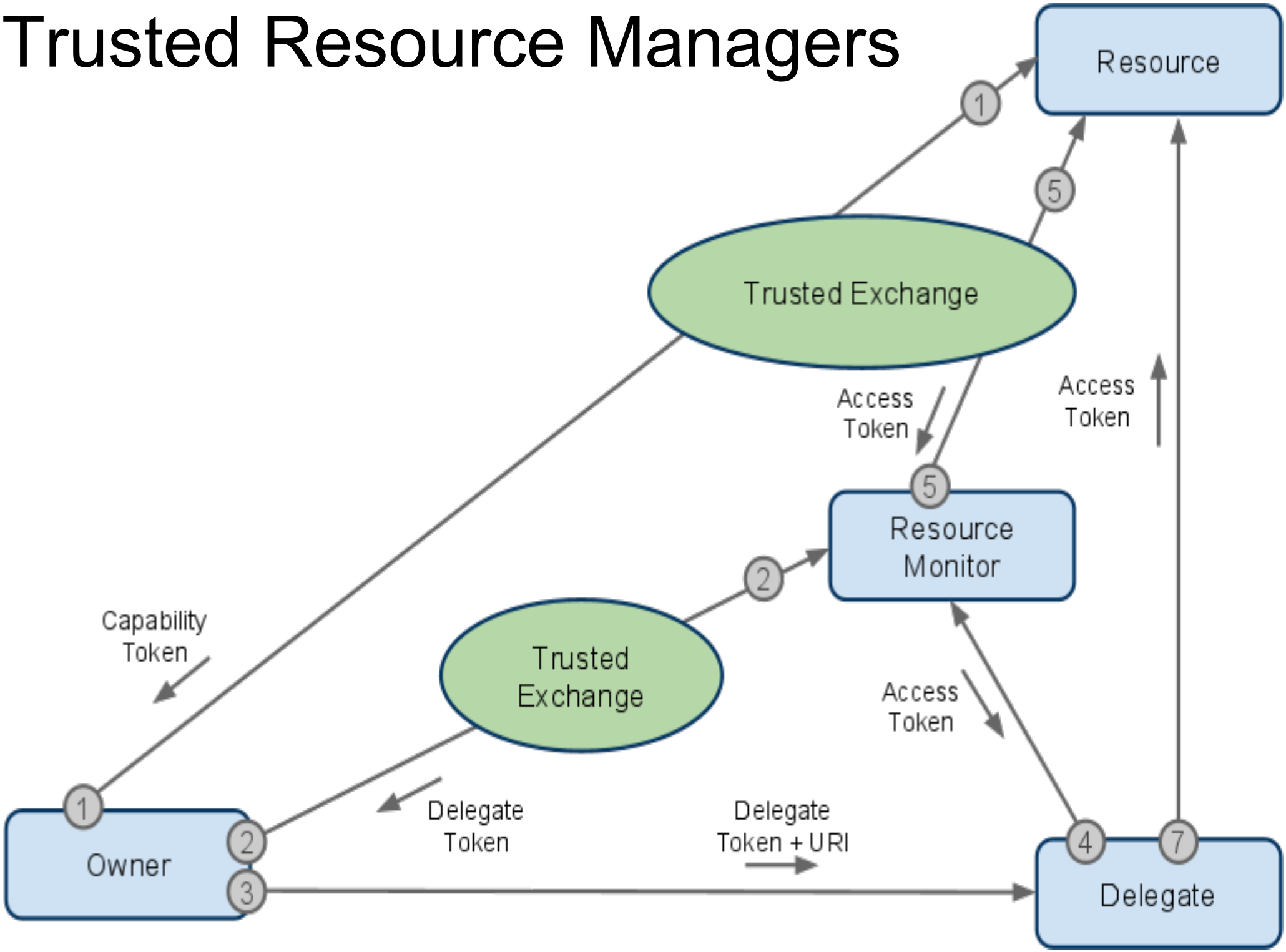
Capability Based Security

- Originates in the 80's when different security models were being developed for "secure" versions of Unix.
- A **Capability** is an encapsulation of:
  - a reference to a group of data,
  - authorization, access rights, and rules for use of the data.
- A capability's use is checked by an actor called the **Reference Monitor** where:
  - the process's access,
  - and operation against data,
  - are checked against the capability granted.

Capability Trust Exchange Model



Trusted Resource Managers

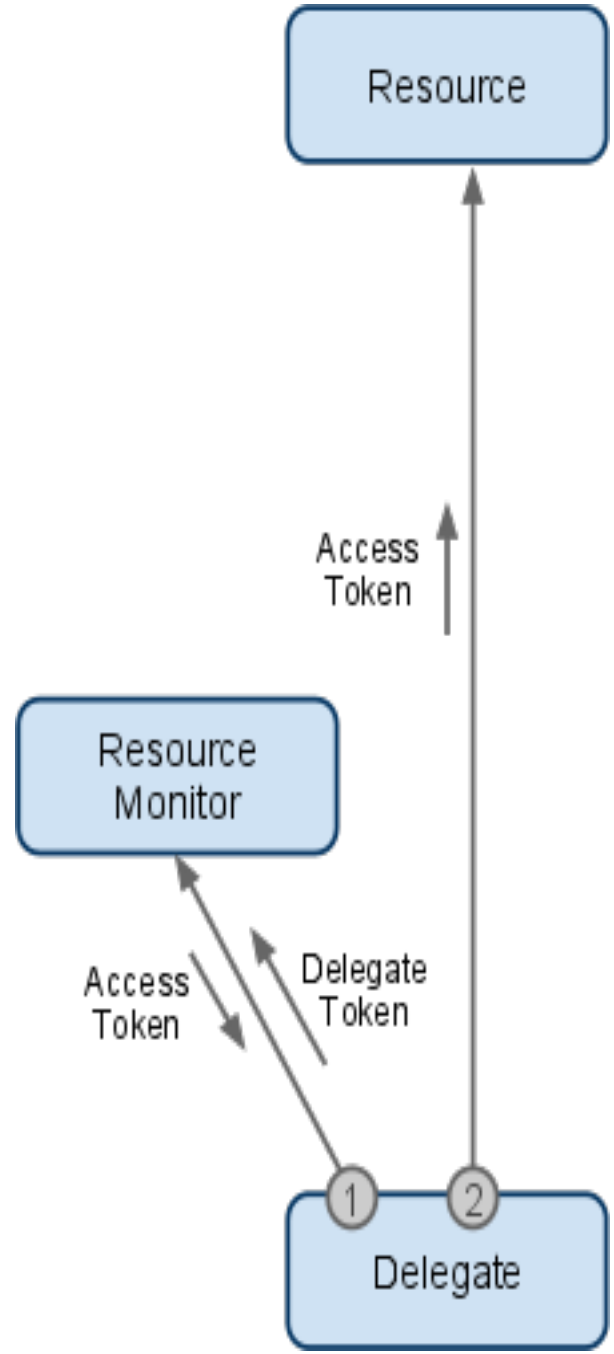


Defining Capabilities

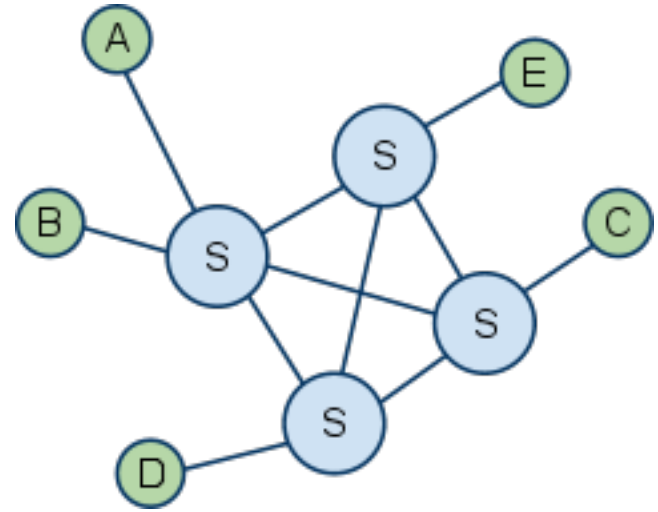
- On the web, a capability is the pair:
  - a target resource set (i.e. a set of URIs),
  - a set of operation constraints.
- Example: "Upload a single picture" constraints
  - content type: image/*
  - size less than 1MB
  - only once
- Example description:
  ({ http://upload.example.com/gallery/12345 },
  { (POST, 1, { starts-with(content-type(),"image/*"),
  size() < 1048576,
  uses() < 1 }) })

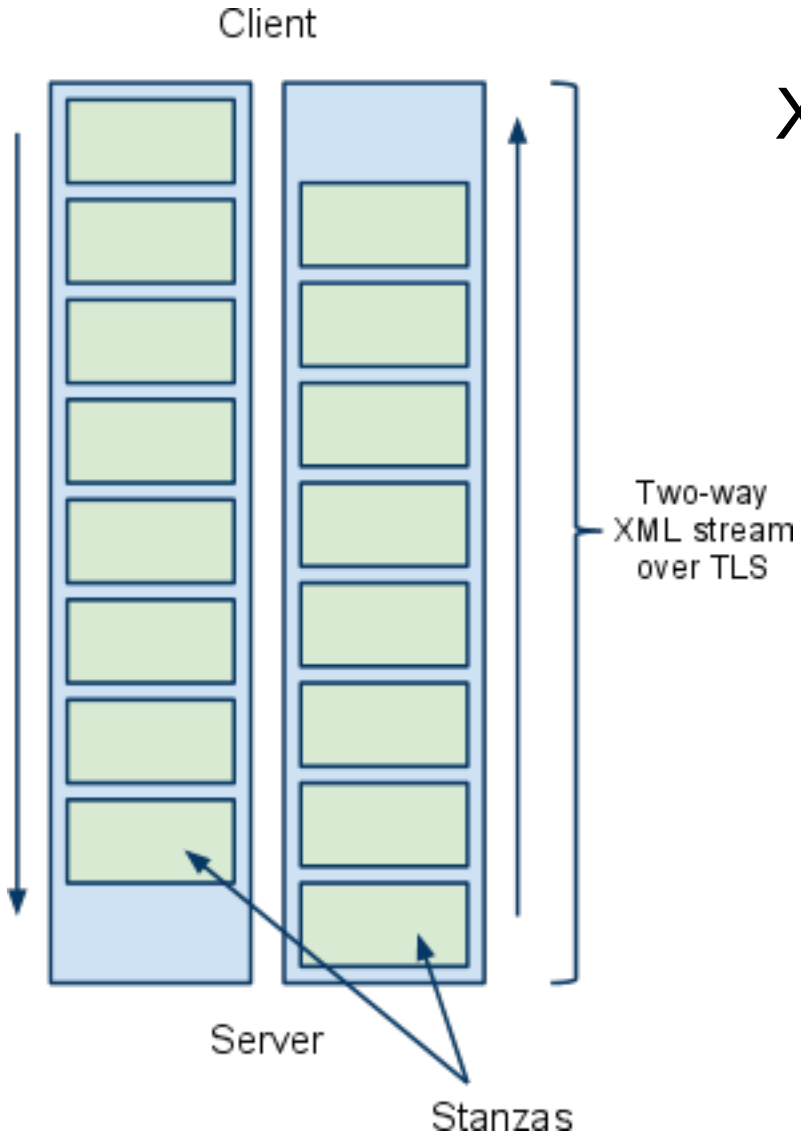
Using OAuth for Delegate Access

- OAuth is IETF RFC 5849 for transferring access to web resources between origins.
- OAuth 2.0 provides new facilities:
  - allows use of assertions,
  - but it is only a draft.
- The Delegate uses an OAuth exchange with the Resource Monitor to get an access token.
- The Delegate uses the access token as it normally would to access the Resource.



XMPP for Exchanges





XMPP is:

- server P2P,
- client/server for the client,
- the client is always connected over an outgoing & usually TLS secure connection (firewall friendly),
- a stream is a sequence of XML elements called "stanzas",
- each kind of stanza (based on element name), represents a particular kind of message (e.g. "message" is used for chat).
- Identity: you@domain/resource

Using MUC for Exchanges

Extensions:

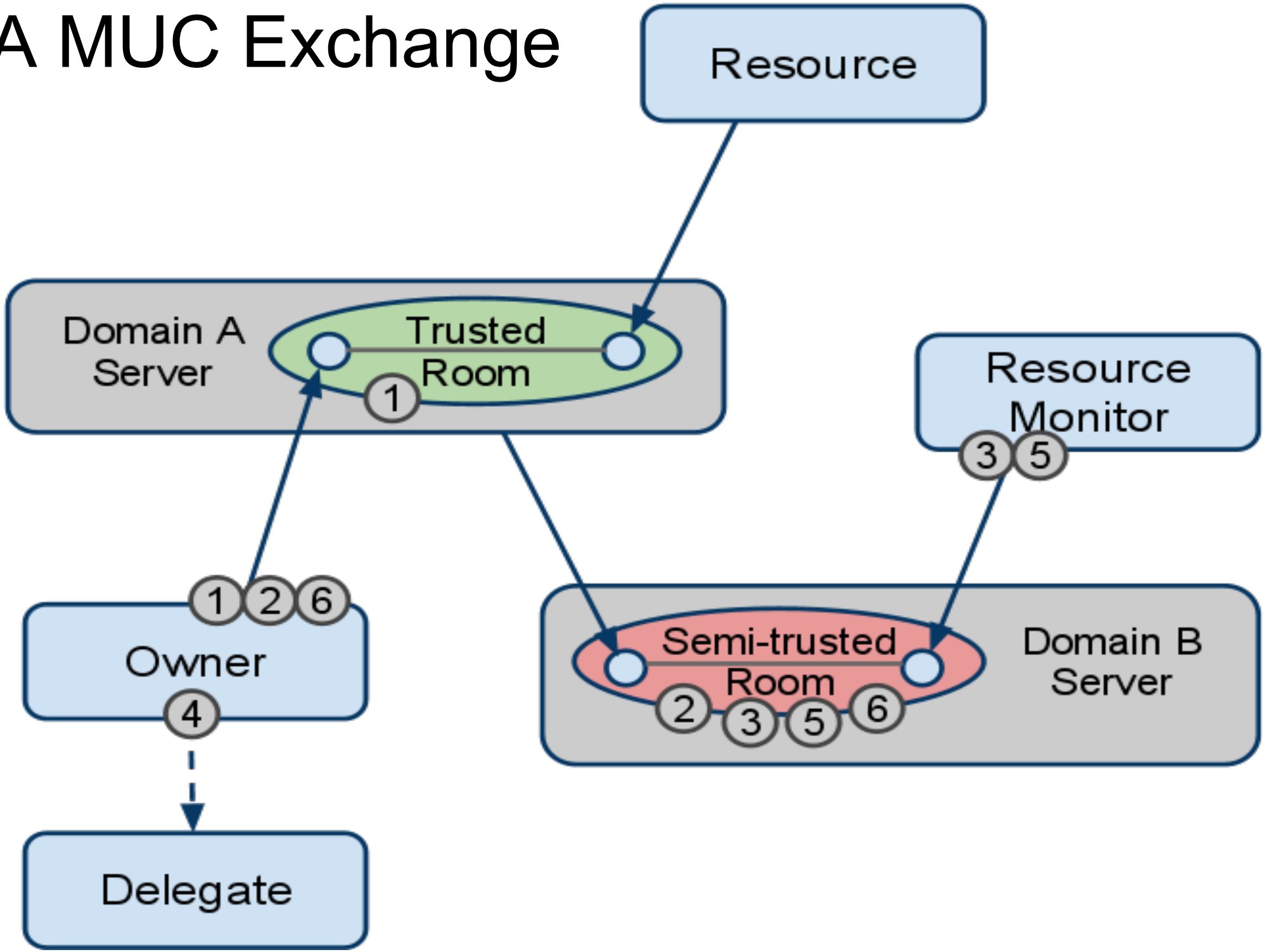
- MUC [1] is just one of about 277 different extensions,
- Extensions add to stanza "semantics",
- See: http://xmpp.org/xmpp-protocols/xmpp-extensions/

MUC:

- Allows for broadcasting of messages to a group of recipients,
- individuals "join rooms" and have in-room identities,
- real identity is protected and rooms route messages,
- joining a room is a negotiated process,
- can be hosted by the server not associated with the client.

[1] http://xmpp.org/extensions/xep-0045.html

A MUC Exchange



Final Thoughts

- Capability exchanges model inter-organization trust.
- This may help organizations share data where:
  - previously they wouldn't have done so,
  - out of concern about control.
- This enables more:
  - automation of services,
  - data sharing,
  - secure collaboration between individuals and agents,
  - who don't share the same infrastructure.
- Check out Amazon S3 "bucket policies" -- this is already happening!
- Need customers! Ideas?