Actors in the ACE Architecture

draft-gerdes-ace-actors-03

Stefanie Gerdes, **Carsten Bormann**
{gerdes | cabo}@tzi.org

IETF-92, ACE Meeting, 2015-03-24

Purpose of the Actors Draft

- What are the tasks that must be performed for authentication and authorization in constrained environments?
- How can these tasks be assigned to actors in the architecture?

Scenario

- RESTful architecture: a client (C) attempts to access a resource (R) which is hosted by a server (S).
- C and/or S are constrained.
- C and S may not know each other, have no trust relationship.
- C and S may not have the same principal (belong to the same person / company).
- How can principals keep the control over their data and devices?

Security Objectives and Authorization

- Integrity (Authorization required).
- Confidentiality (Authorization required).
- Availability (might be breached by misconfigured or wrongly designed authorization solution. Authorization might also help to reduce the burden on system resources).
- Accountability (cannot be achieved with authorization, requires authentication).
- Authorization policies are designed to achieve security objectives.

Lessons Learned from the Use Cases: Security Objectives

- Devices handle sensitive data that needs to be protected.
- Different stakeholders have different security objectives.
- Authorization policies might change any time.

Consequences:

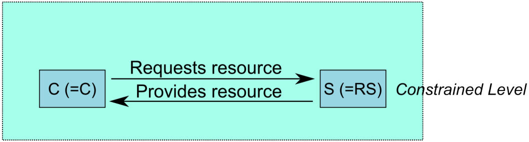
- Authorization policies must be enforced by devices that send or receive sensitive data.
- The authorization policies must be made available to the devices to make them enforceable (in some cases dynamically).

Actors

- Actors are **model**-level
  - defined by their tasks and characteristics
- Several actors **MAY** share a single device.
- Several actors **MAY** be combined in a single piece of software.
  - for a specific application
  - for a specific protocol
- Do not prematurely reduce model to one application/protocol

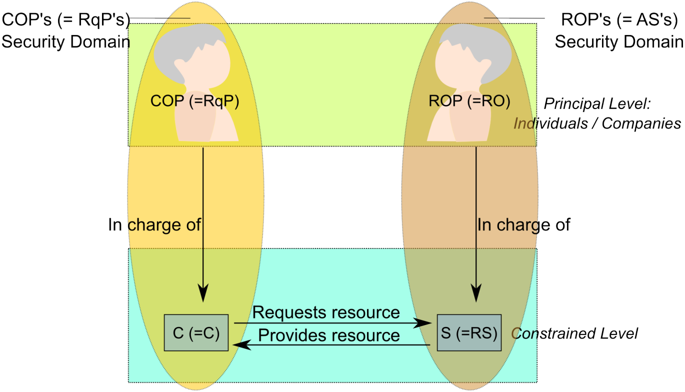
Constrained Level Actors

- C and S are constrained level actors: able to operate on a constrained node.
- Must be able to answer the question: am I supposed to send data to / receive data from this device?
- C and S must perform the following tasks:
  - Validate that an entity is authorized to provide / receive a piece of information.
  - Validate that received messages are authentic.
  - Securely transmit messages.x
- To securely participate in a conversation, an endpoint must at least be able to perform the constrained level tasks.



Principal Level Actors

- C and S are under control of principals in the physical world.
- Client Overseeing Principal (COP) is in charge of C: Configures authorization policies, e.g. with whom C is allowed to communicate.
- Resource Overseeing Principal (ROP) is in charge of S: Configures authorization policies.



Lessons Learned from the Use Cases: Absent Users

- Often no active user at the time of access.
- Authorization policies cannot always be configured manually for each device.
- Devices often have no user interfaces and displays.

Consequences:

- Principals will not intervene in the communication (e.g., not control the client).
- Principals cannot make authorization decisions at the time of access (e.g., no authorization via pop-ups).
- Devices must be able to enforce authorization policies on their own.

Benefits of Offloading Tasks

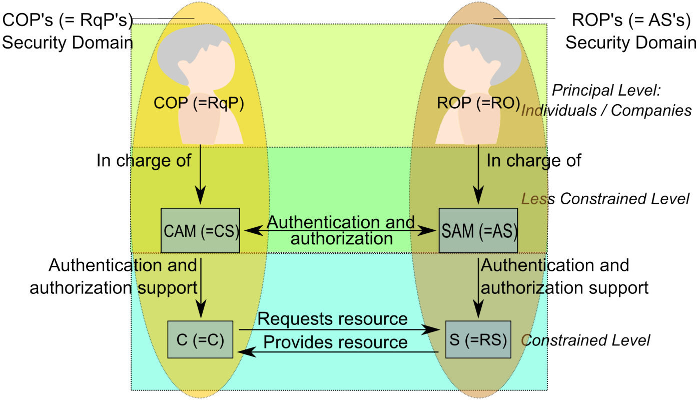
- There might not be an active user at the time of access.
- Devices often don't have user interfaces and displays and thus cannot be controlled by the user at the time of access.
- One or both of C and S are "constrained"
  - in terms of power, memory, storage space.
  - can only fulfill a limited number of tasks.
  - may not have network connectivity all the time.
  - may not be able to manage complex authorization policies.
  - may not be able to manage a large number of keys.
- Address this by associating a *less-constrained device* to each constrained device for one or more of those difficult tasks -> Devices still have to enforce the principal's policies on their own.

Less-Constrained Level

- The Client Authorization Manager (CAM) is aiding C in authenticating S and determining if S is an authorized source for R.
- The Server Authorization Manager (SAM) is aiding S in authenticating C and determining C's permissions on R.
- CAM and SAM act on behalf of their respective principal.
- CAM and SAM provide a user interface for their principal.

Less-Constrained Level (2)

- Without CAM, C's principal will not be able to keep the control over C.
- Without SAM, S' principal will not be able to keep the control over S.

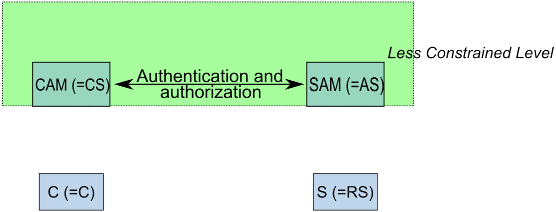

COP's (= RqP's) Security Domain
ROP's (= AS's) Security Domain
COP (=RqP)
ROP (=RO)
Principal Level: Individuals / Companies
In charge of
In charge of
Less Constrained Level
CAM (=CS)
Authentication and authorization
SAM (=AS)
Authentication and authorization support
Authentication and authorization support
C (=C)
Requests resource
Provides resource
S (=RS)
Constrained Level

Tasks

- Tasks that each device must perform (constrained level tasks).
- Tasks that can be outsourced (less constrained level tasks).
- Combination of constrained level tasks and less-constrained level tasks possible.
- If a device can only perform the minimum tasks, it can still securely participate in the communication.

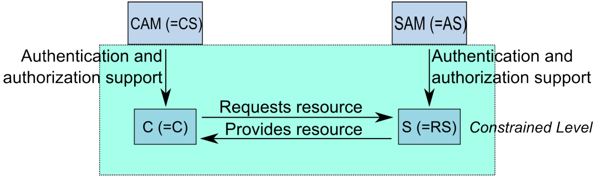
Less-Constrained Level Communication

- No limitations for the use of existing protocols (HTTP, TLS, OAuth,..)



Constrained and Cross Level Communication

- Communication protocol between constrained level actors.
- Support protocol between constrained level actors and less-constrained level actors.
- Protocols must consider the limitations of their constrained endpoints.



Lessons Learned from the Use Cases: Constrained vs Less-Constrained

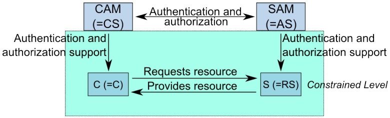
- Limitations of the communicating devices may vary.
- Constrained device to less-constrained device useful.
- Constrained to constrained communication allows for additional benefits (e.g., direct communication between the sensor and the cooling unit in the container monitoring use case enables more efficient cooling).
- Devices might have only some constraints (e.g., no user interface).

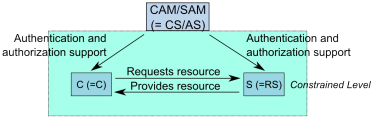
Consequences:

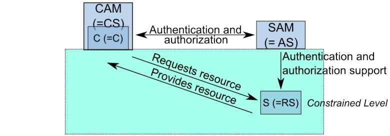
- Constrained devices communicate among themselves as well as with less-constrained devices.

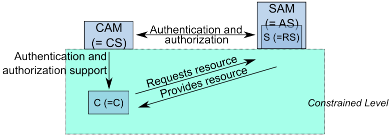
Constrained Level Communication: Variants

- Protocols must consider the limitations of their constrained endpoints.
- Communication protocols are still constrained level protocols.


CAM (=CS)
Authentication and authorization
SAM (=AS)
Authentication and authorization support
Authentication and authorization support
C (=C)
Requests resource
Provides resource
S (=RS)
Constrained Level







Questions the Actors Draft deals with

- How do we handle authorization without an active user?
- How do we cope with the lack of displays and user interfaces?
- How do we cope with dynamic changes in a setting (e.g., outage of the communication partner (server or client), need for a replacement)?
- How do we consider the different security objectives of the principals on both sides?
- How do we combine the constrained world with the less-constrained world?
- How do we manage the different possible client/server settings?

How to proceed?

- Make this a WG document.