ACE Architecture: Actors

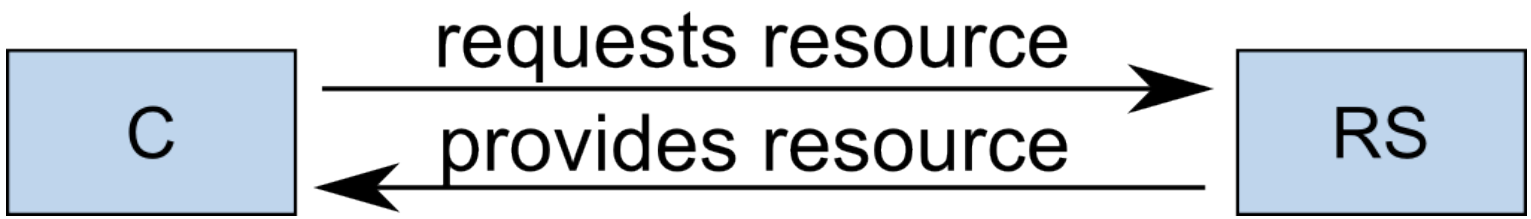
draft-gerdes-ace-actors-02

Stefanie Gerdes, **Carsten Bormann**

IETF-91, ACE Meeting, 2014-11-12

Problem Statement

- A Client (C) wants to access an item of interest, a resource (R) on a Resource Server (RS).
- A priori, C and RS do not know each other, have no trust relationship. They might belong to different owners.
- C and / or RS are located on a constrained node.



Constraints

- "constrained" is defined in RFC 7228
  - i.e., Class-1 ($\approx$ 10/100 KiB) or Class-2 ($\approx$ 50/250 KiB)
- One or both of C and RS are "constrained"
  - in terms of power, memory, storage space.
  - may not have user interfaces and displays.
  - 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.
- Owner may not be present at the time of access (cf. draft-seitz-ace-usecases).

Possible Scenarios

Constrained or not constrained:

1. C is constrained and RS is less constrained
2. RS is constrained and C is less constrained
3. C and RS are constrained

Ownership:

1. C and RS belong to the same owner
2. C and RS belong to different owners

Basic Security Requirements

- Confidentiality and integrity of R: No unauthorized device must be able to access (or otherwise gain knowledge of) R.
  - RS needs to know if C is allowed to access R
  - RS needs to make sure that it provides the resource only to C.
  - Access requests and the corresponding answers can both contain resource values and must be protected accordingly.
- Authenticity of R: C must access the proper R.
- C needs to know if R as offered by RS is the resource it wants to access.

New section in Version 02: Authenticated Authorization

- Determine if the owner of an item of interest allows an entity to access this item as requested.
- Authentication: Verify that an entity has certain attributes (cf. RFC4949).
- Authorization: Grant permission to an entity to access an item of interest.
- Authenticated Authorization: Use the verified attributes to determine if an entity is authorized.

New section in Version 02: Authorization and Security Objectives

- Confidentiality (Authorization required)
- Integrity (Authorization required)
- Availability (can be breached by misbuilt Authorization Solution)
- Accountability (cannot be achieved by Authorization. Requires Authentication)

New terms in Version 02: Authorization Level of Granularity

- Device Authorization: Authorization is granted based on the unique identity of a device.
- Owner Authorization: Authorization is granted based on the ownership of a device.
- Conditional Authorization: Authorization is granted because of contextual factors such as time or location.
- Binary Authorization: All authenticated entities have the same authorization.
- Unrestricted Authorization: All entities can access everything on a device as they see fit.

Machine to Machine

- In many common authorization solutions the user initiates the access.
  - User controls the client at the time of access and thus does the authorization for the client.
  - No need for further authorization on the client side.
  - Authorization is only needed on the server side.
- In M2M scenarios, C might need to decide on its own which RS it needs to contact and whether RS is authorized to provide representations of R for C.

M2M Authorization with Perimeter Security

- The constrained devices are not able to do authentication and authorization on their own.
- “Guardian” device to protect the constrained devices from incoming traffic.
- Protection against local attackers?
- Stuxnet?

Tasks for Authenticated Authorization

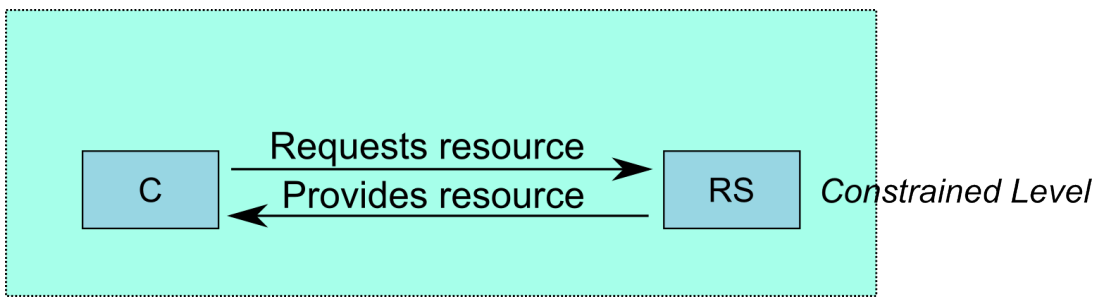
- Constrained devices must be able to limit their tasks
- Some tasks must be performed on constrained devices for security
- Beforehand: Provide information for Authenticated Authorization
  - Make attribute-verifier-binding verifiable: Validate that an entity actually has the attributes it claims to have (e.g. that it belongs to a certain user) and bind the attributes to a verifier (e.g. a key) using the endorsement info.
  - Define access policies (entity with attribute x has this set of permissions).
- At the time of the request: Check access request against the provided information
  - Check the verifier a received access request is bound to.
  - Check the verifier-attribute binding.
  - Determine the authorization using the attributes.
  - Enforce the authorization.

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

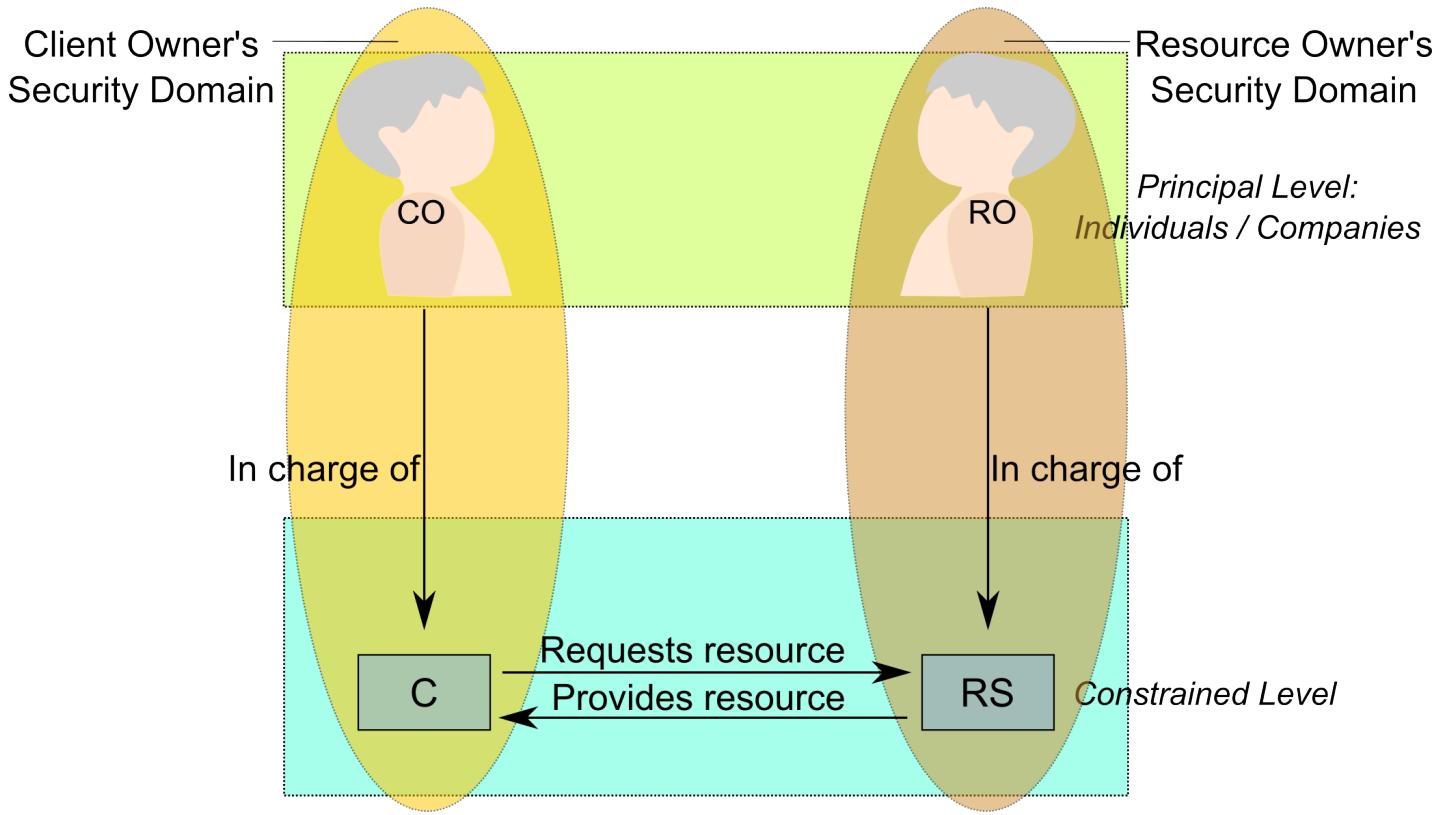
Constrained Level Actors

- C and RS are constrained level actors: able to operate on a constrained node.
- C and RS must perform the following tasks:
  - Validate possession of attributes and authenticate
  - Validate and enforce authorization
  - Securely transmit messages



Principal Level Actors

- C and RS are under control of principals in the physical world.
- CO is in charge of C: Configures security policies, e.g. with whom RS is allowed to communicate.
- RO is in charge of RS: Configures security policies, e.g. authorization policies.



Constraints

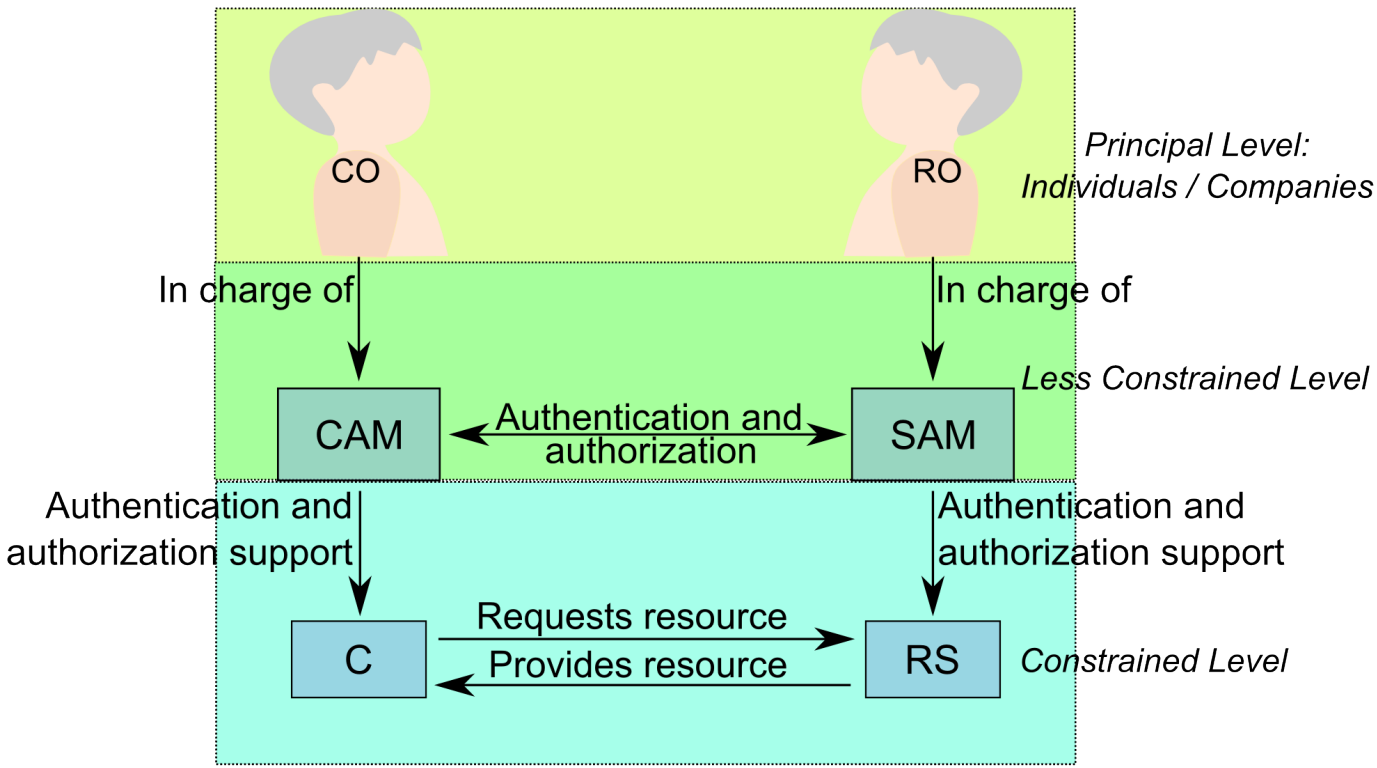
- "constrained" is defined in RFC 7228
  - i.e., Class-1 ($\approx$ 10/100 KiB) or Class-2 ($\approx$ 50/250 KiB)
- One or both of C and RS are "constrained"
  - in terms of power, memory, storage space.
  - may not have user interfaces and displays.
  - can only fulfill a limited number of tasks.
  - 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.
- Owner may not be present at the time of access (cf. draft-seitz-ace-usecases).
- Address this by associating a *less-constrained device* to each constrained device for one or more of those difficult tasks
  -> Devices have to enforce the owner's policies on their own.

Less-Constrained Level

- New Terminology: CAM and SAM instead of AM and AS: Avoid mixup with authorization servers with different functions.
- CAM is aiding C in authenticating RS and determining if RS is an authorized source for R.
- SAM is aiding RS in authenticating C and determining C's permissions on R.
- CAM and SAM act on behalf of their respective owner.
- CAM and SAM provide a user interface for their owners.

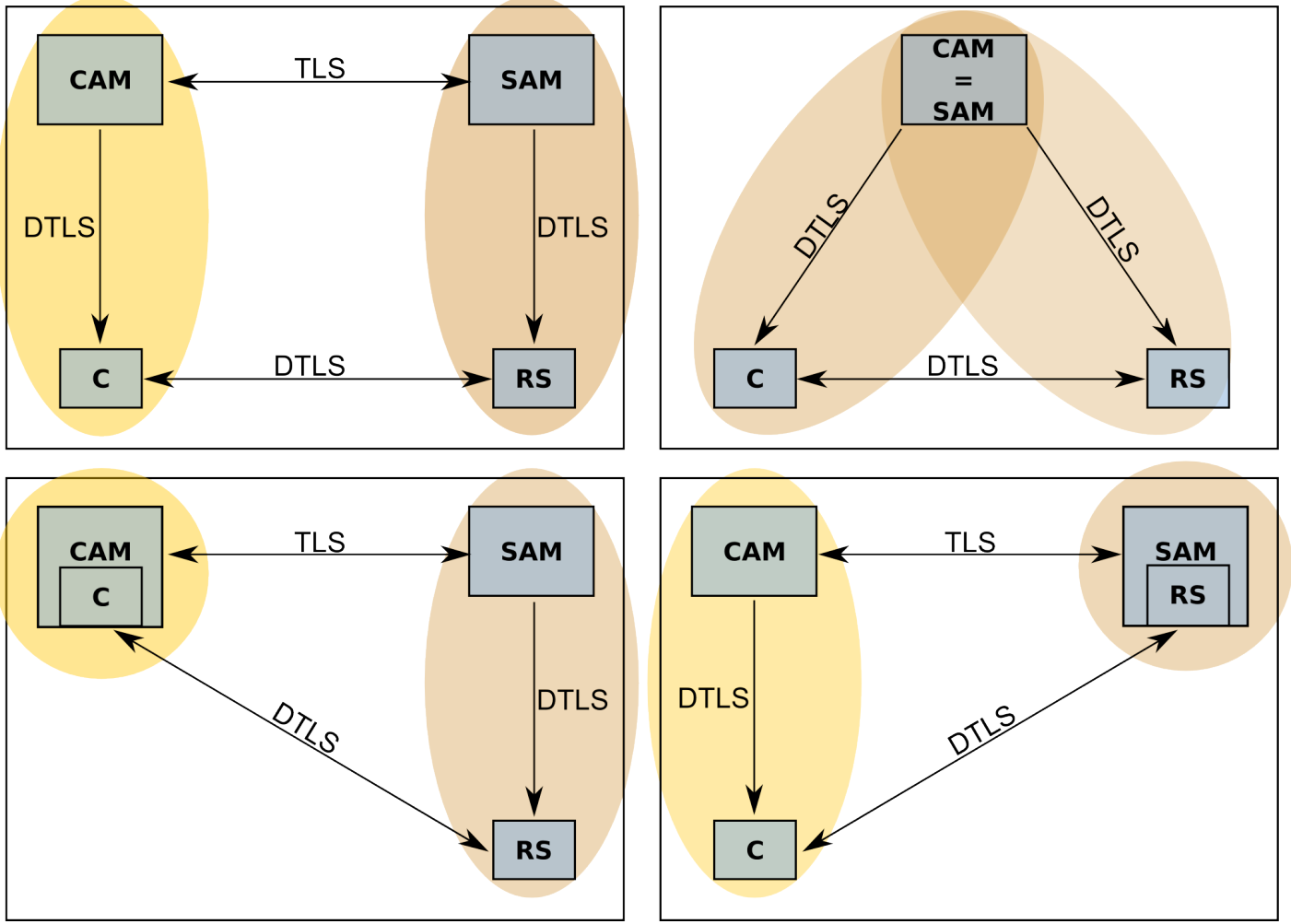
Less-Constrained Level (2)

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



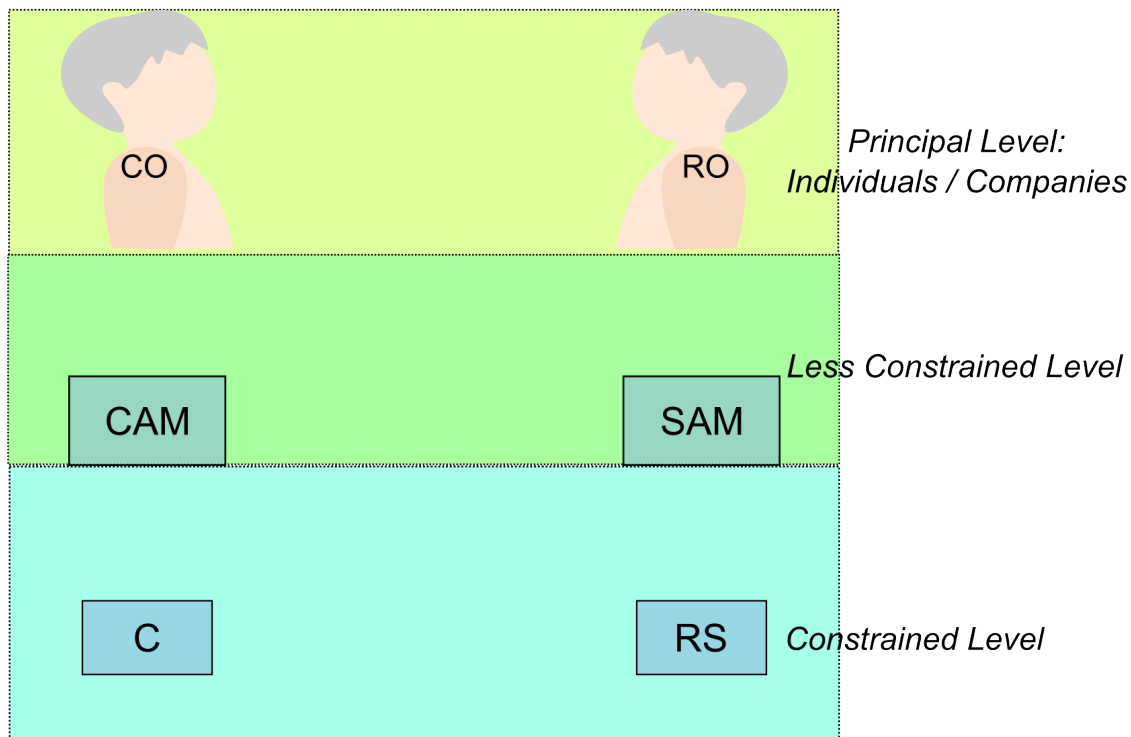
Actors vs. Entities (Devices / Software)

- Several actors may share a single device.
- Several actors may be combined in a single piece of software.



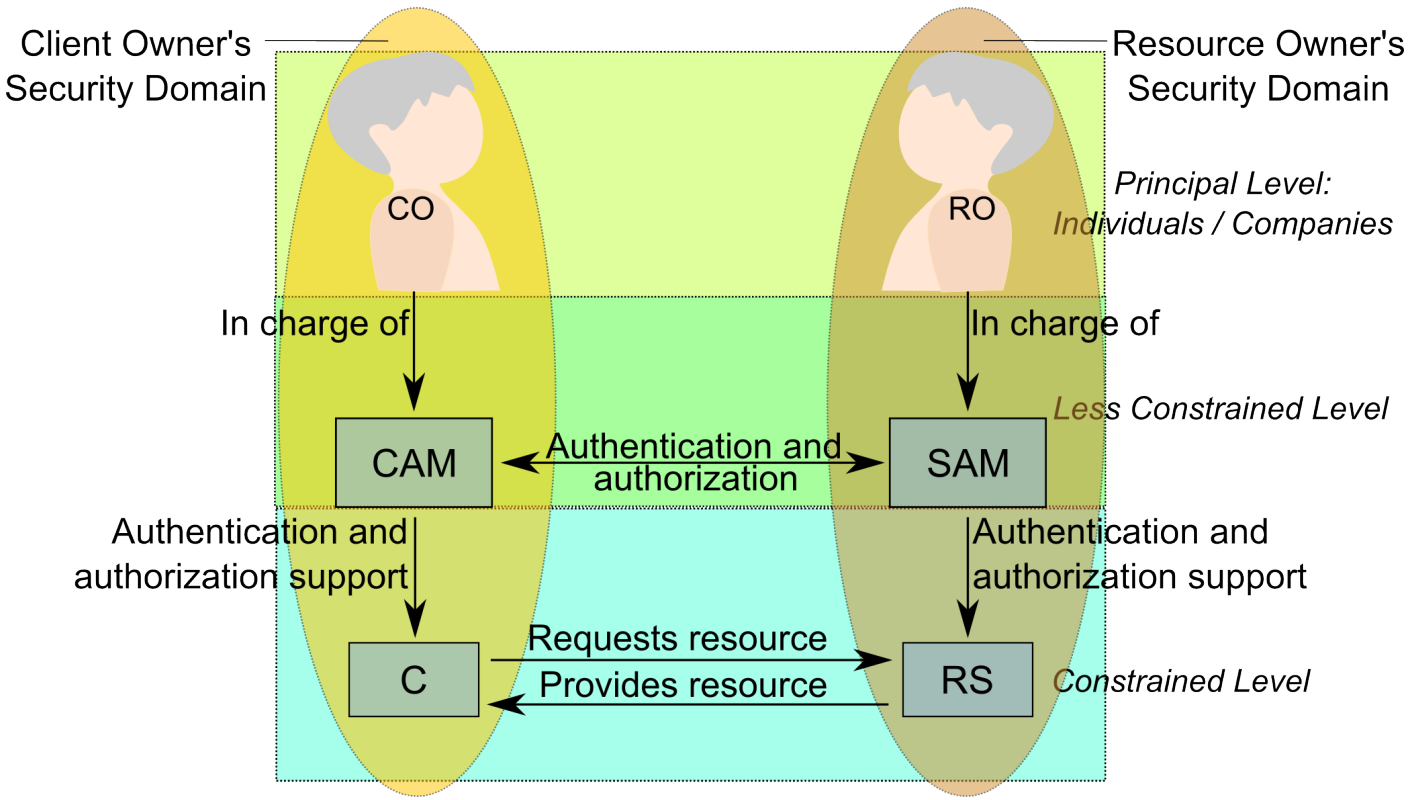
Levels

- Three Levels of Competence: Constrained Level, Less-Constrained Level, Principal Level.
- Different Requirements on each level.
- Principal Level out of Scope in ACE.



Security Domains

- A priori, C and RS do not know each other, may belong to different security domains
- Owners want to keep control over their data.
- Representable with our model.



Next steps

- Focus here: Analysis of Security Relationships in communications with constrained devices
- Align terminology between different drafts
- Decide on final deliverable resulting from this work

Thank you!

Further Reading:
`http://tools.ietf.org/pdf/draft-gerdes-ace-actors-02.pdf`

Send your feedback to: Stefanie Gerdes (gerdes@tzi.org)