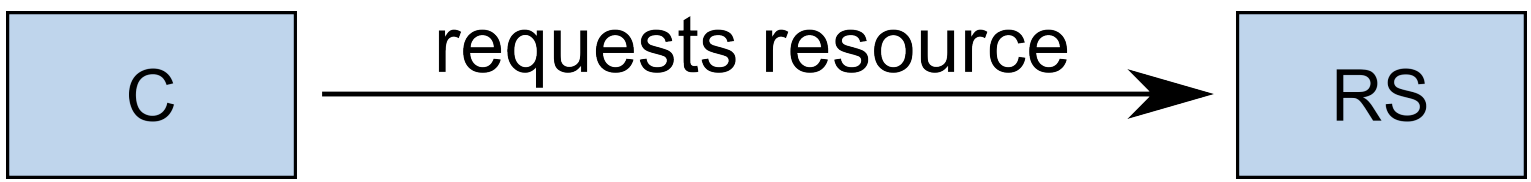
ACE Architecture: Actors

Stefanie Gerdes, **Carsten Bormann**

IETF-90, ACE Meeting, 2014-07-23

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 security domains.
- 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.
- address this by associating a *less-constrained device* to each constrained device for one or more of those difficult tasks

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.

Tasks

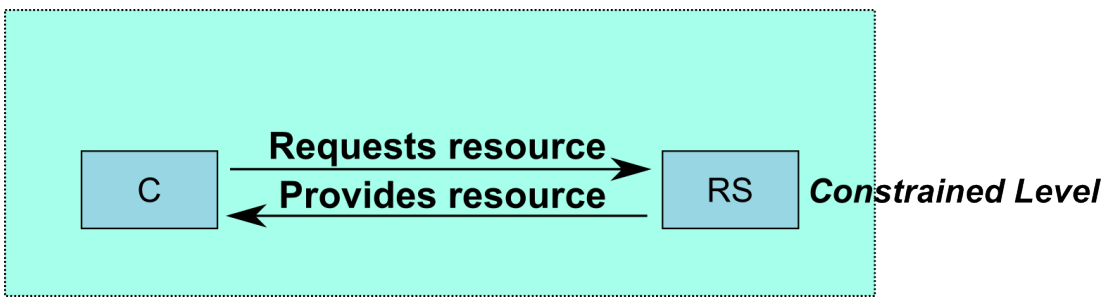
- Constrained devices must be able to limit their tasks
- Some tasks must be performed on constrained devices for security
- Authentication-Related Tasks:
  1. Attribute-Binding: Validate that the entity in possession of a certain verifier (a key) really has certain attributes and make that verifiable by adding endorsement information.
  2. Verifier Validation: Check the endorsement information.
  3. Authentication: The verifier is used for authentication.
- Authorization-Related Tasks:
  4. Configuration of authorization information.
  5. Obtaining the authorization information.
  6. Authorization Validation: map the attributes which are validated by authentication to the authorization information
  7. Authorization Enforcement: Act according to the result of the authorization validation, e.g. grant access to a resource.

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

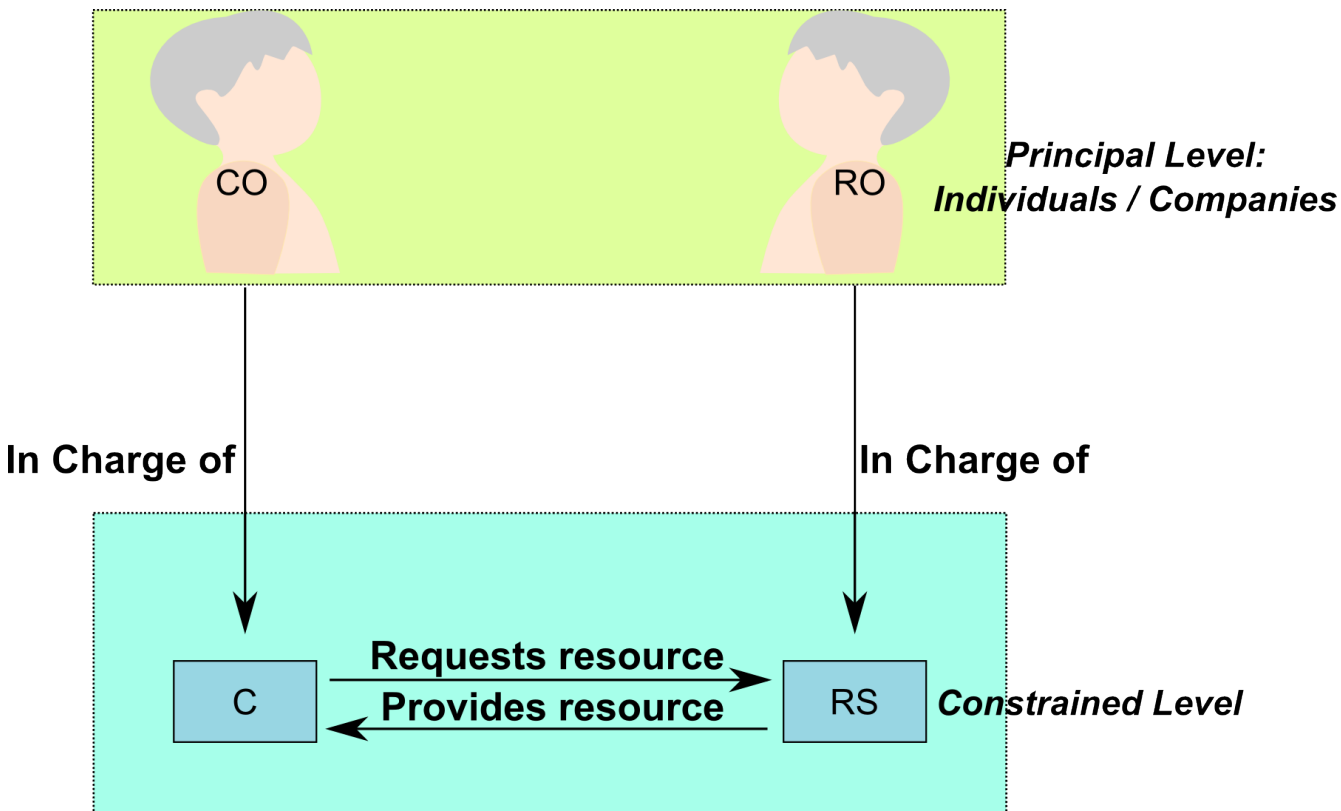
Constrained Level Actors

- 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



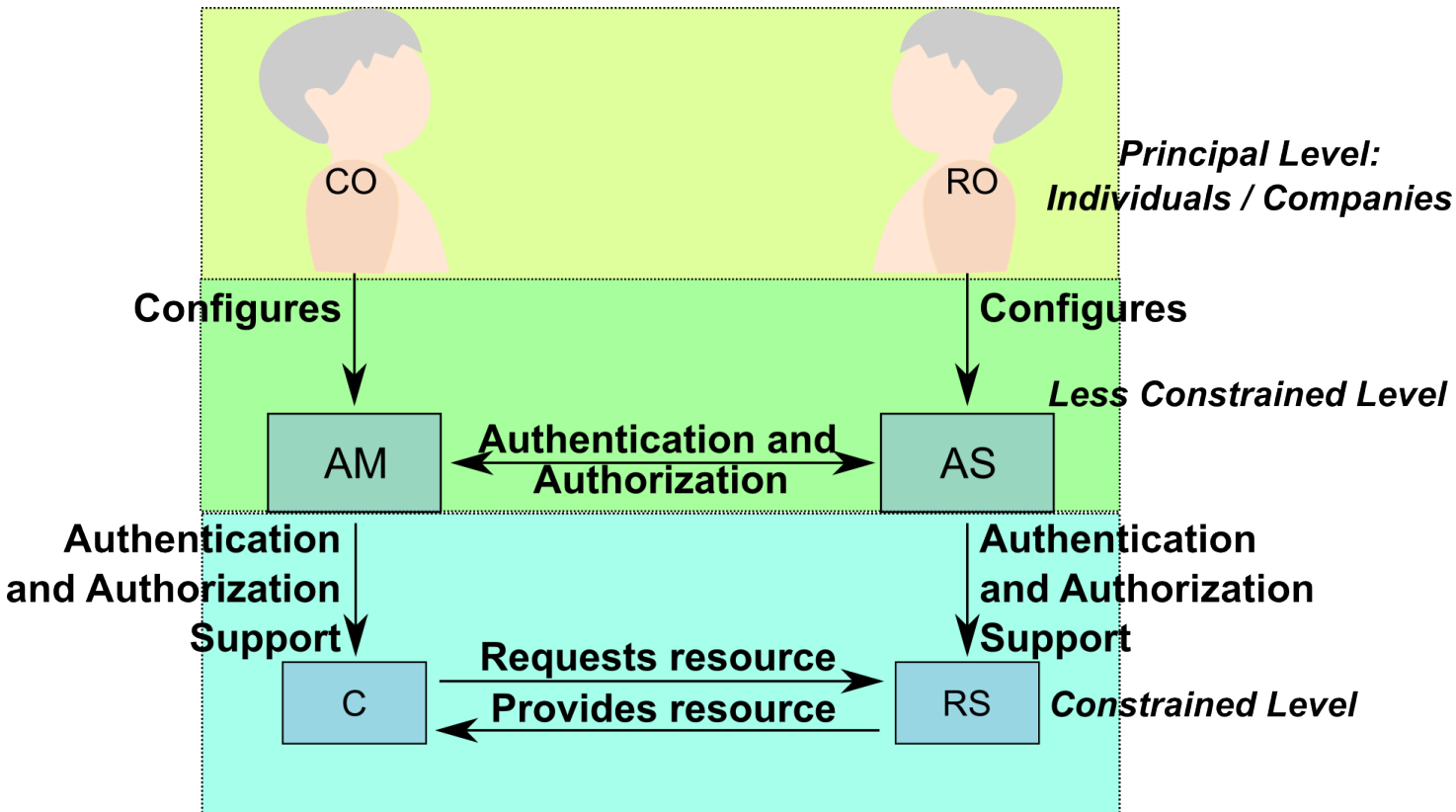
Principal Level Actors

- 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.



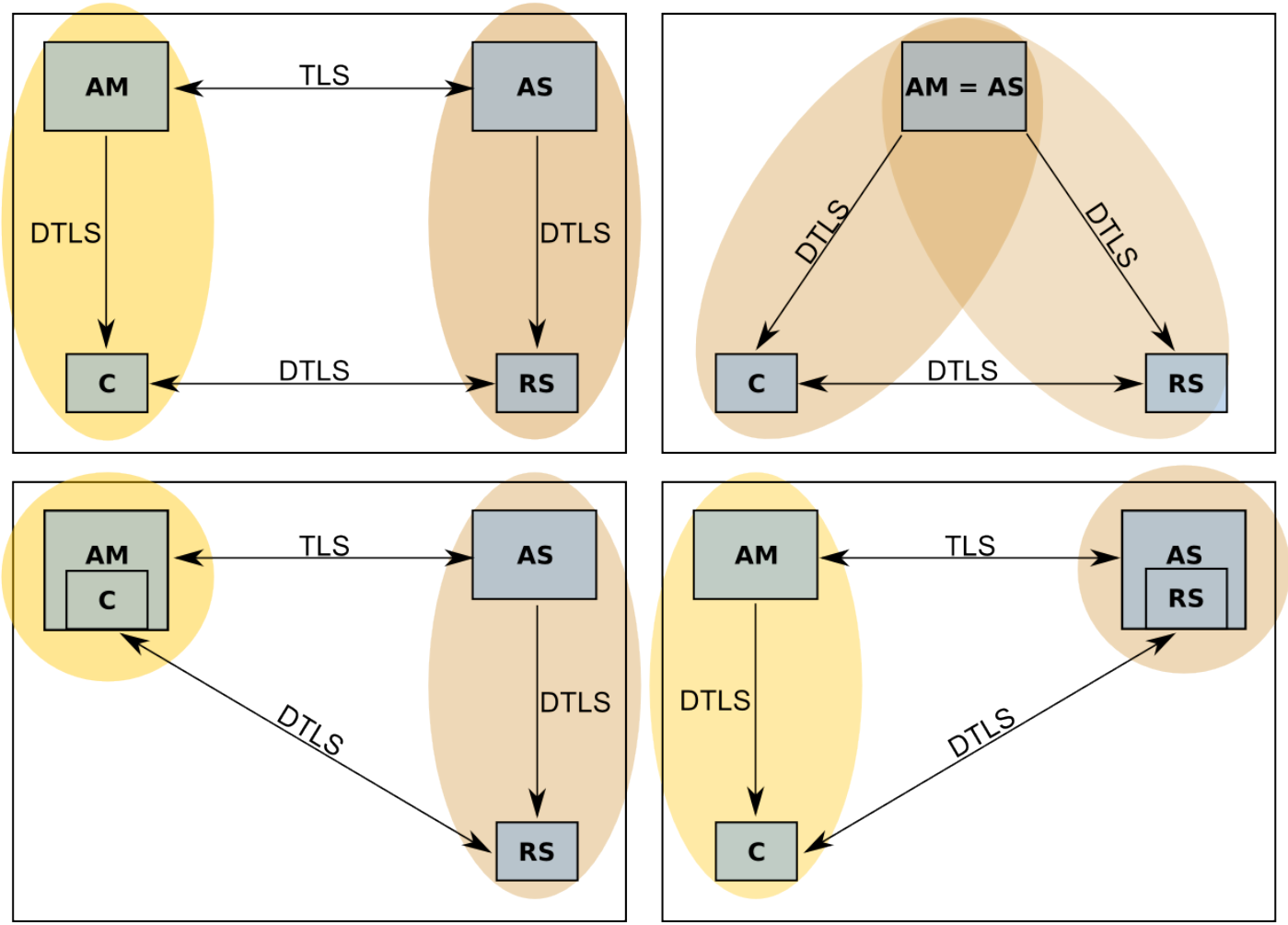
Less-Constrained Level

- AM is aiding C in authenticating RS and determining if RS is an authorized source for R.
- AS is aiding RS in authenticating C and determining C's permissions on R.
- AM and AS act on behalf of their respective owner.



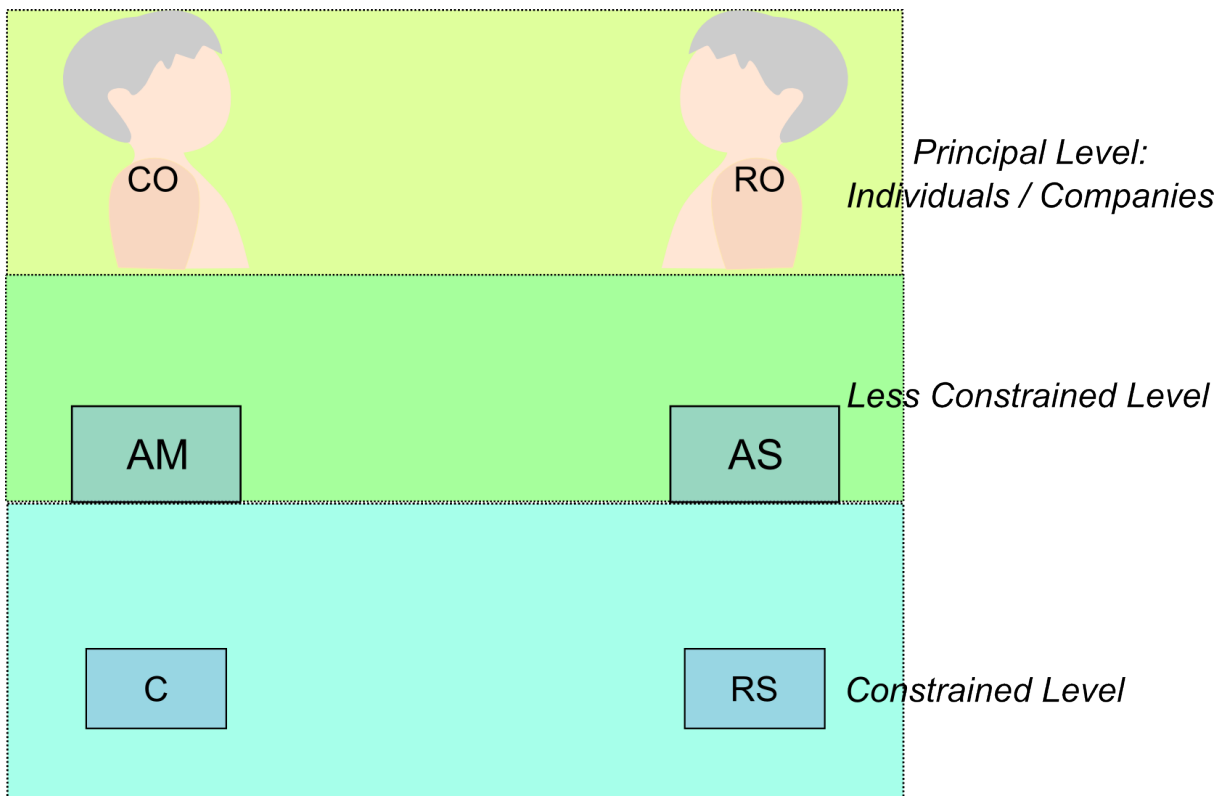
Actors vs. Entities (Devices / Software)

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



Levels

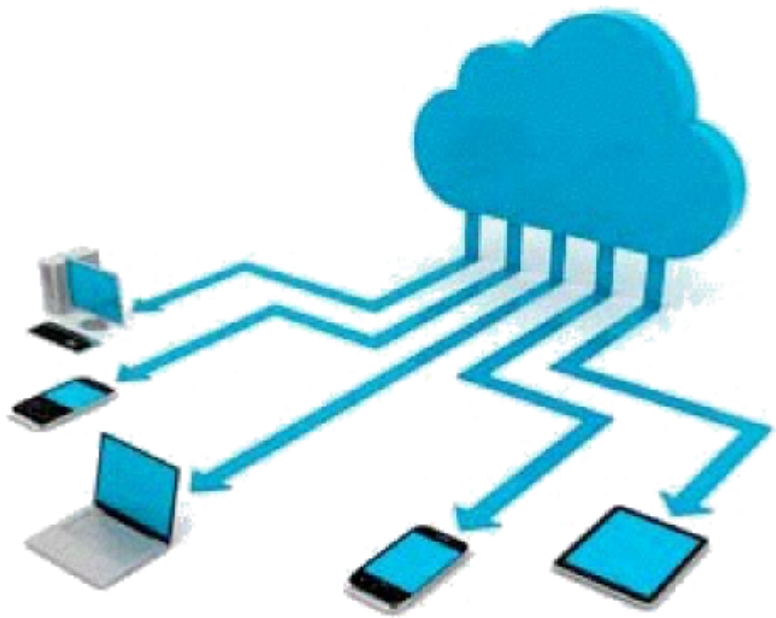
- Three Levels of Competence: Constrained Level, Less-Constrained Level, Principal Level.
- Different Requirements on each level.
- Principal Level out of Scope in ACE.



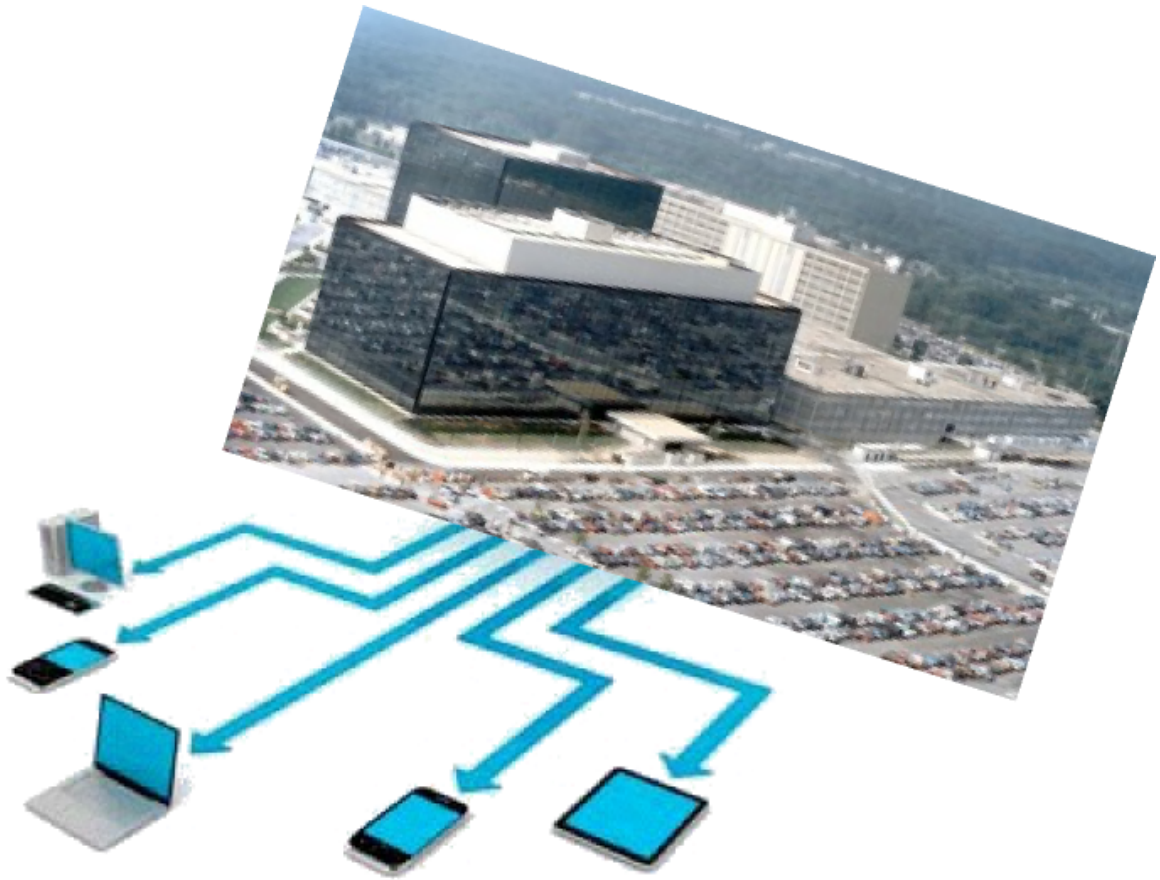
Do we need to model the principals?

Do constrained devices even talk among themselves?

Or just with the Cloud?

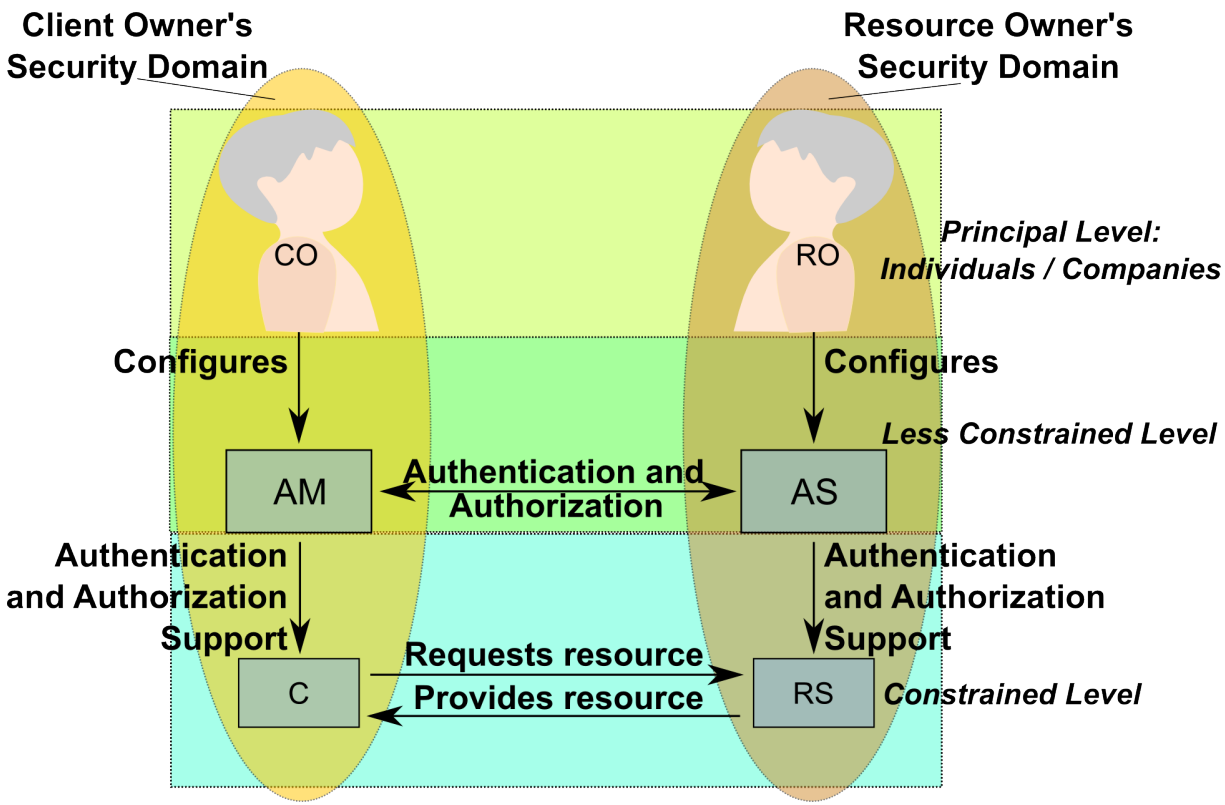


Cloud?



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.

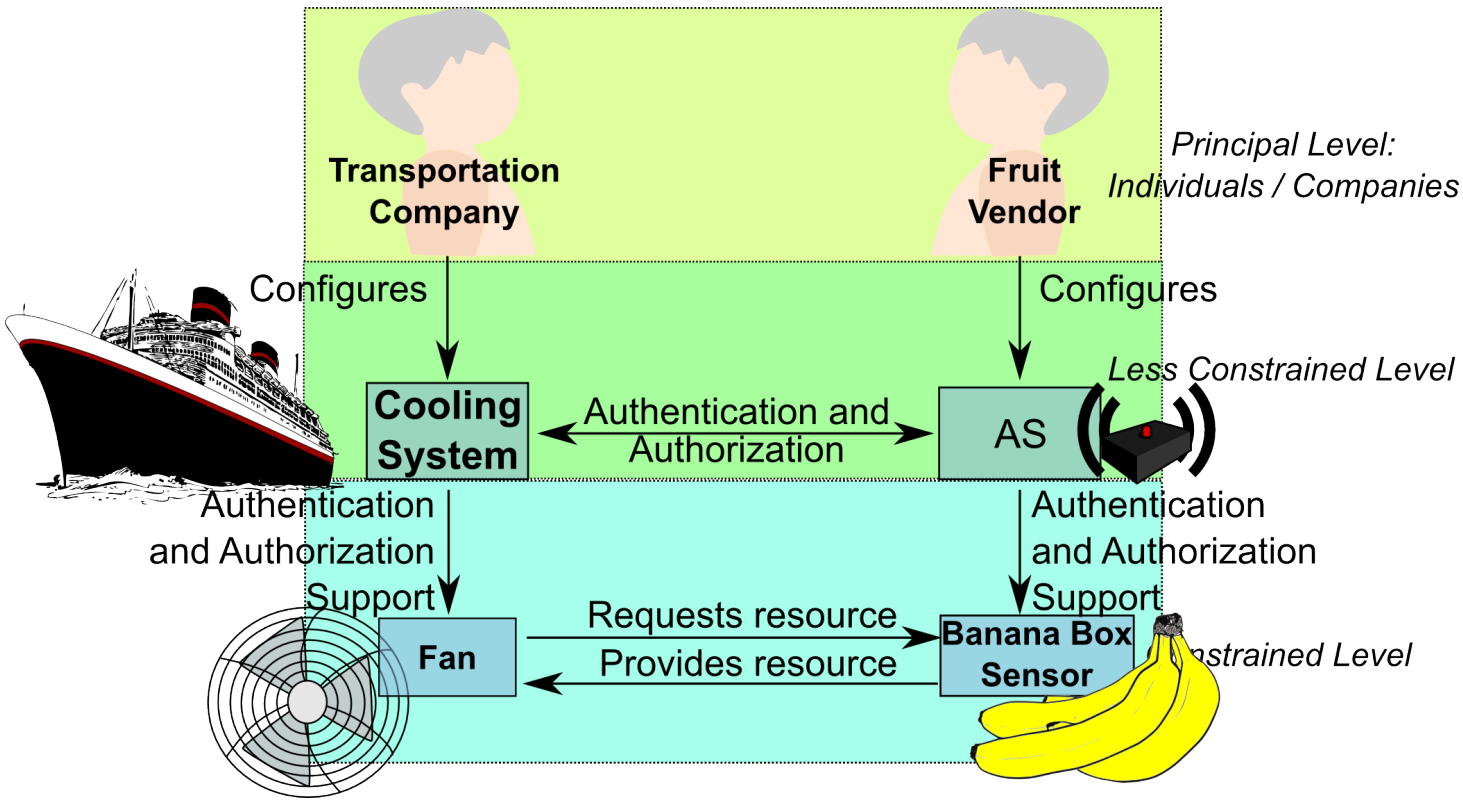


Example: Container Monitoring Use Case

- A fruit vendor grows bananas in Costa Rica for the German market.
- The fruits have to be transported to Germany and stored in a ripening facility.
- During transport and storage the fruits have to be cooled and ventilated.
  - Fruits need to be cooled constantly and evenly spread.
  - Fruits need to be ventilated evenly: Ethylene gas is needed for ripening but too much ethylene leads to early decay of the fruits.
- Use sensors to control temperature and ventilation.

Seamless Cooling and Ventilation

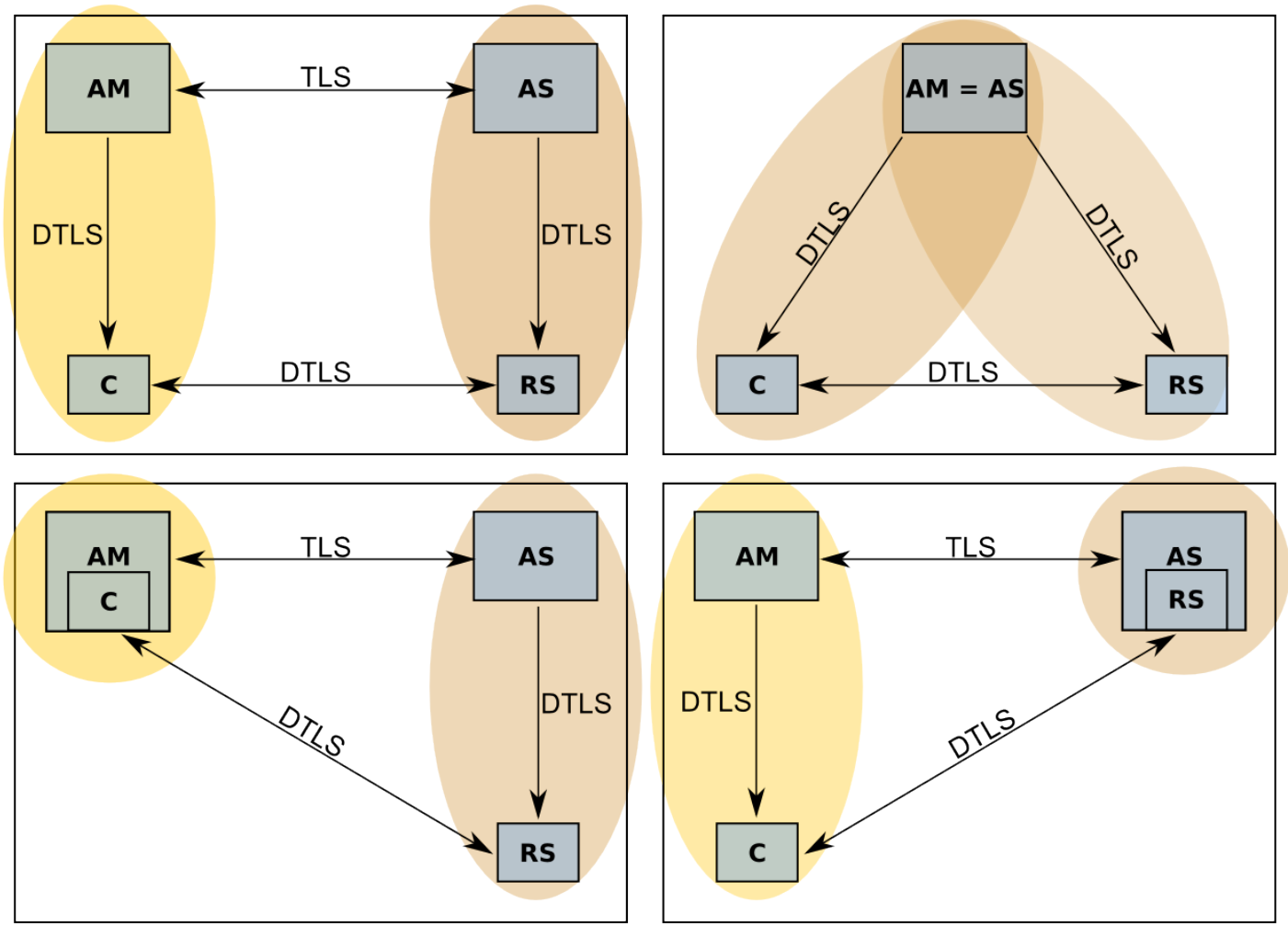
- The cooling and ventilation system of the transportation vehicle needs to communicate with the banana box sensors.
- The fruit vendor configures authorization policies for the sensors.
- The transport company configures authorization policies for the fans.



Constrained to Constrained, Cross-Domain

- Enable constrained devices of different owners to communicate
- Enable dynamic seamless integration with minimal configuration: Once configured, an owner does not have to touch the device
- Flexibility: No painful reconfiguration for every interaction with a foreign device (and the respective authorization server)

Thank you!


AM
TLS
AS
DTLS
DTLS
C
DTLS
RS
AM = AS
DTLS
DTLS
C
DTLS
RS
AM
C
TLS
AS
DTLS
DTLS
RS
AM
TLS
AS
RS
DTLS
DTLS
C