Interaction of "Things" with the "big" Internet: Authentication and Authorization

Stefanie Gerdes, Olaf Bergmann, Carsten Bormann
{gerdes | bergmann | cabo}@tzi.org

2015-07-18

Communication in Constrained Environments

- Constrained Application Protocol (CoAP, RFC 7252)
  - designed for special requirements of constrained environments
  - Similar to HTTP (RESTful architecture style)
    - server has items of interest
    - client requests representation of current state
- Datagram Transport Layer Security (DTLS) binding
- Authorization?

Problem Statement

- A Client (C) attempts to access an item of interest, a web resource (R), on a Server (S) (also called Resource Server (RS).
- A priori, C and S do not know each other, have no trust relationship. They might belong to different owners.
- C and / or S are located on a constrained node.
- How can owners keep the control over their data and devices?
- How can constrained devices establish trust relationships and communicate securely?

Actors

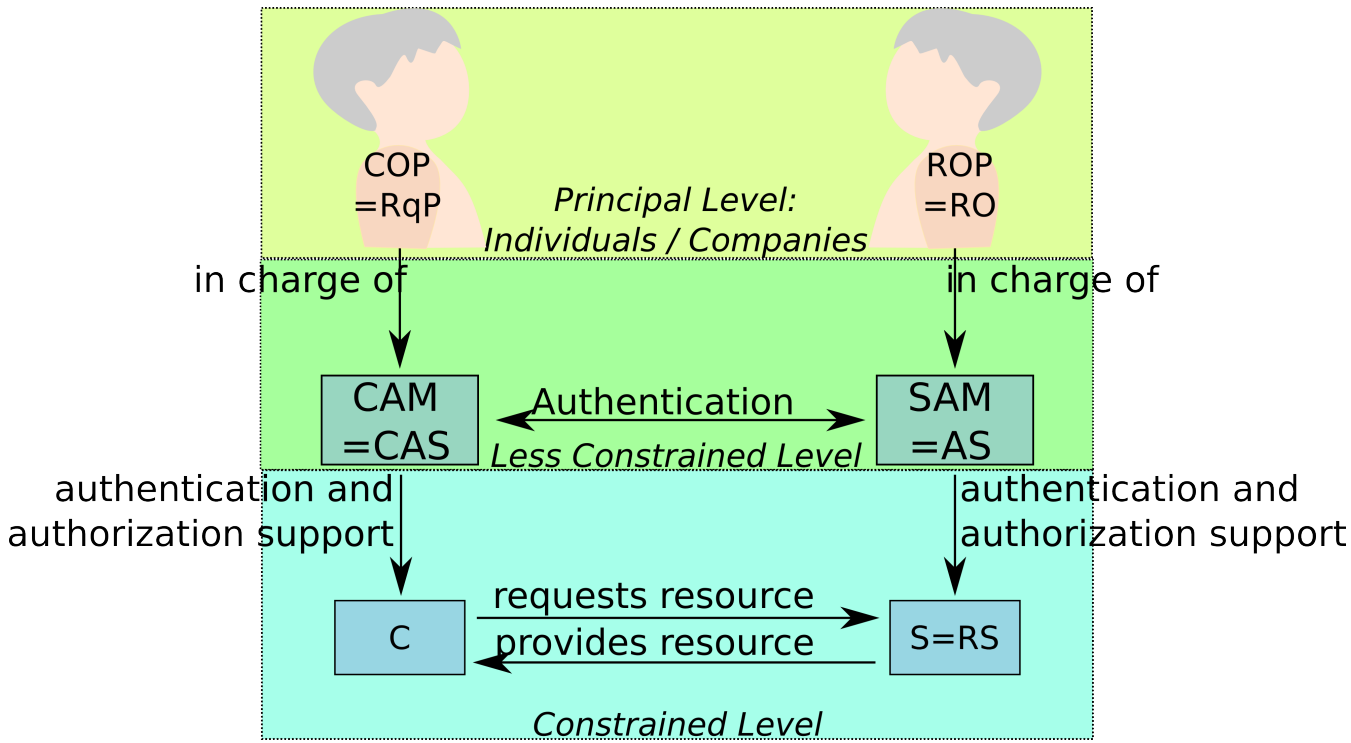
- Actors are **model**-level
  - defined by their tasks and characteristics
- Several actors **MAY** share a single device.
- Several actors **MAY** be combined in a single piece of software.
  - for a specific application
  - for a specific protocol
- Do not prematurely reduce model to one application/protocol

Constraints

- C and S
  - may not have user interfaces and displays.
  - are not able to manage complex authorization policies.
  - are not able to manage a large number of keys.
- Use less-constrained devices for more difficult tasks.

Actors in the Architecture

- C and S are constrained level actors: must be able to operate on a constrained node.
- C and S are controlled by principals in the physical world who specifiy security policies. C and S must enact these policies.
- Authorization Managers CAM (=CAS) and SAM (=AS) help their constrained device with authentication and authorization.



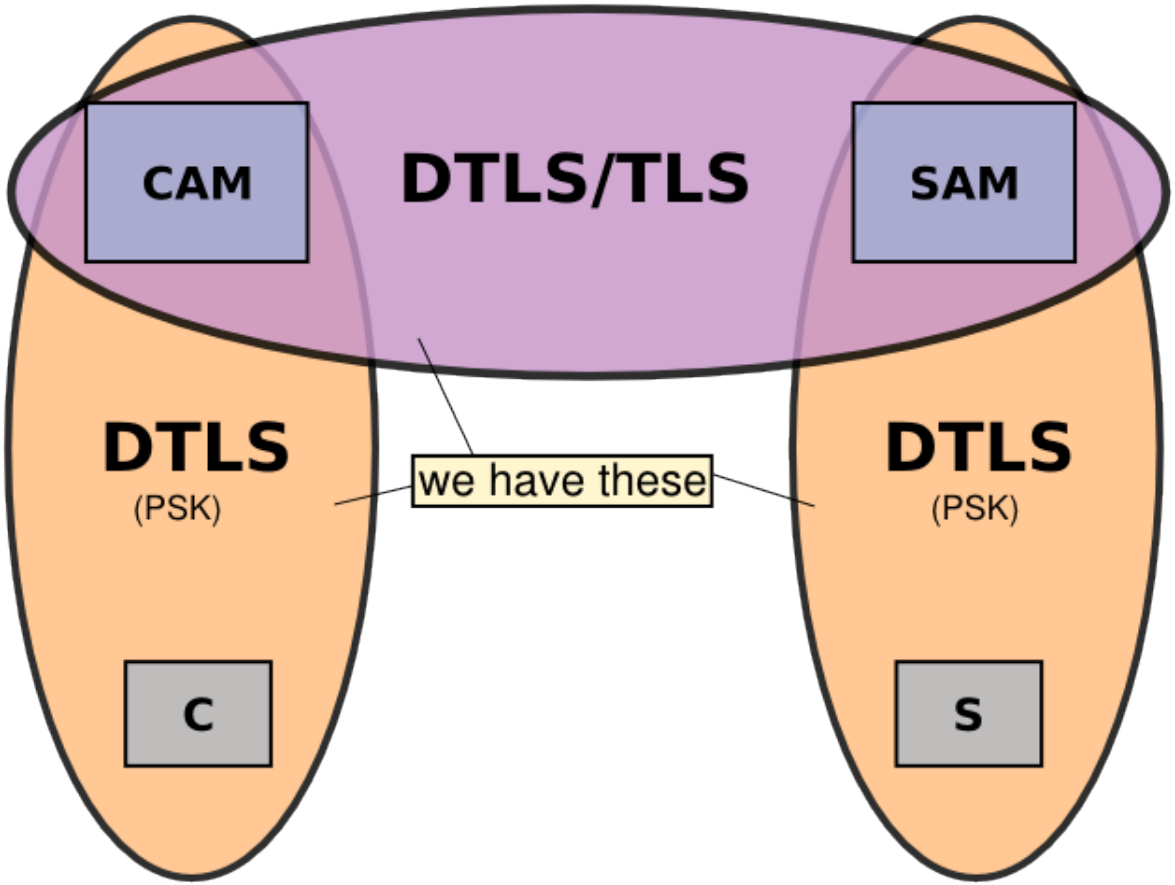
Authorization Managers' Tasks

- Obtain the security objectives from their principal (Provide a user interface).
- Authenticate the other party.
- Use common security protocols.
- Provide simplified authorization rules and means for authentication to their constrained devices.

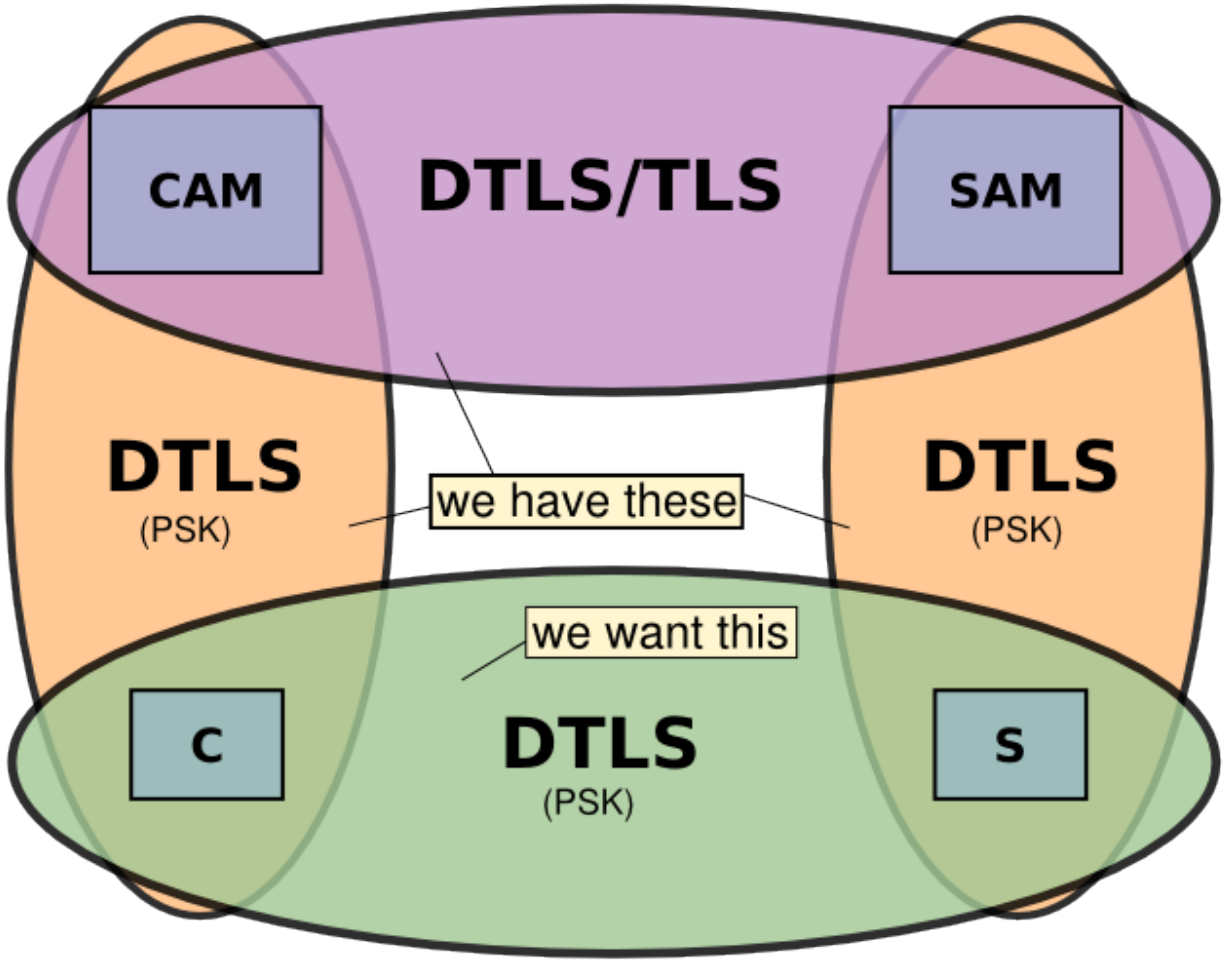
Features of the Delegated Authentication and Authorization Framework (DCAF)

- Optimized for constrained environments.
- Secure exchange of authorization information.
- Secure exchange of session keys.
- Establish DTLS channel between client and server.
- Support of class-1 devices (RFC 7228).
- Use only symmetric key cryptography (DTLS with PSK) on the constrained nodes.
- Support of CoAP Observe and blockwise transfer without additional overhead.
- Relieve constrained nodes from managing complex authentication and authorization tasks.
- Fine-grained Authorization on the client and on the server side possible.

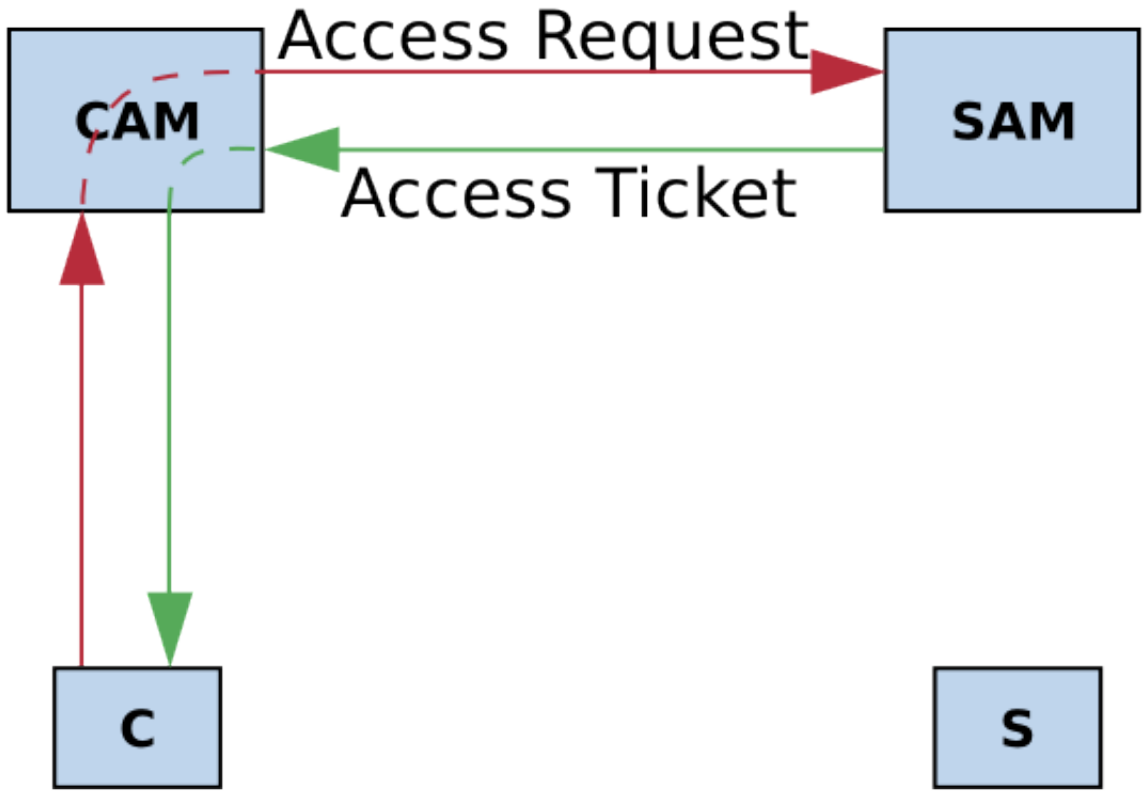
Initial Trust Relationships



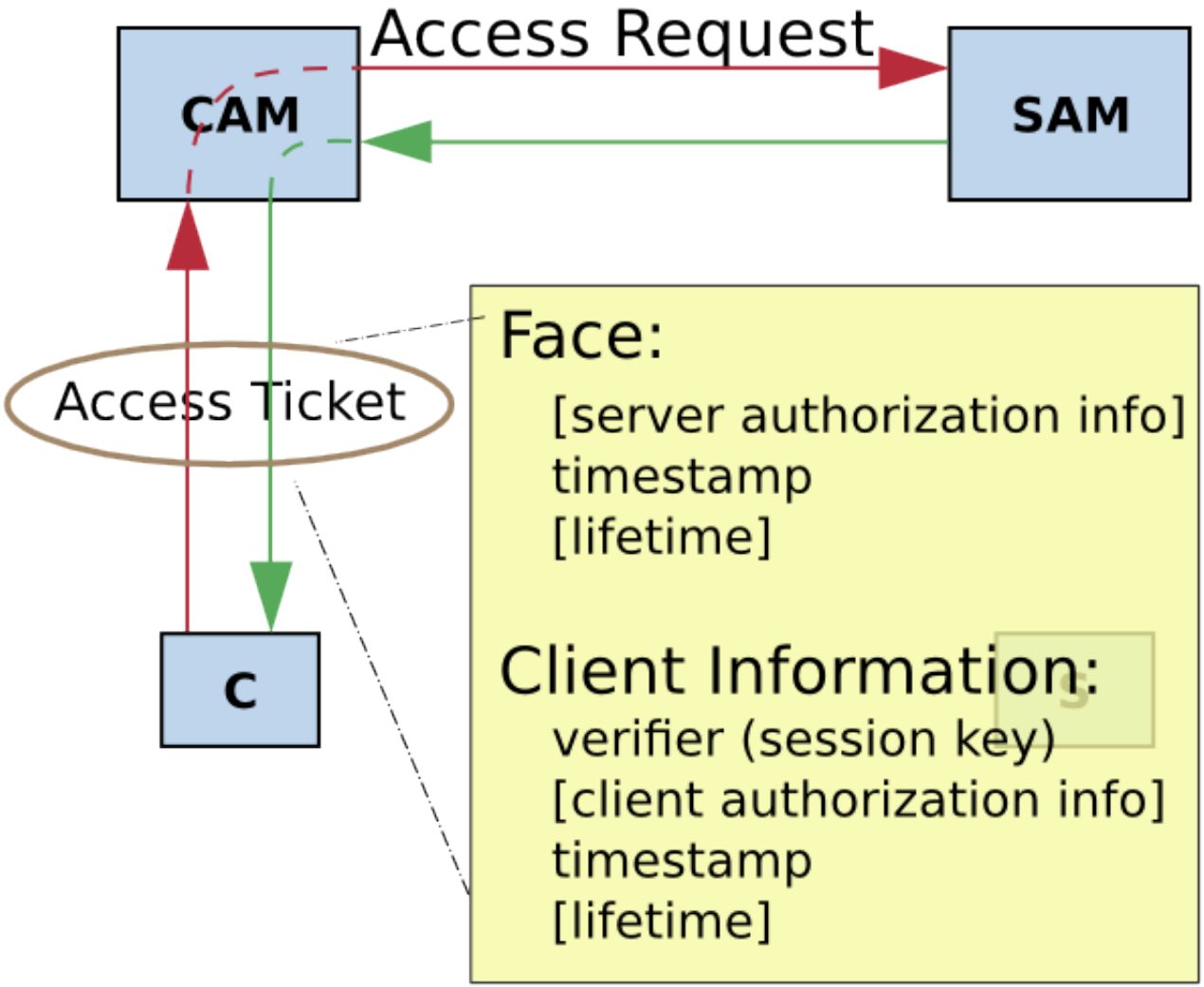
Trust: The Complete Picture



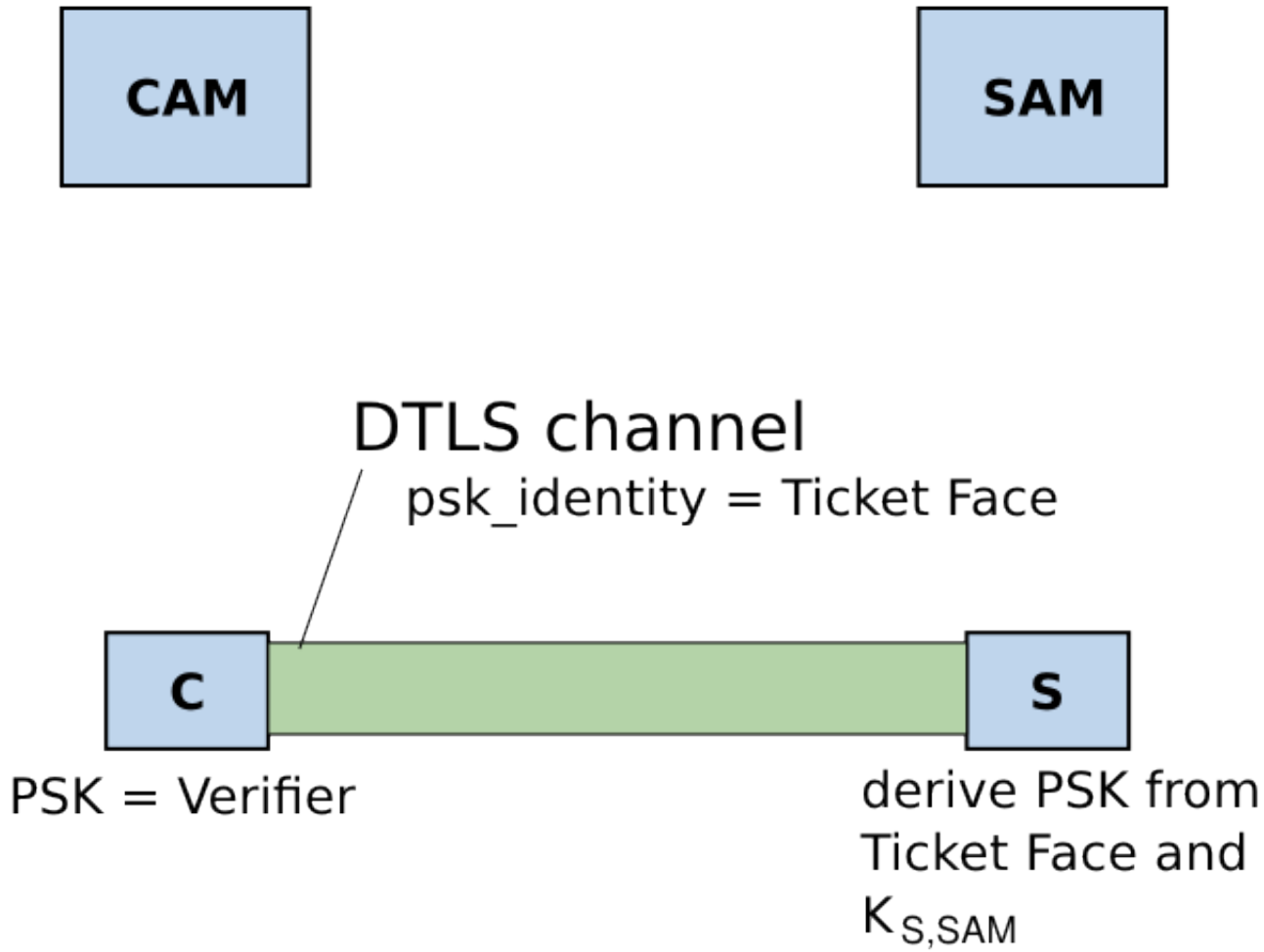
Contact S's Less Constrained Device for Authorization



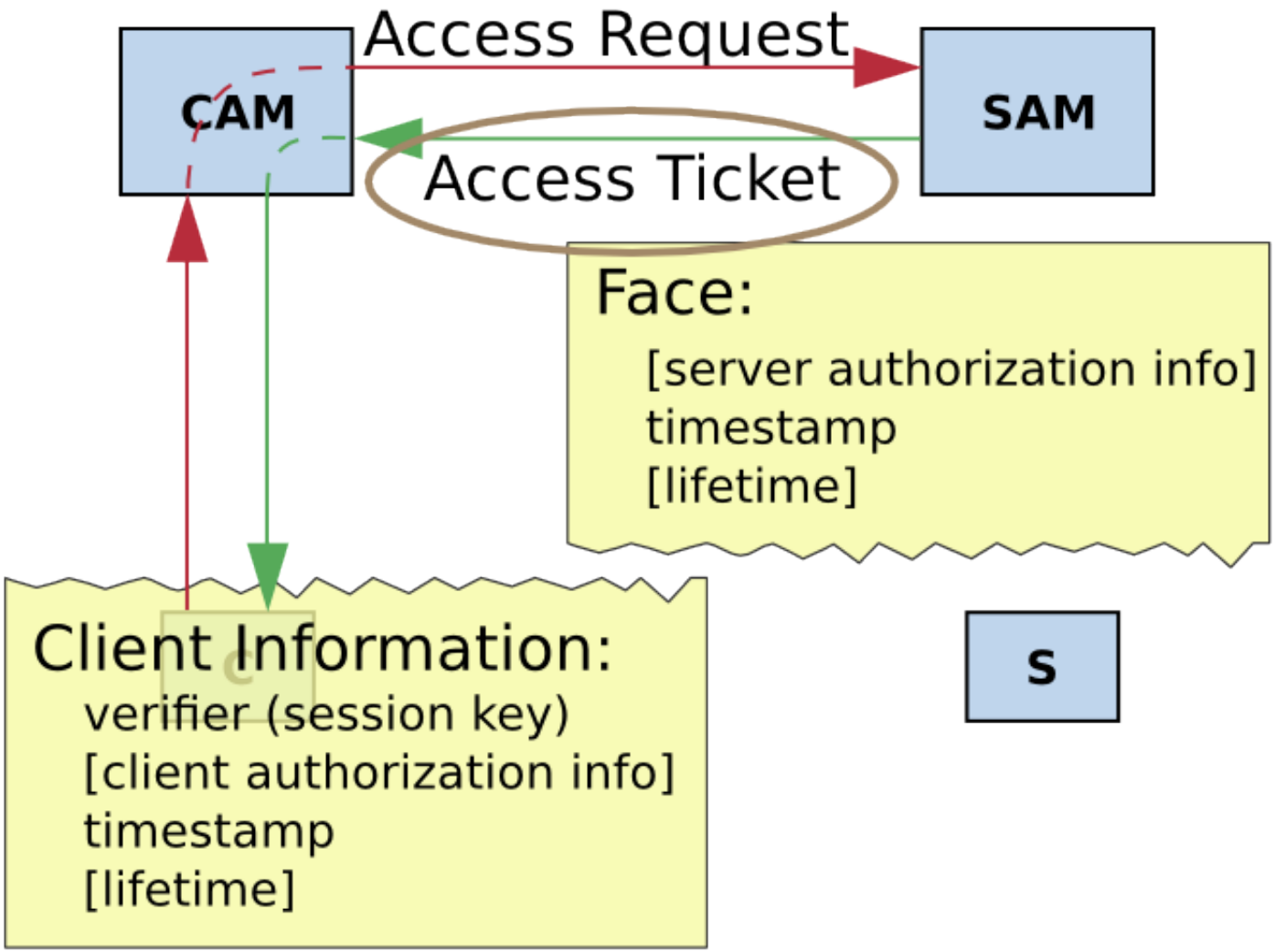
Access Ticket: Obtaining an Access Ticket



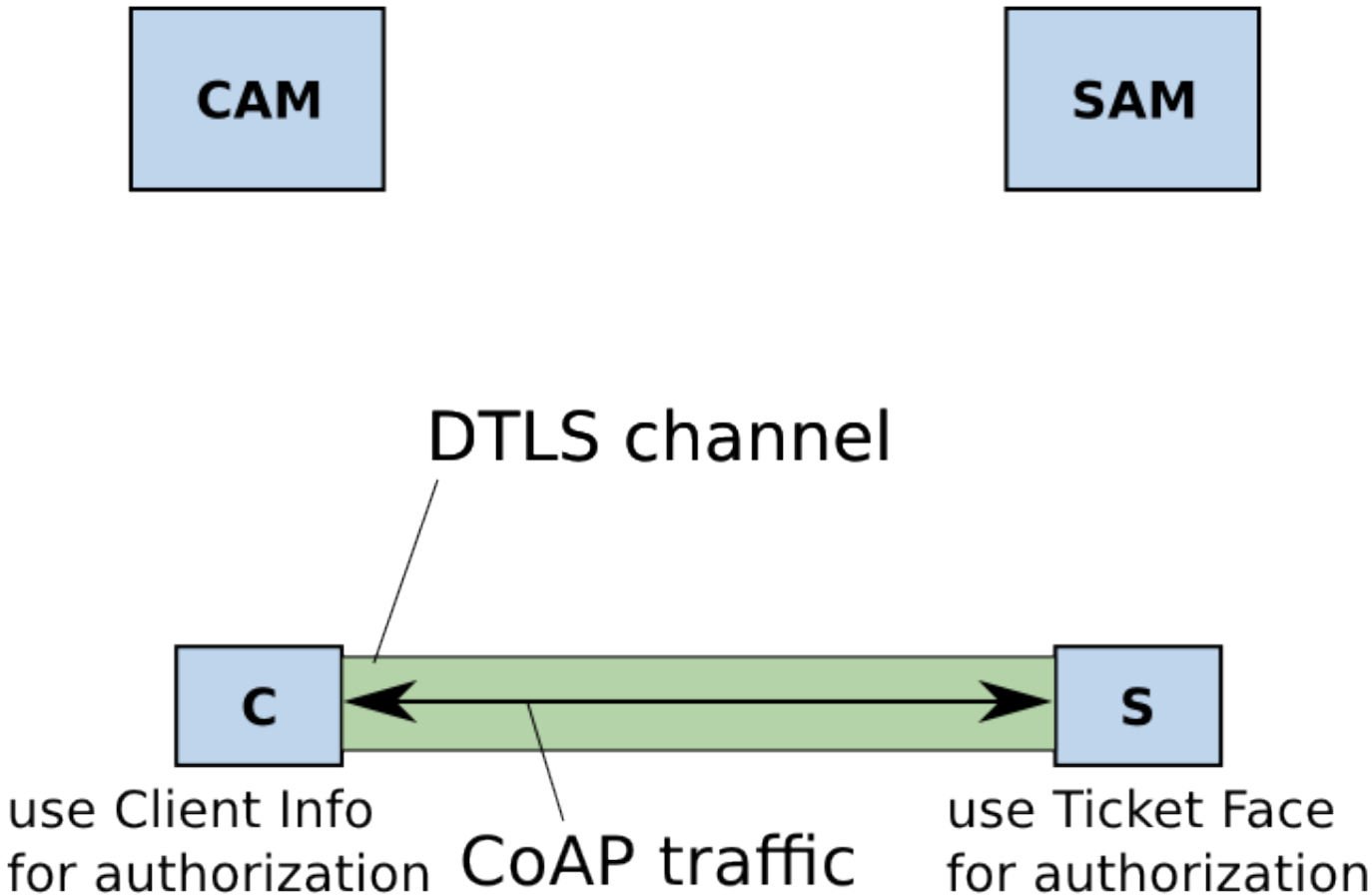
PSK Derivation



Access Ticket Parts



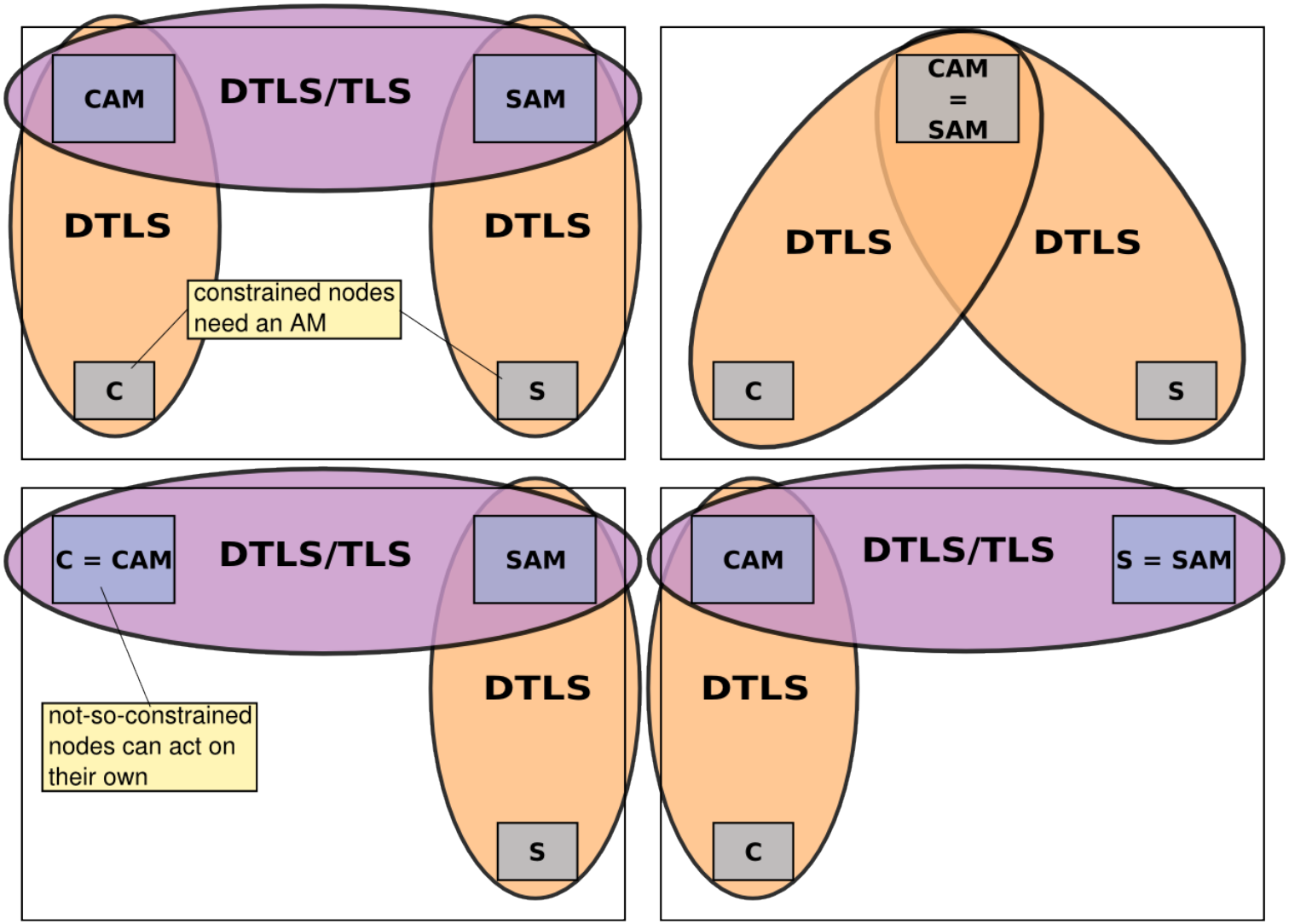
Authorized Requests Over DTLS



Flexibility

- DCAF can be used as a simple protocol for secure transmission of DTLS pre-shared keys (authentication and implicit authorization).
- DCAF can additionally securely transmit authorization information to the server and / or the client.
- DCAF defines how combinations of actors work together.
- DCAF can be used as needed.

Combined Actors



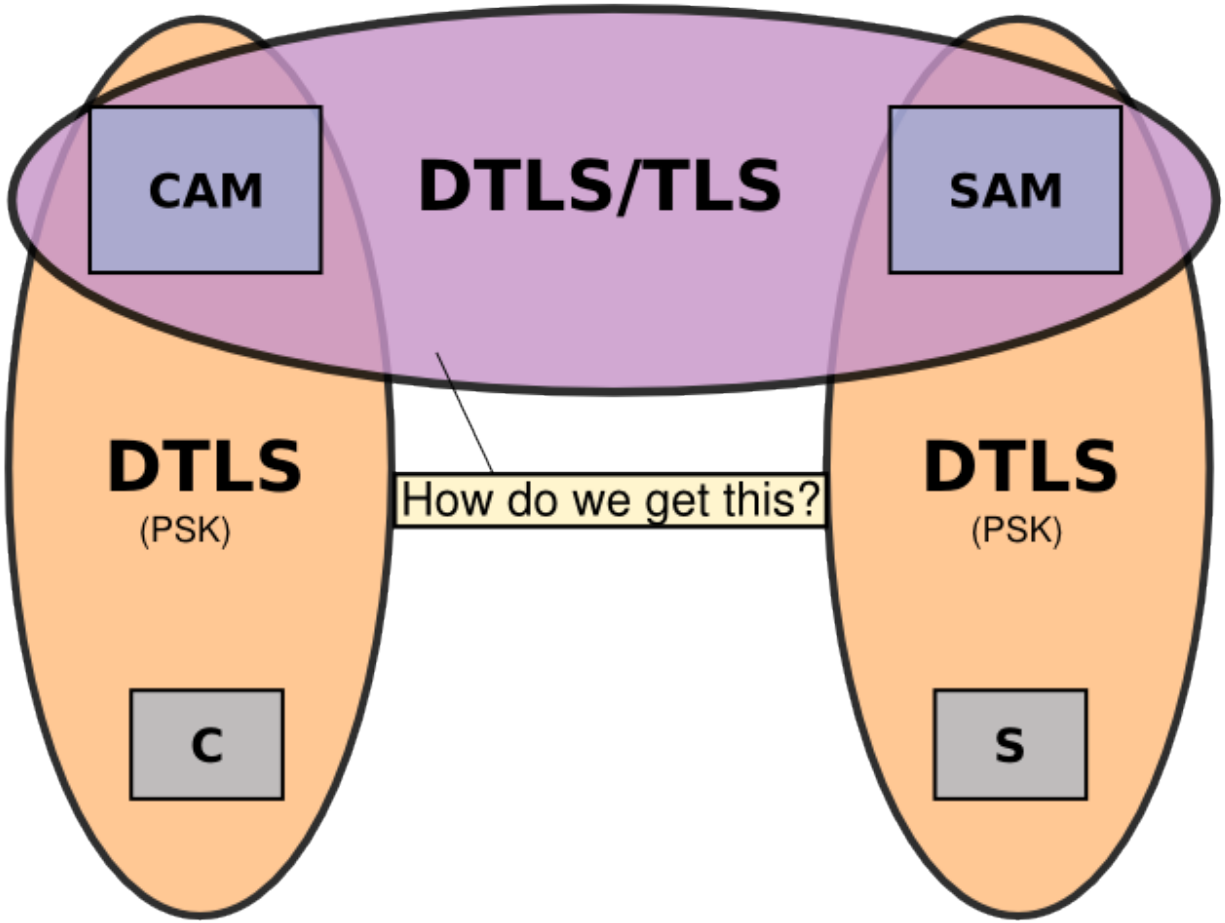
Evaluation

Reference implementation adds

- about 440 Bytes Code
- 54 Bytes data for ticket face
- 722 Bytes parser for CBOR payload

to existing CoAP/DTLS server (ARM Cortex M3).

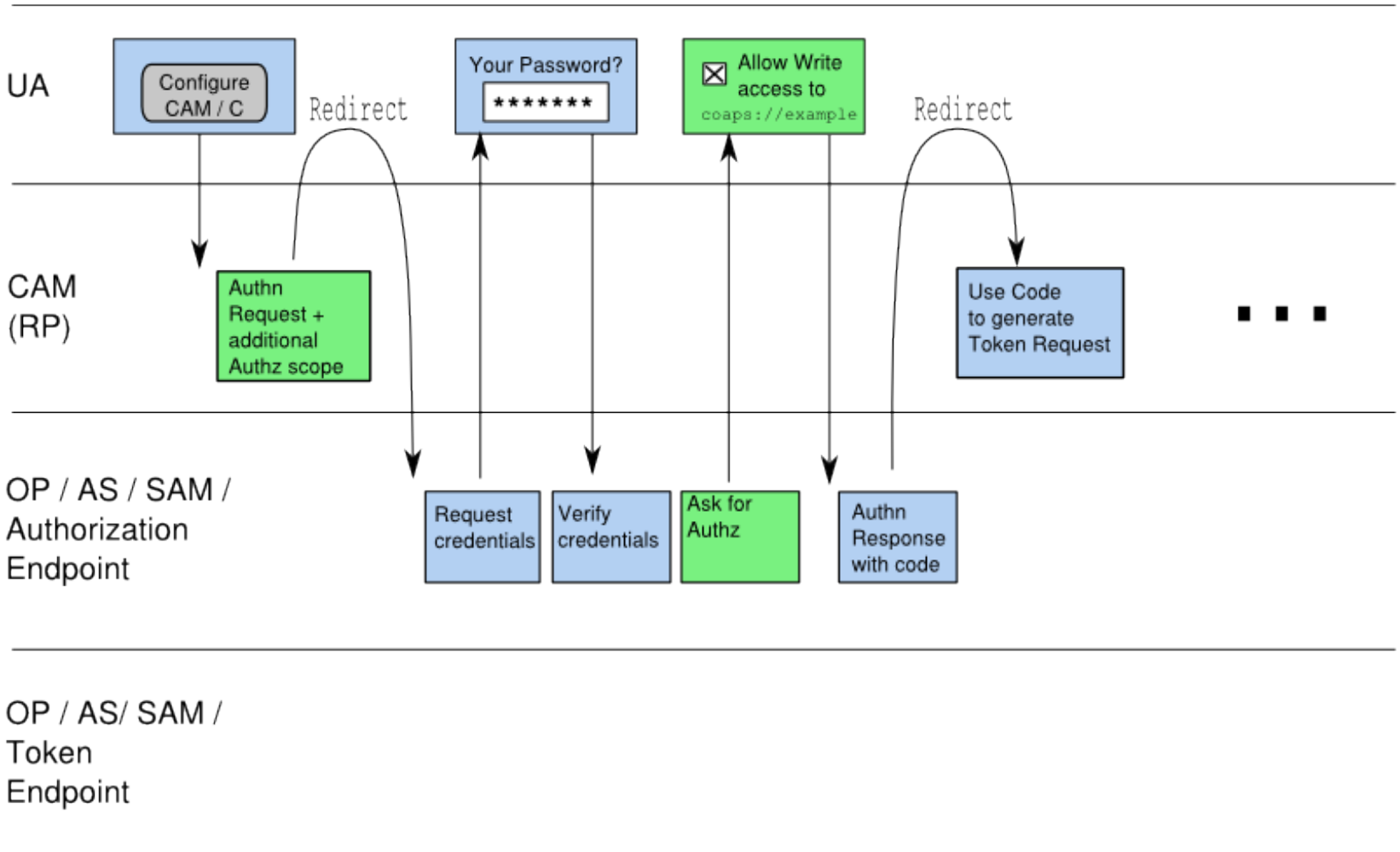
Authenticated Authorization on the Less Constrained Level



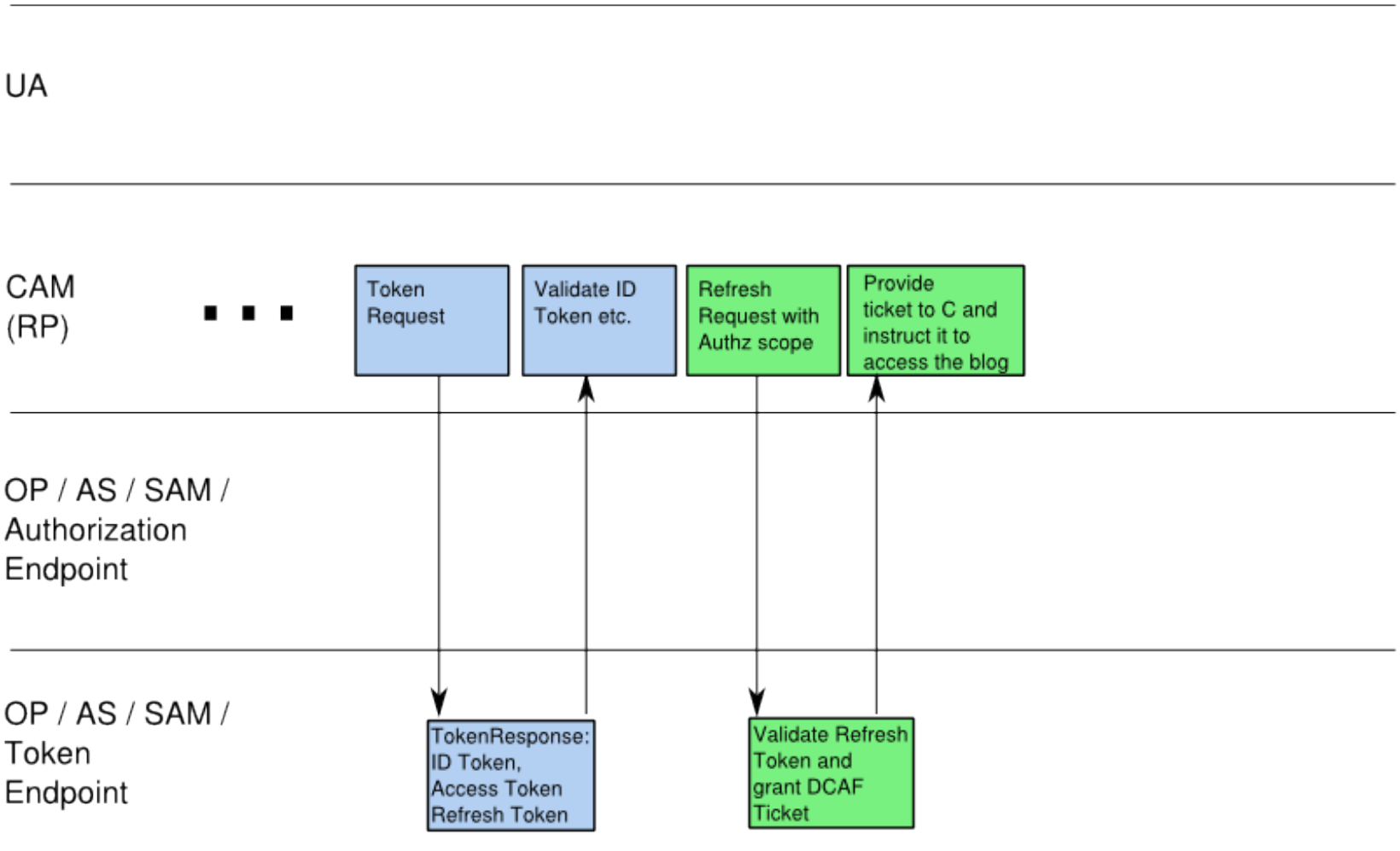
Example 1: Publish Sensor Values in a Blog

- Use OpenID Connect with OAuth and DCAF:
  - Use OpenID Connect for authentication on the less constrained level.
  - Use OAuth for authorization on the less constrained level.
  - Use an OAuth Authorization Endpoint that can speak OAuth and issues DCAF Tickets.

Flow Part 1: Authentication with OpenID Connect



Flow Part 2: Authorization with OAuth/DCAF



Summary

- Use DCAF for authentication and authorization on constrained devices.
- Enable constrained devices to enact the principals' security objectives.
- Use less constrained nodes for the more difficult tasks.
- Use common protocols on the less constrained level that interoperate with DCAF.
- Use OAuth scopes for authorization on the less constrained level.
- Translate OAuth scopes to DCAF tickets for authorization on the constrained level.

Next Steps

- Collect more examples for interoperation flows