Architecture, terminology and problem statement for ACE

Updated after side meeting March 23

Göran Selander
Ludwig Seitz

IETF 92 ACE WG, Dallas, March 24, 2015

Architecture, terminology and problem statement

- Several drafts are discussing architecture for ACE, including (in order of appearance):

  [1] draft-seitz-ace-problem-description

  [2] draft-gerdes-ace-actors

  [3] draft-tschofenig-ace-oauth-bt

  [4] draft-greevenbosch-ace-comparison

  [5] draft-maler-ace-oauth-uma

- [1] and [2] also make an attempt to set the scope of ACE in terms of more comprehensive requirements
- All build on existing terminology, such as OAuth/UMA except [2] which has its own terminology
- There is no common formulation (beyond the charter) of what is the problem ACE should work on

Questions for ACE WG

1. Should the WG try to agree on
   a. Architecture (= type of nodes/functions between which protocols in the scope of ACE are running)?
   b. Names of nodes/functions in the architecture?
   c. Other terminology (e.g. “authorization”)?
   d. Requirements?
   e. High level "problem statement” (e.g. information flow between nodes)?

2. If “yes” on any of the above, what should be the process and timeline for coming to an agreement?

… assuming there is a “yes”,
what could this
architecture, terminology,
and problem statement
be?

Hybrid proposal

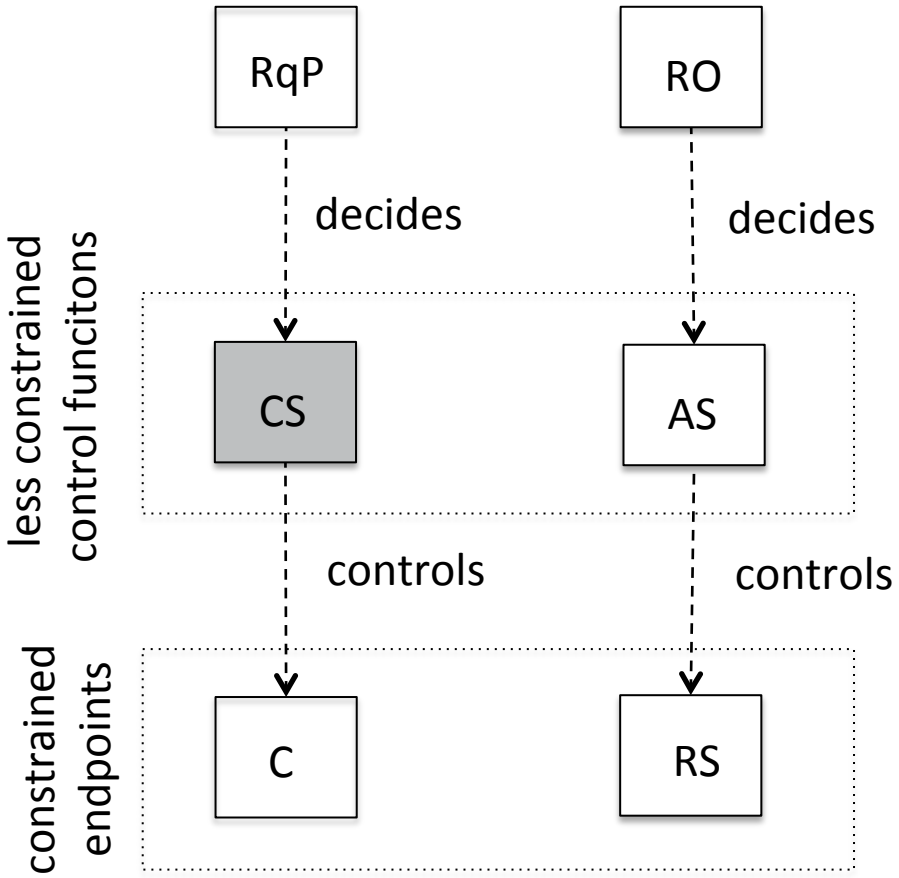
1. Architecture inspired by [2]
2. OAuth/UMA terms, as a starting point
   – adapted to the constrained setting
3. Problem statement inspired by [1] and other drafts

High level problem statement: Overall goal of ACE is to **control** and **protect** interaction between **potentially constrained** endpoints.

Architecture and terminology

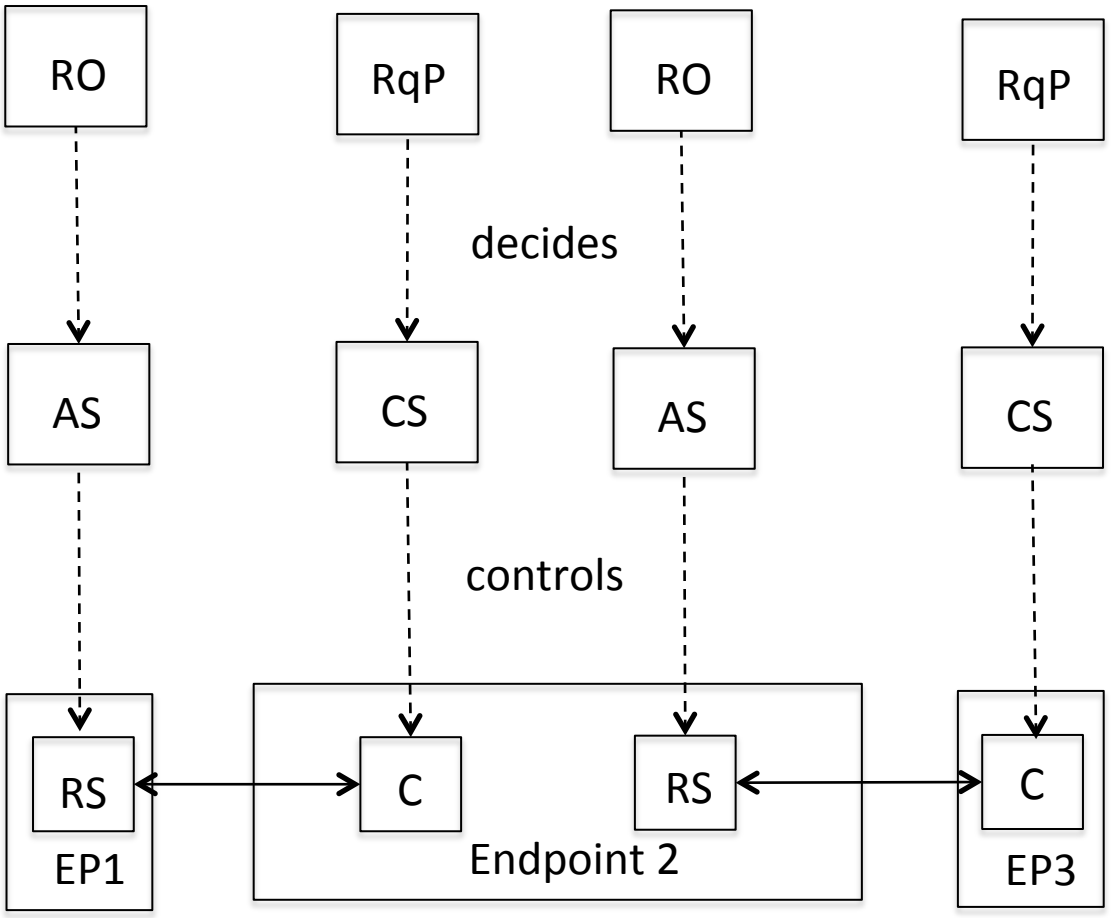
- Resource Server (RS)
  - hosting resources
- Resource Owner (RO)
  - deciding about the resources
- Authorization Server (AS)
  - acting on behalf of RO
    controlling how RS interacts with C

- Client (C)
  - requesting access to resources
- Requesting Party (RqP)
  - deciding about the client
- *Configuration Server (CS)*
  - acting on behalf of RqP
    controlling how C interacts with RS

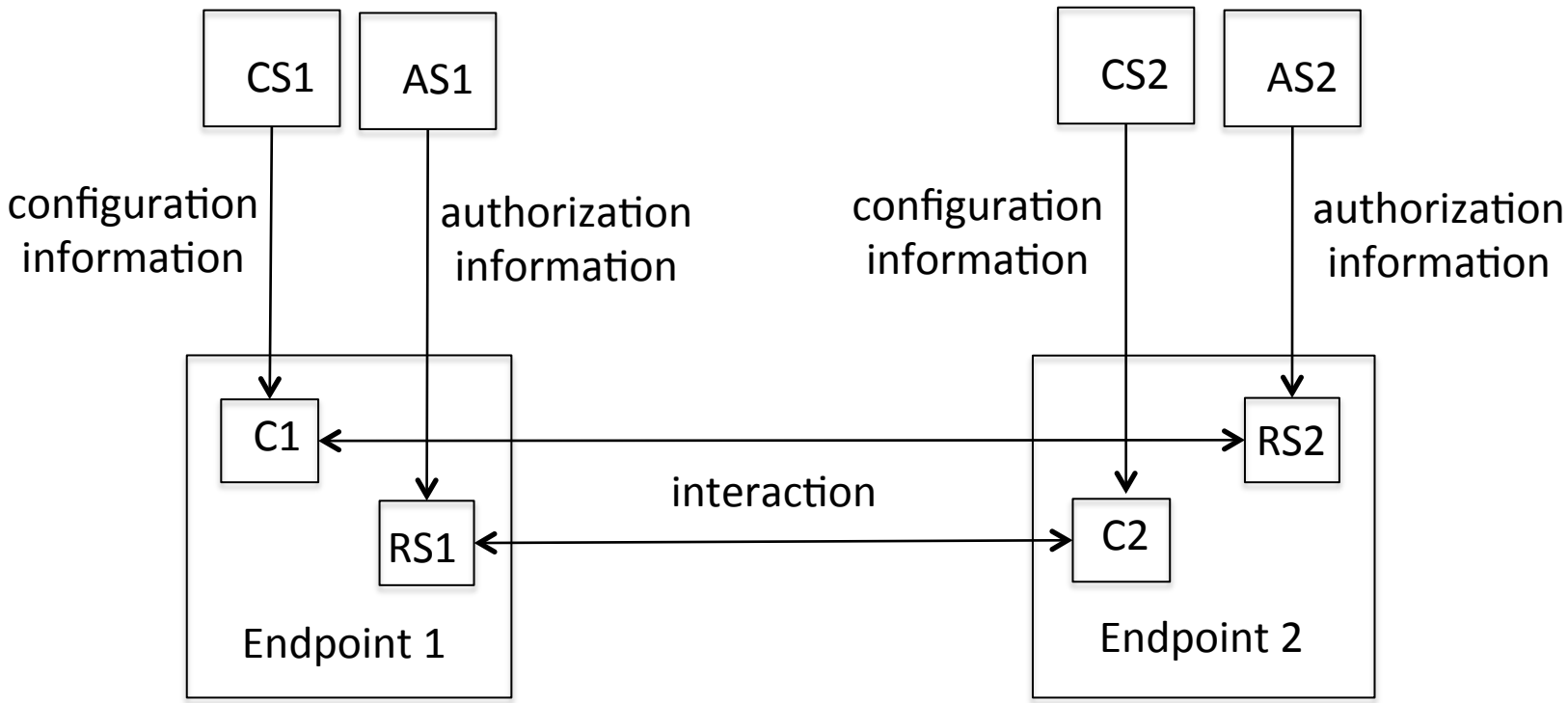


Endpoints and roles

- Each endpoint may host both C and RS; only C; or only RS functionality
- Thus multiple control functions (CS / AS) may be associated to each endpoint



Problem Statement, part 1
Information flows for interaction and control

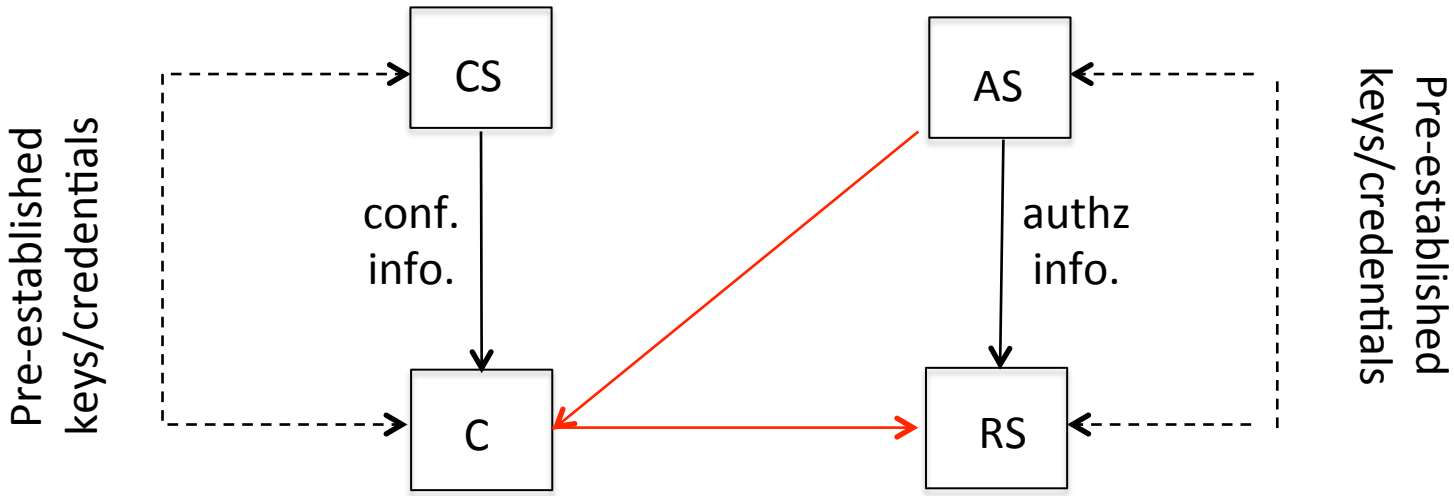


These information flows needs to be protected end-to-end

Problem Statement, part 2
Protection of control information

A. The information controlling the endpoints needs to be protected end-to-end, through intermediary nodes
   Ex. (Pull): AS → C → RS (authorization information)

B. We assume that the necessary keys between the constrained nodes and their control functions are pre-established



Problem Statement, part 3
Protection of interaction

A. Messages between endpoints needs to be protected end-to-end, through intermediary nodes

   Ex.: C1 <-> Forward Proxy <-> RS2 (request/response)

B. Keys/credentials needs to be established in the endpoints to protect the interaction