Proposal:
Security for Nice F2F Plugfest

Johannes Hund
(johannes.hund@siemens.com)

Oliver Pfaff
(oliver.pfaff@siemens.com)

Given Things, Implication

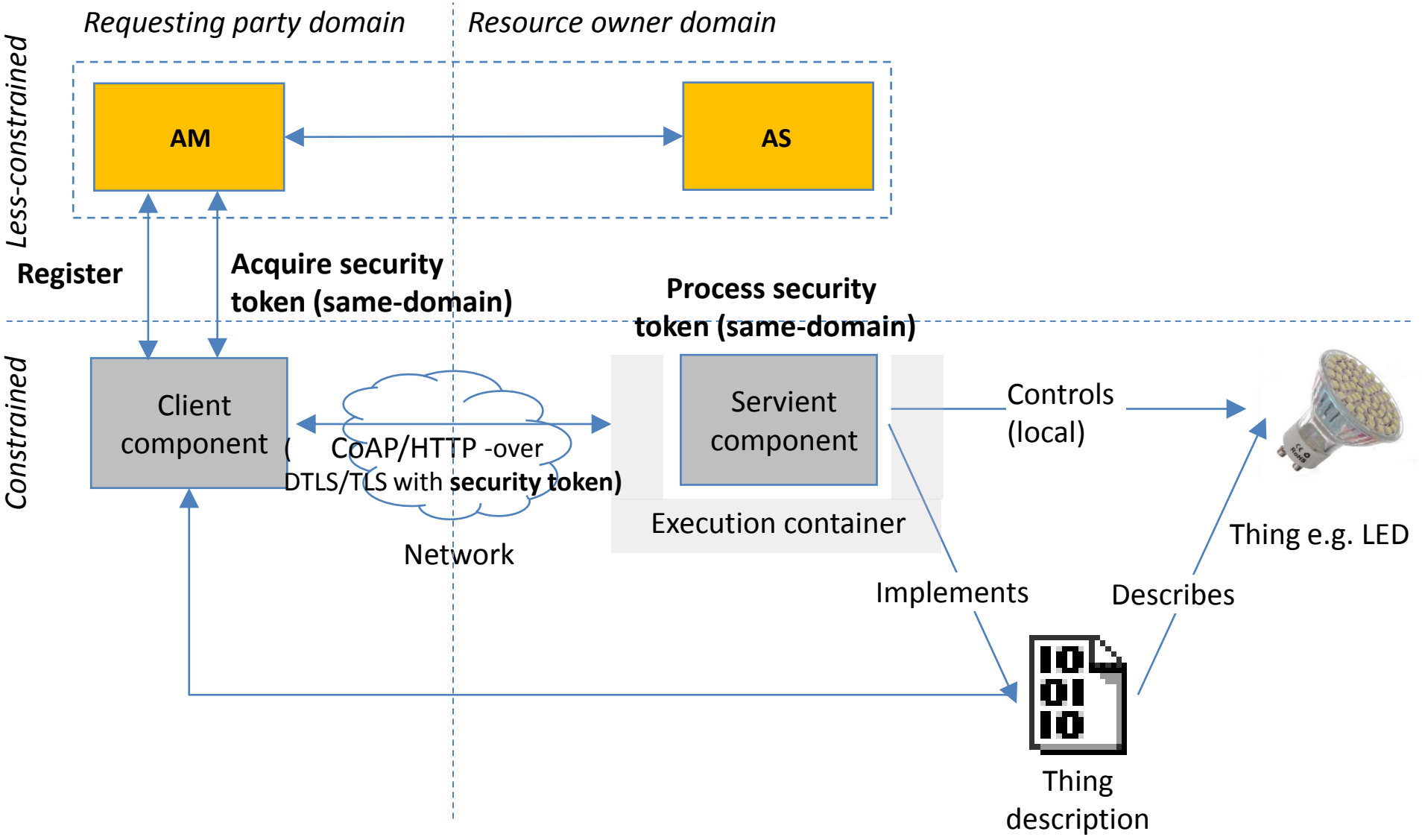
- Target audience:
  - Those who participated in the Sapporo F2F (without security)
  - Others who want to get started in the Nice F2F
- Less than 8 weeks remaining
  - Security-enabling can not demand detailed security domain knowledge
  - Security-enabling clients and resource servers shall not demand more than 1 PW of efforts

➢ Must go with a **low entry** option - in style of a minimal subset

Proposed Recipe for Security-Enabling

1. **Add secure communications** but do not limit security-enabling to that
   - Plumbing DTLS underneath CoAP resp. TLS underneath HTTP is well-understood and straight-forward – *communications should be secured but 2016 is ca. 10-20 years to late to promote this as a highlight*
   - Use a simple key management and configuration model
2. **Focus** on **authorizing** and hence **authenticating** the requests sent over the network
3. **Adopt** the **architectural model** of **IETF ACE**
   - *Client (C)*: constrained, accesses resources
   - *Resource server (RS)*: constrained, serves multiple resources
   - *Authorization manager (AM)*: less/non-constrained, represents multiple Cs
   - *Authorization server (AS)*: less/non-constrained, represents multiple RSs
4. **Rely on existing standards** where-ever possible
5. **(Re-)use well-known artifacts** esp. OAuth, JWT (implementing libraries exist and are interoperable) in incarnating this model

Plugfest@Nice F2F – Security-Addition



Security-Related Processing Tasks

- Registration:
  - C: request/response according OAuth dynamic registration (RFC 7591)
  - RS: avoided
- Token acquisition:
  - C: request/response according OAuth client credentials grant (RFC 6749)
  - RS: not needed
- Token supply and consumption:
  - C (for supply): for HTTP according RFC 6750, for CoAP in style of RFC 6750 (see draft-tschofenig-ace-oauth-bt-01)
  - RS (for consumption): for HTTP according RFC 6750, for CoAP in style of RFC 6750 (see draft-tschofenig-ace-oauth-bt-01)
- Token validation:
  - C: n.a.
  - RS: according RFC 7519 (low entry option uses a minimal JWT)

Supply of Components

- C: **any Plugfest participant** – like in Sapporo
  - The security-enabling of Cs is described in the accompanying How-To. This security-enabling aims at minimizing the impact on C
- RS: **any Plugfest participant** – like in Sapporo
  - The security-enabling of RSs is described in the accompanying How-To. This security-enabling aims at minimizing the impact on RS
- AM and AS: Siemens
  - Not meant to exclude other interested parties - just to make sure they will exist
  - If there is another interested party then:
    - For simplicity it is suggested not to consider AM / AS interop as a goal for the Nice Plugfest
    - I.e. the (AM, AS)-tuple that C and RS are utilize are either Siemens or TBD

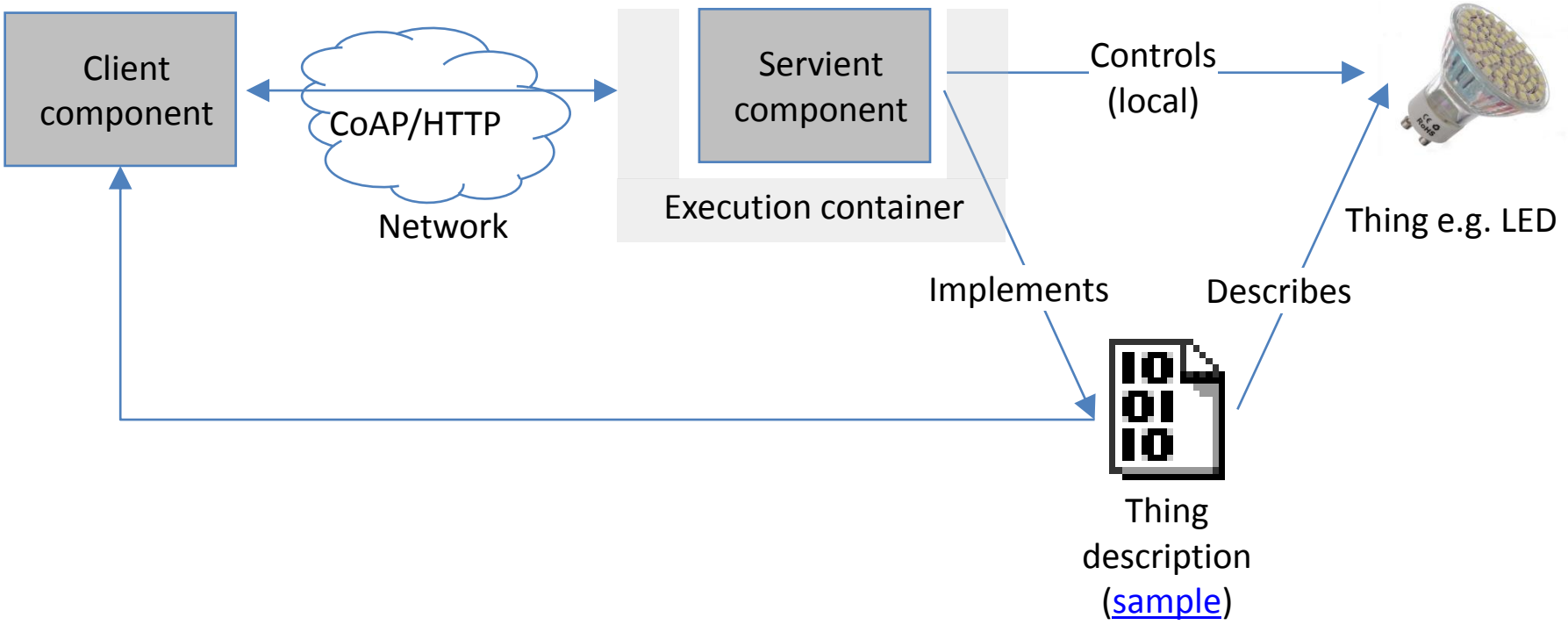
Security-Related Material for Plugfest

- **Overview slides**
  - Does exist, this deck
- **HowTo description** incl. request/response/object prototypes
  - Does exist, accompanying document
- **Cheat sheet** incl. code snippets
  - Does exist, accompanying document
- **Running AM and AS instances** (deployed in e.g. AWS IaaS)
  - Available soon
- **Hands-on help** (via mail or mailing lists)

Security-Related Implementation Options

- **TLS and DTLS usage options:**
  - Default: use HTTP-over-TLS resp. CoAP-over-DTLS plus security tokens for security-enabled interactions between C and RS
  - Alternative: use HTTP-plain resp. CoAP-plain plus security tokens for security-enabled interactions between C and RS
- **AS token content options:**
  - Minimal: no specific access control information
  - Normal: access control information in AIF style (draft-bormann-core-ace-aif-03)
- **AS token signature options:**
  - Default: ES256 (asymmetric, elliptic curve cryptography supported by many JWT libraries)
  - Alternative: HS256 (symmetric, supported by almost all JWT libraries)

Recap:
Plugfest@Sapporo F2F – Security-Unaware



- Plugfest description
- Plugfest results
- Architectural model