The component Caller API is a service that exposes ACS (CORBA) components and their methods through a REST API. Currently, it is deployed as part of the ACS services in the production environment. Although, there is no known application consuming from the REST API.

It was originally envisioned to have at least authentication as a security mechanism. However, this implementation has been postponed for multiple reasons.

Recently, a risk analysis memo was prepared by ICTTAG, depicting a couple of vulnerabilities present in the ALMA SW APE (ALMA Production Environment). One of these vulnerabilities was the Component Caller API, which runs inside the APE environment. The APE environment is a set of servers, hardware, and network devices configured in an enclosed environment and protected by a firewall: The APE firewall. The APE firewall, however, has deliberately been configured to allow several actions directly on the internal machines, including access to the Component Caller API REST from inside the organization network. There's a second firewall: JAO firewall, protecting the organization's network from outside threats.

It can be seen that the vulnerability exposes the main system to be accessed and manipulated from inside the JAO network by either careless or malicious actions. At the moment, there is no mechanism to restrict the user, IP, or limit the ability to interact with certain components/methods, which gives full control to anyone connecting to this API.

There are several ways to reduce or eliminate the risks that have been identified, like adding authentication and roles to the software, running the REST API through a proxy, or even fine-tuning the firewall configuration. A combination of the three could probably achieve the most secure and flexible solution. There's likely a whole set of additional ways to solve the situation, but we will consider three (complementary) alternatives in this proposal.

Custom Implementation

A custom implementation could allow several levels of granularity on its configuration, which includes the following:

- Authentication
- Roles
- Access restriction (Components, types (IDLs), methods. etc.)
- Access logging

The CDB configuration of the Component Caller API could be used to easily configure the different security profiles.

For authentication, it could either be used the CDB configuration (cdbRead could access credentials/tokens), or a service such as LDAP, ALMA Registry, etc.

Proxy

A proxy could be configured either for the server or the APE environment as a whole, which limits access to the resources. It could be used for:

- Authentication (Through CAS / KeyCloak)
    - Would most likely require changes to the software to take advantage of CAS / KeyCloak tokens
- Roles (Associated with the source IP)
- Access restriction (Components, types (IDLs), methods. etc.)
    - Would require software changes to consider /<component>/<method> endpoints.

Firewall

The existing firewall could be configured for the APE environment to limit the access to the Component Caller API, allowing specific services to make use of the REST API

- Access restriction
    - Black / White access limitation

To achieve a reasonable level of security, while not imposing big changes in software and the ALMA Production Environment (APE), we believe a mixture of the Firewall and a reduced Custom implementation would allow a secured environment to work with.

- The firewall would be configured to block all communications to port 9000 of gas02, with the exception of specific allowed services (source IPs), such as the ObopsOnlineServer
- The ComponentCallerAPI would block all component / method calls, except the allowed ones, such as the weather station information
    - Allowed methods would be configured in the ComponentCaller XMLDoc

The conclusions will be prepared after reviewing by the stakeholders and possible discussion meetings are held.