

- [Introduction](#)
- [Presentation](#)
- [Hands-On Exercise](#)
 - [Component Caller API](#)
 - [Deployment / Configuration](#)
 - [Interaction](#)
 - [Event Converter](#)
 - [Deployment / Configuration](#)
 - [Interaction](#)
- [Discussion](#)

The Component Caller API is a component that works as a proxy that makes the ACS / CORBA world in a REST API. This enables different functionality, such as connecting standalone applications, plugins and/or web services to the system.

The Event Converter is a component that serves a pre-configured set of events by converting them from ACS' Notification Channel (Based on CORBA's Notification Service) into Redis PubSub model for unreliable subscribers and into Redis queues for reliable subscribers.

At ALMA, these services are being used in a couple of places:

- Integrated Alarms System Plugins to obtain device information
- ObopsOnlineServer to prepare observation and operational reports (Shift Logs)
- Some prototypes have been done to check feasibility of web services to control the system
- Scope
 - Background
 - Component Caller API
 - Event Converter
- Duration: 10 minutes
- [ComponentCallerAPI_EventConverter.pdf](#)

Component Caller API

Deployment / Configuration

- Python Container
 - For instance aragornContainer with the demo CDB
- Component configuration:

CDB/MACI/Components/Components.xml

```
...
<e
  Name="ComponentCallerAPI"
  Code="acs.ComponentCallerAPI"
  Type="IDL:alma/ACS/ACSComponent:1.0"
  Container="aragornContainer"
  ImplLang="py"
  Autostart="true"/>
...
```

Interaction

- Start ACS
- Start a Container
- Interact with a component using curl. For instance to retrieve name and componentState attributes:

```
> curl -XPOST http://127.0.0.1:9000/ -d '{"componentName": "EventConverter", "methodName": "_get_name",
"arguments": {}}'
{"data": "EventConverter"}

> curl -XPOST http://127.0.0.1:9000/ -d '{"componentName": "EventConverter", "methodName":
"_get_componentState", "arguments": {}}'
{"data": {"meta": {"type": "omniORB.EnumItem"}, "data": {"_n": "COMPSTATE_OPERATIONAL", "_v": 3,
"_parent_id": "IDL:alma/ACS/ComponentStates:1.0"}}}
```

Event Converter

Deployment / Configuration

- Install redis

```
> sudo yum install redis
```

- Python Container
 - For instance aragornContainer with the demo CDB
- Component Configuration

CDB/MACI/Components/Components.xml

```
...
<e
  Name="EventConverter"
  Code="acs.EventConverter"
  Type="IDL:alma/ACS/ACSComponent:1.0"
  Container="aragornContainer"
  ImplLang="py"
  Autostart="true"/>
...
```

- XMLDoc

CDB/alma/EventConverter/EventConverter.xml

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<EventConverter xmlns="urn:schemas-cosylab-com:EventConverter:1.0" xmlns:baci="urn:schemas-cosylab-com:
BACI:1.0" xmlns:cdb="urn:schemas-cosylab-com:CDB:1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
  <RedisConn host="localhost" port="6379"/>
  <PubSub>
    <Channel name="example">
      <EventType type="workshop.ExampleEvent"/>
    </Channel>
  </PubSub>
  <ReliableSubscribers>
    <Subscriber name="example" queueLimit="20">
      <Channel name="example">
        <EventType type="workshop.ExampleEvent"/>
      </Channel>
    </Subscriber>
  </ReliableSubscribers>
</EventConverter>
```

- IDL Event Definition

```
#ifndef _ExampleEvent_IDL_
#define _ExampleEvent_IDL_

#pragma prefix "alma"

module workshop {
  const string CHANNELNAME_EXAMPLE = "example";
  struct ExampleEvent {
    string msg;
    long value;
  };
};
#endif
```

Interaction

- Start redis-server

```
> redis-server
```

- Start ACS
- Start Python aragornContainer
- Send events with ExampleSupplier.py

```
import workshop

from Acspy.Nc.Supplier import Supplier

event = workshop.ExampleEvent("Example Supplier", 10)

sup = Supplier(workshop.CHANNELNAME_EXAMPLE)
sup.publishEvent(simple_data=event)
sup.disconnect()
```

- Check Redis

```
> redis-cli
#Get all queues
127.0.0.1:6379> keys *
1) "example"

#Check elements of 'example' queue
127.0.0.1:6379> LRANGE example 0 -1
1) "{\"meta\": {\"type\": \"workshop.ExampleEvent\"}, \"data\": {\"msg\": \"Example Supplier\", \"value\": 10}}"
```

- Limitations: Callbacks
- Improvements
- New Ideas