# Authorization Policy

Authorization policy defines the method of authorization.

An authorization policy is a connector that gets the parameters from the request and maps to a specified service for authorization. In other words, it executes an already defined service for authorization (*authorization service*) and use some of the request parameters for input.

An Interface *references* to a defined authorization policy to restrict the access of the interface.

A service *references* to a defined authorization policy to restrict the usage of the service.

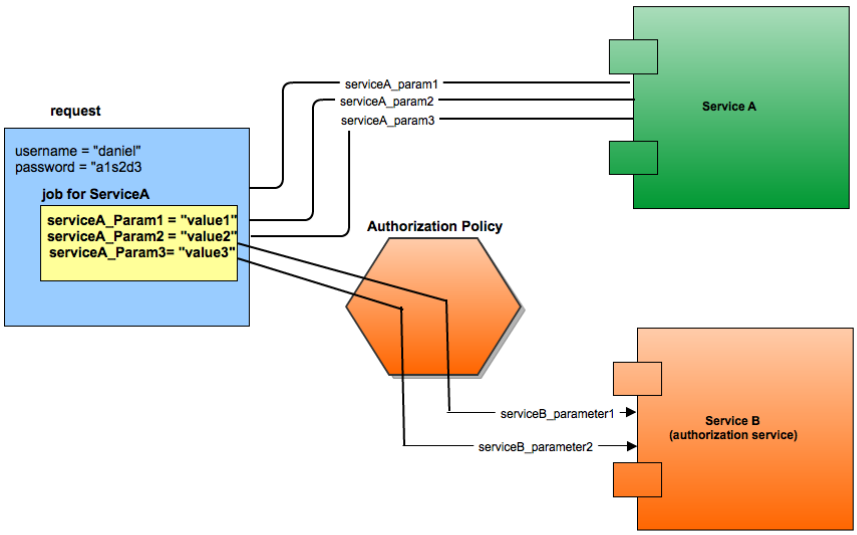
## Authorization Policy Identifier

An interface or a service can use a defined authorization policy for usage restriction.

Authorization policy  should have a unique name so that it can be referenced from an interface or service definition.

## Parameter Mapping

Parameter mapping keeps the information of which request parameter is used for input of authorization service.




As seen on figure, the request parameters first sent to Service B, after successful authorization, request parameters are sent to Service A (the target service).

Service B (Authorization service) requires *serviceB_Parameter1* and *serviceB_Parameter2* parameters. Authorization policy passes serviceA_Param2 to Service B as serviceB_Parameter1.

The parameter mapping should be as follow:

| Authorization Parameter | Service Parameter |
|---|---|
| serviceA_Param2 | serviceB_Parameter1 |
| serviceA_Param3 | serviceB_Parameter2 |

You can use the *username* and *password* parameters as authParam.

There are three kinds of authorization policies:

# Executable Authorization Policy

This policy use an executable service for authorization. The output of the executable is checked with one of the verification methods:

## Text Verification

The output is verified with constant predefined string. Authorication succeed if the output is same as predefined string.

If the executable prints "OK" for correct parameters, you can use "OK" with text verification.

## Parameter Verification

The output is verified with one of the request parameters. Authorication succeed if the output is same as the value of the specified parameter.

## 'Not Equals' Verification

The output of the executable service is checked if it is not equals a predefined text value.

If the executable prints "**NOK**" for incorrect parameters and print a generated ID for correct parameters (unpredictable value), you can use "**NOK**" with 'Not Equals' verification.

## Numeric Expression Verification

The check output of the executable service is checked with a numeric expression. Numeric expression may be *smaller than* or *greater than*.

If the executable prints a code number between 200 - 300 for incorrect parameters and print a code number greater than 300 for correct parameters (unpredictable value), you can use "**300**" with ***greater than*** numeric expression verification.