# CSAP

## COMMON SECURITY ARCHITECTURE *for* PRODUCTION

VERSION 1.3

# PART 2:
# INTERFACES

Motion Picture Laboratories, Inc. August 1, 2023

# Contents

# 1 Introduction

This document is Part 2 of the group of documents that describe our security architecture. Part 1 is the overall architecture description, and familiarity with that document is necessary to understand this document.

This document illustrates the interfaces between:

- Core components and supporting components
- Core components and production management
- Between core components

It uses a canonical form as a means of explanation; however, this is not intended to be a specification for APIs.

**Changes from CSAP Part 1 v1.1**

- The name of the *authorization policies* has been changed to *authorization rules*.
- The functions of the policy manager moved into the authorization service, the policy service in v1.0 now consists only of the Authorization Rules Distribution Service (ARDS), formerly called the policy engine. This does not change the functions necessary to create an authorization policy, but consolidation simplifies this part of the architecture.

**Changes from CSAP Part 1 v1.2**

- The functions of the Asset Protection Service have been merged into the authorization service. There is no change in function. Descriptions of the interface between the asset protection service and the authorization service have been removed since they are internal to the authorization service.
- The CSAP supporting security functions Trust Inference and Continuous Trust Validation have been merged to reflect the direction of the market place.
- The diagrammatic representation is now three services (authorization, authentication and the ARDS) as the part of the CSAP infrastructure.

# 2 Component Interfaces

Interactions between components are classified according to four types.

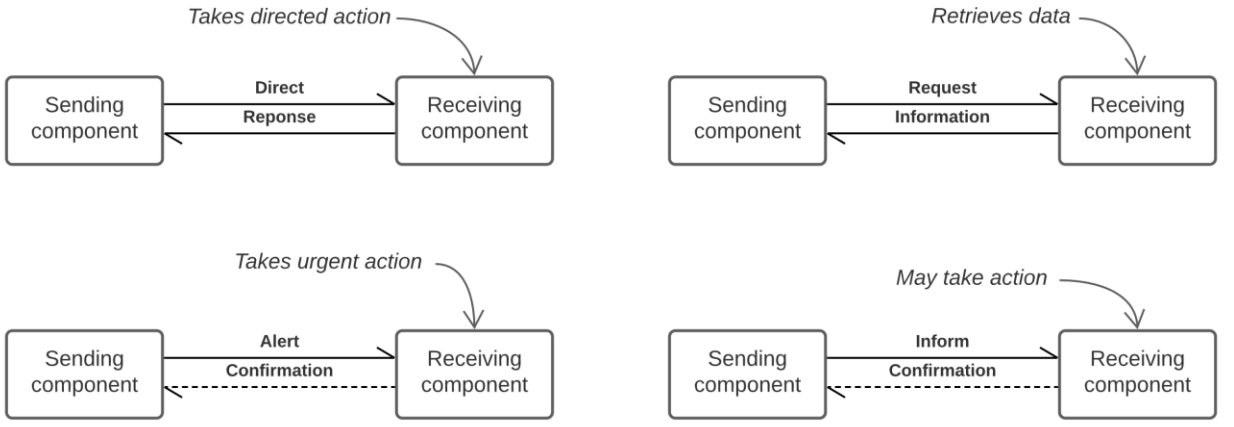


*Figure 2-1 Types of interactions between components*

1. *Directs* – a message is sent from one component to another, and the receiving component is required to act on the message.
2. *Requests* – a message is sent from one component to another requesting information, e.g., the requester is registering with a service to receive alerts.
3. *Alerts* – an urgent notification, likely an asynchronous message, is sent from one component to another notifying the receiving component of a material change of state. The expectation is that the receiving component will act on the alert.
4. *Informs* – a notification, likely an asynchronous message, is sent from one component to another notifying the receiving component of current state or change in state.

*Alerts* are distinguished from *informs* since a different mechanism might be used to deliver urgent messages. However, the mechanism for transmitting messages is beyond the scope of this architecture.

In the rest of the document, the interactions of components are listed in the format:

***Component A*** interaction ***component B***

- Parameters: <a list of parameters sent with the interaction>
- Returns: <a list of values returned to the sending component>

Lists of parameters and returned values shown in this document are not necessarily complete.

## 2.1 Parameters

The parameters used in the architecture are:

*Identifier*: the value used to identify the entity (something that is taking part in the workflow: e.g., asset, resource, human).

*Credentials*: data providing evidence for claims about the identity.

*Contextual Attributes*: data describing characteristics of the contextual attributes of an authentication request. That might include:

- IP address
- Geolocation
- System
- Previous times the entity has been authenticated
- Production

*Trust Score*: the trust score is a number assigned by the trust inference and continuous trust validation indicating the level of trust appropriate for an artifact. We might use a trust score that is in the range of 1 to 100 where:

- A trust score of 100 represents the highest confidence that an entity is the trusted entity it claims to be, if continuous trust validation confirms that score then the lifetime of a previous authentication can be extended.
- A trust score of 0 means the entity is not, or must not be regarded as being, the trusted entity it claims to be.

*Access Token*: an access token contains the security credentials of an authenticated entity.

*Permissions List*: a set of permissions that control the ability of an entity to read, write, change, and execute an asset or application.

*Security Status Request*: a security status is requested for a listed set of artifacts.

*Security Status*: the security status of the artifacts in the security status request.

*Alert Code*: a context specific code describing the reason for the alert.

## 2.2 Document Organization

The document is organized around the core and supporting security components. Interfaces are listed under the initiating component.

## 2.3 Abbreviations

***ARDS*** **-** Authorization rules distribution service.

***PEP*** **-** Policy enforcement point.

# 3 Supporting Security Component Interfaces

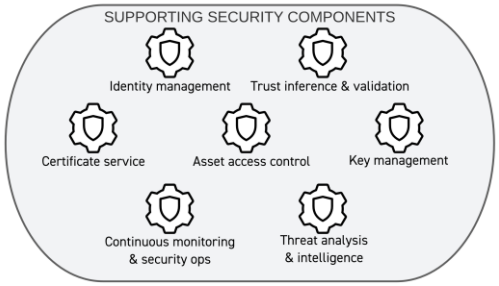


*Figure 2 Supporting security components*

## 3.1 Identity Management

***Identity management*** alerts ***authentication service***

- Parameters: identifier, reason code

If the identity management is an IAM system:

***Identity management*** notifies ***authorization service***

- Parameters: identifier, permissions list or NULL

## 3.2 Trust Inference

***Trust inference*** informs ***authentication service***

- Parameters: identifier, trust score

***Trust inference*** alerts ***authentication service***

- Parameters: identifier, trust score

***Trust inference*** informs ***authentication service***

- Parameters: identifier, trust score

## 3.3 Certificate Service

***Certificate service*** alerts ***authentication service***

- Parameters: Authentication certificate, REVOKED

***Certificate service*** alerts ***authorization service***

- Parameters: Authentication certificate, REVOKED

## 3.4 Continuous Monitoring and Security Operations

***Continuous monitoring and security operations*** alerts ***authorization service***

- Parameters: set of resources, security status of resources

***Continuous monitoring and security operations*** informs ***authorization service***

- Parameters: set of resources, security status of resources

***Continuous monitoring and security operations*** alerts ***authentication service***

- Parameters: identifier, security status

***Continuous monitoring and security operations*** informs ***authentication service***

- Parameters: identifier, security status

***Continuous monitoring and security operations*** alerts ***authorization service***

- Parameters: asset identifier, asset location, security status

## 3.5 Threat Analysis and Intelligence

***Threat analysis and intelligence*** informs ***authorization service***

- Parameters: set of resources, security status of resources

***Threat analysis and intelligence*** alerts ***authorization service***

- Parameters: set of resources, security status of resources

# 4 Core Security Component Interfaces

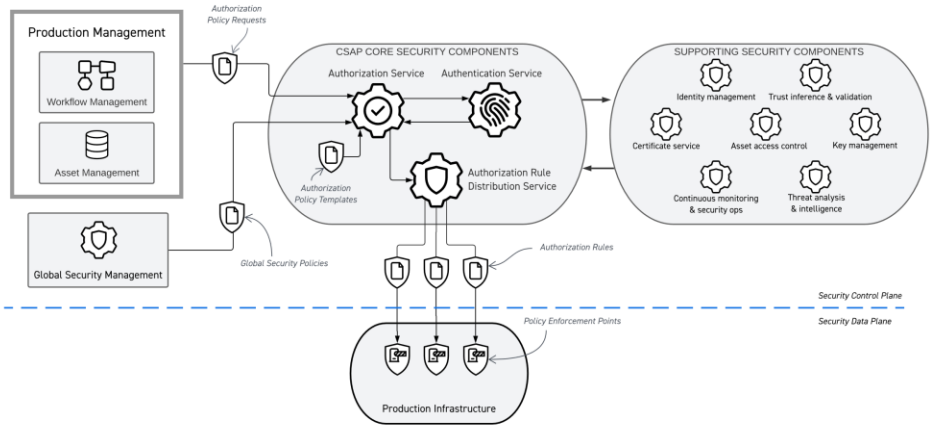


*Figure 3 Detail of core components*

## 4.1 Authentication Service

***Authentication service*** directs ***identity management***

- Parameters: identifier, credentials, context attributes
- Returns: access token, FAIL

***Authentication service*** requests ***trust inference***

- Parameters: identifier
- Returns: trust score

***Authentication service*** requests ***continuous trust validation***

- Parameters: identifier, trust score
- Returns: trust score

***Authentication service*** directs ***certificate***

- Parameters: identifier, public key
- Returns: authentication certificate

***Authentication service*** alerts ***authorization***

- Parameters: identifier, alert code

***Authentication service*** alerts ***production management***

- Parameters: identifier, alert code

***Authentication service*** alerts ***ARDS***

- Parameters: identifier, alert code

The alert code describes the reason for the alert, such as an authentication policy has been revoked.

## 4.2 Authorization Service

***Authorization service*** directs ***authentication service***

- Parameters: identifier
- Returns: access token, FALSE

***Authorization service*** directs ***ARDS***

- Parameters: authorization rules

***Authorization service*** alerts ***ARDS***

- Parameters: authorization rules, change reason code

***Authorization service*** alerts ***production management***

- Parameters: identifier list, asset handle list, alert code

***Authentication service*** requests ***continuous monitoring and security operations***

- Parameters: set of resources
- Returns: security status of resources

***Authentication service*** requests ***threat analysis and intelligence***

- Parameters: set of resources

If the identity management is an IAM system:

***Authorization service*** requests ***identity management***

- Parameters: identifier
- Returns: permissions list, NULL

***Authorization service*** directs ***certificate service***

- Parameters: identifier, public key
- Returns: certificate

***Authorization service*** directs ***policy enforcement point***

- Parameters: identifier, access permissions, encryption keys
- Returns: ACK, ERROR

## 4.3 ARDS

***ARDS*** requests ***authorization service***

- Parameters: asset handle, asset location

- Returns: SUCCESS, FAIL, error code

***ARDS*** directs ***policy enforcement point***

- Parameters: authorization rule(s)
- Returns: security status of resources

## 4.4 Policy Enforcement Point

***Policy enforcement point*** alerts ***ARDS***

- Parameters: exception

# 5 Production Management Interfaces

***Workflow management*** directs ***authentication service***

- Parameters: Identifier
- Returns: ACK, ERROR

***Workflow Management*** directs ***authorization service***

- Parameters: Resource list, participant list, asset list
- Returns: ACK, ERROR

***Asset management*** directs ***authorization service***

- Parameters: Asset handle, asset location
- Returns: ACK, ERROR