



**MOVIELABS  
SECURITY ARCHITECTURE  
PART 2:  
INTERFACES  
VERSION 1.0**



## Contents

1	Introduction .....	1
2	Component Interfaces .....	1
2.1	Parameters .....	2
2.2	Document Organization .....	3
3	Supporting Security Component Interfaces.....	3
3.1	Identity Management.....	3
3.2	Trust Inference .....	4
3.3	Continuous Trust Validation .....	4
3.4	Certificate Service.....	4
3.4.1	Interface Parameters.....	4
3.5	Continuous Monitoring and Security Operations.....	4
3.6	Threat Analysis and Intelligence .....	5
4	Core Security Component Interfaces.....	5
4.1	Authentication Service .....	5
4.2	Authorization Service .....	6
4.3	Asset Protection Service .....	7
4.4	Policy Service .....	7
4.5	Policy Enforcement Point .....	8
5	Production Management Interfaces.....	8

© 2021 Motion Picture Laboratories, Inc.

This document is intended as a guide for companies developing or implementing products, solutions, or services for the future of media creation. No effort is made by Motion Picture Laboratories, Inc. to obligate any market participant to adhere to the recommendations in this document. Whether to adopt these recommendations in whole or in part is left to the discretion of individual market participants, using independent business judgment. Each MovieLabs member company shall decide independently the extent to which it will utilize, or require adherence to, these recommendations. All questions on member company adoption or implementation must be directed independently to each member company.

# 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.

# 2 Component Interfaces

Interactions between components are classified according to four types.

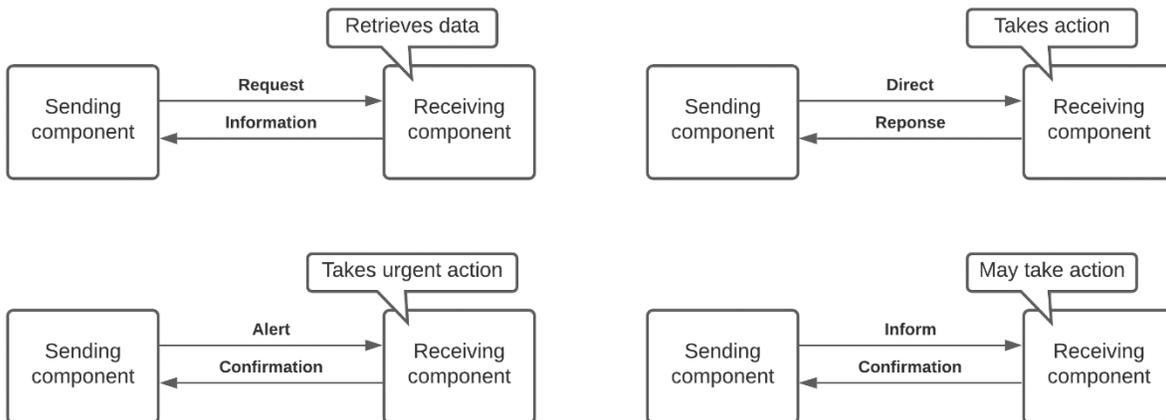


Figure 2-1 Types of interactions between components

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

*Trust Score*: the trust score is defined to be in the range -1 to 1.

- A trust score of 1 means that the lifetime of a previous authentication can be extended.
- A trust score of 0 means that full authentication should be conducted.
- A trust score of -1 means that even if the credentials presented are correct, there are circumstances surrounding the request that warrant the request be rejected or further authentication mechanisms be invoked.

*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.

## 3 Supporting Security Component Interfaces

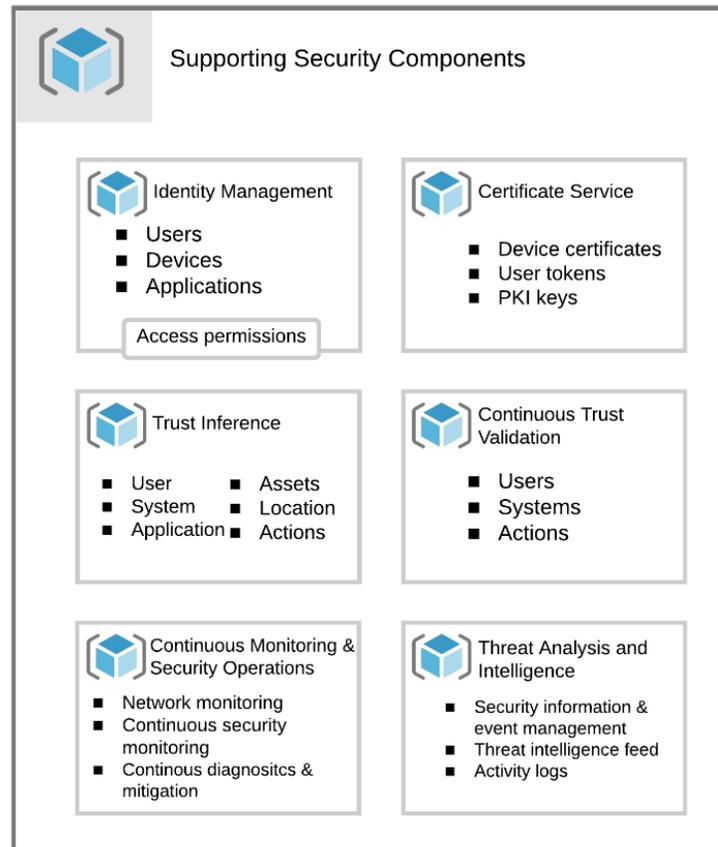


Figure 3-1 Detailed View of supporting components

### 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

## 3.3 Continuous Trust Validation

***Continuous trust validation*** alerts ***authentication service***

- Parameters: identifier, trust score

***Continuous trust validation*** informs ***authentication service***

- Parameters: identifier, trust score

## 3.4 Certificate Service

### 3.4.1 Interface Parameters

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

- Parameters: Authentication certificate, REVOKED

***Certificate service*** alerts ***asset protection service***

- Parameters: Authentication certificate, REVOKED

## 3.5 Continuous Monitoring and Security Operations

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

- Parameters: set of resources, security status of resources

***Continuous monitoring and security operations*** informs ***policy 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 ***asset protection service***

- Parameters: asset identifier, asset location, security status

### 3.6 Threat Analysis and Intelligence

**Threat analysis and intelligence** informs **policy service**

- Parameters: set of resources, security status of resources

**Threat analysis and intelligence** alerts **policy service**

- Parameters: set of resources, security status of resources

## 4 Core Security Component Interfaces

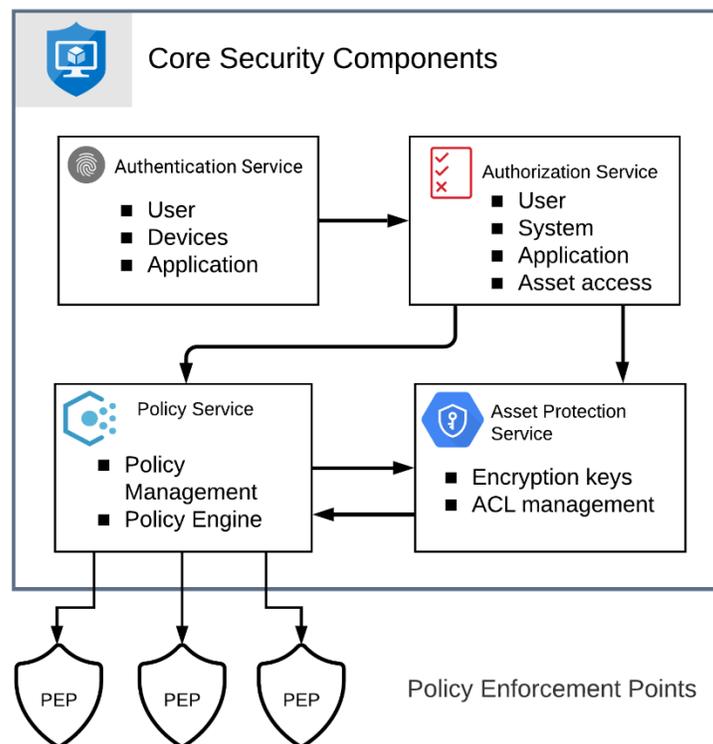


Figure 4-1 Detail of core components

### 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 *policy service*

- Parameters: identifier, alert code

The alert code describes the reason for the alert, such as a user account has been closed.

## 4.2 Authorization Service

**Authorization service** directs *authentication service*

- Parameters: identifier
- Returns: access token, FALSE

**Authorization service** directs *policy service*

- Parameters: static policy, dynamic policy

**Authorization service** alerts *policy service*

- Parameters: static policy, dynamic policy, change reason code

**Authorization service** directs *asset protection service*

- Parameters: asset handle
- Returns: SUCCESS, FAIL, error code

**Authorization service** alerts *production management*

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

If the identity management is an IAM system:

**Authorization service** requests *identity management*

- Parameters: identifier
- Returns: permissions list, NULL

### 4.3 Asset Protection Service

The asset protection service is supplied with:

***Asset protection service*** requests ***authentication service***

- Parameters: identifier
- Returns: TRUE, FALSE

***Asset protection service*** directs ***certificate service***

- Parameters: identifier, public key
- Returns: certificate

***Asset protection service*** directs ***policy enforcement point***

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

### 4.4 Policy Service

***Policy service*** directs ***authentication service***

- Parameters: identifier
- Returns: TRUE, FALSE

***Policy service*** directs ***asset protection service***

- Parameters: asset handle, asset location
- Returns: SUCCESS, FAIL, error code

***Policy service*** directs ***policy enforcement point***

- Parameters: combined policy

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

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

***Policy service*** requests ***threat analysis and intelligence***

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



## 4.5 Policy Enforcement Point

***Policy enforcement point*** alerts ***policy service***

- Parameters: exception

***Policy enforcement point*** alerts ***production management***

- Parameters: exception

***Policy enforcement point*** alerts ***continuous monitoring and security operations***

- Parameters: alert code

## 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

***Workflow management*** directs ***policy enforcement point***

- Parameters: Resource list, participant list, asset list
- Returns: SUCCEED, FAIL

***Asset management*** directs ***asset protection service***

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