The University of Kansas

Information and Telecommunication Technology Center

Technical Report

# ACE Project ACE Authorization Interface Specification Version 0.1

James Mauro, Leon Searl, and Gary Minden

ITTC-FY2001-TR-23150-08

June 2001

Project Sponsor:
U.S. Air Force and the Defense Advanced Research Projects Agency under contract no. F30602-00-2-0581 and The National Science Foundation under grant EIA-9972843

# 1 Introduction

The following document specifies the interfaces to communicate with a piece of software and to communicate with services in order to get authorization and permission to access a service. The methods and commands specify the communications of certificates and the communication of access requests to the server.

This document extends from the ACE Authorization Requirements and the ACE Security Requirements. The specifications within this document implement those commands as needed.

The following document describes whom the software affects in the problem domain, and then gives specifications in the following four categories: Query Interface, Behavioral Interface, Mapping Interface, and Software Interfaces.

## *1.1 Notation*

The names of documents are in *italics*.

Each individual specification in the Specification Document has an identifier. The identifier is of the form **'S-X.X.X'** where X.X.X is a dotted number. The number shall consist of the section number the specification is in plus a dotted number. Specification identifiers are used in Design Documents and source code to refer to specifications in the Specifications Document. . An example of a specification follows:

**S-2.1.3 Program Name Specification**

The name of the control program shall be 'bamboozel'. The program takes no arguments.

# 2 Specifications

The following sections specify the parts of the Authorization information. They include the Problem Domain, which specifies the actors, the Query Interface Specifications, which specify the information, which can be given by the software, the Behavioral Rule which specifies the actions the software takes on actors within the environment, the Mapping Specification which specify the mapping from the input to the output, and the Operations specifications which specify which commands can be used to command the software.

## *2.1 Problem Domain*

The ACE Authorization Interface allows for any user who wishes to communicate with a service within the ACE to determine access levels and access permissions to a device. These permission levels are constant and defined within each ACE device. All ACE services must verify the user's permission levels and user's identity before granting access to any services.

The services specified within this document specify those interfaces used over the ACE

Command Connection channels in order to communicate security information. This includes protocol information and ACE Commands that represent protocol information that the clients and services must implement.

## 2.2 Query Interface Specifications

The following commands represent information that can be requested from a service in order to gain information about the current permission levels, access to functions, etc.

### S-2.2.1 Permission Levels

The following string represent the possible permission levels as specified by R-XXX.

`FULL_ACCESS`: All access is granted to the current system, for any service with only one reader, this access overrides all other readers. This level implies `INFORMATIONAL_ACCESS` and `QUERY_ACCESS`.

`INFORMATIONAL_ACCESS`: Access is granted to any command that provides information about the device, but cannot change the internal state of the device. This level implies `QUERY_ACCESS`.

`FULL_ACCESS_NO_OVERRIDE`: Access is granted to change the state of the device, but access is denied if another user has control of the device at the specified time. Implies `INFORMATIONAL_ACCESS` and `QUERY_ACCESS`.

`QUERY_ACCESS`: Access is given to request non-specific commands that every service must implement. No access to service specific commands is granted.

`NO_ACCESS`: No permission is given to run any command. The service must then terminate the connection.

`UNKNOWN_ACCESS`: Current permissions level is unknown and must be determined. Access to any functions of the service is denied, but the connection is not terminated until the access level is determined. Only commands that help discover current permission levels are available to be accesses.

### S-2.2.2 Current Permission Level

The current level of permission a user has to access a service is determined using the getCurrentPermissionLevel function. This function will return a representation of the permission levels defined in S-2.2.1. Both the command interface method and the network command are available to all users, regardless of their connection levels.

```
public int getCurrentPermissionLevel();
```

*Arguments:* None

*Returns:* An integer that represents the current permission level. The defined values for the integer are specified in S-2.2.1.

```
ServiceGetCurrentPermissionLevel;
```

*Arguments:* None

*Returns:* None

```
ServiceGetCurrentPermissionLevelResult level="my_level";
```

*Arguments:*

**level** – The current level or permissions, as specified in a string. The possible values for the string and their meanings are specified in S-2.2.1

| | |
|---|---|
| Argument Occurrences | : CMDVAL_OCCURS_ONCE |
| Value Data Type | : CMDVAL_STRING |
| Value Extent | : See S-2.2.1 |
| Mutually Dependent Arguments | : None |
| Mutually Exclusive Arguments | : None |

**status** – Specifies the status, success or fail of the command.

| | |
|---|---|
| Argument Occurrences | : CMDVAL_OCCURS_ONCE |
| Value Data Type | : CMDVAL_STRING |
| Value Extent | : {Success, Fail} |
| Value Unit | : N/A |
| Mutually Dependent Arguments | : None |
| Mutually Exclusive Arguments | : None |

**msg** – A message about the error that has occurred. It is more specific that an error number.

| | |
|---|---|
| Argument Occurrences | : CMDVAL_OCCURS_ONCE |
| Value Data Type | : CMDVAL_STRING |
| Value Extent | : CMDVALEXTENT_ANY |
| Value Unit | : N/A |
| Mutually Dependent Arguments | : errorno, status |
| Mutually Exclusive Arguments | : None |

**errorno** – An error number for a failure of the command.

| | |
|---|---|
| Argument Occurrences | : CMDVAL_OCCURS_ONCE |
| Value Data Type | : CMDVAL_STRING |
| Value Extent | : CMDVALEXTENT_ANY |
| Value Unit | : N/A |
| Mutually Dependent Arguments | : errorno, status |
| Mutually Exclusive Arguments | : None |

### S-2.2.3 Accessible Commands

Accessible commands and methods retrieve those commands that can be executed using the current permission level.  The method returns an array of the commands that can be executed. These commands can be executed without permission errors within a limited time frame.  If the current conditions change before the command can be executed.

```
public String[] getAvailableCommands();
```

*Arguments:* None

*Returns:* An array of strings.  These string represent the commands that there is currently permission to run.  Overloaded methods that have different permission levels are not separated.  If permission to one of the overloaded methods is given, then the method is listed in the array.

```
ServiceGetCurrentAccessibleCommands;
```

*Arguments:* None

*Returns:* None

```
ServiceGetCurrentAccessibleCommandsResult commands={my_cmd, my_cmd2,. . .}
```

*Arguments:*

**level** – A list of the commands that the current user has permission to run.  Any command that the user have permission to run is contained within this array.

| | |
|---|---|
| Argument Occurrences | : CMDVAL_OCCURS_ONCE |
| Value Data Type | : CMDVAL_STRING_VECTOR |
| Value Extent | : Any string. |
| Mutually Dependent Arguments | : None |
| Mutually Exclusive Arguments | : None |

**status** – Specifies the status, success or fail of the command.

| | |
|---|---|
| Argument Occurrences | : CMDVAL_OCCURS_ONCE |
| Value Data Type | : CMDVAL_STRING |
| Value Extent | : {Success, Fail} |
| Value Unit | : N/A |
| Mutually Dependent Arguments | : None |
| Mutually Exclusive Arguments | : None |

**msg** – A message about the error that has occurred.  It is more specific that an error number.

| | |
|---|---|
| Argument Occurrences | : CMDVAL_OCCURS_ONCE |
| Value Data Type | : CMDVAL_STRING |
| Value Extent | : CMDVALEXTENT_ANY |
| Value Unit | : N/A |
| Mutually Dependent Arguments | : errorno, status |
| Mutually Exclusive Arguments | : None |

**errorno** – An error number for a failure of the command.

| | |
|---|---|
| Argument Occurrences | : CMDVAL_OCCURS_ONCE |
| Value Data Type | : CMDVAL_STRING |
| Value Extent | : CMDVALEXTENT_ANY |
| Value Unit | : N/A |
| Mutually Dependent Arguments | : errorno, status |
| Mutually Exclusive Arguments | : None |

## S-2.2.4 Request for the public key of the service

A public key can be requested from a service using the method or a command. This public key represents a key that can be used to create an assertion later. The format of the key is implementation specific.

```
public String getPublicKey();
```

*Arguments:* None

*Returns:* Returns a properly formatted public key that can be used to create other assertions. The format of the key is implementation specific.

```
ServiceGetCurrentPublicKey;
```

*Arguments:* None

*Returns:* None

```
ServiceGetCurrentPublicKeyResult key=”MyKey”;
```

*Arguments:*

**key – The requested key formatted properly.**

| | |
|---|---|
| Argument Occurrences | : CMDVAL_OCCURS_ONCE |
| Value Data Type | : CMDVAL_STRING |
| Value Extent | : Any string. |
| Mutually Dependent Arguments | : None |
| Mutually Exclusive Arguments | : None |

**status** – Specifies the status, success or fail of the command.

| | |
|---|---|
| Argument Occurrences | : CMDVAL_OCCURS_ONCE |
| Value Data Type | : CMDVAL_STRING |
| Value Extent | : {Success, Fail} |
| Value Unit | : N/A |
| Mutually Dependent Arguments | : None |
| Mutually Exclusive Arguments | : None |

**msg** – A message about the error that has occurred. It is more specific that an error number.

| | |
|---|---|
| Argument Occurrences | : CMDVAL_OCCURS_ONCE |
| Value Data Type | : CMDVAL_STRING |
| Value Extent | : CMDVALEXTENT_ANY |
| Value Unit | : N/A |
| Mutually Dependent Arguments | : errorno, status |
| Mutually Exclusive Arguments | : None |

**errorno** – An error number for a failure of the command.

| | |
|---|---|
| Argument Occurrences | : CMDVAL_OCCURS_ONCE |
| Value Data Type | : CMDVAL_STRING |
| Value Extent | : CMDVALEXTENT_ANY |
| Value Unit | : N/A |
| Mutually Dependent Arguments | : errorno, status |
| Mutually Exclusive Arguments | : None |

## *2.3 Behavioral Rule Interface Specifications*

### S-2.3.1 Request for Credentials

A service or a user can request for credentials from each other. The returned credentials must be in the proper format. The set of credentials must be able to be entered into keynote without extra trouble.

```
public Assertions[] requestCrendentials();
```

*Arguments:* None

*Returns:* Returns a set of credentials that can be passed to the authorization mechanisms. The actual implementation of the Assertion objects is implementation specific.

```
ServiceRequestCredentials;
```

*Arguments:* None

*Returns:* None

```
ServiceRequestCredentialsResult assertions={ "Assert", "Assert", … };
```

*Arguments:*

**assertions** – A properly formatted assertion

| | |
|---|---|
| Argument Occurrences | : CMDVAL_OCCURS_ONCE |
| Value Data Type | : CMDVAL_STRING_VECTOR |
| Value Extent | : Any string. |
| Mutually Dependent Arguments | : None |
| Mutually Exclusive Arguments | : None |

**status** – Specifies the status, success or fail of the command.

| | |
|---|---|
| Argument Occurrences | : CMDVAL_OCCURS_ONCE |
| Value Data Type | : CMDVAL_STRING |
| Value Extent | : {Success, Fail} |
| Value Unit | : N/A |
| Mutually Dependent Arguments | : None |
| Mutually Exclusive Arguments | : None |

**msg** – A message about the error that has occurred. It is more specific that an error number.

| | |
|---|---|
| Argument Occurrences | : CMDVAL_OCCURS_ONCE |
| Value Data Type | : CMDVAL_STRING |
| Value Extent | : CMDVALEXTENT_ANY |
| Value Unit | : N/A |
| Mutually Dependent Arguments | : errorno, status |
| Mutually Exclusive Arguments | : None |

**errorno** – An error number for a failure of the command.

| | |
|---|---|
| Argument Occurrences | : CMDVAL_OCCURS_ONCE |
| Value Data Type | : CMDVAL_STRING |
| Value Extent | : CMDVALEXTENT_ANY |
| Value Unit | : N/A |
| Mutually Dependent Arguments | : errorno, status |
| Mutually Exclusive Arguments | : None |

### S-2.3.2 Passing Credentials without a Request

A credential can be passed to the user without a request from the authorizing server. This request is processed and any added credentials are added to the current keynote session. The permissions of the session are then re-evaluated.

public void supplyCredentiatials( Assertions MyAssertions[]);

*Arguments:*

**MyAssertions** – An array of assertions to pass to the authorizing service.

*Returns:* None.

```
ServiceRequestCredentials assertions={ "Assert", "Assert", … };
```

*Arguments:*

**assertions –** A properly formatted assertion

| | |
|---|---|
| Argument Occurrences | : CMDVAL_OCCURS_ONCE |
| Value Data Type | : CMDVAL_STRING_VECTOR |
| Value Extent | : Any string. |
| Mutually Dependent Arguments | : None |
| Mutually Exclusive Arguments | : None |

*Returns:* None

`ServiceRequestCredentialsResult;`

*Arguments:*

**status** – Specifies the status, success or fail of the command.

| | |
|---|---|
| Argument Occurrences | : CMDVAL_OCCURS_ONCE |
| Value Data Type | : CMDVAL_STRING |
| Value Extent | : {Success, Fail} |
| Value Unit | : N/A |
| Mutually Dependent Arguments | : None |
| Mutually Exclusive Arguments | : None |

**msg** – A message about the error that has occurred.  It is more specific that an error number.

| | |
|---|---|
| Argument Occurrences | : CMDVAL_OCCURS_ONCE |
| Value Data Type | : CMDVAL_STRING |
| Value Extent | : CMDVALEXTENT_ANY |
| Value Unit | : N/A |
| Mutually Dependent Arguments | : errorno, status |
| Mutually Exclusive Arguments | : None |

**errorno** – An error number for a failure of the command.

| | |
|---|---|
| Argument Occurrences | : CMDVAL_OCCURS_ONCE |
| Value Data Type | : CMDVAL_STRING |
| Value Extent | : CMDVALEXTENT_ANY |
| Value Unit | : N/A |
| Mutually Dependent Arguments | : errorno, status |
| Mutually Exclusive Arguments | : None |

### *2.4 Mapping Interface Specifications*

None

### *2.5 Software Operation Specifications*

None

## 3 Design Constraints

None

## 4 Deprecated Specifications

None

## 5 Glossary

None

## 6 Change Log

| Version | Date | Changes |
|---|---|---|
| 0.1 | <date> | Initial Working Version |
| 1.0 | <date> | Initial Released Version |

## 7 Notes

-