Authentication Step-Up Protocol and Metadata Version 1.0

Working Draft 04b

04 September 2015

Technical Committee:

[OASIS Electronic Identity Credential Trust Elevation Methods (Trust Elevation) TC](https://www.oasis-open.org/committees/trust-el/)

Chairs:

Abbie Barbir (barbira@aetna.com), Aetna

Don Thibeau ([don@openidentityexchange.org](mailto:don@openidentityexchange.org)), [Open Identity Exchange](http://www.openidentityexchange.org/)

Editors:

Andrew Hughes ([AndrewHughes3000@gmail.com](mailto:AndrewHughes3000@gmail.com)), Individual

Peter Alterman (palterman@safe-biopharma.org), [SAFE-BioPharma Assn](http://www.safe-biopharma.org/).

Shaheen Abdul Jabbar ([shaheen.abduljabbar@jpmchase.com](mailto:shaheen.abduljabbar@jpmchase.com)), [JPMorgan Chase Bank, N.A.](http://www.jpmorganchase.com/)

Abbie Barbir (barbira@aetna.com), Aetna

Mary Ruddy ([mary@meristic.com](mailto:mary@meristic.com)), [Identity Commons](http://www.identitycommons.net/)

Additional artifacts:

* None

Related work:

This specification replaces or supersedes:

* Specifications replaced by this specification (hyperlink, if available)

This specification is related to:

* Related specifications (hyperlink, if available)

Declared XML namespaces:

* None

Abstract:

Summary of the technical purpose of the document.

Status:

This [Working Draft](https://www.oasis-open.org/policies-guidelines/tc-process#dWorkingDraft) (WD) has been produced by one or more TC Members; it has not yet been voted on by the TC or [approved](https://www.oasis-open.org/policies-guidelines/tc-process#committeeDraft) as a Committee Draft (Committee Specification Draft or a Committee Note Draft). The OASIS document [Approval Process](https://www.oasis-open.org/policies-guidelines/tc-process#standApprovProcess) begins officially with a TC vote to approve a WD as a Committee Draft. A TC may approve a Working Draft, revise it, and re-approve it any number of times as a Committee Draft.

URI patterns:

Initial publication URI:  
http://docs.oasis-open.org/trust-el/trust-el-protocol/v1.0/csd01/trust-el-protocol-v1.0-csd01.doc

Permanent “Latest version” URI:  
http://docs.oasis-open.org/trust-el/trust-el-protocol/v1.0/trust-el-protocol-v1.0.doc

(Managed by OASIS TC Administration; please don’t modify.)

Copyright © OASIS Open 2015. All Rights Reserved.

All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual Property Rights Policy (the "OASIS IPR Policy"). The full [Policy](https://www.oasis-open.org/policies-guidelines/ipr) may be found at the OASIS website.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published, and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright notice or references to OASIS, except as needed for the purpose of developing any document or deliverable produced by an OASIS Technical Committee (in which case the rules applicable to copyrights, as set forth in the OASIS IPR Policy, must be followed) or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Table of Contents

TOC \o 2-2 \t "AppendixHeading1, 3,Heading, 4,Heading 1 WP, 5"

1. Introduction PAGEREF \_Toc \h 6
   1. Terminology PAGEREF \_Toc1 \h 6
   2. Normative References PAGEREF \_Toc2 \h 6
   3. Non-Normative References PAGEREF \_Toc3 \h 6
2. Landscape and Context PAGEREF \_Toc4 \h 8
   1. Goals of the Fourth Deliverable PAGEREF \_Toc5 \h 8
   2. Trust System Context PAGEREF \_Toc6 \h 8
   3. Assumptions for Trust Elevation Approaches PAGEREF \_Toc7 \h 8
3. Conceptual Models PAGEREF \_Toc8 \h 9
   1. Trust Elevation Core Model PAGEREF \_Toc9 \h 9
   2. Trust Elevation Concepts PAGEREF \_Toc10 \h 9
   3. Use of Authorization Architectures and Models PAGEREF \_Toc11 \h 10
4. Architecture & Design Considerations PAGEREF \_Toc12 \h 14
   1. Architecture & Design Factors PAGEREF \_Toc13 \h 14
   2. Trust Elevation Architecture Components PAGEREF \_Toc14 \h 15
   3. Other Related Architecture Components PAGEREF \_Toc15 \h 16
5. Implementation Considerations PAGEREF \_Toc16 \h 18
   1. Orchestration PAGEREF \_Toc17 \h 18
   2. Protocol Capabilities PAGEREF \_Toc18 \h 18
   3. Assignment of Roles and Responsibilities PAGEREF \_Toc19 \h 18
   4. Enumeration of Authentication Methods PAGEREF \_Toc20 \h 18
   5. User Enrolment PAGEREF \_Toc21 \h 18
   6. Risk Engine Integration PAGEREF \_Toc22 \h 19
6. Trust Elevation Sequence Example PAGEREF \_Toc23 \h 20
   1. Use Case: Online banking transactions PAGEREF \_Toc24 \h 20
7. Metadata and Assertions PAGEREF \_Toc25 \h 23
   1. LoA Assessment (LA) PAGEREF \_Toc26 \h 23
   2. Method Determination (MD) PAGEREF \_Toc27 \h 23
8. Use cases PAGEREF \_Toc28 \h 25
   1. Trust-El Use Case 1 PAGEREF \_Toc29 \h 25
   2. Trust-El Use Case 2 PAGEREF \_Toc30 \h 25
   3. Trust-El Use Case 3 PAGEREF \_Toc31 \h 26
9. # Conformance PAGEREF \_Toc32 \h 27
10. Acknowledgments PAGEREF \_Toc33 \h 28
11. State Models for Assurance Level Evaluation PAGEREF \_Toc34 \h 29
    1. Evaluation of Assurance Requirements at Transaction Time PAGEREF \_Toc35 \h 29
12. Revision History PAGEREF \_Toc36 \h 33

# Introduction

[All text is normative unless otherwise labeled]

## Terminology

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in **[RFC2119]**.

## Normative References

**[RFC2119]** Bradner, S., “Key words for use in RFCs to Indicate Requirement Levels”, BCP 14, RFC 2119, March 1997. <http://www.ietf.org/rfc/rfc2119.txt>.

ITU-T X.1252

ITU-T X.1254

ITU-T X.1255

ISO 29115

NIST SP 800-63-2

NIST SP 800-162 ABAC

OAuth

UMA

OpenID Connect

Other TC Deliverables

[XACML3] OASIS Standard, eXtensible Access Control Markup Language (XACML) Version 3.0, 22 January 2013. <http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-en.doc>

[SAML2] OASIS Standard, Security Assertion Markup Language (SAML) Version 2.0, 2 December 2009. <https://www.oasis-open.org/committees/download.php/35711/sstc-saml-core-errata-2.0-wd-06-diff.pdf>

[SAMLAC] OASIS Standard, Authentication Context for the OASIS Security Assertion Markup Language (SAML) Version 2.0, 15 March 2005. <https://docs.oasis-open.org/security/saml/v2.0/saml-authn-context-2.0-os.pdf>

## Non-Normative References

[IDMgmt] <https://www.idmanagement.gov/sites/default/files/documents/BAE_v2_SAML2_Profile_Final_v1.0.0.pdf>

**NOTE: The proper format for citation of technical work produced by an OASIS TC (whether Standards Track or Non-Standards Track) is:**

**[Citation Label]**

Work Product [title](http://docs.oasis-open.org/specGuidelines/ndr/namingDirectives.html#workProductName) (italicized). Edited by Albert Alston, Bob Ballston, and Calvin Carlson. Approval date (DD Month YYYY). OASIS [Stage](http://docs.oasis-open.org/specGuidelines/ndr/namingDirectives.html#stage) Identifier and [Revision](http://docs.oasis-open.org/specGuidelines/ndr/namingDirectives.html#revision) Number (*e.g.*, OASIS Committee Specification Draft 01). Principal URI ([version-specific URI](http://docs.oasis-open.org/specGuidelines/ndr/namingDirectives.html#this-version), *e.g*., with stage component: somespec-v1.0-csd01.html). Latest version: ([latest version URI](http://docs.oasis-open.org/specGuidelines/ndr/namingDirectives.html#latest-version), without stage identifiers).

For example:

**[OpenDoc-1.2]** *Open Document Format for Office Applications (OpenDocument) Version 1.2*. Edited by Patrick Durusau and Michael Brauer. 19 January 2011. OASIS Committee Specification Draft 07. <http://docs.oasis-open.org/office/v1.2/csd07/OpenDocument-v1.2-csd07.html>. Latest version: <http://docs.oasis-open.org/office/v1.2/OpenDocument-v1.2.html>.

# Landscape and Context

This document, the fourth deliverable of the OASIS Trust Elevation Technical Committee, builds on the work of the first three. To recap: the first deliverable, *Survey of Methods of Trust Elevation Version 1.0*, consists of a broad overview of current and near-future online trust elevation techniques used for (or capable of) elevating a relying party’s assurance that the user requesting access to its resources is actually the person he or she claims to be. The second deliverable, *Analysis of Methods of Trust Elevation Version 1.0*, evaluated how each of the identified trust elevation mechanisms operated and what threats they mitigated that added to the relying party’s confidence in the identity asserted. A discussion of the methodology used to analyze the identified mechanisms has been included in that deliverable. The third deliverable, *Electronic Identity Credential Trust Elevation Framework Version 1.0*, is an abstraction intended to help to develop applications conforming to an accepted way of elevating trust of a digital identity.

As has been the pattern for this TC’s deliverables, this fourth deliverable builds on the work of the first three and specifies protocols for the elevation of trust.

## Goals of the Fourth Deliverable

Trust Elevation Methods are used to increase assurance in entity identification using authentication events and related entity information for the purpose of risk mitigation when making access control policy decisions.

The goals of this Fourth Deliverable are:

* To propose simple Trust Elevation architectural patterns demonstrating the use of Trust Elevation in modern Access Control architectures.
* To describe a common metadata set, mechanisms and protocol elements for Trust Elevation information exchanges.
* To promote the use of Trust Elevation elements to facilitate standardization among the many technologies and approaches currently in use for credential & authentication risk mitigation.

## Trust System Context

The context for the Trust Elevation techniques described in this document is a closed trust system. The participants, authentication methods, communication protocols and authorization methods must be agreed upon among the participants (possibly excluding Subjects). New participants and/or methods may be introduced to the trust system using appropriate onboarding processes.

The trust system must be closed due to the lack of generally agreed-upon criteria and evaluations of an authentication method’s efficacy to counter threats, mitigate impacts or reduce negative occurrence frequency, as well as local extrinsic concerns. For example, one trust system may consider passwords to be sufficient for identification whereas another trust system may require additional fraud detection infrastructure to realize the same degree of sufficiency.

The term Trust System could refer to: federated systems; systems controlled by a single governing entity; or a single system. The critical factor is the shared business rules and technologies related to authentication and authorization for performing trusted transactions.

## Assumptions for Trust Elevation Approaches

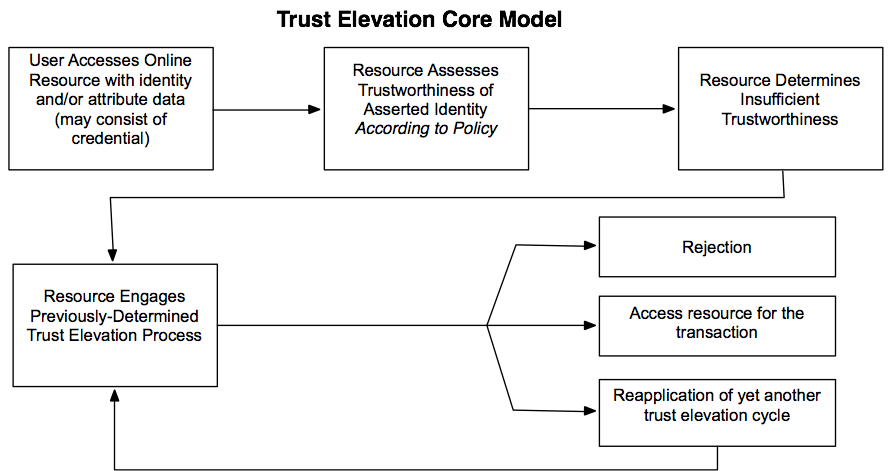
There are several assumptions that help set the context for this work:

* The Resource Owner has a defined set of requirements for authentication and/or authorization control. The requirements may include combinations of static rules and dynamic risk evaluations.
* In the case of Federated services, the Federation agreement defines the available identification and authentication methods and their relationship to discrete ‘levels’ of assurance that map to risk mitigation or compensating controls.
* Authentication methods are described sufficiently to allow creation of sets of compatible methods that cover identifiable risks or threats. For example, password authentication and hard token authentication are known to cover independent authentication factors.

# Conceptual Models

## Trust Elevation Core Model

As described in *Electronic Identity Credential Trust Elevation Framework Version 1.0,* the following depicts the core model for Trust Elevation.



## Trust Elevation Concepts

While the flow diagram above is easy to understand, implicit in the Core Model are several key components and processes. The first of these is a component within the resource which functions as a policy engine capable of consuming the asserted user data and making a determination as to whether that data satisfies the resource’s policy for authentication risk mitigation. Of course, the resource must have previously performed a risk assessment and adopted a risk mitigation strategy (NIST RMF and ISO ISMS are good examples of methodologies for these antecedent steps).

The second key component is again an antecedent service generated during the risk assessment and mitigation process. It is composed of a capability to recognize which, if any, risks have been adequately mitigated by the initial transaction, which risks remain to be mitigated and preferred methods for satisfying the remaining needs. The third key component is a component for evaluating the success of the trust elevation transaction. This could be an iteration of the first component, but it has been broken out in the above graphic to clarify the decision flow.

While these components are necessary to implement trust elevation of a presented online identity, they require the resource manager to have engaged in prior planning and assessments in order to generate the information necessary to direct the behavior of the components. In addition to implementing recognized, standards-based risk assessments, the prior work of this Technical Committee provide the necessary guidance for populating the decision-making components of the Core Model as well as most comparable models.

Trust Elevation methods are used to increase confidence in entity identification using authentication events and related entity information for the purpose of increased risk mitigation when making access control policy decisions.

Levels of Assurance models are structured such that increased risk mitigation results in increased Credential or Identity Assurance Level trust. These models require determination of a given transaction’s identity and authentication risk, expressed in terms of Level of Assurance Requirements. Policies are designed such that Credential or Identity Assurance Level must meet or exceed the Transaction Level of Assurance Requirement.

As described in *Electronic Identity Credential Trust Elevation Framework Version 1.0*, Entity identification confidence may be increased by: mitigating an authentication threat not addressed by the original authentication exchange; improved mitigation of the original authentication threat, or examination of contextual or environmental factors to corroborate the existing identification.

The definition of the composition of a particular Assurance Level scheme, and related policy evaluation criteria, is the responsibility of the parties involved in the transactions. The scheme should be tailored to the risk tolerance and requirements of the Relying Party. In other words, it is up to the resource manager to determine when sufficient mitigations of risk have occurred.

## Use of Authorization Architectures and Models

Another way to look at Trust Elevation is as a species of Transaction or Access Control Authorization. From this perspective, evaluation of the current state versus policy requirements results in decisions to ‘Permit’, ‘Deny’, or ‘Require Elevation’.

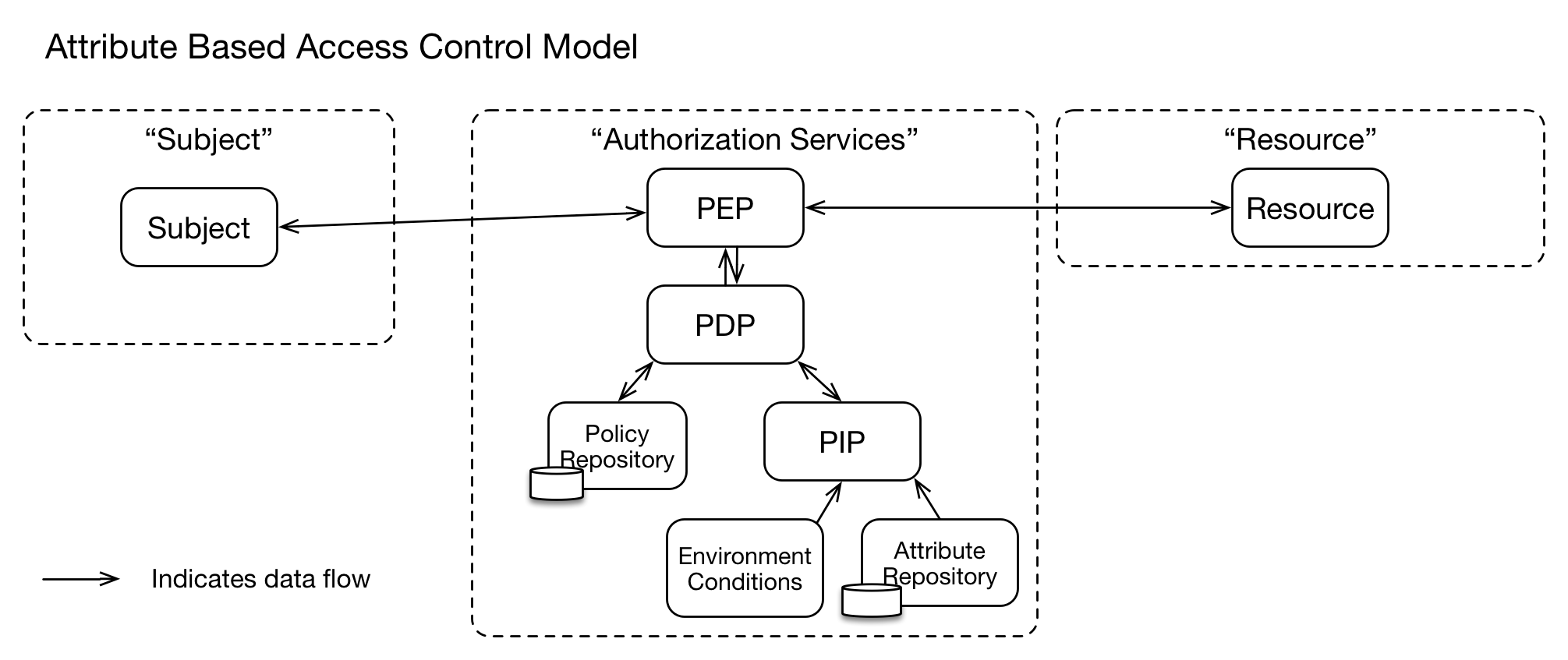
The Trust Elevation Core Model is compatible with other published Authorization Models, such as: Attribute Based Access Control (ABAC), OAuth and XACML, User Managed Access (UMA).

### Attribute Based Access Control Model

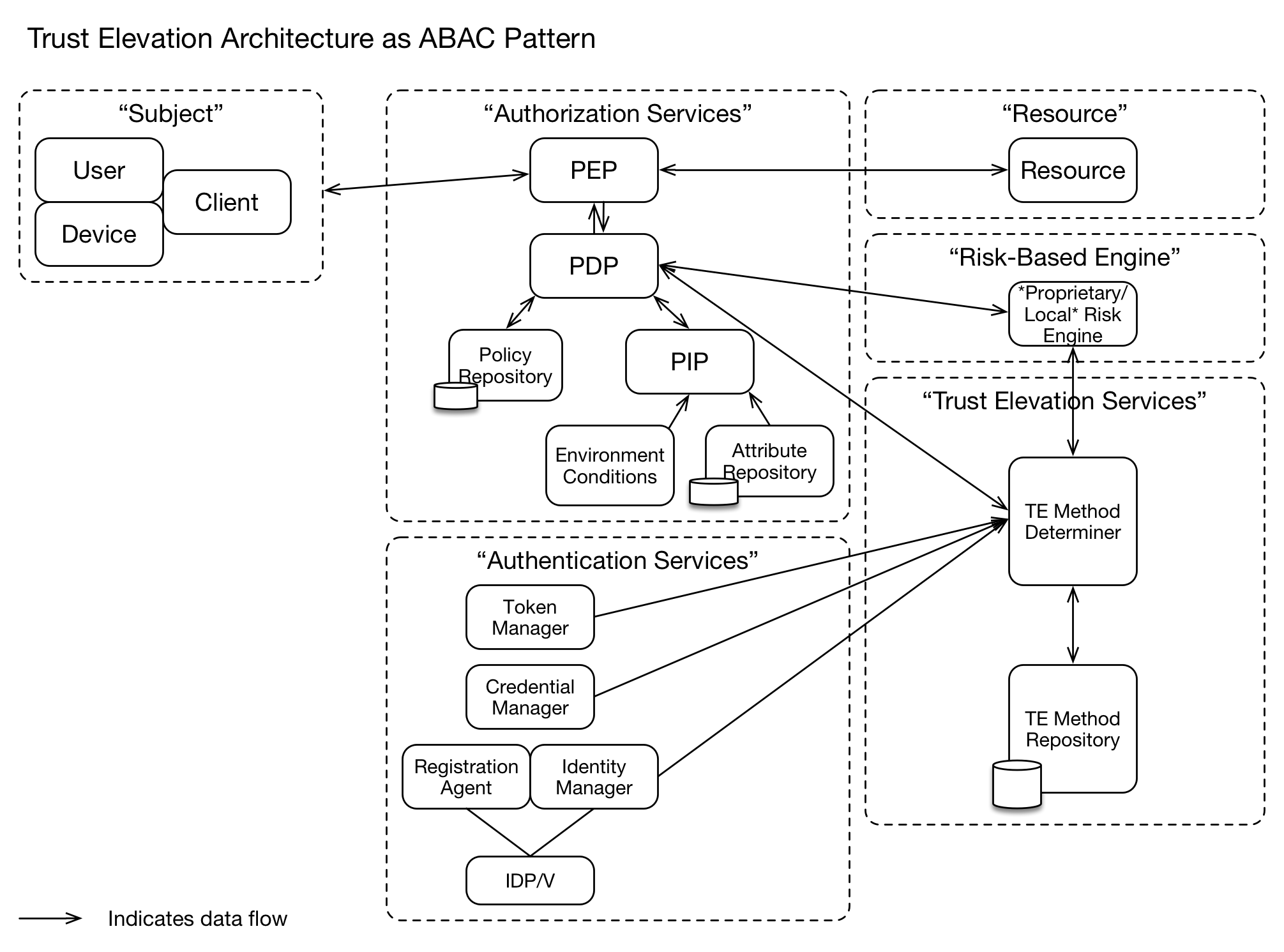
This section illustrates how Trust Elevation would fit into an Attribute Based Access Control model.

[NIST SP800-162] describes the elements of an Attribute Based Access Control Model.

As shown in the figure below, the primary components of Authorization Services are the Policy Enforcement Point (PEP) which intercepts resource requests; and, the Policy Decision Point (PDP) which checks supplied attributes versus access control policy. The PDP can obtain additional attributes from Environmental Conditions, Policy Information Point and other sources. Based on the policy evaluation, the PDP instructs the PEP to permit or deny access to the resource.



In the diagram below, when the Authorization Services determines that Trust Elevation is required, the Trust Elevation Services take information from “Authentication Services” and “Risk-Based Engine” to evaluate what Trust Elevation Method should be used to achieve the desired result.



### User Managed Access Authorization Model

The User-Managed Access protocol (UMA) defines a mechanism for a policy enforcement point – known as the resource server – to delegate authorization of a requesting party to a policy decision point – known as the authorization server – using elements of the OAuth 2.0 authorization framework.

To gain access to a protected resource, an UMA client (web or mobile application operating on behalf of a requesting party) must present a valid access token, called a requesting party token (RPT), to the resource server. The RPT must be valid and associated with sufficient authorization data, issued through a trust elevation process, before the resource server can grant access.

The authorization server, guided by policies set by the owner of the protected resource, elevates trust by testing whether the requesting party meets the policies. As part of this process, it can demand, for example, that the requesting party (or the client on their behalf) provide claims, such as identity information or even promises to adhere to constraints set by the resource owner, such as an embargo on information release until a certain date.

One policy the authorization server can consider is what mechanism was used to authenticate the person. UMA doesn’t require use of any particular authentication protocol, but works especially well with OpenID Connect.

The OpenID Connect Core specification defines two claims in the ID Token format called acr and amr, which provide details about what type of authentication was performed. Their values can be defined by a domain, a federation, a global registry, or some other trust framework. An UMA authorization server can test a requesting party against policies to evaluate the sufficiency of the authentication mechanism as provided in values of these claims.

In the event that the mechanism was not sufficient, the authorization server can indicate the reason for the authorization failure and what type of credentials would satisfy the policy. At this point, the client can request re-authentication from the OpenID Provider and ultimately re-request the RPT token. This flow would constitute trust elevation by step-up authentication.

### 4.3.3 XACML Authorization Model

The eXtensible Access Control Markup Language (XACML) standard defines a reference architecture for Attribute-Based Access Control (ABAC), a language for expressing access control rules and policies, and a protocol for generating and processing access control requests and returning responses.

Access to resources is mediated by a Policy Enforcement Point (PEP), which relies on decisions from a Policy Decision Point (PDP). When a user attempts to access a protected resource, the PEP assembles a request, which provides attributes about the user, the resource, the environment, and the action requested. The PEP communicates the request to the PDP, which evaluates it according to pre-defined policies.

To perform Trust Elevation, the access control policy can specify how users must be authenticated, including parameters such as authentication method, credentials accepted, and levels of assurance. Trust elevation in this context means enhancing authentication and/or authorization by means of requiring additional attributes.

Consider the following example: a user requests access to a protected resource. The access control policy governing the resource requires multi-factor authentication using a strongly vetted identity credential by means of setting the MustBePresent attribute to TRUE. The PEP controlling access to the resource has only hitherto validated the user identity by means of a lower assurance username/password combination. When the PEP initially formulates the request, it bases the user identity attribute on the previous username/password authentication event. When the PDP receives the request, it evaluates the request according to the appropriate policy, based on the resource. Since MustBePresent = TRUE, the PDP renders an “Indeterminate” decision, with a status code of “urn:oasis:names:tc:xacml:1.0:status:missing-attribute”. Upon receiving this “Indeterminate” with MissingAttribute status decision from the PDP, the PEP may resubmit a request after acquiring the proper attributes. In this case, the proper attributes could only be gathered through a step-up authentication event. This sequence constitutes a sample Trust Elevation event.

Alternatively, security administrators and resource owners may devise a series of Boolean attributes to test for authentication methods used, i.e.:

subject-id-authenticated-by-password

subject-id-authenticated-by-smart-card

subject-id-authenticated-by-biometric-iris-scan

subject-id-authenticated-by-biometric-fingerprint

subject-id-authenticated-by-two-factors

subject-id-authenticated-by-three-factors

This would allow policy authors to specify which methods are acceptable by testing for a TRUE result among the list they define as meeting security requirements.

Lastly, the Obligation element of XACML could be used to perform Trust Elevation. Any rule that permits access and specifies the authentication level required would add an obligation stating the minimum required authentication level. e.g.,

if “User authorized” then Permit. FulfillOn=Permit -> authenticated-by-two-factors-obligation.

In this case, the PEP does not need any special attributes. It makes a normal authorization request. If the response is Deny or NotApplicable, then the authentication level is irrelevant because the user is not allowed access. If the response is Permit without any authentication level obligations, then access is allowed even at the lowest authentication level. If the response is Permit with specific authentication level obligations, then the PEP must perform step-up authentication to the authentication level of the highest level of the obligations it received. If the highest level is satisfied, then any lower levels are satisfied. If that step-up fails or cannot be attempted, then access is denied. If step-up succeeds then access is allowed without needing an additional authorization request.

### 4.3.4 SAML Backend Attribute Exchange (BAE) Model

The Security Assertion Markup Language (SAML) standard defines a means for representing authentication events between different trusting security domains. A SAML assertion may contain a variety of attributes about the requesting subject and the conditions of the authentication event. Subject and Issuer attributes generally relate the name of the subject and the name of the organization with which the subject is associated in the AuthenticationStatement element. The AuthenticationStatement also contains an AuthenticationContext attribute, which details how the subject was authenticated in the context of the current assertion.

SAML-aware relying party applications can request additional attributes via the AttributeQuery element. Moreover, SAML authorities can request full attribute evaluations via the AuthzDecisionQuery element. Relying parties may specify acceptable authentication methods and credentials by using the RequestedAuthnContext element, and can force a fresh authentication event by setting ForceAuthn to true.

Trust Elevation can be exemplified in the following scenario using SAML: a user attempts to access content protected by a SAML-aware relying party (RP) application. The user posts a SAML assertion containing Subject/Issuer attributes and indicates a low level assurance authentication event to the RP. The RP’s access control policy requires additional attributes and a higher strength credential and authentication event. The RP initiates a SAML authentication request to the user’s home domain. This forces a step-up authentication event and retrieval of additional attributes, as required by the attribute contract. As with the XACML model, trust elevation means enhancing authentication and/or authorization by means of requiring additional attributes.

# Architecture & Design Considerations

## Architecture & Design Factors

There are many potential factors that influence the design specific Trust Elevation architectures. The nature and impact of the factors is determined by local requirements.

### Definition of ‘Elevation’ or ‘Step-Up’

The semantics of combining authentication methods to increase risk mitigation are dependent on local definition of authentication method characteristics within a Trust System.

The risk models of the Relying Party and/or Federation that comprise the Trust System should be considered when defining how combinations of methods modify risk mitigation.

For example, in one Federation repetition of a password authentication to re-confirm the authenticator may change the risk mitigation from ‘Low’ to ‘Medium’. In a different Federation, the same risk mitigation change might require a second authentication method which is different from the first one used.

The full range of permitted combinations and their effect on risk mitigation should be defined for the local entities.

### Use of Shared Definitions

As with authentication method combinations, the specification of each permitted authentication method should be shared within a Trust System.

For example, if a Fingerprint Template biometric is to be used, common specification of sampling mechanics, template calculation and comparison algorithms is essential. Variance in specification within a Trust System will result in different semantic meaning when combining authentication methods.

### Authentication State Tracking

Authentication state per Subject may need to be kept.

The Trust Elevation system may need to know which authentication methods have been attempted in prior transaction attempts in order to select the correct authentication methods to be attempted next.

For example, the policy may state that moving from Low to Medium assurance means use of an additional authentication method that was not used previously by this Subject for the in-scope transaction.

Tracking State per Subject and transaction attempt may prove to be a complex undertaking unless care is taken when designing elevation policy.

### Location of Policy Decisions

The architecture and design should be able to accommodate local, remote and distributed policy evaluation. Policy evaluation for trust elevation purposes may occur within a single system, or may occur in several different systems then combined.

A mechanism for calculating the combined result of the policy evaluation must be designed.

### Consideration of Time or Quality Degradation

When designing the state model for the authorization system, time-related degradation of information quality or authenticator validity should considered. The degradation could be defined as nil, or according to a specified time function.

### Responsiveness to Threat Environment

The effect of changes in the threat environment may cause changes of calculated assurance levels. Designers should determine if and how to respond to changes to the threat environment.

For example, if a system component is observed to be under active attack, the Authorization System may require increased assurance levels through use of additional authentication methods.

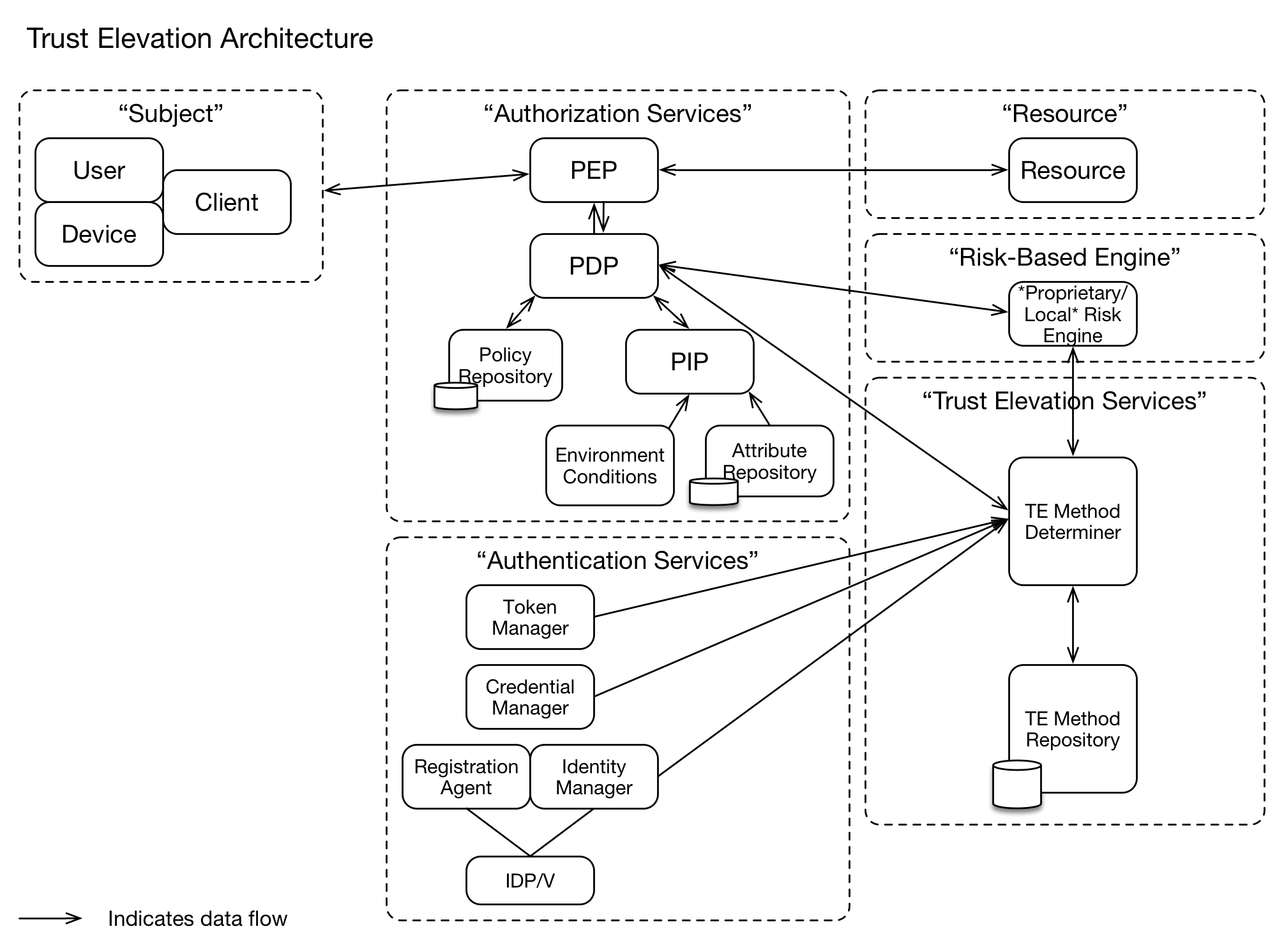
### Inclusion or Separation of Identity Information and Credential

Credential systems may be designed such that the credential directly contains entity identity information. In these systems, presentation of the credential might be the correct method for communication of entity identity information.

Other credential systems may be designed such that the credential authenticator is opaque or “pseudonymous” and cannot be used directly to obtain entity identity information. In these cases, methods to obtain identity information claims may be required to increase certainty of entity identification.

## Trust Elevation Architecture Components

The following architecture diagram shows Trust Elevation components and other components related to Trust Elevation and their core functions. The dashed line boxes represent the boundary for each major component. The solid line boxes represent the functions within the major components. In other authorization model representations, the functions may have different names and may possibly appear within different major component boundaries.



### Trust Elevation Services Component

The Trust Elevation Services Component is comprised of the Trust Elevation Method Determiner and the Trust Elevation Method Repository.

When the Authorization Services Component decides that the Subject is not permitted to access the resources due to insufficient identification and authentication assurance, the Trust Elevation Services Component is used to select an additional authentication method or methods which would allow the Subject to access the resources.

The Trust Elevation Services Component enables the Authorization Services to ask the Subject to retry access using different or additional authenticators.

Significantly, the Trust Elevation Services are aware of the methods and authenticators previously used by the Subject to attempt access. This enables mitigation of identification threats different from the initial authentication methods and authenticators, without having to hard code all combinations of authenticators that could be used.

For example, if the initial authenticator used username/password (a ‘know’ factor), the Trust Elevation Services would not recommend that authenticator if asked for another single factor authenticator: it might return a ‘have’ or ‘are’ factor authentication method, or a ‘know’ factor authentication method that is not username/password.

#### Trust Elevation Method Repository

The Trust Elevation Method Repository contains information necessary to the functions of the Trust Elevation Method Determiner.

The Method Repository contains information about the implemented authentication methods and their characteristics. These characteristics are used in the Trust Elevation Policy when the concepts of ‘stronger’ authenticators or ‘more’ assurance are represented.

For example, if the Trust System uses authentication factors to determine authenticator strength, it may consider a single factor authenticator as ‘weaker’ than a two-factor authenticator. In this case the characteristics should include details of which authentication factors are used.

#### Trust Elevation Policy

The Trust Elevation Policy maps the combinations of authenticators to the desired assurance levels.

Given the desired assurance level, the Trust Elevation Method Determiner is able to evaluate Policy to identify the list of authentication methods that could be used to achieve the desired assurance level. Information about the already-used authentication method can narrow the list of authentication methods if policy indicates that different methods should be used.

#### Trust Elevation Method Determiner

The Trust Elevation Method Determiner makes Trust Elevation policy decisions.

It receives requests from the Authorization Services Component that include current authentication state information of the Subject and the desired Level of Assurance.

The Trust Elevation Method Determiner uses policies stored in the Trust Elevation Method Repository to determine which, if any, authentication methods could be used to achieve the desired Level of Assurance.

The current authentication state information may include data about: authenticators presented to the Authorization Services component; authentication methods that were used by the Subject to achieve the current authentication state; and, the current Level of Assurance of the Subject.

If the authentication capabilities of Subjects (user, device or client) are dynamic or dependent on device, user or software abilities and features, the Method Determiner may need information about the specific capabilities of the specific Subject in order to avoid unnecessary round trips to the Subject.

## Other Related Architecture Components

### Authorization Services Component

The Authorization Services Component must be capable of requesting and processing Trust Elevation information. Trust Elevation Services may be treated as an information source or a remote policy engine.

The Authorization Services Component may need additional functionality to handle and track multiple access attempts by the Subject as the Subject responds to elevation requests.

### Risk-Based Engine Component

If the Risk-Based Engine Component exists, it represents systems that may be used by the Resource Owner to detect, measure and respond to threats in the operational environment. For example, detection of increased online attacks could cause the Resource Owner to require a greater degree of identification or authentication for access to resources.

# Implementation Considerations

## Orchestration

[ED: discussion about the need for orchestration – when an elevation is determined to be needed, how do all the component parts execute the elevation and loop back to the TE Determiner?]

## Protocol Capabilities

[ED: discussion about protocol support for TE-related actions. E.g. SAML and UMA/OAuth have extendable definitions that could be used – but the extensions would have to be created (acr/amr)]

## Assignment of Roles and Responsibilities

[ED: Section to include discussion on which actors get which functions in the flow.]

## Enumeration of Authentication Methods

[ED: discussion about the need to have a complete list of valid authentication methods for the trust system]

### Authentication Services Component

The implemented authentication methods must be enumerated and details captured in some form of Trust Elevation Repository.

The details that must be captured are identified in Deliverable 2, comprised of threats eliminated and risks mitigated. The detailed information will enable analysts to design Trust Elevation sequences that use complementary authentication methods to strengthen risk mitigation.

[ED: more detailed example here]

### Subject Component

Authentication methods recorded in the TE Method Repository may involve any combination of User, Device and Client. The Subject component might present to the Authorization Services in a different configuration than at registration. Ensure that Authentication and Identity Information methods make no assumptions about the relationships between the sub-elements of the Subject.

For example: the same User attempting access from a different device that has an identical device model has lower assurance than use of the originally registered device. Authentication methods involving the device need to be able to differentiate between those devices.

### Effect of Device Capability Changes

Devices may have different authentication method capabilities over time. For example, at enrolment, a smartphone registers the presence of a camera; but at transaction time, the smartphone camera is non-functional.

## User Enrolment

Enrolment is a key phase for future execution of Trust Elevation. At enrollment time, the Trust System must identify, record and possibly provision authentication methods. These authentication methods could include user, device, geo-location, network location and environmental elements.

For example, if geo-location is an available authentication method, the expected geo-location parameters must be captured at enrolment such that they can be compared to the geo-location during the transaction.

## Risk Engine Integration

For implementations with dynamic environmental risk evaluation, the Trust Elevation Component should be integrated. This may be accomplished by adjusting the Trust Elevation Policy such that more or less authenticator strength is needed to achieve a desired level of assurance.

For example, if the risk evaluation engine detects an general increase in fraudulent activity, it may instruct the Trust Elevation scheme to require additional authentications and checks for all transactions.

# Trust Elevation Sequence Example

This section contains a simple Trust Elevation Use Case.

Note that the specific structure and content of the Policy Table and Methods Table are defined within the trust system, driven by the Relying Party’s authentication policies.

In this simple example, a static mapping of Relying Party defined Transaction Risk Levels to pre-defined authentication strengths encoded as Levels of Assurance is shown. The Relying Party defines which LOA transitions are required for each Transaction Risk Level.

The policies are based on the Authentication Factors approach to risk mitigation. The Relying Party policy sets out the permitted combinations of authentication factors required to move from one LOA to another LOA.

Note that all transitions for all risk levels are not necessarily defined. The Policy Table only shows valid policies for this Relying Party within this trust system. If a particular transition is not defined, it is deemed to be invalid.

## Use Case: Online banking transactions

### Description

A bank customer initially logs on to the bank site (through a browser or mobile app) to view their account balance. Then, they decide to perform a higher risk transaction that requires a higher level of authentication: a funds transfer of $X.

### Pre-conditions

* Claimant has an existing relationship with the bank (i.e., is an account holder)
* Claimant has previously registered his authentication methods (e.g., his password, device, biometric)
* There are three LOA strengths

#### Transaction Risk Levels

|  |  |  |
| --- | --- | --- |
| **Transaction Designation** | **Transaction Name** | **Transaction Risk Level** |
| T1 | Check Account Balance | Low |
| T2 | Transfer Funds Out | Med |

#### Policy Table\*

The Policy Table is defined during system design by the Relying Party.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Transaction**  **Risk Level** | **Initial Strength** | **Desired Strength** | **Authentication needed\*** | **Policy designation** |
| Low | LoA0 | LoA1 | One factor, either what you know or have | P1 |
| Med | LoA0 | LoA2 | Two factors, any class | P2 |
|  | LoA1 | LoA2 | One factor, different than used for LoA1 authentication | P3 |
| High | LoA0 | LoA3 | Three factors | P4 |
|  | LoA1 | LoA3 | Two factors, any class, different than used for LoA1 authentication | P5 |
|  | LoA2 | LoA3 | One factor, different than used for LoA1/2 authentication | P6 |

Where LoA0 represents a "user not logged in" state.

\*Authentication policies are set by the relying party.

#### Methods Table:

The Methods Table enumerates the authentication methods available in the trust system.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method designation** | **Method description** | **Class(es)** | **SF strength** | **Threats addressed\*** |
| M1 | PIN (>=4 char) | Know | 1 |  |
| M2 | Password (>=8char) | Know | 1 |  |
| M3 | Device ID | Have | 1 |  |
| M4 | Crypto key (TLS protocol) | Have | 2 |  |
| M5 | Biometric – face | Are | NA |  |
| M6 | Biometric – fingerprint | Are | NA |  |
| M7 | PIN + Device ID | K+H | 2 |  |
| M8 | Crypto key + face | H+A | 3 |  |

\*For the benefit of relying party operators setting up policies.

### Process Flows

#### Transaction 1: Check Account Balance

* Claimant accesses bank site/app
  + Claimant state is LoA0 (not logged in)
* Claimant logs onto their account to Check Account Balance (T1).
  + T1 is a Low risk transaction
    - Claimant needs to go from LoA0 to LoA1
  + Policy P1 is invoked
  + Methods M1, M2, M3, M4 are able to meet the Policy P1 requirement. System designers chose to implement M2 “(Password (>= 8char)” in this case
  + User prompted for and enters password
    - Authentication server verifies password
  + Access is granted, balance displayed

#### Transaction 2: Transfer Funds Out

* Claimant then chooses to Transfer Funds Out (T2)
  + PDP determines current Authn level (LoA1)
    - Looks up current state for that user
  + PDP determines target Authn level (LoA2)
  + PDP demands trust elevation (LoA1-2)
    - Request sent to MD
    - MD selects Policy P3 to meet the TE requirement
    - MD returns response to PDP
  + PDP sends Authn request to Authn server, specifying policy P3
  + Authn server determines currently available methods
    - Authn server issues appropriate challenges to user
    - User/device responds to challenges
    - Authn server verifies Authn data
  + Authn server send Authn results to PDP (standard assertion, e.g., SAML)
  + PDP makes access decision and sends to PEP
  + PEP provides access to requested resource (i.e., funds transfer function)
  + User proceeds with transfer

### Component-Component Communications

Content of PDP-MD request:

* Current level
* Method(s) that were used to achieve current level
* Target level

Content of MD-PDP response:

* List of methods that could be used to achieve target level

Content of PDP-AuthnSvr request:

* Claimant ID
* List of acceptable methods

# Metadata and Assertions

[**EDITORS NOTE**: Depending on where we get to with the Methods Repository, these assertions may change. There are structures in SAML and OAuth (acr and amr) that are potential candidates for expression of Trust Elevation authentication stuff. ]

## LoA Assessment (LA)

**LA Request**

<trustel:AssessLoaRequest>

<trustel:AssertedID>...</trustel:AssertedID> //could be SAML token

<trustel:AssertedAttribute>

<trustel:Attribute>...</trustel:Attribute> //Attribute

<trustel:AttributeProvider>...</trustel:AttributeProvider> //AP OU

<trustel:AssertedAttribute>

<trustel:AssertedAttribute>

<trustel:Attribute>...</trustel:Attribute> //Attribute

<trustel:AttributeProvider>...</trustel:AttributeProvider> //AP OU

<trustel:AssertedAttribute>

<trustel:Loa>....</trustel:Loa> //current LoA Strength in numerical value

<trustel:LoaAssessor>...</trustel:LoaAssessor> //LoA OU

<trustel:AuthnContext>

<trustel:AuthnDeviceSig>..</trustel:AuthnDeviceSig> //Device Fingerprint

<trustel:AuthnLocation>...</trustel:AuthnLocation> //Device location

<trustel:AuthnIP>...</trustel:AuthnIP> //IP of the device

<trustel:AuthnTime>...</trustel:AuthnTime> //time of request

</trustel:AuthnContext>

</trustel:AssessLoaRequest>

**LA Response**

<trustel:AssessLoaResponse>

<trustel:AssertedID>...</trustel:AssertedID> //could be SAML token

<trustel:LoaStrength>...</trustel:LoaStrength> //could be numerical value

<trustel:LoaAssessor>...</trustel:LoaAssessor> //LoA OU

<trustel:LoaAssessorSig>...</trustel:LoaAssessorSig> //LoA OU Signature

</trustel:AssessLoaResponse>

## Method Determination (MD)

**MD Request**

<trustel:MethodTypeRequest>

<trustel:AssertedID>...</trustel:AssertedID> //could be SAML token

<trustel:AssertedAttribute>

<trustel:Attribute>...</trustel:Attribute> //Attribute

<trustel:AttributeProvider>...</trustel:AttributeProvider> //AP OU

<trustel:AssertedAttribute>

<trustel:AssertedAttribute>

<trustel:Attribute>...</trustel:Attribute> //Attribute

<trustel:AttributeProvider>...</trustel:AttributeProvider> //AP OU

<trustel:AssertedAttribute>

<trustel:Loa>....</trustel:Loa> //current LoA Strength in numerical value

<trustel:LoaAssessor>...</trustel:LoaAssessor> //LoA OU

<trustel:AuthnContext>

<trustel:AuthnDeviceSig>..</trustel:AuthnDeviceSig> //Device Fingerprint

<trustel:AuthnLocation>...</trustel:AuthnLocation> //Device location

<trustel:AuthnIP>...</trustel:AuthnIP> //IP of the device

<trustel:AuthnTime>...</trustel:AuthnTime> //time of request

</trustel:AuthnContext>

</trustel:MethodTypeRequest>

**MD Response**

<trustel:MethodTypeResponse>

<trustel:AssertedID>...</trustel:AssertedID> //could be SAML token

<trustel:Method>...</trustel:Method> //could be "|" delemited array of methods

<trustel:MethodDeterminer>...</trustel:MethodDeterminer> //MD OU

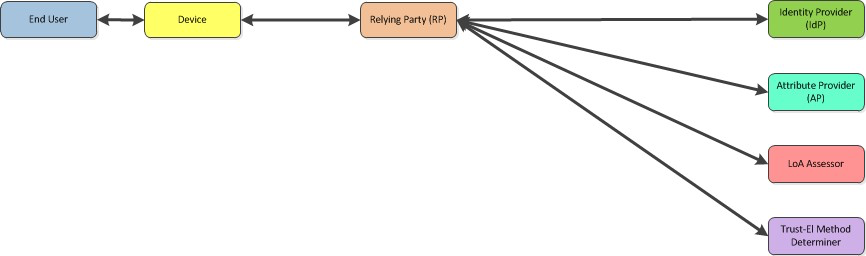
<trustel:MethodDeterminerSig>...</trustel:MethodDeterminerSig> //MD OU Signature

</trustel:MethodTypeResponse>

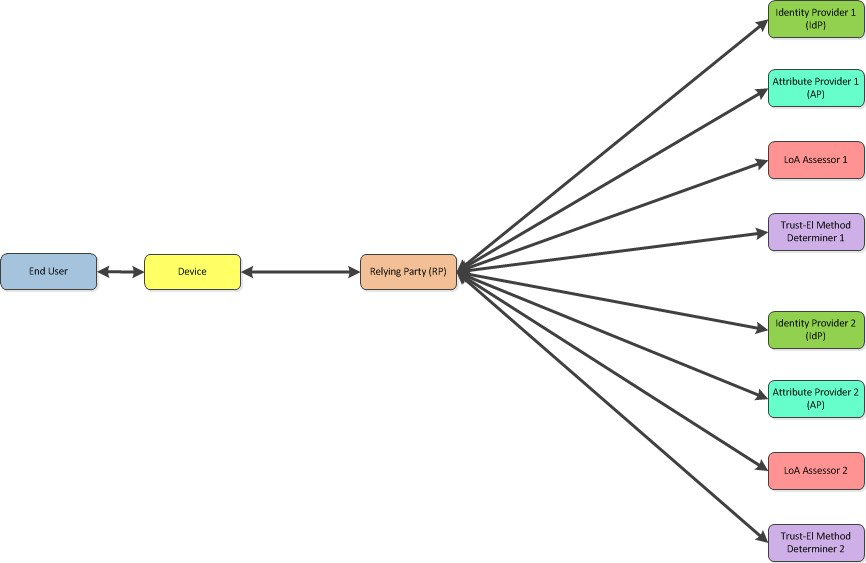
# Use cases

[ED: These Use Case diagrams may be removed once we settle on the contents of Section 6.]

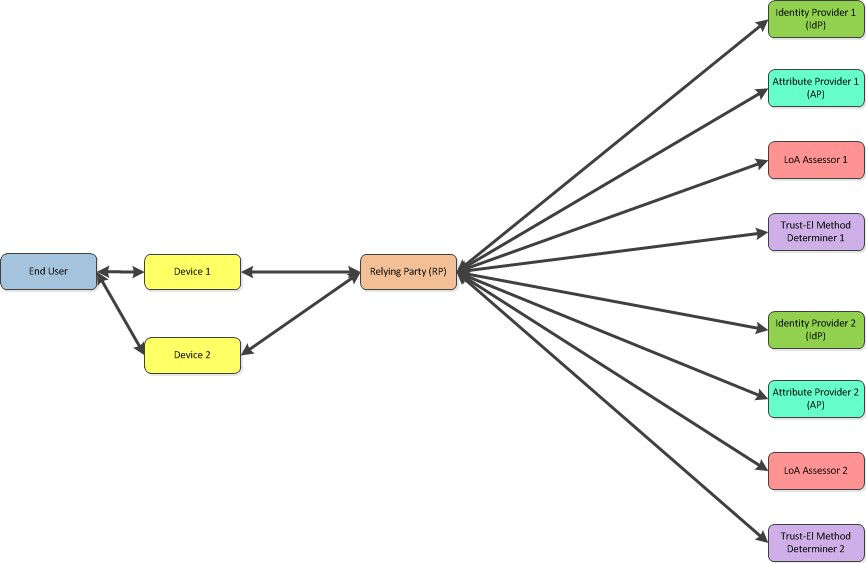
## Trust-El Use Case 1



## Trust-El Use Case 2



## Trust-El Use Case 3



# # Conformance

The last numbered section in the specification must be the Conformance section. Conformance Statements/Clauses go here. [Remove # marker]

1. Acknowledgments

The following individuals have participated in the creation of this specification and are gratefully acknowledged:

Participants:

[Participant Name, Affiliation | Individual Member]

[Participant Name, Affiliation | Individual Member]

1. State Models for Assurance Level Evaluation

## Evaluation of Assurance Requirements at Transaction Time

One of the core assumptions of Trust Elevation is that a subject attempting a transaction is unable to meet the policy requirements for identification certainty unless an Elevation event occurs.

An important concept is that measured assurance levels change over time due to many factors. At the instant of authorization policy evaluation, the current state of identity attribute assurance level and authenticator assurance level are compared to the Transaction’s Assurance Level Requirement. If the measured assurance levels are greater or equal to the requirement, the transaction proceeds.

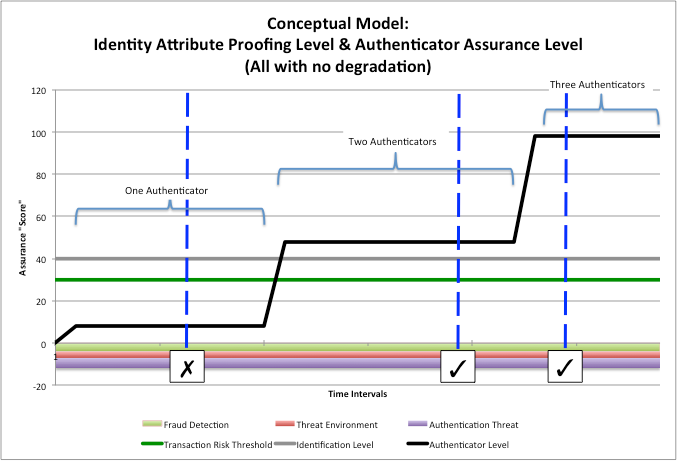
The graphics show that the assurance level of the Identity Information Attributes established via the Identity Proofing and Verification processes are separate and unlinked to the assurance level of the Authentication Event (which includes Credential and Authenticator details). This approach is consistent with the NIST SP800-63 LOA calculation method.

### Up-Front Policy Evaluation of Proofing and Authenticator Levels

This graphic illustrates a scenario where the levels of identity attribute assurance and authenticator assurance are determined in advance and do not degrade over time.

The vertical dashed lines represent the potential points in time of the transaction event. The identity attribute assurance and authenticator assurance levels are compared to the transaction assurance level requirement. If both values are greater than the requirement, the transaction can proceed (check mark). If one or both are lower, the transaction cannot proceed (X mark) and is either rejected or directed to a trust elevation event.

Trust Elevation in this scenario combines authentication factors to step up combined authenticator assurance to meet or exceed the transaction requirement.



Notes:

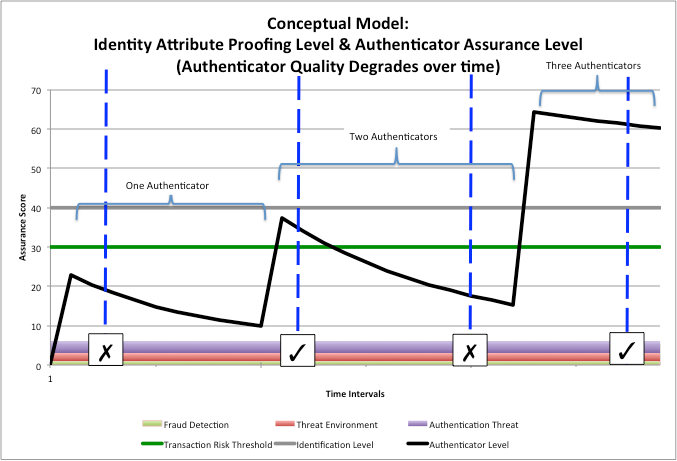
* The ‘Assurance Score’ is a simple numerical representation of the degree of certainty for illustrative purposes. ‘Assurance Level 3’ has been arbitrarily defined as ‘30’ on the scale
* The Grey line represents the assurance level resulting from the Identity Proofing and Verification process; established at Subject Registration time by the Registration Agent.
* The Black line represents the authenticator assurance level resulting from the Authentication event. It takes credential, authentication secrets and authenticator generation factors into account.
* The Green line represents the Resource Owner defined assurance score/level required for the transaction. It is based on the Resource Owner’s risk determination methods. In this example, the Transaction Requirement is ’30’ or ‘LOA3’
* The Black line initially shows the effect of a single authenticator, then two authenticators, then three authenticators.

### Time-Based Degradation of Authenticator Assurance Levels

The assurance level of the Authenticator is important. This graphic illustrates a scenario where the authenticator assurance level changes over time due to time-based degradation of the credential, secrets and authenticator generation processes.

The vertical dashed lines represent the potential points in time of the transaction event. The identity attribute assurance and authenticator assurance levels are compared to the transaction assurance level requirement. If both values are greater than the requirement, the transaction can proceed (check mark). If one or both are lower, the transaction cannot proceed (X mark) and is either rejected or directed to a trust elevation event.

This scenario shows that due to rapid degradation of authenticator assurance for most time periods, Trust Elevation to three authenticators is needed for the transaction policy.



Notes:

* The ‘Assurance Score’ is a simple numerical representation of the degree of certainty for illustrative purposes. ‘Assurance Level 3’ has been arbitrarily defined as ‘30’ on the scale
* The Grey line represents the assurance level resulting from the Identity Proofing and Verification process; established at Subject Registration time by the Registration Agent.
* The Black line represents the authenticator assurance level resulting from the Authentication event. It takes credential, authentication secrets and authenticator generation factors into account.
* The Green line represents the Resource Owner defined assurance score/level required for the transaction. It is based on the Resource Owner’s risk determination methods. In this example, the Transaction Requirement is ’30’ or ‘LOA3’
* The Black line initially shows the effect of a single authenticator, then two authenticators, then three authenticators.
* The downward slopes represent the time-based degradation of certainty of the authenticator
* Although not shown explicitly, refresh to original values could be achieved by re-issuance of credentials, or generation of new keys.

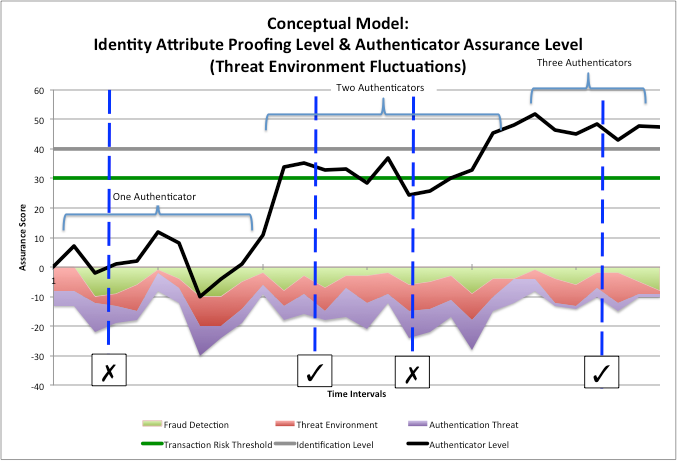
### Threat Environment Effects on Effective Authenticator Level

The last graphic illustrates a more complex example in which the overall threat level affects the Authenticator assurance level. A simplistic calculation is used where increasing threat environment, increasing detected fraud and decreased system security subtract directly from the authenticator assurance score.

This mimics the effect that a risk-based authentication system or risk engine might have on transaction assurance requirement evaluation.

As in the previous illustrations, the vertical dashed lines represent the potential points in time of the transaction event.

Where the increased threat level causes the effective authenticator assurance level to dip below the green transaction requirement line, Trust Elevation could be used to achieve the minimums necessary. Note that in the ‘Two Authenticators’ region, the transaction could proceed or fail depending on the magnitude of the threat levels. If the transaction fails, the Relying Party could choose to retry at a later time, or request additional Authenticators.



1. Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision** | **Date** | **Editor** | **Changes Made** |
| Draft 04b | 2015-09-04 | A. Hughes | Minor updates to base draft. |