This document is a response to the IETF Internet-Draft “Requirements for Message Access Control” (aka “Plasma”) from the OASIS XACML Technical Committee (TC). This document is submitted in accordance with the IETF comment process. Quotations from the IETF Plasma draft are shown in *italicized* black font with page number references. Comments and questions from the XACML TC are shown following in blue font. Spelling, grammar, and formatting errors will not be addressed in this document.

An Access Control Policy defines a set of criteria and evaluation logic that must be satisfied in order to grant access to the information. These criteria are defined in terms of attributes about the subject requesting access. (p. 4.)

Attributes about the resources requested, actions proposed, and environment must also be considered in access control decisions.

Attribute Based Access Control (ABAC) is access control based attributes of the subject. An ABAC policy specifies which attributes are needed to authorize access to a resource. These attributes may be provided by the subject as part of the access request or discovered by the access control engine based on relationships it has with attribute sources such as a directory or personnel database. (p. 7)

Instead: Attribute Based Access Control (ABAC) is an access control methodology based on evaluating the attributes of the subject, resource, actions, and environment according to rules and/or policies. An ABAC policy specifies which attribute values are needed to authorize access to a resource. Subject attributes may be provided by the subject as part of the access request or discovered by the access control engine based on relationships it has with attribute sources such as a directory or personnel database. Resource, action, and environment attributes can be parsed from the request, obtained from other authoritative sources or inferred by policy enforcement points at the time of the request.

In section “4.1 Vocabulary” (p. 28), a number of terms are introduced which have definitions elsewhere. We suggest that the following be considered (from the XACML core specification: http://www.oasis-open.org/apps/org/workgroup/xacml/document.php?document_id=45800)

Policy

A set of ***rules***, an identifier for the ***rule-combining algorithm*** and (optionally) a set of ***obligations*** or ***advice***. May be a component of a ***policy set***

Policy administration point (PAP)

The system entity that creates a ***policy*** or ***policy set***

Policy-combining algorithm

The procedure for combining the ***decision*** and ***obligations*** from multiple ***policies***

Policy decision point (PDP)

The system entity that evaluates ***applicable policy*** and renders an ***authorization decision***. This term is defined in a joint effort by the IETF Policy Framework Working Group and the Distributed Management Task Force (DMTF)/Common Information Model (CIM) in **Error! Reference source not found.**. This term corresponds to "Access Decision Function" (ADF) in **Error! Reference source not found.**.

Policy enforcement point (PEP)

The system entity that performs ***access control***, by making ***decision requests*** and enforcing ***authorization decisions***. This term is defined in a joint effort by the IETF Policy Framework Working Group and the Distributed Management Task Force (DMTF)/Common Information Model (CIM) in **Error! Reference source not found.**. This term corresponds to "Access Enforcement Function" (AEF) in **Error! Reference source not found.**.

Policy information point (PIP)

The system entity that acts as a source of ***attribute*** values

Policy set

A set of ***policies***, other ***policy sets***, a ***policy-combining algorithm*** and (optionally) a set of ***obligations*** or ***advice***. May be a component of another ***policy set***

For the purpose of the PLASMA work, it is desirable that the PEP and PDP be clearly defined as separate services which may be on separate systems. (p.32)

We concur that the architecture should allow for logical and physical separation of the PEP, PDP, PIP, and PAP components. Logical (and even physical) separation of PEPs and PDPs will ensure greater scalability and will meet more business requirements in cases where existing infrastructures cannot easily be modified for new monolithic services.

The PEP just needs opaque references to the policies and defers all decisions to the PDP. The use of policy references also minimizes any policy maintenance issues due to policy updates. The PEP can be required to carry out obligations of the policy such as specific encryption requirements such as key size or algorithm; or data integrity requirements such as signing or HMACing content. (p. 32).

Is the intention in this section to have the PEP provide policy references to PDPs via the request? This would not contravene the XACML core specification. The current XACML Profile of SAML allows policies to be provided by the PEP in the decision request either as the only policies to be used or to be combined with policies already trusted by the PDP. Passing references to policies raises challenging questions of access to those policies including availability, performance and trust.

In order to implement this, it would be best to specify in more detail:

- how policy references are to be determined by the PEP
- how they could be instantiated in metadata
- how the policy references are passed in the request from PEP to PDP, and
- how the PDP should consider policy references sent by the PEP.

The model is fundamentally an Attribute-Based Access Control (ABAC) model. Access is granted to information based on attributes of the subject. (p. 36).

Instead: The model is fundamentally an Attribute-Based Access Control (ABAC) model. Access is granted to information based on evaluation of attributes of the subject, resource, action requested, and environment according to rules and/or policies.

By name. This is where a reference to the policy is directly associated with the data. e.g. a URI or a URN which identifies the policy to be enforced or points to where the policy is published. For example with S/MIME the ESS label identifies the applicable policy by an OID. When an *access request is made to the data, the PDP finds the policy based on the identifier and then compares the access request to the referenced policy. (p. 38)*

The use of URIs/URNs has been shown to be an effective way of representing values in a request context, particularly when the values are fairly static. This would be the preferred structure for passing policy references.

Section 4.5.2 "Advanced policy": Advanced policy is intended to be used where one or more arbitrary policies are required on the content . It is intended to target more complex scenarios such as content with regulated information or content subject to other organization and contractual policies. The input set of attributes is defined by the policies and can be either primordial or derived attributes or both. Multiple policies have a logical relationship e.g. they can be AND or ORed together. It is not expected that all Plasma clients support advances policy. (p.41)

The XACML TC recommends that the formal adoption of the XACML 3.0 rule/policy format and rule/policy combining algorithms for the Plasma architecture. By adopting the XACML policy format, security architects would be able to define policies at the enterprise level which could be distributed and used by a variety of platforms, applications, and services, including the S/MIME messaging applications toward which Plasma is intended. Promoting interoperability at the policy level will reduce the administrative overhead associated with creating, distributing, and maintaining policies for disparate systems. It will also improve adopting organizations' security posture by increasing consistency across various policy domains. Furthermore, the XACML 3.0 rule- and policy-combining algorithms offer a richer set of mechanisms for constructing and evaluating complex policies than just logical ANDs and ORs.

-----

1) The pointer to the PDP MUST be checked against some policy before attempting to query the PDP for a policy decision. 2) Care MUST be taken when processing the responses from a PDP to check that they are well-formed and meet local policy before using the responses. (p. 46).

For "1)", it is unclear which component of the architecture should check the PDP pointer. Is it the PEP, or Plasma client? If so, what process will be used to validate this action? Should the Plasma draft also specify that the PDP should check the received pointers? Or is this the intention of "1)"?

For "2)", the XACML 3.0 Request/Response protocol and the SAML/XACML profile have provisions for validating the PDP decisions.

Plasma represents an innovative and standard way of providing better security and assurance with policy compliance in the email and messaging application domain. The XACML TC recommends that the proponents of Plasma in IETF consider building an XACML profile for Plasma. Specifically, the access control and encryption protocol that is sketched in the referenced draft should define a set of integration points and/or interfaces with XACML PDPs. The XACML TC welcomes profiles that standardize and satisfy particular use cases or are tailored for certain platforms (for example, a REST profile for XACML is currently in work).

For future iterations of this draft, we also suggest that the Plasma authors consider defining how Plasma would work in a federated authorization model. What trust models will be used? How will cross-domain trusts be established and managed? What protocols and transport mechanisms will be used for out-of-band policy management?

We recognize and agree that the use of SAML for providing information about subject attributes and authentication events is a good practice that should be extended to the messaging application domain.

In summary, we request that the Plasma authors add support for the following XACML features in next version of the draft:

- XACML reference architecture, including the notion of resource, action, and environment attributes in addition to subject attributes
- XACML policy format
- XACML rule- and policy-combining algorithms
- A Plasma-to-XACML protocol interface
- A Plasma profile for XACML containing the elements listed above.

Thank you for your consideration.