# Identity on the Cloud (iDCloud)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Abstract**

In an emerging world of digital identity, it is increasingly imperative for firms to have a strategy for implementing authentication, authorization and audit security events for resources within or across their domains. In almost all of the cases, a subject is trying to get access to a resource, which is governed by set of policies. To gain access to the resource, the subject must, as a first step, present its valid credentials for authentication. However, a valid credential alone is not enough. The subject must have valid attributes for authorization to access the resource. Every step in this process also needs to be recorded for audit purposes. All this needs to be coupled with a user management system which provisions and tracks the life cycle of a subject (user). Potentially, such a system can be used by the IT organization of any enterprise to model their identity management, access control, user management, single-sign-on (SSO) capabilities and Neustar can offer and operate this service on a public cloud.

One of the huge benefits of such an abstracted authentication and authorization model is that a firm can interoperate and centralize security across multiple systems and even support federated use cases. Consider the following use case. A typical modern IT firm deals with services on the Cloud. An Operations team might be dealing with AWS, the Engineering team with Google Apps and the Sales team with SalesForce. Each of these services may have different authentication schemes. An abstracted authentication layer will act as the gateway to deal with the complexities of authentication. Besides authentication, this layer will handle the conversion of the requests and responses along with other aspects like SSL. The authorization layer will determine if the request can be fulfilled based on the policies established for the user. For example, the information that a doctor can access about a patient will be more than what a nurse or a hospital administrator can access. Such decisions can be controlled via a centralized policy engine. This will allow the application developers to focus on their areas without having to deal with the complexities of authentication and authorization. Once the user is authenticated into the system, then that user can move through the enterprise without having to re-authenticate and thereby enabling SSO. This implementation can leverage Neustar's NexGen services for the Infrastructure and Platform (IaaS and PaaS) and hosted on a public cloud. It can also use or extend its Identity and Management (IAM) service, which currently is used for managing access to Neustar services, whereas the proposed model is for managing access for users within any organization.

To this affect, the current implementation aims at creating a generic model where authentication decisions can be de-coupled from authorizations decisions. The authentication layer can receive access requests through different types of protocols like SAML, OAUTH, LDAP etc. This layer will act like a Cloud Service Broker (CSB) and can provide gateway services like SSL, DDOS, Code Injection, Malware etc. These API gateways will be the bridge between the enterprise and the external APIs.

The authorization layer will be based on Attribute Based Access Control (ABAC). In this model, a Policy Administration Point (PAP) creates and manages polices, a Policy Decision Point (PDP) evaluates and issues authorization decisions, a Policy Enforcement Point (PEP) authenticates a request access and enforces PDP’s decisions and a Policy Information Point (PIP) provides attribute information to the PDP. The PEP in this case will reside in the CSB layer described above. The attribute based authorization engine will be created using the eXtensible Access Control Markup Language (XACML). The XACML standard defines a declarative [access control](http://en.wikipedia.org/wiki/Access_control) policy language implemented in [XML](http://en.wikipedia.org/wiki/XML) and a processing model describing how to evaluate authorization requests according to the rules defined in policies. The implementation intends to use XACML 3.0 and have support for multiple PEPs. A web interface will also be provided for the Administrator of policies. Any new implementation of this model will effectively, only need a custom PEP (if there is no existing PEP implementation). The PDP will only need policies to be defined by the web interface without any changes to the engine.

The Audit layer will record all security transactions that have taken place in the system.

The User Management System (UMS) will be a web based (or a mobile application) interface responsible for provisioning, de-provisioning and tracking the life cycle of a user within the enterprise.

Figure 1 illustrates the authentication and authorization flow.

**Figure 1**