OpenAutonomy High-Level Description

Jeff Disher (Copyright 2013 Open Autonomy Inc.)
Date: 2013-08-22
Version: 1

What is it?

OpenAutonomy is 2 things:

1. a platform and protocol for connecting applications on [potentially] different servers (in different domains) in a trusted and extensible way

2. a protocol for distributed identity management

The use as a distributed application platform is the focus of this document.

Note that there is no central authority across the OpenAutonomy platform. Each user acts as the centre of their own application universe and there is never a reason to rely on any single provider of OpenAutonomy services. This means that the platform can be used in non-public network environments (corporate intranets or isolated networks) or by users who wish complete control over their data and applications.

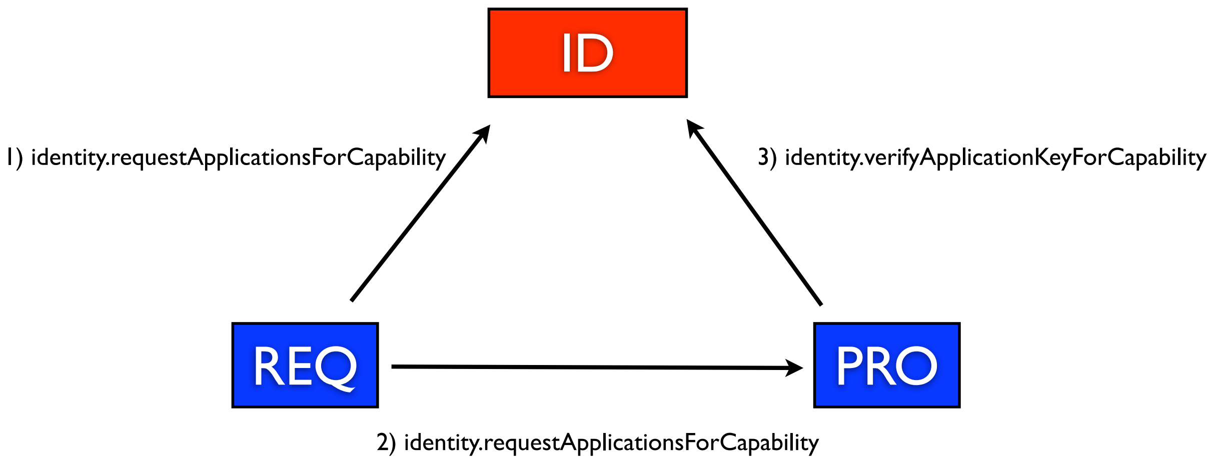
What does it do?

OpenAutonomy starts with an identity. This can be visualized as a combination between a microkernel name server and OpenID.

The **identity** is what anchors the user to the platform and it provides the means for distributed identity, trusted user group management (grouping other identities so that they can be granted specific access to applications), the point where new application instances owned by the user register to become trusted, find other applications and services, and as a trusted referral signing authority when an application wishes to find applications owned by other identities.

Connecting applications

The power in the platform is how applications can provide and consume services to and and from each other. If an application represents, for example, a cloud storage system, it can register that protocol with its owning identity. Then, if another application needs a storage capability (either to load or store data), it can ask the identity which applications provide that protocol.


ID
1) identity.requestApplicationsForCapability
3) identity.verifyApplicationKeyForCapability
REQ
PRO
2) identity.requestApplicationsForCapability

Extensibility

The protocols provided by an application are merely strings (encoded as the Java-style reverse-domain package names in order to avoid collisions without a dedicated central governing body). This means that any application developer can build their own protocol to do whatever they want, however they want it to work.

While the identity protocol uses XML-RPC for cross-server communication (which requires that all web applications implement a small set of RPCs), there is no requirement on the mechanism used by other protocols.

Web and native applications

From the perspective of OpenAutonomy, there is no fundamental difference between where applications are deployed, what languages they are written in, or what platforms they are running on with the exception of one factor: can it receive incoming communication attempts from an identity or application?

This changes initial application registration in one key way: web apps can be reached over HTTP so the identity can call them back to exchange trust tokens, thus verifying that the application is who they say they are while native apps must acquire their initial trust tokens by the user manually performing the exchange.

This also restricts the activity of non-web apps in one key way: they cannot provide services to other applications since the other application would have no way of contacting them.