# OA Storage for Developers

Problem to solve:

I have a great idea for a new cloud service! I can take arbitrary files and mash them up into an entertaining and informative interactive view. I think people would find this fun and interesting. The problem is that **I need users** and **access to their data**.

I don't think I can convince the users to sign up for yet another service, nor do I want to waste time developing a user management system. Also, I can't afford to build a data centre to store all the user data, nor do I think they will want to upload all of it just to play with this application.

OpenAutonomy solution:
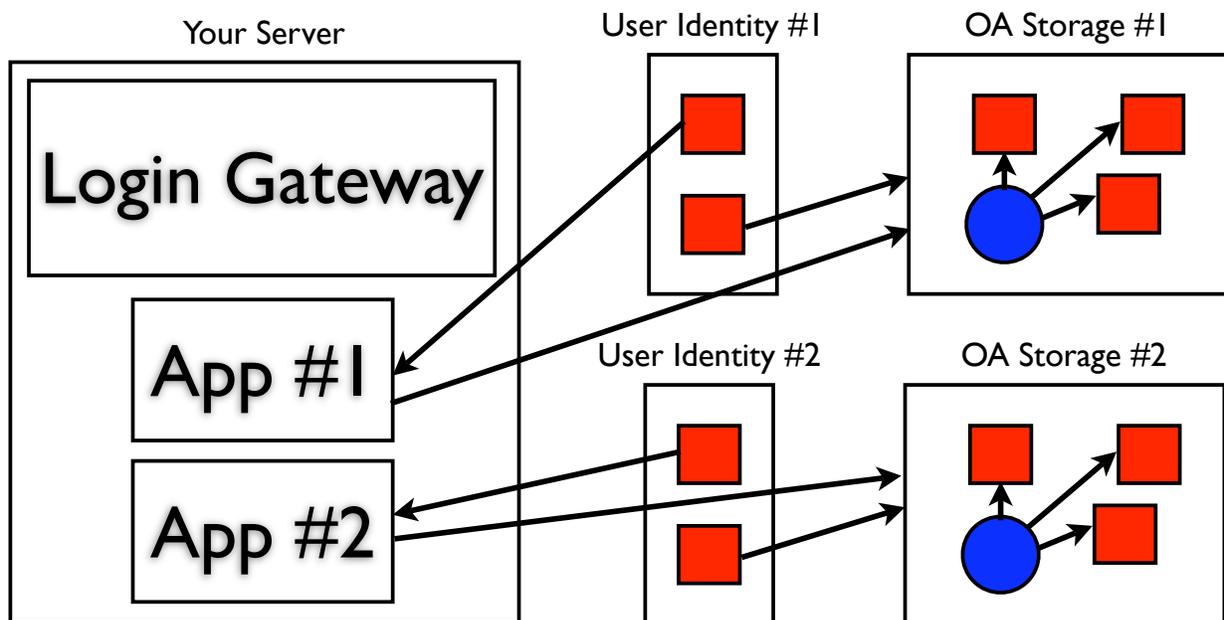
OpenAutonomy can solve both parts of this problem:

Access to OA users:
OpenAutonomy assumes that there is no difference between a user logged in on the server where their identity "lives" versus any other server. The only difference is the specific log in process but, after that, the system treats them the same (since the system is made to be distributed and extensible, adding any special treatment would cripple the extensibility).

Adding a log in gateway means that you will have a potential user base of all users who have an OpenAutonomy identity on any server in the world (demo code will be provided in the reference implementation of the PHP OA Server). Any user management you choose to add to filter users (due to spam or terms of use violations) or for analytics purposes is a secondary concern, specific to your use-case but possible to add.

Access to OA Storage:
In order for your application to have access to a user's storage, you must allow each user to instantiate it. The instance then tries to attach to the user's identity. The default **trust mode** used by most identities is to assume that newly attached applications are **NOT to be trusted** (since any application can "claim" to be a new application and try to get access to the user's other applications). This means you will need to tell the user to go to their identity and set the **trust for the new instance to MAX** before proceeding (as their storage will otherwise deny access).

Once this is done, your application can request access to their OA Storage instance(s) and navigate them like traditional hierarchical file systems, requesting to download the files of interest.