

# DuraCloud Architecture

## DuraCloud Storage

The primary DuraCloud application is called DuraStore, this is where storage is managed. DuraStore has a variety of storage adapters, each of which is built to communicate with a specific storage system. All calls to work with content go through a thin mediation layer, to ensure consistency between the storage providers, and then are passed along to the providers themselves to complete the work.

DuraCloud storage begins with spaces. Spaces are the containers into which content is placed. Spaces are also where access control is defined. Once a space has been created, content can be stored there. All content stored through DuraCloud lands first in a primary Storage Provider and is then copied as needed to other providers. Actions that occur with content always take place through the storage REST API (even if this is hidden by client tools or nice UI controls.)

### Storage REST Interface

#### Space Actions

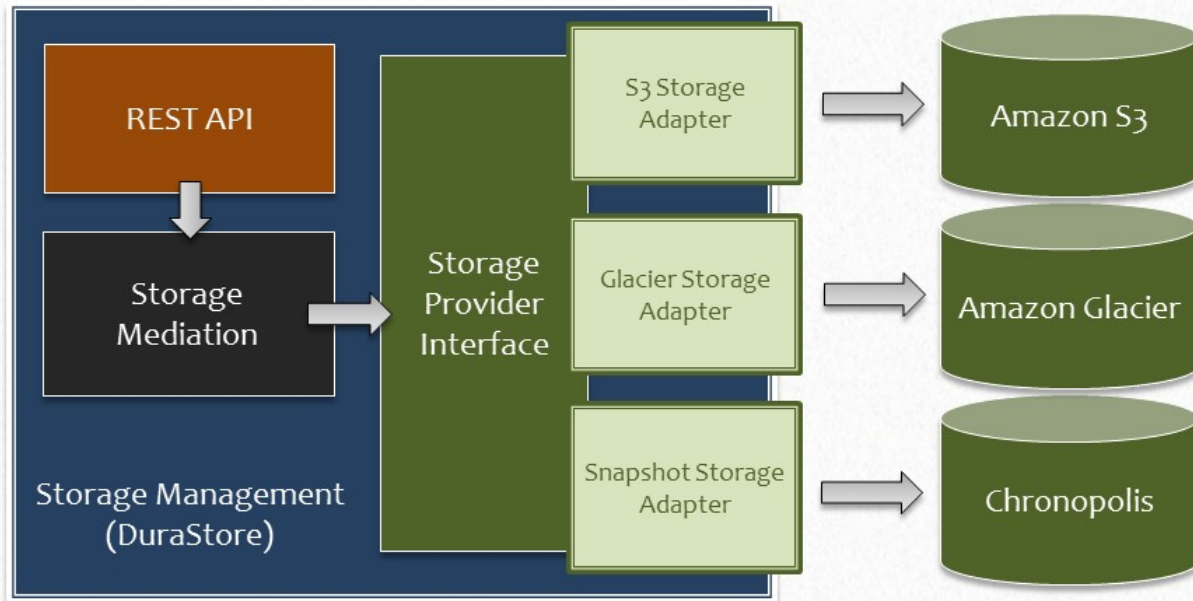
- Add Space
- Get Space Properties
- Get Spaces List
- Get Space Content List
- Get/Set Space Access
- Delete Space

#### Content Actions

- Add Content
- Get/Set Content Properties
- Get Content
- Copy Content
- Delete Content

#### Other Actions

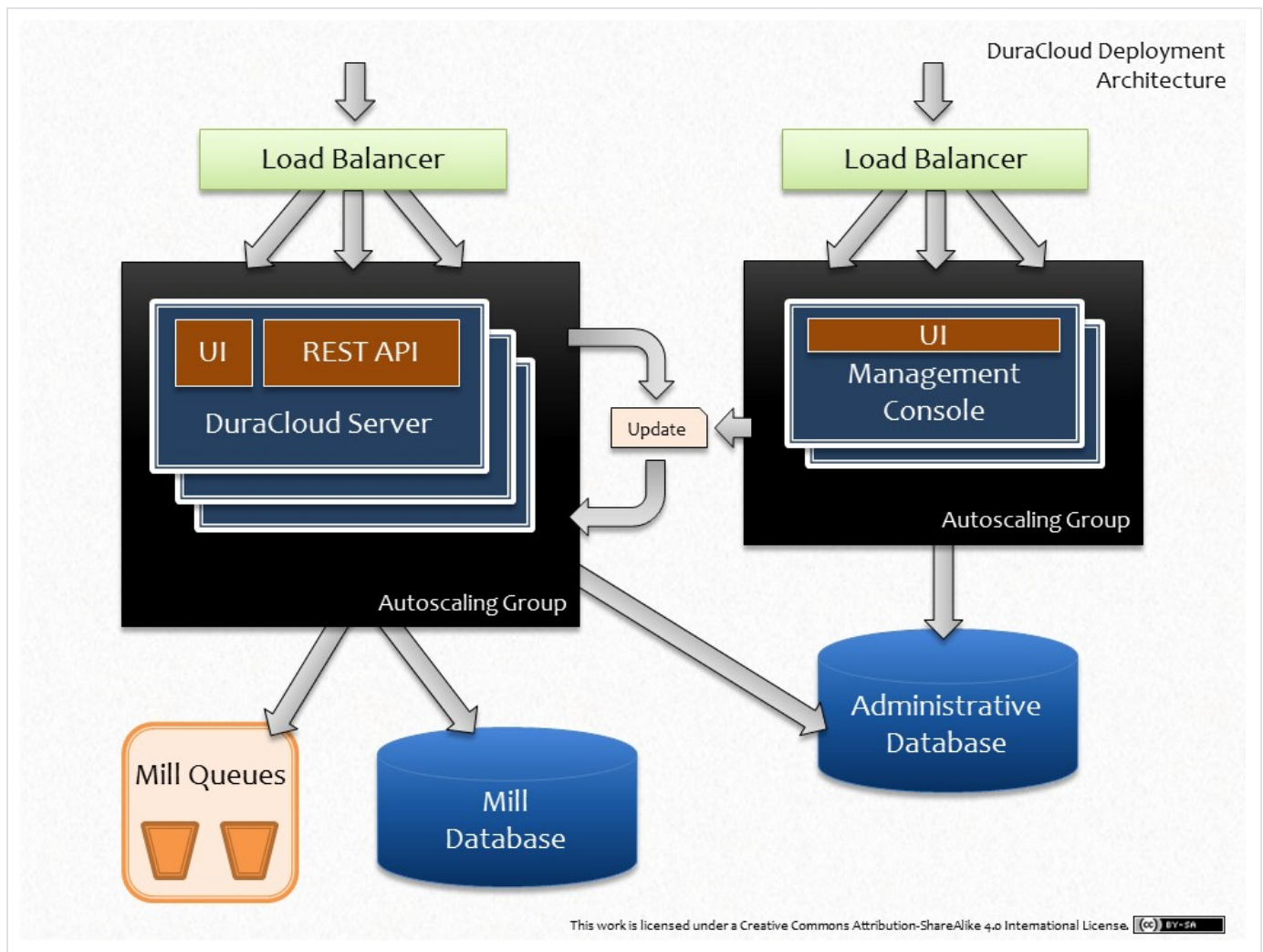
- Get Stores
- Get Tasks List
- Perform Task



## DuraCloud Deployment

The DuraCloud applications are deployed via a cluster of instances each running both the DuraCloud Storage (DuraStore) and DuraCloud UI (DurAdmin) applications. The cluster uses auto-scaling to set the number of servers based on load; in front of the cluster is a load balancer to distribute load between the available servers. The DuraCloud UI application makes calls through the REST API, which in turn communicates with storage providers to manage data. Storage actions are sent to queues that are watched by the DuraCloud Mill for processing. The processed Mill data is stored in the Mill database, which the DuraCloud applications consult to display audit logs, spaces manifests, bit integrity reports, and other results of Mill processing.

Users and accounts in DuraCloud are maintained via the Management Console application. This application also runs in a auto-scaled cluster with a load balancer. The Management Console interacts with a database to update and store account and user information. The DuraCloud storage and UI applications also consult the administrative database to retrieve account and user information. When updates occur in the Management Console, notifications are sent to ensure changes are picked up properly by the DuraCloud applications. Similar notifications are sent by the DuraCloud applications themselves on changes to ACLs and other security settings.



## DuraCloud Mill

The DuraCloud Mill is the back-end system which handles much of the data processing for the DuraCloud system. The Mill makes use of a set of queues where all work tasks that need to be performed are placed. Each DuraCloud instance puts a task in one of the Mill queues every time any storage action occurs. These tasks are then worked, resulting in updates to the audit and manifest details for each space. This allows the Mill to keep track of all content in the DuraCloud system.

A series of workers perform the jobs outlined in the work tasks. These workers scale automatically based on the number of items in each queue, allowing the number of workers to increase and decrease as the amount of work to be done changes. The workers communicate with the storage providers, as well as with a database that is used to maintain system state.

DuraCloud maintains a set of duplication policies which define which spaces should be duplicated between which providers. This information is used by the workers to perform the duplication actions that are needed to keep all content in sync between providers. As a secondary measure to ensure consistency, another part of the Mill system, called the Duplication Producer, looks at the duplication policies, and adds a task for each content item that should be duplicated to a task queue. The workers use these tasks to verify that, indeed, all content is duplicated as it should be.

Another type of producer, the Bit Integrity Producer, is used to perform checks for content bit-level integrity. Much like the Duplication Producer, the Bit Integrity Producer adds a task to the task queue for each content item in the DuraCloud system. The workers then execute those tasks by retrieving the content, and verifying that the checksum computed for the content matches that provided by the storage provider and maintained in the DuraCloud manifest.

The DuraCloud manifest and audit cache are maintained in the Mill database. This database is kept up-to-date by the workers as they process audit tasks. The DuraCloud instance consults the Mill database when manifest or audit information is required. The audit cache maintained in the database is transitioned into long-term file-level audit storage after a short period of time.

The capabilities of the DuraCloud Mill are likely to expand over time, as it is designed to be a generic and scalable task processing engine.

