Authentication and Authorization Integration Road Map

Table of Contents

Overview

This document contains a road map for integrating the authentication and authorization components into the cloudfoundry.com production system. It describes expected standards and processes for code review, testing, packaging, and deployment.

We will describe a series of phases to integrate the User Account and Authentication (UAA) component and the Access Control Manager (ACM) component into cloudfoundry.com, and then describe how those components could also be integrated by the BOSH layer.

Components

There are three components in view for the following road map:

1. the user account and authentication (UAA) service.
2. the access control manager (ACM).
3. a token processing library.

CF Team Prerequisites

API documents for the UAA and ACM, and an interactions document, are up-to-date and need the dashboard review status updated.

The UAA and ACM are implemented as per the API documents and are complete for this milestone.

CF Deployment Process for the UAA and ACM

Prerequisites, standards, and steps to releasing a component in cloudfoundry.

Source code and unit tests:

- source code control and review via gerrit
- unit tests run by jenkins to verify changes before merge
- determine unit test coverage
- add monitoring collector, /varz

Integration with vcap:

- add submodule to vcap
- write chef recipes for components to be deployed by dev_setup
- add appropriate BVTs to vcap-tests
- run BVTs before merge to master
- add stress tests to grinder with QA team
- integrate varz endpoints with dashboard.

BOSH packaging:

- create packages and jobs
- integrate packages and submodule into branch of release repository
- test on dev instance
- If there is data that needs to be persistent, use a disk for the job.
- In the bosh deployment, test a stemcell update scenario - or simulate it by a "bosh recreate <jobname> - to verify that the job control scripts work and that persistent data are not lost during OS updates.
- Document verification plan against dev instance (it doesn't have to be complicated) - some of this is BVTs, but if you believe that there are aspects of your service that isn't covered by BVTs - such that we/you can verify against Staging once we deploy it there.

Release process:

- schedule release on calendar (page on wiki)
- add on-call staff to pager duty (get production access)
- merge release branch to master
- SRE deploys to staging
  - id team manually checks
  - run stress tests
- SRE deploy to prod
  - id team manually checks

Phase 1: UAA endpoints for limited testing, CC database prep

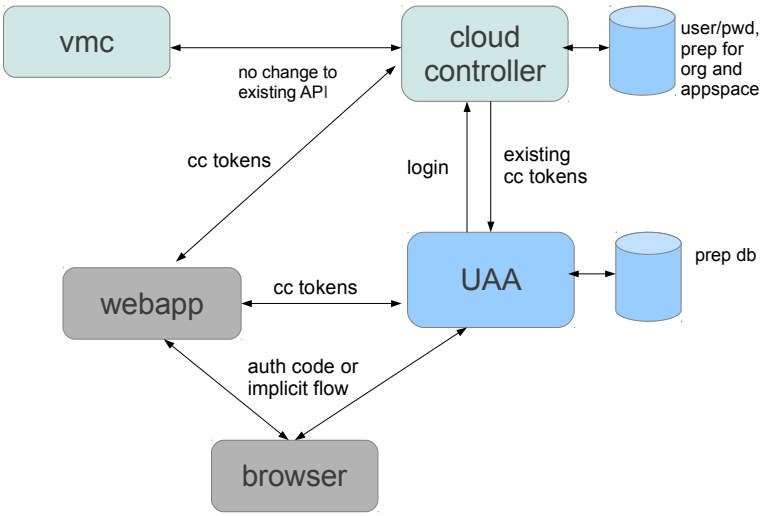
Objective: Get the UAA through the deployment process and operational for traffic in the production environment, and to allow for some preparation of the CC database.

In this phase:

- deploy the UAA through the system
- make changes to the CC database to ease future transition to orgs and appspaces.
- the only new capability would be that UAA endpoints would be available for testing in a

production environment, but the UAA would return CC tokens using existing CC APIs and would hold no passwords.

Target date: Monday, 16 Jan 2012



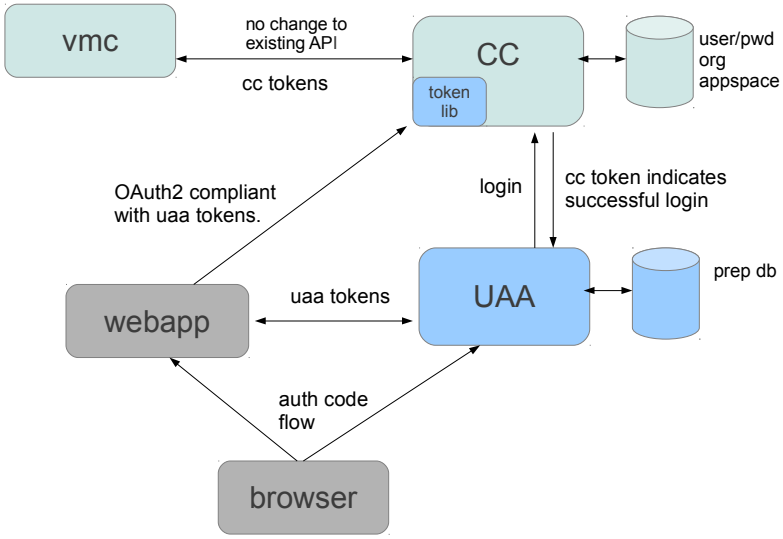
Phase 2: New token issued by UAA, CC accepts old and new

Objective: UAA deployed with full OAuth2 support and new token format.

In this phase:

- User accounts still in CC
- UAA checks password via existing API and issues new token format
- CC integrates token processing library and can accept old or new tokens

Target date: Monday 23 Jan 2012 – one week after phase 1.



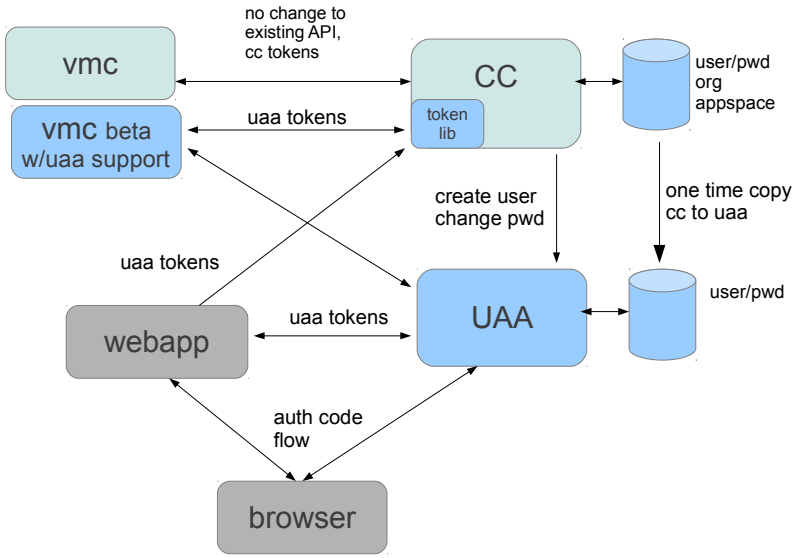
Phase 3: UAA issues tokens from its own DB

Objective: Copy the CC user accounts to the UAA database, and add calls from CC to UAA to support create user, change password.

In this phase:

- User account database copied from CC to UAA.
- UAA authenticates user from its own db and issues new format tokens
- a modified vmc that supports tokens from the UAA could be in limited beta.

Target date: Monday, 6 Feb 2012 – 2 weeks after phase 2.



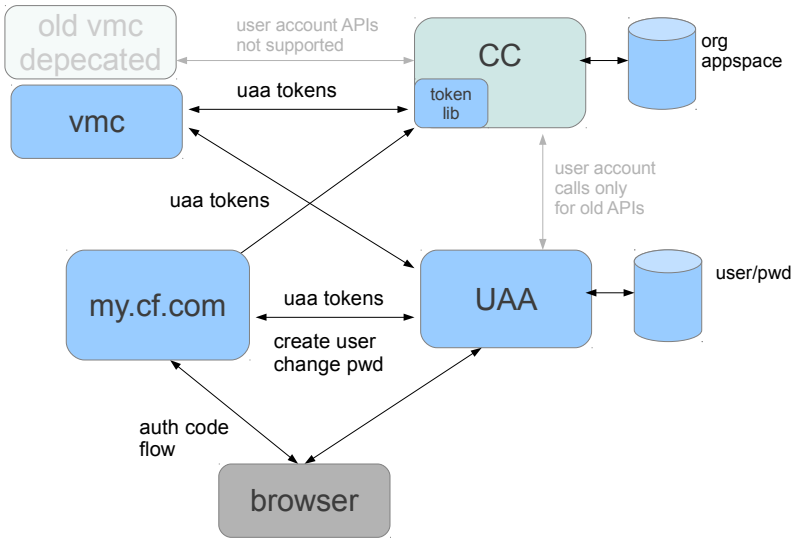
Phase 4: UAA owns users, old API support removed

Objective: remove legacy and transitional pieces, switch vmc and www apps to use uaa apis.

In this phase:

- User accounts can be dropped from cc db
- release vmc which supports the uaa
- my.cloudfoundry.com (aka the www app) should switch to uaa apis for user account creation and management.
- old vmc and apis can be deprecated

Target date: Monday, 13 Feb 2012 – one week after phase 3.



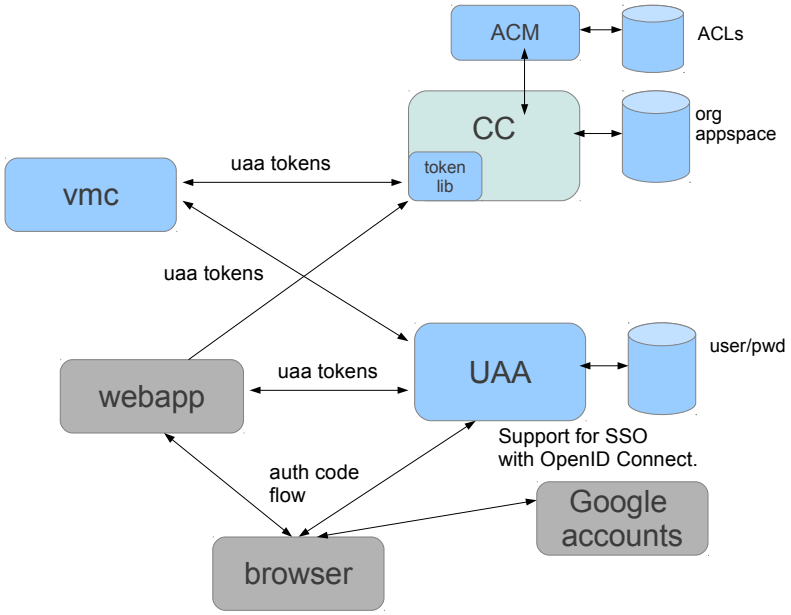
Phase 5: Add ACM to CC, support web SSO

Objective: add ACM to production environment with CC, and add support for web SSO in the UAA.

In this phase:

- CC creates objects and ACLs in the ACM
- CC calls ACM for authorization decisions
- UAA supports SSO with OpenID Connect providers, e.g. Google and Horizon

Target date: Monday, 20 Feb 2012 – one week after phase 4.



BOSH Integration and Deployment

This section refers to the steps involved in integrating the uaa and acm for use by bosh itself, not deployment by bosh into cloudfoundry as is described in the phases above. This is so that a bosh admin making API calls to the bosh director are authorized by uaa tokens and acm permissions.

Integration with bosh should be simplified in that much of the work of creating bosh packages, chef scripts, perhaps some BVTs, etc., are reusable from the CF integration. Also, the user account migration situation is quite simple since there is no existing user account database.

Steps:

1. package and deploy the uaa and acm in bosh (similar work already in progress for CF)
2. update cli and director to go to uaa for authentication and use new token processing library on branch
3. test, review, test, merge
4. test in staging
5. move to production

Approvals Needed External to CF

These can be done in parallel with some of the earlier phases, but need to be done before each component is publicly available.

1. Each component is expected to be released as open source and will need to be properly approved by VMware open source review process.
2. Each component should be submitted to a formal security review.