MOC-UI Project Proposal

Thomas Hsu, William Vangos, Alex Wong, Hang Xu

Vision and Goals of the Project

The Massachusetts Open Cloud UI (MOC-UI) will be the interface seen by users of the MOC. This project expands the goals and progress achieved by the previous MOC-UI development group by:

- Creating a user registration/login system
- Creating a marketplace that allows users to choose between multiple service providers
- Enhancing the existing UI and transforming it into a more usable interface.

Users of the Project

The users of the MOC UI are those who wish to utilize the services of the Massachusetts Open Cloud. These primarily include:

- Member colleges and universities (Boston University, Northeastern University, Harvard University, the Massachusetts Institute of Technology, the University of Massachusetts).
- The Massachusetts Green High Performance Computing Center (MGHPCC).
- Other industry and government partners and the general public.

The UI is geared towards the end-users, and as such simplicity and ease of use are a primary goal.

Scope and Features Of The Project

MOC-UI

- A simple UI that presents a robust yet easy-to-use user experience.
- User Management
    - Administrators will be able to add and remove users.
    - Users will be able to add other users to their projects.
    - Clients will be able to add themselves as users.
- Project Management
    - Adding and removing projects from the system.
    - Resource Management
        - Users will be able to configure their resource needs (processing power, data storage, and networking)
        - Users will be able to manage their stored data
        - Users will be able to manage network connections.
    - User will be able to perform instance management (launching, suspending, stopping) through VNC console.
    - Users will be able to manage high-level resources (Hadoop, Spark, and other appliances)
    - Users will be able to interface with multiple MOC deployments. (Harvard University, Northeastern University, Boston University)
    - The Marketplace
        - Users will be able to purchase resources for their projects a la the Amazon Web Services marketplace.

- Users will be able to purchase resources from different deployments for their projects.
- The Marketplace UI will be able to accommodate and scale to a large number of users.
- The UI will provide an interface for third party services to interact with the UI.

Solution Concept

System Components of the Global Architectural Structure:

- OCX Library: this will be used to authenticate user credentials, create project space, access service directories, and state and credential databases.
- OCXi: user interface for OCX
- OCX-login: login interface for OCX and project lists
- OCX-auth: authentication server for user login
- Service-Directory: maintains service endpoints, including service information retrieved from specific resources. For example, availability, image size, SSH, CPU, price, version, name, description, etc.
- Service Deployment: low and high level services, can be a VM, appliance, or a high-level service
- State-DB: contains all of the project's purchased services
- Service Credential-DB (SC-DB): contains all credentials for project services.
- Project Space: virtual space for users (tenant ui, state db, sc-db)

User credentials are authenticated by OCX-auth, an authorization server. After authentication, the user will create a new project-space, or login to an existing project-space. After this, a Tenant-UI is spawned for the user and performs all interactions. Within the project-space, there are databases for managing project state and service credentials. The OCX Library allows the Tenant-UI to acquire information on all services included with the project-space. The Service-Directory is called by the OCX Library to retrieve endpoints, which are in turn used to acquire information associated with each service included in the project-space. The credentials of included services are stored in the Service-Credential Database and a list of these services are stored in State Database. Both of these databases are managed by the OCX Library. The Tenant-UI uses it's stored endpoints with the API to connect with services. Authentication is required with the central authority to interact with it's services. After this process, the Tenant-UI provides a marketplace of available services with a project-management interface, or an extensible, third party interface to manage interactions between services.

Design Implications and Discussion

Core Concepts and Technique behind the MOC-UI

- **Django:** This web framework will be used to connect end users to their project on the MOC system.
- **Shared Service Directory:** This is part of the larger OCX project and is not the main focus of our project. It is being developed as a new service due to it's ability to cross multiple regions and service providers. It will provide a single service that different providers can advertise their services on, to inform users about services provided by OCX.
- **Project Space uses SC-DB:** Until OpenStack supports federated authorization, this would be a stopgap solution to authenticating credentials across various deployments of the MOC.

Preferably, it is better for a single set of credentials to be used across the deployments of the MOC.

- **Extensible UI:** Rather than tailoring the MOC UI backend to the UIs of the MOC constituent deployments, we will assume that each provider deployment has a compatible UI. For a multi-provider cloud this is necessary to ensure the same experience regardless of which providers are used.

Release Planning

Release #1 (Due by Week 5):

- User login and registration screens.
- Ability to add a user to the MOC system.
    - Ability to register a user with a username, email, and password.
    - Ability to add a user (admin).
- Ability to remove a user (admin).
- Store user info.
- User login/logout
    - Using registered username or email and password.
- User's projects page
    - Displays current projects user is a part of.
    - User selects projects and gains access to project via a VM.
- Successful testing of the user creation system with a dummy account

---

Release #2 (Due by Week 6):

- Research UI

Release #3 (Due by Week 7)

- Will be uploaded in the future by the team members