

Program Management Backlog

as of 9/30/2013 delivery to AIR

The following captures the known outstanding functionality left for Program Management as well as any known remaining technical tasks. The schedule for completion of the backlog is not determined. External dependencies have been noted where given as reason for being in the backlog. Please note that the component cannot be considered finished until this remaining backlog is completed. Scheduling for completion of these tasks will need to be coordinated by all involved stakeholders.

Entity and Entity Type Domain

Currently the Tenant domain object has `Tenant.name` and `Tenant.TenantType` attributes that should really be a relationship to an Entity. The Entity's lifecycle will eventually be owned by the Test Registration component. Once this is completed, the Program Management UI should be adjusted to search Test Registration for Entity Name, rather than allowing free form text entry. Additionally, the `EntityType` is currently an enumerated list owned (and served via a webservice) by Program Management. Once Test Registration is completed, the domain ownership and associated webservice should be removed from Program Management. This is further documented in the 'Future State' portion of the Entity Diagram in the Program Management design documentation.

Authentication/Authorization

This task remains as an open issue until such time the SSO/Authorization mechanism for Smarter Balanced RFP#11 (SB11) is decided upon and delivered. Once user credentials (signed in user) are available to the web application, `PropertyConfigServiceImpl` should be modified to capture the user as part of the Audit. This is noted by the following comment:

```
// TODO will need to add user when we can get it from SSO/session
```

Additionally, based on the following assumptions, the level of effort remaining will be a matter of applying the authentication pattern to the web application that hosts the web services, as well as annotating the web service method with:

```
@Secured(Role="roleGoesHere") org.springframework.security.access.annotation.Secured
```

Assumptions:

- Authentication will be handled by a separate component.
- The authentication method will be supported by Spring Security
- Authentication will be on a component client basis. Example: integrating components will be authenticated and granted authorization (by the security component) in the form of a statically defined role.
- The web service endpoints will be secured (via the `@Secured` annotation) and configured

to be protected by a statically defined role, and improper access will result in a HTTP 401 (unauthenticated) or a HTTP 403 (unauthorized).

- Additional business (non-role) based security rules have not been specified, and will require additional effort if required.

Dependency Configuration

Program Management has an external dependencies configurations (MongoDB) that allow for flexible configuration options. Please see installation instructions for more details on the configurations needed.

Deployment

Fault Tolerance

Program Management is required to be a high availability component in the SB11 Architectural Report. The ability to make a component highly available will be extremely dependent upon the deployment environment. The deployment model for the SB11 has not been finalized, and as such, plans for clustering/and failover can not be finalized for Program Management. Depending upon the clustering method chosen, caching and/or other optimizations for high volume transactions may have to be implemented.

Technical Tasks

There are various technical tasks, and investigation projects that were not considered critical path to required functionality. These tasks have been noted in internal DRC task tracking. If the schedule allows and priorities dictate, these items can be scheduled to be worked on.