1. Appendix
   1. SoW Tasks list

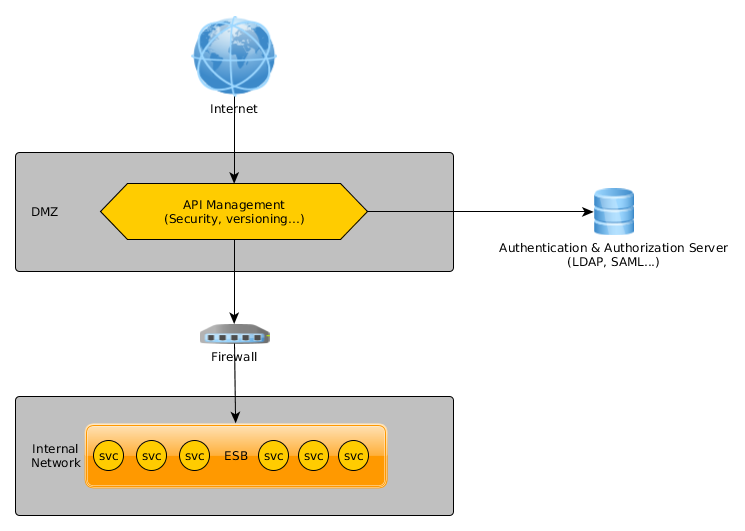
|  |  |  |
| --- | --- | --- |
| **Task** | **Status** | **Comment** |
| Setting up of fuse fabric on single node | DONE | Dev platform |
| Setting up container for easy deployments of interfaces | DONE | Child containers in dev & SSH in prod. There are containers for apps, amq and insight |
| Setting up fuse as default auto boot service | NOT DONE | We recommend to use custom “.service” scripts with systemd, based on the FUSE ./start, ./stop and ./status executable. |
| Configuring security on fuse container | DONE (N/A) | No requirement. Protocol security is to be handled by the applications if required. |
| Configuring container for remote management using ssh or CLI | DONE | OOTB. SSH keys are used to communicate between the hosts. |
| Configuring persistence for local cache, generated-bundles, JBI, maven and txlog | DONE | MySQL is used for the applications. KahaDB/LevelBD are used for the brokers. Artifacts are stored under the Nexus filesystem |
| Configuring the hot deployment system | DONE | Use the maven plugin directly from the workstations |
| Configuring JMX within fuse fabric | DONE | OOTB |
| Configuring JAAS security from wide choice of authentication mechanism | DONE (N/A) | No requirement to secure the Fuse Web console. |
| Configuring fabric’s maven proxy | DONE | Fabric agents are linked to Nexus |
| Securing the jetty http server | DONE (N/A) | As part of the application code. No requirement for SSL. |
| Managing of certificates within fuse fabric environment | DONE (N/A) | SSL is handled and offloaded by F5 |
| Configuring transaction manager in fuse | DONE (N/A) | No transaction Manager on Karaf. Transactions can be handed on the endpoints directly (JMS/JDBC). |
| Securing redhat JBoss A-MQ container | DONE (N/A) | Not required. Brokers are not exposed externaly. |
| Configuring JMX on Jboss A-MQ container | DONE  (N/A) | Leveraging Fabric |
| Configure of message persistence stores. One of the persistence store (kaha DB or level DB) | DONE | LevelDB replication is used in PROD. Another solution is to use the MySQL database.  KahaDB is used in Dev (single node, no HA). |
| Finally fine-tuning of the above environment to meet NFR of the customer | NOT DONE | Mainly Java Heap size and HTTP threads pool. This will depends on the workload so need to have this evaluated first. |

* 1. API Management concept

When exposing ESB services to the external world (internet), it's recommended to use an API gateway to separate the technical aspect of the flow (security, authentication & authorization) with the funcitonal aspect (core logic of the ESB applications).

At best this API management component should be located behind a firewall in the DMZ network.

The concept is represented in the diagram below.



Unfortunately, Red Hat doesn't have any Open Source API Management product at the moment.

API management principle: