

**High Level Design Document**

Search

**Document Version: Ver. 0.0.1**

**Last Update: March-18-2021**

Revision History

Please keep the latest version on top

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ver.** | **Change Description** | **Sections** | **Date** | **Author** | **Reviewer** |
| 1.0 | ALL | ALL | May-23 | Bipin | Bipin |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

References

Reader should read this document in conjunction with the following documents

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Document Name** | **Ver** | **Location** |
|  |  |  |  |

Definitions, Abbreviations and Acronyms

|  |  |
| --- | --- |
| LLD | Low Level Design |
| BRD | Business Requirement Document |
| UDO | Universal Data Object |

Table of Contents

[Revision History 2](#_Toc68110026)

[References 2](#_Toc68110027)

[Definitions, Abbreviations and Acronyms 2](#_Toc68110028)

[**1.** **Environments** 5](#_Toc68110029)

[**2.** **Branching & Merging** 5](#_Toc68110030)

[**3.** **Deployments in Multiple Environments** 5](#_Toc68110031)

[**4.** 5](#_Toc68110032)

**Introduction**:

The document contains detailed requirements and high-level design for integrating payment gateway for various Dyson markets

**Document Reference:**

* Magento2\_ Application Handbook\_v0.3

**Stakeholders**

* Business Owners:<Add name here/else add Unknown >
* IT Owner: <Add name here/else add Unknown >
* Business Analyst Mindtree: <Add name here/else add Unknown >
* Tech design lead – Mindtree: <Add name here>

**1. Environments:** The Software development cycle typically consist of 4 to 5 environments. These environments are Local, Staging, Sandbox, Pre-Production and Production. The purpose of each environment is as follows:

1.1) Local – Vagrant environment where initial development and initial testing happens.

1.2) Sandbox – Digital ocean environment where first deploy and initial QA takes place if feature is market-wide.

1.3) Staging – AWS environment where all QA, end to end testing and sign off/approval takes place here.

1.4) Pre-Production – Digital ocean environment that mirrors Production in terms of codebase where final smoke test takes place.

1.5) Production - public AWS environment where the final deployment for the market or group of markets happens.

**Requirements for the environment setup:**

* AWS account
* Digital Ocean Account
* Vagrant
* Ansible

**2.1 Branching Strategy:** During the development process, both site and module repositories can be updated depending upon the requirement. Sometimes change might be required in India’s site repository say as it has to deal with payment charge things and sometimes it could be a module change say Dyson CMS just needs to update and in other times, we might need to update both site and module repositories. This process is followed by new feature work as well as fixes for support tickets. Each module has develop/xx branch per market and one for sandbox e.g. for India the branch is develop/in. This allows us to merge changes to sites at different times without interruption of the other sites. Each market has develop, release\_candidate and main branch for various phases and environments. The develop branch is for sandbox and staging environments whereas release\_candidate is for Pre-Production environment. Branch names are kept consistent across site and module repositories for easy identification. For e.g. feature/1234-new-feature will be the name of the feature branch for market and module repositories.

**Requirements for Branching Strategy**:

* Vagrant
* Bitbucket Dyson Vagrant repository

**2.2** **Merging Strategy for different environments**:

**2.2.1** **Sandbox**: Sandbox is used to test the features before they’re rolled out to market staging sites. When work is initially complete on local environment, assuming the feature is market-wide, the first step will be to compile the work ready for sandbox. There is no “production” branch or “main” branch for sandbox because there’s no production site as sandbox so we always use the develop branch and merge our feature branch into it:



If any modules are worked in, these also need to be merged into develop/sandbox branch of those modules:



2.2.2 **Staging**: Staging environment will be the first merge to get it to the specific market. Either way, the steps are the same. If the feature is for all markets, we will group markets by certain criteria and deploy to a selected market candidate from each group first. E.g. Magento Commerce sites, Right to Left (RTL) markets etc. By doing this, if any bugs are found on the initial candidates, we can resolve before rolling out to the remaining markets and save time on compiling and deployments. Merging a feature branch into the market repository’s develop branch:



For the module work, merging a feature branch into develop/xx branch for that module. 

2.2.3 **Pre-Production:** When everything goes well in staging environment the next environment is Pre Production where we do final internal smoke tests. Each market has a Pre-Production site which sits on a Digital Ocean droplet. Each market has a release\_candidate branch which is an exact copy of the main branch of the market, and used for deploying to their Pre-Production environment. So this is where our market feature branch is merged into:



For modules, this is the point where we create a new shippable version. To do this we merge the feature branch into the latest main branch of the module: 

We then tag the commit according to semantic versioning and use composer to “require” the latest version of the module:



**2.2.4 Production**: Everything has been merged into the release\_candidate branch in the Pre-Production environment. All the composer changes and feature branches merges into the release\_candidate branch will become the new shippable version of market and we finally merge our release\_candidate branch to ‘main’ branch for the deployment:



**Requirements for Merging Strategy**:

* Vagrant
* Bitbucket Dyson Vagrant repository

**3. Deployments in Multiple Environments:** Irrespective of deployment to various environments, we always use the same tools and commands which are written in ansible. Dyson-cli is a bespoke wrapper for ansible playbooks that semi-automate the config of environments and deployment of code. These will always be executed by a developer from their local or remote vagrant which sits inside a vagrant machine, and uses their ssh-key and the AWS cli to run commands. Dyson CLI commands can be re-run without damage to an environment (e.g. site:setup command can be re-used to enable/disable basic auth). These commands uses the vars and state of in dyson-vagrant group\_vars where we have all our configurations for e.g. branch names, if branch name is equal to release\_candidate for the Pre-Production config etc. If there is an emergency of deploying to a different branch to your staging or pre-production environment. We don’t need to go for merging of newly created branch to develop or release\_candidate branch. We can map a new branch name with branch name in dyson-vagrant group-vars for the deployment without altering the common environments. Usage is dyson <command> <environment> <site-code> (can use the same commands on dev, staging, pre-prod, prod). Some of the frequently used commands are:

* dyson provision - installs all prerequisites to an environment (php modules, node, local services: mysql, redis, etc)
* dyson site:setup - configures market-specific settings for the environment (e.g. nginx and postfix configs)
* dyson site:db:pull/push - pull or push databases from/to remote and local environment.

Note, you can only ‘push’ to a Pre-Production environment on Digital Ocean and we can pull from Production environment.

* dyson site:uploads:pull/push - pull or push shared files from/to remote and local environment
* dyson site:admin/frontend:ip - display the IP addresses for admin or frontend EC2 instances

The above mentioned Dyson CLI commands will run the ansible playbooks and those playbooks were kept in Dyson vagrant repository. For e.g. Deployment command of Dyson CLI in Production environment for Australian market is: dyson deploy production aws-au

**Requirements for Deployments**:

* Vagrant
* Bitbucket Dyson Vagrant repository
* Dyson CLI
* Dyson Commands