**Top Level Magento Infrastructure/Architecture**

**Architecture Overview:**

* Magento Shop Fronts (AWS & Digital Ocean): Magento Shop Fronts in AWS can be accessed always via Akamai and AWS load balancer - all ports are closed except 80/443.Anyone in any region/country can access Magento Shop Fronts. we do HTTP basic auth during staging sign on with username and
* Magento Admin (AWS & Digital Ocean): It is mostly accessed through Akamai through AWS Load Balancer. Only G&V Engineers and approved Dyson staff can have admin access. we can have different passwords for staging, pre-pro and prod and environments are linked.
* SSH (AWS): All G&V Engineers have SSH access. This uses Unique developer users/keys. passphrase is stored in G&V password vault and only available to most senior personnel
* d. SSH (Digital Ocean): They have shared user and key between developers
* Console(AWS):. individual accounts are created per developer and MFA protected. All G&V, Rackspace engineers have access to console and no one at Dyson

**Local Dev Architecture**: Inside Mac (ex: IP: 192.168.0.12) we have Vagrant VM running Ubuntu 16 (ex: 192.168.33.99). Here actual repositories will be on MAC and vagrants NFS to it and here if we destroy the vagrant it won’t remove repositories. MariaDb will also be on local developer site and by destroying vagrant we destroy database. Provision, setup, deploy management is done by Dyson CLI and then SSH to remote environments

**Digital Ocean Architecture**: Digital Ocean Droplets are individual cloud servers. we run one with many virtual hosts as we need and they normally have subdomains. We do not have load balancers or firewall in place. By using basic http auth it is protected from general access. we can't transfer account and domain so we snapshot the droplet and another digital ocean account. All the sites here will have new domain's

**AWS Architecture:** Customers connect ec2 auto-scaling group through EC2 ASG Magento - KR C5.xlarge. The security group for Auto-scaling group in the RDS security group connected to database. We authenticate using Azure MFA , we name instances based on the country code. Shared components accessed via security groups (sg-Redis , sg-RDS , sg-FTP , sg-NFS). We C&V/Dyson Vagrant Developer Environment with tooling for provision, setup and development automation. Access controlled by Key based auth and IAM Roles. We have ECC Remote System we frim (stock and Shipment updates). Dropbox shipment Requests shipment and stock updates

**Admin Pages Customization**: The main areas of this admin pages’ customization can be very well done in Categories and Products. Otherwise its mostly new modules that will create new part of magento admin. Create new menu items and new admin pages to keep themselves quite contained. These customizations will happen over Stores configuration, Dyson CMS and also some new sections under existing menu sections. Customization of categories and products have a number of attributes and attribute groups. Those are created programmatically and not in admin or in database.If the attributes are specific to Dyson and products, but perhaps not a specific piece of functionality , they are managed/created by Module Leap. If the attributes are specific to a module or a piece of functionality, they will be created or governed by their module

**Custom modules**: The custom modules are already covered. Custom modules can extend the Magento admin UI in several areas like side menu, sections and links inside flyout menu, Adding to existing, or making new items, in the accordion menus in an admin sub section. The only additions to the slide menu at a root level are for very signiﬁcant modules – e.g. Dyson CMS and GDPR Cleanser. Catalog has had it’s ﬂyout extended to include the Legacy Product Import module. Marketing has two items added under Dyson Communications, this is for New Product Notiﬁcations and Promo Code Sign-ups. Stores Conﬁguration has had many sections added to it but we have always tried to scope these as the vendor Dyson and have a new section for each custom modules options

**UI Elements:** We rarely, if ever, create custom/new UI components when working in the Magento admin. We do things in the standard Magento way. Menu changes and routes are by the module they relate to in module-name/etc/adminhtml/menu.xml and routes.xml. Controllers are in the module they relate to in module-name/Controller/Adminhtml/. Actual UI components are in modules-name/view/adminhtml/, including Layout XMLs,Occasionally phtml templates,and some more XML layouts in ui\_component

**API Endpoints Conﬁguration and Customization**: For APIs, an endpoint can include a URL of a server or service. Each endpoint is the location from which APIs can access the resources they need to carry out their function. The following are the key API Endpoints that we actively program against and maintain code for in order to consume

**Engage (CRM for Multiple Directs**): Engage API is a RESTful API for retrieving and managing data in Dyson site.Sends Shipment Requests for completed and paid orders only. Speciﬁcally in vendor/dyson/xml-back-ofﬁce/Controller/ShipmentRequest/ApiShipmentRequest.php. Ln 328 sendShipmentRequestData() within uses CURL to built the call and actually call the engage API

Module Dyson Product Registration will be sending to EngageSpeciﬁcally in vendor/dyson/module-product-registration/Cron/Engage.php. Ln 167 sendToEngage() within uses CURL as above. Global Engage API conﬁg is managed in Ansible Group Vars in the Dyson Vagrant on a per site basis.Ansible injects the Engage API conﬁg if present into the market’s env.php. Speciﬁcally see dyson-vagrant/ansible/group\_vars/tag\_Environment\_Production/vars.yml. Ln 164 is TR’s Groups\_vars, Ln 197 shows backofﬁce c6rd4 edx cf56y7hbu890onﬁg, Ln 202 has the Engage API conﬁg deﬁned within

**Oracle (CRM for HK on Distributor Module)**: The Oracle API Manager facilitates the creation of APIs that expose the functionality of backend systems(vendor/dyson/oracle-back-ofﬁce) and services. Sends Shipment Requests for completed and paid orders only in vendor/dyson/oracle-back-ofﬁce/Cron/ SendOrder.php. Requests Shipment Updates in vendor/dyson/oracle-back-ofﬁce/Cron/ UpdateDelivery.php. Requests Stock Updates in vendor/dyson/oracle-back-ofﬁce/Cron/ StockUpdate.php. Oracle API Conﬁg is in Stores > Conﬁguration > Dyson > Oracle Back Ofﬁce

**Sorted Pro (Scheduled delivery and click & collect on AU and PL):** Code wise, Sorted Pro API is only consumed by Module Dyson Sorted Pro by the AU and PL markets. You cannot just deploy Sorted Pro to a new market and expect it to start consuming the Sorted API and work.Market must be conﬁgured in Sorted Pro.The Module will almost certainly need some work to ﬁt it into and around the market speciﬁc checkout conﬁguration, custom ﬁelds, and form validation rules API helper class managed connection and requests in vendor/dyson/module-sortedpro/Helper/Api.php. Click & Collect related calls are managed in vendor/dyson/module-sortedpro/Controller/Api/ClickAndCollect.php. Scheduled Delivery related calls are managed in vendor/dyson/module-sortedpro/Controller/Api/ScheduledDelivery.php, API conﬁg can be found in Stores > Conﬁguration > Sales > Shipping Methods > Sorted Pro

**10 to 8 (IN In Home Demo):** Deep dive on this module on hold until we establish whether it’s been withdrawn or not. Others managed by third-party extensions/libraries that we consume transparently (and won’t be covering). Payments (Many/misc),Riskiﬁed (Fraud on some direct markets), Many of our Javascript supplied integrations are probably using an API

**.Net Application and Usage:** The .Net application being referenced is assumed to mean SiteCore, managed by Valtech. we have iframed in the UI/functionality from SiteCore. However,functionality that integrations with other Dyson systems exists in SiteCore, but the same functionality is still in the Magento backlog. Magento CMS pages have an option to show an iframe of content instead of content, but this is generally not used anymore due to Dyson CMS

**Unit & Integration Test Architecture:** As of now, there is no unit testing being carried out. All testing, functional and integration is done manually. More info regarding automated smoke testing will be covered in later sessions