Cache Configuration: its built in layers as we go further down through layers we start find ALL markets. Currently we are using built in Magento Page Cache and we store server managed in Redis in all markets will be using Redis in all environments. We install Redis on local host in Dev and on Digital Ocean/Prep-prod in all markets. In AWS we use Elastic Cache instances for Redis. We run cluster is RDS for high availability basically we have replica's here for any failures. There is not WAF/CDN/HTML cache in any front Digital Ocean. We do have Akamai WAF/CDN in front of load balancers, Redis in AWS. HTML cache is enabled in Akamai but not in all markets. G&V custom cache module enable HTML cache turned on Redis without Varnish. We need to deploy Varnish between Nginx and load balancer and get all configured. We effectively cherry picked the functionality of Magento varnish integration we need so that we can get good HTML Cache headers to start CDN functionality properly. Varnish is built in Magento but not part of tooling and provisioning. Currently we are moving from G&V custom cache module to varnish using Ansible

Magento Page Cache: it is configured identical across all markets. It is setup in system -> tools ->cache management -> Flush Magento Cache to clear all caches. Full access should have enabled only for administrators. Main configuration is in configuration > advanced > system > full page cache

Note: Running Magento cache does not clear all sessions/baskets. Flushing java script/css we will lose the deployed java script and flushing catalog image cache we will lose printed media images.

Redis:

* We use Redis in local host in dev and pre-prod and elastic cache instances are used in standing and prod on AWS. These run cluster, primary and replica for high availability. There are 3 cache databases per install in Magento

a) Default: System Cache

b) Page\_Cache: full page cache objects

c) Session: individual custom session.

* We always have 15 Redis instance databases in a cluster. 15 databases divided by 3 per market means up to 5 markets can share a Redis cluster. In Ansible group\_vars we are group markets in order by which cluster they share so it will be easier to navigate. By looking at DNS name from AWS in Ansible group vars we can tell which market is in which Redis cluster

Redis Capacity: When Redis is overloaded. Redis algorithms perform evictions based on the least frequently accessed oldest data. And initially RAM gets fill's first not the CPU. By deploying Varnish in every market or using G&V custom module we can cut our Redis usage to <90% of traffic would be served by Akamai CDN and we can reduce cache regularly.

Redis Dev/Prep-prod: we have Ubuntu 16.04 running in vagrant or Digital Ocean and also Maria dB, Ngnix, Redis running in our local host and we have no WAF, load balancer or CDN

Redis Staging/Prod AWS: we have AWS VPC for prod/staging and Magento ASG , Load balancer, Redis Elastic cache cluster and RDS cluster. User request goes to Akamai WAF/CDN (HTML Cache is enabled based on markets) then to Load balancer into EC2 ASG and servers the page request from Cache

HTML Cache / Akamai: we got Magento Core and id G&V custom cache module is installed Akamai is explicitly set to not cache HTML objects:

* Redis full page cache is used and cache headers not set / modified
* Request originates from Akamai
* Request page served from Magento Cache

We got Magento Core and id G&V custom cache module is installed Akamai is explicitly set to cache HTML objects unless Magento is set no-cache header:

* Redis full page cache is used and cache headers set / modified.
* Request originates from Akamai
* Request page served from CDN

G&V custom cache: Repo: vendor/dyson/module-custom-cache-config.

* if Varnish is not being used Magento doesn't send good cacheable headers and treat URLs with unique strings as unique cache objects, even if it’s needed or not.

Module Functionality:

* It used plugin's to overwrite the cache headers set by Magento and other plugin to override process Magento decides by checking if it new cache object or not
* if page is set to "cacheable = false" at a layout xml level, then it isn’t even picked by magento's full page cache process anyway
* module provides an admin area to explicitly disallow URLs from the cache calculations if required

G&V Custom Cache Module Flow vs Normal flow:

Normal flow: it has no custom cache module

* if URLs requested is not cacheable no caching possible server page
* if URLs requested is cacheable hash requested page into cache object if hash already exists serve page straight from Redis cache
* if URLs requested is cacheable hash requested page into cache object if hash doesn’t exists create new cache object in Redis, finish processing request and server new page

Custom Cache Module:

* if URLs requested is not cacheable no caching possible server page
* if URLs requested is cacheable custom cache plug in checks if URL is cacheable and not in disallow list, then strips querying from URL and hash requested page into a cache object. if hash doesn’t exists create new cache object in Redis, finish processing request and server new page, Custom cache plugin’s create a new etag for the cacheable object and overwrites HTTP cache headers "dyson/module-custom-cache-config/Plugin/IdentifierPlugin.php-afterGetValue()"
* If URLs requested is cacheable custom cache plug in checks if URL is cacheable and not in disallow list, then strips querying from URL and hash requested page into a cache object. If hash exists it servers page straight from the Redis cache.

Dyson Site Feature Matrix: This matrix tells us about all the Dyson site features across various countries. Some of those features are Magento version, payment gateway, php version, Risk Field, Dual language Back Office SAP, Back Office Engage and so on. For each of this features the value for the respective country is filled across the matrix.

For Example, in case of Payment gateway, in case of Australia (AU) its ADYEN and in case Of India (IN) is PAYU likewise for the rest of the countries. In this way the all site feature matrix contains fields of various countries covered across multiple features.

* Magento Version: It tells about the Magento version that the site is running on across various countries
* Payment Gateway: It will be discussed in detail in payments session. If we see, in the all site feature matrix, Australia is using ADYEN, and UAE is using PAYFORT and so on.
* PHP Version: It tells about the php version that the Magento is running on across various countries
* Requires Translation: It tells about the markets that’s need translation. So it is basically not in English. Example Arabic in case of UAE, Hungarian in case of Hungary, Czech in case of Czech and so on.
* Dual Language: Dual Language deals with the multilingual website with true native translations are considerable across various countries.
* RTL: It tells us that the site is right to left enable. For example in case dyson UAE site, by default it is in English, but when we translate it to Arabic native language its right to left enabled.
* Riskified: Riskified plugins are used for fault detection by couple of markets(Australia and New Zealand and Turkey)
* Installments 3rd party: This is where those payments and installments are available.
* Back Office Engage: It tells whether the back office is using engage.
* Back Office SAP: It tells whether the back office is using SAP.
* Back Office Other: Specifies whether it uses another back office.
* Distributor: Specifies whether it is
* Single Page checkout – NEW: It deals with the big new update. Countries like Australia work on this newer versions.
* Single Page checkout – OLD Countries like Hong Kong work on the older existing versions.
* Bespoke/ Extra T&Cs
* Bold Chat: Bold Chat is provider of live chat services. It specifies whether the countries across the all site feature matrix has this feature or not.
* GDPR module: It’s almost live on all countries except Romania and Hungary.
* Soft Bundle: Dyson has come with Soft Bundle functionality as a single unit i.e., wrapping up of products where multiple products/ items can be added into the basket (Customized).
* Promo code / Notify me
* Pin code Lookup: It has been India specific functionality.
* In-home Demo

Note: will be discussed in detail in further sessions

* Sorted Pro

Note: will be covered in detail in upcoming sessions. It’s available in Australia and Poland.

* Editable Newsletters: Only in Australia region they can edit newsletters from the admin.
* DSC information: DSC information selling contract is turkey specific legislation.
* Product Debossing: It deals with debossing the customer’s message on to leather cases. There is no market using it and are not updated.
* Perfect Product Pages: In Perfect Product Pages, more flexibility and rich functionalities are introduced.
* Sticky Promo Banner: Specifies Sticky banner in product page.
* CMS Enhancements: It will be Adding new product types
* Popular Search Pages: Every time the customer clicks search he will be shown popular searches. These are not popular searches though, but are configured in search items.
* Queue-IT JavaScript: That was a situation that turns the queue on and off. It’s the java script into the code in the git system.
* Custom Cache Config: Custom Cache Config which is the caching module.