Node/npm versions/legacy: The Dyson vagrant ships with Node10 and NPM6. It is recommended using Homebrew and install same versions to install Node and NPM on Mac hardware. If we exceed node version 10 frontools won’t be able to run the gulp file and frontend styles can’t be build etc. Node/npm performance in the virtual machine is slower compared to the Node/npm performance directly on Mac as it’s nearly twice as fast. Upgrading the versions of Node/npm will affect the:

* Updating Node/Npm
* Updating Snow dog frontools
* Fixing Dyson Theme leap Sass errors

Ease vs Performance in Dev: Standard environment setup documentation directs you to run npm install in the vagrant. Backend developers can only run gulp styles after setting up new site or pull updated release. Frontend developers use the gulp dev task frequently are going to run gulp and Sass directly on Mac hardware. Below are the some of the instructions.

* Don’t install the Node and NPM directly onto Mac
* Install Homebrew first (package management of Mac)
* Install Node/npm via Homebrew and force it to Node10
* Run the individual websites ‘npm install’ on the Mac not in Vagrant
* Sass will be compiled for Mac binary not Linux

Snowdog Frontools Recap: Magento uses Grunt as its frontend build tool and Snowdog frontools are publically available. Magento module will let you use Gulp and Sass.Snowdog frontools were installed in [site\_dir]/vendor/snowdog/frontools but each site repo ships with a symlink to the folder via [site\_dir]/tools

* Gulp styles – it’ll perform a onetime build of frontend CSS
* Gulp dev- it’ll performs one time build and then watches for changes and rebuilds as needed
* Public repository https://github.com/SnowdogApps/magento2-frontools

Note: we need to manually run the gulp styles and gulp dev tasks in the local environment whereas deploying to a remote environment will automatically run gulp styles

Dyson Theme Leap Overview: Dyson Theme Leap is built using naming conventions from BEM. Theme was built in isolation from OXP’s, and it used reverse Nginx and visually in browser. OXP also appears to use BEM naming in its classes and CSS and G&V was using BEM prior to Dyson. Dyson currently carrying 3 forks of Dyson Theme Leap, one for Magento 2.2 Open source, one for 2.3 commerce and an experimental for 2.4, these need to be refactored and merged back together. We have two types of themes in Dyson. a) Dyson Theme Leap b) Dyson Theme V5

Theme Hierarchy: If we are edited the main theme and our changes aren’t reflecting, it’s because there’s a child theme version of a file we're editing it. All Dyson sites bar ZA are using the Dyson Theme. It Installs via Composer. Market specific overrides were made in a child theme but we’ve been trying to limit this overtime as they can create technical debt against the main theme

Standard theme components: Some of the standard theme components are listed as follows:

* Layout XML’s: its path is "vendor/dyson/dyson-theme-leap/Magento\_Theme/layout/default.xml" All pages use the default xml layout as the Dyson theme is essentially single column across all pages Custom modules make themselves available by adding to other module layout XML’s e.g.: Dyson Badge adds itself to "catalog\_product\_view.xml in vendor/dyson/module-badge/view/frontend/layout/".
* .phtml – Standard Magento template files, used for almost everything except the checkout
  + - Modules can bring their own e.g.: Dyson Badge = vendor/dyson/module-badge/view/frontend/templates/badge.phtml
    - Magento core ones can be overridden in Dyson theme leap e.g.: the product final price template is overwritten in vendor/dyson/dyson-theme-leap/Magento\_Catalog/templates/product/price/final\_price.phtml
* Html – As above but used by Knockout areas of the site instead of PHP

Dyson Theme v5: Dyson Theme v5 only runs on Dyson ZA .Majority of the custom modules are for leap theme only not for v5 theme. Dyson Theme v5 is compatible with Dyson Vagrant and Dyson CLI etc. It runs on a shop. Dyson subdomain instead of www.dyson

CSS Methodology: For the naming conventions, Dyson CSS and Class name follows a frontend methodology called BEM, BEM means things are named sensibly but are easy to find. BEM stands for Block, Element and Modifier. Blocks are tangible things or modules e.g.: button, header, trade-up card. Elements are the stuff that made the block. Modifiers are variations on blocks or elements:

* Block=.button (the wrapper of a button)
* Element=.button\_text (the text or label inside a button)
* Modifier=.button-wide (a button that is full-width)

We identify blocks and elements to build out as classes and we style only them. Cascading isn’t automatically bad, but in a big code-base with multiple markets and developers, we give some benefits of cascade in favor of having CSS more scoped

Finding the CSS for .card\_perks?

By using BEM naming convention, we can find block called ‘.card’, and Sass file called \_card.scss. Developers will mention blocks and file names while creating new blocks. Before going for \_card.scss file, we need some more folders to navigate:

* Blocks – It’s a Magento core terminology, things will be copied from Snowdog/blank theme
* Core – these are the fundamental of fonts, themes and colors
* Layouts – these are biggest things in the theme, often collection of Blocks that will make up bigger things like header, footer and navigation
* Mixins – carried from Snowdog/Blank theme, just ignore
* Modules – these are intermediate sized things and are mostly tangible blocks e.g.: think button, badge and rating the card
* Top – we need to Ignore "Top" folder
* Vendor – namely grids and css reset