Technical Design (TD) - KCP Payment Gate away Korea: NHN KCP (https://kcp.co.kr/) is the payment gateway in Korea. It handles all Magento 2 web transactions in Korea. The integration utilizes a Magento extension provided by Seoul Web Design. The provider of the extension was changed from Eguana Commerce to Seoul Web Design due to issues with transaction processing and the lack of ability to cancel transactions. Some of the useful links were:

* Dyson Korea: https://www.dyson.co.kr/
* Magento Extension: https://www.seoulwebdesign.com/product/kcp-korea-magento-2/
* Codebase (request permission from link for access):https://drive.google.com/file/d/14xBs\_xo-20wpcvHxSYl4MIXHujOdQw7N/view?usp=sharing\_eil&ts=5f475c9f - Connect to preview
* Test Payment Details: https://cyclones.atlassian.net/wiki/spaces/DR/pages/1732511790/RoW+-+Test+Payment+Details

Functional flow:

* Add product to basket
* View basket
* Proceed to checkout
* Enter email, name, address and phone no.
* Proceed to payment
* Choose payment method
* Enter account transfer payment details in KCP hosted lightbox
* Choose card provider in KCP hosted lightbox
* Enter card payment details in card issuer hosted lightbox (e.g. Hyundai Card)
* Confirm payment in KCP hosted lightbox
* Proceed to Magento 2 order confirmation screen

Technical flow: The KCP Magento extension is provided by Seoul Web Design (the developer) as a standard plugin which is purchased ‘off the shelf’. The extension is installed on the Magento server via the standard package management system, composer. This results in the traffic being limited to the browser, Magento server, KCP server and the financial institutes. There is no traffic which passes through Seoul Web Design servers. The KCP logical steps were as follows:

* Customer selects payment method on the checkout page and proceeds to payment
* Magento customer to the KCP hosted payment page
* KCP request certificate from financial institute
* Financial institute return certificate to KCP
* Payment data is encrypted by KCP using key specific to market/channel and sent to Magento.
* Magento uses KCP provided pp cli hub to communicate with KCP
* KCP makes payment request to financial institute
* Financial institute returns payment result to Magento
* Magento redirects customer to order confirmation page on success.

Technical Design (TD) - Adyen - Australia, New Zealand, Poland, Turkey, and Romania: Ayden (https://www.adyen.com) is a payment gateway option on Australia, New Zealand, Poland, Turkey and Romania. The integration utilizes a Magento extension provided by Adyen. On Turkey, Adyen is used as a wrapper to the Turkish payment gateway Iyzico (https://www.iyzico.com/). Some of the useful links were:

* Dyson Australia: https://www.dyson.com.au
* Dyson New Zealand: https://www.dyson.co.nz
* Dyson Poland: https://www.dyson.pl
* Dyson Turkey: https://www.dyson.com.tr
* Dyson Romania: https://www.dyson.com.ro (Est live March '21)
* Magento Extension (public repository): https://github.com/Adyen/adyen-magento2
* Test Payment Details: https://cyclones.atlassian.net/wiki/spaces/DR/pages/1732511790/RoW+-+Test+Payment+Details

Functional flow:

* Add product to basket
* View basket
* Proceed to checkout – shipping details
* Proceed to checkout – payment step
* Credit card fields are encapsulated in an iframe hosted by Adyen servers.
* Return to Magento server on Order confirmation page

Technical flow: The Adyen Magento 2 extension is provided by via public repository. The extension is installed on the Magento server via the standard package management system, composer. This results in the traffic being limited to the browser, Magento server, Adyen server and the financial institutes. The Adyen logical steps were as follows:

* Customer proceeds to pay on the checkout page and enters payment information
* Magento Adyen Plugin request secure iframe from Adyen API
* Customer enters card details into the secure iframe, thinking it’s part of Magento
* Adyen contacts issuer in background to make payment request
* Issuer responds to Adyen
* Adyen API responds to original request from Magento Adyen plugin that request was successful
* Magento sends customer to order success page
* Adyen follows up via API endpoint in the Magento Adyen plugin to confirm payment was captured
* Magento confirms with customer via order success email

Technical Design (TD) - PayPal Express - Australia, Czech Republic, Israel, Mexico, and Singapore: PayPal Express is a payment gateway option that ships with the Magento core, and is available on Australia, Czech Republic, Israel, Mexico, Singapore. Customers who chose to pay by PayPal Express are taken off site to the PayPal servers to complete their payment. They can pay by card, finance or account balance but Magento doesn’t need to know this, only that the order is paid for. Some of the useful links were:

* Dyson Australia: https://www.dyson.com.au
* Dyson: Czech Republic: https://www.dyson.cz
* Dyson Israel: https://www.dyson.co.il
* Dyson Mexico: https://www.dyson.com.mx
* Dyson Singapore: https://www.dyson.com.sg
* Test Payment Details: RoW - https://cyclones.atlassian.net/wiki/spaces/DR/pages/1732511790/RoW+-+Test+Payment+Details

Functional flow:

* Add product to basket
* View basket
* Proceed to checkout – shipping details
* Proceed to checkout – payment step – choose PayPal
* On the PayPal portal, customer is given option to login to their PayPal or proceed to pay with Credit/Debit as a guest.

Note: step 5 and 6 can take place either in an iframe over the checkout, or offsite and in the current browser window,

* Depending on the market. Either way, the data entry and transaction is always taking place in PayPal UI and servers.
* Customer chooses pay with Credit/Debit card (finance and balance are assumed OOS), and fills in their payment information.
* Return to Magento server on Order confirmation page

Technical flow: The PayPal Express payment option is shipped with the Magento core and as such it is always deployed whenever the Magento core itself is deployed. The PayPal Express logical steps are as follows:

* Customer proceeds to pay on the checkout page and choses PayPal Express
* Customer redirected to PayPal Express portal, or presented it overlaying the checkout in an iframe
* Customer enters card details in PayPal Express portal and completes the transaction
* Customer Redirected to Magento order success page
* PayPal contacts issuer in the background to actually capture the payment
* PayPal sends an Instant Payment Notification to an API endpoint in the Magento PayPal Express module
* Magento confirms with customer via order success email

Technical Design (TD) - PayPal Plus - Mexico: PayPal Plus (https://developer.paypal.com/docs/integration/paypal-plus/mexico-brazil/paypal-plus-integration-guide-mexico-brazil/) is a payment option on Mexico. The integration uses a Magento 2 extension provided by PayPal but developed for them by QBO (https://www.qbo.tech/).Customers who chose to pay by PayPal Plus do so into an iframe of form fields hosted by PayPal servers, but look like they are part of the Magento checkout. Some of the useful links were:

* Dyson Mexico: https://www.dyson.com.mx
* Magento Extension (public repository): mkastaneda/magento2-paypalplusmx
* Test Payment Details: https://cyclones.atlassian.net/wiki/spaces/DR/pages/1732511790/RoW+-+Test+Payment+Details

Functional flow:

* Add product to basket
* View basket
* Proceed to checkout – shipping details
* Proceed to checkout – payment step – choose Credit/Debit Card (not Pay with PayPal is actually PayPal Express option)
* Credit card fields are encapsulated in an iframe hosted by PayPal Servers (note the iframe src is obfuscated and injected via PayPal module java script)
* the iframe is only returned if valid PayPal API credentials are present when requesting it)
* Return to Magento server on Order confirmation page

Technical flow: The PayPal plus Magento 2 extension is provided by QBO via a public repository. The extension is installed on the Magento server via the standard package management system, composer. This results in the traffic being limited to the browser, Magento server, Adyen server and the financial institutes. The PayPal Plus logical steps were as follows:

Customer proceeds to pay on the checkout page and enters payment information

Magento PayPal Plus Plugin request secure iframe from PayPal API

Customer enters card details into the secure iframe, thinking it’s part of Magento

PayPal contacts issuer in background to make payment request and captures (@Dave Merrin no follow up calls are made to Magento so assume capture is immediate)

Issuer responds to PayPal

PayPal API responds to original request from Magento Adyen plugin that request was successful

Magento sends customer to order success page

Magento confirms with customer via order success email

Technical Design (TD) - PayU India: PayU India is the sole real-time payment gateway option for the Indian market (Cash on Delivery is also available). PayU is also available on South Africa but PayU India and PayU South Africa are distinct entities and implementations. Customers who chose to pay by Credit/Debit card or instalments are taken off site to the PayU servers to complete their payment. When the process is completed, customers are returned to Magento. Some of the useful links were as follows:

* Dyson India: https://www.dyson.in
* Test Payment Details: https://cyclones.atlassian.net/wiki/spaces/DR/pages/1732511790/RoW+-+Test+Payment+Details

Functional flow:

* Add product to basket
* View basket
* Proceed to checkout – shipping details
* Proceed to checkout – payment step – choose Credit/Debit card
* On the PayU portal, customer is given option to pay by Credit/Debit cards as well various net banking, wallet, EMI and
* Third-party payment methods but all are wrapped in PayU.
* Customer completes payment and returns to Magento server on Order confirmation page

Technical flow: The version of the PayU India payment gateway plugin we are using is provided as a zip file which we have committed to the private Dyson India site repo and is versioned and deployed as part of the site’s codebase. A newer version (which is a distinct different version) is available via public repository and could be composer installed. The PayU India logical steps were as follows:

* Customer proceeds to pay by Credit/Debit card on the checkout page
* Customer redirected to PayU India portal
* Customer enters card details in PayU India portal and completes the transaction
* Customer Redirected to Magento order success page
* PayU India contacts issuer in the background to actually capture the payment
* PayU India sends a Payment Notification to an API endpoint in the Magento PayPal Express module
* Magento confirms with customer via order success email