ID Resolver – Dependent Systems, Hardware and Software

3rd Party - Card Payments System • Provide mocked responses of their system via requesting certain names and addresses for positive and negative responses. • The system has an information endpoint which can be contacted for availability • It is unknown if there are further test environments available or what the ultimate capacity is for this system. • As a 3rd party system, comprehensive online documentation regarding each API operation and authentication instructions.	**Internal – Credit Search System** • Recently re-architected as a microservice architecture, leading to significant capacity increases. • Able to deploy on demand in an automated fashion to multiple environments. • A mock service for each micro service was created as part of the new architecture. • Extensive API documentation using Swagger is available to all teams at the organisation.
Internal – Money Laundering System • The system is a tightly coupled, very dense, legacy system with limited automated testing. • The system has extensive logging of audit records • There are only two environments for this system, a test environment and production, neither of which are the same. • A few experienced developers hold most of the knowledge for this system.	**IDResolver – CentOS for Web and Database Layers** • CentOS uses yum as a package manager, therefore dependencies can be limited. • CentOS has the standard logging tooling of Linux, including syslog and with Apache installed also access logs. • As CentOS is built upon the Linux Kernel, it has wide configuration options. • Although CentOS is used widely, it does not have the support, security focus and updates of RedHat enterprise.
IDResolver – Perl for Batch Processing • The batch processing in Perl is not thread safe, so is only run on one webserver, leaving no redundancy. • The batch processing system writes log messages to the audit database and sends emails for successfully completed jobs. • Jobs are scheduled on a time-based basis, therefore can be separated to assist • There are only two Perl Developers remaining and they are transitioning to php, as there is only one system in Perl.	**IDResolver – Load Balancing** • The Load Balancer for test environment and production are held on separate instances, production needs high level of access. • As the load balancing is a third-party product, it has in built monitoring and alerting, plus an API for configuring alerts in other tools. • There is a web interface for observing and changing IP rules and pool configuration that is accessible to development teams for the test environments. • Even though it is a third party, online documentation is limited, as the company wishes to train onsite.