AUTODESK UNIVERSITY

MFG125084

Hybrid Cloud Solutions

Tony Mandatori
Autodesk, Inc.

Learning Objectives

- Identify Hybrid Cloud systems
- Recognize the challenges of deploying systems that that use different platforms
- Categorize the concerns specific to these environments
- Optimize for environments

Description

A Hybrid Cloud is a cloud computing environment using a mix of applications on different infrastructures relying on orchestrated integration between the platforms. The infrastructures can be on-premises, private cloud and third-party, public cloud services. This a legitimate option even if all the components are not pure cloud. Our Fusion Lifecycle Implementations are typically Hybrid Cloud: we use Vault (deployed as either on premise or Private Cloud) and FLC (pure cloud). These environments come with their own challenges. In this class, we discuss these challenges and provided recommendations on how to optimize these heterogeneous environments.

Speaker(s)

Tony Mandatori

AUTODESK UNIVERSITY

Cloud systems

Hybrid Cloud is a common term - http://searchcloudcomputing.techtarget.com/definition/hybrid-cloud

It represents an environment using a combination of applications on different infrastructures, and relies on integration between the systems. The infrastructures can be on-premises, private cloud and third-party, public cloud services.

Cloud Flavors

http://www.securityweek.com/three-categories-cloud-computing-whats-your-flavor

- IaaS - Infrastructure as a Service
    - Analogous to The Road –
    - Infrastructure – Services, Memory, Disk, Operating Systems
    - Examples – AWS, Azure
- PaaS - Platform as a Service
    - Analogous to The Car
    - Database, File Storage
    - Amazon RDS, Azure SQL Storage, Amazon S3/Azure Storage
- SaaS - Software as a Service
    - Analogous to Transportation from point to another
    - Application which makes use of
    - Fusion Lifecycle (PLM), NetSuite (ERP)

Cloud Deployment Alternatives

Using the Cloud is an alternative to local on-site systems.

- Private Cloud
    - This is typically the first stop for companies that develop their own systems
    - Instead of purchasing assets that depreciate, the entire solution can be hosted on someone else's hardware. There is an added benefit that the hardware scale incrementally to be as little or as large as needed. You only pay for the portion you use.
    - This usually stops at the IaaS level. There is no need for PaaS, or SaaS since the customer feels comfortable with the custom solution they've created and is still new to using the Cloud.
- Managed Services
    - As software providers (or their partners) get more comfortable with cloud technology, they start offering a managed alternative for their solutions. Instead of asking the customer to purchase software, the provider/partner can deploy and maintain the software better and more easily than their customers.
    - The solution is similar to the Private Cloud, but it is managed by a third party
    - Remote monitoring tools are employed to ensure a more stable experience for the end user. The provider/partner anticipates failures and applies upgrades as needed.

  - Customer is in charge of making decisions on upgrades/changes that are applied to the service, but relies on a third party to implement those changes.
  - This is typically in between IaaS and PaaS. The partner may make use of third party components like a common database (RDS) and common file storage (OSS),
  - Customizations to the application are owned by the customer and need to be maintained.
- Public / Multi-Tenant / Shared
  - This is the ultimate hands-free experience
  - The system can be configured, but customizations are replaced with integrations.
  - ReST API is usually available
  - No Customization is possible on the Public System since it would disrupt other customers/tenants
  - Integrated systems may or may not be cloud systems

SaaS Cloud Systems

Multiple systems are needed for business regardless of size. Cloud alternatives exist for each.

- xCAD
  - Fusion 360 Mechanical
  - OnShape
- CRM
  - NetSuite
- PDM
  - CloudPDM
- PLM
  - Fusion Lifecycle
- ERP
  - NetSuite
- BI
  - Tableau
- Integration
  - Jitterbit Harmony
  - MuleSoft CloudHub

Mixing Cloud with Local systems

Moving to the cloud will typically happen incrementally. Prudent users will want upgrade systems in sequence based on need/priority. This will cause more work as integrations need to be created to multiple systems.

To offset the need to recreate integrations – the optimal solution would be to create a service bus that interacts with each application. However, this too would require a big effort to retrofit each application before starting. Furthermore, each of the existing system may already expose ReST API that already provides an integration layer that can easily be called.

The recommended strategy is to choose the systems that are most visible and to upgrade those first. Create the integrations that provide the most immediate benefit, and then retrofit the others as needed.

Identify the systems with the most immediate benefit by concentrating on your pain points. Errors in production are typically the most expensive. So centralize the change process. Then use this as the vehicle to publish to the ERP system.

PLM and ERP can be handled exclusively in the cloud. Typically, the CAD integration will pose greater issues.

PDM Integration

Although we've provided CloudPDM in the past, our robust solution is Vault. Vault is a client server tool that requires both a server and client software for each of the connected clients. Integration between Vault and PLM is possible.

PDM Requirements are different from PLM. PLM focusses on status and readiness whereas PDM focusses on detailed design. PLM is concerned with approvals, whereas PDM includes all the content to receive the approvals. PLM data focusses on readiness whereas PDM data is detailed design data.

PDM data is highly sensitive

Export controls usually focus on the design data. The approvals received are not as sensitive as the content being approved.

PDM data is big

PLM data is mostly metadata whereas PDM data is mostly content.

Design is localized

A collocated team is usually employed to design a product. This isn't always the case. Aerospace manufactures will create centers of excellence for different pieces of the aircraft (fuselage, wing, etc). The personal interaction is usually important in optimizing the design. This dynamic is changing as modern tools allow for better collaboration – but the traditional business strategies are hard to break.

CRM/ERP Integration

Each geography builds the product a different way depending the capabilities of the local factory and the sourcing vendors used. Although separate factories build the product, the configuration

is usually pushed to each factory from a central instance. With modern ERP systems, the central system is cloud based.

There is no strong argument for making an ERP system cloud based apart from maintenance and cost of ownership. A mix of both exists with a trend towards cloud systems.

Integrating CAD Clients

Most CAD systems come with a CAD Data Management tool (CDM). ECAD data has complexities which make it easier to manage in the native CDM system. These vendors prefer integrations to their CDM systems instead of direct integrations from the CAD tool to the PLM System. This highlights the idea of different areas of the BOM are managed by different PDM systems. ECAD in an electrical CAD system, MCAD in a mechanical CAD The combined BOM is then managed in PLM.

The PDM system doesn't necessarily have all the information for the BOM. The PLM system amalgamates the different BOMs in a heterogeneous BOM.

NOTE: Fusion 360 AnyCAD offers a different perspective. The heterogeneous CAD BOM can be visualized properly in Fusion 360. If adopted, it would provide a single cloud solution for PDM which can be integrated directly to PLM bypassing the need for CAD specific CDM.

Hybrid Strategies

The nature of PLM Systems lends itself to being an ideal choice for migration to the cloud.

PLM/ERP

ERP systems would move to the cloud depending on choices made by the customer. As long as simple functionality is required, the system can exist in the cloud. NetSuite is an obvious choice because of its cloud architecture. By contrast, complex systems like SAP and Oracle Manufacturing (which have the lion share of the market would remain as local expensive deployments because of a lack of alternatives.

So Simplicity is our first Hybrid requirement.

PDM/PLM

As discussed earlier, PDM has many challenges when migrating to the cloud:

- Security
- Performance

CloudPDM system can sort these issues out with added features. But it will be difficult to convince people to switch from well performing active products to cloud technologies. Especially when they may have heard of, or even experienced, poor web performance. There is no need to change something that is already working well.

PDM/PLM Implemented Solutions

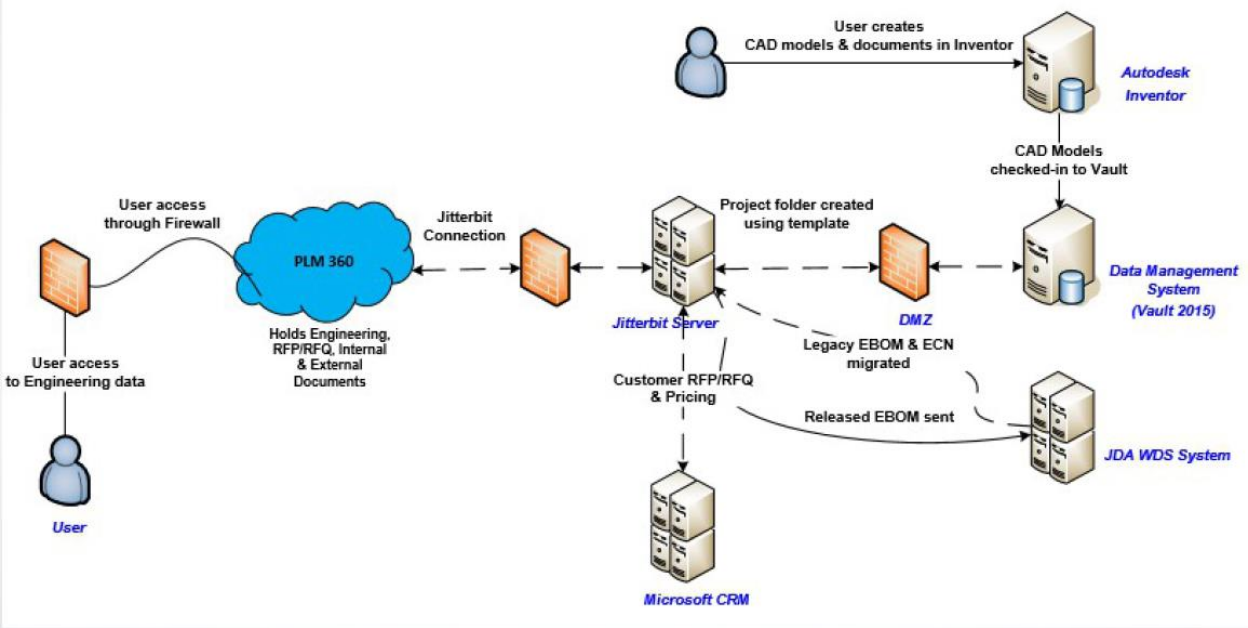
A few examples exist

Sample Implementations

The following examples exist:

Vault (Local) and Fusion Lifecycle (Cloud) via Jitterbit (Harmony)

Many deployments of this nature exist.



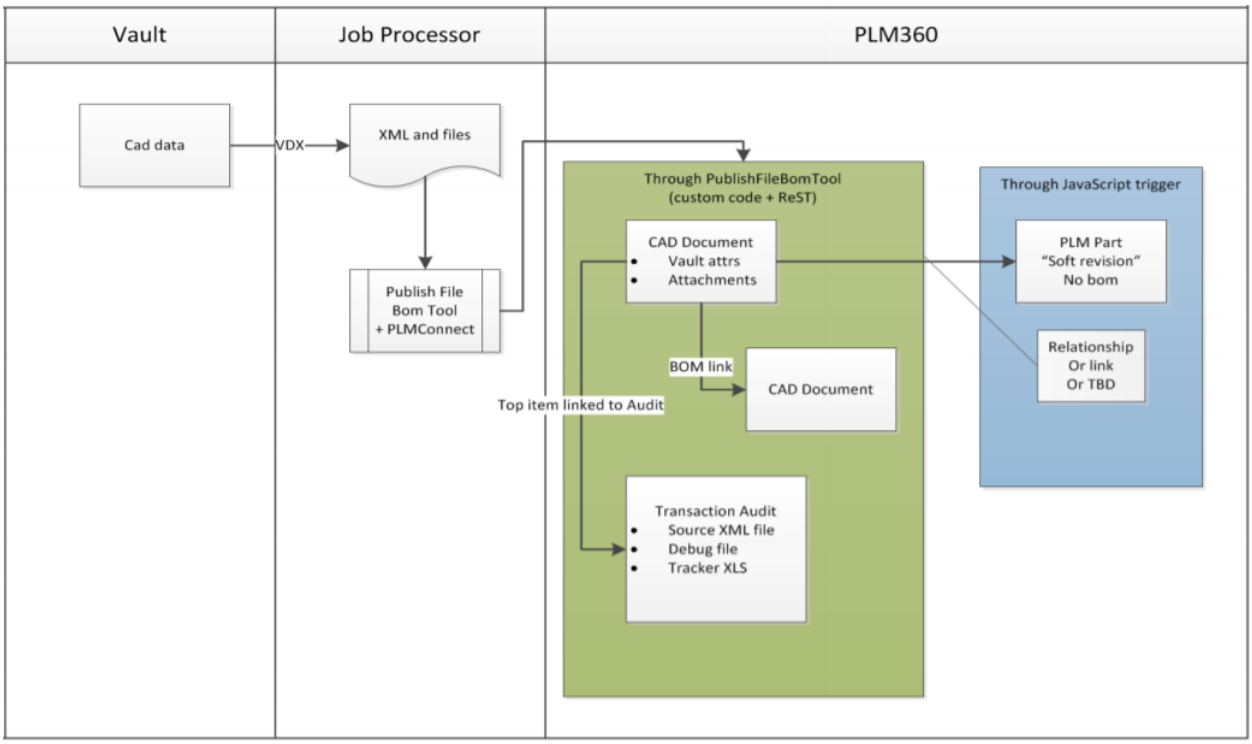
Usually implemented with a plugin to create a synch on a part by part basis with an Item in PLM.

On a one by one basis, a relationship is created with an item in PLM.

Jitterbit can be substituted for an Integration tool. Initially we used Jitterbit, but in the future, we started using ASP.NET server

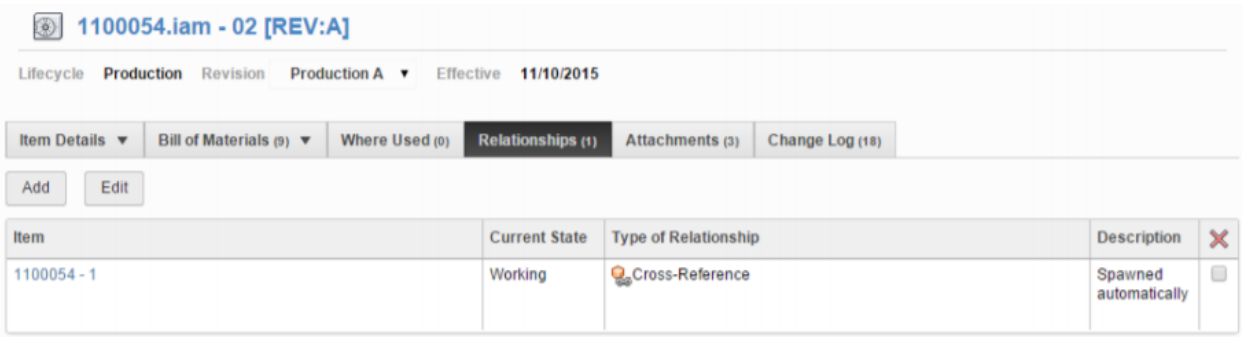
Vault (Local) and Fusion Lifecycle (Cloud) via custom Import/Export Tools (VAO, Utility Pack)

Instead of using Jitterbit, we can use a combination of Extract tools (VAO) and Load tools



The BOM is extracted

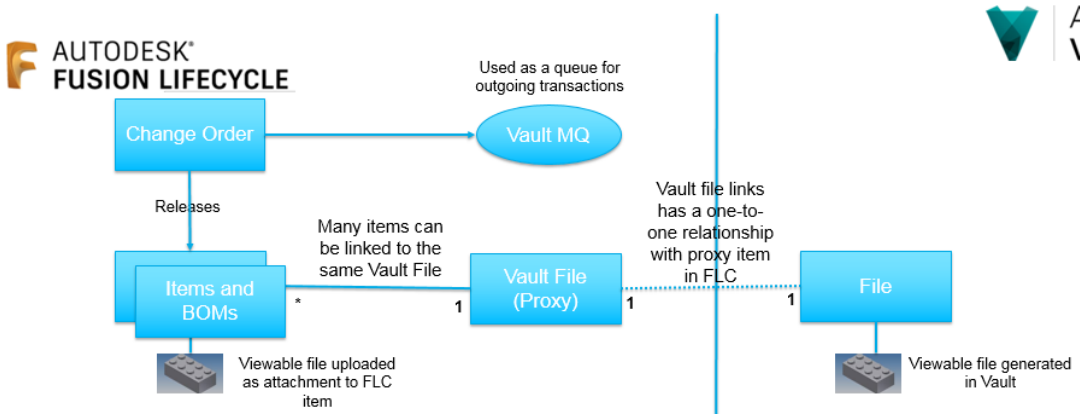
The relationship is managed in PLM



Commonality in PDM/PLM

Usage of a Proxy Workspace for Documents

On early implementations, we created a relationship to the Item (part) directly on later implementations we decided that the relationship should be managed in PLM. So the Vault item was related to a document object in PLM.



Also note that a Vault Queue is used to communicate between Cloud PLM system to an internally protected vault server. The integration needs to poll (query) the Vault MQ workspace to identify required updates that it can make to Vault.

Use of Queuing Workspace

For Communicating with the protected system, an internal component like Local Agent or .NET standalone process needs to reach out to get instructions

Error Logging Workspace

Errors cannot be grounded. Integrations are prone to break, and when they do, the proliferation of bad can be expensive to fix. We need to stop issues early in order to mitigate expenses in fixing the failure.

Differences

The following differences should be noted

Whether the BOM is extracted

Getting the BOM information is complex. There is no out-of-the-box way of propagating the BOM to FLC, but it has been done before. Data is available in Vault and be queried using API or can be exported. It is easier to export the data to a file.

Integration Mechanism

Orchestration vs File Loaders. The mechanism used depends on a few factors:

- Complexity
- Delineation of responsibility – decoupling the integration to be based on files is an easy of delineating the responsibility. A common “contract” must be agreed upon.

Forge

Forge is another service that can be called upon for viewing. Again, integration is not available out-of-the-box. It has been used where the value of the visualization was considered important enough to warrant the work involved in scripting the API.

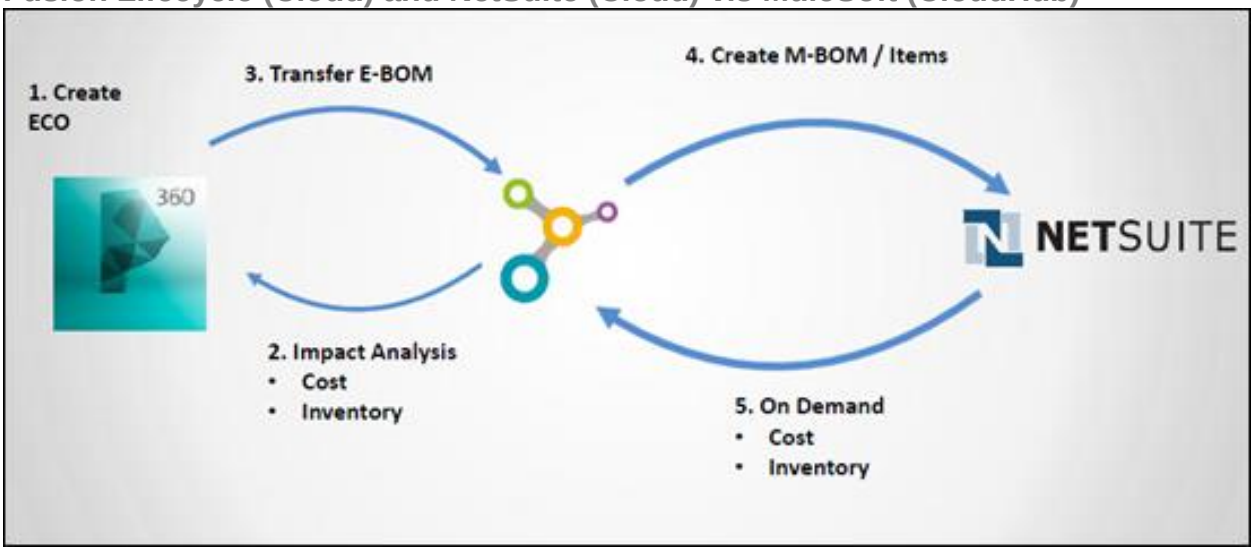
PLM/ERP Implemented Solutions

Sample Implementations

Fusion Lifecycle (Cloud) and SAP (Local) via custom Import/Export Tools (Utility Pack and ERP Import Tools)

SAP users are generally more familiar with Import mechanisms than they are the BAPI or IDOC API. So using a file import mechanism is common with these systems.

Fusion Lifecycle (Cloud) and NetSuite (Cloud) vis MuleSoft (CloudHub)



NetSuite, by contrast, is a cloud system. In cloud systems, importing is a little trickier because the system is remote. In the case of NetSuite, we have built Jitterbit customizations which we readily reuse with small tweeks.

Fusion Lifecycle (Cloud) to JDE (Local) via Jitterbit (Harmony)

Although JDE is a local system, we have used Jitterbit

Fusion Lifecycle (Cloud) to multiple (local) systems, Oracle and Vantage, via Jitterbit (Harmony)

Using multiple ERP systems isn't very common, but manufacturing in multiple is common.

Commonality in PLM/ERP Solutions

Single PLM System

It is common to see multiple CAD systems each with their preferred PDM system. So multiple PDM systems feeding a single PLM system is common. Also, although the ERP system is usually unique, the number of plants can be greater than one. And with acquisition multiple ERP systems are sometimes a reality.

Use Cases

The typical uses cases are seen:

- Publish to ERP
- Query Cost and Inventory on the Fly
- Calculating cost of Change

In all cases, Orchestration is usually the preferred mechanism – in this case, it may delegate to a File Loader

Differences in PLM/ERP

Diversity of ERP Systems

The Systems used
Cloud or Local ERP System

Integration Tool

Orchestration vs File Loaders. Again we are faced with two choices: Use Target System API or Import Mechanism

Number of ERP Integrated Systems or Plants

There is clearly only one PLM system, but there could be multiple ERP boms.

Location of the BOM

EBOM, MBOM and Factory BOM
Or EBOM and Multiple MBOMs

Existence of Change Order and Versions in ERP

NetSuite doesn't hold change orders, but a lot of ERP systems do.

Lessons Learned

Key Decision Points

Is there a system of record where parts are mastered?
Is there a mapping between users and systems based on role?
What are the triggers for copying data from one system to the other?
How do I draw the line between each of the systems? What are the integration contracts (data protocol or schema) that each system must follow?

Usage Scenario

Identify main system for each user. Minimize the number of systems each user needs – if they need to switch between systems, make the transition easy and infrequent.

Avoid Making user wait for integrations to complete. With Synchronous loading, this is inevitable.

Maintain Appropriate Data in each system. Constituents of purchased assemblies may not be needed PLM.

Use PDM systems appropriately – keep ECAD data in Vendor's CDM, keep mechanical in PDM, combine the structure in PLM.

Decide where each BOM resides EBOM, MBOM, SBOM …. It may only be needed in the ERP system.

Other BOMs, Compliance, Sourced BOM Scenarios are important, but unless the tool can manage them appropriately, it's a waste of time.

Limit the amount of data being copied around (e.g. replicating Classification). Isolate the true purpose of each system

Solution

Consolidate tools to ensure that the expertise is developed in the appropriate tool. Point to Point solutions are rich, but expensive to maintain. Building capability in multiple tools is expensive. Choose a tool a stick with it.

Use ReST API as a trigger for the middleware tool.

- The heavy lifting should be done by the tool that doesn't have timeout.
- Avoid manipulating data directly from one enterprise system to the other (for example instead of loading the BOM from PDM to PLM without using the middleware tool, have the PDM system trigger the middleware tool which will then take on the responsibility of loading the PLM system)
- Don't skip the middleware tool just because it's easy to access API from Script.

Use a queue workspace to access systems that are behind a firewall. The protected system needs to pole for updates.

Managing complex relationships via integration. Proxy objects with 1-1 relationships should be represented in the system instead of creating 1-many relationships in the integration. Part – Drawing relationships should be visible in the tool.

Error Handling

Keep an Error Log – Integration failures should be communicated and escalated. Grounding the issue in the middleware tool only will make it difficult to correct and may cause hidden errors.

OCM

Documentation and Handoff is important.

Development

Use a development and test environments with full replicas of each production system (same size and version). The entire set of systems needs to be replicated.

Ensure that the solution can scale and support simultaneous transactions

Stress Test Integration: Perform a thorough stress test of the integration before deploying to production

Avoid Infinite Loop: Ensure that an on-update trigger based bi-directional synchronization does not go into an infinite loop

System Specific

The following is a list of system specific suggestions.

Vault

Use the ADMS to Debug: Use the ADMS console to debug server side errors in Vault

When setting up the integration service on the same machine as the Vault server, IIS needs to configured to prevent conflict of HTTP port

NetSuite

Standard NetSuite Connector Functions: Try to use the standard NetSuite connector functions instead of direct web service calls

Concurrent Transactions: Executing concurrent transactions in NetSuite could be restricted by licensing in some cases

Show Internal ID's: Activate the 'Show Internal IDs' option in NetSuite during development

Web Services Log to Debug: Use the web services log in NetSuite to debug failures

Future Strategy

There should be questions about the future strategy in light of the regression to Classic Attachments.

CloudPDM is the future.
Development is ongoing with Fusion Team.

In the future, CloudPDM will be the exclusive integration point for content in Fusion Lifecycle.
All PDM apps wishing to integrate with Fusion Lifecycle must do so via Fusion Team.
This includes Vault – Vault will integrate with Fusion Team and Fusion Team will integrate with Fusion Lifecycle.

Vault to Fusion Team integration answers limitations of CloudPDM:

- IP restrictions
- ITAR
- Performance