

Evolution of Interfaces as Cloud Infrastructures Increase

Robert Garbus, Mainsaver Software

In the days when mission critical business systems including the CMMS (Computerized Maintenance Management System) were on the same or nearby servers, exchanging data between systems could be as simple as inserting records into an intermediate database table or placing flat files on a common file server accessible by both systems. As more enterprises have either moved some or many of their core business systems to cloud based applications or outsourced them completely the requirement has emerged to pass data from one system to another utilizing alternative methods such as file transfers or application calls. The purpose of this paper is to discuss methodologies and lessons learned in data exchanges from cloud based Mainsaver CMMS to other core business systems which are either in-house, cloud based or outsourced.

This paper will use the following example where a Mainsaver customer exports payroll data to a payroll processor such as ADP® or Paychex® and also receives daily spare parts inventory from a legacy ERP application such as Oracle® Financials or SAP® software. The decision on which data elements (suppliers, materials, purchase orders, invoices, timecards) are maintained in which system was decided early in the business analysis of the overall project. For example, the ERP package may be the system of record for suppliers because that is where the accounts payable resides. In this case it was decided to pass supplier create, replace, update and delete (CRUD) transactions to Mainsaver.

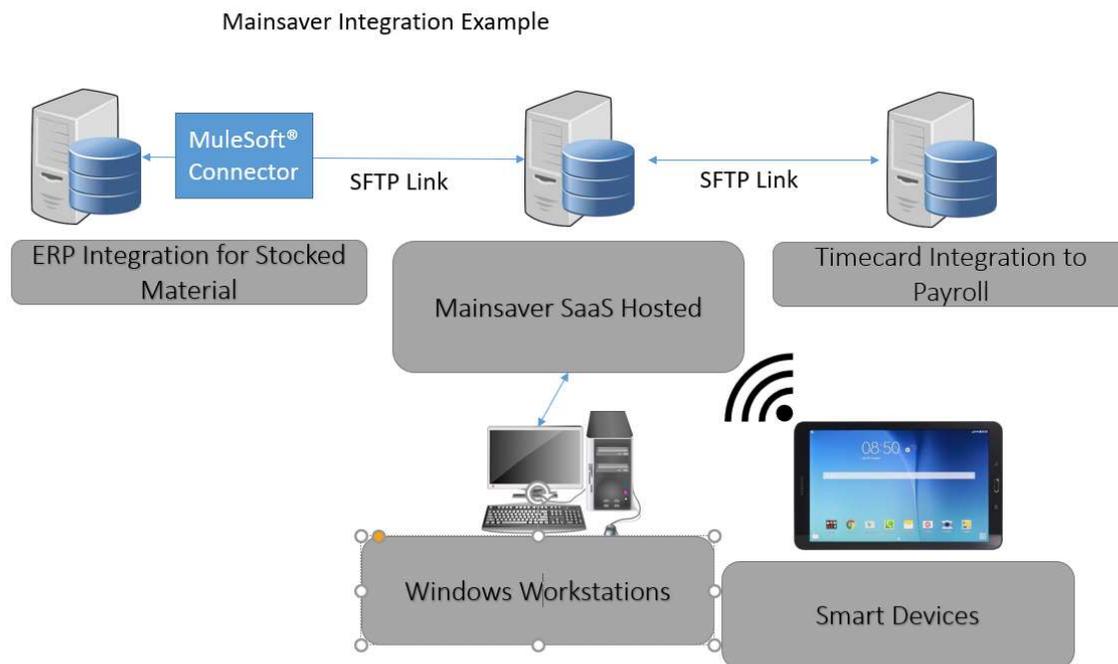


Figure 1 – Architecture Diagram

Business Analysis

In building the technical design document for the interfaces the following areas are explored.

- 1- Does the target system have a pre-determined record format for data import?
- 2- What are the data elements and exact formats required for each exchange? Which fields are required?
- 3- If using comma separated value (CSV) files, ensure that the comma character is properly encased in quotes. Ensure any other illegal characters are properly parsed from the data.
- 4- Upon import, which data elements must be validated against existing tables (such as account codes) versus when a new data value may be inserted into a table.
- 5- If certain data elements are critical for data integrity (such as supplier ID) should that data be passed to the exporting system in order to avoid rejected records due to invalid values.
- 6- Error Handling: Notification and re-processing of rejected records. For example, a purchase requisition is passed from Mainsaver to SAP however to supplier is not a valid SAP supplier.
- 7- Error Handling: If the scheduled import and export tasks stop running for some extraneous reason, when the tasks are restarted how will multiple data files be processed.
- 8- Timing of file transfer: ensure that the sequence is logical. If a new supplier is added, that should be passed before a new PO with the supplier number is passed.
- 9- Testing: Development, scheduling and execution of a comprehensive test plan.
- 10- Agreement: Since there are multiple applications and thus multiple entities involved in the exchange of data, it is important that all parties agree on the design specification and do not deviate without consulting the project manager and updating the design document.
- 11- Filenames should include a date and timestamp so if multiple files build up on one side of the interface, when they are processed it is done in chronological order.
- 12- Record lock-out rules, stopping simultaneous changing of the same record/transaction/data.
- 13- File Retention: Once the data files are processed they are deleted off of the SFTP site however they should be retained in an archive directory for audit and support purposes.

Technical Architecture

If Mainsaver and the target application are both on local servers, the data may be exchanged through a direct Insert from a database table in Mainsaver to an intermediate or import/export table within target application. Then the staged records may be processed immediately through a table trigger or in batch at a pre-determined interval.

When the applications are not local, the data is exchanged though the World Wide Web. Rather than an API (Application Programming Interface) that uses HTTP requests, Mainsaver exchanges data with other systems through the utilization of data transfers.

The data transfers are usually flat file exchange in various formats such as iDoc, csv or xml. As data is most commonly confidential for customers, Mainsaver uses SFTP (SSH File Transfer Protocol) which utilizes a secure shell to prevent malicious sniffing. Part of the security protocol involves whitelisting (authorizing) only the IP address of the target systems and rejecting connections from other sources.

This allows Mainsaver to send or receive data from virtually any system. The use of flat files also mitigates the risk of changes to record layouts when applications are upgraded which could lead to failure of the interface due to a newly introduced incompatibility.

The target applications can place or pick up files from the SFTP site using file transfer software such as WinSCP. In the example Figure 1 above, instead of the customer creating a file direct from SAP and transferring that file to the SFTP site, they use an intermediate application supplied by the MuleSoft® Integration Platform which provides a connector into SAP. This provides an additional layer of protection.

Process Timing

The business requirements of the interface functionality should determine how often the data files are sent and processed. In most situations, the data ~~is may be~~ sent at a predetermined time interval. In other cases, such as time export for payroll, the export may be manually triggered once it is verified all employee time is validated. It is also important the data processing times be sequenced and staggered based on the requirements. The table below illustrates an example of a purchasing interface.

Function	Description	Timing
Supplier Import	Create, update or Deactivate	Hourly on the hour
Purchase Request Export	Following Approval	Sem-Hourly at 00:05 and 00:35
PO Import	PO Created in ERP	Hourly at 00:10
Receipt Import	Populated Inventory	Hourly at 00:15

Reliability and Performance

Failure of the interface means that data in system A is not showing up in system B or vice versa. There are a few scenarios which can cause failures.

- Record Format: The import/export file was created in a format not matching the specified layout. For example, fields are missing.
- Record Content: A date or numeric field does not match the specified format; an alphanumeric field is too large or a data element does not pass validation. An example might be a receipt transaction from ERP system being passed for a purchase order number that does not exist in Mainsaver.
- Process Job Failure: The scheduled task running the import/export job failed for some reason such as an expired password or change in network security.
- No Data: The application routine to create the output data has failed and therefore there is not producing the required import/export data.

Since the scheduled tasks are running autonomously, these anomalies might not be known until the end users discover some issue. If invoices are passed from the CMMS system to the accounts payable system, failure of the interface might result in suppliers not being paid. For this reason, it is important that there is a notification system that would send emails where process failures occur. Also, the system owners should have access to the error log to monitor problems as well as to verify normal performance.

Summary

The exchange of data between 2 systems essentially provides a data bridge to satisfy business requirements efficiently and accurately. The interface reduces duplication of effort and keystroke errors. Proper planning, execution and monitoring will ensure that the data exchange is effective.