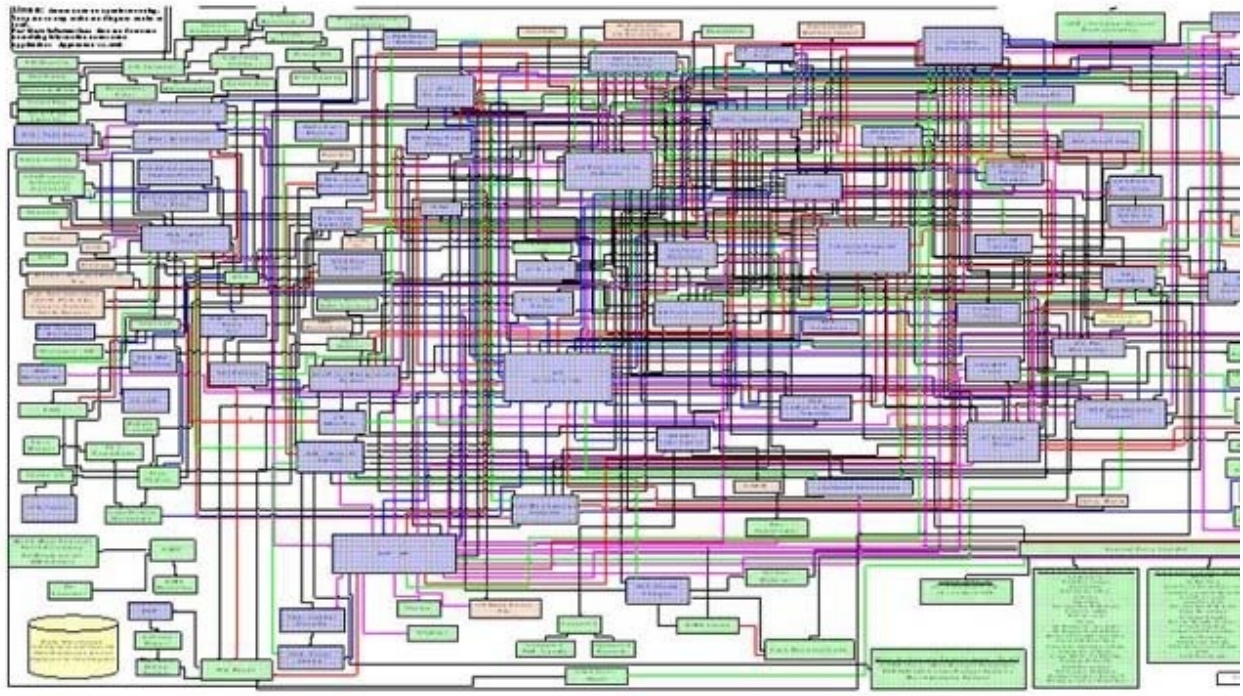
Avoiding an information hairball

Date: 2016-07-07 10:20:00



Why are there data integration challenges?

Typical reasons for data integration challenges include:-

- Multiple systems created to provide the same business function e.g. business units choosing their own sales order processing systems either prior to or after acquisition.
- Each system has it's own view as to how data should be stored (aka physical data model)
- Systems needing to be replaced over time e.g. because they are no longer supported, no longer capable of meeting business requirements or the business function has moved between divisions and new technology choices are implemented.

Problems with a typical data integration approach

Integration of data between systems is primarily done via peer-to-peer bespoke methods most typically moving data in batch. Problems with this approach include:-

- Time delays whilst data is being batch processed which reduces the customer experience e.g. where a sales order completes, but a billing invoice is generated only on a weekly cycle and can lead to inconsistencies in reported information e.g. where payment is made, but there is a delay in feeding this to a reporting system.
- Complexity of peer to peer integration grows exponentially as more systems are added and the connections between systems increase – commonly referred to as the "information hairball" problem.
- Makes it difficult to integrate systems owned by new companies that are acquired in to existing systems
- Multiple physical data integration interfaces are developed to cover what essentially is the same logical data integration requirement.
- Whenever a system is replaced, every physical data interface between it and other systems needs to be re-written.
- Knowledge as to how data integration works is retained by delivery projects and often lost if external consultancies were used in delivery

All of this can cost your company significant amounts of money in development, maintenance and opportunity costs - a money pit.

Enterprise Data Integration requirements

There are many different data integration technologies available to you to solve this problem, but first you will need to gather requirements. You'll need to gather information about the following:-

Requirement type	Explanation
Latency	How quickly you want to data to be transferred from 1 system to another
Frequency	Whether you want near real time or periodic transfers, and if the latter, how frequently you want to move the data
Volume	How much data needs to be moved in 1 instance
Type	Whether it's structured e.g. system generated data, semi-structured e.g. email or unstructured e.g. social media posts

Enterprise Data Integration technologies

Dependent on requirements, there are different data integration technologies available.

Table below identifies examples of popular technology solutions which meet different types of requirements today. (Note: You should determine for yourselves, however, which technology is most appropriate for your organisation and have a vendor demonstrate via a proof of concept)

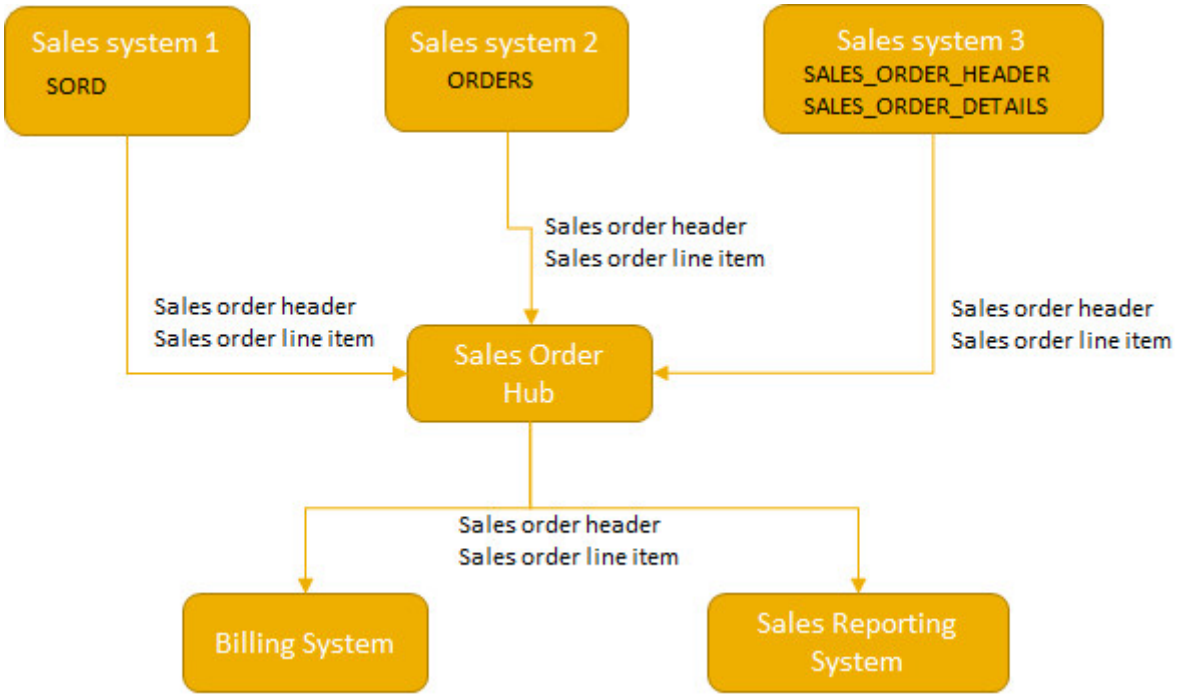
Technology	Latency	Frequency	Volume	Type	Example Use Case
Mulesoft	Low	Near real-time	Low	Structured	Sales order needing to be sent to a billing system for an online sale.
Tibco	Very Low	Real time	Low	Structured	High speed information required for investment bank trading desks
Informatica, SSIS (Microsoft), ODI (Oracle), Infosphere (IBM)	High	Periodic – End of Day	High	Structured	End of day batch population of a data warehouse
GoldenGate (Oracle), Q-Replication (IBM)	High	Near real-time	Low	Structured	Replication of data from a transactional database to a reporting database.
Composite (Cisco)	Instant	Real time	Variable	Structured	Data virtualisation technology which allows querying and integration of data from multiple databases
Pentaho, Talend	Low	Near real-time	High	Structured, Semi-structured, Unstructured	Parsing of raw data held on a Hadoop cluster

Integration Architecture

Rather than building peer to peer, system specific "information hairballs", a better approach is to create a hub and spoke publisher-subscriber architecture.

For example, the integration architecture below shows a situation where you have 3 sales systems, each with different ways of physically storing sales order information. Each sales system publishes their sales orders to a sales order hub which automatically passes on that information to the billing and sales system who have subscribed.

Advantages of this architecture is that you can easily add a new sales system or replace an existing system without impacting the downstream billing or reporting systems. It's also easier to get a holistic view of sales orders.



Interface Design

In order to build a reliable standard interface between systems, that another system owner could reasonably be expected to supply high quality data for, you will need to supply an interface specification containing the following elements:-

1.A message structure down to field level with descriptions. The message structure should indicate the relationships between sections within the message e.g. a message for a sales order might contain a sales order header which can contain 1 or more sales order line items.

2.Information as to restricted list of values (aka codeset) that are applicable to particular attributes

3.Information on specific format restrictions e.g. date formats must be "dd/mm/yyyy"

4.Information as to the validation rules

The interface spec should ideally be machine readable e.g. xml schema definition (xsd) are typically used to define an interface specification which can validate xml messages.

In addition to the interface spec, a word document describing the purpose of the interface and where it would typically be used in a business process within Informa provides context.

A very good example of well defined interface documentation is provided by the ISO20022 organisation whose purpose is to standardise electronic messages which are typically passed between financial institutions e.g. payment information - https://www.iso20022.org/full_catalogue.page. Each message has a supplied xsd containing the information set out above, and accompanying message definition report (MDR) documentation which provides interface spec documentation and typical business process usage.

Monitoring enterprise data integration

As well as the physical movement of data from 1 system to another, there needs to be a capability in place to monitor how well this is operating.

However robust a data integration system is, there will always be messages that are rejected and require manual intervention. There needs to be a process in place to deal with this situation.

Checks also need to be made to ensure that data isn't lost during the transfer. This is typically achieved by capturing statistics about data as it's sent from the source, and then running identical queries against the target system and comparing the difference.

Your IT Operations unit will also need to ensure that the data integration system is operational, and is capable of handling peak transaction rates.