

WELCOME



NARELLE CAREY

Practice Manager i2
(Innovation & Integration) AU/NZ

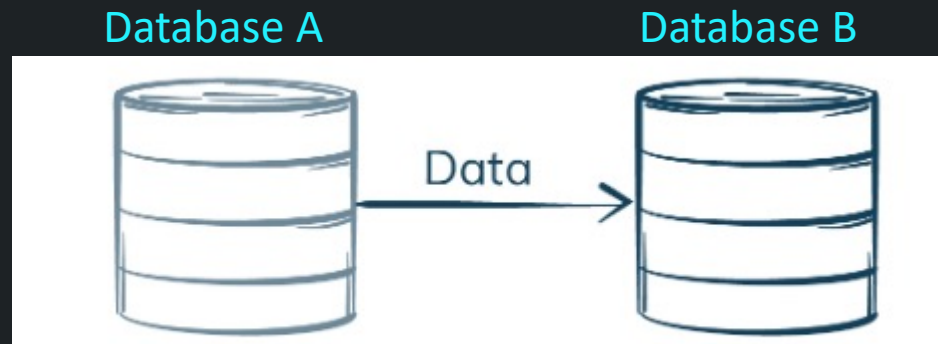
CRITICAL ELEMENTS THAT MUST BE DEFINED
BEFORE YOUR INTEGRATION PROJECT



Common Mistakes

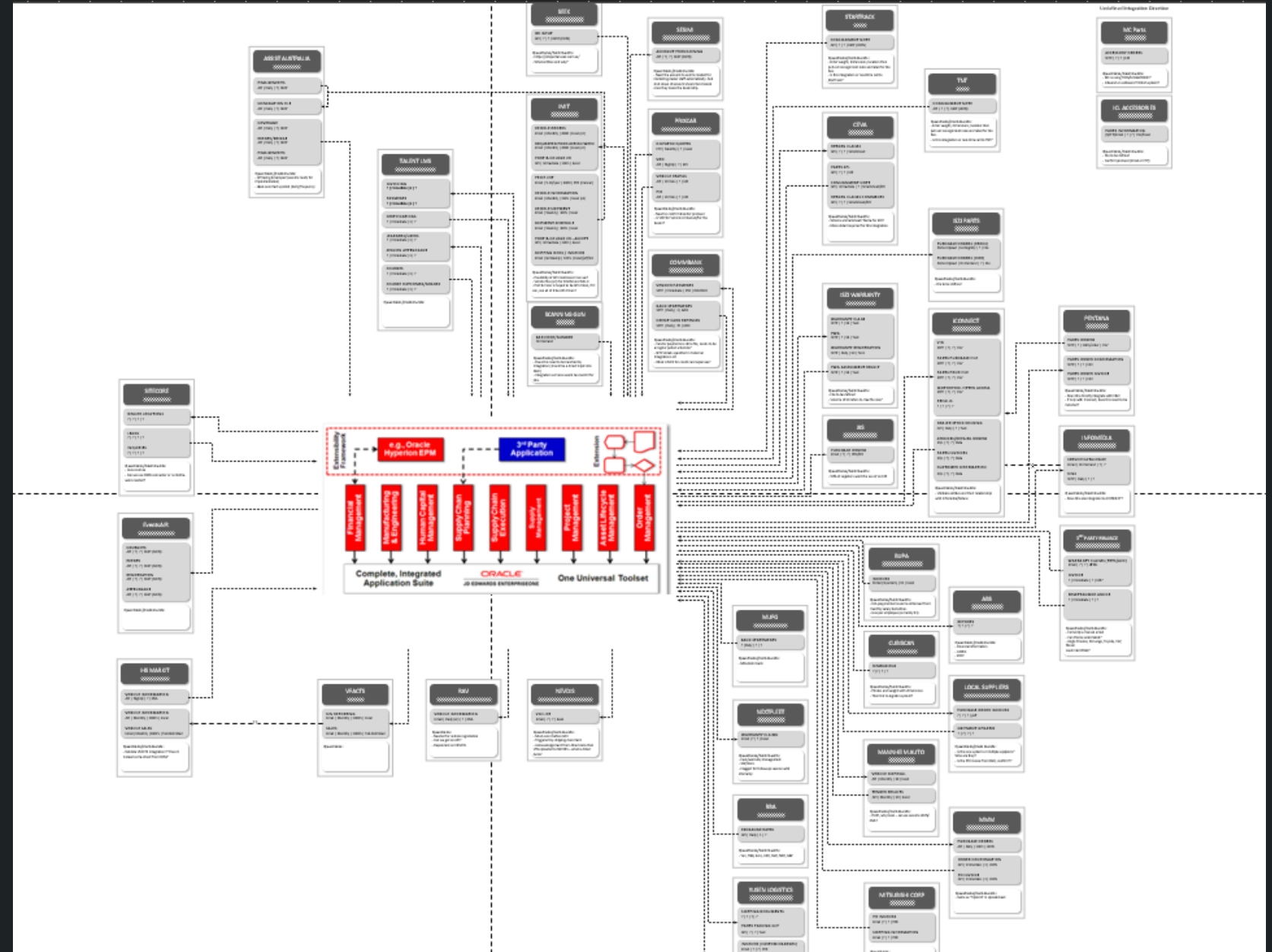
- No Enterprise Architect or Architecture
- Point to point is easier!
- We selected a middleware because someone said it works well with this one application
- We need a low-code option for our enterprise sized organisation
- Don't worry, we'll look after our integrations
- Why do you need to know about our processes? It's just data.
- We want it all, and we want it now
- We want people to be able to update data from any system and it go everywhere else.

HOW MOST PEOPLE DESCRIBE INTEGRATION



How hard can it be?
Right?

THE REALITY



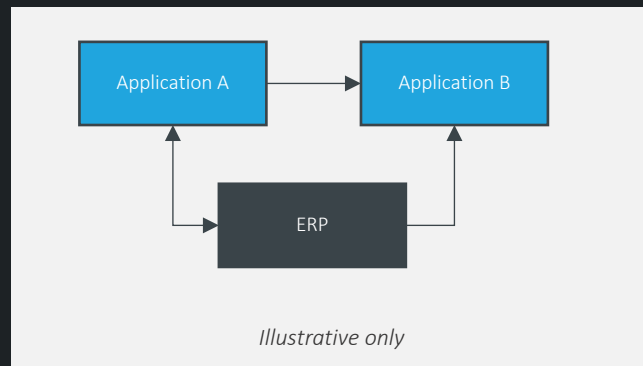
Common Mistakes

- No Enterprise Architect or Architecture
- Point to point is easier!
- We selected a middleware because someone said it works well with this one application
- We need a low-code option for our enterprise sized organisation
- Don't worry, we'll look after our integrations
- Why do you need to know about our processes? It's just data.
- We want it all, and we want it now
- We want people to be able to update data from any system and it go everywhere else.

Integration strategy model

The different approaches to integration

Option 1 – Point to Point



Financial-only ERP is connected directly to other supplementary tools either via API, webhooks or other services.

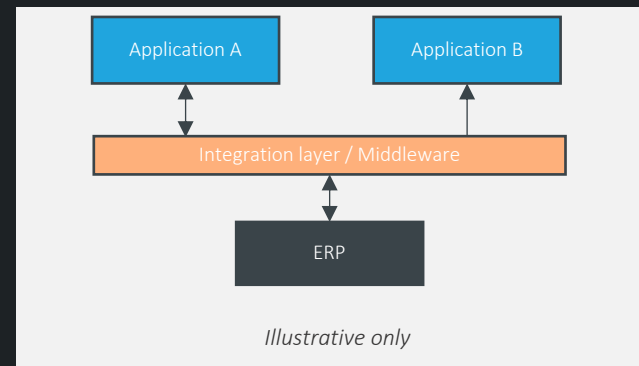
Key Advantages

- Good solution where there are limited integrations
- A potentially quicker implementation due to ability to leverage available adapters or connectors

Key Disadvantages:

- Limited re-useability and scalability across brands and it is likely that each brand will have to build direct integrations to the ERP based on their current state supplementary tools
- Result is a tightly coupled integration between endpoints, resulting to less software choice across the brands.
- Higher customisation requirement “inside the ERP”, i.e., this approach includes ‘connectors’ as well as ‘customising ERP to be the integration’.

Option 2 – Using Common Integration Layer



Financial-only ERP is connected mainly via a Common Integration Layer using an integration platform.

Key Advantages

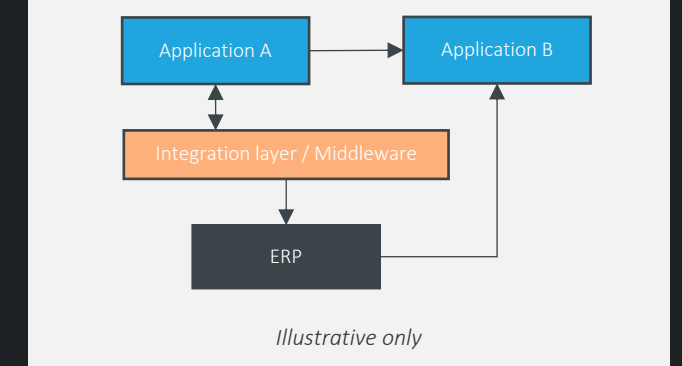
- Decoupled model that allows for more re-useability, scalability and more software choices across brand
- Provides each brand a common integration tool and standards to follow.

Key Disadvantages

- Can be costly depending on choice of tool for integration and the processes and customisation being supported by the tool.
- Low ROI on the tool or may require higher operational cost depending on the support model/structure of the selected tool.

RECOMMENDED APPROACH

Option 3 – Hybrid



Financial-only ERP is connected in multiple ways where it economically make sense, i.e., some point-to-point, some via an integration layer, and some as data loads.

Key Advantages

- Can leverage ‘best of both worlds’ between Option 1 and 2 depending on complexity and economical case.

Key Disadvantages

- A less simple architecture that require specific design depending on customisation needs, and may require different level of governance, support and maintenance
- May result to limited standardisation of approaches across brands if not managed at enterprise level.

Common Mistakes

- No Enterprise Architect or Architecture
- Point to point is easier!
- We selected a middleware because someone said it works well with this one application
- We need a low-code option for our enterprise sized organisation
- Don't worry, we'll look after our integrations
- Why do you need to know about our processes? It's just data.
- We want it all, and we want it now
- We want people to be able to update data from any system and it go everywhere else.

Low code might be easier but is it the right fit?

What do you want to do with your data?

Do end users need to see it?

Is it just for reporting?

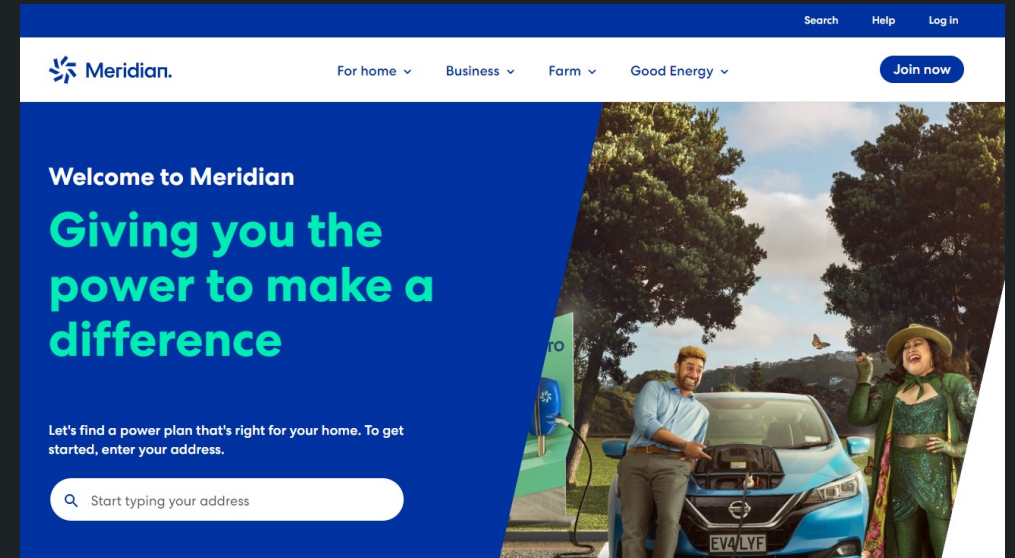
Is it integrated for a more complex need?

Is it real-time or scheduled?

What data volumes do you have?

Does the low-code solution perform efficiently?

How do you enforce efficient rules?



Middleware Platforms – Why choose one over the other?

Agnostic approach to middleware supported by product architecture specialists



Key Strengths	<ul style="list-style-type: none"> Architecture variety – custom built solution with provisioned and serverless tools Scalability 	<ul style="list-style-type: none"> Certified architect and development specialists at Fusion5 Native JDE Connectors Reduce development efforts with Prebuild Oracle integration. Inbuild sFTP server and B2B Support 	<ul style="list-style-type: none"> Fully managed service & platform Flexible integration solution that connects to any service Scalability (AWS) 	<ul style="list-style-type: none"> Wide range of connectors to reduce Development time API designer & management tools 	<ul style="list-style-type: none"> Low cost of entry with faster implementation (within months) High pricing flexibility / competitiveness 	<ul style="list-style-type: none"> Wide range of integration apps and connectors to reduce Development time Best integration with NetSuite ecosystem 	<ul style="list-style-type: none"> Simpler interface Lower cost in comparison with other iPaaS Quick deployment for available connectors or recipes 	<ul style="list-style-type: none"> Strong e2e data management (governance, cataloguing, warehouse etc). Can be quick to deploy but often longer for bespoke integrations
Key Weaknesses	<ul style="list-style-type: none"> High developments costs due to custom builds Speed of services and applications vary 	<ul style="list-style-type: none"> Major focus on Oracle SaaS products. Complex solution for simple Integrations Requirement 	<ul style="list-style-type: none"> IT team can only monitor not develop integration. 	<ul style="list-style-type: none"> Complex coding requiring strong Java developers Steep learning curve for adoption Can be high cost and slow implementation 	<ul style="list-style-type: none"> Jitterbit-script language cannot be ported to other platforms Less cloud centric 	<ul style="list-style-type: none"> Can be high cost and slow implementation NetSuite focused only Can be steep learning curve for adoption 	<ul style="list-style-type: none"> Performance concerns on large or complex integrations with high throughput needs Limited connectors or recipes and error management 	<ul style="list-style-type: none"> Limited middleware functionalities Hard to build or support event-based triggers No queuing OOTB need to add-on Azure Cost comparative to MuleSoft.
Best suited for	<ul style="list-style-type: none"> Organisations with Azure investment or Microsoft ecosystem High throughput integrations Non-standard integration requirements 	<ul style="list-style-type: none"> Best Suited for customers who have invested/planning to invest in Oracle cloud platform. Customers with complex integration requirements with Oracle Apps. Customer who wants to maintain platform by themselves. 	<ul style="list-style-type: none"> Organisations without dedicated integration / IT team Need connections to a wide variety of software, incl. proprietary software 	<ul style="list-style-type: none"> Mature and fully staffed and experienced IT organisation, preferably with Java expertise 	<ul style="list-style-type: none"> Organisations with non-coder technical personnel who wish to support integration. Want to license a platform with strong support structure 	<ul style="list-style-type: none"> Organisations with non-coder technical personnel who wish to support integration. 	<ul style="list-style-type: none"> Organisations with simple integration with technical team to manage errors and debugging Organisations wanting a low-cost platform and does not require high data throughput 	<ul style="list-style-type: none"> Organisations that are looking more on data lake and warehouse rather than integrations. OR those with existing middleware. Medium to large organisations only. Small organisations will find it costly and

Common Mistakes

- No Enterprise Architect or Architecture
- Point to point is easier!
- We selected a middleware because someone said it works well with this one application
- We need a low-code option for our enterprise sized organisation
- Don't worry, we'll look after our integrations
- Why do you need to know about our processes? It's just data.
- We want it all, and we want it now
- We want people to be able to update data from any system and it go everywhere else.

Step 1 in reviewing middleware products

What you want to take ownership for is a key driver



We will take responsibility for the build and support of our integrations



We would like your help to build and support our integrations



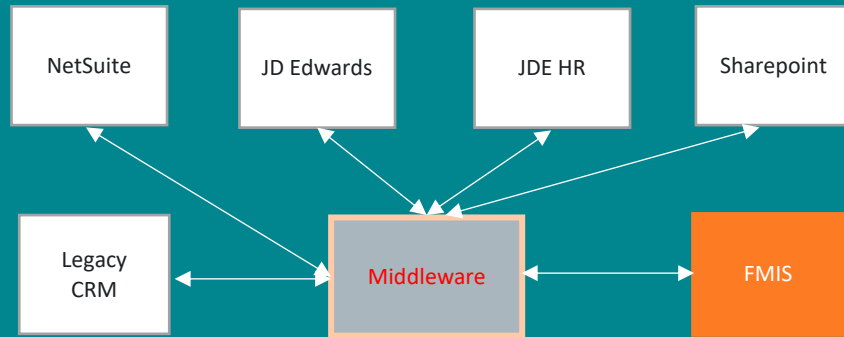
We don't want to deal with integration

Common Mistakes

- No Enterprise Architect or Architecture
- Point to point is easier!
- We selected a middleware because someone said it works well with this one application
- We need a low-code option for our enterprise sized organisation
- Don't worry, we'll look after our integrations
- Why do you need to know about our processes? It's just data.
- We want it all, and we want it now
- We want people to be able to update data from any system and it go everywhere else.

Discuss and align on critical data flows

Students/Invoices/Payments



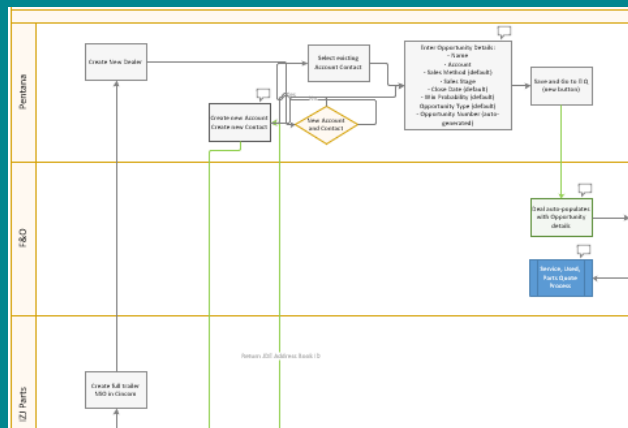
1. Customer Data
2. Invoices
3. Payments
4. Items
5. Inventory

Alignment Changes/Additions

For the past alignment sessions, what are the changes?

1. JD Edwards has not current RTE's setup to schedule data extracts
2. NAV instances are a combination of on-premise and cloud
3. On-premise systems will need an agent installation
4. Need to start working with IT support to pass the security audit

Business Process Flows - Detail



Example:

1. Customers created in legacy CRM
2. On create, triggers create in JDE
3. Student receivables amount updated daily to legacy CRM
4. Invoices created in JD Edwards
5. PDF of invoice available in Sharepoint
6. Link to PDF created in CRM

Alignment Changes/Additions

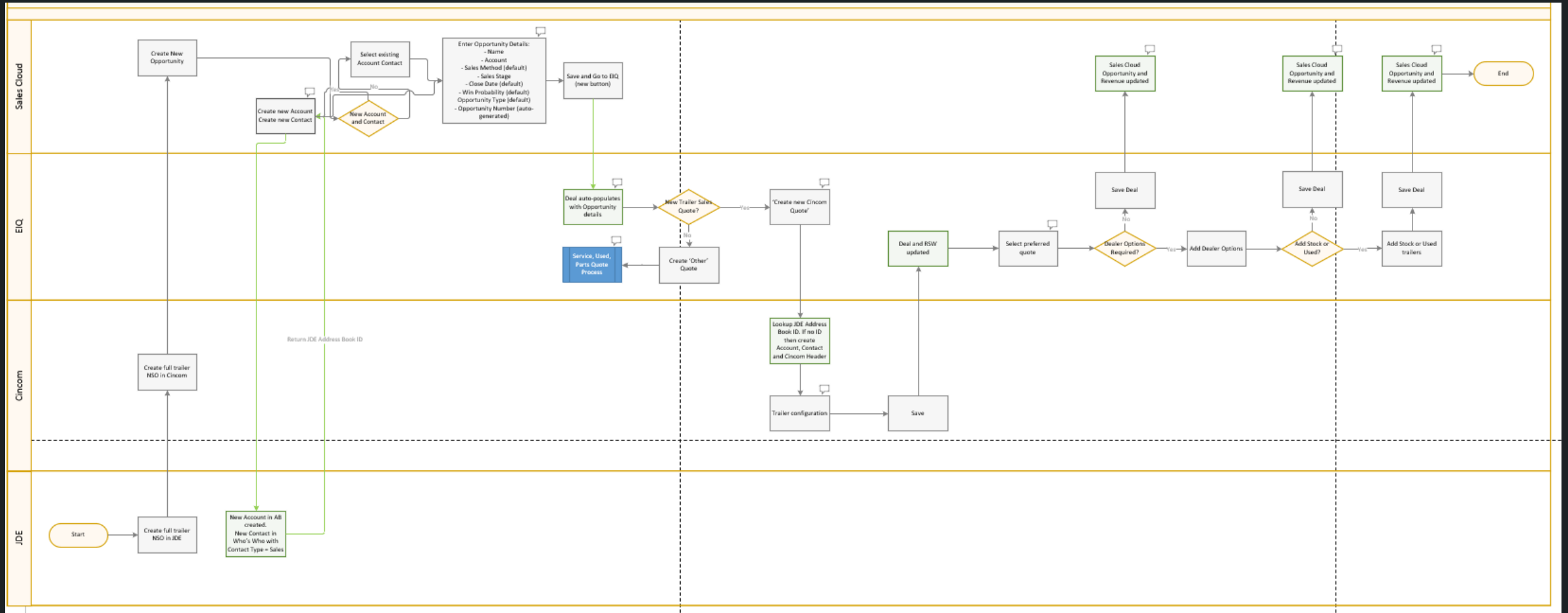
For the past alignment sessions, what are the changes?

1. Data misalignment found between legacy CRM and JD Edwards for customer data
2. Business logic is different depending on Customer Type which changes the NetSuite form
3. Some reporting fields found to be free-text in source of truth
4. Are the amounts more critical or having a holistic view of financials by line item?

Common Mistakes

- No Enterprise Architect or Architecture
- Point to point is easier!
- We selected a middleware because someone said it works well with this one application
- We need a low-code option for our enterprise sized organisation
- Don't worry, we'll look after our integrations
- Why do you need to know about our processes? It's just data.
- We want it all, and we want it now
- We want people to be able to update data from any system and it go everywhere else.

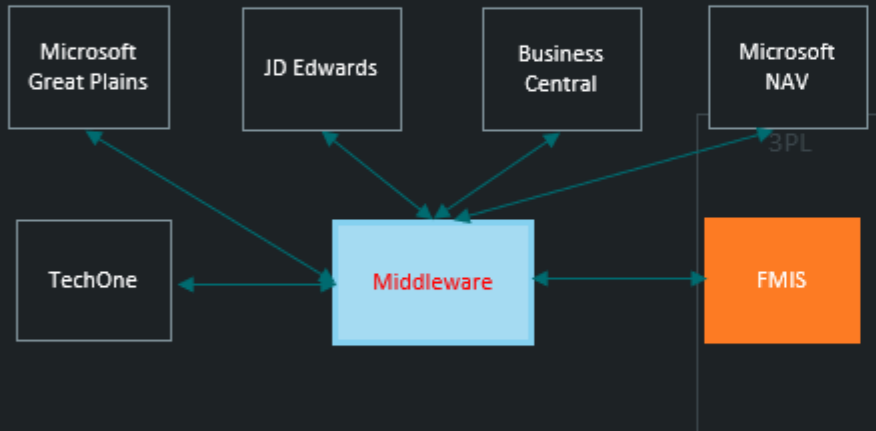
IF THIS IS CRITICAL, HOW WAS IT OVERLOOKED?



Common Mistakes

- No Enterprise Architect or Architecture
- Point to point is easier!
- We selected a middleware because someone said it works well with this one application
- We need a low-code option for our enterprise sized organisation
- Don't worry, we'll look after our integrations
- Why do you need to know about our processes? It's just data.
- We want it all, and we want it now
- We want people to be able to update data from any system and it go everywhere else.

SOURCE OF TRUTH



- Student (Master) –
- Student (Replicated) –
- Invoices (Master) –
- Invoices (Replicated) –
- Product / Items / Pricing (Master) –
- Product / Items / Pricing (Replicated) –
- Payments (Master) –
- Payments (Replicated) –

So, in Summary

- ⇒ Enterprise Architecture is a must
- ⇒ Assess and decide on the strategy you will be utilising for your integrations
- ⇒ Every middleware has its strengths and weaknesses
- ⇒ Low code is not suitable for anything other than basic integrations
- ⇒ Be honest about your internal capability
- ⇒ The business process forms the basis of the integration architecture
- ⇒ Identify critical processes to implement
- ⇒ If you don't have a source of truth, you end up with a looping mess of data

Thank You!

