

ACCOUNTING INTEGRATION BLUEPRINT

A best practice guide to
integrating your Purchase
Orders & Invoices



CONTENTS

OBJECTIVES	01
INTEGRATION REQUIREMENT GATHERING	02
DATA MAPPING	03
API VS ETL	05
SOLUTION #1	07
SOLUTION #2	11
SOLUTION #3	15
API WORKFLOWS	18

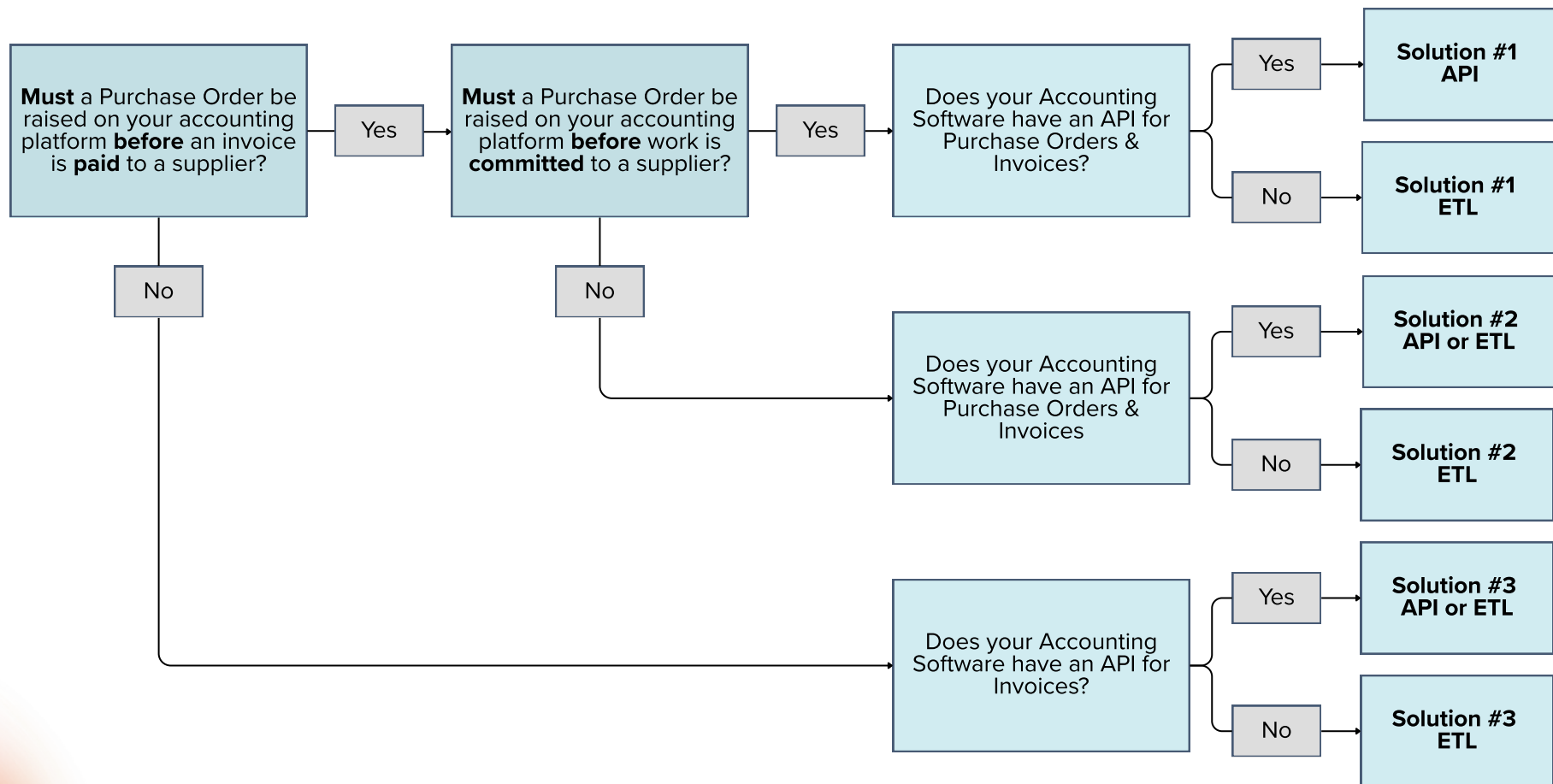
OBJECTIVES

So, you've identified an opportunity to streamline the PO and Invoice process by integrating expansive FM (CAFM) and your accounting platform.

No doubt your objectives will include:

- ✓ Avoid any double handling of PO and Invoice data
- ✓ Mirror any existing PO approval processes
- ✓ Speed up internal processing of POs and Invoices
- ✓ Ensure PO and Invoice data is complete with Cost Centres, Supplier Numbers, Nominal Codes, Budget Tagging etc
- ✓ Speed up external comms with suppliers
- ✓ Achieve a more efficient workflow process resulting in no invoice discrepancies
- ✓ Improve supplier relations due to no invoice discrepancies
- ✓ Support the FM team with a greater depth of financial reporting from within CAFM

INTEGRATION REQUIREMENT GATHERING



DATA MAPPING

An indication of what data
we would expect to be
migrating and mapping as
part of an integration

FINANCE DATA MAPPING

Client Data	EFM Mapping	
Legal Entity Codes	Organisation Reference	Your legal entity or organisation reference
Site Cost Centres	Site Cost Centre	The site's unique accounting reference
Spend Categories	Service Type	Sometimes used to categorise spend other than nominal codes and budgets
Nominal Codes	Cost Code	Your nominal code or general ledger account code
Budget Codes	Budget Name	The unique reference for your budget – if different to your nominal code
Supplier IDs	Contractor External ID	The supplier's reference in your accounting platform
User IDs	User External ID	Used to identify the user that approve quotes and approve spend
Tax Codes	Tax Rate External Code	Used to correctly match any tax rates you may have stored in your accounting platform
Currency Code	Currency Code	Currency code such as GBP, EUR etc

API VS ETL

What's the difference?

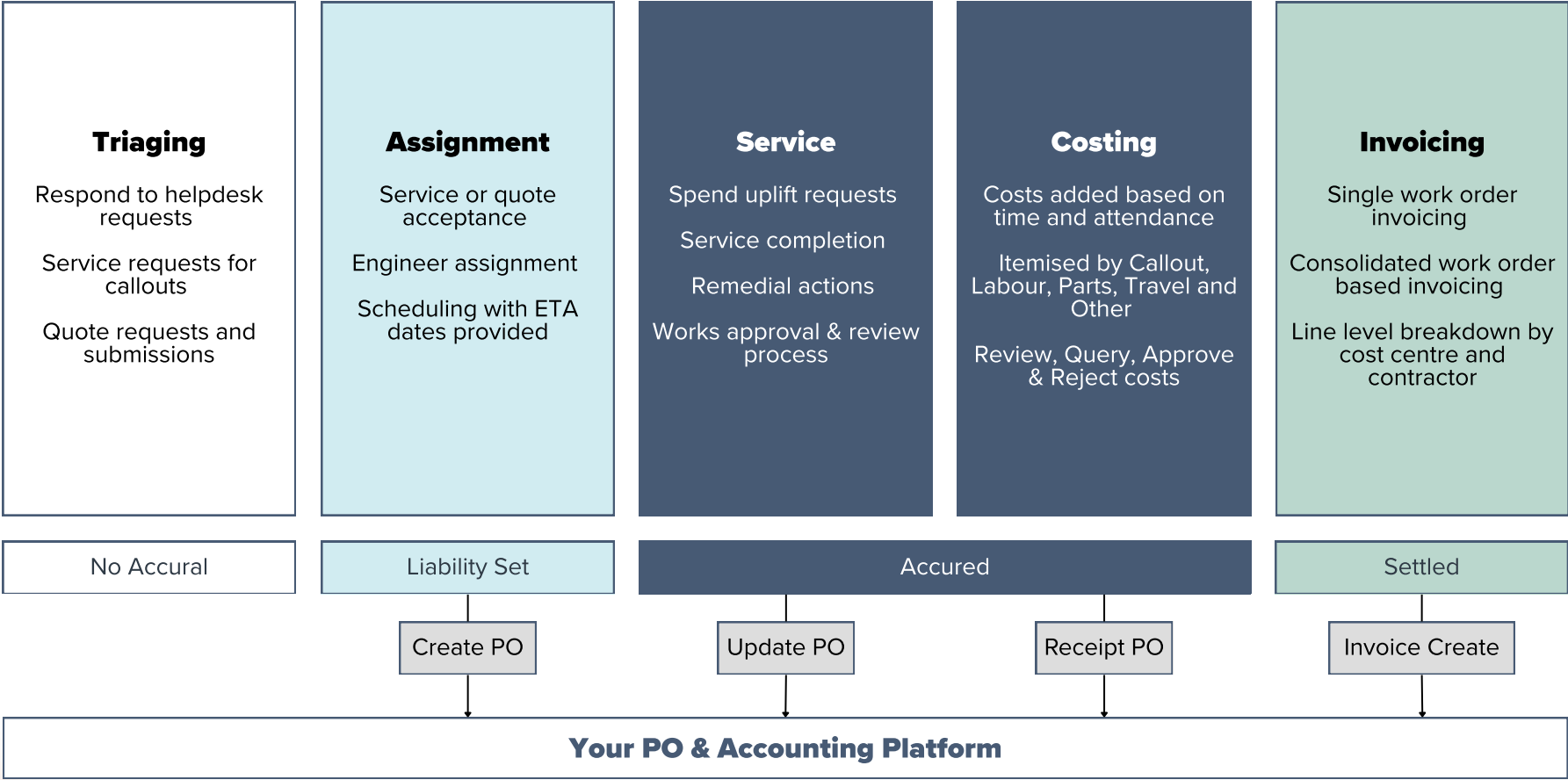
API VS ETL

	API Application Programming Interface	ETL Extract Transform Load
Pros	More real-time Better monitoring Instant feedback / responses on requests Easier to remedy data fixes e.g. PO not created	Less Dev time Often quicker to test and implement Often quicker to make file adjustments if changes are required
Cons	Takes longer to implement / test When EFM product is enhanced / changes are made, this could lead to future changes to the API layer	Requests are often less frequent Monitoring not instantanous Data corrections are harder to disagnose

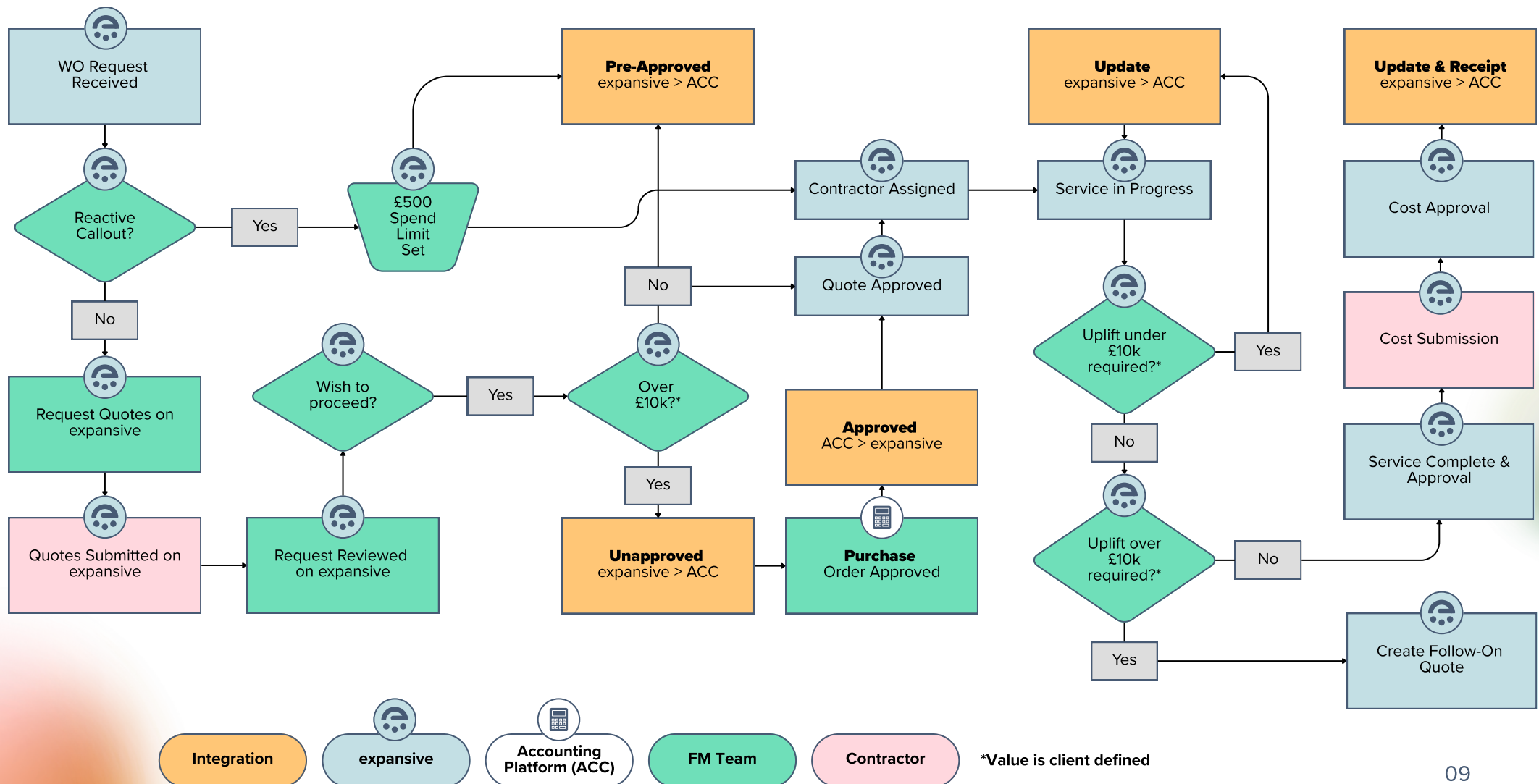
SOLUTION #1

**Purchase orders raised on
commitment**

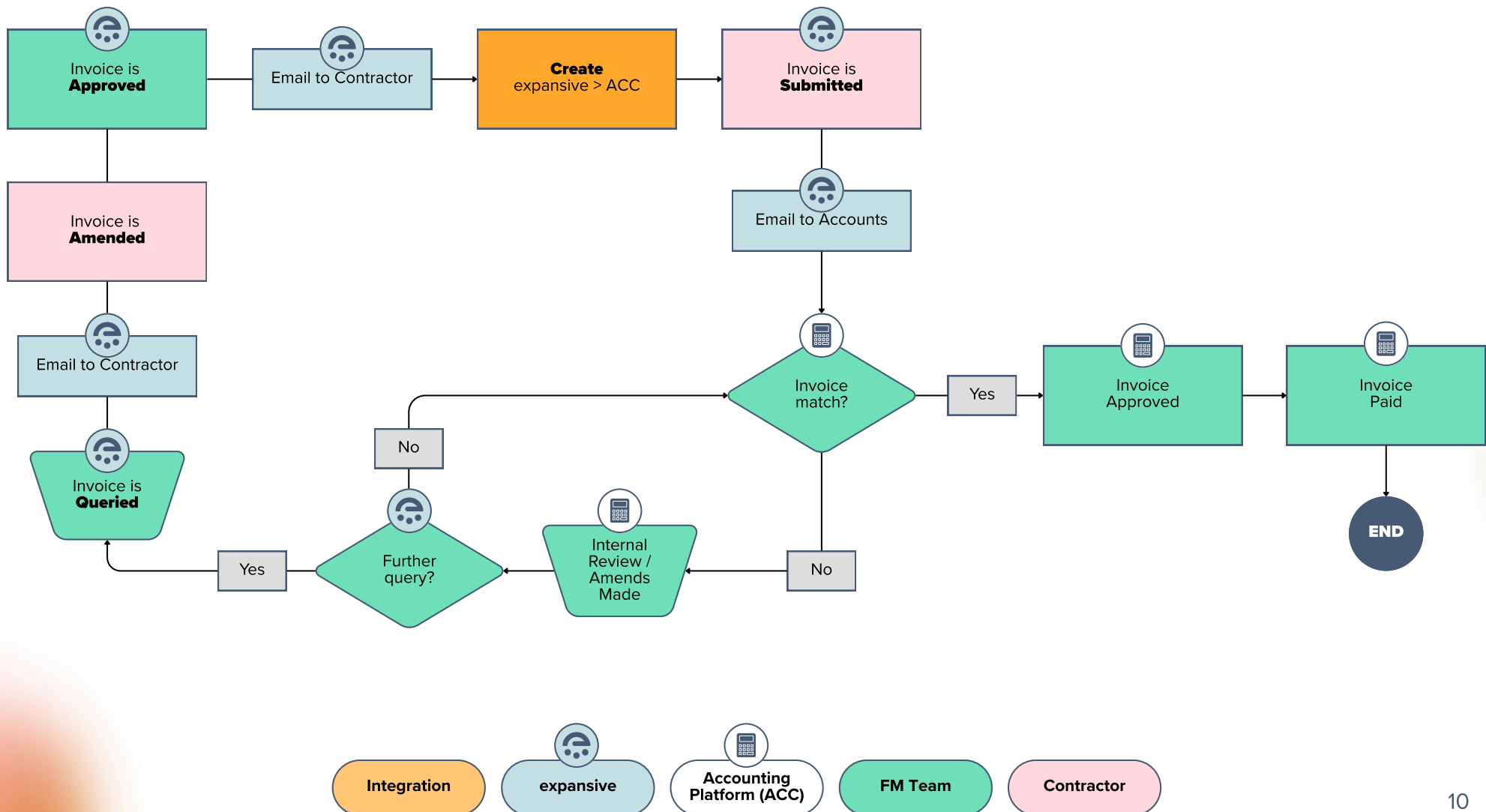
SOLUTION #1 - HIGH LEVEL



SOLUTION #1 - PO PROCESS FLOW



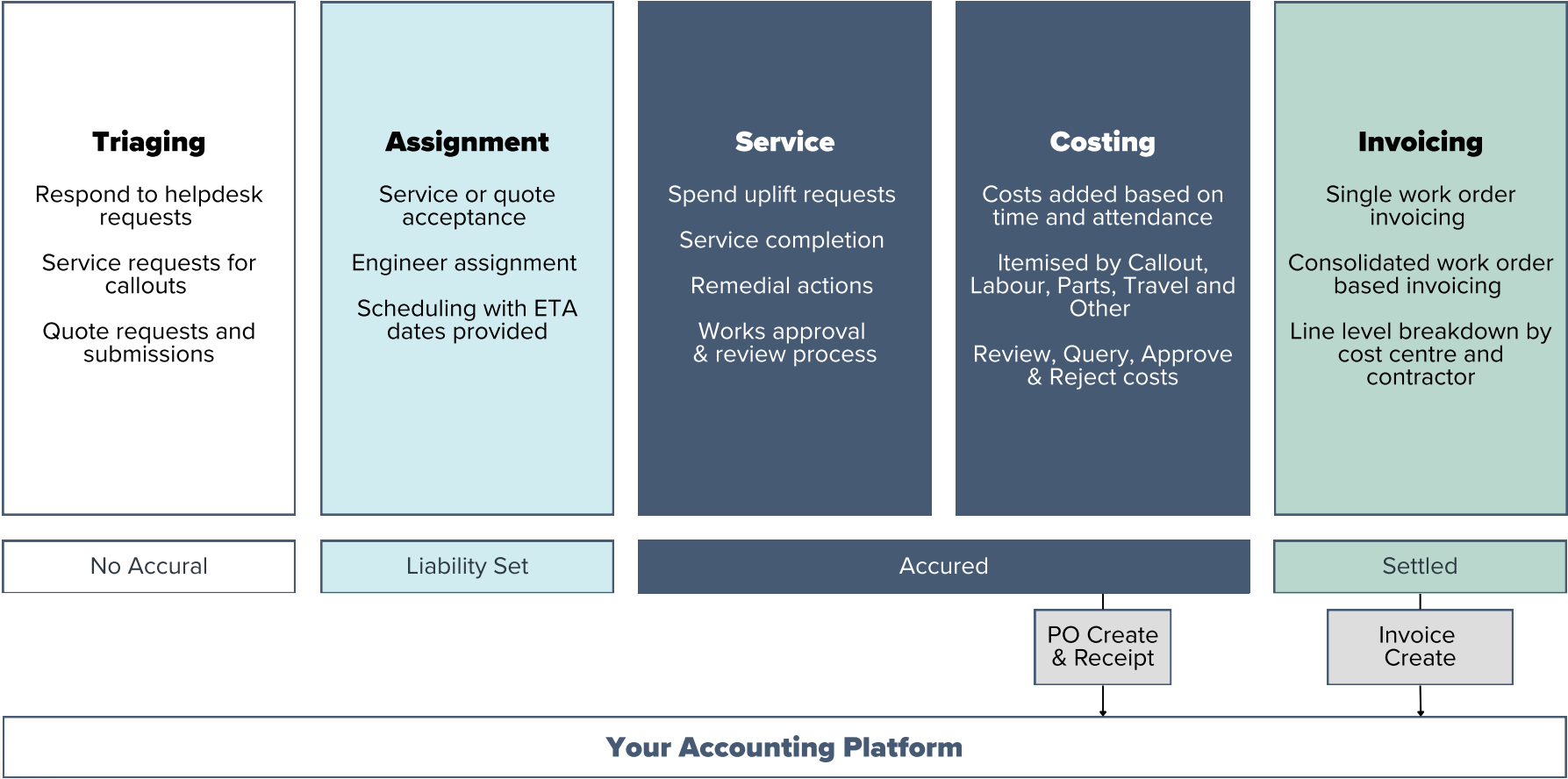
SOLUTION #1 - INVOICE PROCESS FLOW



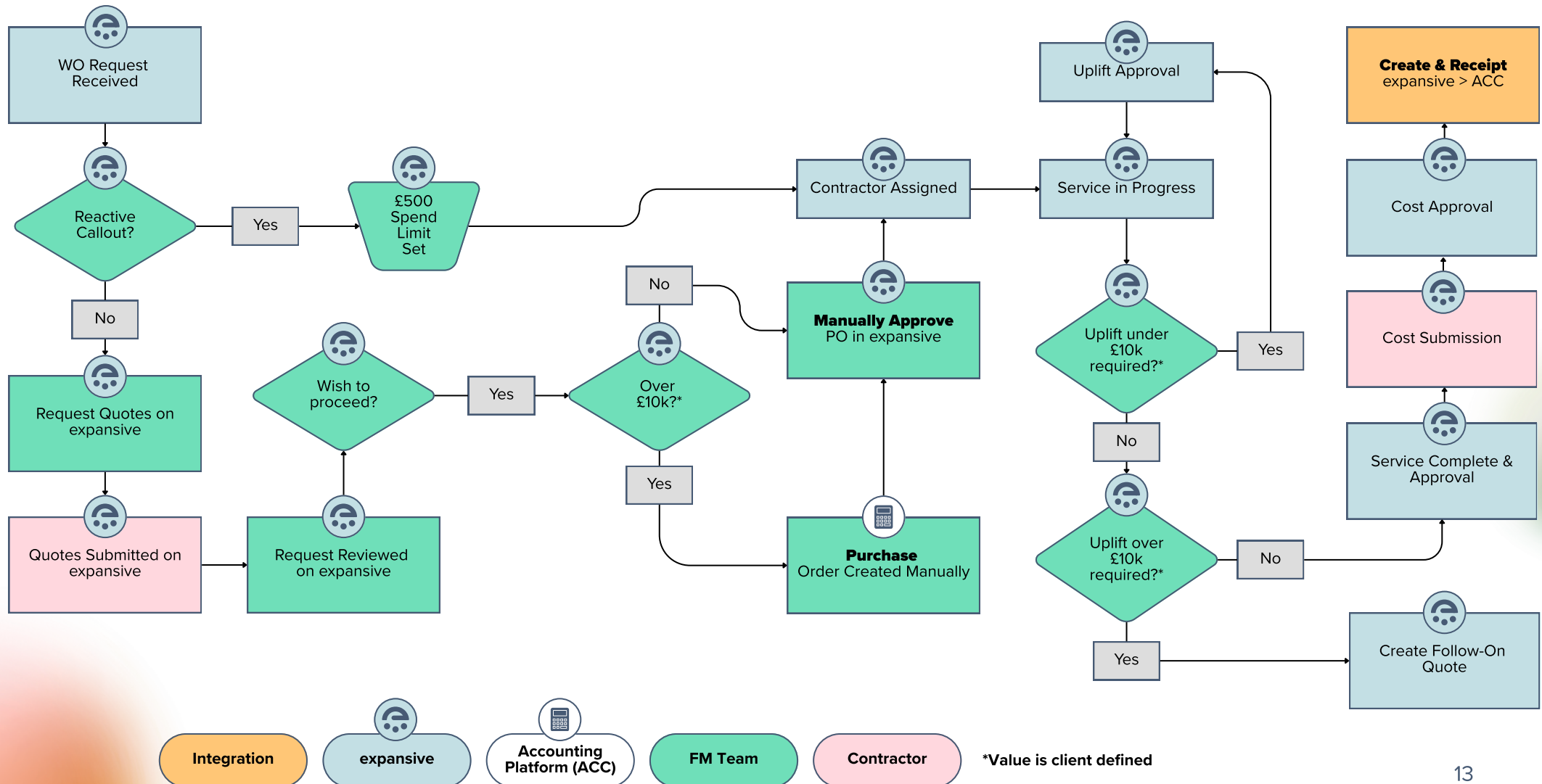
SOLUTION #2

**PO raised and receipted
when complete and costs are
approved**

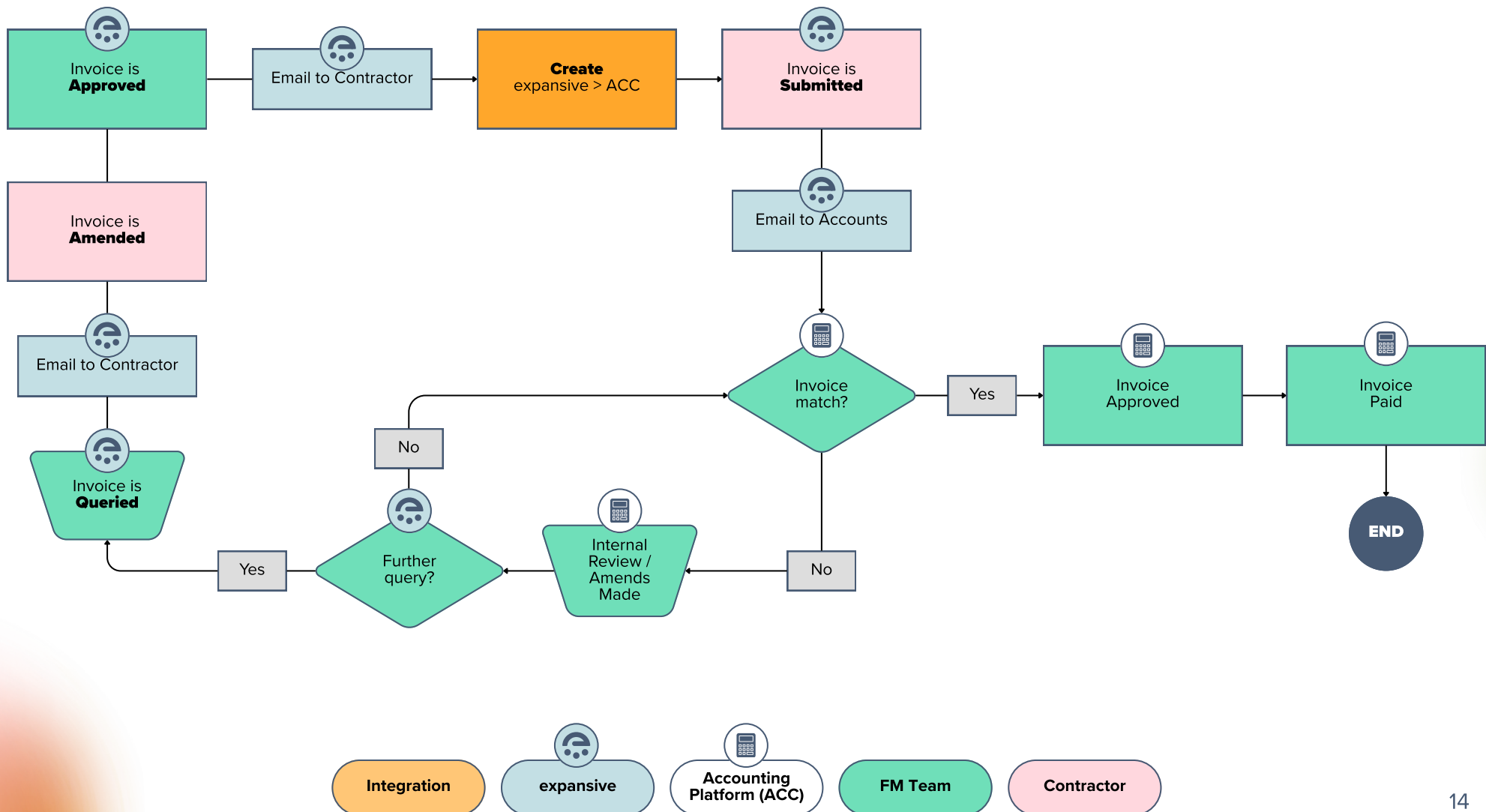
SOLUTION #2 - HIGH LEVEL



SOLUTION #2 - PO PROCESS FLOW



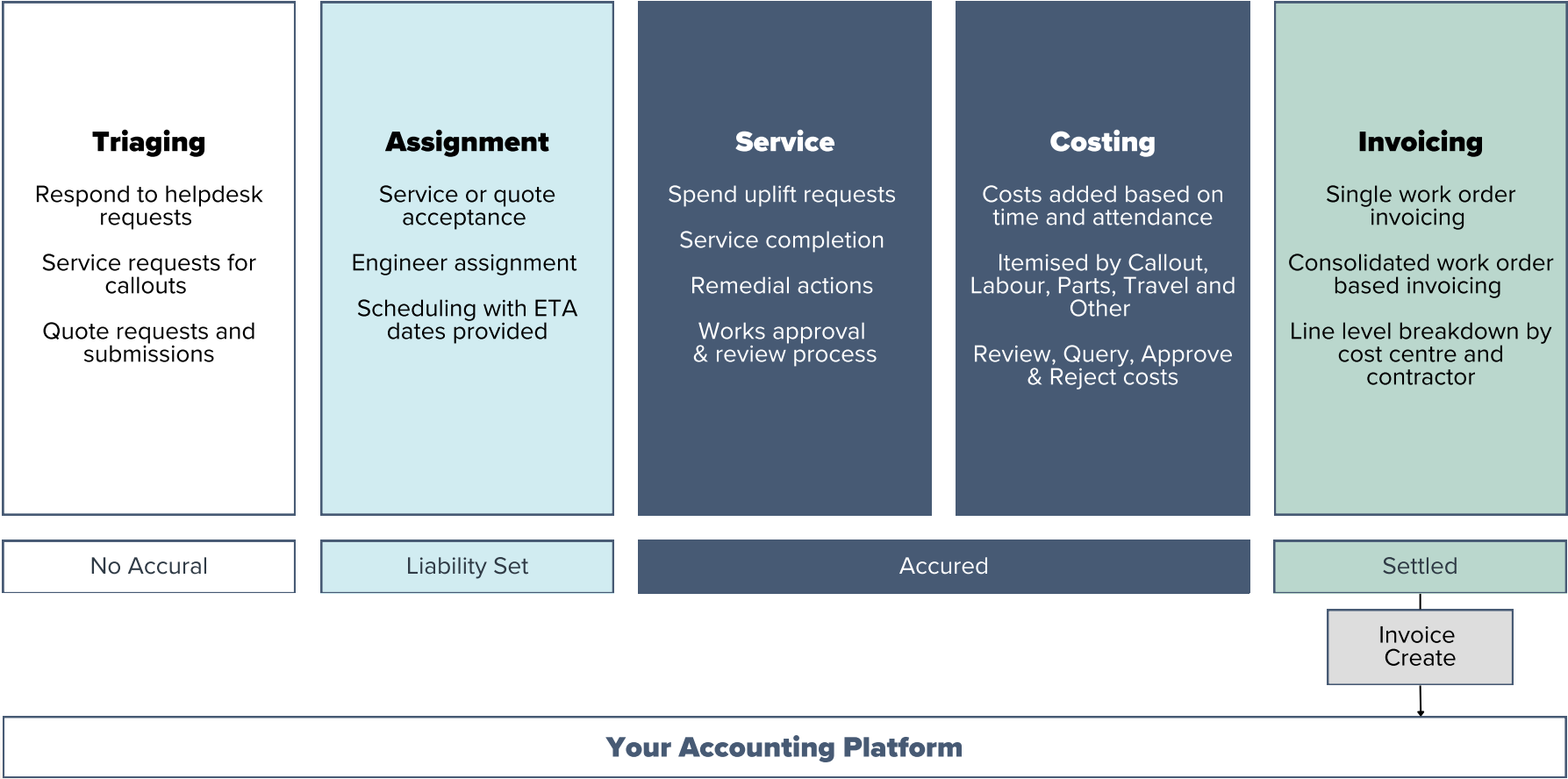
SOLUTION #2 - INVOICE PROCESS FLOW



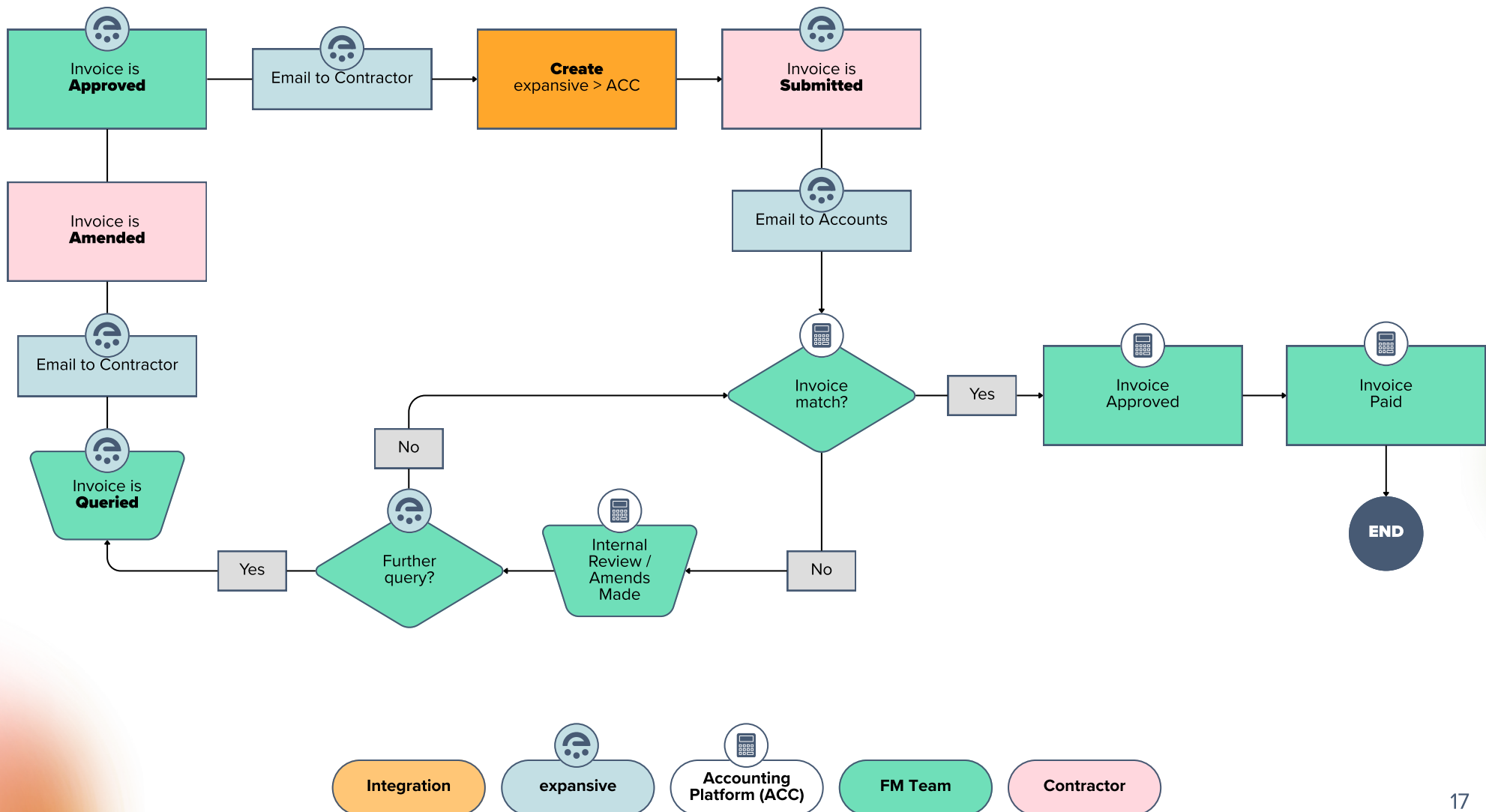
SOLUTION #3

**Invoice only with no
accounting PO requirement**

SOLUTION #3 - HIGH LEVEL



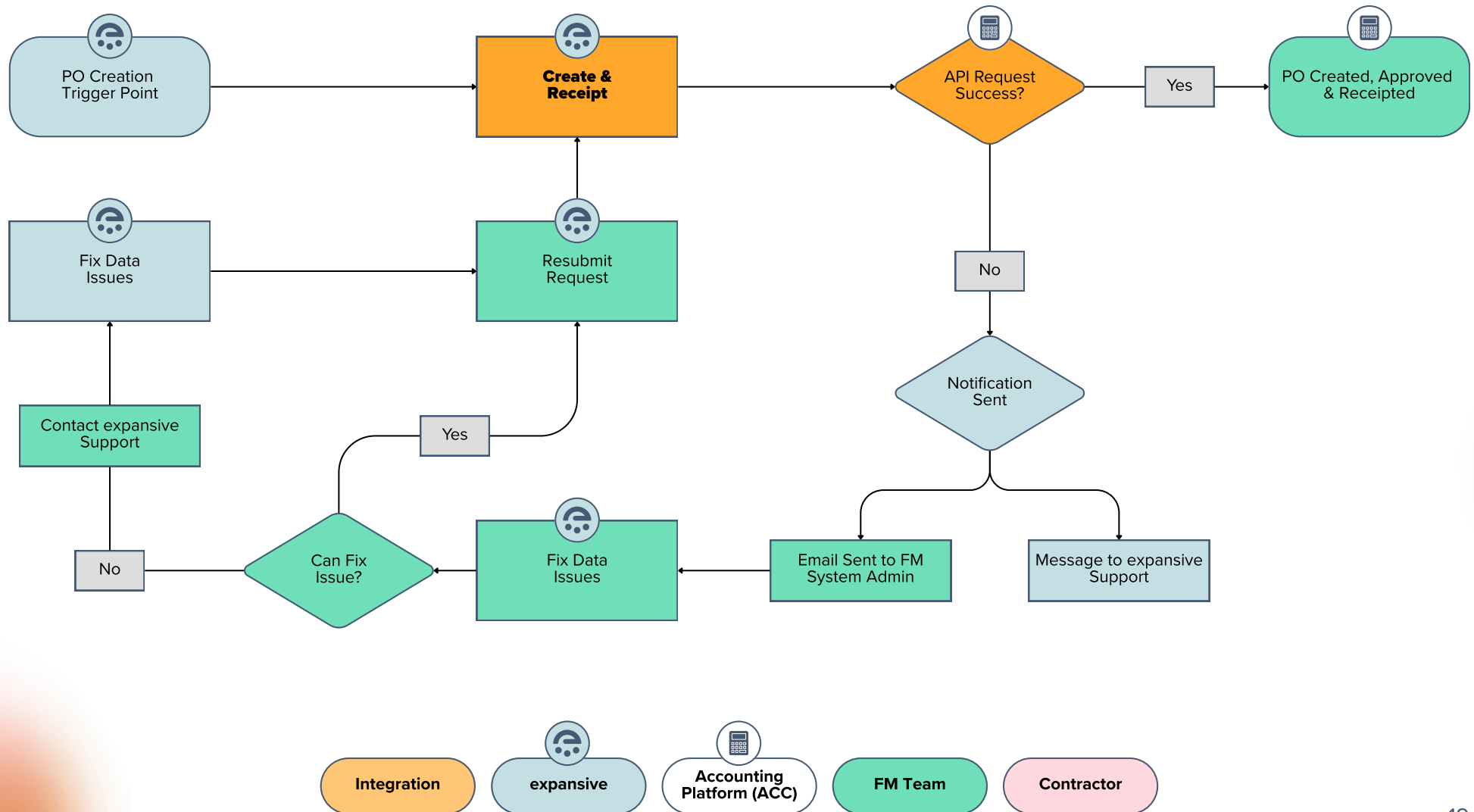
SOLUTION #3 - INVOICE PROCESS FLOW



API WORKFLOWS

The process to keep data
flowing for PO & Invoices

PURCHASE ORDER - API WORKFLOW



INVOICE - API WORKFLOW

