

ERP-Additive Procurement Integration Guide

Plan procurement integration around the ERP with clear object ownership, mappings, timing patterns, security, monitoring, reconciliation, testing, and support.

[Enterprise guide · V4.1 creative review edition](#)

Review context

Audience: Enterprise applications, integration engineering, procurement technology, finance systems, security, architecture, and procurement operations

Author: VendrNova Integration & Architecture Team

Review: Reviewed by VendrNova Product and Procurement Leadership

This public-only review resource contains no submission, analytics, tracking, or production legal activation.

Executive summary

ERP-additive procurement architecture preserves the ERP as the financial system of record while using a procurement orchestration layer to coordinate requests, suppliers, evidence, approvals, exceptions, execution context, and system handoffs. The design does not ask one system to own every field. It defines ownership by object and state.

VendrNova has confirmed production integrations and reusable bidirectional frameworks for SAP ECC, SAP S/4HANA, Oracle ERP, Microsoft Dynamics 365, Microsoft Dynamics NAV / Navision, and NetSuite. Those frameworks still require customer-specific configuration, mappings, policies, workflows, security design, validation, and support. Scheduled synchronization is the normal pattern, approximately every two hours, while real-time synchronization is available for selected objects and customer requirements.

This guide turns that position into an implementation planning method.

Table of contents

1. Set the system-of-record boundary
2. Inventory synchronized objects
3. Choose scheduled and real-time patterns
4. Design customer-specific mappings
5. Govern identity, security, and access
6. Build monitoring, notification, and reconciliation
7. Define ownership and support
8. Plan implementation and testing
9. Cut over and operate

1. Set the system-of-record boundary

Begin with a one-page responsibility model.

Procurement orchestration typically owns

- business request and context;
- workflow route and current owner;
- sourcing and supplier interaction evidence;
- approval, return, rejection, and exception history;
- supplier onboarding and risk-workflow context;
- award rationale;
- purchase execution context;
- receipt or invoice-exception collaboration;
- audit context for decisions completed in the platform.

ERP typically remains authoritative for agreed financial records

- company and organizational structures;
- cost centers and budgets according to customer design;
- supplier/vendor master identifier;
- purchase-order financial record where agreed;
- goods receipt or GRN financial/stock record where agreed;
- supplier invoice accounting record;
- payment and reconciliation status;
- financial posting.

The exact ownership can differ. Record the agreed source for every field and state.

Boundary questions

- Which system creates the object identifier?
- Which system may update each field?
- Which event makes a state final?
- Which data is referenced rather than copied?
- How is a conflict resolved?
- How does a rejected record return for correction?
- What evidence must remain visible to users?

2. Inventory synchronized objects

Common object families can include:

- supplier/vendor master data;
- company and organizational structures;
- cost centers and budgets;
- users and approval data;
- requisitions;
- sourcing awards;
- contracts;
- purchase orders;
- goods receipts and GRNs;
- invoices;
- payment and reconciliation status.

Object inventory fields

For each object, document:

- object ID and name;
- business purpose;
- source and destination system;
- create/update/cancel direction;
- authoritative fields;
- required references;
- expected volume;
- timing requirement;
- retention;
- sensitivity;
- validation;
- error owner;
- reconciliation method.

Do not infer that because an ERP offers an API, every object or field is suitable for the intended integration. Official SAP, Oracle, Microsoft, and NetSuite documentation shows different API, OData, record, custom-service, and batch patterns. Customer product version, permissions, available endpoints, data volumes, and business rules still govern the design.

3. Choose scheduled and real-time patterns

Use the least complex pattern that meets the business requirement.

Scheduled pattern

Use when:

- the process can tolerate bounded delay;
- volume favors batch processing;
- downstream processing is periodic;
- the object does not require an immediate user decision;
- retry and reconciliation are simpler in batches.

VendrNova's normal scheduled synchronization is approximately every two hours, subject to customer architecture and object design. This is an operating baseline, not a universal service-level guarantee.

Real-time pattern

Use when:

- a user cannot proceed without the response;
- an approval or action needs current state;
- the object must be validated before commitment;
- event timing materially affects control or experience.

Real-time synchronization is available by object and customer requirement. "Real time" still requires timeout, retry, duplicate, failure, and fallback design.

Hybrid pattern

Many architectures use real-time validation or status for selected events and scheduled synchronization for bulk updates, reconciliation, or reference data.

Pattern decision record

Record business need, maximum acceptable delay, volume, peak load, dependency, failure consequence, retry, fallback, monitoring, and cost. Do not choose real time simply because it sounds more advanced.

4. Design customer-specific mappings

Mapping is business design, not clerical translation.

Mapping workbook

Include:

- source object and field;
- destination object and field;
- business definition;
- data type and format;
- required/optional;
- allowed values;
- transformation;
- default;
- validation;
- reference or code list;
- ownership;
- privacy classification;
- test case;
- decision status.

High-risk mapping areas

- supplier identity and duplicate resolution;
- company, plant, and purchasing organization;
- cost center, general ledger, project, and budget;
- currency, tax, units, and rounding;
- user and approver identity;
- category and commodity;
- contract and sourcing references;
- purchase-order type and status;
- receipt quantity and service entry;
- invoice and credit/debit note;
- payment and reconciliation status.

Use synthetic or protected test data. Do not expose production credentials or personal data in documentation.

Mapping governance

Every transformation needs an owner and reason. Treat default values cautiously: a default can hide missing information and create an apparently successful but incorrect transaction.

5. Govern identity, security, and access

Integration security should cover:

- system and service identities;
- authentication method;
- least necessary permissions for the specific design;
- credential and secret handling;
- network path;
- data classification;
- encryption requirements approved for the architecture;
- logging and monitoring;
- separation of development, test, and production;
- change approval;
- emergency access;
- incident and support process.

VendrNova can configure enterprise single sign-on subject to supported identity architecture and implementation scope. This guide does not claim universal identity-provider support, SCIM, or a specific encryption/control implementation.

Access decision

For each integration action, document:

- who or what initiates it;
- object and operation;
- allowed environment;
- authorization;
- logging;
- approval;
- review frequency;
- revocation.

6. Build monitoring, notification, and reconciliation

Successful transport is not the same as successful business processing.

State model

Use explicit states such as:

- queued;
- sent;
- transport accepted;
- business accepted;
- rejected;
- retry scheduled;
- mapping issue;
- duplicate held;
- corrected;
- reconciled;
- closed.

Monitoring record

Capture:

- integration and object;
- source identifier;
- destination identifier;
- event timestamp;
- payload or protected evidence reference;
- status;
- rejection or error category;
- retry count;
- owner;
- next action;
- resolution;
- reconciliation status.

VendrNova integration operations monitor failed and rejected records, retries, and mapping issues, with notification support for VendrNova and the client. Notification method and service level depend on the customer arrangement.

Reconciliation

Reconcile counts, identifiers, amounts or quantities where appropriate, status, and unresolved errors. Define tolerance and aging. A dashboard should let an owner move from aggregate status to the affected record without exposing unnecessary sensitive data.

7. Define ownership and support

Responsibility model

- procurement process owner: business behavior and workflow;
- ERP owner: financial system and object policy;
- integration owner: design, mappings, monitoring, and correction;
- security owner: architecture and access review;

- data owner: definition and quality;
- support owner: incident intake and coordination;
- business owner: validates process outcome;
- vendor or supplier contact: corrects supplier-controlled information where appropriate.

VendrNova owns post-go-live integration support under the customer's agreed support arrangement. Commercial terms and service levels vary.

Support runbook

Include:

- supported objects and interfaces;
- hours and contact path under the agreement;
- severity definitions;
- triage information;
- ownership and escalation;
- safe replay and correction process;
- communication;
- change control;
- known dependencies;
- reconciliation;
- closure evidence.

8. Plan implementation and testing

Planning sequence

10. Confirm workflow and system boundary.
11. Inventory objects and versions.
12. Validate endpoints, permissions, and environments.
13. Approve mappings and reference data.
14. Configure workflow and integration.
15. Prepare synthetic and protected test data.
16. Test unit, contract, end-to-end, exception, security, performance, and recovery behavior.
17. Reconcile results.
18. Run business acceptance.
19. Approve cutover and support readiness.

Test matrix

Test:

- valid create, update, cancel, and status;
- missing required field;
- invalid code;
- stale reference;
- duplicate message;
- out-of-order event;
- timeout;
- transport failure;
- ERP business rejection;
- mapping failure;
- partial batch;

- retry;
- correction;
- authorization failure;
- volume and peak;
- reconciliation mismatch.

An integration is not ready because the happy path succeeds once.

9. Cut over and operate

Cutover checklist

- To do: Scope, objects, mappings, and versions are approved.
- To do: Production identities and access are approved.
- To do: Monitoring and notification are active.
- To do: Initial reference data is synchronized and reconciled.
- To do: Open transactions have a migration or coexistence rule.
- To do: Rollback or pause criteria are defined.
- To do: Business, ERP, integration, security, and support owners are available.
- To do: User communication and support paths are ready.
- To do: First-day and first-week reconciliation is scheduled.

Post-go-live review

Review accepted/rejected rates, error mix, retry, aging, data quality, business outcome, support demand, and changes. Separate design defects from training issues, master-data problems, and external system behavior.

Change control

Trigger review when the ERP version, endpoint, object, mapping, policy, workflow, identity, volume, data classification, or support arrangement changes.

Architecture workshop output

Produce:

20. system context and responsibility diagram;
21. object inventory;
22. mapping workbook;
23. pattern decision record;
24. security and identity review;
25. monitoring and reconciliation design;
26. test plan;
27. cutover and support runbook;
28. open decision and dependency log.

Continue the review

Discuss your architecture: <https://vendrnova-public-launch-v4-1.neetlife.chatgpt.site/book-a-demo>