

Plan an integration before building a connector.

Bring data owners and system maintainers together to agree the business meaning and recovery path.

Integration planning agenda

Topic	Decision required
Authority	Which system owns each field and business transition?
Identity	How will applications authenticate and receive permission?
Consistency	Which changes must be immediate and which may arrive later?
Recovery	Who investigates mismatches and safely replays work?

Use real records in the workshop

Include duplicate customers, cancelled orders and incomplete fields. These examples reveal assumptions that a perfect sample payload cannot expose.

Record an exception path

A connector cannot decide every business conflict. Define the queue, context and person needed when the integration cannot proceed safely.

Bring a real transaction to the workshop

Choose one transaction that currently creates delay or manual rework. Bring a redacted example, the systems involved and the people who understand the exceptions. A discussion about an actual order, account update or invoice is more productive than a list of desired connectors.

Trace where the transaction starts, which system gives it an identifier and which system is allowed to change it. Note every manual check along the way. Some checks exist because the process is unclear, while others protect an essential business rule. The integration plan needs to distinguish those cases before deciding what to automate.

Resolve ownership before choosing a connector

For each shared field, identify the authoritative source and the allowed direction of change. If both systems can edit the same value, define how a conflict is detected and resolved. Do not assume that the most recently received update is the most correct one.

Then examine authentication, rate limits, available events and reconciliation options. A provider may expose an API without supporting the delivery guarantees your workflow needs. Document those constraints and try a small end to end transaction in a safe environment. Include duplicate requests and a temporary outage so the plan accounts for recovery as well as successful delivery.

Leave with decisions someone can implement

The workshop output should name the first integration boundary, its owner, the expected data contract and the acceptance evidence. List unanswered questions separately with a person responsible for resolving each one. This keeps an unknown provider limit from disappearing into an optimistic delivery estimate.

Agree how the organisation will detect and resolve records that fail to synchronise. A queue with no owner simply moves the manual problem somewhere less visible. Define the operational view, access needed to investigate and the conditions under which replay is safe. Use that evidence to scope the first delivery increment and any follow-up work.

What should the workshop produce?

A data mapping, a boundary diagram, representative examples, an exception process and the assumptions that still need validation.

Read the current resource online

<https://cobnex-review.rgcsekaraa.workers.dev/resources/partnerlocator/>