INOVE.

QUICK GUIDE · INTEGRATION

# Integration that doesn’t break

Seven questions to ask **before** connecting two systems — and what must be written into the contract.

inovesolutions.com

THE REAL PROBLEM

# Integration doesn’t fail at the connection. It fails at the agreement.

Connecting two systems is the easy part. There is an API, a connector, a middleware. What breaks integration projects is almost never technical: it is **what nobody agreed before starting**.

The pattern repeats. The integration goes live, works for weeks, and then an order arrives twice. Or disappears. Or arrives with the right amount and the wrong date. That is when the expensive part begins: working out whose fault it is, when nobody defined whose responsibility it was.

> **What this guide solves**
> The seven questions below cost one meeting to answer and prevent the four problems that show up most: duplication, silent loss, figures that disagree, and a dependency nobody mapped.

## Why the technical conversation comes second

Choosing between API, file or queue is a decision about means. It only makes sense once it is clear **what needs to happen** — how often, with what guarantee, and what to do when it fails. Starting from the tool is what makes an integration be born without a contract.

THE SEVEN QUESTIONS

# Ask them before writing the first line

**1. Who owns the data?**
If both sides can change the same record, you don't have an integration — you have a scheduled conflict. Define which system rules each field, and what happens when both change at once.

**2. What exactly is an event?**
"When the order is approved" sounds clear until someone approves, cancels and approves again. Write the definition in one sentence and test it against three odd cases before accepting it.

**3. What delay can the other side live with?**
Real time costs far more than every fifteen minutes, and the business almost never needs real time — it needs **predictability**. Ask which delay breaks the process, not which would be ideal.

**4. What happens when the destination is down?**
Resend, queue, discard, or alert? Without that answer, the default behaviour is to lose in silence — and nobody finds out until closing.

**5. How do you know it arrived?**
Sending is not delivering, and delivering is not processing. Each step needs its own confirmation. Without them, the first news of a failure comes by phone, from the customer.

**6. And if it arrives twice?**
It will. Networks repeat, retries duplicate, someone reprocesses. The integration must recognise what it has already seen — by identifier, not by resemblance.

**7. Who gets told, and how fast?**
An alert with no owner is not an alert, it is a log. Define the person, the channel and the window in which someone needs to know — before the next process starts going wrong.

IN THE CONTRACT

# Four clauses that prevent the blame meeting

If the integration involves a vendor, these four need to be written down. Each answers a question that, left unanswered, becomes a three-hour meeting after the incident.

**Volume and window**
How many messages per hour, and at what times. An integration sized for the average day breaks at peak — and peak is exactly when it matters.

**Reprocessing**
Who resends, within what time, and without creating duplicates. It must be clear whether the resend is automatic or requested, and who authorises it.

**Message retention**
How long what travelled stays available for inspection. Without it, investigating a case from three months ago is impossible.

**Boundary of responsibility**
Where one side's obligation ends and the other's begins. Usually at the receipt confirmation — but it must be written which confirmation.

**Need help answering these questions?**
Inove designs and supports integration between SAP, cloud and legacy systems — starting from the agreement, not the tool. **inovesolutions.com**