language io | ENTERPRISE STRATEGY GUIDE

Build, Buy, or Platform?

A Practical Guide to Enterprise Translation Infrastructure

WHAT'S INSIDE

Build, Buy, or Platform?

TABLE OF CONTENTS

language io

CHAPTER 01

Why This Decision Is Harder Than it Looks

The translation tools you already have aren’t the translation infrastructure you need.

01

BUILD, BUY, OR PLATFORM?

THE PROBLEM

Every enterprise has some form of translation today.

Google Translate is bookmarked in the support org. OpenAI is being piloted by three different teams. Salesforce says it handles multilingual. **So why are your international customers still getting worse service than your English-speaking ones?**

The problem isn't access to translation. It's the assumption that translation is a feature you can bolt onto existing infrastructure, rather than a capability that requires the same rigor as any other enterprise system.

This guide gives you a structured way to evaluate the three paths available to any organization: build, buy from your existing platform, or buy from a specialized vendor.

The criteria that matter, the costs that hide, and the questions you need to answer before you choose.

Right now, there are board-level mandates to "do something with AI." CEO compensation packages tied to AI adoption metrics. Real budget moving toward tools that look like progress. None of that is wrong. But at the end of every experiment is a customer relationship. And that customer relationship doesn't care about your innovation roadmap.

A bad translation in a healthcare claim dispute costs you that customer. A misread financial instruction costs you more than that. A support queue where non-English speakers consistently wait longer and get worse answers is a churn problem, a brand problem, and eventually a growth problem.

The decision you're making about translation infrastructure isn't a technology decision. It's a CX decision wearing a tech procurement hat.

The problem isn’t access to translation. It’s the assumption that translation is a feature you can bolt on - rather than a capability that requires the same rigor as any other enterprise system.

The question isn’t whether your system can translate. Every system can translate.

The question is whether it translates accurately, consistently, and securely - inside the workflow your agents actually use, at the volume you actually need.

01

BUILD, BUY, OR PLATFORM?

CHAPTER 02

The Three Paths

When companies decide to get serious about multilingual support, they typically land on one of three options.

Two of them feel faster than they are. One of them feels slower than it is.

PATH ONE - NATIVE PLATFORM

The appeal is obvious. The reality isn’t.

You're already in Salesforce. Translation is listed as a capability. Turning it on requires no new vendor relationship, no security review, no integration work. You're live in days.

What you're actually turning on is AWS Translate, repackaged inside your CRM. It's a solid general-purpose engine. It has no knowledge of your products, your industry, your regulated vocabulary, or the specific language pairs where your customers actually need you.

For English-to-Spanish support tickets about a retail return, native platform translation is fine. For any regulated industry, any language with significant structural complexity, or any customer relationship where accuracy actually matters, it’s not.

Here's what that means in practice.

Japanese has three writing systems and a layered honorific structure where the wrong register doesn't just sound awkward — it signals disrespect.

Korean interpersonal dynamics are encoded directly in verb endings; a customer service interaction that uses the wrong speech level reads as rude before the content even registers.

Arabic has significant dialectal variation across regions that share the same written standard. A general-purpose NMT engine handles none of this with any consistency, and your Salesforce account team is not equipped to tell you when it's failing.

Use the native platform to prove the concept internally before you can get budget for the real solution. Don’t confuse it with one.

PATH ONE - BUILD YOUR OWN

You are now a translation software company.

The argument for building usually starts with cost. Raw API access to OpenAI or Google Translate is cheap per word. Your engineering team can wire it into your existing stack. You own the IP. You control the model.

What the argument leaves out: you are now a translation software company. And unlike a translation software company, you probably don't have computational linguists on staff.

Prompt engineering can get you surprisingly far with major European languages. It will not get you to production quality in Japanese, Korean, Arabic, or any language where meaning is structurally encoded in ways that require real linguistic expertise to configure, test, and maintain.

Beyond the linguistic complexity: every prompt needs maintaining. Every model update is your team's problem.

Quality degrades without measurement infrastructure that is itself a separate build. Zero data retention, a non-negotiable in most regulated industries — requires significant architecture work nobody budgeted for.

Build makes sense in exactly one situation: your use case is genuinely novel, no existing vendor solves it, and you have a dedicated ML team with the capacity, the mandate, and the linguistic expertise to own this permanently.

That is a narrow door as it is easy to do, but hard to do well.

Build makes sense in exactly one situation: your use case is genuinely novel, no existing vendor solves it, and you have a dedicated ML team with the capacity, the mandate, and the linguistic expertise to own this permanently. That is a narrow door.

PATH THREE - BEST-OF-BREED

One vendor. Clear ownership. Architectural fit with your stack.

The reason multi-engine orchestration exists is the same reason you can't build this yourself: no single model is best at every language pair.

The model that handles French customer service with high accuracy is not the model that handles Japanese honorifics correctly. The engine that performs well on Spanish legal terminology is not the engine that navigates Korean speech register.

This is also where glossary enforcement stops being a nice-to-have. General-purpose models and prompt-based solutions handle your product terminology inconsistently — and in complex languages, a terminology error compounds.

A mistranslated product name in French is an embarrassment. Best-of-breed solutions enforce glossary at the infrastructure level, not the prompt level, because prompt-level enforcement drifts.

The other thing a specialized vendor brings that no internal build can replicate: accumulated expertise in the actual linguistics. When your Korean escalation rate climbs, they know where to look.

When a new Arabic dialect variation starts showing up in your queue, they've seen it before. That institutional knowledge is not something you can acquire in an engineering sprint.

A specialized translation vendor routes each interaction to the model best equipped for that specific language pair, in that specific context. Your engineers can’t build that, and your Salesforce account team can’t configure it.

CHAPTER 03

The Decision Matrix

Most build vs. buy evaluations start with cost. Cost is the wrong first question. **The right first question is: what does it cost you when it fails?**

A mistranslated response in a financial dispute. A healthcare instruction that reads as ambiguous. A support interaction where a Japanese customer receives a reply that signals disrespect before they've read the first sentence.

Those aren't translation errors. They're liability events, churn events, and brand events. Evaluate accordingly.

DECISION MATRIX - CRITERIA 1-5

EVALUATION CRITERION	NATIVE PLATFORM	BUILD YOUR OWN	BEST-OF-BREED
Translation Quality One model cannot be best at every language pair. The quality gap between a specialized routing engine and a general-purpose model can make a huge difference. Some models are trained on largely European language sets and have very limited understanding of the complexities of other languages where nuance matters.	LIMITED	PARTIAL	STRONG
Zero Data Retention In financial services, healthcare, and insurance, customer data cannot pass through a vendor's training pipeline. This is a legal requirement, not a preference. Engineering it yourself is a significant architecture project. Assuming your existing platform handles it without verification is a compliance risk.	LIMITED	PARTIAL	STRONG
Glossary & Terminology Enforcement LLMs hallucinate product names. Prompt-based glossary enforcement works until it doesn't, and it degrades over time, especially in long-context interactions and complex languages where terminology errors compound. Infrastructure-level enforcement doesn't drift.	LIMITED	PARTIAL	STRONG
Linguistic Expertise & Ongoing Support This is the criterion missing from most evaluation frameworks, which is exactly why it matters. Your Salesforce account team does not have the capability to help you when your CSAT scores are noticeably lower in certain markets. Your engineering team does not have that capability. A specialized vendor does, and that institutional knowledge compounds over time.	LIMITED	PARTIAL	STRONG
Quality Measurement Translation quality degrades invisibly without active measurement. Without a feedback loop, you won't know your Arabic customers are getting worse answers until CSAT data tells you six months later. By then you've already lost customers you didn't know you were losing.	LIMITED	PARTIAL	STRONG

DECISION MATRIX - CRITERIA 6-10

EVALUATION CRITERION	NATIVE PLATFORM	BUILD YOUR OWN	BEST-OF-BREED
Agent Workflow Fit The best translation engine in the world delivers zero ROI if agents work around it. Context-switching, even thirty seconds to copy-paste into another tool, is enough friction to kill adoption. Integration inside the console your agents already use is not a convenience feature. It is the adoption strategy.	LIMITED	PARTIAL	STRONG
Total Cost of Ownership The best translation engine in the world delivers zero ROI if agents work around it. Context-switching, even thirty seconds to copy-paste into another tool, is enough friction to kill adoption. Integration inside the console your agents already use is not a convenience feature. It is the adoption strategy.	LIMITED	PARTIAL	STRONG
Time to Production Speed to deployment and speed to reliable production quality are different numbers. The native platform gets you the first one fastest. It may never get you the second. Evaluate on the metric that actually affects your customers.	LIMITED	PARTIAL	STRONG
Scalability Entering a new market means a new language pair, a new compliance regime, potentially a new dialect variation, and a new volume profile, often simultaneously. Solutions designed for this scale without architectural rework. Solutions that weren't require an engineering sprint every time your business grows.	LIMITED	PARTIAL	STRONG
Toxicity Controls Entering a new market means a new language pair, a new compliance regime, potentially a new dialect variation, and a new volume profile, often simultaneously. Solutions designed for this scale without architectural rework. Solutions that weren't require an engineering sprint every time your business grows.	LIMITED	PARTIAL	STRONG

CHAPTER 04

The Signals That Point to Each Path

The matrix tells you how each path performs against objective criteria. This section is different. It's about organization fit - the conditions on the ground that make one path genuinely right for where you are now.

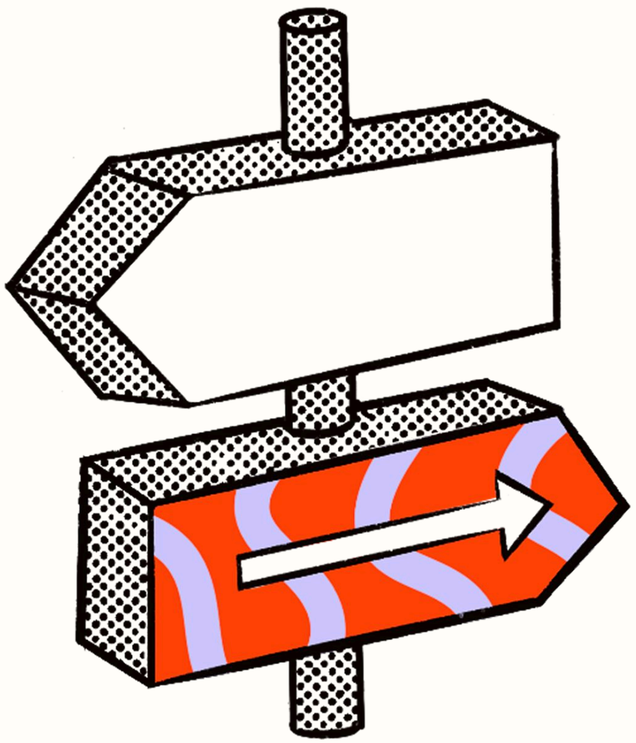
No path is universally wrong. Each one is wrong for specific organization at specific moments.

Here's how to read the signals.

SIGNAL 01

CHOOSE THE NATIVE PLATFORM WHEN:

Your conversations are low stakes, your volume is low, and accuracy is not the point.



That's a real scenario. Internal HR communications in a second language. A small e-commerce operation handling simple order status questions. A team running a limited proof of concept before they can get budget approved for something real. In those situations, the native platform is fine.

Turn it on, see how translation performs in your environment, and collect some data.

What it is not built for: any customer relationship where a wrong answer has consequences. Any regulated industry. Any language pair with structural complexity. Any volume where quality degradation compounds at scale. Any organization where your international customers are supposed to receive the same quality of service as your English-speaking ones.

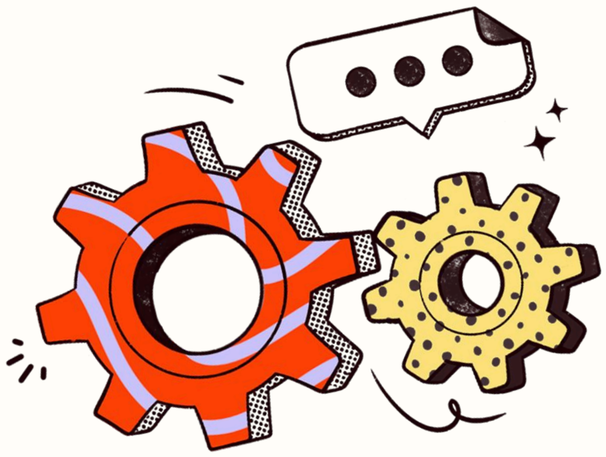
The test is simple. **Ask yourself what happens when the translation is wrong.** If the answer is "the customer gets mildly confused and sends a follow-up," the native platform may be adequate. If the answer involves a compliance event, a lost customer, an escalation, or a support interaction that reads as disrespectful before the agent has said anything substantive, you have already outgrown it. The question is whether you know that yet.

The native platform in Salesforce is AWS Translate with a CRM wrapper. Your Salesforce account team cannot help you when it fails in ways that require linguistic expertise to diagnose. That support structure does not exist.

SIGNAL 02

CHOOSE BUILD YOUR OWN WHEN:

Your use case has no existing solution and you have the team to own it permanently.



You are not buying translation for a support org. You are building a product where translation is a core differentiable capability and owning the IP matters to your investors and your roadmap. You have a dedicated ML team, not a product engineering team with some AI familiarity. You have computational linguists, or reliable access to them. And you have done the 24-month TCO model honestly, including the cost of every engineering sprint that will not go toward something else.

If all of that is true, build. It is a narrow door and most organizations that think they are walking through it are not. Here is what the build path can put the onus of being a language expert on a team not equipped to handle questions about why a certain language is impacting your metrics across the board and even less prepared to identify the root cause and fix it.

Beyond the linguistics: every prompt needs maintenance. Every model update lands on your engineering backlog. Quality degrades without measurement infrastructure that is itself a separate build. Zero data retention, non-negotiable in most regulated industries, requires significant architecture work that nobody budgeted for at the start.

The companies most confident they should build are often the least equipped to evaluate what they are taking on. Enthusiasm for AI capability is not a plan for maintaining translation quality in twelve languages over three years. The maintenance burden does not shrink as you scale. It grows

Build if the door genuinely fits. But walk through it with your eyes open.

SIGNAL 03

CHOOSE BEST-OF-BREED WHEN:

You have outgrown the other two options, or you were never suited for them to begin with.

That covers more organizations than most procurement teams want to admit. If you operate in a regulated industry, if you support more than five languages, if you are actively expanding into new markets, if your agents live in Salesforce and adoption is a real operational concern — you were never a candidate for the native platform or a DIY build. You just may not have known it yet.

Multi-engine orchestration means the model handling your French interactions is not the same model handling your Japanese ones. That is not a marketing claim. It is the only architecturally honest response to the reality that no single model is best at every language pair. A specialized vendor has done the work of evaluating which engine performs best for which language, in which context, at what volume. You are not going to replicate that in an eighteen-month build.

Glossary and terminology enforcement at the infrastructure level means your product names, your regulated vocabulary, and your brand voice travel with every interaction. Not as a prompt instruction that drifts. Not as a configuration your team has to re-verify after every model update. As a foundational layer that holds regardless of what else changes.

Agent workflow fit means your agents do not change how they work. They do not open a second tab. They do not copy and paste. Translation happens inside the console they already use, on the interactions they are already handling. Adoption is not a change management project. It is a configuration.

Translation infrastructure should be evaluated as infrastructure. One vendor, clear ownership, clean integration. That discipline is the difference between a capability and a problem you have deferred.

CHAPTER 05

Total Cost of Ownership

The math that changes every time someone says “We can just use OpenAI.”

Initial cost comparisons are almost always wrong. Not because the numbers are fabricated, but because they measure the wrong thing. Licensing fees, seat costs, and API rates are the most visible inputs in a much larger equation. **They are also the ones that look best for the options that end up costing the most.**

Here is what the honest 24-month model looks like.

01

Engineering & Integration

The initial build is the number everyone uses in the ROI calculation. It is also the smallest number in the actual cost stack. What follows it: prompt engineering cycles every time a model updates. Quality monitoring infrastructure that somebody has to scope, build, and maintain. Security and compliance architecture for zero data retention, if your industry requires it. Integration maintenance every time Salesforce releases a significant update. **None of these are one-time costs. All of them compete with your product roadmap for the same engineering capacity.**

02

Linguistic Maintenance

This is the cost nobody puts in the spreadsheet. Prompts drift. Model behavior changes between versions in ways that are not always documented. A configuration that produced acceptable Japanese output in January may not be producing the same output in July. Detecting that requires measurement infrastructure. Diagnosing it requires linguistic expertise. Fixing it requires both. **If your team does not have computational linguists on staff, every one of those cycles runs slower and costs more than it should.**

03

Quality Assurance

Without automated quality measurement, translation quality is invisible until it becomes a customer problem. Building measurement infrastructure is a project in itself. Running it requires ongoing resources. The alternative is finding out your Arabic customers have been receiving degraded service for six months when your CSAT data finally surfaces it. **That is not a QA cost. That is a revenue and retention cost that dwarfs whatever you saved on licensing.**

04

Agent Adoption

A translation tool that agents work around costs you the full price of the tool and delivers none of the value. Adoption is not free. If the integration requires context-switching, retraining, or any meaningful change to how agents work, you will spend real money on change management and still not get full utilization. **The TCO model needs a line for this. Most don't have one.**

05

Opportunity Cost

This is the number that changes the conversation most reliably when you put it in front of a CFO. Every engineering sprint that goes to maintaining a DIY translation layer is a sprint that does not go to core product. Every month your team spends debugging Korean prompt drift is a month they are not building the features your sales team is promising customers. **That cost is real, it compounds, and it is almost never in the initial build vs. buy analysis.**

06

What the Native Platform Hides

The native platform looks like the cheapest option because most of its costs are invisible. You are not paying for bad translations directly. You are paying for them in churn from markets you are not watching closely enough, in escalations that get attributed to agent performance rather than translation quality, in customer relationships that degrade slowly enough that nobody connects the cause to the effect. **Those costs are real. They just do not show up on the invoice.**

CHAPTER 05

Best-of-breed has a real price. So does everything else.

Best-of-breed has a real price and it should be evaluated honestly. What it buys you is predictability. A known cost, a known capability, a vendor whose entire business depends on the quality of what you're paying for. Set that against the full 24-month model for building and **the math inverts for most organizations somewhere between month twelve and month eighteen.**

> Require the full model before you decide. Every option on the table, including the option of building, deserves the same honest accounting.

CHAPTER 06

An Evaluation Framework

The questions that determine which options are actually on the table.

How To Use This Framework

Most evaluation frameworks ask you to weigh a long list of criteria and arrive at a conclusion through careful deliberation. This one works differently.

A small number of conditions are decisive. One regulated industry requirement disqualifies two paths before the conversation goes any further. One honest answer about engineering capacity does the same.

The chart on the following page maps your actual situation to the right path. If you find a "No" in a column that describes your organization, that path is off the table. Start there.

One "No" in a column that describes you means that path is eliminated.

NATIVE PLATFORM

The translation layer already inside your CRM or CCaaS. Fast to deploy, limited in depth.

BUILD YOUR OWN

A custom solution built by your engineering team using a general-purpose language model.

BEST-OF-BREED

A purpose-built enterprise translation platform with operational infrastructure baked in.

EVALUATION FRAMEWORK

YOUR SITUATION	NATIVE PLATFORM	BUILD YOUR OWN	BEST-OF-BREED
BUDGET & SCOPE			
Low volume, low stakes, accuracy not critical	✓ Yes	✗ No	✗ No
Proof of concept, budget not yet approved	✓ Yes	✗ No	✗ No
COMPLIANCE & RISK			
Regulated industry - financial services, healthcare, insurance	✗ No	✗ No	✓ Yes
Translation errors carry legal or financial consequences	✗ No	✗ No	✓ Yes
LANGUAGE COMPLEXITY & SCALE			
Complex language pairs - Japanese, Korean, Arabic	✗ No	✗ No	✓ Yes
More than five languages or active market expansion	✗ No	✗ No	✓ Yes
When translation fails in a complex language, you need expert diagnosis	✗ No	✗ No	✓ Yes
LANGUAGE COMPLEXITY & SCALE			
Dedicated ML team with computational linguistics expertise	✗ No	Maybe	✓ Yes
Need production quality in under six months	✗ No	✗ No	✓ Yes

EVALUATION FRAMEWORK

YOUR SITUATION	NATIVE PLATFORM	BUILD YOUR OWN	BEST-OF-BREED
OPERATIONS & MEASUREMENT			
Agents are Salesforce-native and adoption is a real concern	✕ No	✕ No	✓ Yes
Handle time, CSAT, and quality are actively measured	✕ No	✕ No	✓ Yes
Need quality measurement and feedback loops over time	✕ No	✕ No	✓ Yes
COMPLIANCE & RISK			
IP ownership is a core business requirement	✕ No	✓ Yes	✕ No
Building a product where translation is a core differentiator	✕ No	✓ Yes	✕ No
Already tried the native platform or a DIY build and hit the ceiling	✕ No	✕ No	✓ Yes

Most organizations land in the best-of-breed column.

The framework eliminates paths based on your actual constraints. For enterprise support teams operating at scale, the decisive factors almost always point the same direction.

The question is no longer whether to use a purpose-built platform.

It is which one fits your operational environment.

CHAPTER 07

Running the POC.

Four weeks is enough to eliminate ambiguity.

Here is the structure that works.

A POC is not a pilot.

A pilot is open-ended. It expands to fill available time, accumulates stakeholders with conflicting opinions, and rarely produces a clean decision. Organizations run pilots when they want to feel like they are making progress without committing to a conclusion.

A POC has a defined end date, defined success criteria, and a decision gate. Build that gate into the process before you start, or you will be running a pilot in four weeks and calling it something else.

Four weeks is the right window. Long enough to surface real failure modes. Short enough to maintain organizational momentum and prevent the evaluation from becoming its own project.

Define the decision gate before you start. Not after.

WEEK 01

Establish the Baseline
Pull real cases. Measure current handle time and CSAT. Document what's already broken.

WEEK 02

Deploy and Configure
Stand up in sandbox. Load your actual glossary. Run first live cases from your real queue.

WEEK 03

Measure and Stress Test
Track numbers against baseline. Test edge cases. Find the volume ceiling before you commit.

WEEK 04

Decide With Data
Compile before and after. Build the TCO model. Make the call on the date you committed to.

01 Establish the Baseline

Do not start translating anything yet. Start measuring.

Pull 200 to 300 real cases from your queue, weighted toward the language pairs that matter most to your evaluation.

Measure current handle time on multilingual interactions. Pull CSAT scores segmented by language where you have them.

Document the translation failure modes your agents already know about, the ones they have workarounds for, the ones that generate escalations.

Talk to three or four agents directly. They know exactly where the current system breaks. They have just never been asked to put it in a formal report.

This baseline is the most important output of week one. Without it, your week four conclusions are opinions. With it, they are evidence.

02 Deploy and Configure

Stand up the solution in a sandbox environment.

Load your actual product glossary and terminology list, not a cleaned-up version of it.

Configure compliance and data retention settings before any live data touches the system.

Brief your agents, and keep that briefing short. If the solution requires more than thirty minutes of explanation to use, that is itself a finding worth documenting.

Run your first live cases. Not a curated sample. Real interactions from your actual queue, in the language pairs you identified in week one.

03 Measure and Stress Test

Track handle time against your baseline. Not directionally. With actual numbers.

Test the edge cases your agents told you about in week one. Long-form interactions. Highly technical language. Emotionally charged customer communications. Profanity and abusive content in languages your agents do not speak.

These are not edge cases in production. They are regular occurrences that your evaluation needs to account for.

Run volume tests. Translation latency that is acceptable at fifty interactions per hour may not be acceptable at five hundred. Find the ceiling before you commit, not after.

Collect agent feedback directly and in writing. Not a survey. A conversation.

Ask specifically what is working, what is not, and what would have to change for them to trust this tool on their most complex interactions.

04 Decide with Data

Compile the before and after metrics.

Handle time, CSAT where available, escalation rates on multilingual interactions, agent feedback, quality sample results.

Build the 24-month TCO model using real volume numbers from your own queue, not vendor-supplied estimates.

Present the findings to stakeholders before the decision date, not on it.

Give people time to ask questions and raise objections while there is still time to address them. Then make the call on the date you committed to at the start.

If the POC surfaces failure modes that disqualify the solution, that is a successful POC. You have learned something real before you committed budget. That is the point.

CHAPTER 08

The Pre-Decision Checklist

Close the gaps with this checklist designed to help you close the last gate before you commit budget.

Work through every section relevant to your evaluation.

Check each box you can honestly confirm is complete.

Count unchecked boxes by section and find out what your answers tell you in the guidance section.

PRE-DECISION CHECKLIST

COMPLIANCE & SECURITY

- [ ] Zero data retention requirements documented and shared with every vendor under evaluation
- [ ] Data residency requirements reviewed with legal, not with the vendor's sales team
- [ ] Security certifications reviewed for each vendor under evaluation
- [ ] Applicable regulatory requirements (GDPR, HIPAA, FINRA) mapped explicitly to vendor capabilities, with gaps documented

QUALITY & ACCURACY

- [ ] All required language pairs tested, not just the ones the vendor chose to demo
- [ ] Complex language pairs including Japanese, Korean, and Arabic tested against real interactions from your queue
- [ ] Actual product glossary loaded and tested for accuracy and consistency
- [ ] Quality measurement methodology agreed and documented before deployment
- [ ] Edge cases tested: technical terminology, emotionally charged language, long-form interactions, abusive content

LINGUISTICS EXPERTISE & SUPPORT

- [ ] Each vendor asked directly: when something goes wrong in a language our team does not speak, what is your diagnostic process?
- [ ] Responsibility for identifying and fixing quality degradation in complex language pairs documented and assigned
- [ ] Vendor support model confirmed to include linguistic expertise, not just technical support

WORKFLOW & ADOPTION

- [ ] Integration reviewed inside the Salesforce Service Cloud agent console, not a standalone demo environment
- [ ] Solution requires no context-switching outside the console agents already use
- [ ] Handle time baseline established with specific post-launch targets
- [ ] Agent feedback collected directly, not inferred from utilization metrics
- [ ] Adoption accountability assigned to a named person with a 90-day target

ECONOMICS

- [ ] 24-month TCO model built for every option under evaluation, including the option of building
- [ ] Engineering cost of the build option calculated explicitly: maintenance, prompt engineering, quality monitoring, and security review cycles
- [ ] Volume pricing and scaling terms reviewed against actual expected capacity, not current volume
- [ ] ROI baseline and targets agreed with finance stakeholders before vendor selection

POC

- [ ] POC used real cases from your actual queue, not a vendor-supplied dataset
- [ ] Success criteria defined and agreed before the POC started
- [ ] Real agents participated and their feedback was collected in writing
- [ ] Decision date set before the POC began and held to
- [ ] 24-month TCO model uses volume numbers from your own POC data, not vendor estimates

PRE-DECISION CHECKLIST

If you have unchecked boxes in COMPLIANCE & SECURITY

Stop. Do not proceed to vendor selection.

A compliance gap discovered during contract negotiation costs significantly more to resolve than one discovered during evaluation.

If your regulatory requirements disqualify options you were considering, better to know now.

If you have unchecked boxes in QUALITY & ACCURACY

You are evaluating on incomplete information.

Vendor demos are optimized to avoid the failure modes your unchecked boxes would surface. Run the missing tests before you decide.

One edge case failure in a complex language pair during a POC is information. The same failure six months into a contract is a problem.

If you have unchecked boxes in LINGUISTICS EXPERTISE & SUPPORT

Most organizations arrive here with the most gaps.

Those unchecked boxes represent a diagnostic capability you are assuming exists somewhere in your organization or your vendor relationship.

Ask the question directly before you commit. The answer will narrow your options faster than anything else on this list.

If you have unchecked boxes in WORKFLOW & ADOPTION

Your ROI projection is theoretical.

Translation infrastructure that agents work around delivers none of its projected value.

Close the workflow gaps before you finalize the business case, or build the adoption risk explicitly into your financial model.

If you have unchecked boxes in ECONOMICS

You are comparing options on incomplete cost data.

The option with the most unchecked boxes in this section is almost always the build path.

That is not a coincidence.

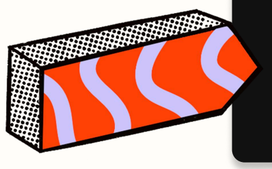
Require the same honest accounting from every option before you decide.

If you have unchecked boxes in POC

You are not ready to decide.

A four-week POC with real data and a defined decision gate is the only way to remove ambiguity before you commit budget.

The cost of running it is measured in weeks. The cost of skipping it is measured in contract terms.



If every box is checked, you have done the work. You have the information you need to make a defensible decision, close the gaps your evaluation surfaced, and walk into procurement with evidence rather than assumptions. The work is done. The decision is yours.

language io

The platform built for the problem you're **actually** trying to solve.

Language IO is the operational layer between your language models and your customer support workflows.

It manages the terminology, prompts, quality signals, and channel integration that raw AI cannot provide on its own, and it adapts as models evolve, so your investment does not start over when a better option emerges.

Request a Demo

languageio.com