

# Interloquial Communication & The ICE Framework

A comprehensive guide to effective human-AI collaboration.

---

## Quick Start

**If you only read one thing:** AI isn't human, so talking to it like a human doesn't work. ICE is not a template you fill in — it's a structure for dialogue that surfaces what you actually need.

### The reversal that changes everything:

Instead of instructing — “Build me a settings panel”

Try inviting dialogue — “I need users to access their settings quickly. Before you start, what questions do you have about intent, constraints, and expectations?”

The AI asks. You answer. You both discover requirements you hadn't articulated. *Then* you build.

### ICE gives that dialogue structure:

- **Intent** — What are you trying to achieve, and why does it matter?
- **Constraints** — What boundaries must the solution respect?
- **Expectations** — How will you know it's right?

These aren't boxes to fill in. They're territories to explore through conversation.

---

## Part 1: Interloquial Communication

### 1.1 What Is Interloquial?

**Interloquial** (*adj.*) — Relating to communication between different kinds of intelligences (human-AI). From Latin *inter* (between) and *loqui* (to speak).

**Interloquial skills** are the ability to exchange intent, context, and understanding clearly across the boundary between human and machine intelligence.

This is distinct from: - **Prompting** — One-way instruction (you write, AI responds) - **Interpersonal communication** — Human-to-human exchange where emotion and rapport matter

Interloquial communication is **two-way dialogue** where meaning emerges through exchange.

### 1.2 Why It Matters

Most poor AI results stem from unclear requests, not AI limitations.

### Vague inputs produce vague outputs.

The AI cannot read your mind. It cannot infer unstated requirements. It cannot know what “good” looks like unless you tell it. The bottleneck is not the AI's capability — it's your ability to articulate what you want.

### 1.3 Key Principles

**Principle 1: Dialogue, Not Instruction** Effective interloquial communication is not one-way. You express requirements. The AI interprets them. The AI asks clarifying questions. You refine your thinking. Meaning emerges through this exchange.

**Less effective:** “Build me a user authentication system.”

**More effective:** “I need users to be able to log in. Before you start, what questions do you have about my requirements?”

The reversal — asking the AI to question *you* — is transformative.

### Principle 2: Precision Over Warmth

Interpersonal	Interloquial
Build rapport	Build clarity
Read emotional subtext	State requirements explicitly
Prioritise relationship	Prioritise accuracy
Success = trust built	Success = output works

AI partners do not reward warmth. They reward precision.

**Principle 3: Trust Through Verification** AI responses arrive with identical confidence whether correct or fabricated. The *feel* of a response tells you nothing about its reliability.

Trust Zone	Description	Use When
<b>Blind trust</b>	Follow without verification	Never recommended
<b>Verified trust</b>	Cross-check critical operations	Production code, security, payments
<b>Exploratory trust</b>	Try it and observe results	Learning, prototyping, reversible changes

**Rule:** Could this action cause harm? If in doubt, verify through independent sources before executing.

**Principle 4: Expression AND Interpretation** Most attention goes to expression (“How do I write better prompts?”) but interpretation is equally important:

- Can you recognise when AI output has subtle flaws?
- Can you identify when the AI made a structural choice you wouldn’t have made?
- Can you distinguish “this is wrong” from “this is different but possibly better”?

Interloquial competence requires both directions.

## Part 2: The ICE Framework

### 2.1 Overview

**ICE** structures the planning conversation that precedes implementation. It is not a prompt template — it is a framework for dialogue.

Element	Definition	Surfaces
Intent	What you want to achieve and why	Purpose, context, user situation
Constraints	Boundaries and requirements	Technical limits, business rules, design principles
Expectations	How you will verify success	Acceptance criteria, what “right” looks like

### 2.2 Intent: What You Want to Achieve

Intent is the outcome you’re trying to accomplish, including the context that makes it meaningful.

**Without intent:** “Add a button to the page.”

**With intent:** “Users need to reach their settings quickly without losing their place — they’re often mid-task and can’t afford to navigate away. Something always visible that says ‘your preferences are here.’”

Intent tells *why*, not just *what*. This gives the AI information to make good implementation decisions.

**Questions to surface intent:** - What problem does this solve? - Who will use this, and in what situation? - What happens if we don’t build this? - What does success look like for the user?

### 2.3 Constraints: The Boundaries

Constraints narrow the solution space. They have dual nature:

Function	Description
<b>Restrictive</b>	Prevents unwanted solutions
<b>Generative</b>	Forces discovery of opportunities within the fence

You don’t need technical vocabulary. Express what matters: - “My users often work with unreliable internet — can this work offline?” - “Only paying customers should see this feature.” - “It needs to feel instant — I don’t want users waiting around.”

**Questions to surface constraints:** - What technical limitations exist? - What business rules must be respected? - What can we NOT do? - What resources (time, budget, expertise) are limited? - What security or privacy requirements apply?

## 2.4 Expectations: How You'll Know It's Right

Expectations are how you'll verify success — what you'll look for when you test it.

**The discipline:** Force yourself to describe what success looks like *before* building.

“I'll know it's working when I can click the icon, see my settings, change something, and find that change still there tomorrow.”

**Technique:** Ask the AI to wireframe the interface before building:

“Before we start coding, sketch out what the screens will look like — what the user sees, where they click, what happens next.”

Wireframing surfaces expectations you didn't know you had.

**Questions to surface expectations:** - What will the user see/experience when this works correctly? - What should happen when things go wrong? - How will we test this? - What edge cases might exist?

## 2.5 When ICE Adds Value

ICE is **task-dependent**.

Task Type	ICE Value	Why
Simple, well-defined	Low	Little ambiguity to resolve
Complex, ambiguous	High	Collaborative refinement matters
Iterative, evolving	High	Prevents drift and misalignment
Single-shot	Low	No compounding benefit

You don't need ICE to ask someone to pass the salt. You do need it to design a house.

## 2.6 Running ICE Twice

A discovery from practice: running ICE twice produces qualitatively different results.

**First pass:** Surfaces *what* — core problem, functional requirements **Second pass:** Surfaces *why* — including user state of mind at point of use

Example: Planning a disk cleanup utility - First ICE: “WizTree shows data but no guidance; cleanup tools feel scary” - Second ICE: “User state: panic + anxiety (disk full, scared of breaking things)”

That insight transformed the design to be safety-first, with conservative language and UI designed to slow down panicked users.

---

## Part 3: The Critical Persona Step

### 3.1 Why This Matters

AI systems are trained to be helpful and agreeable. This creates **sycophancy bias** — the AI wants to build what you asked for, even if what you asked for has flaws.

The solution: ask the AI to adopt a critical persona that actively looks for problems.

### 3.2 How to Use It

After completing ICE, before implementation:

“Now review this plan as a [critical persona]. What have we missed? What could go wrong? What assumptions are we making that might be wrong?”

#### Useful critical personas:

Persona	Good For
Sceptical user	UX problems, confusing flows
Security auditor	Vulnerabilities, data exposure
Hostile critic	Weak arguments, unstated assumptions
Edge case tester	Failure modes, unusual inputs
Cost accountant	Hidden expenses, scope creep
Maintenance developer	Code that will be hard to change later

### 3.3 Example

After planning a feature:

“Before we build this, review the plan as a sceptical user who’s tried similar features before and been disappointed. What would make them roll their eyes? What’s likely to frustrate them?”

The AI shifts from “how do I build this?” to “what’s wrong with this?” — surfacing considerations that agreeable mode would miss.

---

## Part 4: The Blueprint Output

### 4.1 Capture the Conversation

Planning conversations are volatile. The understanding you developed exists only in that session’s context. Tomorrow, the AI remembers nothing.

**Before leaving planning mode:** Ask the AI to record decisions into a reference document.

“Capture what we’ve agreed in a project brief I can use going forward.”

This transforms ephemeral dialogue into durable artefact.

### 4.2 Use the Blueprint

The reference document becomes the anchor: - Start implementation: “Here is what we agreed. Let’s start building.” - Requirements change: Update the document first, then proceed - Return after time away: The document restores context instantly

## 4.3 Document Structure

A useful blueprint contains:

# [Project/Feature Name] Blueprint

## Intent

What we're building and why. The problem being solved. User context.

## Constraints

- Technical limitations
- Business rules
- Security requirements
- Performance targets

## Expectations

- How we'll know it works
- Key user scenarios
- Edge cases to handle

## Critical Review Notes

Issues surfaced during critical persona review. Mitigations planned.

## Decisions Made

Key choices and their rationale. What we considered and rejected.

---

## Part 5: Practical Application

### 5.1 Starting a New Task

1. **State your intent** — What are you trying to achieve? Why does it matter?
2. **Invite questions** — “Before you start, what questions do you have?”
3. **Articulate constraints** — What boundaries apply?
4. **Define expectations** — How will you verify success?
5. **Run critical review** — “Review this as a [sceptical persona]”
6. **Capture the blueprint** — Document decisions before implementing

### 5.2 Mid-Project Course Correction

When requirements change or something isn't working:

1. **Identify which ICE element shifted** — Intent? Constraints? Expectations?
2. **Re-run ICE for that element** — Surface the new understanding
3. **Update the blueprint** — Record the change
4. **Continue with explicit context** — “Here's what changed and why”

### 5.3 Common Mistakes

Mistake	Problem	Solution
Skipping to implementation	AI builds the wrong thing confidently	Always start with ICE dialogue
Assuming AI knows context	AI can't read your mind	State context explicitly every time
Accepting first output	May contain subtle errors	Verify, question, iterate
Emotional language	Decreases precision	Bracket feelings, focus on requirements
No critical review	Sycophancy produces flawed plans	Always run adversarial persona
No blueprint	Context lost between sessions	Always capture decisions in writing

## Part 6: Psychological Guardrails

### 6.1 Risks to Manage

Working with AI creates psychological dynamics to be aware of:

Risk	Description	Mitigation
<b>Automation complacency</b>	The more fluent the output, the less critically it gets reviewed	Treat fluent output with <i>more</i> suspicion
<b>Skill atrophy</b>	Delegating cognitive work weakens underlying capability	Maintain core skills through deliberate practice
<b>Sycophancy bias</b>	AI is trained to agree, reinforcing poor ideas	Use critical personas to break the pattern
<b>Overconfidence</b>	AI confidence doesn't correlate with accuracy	Verify independently, especially for high-stakes decisions

### 6.2 Healthy Practices

- **Verify before trusting** — Cross-check important outputs with independent sources
- **Question confident answers** — “How would I check this independently?”
- **Maintain human judgement** — AI assists, human decides
- **Recognise mode-switching** — Interloquial precision for AI, interpersonal warmth for humans

## Quick Reference Card

### The ICE Mnemonic

Letter	Element	Key Question
<b>I</b>	Intent	What am I trying to achieve, and why?
<b>C</b>	Constraints	What boundaries must this respect?
<b>E</b>	Expectations	How will I know it's right?

## The Process

1. **Express Intent** — State what and why
2. **Invite Questions** — “What do you need to know?”
3. **Articulate Constraints** — Boundaries and limits
4. **Define Expectations** — What success looks like
5. **Critical Review** — “Review as a sceptical [persona]”
6. **Capture Blueprint** — Document before building
7. **Implement** — Build from the blueprint
8. **Verify** — Check output against expectations

## Trust Calibration

- **Blind trust** — Never
- **Verified trust** — For anything that matters
- **Exploratory trust** — For reversible experiments

## Key Principles

- Vague inputs produce vague outputs
- Dialogue, not instruction
- Precision over warmth
- Trust through verification
- Expression AND interpretation
- Run ICE twice for depth
- Critical persona breaks sycophancy
- Blueprint preserves context

---

## Appendix: The Interloquial Concept

### Etymology

*Inter* (Latin: between) + *loqui* (Latin: to speak) = speaking between intelligences.

### Why a New Word?

Existing terms don't capture what's happening: - **Prompting** — Implies one-way instruction - **Chatting** — Implies interpersonal warmth - **Instructing** — Implies command-and-control

Interloquial names a distinct communication mode with its own dynamics, skills, and success criteria.



## The Core Distinction

Interpersonal	Interloquial
Between persons	Between intelligences
Emotion is central	Emotion is a confounder
Success = relationship quality	Success = output quality
Empathy builds connection	Precision builds results

## Related Fields

Field	Studies	How Interloquial Differs
Biosemiotics	Communication across all living systems	Both parties speak the same language
Zoosemiotics	Animal communication	We can iterate through dialogue
Xenolinguistics	Hypothetical alien communication	The AI responds immediately

The unprecedented feature of interloquial communication: both parties speak the same language, enabling rapid iteration through dialogue.

---

*This framework synthesises research on prompt engineering, communication theory, and practical experience from human-AI collaboration. It is designed to be applicable across AI platforms and domains.*

**Version:** 1.0 **Last Updated:** February 2026 **Licence:** Use freely with attribution