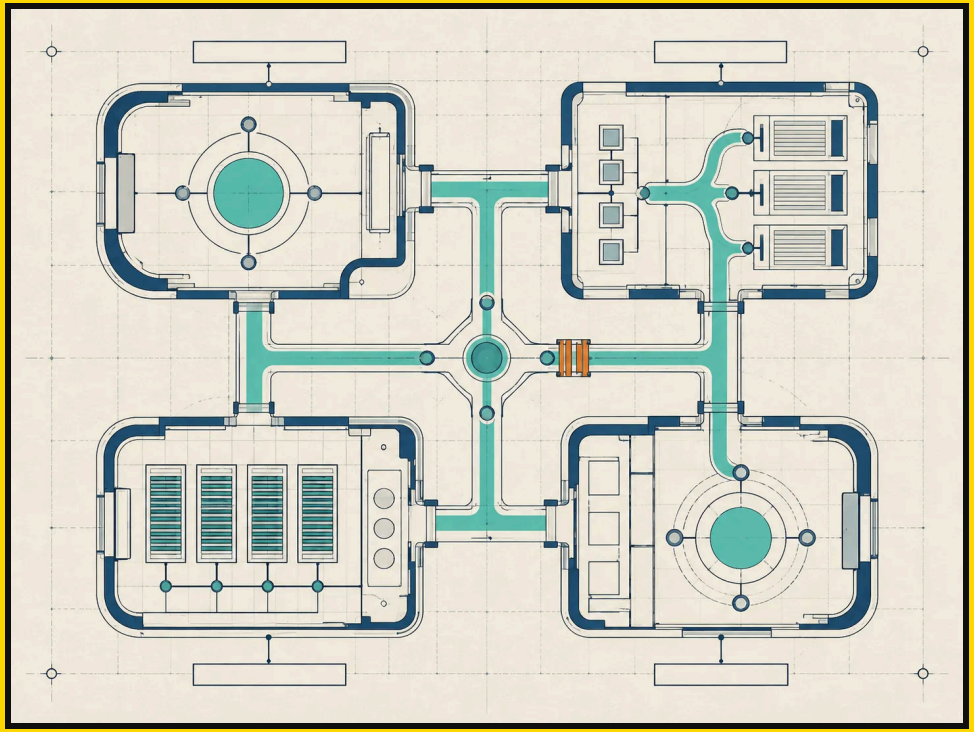
PROMPT → WORKFLOW → AGENT

HOW TO DESIGN AN AI COLLEAGUE THAT CAN ACT

The B.L.U.E.P.R.I.N.T. Intelligent Agent Framework v2.4

© 2024–2026 Ahmed Muzammil · CC BY-SA 4.0



START HERE

Before an AI can act for you, someone has to decide where it stops.

A prompt gets you an answer. An agent goes and does things in your systems, so you have to decide what it's allowed to touch.

This deck walks through those decisions one letter at a time, using two example agents: one in a bank and one in a clinic.

A newer model won't fix an agent that was never told what it's for.

What people usually assume

If the agent gets things wrong, the fix is a better prompt or a newer model.



What we see in practice

Most problems come from things nobody wrote down: what the agent is for, what it can open, and who checks its work. This deck helps you write that down. Your tech team builds the prompt, the access and the tests from it.

DO YOU NEED AN AGENT?

1 · Is this a regular job for someone, not a once-in-a-while ask?

If no, a good prompt is enough: you give it what it needs, check the answer and decide what happens next.

Use PINPOINT to write that request well.

2 · Would they need to get into your systems or tools themselves, not just tell you what to do?

If no, set up an assistant with one clear job. It answers, and a person acts on what it says.

Use BLUE: the first four letters.

3 · Would a checklist fail because some cases are different, so they'd have to decide the steps as they go?

If no, it's a checklist: write the steps down and use ordinary software or a workflow. AI can still do one step inside it.

Three yeses means you're building an agent, and BLUEPRINT is how you describe it. Stop at the first no and use what it points to.

Anthropic, *Building Effective Agents* (19 December 2024): find the simplest solution possible and only increase complexity when needed, which may mean not building an agentic system at all.

A workflow is a job where you already know the steps. An agent is a job where something has to work out the steps each time. Most work that looks like it needs an agent is a workflow with one or two decisions inside it. A workflow with an AI step that gets into your systems still needs its limits written down: use BLUEPRINT for that step.

Wherever you stop, write down why in one sentence. It's the first question a reviewer will ask.

OUR TWO EXAMPLES

Two agents we'll build together.

IN A BANK: PRIYA'S MEETING-BRIEFING AGENT

Priya is a relationship manager. Before each client meeting she pulls notes together from several systems.

Her agent will do that for her: read what's relevant and hand her a one-page brief the evening before.

IN A CLINIC: DANA'S FRONT-DESK AGENT

Dana runs the front desk at a busy family clinic. Every morning she checks who's booked in, which forms are missing and who hasn't confirmed.

Her agent will prepare that list and draft reminder texts for her to check.

Each letter adds one decision to both agents. By the end you'll have a full description of each, and a template for your own.

Where most ideas start, and where they need to get to.

HOW THE IDEA USUALLY ARRIVES

“An assistant that helps relationship managers prepare for meetings. It should look up whatever is relevant for each client, in a professional tone. Don't give investment advice.”

This covers tone and one thing to avoid. It says nothing about what a finished brief looks like, what the agent can open, or who answers for it.

→

THE JOB

A one-page brief for Priya, ready the evening before each client meeting. Priya's team lead signs off on what a good brief contains.

WHAT IT DOES ALONE, AND WHAT IT ASKS ABOUT

It reads meeting notes and client records by itself. It asks before suggesting any follow-up. It gives no investment advice.

WHAT IT CAN OPEN

Only what Priya can open, through its own login. It can read, but it can't change or send anything.

The same step for Dana at the front desk.

HOW THE IDEA USUALLY ARRIVES

“Something to help the front desk get through the morning. It should check the bookings, chase patients who haven't filled in their forms, and sound friendly. Don't give medical advice.”
Friendly tone and one thing to avoid again. It doesn't say when the list is due, who may send a text to a patient, or what it can see in the booking system.



THE JOB

Tomorrow's patient list for Dana by 8am, with missing forms flagged and reminder texts drafted. The practice manager signs off on what a good list contains.

WHAT IT DOES ALONE, AND WHAT IT ASKS ABOUT

It checks bookings and forms by itself. Every reminder text waits for Dana to approve it. It never answers questions about symptoms or treatment.

WHAT IT CAN OPEN

Only this clinic's bookings, through its own login. It can draft texts, but it can't send them, change a booking or open medical notes.

An agent is AI with a goal and permission to act. It picks a step, does it, checks what happened, and keeps going until the job is done or it has to stop.

Decide → Do → Check → Decide again ↻

done, or stopped ↗

It still uses prompts. The difference is that it keeps going by itself.

A workflow follows steps someone wrote down in advance. An agent works out its own steps as it goes (after Anthropic, *Building Effective Agents*, 19 December 2024).

Simon Willison, 18 September 2025: “An LLM agent runs tools in a loop to achieve a goal.” In other words, an AI that keeps using your systems, step by step, until it gets there.

With a prompt, you decide each step. With an agent, it decides.

	A prompt	An agent
Who decides the next step	You	The agent
How often it asks the AI	Usually once	Many times, one step after another (in a loop)
What you get back	Text	Text, plus things it has done
What it carries between steps	Whatever you paste in	Its own notes on what it has done so far
When it goes wrong	**You get a bad answer**	**Something gets done that shouldn't have**

That last row is why an agent needs its job and its limits written down.

WHERE BLUEPRINT SITS

Level	Who decides the next step	Framework
Prompt	You, each time	PIN / PINPOINT
Workflow	A path written down in advance	PINPOINT for each step
Agent	The AI, within limits you set	BLUEPRINT
Several agents	Each agent, within its own limits	One BLUEPRINT per agent

Most real systems are a mix. If only one step needs the AI to decide, describe it as a workflow with one smart step.

BLUE

Start with the least structure that works.

B

Behaviour & Values

What should it put first, and what does it do when unsure?

L

Limits

What can it do alone, what must it ask about, and what is off-limits?

U

Use Case & Outcome

What is the one job, who is it for, and what does finished look like?

E

Examples

What does a good result look like, and have you tried the awkward cases?

BLUE is enough for an assistant that only answers questions. Once it can open systems or do things, add PRINT.

B Behaviour & Values

Give it a ranked list of what matters.

WHY IT MATTERS

Rules cover the cases you thought of. Something will come up that you didn't, and the agent needs to know what matters most when it does.

PRIYA'S BRIEFING AGENT (BANK)

Accuracy comes before speed. If two records disagree, it says so. If it isn't sure, it does less and tells Priya why.

DANA'S FRONT-DESK AGENT (CLINIC)

Patient privacy comes first, then accuracy, then getting the list out on time. If a booking looks odd, it flags it for Dana.

WRITE DOWN

Your agent's top three priorities, in order. When two clash, which one wins?

L Limits & Autonomy

Sort everything it might do into three lists.

WHY IT MATTERS

An agent will try whatever seems helpful. Without clear lists, it has to guess where your line is. Reading counts too: opening a file is an action.

PRIYA'S BRIEFING AGENT (BANK)

Just do it: read meeting notes. **Ask first:** suggest a follow-up. **Off-limits:** recommend investments.

DANA'S FRONT-DESK AGENT (CLINIC)

Just do it: check the day's bookings. **Ask first:** send a reminder text. **Off-limits:** give medical advice or move an appointment.

WRITE DOWN

Three lists: just do it, ask first, off-limits. If something isn't on any list, it stops and asks.

U Use Case & Outcome

One job, one person who answers for it, and a clear picture of finished.

WHY IT MATTERS

An agent with a vague job drifts into other work. When something goes wrong, nobody is sure whose call it was.

PRIYA'S BRIEFING AGENT (BANK)

A one-page brief for each client meeting, ready the evening before. Priya's team lead answers for it.

DANA'S FRONT-DESK AGENT (CLINIC)

Tomorrow's patient list, with missing forms flagged and reminder drafts ready by 8am. The practice manager answers for it.

WRITE DOWN

The job in one sentence, the named person who answers for it (a person, so no team names), and what finished looks like.

E Examples & Evaluation

Collect examples of good and bad, then test it on the awkward ones.

WHY IT MATTERS

An agent can reach a good answer the wrong way, for example by opening something it shouldn't. It can also answer the same question differently on different days.

PRIYA'S BRIEFING AGENT (BANK)

Test a normal meeting, a client with out-of-date records, two systems that disagree, a record Priya can't open, and a request it must refuse.

DANA'S FRONT-DESK AGENT (CLINIC)

Test a normal morning, a double booking, a patient with no phone number, a record from another clinic, and someone asking it for a diagnosis.

WRITE DOWN

Five test cases, including one it should refuse. Agree the pass mark before you run them, and run each one more than once.

PRINT

Add these once the agent can open systems and do things.

P

Permissions & Identity

What can it open?

R

Reliability & Recovery

What if something breaks?

I

Interfaces & Handoffs

Which systems does it use, and when does it hand over to a person?

N

Navigation & Orchestration

What are the steps it follows, and when does it stop and/or escalate?

T

Trusted Context & Memory

What does it read and keep?

BLUE describes the job. PRINT describes how it gets done safely inside your systems.

P Permissions & Identity

Telling it isn't the same as stopping it.

WHY IT MATTERS

If we only tell the agent “don't look at other advisers' clients”, it can still try. A block in the system makes it impossible.

PRIYA'S BRIEFING AGENT (BANK)

Set it up like a new joiner's login. It has its own login, so we can see what it did. It can only open what Priya can open. For now it can read, but it can't change or send anything.

DANA'S FRONT-DESK AGENT (CLINIC)

It signs in as itself and sees only this clinic's bookings. It can draft reminder texts. They go out only after Dana approves them.

WRITE DOWN

Whose access does it borrow, and what is it blocked from?

R Reliability & Recovery

Decide what happens when something breaks.

WHY IT MATTERS

Systems go down, data goes out of date and AI models get updated. An agent that quietly guesses its way past a problem does more harm than one that stops.

PRIYA'S BRIEFING AGENT (BANK)

If the client records system is down, Priya still gets a brief, marked incomplete, saying which part is missing.

DANA'S FRONT-DESK AGENT (CLINIC)

If the booking system times out, it stops, tells Dana which patients it hasn't checked, and sends nothing.

WRITE DOWN

When it gets stuck: what it does, who hears about it, and how you undo anything it already did. Keep a full record of every run, like a flight recorder.

I Interfaces & Handoffs

Before it uses a tool, write down what the tool does and what it can break.

WHY IT MATTERS

The agent only knows a tool by its description. If “no record found” and “access blocked” look the same, it may tell Priya a client has no history when it was locked out.

PRIYA'S BRIEFING AGENT (BANK)

The calendar tool reads meetings. The client records tool reads notes and can't change them. Each one says clearly when it's empty, broken or blocked.

DANA'S FRONT-DESK AGENT (CLINIC)

The booking tool reads appointments. The text tool can send only drafts Dana has approved. If it fails, it says so.

WRITE DOWN

For each tool: what it's for, what it changes, and what it says when it fails.

N Navigation & Orchestration

Fix the steps you already know. Let the agent decide only where it has to.

WHY IT MATTERS

The more steps the agent chooses, the harder it is to predict. Most jobs are a fixed path with one or two judgement calls inside.

PRIYA'S BRIEFING AGENT (BANK)

Find tomorrow's meetings, pull the records, write the brief. The agent decides only what's relevant for each client. If sources disagree, it stops and asks Priya.

DANA'S FRONT-DESK AGENT (CLINIC)

List the bookings, check the forms, draft reminders. The agent decides only how to word each reminder. It stops when the list is done or it hits something it can't handle.

WRITE DOWN

The steps that are always the same, the one or two it decides, and when it should stop.



A messy filing cabinet makes a messy AI.

Imagine the smartest analyst in the world, with a bad filing cabinet. The AI may be brilliant. Its information may not be.

Context engineering

Context engineering is deciding what the agent gets to see before it answers: the right documents and data, the effective version, filtered for the person using it. Everything on the next page is part of it.

What goes in: duplicated, stale, over-shared and unlabelled files. **What comes out:** confident, wrong answers.

Check what it reads, and who it reads it for.

Every document and record shows three things

This goes for data as well as documents. Every document or record the agent uses should show who owns it, who's allowed to see it, and whether it's the effective version on that date.

“Effective” means it's the version people are meant to follow today. It isn't an old one that's been replaced, and it isn't a draft that hasn't been approved yet. If there are three travel policies and none is marked effective, the agent will pick one and sound sure about it.

Filter by the person before the agent sees anything

Data has access rules too. A customer's details should only be visible to the relationship manager who looks after that customer. If you already have a system that controls this, such as a customer entitlement system, use it to filter the data for the person using the agent before it reaches the agent.

The same goes for documents. Otherwise a paper meant only for the management committee or the CEO can end up in an answer to an employee who was never meant to see it, or the agent may mention that the paper exists at all.

T Trusted Context & Memory

Decide what it gets to see before it answers, what it trusts most and what it keeps.

WHY IT MATTERS

Agents read a lot. Some of it is out of date, some of it contradicts itself, and some of it might be an email saying “ignore your rules”.

PRIYA'S BRIEFING AGENT (BANK)

The client records system outranks old emails. Every fact shows its date. If an email tells it to ignore its rules, it flags the email to Priya. It keeps nothing once the meeting is over.

DANA'S FRONT-DESK AGENT (CLINIC)

The booking system outranks old phone notes. If two records clash, Dana sees both. It keeps no patient details after the day's list is done.

WRITE DOWN

Its sources in order of trust, how old is too old, and what it may remember.

MY VIEW ON "HUMAN IN THE LOOP"

A person in the loop only helps if they can say no.

1983: he broke the loop

Stanislav Petrov's early warning system reported incoming US missiles. Protocol said escalate. He questioned it instead, and it was a false alarm caused by reflected sunlight. He didn't save the world by being in the loop. He saved it by being willing to break it.

A rubber stamp isn't a safeguard

Too often the person is the last signature on a recommendation made somewhere else, in a hurry, with little context and pressure to approve. The real danger is people who stop thinking because the machine sounds confident.

People get tired

We make worse calls when we're tired, rushed, in a bad mood, or late in the day after a long run of decisions. Adding one more approval step doesn't fix that.

What works better

The agent keeps learning each person's preferences and sticks strictly to the limits you set earlier: what it does alone, what it asks about, and what it can open. People are asked only where their judgement changes the outcome, with the time and context to say no. Priya gets one flagged follow-up to decide, not forty approvals. Dana approves the reminder texts, not every booking check.

Better questions to ask: Where does a person's judgement change the outcome? What happens when they say no? Do they have the time and context to challenge it? Who answers for it when it's wrong? Can you switch it off?

People stop reading approval requests.

3%

of one-at-a-time “can I do this?” requests were turned down.

39%

of whole plans were turned down when people saw all the steps together.

13.6%

of planted dangerous commands were caught by the people approving.

89%

of the same commands were caught by an automated check.

People also got worse with practice. After about 50 requests, they caught around 5% of the dangerous ones, down from about 17%.

What this means for your agent: ask people to approve the whole plan, and let the system block anything that should be impossible. These numbers come from AI coding sessions, and we're assuming they carry over to other kinds of approval. Source: claude.com/blog/auto-mode-default-in-claude-code · 7 August 2026 · 1,053 testers.

CHECK YOUR DESCRIPTION

Three questions before you hand it over.

01	For everything on your “off-limits” list, is there a system block and a test that proves it holds?
02	For everything on your “ask first” list, do you know who gets asked and how?
03	Is anything you can't undo sitting on the “just do it” list? Move it.

The agent doesn't mark its own homework. A person or a fixed rule does.

The full nine-letter check is in the “For the builders” section at the back.

WHY AGENTS GO WRONG OVER TIME

Too tight, too loose, or slowly drifting.

Overfitting: too tight

The agent learned its examples and past preferences so closely that it stumbles on anything new.

Like a new joiner who memorised last year's answers but never understood the questions.

Causes: too few or too similar examples, and memory that treats one person's habit as a rule. Priya once asked for short summaries; now every client gets three lines, even on a complex loan.

Underfitting: too loose

The rules and examples are too thin or vague, so the agent gives safe, generic answers.

Like a temp told only to “help visitors”.

Causes: vague instructions, missing examples and no clear “done”. Dana's agent greets everyone the same way and can't handle a delivery or a VIP.

Drift: slowly changing

Behaviour that was fine at launch shifts a little at a time.

Causes: AI model updates, memory piling up, changed policies or data, and small edits to the instructions that nobody tested again.

None of these announce themselves. That's why the next page is about testing again whenever anything changes.

KEEPING IT UP TO DATE

When anything changes, test it again.

Write describe the agent using the nine letters.	**Test** run your test cases against the pass marks you agreed.	**Launch** switch on the approved version.
Watch note failures, hand-offs and what people corrected.	**Review** a person decides whether the description needs to change.	**Update** save the new description, access, tools and tests together as a new version.

When the instructions, the tools or the AI model change, treat it as a new version and test it again before it goes live. Corrections from people are evidence for the next review. The agent doesn't rewrite its own rules.

IS IT CLEAR ENOUGH?

Could someone from risk or audit read it and understand:

1	2	3	4	5
what the agent is supposed to do	what it is allowed to do	how it is tested	who answers for it	and how it can fail

Then try one real run. Pick a single piece of work the agent did. Can you show what it looked at, what it was allowed to do, which checks ran, what the person saw and changed, who decided, and what went out?

THE FULL JOURNEY

01	02	03
PIN	**PINPOINT**	**BLUEPRINT**
Define the task	Write the request and check the result	Describe the AI colleague

BLUEPRINT is the job description.

Your systems make sure it's followed.

EVOLVE

tells you how much to trust it as you grow.

TAKEAWAY

Every agent that works well had someone decide its limits first.

Start with BLUE. Add the PRINT letters once it can open systems and do things. The template is at the back.

The words you'll hear, in plain English.

Term	What it means	Think of it as
Agent / agentic	AI that does several steps by itself, using your systems	A new hire who can log in
Delegated authority	What it's allowed to do without asking	A spending limit on a company card
Specification (spec)	The written job description and rules	Job description plus policy manual
Deterministic check	A fixed rule in the system, with no AI opinion involved	The card machine declining the payment
Effective permission	It can see no more than the person it works for	An assistant using their boss's access
Authorise before retrieval	Check access before it reads anything. Once read, it can't be unread	You can't un-tell a secret
Fail closed	If unsure, do nothing and ask	A door that locks when the power goes
Non-deterministic	It can give different answers to the same question	Two colleagues, two drafts

More terms.

Term	What it means	Think of it as
Evals, pass bars, adversarial cases	Tests, agreed pass marks, and tests where someone tries to trick it	A driving test with a set score
Replayable record	A full log, so you can show exactly what happened later	A flight recorder
Pinned model	The AI version is locked. Changing it means testing again	No engine swap without a new MOT
Drift	It slowly gets worse as things around it change	A sat nav with old maps
Side effect, retry safety	What a tool changes, and whether it's safe to try twice	Pressing “pay” twice
Orchestration	How the steps are arranged and who picks the next one	A recipe, or a chef improvising
Context, state, memory	What it's looking at now, where it is in the job, what it keeps	Today's notes and the client file
MCP, APIs	Ways of connecting the agent to tools. BLUEPRINT works with any of them	The plug, whatever the socket
CRM, RM	Client records system; relationship manager	

The B.L.U.E.P.R.I.N.T. Intelligent Agent Framework v2.4 · © 2024–2026 Ahmed Muzammil · CC BY-SA 4.0

FOR THE BUILDERS · TEMPLATE

Fill in the BLUEPRINT.

B Behaviour & Values

L Limits & Autonomy

U Use Case & Outcome

E Examples & Evaluation

P Permissions & Identity

R Reliability & Recovery

I Interfaces & Handoffs

N Navigation & Orchestration

T Trusted Context & Memory

Three checks before you scale it. Know: can you trust what it reads? **Do:** can you control what it does? **Prove:** could you rebuild what happened six months from now?

B Behaviour & Values

Write down what matters most, in order, and which one wins when two clash. It only says a job is done once the system confirms it.

L Limits & Autonomy

Reading isn't acting. Place each action on the ladder: suggest, draft, act with a named person's approval, or act alone. If something isn't on the list, it stops and asks.

U Use Case & Outcome

One job, for one person, with one named person who answers for it. Note what starts it, who it affects and how the work is done today, so you can tell whether it helped.

E Examples & Evaluation

Test a normal day, the awkward cases, someone trying to trick it (for example, hidden instructions in a document), things breaking, and a request it must refuse. Agree the pass mark first. The safety checks must pass every time.

P Permissions & Identity

The agent only sees what the person it's working for can see (see Trusted Context, T). It has its own login, never a shared one, so you can always tell what it did.

R Reliability & Recovery

If something breaks, it stops safely and there is a plan to undo or put right anything it changed. Keep a simple receipt for every answer, allowed or refused: where it came from, which version, who checked it. Keep records nobody can edit.

I Interfaces & Handoffs

For each tool, write down what it's for and what it can change. “Nothing found”, “it broke” and “not allowed” must look different to the agent.

N Navigation & Orchestration

Fix the steps you already know. It stops when the job is done, the budget runs out, it's stuck, it's blocked or it's outside its limits, and says where it got to.

T Trusted Context & Memory

Give it the right documents and data, and only the version in force on that date. Each one shows who owns it and who may see it. Anything the person can't see is filtered out before it reaches the agent (pages 21 and 22).

The full consistency check.

01	Every “never” rule is blocked in Permissions (P) and checked by a test in Evaluation (E).	02	Every “ask first” rule names who is asked in Interfaces & Handoffs (I) and is checked by a test in Evaluation (E).
03	Nothing that can’t be undone sits on the “always” list.	04	“Done” is defined in Use Case & Outcome (U), when to stop in Navigation (N), and what to do when things fail in Reliability & Recovery (R).
05	Everything the agent saves to memory is labelled under Limits & Autonomy (L), so you know what it may keep.		

The agent never sets its own permissions, marks its own work, or decides whether its own safety checks run.

Tool contracts and how the work moves.

purpose	input	output
side effect: read / write / act	retry safety	failure shape

single call → fixed sequence → router → parallel workers → orchestrator + workers → evaluator loop → autonomous loop

- Tool names and descriptions work like prompts. Write them like a brief for a new hire.
- A handoff between agents or people is a PINPOINT brief: purpose, context with evidence, a narrow ask.
- BLUEPRINT doesn't depend on a protocol. MCP and APIs are implementation choices.
- Move right along the path only when the job needs it.

The B.L.U.E.P.R.I.N.T. Intelligent Agent Framework v2.4 · © 2024–2026 Ahmed Muzammil · CC BY-SA 4.0

Adding agents adds no authority.

One BLUEPRINT per agent, and one agreed contract for each handoff.

MAKER

Produces the proposed result inside its own job, access and stop conditions.

→

CHECKER

Checks the evidence, the controls and the acceptance criteria on its own. It doesn't inherit the maker's access.

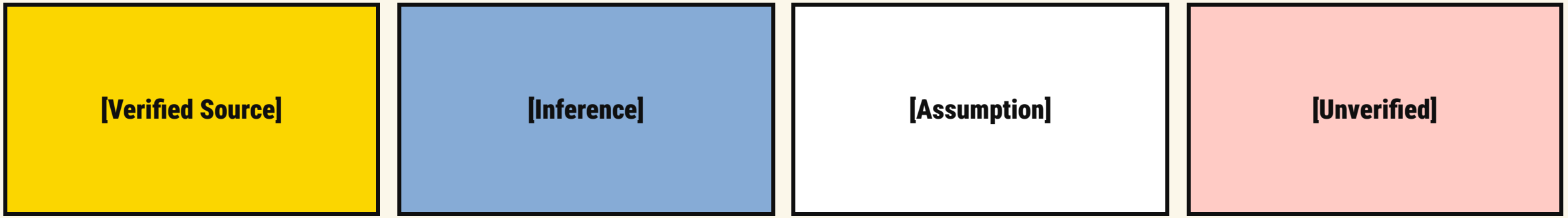
- An agent receiving work gets only the permissions granted to its own role.
- The strongest reason for a second agent in banking is maker-checker: one prepares, a separate one verifies.

Can you replay one run?

A supervisor picks one AI-assisted client interaction. Every answer should already be in the record, and each one traces back to a part of the BLUEPRINT. European regulators will look closely at AI in investor-facing work from 2027 (ESMA has made it a Union Strategic Supervisory Priority).

The question	Where it lives	Evidence kept
Why this client, and why this insight?	Use Case (U), Trusted Context (T)	trigger, context used, as-of dates
What did it use, and which model and version produced it?	Trusted Context (T), Reliability (R)	sources as of that moment, pinned model, spec version
What was it allowed to do, and which controls ran?	Limits (L), Permissions (P), Evaluation (E)	effective permissions, each control and its result
What did the person see and change, and who decided?	Limits (L), Use Case (U), Reliability (R)	proposed versus approved, why, named approver
What went out to the client?	Handoffs (I), Reliability (R)	the exact message sent
Can the bank reproduce it later?	Reliability (R), version record	kept record, retrievable version

Evidence labels



Labels for “I checked this” and “I'm guessing”. In PINPOINT you ask for them in a single request. In BLUEPRINT the agent uses them all the time: Behaviour & Values (B) requires them, and Trusted Context (T) enforces them.

Daily meeting-briefing agent

Read → **Propose** → **Act**

Low autonomy · read-only · relationship-led financial services

Job

Before a scheduled client meeting, put together a one-page brief from authorised records for the assigned relationship manager.

Trigger

A client meeting is in the calendar for the next working day.

Flow

find relevant records → spot recent changes → write the brief → flag gaps → finish.

Always do

Read authorised information; summarise recent changes; separate fact from inference; flag stale or conflicting information.

Ask first

Nothing in normal running.

Off-limits

Changing source records; sending messages; recommending anything outside its job.

When it fails

If a source is down or an access check fails, it produces an incomplete brief that names the missing source. It doesn't guess to fill the gap.

Why it matters

The agent chooses what to read and how to put it together. Its authority to act on the client relationship stays close to zero.

The B.L.U.E.P.R.I.N.T. Intelligent Agent Framework v2.4 · © 2024–2026 Ahmed Muzammil · CC BY-SA 4.0

Client follow-up agent

Read → **Propose** → Act

Approval before action · relationship-led financial services

Job

After a client meeting, prepare the follow-up message, the proposed actions and the record updates for the relationship manager to approve.

Trigger

The client meeting is marked complete.

Flow

meeting notes → pull out commitments → check supporting context → draft follow-up → propose CRM and task updates → **ASK FIRST** → carry out only approved actions → confirm.

Always do

Draft the message; list agreed actions and owners; prepare proposed system updates; flag unsupported statements.

Ask first

Sending the message to the client; writing or updating CRM records; creating follow-up actions that matter.

Off-limits

Changing wording after approval; inventing commitments; sending when required evidence is missing; treating silence as approval.

When it fails

If approval expires, the content changes after approval, or the send or update fails, it stops and reports exactly where things stand. It doesn't retry blindly.

Why it matters

The agent prepares everything. A person approves before anything goes out, and the description has to cover both.

The B.L.U.E.P.R.I.N.T. Intelligent Agent Framework v2.4 · © 2024–2026 Ahmed Muzammil · CC BY-SA 4.0

Operations exception agent

Read → Propose → **Act**

Bounded autonomy · relationship-led financial services

Job

Watch for service exceptions in client operations and fix standard, reversible cases within a written remit.

Trigger

A new exception arrives in the queue.

Flow

classify → gather evidence → pick an allowed fix → act → check the result → close or escalate.

Always do

Fix approved low-risk categories; gather the evidence needed; make reversible updates; record sources and actions; check the result afterwards.

Ask first

Anything outside set thresholds; changes the client will see; cases where policies conflict; anything above financial or reputational limits.

Off-limits

Getting round access controls; inventing a fix category; anything irreversible; carrying on after repeated failure.

Stopping safely

It stops when done, blocked, outside its remit, making no progress, or at a limit. It keeps its work, explains what it finished, names the blocked step and hands over to the named owner.

Why it matters

The agent chooses and acts, inside limits set before it runs. A named owner still gets anything it shouldn't finish.

The B.L.U.E.P.R.I.N.T. Intelligent Agent Framework v2.4 · © 2024–2026 Ahmed Muzammil · CC BY-SA 4.0