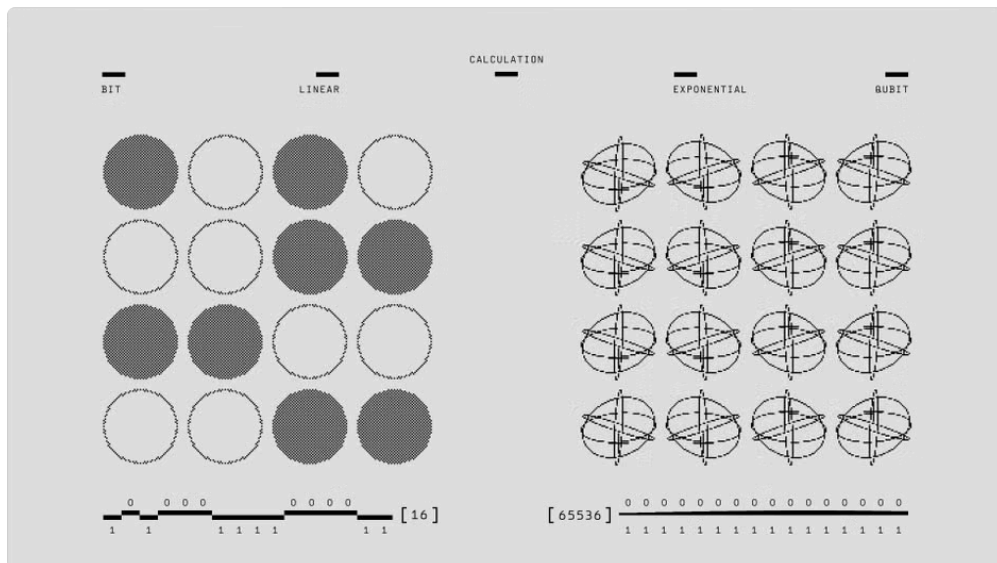


In a shared AI thread, the answer you get depends on more than the wording of your prompt. It also depends on which model you ask to respond. Suprmind's **@mentions** are a way to direct a request to a particular model within the conversation, rather than leaving model selection entirely to a default or orchestration setting.

That distinction matters when a team is comparing options, reviewing policy language, or preparing a decision memo. You may want one model to generate a first draft, another to check the reasoning, and a third to identify missing evidence—all in the same working thread. With model-directed prompts, teammates can **control which model responds** while keeping the discussion and its context together.

This guide explains how to think about @mentions, how they fit into multi-model orchestration, what to validate before relying on an answer, and how plan-based access can affect model seats and features.



What an @mention does

An @mention is a routing instruction in your message. You name an available model—using the mention picker or the supported mention format—and then provide the task. Instead of asking an unspecified assistant to answer,

you make the intended responder explicit.

For example, a team might write:

- **@Model A:** Summarize the main risks in this vendor proposal.
- **@Model B:** Challenge that summary. What assumptions or counterarguments are missing?
- **@Model C:** Turn the discussion into a decision table with evidence and open questions.

The exact model names available to mention depend on the workspace and its enabled models. Suprmind may connect to model providers such as OpenAI and Anthropic, but access to a provider, model version, or related feature can depend on the account configuration and plan. Check the model picker in your workspace rather than assuming every model is available to every user.

How model selection works in a shared thread

In a multi-model workflow, the thread is the shared research.com workspace and the mention identifies the requested responder. That lets a team keep the original question, source material, follow-up instructions, and model responses in one place. You can compare answers without copying the same context across separate chats or losing track of which response came from which model.

1. **Set the decision context.** State what you are deciding, who the answer is for, and what evidence the model should use.
2. **Direct a specific task.** Mention the model you want for a defined job, such as summarizing, critiquing, extracting clauses, or drafting.
3. **Review the response in context.** Compare it with the prompt, source material, and other contributions in the thread.
4. **Route the next step.** Ask a different model to challenge the answer, apply a rubric, or produce a structured output.
5. **Validate before acting.** Confirm important claims against source documents, calculations, and human judgment.

A mention should not be treated as a guarantee that one model is objectively “best.” It is a way to choose a responder for a task. The value comes from matching the task to the model’s strengths, giving it clear instructions, and checking the result.

Example: a decision memo

Imagine a strategy team reviewing three product initiatives. One model can summarize the options using a consistent template. A second can argue against the leading option and surface risks. A third can turn the discussion into a comparison table. Team members can then review the outputs side by side in the same thread and decide which points require evidence or stakeholder input.

This is different from asking several models the same broad question and choosing whichever answer sounds most confident. A stronger workflow assigns each responder a distinct role, uses shared criteria, and preserves the rationale behind the final recommendation.

@Mentions, orchestration, and decision intelligence

Multi-model orchestration means coordinating models or model tasks so they contribute to a larger workflow. An @mention is a useful manual control within that process: it lets a person direct a particular step instead of

relying only on an automatic selection or a single default model.

For decision intelligence, the goal is not simply to generate more text. It is to make a decision easier to inspect. A good workflow distinguishes facts from assumptions, compares options using the same criteria, captures uncertainty, and identifies what needs validation. Model responses can help structure that work, but they do not replace the evidence or the decision owner.

Use structured validation, not just a second opinion

When one model reviews another, give the reviewer a concrete checklist. For example, ask it to:

- Separate claims supported by the supplied documents from claims that are inferred.
- Identify missing information and conflicting evidence.
- Apply the same scoring criteria to every option.
- Flag calculations or legal, financial, and policy claims for human verification.
- Return findings in a table with a source, confidence level, and follow-up action.

Structured validation is more useful than a prompt such as “Is this correct?” A model can sound persuasive while overlooking a key detail, misreading a clause, or relying on information that was not supplied. Requesting evidence and uncertainty makes it easier for a reviewer to decide what to trust and what to check.

How to write effective @mention prompts

A model mention chooses who should respond; the rest of the prompt determines how useful that response is. Give the model enough context to understand the task, but be precise about the expected output.

A reusable prompt pattern

@[Model]: You are supporting [team or decision]. Using only [named materials], complete [specific task]. Apply these criteria: [criteria]. Distinguish evidence from assumptions, cite the relevant source or section where possible, and list unresolved questions. Return the result as [format]. Do not invent missing information.

For example:

@[Model]: Review the attached policy draft for differences from the approved standard. Return a table with the clause, the difference, the likely operational impact, and a question for the policy owner. Use only the supplied documents and flag any interpretation that needs legal review.

For a comparison workflow, use the same prompt and criteria for each model. If you change the instructions between responses, you may be comparing prompt quality rather than model behavior. Keep the source material, evaluation rubric, and desired format consistent, then record where responses agree and where they differ.

Model seats and plan-based access

Model access can be shaped by both the workspace plan and its configuration. A **model seat** is a practical way to think about capacity or access associated with a model in a team workflow. It is not necessarily the same thing as a user seat, and the exact meaning—who can use a model, how usage is allocated, or whether access is limited—should be confirmed in Suprmind’s current plan details.

Plan-based pricing can make costs more predictable than paying solely per token, especially for teams that collaborate across many threads. But a subscription price does not automatically mean every model, feature, or

usage level is unlimited. A plan may define which models can be selected, how many model seats are available, or which orchestration and administration controls are included. Review the plan terms for limits and any separate usage charges.

For example, the Spark plan has a price example of **\$19/month**. Treat that as a starting point for evaluating fit, not as proof that every model or capability is included. Confirm current billing terms, included seats, usage allowances, and any add-ons before purchasing.

Feature gating across Spark, Pro, Frontier, Power, and Enterprise

Suprmind plan names may include Spark, Pro, Frontier, Power, and Enterprise. The important evaluation question is not only “Which plan has the lowest price?” It is “Which plan gives this team the model access and workflow controls it actually needs?” Specific entitlements can change, so verify them in the current plan comparison or with Suprmind before making a procurement decision.

Plan What to verify for an @mention workflow Spark Included model options, model-seat limits, usage allowance, and whether the core shared-thread workflow is available. Pro Whether it expands model choice, collaboration capacity, or usage limits for a growing team. Frontier Which advanced models or orchestration features are included, and whether access differs by workspace or user. Power Whether higher-volume or more complex workflows receive additional capacity, controls, or model seats. Enterprise Administration, security, governance, support, deployment, and commercial terms relevant to organizational use.

This table is an evaluation checklist, not a claim that a particular feature belongs to a particular tier. Ask for the current feature matrix and test the exact models and collaboration pattern your team plans to use.

Plan-based pricing versus per-token billing

With plan-based pricing, a team pays a subscription price for a defined package of access and features. That can help with budgeting and make it easier to invite colleagues into a shared workflow. With per-token billing, charges are tied more directly to the amount of model input and output consumed. Actual cost can vary with model choice, prompt length, attached context, and response length.

Neither approach is automatically cheaper. A plan may be easier to forecast, but usage caps, model-seat limits, and excluded features matter. Per-token billing can give teams more direct visibility into consumption, but frequent or lengthy multi-model reviews may make costs harder to predict. Compare the total workflow cost—not just the headline price—including the number of users, models, review steps, and likely volume.. Pretty simple.

As a comparison point, Gaviti is AR software with pricing by quote; a free trial and demo are available. Paylocity uses custom pricing and does not offer a free trial. Zoho Books is listed at **\$15 per month** and offers a **14-day free trial**. These examples illustrate that software vendors package access differently: subscription prices, quoted contracts, trials, and demos each answer different evaluation questions. They are not direct substitutes for a model-orchestration product.

Before you depend on a model response

Let me tell you about a situation I encountered made a mistake that cost them thousands.. Want to know something interesting? use @mentions to make routing explicit, then establish a validation habit around consequential work. Keep a record of the prompt, source material, selected model, and any human edits when the output informs a significant decision. For sensitive material, follow your organization’s rules for data handling and confirm the product’s current privacy and security terms.

- Check that the model you mention is available to your workspace.
- Confirm plan limits, model seats, and any usage or billing rules.
- Use a clear role and a specific output format.
- Ask for sources, assumptions, uncertainty, and unresolved questions.
- Verify material claims against authoritative documents or subject-matter experts.
- Use the same rubric when comparing responses from different models.

The takeaway

@mentions help teams **control which model responds** inside a shared Suprmind thread. Used well, they support a deliberate sequence: direct a model to draft or analyze, ask another to challenge the result, and use structured validation to surface what still needs review. That makes multi-model orchestration more transparent and useful for decision work.

Before adopting the workflow, test the models your team needs, confirm what each plan includes, and compare plan-based costs with expected usage. A mention can route a task; sound decisions still depend on clear criteria, reliable evidence, and human accountability.