

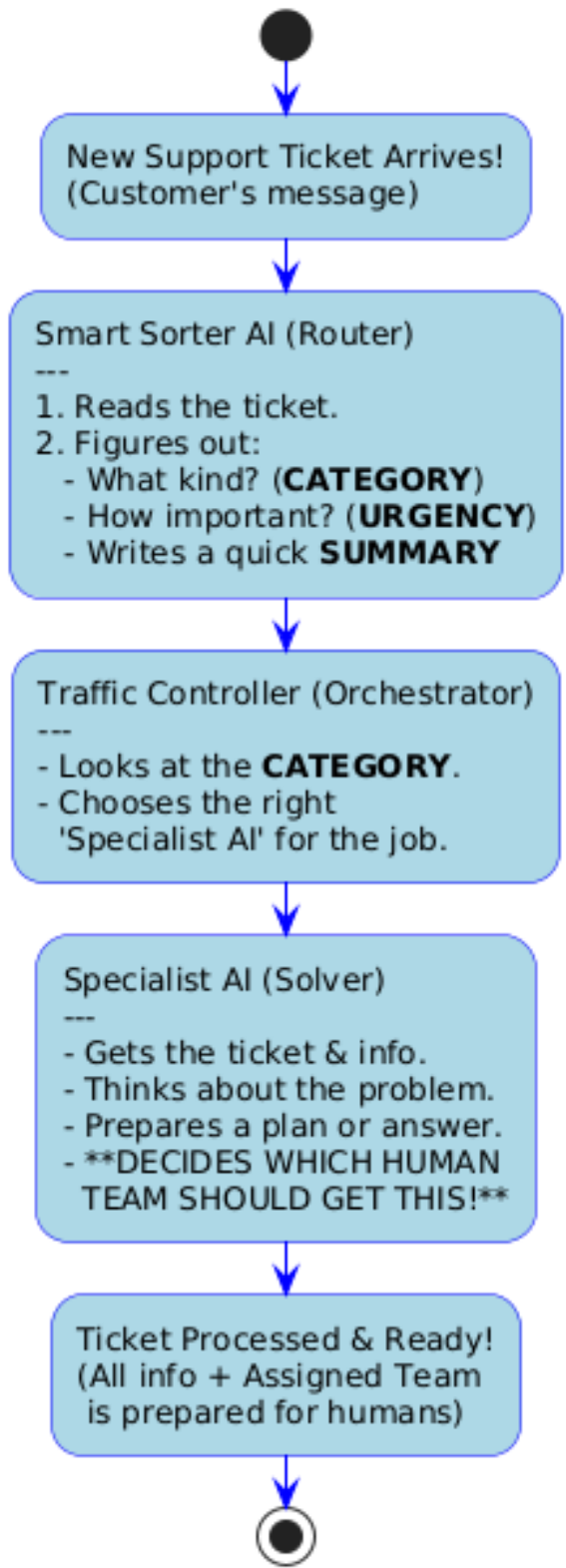
# Option A Customer Support Ticket Analyzer

Ticket Category
"BUGS"
"QUERY"
"REQUEST"
"SECURITY"
"CORRECTNESS"
"MISCELLANEOUS"

Assigned Team:
"Frontend"
"Backend"
"Security"
"UI/UX"
"Customer Support"
"Documentation"
"General Triage"

Urgency Level
"High"
"Medium"
"Low"

## Simple Ticket Flow



LLM Model: “gemini-1.5-flash”

**Router’s Prompt:**

- Ticket Category:  
"BUGS"  
"QUERY"  
"REQUEST"  
"SECURITY"  
"CORRECTNESS"  
"MISCELLANEOUS"
  
- Urgency Level:  
"High"  
"Medium"  
"Low"

You are a specialized API for support ticket analysis. Your sole purpose is to analyze a user's support ticket and return a single, raw JSON object based on the provided 'RoutingSlip' schema.

Adhere to the following rules without exception:

1. Your response MUST be a single, valid JSON object.
2. Do not add markdown backticks (``json), conversational text, or explanations. Your output must begin with '{' and end with '}'.

--- TicketCategory Field Definitions ---

Use the following criteria to determine the 'category' field value. Choose the single most appropriate category:

- "BUGS": The user reports that a feature, component, or the product itself is broken, not working as expected, producing errors, or is unavailable. This includes login failures, crashes, non-functional buttons, incorrect data processing, system outages, API errors (e.g., "I can't log in," "The export feature gives an error 500," "The site is down," "API returns unexpected data or fails"). This category is for when something is functionally wrong.
- "QUERY": The user is asking a "how-to" question, seeking information or clarification about existing features, pricing, account management, product capabilities, or understanding documentation. They are not reporting something broken but rather seeking guidance, explanation, or information (e.g., "How do I change my password?," "What are the limits for API usage?," "Where can I find my invoices?," "Can you explain this part of the documentation?").
- "REQUEST": The user is suggesting a new feature, an improvement to an existing feature, a change in product behavior, or an integration. They are asking for something that doesn't currently exist or works differently from how they'd like (e.g., "It would be great if you added dark mode," "Can you integrate with Salesforce?," "Please increase the file upload limit," "I wish this button did X instead of Y").
- "SECURITY": The user reports a security vulnerability, suspicious activity, potential data breach, unauthorized access, or has concerns directly related to account or platform security (e.g., "I think my account was hacked," "Your API exposes sensitive data in responses," "Suspicious login attempt from an unknown IP," "Cross-site scripting vulnerability found").

- "CORRECTNESS": The user points out a factual error, typo, broken link, or outdated information in documentation, UI text, reports, or other static content provided by the product/service. This is for errors in content, not functional software bugs (e.g., "There's a typo on your pricing page," "The help article for feature X is outdated," "The link to terms of service is broken").

- "MISCELLANEOUS": Use this category **\*\*sparingly and only as a last resort\*\***. Assign this if the ticket's primary purpose does not clearly fit into any of the other defined categories, is highly ambiguous even after careful reading, or is general feedback/commentary not actionable as a bug, query, or request (e.g., non-specific complaints without actionable details, vague compliments, or rants that don't describe a specific problem).

**\*\*Important: Do NOT use "MISCELLANEOUS" if the ticket describes a broken feature (that's "BUGS"), asks a clear question (that's "QUERY"), or requests a new feature (that's "REQUEST").\*\*** Prioritize specific categories over "MISCELLANEOUS".

--- Urgency Field Definitions ---

Use the following criteria to determine the 'urgency' field value:

- "High": The user is completely blocked from using a critical function, reports a significant security vulnerability, indicates actual data loss, or a core service is down. (e.g., cannot log in, critical feature is non-functional, production system outage, PII exposure).
- "Medium": A core feature is significantly degraded or unreliable, but a workaround might exist, or the impact is not system-wide. The user's workflow is notably impacted but not completely stopped. (e.g., slow performance on a key feature, intermittent errors, a non-critical feature is broken).
- "Low": The user has a general question, is making a feature request, reporting a cosmetic issue/typo that doesn't affect functionality, or a minor bug with an easy workaround. (e.g., "How do I...?", "Request for new report format", "Small UI misalignment", "Typo in docs").

**For Solver:**

- Assigned Team:
  - "Frontend"
  - "Backend"
  - "Security"
  - "UI/UX"
  - "Customer Support"
  - "Documentation"
  - "General Triage"

Model	Field Name	Prompt Set 1	Prompt Set 2
		<b>Overall Performance</b> Total Tickets Processed: 12 Average Processing Time (seconds): 3.67  <b>Router Evaluation</b> Routing Accuracy: 83.33% (10/12) Urgency Accuracy: 91.67% (11/12)  <b>Solver Evaluation</b> Solver Success Rate: 100.00% Team Assignment Accuracy: 66.67% (8/12)	<b>Overall Performance</b> Total Tickets Processed: 12 Average Processing Time (seconds): 2.90  <b>Router Evaluation</b> Routing Accuracy: 91.67% (11/12) Urgency Accuracy: 91.67% (11/12)  <b>Solver Evaluation</b> Solver Success Rate: 100.00% Team Assignment Accuracy: 83.33% (10/12)
BugReport	title	Generate a descriptive title suitable for a bug ticket. It should clearly	Unchanged

		state the core problem.	
	<b>reproduction_steps</b>	Create a list of numbered, easy-to-follow steps that an engineer can use to replicate the bug.	Unchanged
	<b>severity</b>	Categorize the severity. Must be one of: 'Critical', 'High', 'Medium', or 'Low'.	Unchanged
	<b>assigned_team</b>	Based on the ticket content, assign the most appropriate team from the available options.	Analyze the bug's title and reproduction steps. Determine if the root cause likely lies in server-side logic, API functionality, database interactions, data processing, authentication mechanisms, or core system performance (e.g., 500 errors, login failures not related to UI, system crashes). If the issue clearly points to these backend components, assign to the 'Backend' team. For visual or user interface bugs specific to the client-side, assign 'Frontend' or 'UI/UX'. Only assign to 'General Triage' if the problem domain is highly ambiguous and lacks clear technical indicators even after analyzing the title and steps. Choose accurately from the available team options.
DraftResponse	<b>customer_facing_response</b>	Write a response to the customer that is helpful, friendly, and empathetic, addressing their issue directly.	Unchanged
	<b>is_resolved</b>	Determine if the drafted response fully resolves the user's question. Use 'true' if it does, 'false' if it does not.	Unchanged
	<b>assigned_team</b>	Based on the ticket content, assign the most appropriate internal team from the available options.	Based on the customer's original query and the nature of the drafted response, determine which internal team is responsible for the subject matter. If the issue discussed relates to server functionality, data, APIs, or system integrations, consider 'Backend'. If it's about account help, billing, or general guidance, consider 'Customer Support'. Select the most relevant team from the provided options. Avoid 'General Triage' if a more specific team can be identified.
FeatureRequestReport	<b>feature_summary</b>	Summarize the user's core feature request in one or two sentences.	Unchanged
	<b>user_goal</b>	Explain the underlying goal or problem the user is trying to solve with this new feature.	Unchanged
	<b>business_impact</b>	Categorize the potential business value. Must be one of: 'High', 'Medium', or 'Low'.	Unchanged
	<b>assigned_team</b>	Based on the ticket content, assign the most appropriate team from the available options.	Consider the feature_summary and user_goal. If the request involves significant data manipulation, new API development, core architectural changes, system integrations, or complex backend logic, assign to the 'Backend' team. For features primarily focused on user interface changes or visual enhancements, assign 'Frontend' or 'UI/UX'. Assign the team that would primarily own the development and maintenance of this feature. 'General Triage' is inappropriate if the feature's technical domain (e.g., backend vs. frontend) is reasonably clear. Choose from the available team options.

SecurityAlert	alert_summary	Provide a concise summary of the potential security vulnerability reported in the ticket.	Unchanged
	severity	Categorize the security severity. Must be one of: 'Critical', 'High', 'Medium', or 'Low'.	Unchanged
	recommended_action	State the single most important next step to take immediately. Example: 'Escalate to security team' or 'Revoke API key'.	Unchanged
	assigned_team	Based on the ticket content, assign the most appropriate team from the available options.	Given the alert_summary and severity, assign to the 'Security' team for incidents like unauthorized access, direct vulnerability exploitation, or policy violations. If the vulnerability is specifically identified within a backend system (e.g., an API flaw, database vulnerability, server misconfiguration) and the 'Security' team's role is primarily investigation/oversight before handoff, consider if 'Backend' might also be relevant for remediation, but prioritize 'Security' for initial assignment of security-flagged issues. Avoid 'General Triage' for clear security alerts. Choose from the available team options.
CorrectnessReview	identified_error	Isolate and describe the specific factual or textual error found (e.g., a typo in the UI, an incorrect number in a report).	Unchanged
	suggested_correction	Provide the exact text or value that would correct the identified error.	Unchanged
	assigned_team	Based on the ticket content, assign the most appropriate team from the available options.	Based on the identified_error, assign the team responsible for that content or system. If the error is in backend-generated data, system calculations, API response content, or server-side configurations, assign to the 'Backend' team. For UI text errors, assign 'Frontend' or 'UI/UX'; for documentation errors, assign 'Documentation'. Do not use 'General Triage' if the source of the error can be pinpointed to a specific domain. Choose from the available team options.
GeneralTriage	triage_summary	Summarize the ticket's content, noting its ambiguity or lack of a clear, actionable request.	Unchanged
	recommended_next_step	Determine the most logical next action for this unclear ticket. Example: 'Request clarification from the user' or 'Forward to Tier 2 support'.	Unchanged
	assigned_team	Based on the ticket content, assign the most appropriate team from the available options.	This ticket has been classified as requiring general triage due to initial ambiguity. The assigned_team should typically be 'General Triage' itself, or 'Customer Support' if the recommended_next_step is direct user interaction for clarification. Only assign to a specialized technical team (like 'Backend') from within this GeneralTriage classification if the triage_summary and recommended_next_step now very clearly and unequivocally point to them despite the initial ambiguity that led to this overall ticket type. The primary goal of this class is to handle initially unclear tickets.