R&D Proposal: Automated Client Reporting System

Date: February 9, 2026

Status: Draft/Proposal

Topic: Automated Technical-to-Client Translation Pipeline

---

1. Problem Statement

1. Redundancy: Developers write a daily log, and managers manually rewrite it for clients.
2. Communication Gap: Technical jargon (e.g., "fixed API latency") often confuses or alarms non-technical clients.
3. Time Loss: Managers spend significant hours acting as manual translators.

2. Objective

To build an automated pipeline that triggers when an employee logs out. The system will ingest raw technical summaries, use Generative AI to rewrite them into professional client updates, and automatically push them to the management dashboard.

3. Proposed Solution Architecture

3.1. The Workflow

1. Trigger: Employee submits daily log/logout.

2. Input: System captures Project_ID and Technical_Summary.

3. Processing: AI Agent strips jargon and reformats for business value.

4. Output: System generates a Client_Friendly_Summary.

5. Action: Database update triggers on the Client Dashboard.

3.2. Technical Stack

LLM Engine: GPT-4o-mini or Gemini 1.5 Flash (Cloud) / Llama-3 (Local).

Backend: Python (FastAPI) or Node.js.

4. The Core Logic: Prompt Engineering

The system uses a "Translation" prompt rather than a simple summary prompt.

System Prompt Template:
"Role: You are an expert Client Relationship Manager.
Task: Rewrite the following internal technical update into a non-technical, professional summary.
Rules: Simplify technical terms (e.g., "Refactored API" -> "Improved system speed"), keep tone professional, and remove sensitive data."

5. Use Case Scenarios

Scenario	Developer Input (Technical)	Automated Client Output
Bug Fix	Fixed null pointer exception in auth flow.	Resolved an issue preventing smooth user logins.
Backend Work	Migrated MongoDB cluster & optimized indexes.	Upgraded database infrastructure for better performance.
UI/Frontend	Fixed CSS alignment on mobile media queries.	Polished the dashboard layout for mobile devices.
Blocking Issue	Server crashed due to memory leak.	Investigating a performance hiccup; maintenance is underway.

6. Implementation Roadmap

Phase 1 (Shadow Mode): AI generates drafts but does not publish. Managers review accuracy.

Phase 2 (Full Automation): System publishes directly. Developers get a "Confirm" prompt before logout.

7. Database Schema Updates

Column Name	Type	Description
update_id	INT	Primary Key
project_id	INT	Links to Client Project
technical_log	TEXT	Input: 'Fixed API bug...'
client_log	TEXT	Output: 'Resolved stability issue...'
visibility	ENUM	'Internal' vs 'Client-Facing'

8. Conclusion

By implementing this layer, we decouple work execution from work reporting, saving estimated 3-5 management hours per project per week.