Le Protocole ALERTE!!!

Du Signal Émotionnel à la Correction Systémique

L'un des défauts majeurs des LLM actuels est leur **"confiance obstinée"** : lorsqu'un modèle part dans une mauvaise direction (hallucination ou boucle logique), il a tendance à justifier son erreur plutôt qu'à la corriger. Pour briser cette inertie, j'ai conçu le **Protocole ALERTE**, un mécanisme de gouvernance activé par un signal organique : **!!!**.

J'ai choisi ce déclencheur pour sa nature intuitive. Dans le feu de l'action, face à un agent qui refuse de comprendre, je n'ai pas envie de taper une commande complexe `/admin --override-mode`. Je tape naturellement des points d'exclamation pour marquer mon agacement. J'ai transformé cette impulsion émotionnelle en un **disjoncteur cognitif**.

1. L'Injection de Contexte Prioritaire

Le Prompt Engineering Dynamique

Au moment où l'orchestrateur (**AgentSemi**) détecte le motif **!!!**, il ne se contente pas d'ajouter une instruction à l'historique. Il modifie fondamentalement la structure du prompt système pour le tour suivant.

Le système injecte un **artefact mémoriel** (un Souvenir de type règle) avec un score de pertinence artificielle de **999.0**, écrasant toutes les autres instructions contextuelles. Ce "Méta-Prompt" force le LLM à abandonner sa persona d'assistant serviable pour adopter celle d'un **auditeur critique**. Il impose une méthodologie de débogage stricte : "Vérifier la syntaxe avant la logique", "Remettre en question les hypothèses précédentes" et "Solliciter la validation humaine étape par étape".

Méthodologie de débogage imposée
• Vérifier la syntaxe avant la logique
• Remettre en question les hypothèses précédentes
• Solliciter la validation humaine étape par étape

Le résultat est immédiat : le modèle cesse de "forcer" sa solution, s'excuse (une seule fois, pour éviter le bruit) et passe en mode **"Doute Structuré"**. Ce changement de contexte radical est la

seule méthode fiable que j'ai trouvée pour stopper net une hallucination en cours sur un modèle local de 14B.

2. La Boucle Réflexive

Apprentissage In-Context Permanent

Cependant, corriger l'erreur immédiate ne suffit pas ; il faut empêcher qu'elle se reproduise. C'est ici qu'intervient la dimension agentique du système. En parallèle de la réponse, le signal **!!!** réveille l'**AgentReflexor**.

Cet agent autonome lance une **analyse post-mortem** de la conversation en arrière-plan. Il examine les tours précédents pour identifier la cause racine du désalignement (ex: "Le modèle a confondu deux bibliothèques Python similaires"). Il génère ensuite une **Règle de Correction Comportementale** qui est vectorisée et stockée dans la mémoire réflexive.

La puissance de ce système réside dans son intégration avec le **RAG** (Retrieval-Augmented Generation). Lors des futures conversations, si un contexte similaire se présente, le moteur de recherche remontera cette règle spécifique. Ainsi, le système ne se contente pas d'être corrigé : **il apprend de mes frustrations**.

> *« Une colère exprimée via **!!!** aujourd'hui devient une règle de gouvernance permanente pour demain, rendant le système de plus en plus robuste et aligné avec ma logique de développement sans nécessiter de fine-tuning coûteux. »*

Architecture du Protocole ALERTE