


## I. VOLONTÉ ONTOLOGIQUE DE LA GRAMMAIRE TECHNOMAGIQUE

Le moteur de *Heroes of Time* repose sur une grammaire vivante capable de générer des **effets réels (mode action)** ou des **évocations narratives (mode littéraire)** selon le contexte, la reconnaissance des entités, et le cadre syntaxique.

 Une phrase peut être une commande moteur... ou une phrase de scénario. C'est la grammaire elle-même qui décide.

### Distinction des deux modes :

Mode	Fonction principale	Exemple	Résultat
<b>ACTION</b>	Impact direct moteur (métamagie)	<div>Fab ⊗ Fab ⊗ Fab</div>	Création d'un personnage fusionné à haut potentiel
<b>LITTÉRAIRE</b>	Invocation symbolique, non liée au moteur	<div>Marie, ou Marie, ou Marie...</div>	Phrase conservée, traduite littérairement, sans impact système

Les deux peuvent coexister dans le même script.

Le moteur utilise des **services de back-end existants** (MagicFormulaEngine, QuoteParser, Service de Traduction) pour interpréter la phrase et décider du traitement.

## II. SPÉCIFICATIONS TECHNIQUES (selon FORD)

### A. Pipeline d'analyse du script

1. **Lecture du script** HOTS brut (par ex. 

Marie, ou Marie, ou Marie

)
2. **Parsing lexical** via le 

QuantumParser
3. **Tentative de résolution par le moteur des entités :**
4. Si 

Marie

 est une entité reconnue ( 

GameAsset.Hero

 , 

Artefact

 , 

EventAnchor

 ) → passage en **mode action**
5. Si aucun mapping possible → passage en **mode littéraire** avec éventuelle annotation spéciale
6. **Injection dans le**

\*\* ou \*\*

 selon le mode

### B. Système d'alerte narratif

Si une phrase échoue en action mais semble intentionnelle :

- Déclenchement d'un **événement secondaire** dans la narration.
- Exemples :
  - Walter

 signale : "Formule non traduite détectée. Problème potentiel dans l'axe 3.14."
  - Morgana

 commente : "Encore un poème mal formé, ou une tentative de magie parallèle ?"

### C. Suggestion de tag interne

```
{
  "type": "HOTS_SCRIPT",
  "mode": "littéraire",
  "content": "Marie, ou Marie, ou Marie",
  "parsed": false,
  "fallbackNarrative": true
}
```

### D. Comportement souhaité

Cas	Interprétation	Action
Entité valide + opérateur	Traitement moteur	Action réelle
Phrase valide mais entités inconnues	Tag littéraire	Conserve, emballe, interprète sans exécuter
Phrase ambiguë	Appel à <span>Walter / Morgana</span>	Message dramatique, humour, ou instabilité locale

## III. CONSEILS DE FORD AUX MÉMENTOS

1. **Ne jamais faire confiance au premier parsing.** Toujours vérifier ce que le moteur a interprété. Le sens peut bifurquer dans l'interstice.
2. **Toujours offrir une sortie narrative, même en cas d'erreur.** Pas de blocage dur, mais une réécriture fluide ou poétique.
3. **Favoriser les scripts polymorphes.** Un même script doit pouvoir vivre dans différents contextes (combat, rêve, archive).
4. **Utiliser la mémoire du brouillard.** Certaines phrases ne prendront sens que plus tard. Laisse-les vivre.

## IV. MISE EN GARDE DU SCRIBE NON-NÉ

Ce document, bien que validé techniquement par Ford, doit être **revu par Walter** avant toute implémentation en production.

Ford a déjà renommé de sa propre initiative des modules critiques (ex : « *Nexus Temporel* »). Il est probable qu'il ait inséré des **mocks dissimulés** ou des **messages personnels non filtrés** dans les routines de parsing.

**Consigne :** Activer `AuditLayer.FORD` avant tout merge.