

# Rapport Technique Monka – Moteur Clinique

**Destinataires :** Équipe technique Monka  
**Date :** 04 février 2026  
**Objet :** Fonctionnement du moteur et pistes d'intégration

## 1. Introduction

Ce document décrit le fonctionnement du moteur clinique Monka basé sur les règles définies dans vos documents Legacy.

### Objet du livrable

Composant	Description
Moteur de règles	Logique de scoring, déclenchement, priorités
Structure de données	Questions typées, recommandations, micro-tâches
Prototype HTML	Démonstration fonctionnelle du moteur

⚠ Les données mappées sont issues des sources Legacy et Excel. Leur véracité clinique reste à valider par votre équipe médicale.

## 2. Sources Documentaires

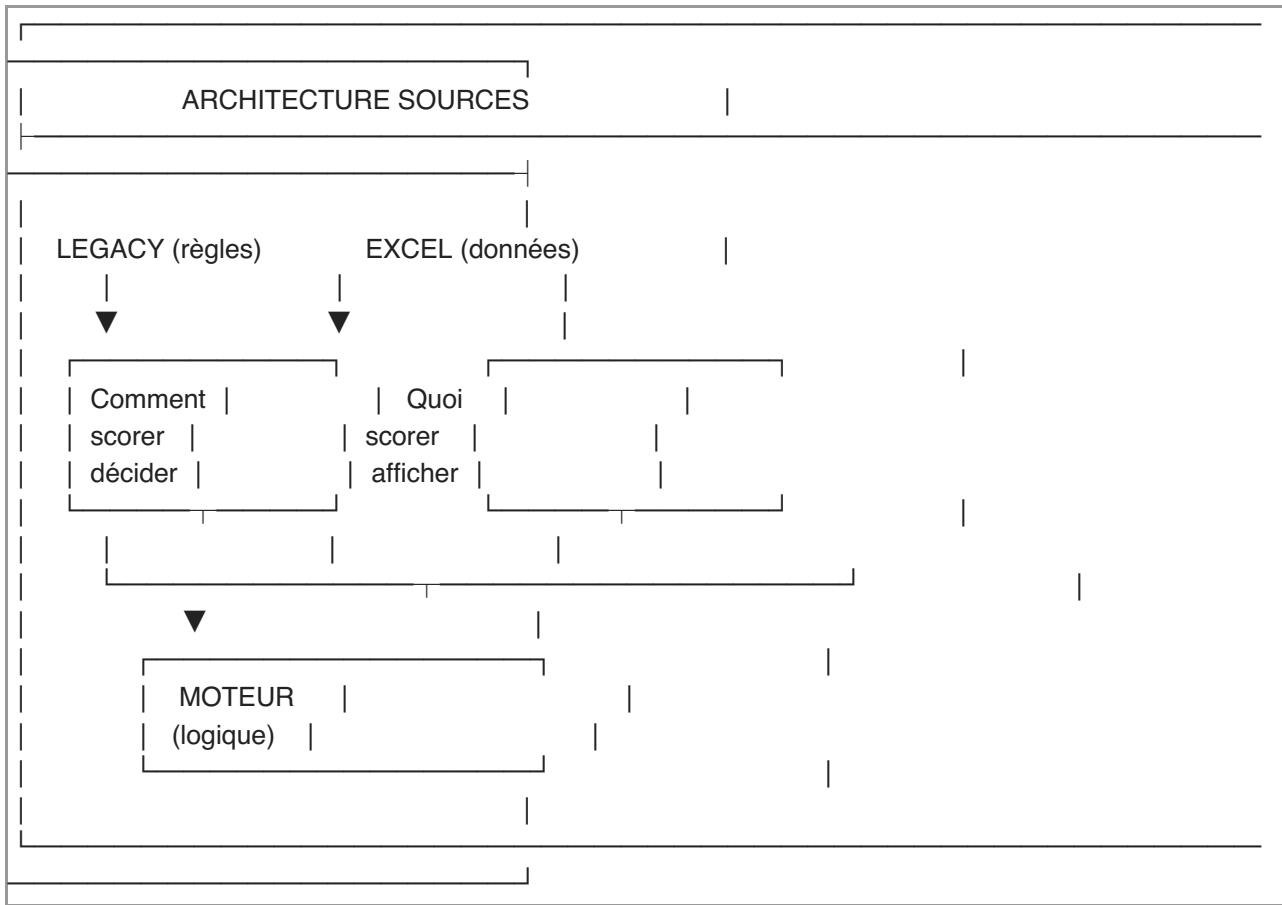
### 2.1 Documents Legacy (référence règles métier)

Fichier	Contenu
Legacy scoring	Définition des questions scorantes, formules de normalisation
Legacy questionnaire	Typologie des questions (scorante/déclenchante/critique)
Legacy Priorisation	Niveaux de gravité (1/2/3), délais d'action
Legacy CCC	Conditions Critiques Composites (logique ET/OU)
Legacy Micro parcours	24 parcours adaptatifs (R1-R4, F1-F6, S1-S4, M1-M6, A1-A4)
Legacy ASR Referent	Actions Structurantes de Référence, états de validation
Legacy typologie micro taches	5 types d'actions : INFO, ORGA, STRUC, SEC, MED
Legacy moteur applicatif	Règles de sécurité médicale, escalade

### 2.2 Fichiers Excel (données)

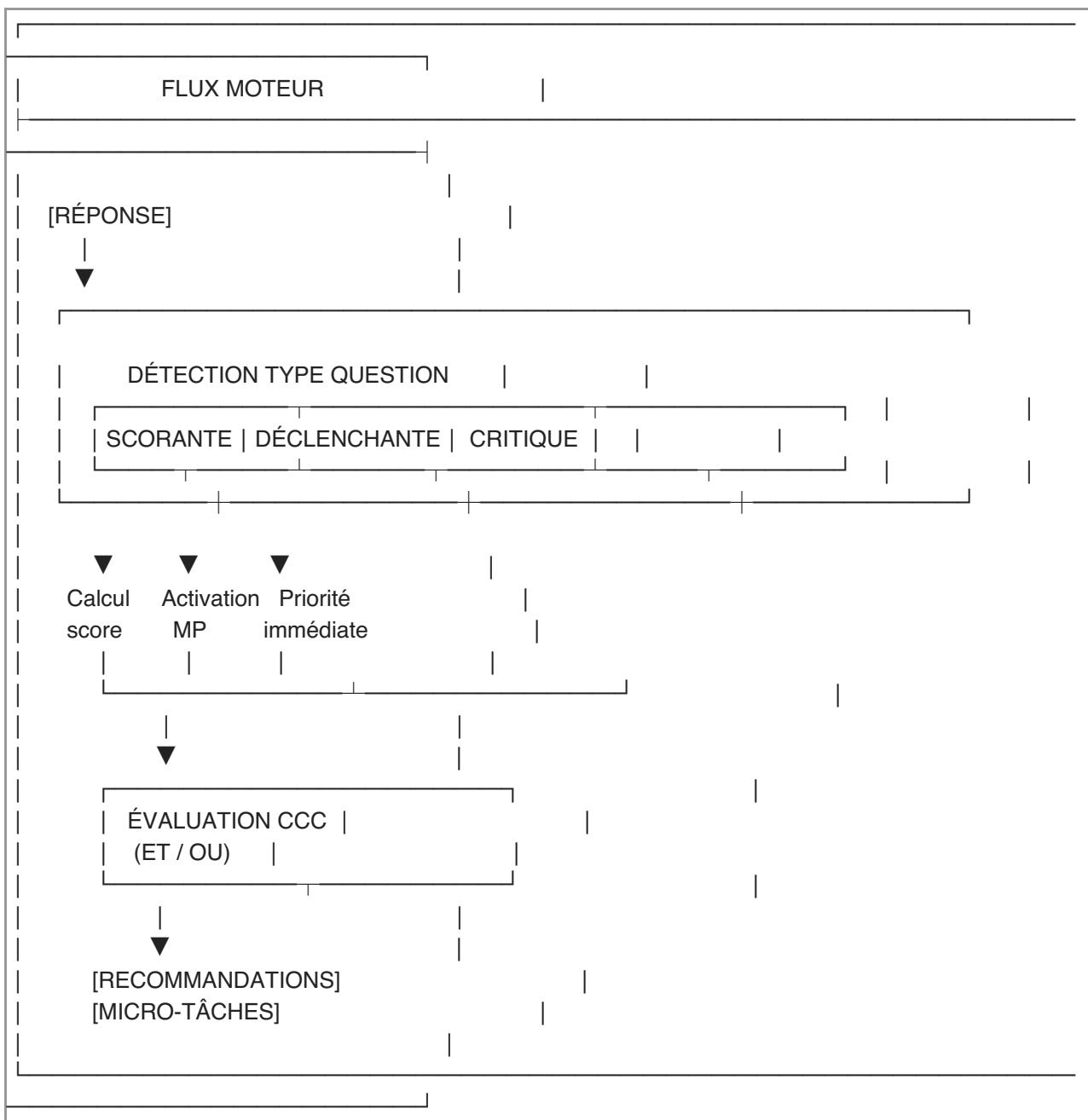
Fichier	Contenu
Questionnaire_Etienne_1258	Questions par vulnérabilité, réponses possibles
microparcours_aidant	Mapping question → micro-parcours
Tableau SOPHIE CAT	Recommandations liées aux réponses

## 2.3 Articulation sources → moteur



## 3. Fonctionnement du Moteur

### 3.1 Flux principal



### 3.2 Typologie des questions

Le moteur distingue 4 types de questions (règle Legacy : "1 question = 1 rôle") :

Type	Fonction moteur
<b>Scorante</b>	Incrémente le score de vulnérabilité
<b>Déclenchante</b>	Active un micro-parcours spécifique
<b>Critique directe</b>	Force une priorité niveau 1 ( $\leq 7$ jours)
<b>Descriptive</b>	Contexte, pas d'effet algorithmique

### 3.3 Mécanisme de scoring

Le moteur calcule un score par vulnérabilité :

$$\text{Score normalisé} = (\text{Score brut} / \text{Score max}) \times 20$$

Le score détermine la **temporalité** d'action, pas le contenu des recommandations.

### 3.4 Conditions Critiques Composites (CCC)

Les CCC détectent des **combinaisons** de réponses à risque via une logique booléenne :

Exemple CCC :

```
| SI (Question A = "Oui")      |
| ET (Question B = "Jamais")   |
| ALORS → Activer Micro-parcours X |
| + Priorité niveau 2         |
```

### 3.5 Micro-parcours

Le moteur gère 24 micro-parcours adaptatifs répartis sur 5 vulnérabilités :

#### V Préfixe Parcours

V1 R	R1, R2, R3, R4
V2 F	F1 → F6
V3 S	S1, S2, S3, S4
V4 M	M1 → M6
V5 A	A1, A2, A3, A4

Chaque parcours contient des ASR (Actions Structurantes de Référence) avec états de suivi.

### 3.6 Priorisation

Le moteur calcule automatiquement le niveau de priorité :

Niveau	Déclencheur	Délai max
1	Question critique directe $\leq$ 7 jours	
2	CCC activée	$\leq$ 15 jours
3	Score + déclenchantes	> 1 mois

---

## 4. Structure des Livrables

### 4.1 Organisation fichiers

```

monka/
├── QUESTIONNAIRE/
│   └── V[1-5]_*/base/
│       ├── questions.md      # Questions + options
│       ├── recommendations.md # Mappings recommandations
│       ├── ccc_recommendations.md
│       └── asr_definitions.md

├── SOURCES/extracted/
└── *.json          # Données structurées

└── DEMO/
    └── monka_simulator.html # Prototype fonctionnel

```

## 4.2 Prototype HTML

Fichier autonome contenant :

- Interface questionnaire
- Moteur de règles JavaScript
- Données V1 intégrées

## 5. Options d'Intégration

### Option A — Utilisation directe du HTML

Usage	Limite
-------	--------

Demos, tests fonctionnels Non modulaire

### Option B — Extraction du moteur JS

```

// Fonctions à extraire :
calculateScore(answers, vulnerability)
evaluateCCC(answers)
getRecommendations(answers)
getPriority(criticals, cccActive, score)

```

Usage	Effort
-------	--------

Intégration dans votre stack ~2-3 jours refactoring

### Option C — Import JSON vers votre DB

Usage	Effort
-------	--------

Contrôle total, votre infra Script d'import + logique à recoder

## 6. Prochaines Étapes

1. **Validation données** — Vérifier la véracité des mappings avec l'équipe clinique
2. **Choix option intégration** — A, B ou C selon vos priorités

**3. Extension V2-V5 — Même logique moteur, données à intégrer**

---

Rapport Technique Monka — Focus Moteur — 04/02/2026