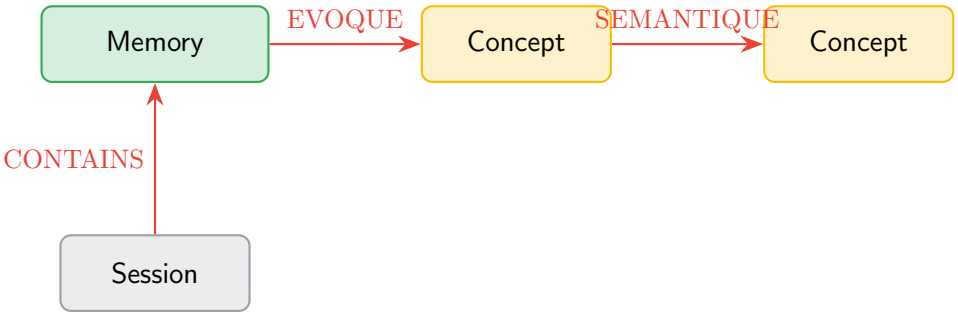
Structure Neo4j

MCEE - Modèle Complet d'Évaluation des États



Version 3.1
Architecture simplifiée - 24 émotions

Documentation technique
22 décembre 2025

Table des matières

1 Vue d'ensemble

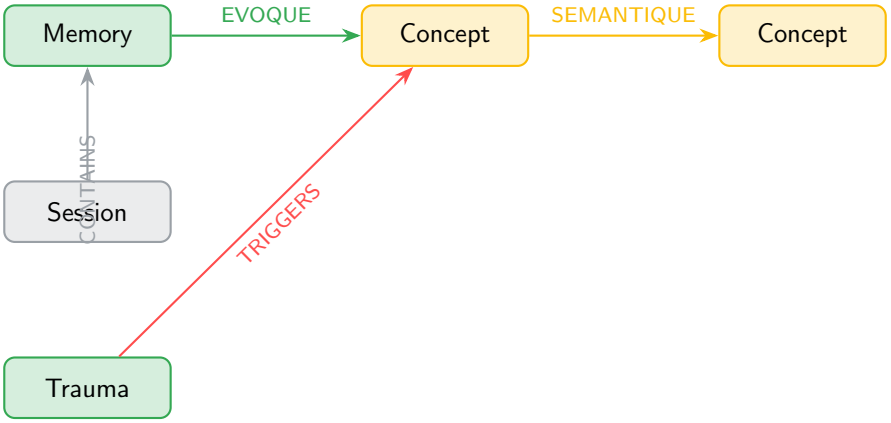
Le système utilise Neo4j comme base de données graphe pour stocker les mémoires émotionnelles, les concepts sémantiques et leurs relations.

1.1 Philosophie de conception

L'architecture v3.1 abandonne les patterns statiques (SERENITE, ANXIETE, etc.) au profit d'un système basé sur **24 émotions continues**. Cette approche permet :

- **Nuances émotionnelles** – La peur à 0.3 + anxiété à 0.4 vs terreur à 0.9 + horreur à 0.8
- **Combinaisons naturelles** – Joie + nostalgie + admiration simultanées
- **Évolution temporelle** – Suivi des changements par `sentence_id`

1.2 Architecture mémoire



2 Nœuds (Nodes)

2.1 Memory

Représente un souvenir/mémoire avec son contexte émotionnel complet.

Memory Node

```
(:Memory {
    id: String,                          -- Identifiant unique

    -- Emotions (24 dimensions)
    emotional_states: String,            -- JSON: {"sentence_id":
       [24 floats]}
    dominant: String,                    -- Emotion dominante
    intensity: Float,                    -- Intensite [0.0 - 1.0]
    valence: Float,                      -- Valence [-1.0 a 1.0]

    -- Metadonnees
    weight: Float,                       -- Poids/importance [0.0 -
       1.0]
    context: String,                     -- Texte du contexte
    category: String,                    -- Categorie semantique
    keywords: [String],                  -- Mots-cles extraits
```

```
    -- Classification memoire
    type: String,                    -- "MCT" | "MLT" |
       "Episodic" | "Semantic"
    consolidated: Boolean,           -- True si consolide en MLT
    active: Boolean,                 -- True si actif

    -- Statistiques
    activation_count: Integer,       -- Nombre de reactivations

    -- Timestamps
    created_at: DateTime,
    updated_at: DateTime,
    last_accessed: DateTime
})
```

2.1.1 Labels additionnels

— `:Trauma` – Pour les mémoires traumatiques (weight ≥ 0.3)
— `:MCT` – Mémoire à Court Terme
— `:MLT` – Mémoire à Long Terme
— `:Archived` – Mémoire archivée

2.1.2 Format emotional_states

Les états émotionnels sont stockés en JSON string avec le `sentence_id` comme clé :

```
{
    "1": [0.8, 0.24, 0.16, 0.0, 0.0, ...],
    "4": [0.1, 0.7, 0.21, 0.0, 0.0, ...],
    "7": [0.0, 0.0, 0.0, 0.9, 0.27, ...]
}
```

2.1.3 Les 24 émotions

Index	Émotion	Index	Émotion	Index	Émotion
0	Joie	8	Calme	16	Triomphe
1	Confiance	9	Intérêt	17	Admiration
2	Peur	10	Fascination	18	Adoration
3	Surprise	11	Confusion	19	Amusement
4	Tristesse	12	Ennui	20	Anxiété
5	Dégoût	13	Gêne	21	Émerveillement
6	Horreur	14	Excitation	22	Nostalgie
7	Douleur emp.	15	Soulagement	23	Satisfaction

Note:

Les 24 émotions permettent de capturer des nuances fines. Par exemple, "peur légère" = Peur(0.3) + Anxiété(0.4), tandis que "terreur" = Peur(0.9) + Horreur(0.8) + Tristesse(0.5).

2.2 Trauma

Extension de Memory pour les souvenirs traumatiques.

Trauma Node

```
(:Memory:Trauma {
    -- Herite de toutes les proprietes Memory
    id: String,
    emotional_states: String,
    ...

    -- Proprietes specifiques Trauma
    trauma: Boolean,                -- Toujours true
    triggers: [String],             -- Mots/concepts
       declencheurs
    protection_level: Float         -- Niveau de protection
       [0.0 - 1.0]

    -- Note: weight a un plancher (floor) de 0.3
})
```

2.3 Concept

Représente un concept sémantique extrait des mémoires.

Concept Node

```
(:Concept {
    name: String,                   -- Nom du concept
       (minuscules)

    -- Emotions accumulees
    emotional_states: String,       -- JSON: {"sentence_id":
       [24 floats]}

    -- References
    memory_ids: [String],           -- Liste des IDs de
       memoires liees

    -- Timestamps
    created_at: DateTime,
    updated_at: DateTime
})
```

Note:

Les `emotional_states` d'un Concept accumulent les états de toutes les mémoires qui l'évoquent, permettant une analyse de l'évolution émotionnelle du concept dans le temps.

2.4 Session

Représente une session d'interaction.

Session Node

```
(:Session:MCT {
    id: String,                          -- Identifiant unique

    -- Metriques emotionnelles
    stability: Float,                    -- Stabilite [0.0 - 1.0]
    volatility: Float,                   -- Volatilite [0.0 - 1.0]
    trend: String,                       -- "stable" | "ascending"
       | "descending"
    state_count: Integer,                -- Nombre d'etats
       enregistres

    -- Timestamps
    created_at: DateTime,
    updated_at: DateTime
})
```

2.5 EmotionalState

État émotionnel ponctuel (lié à une Session).

EmotionalState Node

```
(:EmotionalState {
    emotions: [Float],                   -- 24 valeurs d'emotions
    dominant: String,                    -- Emotion dominante
    intensity: Float,                    -- Intensite
    valence: Float,                      -- Valence
    timestamp: DateTime                  -- Moment de
       l'enregistrement
})
```

3 Relations

3.1 EVOQUE

Lie une mémoire aux concepts qu'elle évoque.

```
(:Memory)-[:EVOQUE]->(:Concept)
```

Propriétés : Aucune (relation simple)

3.2 SEMANTIQUE

Relation sémantique entre deux concepts.

```
(:Concept)-[:SEMANTIQUE {
    type: String,                    -- Type de relation
    count: Integer,                  -- Nombre d'occurrences
    memory_ids: [String],            -- Memoires source
    emotional_states: String         -- JSON des etats emotionnels
}]->(:Concept)
```

3.2.1 Types de relations sémantiques

Type	Description	Exemple
EST	Attribution/Identité	"chat" → "noir"
UTILISE	Action/Utilisation	"chat" → "dort"
LOCALISE	Localisation	"chat" → "canapé"
MODIFIE	Modification/Adverbe	"paisiblement" → "dort"
POSSEDE	Possession	"marie" → "chat"
CAUSE	Causalité	"pluie" → "tristesse"

3.3 ASSOCIE_A

Association générique entre Memory et Concept.

```
(:Memory)-[:ASSOCIE_A {
    strength: Float,                 -- Force de l'association [0.0
        - 1.0]
    created_at: DateTime
}]->(:Concept)
```

3.4 CONTAINS

Lie une session à ses états émotionnels.

```
(:Session)-[:CONTAINS]->(:EmotionalState)
```

3.5 TRIGGERS

Lie un trauma à ses concepts déclencheurs.

```
(:Trauma)-[:TRIGGERS]->(:Concept)
```

4 Index recommandés

```
-- Index primaires
CREATE INDEX memory_id IF NOT EXISTS FOR (m:Memory) ON (m.id);
CREATE INDEX concept_name IF NOT EXISTS FOR (c:Concept) ON
    (c.name);
CREATE INDEX session_id IF NOT EXISTS FOR (s:Session) ON (s.id);
```

```
-- Index de recherche
CREATE INDEX memory_dominant IF NOT EXISTS FOR (m:Memory) ON
    (m.dominant);
CREATE INDEX memory_type IF NOT EXISTS FOR (m:Memory) ON (m.type);
CREATE INDEX memory_weight IF NOT EXISTS FOR (m:Memory) ON
    (m.weight);

-- Index composites
CREATE INDEX memory_type_weight IF NOT EXISTS
    FOR (m:Memory) ON (m.type, m.weight);
```

5 Requêtes Cypher courantes

5.1 Créer une mémoire avec concepts

```
// 1. Creer la memoire
CREATE (m:Memory:MCT {
    id: $id,
    emotional_states: $emotional_states,  // JSON string
    dominant: $dominant,
    intensity: $intensity,
    valence: $valence,
    weight: $weight,
    type: 'MCT',
    created_at: datetime()
})

// 2. Creer/lier les concepts
MERGE (c:Concept {name: $concept_name})
ON CREATE SET c.created_at = datetime(), c.memory_ids = [$id]
ON MATCH SET c.memory_ids = c.memory_ids + $id
MERGE (m)-[:EVOQUE]->(c)
```

5.2 Rechercher par sentence_id

```
// Concepts contenant un sentence_id specifique
MATCH (c:Concept)
WHERE c.emotional_states IS NOT NULL
  AND c.emotional_states CONTAINS '"<sentence_id>":'
RETURN c
```

5.3 Trouver les mémoires similaires

```
MATCH (m:Memory)
WHERE m.intensity IS NOT NULL AND m.valence IS NOT NULL
WITH m,
     1 - abs(m.intensity - $query_intensity) AS intensity_sim,
     1 - abs(m.valence - $query_valence) AS valence_sim
```

```
WITH m, (intensity_sim + valence_sim) / 2 AS similarity
WHERE similarity >= $threshold
RETURN m ORDER BY similarity DESC LIMIT $limit
```

5.4 Consolider MCT vers MLT

```
MATCH (m:Memory)
WHERE (m.type = 'MCT' OR m.type IS NULL)
  AND (m.consolidated IS NULL OR m.consolidated = false)
  AND m.weight >= $threshold
SET m.type = 'MLT', m.consolidated = true, m:MLT
REMOVE m:MCT
RETURN count(m) AS consolidated
```

6 Conventions importantes

6.1 Sérialisation JSON

Important:

Les `emotional_states` sont **toujours** stockés comme JSON string dans Neo4j.

```
# Python -> Neo4j
json.dumps({str(k): v for k, v in emotional_states.items()})

# Neo4j -> Python
json.loads(emotional_states_string)
```

6.2 Clés de sentence_id

Les clés dans `emotional_states` sont **toujours des strings** :

```
{"1": [...], "42": [...]}   // Correct
{1: [...], 42: [...]}       // Incorrect
```

6.3 Recherche dans JSON

Utiliser `CONTAINS` pour rechercher dans les JSON strings :

```
WHERE c.emotional_states CONTAINS '"42":'  // Cherche sentence_id
    42
```

6.4 Syntaxe Neo4j 5+

Ancienne syntaxe	Neo4j 5+
`NOT EXISTS(m.prop)`	`m.prop IS NULL`
`EXISTS(m.prop)`	`m.prop IS NOT NULL`

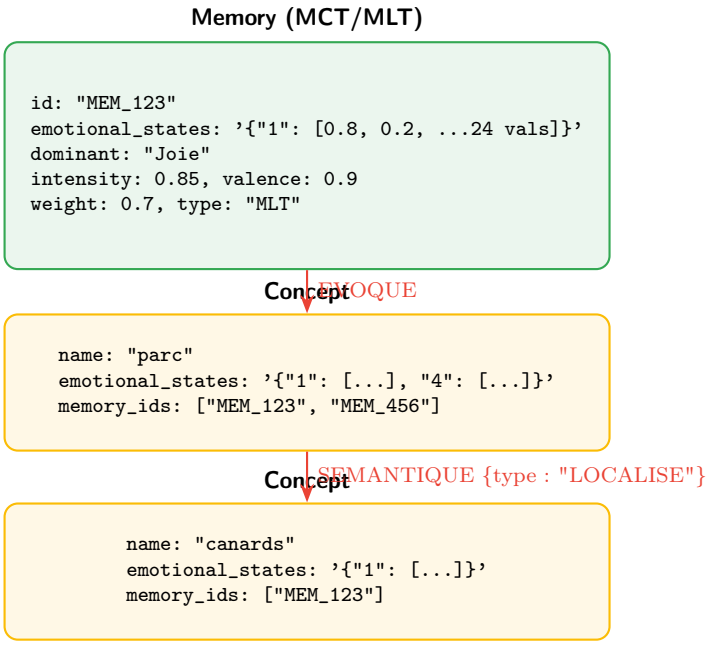
7 Analyse émotionnelle

Lors de la récupération d'un concept avec `get_concept`, une analyse est effectuée :

```
{
    "name": "parc",
    "memory_ids": ["MEM_1", "MEM_2"],
    "emotional_states": {"1": [...], "4": [...]},
    "sentence_ids": ["1", "4"],
    "emotional_analysis": {
        "dominant_emotion": "Joie",
        "average_valence": 0.73,
        "stability": 0.96,
        "trajectory": "stable",
        "trauma_score": 0.12
    }
}
```

Champ	Description
`dominant_emotion`	Émotion la plus fréquente parmi les 24
`average_valence`	Valence moyenne [–1.0, 1.0]
`stability`	Stabilité émotionnelle [0.0, 1.0]
`trajectory`	`"stable"` | `"ascending"` | `"descending"`
`trauma_score`	Score de trauma [0.0, 1.0]

8 Schéma visuel complet



9 Comparaison avec l'ancienne architecture

Élément	v3.0 (Patterns)	v3.1 (24 émotions)
Patterns statiques	Oui (8 patterns)	Non
Transitions	TRANSITION_TO	Implicites
Nuances émotionnelles	Limitées	24 dimensions
Combinaisons	Pattern unique	Multi-émotions
Complexité	Plus élevée	Simplifiée
Nœuds	Memory, Concept, Pattern, Session	Memory, Concept, Session