Memory Strategy x Retrieval

Context Memory Tiers

- **RAM:** Data tagged with 'contexts_active' or part of collection. This data will always be injected or at least prioritized when retrieving data
- **Context Store:** Long-term data that can be retrieved using RAG to find most relevant info from a huge store of data

---

Storage Granularity

- **Elements:** Predefined, system-generated. Can be assigned/modified manually
- **Sub-Elements:** Default values + user values; system-generated. Can be assigned/modified manually
- **Shards:** Not defined, system-generated. Used for salience scoring

---

Retrieval Strategies

RAG pipeline
- embeddings
- shards

Manual Filtering
- on tags
- for temporal (RAM)
- chat sessions

Potential Issues / Improvements

- **Shard Granularity Drift**
  - ↳ shard might become too noisy, fragmented, or redundant.
  - ↳ Define cohesion rules: → temporal (within last X hours)
    - → shared sub-elements > 60%
    - → semantic similarity > .75 cosine
  - ↳ Give each shard a human-readable title using LLM summarization
- **Element/sub-element lock in Risk:** If elements are too rigid, will create bottlenecks for future contexts. Instead of a Tree, think of an expandable Graph. Also, sub elements could be overlapping. (Ex. 'burnout' on • Career • relationships)
- **NEED** versioning or auditing system for ontology shifts
  - ↳ Add 'version', 'history_log', and 'last_modified_by' fields