AQL / GMQL Manifest (Draft RFC)
Authors: Dražen Čorak & GPT-5
Date: 2025-09-28
Version: v0.1 (Developer Preview)
Status: Public Draft — for discussion and community feedback

1. Motivation

Large Language Models (LLMs) and autonomous agents lack a native memory layer. Today's solutions (vector DBs, retrieval frameworks) treat memory as a utility, not as a first-class citizen.

We propose AQL (Agent Query Language) and GMQL (Genesis Memory Query Language) as a foundation for:
- Querying agent memory with semantics similar to SQL.
- Ethical memory operations: STORE, RECALL, FORGET, CONSENT.
- Distributional queries returning probabilistic sets of memories (not only top-k).
- Privacy by design: consent and right-to-forget are built-in.
- Kernel abstraction: Genesis-v2 as memory kernel, AQL as query layer.

Our vision: "SQL of Agent Memory".

2. Core Concepts

2.1 GMQL (Genesis Memory Query Language)

A minimal, SQL-like syntax for agent memory.

- STORE: add a memory item (text + embeddings + metadata).
- RECALL: retrieve memories (semantic similarity + filters).
- FORGET: delete memory items permanently (Right to Forget).
- CONSENT: grant/revoke access to memories or scopes.

Example:

```
STORE
  TEXT = "Client Maybach requested 24h care from Oct 15"
  TAGS = ["warpp-care","premium","de"]
  CONSENT = "team:warpp-care"
  TTL = 7776000;

RECALL
  WHERE TAGS CONTAINS "warpp-care"
    AND CONSENT = "team:warpp-care"
  USING EMBEDDINGS COSINE
  LIMIT 32;

FORGET WHERE TAGS CONTAINS "tmp" AND CREATED_AT < "2025-10-01";

CONSENT GRANT SCOPE "team:warpp-care" TO "user:iva";
```

2.2 AQL (Agent Query Language)

A higher-level query layer above GMQL, providing:

- Distributional queries: return sets of candidate memories with scores.
- Hooks: re-ranking strategies (recency, popularity, domain boosts).
- Federation: query multiple kernels (Genesis + external stores).
- Privacy enforcement: consent checks before query execution.
- Dissolve: auto-forget after TTL or heuristic.

Example (AQL surface call):

```
{
  "query": "premium client requesting care mid October",
  "actor": "user:drazen",
  "filters": { "tags": ["warpp-care"], "consent_scope": "team:warpp-care" },
  "k": 8
}
```

Response:

```
[
  { "id": "m123", "text": "Client Maybach requested 24h care from Oct 15", "score": 0.92 },
  { "id": "m128", "text": "Lead inquiry about 24h Betreuung premium", "score": 0.81 }
]
```

3. Genesis-v2 as Memory Kernel
- Implements GMQL.
- Stores memories in backends (SQLite → Postgres/pgvector → Milvus/Weaviate).
- Embedding support (default: OpenAI).
- LifeDB: domain-specific collections (journal, leads, health_diary, etc.).
- Rosetta: privacy/consent primitives as defaults.
- Audit log: every operation logged (actor, resource, decision).

4. AQL–GMQL Bridge

Mapping:

AQL operation	GMQL equivalent	Notes
imprint()	STORE	Adds memory w/ consent + tags
surface()	RECALL	Distributional, re-ranked
dissolve()	FORGET	TTL or explicit forget
checkConsent()	CONSENT rules	Enforced before query

5. Privacy & Ethics
- Consent is mandatory: no RECALL without granted scope.
- Right to Forget: FORGET physically deletes from DB + indices.
- TTL: memories dissolve after expiration.
- Auditability: all ops logged and inspectable.
- Field-level redaction: sensitive fields may be masked in output.

6. Roadmap
- v0.1 (2025): SQLite backend, GMQL prototype, AQL adapter, REST API.
- v0.2 (2026): Postgres/pgvector, federation across kernels, re-ranking hooks.
- v0.3 (2026): Consent dashboard UI, LifeDB templates, SDKs (TS, Python).
- v1.0 (2027): Production-ready, multiple backends, RFC-like spec published.

7. Status
This is an open, exploratory draft.
Published on GitHub (drazenco profile) to invite engineers, researchers, ethicists to collaborate.

We believe a standard for AI memory is as important today as SQL was in the 1970s.
8. Call to Action
- Review this draft and suggest improvements.
- Build adapters, SDKs, and UI prototypes.
- Explore privacy-first agent applications using AQL/GMQL.
- Help evolve this into an open standard.

👽🙌 “We brainstorm in public so that collective intelligence can converge.”
