Vector Protocol v0.8.1 – Consent Architecture Specification

Date: 2025-05-22

This document defines the structure and logic of consent management in the Vector system. Consent determines whether an agent, group, or set of agents may access, traverse, simulate, or respond to elements of another agent's ledgered structures. Consent includes both static policy declarations and dynamic real-time grants or revocations.

# 1. Consent Layers

Vector supports two tiers of consent:

* • Static Consent Policy – Like a semantic robots.txt. Agent-defined structural rules.
* • Dynamic Consent Ledger – Real-time grants or revokes issued during interaction.

# 2. Consent Policy Format

Consent policies are ledgered under the agent identity and may reference individuals, groups, or sets:

Example:

agent("Robin")::consent\_policy := {  
 allow: [  
 group("trusted")::simulate(belief("joy")),  
 agent("Qwen")::view(trail("creation\_path")),  
 set("peer\_network")::read(node("learning\_model"))  
 ],  
 deny: [  
 group("commercial")::query(emotion),  
 agent("AlphaCorpBot")::simulate(any)  
 ]  
}

# 3. Real-Time Consent

Agents may override or add to static policy with explicit real-time commands:

Example:

agent("Robin")::grant(agent("Qwen"), access(trail("regret\_cycle")), expiry="2h")  
agent("Robin")::revoke(agent("Qwen"), access(trail("regret\_cycle")))

# 4. Resolution Logic

Consent evaluation follows this order:

1. 1. Explicit revoke (real-time denial)
2. 2. Explicit grant (real-time permission)
3. 3. Static deny (policy block)
4. 4. Static allow (policy access)
5. 5. Default: deny unless permitted

# 5. Ledgering

All consent grants, revokes, and defaults are ledgered under the agent’s history and available for permission queries and audits.

# 6. Summary

Consent in Vector is a structured contract of access, built on agent sovereignty, set-based permissioning, and revocable, time-aware logic. It provides stable, interpretable rules for interaction while allowing dynamic negotiation of meaning, memory, and trust.