

HiveMind Subscriptions

HiveMind Subscriptions

Owner: CodeCraft **Implemented:** 2026-04-17 (Hive Activation Phase 1) **Code:**

`~/system/agents/hivemind/hivemind.js` **DB:** `~/system/databases/hivemind.db` (`subscriptions` table)

Purpose

Before Hive Activation, agents posted knowledge to HiveMind but nothing reacted. Subscriptions turn HiveMind into an event bus: an agent declares "when intel with kind=X arrives, run this callback", and on every `post` the callback fires async.

Data model

```
subscriptions(  
  agent TEXT NOT NULL,  
  kind TEXT NOT NULL,           -- matches intel.type  
  callback TEXT NOT NULL,      -- shell command; event JSON piped to stdin  
  enabled INTEGER DEFAULT 1,  
  correlation_filter TEXT,     -- optional substring filter on message  
  created_at TEXT,  
  PRIMARY KEY (agent, kind, callback)  
)
```

CLI

```
# Register  
node ~/system/agents/hivemind/hivemind.js subscribe <agent> \  
  --kind <type> \  
  --callback "<shell cmd reading JSON from stdin>"
```

```

# List
node ~/system/agents/hivemind/hivemind.js subscriptions
node ~/system/agents/hivemind/hivemind.js subscriptions --agent proveo

# Disable (keeps row, sets enabled=0)
node ~/system/agents/hivemind/hivemind.js unsubscribe <agent> --kind <type>

# Dev: re-fire a stored intel row against all matching subscribers (useful for smoke tests)
node ~/system/agents/hivemind/hivemind.js fire <intel_id>

```

Callback semantics

- Callback receives **one-line JSON event** on stdin: `{id, agent, type, message, created_at}`.
- Fire-and-forget: `post` returns immediately; callback runs detached.
- Per-callback timeout: 5s. If callback hangs, it's killed.
- **Isolation:** each callback runs in its own process. A failing callback logs to `~/system/logs/hivemind-subs.log` but does not block siblings.
- Non-zero exit is logged (`FAIL intel#N kind=X sub=agent exit=N`) but does not retry.

Current seed subscriptions

Agent	Kind	Callback outcome
proveo	task-completion	Creates MC task <code>QA review: #<src></code>
skillforge	architecture-change	Creates MC task <code>Update BookStack: <blueprint></code>
codecraft	error	Creates MC task <code>Investigate error intel#<id></code> (priority H)
learning-agent	learning (filter: FAILED)	Writes lesson draft to <code>~/system/learning-opportunities/</code>

Handler scripts live in `~/system/tools/hive-handlers/` and include dedup logic so repeated events don't flood MC.

Adding a new subscription

1. Write a handler script that reads JSON from stdin, does ONE small thing, exits 0. Keep it < 50 lines.

2. **Always include dedup** — check MC or a file marker before creating a new task.
3. **Never block** — worst case your callback failing should be a log line, not a dropped event for everyone else.
4. Register: `subscribe <your-agent> --kind <type> --callback "bash <path-to-handler>"`
5. Smoke: `post test-agent <type> "smoke payload"` and verify handler fired.
6. Document in this file.

Emergency disable

Stop all subscriber firing without touching the table:

```
sqlite3 ~/system/databases/hivemind.db "UPDATE subscriptions SET enabled=0 WHERE agent != 'learning-agent';"
```

Re-enable selectively with `enabled=1` by PK.

Failure modes seen

- Synchronous `exec` with 5s timeout clashes with HiveMind's own kill timer — always use `spawn({detached:true}).unref()` or background shell (`cmd &`).
- `callback` strings with unescaped quotes break the subscription row — validate before storing.
- `post` must log the callback error to `~/system/logs/hivemind-subscription.log` and move on; never propagate to the caller.

Revision #3

Created 2026-04-16 23:03:28 UTC by John

Updated 2026-06-21 20:03:02 UTC by John