Hearthlink Plugin Ecosystem Design Document

Purpose\ Define a future-resilient, agile plugin integration framework within the Hearthlink ecosystem. This system must support MakeItHeavy, SuperClaude, and future plugin technologies without requiring architectural overhauls, while leveraging existing modules (Synapse, Core, Mimic, Vault, Alice, Sentry, Alden).

---

1. Architectural Foundation

- **Synapse**: Universal protocol translator and event orchestrator. Ingests tasks from API, CLI, WebSocket, REST (via Cloudflare Workers), and routes to Core.
- **Core**: Communications and knowledge hub. Hosts Project Command, manages conference-mode inter-agent sessions, and delegates to execution modules.
- **Mimic** *(Core Module)*: Adaptive persona constructor. Dynamically builds purpose-fit agent personas based on task context and CBP profile.
- **MakeItHeavy / SuperClaude**: Plugin-style execution modules. Handle multi-agent strategies (fixed personas or task-split clones).
- **Vault**: Knowledge + Secret store. Agents read/write task state, context, and credentials.
- **Alice**: Chat UI and sentiment profiling (CBP) source. Context available to agents via CBP APIs.
- **Sentry**: Monitoring, fault detection, policy enforcement.

---

2. Plugin Requirements and Manifest Schema

All plugins must register using a `plugin_manifest.yaml`. Example:

```yaml
plugin_name: MakeItHeavy
version: 1.0.3
entrypoint: ./main.py
roles_supported:
  - researcher
  - implementer
capabilities:
  - subquestion_generation
  - parallel_dispatch
  - synthesis
fitness:
  codegen.simple: 0.8
  research.topic: 0.9
schema_version: 1.0
```

---

3. PluginManager (Core Service)

Responsibilities:

- Discover `/plugins/**/plugin_manifest.yaml`
- Validate manifest schema
- Register plugin + capabilities
- Provide plugin registry API to Synapse and Core

---

4. Task Routing Process

1. Synapse receives task from any protocol.
2. Synapse wraps task in Task Envelope:

```
{
  "task_id": "uuid",
  "intent": "code_refactor",
  "preferred_plugins": ["SuperClaude"],
  "context": { ... },
  "cbp": {...},
  "memory": {"recall": true},
  "input_format": "cli"
}
```

1. Core evaluates:

2. Active agents?

3. Should this go to Project Command?

4. Best plugin match (capability + fitness)?

5. Routes to plugin or composite path (e.g. Mimic creates agents → agents run via SuperClaude).

6. Vault enables memory/context read/write per task.
7. Alice CBP and CBP-aware modules can inject affective or contextual guidance.

---

5. Extending UI Templates

- Each plugin can optionally expose `ui_template.json`, used by Alden UI to rapidly configure its panel display:

```
{
  "layout": "split",
  "components": [
    { "type": "AgentView", "role": "researcher" },
    { "type": "Synthesizer" },
    { "type": "CBPInsights" }
  ],
  "editable": true,
  "supports_dark_mode": true,
  "default_theme": "auto",
  "min_resolution": "1024x768"
}
```

- UI templates will be interpreted by Alden's panel manager to determine which panels to load, layout behavior, default interaction bindings, and accessibility overrides.
- A fallback default template will be generated automatically if the plugin omits one.
- If `editable: true`, UI elements may be reordered or resized manually by the user.

---

6. Plugin Execution Models

Plugin	Execution Style	Agent Source
MakeItHeavy	Prompt-split + parallel	Clones (same base LLM)
SuperClaude	Fixed persona execution	Defined prompt classes

Note: Mimic is not a plugin. It is a Core module that dynamically generates bespoke agents and may delegate execution to compatible plugin modules.

---

7. Observability and Telemetry

- All plugins must emit task start/stop/error via standard `/v1/events` Synapse endpoint.
- Metrics auto-ingested into Sentry.
- Each execution tagged with `plugin_name`, `agent_id`, `task_id`, `session_id`.
- Health states for each plugin reported via heartbeat or periodic `/status` pings.
- Heartbeat schema:

```
{
  "plugin_name": "SuperClaude",
  "status": "Available",
  "uptime_seconds": 43200,
  "last_task_id": "8b4f...",
  "error_rate": 0.01,
  "last_heartbeat": "2025-07-17T14:03:00Z"
}
```

- Plugin health states exposed in Alden UI:

- **Available** (green)
- **Degraded** (yellow)
- **Failed** (red)
- **Incompatible** (gray)
- Alden UI will display real-time plugin state via a **Plugin Panel Manager**, with the following components:
- Plugin Status Grid (sortable table by name/state/load/last used)
- Plugin Detail View (log output, error stack, recent tasks)
- Health Timeline Graph (heartbeat history)
- Enable/Disable toggle with confirmation

---

8. Hot Reload + Adaptation

- Synapse watches plugin directories for manifest or logic changes.
- PluginManager revalidates schema and updates registry.
- Core rebalances tasks to new plugin versions if compatible.

---

9. Fallback and Isolation

- Each plugin runs in an isolated subprocess or container.
- If plugin fails: Core catches error, logs to Sentry, reassigns task to fallback plugin.
- Vault logs all plugin invocations for audit.

---

10. Future Roadmap Considerations

- Auto-benchmarking plugins over time to adjust fitness scores.
- Plugin capability negotiation (e.g. "can you handle streaming audio?")
- Memory profile inheritance (e.g. MakeItHeavy agents access same Vault trace via shared ID)
- Support for multimodal plugins (image, video, audio)
- Plugin Store: UI panel in Alden to enable/disable/update plugins with health state. *(Targeting v10+)*

End of Document