AGRM + MDHG + Cmplx-Core System Architecture Overview

1. AGRM Framework (TSP Logic Core)

Purpose:

Massively scalable TSP solver using Golden Ratio sweeps, complexity-aware pruning, and midpoint-locked zone traversal.

Key Modules:

- `agrm_core.py`, `agrm_core_loop.py`: Control logic
- `agrm_path_engine.py`, `agrm_pathbuilder_dual.py`: Pathing
- `sweep_scanner.py`: GR-based logical sweeps
- `salesman_and_evaluator.py`: Reroute validator
- `agrm_zone_density.py`, `agrm_distance_cap.py`: Node zone logic

Execution Steps:

1. Sweep 2. Classify 3. Midpoint unlock 4. Build paths

5. Validate with Salesman 6. Export

2. MDHG Hash Table

Purpose:

Advanced hash system using spatial partitioning and access prediction.

Key Module:

- `mdhg_hash (1).py`

Optimizations:

- Hierarchical layout: Building Floor Room
- Hot key clustering
- Outperforms Python dict under structured loads

3. CLI & Utility Tools

- CLI Launchers: `agrm_cli_launcher.py`, `agrm_cli_allrun.py`
- Profiling: `agrm_profiler_diagnostics.py`, `Hash_testsuite.py`
- Export: `agrm_results_export.py`
- Configuration: `.env`, `.yaml`

4. Documentation Classification

Type	Files
Design Docs	doc_4, 7, 8, 1619
Benchmarks	doc_5, 6
Test Plans	doc_1014
CLI Instructions	doc_0, 1, 9
General Notes	doc_2, 3, 15, 20

System Flow Diagram

```
[TSP Nodes] -> [Sweep Scanner] -> [AGRM Core Loop]
                                  |
                +-----------------+---------------+
                |                                 |
        [Path Builder]                    [Zone Classifier]
                |                                 |
        [Midpoint Unlock]                 [Complexity Modulator]
                |                                 |
                +--> [Salesman Evaluator] --> [Result Export]
```