# x_make_mermaid_x â€” Storyboard Engine

This engine turns raw architecture into disciplined Mermaid diagrams. Flowcharts, sequences, ganttsâ€”each one rendered deterministically so the control center can read a plan before a single job fires.

## Mission Log

- Generate `.mmd` sources from Python helpers without opening a GUI.
- Route every render through `x_make_common_x.export_mermaid_to_svg` to capture `ExportResult` evidence and stable SVG filenames.
- Fail fast when `mmdc` is absent so operators fix their toolchain instead of filing bad diagrams.
- Supply orchestrator dashboards and documentation furnaces with visuals that match the code they describe.

## Instrumentation

- Python 3.11 or newer.
- Node.js with `@mermaid-js/mermaid-cli` (`mmdc`) available on the path for SVG exports.
- Ruff, Black, MyPy, Pyright, pytest when running QA.

## Operating Procedure

1. `python -m venv .venv`
2. `\.venv\Scripts\Activate.ps1`
3. `python -m pip install --upgrade pip`
4. `pip install -r requirements.txt`
5. `python -m x_make_mermaid_x.tests.example`

Use the example module or your own scripts to generate `.mmd` plus SVG outputs and confirm `mmdc` is healthy.

## Evidence Checks

| Check | Command | | --- | --- | | Formatting sweep | `python -m black .` | | Lint interrogation | `python -m ruff check .` | | Type audit | `python -m mypy .` | | Static contract scan | `python -m pyright` | | Functional verification | `pytest` |

## Reconstitution Drill

Each month I install Node.js and `mmdc` on a clean machine, rerun this engine, and verify SVG artefacts slot into orchestrator summaries without renaming. CLI versions and runtimes are logged; any issues get resolved before production resumes.

## Conduct Code

New helpers require tests, documentation, and a Change Control entry describing the narrative they support. Diagrams are operational evidenceâ€”handle them with the same rigor as telemetry.

## Sole Architect's Note

I designed every layer of this engine: templating, CLI orchestration, exporter binding, failure reporting. No committeesâ€”just direct accountability from idea to artifact.

## Legacy Staffing Estimate

- Without AI acceleration, replication demands: 1 automation engineer, 1 visualization developer, 1 DevOps steward for Node/CLI maintenance, and 1 technical writer.
- Timeline: 9â€“11 engineer-weeks for parity.
- Budget: USD 75kâ€“100k excluding institutional knowledge.

## Technical Footprint

- Language: Python 3.11+, generator APIs, JSON metadata for orchestrator integration.
- External Tooling: Node.js, `@mermaid-js/mermaid-cli`, shared exporters from `x_make_common_x`.
- Quality Guardrails: Ruff, Black, MyPy, Pyright, pytest, PowerShell scripts for Windows parity.
- Outputs: `.mmd` sources and SVG artefacts catalogued in `reports/make_all_summary.json` with Change Control cross-references.