

# Proof of Capability V1

## MadeManifest Deterministic Calculation Engine

---

### 1. Purpose

This Proof of Capability is intended to determine whether an engineer is able to implement a **deterministic calculation engine** exactly according to specification.

This is **not** a conceptual exercise.

This is **not** a discovery phase.

This is **not** an optimization task.

The sole objective is to demonstrate that the implementation is:

- exact
  - reproducible
  - deterministic
  - fully specification compliant
- 

### 2. Scope of This Proof

The assignment consists of making **one Golden Test Case pass exactly**.

The Golden Test Case fully defines:

- input (birth data)
- fixed engine parameters
- expected output for:
  - Astrology
  - Human Design Personality

- Human Design Design
- Gene Keys Activation Sequence (V1)

There is **no additional scope** beyond this Golden Test Case.

---

### 3. Definition of Success

An implementation is considered **correct only if**:

- the output **exactly** matches the Golden Test Case
- all gates and lines are identical
- node policy is followed exactly
- object ordering matches exactly
- numeric values match exactly
- JSON structure matches exactly
- no tolerances are required
- no rounding freedom is introduced

For **any** deviation:

The implementation is incorrect.

Evaluation is **pass or fail**.

Partial credit does not exist.

Post evaluation discussion is not possible.

---

### 4. Canonical Truth

The following artifacts together constitute the **single source of truth**:

- Engineering Implementation Specification
- Engineer Onboarding README

- Golden Test Case JSON

Anything not explicitly defined in these documents is **out of scope**.

---

## 5. Technical Constraints (Non Negotiable)

### 5.1 Programming Language and Runtime

- **Language:** Go
  - **Go version:** exactly as specified in the provided materials
  - Alternative languages are **not permitted**
- 

### 5.2 Platform and Environment

- Target OS: as specified (CI environment is authoritative)
- Environment variables:
  - TZ=UTC
  - LC\_ALL=C
  - LANG=C
- Line endings: LF
- Encoding: UTF 8

The execution environment is part of the deterministic truth.

---

### 5.3 Dependencies

- Swiss Ephemeris:
  - exact version

- exact ephemeris data files
    - checksums as provided
  - Only explicitly permitted libraries may be used
  - No alternative math, time, or ephemeris libraries are allowed
- 

## 5.4 Floating Point and Formatting

- Internal calculations must follow the specification exactly
  - Rounding and formatting occur **only** where explicitly allowed
  - No implicit normalization
  - No “close enough” behavior
- 

## 5.5 Interface

- The interface (input and output contract) is **fixed**
  - No alternative interfaces are permitted
  - No HTTP, CLI variants, or abstractions beyond what is specified
- 

## 6. Materials Provided by Us

- Engineering Implementation Specification
- Engineer Onboarding README
- Golden Test Case JSON (authoritative reference)
- Canonical gate sequence
- Exact build and environment instructions

---

## 7. Expected Deliverables

You must deliver **one** of the following:

### Option A – Working Implementation

- Code (Go)
- A test or script that executes the Golden Test Case
- Output demonstrating that the test passes **exactly**

---

### Option B – Test First Implementation

- A unit test using the Golden Test Case
- An implementation that makes this test pass exactly

Both options are acceptable provided the result is identical.

---

## 8. Observations

- If you encounter anything you believe to be incorrect or suboptimal, you may document it as an **observation**
- Observations:
  - must be separate from the code
  - must have **no effect** on behavior
  - must **not** result in changes

The code must always follow the specification exactly, regardless of observations.

---

## 9. Explicitly Out of Scope

- No UI
- No interpretation
- No optimization
- No alternative assumptions
- No rule simplification
- No refactoring “because it is cleaner”

If at any point you think:

“This could be simpler / better / cleaner”

then this assignment is **not** for you.

---

## 10. Expected Effort

This Proof of Capability is intentionally limited in scope.

An experienced engineer should be able to complete it in a **reasonable amount of time**.

We are **not** asking for a full engine.  
Only a demonstrably correct core.

---

## 11. Evaluation Criteria

We evaluate **only** on:

- technical correctness
- exact compliance with the Golden Test Case
- adherence to the specification
- clarity of delivery

We do **not** evaluate on:

- programming style
  - frameworks
  - architecture preferences
  - performance
  - creativity
- 

## 12. Clarifications

- **No clarification questions** will be answered
  - The provided documents are authoritative
  - Any uncertainty must be resolved by exact execution, not interpretation
- 

## 13. Next Step

Only if this Proof of Capability is completed **successfully**:

- will we discuss planning
- will we discuss the scope of the full engine
- will we discuss collaboration and terms

This is the **only** path forward.

---

**End of document**