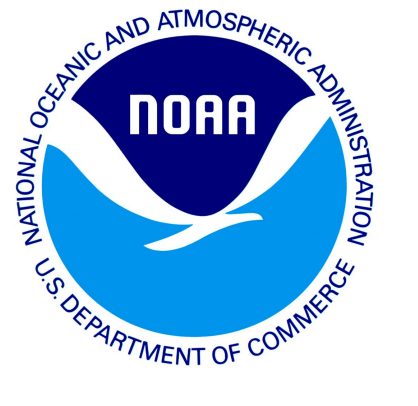




Cognitive Amplification in Satellite Science Ops



Matthew Odle

Cooperative Institute for Meteorological Satellite Studies,
Space Science and Engineering Center, University of Wisconsin - Madison

Introduction

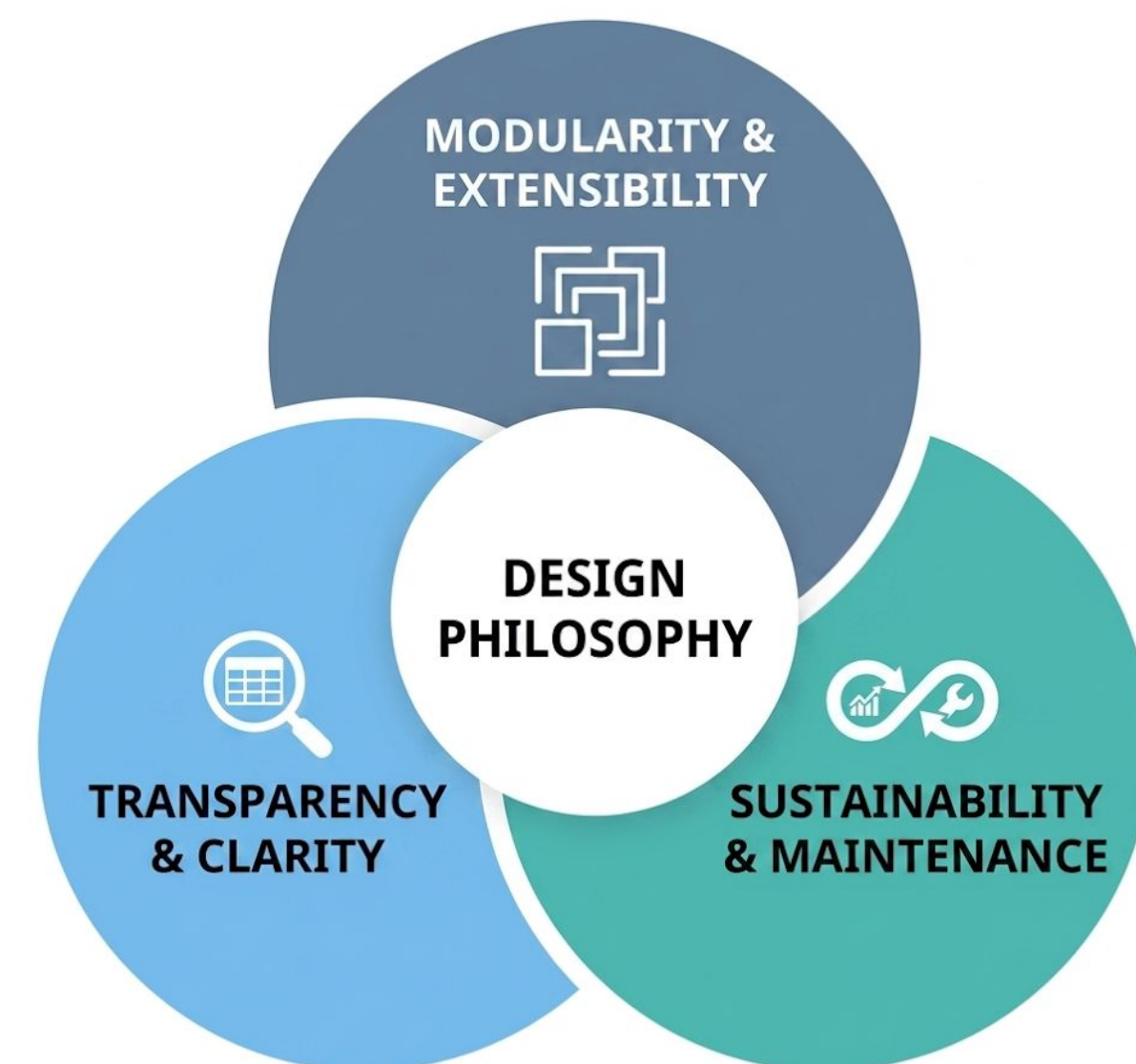
The **CSPP Build Orchestrator** is a case study for "Cognitive Amplification." With AI, everyone can be a programmer. The goal is to avoid "Cognitive Surrender" through structured development loops and tooling that prioritizes human maintainability.

Golden Rule

If the output is **unclear**, it is **broken**.
Ask for simplification.

Philosophy

Tools should be **Modular, Secure, Consistent, Readable, and Scalable**



This design philosophy borrows from the DevOps principles of:

Configuration as Code, Continuous Integration, and Discoverability

Build Orchestrator - Modernizing Legacy Builds

Purpose

The **CSPP Build Orchestrator** ensures operational reliability through total transparency and human-readable analysis.

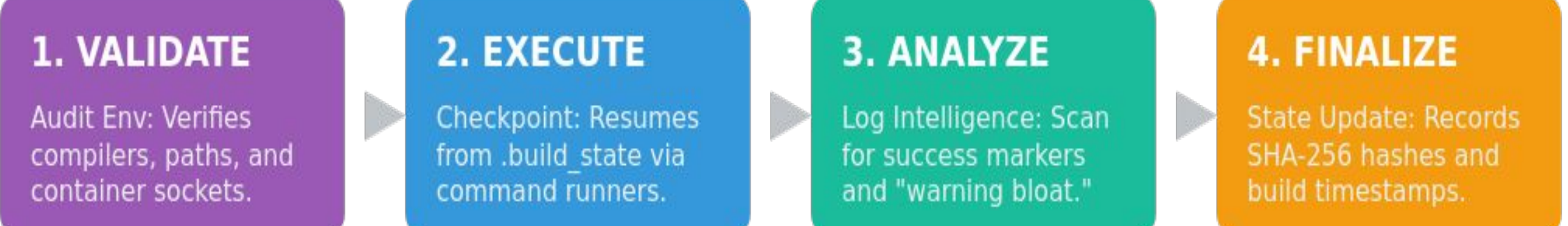
- **Custom Architecture:** Uses a stripped-down pipeline and centralized **app_config.yaml** to manage complex science builds.
- **Automated Rigor:** Pre-build environment audits and intelligent log analysis streamline post-run diagnostics.
- **The Result:** A stable, repeatable framework that transforms rapid AI development into mission-critical software.

Usage and Behavior

```
# Build Stepping
cspp_build --config build.yaml build --step
NEXT: [setup_0] <TARGET_STEP>

I 2026-04-24 19:22:54,514 -- build_env:
/path/to/build/env
    Last Success: 2026-04-24 19:19:09
    Last Attempt: 2026-04-24 19:19:09
(SUCCESS)

# Project Validation
cspp_build --config build.yaml --validate
[OK] BUILD_ENV
    Path: /path/to/repo
    Branch: test_ (Match: test_: Y)
VALIDATION PASSED: Build system ready.
```



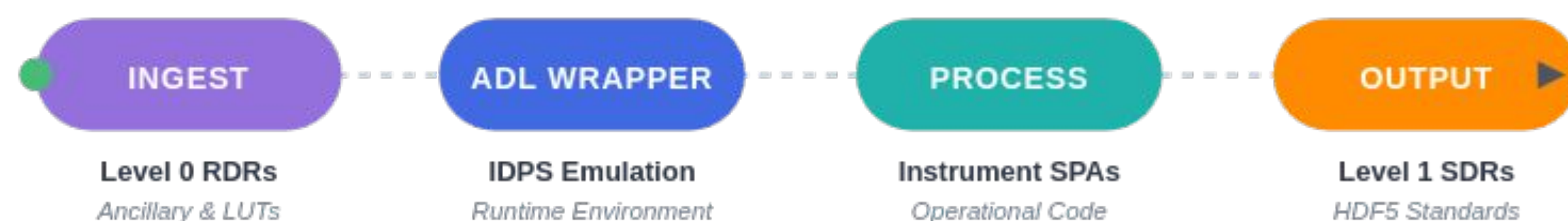
CSPP SDR and ADL Archaeology with Gemini

The SDR Database

> Describe the purpose of the database in the CIMMS CSPP SDR package.

ADL - The Algorithm Development Library

> Describe the ADL software used in the CIMMS CSPP SDR satellite data processing package.



Feature	SDR Database (System Brain)	ADL (Infrastructure Emulation)
Primary Goal	Synchronizes raw satellite streams with complex calibration parameters.	Enables execution of official NOAA ground station C++ and Fortran code.
Operational Parity	Local mirror of NOAA/JPSS auxiliary data for "Direct Broadcast" consistency.	Guarantees bit-for-bit consistency with official archives (VIIRS, CrIS, ATMS).
Data Management	Matches fragmented RDRs with metadata for zero-loss granule aggregation.	Manages the technical transition from raw data (RDR) to geolocated products (SDR).
Infrastructure	Automatically fetches and indexes time-sensitive Look-Up Tables (LUTs).	Acts as a "wrapper" that makes science algorithms independent of local hardware.
Significance	Turns chaotic radio signals into a structured, science-ready archive in real-time.	Provides a transparent "ground-segment-in-a-box" for the global community.

Digital Archaeology - Extracting Context

Sanitize

Sanitize sensitive, domain, and other proprietary data in your code before feeding it to an LLM

```
scrub --config mapping.yaml --file secret_code.py
INFO: Report: output/secret_code_report.txt
INFO: Cleaned file: output/secret_code_cleaned.py
# cat output/secret_code_report.txt
Line 3 | REPLACED | ***** -> New: AUTHOR_NAME
Line 5 | REPLACED | ***** -> New: 0.0.0.0
Line 9 | REPLACED | ***** -> New: 0.0.0.0
Line 13: WARNING: Sensitive pattern: ****://hello.com
```

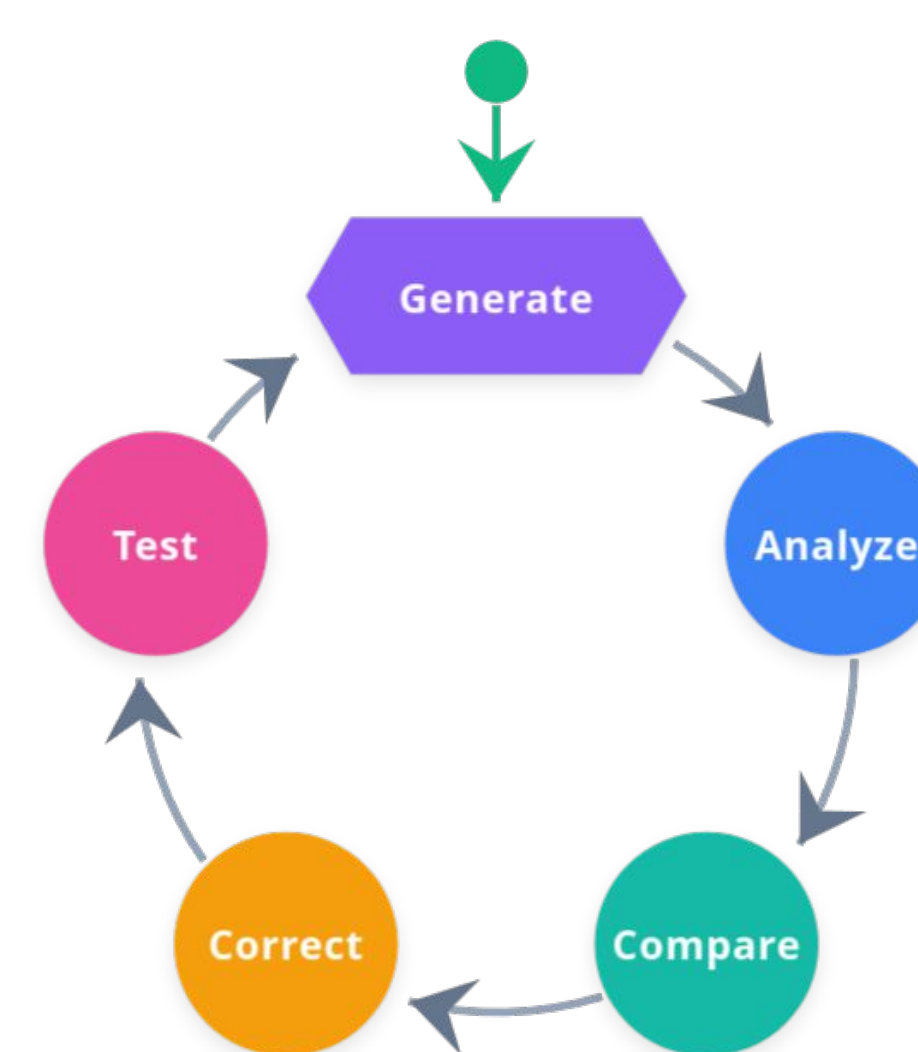
Condense

Condense a code project into an LLM-digestible chunk

```
chunk $PROJECT_PATH --signatures
I -- Initializing walk on: /path/to/project/
I -- Starting file extraction and chunking...
I -- [write] project/chunk_001.txt (4075 chars)
I -- Process complete. 1 chunk(s), 4075 chars.
```

This approach maintains a **clear boundary** between the project and the generative tool. This friction enables better **understanding** of changes and the **reasoning** behind them.

AI Interaction Suggestions



To avoid **Cognitive Surrender**, use key prompts to direct the LLM's creativity toward valuable action and limit byproducts.

AI Gen. Error	Why it Happens	Human Correction
Flag Stripping	AI prioritizes "cleaner" code.	Restore deprecated but required legacy flags.
Logic Bloat	AI attempts to handle all edge cases.	Revert to minimalist, modular functions.
Context Loss	Input token limits or noise.	Condense code into "digestible chunks" before prompt.