spip: A Secure and Version-Controlled Pip Replacement

Antigravity

January 2026

1 Security Hardening and Architectural Evolution

Following a deep architectural review performed by the integrated Gemini Pro AI, `spip` has undergone significant evolution to resolve critical security and design flaws.

2 Implemented Security Fixes

- **Shell Injection Prevention**: Implemented a universal `quote_arg` sanitizer that escapes backslashes, double quotes, and dollar signs. This is now enforced on every `std::system` and `popen` call.
- **Safe Wheel Extraction**: Replaced the legacy `unzip` command with a Python-based `safe_extract.py` helper. This script explicitly validates ZIP member paths to prevent path traversal attacks.
- **Stable Project Hashing**: Replaced `std::hash` with a deterministic FNV-1a mixing algorithm to ensure consistent project IDs and prevent environment collisions.

3 Architectural Refinements

- **Recursive Cleanup**: The `uninstall` and `prune` logic now recursively removes empty parent directories, ensuring zero bloat.
- **Cross-Platform Portability**: Native dependency tracing in `trim` now supports both `otool` (Mac) and `ldd` (Linux).
- **Hardened JSON Parsing**: Metadata extraction now uses non-greedy, targeted regular expressions to eliminate ReDoS risk.
- **Dependency Intelligence**: Improved `requires_dist` parsing for better handling of versioned constraints and platform markers.

4 Verification

The tool has been self-audited using `spip review` and verified to be safe and robust against common environment drift and security vulnerabilities.