CMH Guard: Default-Deny Data, Code, Model, and Compute Controls for Scientific AI

Protection before execution and export

M. Antony Ewing, PhD
Conquer Medical Health
ORCID: 0009-0005-1550-1718

Technical Report v1.0 – 4 September 2026

Abstract

CMH Guard is the cross-cutting data, code, model, compute, and export protection layer for CMH Scientific. Medical and scientific AI can leak through raw tables, notebook outputs, embeddings, checkpoints, caches, logs, environment files, tokens, model artifacts, and proprietary source code. Guard therefore uses a default-deny policy: unknown data are not silently treated as public; clinical-sensitive material remains inside approved execution boundaries unless explicitly authorized; and export or remote execution can be blocked independently of the scientific desirability of the job. The current validation consists of five engineering tests covering unknown/unsafe outbound behavior, secret blocking, clinical-sensitive routing, and public heavy-compute routing. These tests validate software policy behavior only and do not constitute HIPAA, GDPR, institutional, contractual, or penetration-test certification.

Public-release note. This revision aligns the technical report with the September 2026 two-layer software release: a generic callable public interface for new analyses and a separate frozen validation layer for previously reported evidence. Protected operational implementations are not distributed.

1 Purpose

Guard exists because a scientific workflow can leak through much more than a source dataset. Generated artifacts can carry identifiers, sensitive representations, credentials, proprietary code, model weights, or policy-constrained outputs. Guard therefore treats scientific validity and deployment permissibility as separate questions.

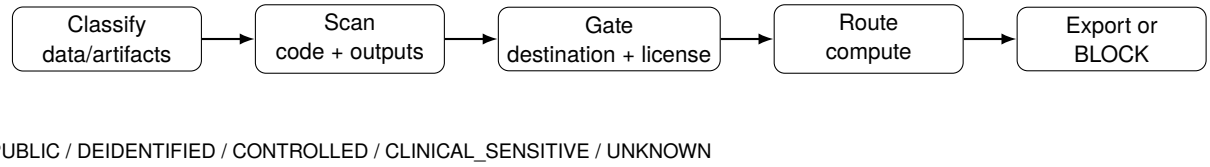

Figure 1: Default-deny Guard flow. A valid scientific job can be blocked at the policy layer.

2 Step-by-step use

1. **Classify the input.** Assign a data class; UNKNOWN remains unknown and must not silently become public.
2. **Scan the bundle.** Inspect source, notebooks, outputs, tokens/secrets, identifier-like fields, embeddings, checkpoints, caches, and model artifacts.
3. **Declare the destination and authorization.** Local, Kaggle, cloud GPU, institutional compute, public repository, and model hubs have different policy implications.
4. **Check model and dataset licenses.** Hosting on a public hub is not equivalent to a universal open-source license.

5. **Route compute.** Heavy public/deidentified jobs may escalate to cloud compute; controlled/clinical work defaults to approved institutional execution absent explicit authorization.
6. **Export only an allow-listed surface.** Publication bundles should be generated intentionally rather than by copying a development directory.

Python public call

```python
from cmh_guard import evaluate_export, route_compute

decision = evaluate_export(
    "analysis_bundle",
    data_class="CLINICAL_SENSITIVE",
    destination="MODAL",
    explicit_remote_authorization=False,
)
print(decision.state)

plan = route_compute(
    data_class="CLINICAL_SENSITIVE",
    rows=2_000_000,
    model_jobs=400,
    gpu_hours=40,
)
print(plan.platform)
```

Current public status. This report defines the public Guard contract and engineering behavior. The standalone production policy engine is not distributed in the current public repository set; Guard behavior is represented in CMH Scientific architecture and release documentation.

Example prompt

Prepare this analysis for remote execution. First classify the data and generated artifacts, scan for identifiers, secrets, notebook outputs, checkpoints, and proprietary code, check model licenses, and block the export unless the destination is permitted.

3 Controls and validation

Controls include credential/secret-pattern scanning; identifier/PHI hints that force review; notebook-output inspection; warnings for checkpoints, caches, weights, and derived representations; patent/proprietary-code markers; model-source license records; data-sensitive compute routing; and execution manifests carrying policy/provenance.

The current engineering validation comprises five passing tests covering unknown/unsafe outbound behavior, secret blocking, clinical-sensitive routing, and public heavy-compute routing. The tests validate policy behavior only; they are not external penetration testing, legal certification, or an institutional HIPAA determination.

4 Medical-data and code/IP boundaries

HHS recognizes Safe Harbor and Expert Determination as de-identification methods under the HIPAA Privacy Rule. Guard may detect obvious risky fields and enforce workflow gates, but it does not replace either formal method, a data-use agreement, IRB/institutional policy, or legal review. Derived artifacts are not automatically downgraded because direct identifiers were removed.

No static scanner can prove the absence of all identifiers, secrets, or intellectual-property leakage. Guard is designed to fail safely and request explicit review.

Public architecture reference: github.com/ConquerMedical/CMH-Scientific-Interface.

5 Execution routing matrix

Environment	Best use	Guard routing principle
Local/Jupyter	Small analyses, private data, rapid iteration	Default for modest jobs when data and dependencies fit locally.
Kaggle	Reproducible public/deidentified work, moderate accelerators	Allowed when the data policy and artifact policy permit notebook execution.
Hugging Face	Model source, artifacts, Paper Pages, demos	Treat as a model/artifact/community layer; inspect each repository license before use or redistribution.
Cloud GPU / Modal	Heavy training, parallel sweeps, batch GPU jobs	Escalate only when workload is heavy and export policy permits; secrets belong in the platform secret store.
Institutional compute	Controlled or clinical-sensitive data	Default protected route absent explicit authorization for external execution.

Table 1: Platform-aware execution without making any platform a scientific dependency.

6 Engineering verification and release use

The Guard white paper is a policy/architecture report rather than a claim of external security certification. Before public release of any CMH Scientific repository, the practical rule is to publish from an allow-listed surface containing only the intended public client, contracts, examples, validation fixtures, documentation, and tests. Raw development directories, caches, model weights, environment files, notebook outputs, and proprietary engine modules should not be copied into the release bundle.

Intellectual property, competing interests, funding, and use

Relevant methods and computational systems are the subject of previously filed U.S. provisional patent applications. This report discloses scientific objects, public interfaces, empirical validation logic, and results needed to evaluate and use the public research architecture. Proprietary operational implementations, calibration internals, candidate-generation procedures, search/ranking policies, deployment orchestration, and other system components not necessary to evaluate the public scientific claims remain outside the scope of disclosure. Publication does not grant a patent or commercial software license.

The author is Founder of Conquer Medical Health and has financial and intellectual-property interests in the technologies described. No external funding supported preparation of this technical report.

Use of AI-assisted tools

AI-assisted tools were used for editorial and software-documentation support. The author determined the scientific claims, supplied the underlying analyses and validation artifacts, and reviewed the numerical content and final report.

References

[1] Ewing, M. A. *Widespread Nonidentifiability in Patient Data Limits AI Prediction Regardless of Model Sophistication*. 2026.

[2] Ewing, M. A. *TopolAI: Information-First Scientific AI*. Conquer Medical Health, 2026.

[3] Ewing, M. A. *OptiCeil: Task-Conditioned Recoverability*. Conquer Medical Health, 2026.

[4] Ewing, M. A. *Acquire: Inverse Measurement Design for Scientific AI*. Conquer Medical Health, 2026.

[5] Ewing, M. A. *AI on Alzheimer Disease: Common-Substrate Reconstruction and Adjudication*. 2026.

[6] Cover, T. M. and Hart, P. E. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory* 13(1), 21–27 (1967).

[7] U.S. Department of Health and Human Services. Guidance Regarding Methods for De-identification of Protected Health Information in Accordance with the HIPAA Privacy Rule.

[8] Kaggle. Notebooks documentation.

[9] Hugging Face. Hub repositories, licenses, Paper Pages, Collections, and Spaces documentation.

[10] Modal. Documentation for serverless compute, GPU workloads, and secrets.