Biosafety Cloud Lab Compliance Screener

Dustin Gault
Independent Researcher
Research conducted at the AIxBio Hackathon, April 2026

Abstract

Cloud labs make it easy to submit and run experiments remotely, but they make oversight harder when review focuses on individual reagents or single steps instead of the whole experiment. This project is a proof of concept biosafety workflow analyzer that screens structured protocols for biosafety, biosecurity, chemical hygiene, shipping, hazardous waste, human-material, facility capability, or custom policy triggers. The process is this: 1) it parses JSON and YAML and normalizes Native JSON, Native YAML, Autoprotocol-like, and Emerald Cloud Lab inputs into a common workflow model; builds a workflow graph from the protocol; applies configurable policy profiles; and optionally enriches the result with an LLM reviewer, which examines it holistically. The dashboard outputs a structured triage: threat level, confidence, flagged steps, missing metadata, and a human-review queue. Eight bundled example protocols and ten deterministic checks cover the main risk patterns I wanted to catch.

1. Introduction

Cloud laboratories let researchers submit protocols as machine readable workflows. That is good for reproducibility and access, but it changes the oversight problem. A protocol can look fine step by step but still need additional review when you read it as a whole experiment. Step level keyword scanning misses the whole picture: aggregate intent, material lineage, facility fit, missing metadata, waste streams, and transfer conditions also all matter.

The goal here is early triage, not adjudication. I am trying to help give a screening layer that sits in front of execution and routes a submission into the right review path.

The contributions:

1. A dashboard for ingesting and screening structured cloud lab protocols.
2. A normalization layer that maps several protocol styles into one workflow model.

3. A configurable policy profile system that exposes which rule domains are active at any given moment.
4. A structured reporting interface with a workflow graph, threat level, confidence, rule findings, missing metadata, queues, and an optional LLM review pass.
5. The ability to export protocols and reports for auditing.

2. Related Work

Biological safety assessment is contextual and workflow dependent; the CDC/NIH BMBL bases risk assessment on agent, procedure, personnel, and facility factors rather than a single checklist [1]. The NIH Guidelines define the institutional governance model for recombinant and synthetic nucleic acid research [2]. OSHA's Laboratory Standard requires a written Chemical Hygiene Plan [3]. The Federal Select Agent Program covers controlled biological agents and toxins [4].

Protocol automation tools, including Autoprotocol style schemas and commercial cloud lab platforms, show that experiments can be represented as machine readable sequences. Most of those representations are built for execution, scheduling, and reproducibility, though, not for review triage. The question I am asking is given a structured workflow, are there any holistic red flags and what oversight path should it enter based upon its threat assessment?

3. Methodology

The project is a front-end browser application that uses Local Storage so it can run on GitHub Pages without needing a backend. The pipeline is:

1. Ingest a protocol from a sample, an upload, or pasted text.
2. Parse the protocol as JSON or a constrained YAML subset.
3. Normalize supported protocol styles into a common model containing materials, operations, facility information, requested execution metadata, and oversight metadata.
4. Validate required fields and surface missing or ambiguous metadata.
5. Build a workflow graph from materials, operations, outputs, and dependencies.
6. Apply the rule domains active in the selected policy profile to derive deterministic findings.
7. Optionally send a normalized screening packet to an LLM endpoint for a holistic overview of the protocol with its findings being used in the final score.
8. Finally, give a risk score and confidence level, then route the submission to auto-triage, clarification, human review, or mandatory human review.

Rule domains currently include biosafety, recombinant nucleic acid, biosecurity, chemical hygiene, hazardous waste, shipping, human materials, and facility capability for the purposes of this POC, but these can be updated and better tuned to actual government policy. The catalog has ten checks covering recombinant or synthetic nucleic acid workflows, biological material propagation, human-derived material handling, hazardous chemical handling, regulated waste streams, shipping or transfer review, controlled-material terms, and facility capability mismatches.

Policy profiles are editable in the dashboard and persisted in browser Local Storage. A profile is a named set of enabled rule domains.

The LLM integration supports Gemini 2.5 Flash (free version for testing) through the Gemini `generateContent` REST API and better models for actual screening. The model is asked to return only structured JSON containing a summary, confidence, threat level, and rules violated. If the LLM returns a threat level, the dashboard updates the final result to reflect that recommendation while preserving the deterministic findings underneath.

4. Results

The prototype ships with eight example protocols: routine low-risk work, incomplete metadata, recombinant/synthetic nucleic acid workflows, human-derived samples, shipping review, controlled-material terms, an Autoprotocol style input, an Emerald Cloud Lab input, and a YAML input.

The dashboard produces:

- Schema validation status and warnings.
- A workflow graph generated from normalized protocol dependencies.
- Deterministic rules and triggers for the selected policy profile.
- Overall status, risk level, confidence score, review route, user, protocol title, and LLM review status.
- A human review queue, an auto-approved list, a submissions table, CSV export, and per-run JSON export.

The test suite is dependency-free. It loads the application in a mocked DOM, validates the default sample, screens it, confirms every bundled sample produces a bounded compliance report. The command `npm test` can be used to verify locally.

The result that matters most is that the screening operates at the experiment level, in addition to the reagent level. The system looks at combinations: biological material plus propagation, recombinant construct plus modification, human-derived material plus

missing review metadata, shipment plus regulated material, waste stream plus facility capability as pertinent examples of biosafety policies.

5. Discussion

Biosafety Workflow Analyzer is a triage and reviewer assistance layer, which I tried to make as transparent and lightweight but still useful. Policy profiles show which rule domains are active. Findings list which rule was triggered and why. The graph gives a fast visual read on the workflow. The LLM layer is optional and structured, and it cannot silently override the deterministic evidence without leaving a report trail.

With more time, the next step is a backend service that stores policy profiles and protocols linked to users, secures API keys, and records immutable audit logs. A production version would also bring in institution specific policy cards and more precise reviewer workflows.

6. Conclusion

This project shows a lightweight approach to cloud lab protocol compliance screening. The core idea is that oversight should be evaluated at the workflow level, because a digital protocol already contains enough structure to derive material lineage, procedural combinations, missing metadata, facility mismatches, and likely review requirements before the work is executed.

The prototype is intentionally conservative. Missing metadata and high consequence signals always route toward human review. That makes it more useful as a compliance and auditing aid.

Code and Artifacts

- GitHub repository: https://github.com/DGault2007/cloud-lab-compliance
- Live demo: https://dgault2007.github.io/cloud-lab-compliance/
- Primary artifact: browser dashboard implemented in HTML, CSS, and JavaScript/React
- Tests: `npm test`

Author Contributions

I was the sole author and developer for all features. My backgrounds are in microbiology, clinical lab science, and software engineering. AI biosecurity is a natural blending.

LLM Usage Statement

OpenAI Codex was used as an implementation assistant for the dashboard work.

References

1. Centers for Disease Control and Prevention and National Institutes of Health. Biosafety in Microbiological and Biomedical Laboratories (BMBL), 6th Edition. https://www.cdc.gov/labs/bmbl/
2. National Institutes of Health. NIH Guidelines for Research Involving Recombinant or Synthetic Nucleic Acid Molecules. https://osp.od.nih.gov/policies/biosafety-and-biosecurity-policy/
3. Occupational Safety and Health Administration. Occupational Exposure to Hazardous Chemicals in Laboratories, 29 CFR 1910.1450. https://www.osha.gov/laws-regs/regulations/standardnumber/1910/1910.1450
4. Federal Select Agent Program. Select Agents and Toxins Regulations. https://www.selectagents.gov/
5. Google AI for Developers. Gemini API Quickstart and generateContent REST examples. https://ai.google.dev/gemini-api/docs/quickstart

Appendix: Limitations and Dual-Use Considerations

Limitations

This is a frontend only MVP. It uses a constrained YAML parser, Local Storage persistence, LLM review, and deterministic heuristics. However, a production ready app would require more specific compliance knowledge. False positives are likely when benign protocols contain words that resemble risk terms and are flagged by LLM. False negatives are possible when users omit material details. The workflow graph is useful for triage but it is not a full provenance model. Improving this would be a helpful extension. Scalability is also limited; there is no backend database currently (although easy to implement), no authentication, no persistent audit log, and no reviewer assignment system.

Dual-Use Risks

The tool is meant to identify protocols that need oversight, but any screening tool can be misused if an attacker probes it to learn which terms or metadata cause escalation. To reduce that risk, the prototype does not provide procedural optimization or execution

instructions. The LLM prompt explicitly frames the model as a compliance reviewer and tells it not to provide experimental optimization or explication.

Responsible Disclosure

No real cloud lab vulnerability, institutional policy gap, or platform specific bypass was discovered during this project as that is beyond its scope.

Ethical Considerations

The system should encourage human oversight when appropriate. It should be transparent to researchers, configurable by authorized personnel, and auditable. It should not collect unnecessary sensitive data. In production, API keys and protocol submissions should not be handled entirely in the browser.

Future Improvements

Future work should add backend validation, role based access control, persistent audit trails, richer graph provenance, and further automated testing for both false negatives and misuse resistance.