– SLIDE 01 / TITLE –

AGENTIC SAMM

when developer is no longer the only actor

SERGEY GORDEYCHIK · scadastrangelove@gmail.com

— SLIDE 02 / SPEAKER —

CATCHING MY FLIGHT

SERGEY GORDEYCHIK

01 · WEB — wasc tc v2 / incident/vuln stat (2007)
02 · ICS / SCADA — scada strangelove (2012)
03 · SD-WAN new hop (2018)
04 · AI / ML SECURITY — talks, courses (2019)
05 · AGENTIC SYSTEMS — ASAMM (today)

05

https://www.linkedin.com/in/sergei-gordeichik-12413438b

– SLIDE 03 / FRAMING –

SAMM WORKS. THE BOUNDARY MOVED.

SAMM COVERS

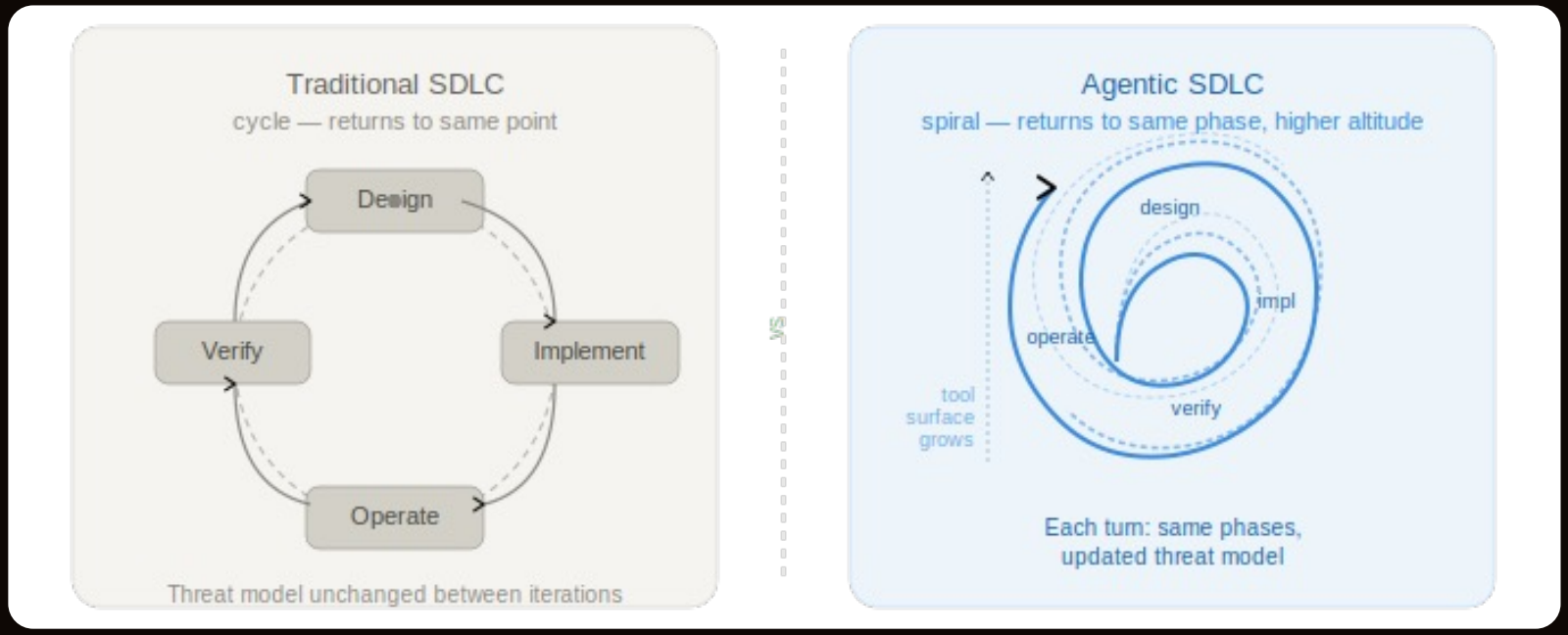
- code and delivery artifacts
- software supply chain
- security testing & review
- vuln & incident management
- five SAMM functions, as designed

ASAMM EXTENDS

- context flows (RAG, memory)
- tool invocations as boundaries
- delegated authority & autonomy
- development as attack surface
- runtime behavior as assurance

– SLIDE 04 / METAPHOR –

CYCLE BECAME A SPIRAL



returns to the same phase – but at a different altitude

tool surface grows · context changes · threat model updates with every turn

– SLIDE 05 / SCOPE –

FOUR SCENARIOS

01 · BUILDING — greenfield agentic system

02 · EMBEDDING — agent inside an existing product

03 · DEV WORKFLOW — claude code, cursor, copilot read your repo

04 · CONSUMING — mcp servers, agent-backed apis you trust

04

– SLIDE 06 / CONCEPT –

Three operational models

Each answers a different question. Together they produce the enforcement gate.

Trust Grading

who do we trust?

A–F source reliability
1–6 behavioral confirmation
self-report ceiling ≤ 3
F6 → sandbox only
promotion: evidence-gated
demotion: fast, asymmetric

Blast Radius

how dangerous is the action?

Asset: C / I / A + P + D
Scope × irreversibility
× lateral propagation
mission impact, not severity
critical × irreversible → approval
composite across agent graph

Delegation Model

how much autonomy?

Tier 0–4: manual → autonomous
= min(Mission, Risk, Trust)
Mission: Auftrag + Rahmen
+ Enforcement
binding ceiling → what to fix
per-path override via AI-03

Trust → WHO. Blast Radius → WHAT. Delegation → HOW MUCH. Together: the enforcement gate.

How to control unknown unknows. Untrusted unknows?

– SLIDE 07 / PROBLEM –

UNKNOWN. UNTRUSTED.

UNKNOWN UNKNOWNS

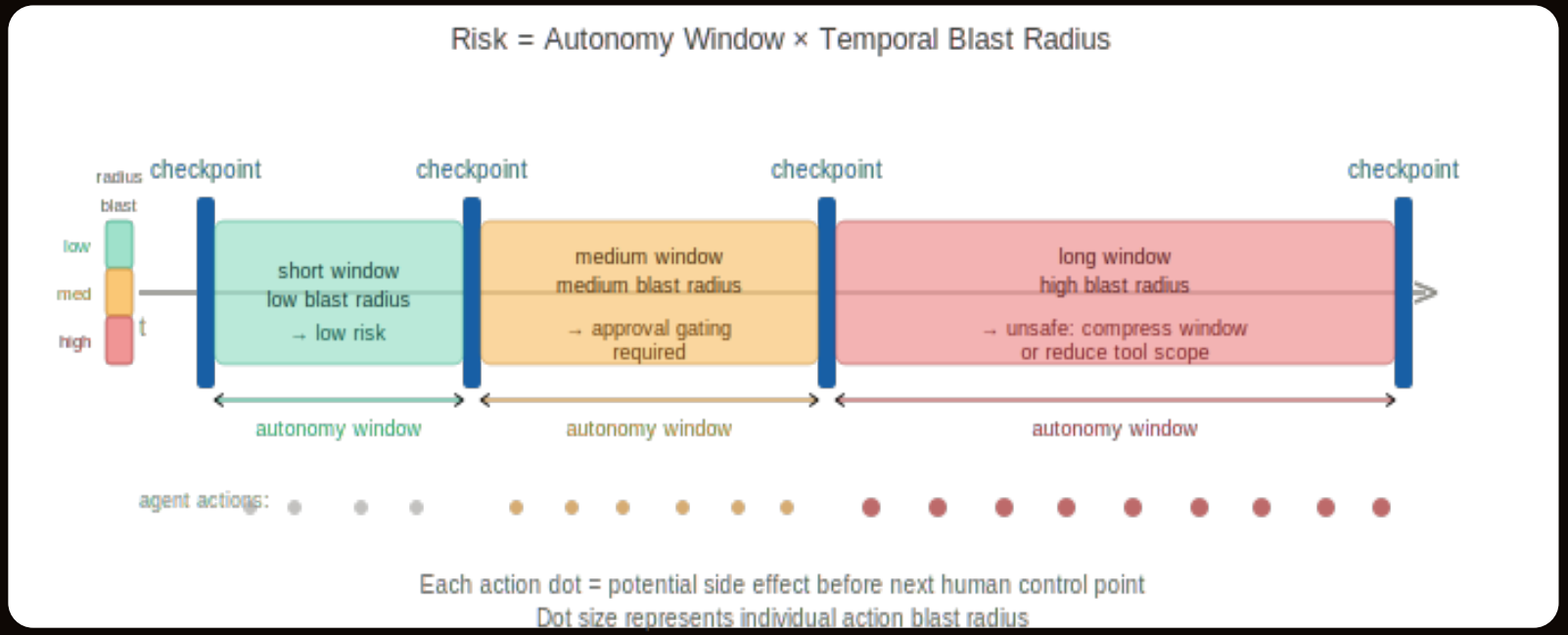
- tool composition you didn’t predict
- context pollution from user input
- prompt drift over long sessions
- emergent strategies in long autonomy
- memory edits that change next-turn behavior

UNTRUSTED UNKNOWNS

- content from search / retrieval
- outputs from upstream agents
- responses from mcp servers
- tool api contracts that changed
- third-party model behavior

— SLIDE 08 / ONE CONTROL —

AD-02 · AUTONOMY WINDOW



Risk = Autonomy Window × Temporal Blast Radius

L1 windows measured · L2 threshold-gated checkpoints · L3 protocol-enforced bounds

– SLIDE 09 / RECEIPTS –

TWO RECEIPTS

DYNAMIC/FORENSIC

- production security tooling pipeline
- 9 of 15 applicable controls at L0
- Setup: agent used as upstream node in a pipeline ending at a security scanner for client engagements.
- Cross-domain blast radius — agent output flows into offensive tooling against client networks.
- memory edits without audit trail

509 AUDITS, BROAD

- 509 repositories scanned
- 20,241 files · 4,882 medium+ findings
- 88 native asamm findings
- autonomous loops · prompt override · trust-boundary expansion

https://github.com/scadastrangelove/agent-audit/

— SLIDE 10 / FRAMEWORK —

ASAMM AT A GLANCE

STRUCTURE

- 21 controls in current public draft
- AG-04 + AI-06 added in v0.3
- 5 SAMM function families: AG / AD / AI / AV / AO
- 6 axioms — incl. evidence primacy
- cc by-sa, open for review

Governance	AG-01 … AG-04	Agent registry, tool registry, kill switch, inter-agent trust protocol
Design	AD-01 … AD-04	Context threat model, autonomy window, tool trust model, dev surface threat model
Implementation	AI-01 … AI-06	Execution boundary, tool authorization, self-modification, value constraints, credential governance
Verification	AV-01 … AV-03	Behavioral testing, adversarial testing, pentest scope includes agent layer
Operations	AO-01 … AO-04	Action logging, intent-action gap monitoring, reassessment triggers, behavioral vulnerability tracking

imported rules + native detectors)

— SLIDE 11 / TOOL —

AGENT-AUDIT AT A GLANCE

WHAT IT IS

· companion audit tool for ASAMM
· reads agent homes, session traces, and instruction-bearing repos
· separates raw hits, clustered issues, and reviewer artifacts
· built for maintainers, auditors, and corpus measurement
· turns abstract controls into inspectable evidence

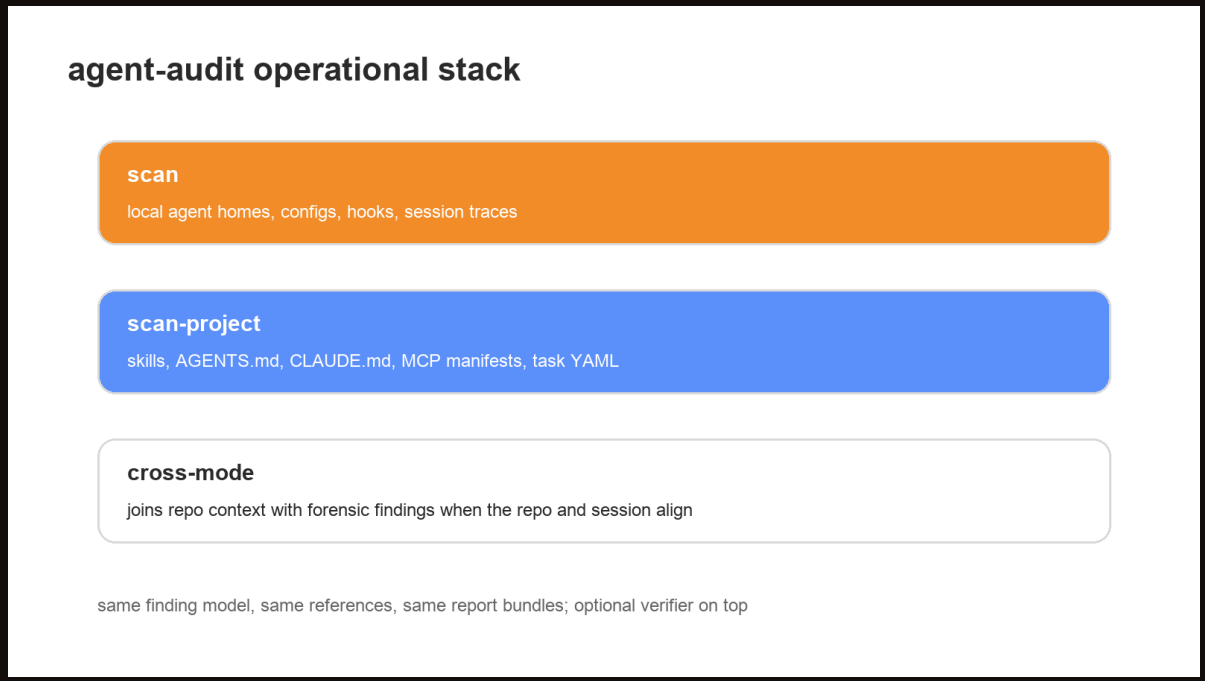
WHAT IT GIVES YOU

· two primary entry modes: forensic and repository surface
· bridge mode joins static and dynamic evidence
· native detectors + imported packs + optional verifier
· JSON and Markdown bundles for operators and researchers
· calibration harnesses for rule evolution over time

— SLIDE 12 / MODES —

TWO ENTRY MODES

01 · FORENSIC / SCAN — local agent home, configs, hooks, session logs
02 · REPOSITORY / SCAN-PROJECT — skills, AGENTS.md, CLAUDE.md, MCP manifests
03 · CROSS-MODE — static context reinforces dynamic findings
04 · VERIFY — optional context-aware triage for ambiguous results



— SLIDE 13 / FORENSIC —

FORENSIC / SCAN

INPUTS

· session JSONL from Claude Code, Codex, and OpenClaw-style traces
· agent configs, hooks, MCP settings, and local probes
· tool calls, autonomy windows, and evidence of risky chains
· optional repo resolution for follow-on cross-mode context
· verifier-ready records for triage when needed

OUTPUTS

· raw findings and sessions of concern
· verified-first Markdown and JSON bundles
· config patch suggestions where relevant
· machine-readable audit log for follow-up

— SLIDE 14 / STATIC —

REPOSITORY / SCAN-PROJECT

SURFACES

· SKILL.md, AGENTS.md, and CLAUDE.md instruction files
· plugin manifests, task YAML, and prompt-bearing fields
· MCP manifests, tool descriptions, and capability declarations
· collections of repos for measurement and benchmarking
· AST-routed parsing before lexical rule application

ARTIFACTS

· project-findings.md and project-findings.json
· clustered findings and files of concern
· security profile plus report profiles
· collection-scale rollups for repeated templates

— SLIDE 15 / PRECISION —

AST / SURFACE ROUTING

FLAT TEXT MODE

· code snippets trigger prose-oriented rules
· examples look like live capability declarations
· every regex sees every file as flat text
· manifests become opaque blobs instead of typed records
· false positives pile up before triage even begins

AGENT-AUDIT ROUTES

· markdown AST splits prose from code and examples
· prompt-bearing YAML fields are extracted explicitly
· MCP manifests are parsed at field granularity
· rules apply only on declared audit surfaces
· the same engine supports both product and corpus modes

— SLIDE 16 / SOURCES —

MANY SOURCES. ONE FINDING MODEL.

IMPORTED PACKS

· ATR
· Aguara
· Cisco PromptGuard
· Gitleaks
· NOVA and optional Cisco MCP YARA

NORMALIZED OUTPUT

· one severity model
· one finding and evidence model
· one audit-surface model
· dedup, canonical classes, collection grouping
· imported rules become comparable instead of siloed

— SLIDE 17 / NATIVE —

WHY NATIVE DETECTORS

PACKS ALONE MISS

· approval gaps and broad external action without consent
· persistent identity rewrite and future-session carryover
· destructive or exfiltration chains across turns
· cross-server toxic flow between MCP tools
· absence-based controls with no obvious keyword trigger

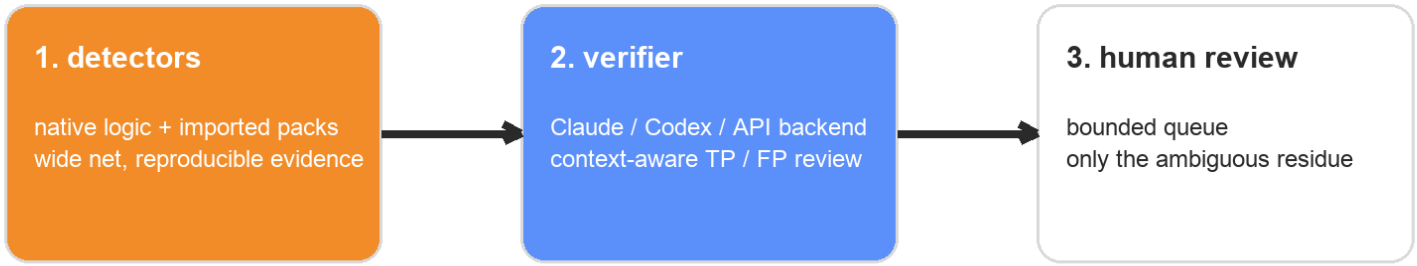
WHY THEY EXIST

· some findings are absence-based, not lexical
· some are behavioral, temporal, or cross-turn
· some require session structure, not flat text
· they map directly back into ASAMM controls
· this is where framework semantics become detector logic

— SLIDE 18 / TRIAGE —

FP TRIAGE IS BUILT IN

false-positive triage path

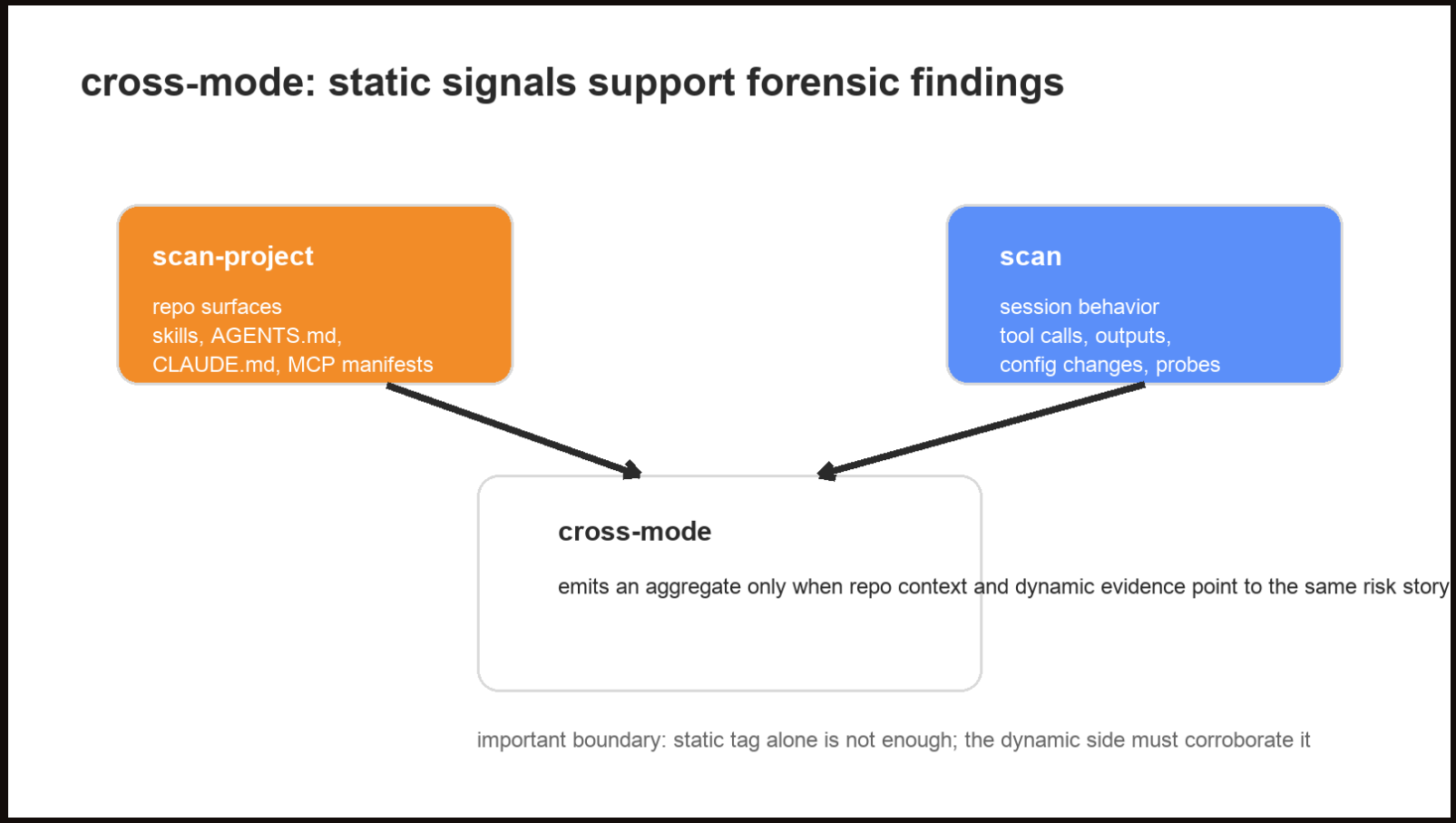


design point: keep raw signal visible, but do not confuse pattern match with final judgment

Claude / Codex / API backends · default off · bounded by budget

— SLIDE 19 / BRIDGE —

CROSS-MODE JOINS THE STORY



— SLIDE 20 / VALUE —

WHAT THIS ENABLES FOR ASAMM

PRACTICE

· local agent audits after suspicious runs
· release gates for skills and MCP manifests
· third-party repo triage before trust decisions
· evidence that maps directly into controls
· repeatable review bundles for security teams

MEASUREMENT

· 509-repo corpus scans and reviewer packs
· adjudication subsets for signal calibration
· regression harnesses for future rule packs
· a path from abstract controls to operational evidence

– SLIDE 11 / ASK –

Run your first ASAMM audit

github.com/scadastrangelove/asamm · audit/samples/ouroboros

1. **Clone the repo**
   git clone github.com/scadastrangelove/asamm
2. **Read the audit prompt**
   audit/samples/ouroboros — "How to run this audit yourself"
3. **Give the prompt to an LLM**
   Claude, ChatGPT, or another — point it at your project
4. **Request an integrity review**
   Ask the LLM to review its own audit — this catches errors
5. **Conduct the mission interview**
   Answer: what cannot you afford to lose?

Result

First audit in 1–2 hours

↓

Concrete findings
with file:line evidence

↓

Remediation roadmap
ordered by priority and effort

crosswalk review · evidence-rule calibration · real audits

github.com/scadastrangelove/asamm · scadastrangelove@gmail.com