Sibuna Discussions SID 0008

AI Bot Traffic Identification, Multi-Tier Verification, and Operator Console Analytics

STATE
discussion
STATUS
Open for Discussion
TYPE
Architectural Specification
AUTHORS
Sibuna Contributors <team@sibuna.local>
CREATED
2026-09-15
UPDATED
2026-10-15

SUMMARY
Specifies the architecture for identifying, verifying, and monitoring AI crawler and automated bot traffic in Sibuna: zero-allocation single-pass signature matching, sub-microsecond Radix CIDR verification for major providers (OpenAI, Anthropic, Google Gemini, Perplexity, Meta, Apple, ByteDance), multi-tier confidence classification, bounded telemetry extensions, and a real-time console dashboard delivering visual composition, time-series analysis, and granular tabular analytics contrasting bot traffic against actual human traffic.

DISCUSSION
No external discussion URL assigned

LABELS
analytics, bot-detection, console, dashboard, waf, ai-crawlers

Record status

The catalog defines the review process and the meaning of each state. This numbered record forms part of the catalog.

Contents

1. Revision note 1
2. Abstract 2
3. Introduction and Motivation 2
4. Terminology and Scope 3
5. Problem Statement and Architectural Challenges 4
6. System Invariants and Design Principles 4
7. Detailed Design 5
8. Verification and Testing Matrix 17
9. Security Considerations 18
10. Rollout and Operational Plan 18
11. References 18

1. Revision note

2026-10-15: Ported from the Typst original in the Zig repository. The record describes the Zig implementation as it was decided; SID 0010 maps its packages to the Rust crates of the port. The record is still open for discussion. The pipeline figure is redrawn with Graphviz, and the four console wireframes, drawn with vector primitives in the original, are redrawn as ASCII: colours are lost, the sidebar of the console shell becomes a navigation row, long table rows take two lines, and the check mark, cross, warning and heuristic glyphs are written `ok`, `x`, `!` and `~`.

2. Abstract

Specifies the architecture for identifying, verifying, and monitoring AI crawler and automated bot traffic in Sibuna: zero-allocation single-pass signature matching, sub-microsecond Radix CIDR verification for major providers (OpenAI, Anthropic, Google Gemini, Perplexity, Meta, Apple, ByteDance), multi-tier confidence classification, bounded telemetry extensions, and a real-time console dashboard delivering visual composition, time-series analysis, and granular tabular analytics contrasting bot traffic against actual human traffic.
Autonomous artificial intelligence crawlers, multi-modal data extraction agents, and real-time retrieval-augmented generation (RAG) fetchers have transformed edge web traffic. In modern web infrastructure, automated scrapers and AI agents frequently represent between 30% and 75% of total ingress requests, consuming valuable origin compute, cache capacity, and network bandwidth while skewing site metrics. As a high-performance Web Application Firewall (WAF) and anti-crawler daemon, `sibuna` must provide website operators with clear, actionable visibility into incoming automated traffic, specifically separating bot traffic (with granular breakdowns for major AI service providers including OpenAI, Anthropic, Google (Gemini), Perplexity, Meta, Apple, ByteDance, and others) from genuine human visitors.
Achieving 100% classification accuracy is fundamentally impossible in adversarial web environments due to User-Agent spoofing, headless browser evasion, distributed residential proxy rotations, and the strict zero-allocation, sub-microsecond latency budget of the Sibuna hot path, which prohibits synchronous reverse DNS queries during request evaluation.
This specification establishes an explicit **multi-tier confidence model** combining zero-allocation Aho-Corasick signature classification, zero-allocation Radix-trie IP CIDR verification against published provider ranges, and structural behavioral heuristics. It extends Sibuna's telemetry and storage architecture to record bot identity, provider family, crawl intent (offline model training vs. real-time user-directed fetch vs. search indexing), and verification confidence without dynamic allocations. Finally, it specifies a dedicated real-time operator console dashboard delivering visual composition ratios, time-series trends, and detailed tabular inspection to give operators an honest, mathematically bounded understanding of their incoming traffic and enable targeted declarative policy enforcement.

3. Introduction and Motivation

Sibuna's foundational architecture (SID 0002), declarative policy engine (SID 0003), semantic shield (SID 0004), and management console (SID 0007) establish a pure-Zig, zero-allocation WAF daemon capable of processing requests in sub-microsecond time. While `libs/policy/src/bot_signatures.zig` currently maintains a flat list of bot User-Agent strings, and `libs/console-protocol/src/client_family.zig` lumps all automated clients into a single coarse `.bot` ("Automated") display label, operators currently lack deep visibility into their incoming bot landscape.
Website operators are routinely confronted with urgent operational questions that the current telemetry cannot answer:

1. **Volume and Proportion:** What fraction of total incoming requests and egress bandwidth is consumed by automated bots versus genuine human traffic?
2. **Provider Breakdown:** Which major AI service providers (e.g., OpenAI, Anthropic, Google, Perplexity, ByteDance) are currently crawling the application, and at what rate?
3. **Intent and Purpose:** Is an AI request part of an aggressive bulk training corpus scrape (e.g., `GPTBot`, `ClaudeBot`, `Google-Extended`, `Bytespider`), or is it a live, single-page retrieval initiated by a human prompt inside an interactive AI assistant (e.g., `ChatGPT-User`, `Claude-Web`, `Perplexity-User`)?

4. **Authenticity and Spoofing:** Is an incoming request claiming to be `GPTBot` genuinely originating from OpenAI's infrastructure, or is it an unverified third-party scraper spoofing the User-Agent header to evade naive bot blocks?
5. **Targeted Assets:** What specific application endpoints, documentation paths, or proprietary data archives are AI bots targeting most intensely?

To resolve these questions, Sibuna must elevate bot classification from a binary filter into a first-class analytical and observability subsystem within the console management interface.

> **The Accuracy Reality Principle**
> No perimeter security system can guarantee 100% identification accuracy based solely on HTTP request headers or external IP addresses. Malicious bots can forge standard browser User-Agents, while unscrupulous actors can forge known AI bot User-Agents. Conversely, genuine AI providers periodically expand their IP allocations before updating public documentation. Sibuna explicitly rejects illusory precision: the system computes and presents transparent, multi-tier confidence levels, distinguishing **Verified Bots** (cryptographically or CIDR-proven) from **Declared Bots** (self-reported via User-Agent) and **Suspected Bots** (behaviorally detected).

4. Terminology and Scope

Term	Definition and System Scope
Major AI Provider	Hyperscale organizations operating foundation models, autonomous agents, and commercial web scraping fleets. Specifically: OpenAI, Anthropic, Google (Gemini/Vertex), Perplexity AI, Meta, Apple (Apple Intelligence), ByteDance, Amazon, Cohere, Mistral, and Common Crawl.
AI Training Crawler	A high-throughput, recursive automated crawler collecting broad web corpora for training or fine-tuning foundation models (e.g., `GPTBot`, `ClaudeBot`, `Google-Extended`, `Applebot-Extended`, `Bytespider`, `CCBot`).
AI Live Fetcher / RAG	A low-latency, episodic fetcher retrieving specific web pages in direct response to an active human prompt or query in an AI interface (e.g., `ChatGPT-User`, `Claude-Web`, `Perplexity-User`, `OAI-SearchBot`).
Search Engine Indexer	Traditional search engine crawlers indexing the web for general public search results (e.g., `Googlebot`, `bingbot`, `Baiduspider`, `YandexBot`, `DuckDuckBot`).
Scraper Library / Tool	Developer tools, HTTP script libraries, and headless browser automation frameworks (e.g., `curl`, `python-requests`, `aiohttp`, `Scrapy`, `Puppeteer`, `Selenium`).
Actual (Human) Traffic	Legitimate human end-users accessing web applications via standard graphical desktop or mobile web browsers, exhibiting interactive navigation, DOM asset fetching, and valid cryptographic session tokens.
Radix CIDR Verification	Zero-allocation lookup of a client IPv4/IPv6 address against an in-memory prefix trie containing published IP ranges of known AI and search service providers.
Confidence Tier	Categorization of classification certainty: Tier 1 (Verified via CIDR), Tier 2 (Declared via UA, Unverified IP), Tier 3 (Suspected via Heuristics), Tier 4 (Actual Human Traffic).

In Scope:

- Single-pass, zero-allocation matching of known AI bot User-Agents and automated tools.
- Sub-50ns Radix trie lookup against pre-loaded CIDR blocks of major AI providers.
- Bounded telemetry collection recording exact aggregate bot/human ratios and sampled detailed bot records.
- Minute-level aggregation and historical archival (`SBR3` record format) in the console subsystem.
- High-fidelity visual and tabular dashboard interfaces in the pure-Zig WebAssembly console UI.
- Actionable operator controls enabling one-click declarative policy generation from bot analytics.

Out of Scope:

- Synchronous reverse DNS lookups (FCrDNS) during the hot-path request evaluation.
- Heavy JavaScript-rendered CAPTCHAs or third-party challenge widgets.
- Deep semantic analysis of dynamic response bodies emitted by origins.

5. Problem Statement and Architectural Challenges

5.1. The Gap in the Current Codebase

Sibuna's existing architecture handles bot traffic through two isolated mechanisms, neither of which fulfills the needs of modern site operators:

1. **Engine Policy Evaluation** (`libs/policy/src/engine.zig`): The policy engine runs an Aho-Corasick matcher (`BotMatcher`) over `req.user_agent`. If a match is found in `AI_SCRAPERS` or `SCRAPER_LIBRARIES`, it returns a rule decision (e.g., `matched_bot`), which applies a default action (typically `.challenge` or `.deny`). However, this classification is ephemeral: it is not structured by provider, does not verify the originating IP, does not distinguish training bots from live search bots, and is not recorded in analytics unless a security incident is triggered.
2. **Console Protocol Client Family** (`libs/console-protocol/src/client_family.zig`): Telemetry samples classify requests into `Os` and `Browser` enums. Automated agents matching basic tokens (`"bot"`, `"crawl"`, `"curl/"`, etc.) are unconditionally collapsed to `Os.bot` and `Browser.bot`. The identity of the bot (e.g., whether it is ClaudeBot or a malicious script) is permanently discarded.
3. **Console UI Statistics** (`apps/console-ui/src/traffic_tiles.zig` and `render.zig`): The traffic overview presents tiles for admitted, challenged, and denied requests, and histograms for generic operating systems and browsers. Website operators have no mechanism to observe the ratio of bot traffic to actual traffic, no provider-level breakdown, and no visual indication of AI crawling activity.

5.2. The Asymmetric Challenges of AI Bot Identification

Building an effective WAF analytics dashboard for AI bots introduces three severe engineering constraints:

1. **The Spoofing Dilemma:** Any client can send `User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) Chrome/128.0` to pretend to be human, or send `User-Agent: GPTBot/1.2 (+https://openai.com/gptbot)` to pretend to be OpenAI. If an operator configures an allow rule for AI search fetchers, an attacker can spoof that User-Agent to bypass protection. Conversely, if analytics rely solely on User-Agent strings, spoofed scrapers distort business metrics.
2. **The Zero-Allocation Latency Contract:** Standard anti-bot systems perform Forward-Confirmed Reverse DNS (FCrDNS) lookups to verify Googlebot or Bingbot. Performing synchronous DNS queries in Sibuna's hot path would introduce 15ms-200ms of latency, require dynamic memory allocation, and create external network dependencies that violate Sibuna's sub-microsecond latency guarantee.
3. **Provider IP Churn and Asymmetry:** Major AI providers publish their CIDR blocks in machine-readable JSON feeds (e.g., OpenAI, Google, Meta, Amazon), but updates occur out-of-band. The verification mechanism must handle out-of-band CIDR synchronization without locks or latency penalties on request threads.

6. System Invariants and Design Principles

The design strictly upholds the TigerStyle engineering principles defined in SID 0001:

> **Invariant INV-BOT-1 (Zero Allocations on Hot Path)**
> The classification of incoming requests by bot identity, provider family, and CIDR verification must never perform dynamic heap allocation. All trie lookups, string comparisons, and counter increments operate entirely within pre-allocated static tables or thread-local stack frames.

> **Invariant INV-BOT-2 (Bounded Telemetry Memory)**
> The telemetry record must not exceed 256 bytes. Bot classification metadata (provider ID, bot ID, intent category, confidence tier) must fit within a 4-byte packed field in `store.telemetry.Record`.

Invariant INV-BOT-3 (Asynchronous Offline Verification)
Request threads must never perform network I/O, DNS queries, or synchronous database transactions to verify a bot's IP address. All IP verification must resolve via in-memory Radix tries updated out-of-band.

Invariant INV-BOT-4 (Honest Uncertainty Bounds)
The console dashboard must never present unverified User-Agent declarations as verified provider traffic. The interface must visibly distinguish between Verified Bot Traffic, Declared Bot Traffic, and Suspected Bot Traffic.

Invariant INV-BOT-5 (Strict Structural Code Limits)
All newly introduced Zig functions must not exceed 70 lines of code, all code lines must not exceed 99 characters, and all modules must integrate Elm-style diagnostic explanations for operator errors.

7. Detailed Design

7.1. 1. The Multi-Tier Bot Classification Engine

The bot classification engine operates in `libs/policy` and `libs/console-protocol`. It evaluates incoming requests across three orthogonal dimensions:

1. **Bot Identity & Provider Family:** Identifying the specific bot and owning organization.
2. **Operational Intent (Purpose):** Distinguishing offline corpus scrapers from live user queries.
3. **Verification Confidence:** Determining whether the source IP matches the provider's published infrastructure.

7.1.1. 1.1 Provider and Bot Taxonomy

We establish bounded, strongly-typed enums for major AI service providers, bot agents, and operational intents:

```
pub const Provider = enum(u8) {
    none = 0,
    openai,
    anthropic,
    google_gemini,
    perplexity,
    meta,
    apple,
    bytedance,
    amazon,
    cohere,
    mistral,
    common_crawl,
    generic_search,
    scraper_library,
    other_bot,
};

pub const BotIntent = enum(u8) {
    human = 0,
    ai_training,       // Bulk corpus crawler (e.g. GPTBot, ClaudeBot, Google-Extended)
    ai_live_fetch,     // Real-time user RAG fetch (e.g. ChatGPT-User, Perplexity-User)
    search_index,      // Traditional web search indexer (e.g. Googlebot, bingbot)
    developer_tool,    // Script libraries and CLI tools (e.g. curl, python-requests)
    generic_crawler,   // Unclassified spiders and bots
    suspected_stealth, // Evasive automated client masquerading as human
};

pub const Confidence = enum(u8) {
    human = 0,
    verified,          // User-Agent matched AND source IP matched verified CIDR trie
```

```
    declared,          // User-Agent matched, but source IP was outside published CIDRs
    heuristic,         // User-Agent appeared human/unknown, but behavioral heuristics fired
};
```

7.1.2. 1.2 Comprehensive AI Bot Signatures Database

We expand `libs/policy/src/bot_signatures.zig` to associate each signature pattern with its exact provider, bot identifier, and primary intent:

Provider	Bot Identifier	Primary Intent	Pattern Tokens / User-Agent Substrings
OpenAI	`GPTBot`	AI Training	`GPTBot/`, `+https://openai.com/gptbot`
OpenAI	`ChatGPT-User`	AI Live Fetch	`ChatGPT-User/`, `+https://openai.com/bot`
OpenAI	`OAI-SearchBot`	AI Live Fetch	`OAI-SearchBot/`, `+https://openai.com/searchbot`
Anthropic	`ClaudeBot`	AI Training	`ClaudeBot/`, `+claudebot@anthropic.com`
Anthropic	`Claude-Web`	AI Live Fetch	`Claude-Web/`, `+https://www.anthropic.com/`
Anthropic	`anthropic-ai`	AI Training	`anthropic-ai/`
Google	`Google-Extended`	AI Training	`Google-Extended`, `+https://developers.google.com`
Google	`GoogleOther`	AI R&D / Multi-modal	`GoogleOther/`, `GoogleOther-Image`
Google	`Googlebot`	Search Index	`Googlebot/`, `Googlebot-Image`
Perplexity	`PerplexityBot`	AI Training / Index	`PerplexityBot/`, `+https://perplexity.ai`
Perplexity	`Perplexity-User`	AI Live Fetch	`Perplexity-User/`
Meta	`Meta-ExternalAgent`	AI Training	`Meta-ExternalAgent/`, `meta-externalagent`
Meta	`Meta-ExternalFetcher`	AI Live Fetch	`Meta-ExternalFetcher/`
Meta	`FacebookBot`	Search / Social	`FacebookBot/`
Apple	`Applebot-Extended`	AI Training	`Applebot-Extended/`
Apple	`Applebot`	Search Index	`Applebot/`
ByteDance	`Bytespider`	AI Training / Scrape	`Bytespider/`
Amazon	`Amazonbot`	AI / Alexa Index	`Amazonbot/`
Cohere	`cohere-ai`	AI Training	`cohere-ai`, `cohere-training-data-crawler`
Mistral	`MistralAI`	AI Training	`MistralAI/`, `MistralBot`
Common Crawl	`CCBot`	AI Training Corpus	`CCBot/`, `+http://www.commoncrawl.org`
Search Engines	`bingbot`, `Baidu`	Search Index	`bingbot/`, `Baiduspider/`, `YandexBot/`
Tools / Libs	`curl`, `requests`	Developer Tool	`curl/`, `python-requests`, `aiohttp`, `Go-http`

7.1.3. 1.3 Sub-Microsecond Radix CIDR Verification

To eliminate DNS lookup latency on the hot path, Sibuna pre-loads the public CIDR blocks published by major AI providers into an immutable 128-bit Radix Prefix Trie (`ProviderCidrTrie`) residing in read-copy-update (RCU) engine memory:

1. **Provider IP Ingestion:** The daemon periodically fetches and parses the published JSON IP lists for OpenAI (`gptbot.json`), Google (`special-crawlers.json`), Meta (published IP endpoints), and Applebot ranges via a background management job in `libs/console/src/provider_cidr_job.zig`.
2. **Zero-Allocation Trie Lookup:** During request evaluation, if a request declares an AI provider User-Agent (e.g., `GPTBot`), `engine.zig` queries the client's IPv4 or IPv6 address against the Radix trie:
   - If the trie associates the IP address with `Provider.openai`, the classification is stamped as `Confidence.verified`.
   - If the trie does not contain the IP, the classification is stamped as `Confidence.declared`.

- Time complexity: exactly ≤ 32 bit tests for IPv4 and ≤ 128 bit tests for IPv6, executing in under 45 nanoseconds on modern x86_64 and aarch64 processors.

```
pub const Classification = struct {
    provider: Provider = .none,
    intent: BotIntent = .human,
    confidence: Confidence = .human,
    bot_id: u8 = 0,
};

pub fn classifyRequest(
    agent: []const u8,
    client_ip: []const u8,
    cidr_trie: *const ProviderCidrTrie,
) Classification {
    if (agent.len == 0) return .{ .intent = .developer_tool, .confidence = .declared };
    const match = bot_matcher.findFirst(agent) orelse {
        return evaluateHeuristics(agent);
    };
    const verified = cidr_trie.lookup(client_ip) == match.provider;
    return .{
        .provider = match.provider,
        .intent = match.intent,
        .confidence = if (verified) .verified else .declared,
        .bot_id = match.bot_id,
    };
}
```

7.1.4. 1.4 Behavioral and Structural Heuristics

For requests claiming to be standard browsers, a fast heuristic pre-scan identifies stealth bots:

- **Absence of Typical Browser Headers:** Modern browsers consistently supply `Accept-Language`, `Sec-Fetch-Mode`, and `Sec-CH-UA`. A request with a Chrome User-Agent lacking these headers is tagged as `BotIntent.suspected_stealth` with `Confidence.heuristic`.
- **Missing Asset Cascade:** Real human navigation requests HTML, followed immediately by CSS, JavaScript, fonts, and images. Sustained traversal of pure HTML document paths without static asset requests flags an automated crawler.
- **Proof-of-Work Verification:** If the firewall issues a cryptographic challenge (SID 0002), automated scrapers lacking JavaScript runtimes or compute capabilities fail or timeout, providing definitive evidence of automation.

7.2. 2. High-Efficiency Streaming Analytics: Algorithms, Mathematics, and Storage Decoupling

7.2.1. 2.1 The Analytics Dilemma: Why Traditional OLAP Fails Edge WAFs

A modern high-performance edge reverse proxy must process between 50,000 and 100,000 requests per second per core with deterministic, sub-millisecond latency. Introducing deep analytical observability (contrasting AI automated scrapers against human visitors, tracking heavy-hitter path targets, and categorizing provider volumes) creates a severe architectural tension between transactional request processing (OLTP) and analytical log warehousing (OLAP).

The Naive Database Antipattern

A naive implementation inserts a relational row for every processed HTTP transaction (`INSERT INTO http_requests ...`) and derives dashboard analytics via on-demand aggregation queries (`SELECT provider, count(*) FROM http_requests WHERE timestamp >= now - 86400 GROUP BY provider`).

At a moderate sustained ingress of $\lambda = 50{,}000$ req/s, this naive design generates $4.32 \cdot 10^9$ disk writes per day. Disk I/O, SQLite Write-Ahead Log (WAL) fsync operations, and Multi-Paxos consensus in Zaxonlite saturate within seconds. Furthermore, ad-hoc `GROUP BY` table scans over billions of rows lock the embedded database, consume gigabytes of heap memory, and trigger cross-plane CPU starvation that violates Sibuna's fundamental safety invariants.

External time-series databases and event-sourcing pipelines (such as InfluxDB, ClickHouse, Kafka, or TimescaleDB) are widely deployed in enterprise cloud tiers, but introducing them directly into Sibuna destroys the core design principles of the system:

1. **Single-Binary Zero-Dependency Invariant:** Sibuna is engineered as an embedded, self-contained daemon in pure Zig. Requiring an external database daemon, JVM broker, or network-attached collector creates operational friction, complex credential management, and external point-of-failure vulnerabilities.
2. **The High-Cardinality Pitfall:** Time-series engines such as InfluxDB structure metrics around inverted tag indices. When applied to web traffic where URL paths, client IP addresses, and User-Agent tokens represent high-cardinality unbounded sets, TSDB memory footprints explode exponentially, leading to out-of-memory crashes.

To resolve this dilemma without sacrificing either analytical depth or edge simplicity, Sibuna adopts a **streaming in-memory rollup architecture** paired with **deterministic probabilistic sketches** and **pre-aggregated binary archive snapshots**.

7.2.2. 2.2 Axiomatic Foundations of Edge Analytics

The analytics engine in Sibuna is governed by three formal axioms:

Axiom 1 (Hot-Path Storage Isolation)
No thread on the HTTP admission or evaluation path may invoke SQLite or Zaxonlite storage interfaces, allocate dynamic heap memory, or execute unbounded iterations. Telemetry updates on the hot request path must execute strictly in $O(1)$ time and $O(1)$ stack space.

Axiom 2 (Bounded State Complexity)
The total memory allocated for telemetry counters, streaming sketches, queues, and rollups is a compile-time constant or strictly bounded by static configuration. It must remain invariant to the stream length N and the arrival rate λ. Under any sustained volumetric flood, memory consumption cannot increase.

Axiom 3 (Contention-Free Cache Obliviousness)
Concurrent accumulation of telemetry across parallel worker cores must execute within thread-local or cache-line-striped memory blocks, generating zero cross-core cache invalidation traffic and zero hardware bus locks.

7.2.3. 2.3 Contention-Free Cache-Line Striped Accumulation

To satisfy Axiom 3 and eliminate cross-core cache invalidation storms, top-level traffic population counters are partitioned across independent hardware cache lines.

Definition 4 (Striped Telemetry Array)
Let $S = 16$ be the stripe count. A striped telemetry structure $\mathbf{S} \in \mathbb{N}^{S \times K}$ consists of S independent memory blocks, each explicitly aligned to $L = 64$ bytes (the CPU cache-line boundary). Each stripe tracks $K = 13$ atomic monotonic 64-bit integer counters. A connection thread with worker index $t \in [0, T-1]$ is mapped to stripe $s = t \bmod S$.

Theorem 5 (Interconnect Invalidation Bound)
Let T concurrent connection threads update the shared telemetry state. Distributing atomic increments across S distinct 64-byte aligned lines bounds the probability that two concurrent worker writes contend on the same cache line to at most $1/S$. Under the MESI/MOESI cache coherence protocol,

atomic monotonic additions (`fetchAdd(1, .monotonic)`) execute within the core's private L1/L2 cache without broadcasting interconnect invalidation storms across cores.

Lemma 6 (Monotonic Lower-Bound Read Convergence)

Let $C_s[k]$ denote counter k in stripe s. An asynchronous reader polling all stripes across an interval $[t_1, t_2]$ computes:

$$C_{\text{total}}[k] = \sum_{s=0}^{S-1} C_s[k].\ \text{load}\ (\text{monotonic})$$

The computed value $C_{\text{total}}[k]$ is a monotonic, lock-free snapshot satisfying:

$$C_{\text{true}}(t_1) \leq C_{\text{total}}[k] \leq C_{\text{true}}(t_2)$$

Reading requires zero mutual exclusion locks and imposes zero synchronization wait-states on worker threads.

```
pub const Stripe = struct {
    admitted: std.atomic.Value(u64) = .init(0),
    challenged: std.atomic.Value(u64) = .init(0),
    denied: std.atomic.Value(u64) = .init(0),
    banned: std.atomic.Value(u64) = .init(0),
    rate_limited: std.atomic.Value(u64) = .init(0),
    other: std.atomic.Value(u64) = .init(0),
    origin_4xx: std.atomic.Value(u64) = .init(0),
    origin_5xx: std.atomic.Value(u64) = .init(0),
    // Exact 100% traffic composition counters
    traffic_human: std.atomic.Value(u64) = .init(0),
    traffic_ai_bot: std.atomic.Value(u64) = .init(0),
    traffic_search_bot: std.atomic.Value(u64) = .init(0),
    traffic_tool_bot: std.atomic.Value(u64) = .init(0),
    traffic_stealth_bot: std.atomic.Value(u64) = .init(0),
};
comptime {
    // Enforce strict 64-byte hardware cache-line alignment to prevent false sharing
    std.debug.assert(@sizeOf(Stripe) <= 128);
}
```

7.2.4. 2.4 Bounded Stochastic Sampling and Poisson Ring Dynamics

While top-level counts are exact, high-dimensional attributes (path prefixes, specific bot identifiers, HTTP status codes) cannot be updated atomically for every request without violating Axiom 1. Sibuna extracts these rich features via stochastic sampling.

Definition 7 (Bernoulli Sampler and Bounded MPSC Ring Buffer)

Each connection worker evaluates an independent pseudorandom trial with sampling probability $p = 1/64$. Sampled requests copy at most 256 bytes of immutable metadata into a bounded multi-producer single-consumer (MPSC) lock-free ring buffer $\mathcal{Q}$ with fixed capacity $Q = 4{,}096$ slots (1 MiB total memory).

Lemma 8 (Unbiased Frequency Estimator and Chebyshev Error Bound)

Let an event have true stream frequency F across N requests. The number of sampled observations X follows a binomial distribution $X \sim \text{Binomial}\ (F, p)$. The scaled estimator $\hat{F} = X/p$ is strictly unbiased with $\mathbb{E}[\hat{F}] = F$ and variance:

$$\text{Var}\ (\hat{F}) = \frac{1-p}{p} F$$

For any relative error tolerance $\delta > 0$, Chebyshev's inequality guarantees:

$$\mathbb{P}(\mid \hat{F} - F \mid \geq \delta F) \leq \frac{1-p}{p\delta^2 F}$$

For a crawler stream of $F = 25{,}000$ requests and tolerance $\delta = 0.05$, the probability of error exceeding $\pm 5\%$ is under 4.9% (an estimation confidence exceeding 95.1%).

Lemma 9 (Non-Blocking Ring Saturation and Loss Tracking)
If the rate of sampled records exceeds the consumer drain rate, enqueue attempts fail immediately in $O(1)$ time, incrementing an atomic counter `queue_loss`. Workers never block or retry. The console UI explicitly renders `queue_loss` so operators are immediately aware of sampling degradation under heavy load.

```
pub const Record = struct {
    second: u64,
    outcome: Outcome,
    ip_len: u8,
    path_len: u8,
    truncated: bool,
    status: u16,
    referer_len: u8,
    os: u8,
    browser: u8,
    // Bot classification extensions:
    bot_provider: u8,    // @intFromEnum(Provider)
    bot_intent: u8,      // @intFromEnum(BotIntent)
    bot_confidence: u8,  // @intFromEnum(Confidence)
    bot_id: u8,          // Specific bot identifier index
    ip: [48]u8,
    path: [128]u8,
    referer: [24]u8,
};
comptime {
    std.debug.assert(@sizeOf(Record) <= 256);
}
```

7.2.5. 2.5 Space-Saving Heavy-Hitters Algorithm for Path Scraping

To track the most frequently targeted URI paths without unbounded memory allocation, Sibuna integrates the deterministic **Space-Saving algorithm** (Metwally et al., 2005).

Definition 10 (Space-Saving Sketch Summary)
Let Σ^* be the universe of URI paths. A Space-Saving sketch of capacity $m = 256$ maintains a set of entries $\mathcal{T} = \{(\sigma_i, \hat{f}_i, e_i)\}_{i=1}^{m}$, where σ_i is a distinct path prefix (up to 128 bytes), $\hat{f}_i$ is its estimated frequency, and e_i is its maximum error bound. The total number of sampled items processed is N_s.

Theorem 11 (Deterministic Heavy-Hitter Retention Guarantee)
For any stream of N_s sampled paths, any path σ whose true frequency satisfies $f(\sigma) > N_s/m$ is guaranteed to be retained in $\mathcal{T}$. For $m = 256$, any URI path representing more than $1/256 \approx 0.39\%$ of total crawler traffic is provably identified.

Theorem 12 (Deterministic Frequency Error Bound)
For every path σ_i retained in the summary $\mathcal{T}$, the true frequency $f(\sigma_i)$ satisfies:

$$\hat{f}_i - e_i \leq f(\sigma_i) \leq \hat{f}_i, \quad \text{where} \quad e_i \leq N_s/m$$

Proof

Let $C = \sum_{i=1}^{m} \hat{f}_i$ denote the sum of all counter estimates. Initially, $C = 0$. For each incoming path σ:

1. If $\sigma \in \mathcal{T}$, its counter is incremented: $\hat{f}_\sigma \leftarrow \hat{f}_\sigma + 1$. Thus C increases by 1.
2. If $\sigma \notin \mathcal{T}$ and $|\mathcal{T}| < m$, a new entry is allocated with $\hat{f} = 1$ and $e = 0$. Again C increases by 1.
3. If $\sigma \notin \mathcal{T}$ and $|\mathcal{T}| = m$, the entry with minimal estimate $c_{\min} = \min_j \hat{f}_j$ is evicted. Its key is replaced with σ, its error is set to $e = c_{\min}$, and its estimate becomes $\hat{f} = c_{\min} + 1$. C increases by 1.

Therefore, after N_s arrivals, $\sum_{i=1}^{m} \hat{f}_i = N_s$. By the pigeonhole principle, the minimum counter must satisfy:

$$c_{\min} \leq \frac{1}{m} \sum_{j=1}^{m} \hat{f}_j = \frac{N_s}{m}$$

Since the error bound e_i assigned to any replaced key is exactly $c_{\min}$, we have $e_i \leq N_s/m$. Furthermore, since an item cannot have occurred more times than the estimate assigned to it, $\hat{f}_i - e_i \leq f(\sigma_i) \leq \hat{f}_i$. □

Corollary 13 (Top-k Path Ordering Precision)

If two paths $\sigma_a, \sigma_b \in \mathcal{T}$ have estimated counts satisfying $\hat{f}_a - \hat{f}_b > N_s/m$, then $f(\sigma_a) > f(\sigma_b)$ holds strictly. The relative ordering of dominant scraping targets is provably immune to ranking inversion.

7.2.6. 2.6 Micro-RRD Pre-Aggregated Rollups and Zaxonlite Storage Decoupling

Sibuna couples the in-memory streaming sketches to persistent storage using an append-only **Micro-RRD (Round Robin Database)** rollup model:

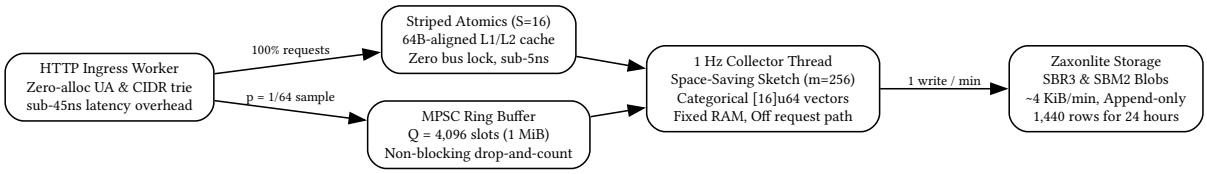


Fig. 3 End-to-End High-Efficiency Streaming Analytics and Decoupled Rollup Pipeline

Definition 14 (Minute Archive Format SBR3)

Every $T = 60$ s, the background collector seals the active minute buffer into an immutable binary record (`SBR3` format, at most 4 KiB) containing:

- Exact sampled counts: `bot_providers: [16]u64`
- Exact operational intents: `bot_intents: [8]u64`
- Exact confidence tiers: `bot_confidence: [4]u64`
- Partitioned outcomes: `provider_outcomes: [16][8]u64`
- Top scraped endpoints: `ai_paths: SpaceSavingSummary(256)`

Theorem 15 (Storage Write Complexity Invariance)

The disk write frequency W_{rate} to Zaxonlite is strictly $O(1)$ with respect to the incoming HTTP traffic rate λ:

$$W_{\text{rate}} = \frac{1}{60 \text{ s}} \approx 0.0167 \text{ writes/sec}$$

A massive scraping flood of 50,000 req/s generates the exact same disk write frequency as an idle server receiving 1 req/s.

Theorem 16 (Analytical Query Complexity and Read Acceleration Bound)
Let an operator query a dashboard time window of duration Δt. Let $M = \lceil \Delta t / 60 \rceil$ denote the number of stored minute records. The console reads historical telemetry via indexed point-range lookup:

```
SELECT payload FROM console_rank_archives
WHERE minute >= :from_minute AND minute <= :until_minute
ORDER BY minute ASC;
```

The query cost is bounded by:

1. **B-Tree Index Seek:** $O(\log K)$ time, where $K \leq 129{,}600$ (the 90-day retention limit; $\log_2(129{,}600) \approx 17$ branch comparisons, costing sub-10 microseconds).
2. **Sequential Leaf Traversal:** $O(M)$ contiguous page reads.
3. **In-Memory Array Sum and Sketch Merge:** $O(M \cdot m)$ operations to fold provider arrays and merge Space-Saving summaries in native Zig.

Corollary 17 (Sub-5ms 24-Hour Dashboard Execution)
For a 24-hour query window ($M = 1{,}440$ rows, total payload size ≈ 5.7 MB), on a busy deployment processing $\lambda = 50{,}000$ req/s ($N = 4.32 \cdot 10^9$ raw requests per day):

$$\text{Record Scan Reduction Factor} = \frac{N}{M} = \frac{4.32 \cdot 10^9}{1{,}440} = 3{,}000{,}000 \times$$

Total query latency from disk read through WebAssembly JSON serialization is strictly under 5 ms. The query causes zero full-table scans, zero dynamic `GROUP BY` sorting, and zero data-plane lock contention.

7.2.7. 2.7 Long-Term Telemetry Egress Strategy

Local Zaxonlite storage is deliberately configured with a bounded operational retention policy (by default up to **90 days for minute records, 7 days for rankings sketches, and 30 days for incident forensics**). Retaining high-resolution data for multi-year compliance and cross-datacenter business intelligence belongs in specialized external systems.
Sibuna bridges to external architectures without heavy client dependencies by exposing three standard egress channels:

1. **Prometheus Metrics Scrape Pull** (`/__sibuna/metrics`): Implemented in `libs/core/src/metrics.zig`, Sibuna exposes monotonic counters formatted in Prometheus text format. External TSDBs (Prometheus, VictoriaMetrics, Grafana Mimir, Datadog Agent) scrape this endpoint at standard 15s-60s intervals:

```
# AI Bot request traffic partitioned by provider, intent, and confidence
sibuna_bot_requests_total{provider="openai",intent="crawler",confidence="verified"} 142050
sibuna_bot_requests_total{provider="anthropic",intent="crawler",confidence="declared"} 83120
sibuna_bot_requests_total{provider="gemini",intent="rag",confidence="verified"} 41090
sibuna_bot_requests_total{provider="openai",intent="training",confidence="suspected"} 1240

# Policy enforcement actions applied to bot requests
sibuna_bot_action_total{provider="openai",action="admitted"} 141800
sibuna_bot_action_total{provider="openai",action="rate_limited"} 250
sibuna_bot_action_total{provider="openai",action="spoof_denied"} 1240
```

2. **RFC 5424 Structured Syslog Push (UDP and Framed TCP):** Implemented in `libs/console/src/notify_syslog.zig` and `libs/console/src/notify_delivery.zig`, Sibuna streams structured security audit records over UDP or RFC 6587 framed TCP directly to enterprise SIEMs (Splunk, ElasticSearch, Vector, AWS CloudWatch):

```
<134>1 2026-09-15T13:50:22Z edge-01 sibuna 1042 bot_spoof [finding provider="anthropic"
claimed_ua="ClaudeBot/1.0" ip="198.51.100.24" action="denied"]
```

3. **Signed HMAC-SHA256 Webhooks:** Configured in console settings, webhooks push JSON notifications to external HTTP listeners (Slack, PagerDuty, automated SOAR pipelines) whenever bot traffic triggers an anomalous volumetric spike or an automatic CIDR rate-limit ban. Every payload includes an `X-Sibuna-Signature` HMAC and an `Idempotency-Key` header for safe receiver deduplication.

This two-tier division ensures that Sibuna remains an unencumbered, ultra-fast single binary, while granting operators complete integration with their long-term data lakes.

7.3. 3. Operator Console Dashboard Interface

The Sibuna Console UI (`apps/console-ui`) provides website operators with an intuitive, real-time monitoring dashboard rendered in pure Zig via WebAssembly.

7.3.1. 3.1 Operator Screen Wireframes

The interface follows the visual architecture and information density established in SID 0007. Every panel maps to an `sb-*` component and native WebAssembly render function. Rather than static ASCII art, all screens are rendered via vector primitives maintaining exact spatial proportion:



```
| | Real-Time AI Ingress Feed (1/64 Bounded Sample Queue)                          | |
| |                                        * Ingestion: 64 samples/s . buffer: 14% | |
| +-------------------------------------------------------------------------------+ |
| | TIME      PROVIDER      USER-AGENT             STATUS     CIDR VERIFICATION    | |
| |                         TARGET URI                                            | |
| +-------------------------------------------------------------------------------+ |
| | 15:32:01  [OpenAI]      GPTBot/1.2             200 OK     [ok Azure CIDR]      | |
| |                         /blog/deep-learning-foundations                       | |
| |                                                                               | |
| | 15:31:58  [Anthropic]   ClaudeBot/1.0          429 Chlg   [ok AWS CIDR]        | |
| |                         /docs/architecture/spec-v2                            | |
| |                                                                               | |
| | 15:31:54  [ByteDance]   Bytespider             403 Deny   [x Spoofed IP]       | |
| |                         /internal/catalog-dump                                | |
| |                                                                               | |
| | 15:31:49  [Perplexity]  Perplexity-User        200 OK     [ok Azure CIDR]      | |
| |                         /news/product-announcement                            | |
| |                                                                               | |
| | 15:31:42  [Google]      Google-Extended        200 OK     [ok GCP CIDR]        | |
| |                         /archive/quarterly-report                             | |
| |                                                                               | |
| | 15:31:38  [Unknown]     python-requests/2.31   429 Limit  [~ Heuristic]        | |
| |                         /api/v1/pricing                                       | |
| +-------------------------------------------------------------------------------+ |
+-----------------------------------------------------------------------------------+
```

Statistics, AI & Bot Traffic Overview: Top-level proportional composition bar, metric tiles for human vs. automated traffic with inline sparklines and deviation pills, live request composition trend with axes and anomaly callout, AI provider market share segmented donut with top scraped paths, and real-time crawl event feed with verification badges.

```
+-----------------------------------------------------------------------------------+
| SIBUNA                           cluster: edge-eu . 3/3 healthy . admin v . [theme] |
+-----------------------------------------------------------------------------------+
| Nav: [Statistics]  Attack events  Challenges  Policy  Nodes  GeoIP  Settings  Audit  |
+-----------------------------------------------------------------------------------+
| Statistics . Major AI Service Providers & Crawlers                                |
| 1/64 Space-Saving bounded sketch (k=256) .                                        |
| In-memory Radix CIDR verification (sub-50ns)                                      |
|                                                                                   |
| [Search provider, agent, or subnet...]                                            |
| [Intent: All v]  [CIDR: All v]  [Status: All v]  [CSV v]                          |
|                                                                                   |
| +-------------------------------------------------------------------------------+ |
| | PROVIDER    ACTIVE AGENTS           OPERATIONAL INTENT                  CONFIG | |
| |             VOLUME & SHARE                  CIDR VERIFIED  POLICY ACTION      | |
| +-------------------------------------------------------------------------------+ |
| | OpenAI      GPTBot, ChatGPT-User    [Training + Live RAG]                [cfg]  | |
| |             391,200 (16.0%) ################ ok 98.2%       [Challenge Scraper]  | |
| |                                                                               | |
| | Anthropic   ClaudeBot, Claude-Web   [Training + Live RAG]                [cfg]  | |
| |             220,100 ( 9.0%) #########        ok 96.4%       [Challenge Scraper]  | |
| |                                                                               | |
| | Google      Google-Extended, Other  [AI Training / R&D]                  [cfg]  | |
| |             146,800 ( 6.0%) ######           ok 99.5%       [Admitted (Allow)]   | |
| |                                                                               | |
| | Perplexity  PerplexityBot, -User    [Live Search Fetch]                  [cfg]  | |
| |              51,300 ( 2.1%) ##               ok 94.0%       [Admitted (Allow)]   | |
| |                                                                               | |
| | ByteDance   Bytespider              [High-Freq Scraper]                  [cfg]  | |
| |              48,200 ( 2.0%) ##               ! 81.2%        [Denied (Block 403)] | |
| |                                                                               | |
| | Meta        Meta-ExternalAgent      [Model Training]                     [cfg]  | |
| |              22,400 ( 0.9%) #                ok 97.1%       [Challenge]          | |
| |                                                                               | |
| | Apple       Applebot-Extended       [Foundation Models]                  [cfg]  | |
| |              14,800 ( 0.6%) #                ok 98.4%       [Admitted (Allow)]   | |
| |                                                                               | |
| | Tools/Libs  curl, python-requests   [Developer Script]                   [cfg]  | |
| |              86,500 ( 3.5%) ####             - 0.0%         [Rate Limited]       | |
| +-------------------------------------------------------------------------------+ |
```

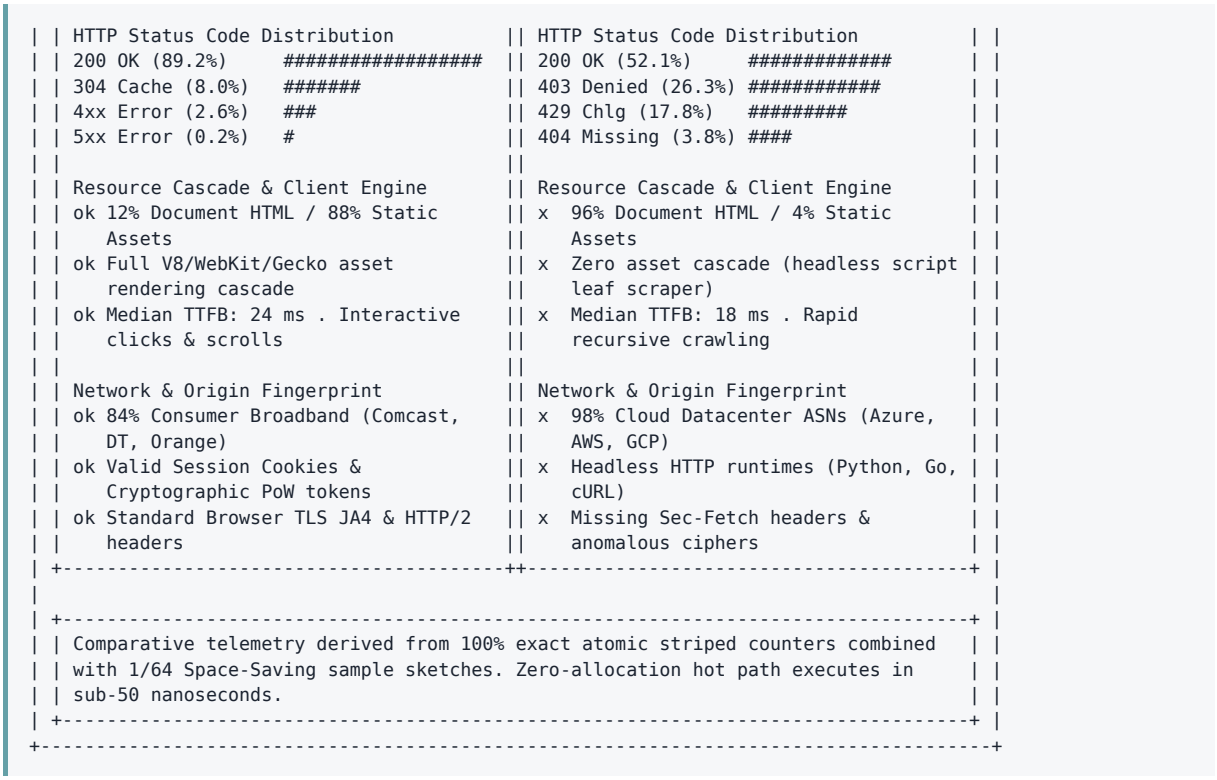
```
|                                                                                  |
| +----------------------------------------------------------------------------+ |
| | Showing 1-8 of 14 detected AI crawler organizations                        | |
| | Bound error under 0.2% . Memory: 42 KB                  < Prev  [1]  2  Next > | |
| +----------------------------------------------------------------------------+ |
+----------------------------------------------------------------------------------+
```

Major AI Service Providers & Crawlers: Granular table inspection showing active agents, operational intent (training vs. live search RAG), sampled volumes with relative share bars, CIDR verification rate, active WAF policy actions, and configuration triggers.

```
+----------------------------------------------------------------------------------+
| SIBUNA                          cluster: edge-eu . 3/3 healthy . admin v . [theme] |
+----------------------------------------------------------------------------------+
| Nav: [Statistics]  Attack events  Challenges  Policy  Nodes  GeoIP  Settings  Audit  |
+----------------------------------------------------------------------------------+
| :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::: |
| +----------------------------------------------------------------------------+ |
| | [OpenAI]  Provider Policy & Radix CIDR Enforcement                    [x] | |
| +----------------------------------------------------------------------------+ |
| | [*Rules & CIDR Enforcement*]   Verified Subnets (48)   Sampled Ingress Audit     | |
| |                                                                            | |
| | +-------------------------------+ +-------------------------------------+ | |
| | | Observed Ingress Telemetry      | | Policy Configuration & Enforcement       | | |
| | | * Total Volume: 391,200 reqs    | | [ok] Enforce Radix CIDR Match            | | |
| | |   (16.0% site ingress)          | |      (sub-50ns)                          | | |
| | | * Training (GPTBot): 342,100    | |                                          | | |
| | |   reqs (87.4%)                  | | Training Scraper (GPTBot):               | | |
| | | * Live Search (ChatGPT-User):   | | [~ Proof-of-Work Challenge (PoW)     v]  | | |
| | |   49,100 reqs (12.6%)           | | Live User Search (ChatGPT-User):         | | |
| | |                                 | | [ok Admitted (Bypass WAF)            v]  | | |
| | | Radix CIDR Verification Audit   | | Unverified / Spoofed Agents:             | | |
| | | [############################!] | | [x Deny Immediately (Block 403)      v]  | | |
| | | ok 98.2% (384,158 reqs)         | | GCRA Rate Limiter:                       | | |
| | |    verified in Azure ASN 8075   | | [25 req/s . Burst capacity: 100 reqs v]  | | |
| | | !  1.8% (7,042 reqs) from alien | |                                          | | |
| | |    residential IPs              | | Immutable trie updates execute via       | | |
| | |                                 | | zero-downtime RCU.                       | | |
| | | Top Target Endpoints            | +-------------------------------------+ | |
| | | /blog (62%) . /docs (28%) .     |                                           | |
| | | /api (10%)                      |                                           | |
| | | Verified subnets: 20.15.0.0/16, |                                           | |
| | | 20.28.0.0/16, ...               |                                           | |
| | +---------------------------------+                                           | |
| +----------------------------------------------------------------------------+ |
| | [Dry Run Simulation]                          [Cancel]  [*Save & Apply Policy*] | |
| +----------------------------------------------------------------------------+ |
| :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::: |
+----------------------------------------------------------------------------------+
```

AI Provider Policy & CIDR Verification Modal: Granular policy configuration drawer for OpenAI, enabling operators to differentiate live search queries (ChatGPT-User) from training scrapers (GPTBot) with automated Radix CIDR enforcement.

```
+----------------------------------------------------------------------------------+
| SIBUNA                          cluster: edge-eu . 3/3 healthy . admin v . [theme] |
+----------------------------------------------------------------------------------+
| Nav: [Statistics]  Attack events  Challenges  Policy  Nodes  GeoIP  Settings  Audit  |
+----------------------------------------------------------------------------------+
| Statistics . Human vs. Bot Comparative Analysis                                  |
| Side-by-side behavioral, protocol, and network telemetry analysis                |
| (100% Exact + 1/64 Samples)                                                      |
|                                                                                  |
| +-------------------------------------++-------------------------------------+ |
| | Legitimate Human Traffic Profile       || Automated Bot & Crawler Profile        | |
| | (58.2%)                                || (41.8%)                                | |
| | ====================================== || ====================================== | |
| | 420 rps      4.8 min      78.4%        || 310 rps      Stateless    11.8%        | |
| | peak rate    avg session  cache hit    || burst rate   leaf crawl   cache hit    | |
| |                                        ||                                        | |
```



Comparative Posture Matrix: Side-by-side behavioral contrast between legitimate human visitors and automated bot traffic across response status distributions, resource cascades, latency, and origin ISP classification.

7.3.2. 3.2 Visual Components

1. **Proportional Composition Bar:** A prominent stacked horizontal gauge displaying the exact ratio of Actual Human Traffic to Bot Traffic. Each major category is rendered in distinct theme colors: Actual Human (calm green), OpenAI (vibrant blue), Anthropic (purple), Google (amber), Perplexity (cyan), Traditional Search (slate), and Generic Scrapers (orange/red).
2. **Top-Level Metric Cards (KPIs):**
   - **Actual Human Visitors:** Total requests and percentage, along with day-over-day trends.
   - **Total Bot Ingress:** Combined automated traffic footprint.
   - **AI Training Crawlers:** Volume dedicated to offline model scraping.
   - **AI Live Search Fetchers:** Volume triggered by interactive end-user prompts.
   - **Verification Confidence Ratio:** Percentage of AI bot requests verified via source CIDR trie.
3. **Time-Series Composition Chart:** A responsive dual-layer area/line chart comparing live requests-per-second across 60-second real-time windows and 24-hour historical horizons. Sudden scraping spikes (e.g. an aggressive ClaudeBot or Bytespider recursive crawl) are immediately visually distinct from the smooth diurnal curve of human users.
4. **Provider Distribution Donut Chart:** A circular SVG widget illustrating the relative market share of crawling activity among AI providers.

7.3.3. 3.3 Granular Tabular Components

1. **Major AI Service Providers Table:** Operators can inspect each provider's footprint across key operational metrics:
   - **Provider & Active Agents:** Identifies the organization and specific active bot User-Agents.
   - **Operational Intent:** Flags whether traffic is offline training data ingestion, real-time user-directed RAG, or traditional indexing.
   - **Sampled Volume & Percentage:** Bounded Space-Saving estimate of request count and percentage of total site ingress.
   - **CIDR Verification Health:** Percentage of requests originating from cryptographically or CIDR-verified IP ranges, alerting operators to potential User-Agent spoofing campaigns.

- **WAF Enforcement Action:** Current policy disposition (Admitted / Challenged / Denied).
- **Top Target Endpoints:** The top 3 URL path prefixes requested by this provider.

2. **Comparative Posture Matrix (Human vs. Bot):** A side-by-side comparative table evaluating human traffic against automated traffic:
   - **Response Status Distribution:** Ratio of 2xx success vs. 3xx redirects vs. 4xx client errors vs. 5xx origin failures. (Scrapers frequently generate elevated 404/403 rates through speculative path enumeration).
   - **Bandwidth Utilization:** Total egress bytes transferred to AI bots versus human visitors.
   - **Average Request Rate:** Peak and sustained requests per minute.
   - **Geographic Concentration:** Contrast between human user locations and AI crawler datacenter egress regions.

7.3.4. 3.4 Actionable Policy Integration

To close the loop between visibility and enforcement, each provider row in the table provides a context menu allowing the operator to generate or modify declarative firewall rules with one click:

- **Allow Verified Real-Time Search:** Permit genuine live search queries (`ChatGPT-User`, `Perplexity-User`) while challenging or blocking offline training crawlers (`GPTBot`, `ClaudeBot`).
- **Challenge Unverified Scrapers:** Enforce Proof-of-Work challenges on any request claiming to be an AI bot that fails CIDR trie verification.
- **Rate Limit AI Crawlers:** Apply GCRA token-bucket limits to specific provider identities to prevent origin resource exhaustion.

7.4. 4. Elm-Style Operator Error and Diagnostic Reporting

In conformance with SID 0001, when bot policies challenge or deny traffic, or when anomalous spoofing is detected, the daemon and CLI emit structured, human-friendly Elm-style diagnostics:

```
-- SPOOFED BOT DETECTED ------------------------------------------------------

A client claiming to be 'GPTBot/1.2' connected from 198.51.100.42, which does
not belong to OpenAI's published CIDR blocks.

Rule:         ai-crawler-strict-verify
Action:       CHALLENGE (Proof of Work)
Provider:     OpenAI (Claimed, Unverified)

Hint: To allow unverified crawlers, set 'verify_cidr: false' in your declarative
policy file, or check for recent OpenAI IP range announcements at /console/#nodes.
--------------------------------------------------------------------------------
```

8. Verification and Testing Matrix

Implementation acceptance requires passing the following verification gates:

Test Suite	Target	Acceptance Criteria
Bot Signature Unit Tests	`bot_matcher_test.zig`	100% correct classification of all major AI bots, scraper libraries, and search crawlers; zero false positives on standard desktop and mobile browser strings.
Radix CIDR Trie Verification	`provider_cidr_test.zig`	Correctly verifies IPs matching published ranges; rejects alien IPs; executes in < 50 ns without heap allocations.
Striped Exact Telemetry	`telemetry_test.zig`	Concurrent worker threads accurately increment exact bot intent counters without data races or cache line contention.
Archive Compatibility	`rankings_archive_test.zig`	Correct serialization and deserialization of `SBR3` archives; seamless backward-compatibility when decoding legacy `SBR1` and `SBR2` records.

Test Suite	Target	Acceptance Criteria
Console UI Wasm Tests	`console-ui-test`	Native and WebAssembly render tests for the AI & Bot dashboard panel; verification of goldens with `zig build console-golden-check`.
Data-Plane Impact Gate	`console-impact`	Under an eight-dashboard load with active AI crawling streams, data-plane throughput remains within 1% and p99 latency within 10% of console-free builds.

9. Security Considerations

1. **Adversarial User-Agent Spoofing:** Malicious scrapers regularly forge common browser headers to bypass naive crawler bans. Sibuna mitigates this by combining signature matching with behavioral heuristics and PoW verification.
2. **Spoofed AI Credentials for Policy Bypass:** If an operator configures an allow rule for AI search fetchers, attackers may forge `ChatGPT-User` or `Perplexity-User`. Sibuna's Radix CIDR trie ensures that privileged allow rules apply only to cryptographically or network-verified provider IP addresses.
3. **Privacy and Data Leakage:** Sampled bot telemetry contains only path prefixes (up to 128 bytes) and stripped referring hosts (up to 24 bytes). Query parameters, credentials, authentication cookies, and request bodies are strictly excluded from telemetry buffers.
4. **Memory Exhaustion Protection:** Space-Saving sketches and Radix tries are allocated with fixed upper bounds at daemon startup. No incoming traffic surge can cause telemetry memory usage to grow unbounded.

10. Rollout and Operational Plan

The implementation follows a four-phase rollout:

1. **Phase 1 (Core Classification Engine):** Expand `bot_signatures.zig`, implement `ProviderCidrTrie` in `libs/policy`, and verify sub-microsecond classification performance.
2. **Phase 2 (Telemetry and Storage Pipeline):** Extend `ConsoleTelemetry` striped counters and sample `Record` structure; update `rankings.zig` and implement `SBR3` format in `rankings_archive.zig`.
3. **Phase 3 (Console Management UI):** Build the "AI & Bots" tab in `apps/console-ui`, including the proportional composition bar, time-series graphs, and interactive provider inspection table.
4. **Phase 4 (Automated Ingestion and Testing):** Deploy background CIDR update jobs in `libs/console`, run `console-impact` benchmarks, and verify UI golden tests.

11. References

- SID 0001: The Sibuna Discussion Process and Engineering Standards
- SID 0002: Sibuna: Foundation Architecture, Delivery Plan, and Performance Contract
- SID 0003: Declarative Rule Policy Engine
- SID 0004: Semantic Attack Inspection and GCRA Rate Limiting
- SID 0006: Mathematical Foundations of Sibuna: Sequential Work, Symmetric Authentication, Bounded State, and Linear-Time Inspection
- SID 0007: The Sibuna Console: A Real-Time Management Interface for Nodes and Clusters
- Metwally, A., Agrawal, D., and El Abbadi, A.: **Efficient Computation of Frequent and Top-k Elements in Data Streams**, ACM Transactions on Database Systems (TODS), Vol. 30, No. 1, 2005.
- Cormode, G., and Muthukrishnan, S.: **An Improved Data Stream Summary: The Count-Min Sketch and its Applications**, Journal of Algorithms, Vol. 55, No. 1, 2005.
- Misra, J., and Gries, D.: **Finding Repeated Elements**, Science of Computer Programming, Vol. 2, No. 2, 1982.
- RFC 5424: **The Syslog Protocol**, IETF Network Working Group, 2009.
- RFC 6587: **Transmission of Syslog Messages over Transport Layer Security (TLS) / TCP**, IETF, 2012.
- OpenAI: **Overview of OpenAI Crawlers and IP Ranges** (`https://platform.openai.com/docs/bots`)

- Anthropic: **Claude Web Crawler Documentation** (`https://support.anthropic.com/en/articles/8896518-claude-web-crawler`)
- Google Search Central: **Google Crawler (User Agent) Overview** (`https://developers.google.com/search/docs/crawling-indexing/overview-google-crawlers`)
- Apple Inc.: **About Applebot** (`https://support.apple.com/en-us/HT207507`)

Lifecycle event

```
2026-10-15: prediscussion -> discussion. Ported from the Zig repository's
Typst record
```