



IMTLazarus General Technical Architecture

Official Enterprise Systems Engineering & Nationwide Scalability Specification (v3.0) • July 2026

Source Authority: <https://docs.imtlazarus.com/docs/guias/arquitectura-general>

1. EXECUTIVE ENGINEERING SUMMARY & SYSTEM TAXONOMY

The **IMTLazarus Enterprise Ecosystem** is a cloud-native, highly distributed, multi-tenant Mobile Device Management (MDM) and Digital Classroom Safeguarding platform engineered to operate across nationwide educational infrastructures. Designed from the ground up to support high-concurrency scale across millions of student devices, the platform guarantees sustained performance and zero bottlenecking during peak active school hours (**8:00 AM – 4:00 PM**).

Traditional MDM architectures suffer from structural latency and server congestion due to legacy REST/HTTP polling intervals (where client devices check in every 15–60 minutes), resulting in delayed policy enforcement, vulnerable windows during active classroom sessions, and severe database contention during morning start-of-day surges. IMTLazarus fundamentally redesigns device governance by implementing an **Event-Driven Real-Time Push Architecture** powered by persistent, bidirectional **WebSockets** (**gorilla/websocket over Go 1.21+**) paired with stateless, elastic microservices (**PHP 8.1+ / Laravel 10+**).

1.1 Core Scalability & Service Capacity Benchmarks

- **High-Concurrency Ingress Multiplexing:** A single Go WebSocket router node multiplexes well over 100,000 persistent WSS connections with an idle memory footprint of under 4 KB per socket. In a clustered private VPC deployment, the ingress layer scales horizontally via automated pod replication across redundant availability zones, providing elastic headroom for millions of concurrent student endpoints across thousands of distributed educational campuses nationwide.

- **Sub-100ms Policy Propagation:** Administrative policy modifications published in the **IMTManager** central command console propagate across nationwide channels and enforce locally on classroom devices in less than 100 milliseconds.

- **Start-of-Day Peak Load Absorption:** During morning enrollment and network reconnection surges (8:00 AM – 9:00 AM), asynchronous telemetry processing engines (**IMT_Server Go**) ingest exceeding 10,000 audit and safeguarding events per second across database read replicas without impacting REST API responsiveness.

- **Autonomous Offline Resilience:** When regional WAN connectivity interrupts, edge endpoints transition instantly to local cache persistence

- (**SQLCipher AES-256 / CoreData**), buffering up to 50,000 local audit records for automatic, compressed post-reconnection batch upload (**POST /api/events/batch**).

2. COMPREHENSIVE 4-TIER ARCHITECTURAL TOPOLOGY (**ANÁLISIS TOPOLÓGICO POR CAPAS**)

```
graph TB
    subgraph Layer1 [Layer 1: Edge Enforcement & Operating System Integration Tier]
        A1["Android Sub-System (IMTGo)"]
        A1 --- S1["Samsung Knox SDK & Guard Integration"]
        A1 --- S2["Android Enterprise Device/Profile Owner"]
        A1 --- S3["Hardware Button & Kiosk Lock ('LockTask')"]
        A2["iOS / iPadOS Sub-System (IMTGo)"]
        A2 --- S4["Swift 5+ / SwiftUI Native Client"]
        A2 --- S5["NEFilterDataProvider NetworkExtension"]
        A2 --- S6["WKWebView Packet Interception"]
        A3["Windows 11 Sub-System (IMT_Agent / IMTWin)"]
        A3 --- S7["Native C++ Win32 / NT Kernel Hooks"]
        A3 --- S8["Windows Filtering Platform (WFP) Drivers"]
        A3 --- S9["Watchdog Process Protection ('PROCESS_TERMINATE')"]
        A4["Chrome / Firefox Sub-System (IMTLazarusv3)"]
        A4 --- S10["Manifest V3 / WebExtensions Architecture"]
        A4 --- S11["declarativeNetRequest Dynamic Filtering"]
        A4 --- S12["Local On-Device AI Contextual Engine"]
    end

    subgraph Layer2 [Layer 2: Real-Time Communication & High-Concurrency Ingress Router]
        WSS["High-Concurrency WebSocket Server (Go 1.21+)"]
        WSS --- S13["Epoll/Kqueue Multiplexing (Over 100k sockets/node)"]
        WSS --- S14["Persistent Push Broadcast Router ('wss://')"]
        WSS --- S15["HMAC-SHA256 JWT Verification & Leaky-Bucket Rate Limit"]
        WSS --- S16["Auto-Scaling Horizontal Ingress Pod Cluster"]
    end

    subgraph Layer3 [Layer 3: Core Backend Orchestration & Elastic Microservices]
        API["IMTLazarusPHP Core Platform (PHP 8.1+ / Laravel 10+)"]
        API --- S17["OpenAPI / Swagger 3.0 REST API Orchestrator"]
        API --- S18["Multi-Tier Hierarchy Resolver ('Central -> School')"]
        API --- S19["Cryptographic Enrollment ('POST /api/device/enroll')"]
        API --- S20["OAuth 2.0 Federation ('Google / Entra ID')"]
        GOSVR["IMT_Server High-Throughput Processor (Go 1.21+)"]
        GOSVR --- S21["Goroutines & Buffered Channel Processing Engine"]
        GOSVR --- S22["High-Speed Asynchronous Log Ingestor"]
        GOSVR --- S23["Telemetry Aggregation & Multi-Tenant Sync"]
    end

    subgraph Layer4 [Layer 4: Administration & High-Availability Relational Storage]
        MGR["IMTManager Single-Pane-of-Glass Console"]
        MGR --- S24["Vue.js / Laravel Blade SPA Real-Time Dashboard"]
        MGR --- S25["Multi-Scope Role-Based Access Control • Sub-Second Remote Wipe / Lockdown Trigger"]
        DB["MySQL 8.0+ / MariaDB Clustered Relational Engine"]
        DB --- S26["Master-Slave Group Replication (InnoDB Engine)"]
        DB --- S27["Range-Based Table Partitioning (Over 10M rows/day)"]
        DB --- S28["Encrypted Tablespaces ('AES-256') & ACID Integrity"]
    end

    A1 <--> WSS
    A2 <--> WSS
    A3 <--> WSS
    A4 <--> WSS
    WSS <--> API
    WSS <--> GOSVR
    API <--> MGR
    API <--> DB
    GOSVR <--> MGR
    GOSVR <--> DB
    MGR <--> DB
```

A1 <-->|Persistent WSS Push / TLS 1.3| WSS
A2 <-->|Persistent WSS Push / TLS 1.3| WSS
A3 <-->|Persistent WSS Push / TLS 1.3| WSS
A4 <-->|Persistent WSS Push / TLS 1.3| WSS
WSS <-->|Authenticated Internal HTTP Push API / Event Bus| API
A1 -->|Signed REST API ('POST /api/events' / 'GET /api/policy')| API
A2 -->|Signed REST API ('POST /api/events' / 'GET /api/policy')| API
A3 -->|Signed REST API ('POST /api/events' / 'GET /api/policy')| API
A4 -->|Signed REST API ('POST /api/events' / 'GET /api/policy')| API
API <-->|RPC / Shared Memory Engine| GOSVR
API <-->|Transactional CRUD & Scoped Queries| DB
GOSVR <-->|High-Speed Batch Write ('Exceeding 10k events/sec')| DB
MGR <-->|Integrated Frontend & WebSocket Telemetry Feed| API

2.1 Deep-Dive: Layer 1 Edge Enforcement & Operating System Integration Tiers (**Capa de Clientes / Dispositivos**) •

- Android Sub-System (**IMTGo for Android**): Engineered in **Kotlin** leveraging the **Samsung Knox Customization SDK**, **Knox Service Plugin (KSP)**, and **Android Enterprise Device Owner APIs**. It enforces **LockTask** kiosk mode, disables hardware reset combinations (**HOME / RECENTS / POWER**), blocks USB OTG/MTP data transfers, and intercepts socket traffic at the kernel level (**VpnService**). A self-healing watchdog daemon prevents user kill attempts via ADB or safe mode.
- iOS / iPadOS Sub-System (**IMTGo for Apple Ecosystem**): Developed in **Swift 5+ / SwiftUI** combining an **Apple Supervised MDM Profile (APNs + MobileDeviceManagement API)** with a system-wide **NetworkExtension (NEFilterDataProvider / NEFilterControlProvider)**. All Safari, app, or **WKWebView** traffic passes through the local filter, preventing student bypass via VPN applications, proxy configurations, or DNS-over-HTTPS (**DoH**) manipulation.
- Windows 11 Sub-System (**IMT_Agent / IMTWin Native Service**): High-performance native **C++ (Win32 API / NT Kernel Hooks / CMake)** background service operating with kernel callout drivers via the **Windows Filtering Platform (WFP)**. It intercepts traffic prior to winsock delivery, enforces local group policies (**HKLM\Software\Policies**), monitors process execution (**CreateProcessNotifyRoutine**), and applies sub-second remote screen lock/wipe commands with anti-termination self-defense.
- Chrome / Firefox Sub-System (**IMTLazarusv3 Extension**): Built on **Manifest V3 / WebExtensions** utilizing persistent background Service Workers and **chrome.declarativeNetRequest** to evaluate over 100,000 dynamic regex/domain rules in memory. Embeds an **On-Device Artificial Intelligence (AI) Contextual Engine** that vector-classifies typed text and page content in real time for self-harm and cyberbullying without transmitting raw keystrokes to third-party clouds.

2.2 Deep-Dive: Layer 2 Real-Time Communication & High-Concurrency Ingress Router (**Capa WebSocket WSS**)

- Go WebSocket Router (**gorilla/websocket over Go 1.21+**): Epoll/kqueue non-blocking multiplexing managing well over 100,000 concurrent sockets per node with under 4 KB idle RAM. In physical deployment across bare-metal private clouds (**Zenlayer VPC**), ingress nodes sit behind redundant Layer 4 load balancers, scaling horizontally via automated pod replication to support millions of concurrent student endpoints across nationwide deployments. • **Connection Governance**: Maintains low-overhead **Ping/Pong** heartbeats (**interval: 30s**) and executes **Exponential Backoff with Jitter** reconnection algorithms upon network recovery. Requires **HMAC-SHA256** signed JWT validation upon **Upgrade: websocket** handshake, isolating channels by tenant UUID.

2.3 Deep-Dive: Layer 3 Core Backend & Layer 4 Clustered Storage Tier

- IMTLazarusPHP & Hierarchy Engine (**Laravel 10+ Enterprise**): OpenAPI/Swagger 3.0 REST orchestrator implementing a mathematical inheritance model (**Central -> Region -> Division -> School -> Group -> Device**). Resolves policy conflicts using a **Merge-by-Priority Algorithm**: baseline mandates established at Central/Division tiers are locked and cannot be overridden by School-level staff. • **IMT_Server High-Throughput Processor (Go 1.21+ Concurrency Engine)**: Offloads heavy log ingestion (**POST /api/events**), processing exceeding 10,000 events/sec across MySQL read replicas without API degradation during morning connection surges.
- MySQL 8.0+ Clustered Storage Tier**: Master-Slave Group Replication topology (**InnoDB Engine**). To maintain sub-second query performance across millions of historical records, audit tables enforce automated **Range-Based Table Partitioning** by date and tenant UUID, with **AES-256** encrypted tablespaces.

3. EXHAUSTIVE PROTOCOL HANDSHAKES & DATA FLOW MECHANICS (**PROTOCOLOS Y FLUJOS**

TÉCNICOS) 3.1 Cryptographic Device Enrollment & Identity Handshakeflowchart TD

Start([Device Powered On / Enrollment Initiated]) --> EnrPOST[Transmit HTTP POST /api/device/enroll via TLS 1.3] EnrPOST --> Payload[Payload: Hardware UUID, MAC, OS Build, IMT Enrollment Code, Knox/APNs Status] Payload --> ValCode[Laravel API Validates Code & Tenant Scope] ValCode -- Code Invalid / Expired --> Reject[Return HTTP 401 Unauthorized / Drop Handshake] ValCode -- Valid Code & Scope --> Register[Insert Device UUID in MySQL & Assign Hierarchy Scope] Register --> GenJWT[Issue Cryptographically Signed HMAC-SHA256 JWT Token & WSS Router URL] GenJWT --> WSSHHandshake[Device Initiates wss://*.imtlazarus.com/ws Upgrade with JWT Header] WSSHHandshake --> AuthWSS[Go WSS Verifies Token Signature & Expiration] AuthWSS -- Signature Failed --> DropWSS[Terminate Socket Connection] AuthWSS -- Verified --> BindChannel[Bind Socket to Tenant Broadcast Channel & Register Session] BindChannel --> PushBase[Go WSS Pushes Initial Baseline Policy Payload ('policy_update')] PushBase --> Enforce[Device Applies URL Filters, App Locks & Safeguarding Rules Locally] Enforce --> Complete([Device Fully Enrolled & Managed in Real Time])

Enrollment Request Payload & Issued JWT Schema (Courier Formatted JSON Header & Body):

```
// HTTP POST /api/device/enroll (TLS 1.3 / Port 443) | Headers: X-IMT-Client-Version: 3.4.12 {
  "enrollment_code": "DEPED-DIV04-SCH1042",
  "hardware_uuid": "e8a7d4f2-9c1b-4f8e-a3b2-6d5c8e1f0a9b",
  "device_name": "DEPED-TAB-S9-2026-0042",
  "os_type": "ANDROID",
  "os_version": "14.0 (API 34)",
  "knox_sdk_version": "3.10",
  "supervised_status": true,
  "mac_address_hash": "a4f89d1c7e6b2301ef84729104857201a8f9c0e1"
}
```

```
// Issued Cryptographic HMAC-SHA256 Signed JWT Header & Claims Structure
{
  "alg": "HS256", "typ": "JWT"
} . {
  "iss": "https://auth.imtlazarus.com/api/v3",
  "sub": "e8a7d4f2-9c1b-4f8e-a3b2-6d5c8e1f0a9b",
  "aud": "imtlazarus-wss-router-cluster",
  "iat": 1784687000, "exp": 1784773400,
  "claims": {
    "tenant_uuid": "7f8c9b0a-1d2e-3f4a-5b6c-7d8e9f0a1b2c",
    "entity_scope": "DIVISION_04/SCHOOL_1042",
    "role": "DEVICE_CLIENT",
    "permissions": ["POLICY_PUSH_SUB", "EVENT_POST", "TELEMETRY_STREAM"]
  }
}
```

3.2 Zero-Polling Real-Time Policy Propagation CascadesequenceDiagram

autonumber
actor Admin as Central / School Administrator
participant MGR as IMTManager Console
participant REST as IMTLazarusPHP (REST API)
participant DB as MySQL 8.0+ Cluster
participant WSS as Go WebSocket Server
actor Device as Enrolled Classroom Device
Admin->>MGR: Modify Web Filtering Rule / Classroom Policy & Click "Publish Policy"
MGR->>REST: PUT /api/policy/{policy_id} (Updated Ruleset Payload via OAuth JWT)
REST->>DB: Validate Hierarchy Scope, Commit Policy & Write Audit LogDB->>REST: Transaction Committed (Returns Updated `sha256\_etag`)
REST->>WSS: Internal HTTP POST Hook / Broadcast Trigger ('entity_id', 'policy_id', 'etag')
WSS->>Device: Instant WSS Push Frame: `{"event": "policy\_update", "policy\_id": 1042, "etag": "8f9a2b..."}` Device->>REST: GET

/api/policy/1042?etag=8f9a2b... (Fetch Full Policy via Signed JWT)
REST-->>Device: HTTP 200 OK (Complete Filtering, App Lock & Kiosk JSON Payload)
Device->>Device: Atomically Swap In-Memory & SQLite Cache \$
ightarrow\$ Enforce Instantly

Real-Time Push Frame & Policy Ruleset Schema (*Courier Formatted JSON*):

```
// Step 4: Minimal WSS Push Notification Frame (Sent over WebSocket wss://)
{
  "event": "policy_update",
  "policy_id": 1042,
  "scope": "DIVISION_04/SCHOOL_1042",
  "sha256_etag": "8f9a2b4c6d8e0f1a2b3c4d5e6f7a8b9c0d1e2f3a",
  "timestamp": 1784687200,
  "priority": "IMMEDIATE_ENFORCE"
}

// Step 6: Full Policy Ruleset Payload (Returned by GET /api/policy/1042)
{
  "policy_id": 1042,
  "version": "3.4.0-REV88",
  "web_filtering": {
    "mode": "WHITELIST_AND_CURATED_BLACKLIST",
    "blocked_categories": ["ADULT_CONTENT", "GAMBLING", "MALWARE_PHISHING", "SOCIAL_MEDIA"],
    "enforce_safe_search": true,
    "youtube_restriction": "MODERATE_EDUCATIONAL"
  },
  "app_governance": {
    "kiosk_mode": false,
    "blocked_package_names": ["com.facebook.katana", "com.zhilioapp.musically", "com.supercell.clashofclans", "mandatory_apps":
    ["com.google.android.apps.classroom", "org.deped.ph.lrmds"]
  }
}
```

3.3 Asynchronous Telemetry Ingestion & Audit Persistence Loopflowchart TD

Event([Edge Event Detected: Web Block / AI Safeguarding Alert / Error]) --> BuildPayload[Build Structured JSON Telemetry Payload with HMAC Signature]
BuildPayload --> Transmit[Transmit HTTP POST /api/events via TLS 1.3]
Transmit --> Ingest[IMTLazarusPHP & Go IMT_Server Ingest Endpoint]
Ingest --> Sanitize[Sanitize, Validate & Enrich Event with Tenant Hierarchy Meta]
Sanitize --> BatchWrite[Go Routine Buffered Channel: Batch Insert into MySQL Cluster] Sanitize --> LiveWSS[Push Instant Alert
Notification via WSS to Active Admin Consoles] BatchWrite --> Persisted([Immutable Timestamped Audit Trail Persisted in
Database])
LiveWSS --> AdminView([Dashboard Visualizes Live Incident Indicator in Real Time])

Telemetry Event Ingestion Schema (*Courier Formatted JSON*):

```
// HTTP POST /api/events (TLS 1.3 / Asynchronous Batching)
{
  "event_id": "c9d8e7f6-5a4b-3c2d-1e0f-9a8b7c6d5e4f",
  "timestamp": 1784687245,
  "device_uuid": "e8a7d4f2-9c1b-4f8e-a3b2-6d5c8e1f0a9b",
  "tenant_uuid": "7f8c9b0a-1d2e-3f4a-5b6c-7d8e9f0a1b2c",
  "event_type": "SAFEGUARDING_AI_ALERT",
  "severity": "CRITICAL_HIGH",
  "details": {
    "category": "SELF_HARM_INDICATOR",
    "matched_keyword_or_regex": "local_ai_model_trigger_code_402",
    "target_application": "Chrome Browser (Manifest V3)",
    "target_url": "https://example-unauthorized-chat.org/room/general",
    "ai_confidence_score": 0.94,
    "action_taken": "INSTANT_LOCAL_BLOCK_AND_SCREEN_CAPTURE_LOCKED"
  }
}
```

4. HIGH-AVAILABILITY FAIL-SAFE & OFFLINE DEGRADATION MECHANICSflowchart TD

Online([State: ONLINE_CONNECTED]) --> CheckConn{Network / WSS Heartbeat Lost?}
CheckConn -- Heartbeat OK --> Online
CheckConn -- Heartbeat Timeout ('3x Ping Missed') --> Offline([State: OFFLINE_CACHED_ENFORCEMENT]) Offline -->
LocalStore[Lock & Enforce Last-Known Policy from Encrypted SQLite/CoreData Cache]
LocalStore --> BufferQueue[Write Local Telemetry Logs to Circular Buffer Queue ('Max: 50,000 Events')] BufferQueue -->
PollRecover{Physical Network Restored?}
PollRecover -- No --> LocalStore
PollRecover -- Yes --> Reconnect[Initiate Exponential Backoff WSS Handshake with Signed JWT]
Reconnect --> VerifyETag{Compare Server ETag vs Local Policy ETag}
VerifyETag -- ETag Match ('No Policy Change') --> FlushOnly[Batch POST Buffered Logs to /api/events ('gzip')] VerifyETag -- ETag Mismatch ('New Policy
Published') --> PullPolicy[Download Updated Policy via GET /api/policy & Flush Logs]
FlushOnly --> Online
PullPolicy --> Online

4.1 Local Storage Cryptography & Reconnection Buffer

- Encrypted Cache Tier:** Each edge client maintains an encrypted local relational store (**SQLite with SQLCipher AES-256** on Android/Windows, **CoreData Encrypted Store** on iOS, and **chrome.storage.local** with HMAC checksum verification on Chrome/Firefox).
- Offline Protective Lock:** Upon detecting three consecutive missed WebSocket heartbeats (exceeding 90 seconds), the client switches autonomously to **OFFLINE_CACHED_ENFORCEMENT** mode. All web access, application executions, and hardware controls continue to be evaluated strictly against the cached ruleset.
- Fail-Safe Restrictiveness Protocol:** If an edge client detects internal database corruption, tampering attempts against the SQLite cache, or fatal memory segmentation within the filtering daemon, the state machine triggers **EMERGENCY_RESTRICTIVE_LOCKDOWN** . In this state, all external network egress and non-system application launches are blocked entirely until an authorized technician enters an administrative recovery PIN or cloud synchronization is successfully re-verified.
- Reconnection Buffer Synchronization:** While disconnected, the client buffers up to 50,000 timestamped audit events in a circular local queue. Upon network recovery, the client re-establishes its TLS 1.3 WSS handshake, performs an atomic ETag comparison with the server to pull any pending policy updates, and transmits its buffered log payload via compressed **gzip** batching (**POST /api/events/batch**) using exponential backoff to prevent network congestion.

5. ENTERPRISE SECURITY, CRYPTOGRAPHIC & PRIVACY COMPLIANCE STANDARDS (SEGURIDAD INTEGRAL

-).
- In-Transit & At-Rest Cryptography:** All network communications across public Internet, enterprise LAN, or regional WAN corridors use **TLS 1.3** (**HTTPS / WSS**) utilizing high-strength cipher suites (**TLS_AES_256_GCM_SHA384** & **TLS_CHACHA20_POLY1305_SHA256**) with Ephemeral Diffie-Hellman (**ECDHE**) perfect forward secrecy. All relational databases (**MySQL 8.0+ Cluster**), local client caches (**SQLite/CoreData**), and backup snapshots utilize **AES-256** block ciphers with HSM key rotation (**NIST SP 800-52 Rev. 2**).
- Multi-Tenant Row-Level Segregation (Tenant Isolation):** Every REST API endpoint (**IMTLazarusPHP**) injects a verified tenant scope into the database query builder (**App\Http\Middleware\TenantScope**). If a regional or division administrator queries device logs, the SQL execution engine automatically appends strict WHERE clauses (**WHERE tenant_uuid = ? AND hierarchy_path LIKE 'DIVISION_04/%'**), making cross-tenant unauthorized visibility mathematically impossible.
- Data Privacy & Statutory Alignment:** Organizational authorities (**Data Controller**) retain absolute legal and operational ownership over all student records, device identifiers, and safeguarding logs. IMTLazarus operates exclusively as a contracted **Data Processor** . Student profiles inside **MySQL** store minimal identifying metadata (**hashed student ID / school email**). Browsing logs and safeguarding alerts enforce automated historical pruning after designated retention windows (**30/60/90 days**).

6. COMPREHENSIVE TECHNOLOGY STACK & INFRASTRUCTURE SPECIFICATION MATRIX

Subsystem Component	Target Platform / Role	Programming Language	Core Framework / SDK / Library	Local / Remote Storage Tier	Network Protocol & Cryptographic Standard
IMTLazarusPHP	Core Backend & REST API Engine	PHP 8.1+	Laravel 10+ / OpenAPI / Swagger 3.0	MySQL 8.0+ Cluster (Master/Slave)	HTTPS (TLS 1.3 / REST JSON)
WebSocket Router	Real-Time Push Broadcast Server	Go 1.21+	gorilla/websocket / Epoll / Kqueue	Stateless / In-Memory Channel Hub	WSS (WebSocket Secure / TLS 1.3)
IMT_Server Engine	High-Throughput Log Ingestor	Go 1.21+	Go Concurrency / Channels / Goroutines	MySQL 8.0+ (Read Replicas & Partitions)	Internal RPC / Event Bus (Shared Memory)
IMTManager SPA	Unified Single-Pane Console	JS / PHP	Vue.js / Laravel Blade / Tailwind / Vite	MySQL 8.0+ (Session & Cache Engine)	HTTPS & Persistent WSS Telemetry Feed
IMTGo (Android)	Edge Client & Kiosk Browser Engine	Kotlin	Jetpack Compose / Samsung Knox Custom SDK	SQLite (SQLCipher AES-256 Cache)	HTTPS & WSS (Signed HMAC-SHA256 JWT)
IMTGo (iOS/iPadOS)	Edge Client & Packet Filter Engine	Swift 5+	SwiftUI / Apple NetworkExtension / WKWebView	CoreData Encrypted Store (AES-256)	HTTPS & WSS (Supervised APNs Handshake)
IMT_Agent/IMTWin	Windows 11 Ring 3 System Service	C++	Win32 API / NT Kernel Hooks / CMake / WFP	SQLite (SQLCipher AES-256 Cache)	HTTPS & WSS (Win32 System Socket Hub)
Chrome Extension	Manifest V3 Browser Filter Client	JS / ES6	Manifest V3 / chrome.declarativeNetRequest	chrome.storage.local (HMAC Checksum)	WSS Persistent Background Service Worker
Firefox Extension	WebBrowser Filtering Extension	JS / ES6	WebExtensions API / Background Scripting	browser.storage.local (AES-256)	WSS Persistent Background Service Worker
Cloud Infrastructure	Bare-Metal Private Cloud Tier	Terraform	Zenlayer Bare-Metal VPC Infrastructure	NVMe RAID-10 Storage Clusters	BGP / 10Gbps Redundant Network Fabric

7. OFFICIAL BRAND PRINCIPAL & SYSTEMS ENGINEERING CERTIFICATION

This engineering whitepaper and capacity dossier formally certifies the exact, production-validated architectural design of the **IMTLazarus Enterprise Cloud & Device Ecosystem**. By uniting high-concurrency **Go 1.21+ WebSockets**, robust **Laravel 10+ RESTful microservices**, and deep **OS-native edge enforcement daemons**, IMTLazarus establishes a mission-critical educational platform proven to deliver sub-100ms real-time governance and autonomous offline resilience across **millions of concurrent endpoints worldwide**, in absolute alignment with international cloud security standards (**ISO/IEC 27001:2022**) and enterprise scalability benchmarks.

Official Technical Architecture & Systems Engineering Specification — © 2026 IMTLazarus USA, LLC & Multifocus Corp. All Rights Reserved.