iMouseGuard – Full System Architecture Document

1. System Overview

iMouseGuard is an intelligent event-driven monitoring system built on top of ZoneMinder.
It captures real-time surveillance events, processes them via ZMES (ZoneMinder Event Server), forwards them through a Python WebSocket client, applies business rules, and sends alerts via Telegram, Slack, and WhatsApp.

Primary Goal:
Real-time intelligent alerting based on surveillance events.

2. High-Level Architecture Flow

```
ZoneMinder (Docker Container)
        ↓
ZMES (Perl WebSocket Event Server – Port 9000)
        ↓
Python WebSocket Forwarder (zmes_ws_to_telegram.py)
        ↓
Hook Script (imouse_hook_alert.py)
        ↓
Alert Channels (Telegram / Slack / WhatsApp)
```

3. Component Breakdown

A. ZoneMinder
- Captures surveillance events
- Stores events in database
- Generates alarm events

B. ZMES (zmeventnotification.pl)
- Listens to ZoneMinder events
- Publishes events over WebSocket
- Handles authentication
- Uses configuration from zmes_ws_only.ini

C. WebSocket Forwarder (Python)
- Connects to ws://127.0.0.1:9000
- Authenticates
- Subscribes to alarm events
- Parses EventId and MonitorId
- Invokes hook script per event

D. Hook Script

```
- Applies rule logic (rules.yaml)
- Formats messages
- Sends notifications via APIs

E. Environment Config (.env)
- Stores tokens and API keys
- Loaded before forwarder execution
```

4. Directory Structure

```
/opt/iMouseGuard
    ■■■ zmeventnotification/
    ■       ■■■ zmeventnotification.pl
    ■
    ■■■ iMouseGuard/
    ■       ■■■ bin/
    ■       ■     ■■■ zmes_ws_to_telegram.py
    ■       ■     ■■■ imouse_hook_alert.py
    ■       ■■■ config/
    ■       ■     ■■■ zmes_ws_only.ini
    ■       ■     ■■■ rules.yaml
    ■       ■■■ logs/
    ■       ■■■ .env
    ■       ■■■ venv/
```

5. Data Flow Details

```
1. Motion detected in ZoneMinder
2. ZMES detects new event via database polling
3. ZMES emits JSON payload via WebSocket
4. Forwarder receives JSON
5. parse_events() extracts:
   - EventId
   - MonitorId
   - Name
   - Cause
6. Forwarder calls hook:
   imouse_hook_alert.py <EventId> <MonitorId>
7. Hook applies rule engine
8. Alert sent via configured channels
```

6. Networking

```
WebSocket Port: 9000
Protocol: WS (SSL disabled in current configuration)
Forwarder connects to: ws://127.0.0.1:9000
```

Docker Environment:
ZoneMinder runs inside container.
ZMES and Forwarder run within same environment context.

7. Process Management (Current Setup)

ZMES started via:

```
perl -T zmeventnotification.pl --config=<path>
```

Forwarder started via:

```
source .env
source venv/bin/activate
python bin/zmes_ws_to_telegram.py
```

Background execution uses nohup with PID tracking.

8. Logging Strategy

ZMES Logs:
/opt/iMouseGuard/iMouseGuard/logs/zmes.log

Forwarder Logs:
/opt/iMouseGuard/iMouseGuard/logs/ws_forwarder.log

System monitoring:

```
ss -ltnp | grep :9000
ps aux | grep zmes
```

9. Failure Scenarios & Handling

- SSL enabled without cert → Fatal error
- Wrong EventId parsing → No alerts triggered
- Hook permission denied → chmod +x fix
- Missing JSON fields → parse_events extended
- WebSocket NOTSUPPORTED → reduce subscription types

10. Future Production Enhancements

- Separate Docker containers for:
  * ZoneMinder
  * ZMES
  * Forwarder

- docker-compose with restart: unless-stopped
- Healthcheck scripts
- Systemd services (if host-level deployment)
- Centralized logging (ELK / Loki)
- Rule-based severity levels
- Event deduplication logic