Zeek SIEM Architecture —

Subject: SoftwareArchitecture (SSZG653)

Presenter: Ashwin Jayaprakash

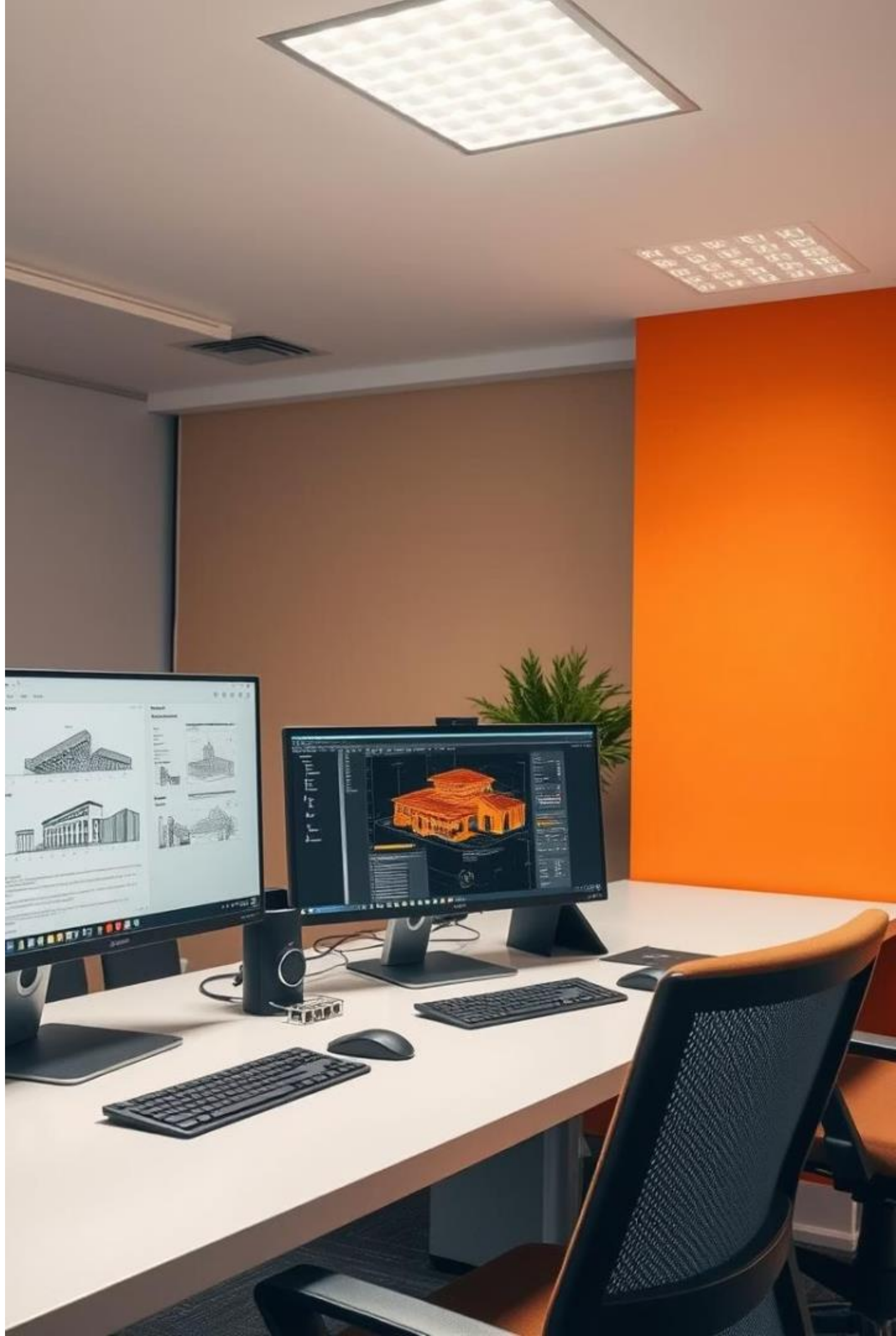
Email ID: 2024mt12030@wilp.bits-pilani.ac.in

Bits ID: 2024MT12030

Software Architecture

Zeek SIEM Machine

Zeek is **a powerful, open-source network security monitoring (NSM) and analysis framework that provides deep insights into network traffic**. It's often used for intrusion detection, forensic analysis, and network auditing. Unlike traditional security tools, Zeek doesn't actively prevent attacks but instead observes and analyzes network traffic, creating detailed logs and customizable outputs for security analysts.



Architecturally Significant Requirements (ASRs)

1 Performance

Must handle 10+ Gbps traffic to support high-throughput environments without packet loss.

2 Accuracy

Detect threats effectively with minimal false positives for reliable incident response.

3 Scalability

Ensure storage of petabytes of logs and scaling detection capabilities in cloud and on-prem clusters.

Tactics for ASR 1 - Performance

Zeek Cluster Mode

Distribute traffic load with manager and worker nodes coordinating capture and analysis in parallel.

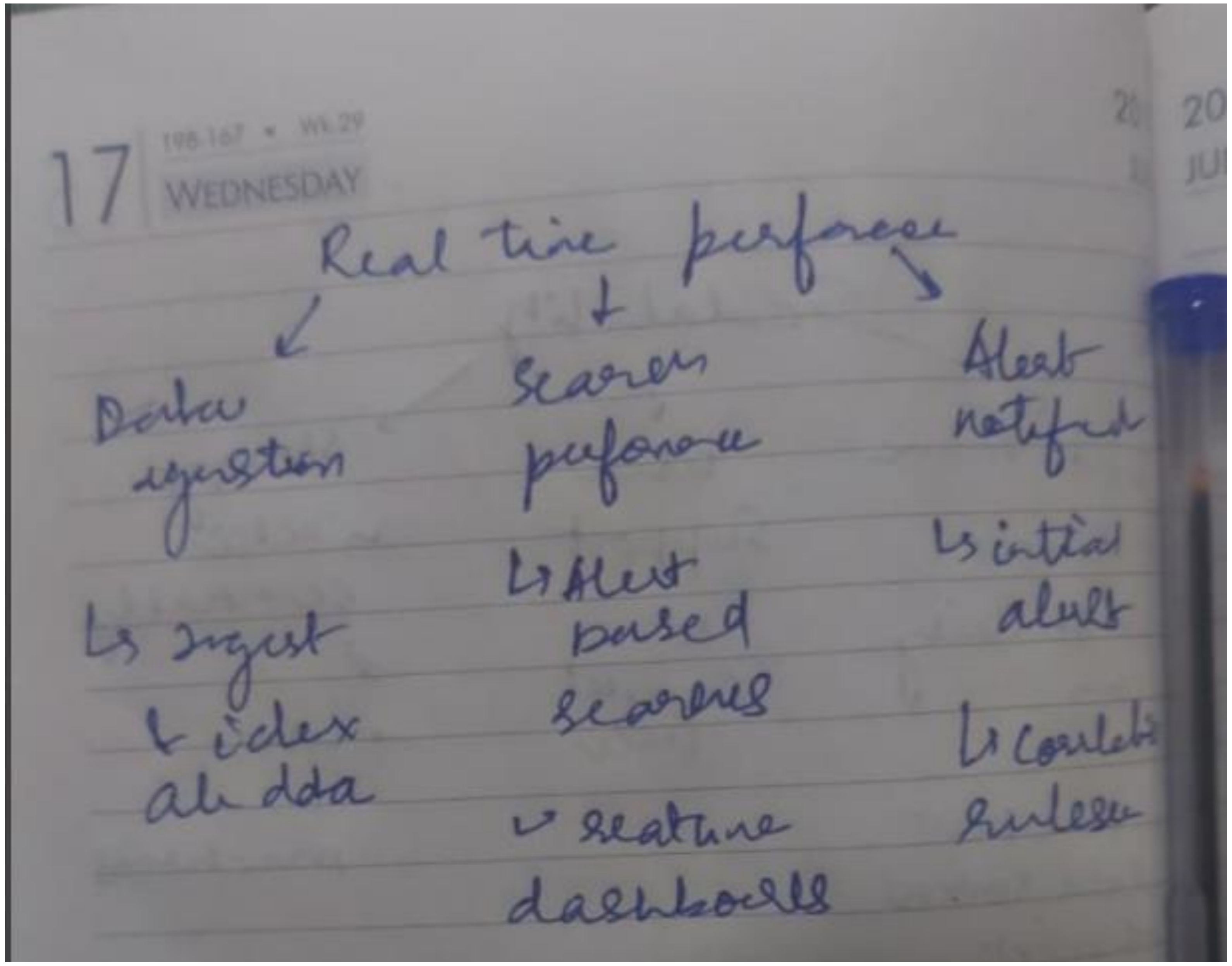
Load Balancing Techniques

Utilize PF_RING and hardware load balancers like Solarflare NICs to achieve line-rate packet capture.

High-Speed NICs

Deploy network interface cards optimized for high throughput and low latency in capture servers.

Tactics for ASR 1 - Performance



17 198-167 • Wk 29 WEDNESDAY

Real time performance

Data ingestion
↳ Ingest & index all data

Search performance
↳ Alert based searches
↳ realtime dashboards

Alert notification
↳ initial alerts
↳ Correlation rules

Tactics for ASR 2 - Accuracy

Custom Zeek Scripts

Tailor detection rules and protocol analyzers to specific network threat signatures.

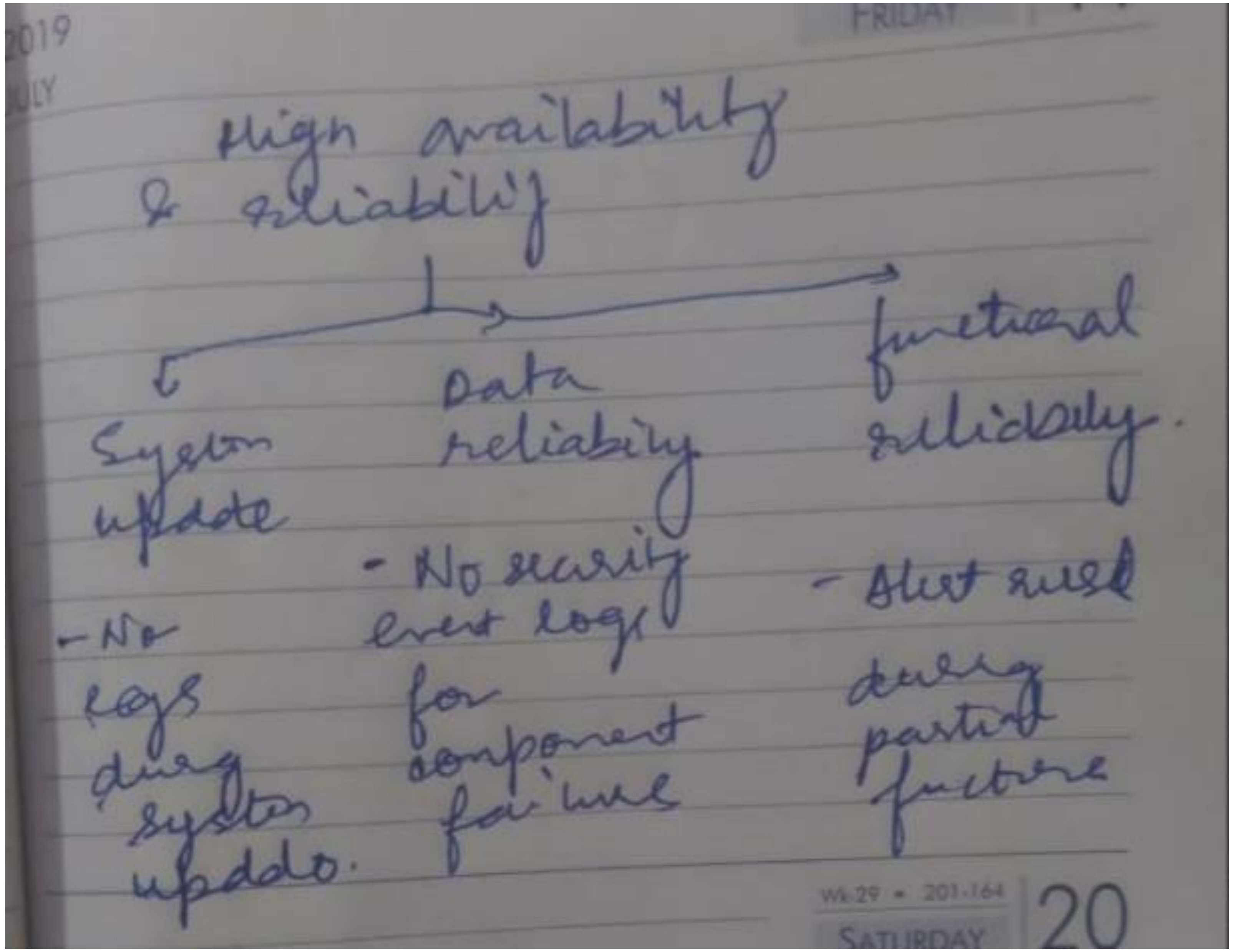
Threat Intelligence Integration

Incorporate updated external threat feeds to enrich detection with current indicators of compromise.

Post-Processing in SIEM

Use Elasticsearch for enrichment, correlation, and reducing false positives effectively.

Tactics for ASR 2 - Accuracy



High availability & reliability

System update
- No logs during system update.

Data reliability
- No security event logs for component failure

functional reliability.
- Alert rules during partial failure

Tactics for ASR 3 - Scalability

Kafka Ingestion

Handles high-volume, fault-tolerant log ingestion to decouple producers and consumers.

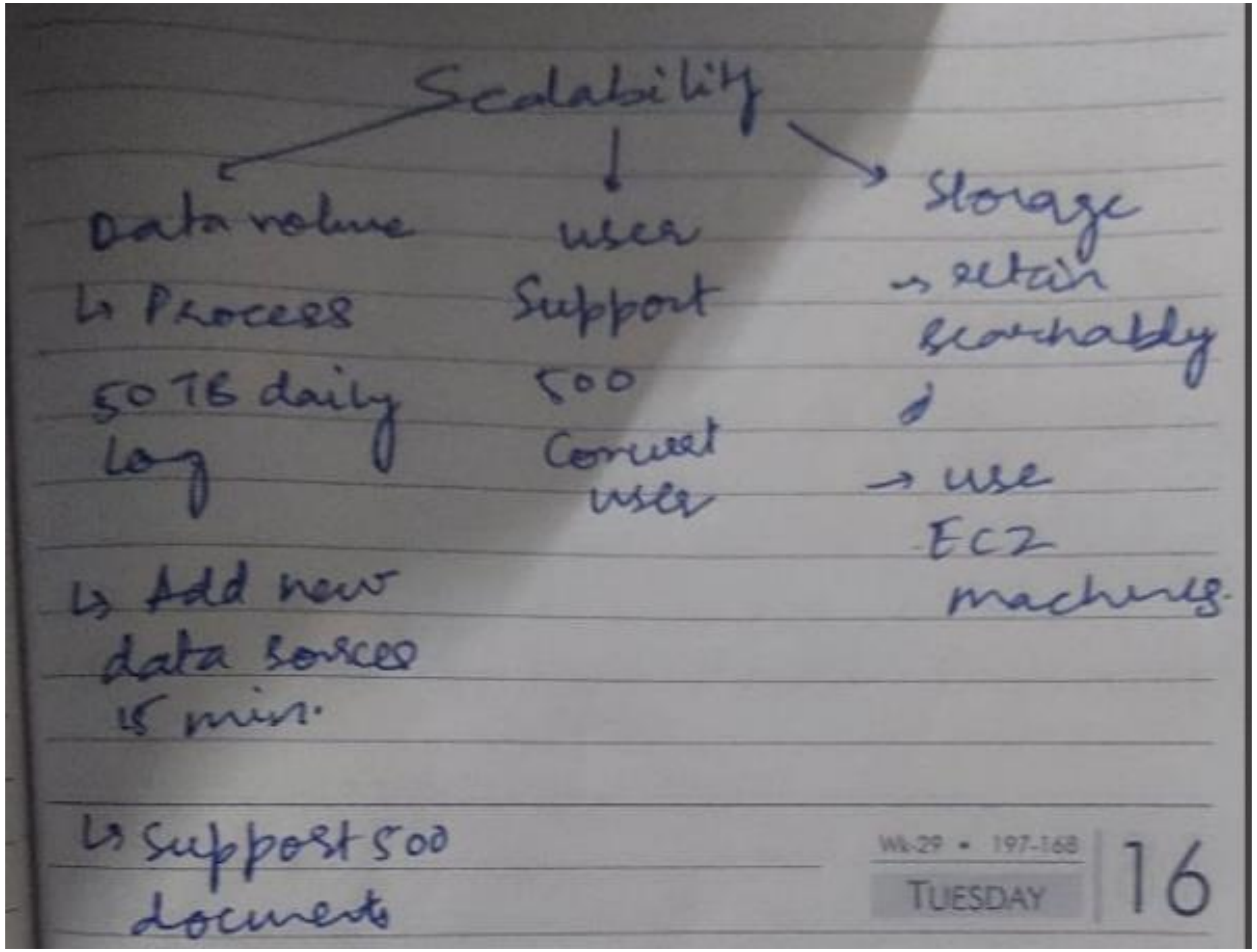
Distributed Storage

Elasticsearch and OpenSearch clusters provide scalable indexing and search capabilities.

Archival Solutions

S3-compatible storage like MinIO ensures cost-effective long-term log retention.

Tactics for ASR 3 - Scalability



Scalability

Data volume
↳ Process 50 TB daily log
↳ Add new data sources 15 min.
↳ Support 500 documents

user Support 500 Concurrent user

Storage
→ retain searchably
&
→ use EC2 machines.

Wk 29 • 197-168
TUESDAY
16

Component & Connection View

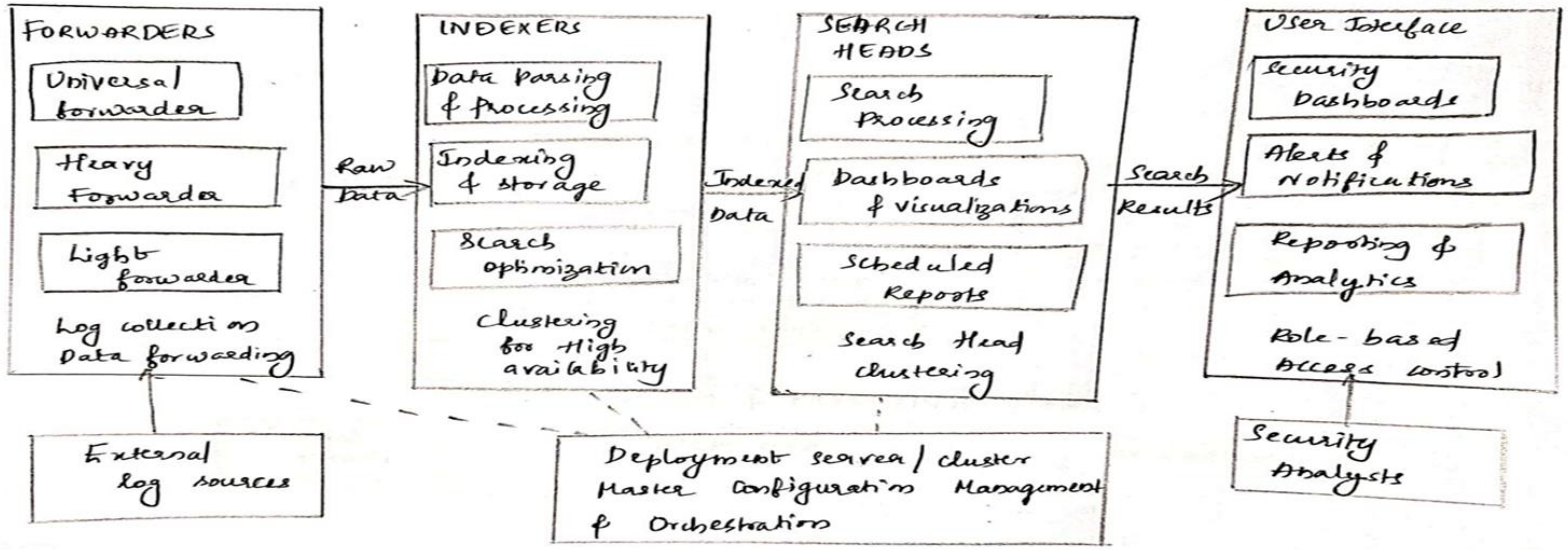
Components

- Zeek Sensors
- Kafka Messaging
- Logstash Processing
- Elasticsearch Storage
- Kibana Visualization

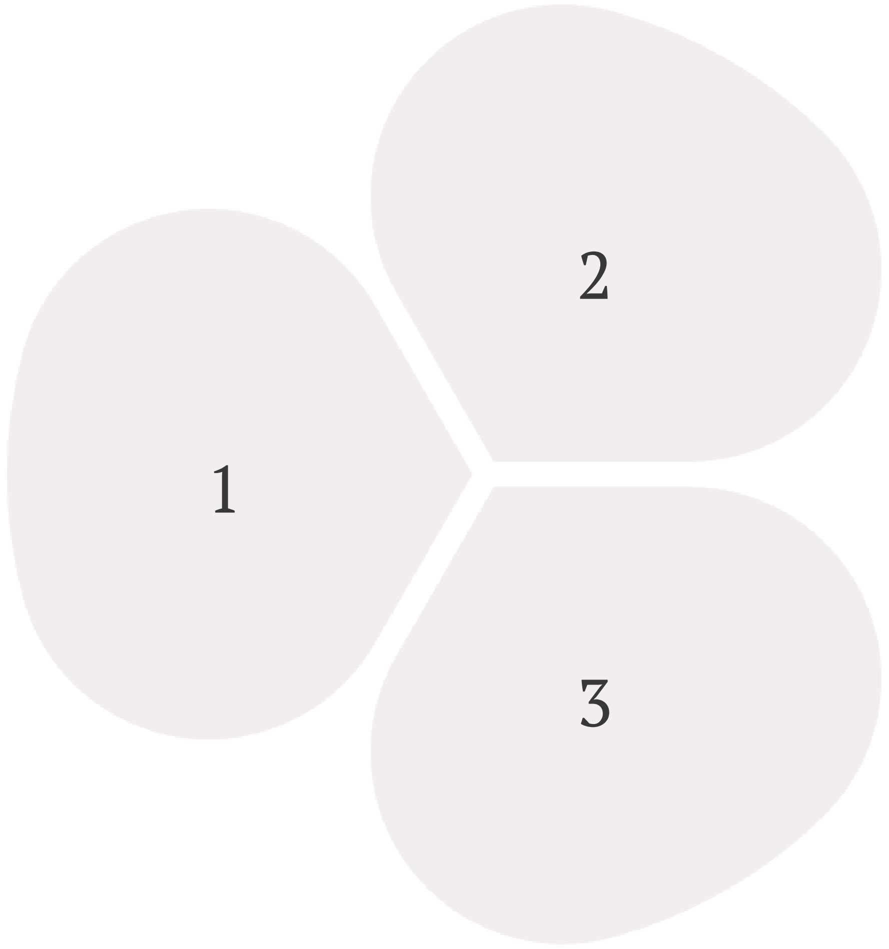
Communication

Zeek captures packets and sends JSON logs over Kafka. Logstash ingests and enriches data, forwarded via HTTP APIs to Elasticsearch, visualized through Kibana dashboards.

COMPONENT & CONNECTION DIAGRAM



Deployment View



On-Premise

Zeek deployed in clustered mode with workers and manager ensuring high capture throughput.

Cloud

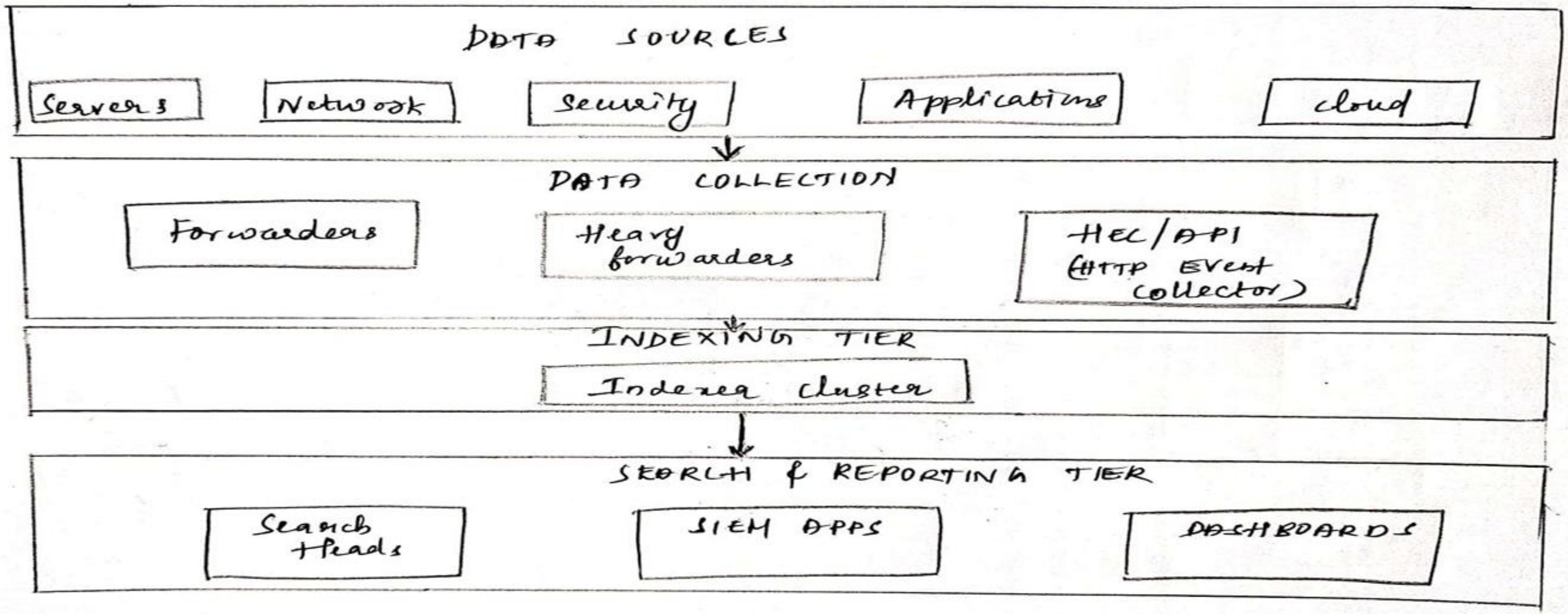
Kafka, Logstash, Elasticsearch, and Kibana reside in cloud infrastructure for scalable processing and access.

Analyst Clients

Users access Kibana dashboards through web browsers to monitor security alerts and investigate incidents.

Made with GAMMA

Deployment View



Sequence Diagram — Malicious DNS Alert

1

Capture

Real-time DNS request packet captured by Zeek sensor for analysis.

2

Detection

Zeek detects anomalous DNS behavior via scripted rules.

3

Forwarding

Alert data is pushed to Kafka message queue for processing.

4

Enrichment

Logstash enriches logs with context such as geoIP and threat info.

5

Indexing

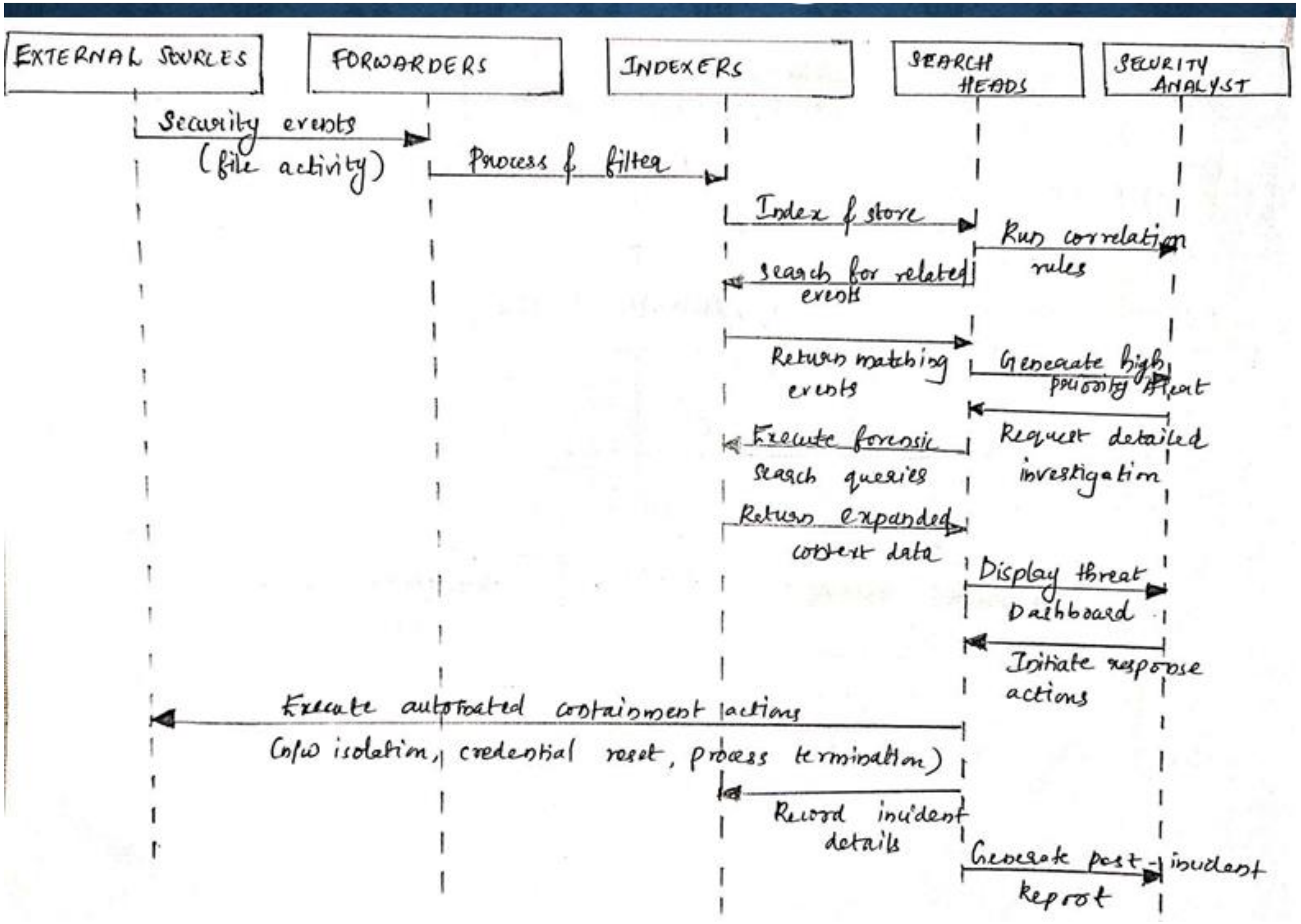
Elasticsearch indexes enriched data for search and analytics.

6

Visualization

Kibana dashboard displays alert for analyst investigation.

Sequence Diagram — Malicious file access Alert



EXTERNAL SOURCES
FORWARDERS
INDEXERS
SEARCH HEADS
SECURITY ANALYST
Security events (file activity)
Process & filter
Index & store
Run correlation rules
search for related events
Return matching events
Generate high priority Alert
Request detailed investigation
Execute forensic search queries
Return expanded context data
Display threat Dashboard
Initiate response actions
Execute automated containment actions (n/w isolation, credential reset, process termination)
Record incident details
Generate post-incident report

Architecture Patterns Used

Microservices Pattern

Decouples alerting, processing, and storage into independently deployable services, enhancing modularity and resilience.

Event-driven Architecture

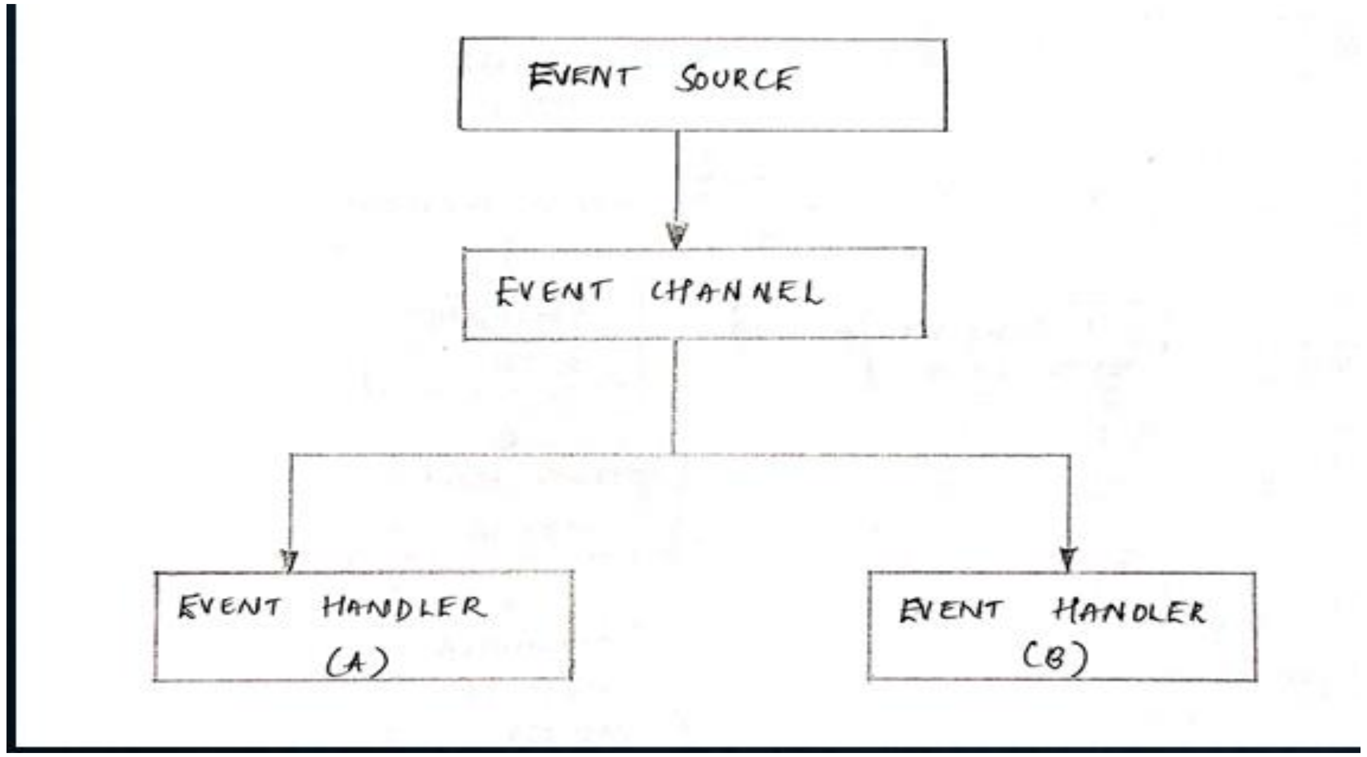
Kafka pipelines orchestrate asynchronous log flow enabling scalable, reliable, and real-time processing of security events.

Architecture Patterns Used

Event driven architecture

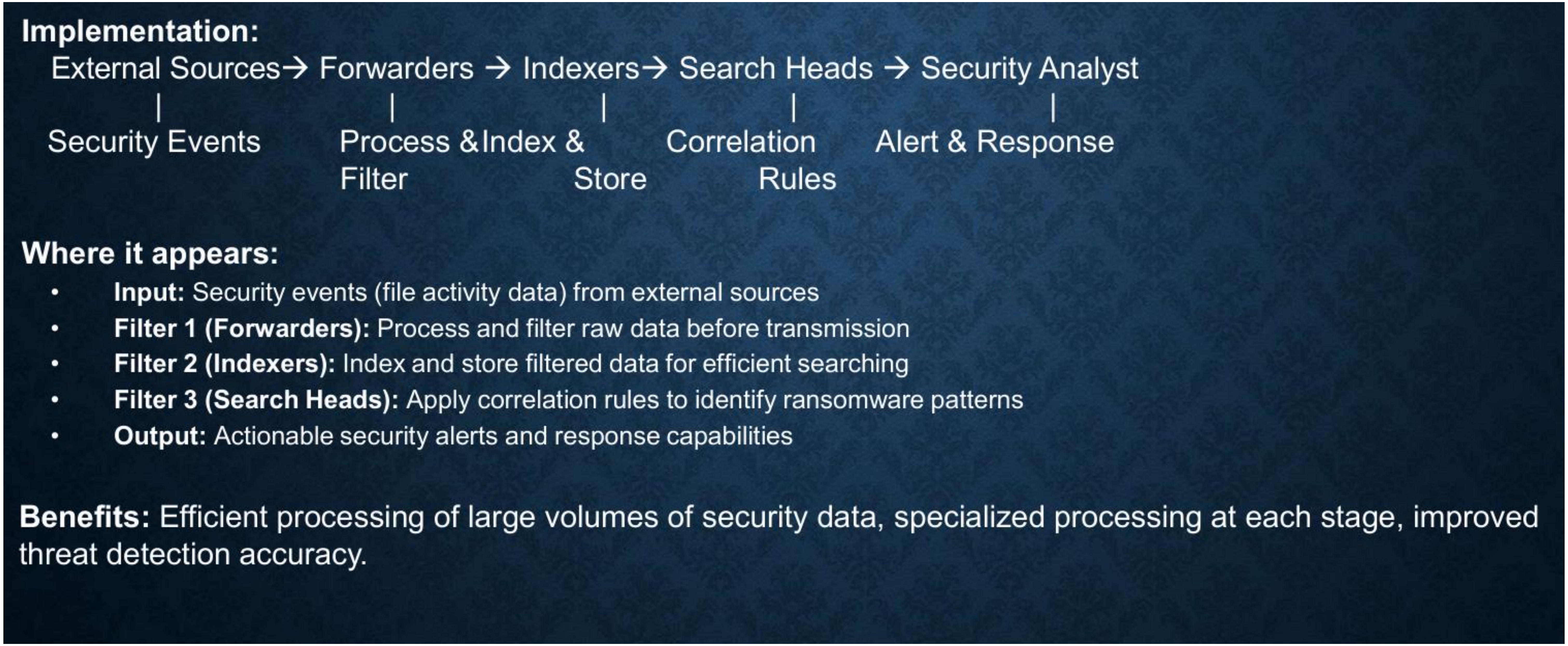
Definition: A software architecture pattern promoting the production, detection, consumption of, and reaction to events.

Key characteristics: • Components communicate through events rather than direct calls • Loose coupling between system components • Components react to events as they occur • Enables real-time processing and response



Made with GAMMA

Example – Pipes and filter in ransomware detection



Implementation:

```
External Sources→ Forwarders → Indexers→ Search Heads → Security Analyst
        |                |              |              |                 |
Security Events      Process &Index &     Correlation     Alert & Response
                     Filter        Store          Rules
```

Where it appears:

- **Input:** Security events (file activity data) from external sources
- **Filter 1 (Forwarders):** Process and filter raw data before transmission
- **Filter 2 (Indexers):** Index and store filtered data for efficient searching
- **Filter 3 (Search Heads):** Apply correlation rules to identify ransomware patterns
- **Output:** Actionable security alerts and response capabilities

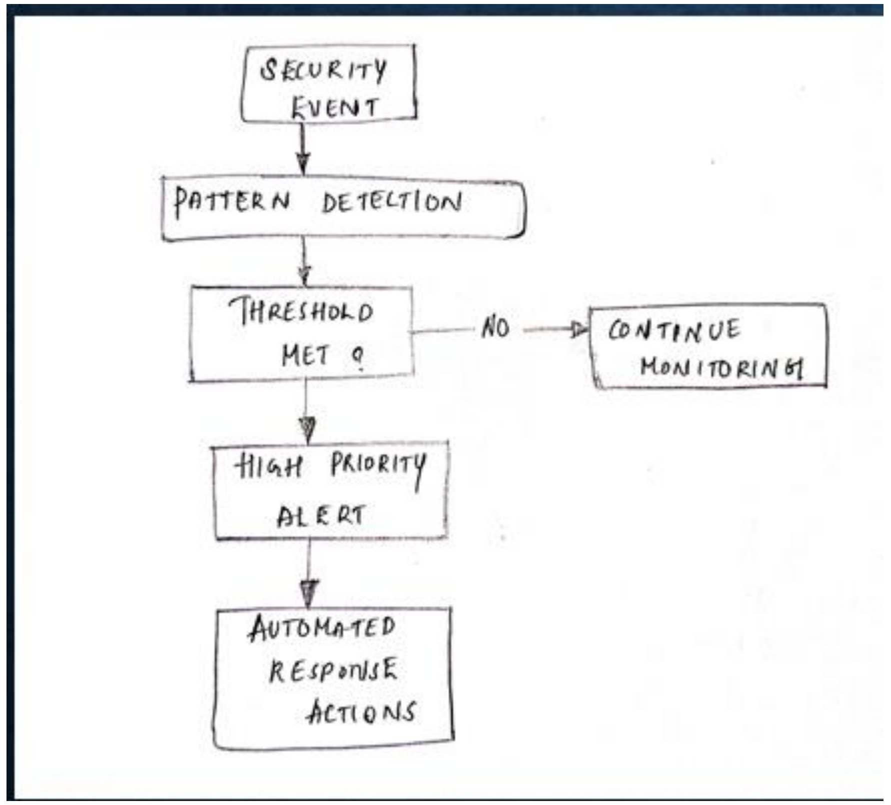
Benefits: Efficient processing of large volumes of security data, specialized processing at each stage, improved threat detection accuracy.

Event driven ransomware detection

Where it appears:

Event Sources:Detection of suspicious file encryption activities
Event Processing:Search Heads running correlation rules against event patterns

Benefits: Alert generation when ransomware patterns are detected Forensic investigation workflows Automated containment actions (network isolation, credential reset) Incident recording and reporting Implementation: Real-time threat detection, immediate response to security incidents, reduced attack impact through early intervention.



Key learnings

a. Balancing Real-time Performance with Scalability:
• Found balance between speed and handling large data volumes • Prioritized critical security events while maintaining system throughput
• Data collection design directly impacts downstream processing effectiveness

b. Security-Specific Availability Requirements:
• Security monitoring needs specialized availability approaches
• Both system uptime and data completeness are essential
• Implemented solutions for component redundancy and data integrity

c. Cost-Effective Resource Allocation in SIEM Architecture:
• Allocated resources based on data importance and lifecycle
• Used tiered storage for different data retention needs
• Balanced performance and cost through smart data routing

Thank you