Ref. Ares(2025)4891322 - 19/06/2025

expertware

Project: 101127939 – SIEMBIOT High Level Solution Design Document

Co-funded by the European Union

ECCC
EUROPEAN CYBERSECURITY COMPETENCE CENTRE

Document Control

Title	**SIEMBIOT High Level Design Document**
Client:	Internal
Reference:	2025-23
Version:	1.1
Release Date:	30/11/2023
Author:	Expertware, DNSC
Total Pages:	
Security Class:	Commercially Confidential
Distribution:	Restricted to EU experts and SIEMBIOT consortium members
Disclaimer:	The document or parts of it cannot be reused outside of SIEMBIOT project without the written approval form the authors

Document Approval

Approved By:	Lelcu Ion Didel	General Manager

Revision History

Version	Date	Issued By	Status	Comments (Update this history table for each version)
0.1	15/09/2023	Iulian Tarus	Draft	Initial layout, use case1
0.2	20/09/2023	Andrei Polocoseri	Draft	Added CTI MISP
0.3	10/10/2023	Liviu Sacu; Iulian Tarus	Draft	Added UC2, UC3, UC4
0.4	15/10/2023	Viorel Horga	Draft	Added Shared Services architecture
0.5	20/10/2023	Andrei Tarus	Draft	Virtualization, Networking, Storage
0.6	25/10/2023	Andrei Polocoseri, Viorel Horga	Draft	Design demo tenants
0.7	10/11/2023	Iulian Tarus, Andrei Tarus	Draft	Design Training and research platform
0.8	20/11/2023	All	Draft	Joint review
1.0	30/11/2023	All	Final	Published
1.1	18/06/2025	All	Draft	Added hypervisor farm details and security architecture

Contents

1 Introduction 5
1.1 Background 5
1.2 Scope 5
2 SIEMBIOT Use cases 6
2.1 [UC1] Demo SOCaaS for SMEs 6
2.1.1 UC1 Functional Requirements 7
2.1.2 UC1 Integration Requirements 8
2.1.3 UC1 Security requirements 8
2.1.4 UC1 Capacity & Performance requirements 9
2.1.5 UC1 Availability requirements 9
2.2 [UC2] Cyber security research data lake 10
2.2.1 UC2 Beneficiaries 10
2.2.2 UC2 Functional Requirements 10
2.2.3 UC2 Integration Requirements 10
2.2.4 UC2 Capacity and Performance Requirements 11
2.2.5 UC2 Availability Requirements 11
2.2.6 UC2 Security & compliancy requirements 11
2.3 [UC3] Cyber security training range 12
2.3.1 UC3 Functional requirements 12
2.3.2 UC3 Capacity and performance requirements 13
2.3.3 UC3 Security Requirements 13
2.3.4 UC3 Availability Requirements 13
2.4 [UC4] CTI exchange platform 13
2.4.1 UC4 Functional requirements 13
2.4.2 UC4 Capacity and performance requirements 13
2.4.3 UC4 Security and compliance requirements 14
2.4.4 UC4 Availability Requirements 14
2.5 Use cases relationships. 14
3 Platform Technical architecture 15
3.1 Architecture principles 15
3.2 Multi-tenancy 15
3.3 High level system architecture 17
3.4 Networking services architecture 18
3.5 Storage platform architecture 19
3.5.1 Storage HA design 20
3.5.2 Storage Scalability 20
3.6 Virtualization layer 20
4 Logical components architecture 22

4.1 SIEMBIOT Demo Tenants (SOCaaS) Platform 22
4.1.1 DNS configuration 22
4.1.2 IPv4 allocation 23
4.1.3 Platform sizing 23
4.1.4 High Availability Configuration 25
4.2 SIEMBIOT Shared Services Platform 28
4.2.1 SIEMBIOT shared services platform hosts components 28
4.2.2 IPv4 addresses allocation. 29
4.2.3 Platform sizing 30
4.2.4 CTI tools and integrations / feeds 31
4.2.5 Continuous Vulnerability Management system architecture 34
4.3 SIEMBIOT Training & Research 42
4.3.1 High Level Design 42
4.3.2 Training and Research Platform Sizing 42
4.3.3 Platform High Availability 42
4.4 Integrations / Traffic Flows 43
4.5 Security architecture principles 43
4.5.1 Tenants' data segregations principles 43
4.5.2 Data Privacy principles 44
4.5.3 Network traffic segregation 44
4.5.4 Authentication and Authorization 46
4.5.5 Security Protection 46
4.5.6 Audit features 47
4.5.7 GDPR considerations 47
5 Hosting infrastructure 48
5.1 Hypervisor host detailed diagram 50
5.2 Networking services deployment 51
5.3 Hypervisor storage connectivity 51
5.3.1 vSAN configuration 51
5.3.2 Shared SAN configuration 52
6 Platform security architecture 54
6.1 Physical security 54
6.2 Network traffic segregation 54
6.3 Communication flows 56
6.3.1 Securing data in transit 58
6.3.2 Securing data at rest 58
6.4 DevOps, IT Management and SOC analysts' access 58
6.5 Data protection 59
6.6 Security Monitoring 59

7 Web application security architecture 60
7.1.1 Web App Authentication Flows 60
7.1.2 Authentication Mechanisms 61
7.2 Authentication and Authorization Architecture (ASP.NET Core + JWT) 61
7.2.1 Overview 61
7.2.2 Authentication Workflow 61
7.2.3 Token Lifecycle and Validation 62
7.2.4 Authorization Strategy 62
7.2.5 Security Considerations 62
8 Glossary of Acronyms and Abbreviations 64

1 INTRODUCTION

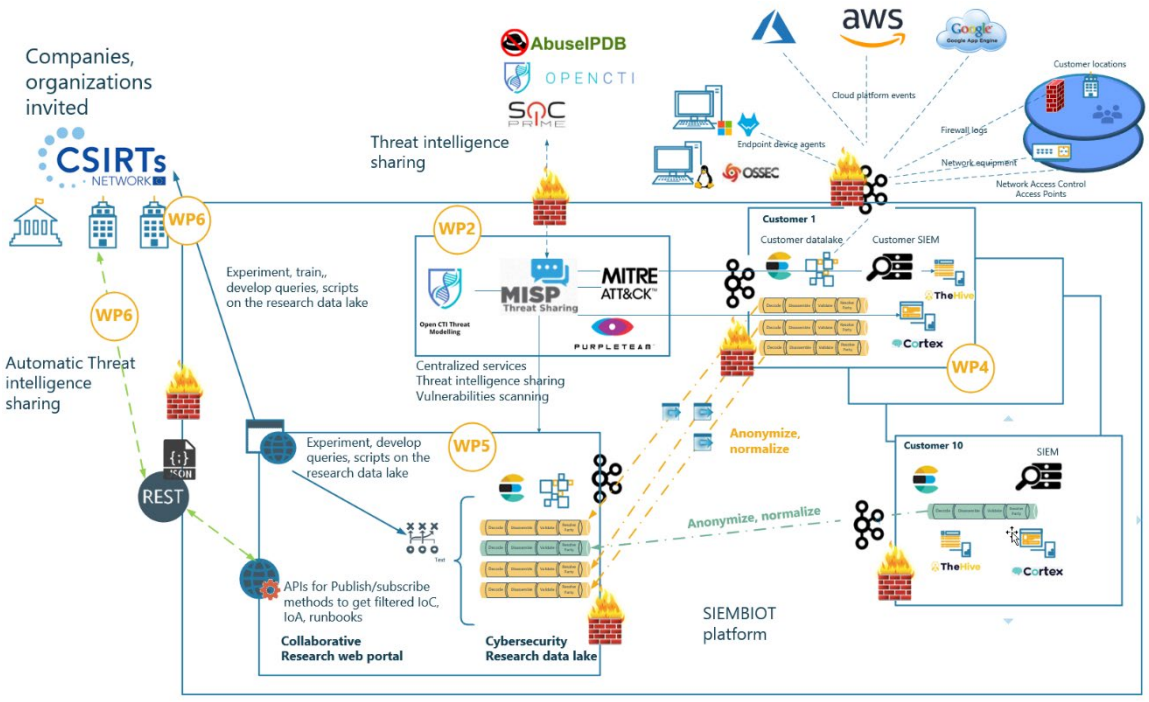
1.1 Background

The overall objective of the SIEMBIOT project is to increase national and regional resilience against cyber-attacks by creating a state-of-the art flexible, modular, and resilient defence and research platform.

SIEMBIOT represents an advanced, modular SIEM, NOC, SOC solution based on a mix of open-source technologies recommended by ENISA research, to enrich and facilitate collaboration among trusted EU entities. The solution accommodates complex IT landscapes distributed across multi-clouds and can leverage customer investments in existing commercial cybersecurity tools & products, integrating events and logs into a normalized and anonymized data lake. SIEMBIOT integrates seamlessly into the customer's IT infrastructure automatically collecting telemetry, events and alerts from different network inflexion points and devices, sending them securely to a SIEM collection platform and corelating them with live CTI feeds. Investigation of suspicions events are fed back to CTI partners and to our own collaboration research platform.

By integrating in this way, we will provide an advanced multi-angle view of any environment allowing for rich reporting and time-based analysis of cyber security threats, risks, and threat profiles.

Target platform



1.2 Scope

This document highlights SIEMBIOT use case requirements and the high-level design for the technical solution we will implement.

The high-level design will cover the architectural principles, technical components, sizing, availability, scalability, and security considerations.

2 SIEMBIOT USE CASES

Use case scenarios

Platform use cases refer to specific scenarios or situations in which SIEMBIOT platform is designed to be used. Use cases help define the practical and functional aspects of a platform by describing the tasks, actions, and interactions that users, systems, or components can perform within the platform. These use cases help clarify the purpose and functionality of the platform and serve as a basis for designing and testing the system.

Key points to consider when defining platform use cases:

- **Beneficiaries:** Describe who are the target beneficiaries for the SIEMBIOT platform.
- **Functional Requirements**: Each use case outlines the functional requirements of the platform. It describes what the platform is expected to do in response to specific actions or inputs. Use cases help define the system's behaviour in various scenarios.
- **Integrations / Data flows**: Use cases often illustrate the flow of actions and events within the platform. These could include integration with third-party APIs, or other software components.
- **Security Requirements**: Describe security requirements, such as authentication, authorization, and data protection. They help identify potential security threats and how they should be mitigated.
- **Capacity & Performance Requirements**: Use cases may specify performance expectations, such as response times or throughput requirements.
- **Availability Requirements**.
- **Scalability and Future Enhancements**: Consider scalability requirements and potential future enhancements when defining use cases. This helps ensure that the platform can grow and adapt to changing needs.
- **Regulatory and Compliance Requirements**: If the platform needs to adhere to specific industry regulations or compliance standards, use cases should address how these requirements are met.
- **Use Case Relationships**: Consider how different use cases relate to each other. Some use cases may be dependent on others, and understanding these dependencies is important for designing the platform's architecture.

2.1 [UC1] Demo SOCaaS for SMEs

Many small and medium organisations lack the skilled resources and the technical capabilities to be able to protect their IT systems and processes. Most organisation rely on isolated solutions like antivirus, patching of devices which provide a level of protection but are incomplete without a continuous security monitoring solution and a proper set of actions. SIEMBIOT aims to facilitate the adoption and administration of a consistent set of tools designed for security monitoring, security events recording, and complex cross platforms correlations required to improve visibility and help organisations act against cyber security threats. SIEMBIOT will deliver on one hand a shared threat intelligence platform that will

contain the core services common to all tenants, and individual SIEM platforms dedicated to each tenant.

The shared platform aim is to collect and consolidate threat intelligence using Structured Threat Information Expression (STIXs) machine readable language and the TAXII protocol services available on either commercial or open source vetted communities.

The individual tenants with connectivity into customer IT infrastructure will comprise a consistent set of virtual appliances and tools ready to use as well as a common portfolio of rules for detection and alerting, escalation flows and Incident Response ticketing systems, complex use cases definitions based on the MITRE Att&ck matrix.

The complexity of the security data extraction, consolidation and the alerts generation are transparent for the end-user organisation and access the up-to-date alert definitions and translation of new techniques and tactics provided to all tenants.

SOCaaS for SMEs include all the components which we deploy and use for each customer to collect relevant events, logs, metrics, index, normalize and analyse by security analysts (SOC teams). SIEMBIOT is by design meant to fit on almost any modern IT infrastructure without the need for major IT platform updates and its associated cost.

Beneficiaries:

A. Expertware
B. SMEs not having SIEM solutions onboarded on demo tenants (work package 4).

Example classification based on an average ratio of security events generated per day:

SME classification based on the number of items protected (users + devices)

SME Size	Cis	Events ingestion	Daily storage needs	12 month archive
Small [S]	0-99	<= 2 mil/day	~10 GB	~4 TB
Medium [M]	100-500	<= 20 mil/day	~30 GB	~12 TB
Large [L]	501-1000	<= 40 mil/day	~60 GB	~24 TB
Extra-Large [XL]	1001-5000	<= 60 mil/day	~120 GB	~48 TB

SIEMBIOT platform is sized so it can accommodate a minimum of 10 SME tenants with a distribution of 5 [S] + 3 [M] + 1 [L] +1 [XL] = 10 SMEs.

SIEMBIOT UC1 validates the multi-tenancy for the SOCaaS demo platform for each of the SME sizes above.

2.1.1 UC1 Functional Requirements

Req id	Description
UC1.FR1	The demo (SOCaaS) tenant must be able to collect & corelate security events from SME's network & security devices, servers, end-user devices, EDRs, & applications.
UC1.FR2	The demo platform must be able to ingest different log & events format: syslog, windows events, APIs, SNMPs, NetFlow, other.
UC1.FR3	The demo platform must be capable to integrate and use multiple CTI feeds.
UC1.FR4	The demo platform must provide on-demand & scheduled vulnerability management scanning capabilities.
UC1.FR5	The platform must provide real-time KPIs, dashboards & security reports.

UC1.FR6	The demo platform must be capable of pushing selected anonymized Security events received from customer systems to the SIEMBIOT research data lake.

2.1.2 UC1 Integration Requirements

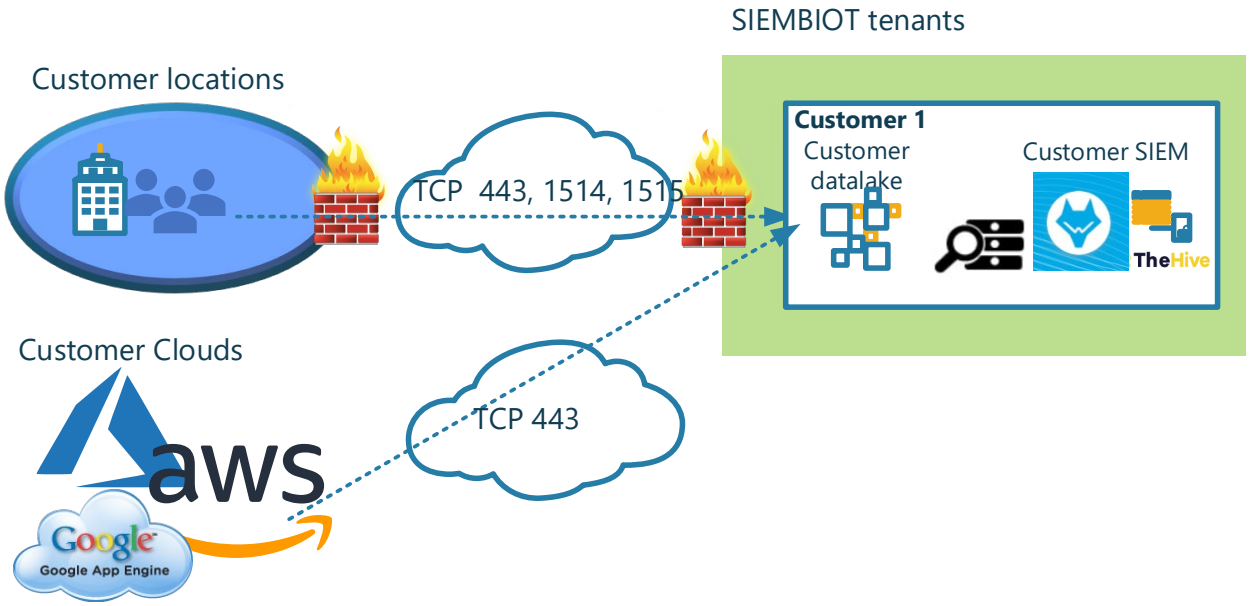


Figure 1

Req id	Description
UC1.INT1	Events & logs streamed from customer locations to the demo SOCaaS customer tenant (e.g. TCP 443, TCP/UDP 514, others)
UC1.INT2	API integration from customer security applications (e.g., EDR, AV manager, public cloud platforms) to the demo SOCaaS customer tenant.
UC1.INT3	User access from customer locations and\or devices to SOCaaS web applications.
UC1.INT4	API integration between customer demo SIEM and security incident and response platform SIRP.
UC1.INT5	API integration between the customer SIRP and the SIEMBIOT shared services analysers.
UC1.INT6	One way Event push from the demo customer SOCaaS to the SIEMBIOT training & research data lake.

2.1.3 UC1 Security requirements

Req id	Description
UC1.SEC1	All communication from customer environments to SIEMBIOT platform must be done over encrypted channels (TLS, SSL)
UC1.SEC2	All information transfer form external systems to SIEMBIOT platform must be validated by firewall IDS modules.

UC1.SEC3	Permit only secure point-to-point connection to the SIEMBIOT shared services platform.
UC1.SEC4	Access to customer demo SOCaaS applications must always be authenticated (no public/anonymous access).
UC1.SEC5	Access to "SIEMBIOT Shared services" must always be authenticated (no public/anonymous access).
UC1.SEC6	Role based access for customer users on the SIEMBIOT demo platform.

2.1.4 UC1 Capacity & Performance requirements

Req id	Description
UC1.CAP1	Demo tenants platform ingestion capacity for 10 demo tenants (5 small + 3 medium + 1 large + 1 extra-large) organisations.
UC1.CAP2	Demo tenants' platform operational storage capacity (3 months retention) for 10 demo tenants.
UC1.CAP3	Demo tenants platform archive capacity (12 months retention) for 10 demo tenants.
UC1.CAP4	Demo tenants' platform compute capacity (query, corelations, visualisation) capacity for 10 demo tenants.

2.1.5 UC1 Availability requirements

Req id	Description
UC1.HA1	Desired End-to-end availability of 99% yearly

2.2 [UC2] Cyber security research data lake

SIEMBIOT project aims leveraging customers onboarded in SIEMBIOT demo tenants and building a research data lake with the anonymized information collected from these demo customers. The purpose is to include diverse customers from various industries, to collect events from multiple devices, technologies, and geographies, to corelate and use them for training and research.

To achieve this the project must develop the anonymization functions and to ensure that they are efficient so events from multiple customer demo tenants can be funnelled, updated, and indexed concurrently into the research data lake.

2.2.1 UC2 Beneficiaries

The training & research data lake will be made available to all organisations onboarded on SIEMBIOT demo tenants. The beneficiaries list can be extended to all cyber security organisations who want to train their cyber security teams using the anonymized data lake or want to perform discovery & research with the aim of building new detect hunting queries and methods.

The list of beneficiaries includes:

A. Expertware, DNSC.
B. SIEMBIOT partners – enrolled via UC4.
C. SMEs organisations enrolled via UC1 and/or UC3.

2.2.2 UC2 Functional Requirements

The minimal set of functional requirements is detailed below:

Req id	Description
UC2.FR1	Perform anonymization for security events sent as JSON
UC2.FR2	Training and research data lake must be capable to ingest concurrently anonymized events sent from multiple demo SOCaaS tenants.
UC2.FR3	Training and research data lake must be capable to normalize, store and index security events received from different tenants into a single namespace.
UC2.FR4	Training and research data lake must be capable or running 20+ concurrent leaf queries.
UC2.FR5	Training and research data lake must be capable of running 10+ concurrent compound queries.
UC2.FR6	Permit multiple concurrent user sessions
UC2.FR7	Training and research data lake information should not be revertible to the original customer events.

2.2.3 UC2 Integration Requirements

Req id	Description
UC2.INT1	Receive events from customer demo SOCaaS tenants
UC2.INT2	Accept authenticated queries created via the training and research portal (UC3).
UC2.INT3	Display query results in the training and research portal.

2.2.4 UC2 Capacity and Performance Requirements

The capacity and performance for the research data lake use case must ensure that it can process the anonymization, the indexing and storage of events from all hoisted customer demo SOCaaS tenants. UC2 data lake aims facilitating the research into developing novel methods & queries which identify suspicious, malicious activities and anomalies.

Req id	Description
UC2.CAP1	Capable to perform anonymization up to 300 events/sec
UC2.CAP2	Capable to index & store up to 300 events/sec
UC2.CAP3	Capable to store events for operational queries up to 6 months
UC2.CAP4	Capable to support up to 30 concurrent query sessions

2.2.5 UC2 Availability Requirements

As a demo research platform SIEMBIOT does not require a strict availability nor disaster recovery.

Req id	Description
UC2.HA1	Desired end-to-end availability of 99% yearly

2.2.6 UC2 Security & compliancy requirements

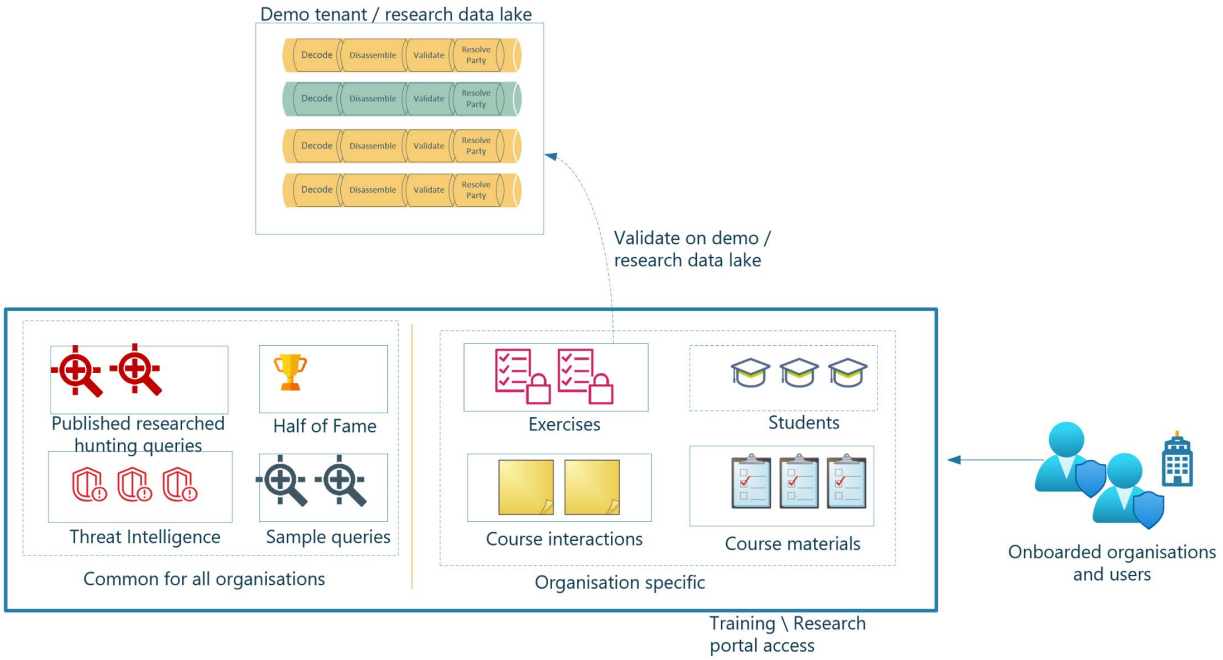
Req id	Description
UC2.SEC1	All communication customer demo tenants must be done over encrypted channels (TLS, SSL)
UC2.SEC2	The training and research data lake should permit only authenticated access for both users and APIs.
UC2.SEC3	The training and research data lake will implement RBAC authorization based on “minimum required access” principle.
UC2.SEC4	All the training and research inbound interfaces analysed must be analysed by firewall intrusion detection module
UC2.SEC5	The solution by be compliant with GDPR rules.

2.3 [UC3] Cyber security training range

SIEMBIOT UC3 aims developing a role-based web portal which complements the demo SOCaaS tenants allowing onboarded customer organisations and cyber security partners to train their teams for security incident management and threat hunting.

Organisations onboarded on the SIEMBIOT demo platform will receive a dedicated portal space where they can demo attack scenarios, work collaboratively taking notes, storing knowledge articles, having access to cyber security news, developing, and saving new hunting queries. If they wish they can submit advanced hunting queries which after validation from the portal admins can be made available for the other organisations onboarded.

The portal will allow uploading course documents, defining exercising, collecting students results and evaluating them.



2.3.1 UC3 Functional requirements

Req id	Description
UC3.FR1	Onboarding mechanism for organisations and users/students
UC3.FR2	Provide the capability to define courses and upload course materials (documents, media files).
UC3.FR3	Provide the capabilities to define sets of exercises.
UC3.FR4	Assign courses and exercises to students
UC3.FR5	Students: write and validate hunting queries against demo tenants \ research data lake
UC3.FR6	Provide interactions capabilities between students and tutors (notes, chats)
UC3.FR7	Common area available for all organisations with sample queries, threat intelligence, cyber news, hall of fame, new CTIs developed on SIEMBIOT research data lake.

2.3.2 UC3 Capacity and performance requirements

The requirements must ensure that we can onboard and support 50+ organisations and 500+ users.

Req id	Description
UC3.CAP1	Capable to host 50+ organisations, 500+ named users
UC3.CAP2	Capable to support 50 concurrent sessions
UC3.CAP3	Capacity to store course documents & media up to 100GB.

2.3.3 UC3 Security Requirements

Req id	Description
UC3.SEC1	All communication customer demo tenants must be done over encrypted channels (TLS 1.2)
UC3.SEC2	RBAC authorization based on “minimum required access” principle
UC3.SEC3	Server-side SSL certificates with a public key length of minimum 2048 bits
UC3.SEC4	User sessions analysed by firewall IDS/IPS modules
UC3.SEC5	1 year audit and retention for web server events

2.3.4 UC3 Availability Requirements

Req id	Description
UC3.HA1	Desired End-to-end availability of 99% yearly

2.4 [UC4] CTI exchange platform

SIEMBIOT UC4 refers to the development of Web APIs for the exchange of threat hunting queries developed by the SIEMBIOT partners and customers.

Partners and customer having access to the SIEMBIOT research portal will be able to develop and validate novel threat hunting queries. Such queries will be made available via web API get calls for registered beneficiaries.

2.4.1 UC4 Functional requirements

Req id	Description
UC4.FR1	Enrolment page for beneficiaries who want to get SIEMBIOT develop threat hunting queries
UC4.FR2	Web API GET method to retrieve a list of threat hunting queries.
UC4.FR3	Web API PUT method to submit new threat hunting queries
UC4.FR4	Cyber security news RSS feed

2.4.2 UC4 Capacity and performance requirements

Req id	Description
UC4.CAP1	50 GET Web API concurrent sessions
UC4.CAP2	20 POST Web API POST concurrent sessions
UC4.CAP3	50 RSS concurrent feeds

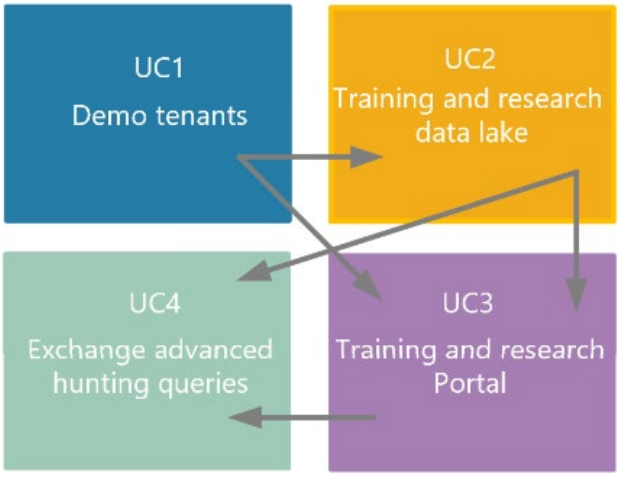
2.4.3 UC4 Security and compliance requirements

Req id	Description
UC4.SEC1	All communication customer demo tenants must be done over encrypted channels (TLS 1.2)
UC4.SEC2	Server-side SSL certificates with a public key length of minimum 2048 bits
UC4.SEC3	API authorization required
UC4.SEC4	Traffic inspected by IDS/IPS firewall modules
UC4.SEC5	Web Audit logs retained for 365 days

2.4.4 UC4 Availability Requirements

Req id	Description
UC4.HA1	Desired End-to-end availability of 99% yearly

2.5 Use cases relationships.



3 PLATFORM TECHNICAL ARCHITECTURE

3.1 Architecture principles

Architectural principles applied were:

- Secure end-to-end.
- Apply RBAC authorization based on “minimum required access”.
- Define reusable functional building blocks.
- Minimize the number of technologies used.
- Prioritize open-source components removing vendor lock-in.
- Standardize where possible (industrialization).
- Consolidate where possible (synergies for shared services).
- Provide end to end redundancy (no single point of failure).
- Simplify the design (chose shortest path).

At the same time some major changes to the configuration paradigms for infrastructures have been implemented.

3.2 Multi-tenancy

Multi-tenancy allows multiple tenants (organisations or users) to securely share the same SIEM instance, while maintaining data isolation and access control. The main multi-tenancy requirements to consider are:

- **Data Isolation**: Tenants' data must be logically and securely isolated from each other. Ensure that each tenant's data is stored separately and cannot be accessed by other tenants.
- **Authentication and Authorization**: Implement robust authentication and authorization mechanisms. Tenants should only have access to their own data and resources.
- **Tenant Onboarding**: Develop a process for onboarding new tenants efficiently. This should include provisioning resources, setting up databases, and configuring tenant-specific settings.
- **Scalability**: Design your system to scale horizontally to accommodate a growing number of tenants. This may involve load balancing, resource allocation, and dynamic provisioning.
- **Performance Isolation**: Ensure that the activities of one tenant do not impact the performance of other tenants. Resource allocation and performance monitoring are crucial for this.
- **Resource Allocation**: Define resource allocation and resource limits for each tenant, including CPU, memory, and storage quotas. Implement mechanisms to prevent resource hogging by any single tenant.
- **Customization**: Allow tenants to customize their SIEM dashboards, reports, and alerts to meet their specific requirements and branding guidelines. However, ensure that customizations don't compromise security or the stability of the system.
- **Data Backup and Recovery**: Implement data backup and disaster recovery plans for each tenant's data to ensure data availability and business continuity.
- **Data Retention Policies**: Enable each tenant to set its own data retention policies and archival rules to comply with regulatory requirements or internal policies.

- **Monitoring and Reporting**: Provide monitoring and reporting tools for tenants to track their usage, performance, and security. Centralized monitoring can also help you identify and address issues across all tenants.
- **Upgrades and Maintenance**: Plan for regular maintenance, updates, and patch management. Ensure that upgrades can be applied without disrupting tenant services.
- **Compliance and Security**: Address compliance requirements, especially if tenants are subject to specific regulations. Implement security measures to protect tenant data and privacy.
- **Support and Helpdesk**: Provide tenant-specific support channels and helpdesk services. Tenants should have a way to reach out for assistance or report issues.
- **Tenant Offboarding**: Develop a process for offboarding tenants when they no longer use your services. This should include data deletion and resource deprovisioning.
- **Tenant Isolation in Development and Testing**: Maintain separate environments for development, testing, and staging to prevent tenant data exposure during the development lifecycle.
- **Documentation**: Create clear and comprehensive documentation for tenants, including user guides, FAQs, and API documentation.
- **Security Incident Response**: Have a well-defined incident response plan in place to address security breaches or incidents that may affect one or more tenants.
- **Cross-Tenant Collaboration** (*if required*): Consider features that allow secure collaboration between tenants when necessary, such as sharing threat intelligence or incident data.

3.3 High level system architecture

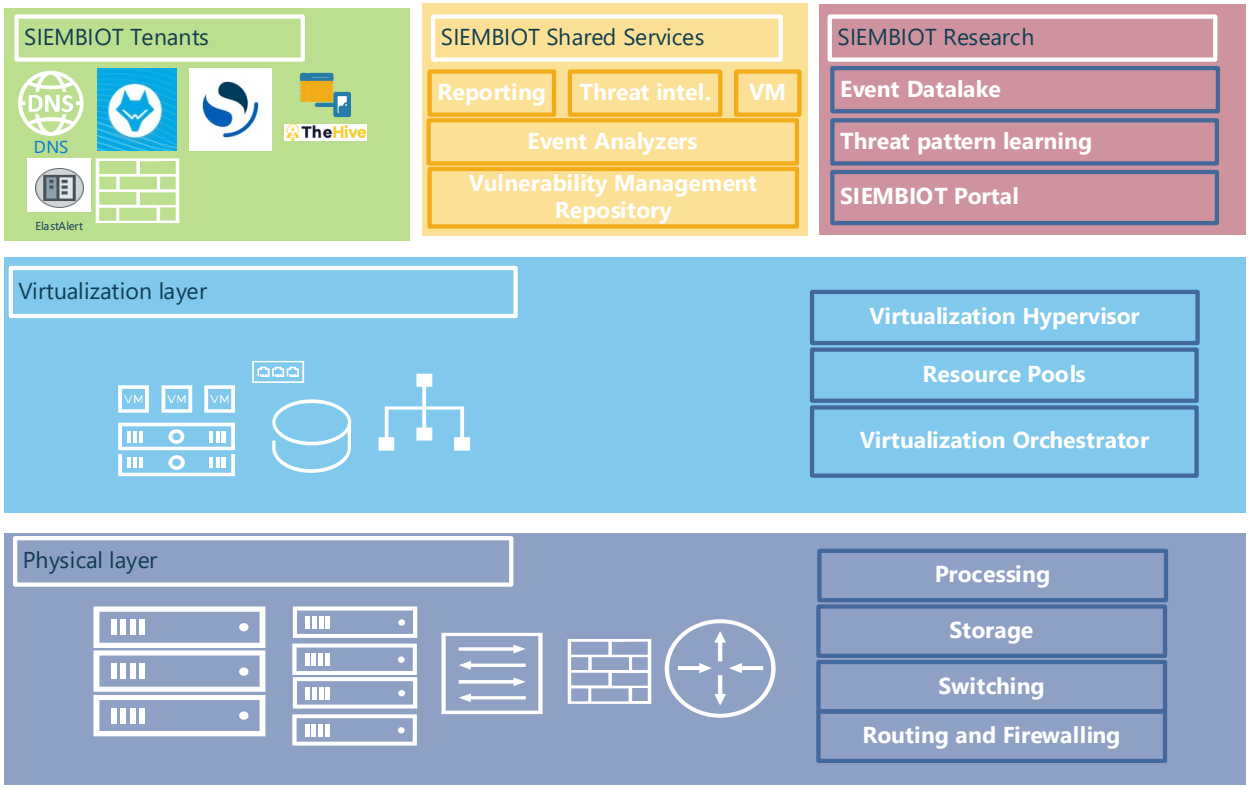


Figure 2

There are 3 distinct SIEMBIOT logical environments:

- SIEMBIOT demo tenants (SOCaaS) platform
- SIEMBIOT Shared Services platform
- SIEMBIOT Training & Research platform

All environments leverage a common virtualization layer to access the physical infrastructure: network & security devices, servers, storage area network optimizing the resource allocation and facilitating flexible capacity needs for each of the 3 logical environments described above. Compute capacity, networking, storage and security devices will be configured as virtual resources which can be easily reallocated, scale up or shrink down as demanded by the customers specifics.

3.4 Networking services architecture

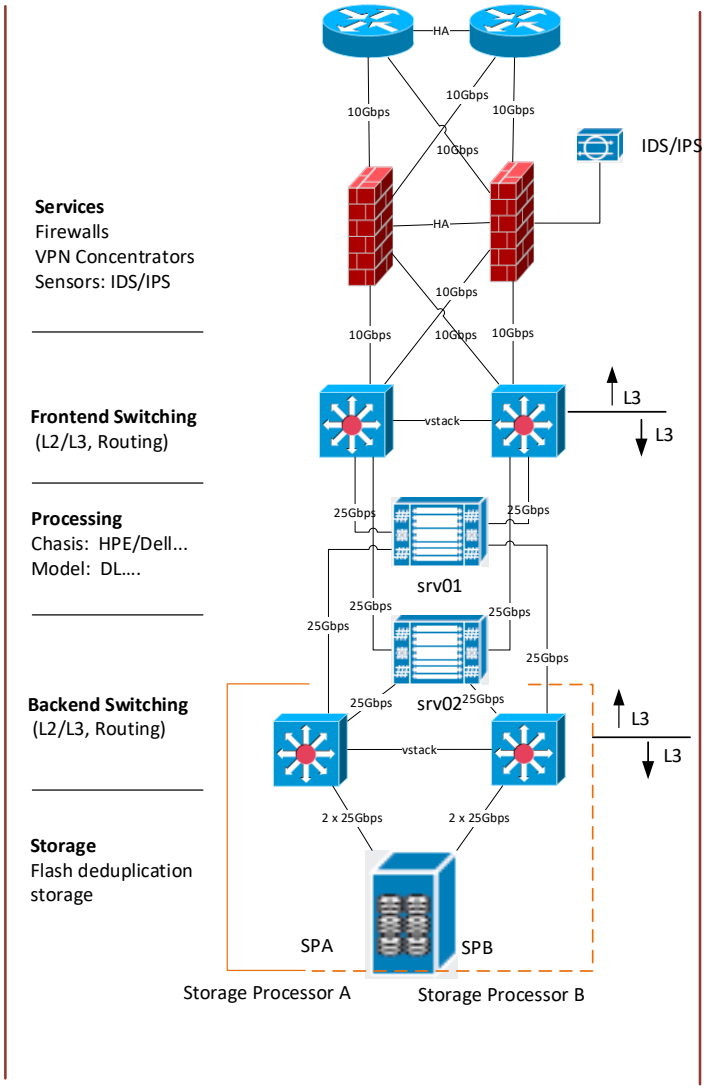


Figure 3

A well-designed data network is critical for any organisation to maintain a competitive edge in today's rapidly evolving digital landscape. An effective network design ensures that the infrastructure is agile and reliable and makes it scalable, resilient, and secure. By following best practices in network design, we can maximise the network's potential and ensure it can effectively support mission-critical applications and services. The benefits of having a well-designed data network are numerous, from reducing downtime and improving performance to enhancing security. Also, it can help reduce downtime by automating failure recovery processes and ensuring the network is always available. This improves the end-user experience and helps maintain a competitive edge in today's fast-paced digital landscape.

Taking these into consideration, our design will be based on:

- All networking components and services are redundant, providing HA for every service (routing, firewalling, IDS/IPS, switching, QoS etc).
- Minimum connection is 10Gbps, bundled in aggregates, to provide redundancy and higher throughputs.

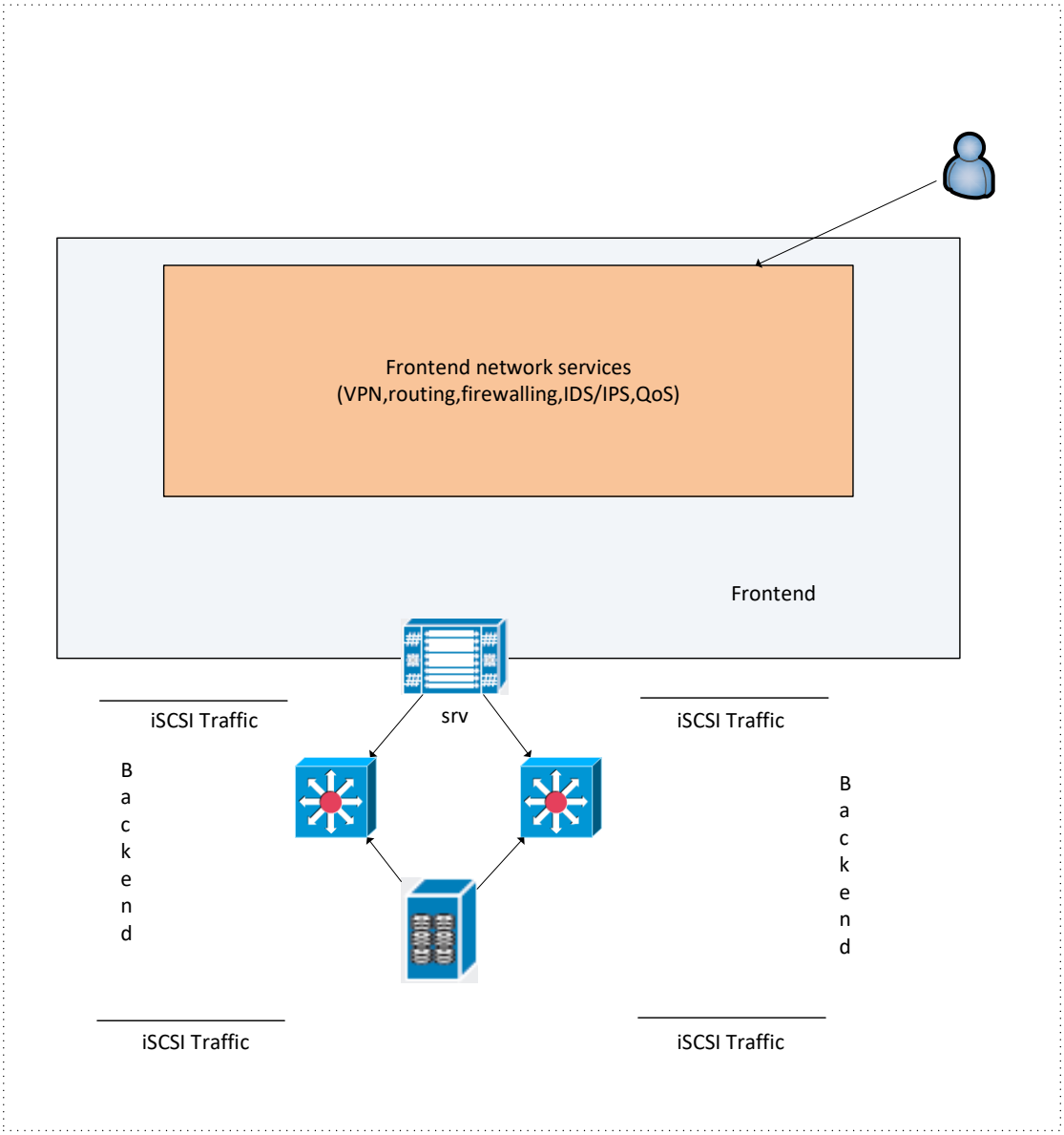
- Each node will have 2 x 25Gbps interfaces connected for frontend and 2 x 25Gbps interfaces connected for backend.
- All connections will be optical fibre based, to avoid eventual RF interferences.

3.5 Storage platform architecture

SIEMBIOT platform requires configuration and an enterprise SAN fabric to guarantee performance and availability.

Each physical server host will be connected via redundant paths to the underlying storage enclosures ensuring that any hardware error (network card, switch, router, storage processor) won't trigger unavailability of storage volumes used by the SIEMBIOT components.

SAN traffic will be isolated from the frontend normal network traffic, using dedicated switches and connections.



SAN storage manages NVMe SSD disks creating protected RAID 6 groups. Virtual disk volumes are provisioned on each of the controller nodes and presented over 2 controllers, 4 redundant paths.

3.5.1 Storage HA design

We build several layers of protection for the data saved:

- DRAID 6 groups (double parity)
- Dual controllers
- Dual SAN fabrics
- Replication to the 2nd node for selected virtual disks.

All equipment benefits of dual power supply and dual circuit feeds.

3.5.2 Storage Scalability

There are 3 different resources which might require extension if the demand increases.

The solution has several layers for the storage virtualization: disks pools, virtual volumes, volumes groups and each of them allows scaling horizontally.

Disk capacity can be extended using horizonal scaling adding disk enclosures or additional controller nodes or vertically enabling software compressions.

3.6 Virtualization layer

Virtualization layer aggregates physical resources and allows flexible slicing of virtual machines, virtual disks, virtual networks etc so we can separate securely SIEMBIOT SOCaaS tenants from shared services and the training & research lab.

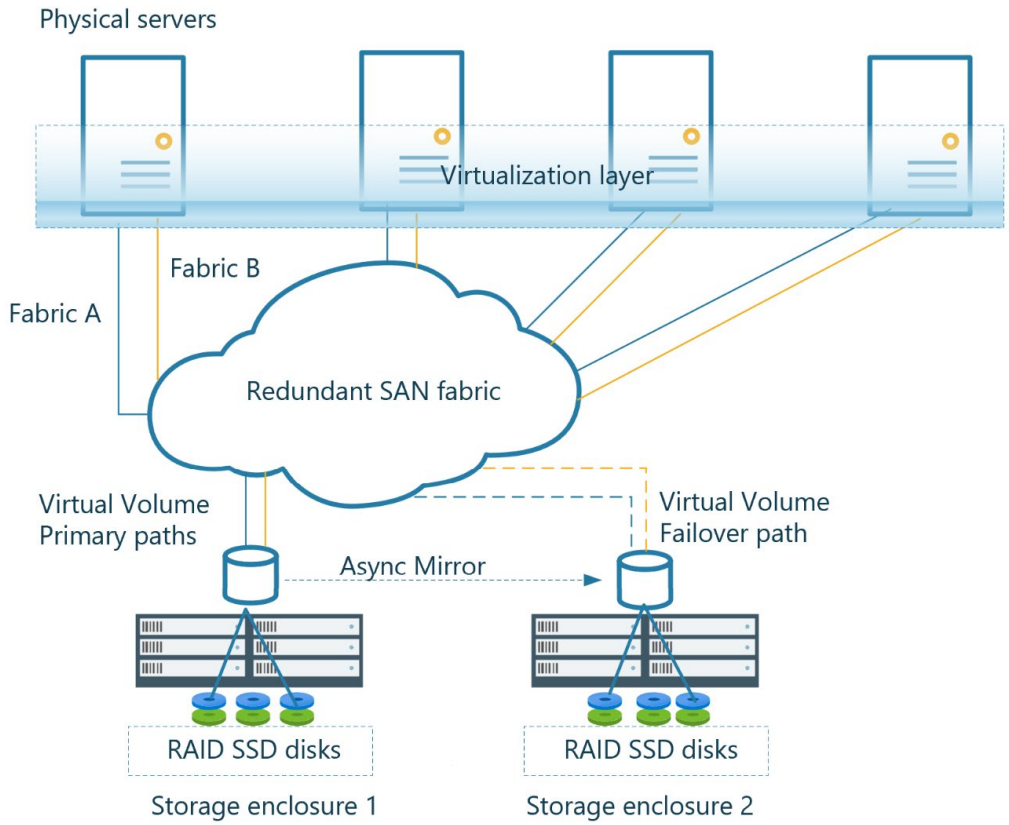


Figure 4

The networking will benefit from the creation of virtual switches and routing tables so customer demo tenants are isolated. We will ensure that there are no promiscuous network cards connected on the virtualization layer and that if

The virtualization layer adds several benefits:

- Capability to combine, share and optimize the hardware resources among different platforms.
- Capability to flexibly adapt resource allocation.
- Capability to deploy high available HA (N+1) clusters leveraging common failover hosts for multiple platforms.
- Capability to logically separate the environments although they use common servers. RAM memory will be reserved for workloads if they are from different platforms and they share the same physical server to ensure the separation.

4 LOGICAL COMPONENTS ARCHITECTURE

4.1 SIEMBIOT Demo Tenants (SOCaaS) Platform

Each organisation invited to trial will benefit from a dedicated SOCaaS tenant provisioned solely for their use. The main components span 3 distinct categories:

A. Ingestion layer = Apache NiFi, Apache Kafka, Apache Spark
B. Storing & indexing = Logstash, Wazuh Manager, OpenSearch
C. Analyse, visualise & alerts = TheHive, ElastAlerts, Kibana.

Besides the functional categories we need to provision supporting core IT management services (DNS, NTP, remote management).

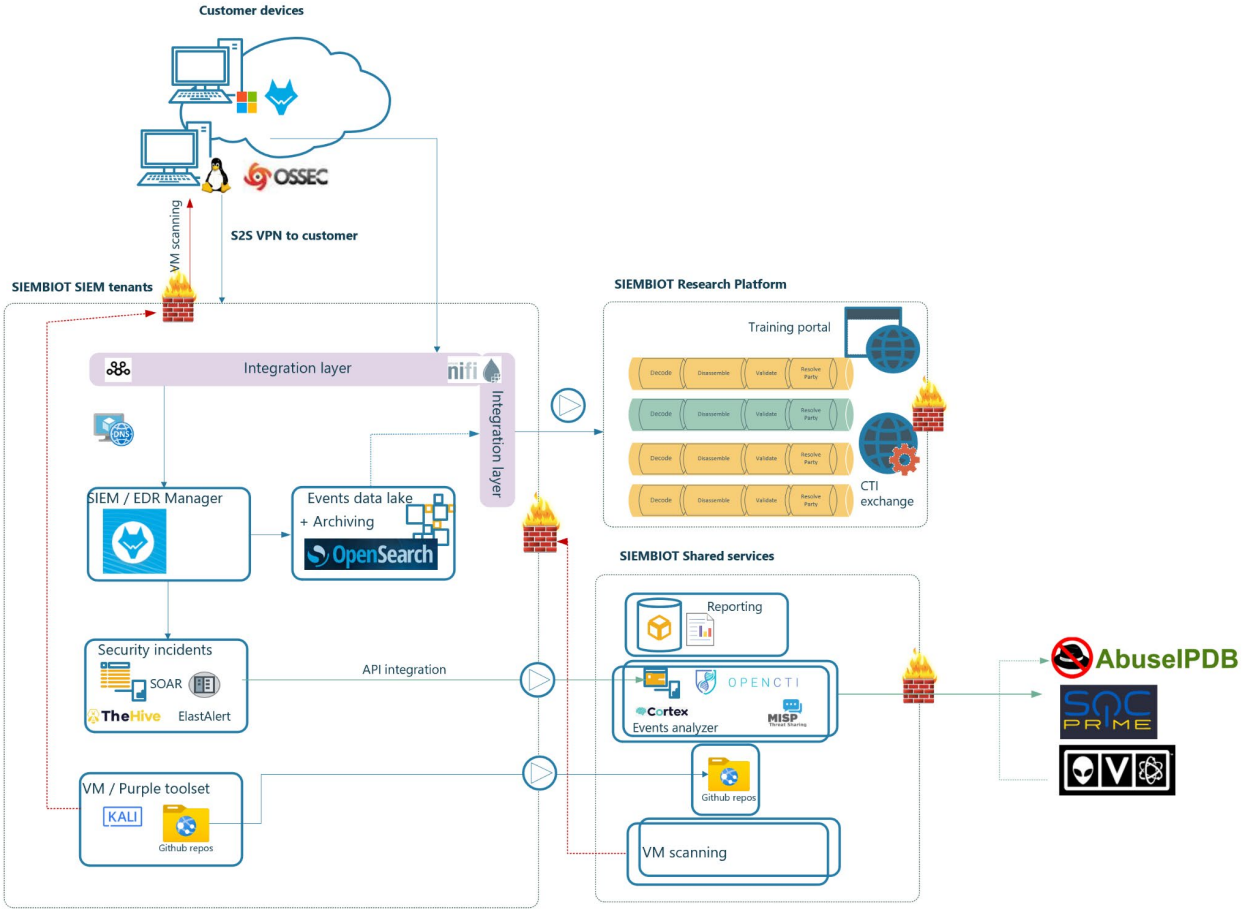


Figure 5

Virtualization layer will facilitate the dynamic provisioning of the logical components required. SIEMBIOT leverages KVM & docker capabilities to ensure optimal use of physical resources while providing isolation among tenants.

4.1.1 DNS configuration

We propose a standardized taxonomy based on customer 2 digits id depicted as *C[nn].*

ingest[0-9].*C[nn].*siembiot = virtual name for the integration endpoints (NiFi, Kafka)

siem[0-9].*C[nn].*siembiot = virtual name for the SIEM endpoint (Wazuh, Elastic)

secincident[0-9].*C[nn].*siembiot = virtual name for the security incident management (Hive)

alert[0-9]. *C[nn].*siembiot = virtual name for the notification engine (elastAlert)

portal[0-9].*C[nn].*siembiot = analysis and visualisation components (Kibana,)

purple[0-9].*C[nn]*.siembiot = purple toolset and management stations

In order interacting with the other SIEMBIOT components and the customers systems we will configure DNS conditional forwarding as follows:

C[nn] domains → forward requests to C[nn] DNS servers via the S2S VPN.

*.shared.siembiot → forward requests to dns[1-2].shared.siembiot

*.expertware.local → forward requests to dns[1-2].expertware.local

4.1.2 IPv4 allocation

Like for the DNS naming we established allocation rules for IPv4 addresses so the platform is easily extentable.

Each customer tenant will receive a /24 subnet from the address space 10.99.0.0/16 following the rule 10.99.*C[nn].*0/16 where *C[nn]* represents the ordinal for the customer tenant onboarding. This taxonomy allows us to provision up to 254 distinct tenants.

Services requiring load balancing like theHive, ElasticCluster will receive virtual IPs following the rule mod(IP) = 0.

Sample IP provisioning for the 5th customer:

IPv4	Role
10.99.5.1	Default network gateway
10.99.5.2	Dns1.C5.siembiot
10.99.5.3	Dns2.C5.siembiot
10.99.5.10	Ingest.C5. siembiot (load balanced VIP)
10.99.5.11	Ingest1.C5.siembiot
10.99.5.12	Ingest2.C5.siembiot
10.99.5.20	Siem.C5.siembiot (load balanced VIP)
10.99.5.21	Siem1.C5.siembiot (1st SIEM node / docker)
10.99.5.22	Siem2.C5.siembiot (2nd SIEM node / docker)
10.99.5.30	Wazuhm.C5.siembiot (Wazuh manager)

4.1.3 Platform sizing

There are different capacity needs triggered by the specific applications components which are to be provisioned as part of the demo tenants.

CPU & RAM capacity

Servers CPU and memory RAM are required to run the applications in virtual machines or containers. We made an estimation based on previous proof of concepts done for small and medium organisation sizes.

Capacity allocation based on use case requirements

Tenant size	Ingestion [Nifi + Kafka] [L1]	Enrich, Index, Store [L2]	Query, Corelate, Visualise [L3]
Small	8 vCPUs, 16 GB RAM	16 vCPUs, 64 GB RAM	8 vCPUs, 32 GB RAM

Medium	12 vCPUs, 24 GB RAM	48 vCPUs, 192 GB RAM	18 vCPUs, 80 GB RAM
Large	16 vCPUs, 32 GB RAM	56 vCPUs, 224 GB RAM	24 vCPUs, 96 GB RAM
Extra Large	24 vCPUs, 48 GB RAM	96 vCPUs, 384 GB RAM	32 vCPUs, 144 GB RAM

Using a virtualized platform, we ensure that we can benefit of the economy of scale as not all demo tenants will require reserved capacity during the exact same interval of time. For performance reasons we will reserve the amount of RAM memory for the [L2] and [L3] components while allowing vCPU oversubscription with a ratio of 1.5 virtual cores:1 physical core.

Demo tenants platform provisioned capacity (5S+3M+1L +1XL)

Tenant size	vCPU allocation	vRAM
Small	160	560
Medium	234	888
Large	96	352
Extra LArge	152	576
TOTAL	642	2376

The estimate capacity remains flexible in case we will have to deploy demo tenants of different sizes. To ensure high availability the solution will leverage virtualization clustering provisioning additional physical servers which allow scalability and provide redundance in case of hardware errors.

Storage & Archiving capacity

Customer demo tenants will store the received security events and logs together with their enrichments into two distinct persistent storage zones:

- operational data lake, fast SSD storage, no compression.
- archive data lake, still SSD storage but with compression and deduplication to gain storage capacity.

Persistent storage allocation

Tenant size	Daily avg capacity [GB]	Operational data lake [GB] [3 months]	Archive data lake [GB] [13 months]
Small	10	900	4000
Medium	15	1350	6000
Large	30	2700	12000
Extra Large	120	10800	48000
TOTAL		15750	70000

Archive data lake will ensure that forensics activities can be done for up to 390 days in the past.

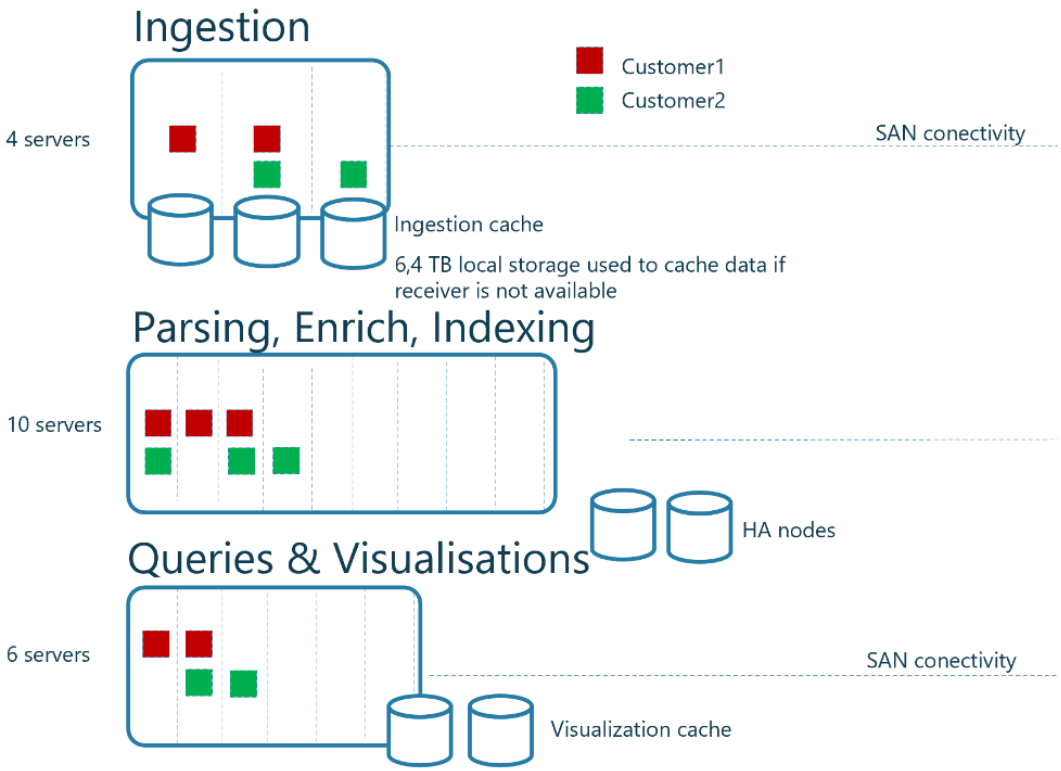
Ingestion [L1] and Query, Analysis [L3] layers require the allocation of an additional cache storage for the following purposes:

[L1] cache storage = will ensure that in case of a bottleneck for the [L2] layer the security events & logs received are queued locally. Such occurrences should take place exceptionally and for short burst so for efficiency reasons we will use servers' local storage.

[L3] cache storage = Query and Analysis layer is used to perform multiple advanced queries against large swaths of data and caching locally brings efficiency and performance (same

data is not read again from the data lake to the L3 servers). Analysis cache is ephemeral as it stores only copy of already existing data from the persistent operational / archive data lakes. Therefore, we will leverage the servers' local storage instead of the shared SAN storage.

Demo tenants platform Servers allocation



Depicted is a sample of physical servers required to ensure hosting for 10 demo customers tenants (5 Small + theh3 Medium + 1 Large + 1 Extra Large), high availability and 20% growth capacity.

4.1.4 High Availability Configuration

As all SOCaaS demo platform components are deployed on top of the hypervisor layer will benefit from the no-SPOF virtualization design.

This section will detail the application specific high availability design considerations meant to protect the solution from logical and human errors.

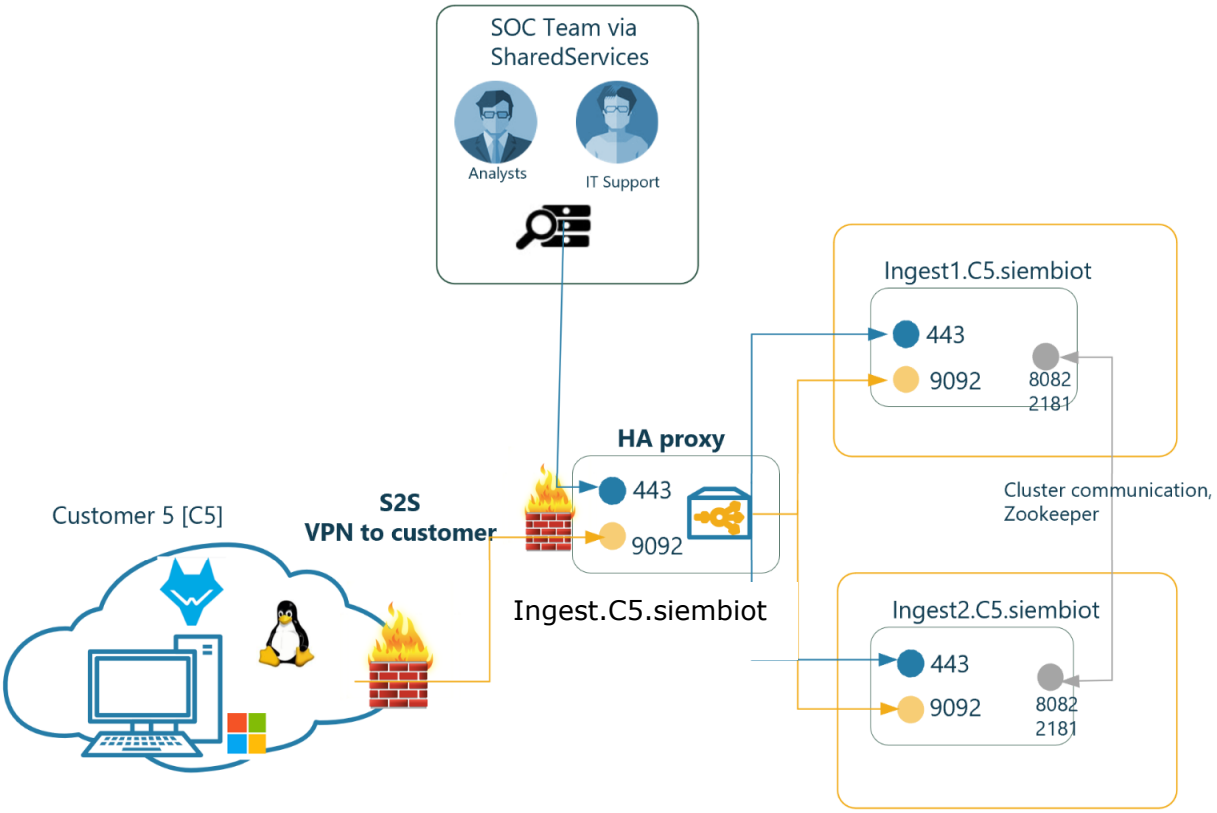
[L1] Ingestion layer [Apache NiFi HA]

For several services like Apache HA there is a need to guarantee that the logical failure of a docker node or virtual machine does not stop the ingestion of events sent form customer locations. The solution supported by both Apache Kafka and Apache NiFi projects leverage the use of a HA proxy / load balancer and application clustering.

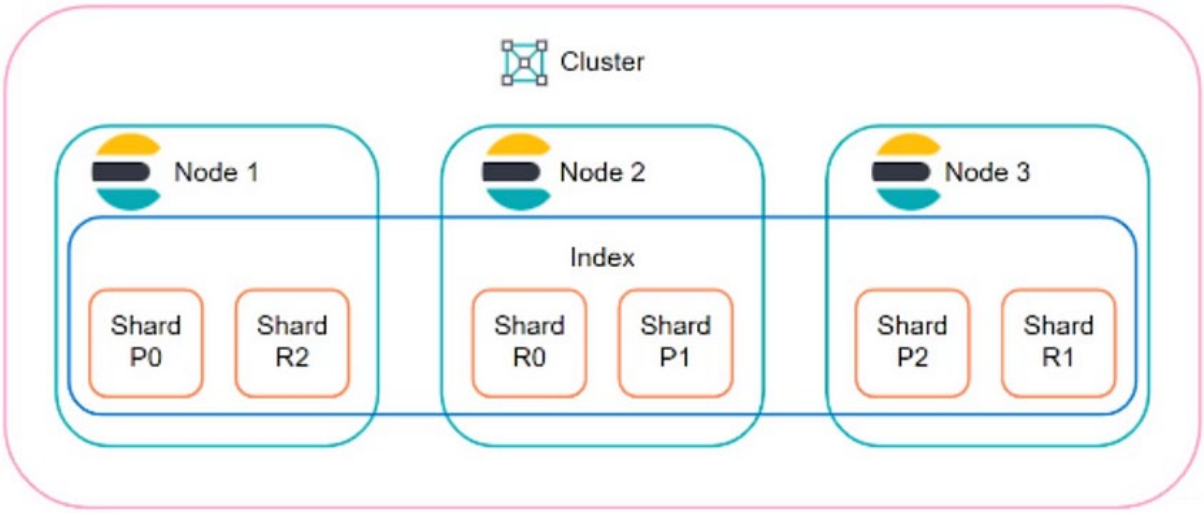
The HA proxy distributes the traffic to back-end nodes and monitors continuously the availability of services on each of the nodes. In case a service on one of the nodes is not responsive, the node is removed from the load balancing and the traffic is redirected to the

remaining functional node(s). All network connections to the servers holding the Ha proxy and the ingestion layer components are redundant as described in the section [4.2]

The HA proxy is stateless, and it can be reprovisioned in a matter of minutes.



[L2] Indexing, enriching layer [Elastic \ OpenSearch cluster HA]



[source: https://nidhig631.medium.com/primary-shards-replica-shards-in-elasticsearch-269343324f86]

Elasticsearch/OpenSearch is designed for high availability and scalability, ensuring that data remains accessible and protected against failures. Its architecture is built around the concept of clusters, nodes, shards, and replicas to achieve this.

Cluster: A cluster is a collection of one or more nodes (servers) that together hold your entire data and provide federated indexing and search capabilities across all nodes.

Node: A node is a single server that is part of the cluster, storing data and participating in the cluster's indexing and search capabilities.

Shards: Elasticsearch divides indexes into shards to distribute data across the cluster. Each shard is a basic unit of scaling and can be hosted on any node in the cluster. There are two types of shards:

- Primary Shards: These hold the main data and are initially created. Each document is indexed into a primary shard.
- Replica Shards: These are copies of primary shards, providing redundancy in case of hardware failures. Replicas also serve read requests, enhancing search performance.

Replication: Replication is the process of creating and maintaining copies of shards across multiple nodes. It ensures high availability by allowing for the continued operation of the cluster even if some nodes or shards fail.

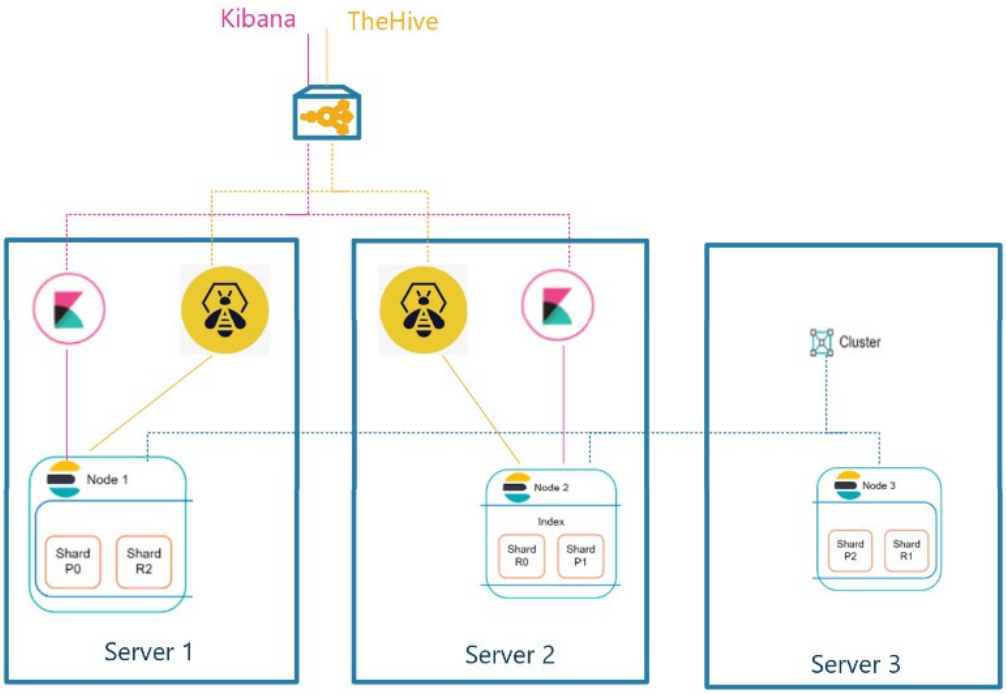
Automatic Failover: If a node hosting primary shards fails, Elasticsearch promotes one of the replica shards to become the new primary shard, ensuring uninterrupted access to data.

This architecture ensure that data is split in chunks / shards , distributed on the available 3 nodes and replicated on each node (1 Primary + 2 Replicas).

[L3] Query and Visualization [TheHive, Kibana HA]

All query and Visualization components rely on the persistent data stored by the [L2] elastic \ OpenSearch data lake.

To protect against logical corruption of front-end components we can apply a similar approach as for L1, introducing a HA proxy/load balancer in front of multiple application nodes.

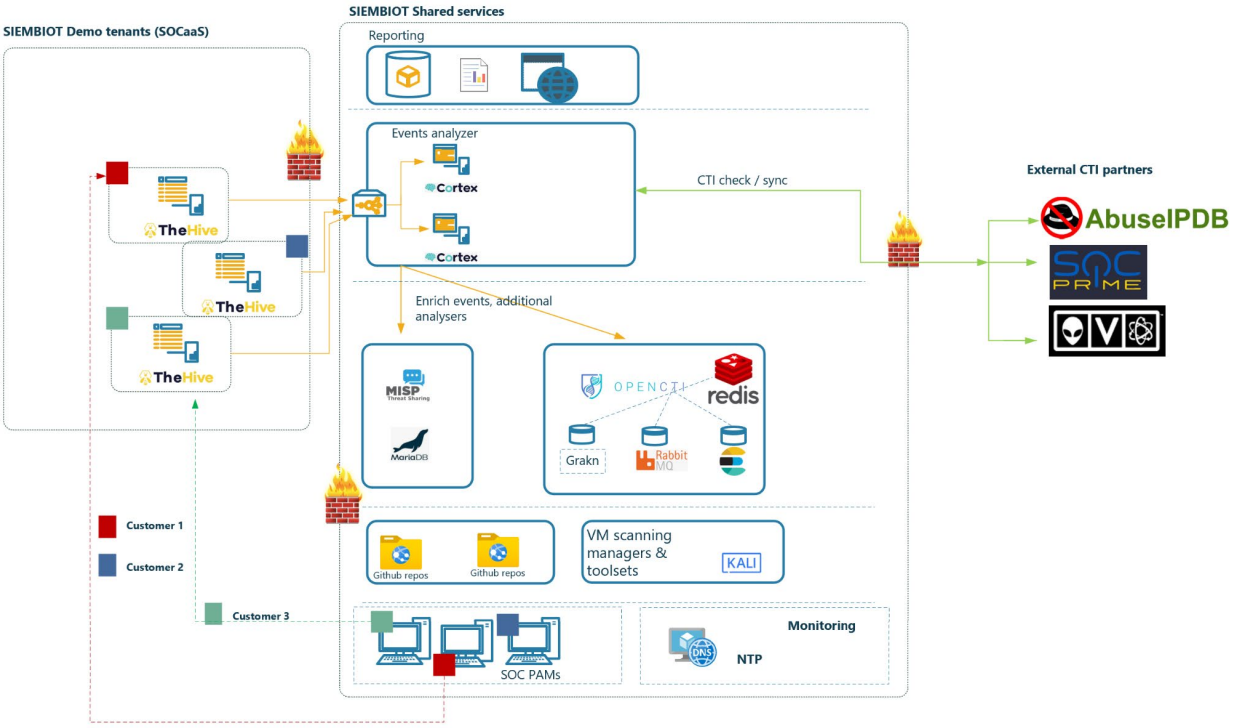


4.2 SIEMBIOT Shared Services Platform

The projects optimize the distribution of certain components centralizing services which can be reused by multiple SaaS tenants without privacy nor security impact. Such components include threat intelligence, vulnerability management, core IT and reporting.

4.2.1 SIEMBIOT shared services platform hosts components

A. Analysers [Cortex instances]
B. Cyber Threat Intelligence [openCTI, MISP Server]
C. Vulnerability Management [tools repositories, shared scanning agents]
D. Core IT components [DNS, NTP, Monitoring]
E. SOC Management PAMs
F. SOC Reporting



The diagram above includes the dependencies for each application component highlighting the segregation of acces to customer 1 & 2.

A. Shared Analysers are based on multi-instance Cortex deployment. Each customer demo SOCaaS tenant will communicate over Ha proxy/load balancer with a farm of Cortex application servers.

B. Cortex customer instance communicate with shared services CTI layer (MISP server, openCTI) to enrich or validate the events against known IoA, IoC.

C. Share Vulnerability Management layer will include shared vulnerability scanners for small customers and the repositories for the VM toolset selected. SOC analysts will be able to download on demand the VM toolsets from the customer demo tenants purple VDIs.

D. Core IT components include the NTP, DNS (shared.siembiot zone) and the NOC monitoring tools.

E. SOC Management PAMs are privileged access management VDIs mapped to the demo customer tenants.

F. SOC reporting will display KPIs and dashboards regarding the security alerts, incidents, and SOC activity.

4.2.2 IPv4 addresses allocation.

SIEMBIOT shared services will be provisioned using 5 class C ranges as follows:

10.99.99.0/24 reserved for core IT (DNS, NTP, Monitoring)

10.99.100.0/24 reserved for Cortex Analyser instances and virtual IPs.

10.99.101.0/24 reserved for shared CTI components (MISP, openCTI) and their virtual IPs.

10.99.102.0/24 reserved for Vulnerability Management shared tools.

10.99.103.0/24 reserved for SOC privileged access management VDIs used as clients to access customer demo tenants (TheHive, Purple VMs, etc).

Services requiring load balancing like Cortex, Reporting portals will receive virtual IPs following the rule mod(IP) = 0.

Sample IP provisioning for the shared services:

IPv4	Role
10.99.99.2	Dns1.shared.siembiot
10.99.99.5	Ntp1.shared.siembiot
10.99.99.10	Monitor.shared.siembiot
10.99.99.11	Monitor1.shared.siembiot
10.99.100.10	CortexC1.shared.siembiot -VIP
10.99.100.11	CortexC1n1.shared.siembiot – instance C1 node 1
10.99.100.12	CortexC1n2.shared.siembiot – instance C1 node 2
10.99.101.10	Misp.shared.siembiot
10.99.101.20	openCTI.shared.siembiot
10.99.102.10	Repo1.shared.siembiot
10.99.102.20	qualys1.shared.siembiot
10.99.103.[1][5]	PamC[1]v[5].shared.siembiot – customer[1], vdi[5]

4.2.3 Platform sizing

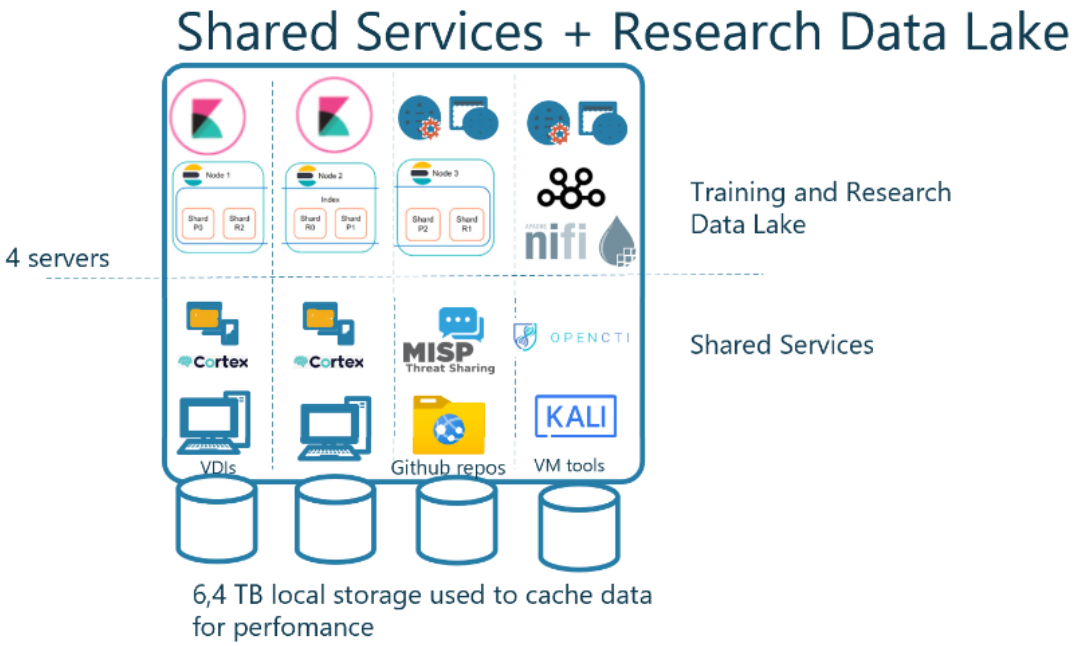
The capacity required by the shared services platform is lower compared with the demo tenants and it has a lower consumption rate as the data stored is much smaller compared with the SIEM data lakes.

CPU, RAM and storage estimated capacity

Components	vCPUs	RAM (GB)	Storage (GB)
Analysers	24	64	200
Shared CTI (openCTI)	64	512	200
Shared CTI (MISP)	16	64	100
Shared VM scanners	32	128	100
Training portal	32	128	200
Total	168	896	800

At the virtualization layer Shared services platform will shared the same physical hosts as the training and research platform.

There will be a strict separation on both networking and storage.



The provisioned physical servers will ensure high availability capacity for both SIEMBIOT shared services and the training research platforms.

In case the provisioned capacity will prove insufficient we will have the possibility to flexible adapt and use additional vCPUs, RAM memory allocated to the demo tenants.

4.2.4 CTI tools and integrations / feeds

CTI (Cyber Threat Intelligence) tools play a crucial role in helping organizations collect, analyse, and disseminate information about cyber threats. MISP (Malware Information Sharing Platform & Threat Sharing) and OpenCTI are popular open-source CTI platforms that facilitate the sharing and collaboration of threat intelligence.

OpenCTI is a stack of components (React frontend, GraphQL, Elasticsearch, Redis, Minio, RabbitMQ, python workers and connectors) that run as separate applications as part of a broader system. Because of this, it is a good fit for container orchestration, and most people install OpenCTI using Docker and run one container for each connector it has enabled.

MISP has a shorter list of dependencies (namely Redis and MySQL) and therefore fewer moving parts compared to OpenCTI. Even so, MISP's architecture also has Redis, a database, workers, a web app, and an API.

Overall, MISP's architecture incorporates fewer different technologies compared to OpenCTI.

4.2.4.1 MISP components:

MISP Feeds:

- MISP supports various threat intelligence feeds, which can be integrated to enrich the platform with up-to-date threat information. Examples include open-source feeds, commercial feeds, and ISAC (Information Sharing and Analysis Center) feeds.

MISP Taxonomies:

- MISP uses taxonomies to categorize and classify threat information. These taxonomies provide a standardized way of describing and sharing threat intelligence. Users can customize taxonomies based on their specific needs.

STIX/TAXII Integration:

- MISP supports the STIX (Structured Threat Information eXpression) and TAXII (Trusted Automated eXchange of Indicator Information) standards. This allows for interoperability with other CTI platforms that adhere to these standards, enabling seamless information exchange.

MISP Modules:

- MISP has modules that allow users to extend its functionality. For instance, there are modules for parsing and importing threat intelligence from various sources, such as log files, emails, and other formats.

MISP Integration with SIEMs:

- Integration with Security Information and Event Management (SIEM) systems enables organizations to correlate threat intelligence with their log and event data, providing a more comprehensive view of the security landscape.
-

4.2.4.2 OpenCTI components:

OpenCTI Adapters:

- OpenCTI supports adapters that allow the platform to connect with different external sources, such as MISP instances, threat feeds, and other CTI platforms. Adapters facilitate the ingestion of threat intelligence into OpenCTI.

Data Sources Integration:

- OpenCTI can integrate with various data sources, including open-source feeds, commercial feeds, and government feeds. This integration ensures that threat intelligence is comprehensive and timely.

Graphical User Interface (GUI) Extensions:

- OpenCTI provides a GUI for visualizing and analysing relationships in threat intelligence. Extensions and plugins can be added to customize the GUI based on user preferences and requirements.

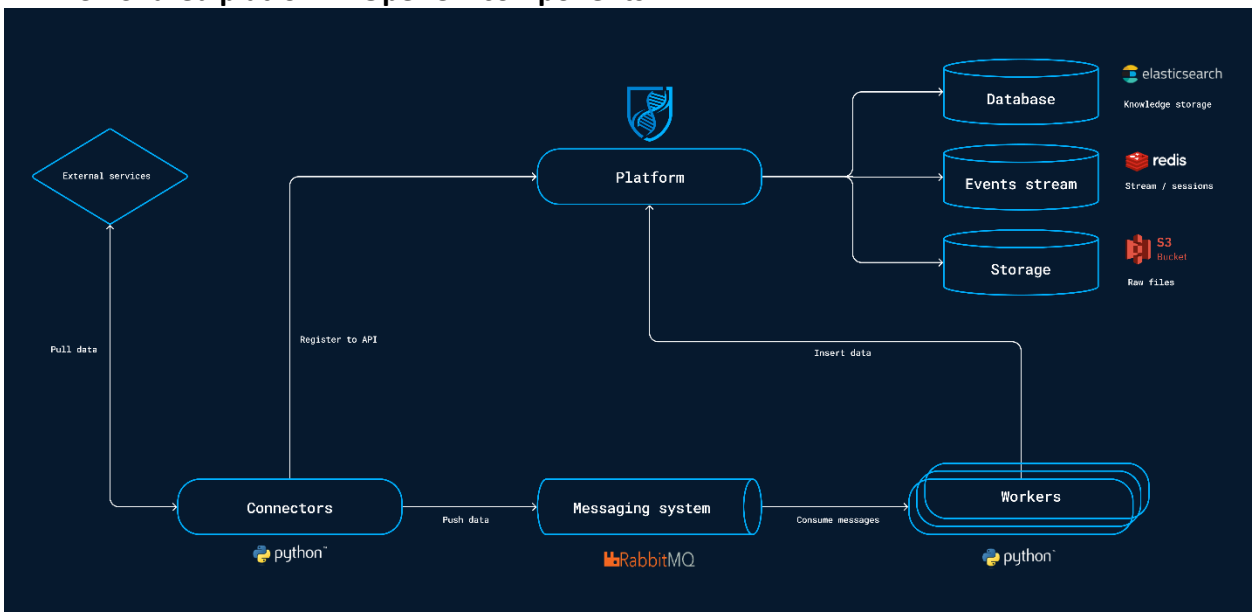
Customization and Automation:

- OpenCTI allows users to customize the platform based on their specific needs. Automation features enable the orchestration of workflows, making it easier to process and act on threat intelligence.

TIP (Threat Intelligence Platform) Integrations:

- OpenCTI can be integrated with other Threat Intelligence Platforms, allowing for information sharing and collaboration across different tools in the cybersecurity ecosystem.

4.2.4.3 Shared platform - OpenCTI components



[Source OpenCTI architecture diagram]

Platform

The platform is the central part of the OpenCTI technological stack. It allows users to access to the user interface but also provides the GraphQL API used by connectors and workers to insert data. As it is a complex platform to setup it will be shared by all the demo SIEMBIOT tenants .

The Connectors

1. MISP connector

Configure MISP connector so events form MISP are ingested in the openCTI platform.

The MISP connector is a standalone Python process that must have access to the OpenCTI platform and the RabbitMQ. RabbitMQ credentials and connection parameters are provided by the API directly.

The MISP connector will check all new events or latest modified event since the last run for import. The import process has the following steps:

- Iterate other MISP events to import with the given parameters and on modified events since the last run.
- Convert each associated galaxy or tags to OpenCTI entities: Threat actors / Intrusion sets / Malwares / Attack Patterns).
- Convert each attribute to Indicators.
- Import all Indicators, Threat actors, Intrusion sets, Malwares and Attack Patterns.
- Create indicates relationships between the Indicators and Threat actors / Malwares.
- Create uses relationships between Threat actors / Intrusion sets / Malwares and Attack patterns.
- Create indicates relationships between the previously created uses relationships.

2. TheHive connector

TheHive Connector allows for integration of TheHive' s alert and case management capabilities into OpenCTI. This connector imports TheHive' s cases, alerts, tasks, and observables into OpenCTI.

Deployment: dockers, ansible, puppets, vagrant

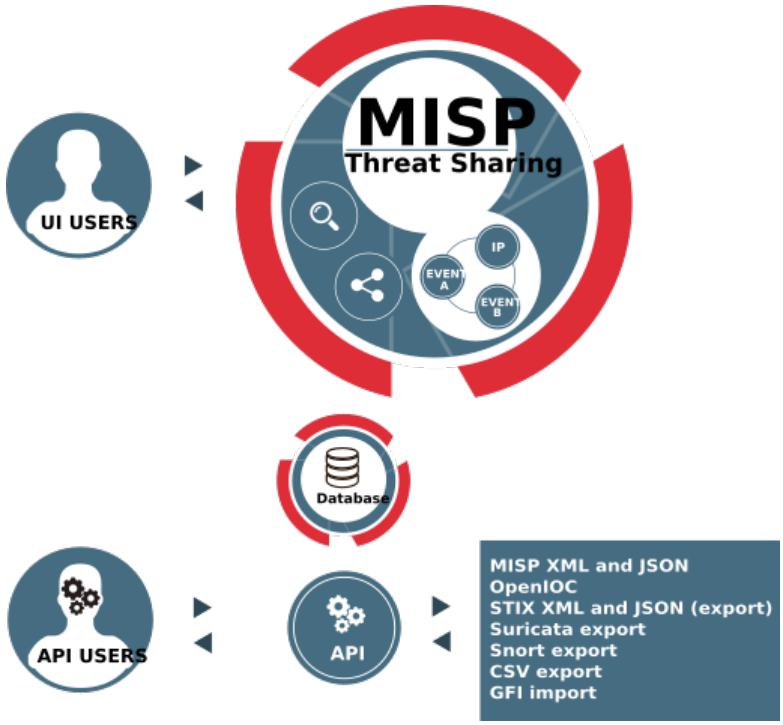
Workers

The workers are standalone Python processes consuming messages from the RabbitMQ broker to do asynchronous write queries.

The recommended setup is to have at least one platform for 3 workers (i.e. 9 workers distributed over 3 platforms).

4.2.4.4 Shared platform – MISP Architecture diagram

Each customer tenant will receive a unique APIkey allowing secure access to the central shared MISP over https. Via API calls, threat intelligence interrogations can be performed from the customer demo tenants to the shared services MISP server.



4.2.5 Continuous Vulnerability Management system architecture

One of SIEMBIOT project goals is to provide near real time information regarding the vulnerabilities present in a customer tenant. To achieve this there is a need of an accurate CMDB containing the operating systems and the software components present in the infrastructure and to validate them against new threat intelligence information fetched from trusted sources.

4.2.5.1 Selected vulnerability scanning components.

The initial list of evaluated technologies has been created based on specific categories:

4.2.5.2 AD audit

Tool	Short Description	Opensource/Free?
PingCastle	Detect critical security issues, get an overview of the technical situation, and provide guidance and advice to fix the issues.	Yes
Purple Knight	The Purple Knight AD security assessment is helping IT and security teams take action to improve overall security, including addressing AD misconfigurations, changing structure and processes to improve overall security posture, and facilitating conversations between IT and security teams.	Yes

4.2.5.3 Web application scanning / Penetration testing

Tool	Short Description	Opensource/Free?
Acunetix	A web vulnerability scanner that helps identify and manage web applications vulnerabilities	No
Arachni	A web application security scanner framework that identifies security issues in web applications, including SQL injection, XSS, and more.	No
Brakeman	Brakeman is a free vulnerability scanner specifically designed for Ruby on Rails applications. It statically analyzes Rails application code to find security issues at any stage of development.	Yes
Burp Suite	Burp Suite is a software security application used for penetration testing of web applications. Both a free and a paid version of the software are available. The software is developed by the company PortSwigger	Yes (Limited)
CMSmap	CMSmap is a python open source CMS scanner that automates the process of detecting security flaws of the most popular CMSs. At the moment, CMSs supported by CMSmap are WordPress, Joomla, Drupal and Moodle.	Yes
Metasploit Community	While Metasploit has a commercial version, the Metasploit Community edition is open source and offers penetration testing and vulnerability assessment capabilities.	Yes
Nikto	A web server scanner that identifies various vulnerabilities in web servers and web applications.	Yes

OWASP	The Open Web Application Security Project, or OWASP, is an international non-profit organization dedicated to web application security.	Yes
OWASP ZAP (Zed Attack Proxy)	An open-source security testing tool for finding vulnerabilities in web applications during development and testing.	Yes
OWTF (Offensive Web Testing Framework)	A framework that combines various security tools and makes them interoperable for web applications security testing.	Yes
oxsecurity	Find and detect vulnerabilities earlier in your SDLC. Scan open-source packages, and dependencies, and ensure compliance maintenance from a single pane of glass.	Yes
RapidScan	A multi-tool web vulnerability scanner designed for efficient and quick web applications assessments.	Yes
Retire.js	A command-line tool for identifying JavaScript libraries with known security vulnerabilities.	Yes
Shift Left Scan	Shift Left Scan is a free open-source security tool for modern DevOps teams. With an integrated multi-scanner-based design, Scan can detect various kinds of security flaws in your application, and infrastructure code in a single fast scan without the need for any remote server.	Yes
Skipfish	A security tool for scanning web applications for security issues, developed by Google.	Yes
W3af	A web application attack and audit framework design to help in finding and exploiting vulnerabilities in web applications	Yes
Wapiti	A web application vulnerability scanner that helps you identify security issues in web applications.	Yes
Wfuzz	A web application brute-forcing tool that can be used to discover vulnerabilities by testing various attack vectors.	Yes

4.2.5.4 Network (assets) scanning

Tool	Short Description	Opensource/Free?
Nessus	Nessus is a platform developed by Tenable that scans for security vulnerabilities in devices, applications, operating systems, cloud services and other network resources.	No
Nmap	Nmap ("Network Mapper") is a free and open-source utility for network discovery and security auditing.	Yes

OpenVAS (Open Vulnerability Assessment System) - Greenbone Community Edition	A powerful and widely used open-source vulnerability scanner that detects and reports security issues in systems and networks. Greenbone OpenVAS: -a vulnerability scanner (OpenVAS Scanner), -a name for a software framework of several components (GVM), -a software project led by Greenbone (Greenbone Source Edition), -a source code release of a vulnerability management solution (OpenVAS 9, GVM 20.08), or -a community of users and developers (Greenbone Community) To have a unique name we decided to use the term **Greenbone Community Edition** as an umbrella for all Open Source/Free Software efforts of the company Greenbone in 2022.	Yes
Qualys (VMDR)	Qualys is a cloud-based solution that detects vulnerabilities on all networked assets, including servers, network devices (e.g., routers, switches, firewalls, etc.), peripherals (such as IP-based printers or fax machines) and workstations. Qualys can assess any device that has an IP address.	No
Rapid7 Nexpose	It can scan and assess physical, cloud, and virtual infrastructures automatically. Nexpose integrates with cloud and virtual environments, too. In addition, the tool provides live and interactive dashboards, solution-based remediation, and risk scoring and prioritization. It also offers a lightweight endpoint agent for processing information, all while using minimal bandwidth.	No
Vuls	A vulnerability scanner for Linux and FreeBSD that focuses on software components and provides detailed vulnerability information.	Yes

4.2.5.5 Password audit

Tool	Short Description	Opensource/Free?
Directory Services Internals (DSInternals)	The DSInternals Framework exposes several internal features of Active Directory and can be used from any .NET application. The codebase has already been integrated into several 3rd party commercial products that use it in scenarios like Active Directory disaster recovery, identity management, cross-forest migrations and password strength auditing.	Yes

From the initial list, these are the selected scanning solutions based on internal testing, experience with the tools, cost, etc:

Tool	Short Description	Category	Opensource/Free?
Directory Services Internals (DSInternals)	The DSInternals Framework exposes several internal features of Active Directory and can be used from any .NET application. The codebase has already been integrated into several 3rd party commercial products that use it in scenarios like Active Directory disaster recovery, identity management, cross-forest migrations and password strength auditing.	Password audit	Yes
OpenVAS (Open Vulnerability Assessment System) - Greenbone Community Edition	A powerful and widely used open-source vulnerability scanner that detects and reports security issues in systems and networks. Greenbone OpenVAS: -a vulnerability scanner (OpenVAS Scanner), -a name for a software framework of several components (GVM), -a software project led by Greenbone (Greenbone Source Edition), -a source code release of a vulnerability management solution (OpenVAS 9, GVM 20.08), or -a community of users and developers (Greenbone Community) To have a unique name we decided to use the term **Greenbone Community Edition** as an umbrella for all Open Source/Free Software efforts of the company Greenbone in 2022.	Vulnerability scanning	Yes
PingCastle	Detect critical security issues, get an overview of the technical situation and provide guidance and advice to fix the issues.	AD audit	Yes
Purple Knight	The Purple Knight AD security assessment is helping IT and security teams take action to improve overall security, including addressing AD misconfigurations, changing structure and processes to improve overall security posture, and facilitating conversations between IT and security teams.	AD audit	Yes
Qualys (VMDR)	Qualys is a cloud-based solution that detects vulnerabilities on all networked assets, including servers, network devices (e.g. routers, switches, firewalls, etc.), peripherals (such as IP-based printers or fax machines) and workstations. Qualys can assess any device that has an IP address.	Vulnerability scanning	No

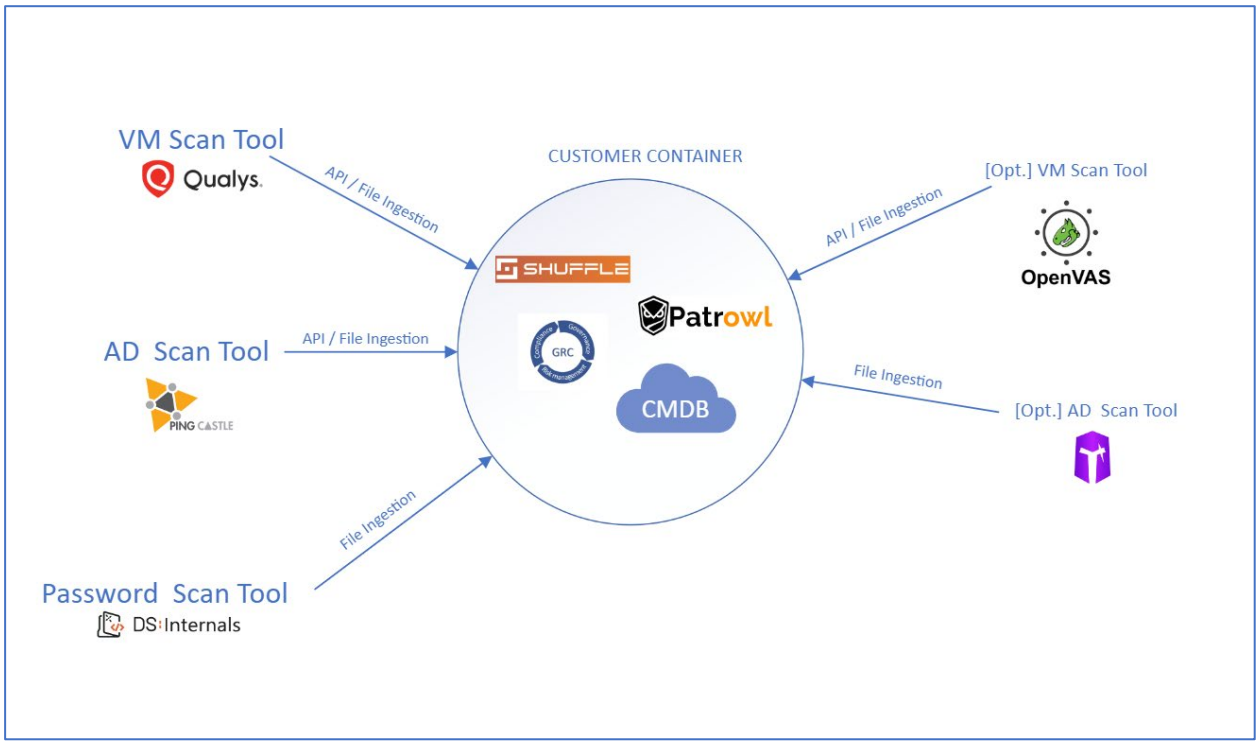
DSInternals has been selected as the main password checking tool, due to the possibility of checking the hashes against an offline database (updated regularly) of already leaked passwords on the internet (amongst other useful checks on accounts level).

PingCastle and **PurpleKnight**(backup) have been selected as the main AD audit tools. PingCastle will be used as the main tool due to the possibility of integrating it with the complete solution (API).

Qualys has been selected as the main vulnerability scanning tool and **Greenbone Community Edition (OpenVAS)** as a backup. The main reason for selecting Qualys is the quality of the scan results and the experience we have with the tool.

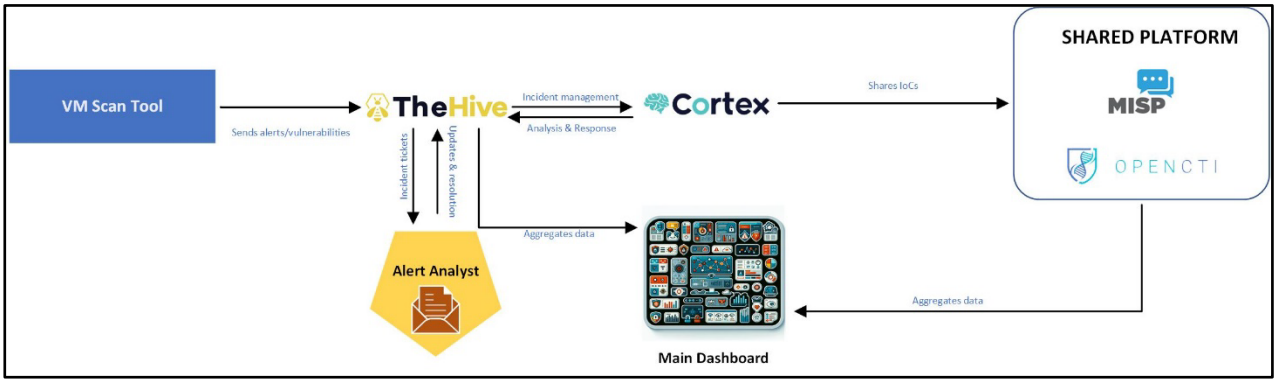
The results from the selected toolset will reside in the customers tenants.

Depending on the size of the customer, these will be ingested in the local customers container (CMDB, specific GRC tool, Vulnerability management solution, etc.).



4.2.5.6 Integration with CTI toolsets and ticketing solutions

Integration with CTI toolsets

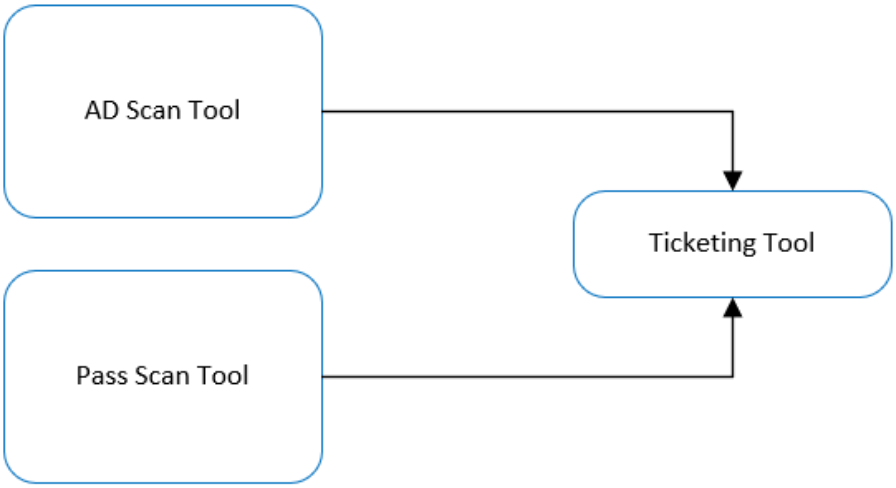


Components and Their Roles:

1. **VM Scan Tool:** Scans for vulnerabilities and threats, acting as an initial data collection point.
   VM Scan Tool -> The Hive:
   - The VM Scan Tool detects vulnerabilities and sends alert to The Hive
   - The Hive creates and managed incident cases based on these alerts.

2. **The Hive:** Central incident management and response platform.
   The Hive <-> Cortex:
   - The Hive forwards cases to Cortex for further analysis.
   - Cortex returns enriched information and automated response recommendations to The Hive.
3. **Cortex:** Analysis and response engine that enriches and automates responses to incidents.
   Cortex <-> MISP/OpenCTI:
   - Cortex queries MISP and OpenCTI for additional intelligence and context during analysis.
   - Shares analysis results and derived IoCs with MISP and OpenCTI for community-wide benefit.
4. **MISP:** Specialized in sharing malware information and indicators of compromise (IoCs).
5. **OpenCTI:** Broadly focused on sharing and managing cyber threat intelligence.
6. **Main Dashboard:** A centralized visualization and control panel that aggregates data from all tools for oversight and management.
   - Aggregates data from all platforms (VM Scan Tool, The Hive, Cortex, MISP, OpenCTI).
   - Provides a unified view of incidents, vulnerabilities, ongoing analyses, and overall threat landscape.
   - Enables central management and oversight, with capabilities to drill down into specific incidents or trends.
   -

Integration with customer ticketing solutions



4.2.5.7 VM Toolset system requirements

The provisioning of VM toolset is a mix of virtual machines deployed in the customer SOCaaS tenants or SIEMBIOT shared services platform.

Tools deployed on demand in the customer demo SOCaaS tenants

DSInternals

HDD: 100 GB
OS: Windows Server with customer AD connectivity
CPU: 2 vCPUs, 8 GB RAM

PingCastle and **PurpleKnight**(backup)

HDD: 5 GB storage for reports storing (history)
OS: Windows Server with customer AD connectivity
CPU: 2 vCPUs, 8 GB RAM

Qualys

Small Customers:

HDD: 40 GB per scanner
OS: Independent
CPU: 4 vCPUs and 8 GB RAM per scanner (ratio 1:2 CPU:RAM)
Special requirement: Internet access for sending/receiving the data to the cloud platform.

Medium/Large/Extra-large Customers:

HDD: 40 GB per scanner
OS: Independent
CPU: 8 vCPUs and 16 GB RAM per scanner (ratio 1:2 CPU:RAM)
Special requirement: Internet access for sending/receiving the data to the cloud platform.

OpenVAS

Small Customers:

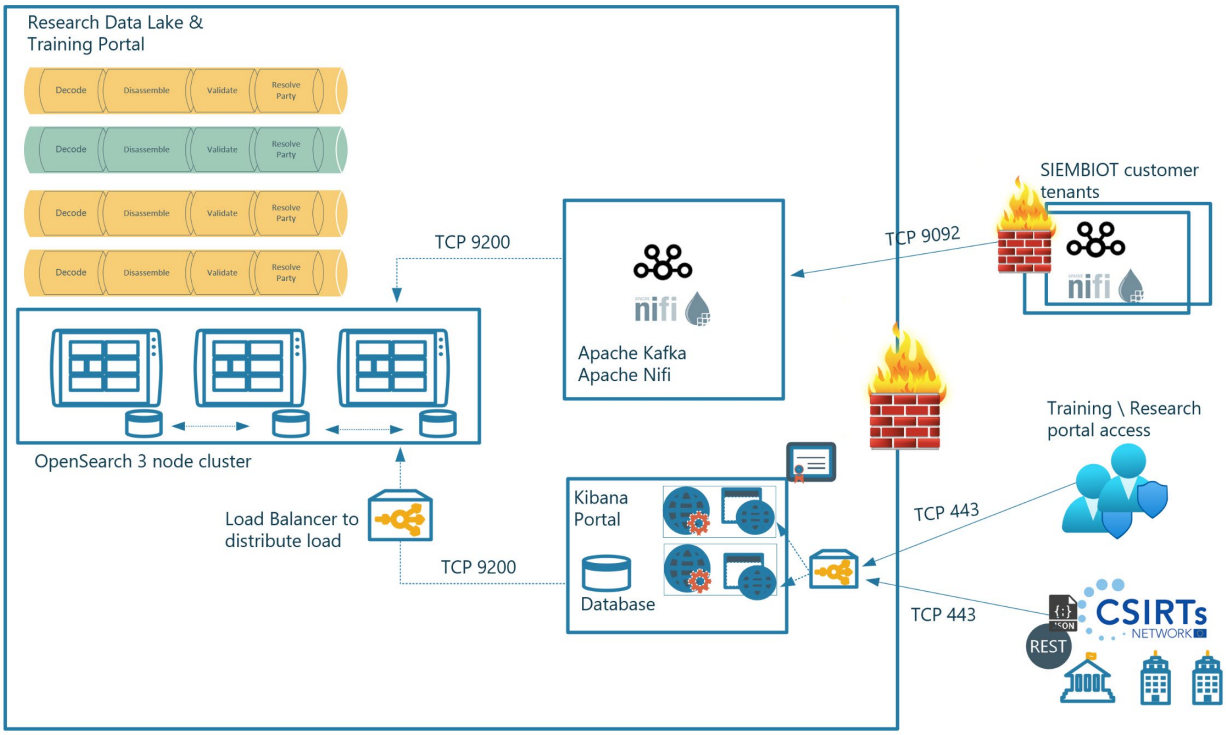
- Platform:
  Sizing: 135 GB
  OS: 64-Bit version Linux OS
  CPU: 2 vCPUs and 8 GB Ram
- Scanner:
  Sizing: 70 GB
  OS: 64-Bit version Linux OS
  CPU: 2 vCPUs and 8 GB Ram

Medium/Large/Extra-large Customers:

- Platform:
  Sizing: 225 GB
  OS: 64-Bit version Linux OS
  CPU: 12 vCPU cores and 24 GB Ram
- Scanner:
  Sizing: 140 GB
  OS: 64-Bit version Linux OS
  CPU: 4 vCPUs and 16 GB Ram

4.3 SIEMBIOT Training & Research

4.3.1 High Level Design



The components included in the SIEMBIOT training and research platform are:

A. Ingestion layer (Apache NiFi, Kafka)
B. Storage layer: 3 node ElasticSearch/OpenSearch cluster
C. Portal layer: NGIX, Kibana, .NET Core, SQL

4.3.2 Training and Research Platform Sizing

The solution will be hosted on the same physical servers as the SIEMBIOT shared services platform.

The expected resource allocation is:

Components	vCPUs	RAM (GB)	Storage (GB)
OpenSearch Cluster	96	384	50 000
2 x Kibana modes	32	128	1024 + 1024 local cache
2 x .NET web servers	32	128	200 + 1024 local cache
Nifi, Kafka	24	96	1024 local cache
Total	184	736	50000 SAN + 2TB local

For the vCPUs we consider the same oversubscription 1:1.5 (physical:virtual) allocation.

The RAM memory will be reserved so we avoid unnecessary data moves from the hypervisor (ballooning) or guests OSs (memory swapping)

4.3.3 Platform High Availability

The shared services and the training and research platform hosted on 4 physical servers will benefit of N+1 clustering.

High availability in a virtualized environment ensures uninterrupted access to applications and services by employing redundancy and failover mechanisms. By leveraging clusters of

interconnected servers (nodes) and shared storage, it safeguards against hardware failures or maintenance downtime. This setup allows for seamless migration of virtual machines between hosts, automatic restarts of affected VMs on healthy nodes, and continuous monitoring to maintain service availability, minimizing disruptions and ensuring business continuity.

4.4 Integrations / Traffic Flows

Source	Target & Zone	Protocol	Description
SIRP Tenant	Threat Intel – shared platform	TCP	Threat Intel GET
SIRP Tenant	Threat Intel – shared platform	TCP	Threat Intel PUT
SIRP Tenant	Analyzer – shared platform	TCP	Analyzers/Responders communication
Worker Node Tenant	Endpoint Agent – public endpoint	TCP	Endpoint Agent monitoring & event collection
Worker Node Tenant	Syslog listener - S2S VPN	TCP/UDP	Syslog collector for Network devices
Worker Node Tenant	Public Cloud / Internet	TCP	Public Cloud datalake for cloud events
Worker Node Tenant	Public Cloud – S2S VPN	TCP	Events from Private infrastructure (eg.private Kafka instances)
Integration Tenant	Integration – research	TCP	Transport for research sample data
Threat Intel Shared	Threat Intel – public	TCP	Threat Intel Sharing

4.5 Security architecture principles

4.5.1 Tenants' data segregations principles

In complex environments, especially in cloud computing, ensuring the separation of tenant data requires sophisticated strategies:

Multi-Instance Architecture:

- Instead of a shared database, each tenant has its own instance of the software and data storage. This provides physical and logical data isolation but can be resource intensive.

Advanced Encryption Techniques:

- Employing advanced encryption methods, which allows data to be processed while still encrypted, thereby enhancing data security.

Containerization:

- Using technologies like Docker and Kubernetes can isolate tenants' applications and environments, providing an additional layer of segregation and security.

Database Sharding:

- This involves splitting databases into smaller, faster, more easily managed parts, or shards, based on tenant.

Zero Trust Model:

- Implementing a Zero Trust framework ensures rigorous identity verification for every person and device trying to access resources on a private network, regardless of whether they are sitting within or outside of the network perimeter.

4.5.2 Data Privacy principles

Data privacy principles form the foundation of ethical and responsible handling of personal information. These principles guide organizations, businesses, and individuals in safeguarding the confidentiality, integrity, and accessibility of personal data.

Integrity and Confidentiality:

- Organizations must implement security measures to protect personal data from unauthorized access, disclosure, alteration, and destruction. This includes encryption, access controls, and regular security assessments.

Accuracy:

- Personal data should be accurate, and reasonable steps should be taken to ensure that inaccurate or incomplete data is rectified or erased promptly.

Data Minimization:

- Only the minimum amount of personal data necessary for the intended purpose should be collected and processed. Unnecessary data should not be collected or retained.

Data Masking:

- Using data obfuscation techniques to protect sensitive information from unauthorized access while maintaining the data's usability.

Accountability:

- Data controllers (organizations or individuals who determine the purpose and means of processing personal data) are responsible for complying with data protection principles. They should implement measures to demonstrate and document compliance.

Storage Limitation:

- Personal data should not be kept for longer than necessary for the purpose for which it was collected. Organizations should establish and adhere to appropriate data retention periods.

Fair and lawful use:

- Personal data should be processed lawfully, fairly, and transparently. Individuals should be informed about how their data will be used, and processing should adhere to applicable laws and regulations.

Purpose Limitation:

- Personal data should be collected for specified, explicit, and legitimate purposes. Any further processing of the data should be compatible with the original purpose.

User Rights:

- Individuals have rights regarding their personal data, including the right to access, correct, delete, or restrict the processing of their data. Organizations should facilitate the exercise of these rights.

Consent:

- Where processing is based on consent, individuals should be given clear, specific, and freely given information about what they are consenting to. Consent can be withdrawn at any time.

Privacy by Design and by Default:

- This approach integrates data privacy into system designs, business practices, and IT infrastructure from the outset, not as an afterthought.

4.5.3 Network traffic segregation

Network segregation is a process that separates critical network elements from the internet and other less sensitive networks. It allows IT teams to control traffic flow between various subnets based on granular policies.

Methods to segregate network traffic:

1. **Virtual LANs (VLANs)** - configure VLANs on managed switches and assign specific ports to each VLAN.
    - Assign each tenant to a separate VLAN. This ensures that broadcast domains are isolated, and tenants cannot directly communicate with each other at the data link layer.
    - Use VLAN tagging to identify and separate tenant traffic on shared network infrastructure.

2. **Subnetting** - Use a router to connect different subnets and control traffic between them.
    - Allocate different IP address subnets for each tenant. This helps in segregating traffic at the network layer.

3. **VRF Lite** - Virtual Routing and Forwarding or VRF allows a router to run more than one routing table simultaneously. When running more routing tables at the same time, they are completely independent. VRFs are typically used in combination with MPLS VPNs. VRFs without MPLS is called VRF lite.

4. **Firewalls** - Deploy firewalls to control traffic flow between different segments of the network.
    - Firewalls can be placed at the network perimeter, between VLANs, or even at the subnet level to control traffic flow.
    - Define rules that explicitly permit or deny communication between tenants based on business requirements and security considerations.

5. **Network Access Control (NAC)** - Deploy NAC systems to authenticate and authorize devices before allowing them to connect to the network.
    - Use NAC solutions to authenticate and authorize devices belonging to each tenant before allowing them to connect to the network.
    - Enforce policies that define the level of access each tenant has and restrict unauthorized devices.

6. **Security Zones** - Use routers or firewalls to create security zones and define policies for inter-zone communication.
    - Define security zones for each tenant and implement policies that control traffic flow between these zones.

7. **Encryption** – use encryption mechanisms for communication between devices.
    - Enforce encryption for data in transit and data at rest, especially for sensitive information belonging to each tenant.
    - Implement protocols such as TLS for secure communication.

8. **Traffic Monitoring and Analysis** - Utilize network monitoring tools to keep track of traffic patterns and detect unusual behaviour.
    - Regularly review logs and reports to detect any anomalies or security incidents.

9. **Segmentation by Role or Function** - Use network design principles to logically separate devices based on their roles or functions.
    - Implement RBAC to control access to network resources and services based on the roles of users within each tenant.

- Ensure that only authorized individuals within a tenant have access to specific resources.

4.5.4 Authentication and Authorization

Combining firewalls and jump servers in a multilayered security strategy can significantly enhance access control, authentication, and authorization within a network environment.

Access Control:

Firewalls: Deploy firewalls at the network perimeter and within the internal network to control incoming and outgoing traffic. This establishes barriers between different network segments, allowing administrators to define rules for traffic flow and access rights.

Authentication:

Jump Servers: Serve as an intermediary between users and critical internal resources. Access to sensitive systems or data is only possible by first accessing the jump server. Users must authenticate themselves at the jump server before initiating connections to other internal systems.

Authorization:

Firewalls: Implement rules within firewalls to restrict access based on IP addresses, protocols, ports, and other contextual information. These rules control traffic flow not just between zones but also between users and specific resources or services.

Enhanced Security Measures:

Multilayered security: Passing from one layer to another will require authentication on a different user realm.

VPN Access to Jump Servers: Require users to establish a VPN connection to the jump server before accessing internal resources, adding an extra layer of security.

Logging and Monitoring: Enable logging and monitoring on both firewalls and jump servers to track access attempts, detect anomalies, and respond to potential security incidents promptly.

By integrating jump servers as intermediary access points alongside firewalls, we'll enforce stricter controls over who accesses critical assets, mitigate potential risks of unauthorized access, and ensure a more robust defence against various cyber threats.

4.5.5 Security Protection

SIEMBIOT ensures that it offers in depth defense and security by design, protecting against security risks at different levels:

- up to date binaries,
- active antivirus and antimalware,
- intrusion detection,
- intrusion prevention,

- application aware firewalls,
- firewall rules based on live CTI feeds,

The protection is multi-layered.

1. Internet facing firewall cluster with IDS and IPS modules enables.
2. Reverse Proxy + Application aware firewall for training & research portal
3. Web servers with the latest security patches and Realtime antimalware / antivirus.
4. Internal segregation and traffic analysis between different security zones & tenants.

4.5.6 Audit features

The platform will be audited on different layers.

- System logs enabled with 365 events retention.
- Web servers logging enabled with 365 days retention.
- Administrative actions audited.
- Journaling on Linux servers.
- Network configuration auditing enabled.

4.5.7 GDPR considerations

SIEMBIOT respects key principles of GDPR, such as data minimization, consent, data subject rights, transparency, and security, are applied to ensure compliance and protect individuals' data privacy.

Data Transfer: Be cautious when transferring data across borders, especially to countries that do not have equivalent data protection laws.

Accountability and Governance: Maintain records of data processing activities, conduct data protection impact assessments for high-risk processing, and appoint a data protection officer if necessary.

Breach Notification: In case of a data breach, GDPR mandates prompt notification to the relevant supervisory authority and, in certain cases, to the data subjects.

Data Processing Agreements: When using third-party services for data processing, ensure that they comply with GDPR through binding agreements.

Anonymization and Pseudonymization: Wherever possible, use techniques like anonymization and pseudonymization to protect the identity of data subjects.

Training and Awareness: Regularly train staff involved in data processing on GDPR requirements and data protection best practices.

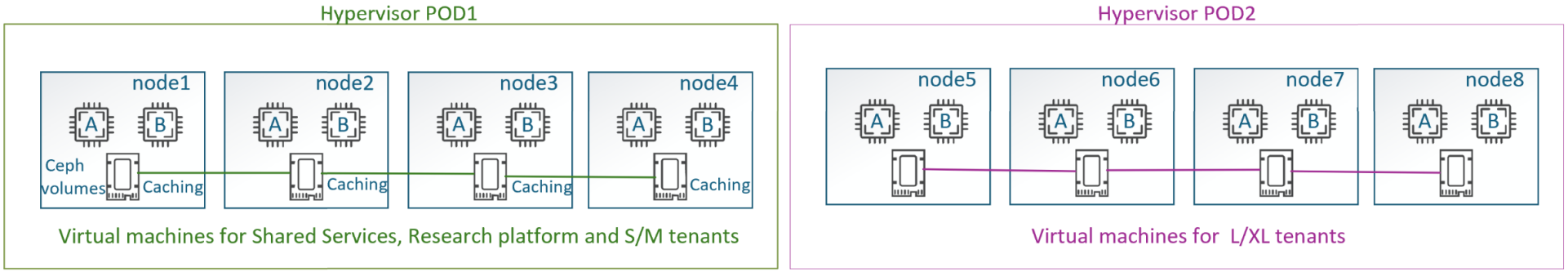
Legal Basis for Processing: Identify and document the legal basis for each data processing activity, such as consent, contract necessity, legal obligation, vital interests, public task, or legitimate interests.

5 HOSTING INFRASTRUCTURE

In order providing flexibility and high availability we envisaged virtualizations for all layers: processing, caching, network & storage.

The design is based on 8 hypervisor nodes, each node offering up to 2 processing hosts and one storage host for caching.

The nodes are split in two different clusters / pods, one for shared services, anonymized data lake and small/medium tenants while the 2nd one for large and extra large tenants.

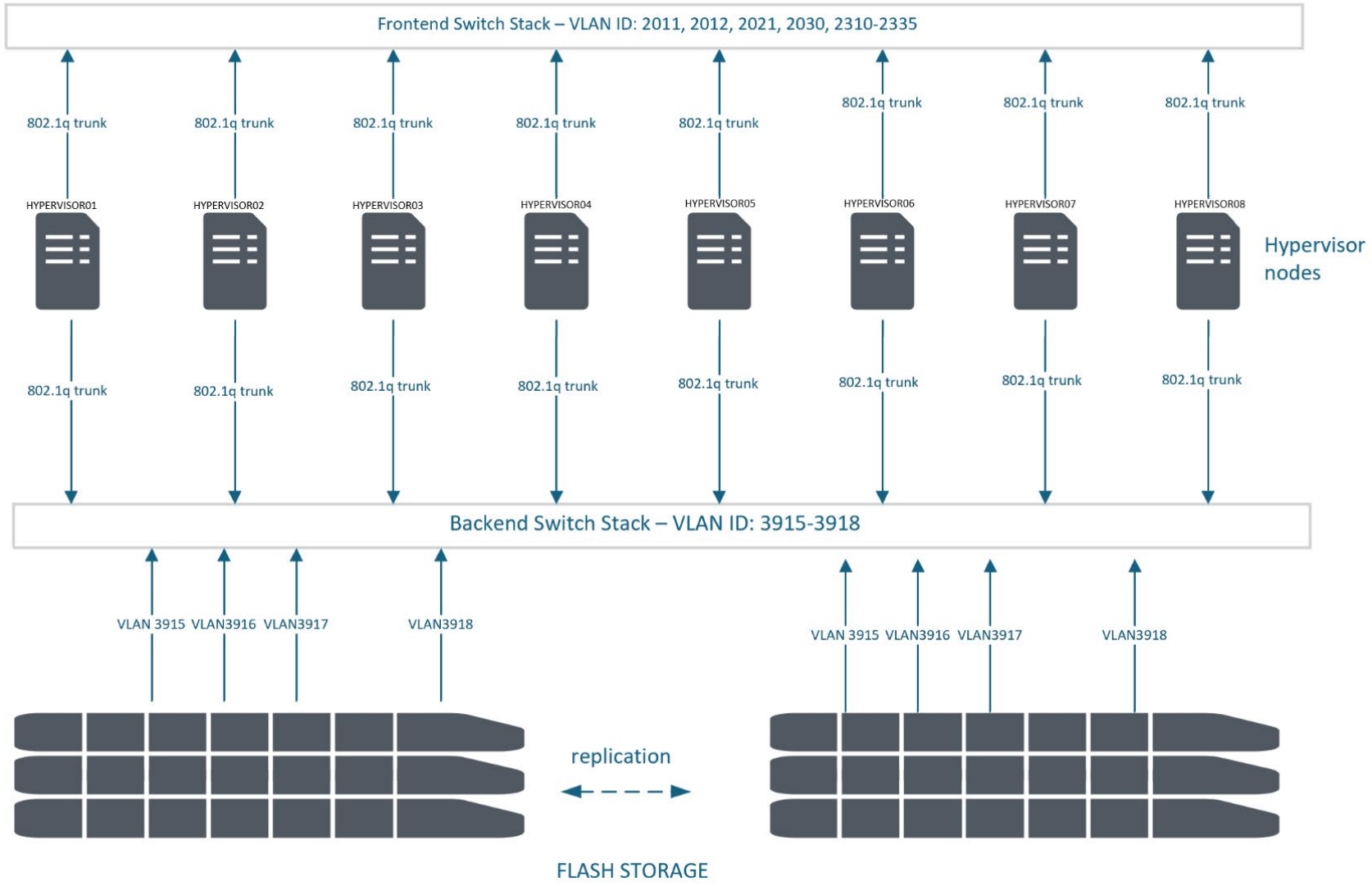


There is a physical segregation between frontend and backend traffic.
Frontend traffic includes all application, client aware connections while the backend traffic includes only the redundant iSCSI connections towards the Storage Area Network (SAN) enclosures.

All Hypervisor nodes have redundant connections to frontend and backend switch stacks via 802.1x trunks.

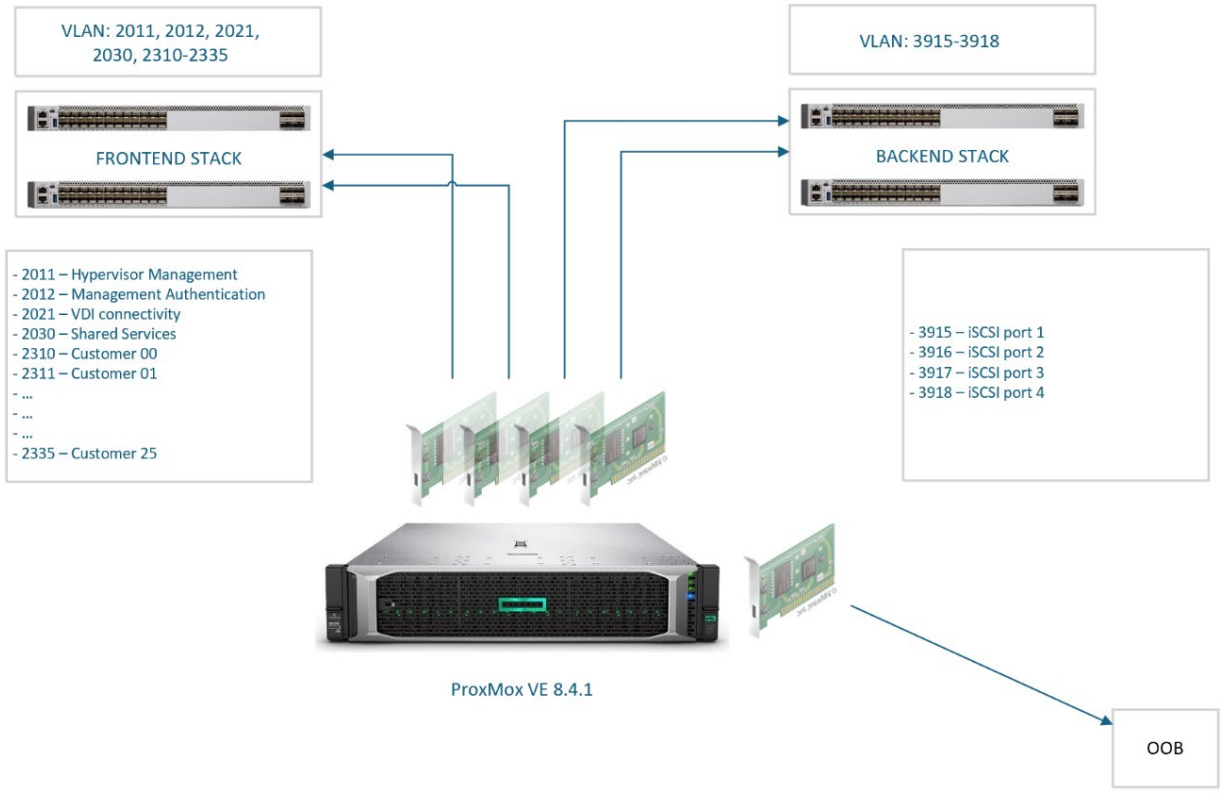
As virtualization solution we select open source hypervisor Proxmox VE as it offers the features required: clustering, high availability, migrations, snapshots, replication, virtual CPU overcommitment, virtual networking (Open vSwitch), virtual storage (logical volumes, Ceph/RBD, ZFS over iSCSI), build-in firewall.

At the time of writing we had Proxmox VE 8.4.1 based on Debian GNU/Linux, free software license AGPL v3.



Layer 2 diagram 1

5.1 Hypervisor host detailed diagram



There is full isolation (different vlans) for the hypervisor management, authentication, VDI connectivity, shared services and customers.

Each host has an out-of-band (OOB) management interface connected on a separate physical infrastructure and accessible only from privileged access management (PAM) stations.

Traffic segregation

At the hypervisor\virtualization level, each server has 4 physical adapters configured as follows:

EtherChannel1: NIC1 + NIC3, 2 x 25Gbps ports, connected to the backend switch stack

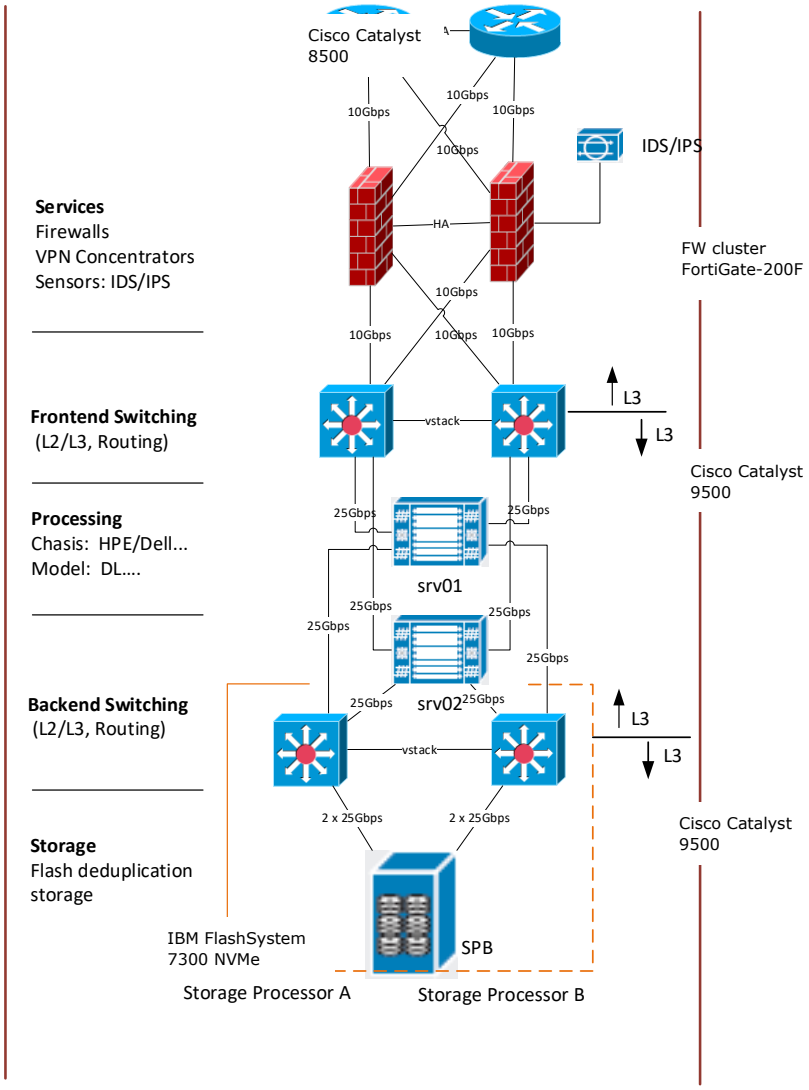
VLANs configured: Core IT Services vlans, Storage replication and access vlans.

EtherChannel2: NIC2 + NIC4, 2 x 25Gbps ports, connected to the core switch stack.

VLANs configured: SIEMBIOT Shared Services vlans, SIEMBIOT beneficiaries' tenants vlans

For each server deployed, EtherChannel 1 is physically separated from EtherChannel 2 ensuring an extra level of protection for the core IT services like: DNS, IT Management, OOTB, Storage access and replication.

5.2 Networking services deployment

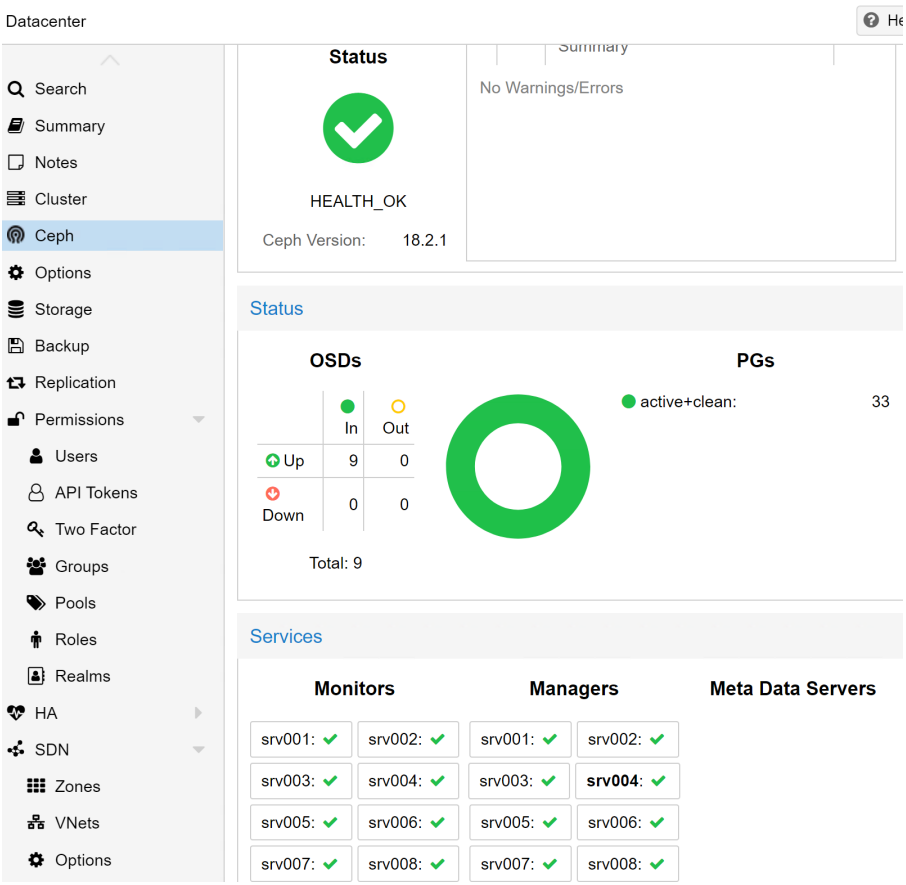


5.3 Hypervisor storage connectivity

5.3.1 vSAN configuration

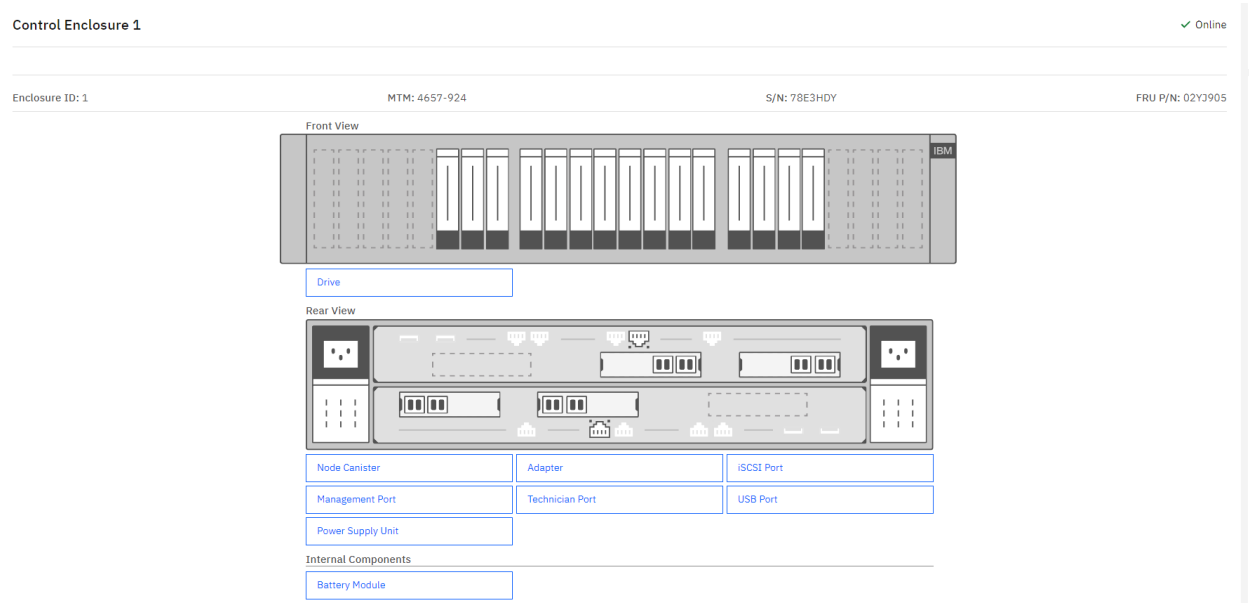
Local storage installed on the storage hosts will be configured to take part in a virtual SAN (vSAN) configuration to hold the transient and caching data. It will use the 2 x 25Gbps management EtherChannel to replicate provisioned disks on peer storage hosts from the same cluster.

Each virtualization POD contains 4 storage nodes which provides a usable capacity of 243.2 TB protected storage capable. The vSAN disk is presented to all nodes in the POD (8+) regardless of they hold local storage or not.

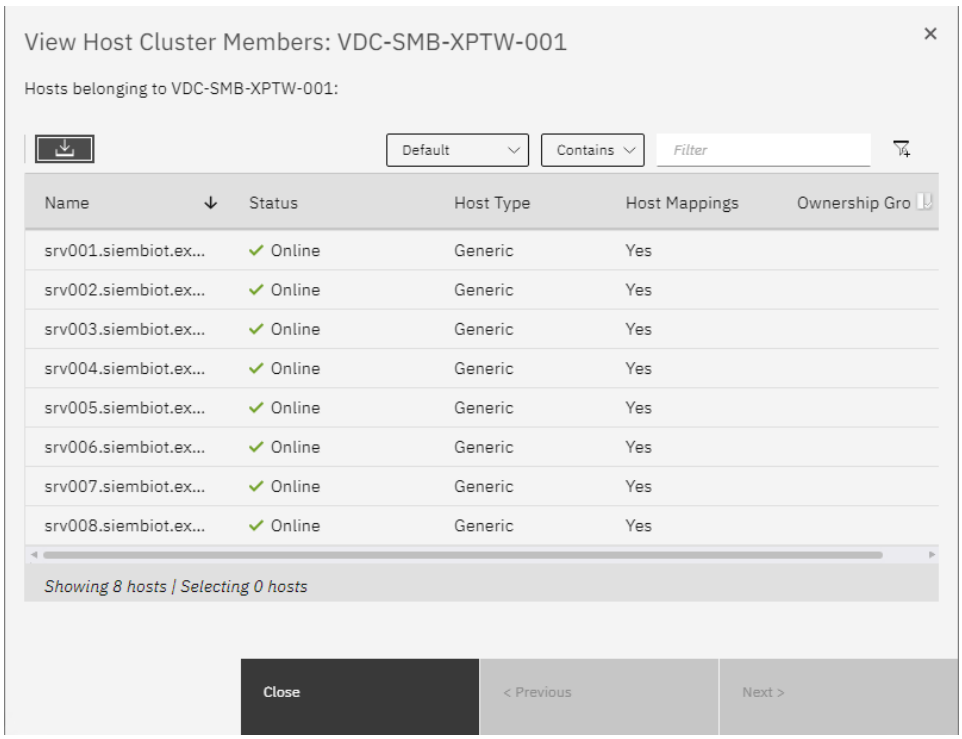


5.3.2 Shared SAN configuration

Siembiot platform is hosted on **2 x IBM Flashsystem 7300 devices,** configured with **200Tb** available for the workloads.
Each SAN device is configured with dual storage processors for high availability, each storage processor having installed 4 x 25Gbps ports for ISCSI traffic, totalling 100Gbps per storage processor available for storage traffic from/towards hypervisors.



Virtual volumes presented to hypervisor POD1


View Host Cluster Members: VDC-SMB-XPTW-001
Hosts belonging to VDC-SMB-XPTW-001:
Default
Contains
Filter
Name
Status
Host Type
Host Mappings
Ownership Gro
srv001.siembiot.ex... Online Generic Yes
srv002.siembiot.ex... Online Generic Yes
srv003.siembiot.ex... Online Generic Yes
srv004.siembiot.ex... Online Generic Yes
srv005.siembiot.ex... Online Generic Yes
srv006.siembiot.ex... Online Generic Yes
srv007.siembiot.ex... Online Generic Yes
srv008.siembiot.ex... Online Generic Yes
Showing 8 hosts | Selecting 0 hosts
Close
< Previous
Next >

6 PLATFORM SECURITY ARCHITECTURE

6.1 Physical security

SIEMBIOT physical IT infrastructure is hosted in a dedicated computer room isle located at -1 and separated from the Expertware core IT isles.

Physical perimeter outside the building and inside benefits of 24x7 security, video and fire protection. Any incidents will trigger automatic escalation to the facilities manager on-call.

Access to the building is permitted only based on identity card + personal alarm code.

There are multiple gates where the ID card must be presented: entrance, elevator, computer room, SIEMBIOT isle. ID Cards are individual and there is a central access management application to grant access for each of the of the specific gates.

All actions are logged and audited.

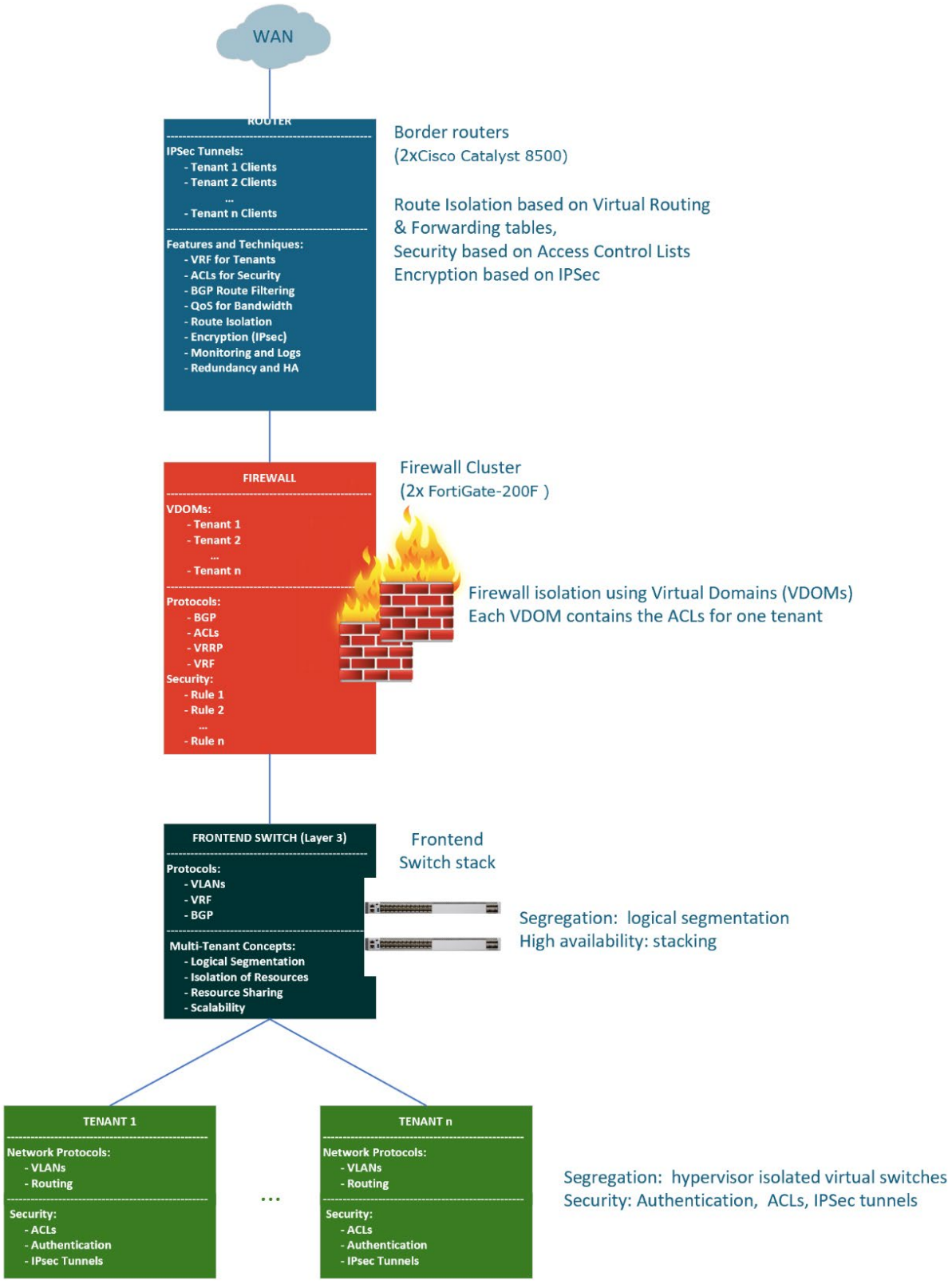
6.2 Network traffic segregation

Besides the physical separation we enforced network traffic SIEMBIOT hypervisors are separated from the Expertware IT Infrastructure.

Routing segregation among tenants is based on dedicated Cisco's Virtual Routing & Forwarding (VRF) tables. Traffic between different vlans in the same VRF is controlled by security ACLs .

Traffic between tenants, shared services, management layer and beneficiaries is evaluated and controlled on dedicated Virtual Domains created on the firewall cluster (drawings below).

Hypervisor networking segregation is based on dedicated virtual switches (vSwitch) per tenant allowing only the tenat's vlans/subnets.


WAN
ROUTER
IPSec Tunnels:
- Tenant 1 Clients
- Tenant 2 Clients
...
- Tenant n Clients
Features and Techniques:
- VRF for Tenants
- ACLs for Security
- BGP Route Filtering
- QoS for Bandwidth
- Route Isolation
- Encryption (IPsec)
- Monitoring and Logs
- Redundancy and HA
Border routers
(2xCisco Catalyst 8500)
Route Isolation based on Virtual Routing
& Forwarding tables,
Security based on Access Control Lists
Encryption based on IPSec
FIREWALL
VDOMs:
- Tenant 1
- Tenant 2
...
- Tenant n
Protocols:
- BGP
- ACLs
- VRRP
- VRF
Security:
- Rule 1
- Rule 2
...
- Rule n
Firewall Cluster
(2x FortiGate-200F)
Firewall isolation using Virtual Domains (VDOMs)
Each VDOM contains the ACLs for one tenant
FRONTEND SWITCH (Layer 3)
Protocols:
- VLANs
- VRF
- BGP
Multi-Tenant Concepts:
- Logical Segmentation
- Isolation of Resources
- Resource Sharing
- Scalability
Frontend
Switch stack
Segregation: logical segmentation
High availability: stacking
TENANT 1
Network Protocols:
- VLANs
- Routing
Security:
- ACLs
- Authentication
- IPsec Tunnels
...
TENANT n
Network Protocols:
- VLANs
- Routing
Security:
- ACLs
- Authentication
- IPsec Tunnels
Segregation: hypervisor isolated virtual switches
Security: Authentication, ACLs, IPSec tunnels

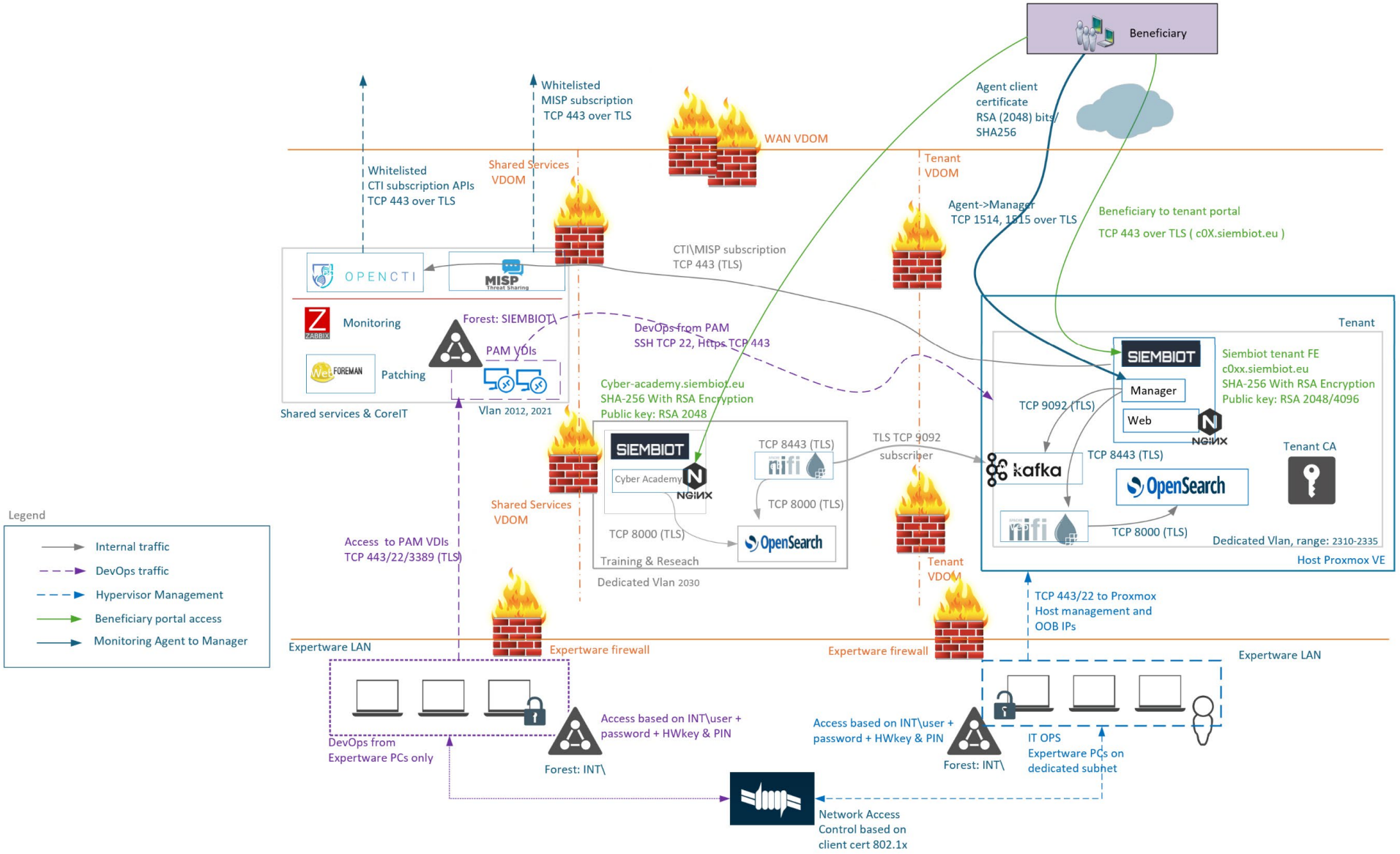
6.3 Communication flows

The diagram below shows the main communication flows: tenant <-> shared services, tenant <-> research data lake, beneficiaries <-> SIEMBIOT tenant & Cyber-Academy portal, IT Management access, SOC Analysts Team access.

The design minimizes the interactions among different security zones as follows:

Name	Description	Protocol and Ports	Virtual firewalls
Agents <-> Managers	SIEMBIOT agents communicate with the corresponding manager. Agents use Client authentication certificate issued from tenant CA.	TCP 1515, 1515 over TLS	WAN VDOM C0x VDOM
Beneficiaries' portal access	SIEMBIOT beneficiaries and Partners access their own tenant and the Cyber-Academy	TCP 443 over TLS	WAN VDOM C0x VDOM
SIEMBIOT Internal traffic	Tenants pulling CTI from Shared services over TLS Cyber academy pulling anonymized events from tenants	TCP 443 over TLS TCP 9042 over TLS	C0x VDOM Shared Services VDOM
DevOps	From Expertware LAN connecting to the Siembiot Shared Services Privileged Access Management VDIs	TCP 443, 22, 3389 over TLS	Shared Services VDOM Expertware LAN VDOM
IT Management	From Expertware LAN connecting to the Siembiot virtualization hosts, the Proxmox management portal, SIEMBIOT networking equipment and SAN over dedicated management network.	TCP 443, 22 over TLS	Expertware LAN VDOM

For the sake of simplicity we did not show all the tenant internal communication taking place as those are completely isolated from each other's, the shared services and the Expertware LAN.


Beneficiary
Whitelisted
MISP subscription
TCP 443 over TLS
WAN VDOM
Agent client
certificate
RSA (2048) bits/
SHA256
Whitelisted
CTI subscription APIs
TCP 443 over TLS
Shared Services
VDOM
Tenant
VDOM
Agent->Manager
TCP 1514, 1515 over TLS
Beneficiary to tenant portal
TCP 443 over TLS (c0X.siembiot.eu)
CTI\MISP subscription
TCP 443 (TLS)
OPENCTI
MISP
Threat Sharing
Monitoring
Forest: SIEMBIOT\
DevOps from PAM
SSH TCP 22, Https TCP 443
PAM VDIs
Patching
Shared services & CoreIT
Vlan 2012, 2021
Tenant
SIEMBIOT
Siembiot tenant FE
c0xx.siembiot.eu
SHA-256 With RSA Encryption
Public key: RSA 2048/4096
Manager
Web
NGINX
TCP 9092 (TLS)
Cyber-academy.siembiot.eu
SHA-256 With RSA Encryption
Public key: RSA 2048
SIEMBIOT
Cyber Academy
NGINX
TCP 8443 (TLS)
nifi
TLS TCP 9092
subscriber
kafka
TCP 8443 (TLS)
Tenant CA
OpenSearch
TCP 8000 (TLS)
Shared Services
VDOM
TCP 8000 (TLS)
OpenSearch
Training & Reseach
Dedicated Vlan 2030
Tenant
VDOM
Dedicated Vlan, range: 2310-2335
Host Proxmox VE
Access to PAM VDIs
TCP 443/22/3389 (TLS)
TCP 443/22 to Proxmox
Host management and
OOB IPs
Legend
Internal traffic
DevOps traffic
Hypervisor Management
Beneficiary portal access
Monitoring Agent to Manager
Expertware LAN
Expertware firewall
Expertware firewall
Expertware LAN
DevOps from
Expertware PCs only
Access based on INT\user +
password + HWkey & PIN
Forest: INT\
Access based on INT\user +
password + HWkey & PIN
Forest: INT\
IT OPS
Expertware PCs on
dedicated subnet
Network Access
Control based on
client cert 802.1x

6.3.1 Securing data in transit

All communications are encrypted, performed over TLS 1.2 using the combination: RSA with key length of 2048/4096 bit, SHA-256

Cipher suites used:

TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256:

Uses ECDHE for key exchange, ECDSA for authentication, AES-128 for encryption (GCM mode), and SHA-256 for message authentication.

TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384:

Uses ECDHE for key exchange, RSA for authentication, AES-256 for encryption (CBC mode), and SHA-384 for message authentication.

6.3.2 Securing data at rest

All virtual Linux machines use Linux Unified Key Setup (LUKS) encrypted partitions. LUKS uses Advanced Encryption Standard (AES) algorithm to scramble the data. The decryption passphrases are kept in a dedicated Privileged Identity Management (PIM) key vault.

Each tenant has different encryption keys.

All workstations and laptops used by DevOps, IT Management or SOC teams are based on Windows 11 and have all disks encrypted using corporate BitLocker. There are active controls via GPOs and CrowdStrike agents blocking the use of unencrypted USB drives.

6.4 DevOps, IT Management and SOC analysts' access

There is layered protection before granting access to the SIEMBIOT tenants for SOC, DevOps or IT Infrastructure management.

The process and various security controls enforced are:

1. The LAN, the Wi-FI and the VPNs networks are protected by an existing Network Access Control (NAC) solution which checks the device certificate, and the AD user credentials before granting access to the LAN. All violations are logged, and they trigger alerts on the Expertware SIEMBIOT tenant.
2. The engineers can connect only from an Expertware corporate device over LAN or via corporate AlwaysON VPN.
3. Acces to corporate PCs is protected by dual factor authentication:
    a. The users must supply the AD INT\ credentials and

   b. present a hardware key connected to a USB-C port and provide a personal PIN (6+ digits).
4. All devices are protected by CrowdStrike and SIEMBIOT agents.
5. For remote users AlwaysOn VPN is enforced before logon based on a 802.1x client certificate so even if outside the corporate network the communication to Expertware LAN is tunneled over the VPN.
6. A dedicated LDAP forest & domain SIEMBIOT\ controls access to SIEMBIOT shared services virtual machines
7. Only from Shared privileged access VDIs we allow connectivity to the tenants' virtual machines.

6.5 Data protection

Backup policies protecting SIEMBIOT shared services virtual machines

- Weekly full backups for VM disks – retention 2 months
- Daily incremental – retention 30 days

For the SIEM data lakes we configured asynchronous replication (one day delay) to the 2nd SAN enclosure combined with monitoring anomalies (deletion, heavy re-writes).

An archiving process to a lower tiering storage is envisaged for tenants as from month 13.

6.6 Security Monitoring

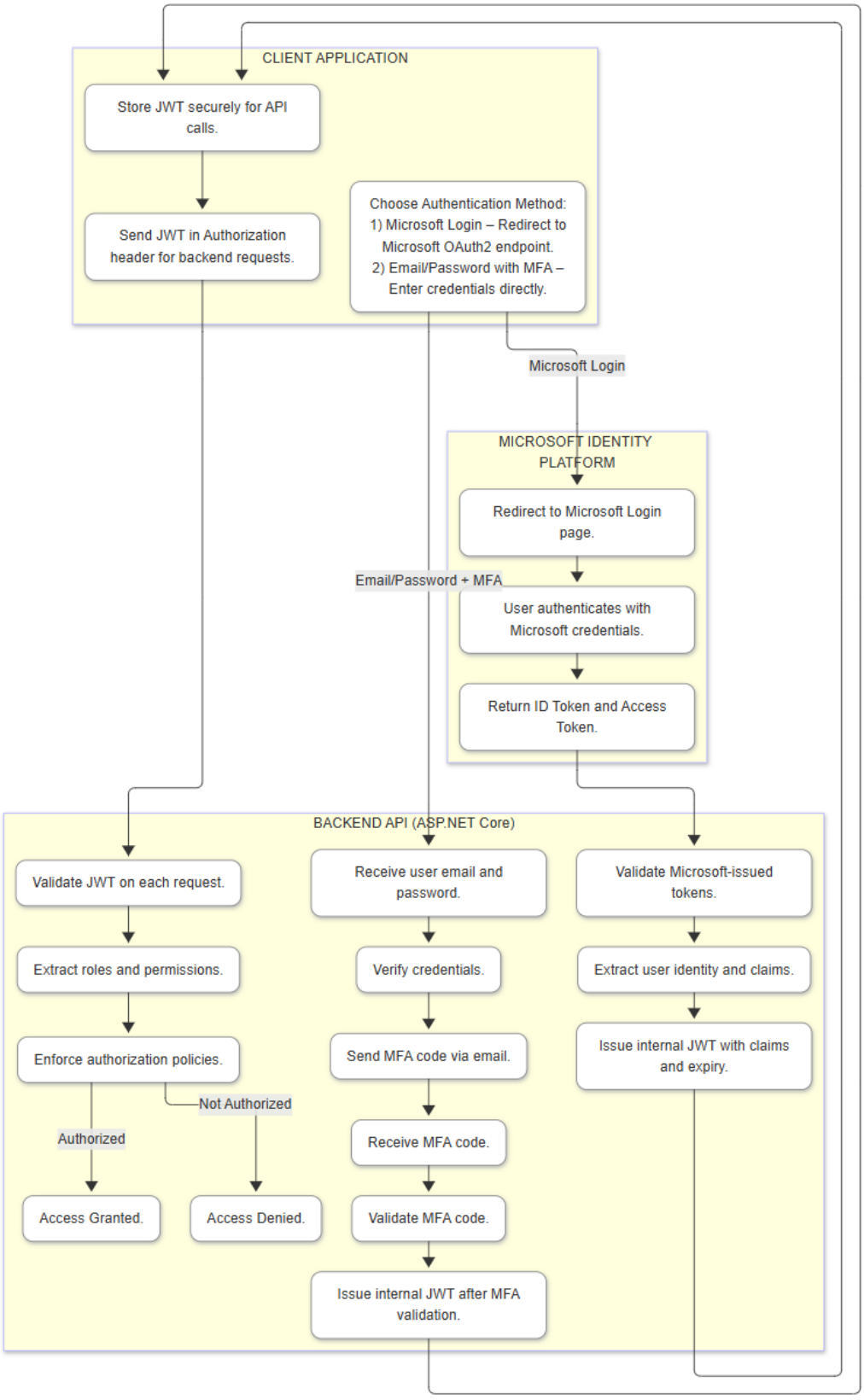
In the SIEMBIOT shared services zone we deploy both 24x7 NOC and SOC monitoring the underlying infrastructure via an internal SIEMBIOT instance. Availability or security alerts are handled by the NOC & SOC team.

We maintain a monthly patching cycle for the underlying hypervisor and virtual machines.

7 WEB APPLICATION SECURITY ARCHITECTURE

SIEMBIOT tenant portals and the SIEMBIOT Cyber-Academy for training & research web application are developed using .NET framework.

7.1.1 Web App Authentication Flows



It implements a robust and flexible authentication and authorization mechanism that supports both federated identity via Microsoft Login and traditional credentials with email-based Multi-Factor Authentication (MFA). It leverages JWT (JSON Web Tokens) to securely authorize users across its backend services.

7.1.2 Authentication Mechanisms

A. SSO based on Microsoft Login

Both web applications allow federated authentication with Microsoft Azure through OAuth 2.0 and OpenID Connect protocols. Upon choosing this login option:

- The user is redirected to the Microsoft Identity Platform.
- After successful authentication, an **ID token and access token** are issued.
- These tokens are sent back to the application, where the backend validates their authenticity and extracts user identity claims.
- A new, internal **JWT** is then issued by the backend to manage access to protected resources within the app.

B. Authentication based on username and passwords with Email based MFA

Alternatively, users can authenticate using a traditional login form:

- The user submits their email and password to the backend.
- After initial credential verification, a **one-time MFA code** is sent to the user's registered email address.
- The user must confirm this MFA code to complete the authentication process.

Once verified, the backend issues a **JWT** for authorized access.

7.2 Authentication and Authorization Architecture (ASP.NET Core + JWT)

7.2.1 Overview

The application implements token-based authentication and authorization using ASP.NET Core's built-in middleware pipeline and JSON Web Tokens (JWT). This design ensures a secure, stateless approach to managing user identity, suitable for modern APIs and distributed systems.

Users can authenticate using either Microsoft Login (via OAuth2/OpenID Connect) or traditional email/password login with multi-factor authentication (MFA) via email. After successful authentication, a JWT is issued to the client, which is then used to access protected endpoints across the system.

7.2.2 Authentication Workflow

Once the user successfully logs in—whether through Microsoft SSO or local credentials—the backend generates a **JWT** that encapsulates user identity and associated claims (e.g., user ID, role, permissions). This token is signed with a symmetric key HMAC256 and returned to the client application. There is a dedicated encryption key for each tenant, keys stored securely in a Azure Key Vault, deployed in the Azure tenant Expertware.onmicrosoft.com. Access to the Azure Key Vault is restricted to Expertware IP range and a dedicated, non-interactive service account from the INT LDAP.

The process relies on a 2nd refresh token which is used to request a new access token upon expiration. The acces JWT has a time to life of 15 minutes and the refresh token has a TTL of 24h.

The client includes this token in the Authorization header (Bearer <token>) of all subsequent API requests. The server uses this token to verify the user's identity and determine access rights.

7.2.3 Token Lifecycle and Validation

The **JWT token** includes:

- User identification (e.g., email or ID)
- Role or permission claims
- Token issue and expiration timestamps
- Signing information (for tamper-proof verification)

ASP.NET Core's **JWT authentication middleware** validates each incoming token by:

- Verifying the signature using the configured security key
- Checking the issuer, audience, and expiration
- Extracting claims to establish the user's identity for the request context

This process is **stateless**—no session data is stored on the server. The token is self-contained and valid if it meets the defined criteria.

7.2.4 Authorization Strategy

Authorization in the application is managed using **policy-based authorization**, a flexible mechanism provided by ASP.NET Core for defining and enforcing access rules.

Although **roles are included as claims within the JWT**, they serve primarily as **labels** that describe the user's identity. These roles, along with other claims, are evaluated through **authorization policies**, which determine access rights at the API endpoint level.

Policies are defined centrally and can include a combination of:

- Role-based conditions
- Custom claim requirements
- Complex business rules implemented via custom policy handlers

Each protected endpoint specifies which policy must be satisfied using the [Authorize(Policy = "PolicyName")] attribute. This ensures that access decisions are based on centralized, maintainable logic rather than hard-coded role checks.

This approach supports:

- Fine-grained control over resource access
- Centralized policy definition and reuse
- Scalability and alignment with enterprise security requirements

In summary, **authorization is enforced through policies**, while **roles and other claims provide the input** evaluated by those policies.

7.2.5 Security Considerations

To ensure security across the authentication and authorization layers, the following practices are in place:

- **Short-lived access tokens**: JWTs have limited lifespans (15 minutes) to minimize risk if compromised.
- **Secure storage**: Secrets used to sign tokens are stored securely (Azure Key Vault).
- **MFA for local login**: Users who authenticate using email and password must complete an MFA step via a one-time code sent to their email.
- **HTTPS enforced**: All token exchanges and authentication flows are conducted over HTTPS.
- **Token validation on every request**: Each request is independently validated by the backend using token validation parameters (issuer, audience, key, and expiration).

This architecture supports a secure, reliable, and extensible foundation for managing identity and access within the application.

8 GLOSSARY OF ACRONYMS AND ABBREVIATIONS

AD	Active Directory
API	Application Programming Interface
AV	Antivirus
CA	Certificate Authority
CTI	Cyber Threat Intelligence
DNS	Domain Name System
EDR	Enhanced Data Rate
FAQ	Frequently Asked Questions
GDPR	General Data Protection Regulation
GVM	Greenbone Vulnerability Management
HA	High Availability
HDD	Hard Disk Drive
HTTPS	Secure Hypertext Transfer Protocol
IoA	Indicator of Attack
IoC	Indicator of Compromise
KPI	Key Performance Indicator
KRI	Key Risk Indicator
KVM	Kernel-based Virtual Machine
MISP	Malware Information Sharing Platform
NTP	Network Time Protocol
NVMe	Non-Volatile Memory
OS	Operating System
PAM	Privileged Access Management
RAID	Redundant Array of Independent Disks
RAM	Random Access Memory
S2S	Site-to-Site Virtual private network
SAN	Storage Area Network
SIEM	Security Information and Event Management
SIRP	Security Incident Response Platform
SME	Subject Matter Expert
SNMP	Simple Network Management Protocol
SOAR	Security orchestration, automation and response
SOC	Security Operation Center
SOCaaS	Security Operations Center-as-a-Service
SPOF	Single Point Of Failure
SQL	Structured Query Language
SSD	Solid State Drive
SSL	Secure Sockets Layer
STIX	Structured Threat Information Expression
TAXII	Trusted Automated Exchange of Intelligence Information
TLS	Transport Layer Security
UC	Use Case
VDI	Virtual Desktop Infrastructure

VM	Vulnerability Management
VMDR	Vulnerability Management Detection & Response
VPN	Virtual Private Network
XSS	Cross-Site Scripting