PACOM
VIGIL CORE™

Cyber Security Solution

PACOM VIGIL CORE provides the ultimate high level security solution to allow installation in extremely secure settings. Strong encryption and authentication procedures are used by connections made by users and devices to secure the network and manage resource access.

- Onion Architecture Docker Container Network
- Micro-Service Implementation
- Domain Driven Design
- Comprehensive Software Testing
- High-Level Encryption
- Certificate-Based Authentication
- Secure Web API
- Access Token Validation
- Configurable Role-based User Access
- Modular Access Policies
- Granular Permission

VIGIL CORE Architecture

The VIGIL CORE Platform and its components are designed to work in a cloud, on-premises, or hybrid environment. VIGIL CORE has been developed on a micro-services architecture to offer high scalability, flexible deployments, resilience, and maintainability.

Onion Architecture

The infrastructure components of the VIGIL CORE Platform are created with appropriate separation between the areas of concern within the application. This makes it possible to develop, test, and maintain.

Advantages:

- Provides better maintainability, therefore, fewer service interruptions and downtime of services.
- Provides better testability for a reliable software platform.
- Any concrete implantation would be provided to the application at runtime
- Domain entities are core and centre part. It can access both the database and UI layers.
- The internal layers never depend on an external layer. The code that may have changed should be part of an external layer.

Domain Driven Design

The Domain Model will be the core of each VIGIL CORE Platform component, containing the business logic and rules. Business logic is separated from the view and data access layers.

Advantages:

- Faster development for updates
- Scalability and improved stability
- Data security

Micro-Services

The functionality of the VIGIL CORE platform is broken down into components, a suite of small, narrowly focused, independently deployable services. Each microservice runs in its own process and communicates with HTTPS endpoint via the Web API. Those services are encapsulated for specific capabilities and are deployed independently using a fully automated mechanism.

Docker Container Network

The VIGIL CORE Platform micro-services are each hosted within a Docker container. Docker container networking provides an extra layer of security so that communications between different components on the system occur on an isolated internal virtual network that is only accessible from within a Docker container, and shall just expose an HTTPS endpoint via the Web API gateway service.

Comprehensive Software Testing

The VIGIL CORE Platform undergoes rigorous testing to ensure quality, functionality, and security.

Unit Testing Framework

- XUnit
- Moq
- FluentAssertions

Integration Testing Framework

- XUnit

Automation Testing Framework

- Selenium

VIGIL CORE Authentication

The VIGIL CORE Platform will authenticate all users, devices, and Integration or other services to allow access.

High-Level Encryption

VIGIL CORE Platform communicates with each component using wolfSSL.

The wolfSSL library is a lightweight SSL/TLS library targeted for embedded, RTOS, and resource-constrained environments - primarily because of its small size, speed, and feature set. It is used in many common platforms because the wolfSSL library supports over 30 different operating environments, industry standards up to the current TLS 1.3 library and offers progressive cyphers such as ChaCha20, Curve25519, NTRU, and Blake2b. User benchmarking and feedback report dramatically better performance when using wolfSSL versus other similar implementations of TLS.

Certificate-Based Authentication

Devices gateway interfaces and other Core VIGIL CORE Platform services are authenticated using shared certificates.

All communications to the S1000 uses HTTP over TLS and is authenticated in both directions using certificates. Certificates are validated or revoked by the VIGIL CORE Platform Integrations Layer. All communications will occur with the S1000 device acting as the TCP/HTTP client on port 443 so that no site-side firewalls need to be configured.

Secure Web API

Authentication from third-party systems is allowed using the same mechanism as system services.

The Integration Layer of the VIGIL CORE platform allows different devices from a variety of vendors to interact with the system in a well-defined way by providing access to the core services through a standard Web API security check.

Devices will have no direct access to the core databases.

Access Token Validation

Users are authenticated by a unique username and password. The VIGIL CORE Platform features a built-in password strength monitor to guide you in creating secure passwords.

Clients receives tokens (Java Web Token), and every action the user performs on the web session shall pass this token.

VIGIL CORE Authorization

The VIGIL CORE Platform enables configurable and finely tuned access, control authorisation on features and information.

Modular Access Policies

Access policies can be configured based on specific organizational security requirement. The VIGIL CORE Platform can be finely tuned on which roles need to be defined, the areas to which each role is allowed access and the type of access allowed.

Configurable Role Based Access

Users are given a set of permissions from the set of configurable roles.

Roles are only allowed to access the information necessary to perform specific tasks effectively. Access can be based on several factors, such as authority, responsibility, and job competency. In addition, access to the VIGIL CORE Platform can be limited to specific tasks such as the ability to view, create or modify a device.

Granular Permissions

The VIGIL CORE Platform is organised according to a tree-based hierarchy for managing user permissions within the system.

Organisations provide a means to partition the system so that the users can only access modules, sites and devices within their scope.

Active Directory

The VIGIL CORE Platform integrates with Active Directory for authentication and authorisation. VIGIL CORE communicates using Active Directory Federation Services (AD FS) and OpenID Connect/OAuth.

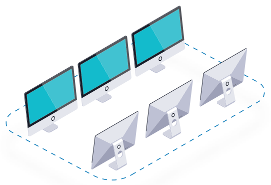
Network and Infrastructure Requirements

The VIGIL CORE Platform is designed to be set up on a cluster of servers or in the cloud. It is possible to run on a single server, but it will not have any redundancy in case of server failures. The initial release of the VIGIL CORE Platform will support Windows Server 2016, and SQL Server 2016. Because the solution is virtualized, it should work on other systems including Linux and Apple MacOS but is currently untested.

Prerequisites

SQL	Installed independently of the VIGIL CORE Platform and accessed according to client IT department policies. VIGIL CORE needs the SQL connection string to be able to connect to the database (as well as appropriate permissions, certificates)
Virtualization	Enabled in client-server BIOS settings. This is generally enabled by default.
Docker	Installed as the first step in the installation process if not already installed.
A private/public certificate pair (Optional)	Required to encrypt the connection between the VIGIL CORE Platform gateway and web-browsers. It is recommended that you purchase a certificate from a standard certificate authority, but if you choose not to; the install can auto-generate one for you. If you use an auto-generated certificate, you will need to install the public certificate on all client machines manually.

Firewall settings needed:



Intranet only

Only required for messaging between the VIGIL CORE Platform servers.

Docker Swarm

TCP (2376, 2377)
UDP (4789)
TCP and UDP (7946)

ElasticSearch Management (optional)

TCP (8881, 8882)

External Ports:

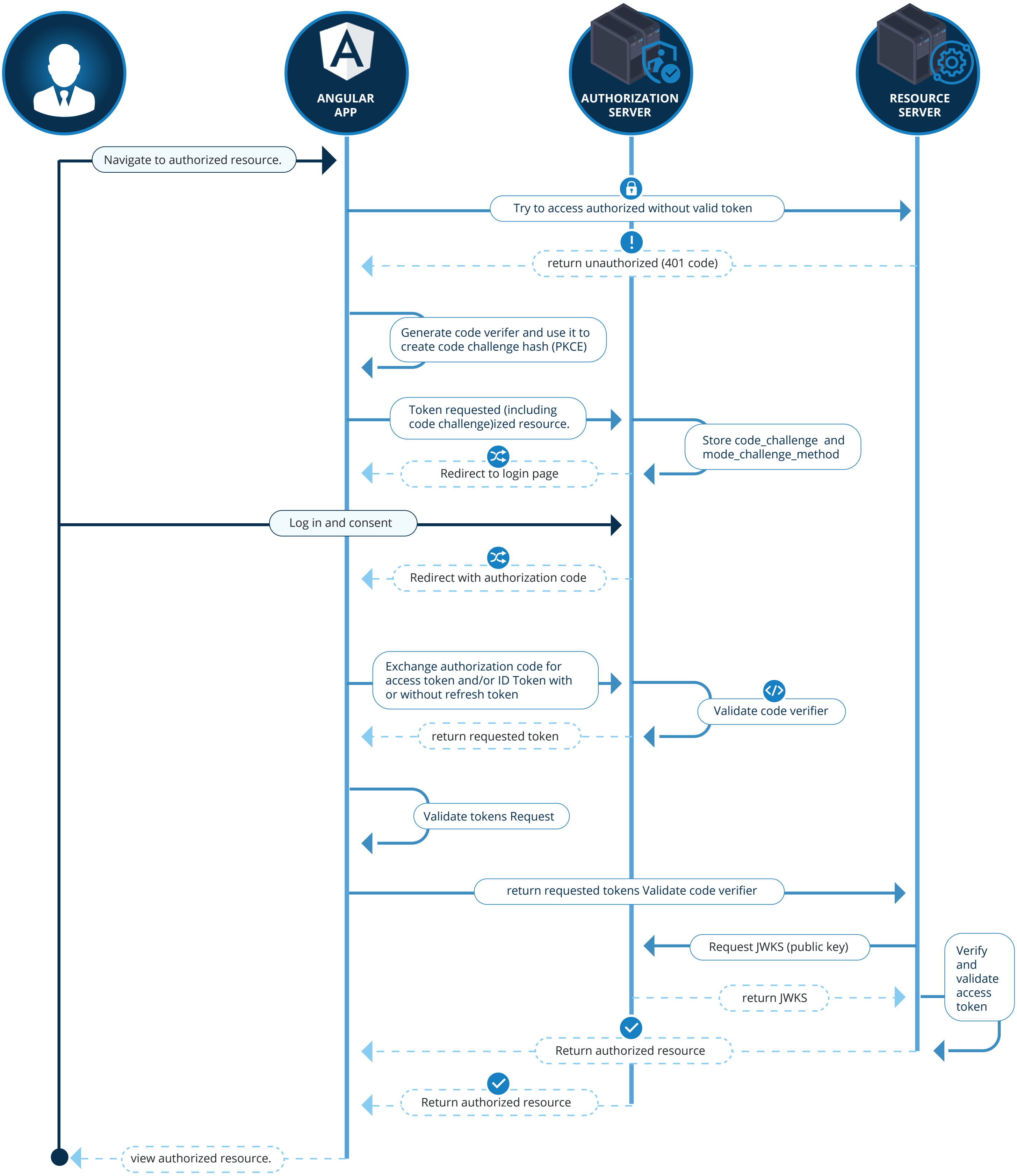
VIGIL CORE Platform web traffic (HTTP/HTTPS)

TCP (80, 443)

S1000 devices:

TCP (7016)

Client Authorisation Code Workflow



ANGULAR APP

AUTHORIZATION SERVER

RESOURCE SERVER

Navigate to authorized resource.

Try to access authorized without valid token

return unauthorized (401 code)

Generate code verifer and use it to create code challenge hash (PKCE)

Token requested (including code challenge)ized resource.

Store code_challenge and mode_challenge_method

Redirect to login page

Log in and consent

Redirect with authorization code

Exchange authorization code for access token and/or ID Token with or without refresh token

Validate code verifier

return requested token

Validate tokens Request

return requested tokens Validate code verifier

Request JWKS (public key)

Verify and validate access token

return JWKS

Return authorized resource

Return authorized resource

view authorized resource.