Laurea Triennale in informatica - Università di Salerno
Corso di *Ingegneria del Software* - Prof.ssa F. Ferrucci

GUARDIAN FLOW

FEELING SAFE

System Design Document

Guardian Flow

Riferimento	
Versione	1.6
Data	05/12/2023
Destinatario	Prof.ssa F. Ferrucci
Presentato da	Intero team
Approvato da	Raffaele Mezza, Martina Mingione

Sommario

Revision History

Data	Versione	Descrizione	Autori
22/11/2023	0.1	Scopo del sistema	Vincenzo Maiellaro
22/11/2023	0.2	Architettura del sistema corrente	Tommaso Nardi
24/11/2023	0.3	Obiettivi di design	Tutto il team
28/11/2023	0.4	Panoramica	Vincenzo Maiellaro
28/11/2023	0.5	Decomposizione in sottosistemi	Danilo Gisolfi
29/11/2023	0.6	Mapping HW e SW	Mattia Guariglia
29/11/2023	0.7	Gestione dei dati persistenti	Edmondo De Simone
29/11/2023	0.8	Diagramma E/R	Tommaso Nardi
30/11/2023	0.9	Controllo degli accessi e sicurezza	Giuseppe Cerella
01/12/2023	1.0	Condizioni limite	Vincenzo Maiellaro
01/12/2023	1.1	Controllo globale del software	Tommaso Nardi
04/12/2023	1.2	Introduzione del documento	Edmondo De Simone Giuseppe Cerella
04/12/2023	1.3	Servizi dei sottosistemi	Vincenzo Maiellaro Tommaso Nardi
05/12/2023	1.4	Glossario	Edmondo De Simone Giuseppe Cerella
05/12/2023	1.5	Revisione del documento	Giuseppe Cerella
05/12/2023	1.6	Approvazione del documento	Martina Mingione Raffaele Mezza

1. Introduzione

1.1 Scopo del sistema

Il software Guardian Flow nasce dalla necessità di implementare una solida infrastruttura digitale per il controllo del traffico dati all'interno di un'azienda, cercando di semplificare e modernizzare il processo esistente. L'analisi del sistema attuale ha rivelato importanti problematiche legate alla rilevazione di anomalie inefficiente del traffico di rete, richiedendo un approccio più avanzato e scalabile. L'obiettivo principale del sistema Guardian Flow è la rilevazione di anomalie attraverso una piattaforma intuitiva e dinamica, costruendo una baseline personalizzata per il cliente e mettendo a disposizione una dashboard, la quale permette al cliente di visualizzare in tempo reale il traffico di rete analizzato e le eventuali anomalie rilevate. La flessibilità del sistema si estende anche alla possibilità di modificare agilmente il piano d'abbonamento, offrendo agli utenti la libertà di adattare le funzionalità e le risorse disponibili alle specifiche esigenze dell'azienda. Con Guardian Flow, si propone di rendere l'analisi del traffico di rete accessibile, efficiente per garantire una gestione ottimale delle risorse e la sicurezza del sistema.

1.2 Obiettivi di Design (Design Goals)

1.2.1 Design Goals

Priorità	ID Design Goal	Descrizione Design Goal	Categoria	Origine	Trade off
Alta	DG_1.1 Scalabilità	Le prestazioni garantite devono essere proporzionali al piano d'abbonamento acquistato, ottimizzando le risorse in base alle esigenze specifiche di ciascun cliente.	Performance	NF_PR_01	Scalabilità e Notifica Anomalie
Alta	DG_2.1 Notifica Anomalie	Il sistema deve notificare qualsiasi anomalia entro un secondo dalla sua rilevazione.	Dependability	NF_PR_02	Scalabilità e Notifica Anomalie
Alta	DG_2.2 Legali	L'analisi del traffico deve essere implementata per rispettare le norme sulla privacy, conformemente al Regolamento Generale sulla Protezione dei Dati (GDPR).	Dependability	NF_LE_01	
Alta	DG_2.3 Accessibilità	Il sistema deve essere costantemente accessibile e operativo, assicurando una presenza continua e affidabile per gli utenti.	Dependability	NF_AF_01	
Media	DG_2.4 Sistema Responsive	Il sistema deve avere una reattività veloce e rispondere tempestivamente a tutte le interazioni.	Dependability	NF_AF_02	
Alta	DG_2.5 Tolleranza ai guasti	Il sistema deve implementare dei backup per garantire sicurezza e disponibilità dei dati.	Dependability	N/A	
Media	DG_3.1 Ambiente	I server del sistema devono essere alimentati con energia rinnovabile, riducendo l'impatto ambientale.	Cost	NF_SO_01	

Alta	DG_3.2 Costi di manutenzione	Il costo previsto per la manutenzione è stimato intorno a 15.000 euro.	Cost	N/A	Supporto utenti vs Costi di manutenzione
Alta	DG_3.3 Costi di sviluppo	Il costo totale in termini di ore è previsto intorno alle 350 ore per la progettazione e lo sviluppo del sistema, distribuite in 50 ore per ciascun membro del team.	Cost	N/A	
Media	DG_4.1 Codice	Il front-end deve essere sviluppato usando il framework Nuxt mentre il back-end usando Python.	Maintenance	NF_IP_01 NF_IP_02	Portabilità vs Codice
Alta	DG_4.2 Portabilità	La dashboard deve essere progettata per essere accessibile senza la necessità di installazioni.	Maintenance	NF_PA_01	Portabilità vs Codice
Media	DG_5.1 Navigazione Intuitiva	La dashboard deve essere progettata con l'implementazione di tasti auto-esplicativi, al fine di garantire una navigazione agevole, intuitiva e priva di ambiguità per gli utenti.	End user	NF_US_01	
Bassa	DG_5.2 Supporto Utenti	Gli utenti devono poter ricevere assistenza tramite un servizio di chatbot.	End user	NF_OP1	Supporto utenti vs Costi di manutenzione

1.2.2 Design Trade Off

Trade-off	Descrizione
Scalabilità e Notifica Anomalie	Offrire un'analisi altamente veloce e precisa del traffico di rete, garantendo un rilevamento affidabile delle anomalie potrebbe richiedere un utilizzo più intenso delle risorse di elaborazione. L'algoritmo di rilevamento delle anomalie sarà accurato e sofisticato, riducendo al minimo i falsi positivi e identificando in modo affidabile i comportamenti anomali. Per ottenere una precisione elevata un'analisi più dettagliata potrebbe richiedere più tempo di elaborazione e una quantità maggiore di risorse hardware, influenzando leggermente la notificazione dell'anomalia.
Supporto utenti vs costi di manutenzione	Mettere a disposizione un servizio di chatbot automatico per il supporto utenti potrebbe non aumentare i costi operativi e di manutenzione. Un servizio di chatbot automatico non richiede risorse aggiuntive.
Portabilità vs codice	Garantire che la dashboard del sistema sia portabile, accessibile senza la necessità di installazioni. L'ottimizzazione per la portabilità potrebbe richiedere l'implementazione di soluzioni più complesse e meno intuitive, incidendo sulla chiarezza del codice sorgente.

1.4 Definizioni, acronimi e abbreviazioni

1.4.1 Definizioni

- Apprendimento supervisionato: l'apprendimento supervisionato è una categoria di tecniche nell'ambito dell'intelligenza artificiale e del machine learning in cui si addestra un modello utilizzando un insieme di dati etichettati;
- Boolean: viene utilizzato nella logica per rappresentare un tipo di dato che può avere solo due valori: vero o falso;
- Date: tipologia di dato per memorizzare informazioni sulla data;
- Design Goals: obiettivi di design progettati per il sistema proposto;
- Integer: tipologia di dato per memorizzare valori numerici interi senza parte decimale;
- Varchar: tipologia di dato per memorizzare stringhe di lunghezza variabile.

1.4.2 Acronimi e Abbreviazioni

- DG: Design Goals;
- DAO: Data Access Object;
- FR: Functional Requirements;
- GDPR: General Data Protection Regulation;
- HTTP: Hypertext Transfer Protocol;
- IDS: Intrusion Detection System;
- IP: Internet Protocol;
- IPS: Intrusion Prevention System;
- RNF: Non Functional Requirements;
- SDD: System Design Document;
- SFTP: Secure File Transfer Protocol;
- TCP: Transmission Control Protocol;
- UC: Use Case.

1.5 Riferimenti

RAD Requirement Analysis Document Guardian Flow V_1.6.

1.6 Organizzazione del documento

- Paragrafo 1: introduce l'obiettivo del sistema, i design goals, un elenco di definizioni, acronimi ed abbreviazioni essenziali per la comprensione dell'intera documentazione;
- Paragrafo 2: esamina le funzionalità e gli aspetti architetturali dei sistemi correlati, che hanno influenzato la definizione dell'architettura del sistema proposto;
- Paragrafo 3: presenta l'architettura del sistema, gestendo la decomposizione in sottosistemi, il mapping hardware/software, i dati persistenti, il controllo degli accessi e la sicurezza, il controllo del flusso globale del sistema e le condizioni limite;
- Paragrafo 4: descrive i servizi offerti dai sottosistemi del sistema, fornendo una visione completa delle funzionalità specifiche implementate per ciascun componente.

2. Architettura del sistema corrente

2.1 Architettura dei sistemi simili

Risorsa: https://www.broadcom.com/products/advanced-threat-protection/network-forensics-security-analytics

Symantec Network Forensics & Security Analytics utilizza un'architettura di cattura e analisi dei pacchetti di rete. Si basa su sensori di rete per raccogliere dati da diverse parti della rete. Questi dati vengono poi inviati al server di Symantec, dove vengono elaborati e archiviati per l'analisi.
Gli amministratori accedono alla console di Symantec per visualizzare i dati, configurare allarmi, eseguire report e gestire la sicurezza della rete.
Risorsa: https://www.solarwinds.com/

SolarWinds NPM utilizza un'architettura centralizzata e distribuita. Si basa su sonde remote per raccogliere dati da posizioni diverse all'interno della rete. Questi dati vengono inviati ai server di SolarWinds, dove vengono elaborati e archiviati per l'analisi. Gli amministratori accedono alla console di SolarWinds per visualizzare i dati, configurare allarmi, eseguire report e gestire la rete.

Risorsa: https://www.paessler.com/

PRTG utilizza sensori che sono distribuiti su dispositivi di rete, server e altre risorse. Ogni sensore raccoglie dati specifici. I dati raccolti dai sensori vengono inviati e elaborati dal server centrale PRTG. Gli utenti accedono all'interfaccia web di PRTG per visualizzare le statistiche, configurare notifiche e analizzare i dati raccolti dai sensori.

Risorsa: https://www.cisco.com/c/it_it/products/collateral/security/stealthwatch/datasheet-it.html

Cisco Stealthwatch utilizza sensori hardware e software distribuiti in vari punti della rete per raccogliere dati sul traffico di rete. I dati raccolti dai sensori vengono inviati a un server centrale per l'analisi e la generazione di report. Gli amministratori accedono all'interfaccia di Cisco Stealthwatch per monitorare il traffico di rete, rilevare anomalie e gestire la sicurezza della rete.

3. Sistema proposto

3.1 Panoramica

Come team di progettazione, in linea con i requisiti e i design goals, si presenta il sistema Guardian Flow, progettato per la rilevazione di anomalie del traffico di rete attraverso una piattaforma intuitiva e dinamica. Il sistema si propone di costruire una baseline personalizzata per ogni cliente, offrendo loro una dashboard in tempo reale per visualizzare il traffico di rete analizzato e le eventuali anomalie rilevate.

Gli utenti possono personalizzare il servizio in base alle dimensioni della propria azienda e alle loro necessità di potenza computazionale, acquistando l'abbonamento più adatto alle loro esigenze. L'obiettivo principale del sistema è fare uso dell'intelligenza artificiale non supervisionata per individuare comportamenti anomali nel traffico di rete che potrebbero sfuggire ai tradizionali sistemi di rilevamento delle intrusioni (IDS), di prevenzione delle intrusioni (IPS) e firewall con apprendimento supervisionato e firme. Inoltre, il sistema notificherà gli utenti per ogni anomalia rilevata.

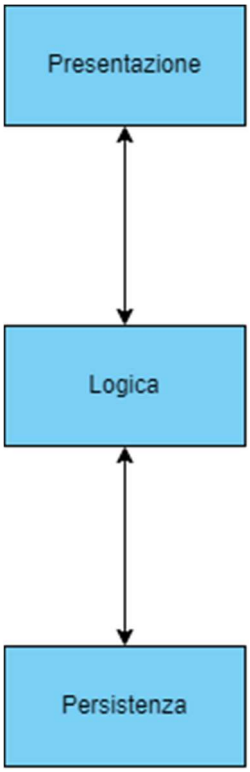
Il pattern architetturale scelto è il modello Three-Tier che prevede la divisione dell'applicazione in tre strati: interfaccia utente, logica funzionale e logica di persistenza, dove le diverse funzioni sono organizzate e distribuite su questi livelli che comunicano tra di loro.

3.2 Decomposizione in sottosistemi

3.2.1 Decomposizione in Layer

La decomposizione usata per il sistema è formata da tre layer:

- Presentazione: contiene tutti gli elementi che consentono l'interazione dell'utente con il sistema;
- Logica: risiedono gli oggetti relativo al controllo e alle entità di elaborazione, verifica e di notifica delle interazioni del sistema;
- Persistenza: si occupa della gestione dei dati persistenti. È responsabile dell'accesso, dell'archiviazione e della manipolazione dei dati.



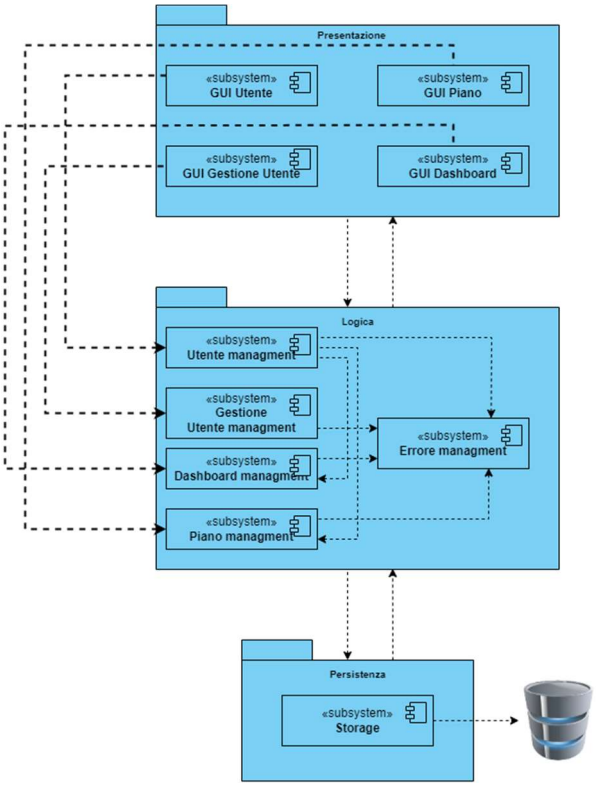
3.2.2 Decomposizione in sottosistemi

Dopo un'accurata analisi funzionale e considerando i Goals & Trade-offs, abbiamo deciso di suddividere le funzionalità per area di gestione; quindi, si è suddiviso il sistema principale in vari sottosistemi che comunicano tra di loro.

In particolare, abbiamo creato l'area funzionale "Utente Management" che comprende le Utenze di Utente Admin e Utente Subordinato.

Si è preso in considerazione un sottosistema che si occupa della gestione degli errori chiamato "Errore Management", abbiamo anche creato, per la gestione e l'organizzazione del piano e degli utenti, due aree funzionali chiamate "Gestione Utente Management" e "Piano Management". Infine, per la gestione e visualizzazione delle anomalie rilevate, abbiamo creato l'area funzionale "Dashboard Management".

Il sistema si compone di dieci sottosistemi:



Il livello di Presentazione prevede quattro sottosistemi GUI per interfacciarsi con la piattaforma e possiamo identificarli come oggetti "boundary" individuati nel RAD:

- GUI Utente: sottosistema che fornisce l'interfaccia utente per interagire con i servizi offerti dalla piattaforma relativi alla gestione dell'utenza;
- GUI Dashboard: sottosistema che fornisce l'interfaccia grafica per i servizi relativi alla dashboard;
- GUI Gestione Utenti: sottosistema che fornisce l'interfaccia grafica agli amministratori per i servizi relativi agli account subordinati;
- GUI Piano: sottosistema che fornisce l'interfaccia grafica agli amministratori per i servizi relativi ai piani d'abbonamento.

Il livello di Logica Applicativa prevede la gestione di quattro sottosistemi:

- Utente Management: sottosistema che permette all'utente di usufruire di tutte le funzionalità offerte dal sistema;
- Dashboard Management: sottosistema che offre funzionalità di gestione e visualizzazione delle anomalie rilevate;
- Gestione Utenti Management: sottosistema che offre funzionalità di gestione, aggiunta, rimozione e modifica degli account subordinati ad un utente admin;
- Piano Management: sottosistema che offre funzionalità di acquisto, gestione, modifica e annullamento di un piano d'abbonamento.

Il livello di Persistenza prevede la gestione di un sottosistema:

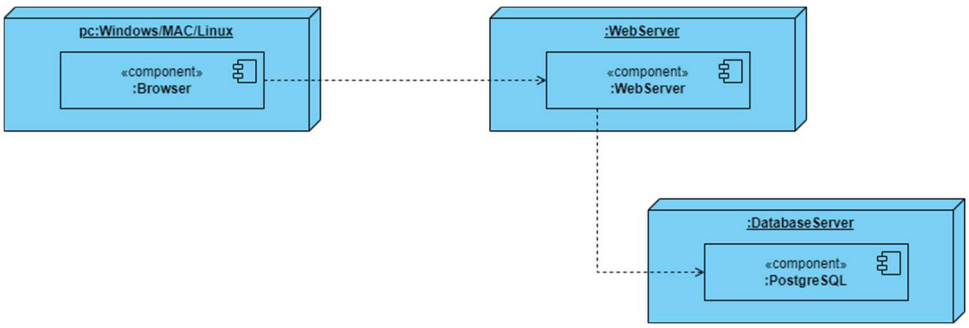
- Storage: sottosistema che si occupa di immagazzinare e prelevare i dati persistenti dal nostro database, in particolare lo stesso sarà diviso in due macro-componenti logiche:
    - Macro-componente DAOs: sottocomponente del sottosistema Storage che si occuperà di offrire i servizi di manipolazione dei dati persistenti;
    - Macro-componente Entity: sottocomponente del sottosistema Storage che racchiuderà tutte le entità rappresentanti informazioni da immagazzinare nel database e direttamente manipolabili dal livello di Logica Applicativa.

3.2.3 Deployment Diagram

Come team di progetto abbiamo deciso di formalizzare il Deployment Diagram con componenti di alto livello per rappresentare l'architettura del sistema a run-time.

L'utente, utilizzando il suo browser, si connette al server web attraverso un link ipertestuale. All'interno del server web, ci sono componenti relative al livello di presentazione e al livello di logica applicativa per soddisfare tutte le richieste dell'utente.

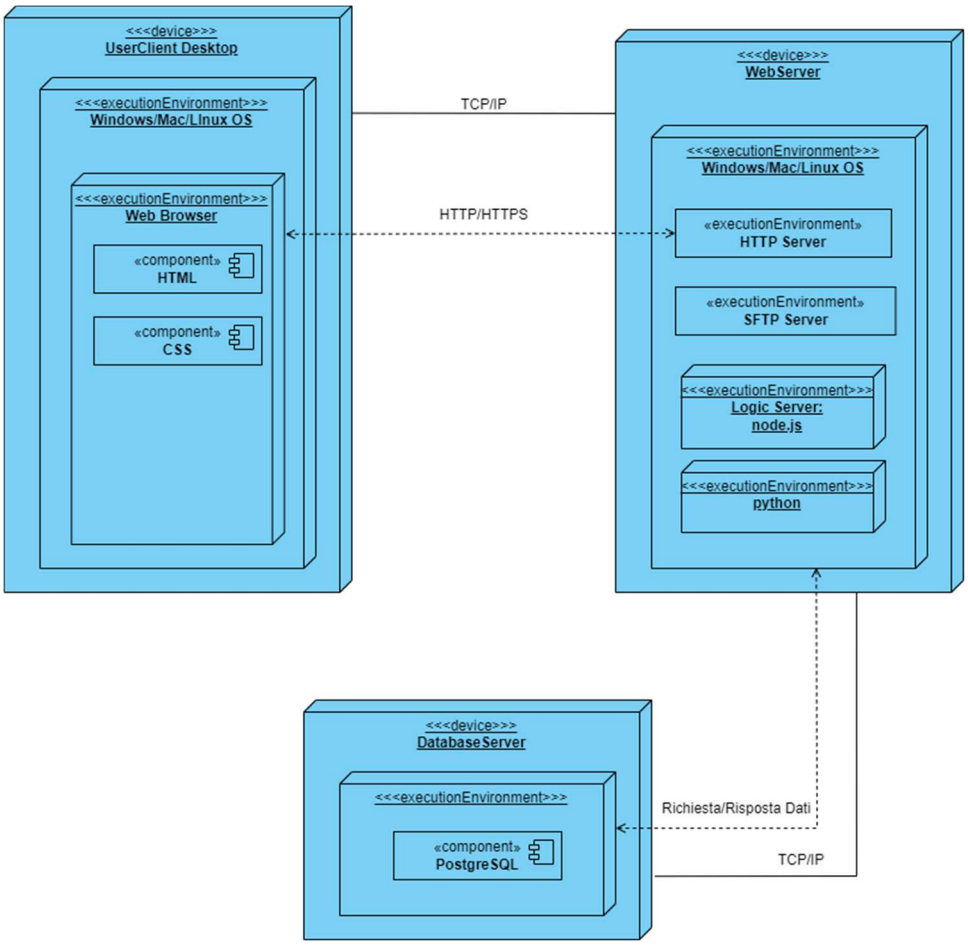
Il DatabaseServer contiene i dati persistenti che verranno forniti al WebServer quando questi vengono richiesti. Non appena le informazioni arrivano al WebServer, quest'ultime vengono inoltrate al client.



3.3 Mapping HW/SW

L'utente può interfacciarsi con il nostro sistema attraverso il proprio browser. I dispositivi utilizzati dovranno avere a disposizione una connessione ad Internet per permettere la comunicazione con l'HTTP Server. Tra l'utente e l'HTTP Server vi sarà una comunicazione basata sul protocollo HTTP in cui il client invierà le richieste e il server provvederà alle risposte. Le richieste e le risposte saranno gestite tramite il protocollo TCP/IP che prevede il three-way handshake.

Il WebServer fa da interfaccia tra l'utente e il Database Server. Il Database Server ospita un database relazionale che verrà interrogato tramite PostgreSQL.


<<<device>>>
UserClient Desktop
<<<executionEnvironment>>>
Windows/Mac/LInux OS
<<<executionEnvironment>>>
Web Browser
«component»
HTML
«component»
CSS
TCP/IP
HTTP/HTTPS
<<<device>>>
WebServer
<<<executionEnvironment>>>
Windows/Mac/Linux OS
«executionEnvironment»
HTTP Server
«executionEnvironment»
SFTP Server
<<<executionEnvironment>>>
Logic Server:
node.js
<<<executionEnvironment>>>
python
<<<device>>>
DatabaseServer
<<<executionEnvironment>>>
«component»
PostgreSQL
Richiesta/Risposta Dati
TCP/IP

3.4 Gestione dei dati persistenti

azienda

Nome	Tipo	Null	Key
id	VARCHAR(45)	Not null	Primary key
nome	VARCHAR(45)	Not null	
email	VARCHAR(45)	Not null	
telefono	VARCHAR(45)	Not null	
p_iva	VARCHAR(45)	Not null	
id_piano	VARCHAR(45)	Not null	Foreign key

user

Nome	Tipo	Null	Key
id	VARCHAR(45)	Not null	Primary key
username	VARCHAR(45)	Not null	
nome	VARCHAR(45)	Not null	
cognome	VARCHAR(45)	Not null	
two_factor	INT		
two_factor_token	VARCHAR(128)		
permessi	VARCHAR(45)	Not null	
azienda_id	VARCHAR(45)	Not null	Foreign key

user_key

Nome	Tipo	Null	Key
id	VARCHAR(255)	Not null	Primary key
user_id	VARCHAR(15)	Not null	Foreign key
hashed_password	VARCHAR(255)		

user_session

Nome	Tipo	Null	Key
id	VARCHAR(127)	Not null	Primary key
user_id	VARCHAR(15)	Not null	Foreign key
active_expires	BIGINT	Not null	
Idle_expires	BIGINT	Not nul	

anomalia

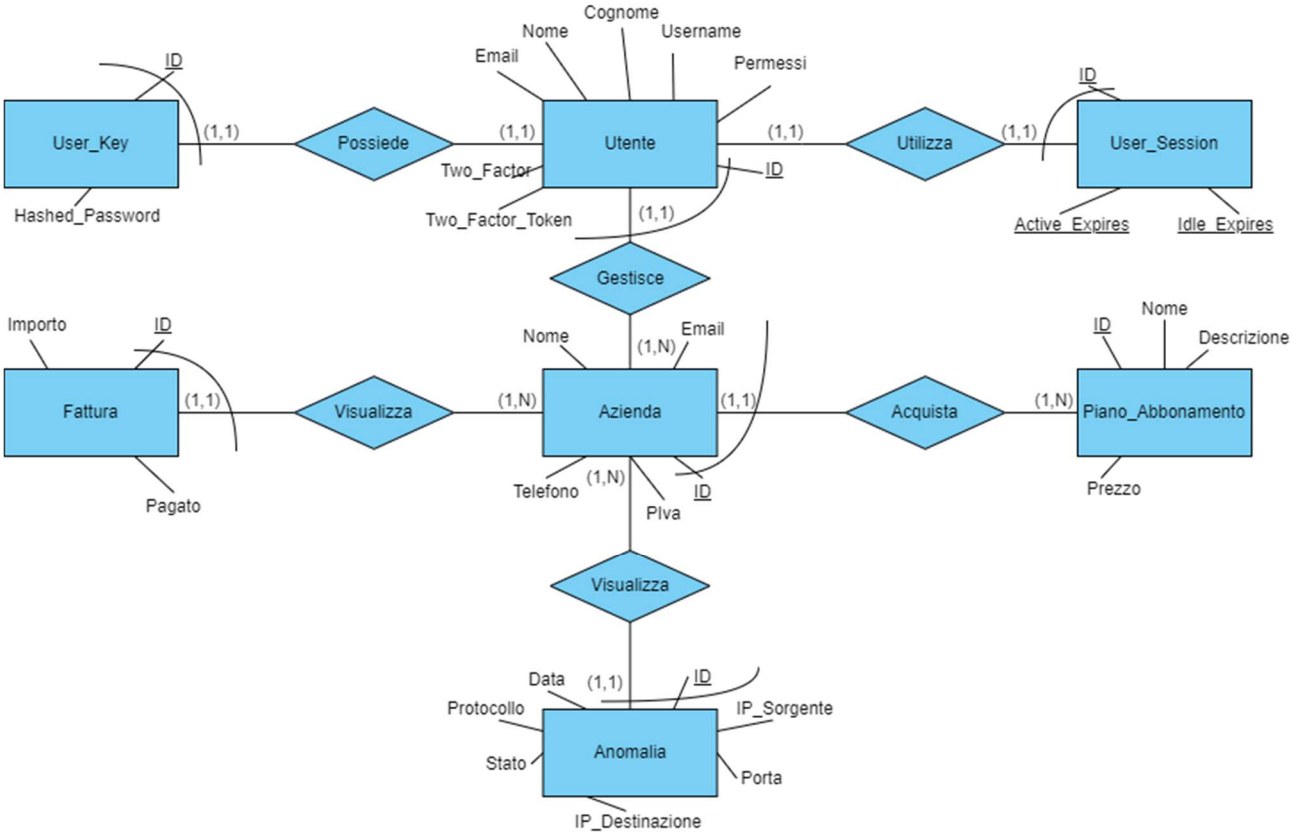
Nome	Tipo	Null	Key
id	VARCHAR(45)	Not null	Primary Key
protocollo	VARCHAR(45)	Not null	
stato	BOOLEAN		
data	DATE	Not null	
porta	INT	Not null	
ip_sorgente	VARCHAR(45)	Not null	
ip_destinazione	VARCHAR(45)	Not null	
id_azienda	VARCHAR(45)	Not null	Foreign key

piano_abbonamento

Nome	Tipo	Null	Key
id	VARCHAR(45)	Not null	Primary key
nome	VARCHAR(45)	Not null	
descrizione	VARCHAR(45)	Not null	
prezzo	DOUBLE	Not null	

fattura

Nome	Tipo	Null	Key
id	VARCHAR(45)	Not null	Primary key
pagato	BOOLEAN	Not null	
importo	DOUBLE	Not null	
id_azienda	VARCHAR(45)	Not null	Foreign key



3.5 Controllo degli accessi e sicurezza

Guardian Flow è un sistema multiutente, vi sono due tipologie di utente aventi permessi differenti.
Il controllo degli accessi è garantito tramite l'utilizzo di e-mail e password, inoltre, ogni utente avrà la possibilità di attivare l'autenticazione a due fattori e potrà cambiare la propria password all'occorrenza.
La sicurezza sui dati sensibili è garantita in quanto solo gli utenti amministratori potranno accedere ai dati relativi ai propri utenti subordinati.
La schematizzazione effettuata suddivide le azioni consentite per aree di gestione al fine di ottenere una visione più compatta e dettagliata grazie alle matrici degli accessi (più precisamente delle Access Control list) riportate di seguito:

Gestione accessi

Attore	Operazione
Utente subordinato / admin	Login( )
Utente subordinato / admin	Logout( )
Utente subordinato / admin	RecuperoPassword( )
Utente subordinato / admin	ModificaPassword( )
Utente subordinato / admin	Attiva2FA( )

Funzionalità Utente

Attore	Operazione
Utente admin	ModificaPiano( )
Utente admin	GestioneSubordinati( )
Utente admin	FalsoPositivo( )

Funzionalità Sistema

Attore	Operazione
Sistema	costruzioneBaseline( )
Sistema	AnalizzaTraffico( )
Sistema	notificaAnomalia( )

3.6 Controllo flusso globale del sistema

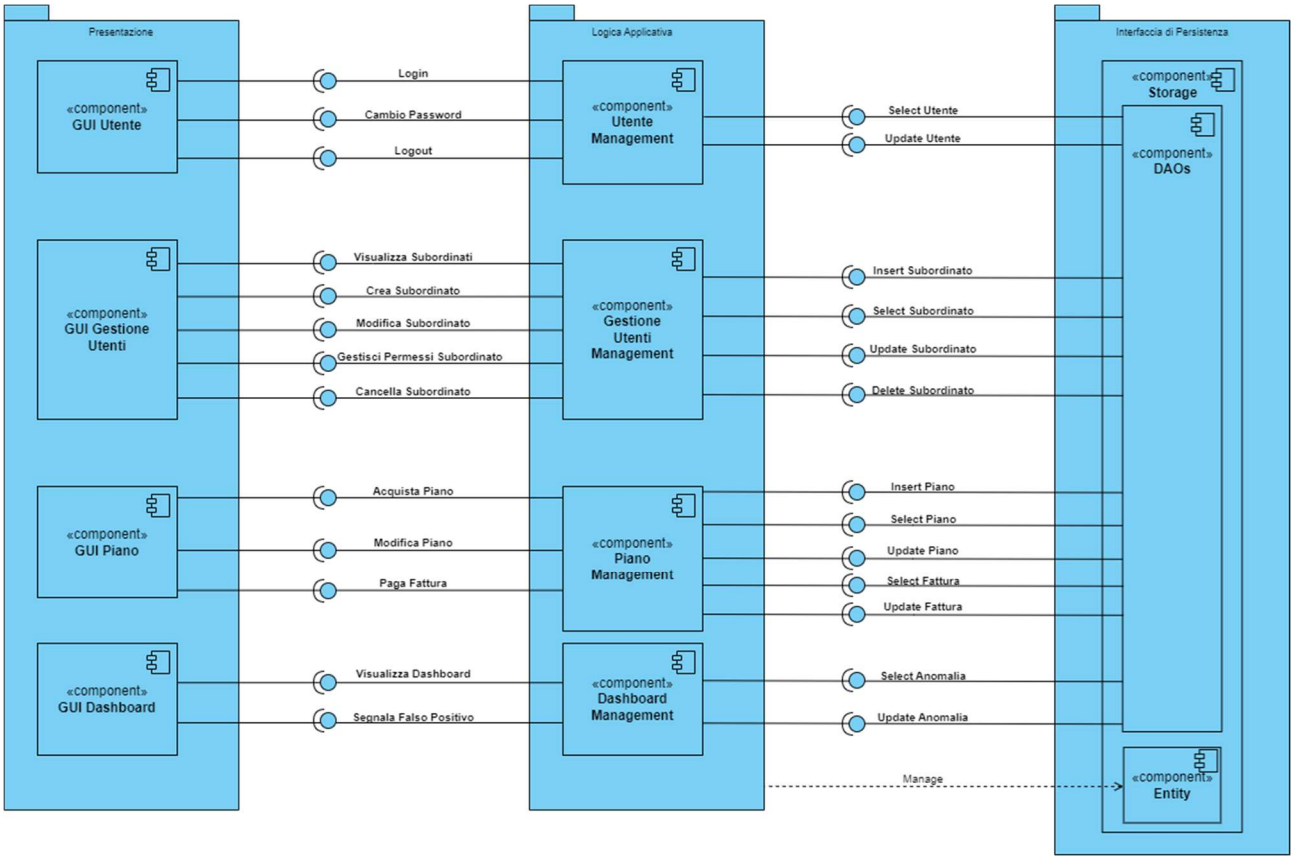
Il flusso del sistema è event-driven perché le sue azioni sono innescate da eventi specifici. Ad esempio, quando si rileva un nuovo tipo di minaccia il sistema reagisce a queste situazioni. Invece di seguire procedure fisse, risponde in modo flessibile agli eventi che si verificano nell'ambiente di sicurezza.

3.7 Definizione Use case per condizioni limite

Identificativo *UC_GS_01*	Picco di dati	*Data*	*01/12/2023*
		Vers.	*0.00.001*
		Autore	*Maiellaro Vincenzo*
Descrizione	Il sistema non è in grado di gestire un picco improvviso di dati che va oltre la sua capacità di rilevamento.		
Attore Principale	Sistema		
Attori secondari	Traffico di rete		
Entry Condition	Il sistema è attivo e monitora il traffico dati dell'azienda.		
Exit condition On success	Il sistema riesce a adattarsi al picco di traffico e continua a rilevare le anomalie senza diminuire la precisione.		
Exit condition On failure	Il sistema non riesce a gestire il picco di traffico, causando un malfunzionamento nel rilevamento delle anomalie causando una diminuzione della precisione.		
Rilevanza/User Priority	Priorità alta		
Frequenza stimata	N/A		
Extension point	N/A		
Generalization of	N/A		
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO			
1	Traffico di rete	Subisce un picco di dati improvviso.	
2	Sistema:	Tenta di elaborare l'improvviso picco di dati e rilevare le anomalie.	
3	Sistema:	Adatta dinamicamente le risorse e le capacità per gestire il picco di traffico, senza compromettere la precisione del rilevamento.	
I Scenario/Flusso di eventi Alternativo:			
2.1	Sistema	Non riesce ad elaborare il picco di traffico.	
2.2	Sistema:	Non riesce a rilevare in modo accurato le anomalie.	
Note	N/A		
Special Requirements	N/A		

Identificativo *UC_GS_02*	Rilevazione di un falso positivo	*Data*	*01/12/2023*
		Vers.	*0.00.001*
		Autore	*Maiellaro Vincenzo*
Descrizione	Il sistema rileva come anomalie un traffico di rete normale.		
Attore Principale	Sistema		
Attori secondari	Utente		
Entry Condition	Il sistema è attivo e monitora il traffico dati dell'azienda.		
Exit condition On success	Il sistema ottimizza i suoi modelli, riuscendo a ridurre la rilevazione dei falsi positivi.		
Exit condition On failure	Il sistema continua a generare falsi positivi.		
Rilevanza/User Priority	Priorità alta		
Frequenza stimata	N/A		
Extension point	N/A		
Generalization of	N/A		
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO			
1	Sistema:	Rileva traffico dati normale come anomalia.	
2	Sistema:	Genera una serie di falsi positivi, segnalando attività normali come comportamenti anomali.	
3	Utente:	Accedono alle segnalazioni e verificano che si trattano di attività normali.	
4	Sistema:	Ottimizza i suoi modelli per ridurre i falsi positivi.	
I Scenario/Flusso di eventi Alternativo:			
4.1	Sistema	Non ottimizza i suoi modelli, quindi non riduce i falsi positivi.	
Note	N/A		
Special Requirements	N/A		

4. Servizi dei sottosistemi



Presentazione

GUI Utente utilizza tre servizi offerti dalla Logica Applicativa del Sistema:

- Login;
- Logout;
- Cambio Password.

GUI Gestione Utenti utilizza cinque servizi offerti dalla Logica Applicativa del Sistema:

- Visualizza Subordinati;
- Crea Subordinato;
- Modifica Subordinato;
- Gestisci Permessi Subordinato;
- Cancella Subordinato.

GUI Anomalie utilizza un servizio offerto dalla Logica Applicativa del Sistema:

- Segnala Falso Positivo.

GUI Piano utilizza tre servizi offerti dalla Logica Applicativa del Sistema:

- Acquista Piano;
- Modifica Piano;
- Paga Fattura.

GUI Dashboard utilizza due servizi offerto dalla Logica Applicativa del Sistema:

- Visualizza Dashboard;
- Segnala Falso Positivo.

Logica Applicativa

Utente Management utilizza due servizi offerti dalla Logica di Persistenza:

- Select Utente;
- Update Utente.

Gestione Utenti Management utilizza quattro servizi offerti dalla Logica di Persistenza:

- Insert Subordinato;
- Select Subordinato;
- Update Subordinato;
- Delete Subordinato.

Piano Management utilizza cinque servizi offerti dalla Logica di Persistenza:

- Insert Piano;
- Select Piano;
- Update Piano;
- Select Fattura;
- Update Fattura.

Dashboard Management utilizza due servizi offerti dalla Logica di Persistenza:

- Select Anomalia;
- Update Anomalia.

5. Glossario

A

Architettura

Organizzazione fondamentale di un sistema, che definisce la sua struttura, i suoi componenti, le relazioni tra di essi e i principi guida per la progettazione e l'evoluzione del sistema nel tempo.

B

Backup

Il termine "backup" si riferisce alla creazione e alla conservazione di copie duplicate di dati o informazioni al fine di proteggerli da perdite accidentali o danni. Queste copie possono essere utilizzate per ripristinare i dati nel caso in cui vengano persi a causa di guasti hardware, errori umani, attacchi informatici, o altri eventi indesiderati.

Back-end

Il "back-end" è la parte di un'applicazione informatica o di un sistema software che gestisce le funzionalità non visibili direttamente agli utenti finali. Si occupa delle operazioni di elaborazione dei dati, della logica di business, dell'accesso al database e di altre attività legate al funzionamento interno di un'applicazione.

E

Event-driven

Un sistema event-driven risponde agli eventi generati da componenti software o hardware. Gli eventi possono includere input utente, cambiamenti di stato del sistema, azioni di rete, e così via. Quando si verifica un evento, viene generato un segnale, e il sistema reagisce eseguendo le azioni associate a quell'evento.

C

Condizioni limite

Limiti specifici o scenari particolari in cui un sistema o un componente deve operare. Queste condizioni spesso definiscono il comportamento estremo o le circostanze critiche.

F

Firme

Vengono utilizzate per identificare e rilevare comportamenti sospetti o dannosi all'interno di sistemi informatici, esse, sono contenute in database di firme, i quali associano ad ogni firma informazioni su modelli di comportamento associati a malware, intrusioni o minacce alla sicurezza.

Firewall

Un firewall è un dispositivo hardware o software progettato per monitorare, filtrare e controllare il traffico di rete tra una rete privata e una rete pubblica (come Internet).

Front-end

Il "front-end" è la parte di un'applicazione informatica o di un sito web con cui gli utenti interagiscono direttamente. È la parte visibile e accessibile agli utenti finali, responsabile dell'aspetto grafico, dell'interfaccia utente e dell'esperienza complessiva dell'utente.

H

Hardware

Componente fisica di un sistema informatico, inclusi processori, memoria, dispositivi di archiviazione e altri componenti fisici.

HTTP

È un protocollo di comunicazione utilizzato per la trasmissione di informazioni su una rete, in particolare su Internet. HTTP è alla base del World Wide Web e consente la comunicazione tra i client e i server web.

L

Layer

Strato o livello all'interno di un sistema o di un'applicazione che svolge specifiche funzioni o fornisce specifici servizi. Gli strati sono spesso organizzati in modo gerarchico per separare le responsabilità e semplificare la progettazione e la manutenzione.

N

Node

Node.js è un ambiente di runtime open-source. È progettato per consentire l'esecuzione di codice JavaScript lato server, consentendo agli sviluppatori di creare applicazioni web e server altamente scalabili e efficienti dal punto di vista delle prestazioni.

Nuxt

È un framework open-source basato su Vue.js, un framework di sviluppo per la creazione di interfacce utente. Nuxt.js semplifica il processo di sviluppo di applicazioni web Vue.js introducendo alcune caratteristiche aggiuntive e convenzioni di progetto.

P

PostgreSQL

È un sistema di gestione di database relazionali open-source. È uno dei database relazionali più avanzati e rispettati disponibili.

Protocollo

Un protocollo di rete stabilisce le regole e le convenzioni che guidano la comunicazione tra dispositivi su una rete.

Python

Python è un linguaggio di programmazione ad alto livello, interpretato e general-purpose. Noto per la sua sintassi chiara e leggibile, che lo rende facile da imparare e utilizzare. È ampiamente utilizzato per lo sviluppo di software, applicazioni web, automazione di processi, data science, intelligenza artificiale e molte altre applicazioni.

S

Software

Software indica l'insieme dei programmi e delle istruzioni che controllano il funzionamento di un sistema informatico. Può includere sistemi operativi, applicazioni e altri programmi.

Sottosistema

Un sottosistema, in una prospettiva generale, si configura come un sistema di livello inferiore e secondario rispetto a un sistema più ampio. Dipendente da quest'ultimo, il sottosistema non può operare autonomamente e ne costituisce una parte integrante.

T

TCP/IP

È un insieme di protocolli di comunicazione utilizzati per la trasmissione di dati su reti, in particolare su Internet. Questo insieme di protocolli fornisce le fondamenta per la comunicazione dati in modo affidabile e interoperabile tra dispositivi diversi all'interno di una rete.

Three-way handshake

È un processo utilizzato nei protocolli di comunicazione di rete per stabilire una connessione affidabile tra due dispositivi. La sua applicazione più comune è nel protocollo di trasmissione di dati TCP, che è uno dei protocolli fondamentali utilizzati per il trasferimento di dati su Internet.

Trade-offs

Si riferiscono alla necessità di fare delle scelte tra due o più opzioni, tenendo conto dei relativi vantaggi e svantaggi. Nella progettazione e nell'analisi di sistemi complessi, gli individui o gli sviluppatori devono spesso bilanciare diversi obiettivi e considerare le conseguenze delle loro decisioni.

V

Vue

Vue.js è un framework JavaScript progressivo utilizzato per la costruzione di interfacce utente a singola pagina e per lo sviluppo di applicazioni web reattive. È diventato popolare per la sua semplicità, flessibilità e capacità di integrazione graduale.

W

Web-server

Applicazione software che in esecuzione su un server è in grado di gestire le richieste di trasferimento di pagine web.