



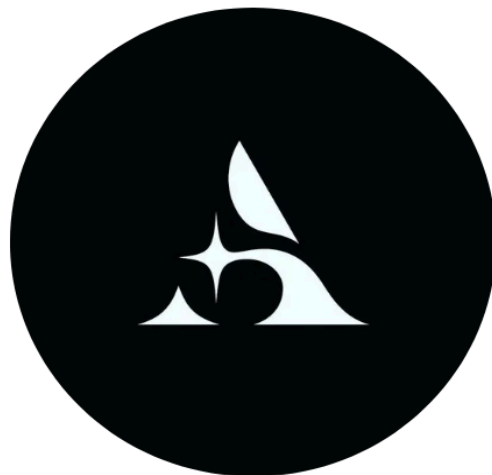
Laurea Triennale in informatica-Università di Salerno  
Corso di *Ingegneria del Software* - Prof. C. Gravino

# SDD

# System Design

# Document

# AstroVerse





Riferimento	NC31 SDD
Versione	2.0
Data	19/11/2024
Destinatario	Professore C. Gravino
Presentato da	Gruppo NC31



## Revision History

Data	Versione	Descrizione	Autori
19/11/2024	0.1	Prima stesura	Adriano De Vita
20/11/2024	0.2	Aggiunta di contenuti al Capitolo 1, Capitolo 2 e parte del 3	Tutto il team
22/11/2024	0.3	Aggiunta di Component Diagram e Diagramma Architetturale	Adriano De Vita, Christian Fontana
23/11/2024	0.4	Aggiunta 3.3 e 3.4	Adriano De Vita
24/11/2024	0.5	Creazione Boundary Conditions	Pellegrino Piccolo
25/11/2024	0.6	Modifiche diagrammi e sottosistemi 3.2 e 3.3	Christian Fontana, Christian Bianco
25/11/2024	1.0	Aggiunte e modifiche ai paragrafi dal 3.4 al capitolo 5 e rilascio prima release per revisione	Tutto il team



12/12/2024	1.1	Aggiunta sezione dati personali	Christian Fontana, Adriano De Vita
13/12/2024	1.2	Piccole modifiche al Capitolo 3	Adriano De Vita
20/12/2024	2.0	Review pre-pubblicazione	Tutto il team

## Team members

Nome	Ruolo progettuale	Contatto
Pellegrino Piccolo	Membro del team	p.piccolo4@studenti.unisa.it
Adriano De Vita	Membro del team	a.devita40@studenti.unisa.it
Christian Fontana	Membro del team	c.fontana7@studenti.unisa.it
Christian Bianco	Membro del team	c.bianco9@studenti.unisa.it



## Sommario

Revision History.....	3
Team members.....	4
Sommario.....	5
1. Introduzione.....	7
1.1 Scopo del sistema.....	7
1.2 Obiettivi di Design.....	7
Design Goals.....	8
Trade-offs.....	14
1.3 Definizioni, acronimi e abbreviazioni.....	15
1.4 Riferimenti.....	15
1.5 Panoramica del documento.....	16
2. Architettura del sistema corrente.....	16
3. Architettura del sistema proposto.....	17
3.1 Panoramica sulla sezione.....	17
3.2 Decomposizione in sottosistemi.....	18
Component Diagram.....	19
Diagramma Architeturale.....	20
3.3 Mapping hardware/software.....	20
Deployment Diagram.....	21
3.4 Gestione dei dati persistenti.....	22
3.5 Controllo degli accessi e sicurezza.....	22
3.6 Controllo globale del software.....	25
3.7 Condizioni limite.....	25
Avvio del sistema.....	26
Spegnimento del sistema.....	27
Fallimento del sistema.....	28
Arresto del sistema per cause naturali.....	29
4. Servizi dei sottosistemi.....	30
Autenticazione.....	30



Sezione dati personali.....	31
Area Spazi.....	31
Modulo IA.....	32
Gestione Post.....	33
Gestione Commenti.....	34
Amministrazione della community.....	35
5. Glossario.....	36



## 1. Introduzione

---

### 1.1 Scopo del sistema

Un gruppo di studenti del dipartimento di Informatica dell'Università degli studi di Salerno, vuole ideare un progetto riguardante una nuova applicazione web, un Social chiamato AstroVerse.

Il compito principale del sistema è quello di connettere gruppi di persone attraverso dei gruppi virtuali (spazi), sia privati che pubblici, dove gli utenti possano condividere post e idee su un argomento scelto dal creatore del canale. AstroVerse, a differenza di altri social comuni dove gli argomenti sono spesso di carattere generico e vengono trattati superficialmente, dove gli utenti condividono spesso post di bassa qualità, incentiva la creazione dei contenuti premiando l'originalità, la qualità e la creatività. Inoltre il sistema attraverso l'uso degli spazi in cui vengono trattati argomenti specifici e atomici, evita di creare ambiguità nelle informazioni lette.

### 1.2 Obiettivi di Design

Nella presente sezione presenteremo gli obiettivi di design del sistema, ovvero le qualità principali del sistema che dovrebbero richiedere maggiore attenzione. Seguendo le linee guida del libro Bernd Bruegge – “Object Oriented Software Engineering” i design goal individuati nel nostro sistema sono stati suddivisi nelle seguenti categorie:

- **Criteri di performance:** ovvero come il sistema dovrebbe comportarsi in termini di **spazio e velocità**
- **Criteri di dependability:** ovvero la quantità di sforzo che deve essere impiegata per **minimizzare gli errori del sistema e le loro conseguenze**
- **Criteri di maintenance:** ovvero quanto deve essere difficile **effettuare modifiche al sistema** dopo il rilascio effettivo
- **Criteri dell'end user:** ovvero le **qualità del sistema desiderabili dall'utente finale** ma che non sono state coperte dai criteri di performance e dependability



Ciascun design goal è descritto da:

- **Rank:** che ne specifica un valore di priorità compreso tra 1 e 15 (nello specifico basandosi sui requisiti non funzionali, da 1 a 6 priorità bassa, da 7 a 10 priorità media e da 11 a 15 priorità elevata).
- **ID Design Goal:** un identificatore univoco e un nome esplicativo.
- **Descrizione:** una descrizione del design goal.
- **Categoria:** ovvero la categoria di appartenenza del design goal.
- **RNF di origine:** ovvero il Requisito Non Funzionale che lo ha generato (ove presente).

## Design Goals

Rank	ID Design Goal	Descrizione	Categoria	RNF di origine
13	DG-1 Tempi di risposta	Il sistema deve rispondere alle operazioni degli utenti in tempi rapidi, entro un massimo di 5 secondi	Performance	RNF-P-5





11	DG-2  Navigazione concorrente	Il sistema deve funzionare correttamente anche con un alto numero di utenti connessi simultaneamente nel 98% dei casi	Performance	RNF-P-2
11	DG-3  Quantità dei dati	Il sistema deve garantire una capacità di memorizzazione adatta a conservare tutte le informazioni necessarie al suo funzionamento, deve avere una capacità minima di 50 gb per supportare i dati degli ultimi 3 anni	Performance	RNF-P-3



14	DG-4  Affidabilità delle operazioni	Il sistema deve garantire un overall Success Rate superiore al 95% e un Task Success Rate 97,5% per tutte le operazioni	Dependability	RNF-U-1  RNF-A-1
15	DG-5  Sicurezza	Il sistema deve garantire sicurezza nel trattamento delle informazioni sensibili nel 100% dei casi in particolare conformità alle leggi sulla privacy, in particolare al regolamento 2016/679 dell'UE in materia di protezione e trattamento dei dati personali	Dependability	RNF-A-2  RNF-L-1



10	DG-6  Fallimento di sistema	Il sistema deve gestire i fallimenti notificando l'utente con messaggi di errore entro al più 3 secondi dal fallimento	Dependability	RNF-A-3
12	DG-7  Riavvio di sistema	Il sistema deve garantire la preservazione dei dati in caso di riavvio improvviso nel 95% dei casi	Dependability	RNF-A-4
13	DG-8  Gestione permessi sistema	Il sistema deve garantire una chiara separazione delle operazioni in base ai permessi degli utenti attraverso l'utilizzo di 4 ruoli diversi	Dependability	RNF-A-5
8	DG-9  Disponibilità	Il sistema dovrà essere disponibile 24 ore su 24	Dependability	RNF-P-4



8	DG-10  Manutenibilità	Il sistema deve essere sviluppato secondo gli standard FMA per la manutenibilità e seguendo le linee guida di manutenzione del software stabilite dagli standard IEEE Std 14764-2006	Maintenance	RNF-IM-4
11	DG-11  Interfaccia intuitiva	L'interfaccia utente della piattaforma deve garantire un Time on Task inferiore a 30 secondi	End User	RNF-U-2
11	DG-12  Feedback esplicito	Ogni azione sulla piattaforma, avviata dall'interazione dell'utente, deve fornire un feedback chiaro, e garantire un Relearning Time di al più 60 secondi dopo un inutilizzo di un mese	End User	RNF-U-3



12	DG-13  Immediatezza di utilizzo	Il sistema deve garantire la completezza di un'attività entro al più 5 Steps to Completion. Esso inoltre deve essere accessibile tramite una pagina web da computer, smartphone e tablet entro un massimo di 1 click	End User	RNF-U-4
15	DG-14  Documentazione e per l'utente	Il sistema sarà dotato di una completa documentazione chiara e completa	End User	RNF-U-5
8	DG-15  Interfaccia responsive	Il sistema dovrà essere dotato di un'interfaccia grafica responsive testata al 100% su dispositivi reali	End User	RNF-P-1



## Trade-offs

Mostriamo di seguito alcuni trade-offs individuati sulla base di due coppie di design goals su cui è stato necessario trovare dei compromessi.

Trade-off	Descrizione
<b>Sicurezza vs Tempi di risposta</b>	Per garantire un elevato livello di sicurezza nel sistema, bisogna avere controlli su ogni accesso o richiesta fatta. Questo potrebbe tuttavia comportare un aumento del tempo di risposta, che potrebbe arrivare fino a 5 secondi.
<b>Navigazione concorrente vs Tempi di risposta</b>	Garantire il corretto funzionamento di navigazione concorrente della piattaforma AstroVerse comporterà inevitabilmente un aumento del carico computazionale, tuttavia rispettando il massimo tempo di risposta accettabile di 5 secondi.
<b>Tempi di Rilascio vs Funzionalità</b>	Per garantire la consegna del programma software entro i termini previsti verranno rilasciate solo le componenti con priorità più elevata, in successivi aggiornamenti verranno rilasciate anche le altre componenti



## 1.3 Definizioni, acronimi e abbreviazioni

In questa sezione del documento verranno elencati tutti gli acronimi e abbreviazioni utilizzati in maniera implicita all'interno del SDD (per una descrizione più dettagliata su alcuni di questi vedi 5. Glossario):

- **DG:** Design Goal
- **SDD:** System Design Document
- **RAD:** Requirements Analysis Document
- **MDT:** Matrice Di Tracciabilità
- **ODD:** Object Design Document
- **RNF:** Requisiti Non Funzionali

## 1.4 Riferimenti

Qui di seguito vengono inseriti i riferimenti agli altri documenti relativi ad AstroVerse:

[W](#) SOW AstroVerse.docx

[W](#) RAD AstroVerse.docx

[W](#) ODD AstroVerse.docx



## 1.5 Panoramica del documento

- Capitolo 1. Introduzione:** descrive in generale lo scopo del sistema, gli obiettivi di design individuati e i trade-off da rispettare.
- Capitolo 2. Architettura software corrente:** viene descritto lo stato dell'architettura del sistema ove presente.
- Capitolo 3. Architettura software proposta:** viene definita l'architettura del software da sviluppare, come verrà suddiviso in sottosistemi, il mapping Hardware/Software, la gestione dei dati persistenti. Verranno poi mostrate la struttura di ogni singolo sottosistema e le boundary conditions dell'intero sistema.
- Capitolo 4. Servizi dei sottosistemi:** vengono esplicate tutte le funzionalità di ogni singolo sottosistema.
- Capitolo 5. Glossario:** descrive i termini tecnici utilizzati nel SDD che non sono stati spiegati nei documenti di precedente fattura.

## 2. Architettura del sistema corrente

---

Al momento, l'unica soluzione software che impiega la quasi interezza delle funzionalità di AstroVerse è il Forum "Quora" che non è però un software che adotta l'approccio open-source. Il software attuale pertanto si ispira soltanto all'interfaccia grafica di tale software.





## 3. Architettura del sistema proposto

---

### 3.1 Panoramica sulla sezione

Il sistema proposto è basato interamente sull'approccio architetturale client-server implementato utilizzando come back-end il framework Spring Boot di Java e come front-end delle pagine HTML rese dinamiche attraverso l'utilizzo del framework Vue.js.

Questa scelta architetturale garantisce:

- **Manutenzione e portabilità:** attraverso la separazione in sviluppo client e sviluppo server
- **Sicurezza:** garantita attraverso le librerie di Spring Security e BCrypt per la protezione dei dati persistenti
- **Integrità e Consistenza dei dati:** la gestione autonoma del database attraverso le Spring JPA
- **Facile gestione dei dati persistenti:** da parte di Spring JPA

Per facilitare la gestione del database e quindi dei dati persistenti verranno utilizzate le Spring JPA di Spring Boot 3+ integrate con le query scritte in linguaggio SQL.

Abbiamo scelto di adottare il modello client/server per il nostro social *AstroVerse* per sfruttare un framework dinamico per il client, come Vue. Questa scelta ci consente di rendere il client più rapido e reattivo, garantendo un'esperienza utente fluida e moderna. Inoltre, il modello client/server permette di ottimizzare le risorse e alleggerire il server che si occuperà esclusivamente della gestione delle richieste provenienti dal client.

Separare le due applicazioni (client e server) quindi, non solo semplifica la manutenzione e l'aggiornamento di ciascuna parte, ma migliora anche la modularità e favorisce il riutilizzo del codice. In questo modo, possiamo implementare le tecnologie più recenti per ottenere un sistema più flessibile, scalabile e performante.



## 3.2 Decomposizione in sottosistemi

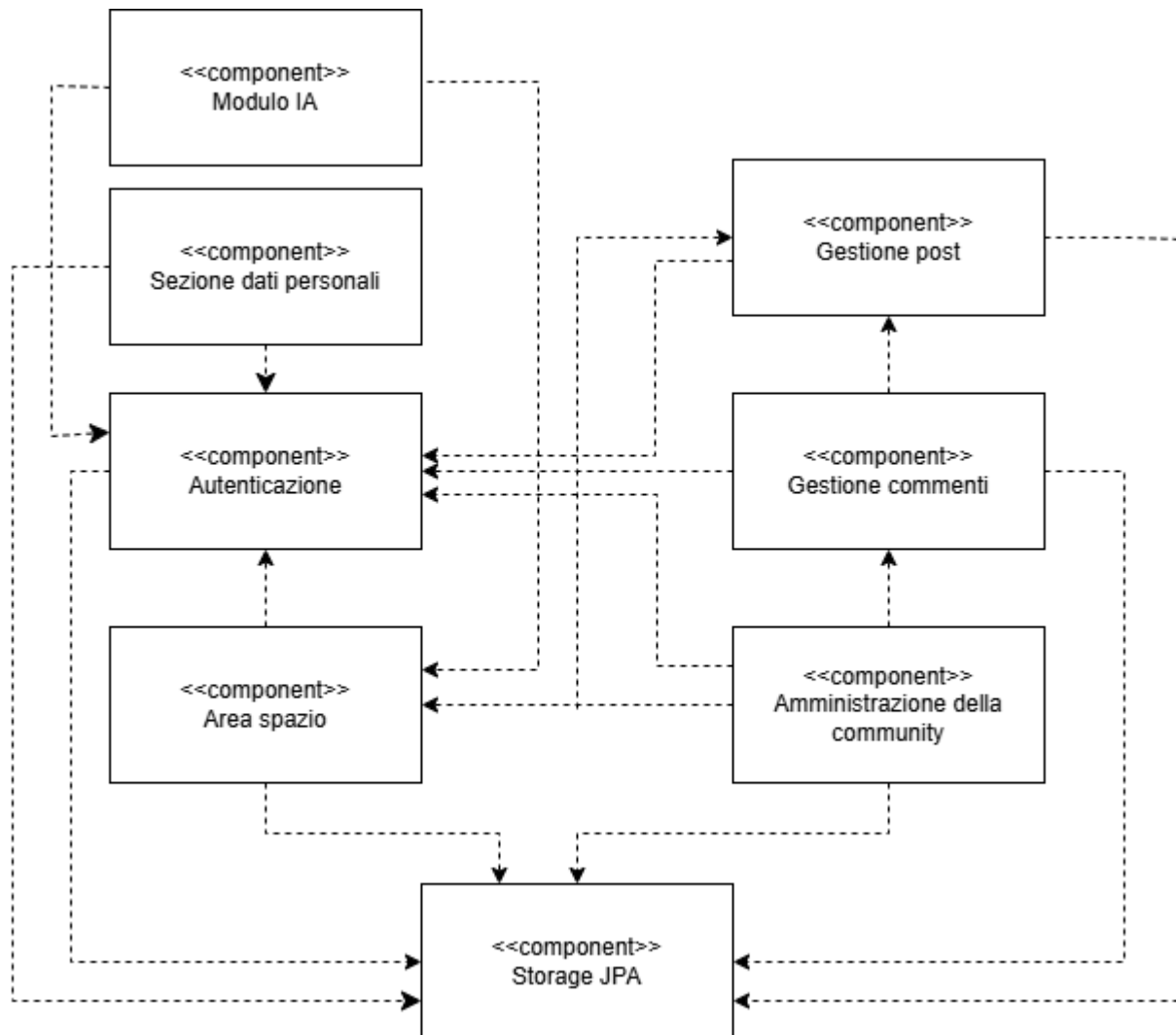
I sottosistemi individuati e che dovranno essere realizzati sono:

- **Autenticazione:** è responsabile delle funzioni di registrazione, login e logout;
- **Area spazio:** responsabile della creazione, modifica, eliminazione, visualizzazione dei membri, iscrizione spazio, abbandono spazio e visualizzazione complessiva degli spazi;
- **Modulo IA:** si occupa di suggerire gli spazi agli utenti registrati;
- **Amministrazione della community:** responsabile dei permessi per la gestione degli utenti, della moderazione e del controllo generale del sistema;
- **Storage JPA:** si interpone tra i vari sottosistemi e garantisce la persistenza dei dati;
- **Gestione commenti:** è responsabile della creazione, modifica, eliminazione proprio commento, eliminazione commento dal proprio spazio, votazione dei commenti, segnalazione commento;
- **Gestione post:** è responsabile della creazione, modifica, eliminazione proprio post, eliminazione post dal proprio spazio, votazione dei post e segnalazione post;
- **Sezione dati personali:** è responsabile della modifica e della visualizzazione dei dati personali dell'utente e dell'eventuale cancellazione dell'account.

Si è scelto di creare un'unica componente che gestisce sia la visualizzazione di spazi a cui l'utente non è iscritto sia le altre funzionalità relative agli spazi singoli a cui è iscritto. Le componenti ModuloIA e Gestione commenti siccome di priorità minore verranno rilasciate in una versione successiva al rilascio di AstroVerse. Di seguito vengono mostrate le dipendenze tra le varie componenti del sistema sopracitate attraverso un component diagram del sistema:

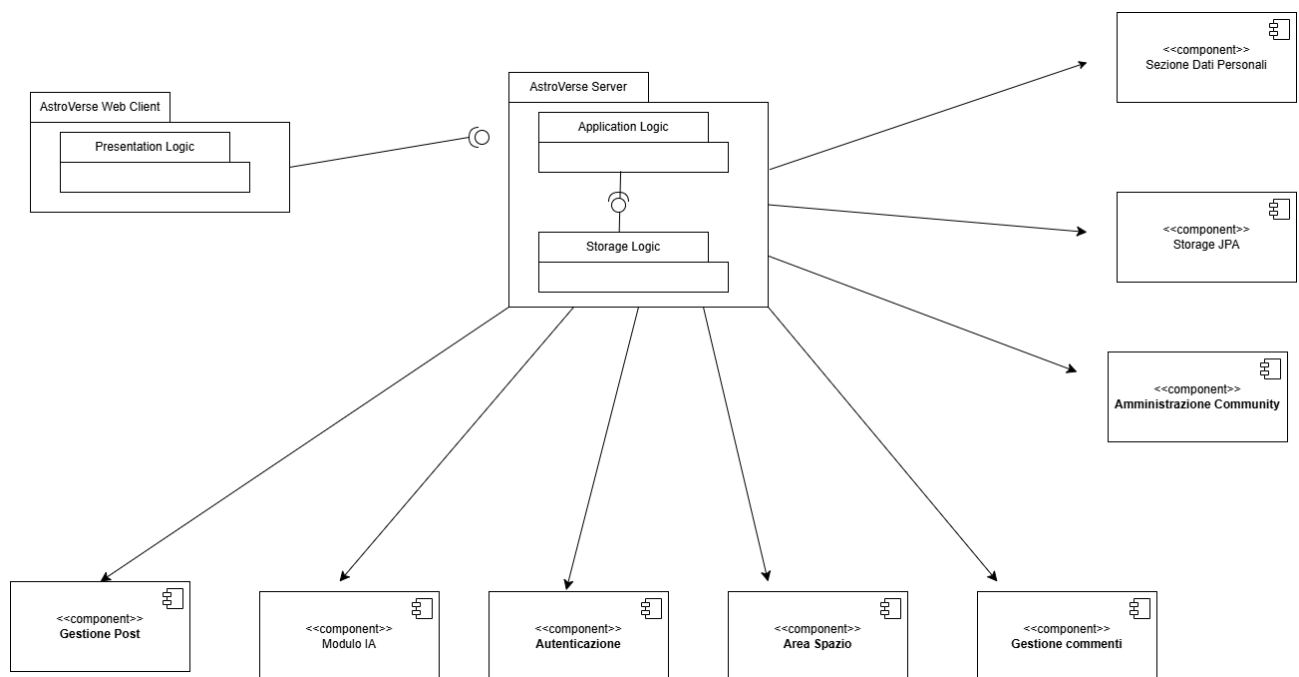


## Component Diagram



Di seguito mostriamo una vista più dettagliata della scelta architetturale client-server del nostro sistema attraverso un diagramma architetturale:

## Diagramma Architetturale

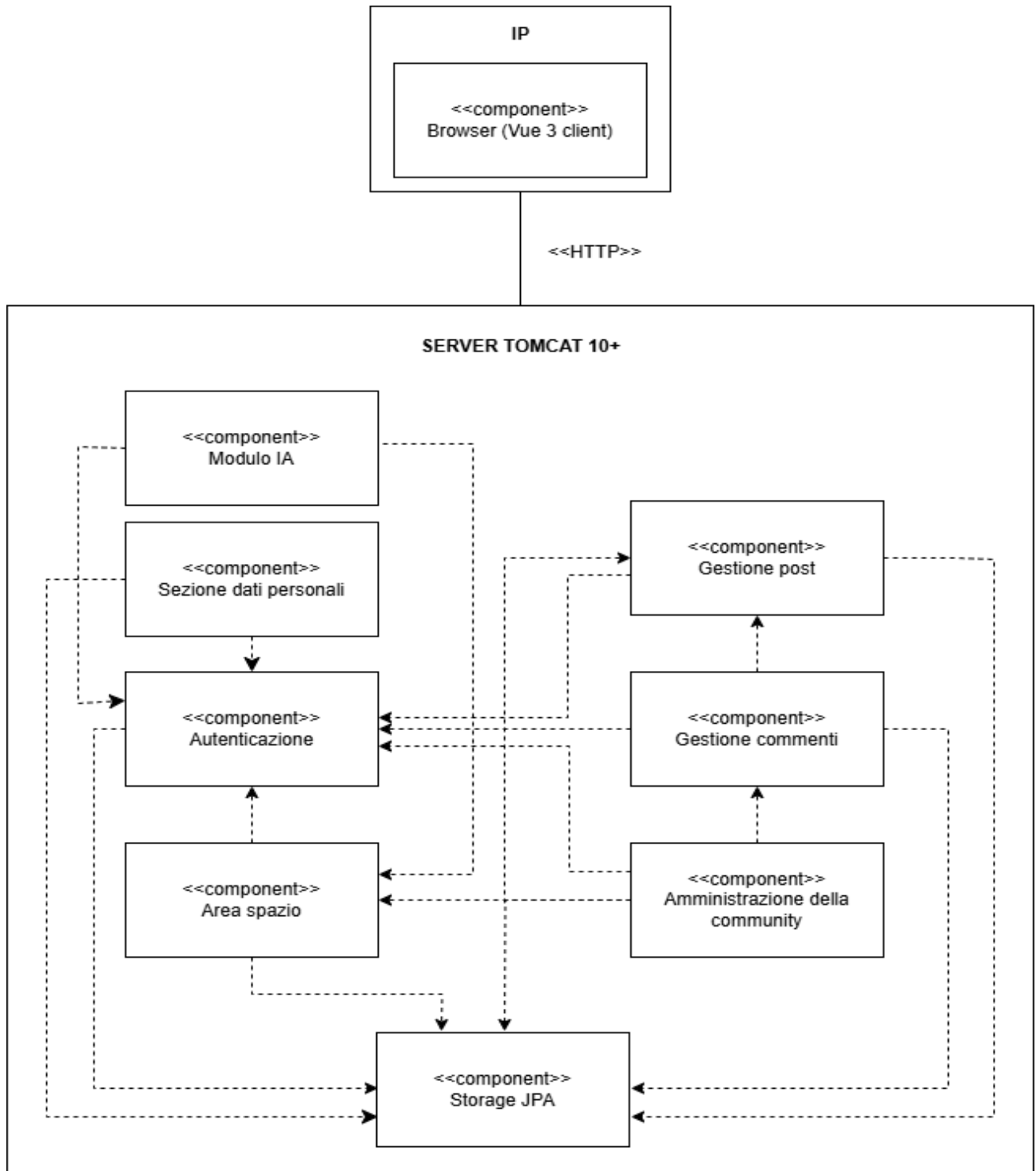


## 3.3 Mapping hardware/software

L'applicazione web in sviluppo si basa su una piattaforma hardware costituita da un server per la gestione degli accessi e dei dati persistenti che risponde alle richieste dei clienti da qualsiasi tipologia di dispositivo che possieda un browser. La nostra applicazione web, in quanto tale, risiede su un web server integrato in Spring Boot, chiamato Tomcat. La versione del server utilizzata è v10+. Di seguito è mostrato un deployment diagram che descrive il mapping hardware/software:



## Deployment Diagram





## 3.4 Gestione dei dati persistenti

Per ciò che riguarda la gestione dei dati persistenti si è pensato di utilizzare il sottosistema delle Spring Java Persistence API (JPA). La gestione dei dati avverrà dunque all'interno del server e nessun utente fatta eccezione dell'amministratore di sistema (che potrà ad ogni modo solo visualizzare alcuni dati) avrà la possibilità di accedere direttamente ad essi, garantendo quindi la trasparenza e la protezione dei dati personali. In allegato la documentazione relativa al salvataggio delle entità persistenti con l'utilizzo del framework Spring: <https://docs.spring.io/spring-data/jpa/reference/jpa/entity-persistence.html>.

## 3.5 Controllo degli accessi e sicurezza

Di seguito viene mostrata la matrice degli accessi per poter tenere traccia di quali attori possono accedere a quali dei servizi offerti dal sistema.

<b>Attori</b> <b>Oggetti</b>	<b>Amministratore spazio</b>	<b>Partecipante spazio</b>
<b>Autenticazione</b>	Login Logout Registrazione	Login Logout Registrazione
<b>Sezione dati personali</b>	Visualizzazione area utente Modifica dati utente Cancellazione account	Visualizzazione area utente Modifica dati utente Cancellazione account
<b>Area spazio</b>	Modifica spazio Eliminazione proprio spazio Visualizzazione membri spazio	Visualizzazione membri spazio Abbandono spazio



<b>Gestione post</b>	Creazione post Modifica proprio post Eliminazione post proprio spazio Eliminazione proprio post Vota post Segnalazione post	Creazione post Modifica proprio post Eliminazione proprio post Vota post Segnalazione post
<b>Gestione commenti</b>	Creazione commento Modifica commento Eliminazione proprio commento Segnalazione commento del proprio post Eliminazione commento del proprio spazio Vota commento	Creazione commento Modifica commento Eliminazione proprio commento Segnalazione commento del proprio post Vota commento

<b>Attori</b>	<b>Amministratore</b>	<b>Utente registrato</b>
<b>Oggetti</b>		
<b>Autenticazione</b>	Registrazione Login Logout	Registrazione Login Logout Visualizzazione area utente



<b>Area spazio</b>	Eliminazione spazio Creazione spazio Modifica spazio Eliminazione proprio spazio Visualizzazione membri spazio Visualizzazione spazi Iscrizione spazio Abbandono spazio	Creazione spazio Suggerimento spazio Visualizzazione spazi Iscrizione spazio
<b>Gestione post</b>	Eliminazione post Creazione Post Modifica proprio post Eliminazione post proprio spazio Vota post	
<b>Modulo IA</b>	Suggerimento spazio	Suggerimento spazio
<b>Gestione commenti</b>	Eliminazione commento Creazione commento Modifica commento Vota commento	
<b>Sezione dati personali</b>	Modifica dati utente Visualizzazione area utente Cancellazione account	Modifica dati utente Visualizzazione area utente Cancellazione account
<b>Amministrazione della community</b>	Sospensione di un account utente Ripristino dell'account utente	





## 3.6 Controllo globale del software

Il sistema AstroVerse è considerato un sistema interattivo per cui ogni funzionalità viene avviata in seguito ad un comando impartito dall'utente attraverso l'interfaccia grafica del sistema. Di conseguenza quando un utente richiede un servizio del sistema lo fa attraverso i controlli dell'interfaccia grafica corrispondenti. L'azione scatenerà quindi un evento che verrà gestito dal suo handler (gestore), esso a sua volta indirizzerà il controllo del flusso di eventi al sottosistema che gestisce la logica di controllo che poi si rivolge ai servizi della logica applicativa. Per tali motivi il sistema utilizza un meccanismo di controllo del flusso di tipo event-driven essendo una web-application.

## 3.7 Condizioni limite

Nel presente paragrafo verranno presentate le boundary condition del sistema, ovvero le condizioni limite del sistema.



## Avvio del sistema

Identificativo	UCBC1-Avvio del sistema	Data	25/11/2024
		Versione	2.0
		Autore	Adriano De Vita
Descrizione	Lo UC descrive la condizione che consente l'avvio del sistema		
Attore principale	Amministratore		
Attori secondari	NA		
Entry Condition	L'amministratore effettua l'accesso al server		
Exit condition On success	Il sistema viene avviato correttamente		
Exit condition On failure	Il sistema non viene avviato		
FLUSSO DI EVENTI PRINCIPALE			
1	Amministratore:	Esegue il comando per avviare il sistema	
2	Sistema:	Si avvia rendendo disponibili le sue funzionalità	
Flusso di eventi di errore: il sistema non si avvia			
2 a1	Amministratore:	Ripete il comando per avviare il sistema <b>come indicato sopra</b>	



## Spegnimento del sistema

Identificativo	UCBC2-Spegni mento del sistema	Data	25/11/2024
		Versione	2.0
		Autore	Adriano De Vita
Descrizione	Lo UC descrive la condizione che consente il corretto arresto del sistema		
Attore principale	Amministratore		
Attori secondari	NA		
Entry Condition	L'amministratore effettua l'accesso al server AND Il sistema è già stato precedentemente avviato dall'Amministratore AND Il sistema è ancora in esecuzione		
Exit condition On success	Il sistema viene spento correttamente		
Exit condition On failure	Il sistema non viene spento		
FLUSSO DI EVENTI PRINCIPALE			
1	Amministratore:	Esegue il comando per arrestare il sistema	
2	Sistema:	Si arresta e termina l'esecuzione delle operazioni	
Flusso di eventi di errore: il sistema non si avvia			



## Fallimento del sistema

Identificativo	UCBC3-Fallimento del sistema	Data	25/11/2024
		Versione	2.0
		Autore	Adriano De Vita
Descrizione	Lo UC descrive la condizione del sistema in una situazione di fallimento		
Attore principale	Amministratore		
Attori secondari	NA		
Entry Condition	Il sistema in conseguenza ad un errore si arresta inaspettatamente		
Exit condition On success	Il sistema viene riavviato correttamente		
Exit condition On failure	Il sistema non viene riavviato		
FLUSSO DI EVENTI PRINCIPALE			
1	Amministratore:	Esegue il comando per riavviare il sistema <b>vedi UCBC1</b>	
Flusso di eventi di errore: il sistema non si riavvia			
2 a1	Amministratore:	Ripete il comando per riavviare il sistema <b>come indicato sopra</b>	



### Arresto del sistema per cause naturali

Identificativo	UCBC4-Arresto del sistema per cause naturali	Data	25/11/2024
		Versione	2.0
		Autore	Adriano De Vita
Descrizione	Lo UC descrive la condizione in cui il sistema si arresta per cause naturali		
Attore principale	Amministratore		
Attori secondari	NA		
Entry Condition	Il sistema si è arrestato improvvisamente per cause naturali (mancanza di alimentazione, malfunzionamento delle piattaforme hardware, calamità naturali)		
Exit condition On success	Il sistema viene riavviato correttamente dopo un certo periodo di tempo		
Exit condition On failure	Il sistema non viene riavviato		
FLUSSO DI EVENTI PRINCIPALE			
1	Amministratore:	Dopo un certo periodo di tempo esegue il comando per riavviare il sistema <b>vedi UCBC1</b>	
Flusso di eventi di errore: il sistema non si riavvia			
2 a1	Amministratore:	Attendo un altro periodo di tempo dopodichè esegue nuovamente il comando per riavviare il sistema <b>come indicato sopra</b>	



## 4. Servizi dei sottosistemi

In questa sezione vengono descritti i servizi di ogni sottosistema elencato nella decomposizione al 3.2.

### Autenticazione

Servizio	descrizione	Interfaccia
<b>Registrazione utente</b>	Questa funzionalità permette di registrarsi sulla piattaforma	RegistrazioneService
<b>Login</b>	Questa funzionalità permette di effettuare l'accesso al sistema tramite le proprie credenziali per sfruttare tutte le funzionalità che offre	AutenticazioneService
<b>Logout</b>	Questa funzionalità permette di disconnettersi dal sistema	AutenticazioneService



### Sezione dati personali

Servizio	descrizione	Interfaccia
<b>Visualizza area utente</b>	Permette di visualizzare i dati relativi alla propria area utente	AutenticazioneService
<b>Modifica dati utente</b>	Permette di modificare i dati relativi alla propria area utente	AutenticazioneService
<b>Eliminazione account</b>	Permette di cancellare il proprio account sulla piattaforma	AutenticazioneService

### Area Spazi

Servizio	descrizione	Interfaccia
<b>Visualizzazione lista Spazi</b>	Questa funzionalità permette di visualizzare la lista degli Spazi	AreaSpazioService
<b>Creazione Spazio</b>	Questa funzionalità permette di creare un nuovo spazio	AreaSpazioService
<b>Modifica Spazio</b>	Questa funzionalità permette di modificare uno spazio dall'amministratore dello spazio	AreaSpazioService



<b>Eliminazione spazio</b>	Questa funzionalità permette di eliminare uno spazio	AreaSpazioService
<b>Visualizzazione dei membri di uno spazio</b>	Questa funzionalità permette di visualizzare tutti i membri di uno spazio	AreaSpazioService
<b>Iscrizione spazio</b>	Questa funzionalità permette di iscriversi ad uno spazio	AreaSpazioService
<b>Abbandono spazio</b>	Questa funzionalità permette di abbandonare uno spazio	AreaSpazioService

## Modulo IA

Servizio	descrizione	Interfaccia
<b>Suggerimento spazio</b>	Questa funzionalità suggerisce all'utente la partecipazione a determinati spazi	ModuloAIService





## Gestione Post

Servizio	descrizione	Interfaccia
<b>Creazione</b>	Questa funzionalità permette la creazione di un post	PostService
<b>Modifica</b>	Questa funzionalità permette la modifica di un post	PostService
<b>Eliminazione proprio post</b>	Questa funzionalità permette l'eliminazione di un post	PostService
<b>Eliminazione post proprio spazio</b>	Questa funzionalità permette l'eliminazione di un post dal proprio spazio dall'amministratore dello spazio	PostService
<b>Votazione dei post</b>	Questa funzionalità permette la votazione dei post	PostService
<b>Segnalazione post</b>	Questa funzionalità permette di segnalare un post	PostService



## Gestione Commenti

Servizio	descrizione	Interfaccia
<b>Creazione</b>	Questa funzionalità permette la creazione di un commento	CommentoService
<b>Modifica</b>	Questa funzionalità permette la modifica di un commento	CommentoService
<b>Eliminazione proprio commento</b>	Questa funzionalità permette l'eliminazione di un commento	CommentoService
<b>Eliminazione commento proprio spazio</b>	Questa funzionalità permette l'eliminazione di un commento dal proprio spazio dall'amministratore dello spazio	CommentoService
<b>Votazione dei commenti</b>	Questa funzionalità permette la votazione dei commenti	CommentoService
<b>Segnalazione commento</b>	Questa funzionalità permette di segnalare un commento	CommentoService



### Amministrazione della community

Servizio	descrizione	Interfaccia
<b>Sospensione di un account</b>	Questa funzionalità permette all'amministratore del sistema di sospendere un account utente	AmministratoreService
<b>Ripristino di un account</b>	Questa funzionalità permette all'amministratore del sistema di ripristinare un account utente	AmministratoreService
<b>Eliminazione post</b>	Questa funzionalità permette all'amministratore del sistema di eliminare un post	AmministratoreService
<b>Eliminazione commento</b>	Questa funzionalità permette all'amministratore del sistema di eliminare un commento	AmministratoreService



## 5. Glossario

Nella presente sezione sono raccolte le sigle o i termini del documento SDD che necessitano di una definizione più appropriata e chiara per l'utente finale e che non sono state definite nel RAD.

Sigla/Termine	Definizione
Trade-offs	Compromessi in termini di vantaggi e svantaggi di un sistema software tra due obiettivi di design
Obiettivi di design	Criteri fondamentali che guidano la progettazione di un sistema software aziendale che definiscono le caratteristiche desiderate dall'utente finale che soddisfano i requisiti funzionali e non funzionali
Performance	Misura dell'insieme di prestazioni che un sistema software deve possedere
Dependability	Insieme di obiettivi di design che un sistema software dovrebbe rispettare al fine di fornire servizi sicuri e affidabili
Maintenance	Insieme di obiettivi di design che garantiscono la manutenibilità del sistema software
End User	Insieme di obiettivi di design richiesti dall'utente finale per scopi di facilità di utilizzo del sistema software



Rank	Livello di priorità di realizzazione assegnato ad un obiettivo di design
Mapping Hardware/Software	Processo che assegna delle funzionalità e dei componenti software a specifiche risorse hardware del sistema
Sottosistema	Componente software che realizza alcuni obiettivi di design e requisiti
Back-end	"Scheletro" del sistema software costituito dal server e da tutte le componenti che realizzano la gestione della logica applicativa e dei dati persistenti memorizzati
Front-end	Rappresenta la parte del software visibile all'utente con la quale può interagire attraverso l'interfaccia grafica
Framework	Struttura software riutilizzabile che fornisce un insieme standardizzato di funzionalità per facilitare lo sviluppo e la gestione del sistema
Component Diagram	Diagramma che fornisce una visione dell'architettura del sistema mostrando i vari sottosistemi impiegati
Diagramma Architetturale	Diagramma che descrive la struttura e l'organizzazione generale del sistema software



Deployment Diagram	Diagramma che mostra il mapping Hardware/Software del sistema software sviluppato
Boundary Conditions	Condizioni limite (critiche) del sistema software
Service	Interfacce che forniscono funzionalità all'utente del sistema
Query	Richieste fatte da parte del sistema ai dati persistenti per visualizzare, modificare, eliminare o interagire con tali dati
Vue	Linguaggio di programmazione per lo sviluppo di componenti front-end dinamiche