Immagine che contiene scatola

Descrizione generata automaticamente

Università degli Studi di Salerno

Immagine che contiene testo, clipart

Descrizione generata automaticamente

Dipartimento di Ingegneria dell’Informazione ed Elettrica e Matematica Applicata

Corso di Laurea in Ingegneria Informatica

**Sistemi Informativi 2021/2022**

**Traccia N. 3 - Emergenza violenza di genere**

|  |  |  |
| --- | --- | --- |
| **Cognome e Nome** | **Matricola** | **E-mail** |
| **Apicella Salvatore** | **0612704656** | **s.apicella31@studenti.unisa.it** |
| **Ferrigno Antonio** | **0612794667** | **a.ferrigno25@studenti.unisa.it** |

**Anno accademico 2021-2022**

**Indice**

[1. Descrizione dell’organizzazione 2](#_Toc104737487)

[1.1 Macrostruttura organizzativa 2](#_Toc104737488)

[1.2 Gerarchia di processo 2](#_Toc104737489)

[2. Descrizione del processo 3](#_Toc104737490)

[3. Flusso del processo 4](#_Toc104737491)

[3.1 Lavori di commissione per redazione report 4](#_Toc104737492)

[3.2 Task ad Hoc Risoluzione parlamentare 5](#_Toc104737493)

[3.3 Task Audizione con gli esperti 5](#_Toc104737494)

[3.4 Task ad Hoc analisi della realtà di interesse da parte degli esperti 6](#_Toc104737495)

[4. Proposta architetturale 7](#_Toc104737496)

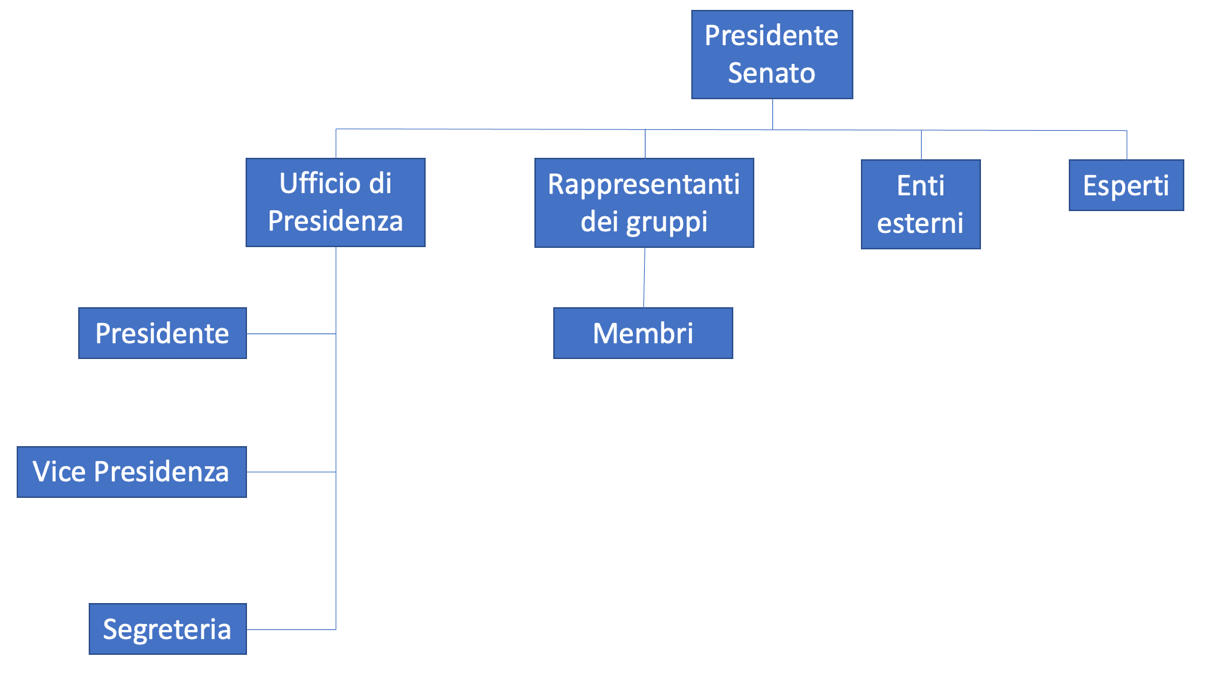
[5. Viste architetturali 8](#_Toc104737497)

[5.1 Astratta 8](#_Toc104737498)

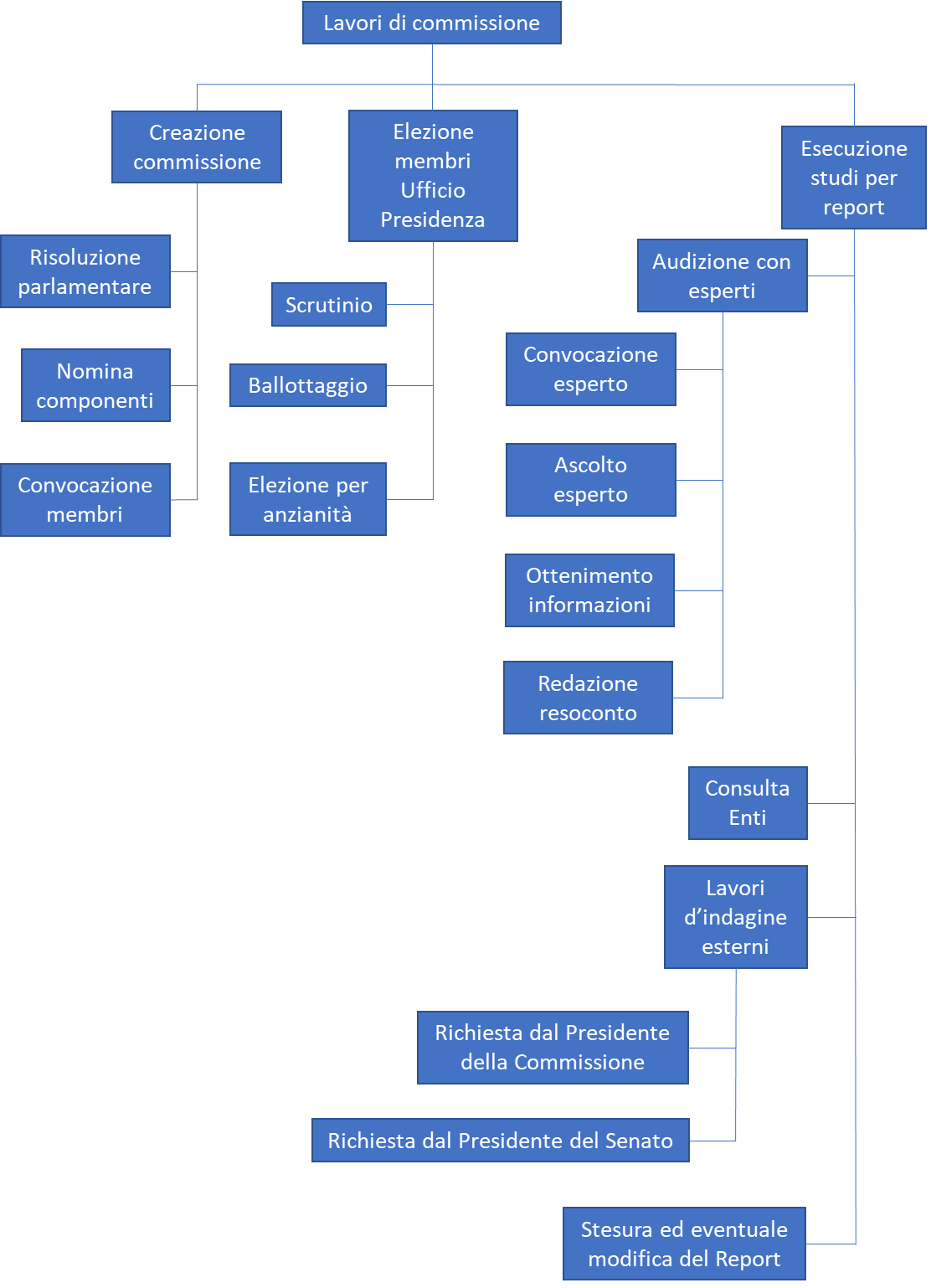
[5.2 Deployment 10](#_Toc104737499)

# Descrizione dell’organizzazione

## Macrostruttura organizzativa



## Gerarchia di processo



# Descrizione del processo

**ELABORAZIONE DI UN REPORT DA PARTE DI UNA COMMISSIONE D’INCHIESTA**

Una “Commissione parlamentare di inchiesta sul femminicidio, nonché su ogni forma di violenza di genere” viene istituita per interesse della camera del Senato per effettuare indagini da affiancare a quelle della magistratura. Lo scopo della Commissione è quella di svolgere indagini sulla reale dimensione del fenomeno del femminicidio e su ogni forma di violenza di genere in Italia e di redigere un report. Il report viene compilato al fine di concedere al Senato un documento a sostegno dell’approvazione di un disegno di legge sulla materia d’interesse.

Il presidente della camera del Senato, attraverso risoluzione parlamentare, stabilisce contemporaneamente i senatori coinvolti in proporzione al numero dei componenti dei gruppi parlamentari, assicura la presenza di un rappresentante per ciascun gruppo e favorisce l’equilibrata rappresentanza di senatrici e senatori.

Il Presidente del Senato, entro dieci giorni dalla nomina dei componenti, convoca la Commissione per la costituzione dell’ufficio di presidenza.

L’ufficio di presidenza, composto dal presidente, da due vicepresidenti e da due segretari, è eletto dai componenti della Commissione a scrutinio segreto. Nell’elezione del presidente, se nessun componente riporta la maggioranza assoluta dei voti, si procede al ballottaggio tra i due candidati che hanno ottenuto il maggior numero di voti. In caso di parità di voti viene eletto il più anziano di età.

Prima di far svolgere ad un membro della commissione ulteriori indagini al di fuori del Parlamento, il Presidente della camera del Senato deve essere informato e deve approvare gli spostamenti del membro direttamente dalla commissione attraverso il Presidente della Commissione.

La commissione convoca l’esperto in Parlamento per svolgere un’analisi del fenomeno consultando testi accademici, analizzando appositi indici che colloca in determinate tabelle esplicative, consultando database con specifici dati statistici, redigendo questionari per l’analisi statistica ed infine svolgendo interviste direttamente ai soggetti d’interesse. In caso di impossibilità di partecipazione per malattia o dimissione dell’esperto, egli comunica la sua mancata disponibilità alla commissione, altrimenti discute la sua analisi nella sede del Parlamento. Se l’esperto è anche un rappresentante di un ente pubblico, la commissione si occupa della redazione di un resoconto dell’intervento dell’esperto.

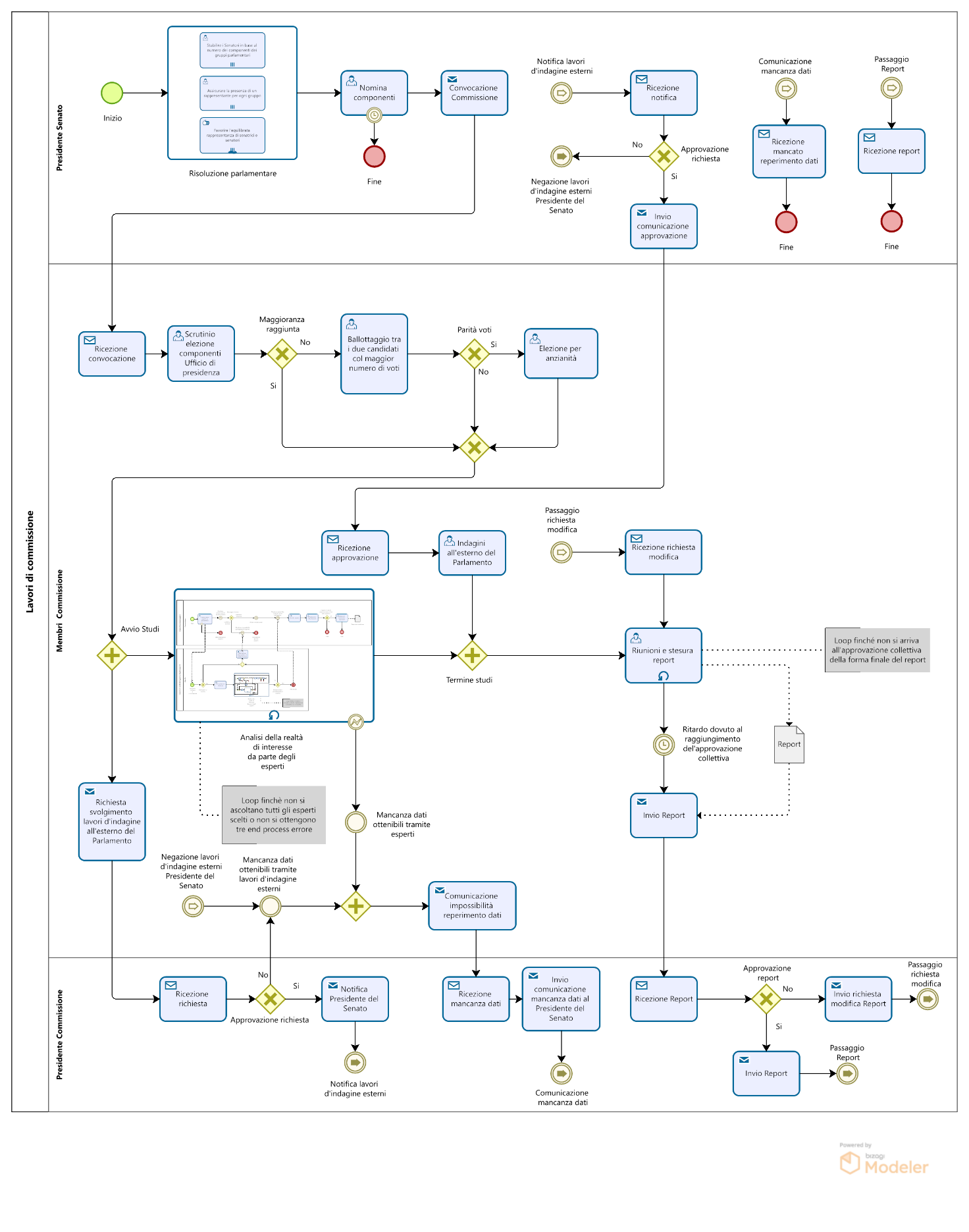
In caso di impossibilità di esecuzione delle indagini dall’esterno del Parlamento e di mancato reperimento dei dati attraverso la consultazione di enti ed esperti per mancata risposta o indisponibilità di partecipazione, i membri comunicano l’impossibilità di elaborazione del report al presidente della commissione, il quale avvisa di tale problematica il presidente del Senato, terminando dunque il processo.

Ottenute tutte le informazioni, vengono svolte opportune riunioni e viene redatto il report complessivo, il quale viene approvato dal presidente della commissione ed infine viene concesso al Senato.

In caso di mancata approvazione da parte del presidente della commissione, egli comunica ai membri di svolgere opportune modifiche al report, le quali verranno apportate col consenso di tutti i membri tramite opportune riunioni.

# Flusso del processo

## Lavori di commissione per redazione report

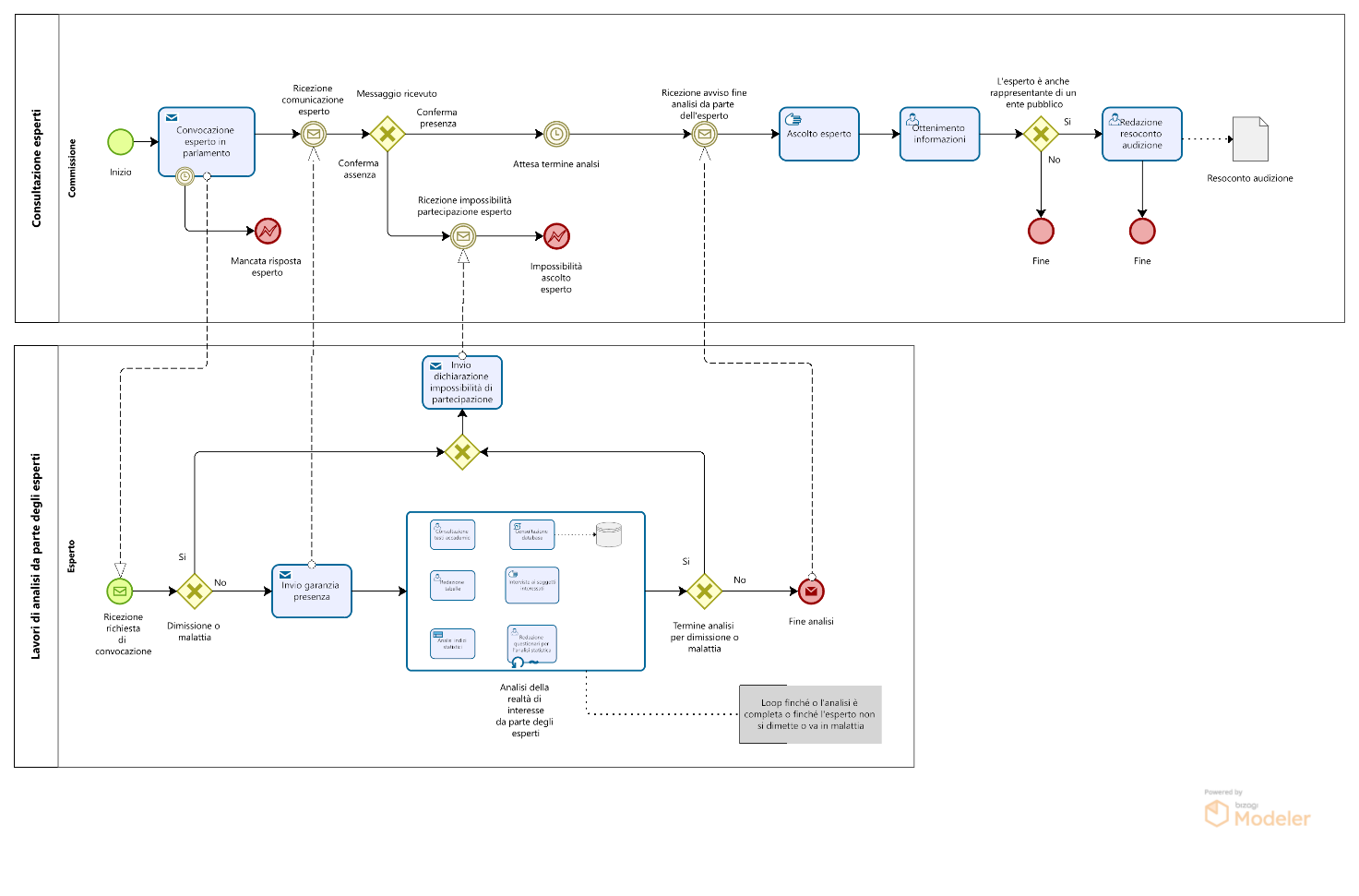


## Task ad Hoc Risoluzione parlamentare

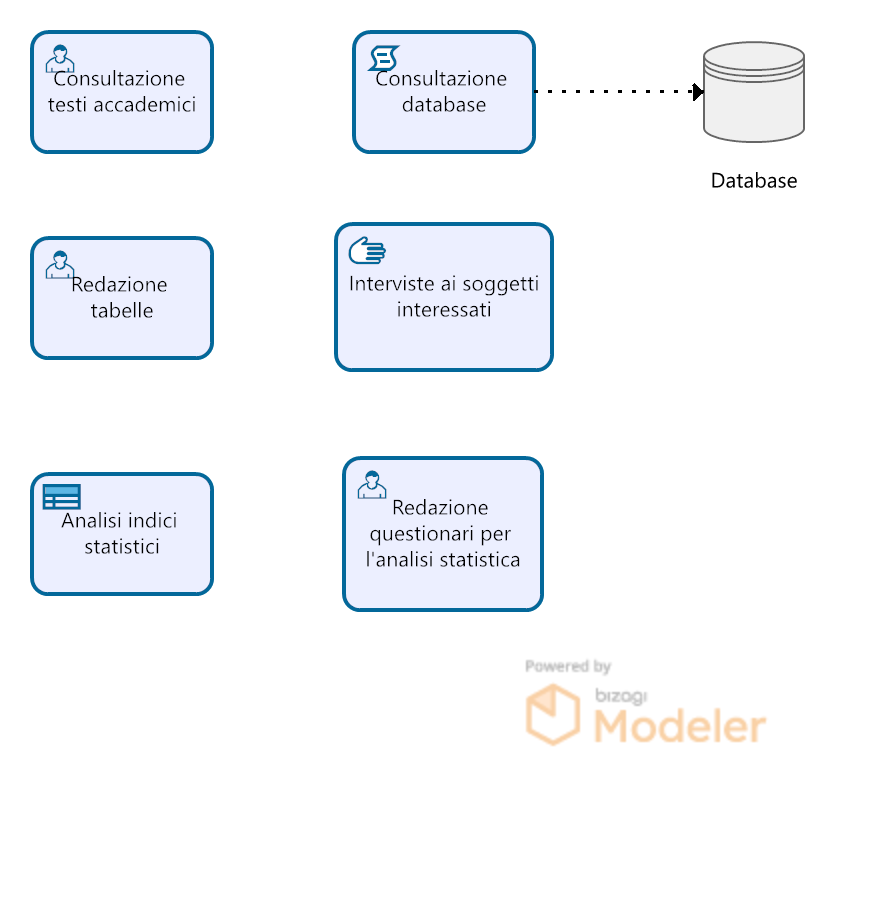
Immagine che contiene testo

Descrizione generata automaticamente

## Task Audizione con gli esperti



## Task ad Hoc analisi della realtà di interesse da parte degli esperti



# Proposta architetturale

L’architettura proposta in base al processo analizzato è un’architettura di tipo micro-servizi distribuiti, implementato nello specifico facendo riferimento a quattro sottosistemi, ovvero sottosistema Client, sottosistema Server, sottosistema Data Storage e sottosistema External Services, impostata con un pattern MVC.

Abbiamo scelto questo stile architetturale per la rapida comunicazione tra i membri della commissione, per la redazione del report condiviso da parte di più membri in contemporanea da più postazioni, per la scelta della propria preferenza in base all’utente che interagisce con il sistema in caso di nomina dei componenti della commissione o dell’ufficio di presidenza ed infine per la semplificazione di accesso al sistema.

L’architettura Client-Server è stata scelta poiché il carico applicativo sul sistema per la gestione della modifica del report condiviso è variabile e dipende dal numero di utenti connessi simultaneamente.

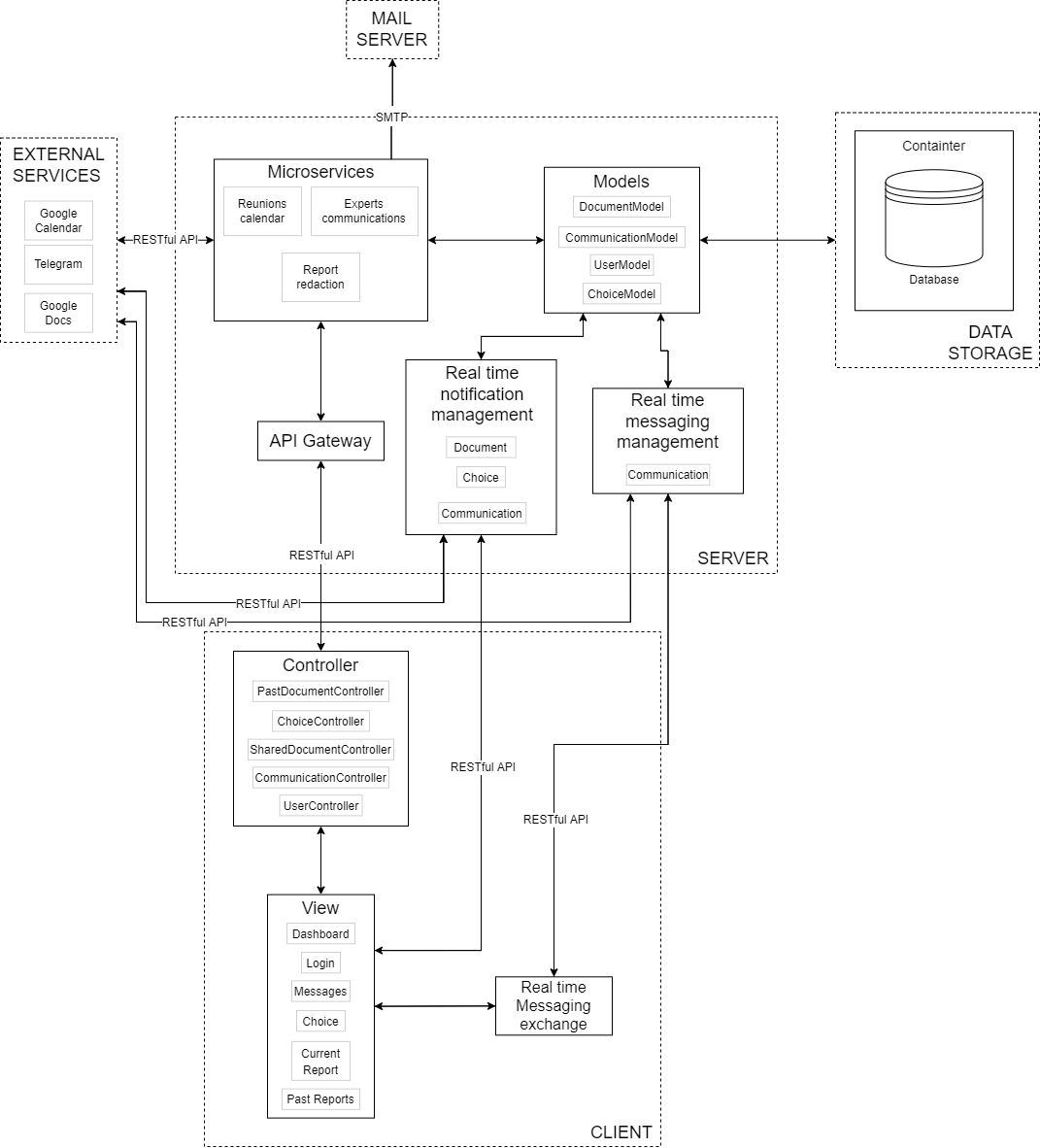
Il pattern MVC è stato prediletto poiché vi sono più modi di visualizzare ed interagire con i dati e per la capacità di consentire ai dati di cambiare indipendentemente dalla loro rappresentazione e viceversa.

Inoltre, quest’ultimo supporta la presentazione degli stessi dati in modi differenti, con le modifiche fatte in una rappresentazione mostrate in tutte le altre.

Un esempio di modi di visualizzazione diversa potrebbe essere il caso in cui il presidente del Senato, nella scelta dei membri per la creazione della commissione, vedrà tutti i senatori della camera del Senato e potrà selezionarne un numero opportuno in base alle sue esigenze, mentre i membri della commissione, al fine di eleggere i membri per la rappresentanza dell’ufficio di presidenza, avranno una visione ridotta soltanto ai senatori che sono già membri della commissione.

# Viste architetturali

## Astratta



Nel seguente modello, il client contiene la View, la gestione dello scambio di messaggi in tempo reale (al fine di garantire un servizio di comunicazione con gli altri membri rapido durante la redazione del report) ed il controller.

La View, tra i diversi moduli presentati, è caratterizzata da una dashboard per scegliere i servizi da selezionare, un modulo di login dell’utente che accede alla piattaforma, un modulo per la visualizzazione e l’invio dei messaggi, un modulo attivabile in caso di scelta dei membri da parte del presidente del Senato o di elezione dei rappresentanti dell’ufficio di presidenza, un modulo per la redazione condivisa del report corrente ed infine un modulo per visualizzare la struttura e i riferimenti dei report precedentemente inseriti nel database.

Nel controller vengono utilizzati moduli per la gestione delle richieste del client dalla View.

Il server è invece caratterizzato da un modulo Models, un modulo per la gestione dei micro-servizi con relativa API gateway per l’implementazione degli stessi, un modulo di gestione in tempo reale di messaggistica ed un modulo di gestione in tempo reale delle notifiche.

I micro-servizi offerti, definiti per la gestione della calendarizzazione delle riunioni o delle elezioni, per la comunicazione con gli esperti delle realtà di interesse e per la redazione del report condiviso, comunicano all’esterno del server tramite apposite RESTful API.

Il modulo Models detiene al suo interno i principali modelli di riferimento per l’ottenimento delle informazioni da parte del database al fine di consentire la visione del report corrente e dei report passati (DocumentModel), i messaggi tra utenti (CommunicationModel), la gestione delle informazioni di accesso e di inserimento all’interno del DB (UserModel) e la gestione della scelta dei membri selezionati in base all’utente o all’evento (ChoiceModel).

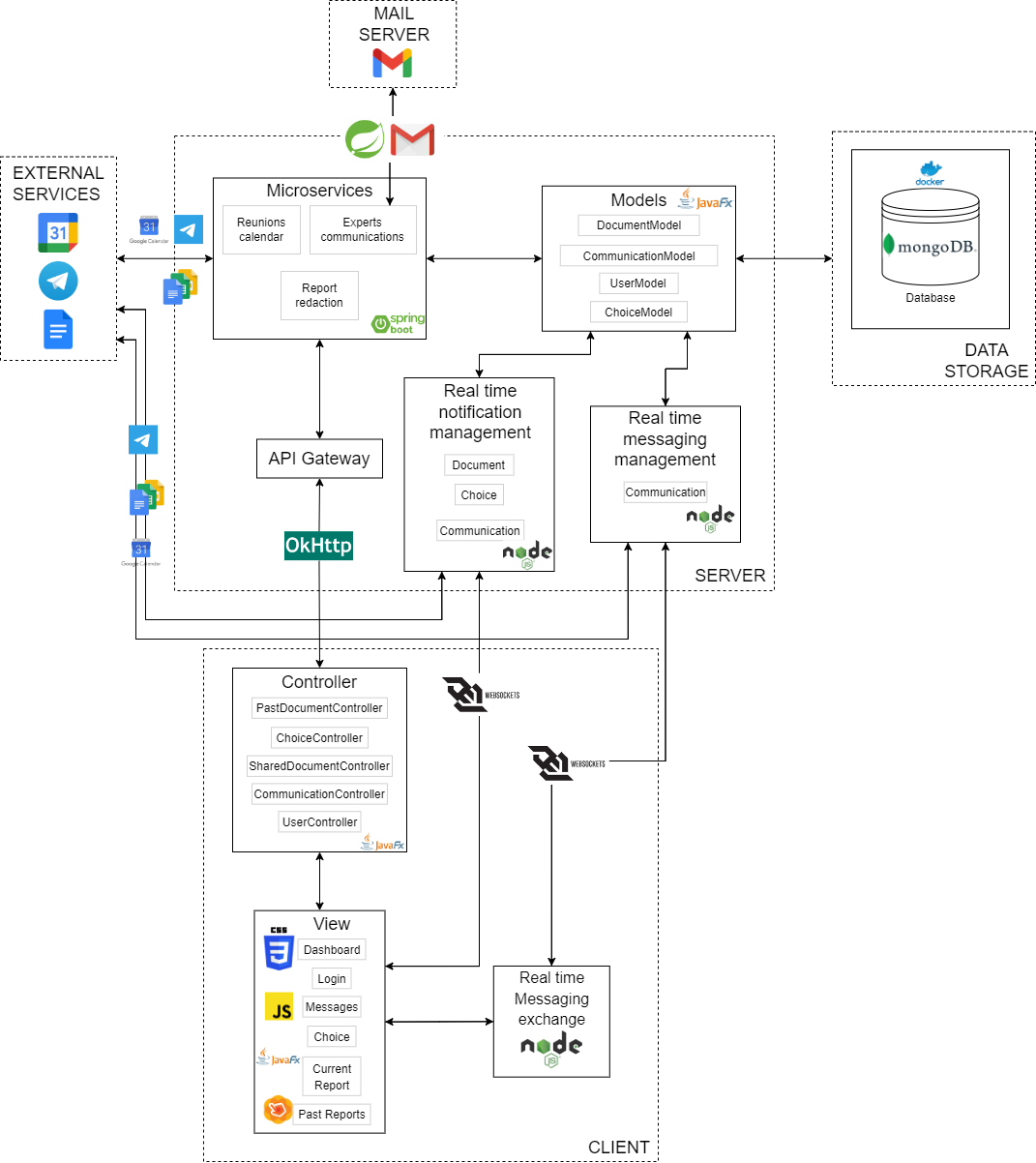
Il database è stato racchiuso in un container per distribuire e ricalibrare le risorse in qualsiasi ambiente e per garantire la tracciabilità dei dati.

Sono infine presenti un modulo di gestione in tempo reale di notifica ed un modulo per la gestione in tempo reale della comunicazione in modo da garantire la visualizzazione delle informazioni mandate verso il modulo Model da parte del modulo di View.

In particolar modo, il primo modulo implementa ulteriori sotto-moduli per la notifica nel caso di modifica del documento corrente, in caso di comunicazione di richiesta partecipazione a far parte della commissione o di vittoria dell’elezione come componente dell’ufficio di segreteria ed in caso di ricezione dei messaggi da parte dei membri durante la stesura del report.

Il secondo modulo viene invece utilizzato per la messaggistica tra utenti che interagiscono con il sistema.

## Deployment



Tra le tecnologie scelte per la realizzazione dell’architettura proposta troviamo:

* JavaFX per la realizzazione dei moduli Models, Controller e View (in particolar modo supportato da Scene Builder);
* NodeJS per i moduli di gestione delle notifiche e dei messaggi in tempo reale;
* Spring boot per la gestione dei micro-servizi;
* MongoDB per la strutturazione del database in documenti, racchiuso nel container modellizzato tramite tecnologia Docker;
* Google Calendar, Google Docs e Gmail per l’implementazione dei micro-servizi e relative RESTful API per la comunicazione con il server;
* CSS3 e JavaScript per il miglioramento e la dinamicità grafica della View;
* WebSockets per lo scambio dei messaggi e delle notifiche ed OkHttp per la comunicazione tra controller e API, utilizzate per la gestione dei micro-servizi.