

Smart Home.

Progetto realizzato da:

Daniele Caliari (816202)
Francesco Mollica (821124)
Filippo Martini (816209)

SOMMARIO

SOMMARIO.....	3
Una breve introduzione.....	4
Ecco quindi com'è nata l'idea... ..	5
Analisi e progettazione	6
Diagramma casi d'uso:	6
Casi d'uso (formato breve):.....	7
CASO D'USO C1:.....	8
CASO D'USO C2:.....	11
Modello di dominio	12
diagramma dell'architettura	13
diagramma delle classi di progetto	14
Dominio	14
Middleware	15
Gui	16
Diagramma delle attività per l'esecuzione di un'operazione	17
Diagramma stati di un dispositivo	18
Pattern Utilizzati... ..	19

Una breve introduzione...

Tra le tante discipline che si occupano di innovazione tecnologica ed automazione, ritroviamo, in particolare, la domotica. Il termine nasce dall'unione di due elementi: la parola "domus", che in latino significa casa, e "ticos", suffisso greco comunemente utilizzato per indicare le scienze applicate. La domotica, difatti, si occupa proprio di questo: di studiare le più avanzate tecnologie e di portarle all'interno del contesto abitativo ai fini di migliorare le condizioni di vita dell'uomo e di rivoluzionare, di conseguenza, il suo modo di abitare. La domotica, punta quindi alla creazione di una abitazione intelligente, in cui l'essere umano si muove ed interagisce con gli strumenti tecnologici a disposizione in maniera intuitiva, ottenendo un controllo mirato ed efficace dell'intera abitazione. In breve in una casa intelligente è possibile:

- usufruire di meccanismi automatici che regolano in maniera indipendente, ad esempio, l'accensione e lo spegnimento degli elettrodomestici, con un conseguente risparmio di elettricità ed una gestione più consapevole dell'impianto elettrico anche da un punto di vista ecologico.
- monitorare tramite sistemi automatici di rilevamento il corretto funzionamento degli impianti casalinghi, sventando per tempo eventuali problemi tecnici o malfunzionamenti.
- verificare i consumi istantanei e su base temporale grazie a meccanismi di previsione affidabili.

In base alle proprie esigenze, ciascun individuo può rendere la propria abitazione più o meno intelligente, scegliendo perciò cosa monitorare quando vuole.

Per installare efficacemente i sistemi domotici all'interno dell'abitazione, è necessario stabilire la connessione necessaria per far sì che tutti i dispositivi installati comunichino correttamente tra loro e che inviino le informazioni rilevanti al sistema di controllo principale. Quest'ultimo, riceve i dati raccolti dai singoli dispositivi e li invia al controller gestito dall'utente, consentendogli di monitorare in tempo reale il corretto funzionamento del sistema domotico e di interagire con lo stesso a suo piacimento.

In ogni caso, le interfacce dedicate alla gestione dei sistemi domotici vengono programmate appositamente da risultare semplici e veloci da usare, permettendo a qualsiasi utente di utilizzare in maniera intuitiva il sistema intelligente della propria casa.

Ecco quindi com'è nata l'idea...

A partire dai documenti forniti dalla prof.ssa Arcelli abbiamo pensato ad un'applicazione che potesse soddisfare quelli che sono i "goals" del progetto sulla base delle conoscenze acquisite durante i tre anni, specialmente per quanto riguarda i corsi di Analisi e Progettazione software e Ingegneria del Software.

Ecco quindi che nasce la nostra idea di sviluppo che mette al centro del lavoro svolto la volontà di costruire un'applicazione che interagisca con un generico framework che comunichi con i dispositivi connessi basandosi sullo standard ZigBee e la realizzazione di una "semplice" interfaccia grafica che permetta all'utente di interagire con la sua casa intelligente.

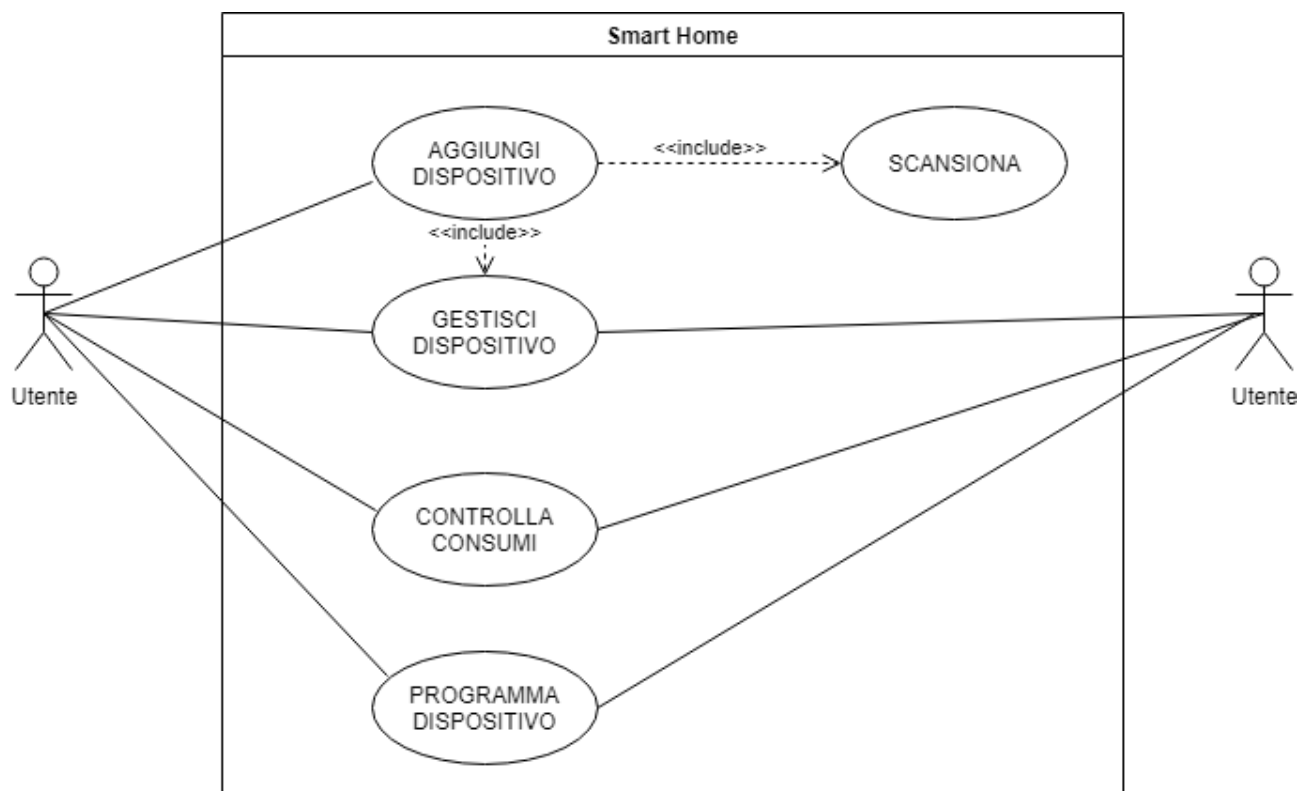
Il disaccoppiamento dell'applicazione da tale framework è stato reso possibile grazie uno strato di software da noi denominato Middleware, aperto al cambiamento a seconda del tipo di protocollo di comunicazione che il framework implementa.

Il sottosistema Middleware da noi concepito implementa la comunicazione con la *REST APIs del framework JEMMA*, in grado di comunicare con gli elettrodomestici, nonché con tutti gli oggetti e dispositivi della casa connessi in rete.

In concomitanza con gli obiettivi del corso, il tempo a disposizione e il numero di componenti del gruppo abbiamo preferito quindi progettare la nostra applicazione interamente in JAVA, lasciando l'accessibilità via web e la multi-utenza come sviluppo futuro.

Analisi e progettazione

Diagramma casi d'uso:



Casi d'uso (formato breve) :

-Aggiungi dispositivo:

Il sistema permette l'aggiunta di un dispositivo domestico alla propria smart home tramite il riconoscimento da parte del gateway (framework) di nuovi dispositivi dotati di uno standard di comunicazione wireless, ciò consentirà di gestire il dispositivo in maniera intelligente direttamente dalla applicazione.

-Gestisci dispositivo in tempo reale:

Con la gestione del dispositivo in tempo reale potremo interagire con i nostri elettrodomestici/dispositivi direttamente dalla nostra applicazione.

-Programma dispositivo:

Con questa funzionalità l'utente potrà gestire i propri elettrodomestici/dispositivi in modo tale da programmare le azioni future degli stessi sulla base delle sue preferenze e abitudini.

-Controlla consumi:

Il sistema fornirà le statistiche dei consumi attuali e sulla base di quest'ultimi garantirà previsioni sui consumi futuri e consigli per un migliore utilizzo degli elettrodomestici.

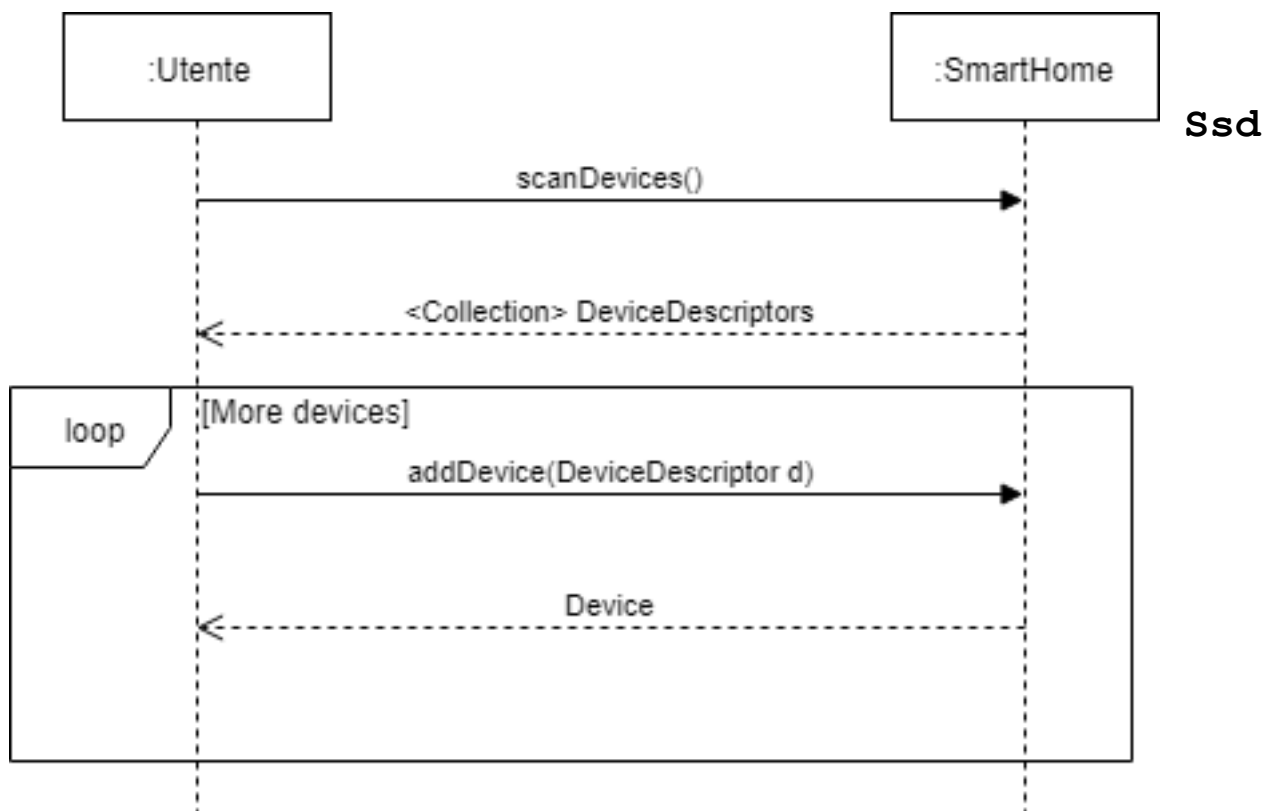
CASO D'USO C1:

Aggiungi un nuovo dispositivo

Attore primario: Utente della smart-home

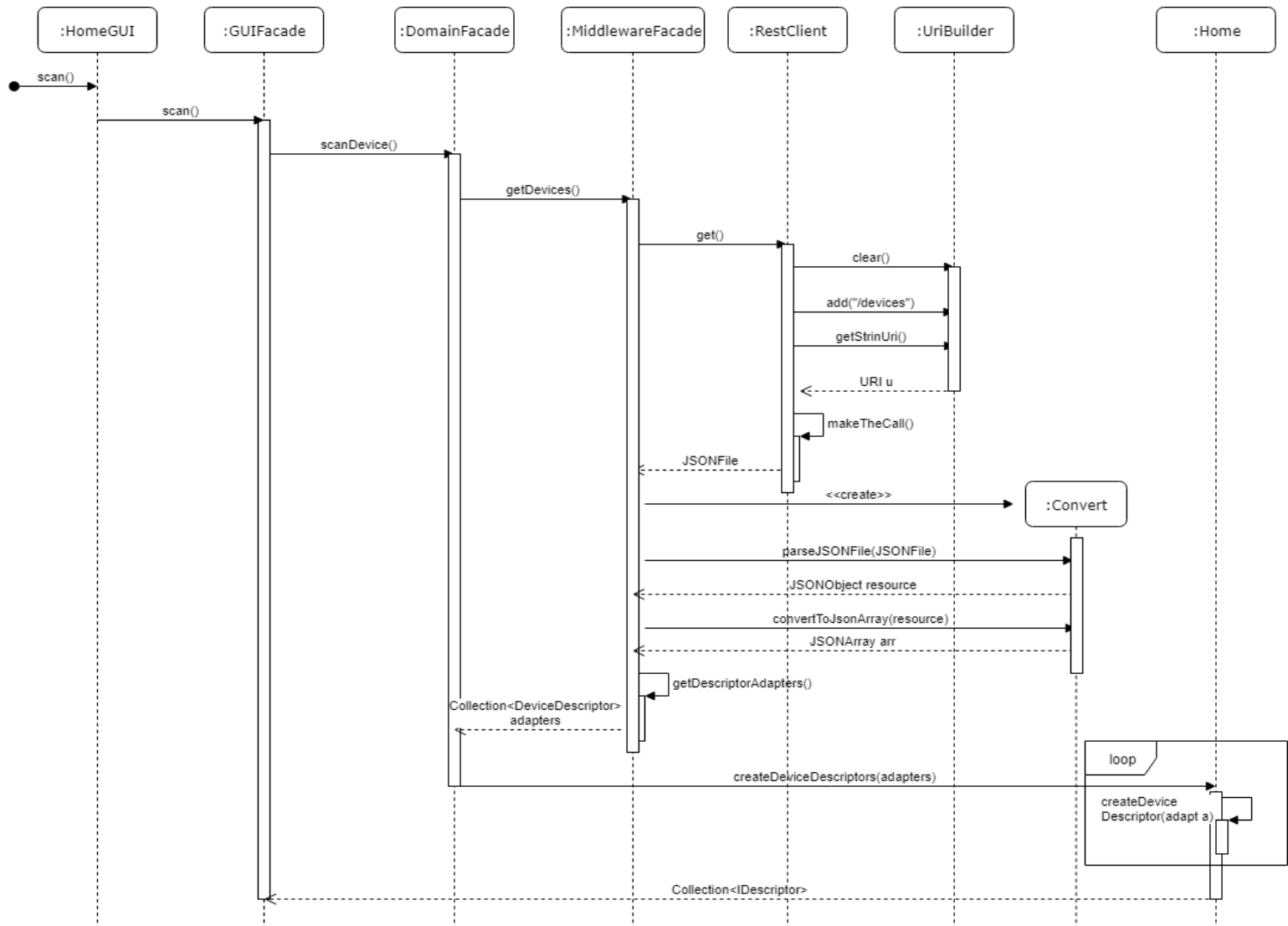
1. L'Utente scansiona i dispositivi collegati alla rete della sua casa.
2. L'Utente entra nella schermata di visualizzazione di tutti i dispositivi scansionati dal sistema.
3. L'Utente clicca sul pulsante "aggiungi" del dispositivo che è interessato ad aggiungere.
 - I passi 2 e 3 vengono ripetuti finchè ci sono dispositivi ancora da aggiungere.

Post-condizione: Il dispositivo è stato aggiunto alla smart-home.

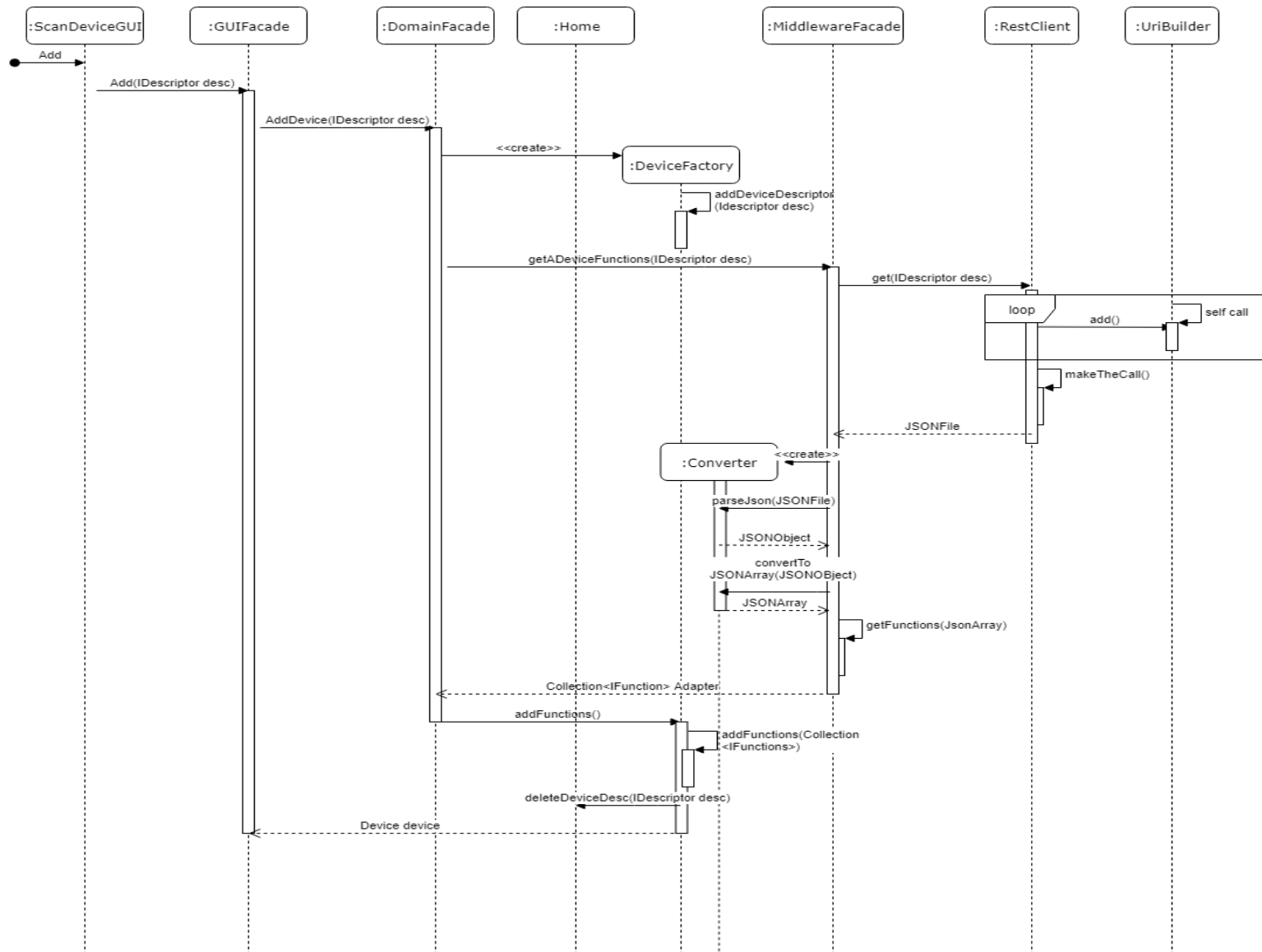


aggiungi dispositivo

Sd scan dispositivo



Sd aggiungi dispositivo



CASO D'USO C2:

Gestisci dispositivi in tempo reale

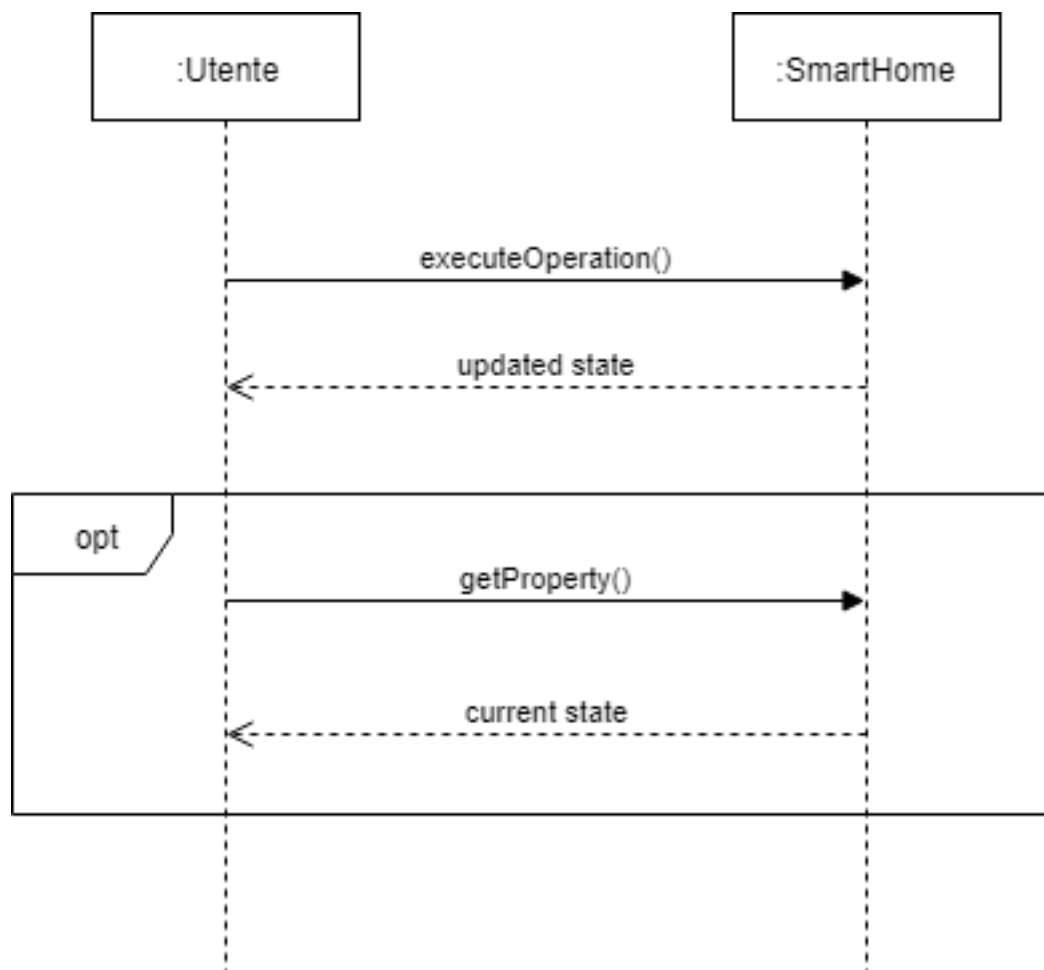
Pre-condizione: L'Utente è nella schermata della sua smart-home.

Attore primario: Utente della smart-home

1. L'Utente clicca sul dispositivo della sua smart-home con cui vuole interagire.
2. L'Utente preme sul pulsante corrispondente all'operazione da eseguire sul dispositivo.

Post-condizione: L'operazione è stata eseguita e lo stato del dispositivo correttamente aggiornato.

Ssd gestisci dispositivo in tempo reale



Modello di dominio

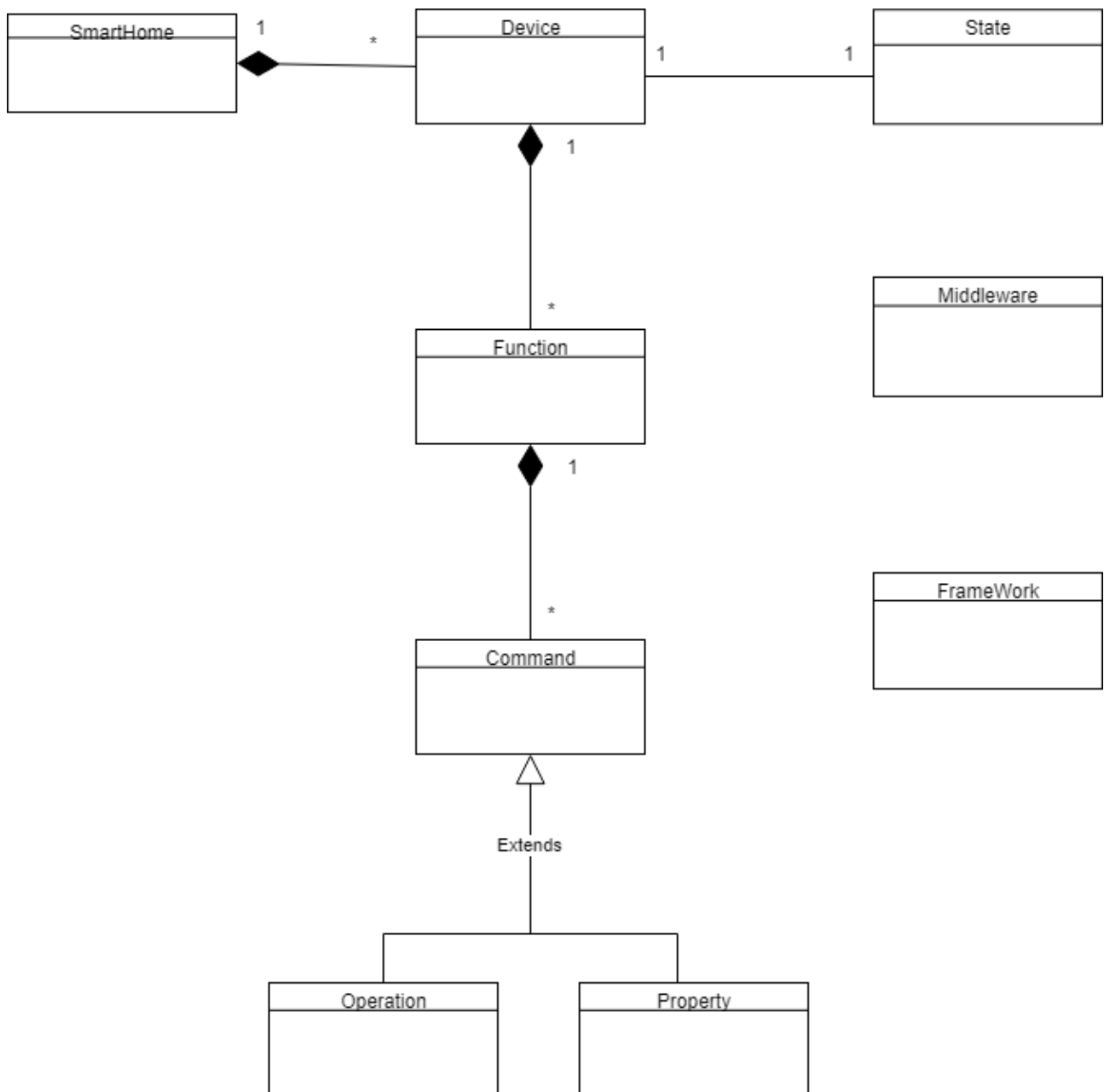
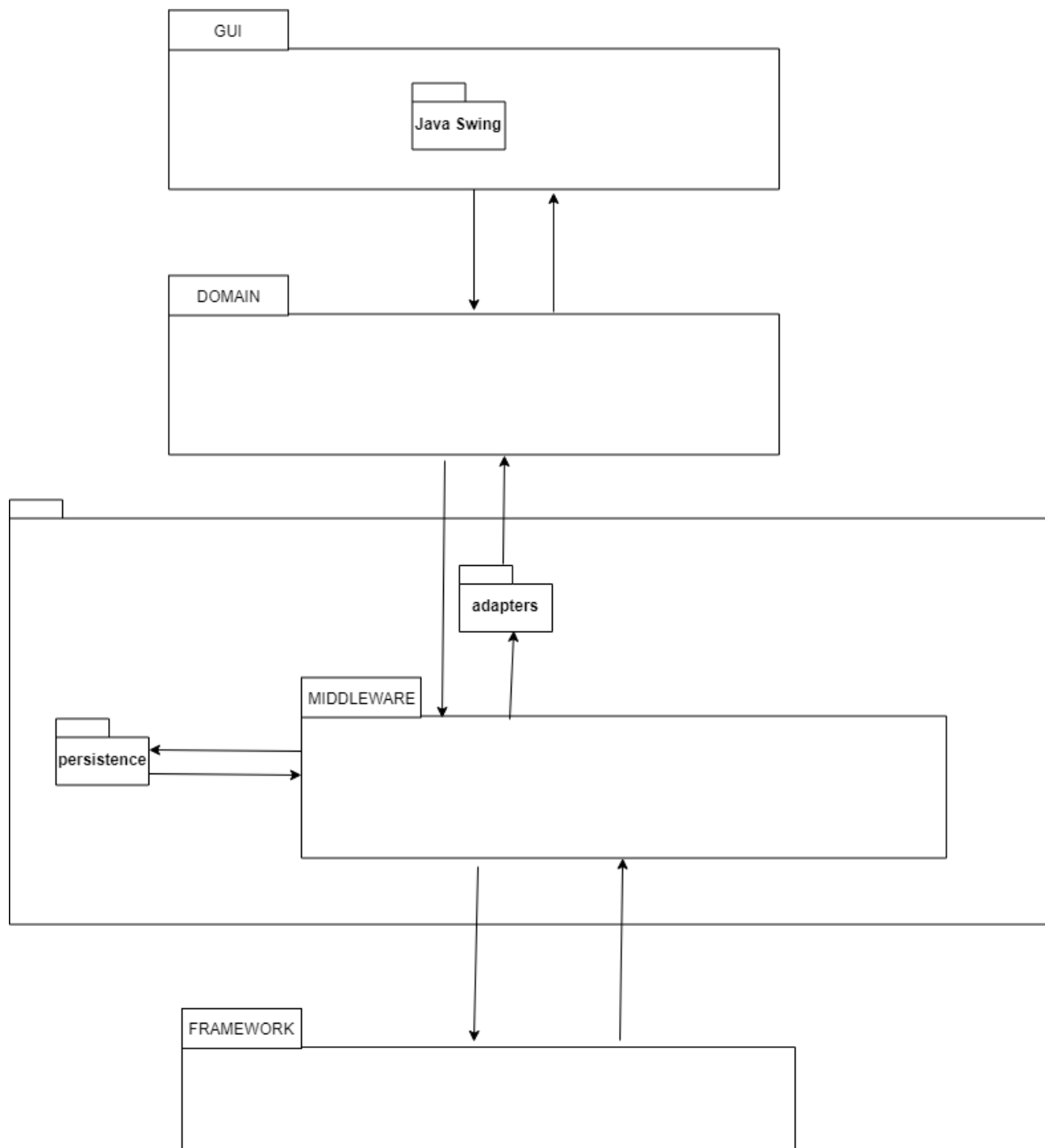
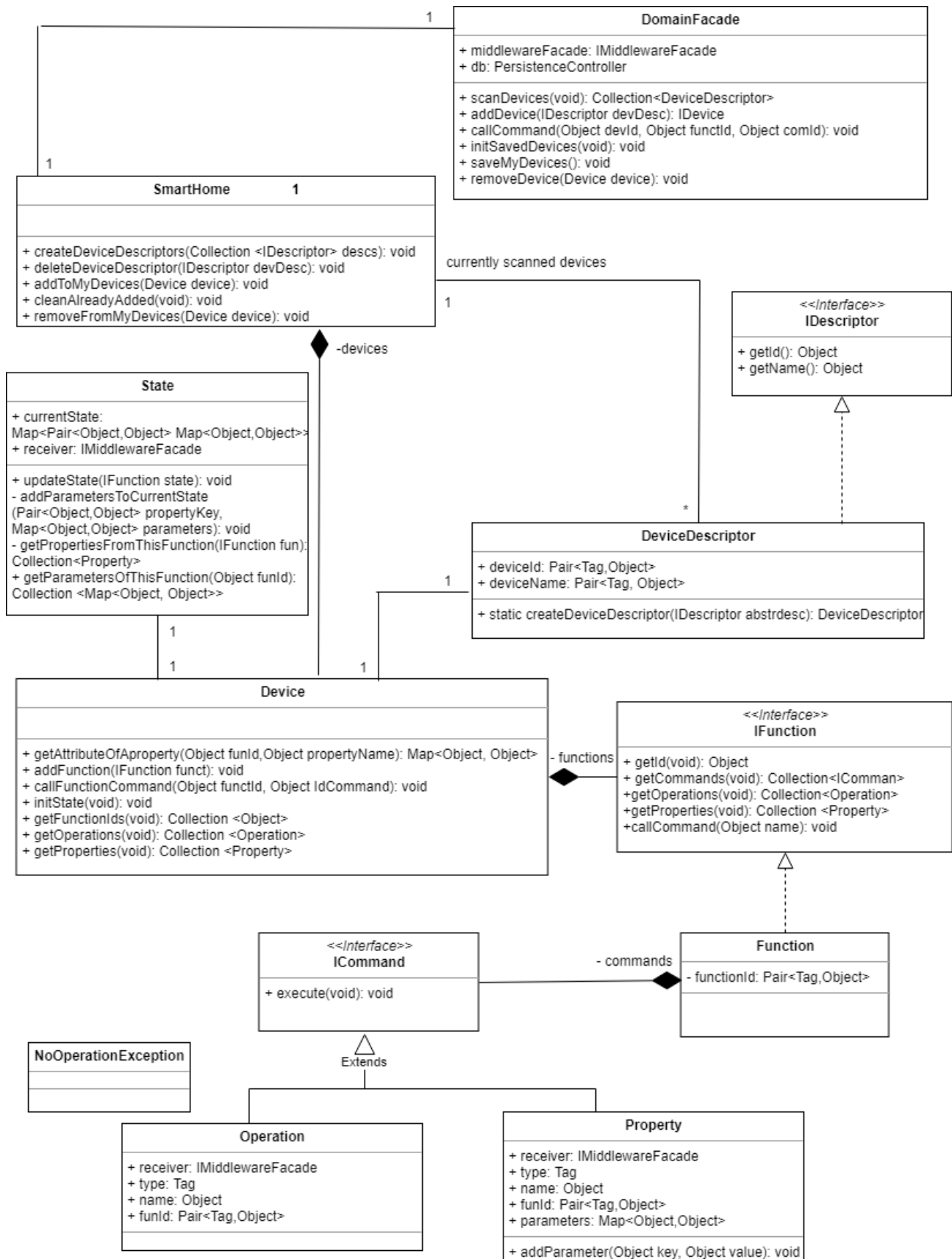


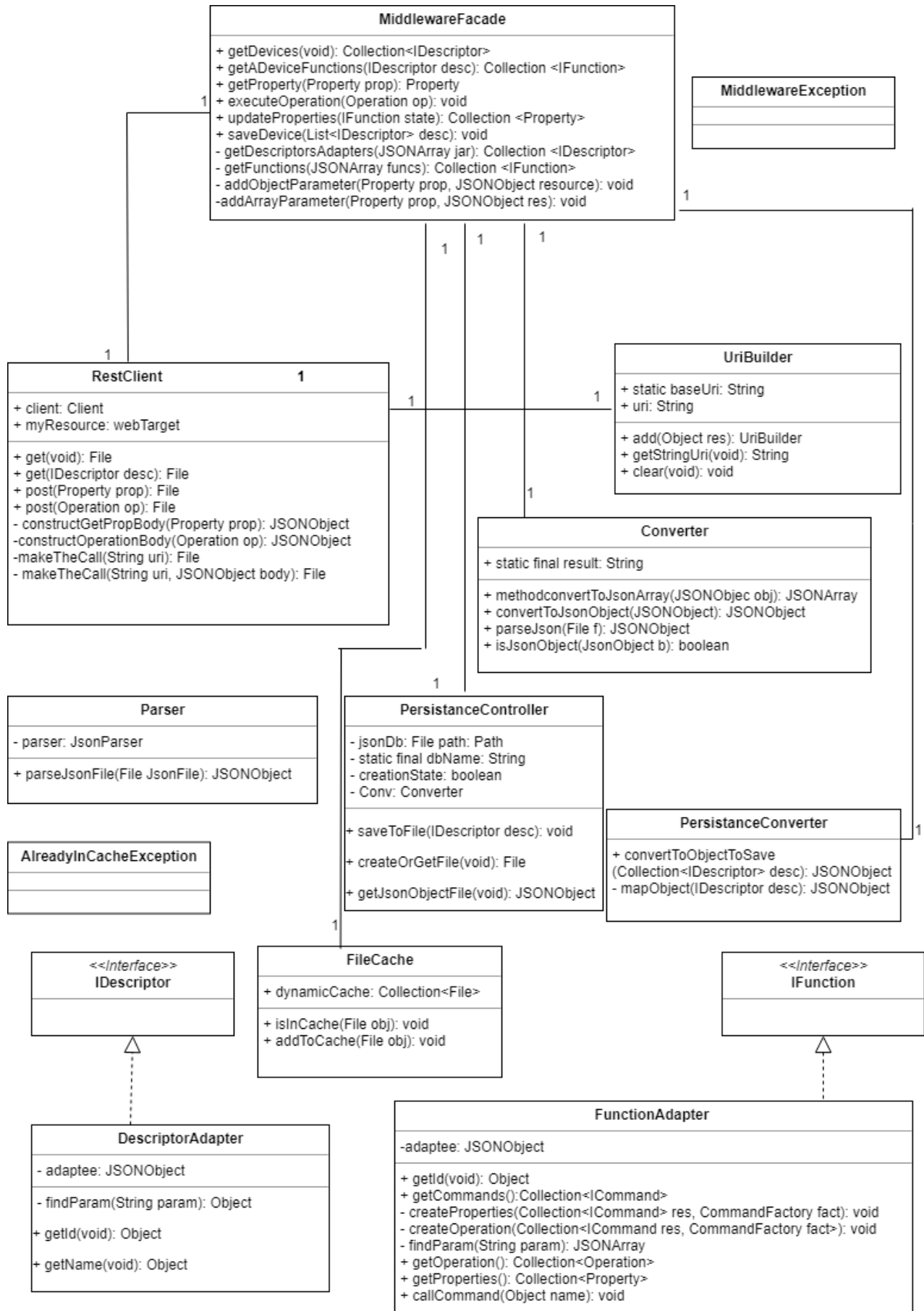
diagramma dell'architettura



Domínio



Middleware



Gui

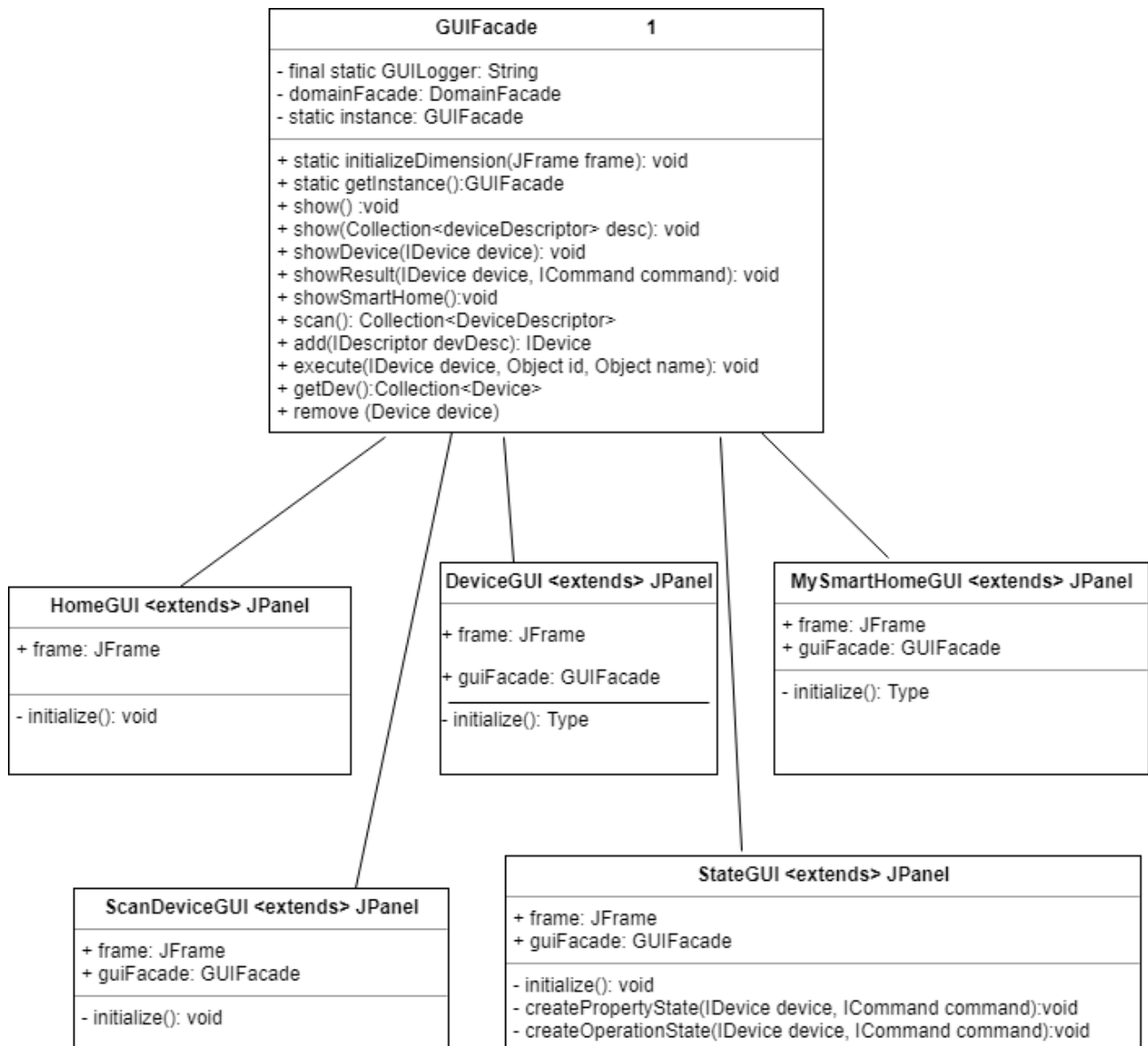


Diagramma delle attività per l'esecuzione di un'operazione

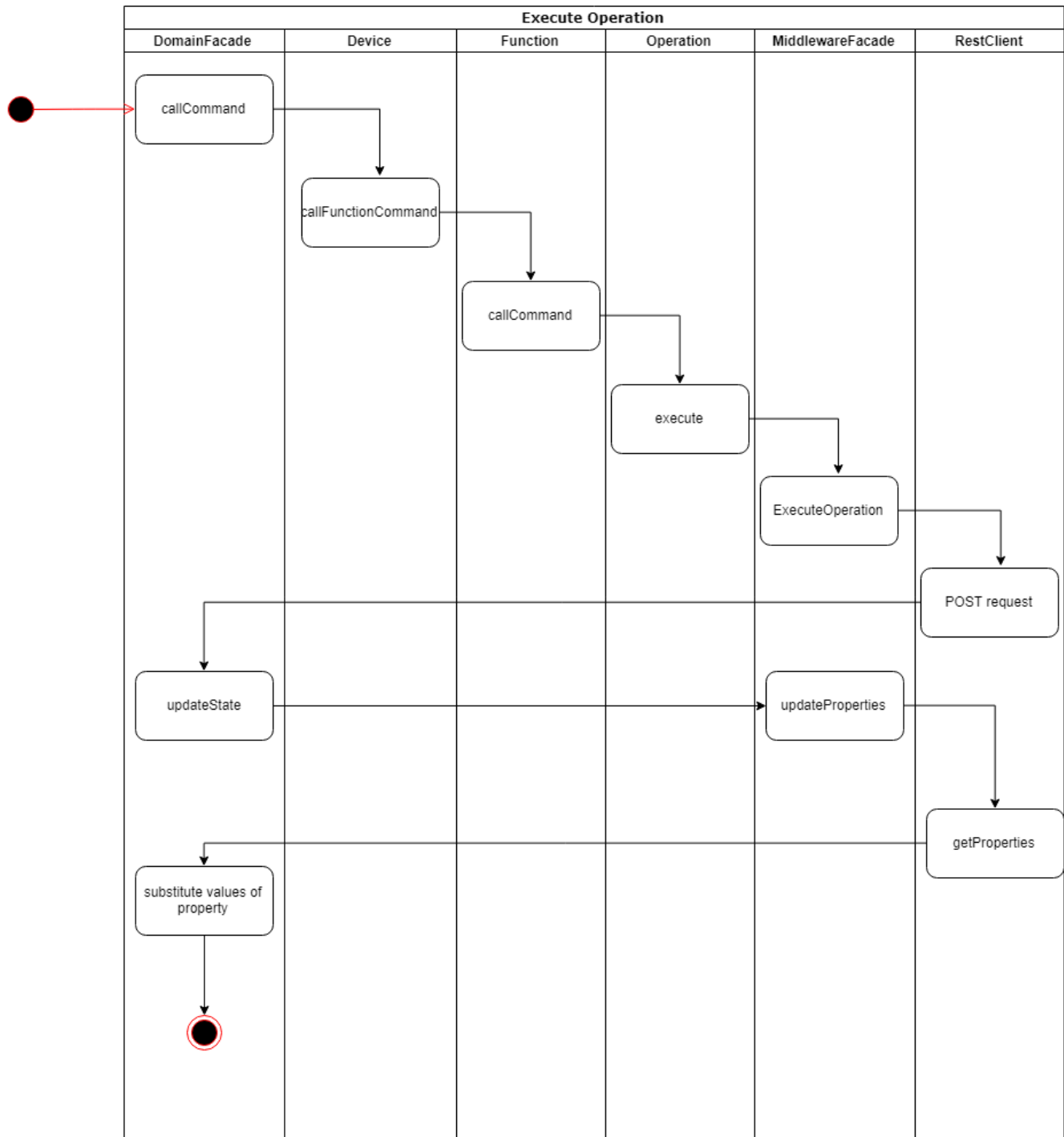
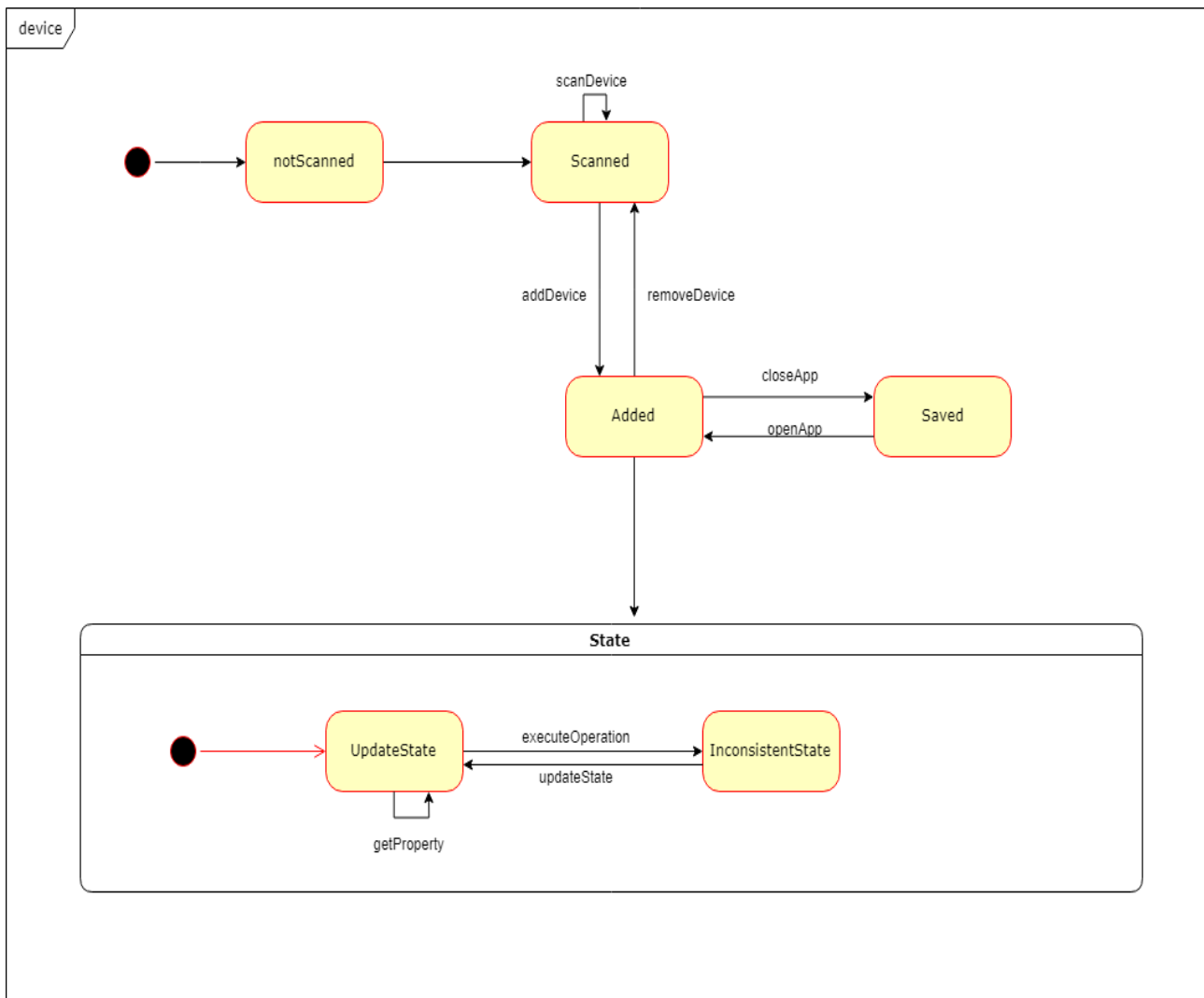


Diagramma stati di un dispositivo



Pattern Utilizzati...

COMMAND:

Il pattern command è stato utilizzato per fornire un'interfaccia univoca all'invoker delle porperties e delle operation ossia le functions. Questo ha semplificato la gestione delle varie categorie di operazioni eseguibili da un device.

FACTORY:

Abbiamo applicato due tipologie di factory, method & class, rispettivamente, una per la creazione di un deviceDesriptor e una per deviceFactory e commandFactory.

FACADE:

Facade è stato utilizzato per semplificare la comunicazione e la collaborazione tra sottosistemi (GuiFacade, DomainFacade, MiddlewareFacade).

ADAPTER:

E' stato utilizzato per fornire maggior disaccoppiamento tra le strutture utilizzate nel Middleware e le classi di dominio(JSONObject, IDescriptor, IFunction).

SINGLETON:

Singleton è stato utilizzato per essere certi di aver un'unica istanza di determinate classi (Dominio-SmartHome) , (Middleware-RestClient).