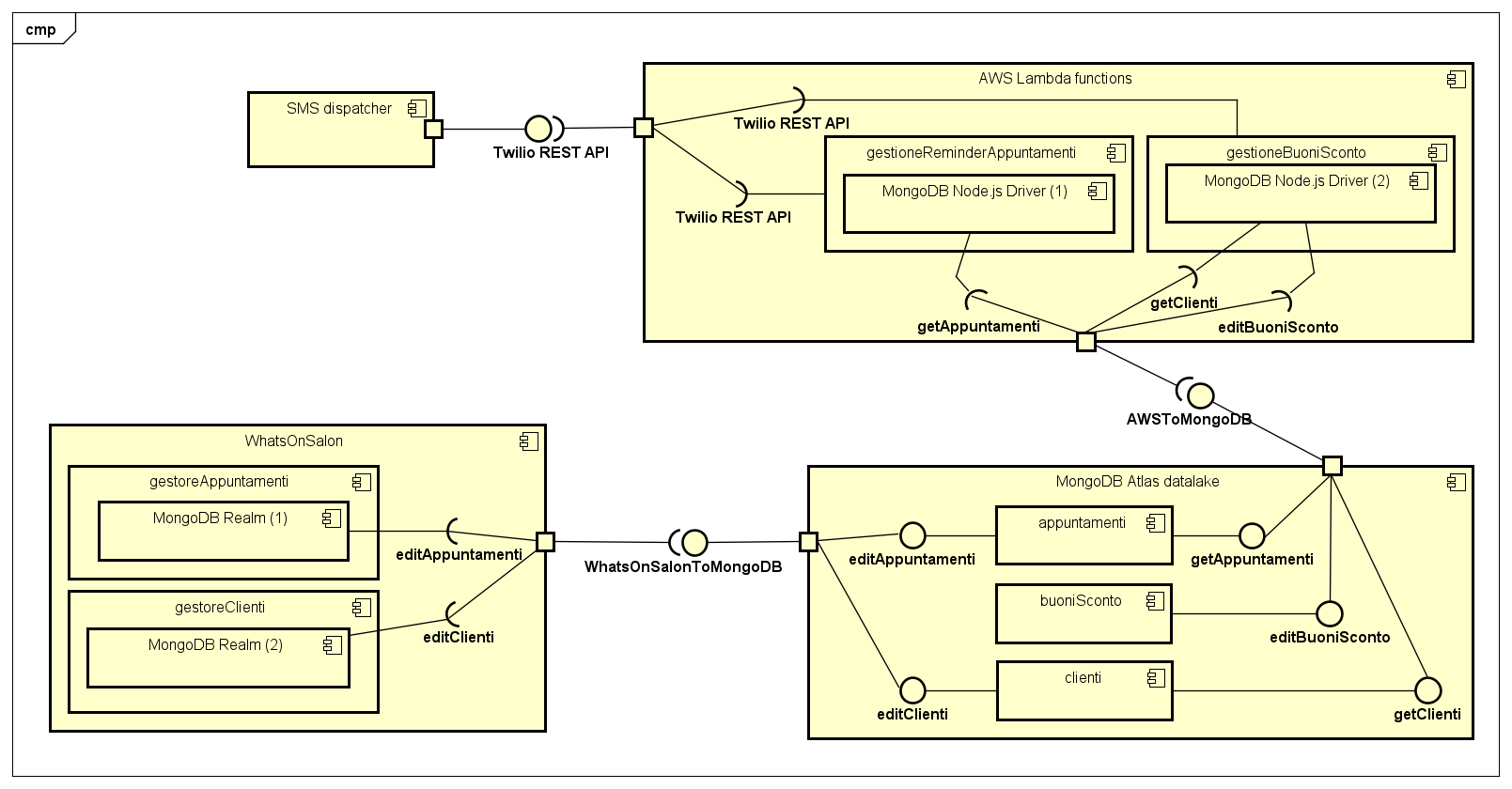
**ITERAZIONE 1**

**ARCHITETTURA SOFTWARE**

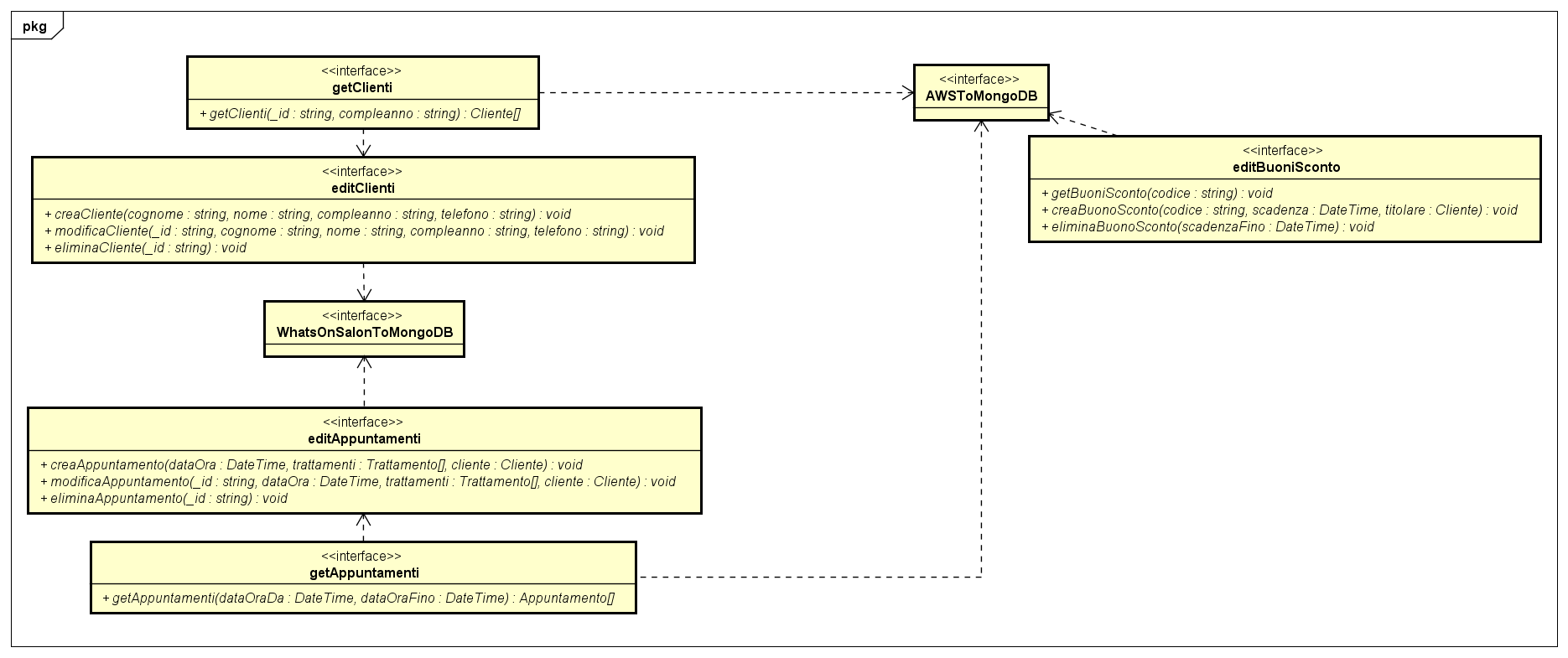
Basandosi sul lavoro svolto nell’iterazione precedente, è stato possibile cominciare a progettare l’architettura del software, riassunta in questo diagramma UML delle componenti.

Il diagramma ha lo scopo di rappresentare la struttura interna del sistema software modellato in termini delle sue componenti principali e delle relazioni fra esse. Per facilitarne la comprensione, in esso è stato fatto largo uso di porte e delegation connectors e dove era stato previsto l’utilizzo di software preesistente questo è stato aggiunto direttamente come subcomponent (es. MongoDB Realm) o interfaccia (es. Twilio REST API).



Le interfacce user-defined presenti nel component diagram sono esplicitate in quest’altro schema UML, che prende il nome di interface diagram.

Come si può osservare, esistono relazioni di ereditarietà tra le interfacce: ogni connessione tra component o subcomponent (visibile nel component model) è infatti realizzata da un’interfaccia che include dentro di sé tutte le interfacce richieste/offerte dalle parti in gioco.  
I metodi specificati all’interno di ogni interfaccia nello schema sono astrazioni di quella che diverrà la reale implementazione durante le fasi successive di sviluppo.



I tipi di dato user-defined presenti nell’interface diagram, che sono anche quelli che popoleranno il database, e le relazioni che intercorrono tra essi sono esplicitati in questo data diagram che, data la particolare architettura venuta a configurarsi in questo caso, è equivalente e funge anche da data transfer model.

