

*Polo territoriale di Como*

SCUOLA DI INGEGNERIA INDUSTRIALE E DELL’INFORMAZIONE

INGEGNERIA INFORMATICA

**Corso di Ingegneria del Software**

****

**AuthOK**

**PROGETTO DEL CORSO DI INGEGNERIA DEL SOFTWARE**

**Parte I – Requisiti**

Ferrario Stefano

Gumus Tayfun

Isella Paolo

Martinese Federico

Indice

[Introduzione 3](#_Toc496730317)

[Fasi di lavoro 3](#_Toc496730318)

[Analisi del problema 3](#_Toc496730319)

[Assunzioni 3](#_Toc496730320)

[Ricerca della soluzione 4](#_Toc496730321)

[Data Dictionary 5](#_Toc496730322)

[Goal Diagram 6](#_Toc496730323)

[SDM (Strategic Dependency Model) 6](#_Toc496730324)

[SRM – Client 7](#_Toc496730325)

[SRM – Server 8](#_Toc496730326)

[SRM – Risorsa 9](#_Toc496730327)

**Introduzione**

Il nostro progetto è “AuthOK”, un sistema di autenticazione basato sull’architettura client server.

Il progetto prevede la realizzazione di una libreria in Java per la comunicazione remota tra sistemi.

La libreria implementa in modo preciso la specifica del protocollo JSON-RPC e consente di implementare client e server JSON-RPC:

* Un server poterà ricevere richieste o notifiche e inviare risposte.
* Un client potrà inviare richieste o notifiche e ricevere risposte.

Sulla base di questa libreria viene realizzato un **sistema di autorizzazione centralizzato** per l’accesso a certe risorse. Il sistema centrale detiene l’elenco delle risorse accessibili. Tale elenco è aggiornabile creando, modificando o cancellando risorse. Analogamente il sistema tiene traccia delle autorizzazioni concesse. Deve essere possibile:

* Richiedere nuove autorizzazioni: fornendo i dati dell’utente, un livello e una scadenza, il sistema genera una chiave unica segreta per l’utente che dura fino alla scadenza. La chiave è associata ad un livello di autorizzazione: l’utente risulta autorizzato ad accedere a tutte le risorse associate a un livello inferiore o uguale alla propria autorizzazione.
* Revocare autorizzazioni (data la chiave)
* Chiedere un token di accesso a una risorsa: il sistema riceve la richiesta con il codice della risorsa e la chiave utente, verifica se la chiave è compatibile con il livello di risorsa richiesto e in caso positivo genera un token della durata di 24 ore per accedere alla risorsa.

Una risorsa può verificare sul sistema in ogni momento la validità di un token: dato token valido, il sistema restituisce per quanto tempo è ancora valido. Dato un token scaduto o inesistente, il sistema restituisce errore.

## **Fasi di lavoro**

Il procedimento che ci ha portato a sviluppare la nostra soluzione può essere suddiviso in due parti: una prima riguardante lo studio del problema e la scelta delle assunzioni necessarie, mentre la seconda orientata alla ricerca di una strategia e di una soluzione efficace.

## **Analisi del problema**

Il primo passo è stato effettuare una lettura approfondita delle specifiche che ci sono state fornite. Una volta terminata la comprensione di ogni singola richiesta, abbiamo cercato di identificare le entità fondamentali e gli obiettivi del nostro progetto.

## **Assunzioni**

Al fine di evitare l’introduzione di ulteriori complicazioni nel progetto e vista l’assenza di specifiche a riguardo, si assume possibile che un utente qualsiasi possa autorizzare “chiunque”.

## 

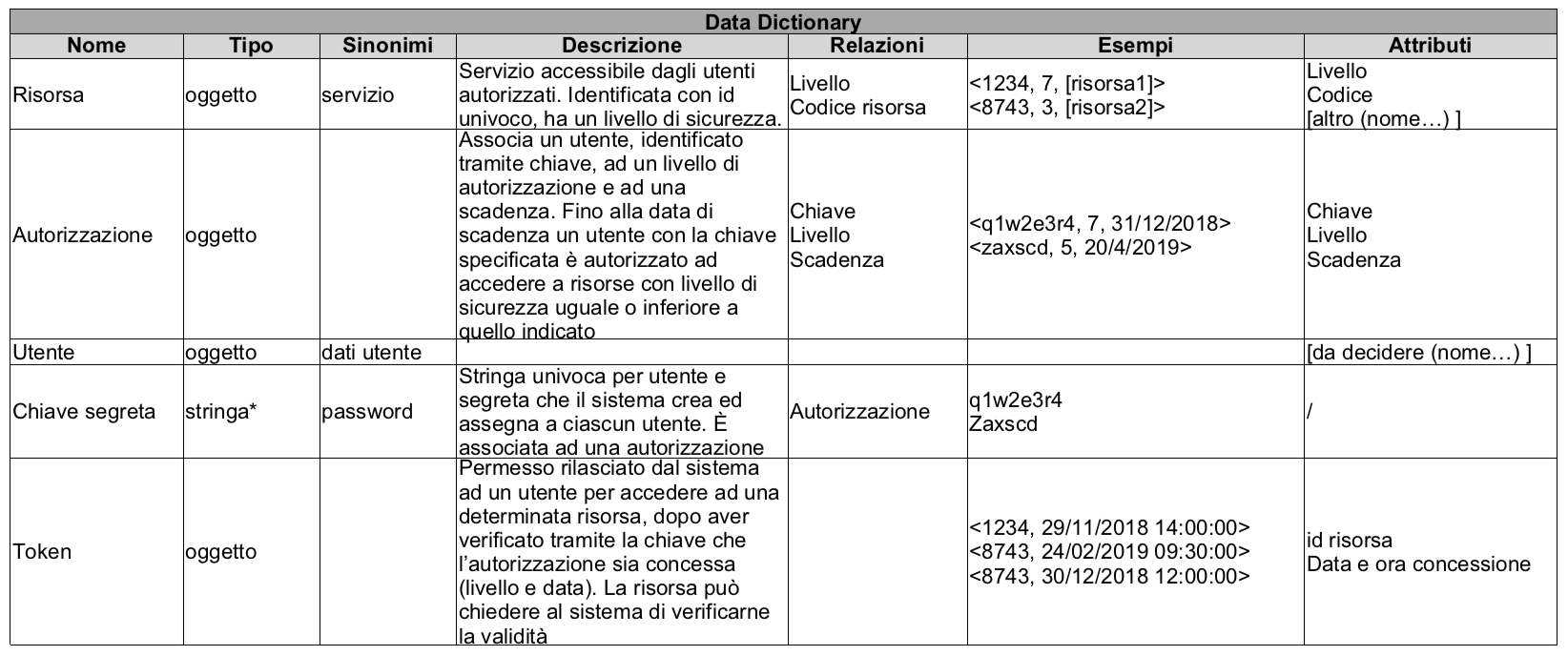
## **Ricerca della soluzione**

Al fine di effettuare l’analisi dei requisiti siamo scesi nel dettaglio utilizzando gli strumenti che ci sono stati forniti. In parallelo alla redazione di un Data Dictionary, nel quale vengono descritte in modo più approfondito le entità principali del progetto, abbiamo sviluppato l’SDM e l’SRM del Goal Diagram. Al loro interno abbiamo scelto attori, obiettivi, relazioni strategiche e, dopo aver esteso i vari agenti e ruoli, ne abbiamo definito le attività principali.

**Data Dictionary**

All’interno del Data Dictionary abbiamo analizzato alcuni elementi che a nostro giudizio costituiscono i concetti fondamentali per la realizzazione e il corretto funzionamento di un sistema di autenticazione: la **risorsa** a cui si vuole accedere, l’**autorizzazione** che il client richiede al fine di ottenere un accesso, l’**utente**, la **chiave segreta** necessaria per l’autenticazione, e il **token**.

Per ciascuno di essi vengono elencati una serie di attributi che, a nostro giudizio, ne permettono la loro descrizione in tutte le sfaccettature in modo da realizzare una visione più completa ed efficace di ogni entità.



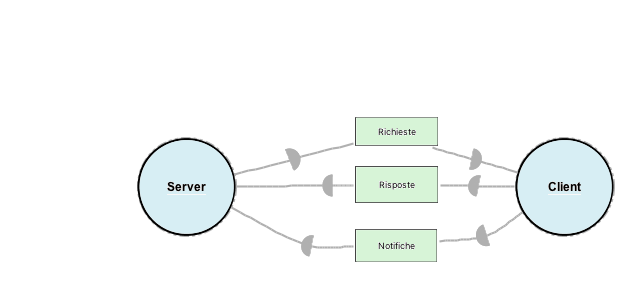
**Goal Diagram**

La stesura di un Goal Diagram ci permette di descrivere come, secondo la nostra visione, l’intero sistema debba essere strutturato.

Abbiamo disposto attori, relazioni, obiettivi e risorse con lo scopo di rappresentare il funzionamento ad alto livello della nostra applicazione.

**SDM (Strategic Dependency Model)**

L’SDM ci guida a una prima visione del progetto. Partiamo con una rappresentazione grafica del funzionamento della libreria java che vogliamo implementare. Tale libreria riproduce un’architettura client e server che infatti costituiscono i due attori nell’SDM.



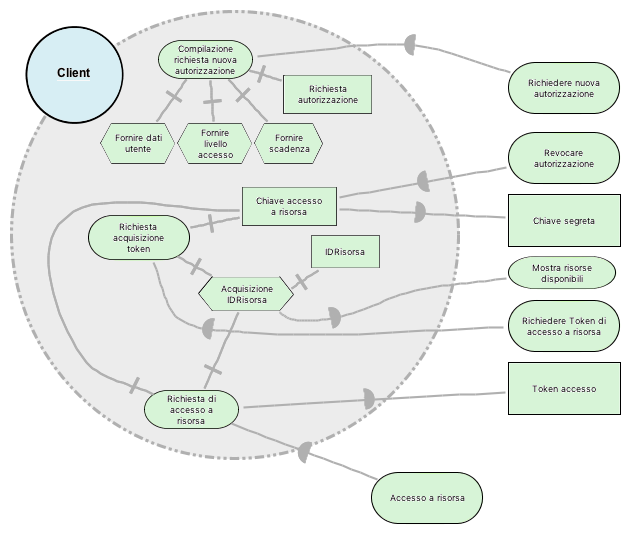
Come si può notare dal grafico, il server riceve richieste dal client che a sua volta riceve risposte dal server. Inoltre il server è in grado di inviare notifiche al client. Richieste, risposte e notifiche vengono identificate come risorse del sistema.

Possiamo ora, partendo dal modello client server, modellizzare la realizzazione del sistema di autenticazione. Otteniamo così il grafico SDM seguente:

Abbiamo scelto una serie di obiettivi per il client che in questo caso specifico può **richiedere nuove autorizzazioni** così come **revocarle** oppure **richiedere un token** per l’accesso a una risorsa. Il client potrà inoltre interagire con la Risorsa, terzo attore che abbiamo aggiunto per questo caso specifico del sistema di autorizzazioni, al fine di richiederne l’accesso. Il server in presenza di richieste da parte del client di una nuova autorizzazione potrà fornire una Chiave segreta oppure, qualora riceva una richiesta per un token di accesso, provvedere alla verifica e fornire un token di accesso. Abbiamo inoltre inserito tra gli obiettivi del server quello di mostrare l’elenco delle risorse disponibili. La risorsa inoltre in corrispondenza di una richiesta di accesso da parte del client potrà verificare la validità del token con il server.

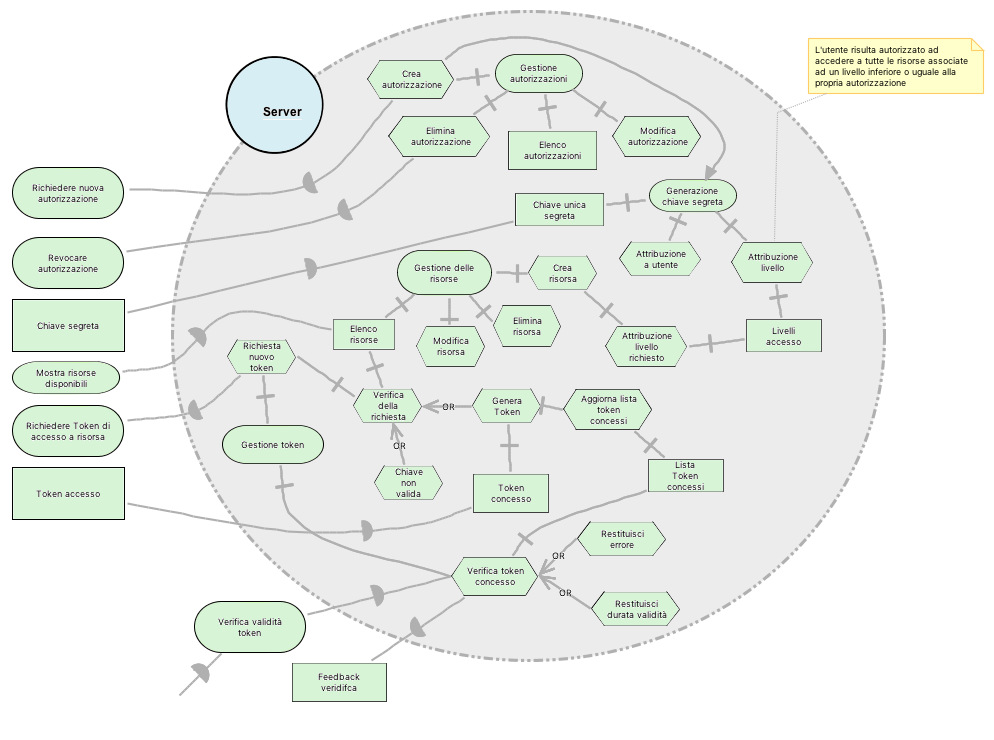
Dopo aver analizzato ogni singolo attore per soddisfare la necessità di rappresentare le motivazioni delle dipendenze in SDM e le dipendenze interne oltre che a quelle tra gli attori procediamo con la costruzione dell’**SRM (Strategic Rationale Model)** che riportiamo scomposto nei diversi attori.

## **SRM – Client**



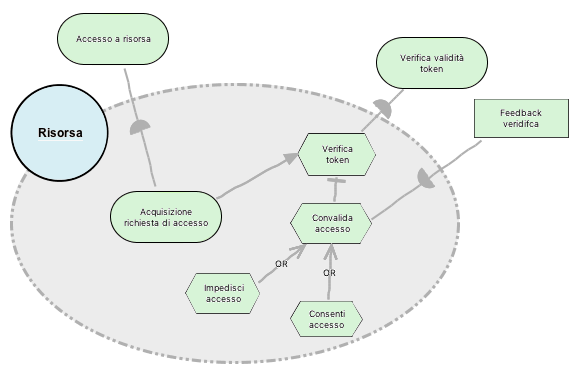
Il primo goal da soddisfare per il Client è “**Richiedere autorizzazione**”. Per fare ciò si dovrà procedere con l’esecuzione dei task che portano alla compilazione di una richiesta. Poiché nelle specifiche non viene precisato tale aspetto, ammettiamo che un utente qualsiasi possa autorizzare chiunque. Il client ha inoltre la possibilità di “**Revocare autorizzazione**” che identifichiamo come secondo goal. Per accedere ad una risorsa sarà necessario “**Ottenere un token**” (terzo goal), obiettivo raggiungibile dopo aver fornito una chiave segreta. Il client ha inoltre la possibilità di “Accedere a risorse” fornendo un token di cui potrà essere verificata la validità da parte della risorsa stessa.

## **SRM – Server**

****

Il Server **gestisce le risorse** mediante le funzioni di creazione, modifica e cancellazione. In modo del tutto analogo **gestisce le autorizzazioni**. La concessione di una nuova autorizzazione dipende dalla richiesta derivante dal Client. Tale task porta alla **generazione di una chiave segreta** che viene restituita al Client come risorsa. Ogni risorsa e ogni chiave segreta hanno un livello di accesso prestabilito così che l’utente risulta autorizzato ad accedere a tutte le risorse associate ad un livello inferiore o uguale a quello della propria autorizzazione. Il Server inoltre si occupa della **generazione di un token**. Per fare ciò procede alla verifica della richiesta proveniente dal Client ed una volta generato il token lo restituisce al Client come risorsa. Il Server potrà essere interpellato da una generica Risorsa qualora voglia verificare la validità di un token.

## **SRM – Risorsa**



Il Client ha la possibilità di richiedere l’accesso ad una Risorsa (risorsa generica). Questa ha la possibilità di verificare in qualunque momento la validità del token che il Client fornisce per ottenere l’accesso.

**Identificativo gruppo**: 10

**Progetto**: Tema A

**Componenti**:

845386 Ferrario Stefano [stefano6.ferrario@mail.polimi.it](mailto:stefano6.ferrario@mail.polimi.it)

843994 Gumus Tayfun [tayfun.gumus@mail.polimi.it](mailto:tayfun.gumus@mail.polimi.it)

854412 Isella Paolo [paolo.isella@mail.polimi.it](mailto:paolo.isella@mail.polimi.it)

845326 Martinese Federico [federico.martinese@mail.polimi.it](mailto:federico.martinese@mail.polimi.it)