



POLITECNICO
MILANO 1863

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
FASI DI LAVORO	3
ANALISI DEL PROBLEMA	3
ASSUNZIONI	3
RICERCA DELLA SOLUZIONE	4
DATA DICTIONARY	5
GOAL DIAGRAM	6
TEMA COMUNE (LIBRERIA JSON-RPC)	6
TEMA A - SDM (STRATEGIC DEPENDENCY MODEL)	6
SRM – CLIENT	8
SRM – AUTORIZZATORE	9
SRM – RISORSA	10

Introduzione

Il nostro progetto è “AuthOK”, un sistema di autenticazione basato su un'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 potrà 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 *qualsiasi utente* possa autorizzare *chiunque*. Il sistema di autorizzazione è quindi non sicuro da specifica.

Un'altra assunzione riguarda la gestione delle risorse, in particolare riguardo creazione, modifica e cancellazione delle stesse. Non essendo specificato *quale attore* agisca con l'obiettivo di

modificare l'elenco delle risorse (ad esempio creandone una nuova) si è deciso di *non* assegnare tale obiettivo al client. La gestione delle risorse rimane quindi un goal interno del sistema autorizzatore, che è in grado di gestire queste funzionalità, ma non è un *dependee* per alcun obiettivo esterno.

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. A causa della semplicità del diagramma I* riguardante il tema comune (libreria JSON-RPC) abbiamo ritenuto superfluo mostrare sia l'SDM sia l'SRM.

All'interno dei diagrammi 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.

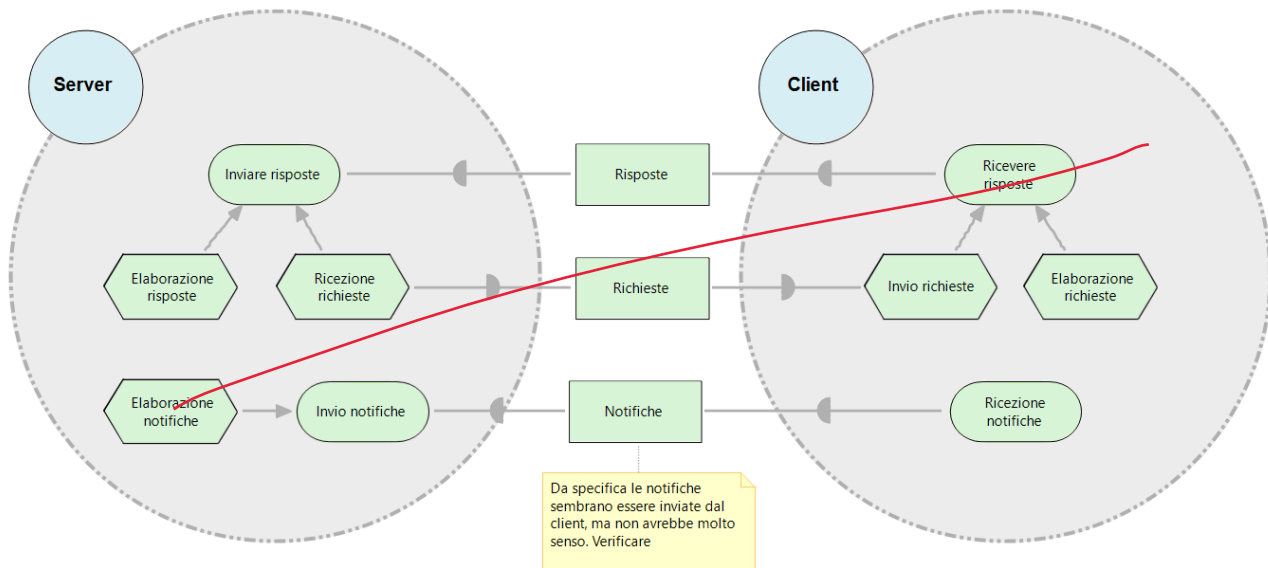
Data Dictionary						
Nome	Tipo	Sinonimi	Descrizione	Relazioni	Esempi	Attributi
Risorsa	oggetto	servizio	Servizio accessibile dagli utenti autorizzati. Identificata con id univoco, ha un livello di sicurezza.	Livello Codice risorsa	<1234, 7, [risorsa1]> <8743, 3, [risorsa2]>	Livello Codice [altro (nome...)]
Autorizzazione	oggetto		Associa un utente, identificato tramite chiave, ad un livello di autorizzazione e ad una scadenza. Fino alla data di scadenza un utente con la chiave specificata è autorizzato ad accedere a risorse con livello di sicurezza uguale o inferiore a quello indicato	Chiave Livello Scadenza	<q1w2e3r4, 7, 31/12/2018> <zaxscd, 5, 20/4/2019>	Chiave Livello Scadenza
Utente	oggetto	dati utente				[da decidere (nome...)]
Chiave segreta	stringa*	password	Stringa univoca per utente e segreta che il sistema crea ed assegna a ciascun utente. È associata ad una autorizzazione	Autorizzazione	q1w2e3r4 Zaxscd	/
Token	oggetto		Permesso rilasciato dal sistema ad un utente per accedere ad una determinata risorsa, dopo aver verificato tramite la chiave che l'autorizzazione sia concessa (livello e data). La risorsa può chiedere al sistema di verificarne la validità		<1234, 29/11/2018 14:00:00> <8743, 24/02/2019 09:30:00> <8743, 30/12/2018 12:00:00>	id risorsa Data e ora concessione

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. 

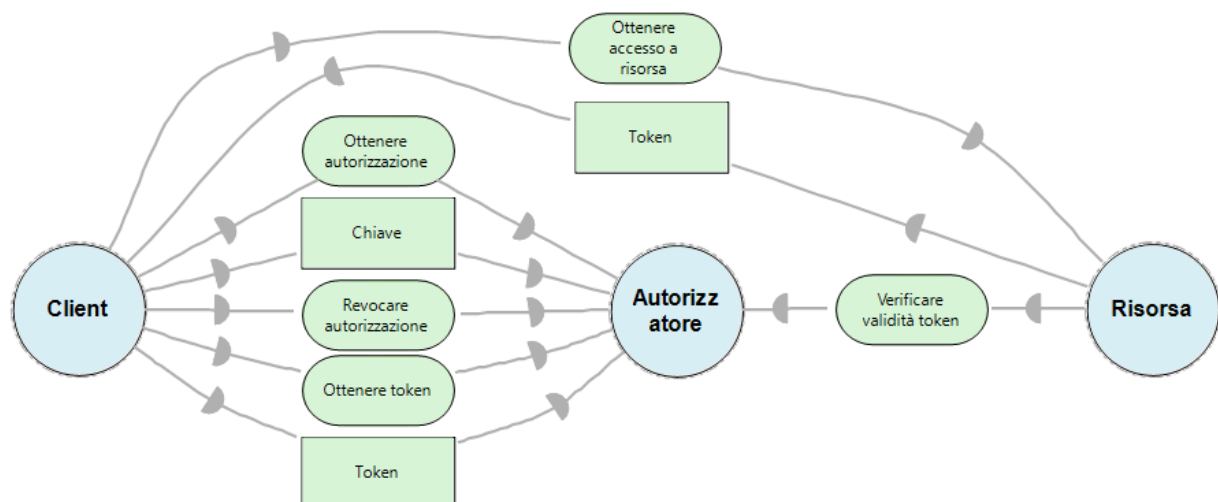
Tema Comune (libreria JSON-RPC)



Il diagramma ci guida a una prima visione del progetto. Partiamo con una rappresentazione grafica del funzionamento della libreria JSON che vogliamo implementare. Tale libreria gestisce il formato di comunicazione tra un client ed un server, i due attori del diagramma.

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. L'invio e la ricezione di richieste, risposte e notifiche (tutte identificate come risorse) sono gli obiettivi dei due attori e vengono raggiunti attraverso i task indicati. 

Tema A - SDM (Strategic Dependency Model)



Abbiamo identificato tre attori; denominati **Client**, **Autorizzatore** e **Risorsa**; ed una serie di obiettivi, nello specifico:

- Per il client
 - **ottenere autorizzazioni**
 - **revocare autorizzazioni**
 - **ottenere un token**
 - **ottenere l'accesso ad una risorsa**
- Per la risorsa
 - **verificare la validità di un token**

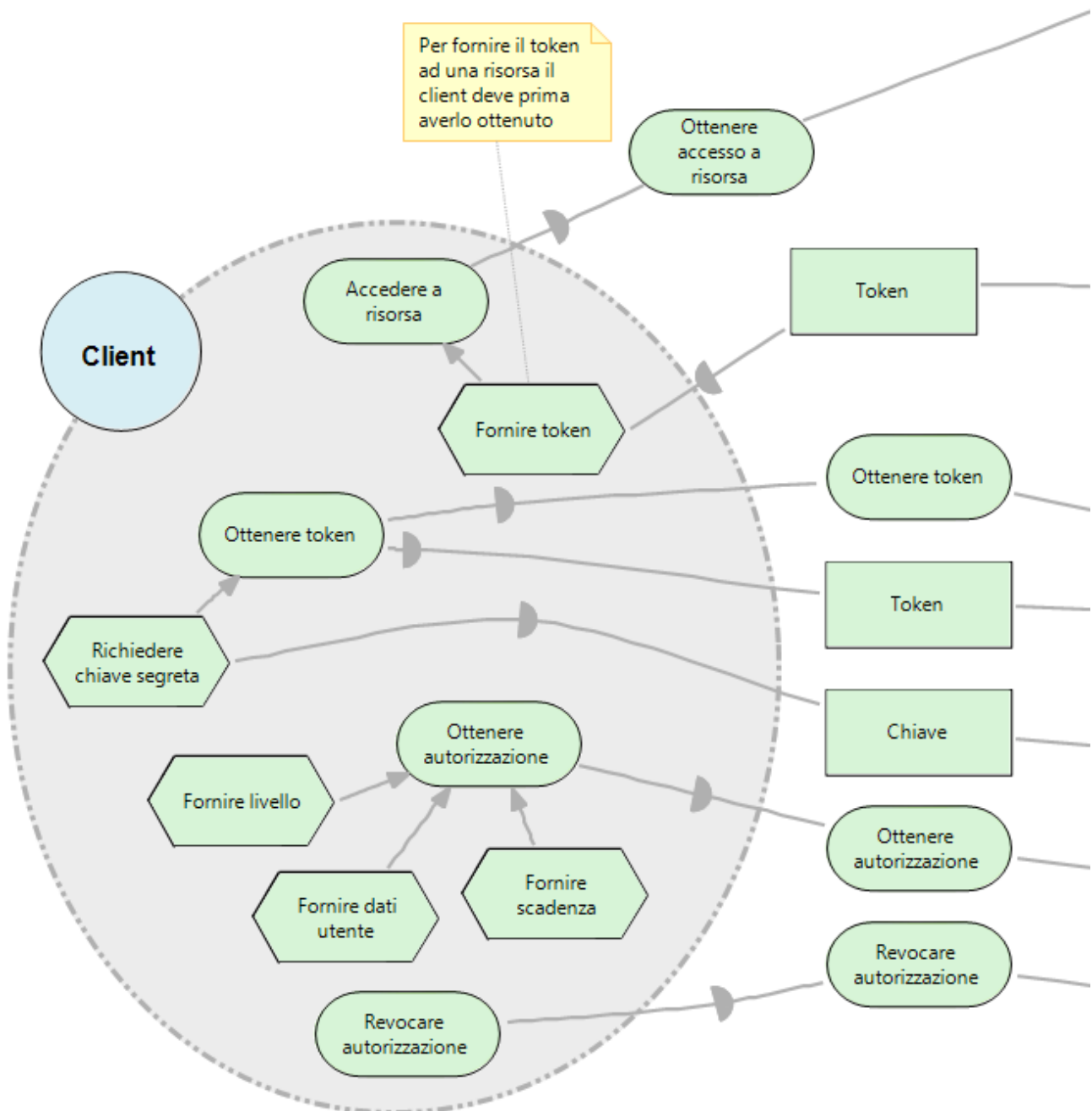
Si noti che l'attore Client del tema comune è un elemento differente dall'attore Client del tema A. Il primo è una delle due entità che comunicano attraverso un protocollo implementato dalla libreria, mentre il secondo è l'utilizzatore di un servizio di autorizzazione e di accesso a risorse.


Per raggiungere i primi tre obiettivi citati il Client dipende dall'Autorizzatore, mentre dipende dalla Risorsa per poterci accedere. Quest'ultima dipende dall'Autorizzatore per poter verificare la validità di un token.

L'autorizzatore invia le risorse *chiave segreta* e *token* al Client, mentre il Client invia il *token* alla risorsa.

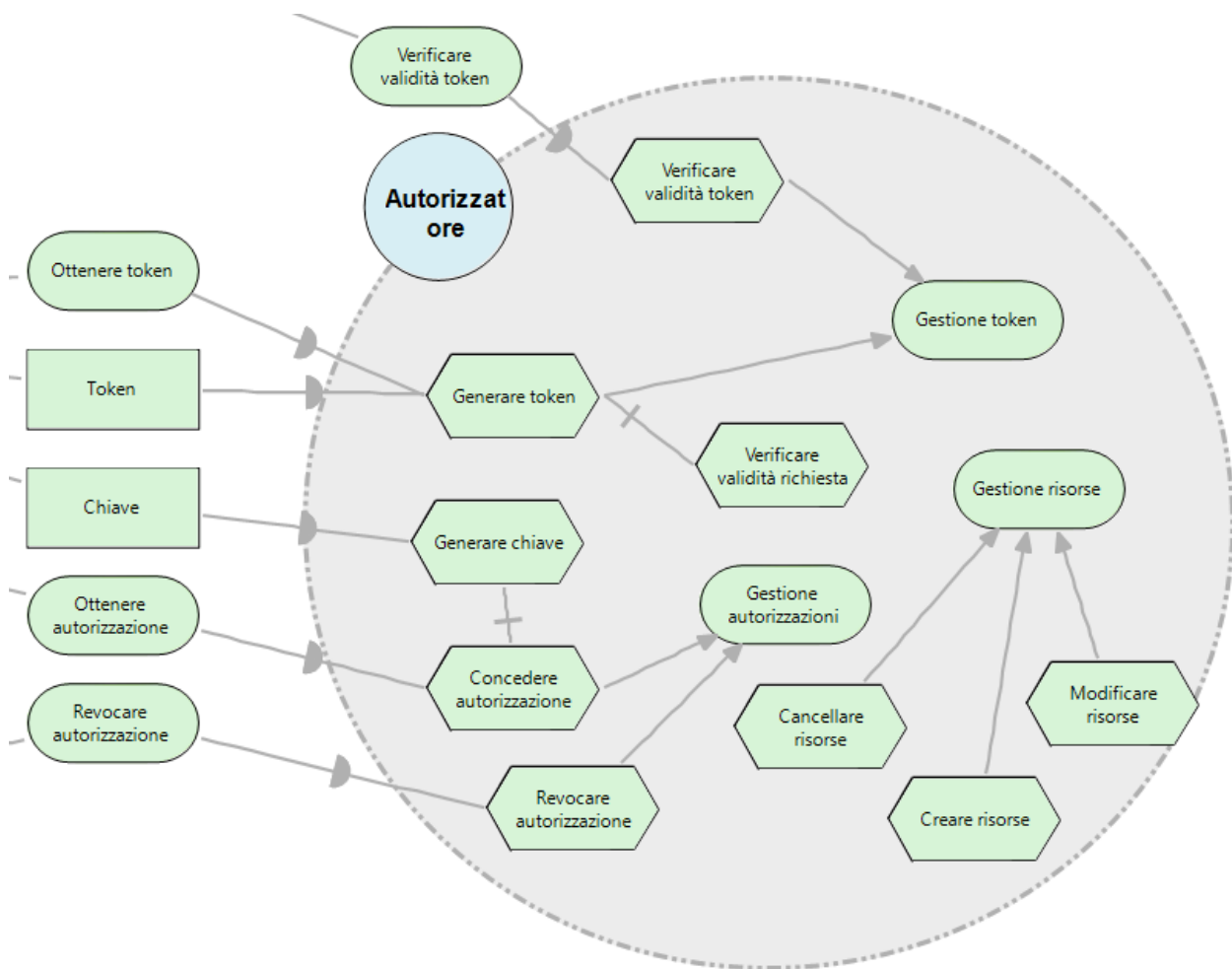
Dopo aver completato l'analisi dei goal e delle dipendenze strategiche si è passati alla realizzazione di uno schema **SRM (Strategic Rationale Model)**, che riportiamo scomposto nei diversi attori per comodità di lettura.

SRM – Client

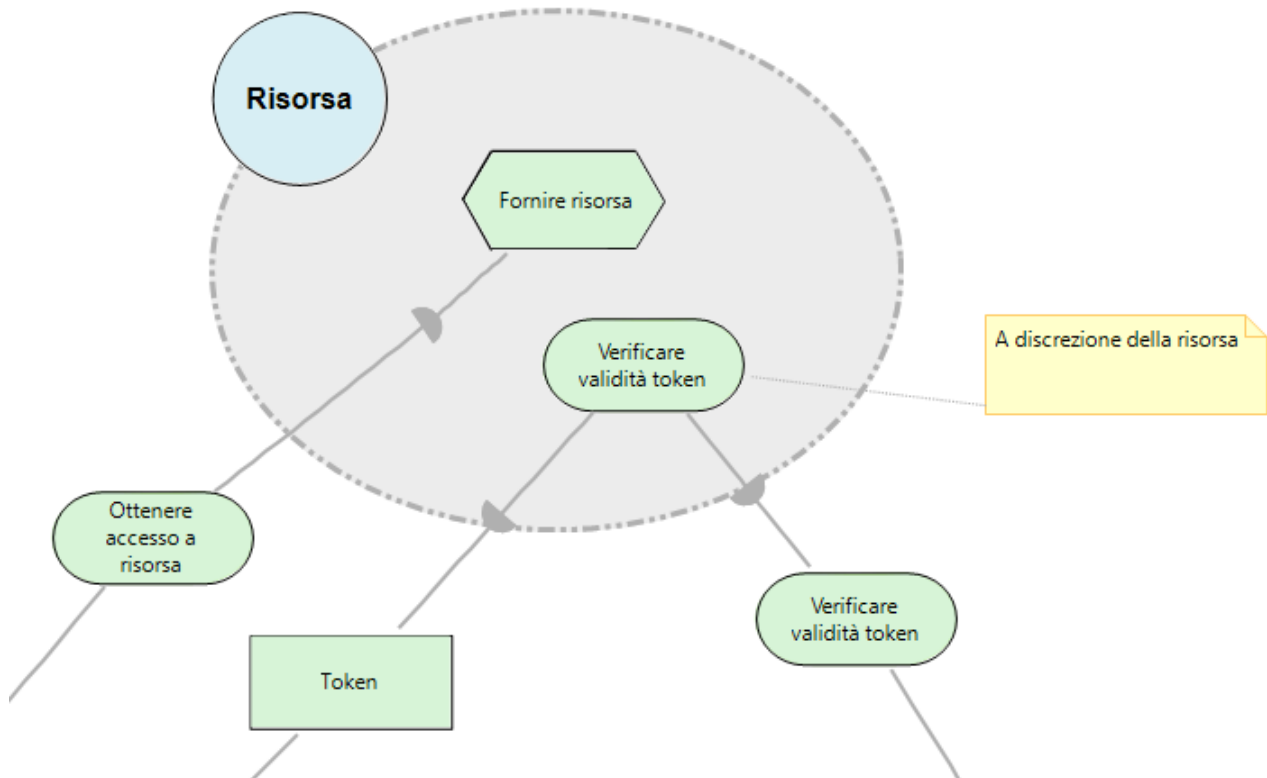


Il primo goal da soddisfare per il Client è **“Ottendere autorizzazione”**. Per fare ciò si dovrà  procedere con l’esecuzione dei task che portano alla compilazione di una richiesta, fornendo le informazioni necessarie. Poiché nelle specifiche non viene precisato tale aspetto, ammettiamo che un utente qualsiasi possa autorizzare chiunque. Il Client ha inoltre la possibilità di **“Revocare autorizzazione”**, identificato come ulteriore obiettivo. Per accedere ad una risorsa sarà necessario **“Ottendere un token”**, goal raggiungibile attraverso la richiesta di assegnamento di una chiave segreta. Il Client ha inoltre l’obiettivo di **“Accedere a risorsa”** fornendo un token di cui potrà essere verificata la validità da parte della Risorsa stessa.

SRM – Autorizzatore



L'Autorizzatore **gestisce le risorse** mediante le attività di creazione, modifica e cancellazione. In modo analogo **gestisce le autorizzazioni**, rendendo possibile il raggiungimento degli obiettivi del client che ne dipendono. La concessione di un'autorizzazione prevede un sottotask necessario alla **generazione di una chiave segreta** che viene restituita al Client come risorsa. L'Autorizzatore inoltre si occupa della **generazione di un token**, da cui dipende il relativo goal del client. Per fare ciò procede alla verifica della richiesta proveniente dal Client ed una volta generato il token lo restituisce al Client come risorsa. L'Autorizzatore inoltre permette alla Risorsa di soddisfare l'obiettivo di "verifica validità di un token" eseguendo il task necessario.

SRM – Risorsa

L'obiettivo del client di accedere ad una risorsa è soddisfatto da un task della risorsa, mentre l'obiettivo della risorsa relativo alla verifica della validità di un token dipende dalla risorsa token fornita dal client e dal task corrispondente dell'autorizzatore.

Identificativo gruppo: 10**Progetto:** Tema A**Componenti:**

845386	Ferrario Stefano	stefano6.ferrario@mail.polimi.it
843994	Gumus Tayfun	tayfun.gumus@mail.polimi.it
854412	Isella Paolo	paolo.isella@mail.polimi.it
845326	Martinese Federico	federico.martinese@mail.polimi.it