Prova finale di Ingegneria del Software  
Documento di progetto

# TEMA COMUNE

Realizzare una libreria in Java per la comunicazione remota tra sistemi.

La libreria deve implementare **in modo preciso** la specifica del protocollo JSON-RPC.

I dettagli del protocollo sono descritti qui: <http://www.jsonrpc.org/specification>

La libreria deve consentire di implementare client e server JSON-RPC:

* Un server deve poter ricevere richieste o notifiche e inviare risposte (ove queste siano previste dalla specifica).
* Un client deve poter inviare richieste o notifiche e ricevere risposte (ove queste siano previste dalla specifica).

Allo scopo di garantire la comunicazione remota, è necessario selezionare un “canale di comunicazione” su cui il protocollo è implementato.

Si suggerisce di rendere la libreria agnostica rispetto al canale di comunicazione, tramite un’apposita interfaccia e/o factory.

Come canale si suggerisce fortemente l’uso di un approccio a code, come ad esempio ZeroMQ.

Le risorse di ZeroMQ sono accessibili qui: <http://zeromq.org/> (e verranno introdotte a lezione).

Una volta realizzato il tema comune, è possibile utilizzare la libreria ottenuta per implementare una specifica applicazione. In particolare:

* Ogni gruppo può scegliere se realizzare l’applicazione A) o B)
* Inoltre, è possibile ottenere 3 punti di bonus aggiuntivo se due gruppi si coordinano per usare le due applicazioni insieme: vedere descrizione dell’applicazione I (Integration).

# Tema B) Broker

Si vuole realizzare un sistema che fa da broker per servizi online, in modo trasparente.

Il sistema detiene una lista di servizi online disponibili e consente di registrare nuovi servizi o cancellarne di esistenti.

Ogni servizio è descritto da un identificativo di servizio e un formato di input e output. Inoltre, sono disponibili una serie di metadati: proprietario, settore, parole chiave, titolo, descrizione, data di attivazione.

Il broker associa ad ogni servizio un metodo JSON-RPC che lo identifica in modo univoco e che ne permette l’invocazione trasparente (più client possono esporre servizi con lo stesso identificativo di servizio, ad esempio “Somma”).

E’ possibile richiedere al broker (anch’esso implementato tramite metodi JSON-RPC):

* Di restituire un elenco di servizi in base a una ricerca (una o più parola, che viene cercata in tutti i metadati). Il risultato è una lista di servizi con tutta la loro descrizione e il metodo JSON-RPC associato. Ad esempio se cerco “Somma” e ci sono più servizi con questo identificativo di servizio, il broker ritorna tutti questi servizi e il loro identificativo JSON-RPC (“SommaXX”)
* Di invocare, in modo trasparente, un servizio, tramite il metodo JSON-RPC associato (“SommaXX”).

Quando un client si disconnette dal broker i suoi servizi vengono automaticamente cancellati ed eventuali richieste pendenti completate con il codice di errore più opportuno.

Si può supporre che il broker e i servizi siano invocabili tramite quanto specificato nel tema comune.

# Sintesi dei Goal

## Goal tema comune:

* Obiettivi Client
  + Generate Request:

Al fine di comunicare con il Server è previsto dal protocollo JSON-RPC che il Client generi un oggetto JSON-RPC Request. Affinchè questo avvenga il Client dev’essere in grado di comporre, secondo lo standard, la richiesta da scambiare con il Server.

* + Handle Response:

Per completare la comunicazione fra Client e Server è necessario che il Client sia in grado di accettare, gestendo eventuali errori, le JSON-RPC Response che riceverà dal Server in seguito ad una Request effettuata.

* Obiettivi Server
  + Generate Response

Per fornire al Client il risultato dell’operazione che ha richiesto oppure per segnalare un errore di composizione nella JSON-RPC Request, il Server deve comporre e inoltrare al Client, rispettivamente, un oggetto JSON-RPC Response oppure un oggetto JSON-RPC Error.

* + Handle Request

È necessario che il Server sia in grado di ricevere e interpretare secondo lo standard JSON-RPC le Requests che riceverà.

## Goal tema B) Broker

* Obiettivi Client
  + Invoke a Service:

Invocare un servizio fra quelli registrati nel sistema (se si conosce il servizio non è necessario richiedere informazioni riguardo i servizi disponibili.). Il Client deve poter dare la possibilità all’utente di usufruire dei servizi disponibili nel Broker.

* + Request a list of desired services:

Richiedere una lista, basata su una ricerca, di servizi registrati nel sistema. Al fine di comunicare all’utente come utilizzare ciascun servizio è necessario che il Client sia in possesso della lista.

* Obiettivi Server
  + Publish a Service:

Pubblicare il servizio che il Server vuole fornire al sistema. Il Server deve poter essere raggiunto dal Broker qualora un Client volesse usufruire (attraverso il canale di comunicazione trasparente stabilito dal Broker) del suo servizio.

* + Delete the own Service:

Dev’essere sempre possibile per il Server rimuovere, attraverso un’opportuna comunicazione, il proprio servizio dall’elenco dei servizi del Broker.

* Obiettivi Broker
  + Make Services available to all connected Clients:

Per rendere disponibili i servizi registrati nel sistema il Broker ha il compito di aggiornare l’elenco dei servizi visibili ai Clients. In particolare l’operazione di aggiornamento consiste nell’aggiunta e rimozione, rispettivamente, di un nuovo servizio e di uno non più disponibile.

* + Create a transparent communication:

È compito del Broker fare da tramite fra il Client e il Server per l’intera durata della comunicazione, senza che il Client sia in possesso delle informazioni necessarie per dialogare con il Server.   
Le Request del Client vengono inoltrate (senza subire alcuna modifica) al Server il quale, allo stesso modo, attraverso il Broker invierà una Response al Client.

* + Manage efficiently services (Soft-Goal):

Affinchè il sistema possa usufruire dei servizi disponibili in modo efficiente è buona norma progettare il Broker in modo che la lista dei servizi disponibili non contenga servizi duplicati o servizi non più disponibili.

* Obiettivi comuni (Server-Broker)
  + Delete an unavailable Service:

Il Broker e il Server possono negoziare la cancellazione di un servizio: il Server può richiedere al Broker la cancellazione del suo servizio dall’elenco dei servizi, il Broker deve cancellare i servizi non più disponibili o raggiungibili nel sistema.