PROGETTO SISTEMI DISTRIBUITI E PERVASIVI

Alessio Quercia

Struttura del codice

- Quattro classi principali:
  - *ServerCloud*
  - *SensorsSimulator*
  - *Node*
  - *Analist*

ServerCloud

- Processo che lancia e gestisce il Server Cloud
- Il Server Cloud mette a disposizione servizi RESTful rispettivamente agli Analisti, ai Nodi e ai Sensori tramite le seguenti classi:
  - *AnalistsServices*
  - *NodesServices*
  - *SensorsServices*

SensorsSimulator

- Prende come input da tastiera il numero di sensori da creare e l'indirizzo del Server Cloud
- Crea n sensori, ognuno dei quali si connette al ServerCloud e inizia ad inviare misurazioni

Node

- Prende come input da tastiera l'ID da dare al nodo, il numero di porta per interfacciarsi con gli altri Nodi, il numero di porta per interfacciarsi con i sensori e l'indirizzo del Server Cloud
- Se c'è una posizione valida nella griglia rappresentante città, viene creato e aggiunto un nuovo nodo in tale posizione. Se il nodo viene aggiunto nella griglia, il Server Cloud gli invia in risposta la lista dei nodi presenti nella griglia.
- Una volta inserito nella griglia:
  - *Se è l'unico nodo, allora diventa il Nodo Coordinatore*
  - *Altrimenti chiede al nodo successivo nella struttura ad anello chi è il Coordinatore*

Node

- Se il nodo scopre di essere il Coordinatore:
  - *Lancia un thread NodeClient per interfacciarsi con il ServerCloud e comunicare con esso (inviandogli statistiche globali e locali) tramite i servizi REST offerti dal Server (nella classe NodesServices)*
- Ogni nodo, compreso il Coordinatore, lancia:
  - *Un thread NodeServerToNodes per gestire la comunicazione con gli altri nodi nella rete. Ogni nodo ha un buffer di messaggi, che vengono gestiti da un server (una pool di thread) lanciato da questo thread*
  - *Un thread NodeServerToSensors per gestire la comunicazione con i sensori. Ogni nodo ha un buffer di misurazioni, che vengono gestite da un server (una pool di thread) lanciato da questo thread*

Analist

- Prende come input da tastiera l’indirizzo del Server Cloud e si interfaccia ad esso
- Un analista può comunicare con il Server Cloud tramite i servizi REST offerti da esso (nella classe AnalistsServices) ed effettuare analisi sulle statistiche inviate al Server

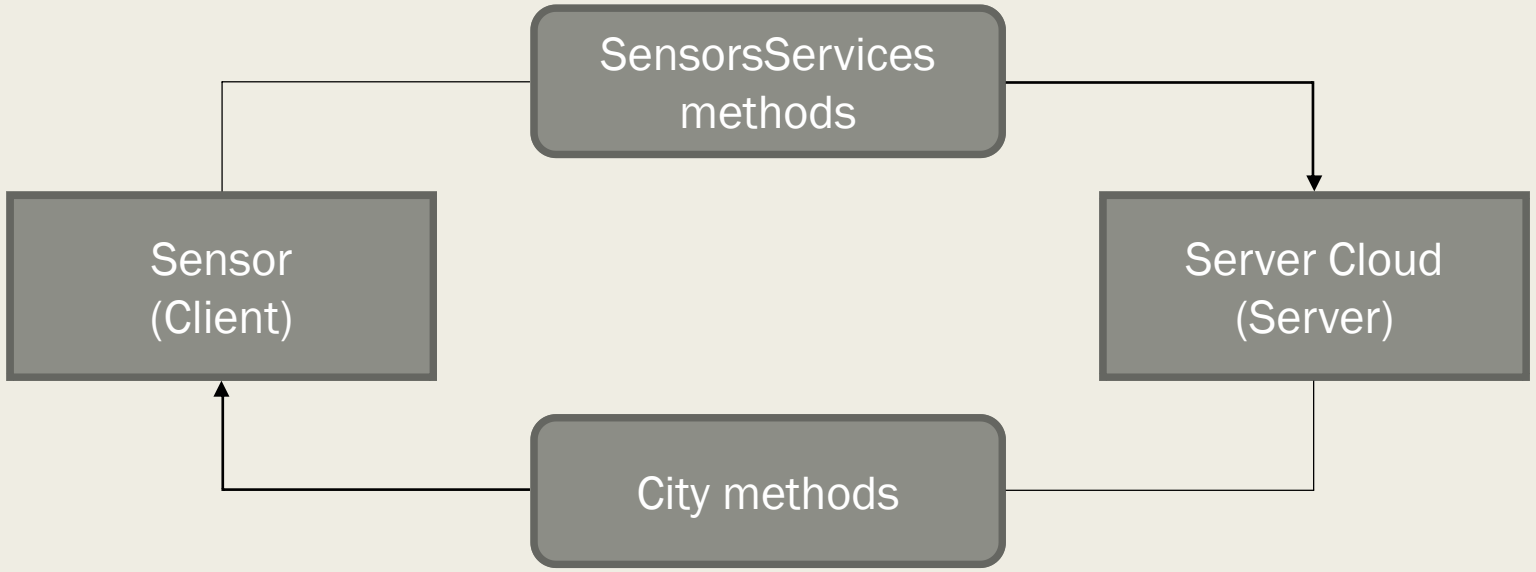
Comunicazione

- Comunicazione con il Server Cloud:
  - *Analist – Server Cloud*
  - *Sensor – Server Cloud*
  - *Node (Coordinator) – Server Cloud*
- Comunicazione con i Nodi:
  - *Sensor – Node*
  - *Node - Node*
  - *Node – Node (Coordinator)*
  - *Node (Coordinator) – Node*

Comunicazione Analist – Server Cloud

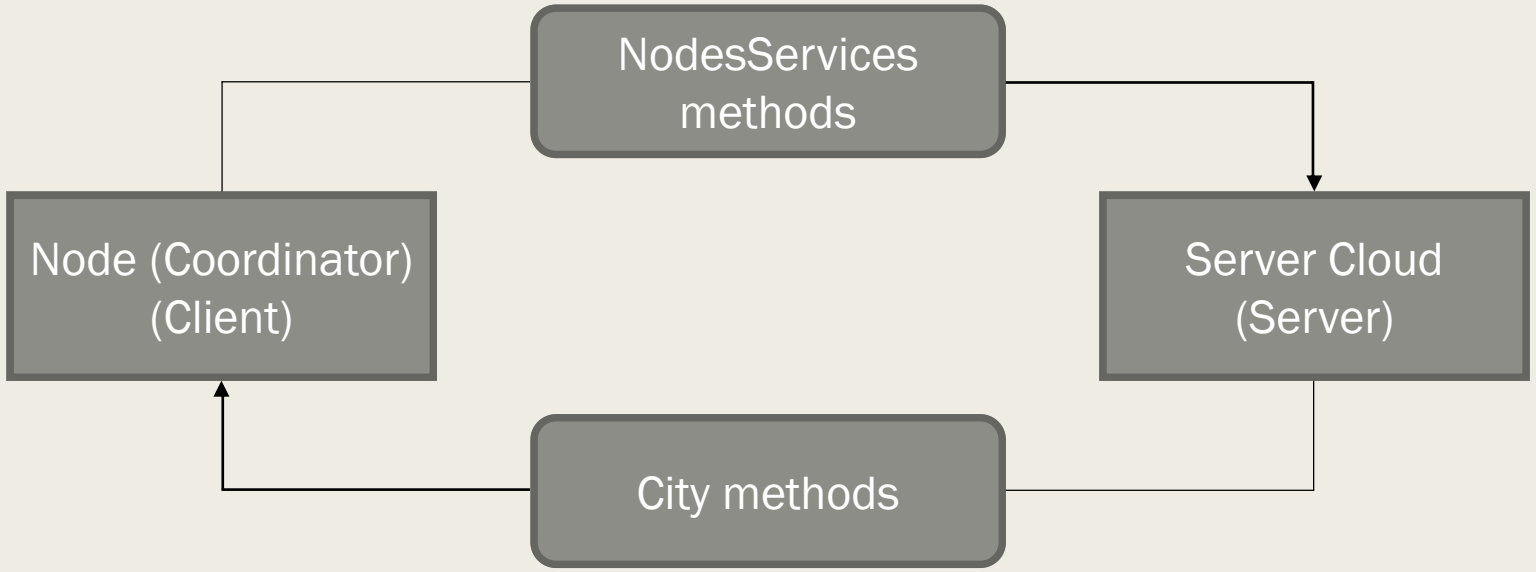
- Comunicazione tramite HTTP e servizi REST

Comunicazione Sensor – Server Cloud



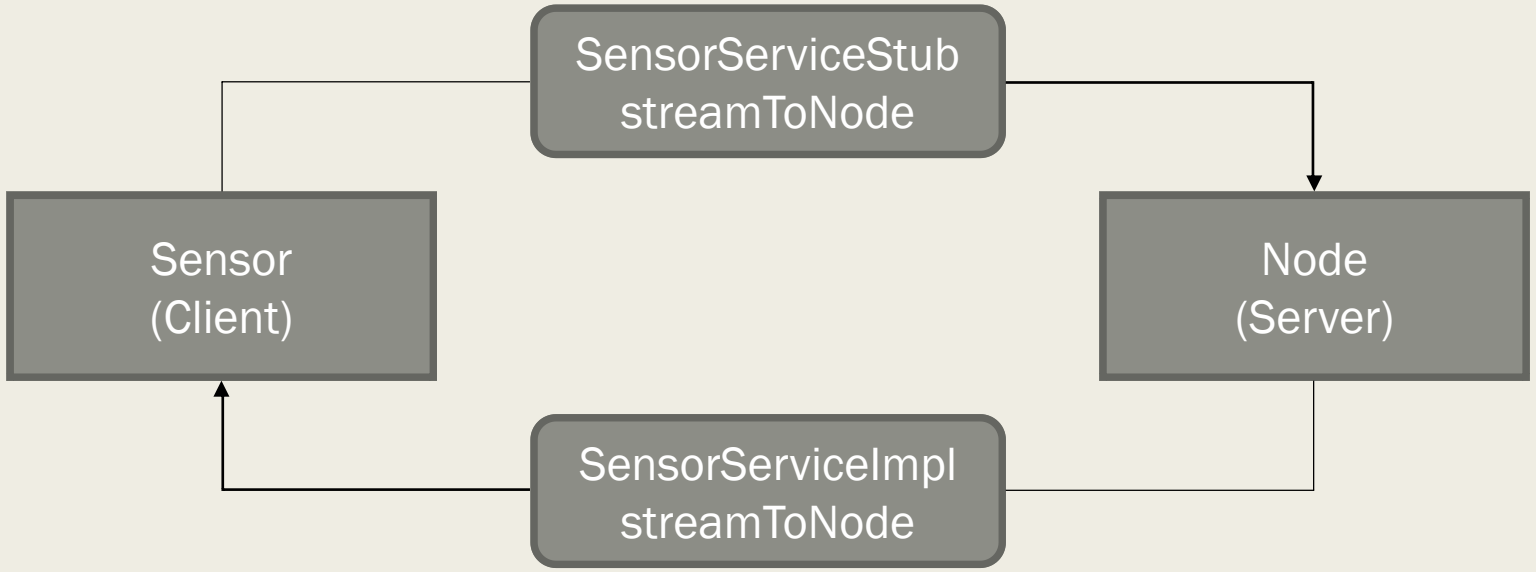
- Comunicazione tramite HTTP e servizi REST

Comunicazione Node – Server Cloud



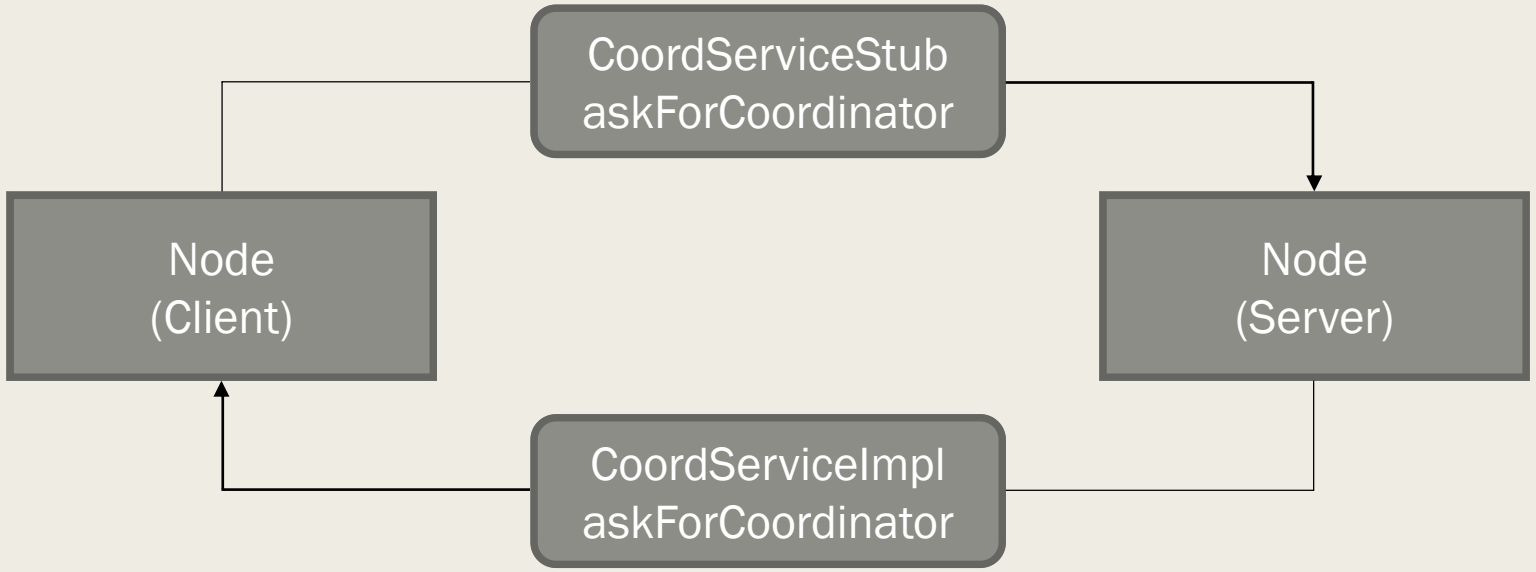
- Comunicazione tramite HTTP e servizi REST

Comunicazione Sensor – Node



- Comunicazione tramite gRPC, utilizzando una nonblocking stub:
  - *Il sensore client chiama streamToNode passandogli una MeasurementRequest*
  - *Il nodo server effettua la chiamata e risponde con una MeasurementResponse*

Comunicazione Node – Node

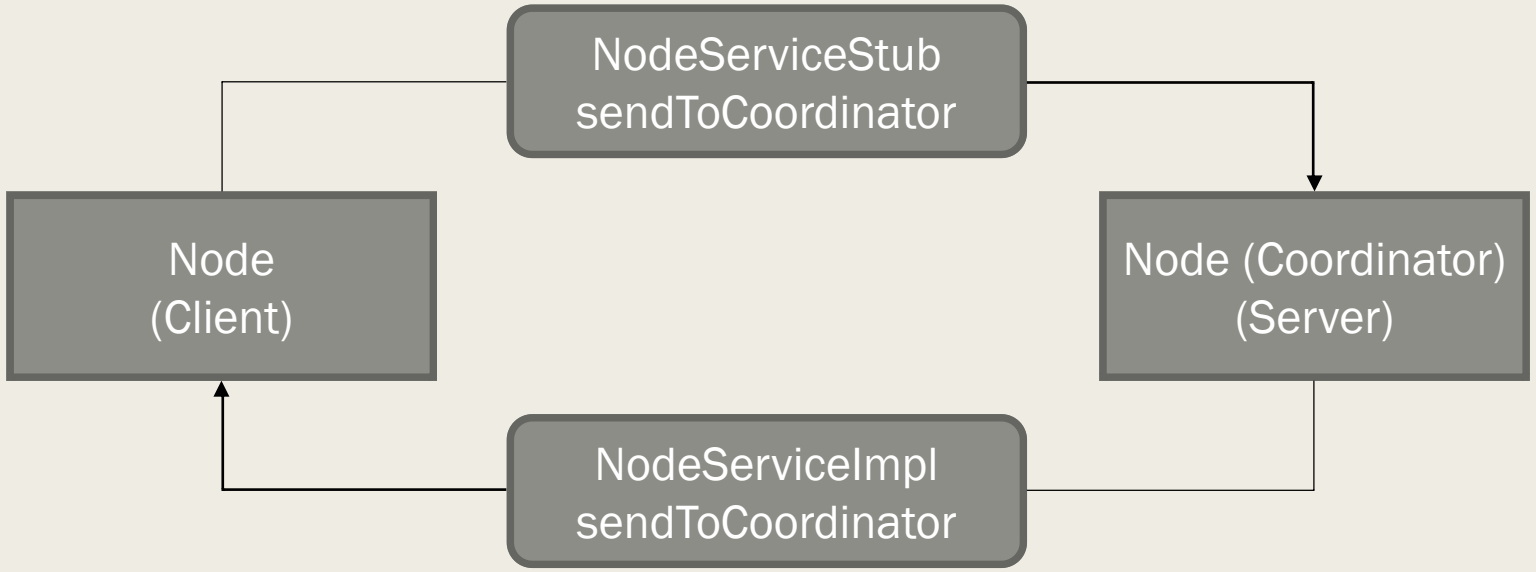


- Comunicazione tramite gRPC, utilizzando una blocking stub:
  - *Il nodo client chiama askForCoordinator passandogli una NodeRequest*
  - *Il nodo server effettua la chiamata e risponde con una CoordResponse*

Comunicazione Node – Node

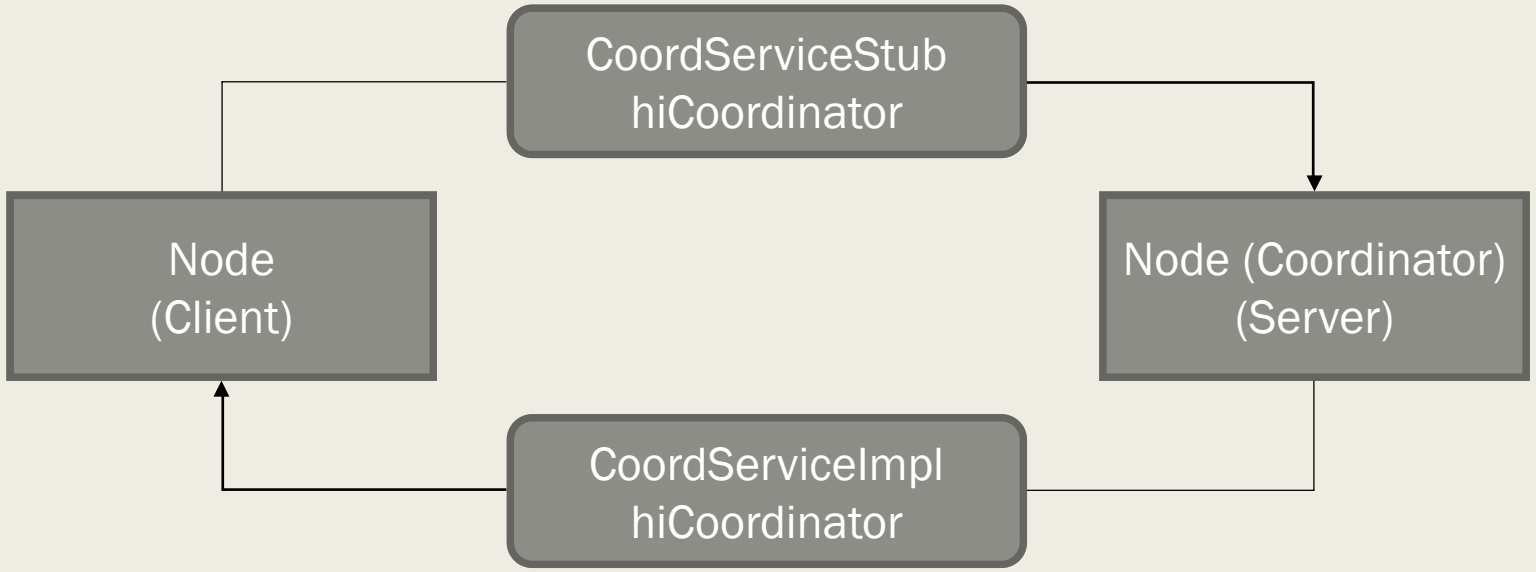
- Comunicazione tramite gRPC, utilizzando una nonblocking stub:
  - *Il nodo client chiama sendElectionMessage passandogli una ElectionRequest*
  - *Il nodo server effettua la chiamata e risponde con una ElectionResponse*

Comunicazione Node – Node (Coordinator)



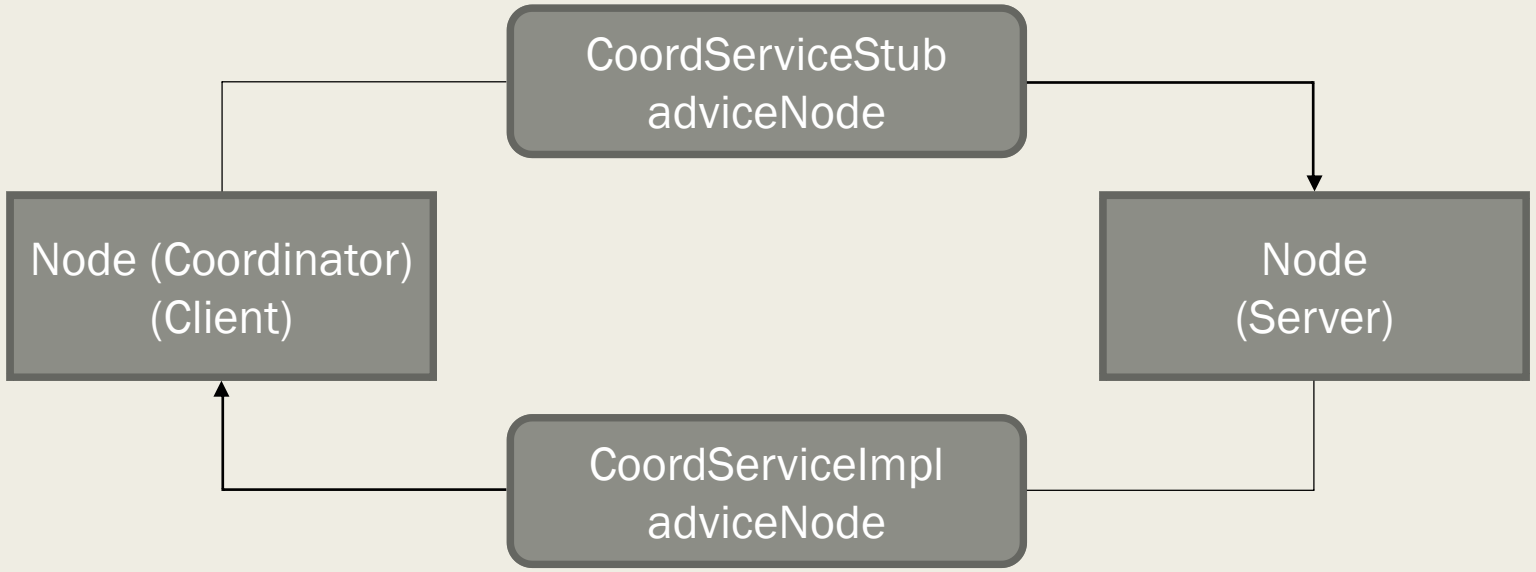
- Comunicazione tramite gRPC, utilizzando una nonblocking stub:
  - *Il nodo client chiama sendToCoordinator passandogli una LocalStatRequest*
  - *Il nodo (Coordinatore) server effettua la chiamata e risponde con una GlobalStatResponse*

Comunicazione Node – Node (Coordinator)



- Comunicazione tramite gRPC, utilizzando una blocking stub:
  - *Il nodo client chiama hiCoordinator passandogli una NodeRequest*
  - *Il nodo (Coordinatore) server effettua la chiamata e risponde con una NodeResponse*

Comunicazione Node (Coordinator) – Node



- Comunicazione tramite gRPC, utilizzando una nonblocking stub:
  - *Il nodo client chiama adviceNode passandogli una NodeRequest*
  - *Il nodo (Coordinatore) server effettua la chiamata e risponde con una NodeResponse*

FINE

Alessio Quercia