

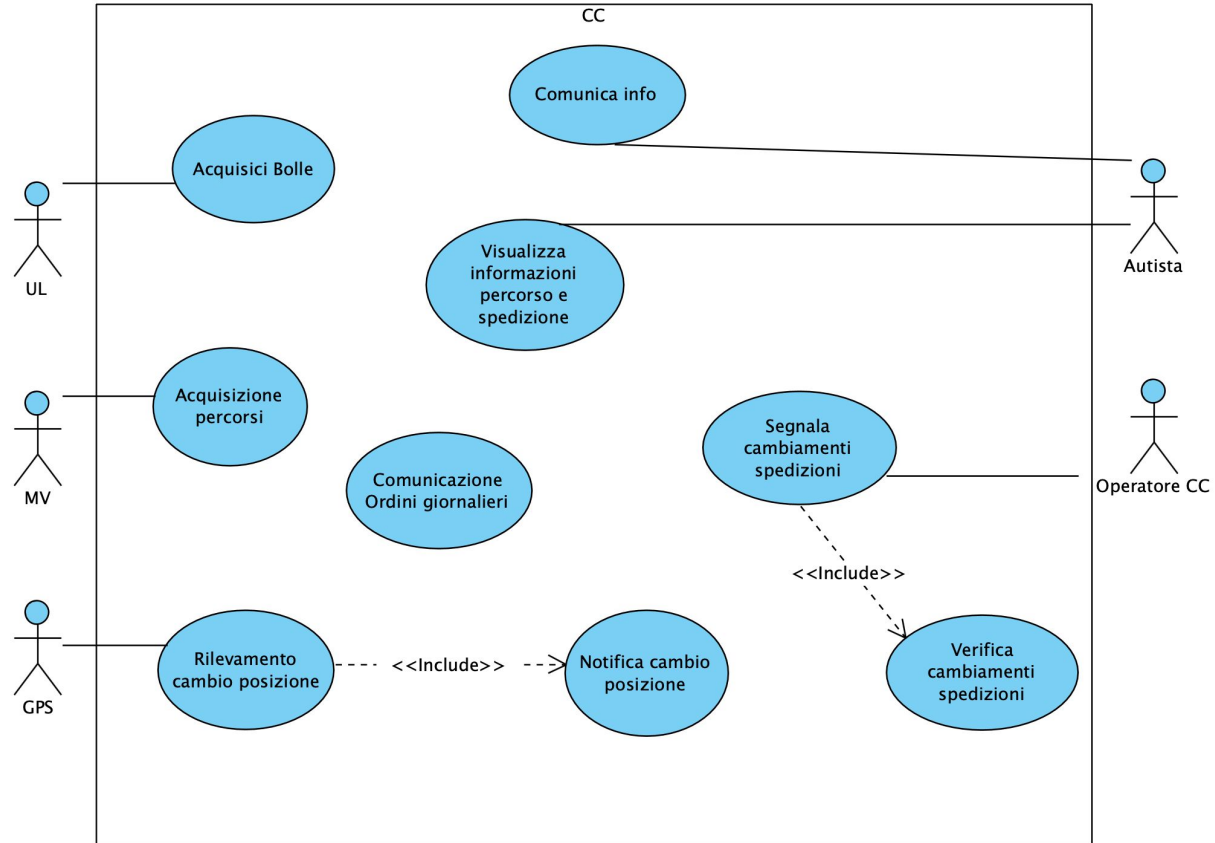


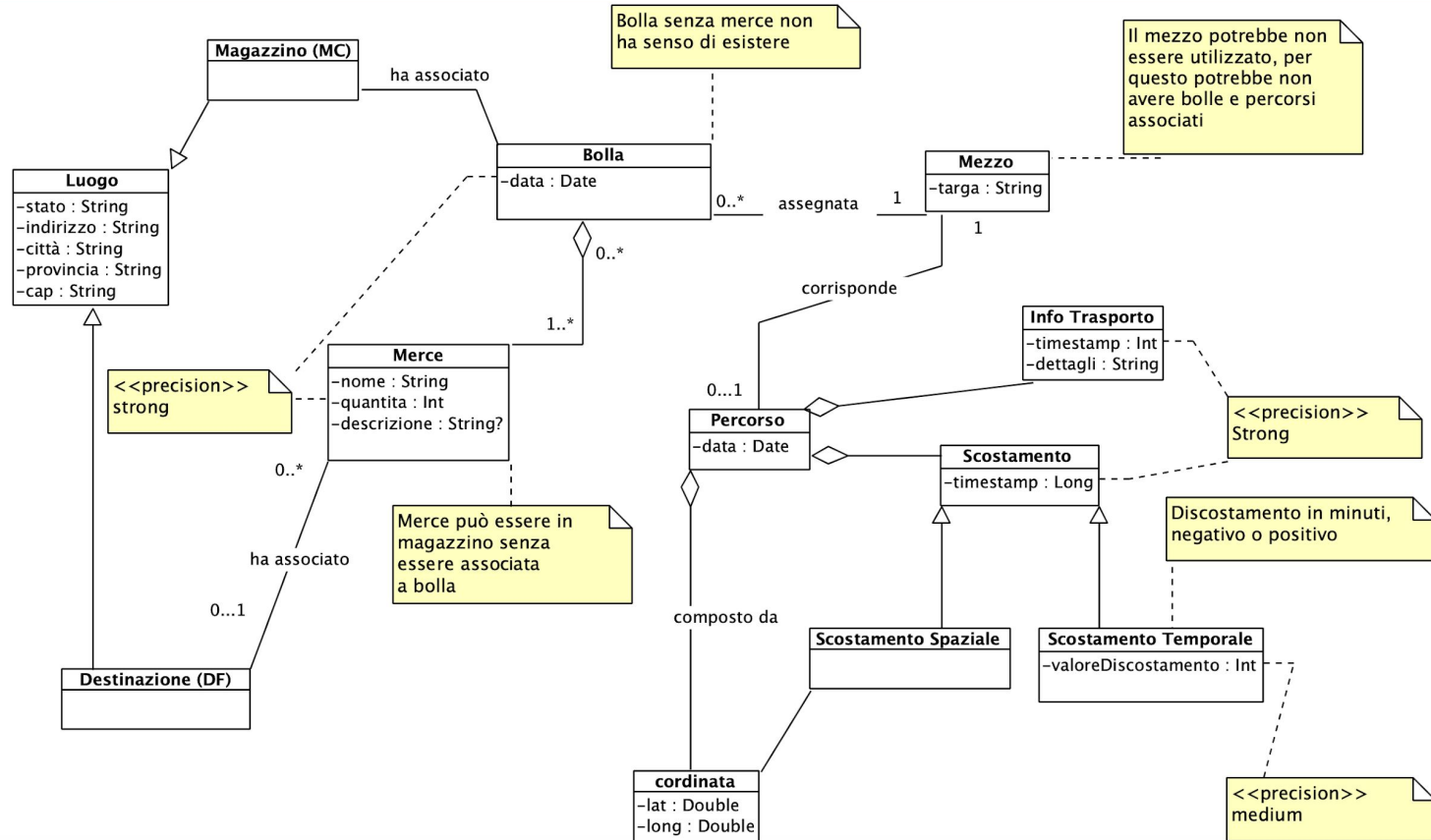
Architettura del SW

2018/2019

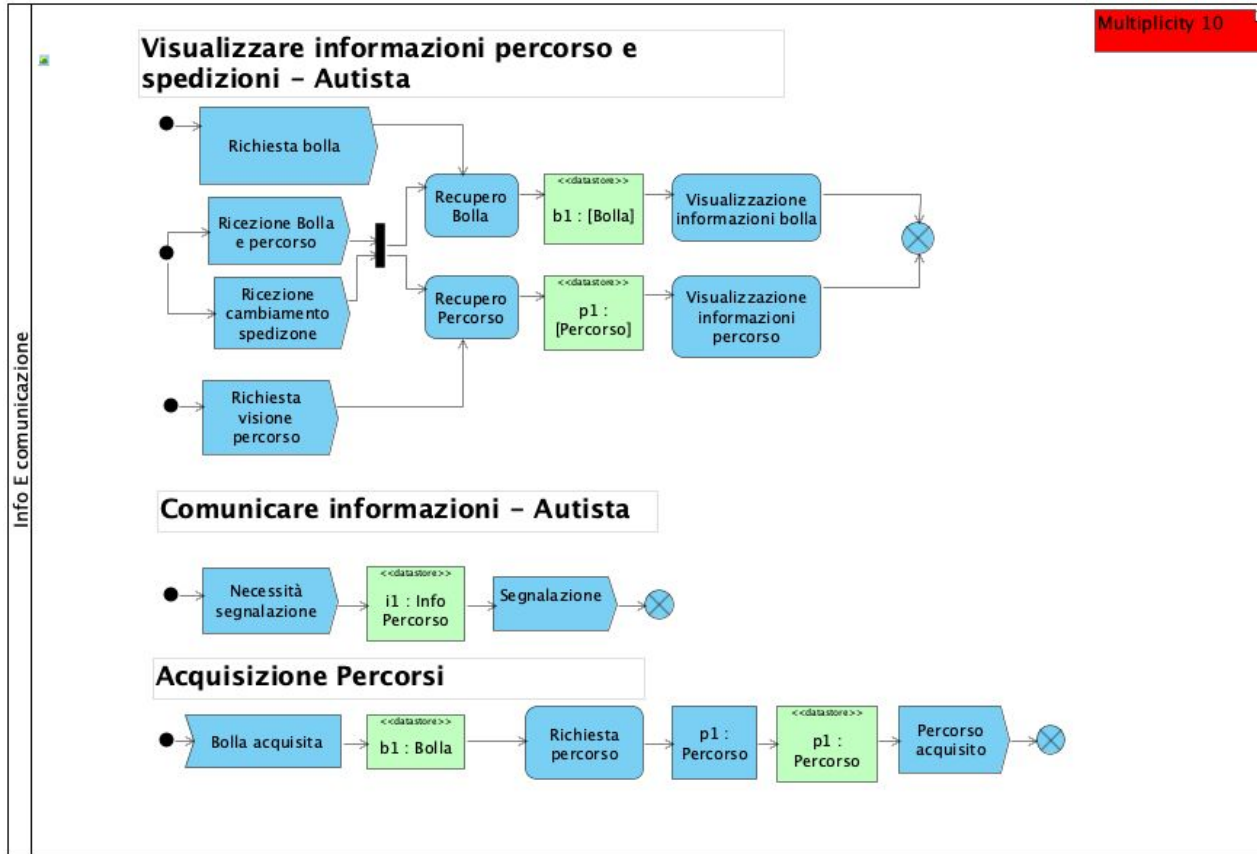
Ceresola Davide
Ceriani Massimiliano
Colombo Paolo

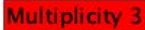
Problem Architecture - Casi D'uso



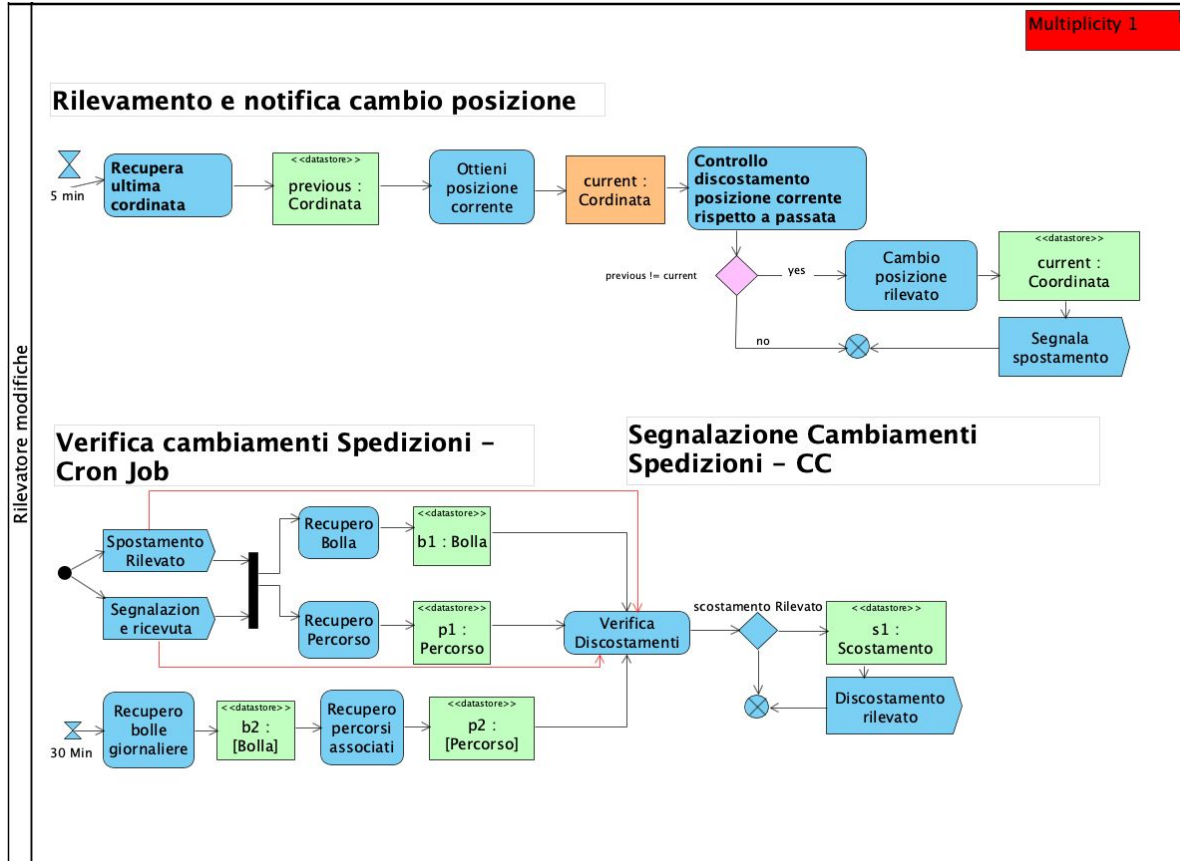


Problem Architecture - Activity Diagrams





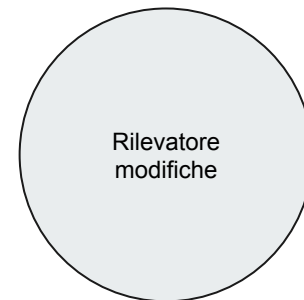
Problem Architecture - Activity Diagrams



Logical Architecture

| Dimensioni |
|-------------|
| Abstraction |
| Complexity |
| Frequency |
| Intra flows |
| extra flows |
| Sharing |

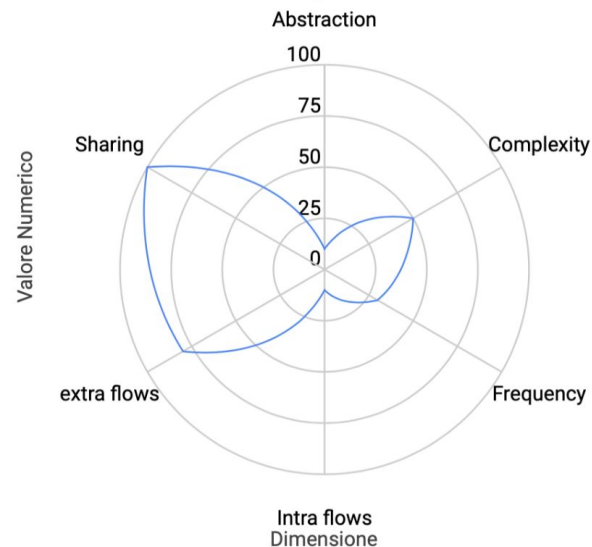
Criterio di partizionamento scelto: Per dominio applicativo



Logical Architecture - Gestore Ordini

| Dimensione | Valore | Valore Numerico | Motivo |
|-------------|--------|-----------------|---|
| Abstraction | Low | 10 | Il componente si interessa solo delle bolle e degli ordini, astraendo completamente il compito di comunicare gli stessi o rilevare modifiche |
| Complexity | Medium | 50 | Il componente ha la funzione di raccordo tra le informazioni prodotte dalle bolle e dal servizio che ne genera i percorsi estraendo le informazioni. |
| Frequency | Medium | 30 | Se considerassimo la produzione di ordine giornalieri sarebbe low, invece l'acquisizione della bolla ne determina un innalzamento. |
| Intra flows | Low | 10 | Il componente pur essendo unico dato che assume un ruolo fondamentale è previsto l'istanziamento di 3 istanze di questo componente che però non interagiscono tra di loro se non in modo irrilevante. |
| extra flows | High | 80 | L'acquisizione dei percorsi e l'invio degli ordini giornalieri coinvolge gli altri due componenti logici. |
| Sharing | High | 100 | L'istanziamento di più istanze di tale componente comporta un alto livello di questa dimensione, in quanto vengono condivise le informazioni su cui poi ogni componente |

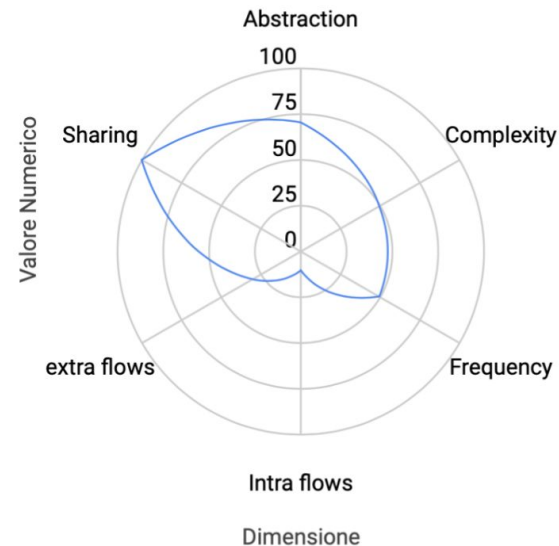
Valore Numerico rispetto a Dimensione



Logical Architecture - Info e comunicazione

| Dimensione | Valore | Valore Numerico | Motivo |
|-------------|--------|-----------------|---|
| Abstraction | High | 70 | Il componente per poter generare i percorsi e le informazioni relative ad essi utilizza anche le bolet |
| Complexity | Medium | 50 | Il componente non include tutte le attività |
| Frequency | Medium | 50 | La frequenza delle azioni svolte da questo componente sono direttamente proporzionali a quelle svolte dal componente gestore degli ordini. |
| Intra flows | Low | 10 | Il componente non necessita passaggio di informazioni tra le diverse istanze di esso in quanto ciascuna indipendentemente e senza stato. |
| extra flows | Medium | 30 | La notifica di acquisizione delle bolle e l'invio del messaggio di segnalazione rappresentano i soli punti di comunicazione con gli altri componenti logico.. |
| Sharing | High | 100 | L'istanziamento di più istanze di tale componente comporta un alto livello di questa dimensione, in quanto vengono condivise le diverse informazioni quali le bolle da dove poi partire per processare le informazioni. |

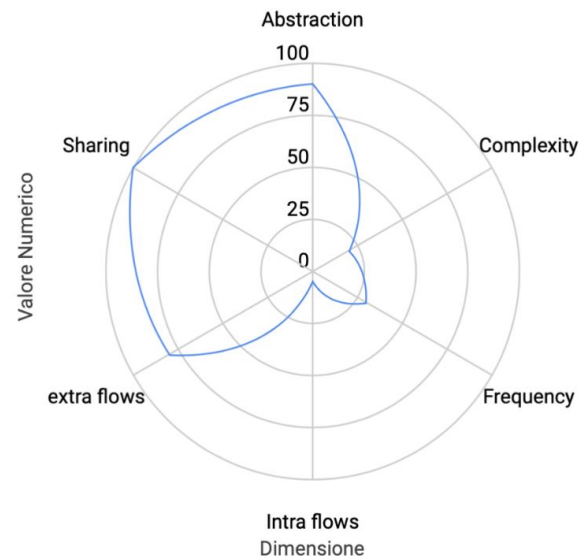
Valore Numerico rispetto a Dimensione



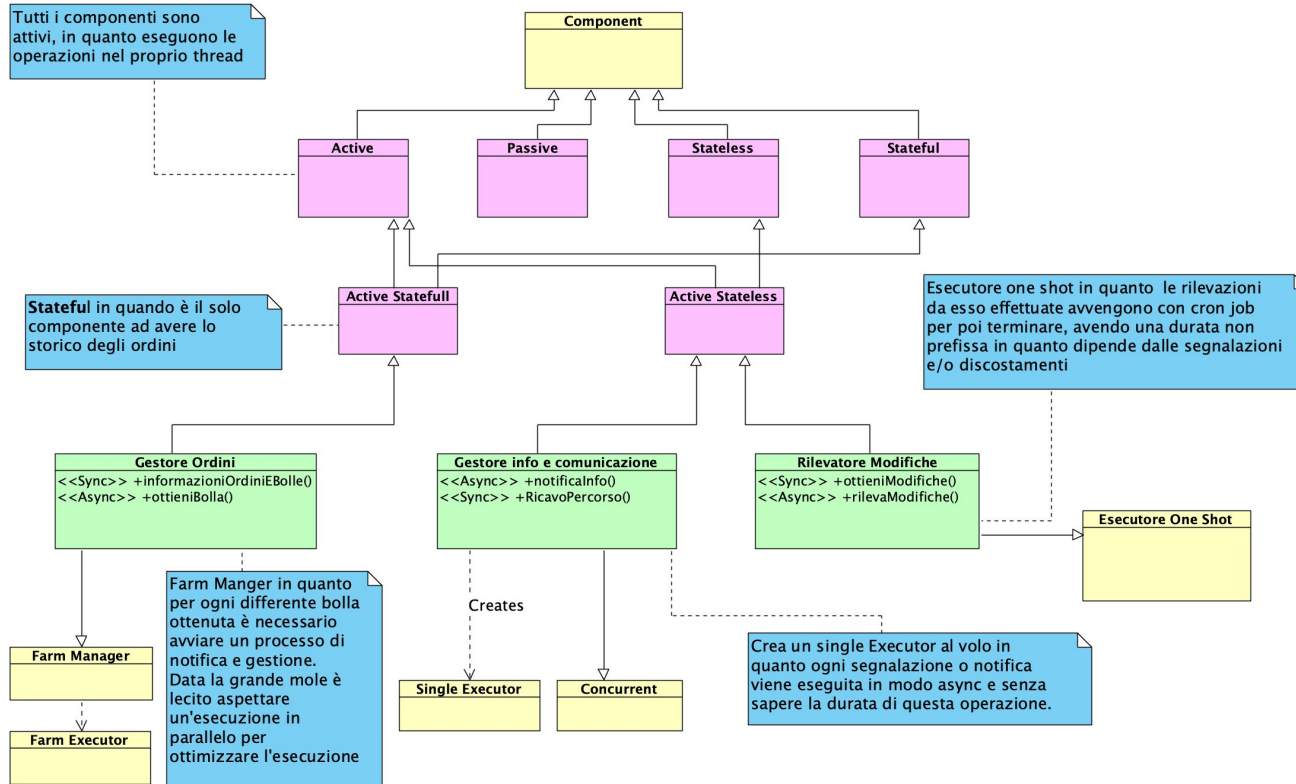
Logical Architecture - Rilevatore Modifiche

| Dimensione | Valore | Valore Numerico | Motivo |
|-------------|--------|-----------------|--|
| Abstraction | High | 90 | Il componente si focalizza principalmente sulle informazioni ottenute dai percorsi, sull bolle e sulle informazioni. Coinvolgendo quindi le principali entita del sistema. |
| Complexity | Low | 20 | Il componente deve recuperare le informazioni e verificare scostamenti temporali o spaziali, task non particolarmente complesso da implementare. |
| Frequency | Low | 30 | le due frequenze da prendere in considerazione per questo componente non sono elevate, in quanto ci si mantiene sull'unità di misura dei minuti. |
| Intra flows | Low | 5 | Il componente con molteplicità 1 non necessita di flussi di informazioni interni. |
| extra flows | High | 80 | Come espresso in precedenza, tale componente pur eseguendo un compito non di complessità elevata, utilizza tutti le informazioni prodotte dagli altri due servizi e li coinvolge al termine delle proprie. |
| Sharing | Hig | 100 | Il componente utilizza tutte le informazioni prodotte dagli altri servizi. |

Valore Numerico rispetto a Dimensione

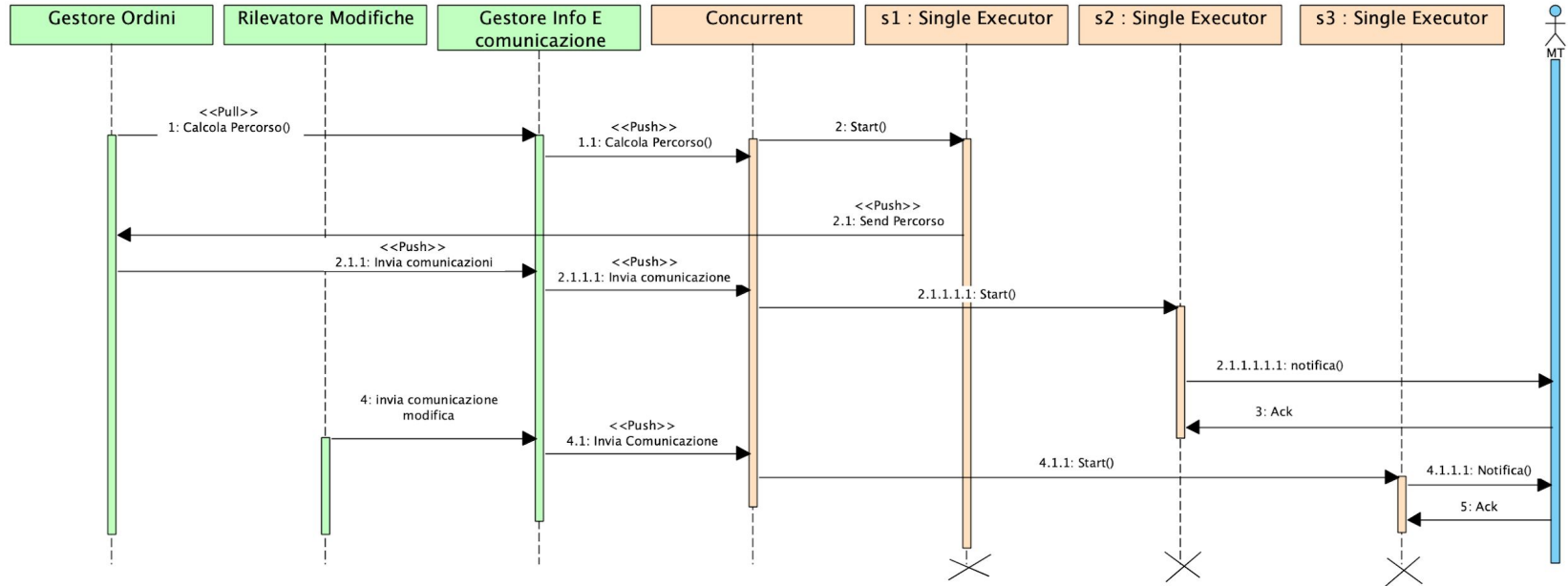


Concrete Architecture - Class Diagram



Concrete Architecture - Sequence Diagram

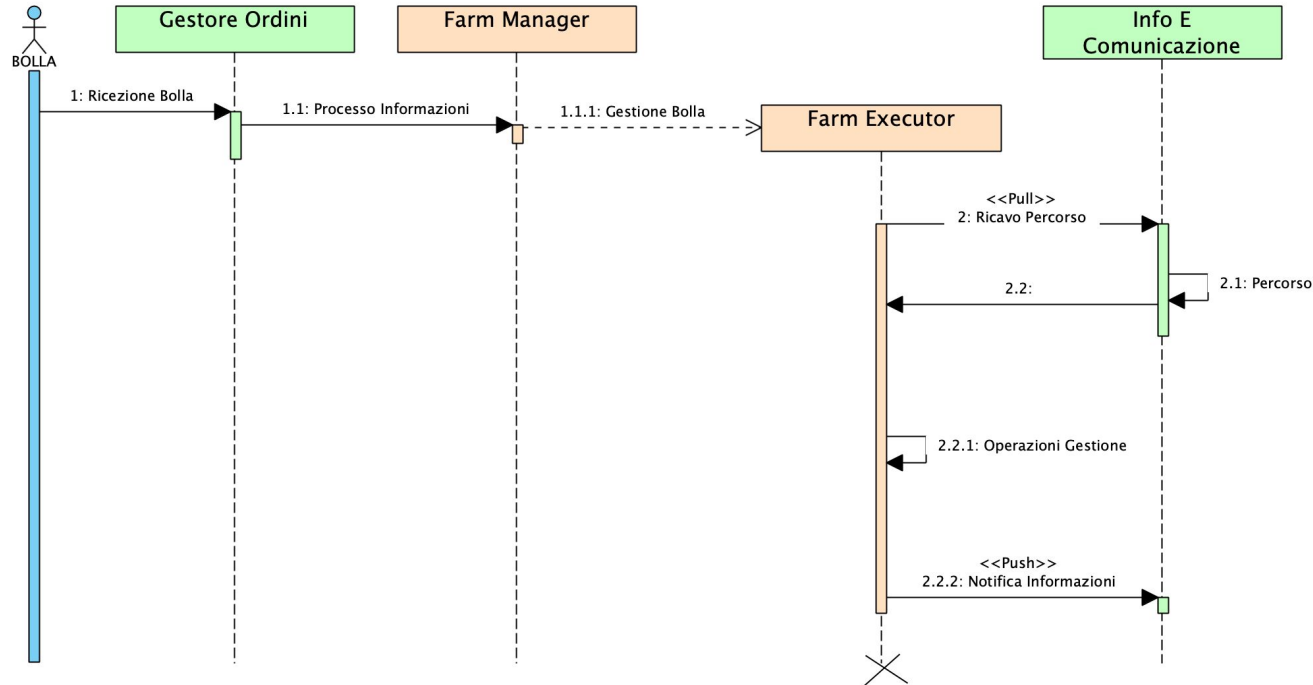
Notifica Info E Comunicazione



Single Executor usati per notificare o calcolare i percorsi. Si noti come la notifica a MT data la sua rilevanza avviene in modo SYNC

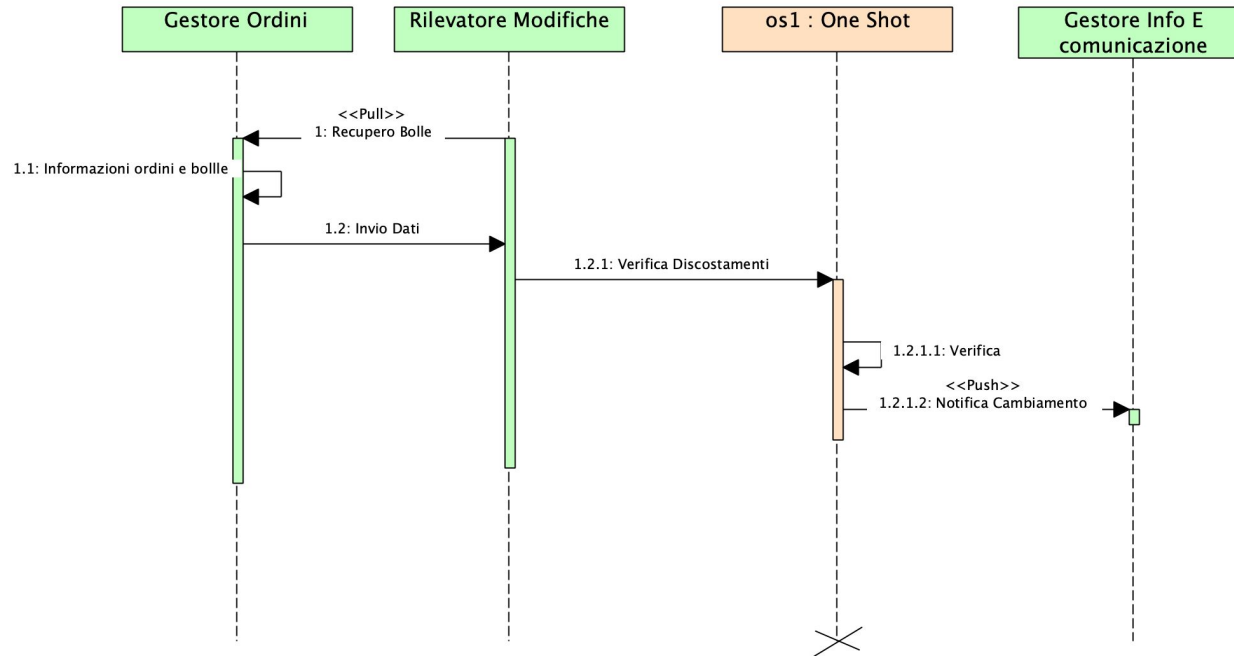
Concrete Architecture - Sequence Diagram

Gestione e ricezione bolla



Concrete Architecture - Sequence Diagram

Rilevazioni Modifiche





System Quality Attributes

Disponibilità

- Il sistema deve essere sempre disponibile, 24/24 7/7, in quanto lo spostamento delle merci non ha orari lavorativi predefiniti. Per esempio anche la notte i mezzi effettuano i loro compiti segnalando continuamente la loro posizione.
- Con il fine di garantire una elevata disponibilità, si è scelto di **Dockerizzare** ognuno dei tre componenti precedentemente descritti, in modo da poter gestire velocemente il loro deployment .
- I vari container vengono successivamente orchestrati i tramite **Kubernetes** su un cluster di almeno 2 nodi, in modo da essere esente da eventuali fault.
- Per la stessa ragione ogni servizio è replicato almeno due volte in nodi diversi.



System Quality Attributes

Sicurezza

- La sicurezza delle comunicazioni tra i dati relativi al GPS di ogni mezzo vengono acquisiti e trasmessi tramite il protocollo **MQTT**. Per proteggere questo flusso di informazioni è stato scelto di utilizzare il servizio di IOT messo a disposizione da **AWS**. Questa scelta deriva dal fatto che il servizio in questione consente una mutua autenticazione tra le parti tramite chiave pubblica/privata e mediante l'utilizzo di certificati. Tale accortezza consente di evitare l'**integrità** e **confidenzialità** di questi dati.
- La visualizzazione delle bolle essendo le stesse confidenziali, è limitata ai soli utenti aventi diritto, mediante un meccanismo gerarchico di autorizzazioni date per step, tale permessi sono richiesti a ogni accesso, anche per utenti già autenticati.
- I servizi disponibili verso l'esterno sono protetti da un ingress (il nome che **Kubernetes** fornisce al proprio servizio di API Gateway come punto di ingresso ai servizi sul cluster) che gestirà le richieste previa autenticazione. L'accesso al cluster, deployato sul servizio EKS di Amazon, è inoltre definito da determinate policy che limitano gli accessi ai soli utenti autorizzati, per prevenire l'eventuale manomissione dei dati.
- Come servizio di **DNS** è stato usato quello di Cloudflare che, tra le altre caratteristiche come la bassa latenza, offre un servizio di protezione per attacchi **DDoS**.



System Quality Attributes

Performance

- Utilizzando **Kubernetes** inoltre risulta facile rendere il sistema reattivo a picchi di carico. Quando è necessario infatti l'orchestratore aumenta le repliche dei servizi di cui lo necessitano, consentendo al sistema di poter reggere il nuovo carico.
- Per rendere possibile ciò si sfruttano gli **HPA**, horizontal pod autoscaler, che permettono di deployare diversi container replicando il servizio sui nodi disponibili quando la CPU supera una certa soglia. Inoltre i servizi di cluster autoscaling di Kubernetes permettono di scalare accendendo o spegnendo nuove macchine **ec2** (incrementando quindi il numero di nodi), in base alla necessità di aggiungere/rimuovere container.

Testabilità e Modificabilità

- Il sistema essendo suddiviso in servizi, rende ogni singolo servizio testabile indipendentemente. La modifica di uno di essi inoltre potrebbe essere deployata nel cluster già in esecuzione, rimuovendola in caso di problemi senza impattare l'intero sistema. Questo permette di analizzare e testare le nuove modifiche in modo safe.