

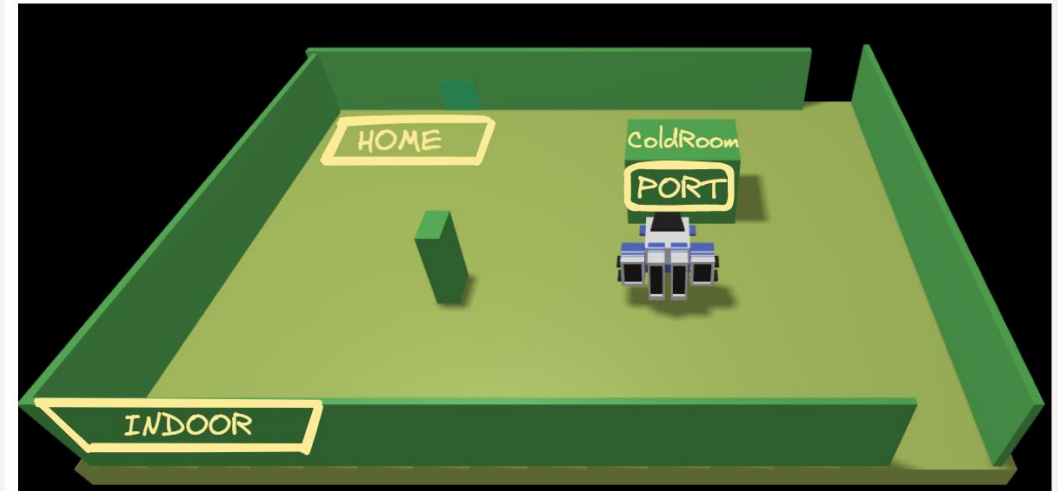
# ColdStorageService Project

PRESENTAZIONE PER L'ESAME DI INGEGNERIA DEI SISTEMI SOFTWARE –  
ING. INFORMATICA LM UNIBO

# Il progetto ColdStorage Service

Lo scenario proposto è quello in cui un committente ci incarica di implementare un sistema di gestione di un 'ColdStorage', fornendoci un BasicRobot che agisce in una scena WebGL come quella in figura, nello specifico il sistema deve essere in grado di:

- Comunicare con questo BasicRobot e simulare il carico e lo scarico di una quantità di cibo in un componente detto 'ColdRoom' che funge da frigorifero.
- Implementare l'applicazione che gestirà le richieste e le validazioni dei tickets che i clienti dovranno richiedere per poter utilizzare il servizio.
- Gestire un sistema di allarmi che comunicherà con la nostra applicazione, simulando la presenza di impedimenti od ostacoli che potrebbero apparire lungo il tragitto del BasicRobot.



# Gestione del lavoro e Tecnologie usate

Per approcciarsi al problema si è deciso di dividere il lavoro in Sprint, nel particolare gli sprint serviranno per:

- Chiarire la consegna ed i requisiti con il committente.
- Identificare e gestire il core dell'applicazione.
- Lavorare sulla connessione coi sistemi di allarme, che dovranno essere indipendenti dall'applicazione principale.
- Implementare un Server Web, tramite il quale agli utenti sarà permesso di interagire con l'applicazione.

Le tecnologie che si è deciso di utilizzare sono:

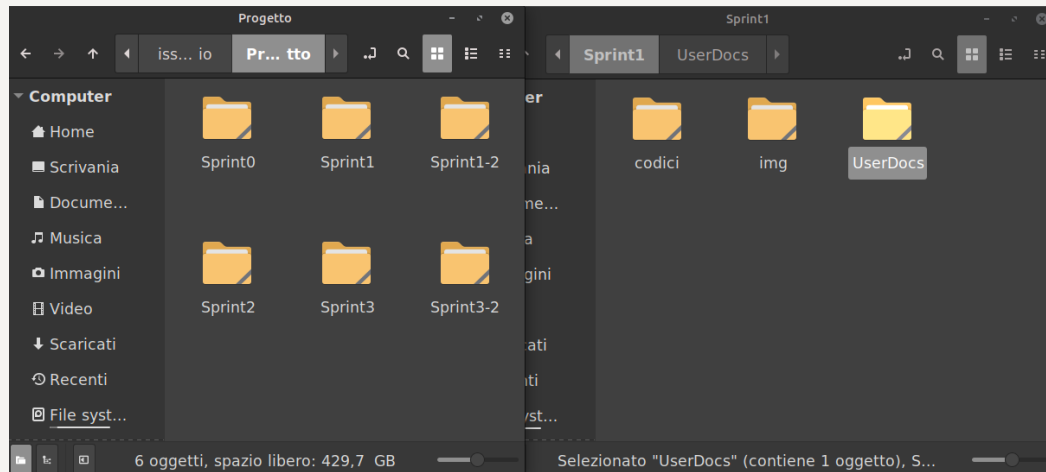
- Linguaggio QAK ed i QActor, che implementano comunicazioni COAP e TCP, per gestire il core dell'applicazione.
- Linguaggio Java e classi che stabiliscono connessioni COAP e TCP con i attori QActor, per gestire la connessione col sistema di allarmi.
- SpringBoot con Thymeleaf e TomCat per gestire il Server Web con il quale gli utenti si interfacciano.

# Directories del Progetto

All'interno del progetto è presente un file README che riassume tutti i collegamenti ai file necessari.

Inoltre è presente una cartella chiamata 'Progetto' dove sono visionabili tutte le cartelle dei vari sprint, contenenti anche codici, immagini e documentazioni che esplicano lo Sprint nel dettaglio, quest'ultimi si trovano all'interno della cartella 'UserDocs' di ogni sprint.

In questo modo si possono vedere i vari step del progetto, e possono essere generati tutti i test descritti nelle documentazioni.

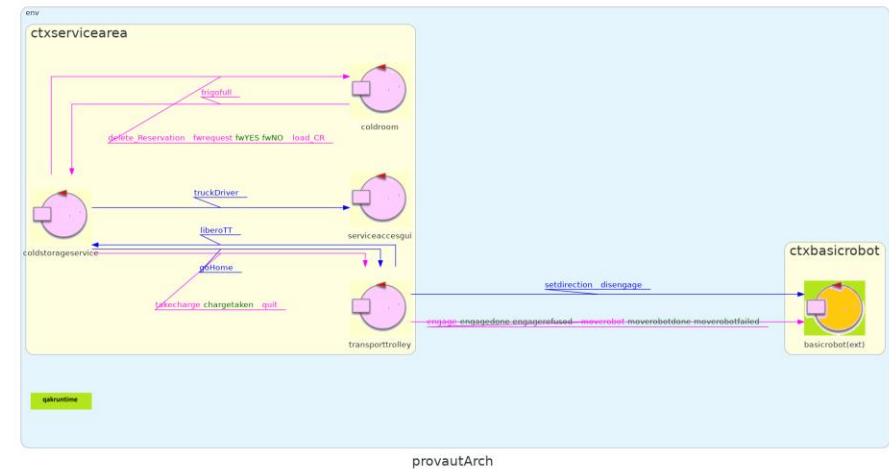


# Sprint 1

**Goal:** produrre una versione funzionante del core applicativo.

1. Identifichiamo i componenti dell'applicazione, e decidiamo di esprimerli come QActors, utilizzando il linguaggio QAK.
2. Ci concentriamo inizialmente sulla generazione e gestione dei tickets e dello stato del 'ColdRoom'.
3. Lavoriamo poi sulla gestione del collegamento con il BasicRobot.

Giungiamo all'applicazione espressa dall'architettura di fianco, sfruttando momentaneamente una versione MockObj per la ServiceAccessGUI.



# Sprint 2

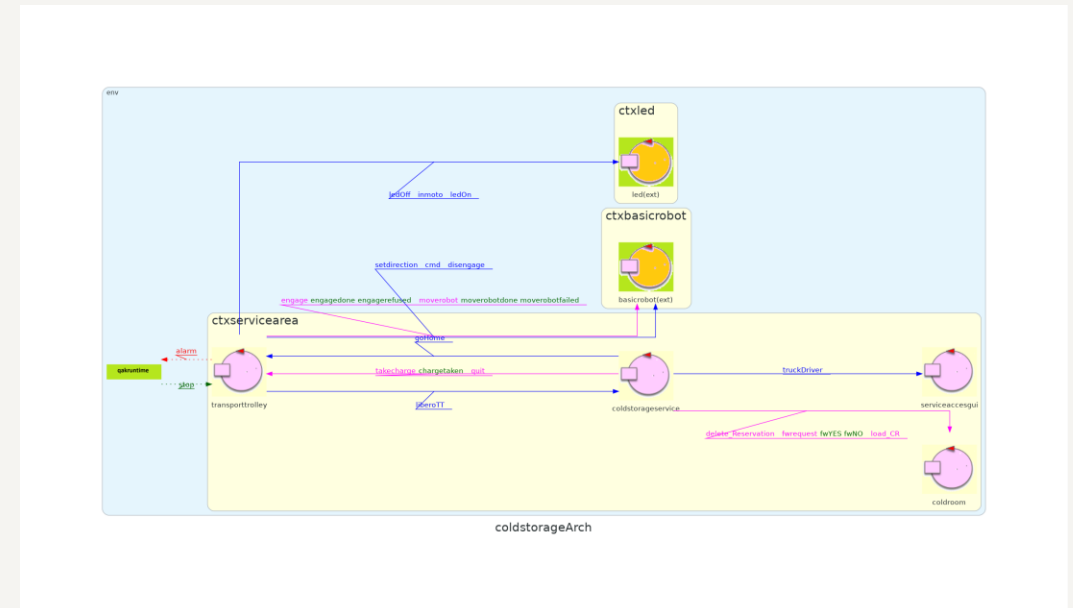
**Goal:** simulare il sistema di allarmi che collegheremo alla nostra applicazione.

Ci viene chiesto di simulare:

- Un led, che esprimiamo come un QActor esterno.
- Un Sonar, che implementiamo come una classe Java.

Inoltre vogliamo questi componenti completamente indipendenti dalla nostra applicazione.

Alla fine il nostro sistema si riassume nell'architettura presente in figura.

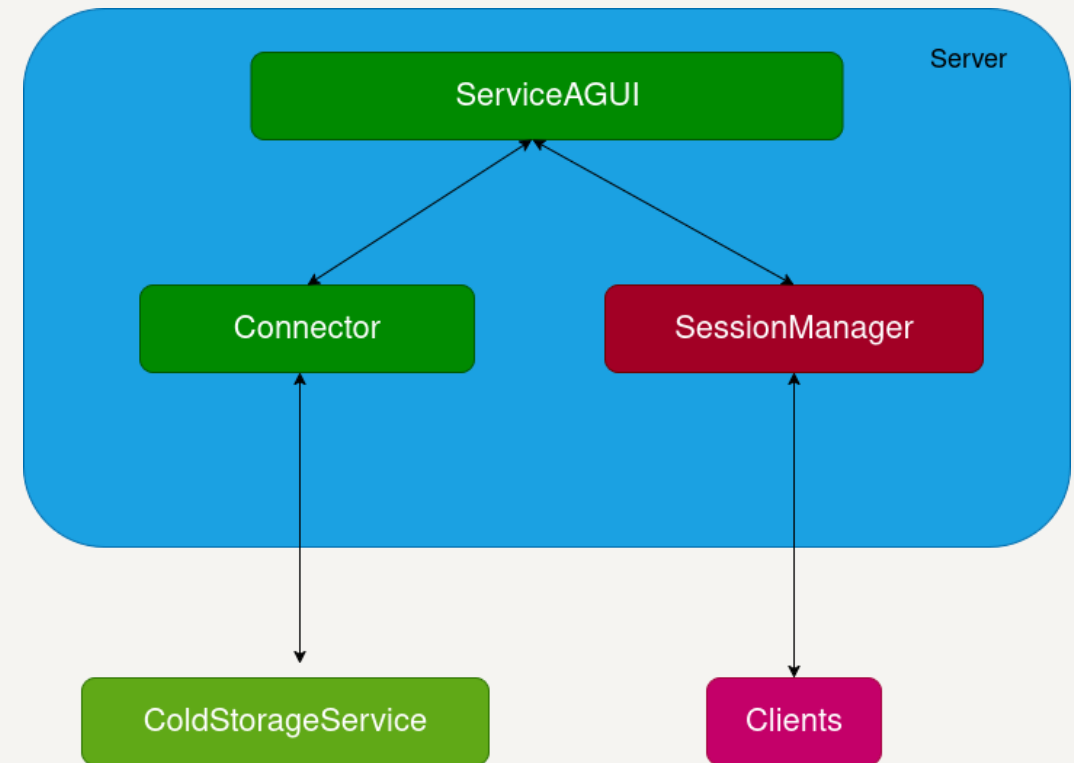


# Sprint 3

**Goal:** creare un Server Web che permetta ai Clients di comunicare con la nostra applicazione per richiedere i tickets, validarli ed osservare le condizioni del 'ColdRoom'.

- Usiamo la tecnologia di SpringBoot che comprende Thymeleaf e Apache TomCat.
- Usiamo connessioni TCP e COAP per comunicare con il nostro ColdStorage.

Ci ritroviamo infine con una architettura a struttura triangolare, come in figura.







GRAZIE PER L'ATTENZIONE

PRESENTATO DA :

CRISTINA NAPOLI – MATRICOLA: 0001103127