

## Introduction

## Requirements

Descrizione requirements a questa pagina

### Goals dello sprint 0:

1. Identificare i componenti principali del servizio **ColdStorageService**
2. Formulare un modello di interazione dei componenti individuati
3. Modellare una architettura logica del sistema che simula le azioni che scatenano le interazioni individuate

## Requirement analysis

### Domain analysis

Termine	Descrizione
<b>ColdStorageService</b>	Sistema distribuito per il deposito di prodotti alimentari all'interno di una stanza refrigerata, automatizzato tramite l'uso di un <u>transport trolley</u> .
<b>Service area</b>	Stanza piatta a pianta rettangolare nella quale il <u>transport trolley</u> può navigare, contiene: <ul style="list-style-type: none"><li>• <u>INDOOR</u>;</li><li>• <u>ColdRoom</u>;</li><li>• <u>HOME</u>;</li><li>• ostacoli con posizione e dimensione predefinita (tra i quali la <u>ColdRoom</u>).</li></ul>
<b>INDOOR</b>	Porta di accesso adibita al trasferimento di un carico dal <u>camion refrigerato</u> al <u>transport trolley</u> .
<b>ColdRoom</b>	Contenitore refrigerato adibito al deposito di cibo ritirato dal <u>camion refrigerato</u> da parte del <u>transport trolley</u> .
<b>MAXW</b>	Limite massimo di chilogrammi che la <u>ColdRoom</u> può contenere in un dato momento.
<b>Transport trolley</b>	Rappresenta un robot virtuale di forma quadrata e lato <u>RD</u> . Astrazione che usa il <u>DDR Robot</u> per muoversi all'interno della <u>service area</u> .
<b>HOME</b>	Posizione di partenza del <u>transport trolley</u> situato nell'angolo tra il muro superiore e quello sinistro della <u>service area</u> .
<b>RD</b>	Lunghezza del lato del <u>transport trolley</u> .
<b>Azione di deposito</b>	Operazione di deposito che coinvolge il <u>camion refrigerato</u> e il <u>transport trolley</u> : <ol style="list-style-type: none"><li>1. il <u>transport trolley</u> preleva un carico di cibo dal <u>camion refrigerato</u> tramite la porta <u>INDOOR</u>;</li><li>2. il <u>transport trolley</u> si sposta da <u>INDOOR</u> a <u>PORT</u>;</li><li>3. il carico viene depositato nella <u>ColdRoom</u>.</li></ol>
<b>Food load</b>	Carico di cibo (frutta, vegetali...) che necessita di essere refrigerato, sia durante il trasporto che nella fase di accumulazione.

<b>Camion refrigerato</b>	Entità esterna al sistema che deposita <u>FW</u> kg di prodotti alimentari tramite la porta <u>INDOOR</u> .
<b>ServiceAccessGUI</b>	Interfaccia grafica che permette di: <ul style="list-style-type: none"> <li>• visualizzare il peso corrente del materiale immagazzinato nella <u>Cold Room</u>;</li> <li>• inviare la richiesta di deposito di una quantità <u>FW</u> kg di prodotti alimentari (ritornando un <u>ticket</u> in caso di richiesta accettata);</li> <li>• immettere il codice del ticket quando il <u>camion refrigerato</u> si trova alla porta <u>INDOOR</u>, all'esterno della service area;</li> <li>• visualizzare il messaggio di <u>presa in carico</u>.</li> </ul>
<b>ServiceStatusGUI</b>	Interfaccia grafica che permette a un utente umano di visualizzare: <ul style="list-style-type: none"> <li>• lo <u>stato del transport trolley</u>;</li> <li>• il <b>peso corrente</b> del cibo nella <u>ColdRoom</u>;</li> <li>• il <b>numero di richieste rifiutate</b> dall'inizio del servizio.</li> </ul>
<b>DDR robot</b>	Differential Drive Robot fornito dal committente insieme al software <u>BasicRobot</u> per controllarlo. Nell'ambito del <u>ColdStorageService</u> concretizza il comportamento del <u>transport trolley</u> .
<b>Service manager</b>	Utente umano che può interagire con la <u>ServiceStatusGUI</u> per monitorare lo <u>stato della service area</u>
<b>Sonar</b>	Dispositivo collegato a un RaspberryPi che, nell'ambito del <u>ColdStorageService</u> , concretizza il comportamento dell' <u>alarm device</u> , è utilizzabile tramite software fornito dal committente.
<b>Alarm device</b>	Dispositivo astratto in grado di rilevare la distanza di oggetti posti di fronte ad esso.
<b>Led</b>	Dispositivo collegato a un RaspberryPi che, nell'ambito del <u>ColdStorageService</u> , concretizza il comportamento del <u>warning device</u> , in particolare: <ul style="list-style-type: none"> <li>• è <b>spento</b> se il <u>transport trolley</u> è situato in HOME;</li> <li>• <b>lampeggia</b> se il transport trolley è in movimento;</li> <li>• è <b>acceso</b> se il transport trolley non è in movimento.</li> </ul> Il dispositivo è controllabile tramite software fornito dal committente.
<b>Warning device</b>	Dispositivo astratto in grado di mostrare tre distinti stati: <ul style="list-style-type: none"> <li>• <u>transport trolley</u> situato in HOME</li> <li>• transport trolley in movimento</li> <li>• transport trolley non in movimento (e non in HOME)</li> </ul>
<b>DLIMIT</b>	Limite di distanza massimo dal <u>sonar</u> per il quale si attivano i relativi <u>alarm requirements</u> : se viene rilevata una distanza minore, viene inviato un segnale di stop al <u>transport trolley</u> .
<b>TICKETTIME</b>	Quantità di tempo che esprime la durata della validità di un <u>ticket</u> , se il camion non è alla porta <u>INDOOR</u> entro questo limite la richiesta viene annullata.
<b>FW</b>	Quantità di carico che un <u>camion refrigerato</u> deve depositare, espressa in chilogrammi.
<b>Stato transport trolley</b>	Stato corrente del <u>transport trolley</u> , questo include: <ul style="list-style-type: none"> <li>• posizione nella <u>service area</u>;</li> <li>• stato di movimento (<b>stopped,moving</b>).</li> </ul>
<b>MINT</b>	Tempo minimo che deve trascorrere prima che il <u>transport trolley</u> possa nuovamente gestire un segnale di stop secondo le condizioni imposte dagli <u>alarm requirements</u> , espresso in millisecondi.
<b>Ticket</b>	Rappresenta una prenotazione da parte dell'operatore di un camion refrigerato che identifica univocamente una azione di deposito in attesa di essere <u>presa in carico</u> , è costituito da:

	<ul style="list-style-type: none"> <li>• codice alfanumerico univoco;</li> <li>• un valore <u>TICKETTIME</u>;</li> <li>• un <b>timestamp</b> che indica il momento di emissione.</li> </ul>
<b>Stato service area</b>	Comprende lo stato di diversi componenti che si trovano all'interno della <u>service area</u> , visualizzabile tramite la <u>ServiceStatusGUI</u> .
<b>Charge taken</b>	Messaggio che rappresenta la presa in carico di un carico di cibo da parte del transport trolley. Questo messaggio è relativo ad una singola richiesta di deposito da parte di un <u>camion refrigerato</u>

## Verbs analysis

Soggetto	Termine	Oggetto	Descrizione
<u>Camion refrigerato</u>	<b>Sends a request</b>	Peso <u>FW</u>	Il (guidatore del) camion invia una richiesta di deposito di <b>FW</b> kg di cibo tramite la <u>ServiceAccessGUI</u> .
	<b>Drives to the INDOOR</b>	-	Il (guidatore del) camion, in caso di richiesta accettata, va fino alla porta di INDOOR, prima della scadenza del ticket ( <u>TICKETTIME</u> ).
	<b>Enters the ticket number</b>	<u>Ticket</u> number	Il (guidatore del) camion, inserisce il numero del ticket tramite la <u>ServiceAccessGUI</u> .
	<b>Waits</b>	Messaggio <u>charge taken</u>	Il (guidatore del) camion attende fino a che il messaggio <u>Charge taken</u> non appare sulla <u>ServiceAccessGUI</u> .
	<b>Leaves the INDOOR</b>	-	Il (guidatore del) camion, in caso di <u>richiesta di deposito</u> rifiutata, o dopo aver <u>atteso il messaggio</u> , lascia la <u>INDOOR</u> .
<u>ColdStorageService</u>	<b>Accepts a ticket</b>	<u>Ticket</u>	Il servizio accetta il ticket che il (guidatore del) camion <u>ha inserito</u> .
	<b>Avvisa il transport trolley</b>	Peso <u>FW</u>	Il servizio comunica al <u>transport trolley</u> che c'è del cibo da depositare.
	<b>Sends message Charge taken</b>	<u>Charge taken</u>	Il ColdStorageService comunica alla <u>ServiceAccessGUI</u> che il transport trolley ha raccolto il carico.
<u>Transport trolley</u>	<b>Accepts a ticket</b>	<u>Ticket</u>	Il transport trolley accetta un nuovo ticket.
	<b>Reaches the INDOOR</b>	-	Il transport trolley raggiunge la <u>INDOOR</u> .
	<b>Picks up the food</b>	Peso <u>FW</u>	Il transport trolley prende il cibo da depositare dalla <u>INDOOR</u> .
	<b>Goes to the ColdRoom</b>	-	Il transport trolley si muove fino alla <u>ColdRoom</u> .
	<b>Stores the food</b>	Peso <u>FW</u>	Il transport trolley deposita il cibo nella <u>ColdRoom</u> , il completamento di questa azione determina la <b>fine dell'azione di deposito</b>
	<b>Return to HOME</b>	-	Il transport trolley ritorna alla <u>HOME</u> dopo aver <u>depositato</u> il cibo.
	<b>Sends message Charge taken</b>	<u>Charge taken</u>	Il transport trolley comunica al <u>ColdStorageService</u> di aver raccolto il carico.

## Problem analysis

In questa fase verranno presi in analisi i macro-componenti del sistema e le interazioni che avvengono fra essi, con il fine di delineare l'architettura logica del sistema.

Abbiamo deciso di utilizzare il modello ad attori per questo progetto in quanto rispecchia i requisiti analizzati dal documento fornito dal cliente: il sistema è costituito da componenti **eterogenei** che interagiscono tra di loro su **base distribuita**; inoltre abbiamo a disposizione librerie e un linguaggio di metamodellazione che supporteranno lo sviluppo orientato agli attori.

Il linguaggio in questione è QAK, un linguaggio di modellazione eseguibile che permette di creare prototipi e sistemi funzionanti in breve tempo (rispetto a linguaggi tradizionali) semplificando gli aspetti più complessi della gestione delle interazioni tramite una struttura basata su attori che operano in contesti.

Oltre alla velocità di sviluppo possibile con QAK, la scelta deriva dal fatto che il cliente ha fornito software per pilotare un DDR robot (o robot virtuali) compatibile con QAK.

### Macro-componenti del sistema

All'interno di questa sezione verranno presentati i componenti che abbiamo individuato come principali, in quanto entità a se stanti che interagiscono tra di loro.

Per ognuno di essi saranno evidenziate le responsabilità che ha, trattando anche eventuali sotto-componenti che possiede.

Ogni componente verrà poi analizzato in maggiore dettaglio nei successivi Sprint.

### ColdStorageService

Il **ColdStorageService** è il componente principale del sistema, ed è composto da:

- ServiceAccessGUI
- ServiceStatusGUI

Il ColdStorageService è incaricato di comunicare al transport trolley la presenza di una nuova richiesta di deposito, mentre il raggiungimento di certi landmark nella service area (HOME, INDOOR, ColdRoom), è delegato al transport trolley.

In questo modo il CSS sarà indipendente dalla particolare disposizione della service area.

Quando un utente (camion) richiede un ticket tramite la ServiceAccessGUI, il ColdStorageService valuterà la richiesta, eventualmente emettendo un nuovo ticket (se la ColdRoom può ospitare il nuovo carico).

Nel momento in cui l'utente inserisce il codice del ticket (sempre tramite la SAG), il ColdStorageService comunicherà al transport trolley che è disponibile un nuovo carico da prelevare.

Si rimanda alla domain analysis per la descrizione della ServiceStatusGUI.

I dati visualizzati nella SSG sono di competenza del ColdStorageService, ad eccezione dello stato del transport trolley (fornito dal transport trolley stesso).

### Camion refrigerato

Consideriamo il camion refrigerato come un componente del sistema, anche se si tratta di un utente umano, in quanto esso dà il via al comportamento del sistema, inviando una richiesta per un ticket tramite la ServiceAccessGUI.

L'interazione tra camion e ColdStorageService avviene attraverso la ServiceAccessGUI:

- l'utente effettua la richiesta del ticket
- se la richiesta è accettata, riceve il ticket
- entro un tempo prefissato inserisce il codice del ticket
- attende il messaggio **charge taken**

A fronte dei piani di testing, il camion refrigerato verrà rappresentato come un **attore**; questo ci permetterà di simulare il comportamento di molteplici utenti (camion) per verificare il funzionamento del sistema.

## Sistema di allarme (RaspberryPi)

Per trattare i requisiti di allarme, abbiamo individuato **sonar** e **led** come componenti appartenenti all'aggregato RaspberryPi, si è deciso di introdurre un ulteriore componente logico, un controller, incaricato di gestire il traffico per i due componenti sopra citati.

Dall'analisi emergono due eventi che il sonar deve comunicare:

1. rilevamento di una distanza inferiore a DLIMIT
2. rilevamento di una distanza superiore a DLIMIT

Dall'analisi dei requisiti è sorto il problema di scegliere se inviare i segnali relativi agli *alarm requirements direttamente al transport trolley o incaricare il CSS di inoltrarli al TT*, si è deciso di coinvolgere nella comunicazione solo il controller e il transport trolley. Il controller sarà incaricato di inviare al transport trolley un segnale di stop alla realizzazione del primo evento e un segnale di resume (riprendere il movimento) all'eventuale realizzazione del secondo evento.

Si specifica che questi eventi possono realizzarsi **molteplici volte**, ma sempre in successione alternata (evento 2 seguirà sempre l'evento 1).

Le comunicazioni inizieranno soltanto dopo la rilevazione del primo evento; a questo riguardo si suppone che l'avvio del sistema comporti la realizzazione dell'evento 2, senza tuttavia inviare il segnale di resume.

Relativamente ai requisiti del led, il controller riceverà dal transport trolley aggiornamenti sullo stato, che può assumere i seguenti valori logici:

- HOME;
- moving;
- stopped ( $\neq$  HOME),

il controller comunicherà poi al led di mostrare lo stato corrente nel rispetto degli alarm requirements.

## Transport trolley

Come già dedotto dall'analisi dei requisiti, il transport trolley è una entità astratta che aggiunge funzionalità di più alto livello a un DDR robot (come riportato dai requisiti), queste funzionalità sono:

- trasportare un carico di cibo da INDOOR a ColdRoom;
- recarsi in HOME quando non sono presenti richieste;
- comunicare il proprio stato a entità interessate.
- comunicare all'utente camion refrigerato che è stato prelevato il carico

Queste funzionalità condividono un requisito fondamentale: la possibilità di navigare la service area, sarà quindi necessario crearne una rappresentazione comprensibile al componente che controllerà il robot concreto.

L'analisi di questa problematica si rimanda a sprint successivi, abbiamo però individuato una serie di componenti che saranno necessari:

- **transport trolley** come astrazione di DDR robot;
- **mapper** per creare la rappresentazione della service area;

- **planner** per decidere il percorso (ottimale o meno) per raggiungere un punto nella rappresentazione della service area;
- **controller** che invia istruzioni di movimento comprensibili dal DDR robot;
- DDR robot concreto, fornito dal committente.

Queste funzionalità sono implementate dal software fornito dal cliente **basicrobot**: esso permette di muovere un robot generico (DDR robot, robot virtuale) seguendo percorsi preparati (non necessariamente ottimali) per spostarsi nella service area.

### Interazioni tra i componenti

Evidenziamo ogni possibile interazione tra i componenti, specificando la tipologia di messaggio tra quelle supportate dal linguaggio QAK (*request-reply, dispatch, event, ...*).

Abbiamo utilizzato sintassi prolog per il formato dei messaggi per mantenere la compatibilità con il metamodello che creeremo con QAK.

1. Camion refrigerato -> ColdStorageService
2. TransportTrolley <-> RaspberryPi
3. ColdStorageService <-> TransportTrolley

In questa tabella non comparirà l'entità camion refrigerato in quanto si tratta di un utente umano rappresentato da un attore solo in **fase di testing**. Si dà per scontato che esista una entità che interagisce con la ServiceAccessGUI allo scopo di scatenare una *storerequest*.

Mittente	Destinatario	Tipologia messaggio	Formato	Significato
ServiceAccessGUI	ColdStorageService	Request	storerequest(FW)	Richiesta di <b>storage</b> di <b>FW</b> chili di cibo
ColdStorageService	ServiceAccessGUI	Reply	storereply(TICKET,OKNO)	La richiesta è stata processata, può essere accettata o meno (OKNO) eventualmente fornisce il codice del ticket (TICKET)
ColdStorageService	ServiceAccessGUI	Dispatch	chargetaken(TICKET)	Il carico associato a TICKET è stato prelevato dal transport trolley, il camion può spostarsi da INDOOR
ColdStorageService	ServiceAccessGUI, ServiceStatusGUI, ...	Event	coldroomupdate(KG)	La quantità di chilogrammi nella ColdRoom è cambiata
ColdStorageService	ServiceStatusGUI, ...	Event	rejqreupdate(NREJ)	E' stata rifiutata un'ulteriore richiesta di deposito
ServiceAccessGUI	ColdStorageService	Request	codeentered(TICKET)	Il codice TICKET è stato inserito nella ServiceAccessGUI da un utente
ColdStorageService	ServiceAccessGUI	Reply	ticketreply(TICKET,OKNO)	La richiesta è stata processata, può essere



				accettata o meno (OKNO) eventualmente fornisce il codice del ticket (TICKET)
ColdStorageService	TransportTrolley	Dispatch	newcharge(FW)	E' presente alla INDOOR un nuovo carico da prelevare
TransportTrolley	ColdStorageService	Dispatch	chargetakentt()	Il carico presente alla INDOOR è stato prelevato dal transport trolley, il camion può spostarsi da INDOOR
TransportTrolley	ColdStorageService	Dispatch	chargedeposited(FW)	Il carico preso dal transport trolley è stato depositato nella ColdRoom, la SAG potrà aggiornare il peso corrente
TransportTrolley	ServiceStatusGUI, Sistema di allarme, ...	Event	statusupdate(POS, STATE)	Lo <b>stato</b> (posizione e/o stato di movimento) del transport trolley è cambiato
Sistema di allarme	TransportTrolley	Dispatch	stop()	La distanza del sonar da un ostacolo è minore di <b>DLIMIT</b>
Sistema di allarme	TransportTrolley	Dispatch	resume()	La distanza del sonar da un ostacolo è maggiore di <b>DLIMIT</b> , eventualmente solo a seguito di un segnale di <i>stop</i>

### Contesti e attori

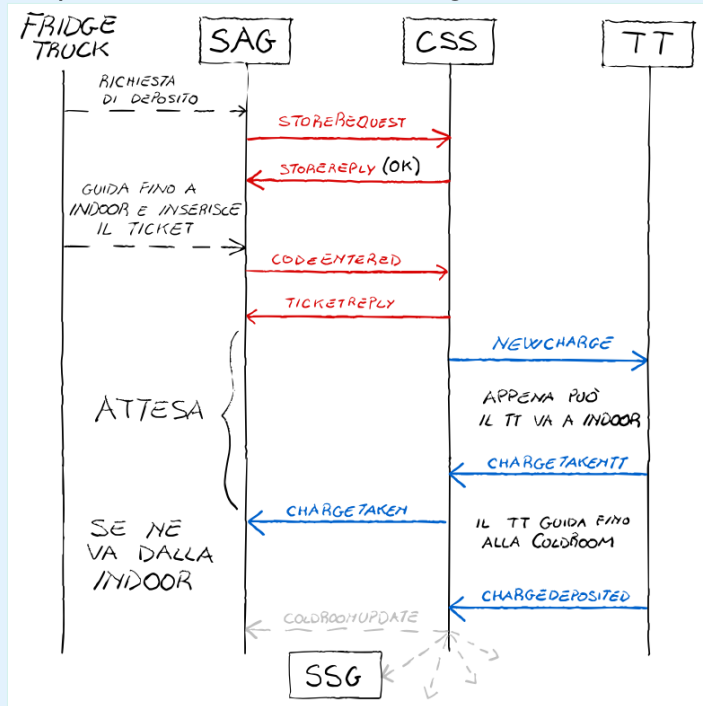
In questa tabella si esplicitano gli attori individuati dall'analisi precedente assegnandoli ai loro contesti; in futuri sprint questa suddivisione potrebbe cambiare a seguito di analisi più approfondite dei gruppi di componenti individuati.

Come si può notare non è presente il componente `controller` individuato nell'analisi del sistema di allarme: questo componente verrà analizzato in dettaglio in sprint futuri.

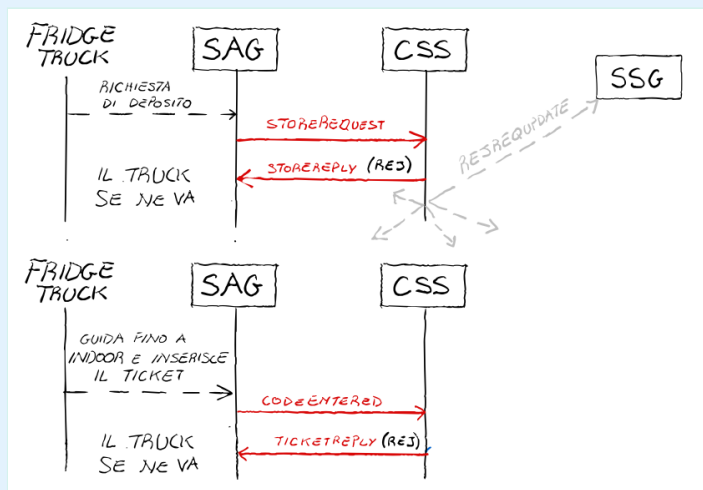
Contesto	Attore
ColdStorageService	ColdStorageService
	Transport trolley
Access	ServiceAccessGUI
Status monitor	ServiceStatusGUI
Sistema di allarme	WarningDevice
	AlarmDevice

Di seguito si riportano alcuni **diagrammi di sequenza** che ci aiutano a comprendere meglio l'ordine di invio e ricezione dei messaggi specificati.

### Sequenza 1. Richiesta di storage riuscita

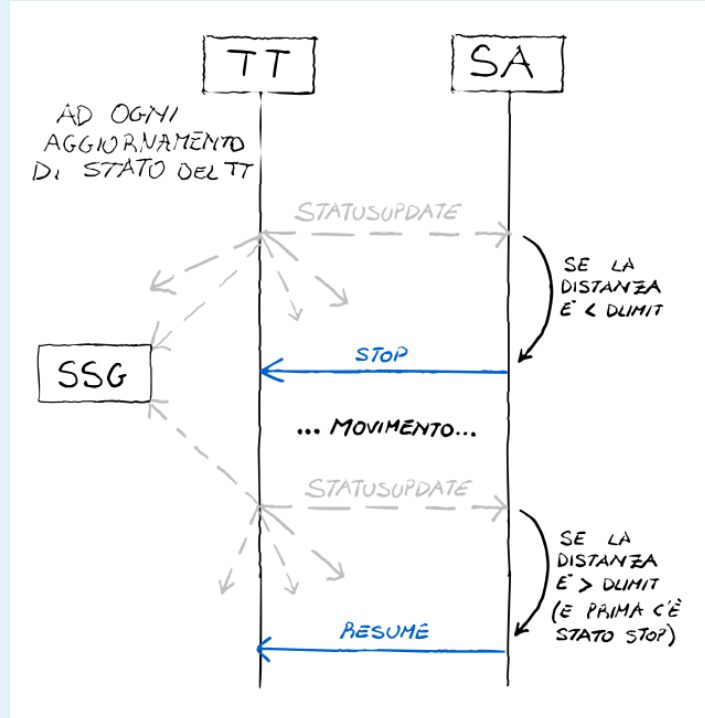


### Sequenza 2. Richiesta di storage fallita, TICKETTIME scaduto



### Sequenza 3. Interazione sistema di allarme con transport trolley





### Interazione TransportTrolley - BasicRobot

Con lo scopo di definire chiaramente il comportamento del TT e di descrivere i suoi stati e le sue transazioni in QAK, si è scelto di aggiungere un attore fittizio **basicrobotsim**.

Tramite esso abbiamo potuto quindi simulare il comportamento effettivo del TT e lo scambio di messaggi che avviene con il BasicRobot.

Di seguito, viene presentato un protocollo base di interazione implementato in QAK. In particolare vediamo i messaggi che si scambiano TT, BRS (basicrobotsim) e ColdStorageService, partendo dal primo messaggio *newcharge(FW)* inviato dal CSS:

1. TT -> BRS : *goToIndoor*
2. BRS -> TT : *arrivedIndoor*
3. TT -> CSS : *chargetakentt*
4. TT -> BRS : *goToColdRoom*
5. BRS -> TT : *arrivedColdRoom*
6. TT -> CSS : *chargedeposited*
7. TT -> BRS : *goToHome*
8. BRS -> TT : *arrivedHome*

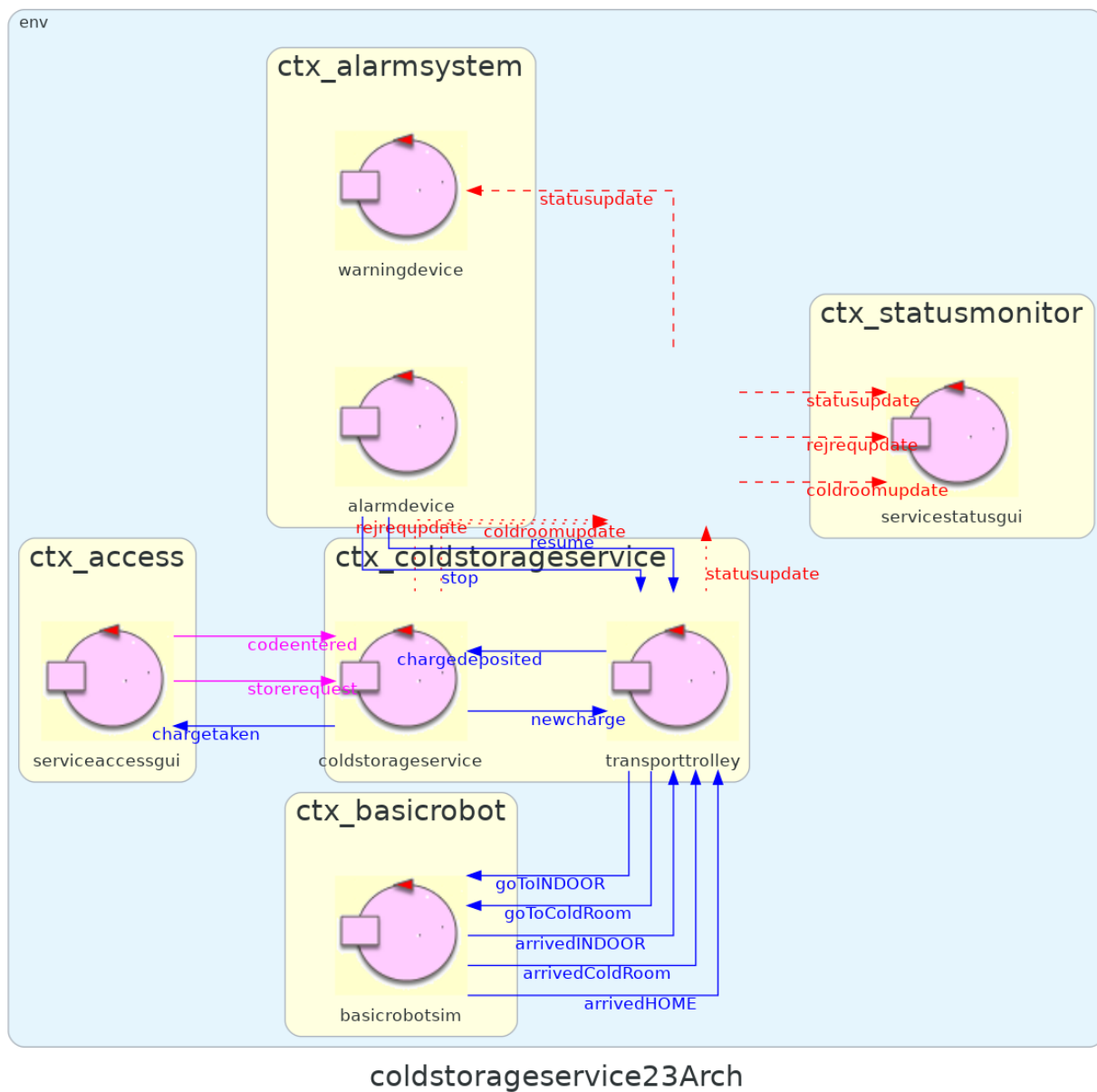
**NOTA:** questo schema di interazione è ridondante, a causa della natura del basicrobotsim, in sprint futuri verrà ridotto il numero di messaggi tra Transport trolley e basicrobot.

### Architettura logica

Di seguito verrà illustrata l'**architettura logica** preliminare del sistema creata sfruttando il linguaggio di metamodelizzazione QAK; i componenti illustrati non rispecchiano l'effettivo funzionamento di ColdStorageService ma **simulano** le azioni che scatenano le interazioni individuate.

In particolare non si mostrerà nel dettaglio l'interazione tra transport trolley e basicrobot (fornito dal cliente) ma verrà utilizzato un componente che simula il comportamento del robot, questo

sarà oggetto di futuri sprint.



## Test plans

GIT repo:  
<https://github.com/RiccardBarbieri/ColdStorageService>