

SerShare

Raport projektowy, wersja z 12.11.2018

Dziennik zmian:

1. 24.10.2018 - utworzenie dokumentu
2. 25.10.2018 - dodanie Danych projektu
3. 26.10.2018 - dodanie Części A
4. 12.11.2018 - dodanie Części B, modyfikacja Części A (zaznaczona na **żółto**)

1. Dane projektu:

- a. Tytuł projektu: SerShare (dawne MilkShare)
- b. Numer zespołu: Zespół A
- c. Skład z podziałem na role:
 - i. **Maciej Wiraszka** - kierownik projektu + specjalista ds. standardów FIPA
 - ii. **Julita Oltusek** - specjalistka ds. algorytmów agentowych
 - iii. **Szymon Borodziuk** - specjalista ds. architektury
 - iv. **Michał Starosta** - specjalista ds. jakości agentów
 - v. **Jędrzej Kalisiak** - specjalista ds. implementacji
 - vi. **Cezary Modzelewski** - specjalista ds. implementacji
- d. Repozytorium: <https://github.com/Zwirek009/SerShare>

2. Część A: Identyfikacja problemu

a. Problem

Ludzie marnują jedzenie, kupując produkty spożywcze, które wykorzystują nie do końca. Poza tym, marnują czas na robienie zakupów. Dodatkowo zanieczyszczają środowisko, bo każdy jedzie po zakupy, często własnym samochodem.

b. Propozycja rozwiązania

System zarządzania zasobami żywieniowymi w społeczności lokalnej (np. w budynku lub na osiedlu). System wykorzystuje połączone, inteligentne lodówki, które monitorują własną zawartość i na tej podstawie automatycznie zamawiają jedzenie przez internet. Produkty dostarczane są zbiorczo, okresowo. System umożliwia także wymianę produktów z sąsiadem, w przypadku kiedy nam coś zostało (np. wykorzystaliśmy niecałą zawartość kartonu z mlekiem), a sąsiadowi akurat było to potrzebne.

c. Koncepcja rozwiązania

Rodzaje agentów:

1. **Fridge Agent (agent lodówkowy)** - agent zajmujący się zapewnieniem wszystkich wymaganych towarów w lodówce i obsługą mechanizmów dzielenia się produktami.

Zadania:

- Monitoring aktualnego stanu jedzenia w lodówce
- Tworzenie planu zawartości danej lodówki
- Obsługa mechanizmu udostępniania jedzenia

Komunikacja z:

- Mobile Agent - uzyskiwanie informacji o planowanym zapotrzebowaniu na produkty
- Storekeeper Agent - wymiana informacji o przewidywanej ilości produktów w lodówce w czasie, informowanie o aktualnych zapasach.
- Fridge Agent - uzgadnianie wymian produktów.

Cel:

- Agent dąży do tego aby lodówka posiadała wymagane produkty

Umiejscowienie:

- Agent lodówkowy będzie zainstalowany na każdej lodówce w ilości jeden

2. **Mobile Agent (agent mobilny)** - agent wydzielony tylko do wprowadzania zmian w planie jedzeniowym lub do zgłaszania potrzeb in-time

Zadania:

- Pobieranie od użytkownika zmian planu jedzeniowego
- Pobieranie od użytkownika zgłoszeń in-time

Komunikacja z:

- Fridge Agent - wysłanie informacji o zmianach planu jedzeniowego i zgłoszeń in-time

Cel:

- Przekazanie agentowi Fridge informacji o planach zakupowych oraz nagłych potrzebach

Umiejscowienie:

- Urządzenia mobilne

3. **Storekeeper Agent** - agent zajmujący planowaniem zakupów produktów na podstawie stanu lodówki oraz planów żywieniowych.

Zadania:

- Zbieranie planów żywieniowych
- Planowanie zakupów
- Zbieranie informacji o stanie lodówki

Komunikacja z:

- Fridge Agent - wymiana informacji o aktualnych zasobach
- Merchant Agent - składanie zamówień na zakupy
- Mobile Agent - pobieranie informacji o planach żywieniowych

Cel:

- Zapewnienie dostępności produktów ustawionych w planach żywieniowych

Umiejscowienie:

- Każda lodówka

4. **Merchant Agent** - agent reprezentujący sklep

Zadania:

- Zbieranie zamówień od wielu Storekeeper Agent
- Planowanie optymalnych zakupów
- Zamawianie produktów w sklepach (przy tym wybór sklepów)

Kontakty:

- Storekeeper Agent - wymiana informacji o aktualnych potrzebach i planowanych dostawach jedzenia

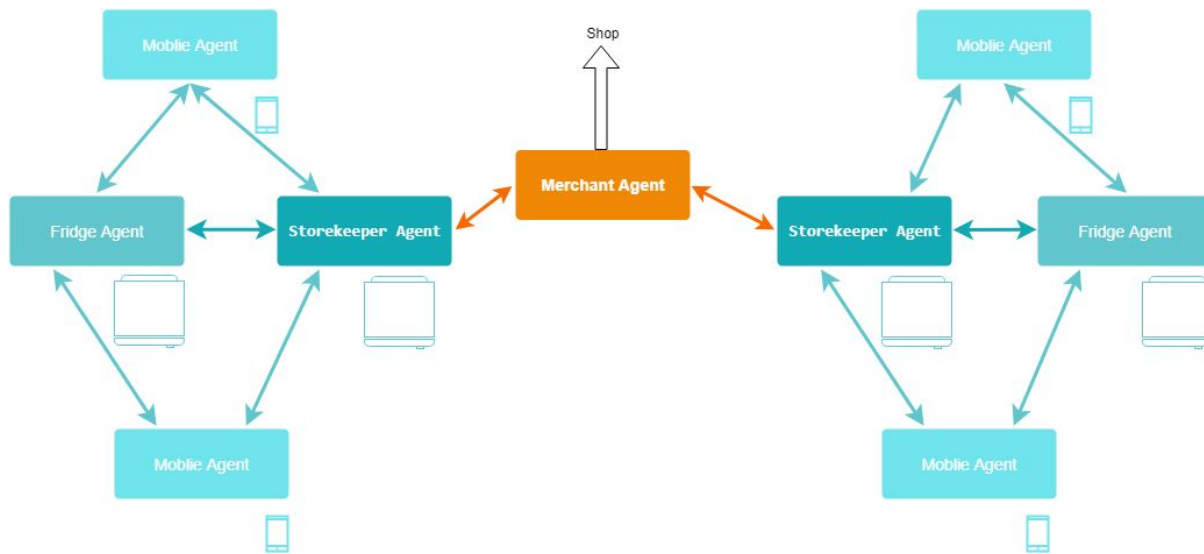
Cel:

- Planowanie jak najrzadszych i najlepiej dopasowanych czasowo zakupów na podstawie zamówień Storekeeper Agentów

Umiejscowienie:

- Chmura

Architektura rozwiązania:



3. Część B: Projekt systemu wieloagentowego (w metodyce GAIA)

a. Identyfikacja ról

- i. **FridgeStateController** - monitorowanie stanu lodówki, czyli ilości produktów w jej wnętrzu.
- ii. **ProductDemandPlanner** - planowanie zapotrzebowania na produkty.
- iii. **MinimumSupplyGuard** - dbanie o zapewnienia produktów dla lodówki w krótkim okresie czasu.
- iv. **SharingHandler** - zarządzanie wymianą produktów pomiędzy klientami.
- v. **OrderManager** - zbieranie zamówień i ich optymalizacja.
- vi. **Customer** - układanie planów żywieniowych.

b. Model ról

FridgeStateController

Aktywności:

- CheckFridgeInternals - pobieranie danych z lodówki za pomocą czujników.

Protokoły:

- GetFridgeInternalStateRequest- zapytanie o stan zawartości lodówki.
- SendFridgeInternalStateResponse - informowanie o aktualnym stanie zawartości lodówki.

Role schema: <i>FridgeStateController</i>
Description: Zadaniem tej roli jest monitorowanie stanu lodówki, czyli ilości produktów oraz terminu ich ważności, a także informowanie o swoim stanie.
Protocols and Activities: GetFridgeInternalStateRequest, <u>CheckFridgeInternals</u> , SendFridgeInternalStateResponse
Permissions: generates: fridge_internal_state - informacja o produktach w lodówce
Responsibilities: Liveness: <i>FridgeStateController</i> = (GetFridgeInternalStateRequest . <u>CheckFridgeInternals</u> . SendFridgeInternalStateResponse) ^w Safety: <ul style="list-style-type: none">• true

ProductDemandPlanner

Aktywności:

- EstimateFridgeStatePlan - estymowanie zapasów lodówki w czasie, z uwzględnieniem planów żywieniowych właścicieli.

Protokoły:

- GetEstimatedFridgeStatePlanRequest - odbieranie żądania przekazania planu lodówkowego.
- SendEstimatedFridgeStatePlanResponse - przekazanie planu lodówkowego.
- SendFridgeInternalStateRequest - pytanie o aktualny stan produktów w lodówce.
- SendFoodPlanRequest - pytanie o plany żywieniowe właścicieli.

Role schema: <i>ProductDemandPlanner</i>
Description: Zadaniem tej roli jest wyznaczanie zapotrzebowania (zakupy) na produkty na podstawie planów żywieniowych i obecnego stanu lodówki. Określa zarówno spodziewane braki produktów jak i potencjalne ich zapasy.
Protocols and Activities: GetEstimatedFridgeStatePlanRequest, SendFridgeInternalStateRequest , SendFoodPlanRequest , <u>EstimateFutureFridgeStatePlan</u> , SendEstimatedFridgeStatePlanResponse
Permissions: reads: fridge_internal_state - informacja o dostępnych produktach food_plans - plany żywieniowe od wszystkich właścicieli lodówki generates: future_fridge_state_plan - informacja o zapasach i brakach produktów w lodówce
Responsibilities: Liveness: <i>ProductDemandPlanner</i> = (GetEstimatedFridgeStatePlanRequest . SendFridgeInternalStateRequest . SendFoodPlanRequest * . <u>EstimateFridgeStatePlan</u> . SendEstimatedFridgeStatePlanResponse) ^w Safety: • true

MinimumSupplyGuard

Aktywności:

- CalculateShortTermProductDemand - wyznaczenie zapotrzebowania na najbliższy czas (np. dzień)

Protokoły:

- SendEstimatedFridgeStatePlanRequest - zapytanie do *ProductDemandPlanner* o plan zapotrzebowania
- SendProductShareRequest - zapytanie do sąsiadów o pożyczenie produktów.

Role schema: <i>MinimumSupplyGuard</i>
Description: Zadaniem tej roli jest zapewnienie produktów potrzebnych w najbliższym czasie, stosując mechanizm pożyczania.
Protocols and Activities: SendEstimatedFridgeStatePlanRequest, <u>CalculateShortTermProductDemand</u> , SendProductShareRequest
Permissions: reads: future_fridge_state_plan - informacja o zapasach i brakach produktów w lodówce generates: short_term_product_demand - aktualne zapotrzebowanie na produkty
Responsibilities: Liveness: $MinimumSupplyGuard = (SendEstimatedFridgeStatePlanRequest . \underline{CalculateShortTermProductDemand} . \underline{SendProductShareRequest}^*)^\omega$ Safety: true

SharingHandler

Aktywności:

- CheckIfCanShare - sprawdzenie czy można pożyczyć produkt bez naruszenia swojego planu.

Protokoły:

- GetProductShareRequest - oczekiwanie na żądanie produktu.
- SendEstimatedFridgeStatePlanRequest - zapytanie o plan stanu lodówki.
- SendCustomerShareRequest - prośba o podzielenie się produktem.
- SendProductShareResponse - informacja o produktach, które udało się uzyskać.

Role schema: <i>SharingHandler</i>
Description: Zadaniem tej roli jest zadbanie o udostępnienie produktów nadmiarowych, o ile klient wyrazi na to zgodę.
Protocols and Activities: GetProductShareRequest , <u>CheckIfCanShare</u> , GetEstimatedFridgeStatePlan, SendCustomerShareRequest, SendProductShareResponse
Permissions: reads: short_term_product_demand - aktualne zapotrzebowanie na produkty customer_share_response - zgoda/odmowa właściciela future_fridge_state_plan - informacja o zapasach i brakach produktów w lodówce generates: fridge_state_balance - informacja o ilości produktów do wymiany, products_shared - informacja o produktach, które zostaną wymienione
Responsibilities: Liveness: SharingHandler = (GetProductShareRequest . CheckSharePossible . SendProductShareResponse) CheckSharePossible = (SendEstimatedFridgeStatePlanRequest . <u>CheckIfCanShare</u> . [SendCustomerShareRequest]) Safety: fridge_state_balance <= 0 ⇒ products_shared = EMPTY customer_share_response = false ⇒ products_shared = EMPTY

OrderManager

Aktywności:

- PrepareMassOrder - przygotowanie zamówienie, wraz z podziałem produktów na odbiorców.
- OrderProducts - zamawianie produktów ze sklepu wraz z podziałem dla kuriera.

Protokoły:

- SendEstimatedFrigdeStatePlanRequest - prośba o przesłanie planów lodówkowych.

Role schema: <i>OrderManager</i>
Description: Zadaniem tej roli jest planowanie zamówień produktów ze sklepu oraz ich optymalizacja.
Protocols and Activities: GetEstimatedFrigdeStatePlan, <u>PrepareMassOrder</u> , <u>OrderProducts</u>
Permissions: reads: future_fridge_state_plan - informacja o zapasach i brakach produktów w lodówce generates: order_list - lista z zamówieniem
Responsibilities: Liveness: $OrderManager = (SendEstimatedFrigdeStatePlanRequest^* . \underline{PrepareMassOrder} . \underline{OrderProducts})^{\omega} (per-day)$ Safety: true

Customer

Aktywności:

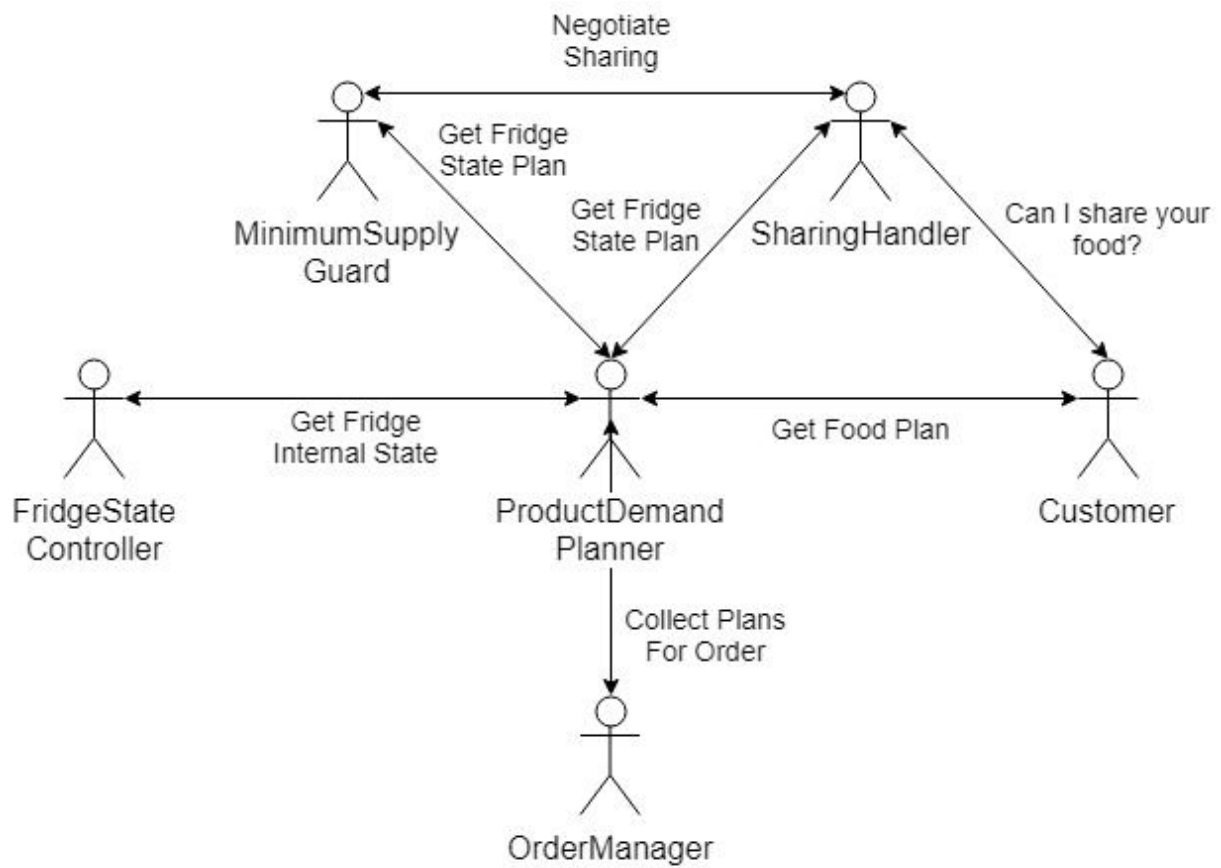
- CheckFoodPlan - pobieranie planu żywieniowego użytkownika
- CheckUserAgreement - pytanie użytkownika o pozwolenie

Protokoły:

- GetShareRequest - czekanie na zapytanie o zgodę na podzielenie się produktami.
- AcceptOrRefuseShare - deklaracja zgody na podzielenie się produktem.
- GetFoodPlanRequest - czekanie na żądanie przesłania planu
- SendFoodPlanResponse - przesłanie planu zapotrzebowania na produkty.

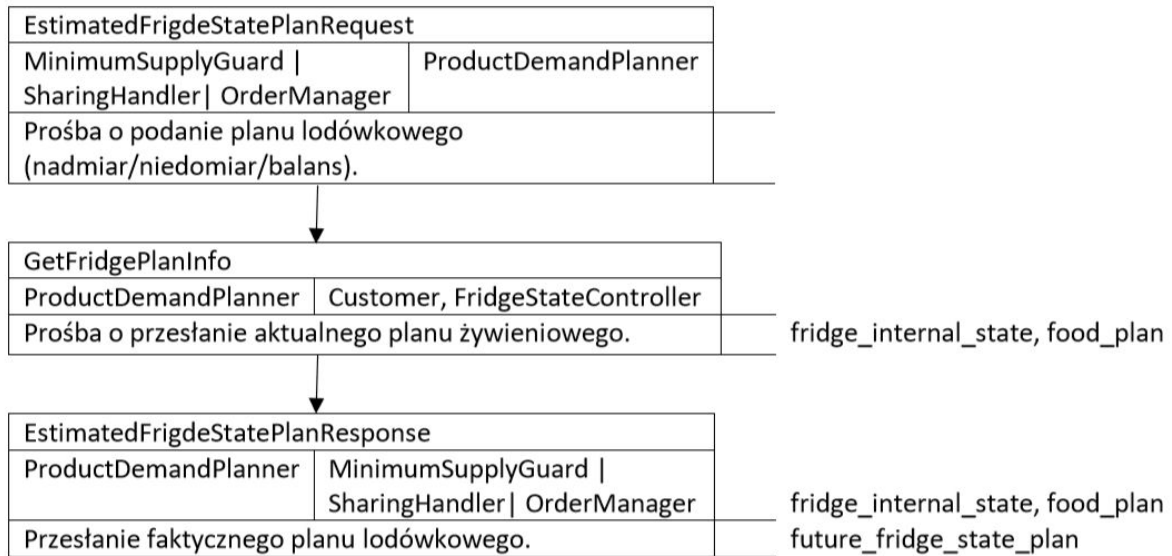
Role schema: <i>Customer</i>
Description: Zadaniem tej roli jest odpowiadanie na zapytanie o plan żywieniowy właściciela i odpowiadanie na zapytanie o podzielenie się produktem.
Protocols and Activities: GetFoodPlanRequest, <u>CheckFoodPlan</u> , SendFoodPlanResponse , GetShareRequest, <u>CheckUserAgreement</u> , AcceptOrRefuseShare
Permissions: generates: food_plan - plan żywieniowy customer_share_response - zgoda/odmowa użytkownika
Responsibilities: Liveness: $Customer = (SendFoodPlan \parallel SendAgreementForShare)^{\omega}$ $SendFoodPlan = (GetFoodPlanRequest . \underline{CheckFoodPlan} .$ $SendFoodPlanResponse)$ $SendAgreementForShare = (GetShareRequest .$ $\underline{CheckUserAgreement} . AcceptOrRefuseShare)$ Safety: true

c. Model interakcji

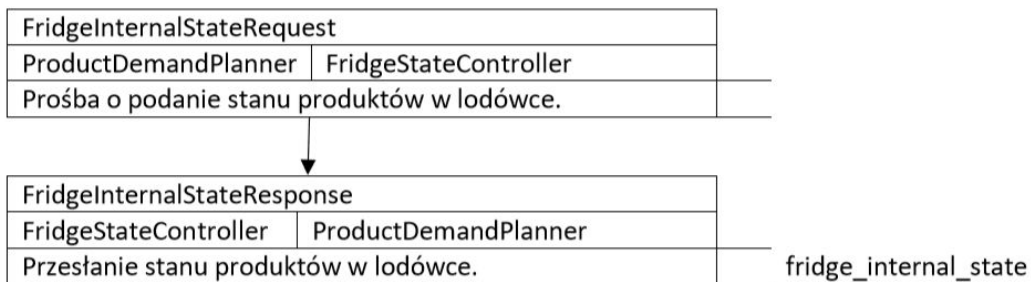


Definicje protokołów

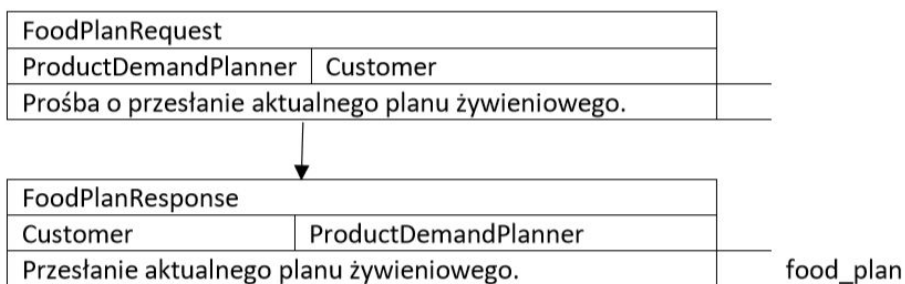
Protokół: GetFridgeStatePlan.



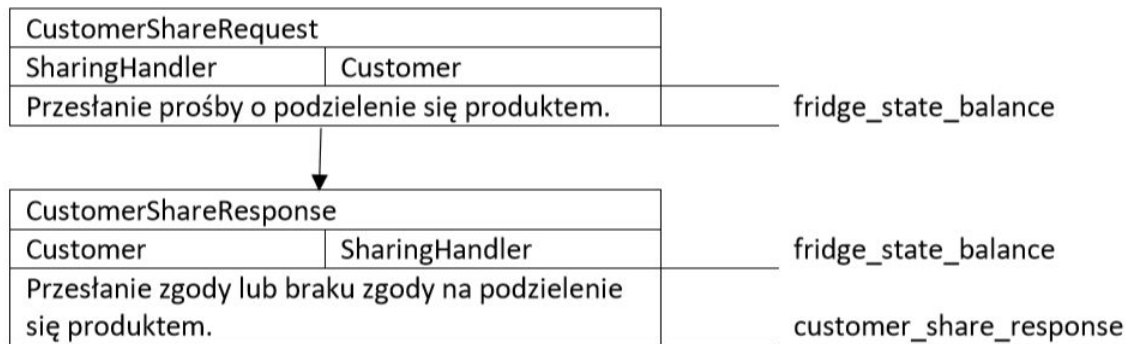
Protokół: GetFridgeInternalState.



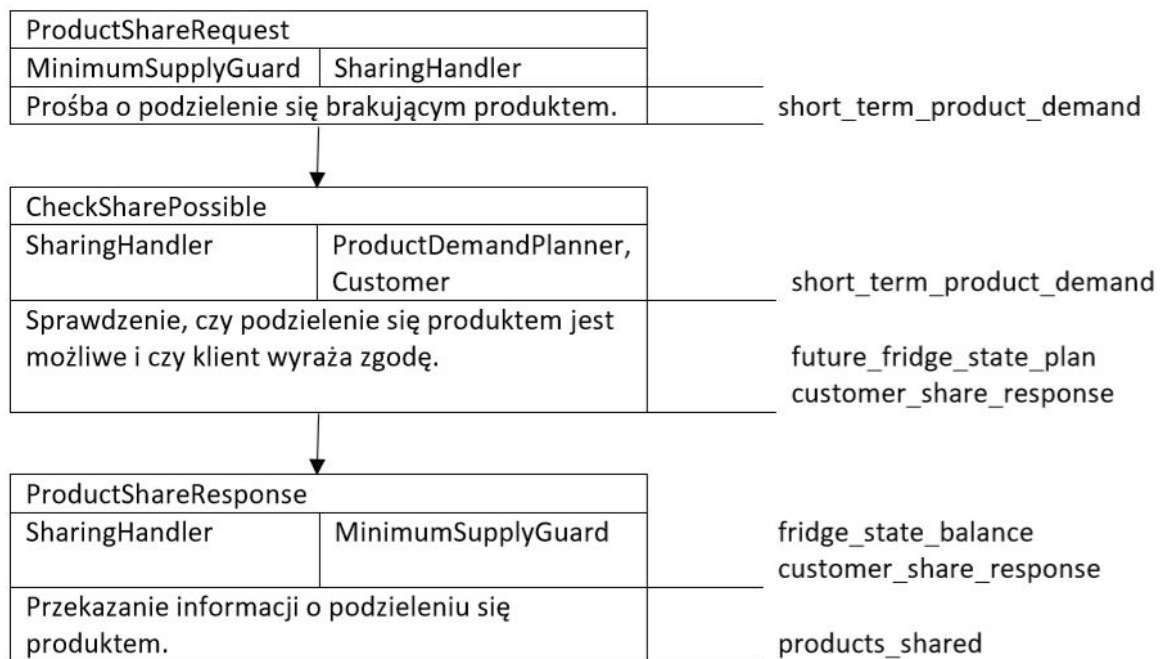
Protokół: GetFoodPlan.



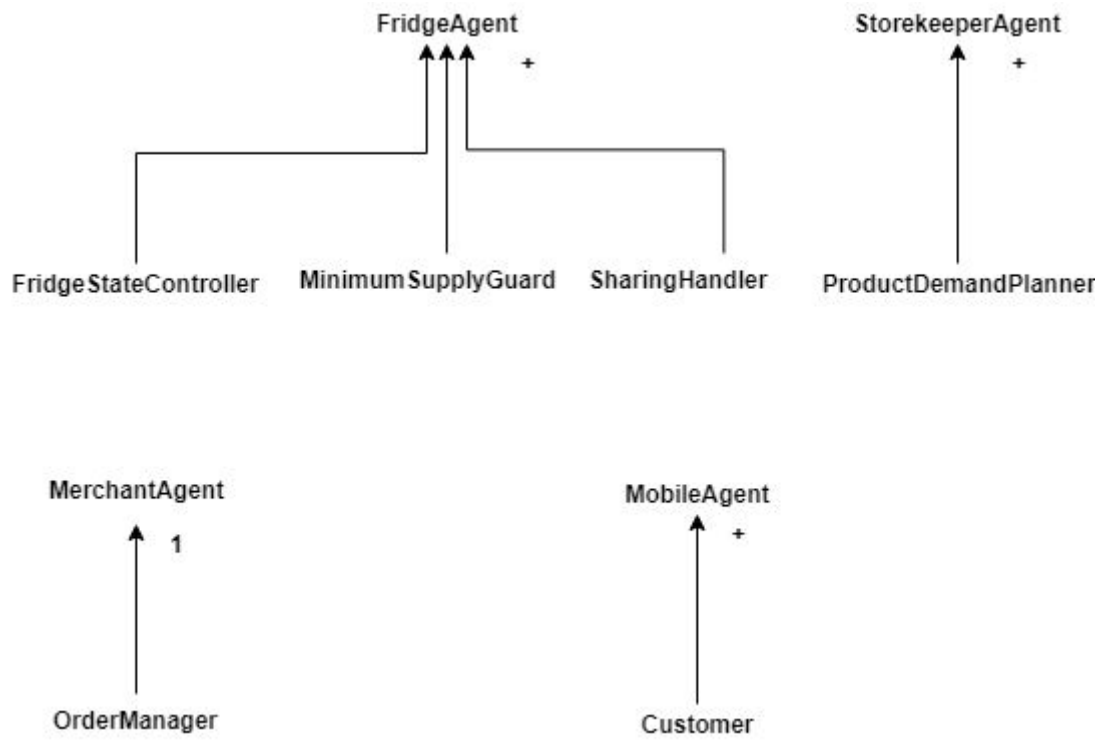
Protokół: GetAgreementForShare.



Protokół: NegotiateShare.



d. Model agentów



e. Model usług

Usługa	Wejścia	Wyjścia	Warunki wstępne	Warunki końcowe
ustalenie planu zapotrzebowania produktów	fridge_internal_state food_plans	future_fridge_state_plan	food_plans != NULL	future_fridge_state_plan != NULL
sprawdzanie stanu produktów	wyniki z kontrolerów	fridge_internal_state	true	true
ustalenie dziennego zapotrzebowania	future_fridge_state_plan	short_term_product_demand	true	short_term_product_demand != NULL
udostępnianie produktów	short_term_product_demand customer_share_response future_fridge_state_plan	fridge_state_balance products_shared	fridge_state_balance <= 0 ⇒ products_shared = EMPTY customer_share_response = false ⇒ products_shared = EMPTY	fridge_state_balance <= 0 ⇒ products_shared = EMPTY customer_share_response = false ⇒ products_shared = EMPTY
przygotowanie zamówienia	future_fridge_state_plan	order_list	true	order_list != NULL
udostępnienie planu żywieniowego	-	food_plan	true	food_plan != NULL
ustalenie pozwolenia na udostępnianie	-	customer_share_response	true	true

f. Model znajomości

