

Dağıtık Sistemler ve Uygulamaları Dönem Projesi

Proje Raporu

Erem Karalar
17401776

Ece Yağdıran
17401756

Döne Hilal Şenkul
17401762

Deniz Edizcan
18401809

25.01.2022

Genel bakış

Bu proje INF 443 Dağıtık Sistemler ve Uygulamaları adlı ders kapsamında gerçekleştirilmektedir. Projenin ana amacı IoT adı altında akıllı sistemlerin arz/taleplerini belirleyip, sistemlere yönelik ihtiyaçların karşılanmasıdır.

Özellikler

Sistemin temel özelliği özel koşullar harici isteğin her zaman istemciden yapılması, cevabın ise her zaman sunucu tarafından gelmesidir. Bu mesajlaşmanın senkron bir şekilde gerçekleşmesi sağlanacaktır. Sistemler peer olarak davranıp (yani hem server hem client) bağlantı kuracaklardır. Bazı sistemlerin senaryo gereği birden fazla istemci kısmı da bulunmaktadır. Bu iletişim socketler aracılığıyla gerçekleştirilmektedir.

Protokol Tasarımı

Peerlar arasında iletişimi tam olarak sağlayabilmek için protokol tasarımı yapılmıştır. Protokolün tasarlanması ve özelliklerin alt başlıkları aşağıdaki gibidir:

I. Kendini Tanıtma

Peerların yapıya dahil olabilmesi için kendilerini tanıtmalara gerekmektedir. Bu tanıtım şöyle gerçekleşir:

RG::<UUID>::<IP adresi>::<Port>::<GPS Koordinatları>::<Tip>::<Kısa Bilgi>

Bağlanmak isteyen UUID Peer'a gelen UUID ile aynıysa bağlantı başarılı mesajı gönderilir.

RO

UUID'ler aynı değilse red mesajı şu şekilde döndürülür:

RN

II. Arz Sorgulama

Arz sorgusu opsiyonel parametrelere göre sınırlandırılabilir veya ek hiçbir parametre koymayarak (Sistem UUID harici) tüm malzeme listesi alınabilir:

[OF][::N::< Num>]

N = Belirli bir sayıda sorgu yapmak istediğimiz durumda kullandığımız opsiyonel parametre

Sorgulanan arzların gönderilme şekli:

OO::BEGIN

OO::<arz uid>::<arz adı>::<arz birimi>::<arz miktarı>::<karşılık adı>::<karşılık birimi>::<asgari karşılık miktarı>

....

OO::END

Sorgu gönderilmezse:

RN

III. Talep Sorgulama

Sorgulanan taleplerin gönderilme şekli:

DM::N::<sayı>

Çıktılar şöyle görünecektir:

DO::BEGIN

DO::<talep uid>::<talep adı>::<talep birimi>::<talep miktarı>::<karşılık adı>::<karşılık birimi>::<azami karşılık miktarı>

....

DO::END

Talep gönderilmezse:

RN

IV. Arz-Talep Sorgulama

Bağlanılan peer'ın tüm arz ve taleplerini görüntülemek için gönderilen sorgu

AT::

Sorgulanan arz ve taleplerin gönderilme şekli:

UB::BEGIN

UB::T::<talep uid>::<talep adı>::<talep birimi>::<talep miktarı>::<karşılık adı>::<karşılık birimi>::<asgari karşılık miktarı>

...

UB::END

UB::BEGIN

UB::A::<arz uid>::<arz adı>::<arz birimi>::<arz miktarı>::<karşılık adı>::<karşılık birimi>::<asgari karşılık miktarı>

...

UB::END

V. Yeni Üyelik

Yeni üyelik aşağıdaki gibi gerçekleşmektedir:

SB::T

Yeni üyelik kabul edilince gelecek yanıt:

SO

Yeni üyelik reddedilince gelecek yanıt:

RN

Üyelikten çıkmak için:

SB::F

Üyelikten çıkıldığına dair bilgi mesajı:

SO

VI. Alışveriş Başlatma-Transaksiyon

Tanıdığı diğer ev sistemlerinden arz veya talep listesi almak için yollanan komut:

TR::<O veya D><arz/talep uid>::<karşılık adı>::<karşılık birimi>::<verilecek/istenen karşılık miktarı>

Onay durumunda dönecek yanıt:

TO

Red durumunda dönecek yanıt:

TN

Kayıtlı olmama durumunda dönecek yanıt:

RN

VII. Mesajlaşma

Bir peer genel olarak bağlı olduğu tüm peerlara mesaj gönderirken:

MS::<mesaj>

Başarılı olursa yanıt olarak şu döndürülür:

MO

Başarısız olursa yanıt olarak:

RN

VIII. Bağlantı Kontrolü

Bağlantının sürekli ve devamlı olduğunu görebilmek için istemci otomatik olarak şu sorguyu atar:

IG::<UUID>

Sunucu yanıt olarak şunu döndürür:

OG::<UUID>

IX. Hatalı Komut Girişi

Herhangi bir hatalı komut girişinde şu yanıt döndürülür:

ER

Senaryo

Tasarlanacak akıllı sistemler yenilenmiştir. Projenin yeni senaryosunda T1 ve T2 tanıştırıcı sistemlerin hardcoded olarak tanıdığı ev sistemleri, tüketiciler ve ürün sağlayanlar olarak ikiye ayrılmıştır.

Bütün sistemlerin talep ve arzlarını tutmak için ise JSON dosya formatı tercih edilmiştir. Bu senaryoda akıllı sistemlerin bağlantı şekilleri de aşağıda belirtilmiştir:

İki tanıştırıcı sistem birbirini tanımaktadır. Bu sistemlere bağlı olan peerlar (ayrı ayrı bağlantılar) tanıştırıcı sistem üzerinden birbirleriyle bilgi alışverişi yapmaktadırlar. İlk önce kendilerini tanıştırıcı sistemlere tanıtmaları gerekmektedir. Abonelik de gerçekleştirildikten sonra bilgi akışı veya transaksyon gerçekleşebilir. Bunun dışında senaryo gereği iki akıllı sistem (A1 ve A5) peer'ı birbirini tanıtıcı sistem olmadan tanımaktadır ve bilgi akışını veya transaksyon işlemlerini gerçekleştirmektedir. Akıllı sistemlerle alakalı detaylar aşağıdaki kısımda belirtilmiştir.

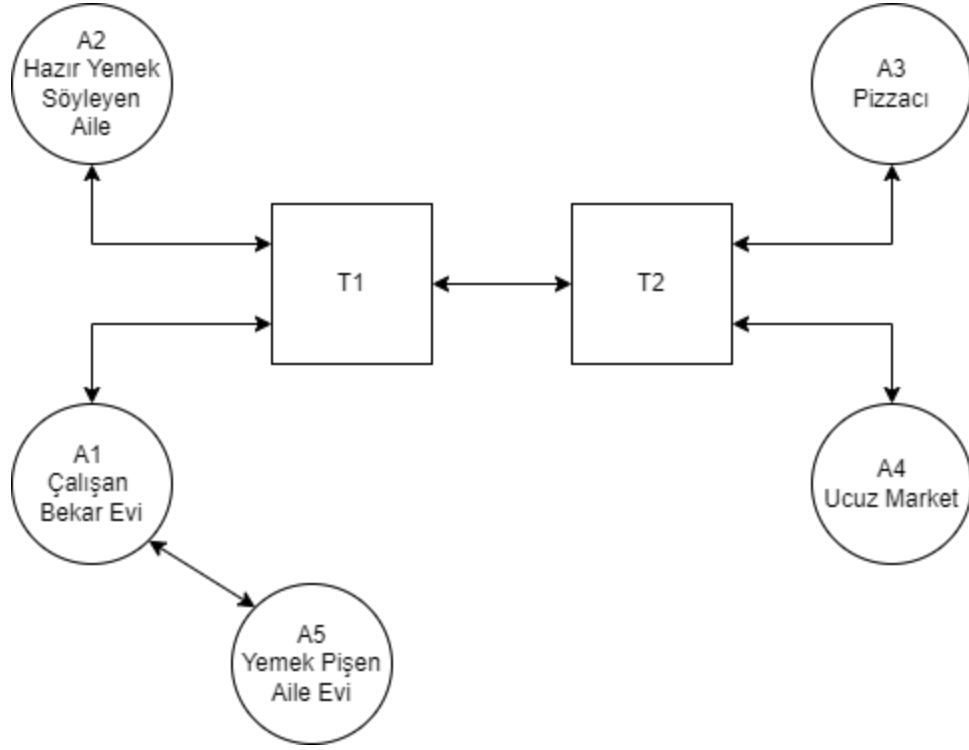
Akıllı Sistemler

I. Tüketiciler

T1 tanıştırıcı sistemin bağlı olduğu A1 akıllı sistemi çalışan bekar evini, A2 akıllı sistemi hazır yemek söyleyen aileyi temsil etmektedir. A1 akıllı sisteminin tanıştırıcı sistemden bağımsız tanıdığı A5 akıllı sistemi ise yemek pişen bir aile evini temsil etmektedir.

II. Ürün sağlayanlar

T2 tanıştırıcı sisteminin bağlı olduğu A3 pizzacıyken A4 akıllı sistemi ucuz markettir. Ayrıca A5 sistemi zaman zaman A1'e de ürünler sağlamaktadır. Senaryo bazlı sistemin gösteriminin yapıldığı grafik aşağıda belirtilmiştir.



Veriler

Veriler bir önceki bölümde bahsedildiği gibi JSON formatında tutulmaktadır. Tüm sistemin bileşenleri gerekli JSON dosyalarını okuyup transaksyon halinde dosyalar üzerinde güncelleme yapılabilmektedir.

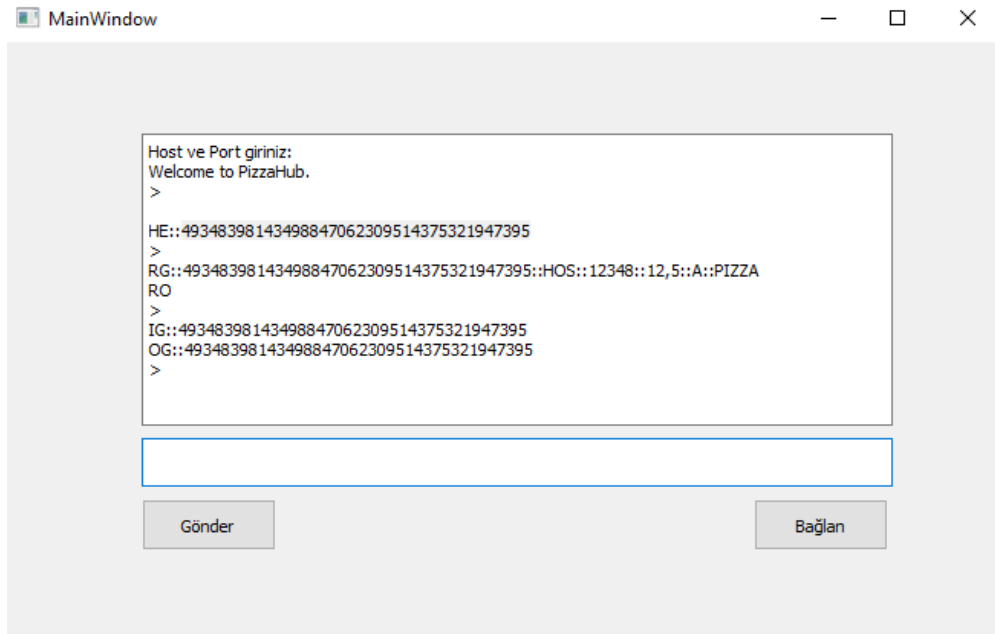
JSON dosyalarında iki temel bölüm bulunmaktadır. Bunlar “offers” yani peer’ın teklif edeceği ve arz edeceği ürünler veya hizmetler, “demands” yani peer’ın talep edeceği ürün ve hizmetlerdir.

Okuma işlemleri genelde talepler üzerine yapılırken güncellemeler genelde arz edilen ürünler veya hizmetler üzerinde yapılmaktadır. Örnek JSON dosyası aşağıda gösterildiği gibidir.

```
{
  "offers":[
    {
      "id": "1",
      "name":"tuna",
      "unit":"slice",
      "quantity": 1000,
      "payment": "cash",
      "currency": "TRY",
      "price": 60
    },
    {
      "id": "2",
      "name":"margarita",
      "unit":"slice",
      "quantity": "1000",
      "payment": "cash",
      "currency": "TRY",
      "price": 40
    },
    {
      "id": "3",
      "name":"pepperoni",
      "unit":"slice",
      "quantity": "1000",
      "payment": "cash",
      "currency": "TRY",
      "price": 60
    }
  ],
},
{
  "demands":[
    {
      "id": "1",
      "name":"flour",
      "unit":"gr",
      "quantity": 1000,
      "payment": "credit",
      "currency": "TRY",
      "price": 20
    },
    {
      "id": "2",
      "name":"salt",
      "unit":"gr",
      "quantity": 100,
      "payment": "TRY",
      "currency": "credit",
      "price": 12
    },
    {
      "id": "3",
      "name":"vegetable",
      "unit":"gr",
      "quantity": 1000,
      "payment": "TRY",
      "currency": "credit",
      "price": 10
    }
  ],
},
{
}
```

Arayüz

Arayüz tasarımı Qt Designer programından yapıлып python dosyalarına transformasyonu gerçekleştirilmiştir. Arayüz ile bağlantı ve komut gönderimleri gerçekleştirilmektedir. Örnek arayüz aşağıda gösterildiği gibidir.



MainWindow

UB::A::5::hawaiian::slice::1000::cash::TRY::50
UB::A::6::four_cheese::slice::1000::cash::TRY::40
UB::END
UB::BEGIN
UB::T::1::flour::gr::1000::credit::TRY::20
UB::T::2::salt::gr::100::TRY::credit::12
UB::T::3::vegetable::gr::1000::TRY::credit::10
UB::T::3::pepperoni::gr::500::TRY::credit::40
UB::END

>
TR::O::1::credit::TRY::20
OK
>

Gönder

Bağlan