İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

GÜVENLİ ETMEN SİSTEMİ KULLANARAK E-BORSA TASARIMI VE GERÇEKLENMESİ

YÜKSEK LİSANS TEZİ

Müh. Cevher Cemal BOZKUR

Anabilim Dalı : BİLGİSAYAR MÜHENDİSLİĞİ

Programı : BİLGİSAYAR MÜHENDİSLİĞİ

EKİM 2007

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

GÜVENLİ ETMEN SİSTEMİ KULLANARAK E-BORSA TASARIMI VE GERÇEKLENMESİ

YÜKSEK LİSANS TEZİ
Müh. Cevher Cemal BOZKUR
504041508

Tezin Enstitüye Verildiği Tarih : 24 Eylül 2007
Tezin Savunulduğu Tarih : 04 Ekim 2007

Tez Danışmanı : Prof.Dr. Nadia ERDOĞAN

Diğer Jüri Üyeleri Prof.Dr. Çoşkun SÖNMEZ (Y.T.Ü.)

Yrd.Doç.Dr. A. Şima ETANER-UYAR (İ.T.Ü.)

EKİM 2007

ÖNSÖZ

Tez çalışmam süresince desteklerini benden esirgemeyen ve yol gösteren sayın hocam Prof. Dr. Nadia Erdoğan'a, teknik konularda yardımlarını esirgemeyen sayın Dr. Suat Uğurlu'ya, sabırla tez çalışmamı başarıyla bitirmemi bekleyen ve her zaman maddi ve manevi destekleri ile yanımda olan aileme –özellikle ANNEME- teşekkürlerimi sunarım.

Ekim 2007 Cevher Cemal BOZKUR

İÇİNDEKİLER

KISALTMALAR **v**
TABLO LİSTESİ **vi**
ŞEKİL LİSTESİ **vii**
ÖZET **viii**
SUMMARY **ix**

1. GİRİŞ **1**
1.1. Problemin İncelenmesi 1
1.2. Çözüm Önerisi 2

2. ETMEN VE HAREKETLİ ETMENLER **3**
2.1. Etmen Tanımı 3
2.2. Etmen Tabanlı Yazılımlar 3
2.2.1. Etmen Tabanlı Yazılım Nedir? 3
2.2.2. Etmen Tabanlı Yazılım Yaşam Döngüsü 3
2.3. Hareketli Etmenler 4
2.3.1. Hareketli Etmenlerin Sağladığı Yararlar 6
2.3.2. Mevcut Hareketli Etmen Sistemleri 7
2.3.3. Hareketli Etmen Sistemlerinin Güvenliği 8
2.4. Etmen Sistemleri ve Elektronik Ticaret 9
2.4.1. E-ticaret Etmenlerinin Temel Karakteristikleri 9
2.4.2. Etmenlerin Kullanıldığı E-ticaret Uygulamaları 10

3. ETMEN SİSTEMLERİ İÇİN JAVA DİLİNİN KULLANILMASI **13**
3.1. Otonomluk 13
3.2. Hareketlilik 13

4. GÜVENLİ ETMEN SİSTEMİ (GES) **16**
4.1. Güvenli Etmen Sistemi Mimarisi 17
4.1.1. GES Sunucusu 17
4.1.2. GES Etmenleri 19
4.2. Etmen Yazılımı Geliştirme Arayüzü 21
4.3. GES Haberleşme Altyapısı 23
4.4. GES Güvenlik Politikaları ve Yönetimi 25

5. BORSA YAPISI **27**
5.1. Borsa Türleri 27
5.2. Gereksinim Duyulan Borsa Karakteristikleri 27
5.3. İstek Türleri 27
5.4. Tacir Türleri 28
5.5. Borsada İsteklerin Eşleştirilmesi 28

6. E-BORSA SİSTEMİ **29**

6.1. E-Borsa Tasarımı 29
6.1.1. Uygulama Etki Alanı (domain) Kavramsal Sınıfları 29
6.1.2. Sistemin GES Sistemi Kullanılarak Modellenmesi 32
6.2. E-Borsa Etmenleri 35
6.2.1. Tacir (Kullanıcı) Etmeni 36
6.2.2. Broker Etmeni 39
6.2.3. Borsa Etmeni 44
6.2.4. Yönetici Etmeni 49
6.2.5. VT (Veritabanı)Etmeni 51
6.2.6. Etmen Politikaları 52
6.3. E-Borsa Kullanıcı ve Yönetim Arayüzleri 53
6.3.1. Kullanıcı Arayüzü 54
6.3.2. Yönetim Arayüzü 56

7. E-BORSA PERFORMANS ANALİZİ 57
7.1. Testlerin Planlanması 57
7.2. Testler 58
7.2.1. Tüm GES Sunucularının Aynı Konakta Olduğu Testler 58
7.2.2. GES Sunucularının Farklı Konaklarda Olduğu Testler 61

8. SONUÇLAR VE TARTIŞMA 65

KAYNAKLAR 66

EK A: PERFORMANS İLE İLGİLİ GRAFİKLER 68

ÖZGEÇMİŞ 77

KISALTMALAR

JNI: Java Native Method Interface
API: Application Programming Interface
GES: Güvenli Etmen Sistemi
RMI: Remote Method Invocation
CGI: Common Gateway Interface
JNDI: Java Naming and Directory Interface
ARA: Agent for Remote Action
CORBA: Common Object Request Broker Architecture
DCOM: Distributed Component Object Model
RPC: Remote Procedure Call
İMKB: İstanbul Menkul Kıymetler Borsası
NASDAQ: National Association of Securities Dealers Automated Quotations
NYSE: New York Stock Exchange
HTTP: Hyper Text Transfer Protocol
HTTPS: Hyper Text Transfer Protocol Secure
TCP: Transmission Control protocol
SECMAP: Secure Mobile Agent Platform
GES: Güvenli Etmen Sistemi
JADE: Java Agent Development Framework
XML: Extensible Markup Language
JASA: Java Auction Simulator API

TABLO LİSTESİ

		Sayfa No
Tablo 4.1	**:** Etmen Çağrı Listesi……	21
Tablo 6.1	**:** Tacir Etmenin Aldığı Mesajlar……...	38
Tablo 6.2	: Tacir Etmenin Gönderdiği Mesajlar ……..	39
Tablo 6.3	: Broker Etmenin Aldığı Mesajlar ……	41
Tablo 6.4	: Broker Etmenin Gönderdiği Mesajlar ……	41
Tablo 6.5	: Borsa Etmeninin Aldığı Mesajlar ……..	45
Tablo 6.6	: Borsa Etmeninin Gönderdiği Mesajlar ……..	45
Tablo 6.7	: Yönetici Etmenin Aldığı Mesajlar …….	50
Tablo 6.8	: Yönetici Etmenin Gönderdiği Mesajlar …….	51
Tablo 6.9	: VT Etmeninin Aldığı Mesajlar ……..	52
Tablo 7.1	: Testler İçin Kullanılan Kısaltmalar ……	58
Tablo 7.2	: Test 1 Sonuçları …….....	58
Tablo 7.3	: Test 2 Sonuçları …….....	59
Tablo 7.4	: Test 3 Sonuçları …….....	60
Tablo 7.5	: Test 4 sonuçları ……..	60
Tablo 7.6	: Test 5 Sonuçları …….....	62
Tablo 7.7	: Test 6 Sonuçları …….	63

ŞEKİL LİSTESİ

		Sayfa No
Şekil 2.1	**:** Hareketli Etmen İle İstemci/Sunucu Modellerinin Karşılaştırılması……..	6
Şekil 4.1	**:** GES Sunucusu Bileşenleri…….....	18
Şekil 4.2	**:** Sarmalanmış Etmen Modeli……...	19
Şekil 4.3	**:** GES Mesajlaşma Mimarisi…….	23
Şekil 4.4	**:** Mesaj Paketi Yapısı……	24
Şekil 6.1	**:** Uygulama Etki Alanı Kavramsal Sınıfları……..	29
Şekil 6.2	**:** Etmen Tabanlı E-Borsa Modeli……..	32
Şekil 6.3	**:** E-Borsa Mimarisi……...	35
Şekil 6.4	**:** Tacir Etmen Oluşturma Mekanizması……	37
Şekil 6.5	**:** Tacir Etmen Yaratma İşlemleri……..	38
Şekil 6.6	**:** İstek İşleme Döngüsü……	42
Şekil 6.7	**:** Alım İsteği Eşleştirme Yordamı…….....	48
Şekil 6.8	**:** Satış İsteği Eşleştirme Yordamı…….....	49
Şekil 6.9	**:** E-Borsa Kullanıcı Giriş Ekranı……...	54
Şekil 6.10	**:** E-Borsa Kullanıcı Kayıt Formu……..	55
Şekil 6.11	**:** Kullanıcı Arayüzü Menüleri……...	55
Şekil 6.12	**:** Yönetim Arayüzü Menüleri……	56
Şekil 7.1	**:** Test 2 İçin SİBoUZ-İstek Sayısı İlişkisi…….	59
Şekil 7.2	**:** Test 4 İçin SİBoUZ-İstek Sayısı İlişkisi…….	61
Şekil 7.3	**:** Birinci Yapılandırma İçin Sistem Yapısı……	62
Şekil 7.4	**:** İkinci Yapılandırma İçin Sistem Yapısı…….....	63
Şekil 7.5	**:** Test 5 - Test 6 – Test 1 Karşılaştırması……..	63
Şekil A.1	**:** Test 5-1 Sırasında Konak 1'deki Sistem Kaynağı Kullanımı	68
Şekil A.2	**:** Test 5-1 Sırasında Konak 1'deki Ağ Trafiği……..	69
Şekil A.3	**:** Test 5-1 Sırasında Konak 2'deki Sistem Kaynağı Kullanımı	69
Şekil A.4	**:** Test 5-1 Sırasında Konak 2'deki Ağ Trafiği…….....	70
Şekil A.5	**:** Test 5-2 Sırasında Konak 1'deki Ağ Trafiği…….....	70
Şekil A.6	**:** Test 5-2 Sırasında Konak 2'deki Ağ Trafiği…….....	71
Şekil A.7	**:** Test 5-3 Sırasında Konak 1'deki Ağ Trafiği……..	71
Şekil A.8	**:** Test 5-3 Sırasında Konak 2'deki Ağ Trafiği……..	72
Şekil A.9	**:** Test 6-1 Sırasında Konak 1'deki Sistem Kaynağı Kullanımı	72
Şekil A.10	**:** Test 6-1 Sırasında Konak 1'deki Ağ Trafiği……..	73
Şekil A.11	**:** Test 6-1 Sırasında Konak 2'deki Sistem Kaynağı Kullanımı	73
Şekil A.12	**:** Test 6-1 Sırasında Konak 2'deki Ağ Trafiği……..	74
Şekil A.13	**:** Test 6-2 Sırasında Konak 1'deki Ağ Trafiği……..	74
Şekil A.14	**:** Test 6-2 Sırasında Konak 2'deki Ağ Trafiği……..	75
Şekil A.15	**:** Test 6-3 Sırasında Konak 1'deki Ağ Trafiği……..	75
Şekil A.16	**:** Test 6-3 Sırasında Konak 2'deki Ağ Trafiği……..	76

GÜVENLİ ETMEN SİSTEMİ KULLANARAK E-BORSA TASARIMI VE GERÇEKLENMESİ

ÖZET

İnternet gün geçtikçe büyümekte; internetin sağladığı uygulama sayısı ve interneti kullanan kullanıcı sayısı da sürekli artmaktadır. Elektronik ticaret uygulamaları da bu gelişimden payını almıştır; gün geçtikçe mevcut uygulamaların kalitesi artmakta ve yeni uygulamalar devreye girmektedir. Sanal borsa uygulamaları da son yıllarda popülerlik kazanan e-ticaret uygulamalarından biridir. Yatırımcıların gerçek risk almadan alım-satış yapabildiği bu uygulamalar sayesinde, yatırımcılar yatırım becerilerini geliştirme ve tecrübe kazanma fırsatını elde etmektedirler. İnternette çalışan her elektronik ticaret uygulaması gibi, borsa uygulamasının da performans, bilgi güvenliği ve ölçeklenebilirlik gibi ihtiyaçları vardır. Son yıllarda, bilişim dünyasında ilgi gören hareketli etmen teknolojisi, elektronik ticaret uygulamalarının ihtiyaçları için çözüm önerileri sunmaktadır.

Hareketli etmen sistemleri, ağ üzerinde hareket edebilen, birbirleri ile haberleşebilen ve bir amacı gerçekleştirmek için çalışan etmen adlı varlıkların bulunduğu sistemlerdir. Hareketlilik etmenlerin, başka konaklar üzerinde çalışmalarını; dolayısıyla ağ haberleşme yükünü azaltmalarını ve performansı arttırmalarını sağlamıştır. Hareketli etmen sistemleri geniş bir uygulama alanına sahiptir. Yapay borsalar, elektronik pazarlar ve açık arttırma sistemleri hareketli etmen sistemlerinin kullanıldığı e-ticaret uygulamalarına örnek olarak verilebilir.

Bu çalışmada, sanal borsanın dağıtık ve güvenli bir etmen sistemi kullanılarak gerçeklenmesi sağlanmıştır. Etmen çatısı olarak -güvenli bir hareketli etmen sistemi olan- GES (Güvenli Etmen Sistemi) kullanılmıştır. GES, sağladığı etkin güvenlik özellikleri yanında, güçlü ve esnek yapısı nedeniyle de kullanılmaya değer yeni bir hareketli etmen sistemidir. Ayrıca, GES etmen programcısına basit bir programlama arayüzü sunmaktadır. Sanal borsanın da gerçeğe en yakın şekilde ve ölçeklenebilir olarak çalışması için dağıtık bir sistem olarak tasarlanması uygun olacaktır. Çalışmada oluşturulan her alt sistem borsanın bir modülüne karşı gelmektedir.

Sonuç olarak, ortaya çıkarılan E-Borsa sistemi, dağıtık çalışmaya uygun olarak, borsayı oluşturan aktör ve sistem elemanlarını temsil eden, etmenlere, gerekli sorumluluklar atanarak; borsa sistemi için gerekli olan güvenlik, gizlilik ve işlem tekilliği gibi şartlar da karşılanarak gerçeklenmiştir. Böylece elektronik borsa sistemlerine, etmen sistemi kullanılarak farklı bir yaklaşım getirilmiştir.

DESIGN AND IMPLEMENTATION OF E-STOCK MARKET USING SECURE AGENT SYSTEM

SUMMARY

The Internet is growing exponentially; and number of users and number of applications are growing respectively. Like other applications, electronic trading applications are affected from this trend, quality of available applications are improving and new applications are created. Virtual stock markets are one of these e-trading applications which are getting more popular in recent years. These applications provide a risk-free environment for investors, so they buy and sell securites without taking risk and they gain experience about the trading strategies. Like other e-trading applications running over Internet, virtual stock market needs data security, performance and scalability. Agent technology that gets attention from IT world, suggests solutions for e-trading application's needs.

Mobile agent systems consist of entities called agents. These agents are able to communicate with each other, and work together to complete a common goal. Mobility provided to agents allow them to run on remote hosts, so that the network load is lessened and performance is improved. Mobile agent systems have a wide application area. Artificial stock markets, electronic marketplaces and online bidding systems are examples of e-trade applications which are implemented by agent systems.

In this work, virtual stock market is implemented by a secure and mobile agent system. SECMAP (Secure Mobile Agent Platform) is used as the secure mobile agent platform. SECMAP is worth being used not only for the security features it presents for agents and hosts, but also for its very flexible and powerful agent programming interface. It will be realistic to model a stock market as a distributed system.

Consequently, electronic stock market system is implemented as a distributed system. Agents represent the actors and elements that constitute the stock market, which have the related responsibilities. Security, privacy and atomicity properties, which are identified as either required or desirable in e-commerce systems are implemeted in our system. As a result, a different approach that takes advantage of agent systems is used for electronic stock market.

1. GİRİŞ

Bu bölümde, etmen metodolojisi kullanılarak gerçeklenen E-Borsa sisteminin ihtiyaçları anlatılacaktır.

1.1 Problemin İncelenmesi

İnternet gün geçtikçe büyümekte; internetin sağladığı uygulama sayısı ve interneti kullanan kullanıcı sayısı da sürekli artmaktadır. Elektronik ticaret uygulamaları da bu gelişimden payını almıştır; gün geçtikçe mevcut uygulamaların kalitesi artmakta ve yeni uygulamalar devreye girmektedir. Sanal borsa uygulamaları da son yıllarda popülerlik kazanan e-ticaret uygulamalarından biridir. Yatırımcıların gerçek risk almadan alım-satış yapabildiği bu uygulamalar sayesinde, yatırımcılar yatırım becerilerini geliştirme ve tecrübe kazanma fırsatını elde etmektedirler. Böylece kullanıcılar, gerçek riske girmeden alım ve satış yapar ve yatırım stratejileri ile ilgili deneyim kazanırlar.

Sanal borsanın gerçeğe yakın bir şekilde ve ölçeklenebilir olarak çalışması için dağıtık bir sisteme gereksinim duyulur.

Ayrıca tasarlanan sistemin aşağıdaki ihtiyaçları karşılaması gerekmektedir [1]:

Güvenlik:

Gönderilip alınan mesajlar gönderici ve alıcı dışında üçüncü bir şahıs tarafından elde edilememelidir.

İşlem Tekillik (Atomicity):

Tüm şartlar altında işlem ya tamamlanmalı (hisse miktarları ve hesaplar tutarlı bir şekilde güncellenmeli) ya da iptal edilmelidir.

Anonimlik:

İstekler takip edilerek yatırımcı profili belirlenememelidir. Yatırımcı profilinin ve başkalarına ait yatırım stratejilerinin öğrenilmesi, kötü niyetli kullanıcıların borsayı manipüle etmesine ve haksız kazanç elde etmesine neden olur.

Gizlilik:

Üçüncü şahıslar gerçekleşen işlemler ile ilgili bilgileri elde edememelidir.

Yukarıdaki ihtiyaçlara ek olarak sistemin performansı açısından, sistemin ölçeklenebilir olması da gerekmektedir.

1.2 Çözüm Önerisi

Yukarıda belirtilen problemin güvenli ve hareketli bir etmen sistemi kullanılarak gerçeklenmesi, güvenlik, gizlilik, anonimlik ve ölçeklenebilirlik şartlarının yerine gelmesini sağlayacağı için uygun olacaktır. İşlem tekilliği şartı için de gerekli önlemler sistem tasarımı sırasında alınacaktır.

Yazılım metodolojileri gün geçtikçe, merkezi çalışma modellerinden dağıtık çalışma modellerine doğru evrim geçirmektedir. Dağıtık çalışma modellerinin gerçeklenmesinde, hareketli etmen sistemleri gün geçtikçe daha sık kullanılmaktadır.

Bunun yanında hareketli etmen sistemleri, beraberinde bazı güvenlik sorunları getirmektedir. Bu güvenlik sorunlarına Güvenli Etmen Sistemi (GES) etkin çözümler bulmuştur. E-borsa sistemi ile hem çoklu bir etmen sistemi kullanılarak bir elektronik ticaret uygulaması gerçeklenecek hem de Güvenli Etmen Sistemi'nin esnekliği, pratikliği ve güvenliği ispatlanacaktır.

Çalışma, literatürde etmen sistemleri kullanılarak gerçeklenen ve kullanıcıları yerine yatırım yapacak akıllı etmenlerin yaratılmasını amaçlayan yapay borsa sistemlerinden farklı olarak güvenli, ölçeklenebilir ve dağıtık bir sistemi gerçekleme amacı üzerine odaklanmıştır. Bu yüzden çalışma, etmen sistemleri kullanılarak gerçeklenen borsa yapılarına yeni bir bakış getirmektedir. Gerçeklenen sistemde akıllı etmenler bulunmamakla birlikte, sistemin modüler yapısı sayesinde, sisteme akıllı etmenlerin daha sonra eklenmesi mümkündür.

2. ETMEN VE HAREKETLİ ETMENLER

2.1 Etmen Tanımı

Üzerinde anlaşılan ortak bir tanım olmamasına rağmen yazılım etmenleri; kullanıcıları adına, belirli bir otonomluk seviyesinde işler yürüten programlar olarak karakterize edilebilir.

Yazılım etmenlerine olan ilgi hızla artmaktadır. Bu ilgi geniş bir uygulama yelpazesinde (haber filtreleme etmenleri, alışveriş etmenleri, akıllı gezi etmenleri, öğrenme yardımcıları vb.) araştırma ve geliştirme yapılmasına öncülük etmiştir. Yazılım etmenlerinin popülerlik kazanmasının arkasında internet ve gelişen elektronik ticaret olguları vardır [2].

2.2 Etmen Tabanlı Yazılımlar

2.2.1 Etmen Tabanlı Yazılım Nedir?

Dünyayı etmen tabanlı düşündüğümüzde sistemde bulunan tek bir etmenin yetersiz olacağı anlaşılacaktır. Çoğu sorunun çözümünde problemin merkezi olmayan yapısı düşünüldüğünde, birden çok etmene ihtiyaç duyulacaktır. Bununla birlikte etmenler kendi hedeflerini gerçekleştirmek ve bağımlılıklarını kontrol etmek için birbirleri ile etkileşime girmeye ihtiyaç duyacaktır. Bu etkileşimler geleneksel istemci/sunucu tipinde olabileceği gibi, eşgüdüm, müzakere gibi sosyal etkileşimler de olabilir [3].

2.2.2 Etmen Tabanlı Yazılım Yaşam Döngüsü

Belirtim:

Bu bölümde etmen sistemi belirtimi problemi çözülmeye çalışılır. Sistemin ihtiyaçları ve temsil edebileceği özellikler belirlenmeye çalışılır. Özellikler şu şekilde sıralanır:

a. Etmenlerin ortam ile ilgili sahip oldukları bilgi;

b. Etmenlerin gerçekleştirmeye çalıştıkları hedefler;

c. Etmenlerin gerçekleştirecekleri eylemler ve bu eylemlerin etkileri;

d. Zaman içinde etmenlerin çevreleri ve diğer etmenler ile olan etkileşimleri.

Gerçekleme:

Bir belirtim verildiğinde, sistemin o belirtime göre doğru bir şekilde gerçeklenmesi gerekmektedir. Bu bölümde soyut belirtimden, somut sayısal sisteme geçiş gerçekleştirilir.

Doğrulama:

Somut sistemi geliştirdikten sonra, sistemin belirtime göre doğru şekilde çalıştığını göstermek gereklidir. Bu süreç doğrulama olarak bilinir [3].

2.3 Hareketli Etmenler

Dağıtık ortamlardaki birimlerin veri işlemek için kullandıkları haberleşme yöntemleri aşağıdaki gibidir:

Mesajlaşma: Bu yöntemde birimler eşit seviyelerde bulunurlar ve birbirlerine iskele (port), mesaj kutusu (mailbox), ya da boru hattı (pipe-line) gibi yapılar üzerinden mesajlar yollarlar. İki birim birbirlerinin adres uzaylarına erişemez. Gönderilen mesajın içerisinde verinin işaretçisi değil kendisi bulunmaktadır. Bu modelde mesajlaşma ayrıntıları ile programcı ilgilenmek zorundadır.

İstemci / Sunucu (client / server) Modeli: İstemci istekleri, mesaj aracılığı ile sunucuya iletir, sunucu mesaj verisini işler ve sonucu istemciye yollar. Uzaktan yordam çağırma (RPC) bu modelde en çok kullanılan yöntemdir. Bu yöntemde mesajlaşma ayrıntıları ile programcı ilgilenmez. RPC arayüzü bu ayrıntılardan programcıyı soyutlar.

Uzaktan Erişilebilen Nesneler: Bu modelde ağda herhangi bir konakta yer alan nesneler, diğer konaklar tarafından kullanılabilmektedir. Temelde uzaktan yordam çağırmadan farklı olmayan bu yöntem, yerel makine dışındaki nesnelere erişimin sağlandığı programlama esnekliği sayesinde daha fazla tercih edilir bir yöntem haline gelmiştir. CORBA, DCOM ve RMI teknolojileri bu sistem kullanılarak yaratılan çözümlerdir.

Dağıtık veri işleme yöntemlerinin son halkasında hareketli etmen sistemleri bulunmaktadır. Hareketli etmenler, programlarını ve durumlarını bilgisayar ağında uzaktaki bir makinede çalışmak üzere taşıyabilen etmenlerdir. General Magic şirketinin Telescript programla dili ile bu konuda öncü olmuştur [4]. Hareketli etmen

yaklaşımı, hareketli etmenlerin esnek, otonom, dinamik ve etkili olmaları nedeni ile araştırma topluluğu tarafından ilgi görmektedir [5].

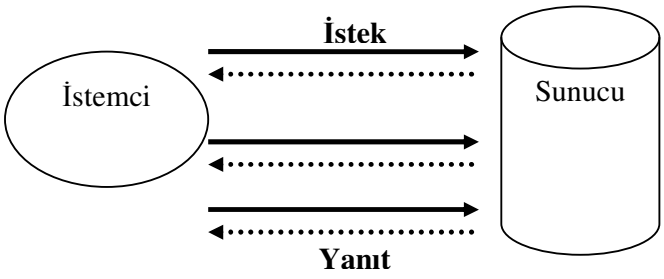
Hareketli etmenlerin arkasındaki mantık basittir. Amaç, proseslerin ağ üzerinde haberleşmesi için uzaktan prosedür çağrılarının yerini almaktır.

$V=B_N.m(argümanlar)$: Bu işlemi A prosesi gerçekleştirir. B_N; B prosesine ait bir nesnedir, m bu nesnenin bir metodu; V ise nesnenin metodunun döndürdüğü değerdir.

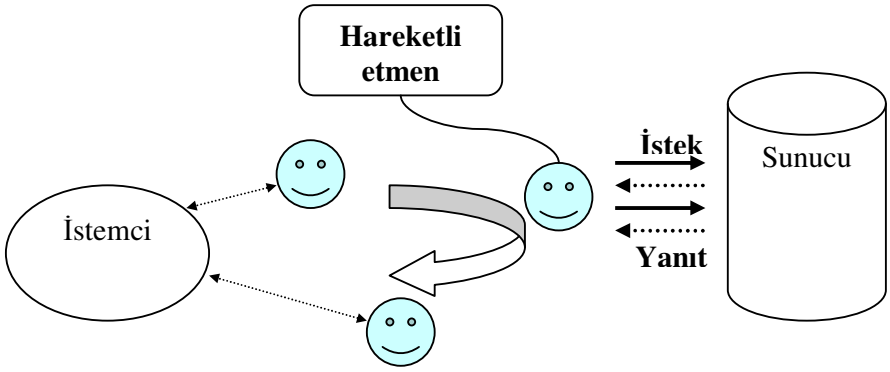
Uzaktan yordam çağrılarında, haberleşme senkrondur. A prosesi, B prosesi değeri döndürene kadar bloke olur. B değer döndürmezse, (Örneğin ağ bağlantısı kesilirse) A sonsuza kadar beklemek zorunda kalabilir.

Ayrıca A ve B arasındaki bağlantı, kullanılmadığı halde açık kalabilir. Bu da kaynakların boşuna kullanılması demektir.

Hareketli etmen mantığında Şekil 2.1'de görüldüğü gibi diğer makineye işlem yapması için bir etmen yollanır. A prosesi işlev çağıracağına bir hareketli etmeni B prosesine yollar. Etmen ile B prosesi iletişime geçer. Etmen ile B prosesi aynı adres uzayını kullandığı için etkileşimler çok daha etkili olur (ağ üzerinden sağlanan etkileşimlere göre). Etmen işlemini bitirdikten sonra, A'ya sonuçlar ile birlikte geri döner. Tüm operasyon sırasında, ağ zamanı sadece A'nın, etmeni B prosesine yollaması ve etmenin işini bitirdiğinde A prosesine dönmesinden oluşur. Bu yöntem, ağ kaynaklarının kullanılması bakımından, uzaktan metod çağırmaya göre çok daha etkili bir alternatiftir.



İstemci / Sunucu iletişimi



Hareketli etmen iletişimi

Şekil 2.1: Hareketli Etmen İle İstemci/Sunucu Modellerinin Karşılaştırılması

Hareketli etmen yaklaşımı, internet üzerinde dağıtılmış konaklar üzerinde paralel işlemleri gerçekleştirmeye de uygundur. Görevler ayrıştırılarak çoklu etmenlere atanabilir. Her etmen, tek başına çalışarak, kendi görevini yerine getirmeye çalışır. Her etmen farklı bir konakta çalışarak, dağıtılmış görevin kısa bir zamanda bitirilmesini sağlayabilir [5].

2.3.1 Hareketli Etmenlerin Sağladığı Yararlar

- Hareketli etmenler bilginin bulunduğu yerde işlenmesine çalışırlar, böylece ağ yükünü ve gecikmesini azaltırlar [6].
- Hareketli etmenlerin otonom olma özelliği, güvenilmeyen ağlarda (sık sık bağlantının kesildiği ağlar) sağlam ve hata hoşgörülü çözümler sağlar.
- Protokol bağımlılığını azaltırlar [7].

- Çevrelerine dinamik olarak adaptasyon sağlayabilirler.
- Heterojen ortamlarda çalışabilirler.

2.3.2 Mevcut Hareketli Etmen Sistemleri

Telescript: General Magic firması tarafından geliştirilmiş nesne tabanlı ve güvenlik desteğine sahip yeni bir dil içeren sistemdir [7].

Tacoma: Gerçekleme dili olarak TCL betik dili kullanılmıştır. Etmenler istemci isteklerini karşılamak için ağda hareket eden prosesler olarak modellenmiştir [8].

Agent Tcl: Sunucular arasında Tcl betiklerinin hareketine olanak sağlar. Etmenler yer bağımlılığı olan DNS isimleri ile tanımlanırlar [7].

Aglets: IBM tarafından java dili kullanılarak geliştirilmiştir. Java'nın applet modelini kullanır. Programcının etmenin nasıl hareket edeceğini belirlediği işlevleri gerçeklediği basit bir yazılım geliştirme çatısı vardır. Aglet, ağ üzerinde Aglet çalıştırabilecek konaklar arasında seyahat eden hareketli java nesnesi olarak tanımlanır. Her Aglet kendi ipliği içinde çalışır (otonom olma) ve kendisine gelen mesajlara yanıt verir (reaktif olma) [8].

Voyager: Java tabanlı bir sistemdir. Uzaktan erişilebilir sanal sınıflar yaratılarak nesnelere sanal referanslar kullanılarak yerden bağımsız olarak erişilmesi sağlanır [7].

Concordia: Java dili kullanarak gerçeklenmiş bir platformdur. Sistemin tasarım amaçları etmen hareketliliğini, etmenlerin işbirliği yapması için gerekli desteğin verilmesini, güvenilir etmen göçünü ve etmen güvenliğini sağlamaktır [9].

Ajanta: Java tabanlı ve güvenlik mekanizmaları diğerlerine göre daha gelişmiş olan bir sistemdir [7].

S-agent: Güvenli bir hareketli etmen sistemi için önerilen diğer bir mimaridir. Kaynaklara ve etmenlere erişmek için erişim politikaları kullanılır [7].

JavaSeal: Güvenlik amacıyla geliştirilmiş bir hareketli etmen sistemidir. Etmen sistemi, hiyerarşik olarak birbirine bağlı, aralarında komşuluk ilişkileri ve haberleşme kanalları olan ve mühür (seal) adı verilen nesnelerden oluşur [7].

JATLite (Java Agent Template, Lite): Etmenlerin internette sağlam bir şekilde iletişim kurmaları için Java dili ile yazılan program paketidir. Etmenler, Etmen

Mesaj Yönlendiricisi (Agent Message Router) aracılığı ile kullanıcı adı ve parola kullanarak internete bağlanabilir, mevcut bir bağlantıyı kesebilir, mesaj alıp gönderebilir, FTP ile dosya transfer edebilir ve diğer etmenler ile veri alışverişinde bulunabilirler [10].

2.3.3 Hareketli Etmen Sistemlerinin Güvenliği

Internet kullanıcılarına birçok imkan tanır, bunun yanında internette bir sistemi güvenilmez, yararsız ve kullanılmaz duruma getirebilecek güvenlik ile ilgili riskler de vardır. Gerçekte internet güvensiz bir ortamdır, bilgi kötü niyetli ara sistemlerde dolaşabilir ve saldırganlar tarafından izlenebilir.

Internet'in yaygın kullanımı ile birlikte, dağıtık sistem güvenlik gereksinimi de dramatik bir şekilde artmış, bu gereksinim güvenlik çözümleri üretecek araştırmaların yapılmasını sağlamıştır [6].

Hareketli bir etmen sistemi, güvenilmeyen etmenlerin yerel olarak çalışmasına, yerel servisleri ve kaynakları kullanmasına ve diğer etmenler ile etkileşime geçmesine izin verebilir. Tehditler, yapı olarak diğer bilgisayar sistemlerinin yüz yüze kaldıklarından farklı değildir [11]:

- **İstenmeyen Erişim:** Etmen veya servis doğru asıllamayı yapmadan verilere erişebilir.
- **İstenmeyen Değişiklikler:** Etmen veya servis veriyi istenmeyen bir şekilde değiştirebilir veya izinsiz bir şekilde silebilir. Konak etmeni silerek ona zarar verebilir.
- **Hizmet Kıtlığı (Denial of service):** Etmen veya servis, sistemi çalışmaz hale getirecek ve diğer programların çalışmasına imkan vermeyecek şekilde paylaştırılmış sistem kaynağı tüketebilir. Konak etmene kaynaklarını kullanma olanağı sunmayabilir.
- **Truva Atları:** Etmen veya servis yanlışlıkla kötü niyetli kod çalıştırabilir.
- **Rahatsız Edici Saldırılar:** Etmen üzerinde çalıştığı konağı rahatsız edebilir, örneğin istenmeyen resimleri ekranda görüntüleyebilir, sürekli ses çıkararak konaktaki kullanıcıyı rahatsız edebilir. Konak, etmenin çalışmasını geciktirerek veya etmenin güzergahını izleyip, etmen ve göndericisi hakkında bilgi toplayarak, etmeni rahatsız edici saldırılara maruz bırakabilir [7].

- **Yanlış Bilgilendirmeye (Dezenformasyon) Dayalı Saldırılar:** Konakların veya etmenlerin yanlış bilgilendirme ile kötüye kullanılması. Örneğin, kötü amaçlı bir etmen ağ üzerinde sistem yöneticisini taklit ederek kullanıcılardan şifrelerini veya önemli bilgilerini isteyebilir veya konaklar, etmenleri yanlış yönlendirerek etmenlerin istenmeyen konaklara göç etmesine neden olabilir.

Yukarıda anlatılan saldırılar kötü niyetli etmenler veya kötü niyetli konaklar tarafından gerçekleştirilebilir.

Etmen bir konağa geldiğinde konak şunlardan emin olmalıdır:

- Etmenin geldiği konaktan ayrılırken taşıdığı kodun aynısını taşıdığı doğrulanmalıdır.
- Etmen programı platformda çalışmak için, doğru kimlik bilgilerine sahip olmalıdır.
- Etmenin yerel kaynak ve servislere erişimi izine bağlı olmalıdır.
- Etmenin yerel etmenler ile etkileşime geçmesi izine bağlı olmalıdır.
- Etmen ve platform arasındaki tüm iletişim doğrulanmalıdır.

2.4 Etmen Sistemleri ve Elektronik Ticaret

En çok dikkat çeken ağ teknolojilerinden biri olan hareketli etmenlerin hareketlilik ve otonomluk gibi özellikleri sayesinde, gelecekteki elektronik ticaret sitemlerinde önemli bir rol oynayacağı düşünülmektedir [12].

2.4.1 E-ticaret Etmenlerinin Temel Karakteristikleri

Etmenin kendi uzmanlık alanına ilişkin bir modeli ve diğer etmenler ile bilgi alışverişinde bulunabileceği bir farkındalık modeli olmalıdır. Farkındalık modeli, etmenin diğer etmenlerin tüm içyapısını ve yeteneklerini bilmesini gerektirmez; etmen kendisi tarafından karşılanamayan bir isteğin diğer etmenler tarafından nasıl karşılanacağını bilmelidir.

Etmen teknolojisi, elektronik ticaret yazılımlarında kullanılacak; zengin ve etkileyici kurumsal modeller sunabilir [13].

Internet ticari amaçlar için kullanılmak üzere tasarlanmamıştır. Bu yüzden ticari amaçlar için kullanım sırasında bazı limitler ile karşılaşılır [4]:

Güven: Hangi satıcının güvenilir olduğunu bilmek zordur.

Mahremiyet ve Güvenlik: Kişisel bilgilerin güvenliği sorun teşkil eder. HTTPS protokolünü kullanmak bu sorun ile başa çıkmanın bir yolu ama yine de güvenlik önemli bir konu.

Ücretlendirme: İnternet gömülü ücretlendirme mekanizmaları ile tasarlanmamıştır. Mekanizmalar, temel internet yapısı kullanılarak gerçeklenmektedir.

2.4.2 Etmenlerin Kullanıldığı E-ticaret Uygulamaları

Etmenler, dağıtık çalışmaya uygun ve otonomluk özelliğinin kullanılabileceği birçok elektronik ticaret uygulamasında kullanılabilir. Etmenlerin kullanıldığı uygulamalar şu şekilde özetlenebilir:

Açık Arttırma Sistemleri: Bu sitemlere örnek olarak açık arttırma robotları verilebilir [4]. Bu robotlar kullanıcıları adına açık arttırmaya katılırlar. Etmenler, kullanıcıları adına hem alım hem de satış yapabilirler. Satıcı etmenlere kullanıcı tarafından 3 adet parametre geçirilir. Bunlar; malın satılacağı tarih, malın satılmak istendiği fiyat ve malın satılacağı minimum fiyat şeklindedir.

Mal istenen fiyattan satılmaya çalışılır. Satış bitiş zamanı yaklaştıkça, fiyat düşürülür (en düşük fiyata kadar). Kullanıcı bu işlem için bir düşürme fonksiyonu tanımlar. Bu fonksiyon lineer, ikinci dereceden ve kübik olmak üzere 3 türlü olabilir. Kullanıcıya ürün satımı ile ilgili olarak veto hakkı verilir.

Alım yapacak etmenler için de daha önce belirtilen 3 parametre (satış etmeninin aksine alış etmeninde maksimum fiyat parametresi vardır) kullanılır.

Elektronik Pazarlar: Elektronik ticaretin en yaygın örneklerinden biri internet üzerinden hizmetlerin ve ürünlerin alınıp satılmasıdır. Geleneksel elektronik pazarlar satıcıları ve alıcıları buluşturur, ancak bu sistemler, alıcılar ve satıcılar arasında yapılacak pazarlık ve anlaşmalar ile ilgili bir çözüm sunamamaktadır [14]. Etmen sistemlerinin kullanıldığı elektronik pazarlarda; etmenler kullanıcıları adına mallar veya hizmetler için pazarlık yaparlar. Etmen, kullanıcısının belirlediği şartlara uyarak, kullanıcısının sağladığı kârı en iyilemek için çalışır.

Yapay Borsalar: Yapay borsalar, gerçek borsalar için birer model oluşturmaktadır ve piyasa dinamiklerini analiz etmek için tasarlanırlar. Etmene dayalı yapay borsalarda yazılım etmenleri borsa üzerindeki tacirleri temsil etmek için kullanılır

[15]. Borsalardaki dinamik ve karmaşık yapı, akıllı etmenlerin kullanıldığı çoklu etmen sistemleri ile gerçek borsa yapısına en yakın şekilde modellenebilir. Gerçek borsalarda pazar fiyatları, yatırımcıların birbirleri ile etkileşimleri aracılığı ile oluşur; yapay borsada yatırımcılar birbiri ile etkileşim halinde olan yazılım etmenleri olarak modellenir.[15]

Aşağıda bazı yapay borsa sistemleri için bilgiler verilmiştir:

Santa Fe Yapay Borsası: Santa Fe Enstitüsü tarafından geliştirilmiştir [16]. Uygulama başlangıçta C dili kullanılarak, Unix istemcilerde çalışacak şekilde yazılmıştır daha sonra etmen tabanlı algoritmalara yardımcı olacak şekilde objective-C dili ile yazılarak, geliştirilmiştir. Sistemde kullanıcıları adına hisse alışverişi yapan tacir etmenler bulunur. Tacir etmenler akıllı etmenler olarak tasarlanmıştır. Ticaret stratejilerini, geçmişte yapılan alışverişleri baz alarak ve genetik algoritmaları kullanarak belirlerler. Başarılı olan stratejiler yaşarlar ve bir sonraki nesildeki stratejileri türetmek için kullanılırlar, zayıf stratejiler ise kullanılmayarak strateji topluluğundan çıkarılırlar.

Yapay Borsalar İçin Etmen Tabanlı Çatı: (Boer et al. 2004)Yapay borsa sisteminin oluşturulması sırasında tasarlanan etmen çatısı, asenkron olarak haberleşen rol tabanlı etmenlerden oluşur. Çatı daha önce tasarlanan etmen tabanlı yapay borsalara göre daha esnek bir yapıya sahiptir. Etmen çatısı, değişik pazar tipleri ve yatırım stratejileri ile çalışılabilecek şekilde tasarlanmıştır. Sistem üç tacir iskeleti içerir (pazar şekillendiriciler, brokerlar ve yatırımcılar). Bu iskeletler, tacir tipleri için temel yapıyı ve davranışları sağlar. Tacirler, aynı anda birden çok işi gerçekleştirebilecek otonom yazılım etmenleri olarak tasarlanmışlardır. Sistem hem sürekli hem de ayrık zamanda yapılan isteklerden oluşan pazarları desteklemektedir.

Yatırımcı etmenler sürekli olarak mesaj alır, bu mesajları analiz ederek verilen bir yatırım stratejisine göre istekleri oluşturup, istekleri broker veya pazar şekillendiricilerine gönderirler. Yatırımcılar, erişilebilir brokerlar'ı arayabilirler. Brokerlar, birincil görev olarak yatırımcılardan gelen istekleri karşılarlar. Brokerlar kendilerine gelen istekleri borsa şartlarına göre daha iyi hale getirip pazar şekillendiricilerine gönderebilirler. Pazar şekillendiricileri, katılımcılardan gelen istekleri kabul ederler. Eğer istek tamamen veya kısmen karşılanabiliyorsa, işlem gerçekleşir ve isteği yapanlar bu durumdan haberdar edilir.

Sistem farklı makinelerde bulunan etmenlerin borsaya istek yollamalarına izin vermektedir. Etmen çatısı JADE kullanılarak gerçeklenmiştir. Süreklilik -sürekli gelen istekler için gerekli olan- JADE'in sağladığı eşzamanlı etmen aktivitesi modeli (Java ipliklerini kullanır.) kullanılarak sağlanmıştır. Yatırımcıların kullandığı stratejiler XML dosyalarında tutulur [15].

JASA (Java Auction Simulator API): JASA değişik açık arttırma metodlarının çalıştırılmasına izin veren, yüksek performanslı açık kaynak kodlu bir açık arttırma sistemidir. Yazılım yeni arttırma tiplerinin eklenmesine izin verecek esnekliğe sahiptir. Ayrıca, yazılım, basit stratejiler kullanarak ticaret ve yatırım yapabilecek etmenlerin gerçeklenebilmesi için gerekli temel sınıfları sağlar. Liverpool Üniversitesi tarafından yürütülmekte olan bir araştırma kapsamında geliştirilmiştir. Sistemin hızlı çalışması tasarım hedeflerinden biridir. Çekirdek sistem, hızlı ve yüksek performanslı olarak tasarlanmıştır [17].

3. ETMEN SİSTEMLERİ İÇİN JAVA DİLİNİN KULLANILMASI

3.1 Otonomluk

Bir yazılım programının otonom olabilmesi için, program ayrık bir proses veya iplik (thread) olmak zorundadır. Uygulamada etmen, kontrolü sağlayan iplik olabilir. Java iplikli uygulamaları destekler.

Akıllı etmenler otonom program ve proseslerdir. Bu yüzden kullanıcı isteğine veya çalışma ortamında gerçekleşen bir değişikliğe yanıt vermeye hazırdırlar.

3.2 Hareketlilik

Java'nın taşınabilir bytecode'ları ve JAR dosyaları derlenmiş Java sınıf gruplarının, ağ üzerinden gönderilmesine ve hedef makinede kodların çalıştırılmasına izin verir. Java appletler'i Java kodunun, web tarayıcısı aracılığı ile uzaktan çağrılmasına izin verir.

Hareketli etmenlerin önemli gereksinimlerinden biri de çalışan proseslerin durumunun saklanabilmesidir. Bu sayede kod başka bir sistemde kaldığı yerden çalışmasına devam edebilir.

Java'nın hareketliliği sağlayan başka özellikleri de vardır. Atama Olay Modeli (Delegation Event Model) Olay kaynaklarını (Event Source) olay dinleyicilerinin dinamik olarak belirlenmesini sağlar. Bu sayede hareketli bir etmen, çalışmakta olan bir sunucu ortama vardığı zaman kendini ortama adapte edebilir ve ortamdan gitme zamanı geldiğinde ortamdan ayrılabilir.

"java.net" paketi, hareketli etmenlerin veya hareketli etmen sistemlerinin diğer sunucular ile konuşmasını ve Java kodu ile durum bilgisinin soketler üzerinden gönderilmesini sağlayacak yeteneklere sahiptir.

Java'yı etmen uygulamaları için ideal yapan özellikler özet olarak şöyledir:

- Java taşınabilirdir, mimariye bağlı değildir, derleme sonucu oluşan "bytecode" Java Sanal Makinesi tarafından yorumlanır ve çalıştırılır. Java

Sanal Makinesini destekleyen herhangi bir sistem, bir Java programını çalıştırabilir.

- Just-In-Time (Tam zamanında üretim) derleyiciler performansı arttırmak için bytecode'u makine diline çevirmek için kullanılır.
- Java işaretçi ve işaretçi işleme operatörleri içermez. Bellek tahsisi "new" operatörü kullanılarak gerçekleştirilir. Artık kullanılmayan nesneler otomatik olarak bellekten çıkarılır.
- Java appletler'i sunucudan istemciye indirilebilen ve bir Web tarayıcısı tarafından, bir HTML dokümanının parçası olarak çalışan küçük programlardır. Güvenlik nedeniyle, appletler'in yerel kaynaklara erişimleri kısıtlanmıştır.
- Java, C++ sözdizimine sahip nesneye dayalı bir programlama dilidir. Bir Java sınıfı, bu sınıftan yaratılan nesnelerin durum ve davranışlarını tanımlayan üye ve işlevlerden oluşur. Ek olarak, Java performans için temel veri tiplerini destekler.
- JNI, Java sanal makinesinin C ve C++ dillerinde yazılmış programları çağırmasına izin verir. JNI ayrıca diğer dillerde yazılan programların Java sanal makinesini başlatmasına ve Java programları çalıştırmasına izin verir.
- Java UTF-16 karakter kümesini kullanır. UTF-16 karakter kümesi dünya üzerinde kullanılan dillerin büyük bir çoğunluğunu destekler.
- Java dili çeşitli paketler aracılığı ile genişletilmiştir. Örneğin ağ iletişimi için "java.net", güvenlik için "java.security", grafik arayüz için "java.awt", uzaktan işlev çağrıları için "java.rmi", veritabanı erişimi için "java.sql" kullanılır.
- JavaBeans, Java'nın yazılım bileşen mimarisidir (software component architecture). Bu özellik mevcut yazılım bileşenlerini (ActiveX bileşenleri de dahil olmak üzere) kullanarak uygulama geliştirme olanağı sağlar.
- Java'nın sağladığı sunucu teknolojisi servlet'ler aracılığı ile CGI betiklerine benzer betikler yazılarak hareketli etmenlere erişim sağlanabilir. JNDI kullanılarak dizin ve isimlendirme hizmetlerine ulaşılabilir [18].

Java etmen tasarımı ve gerçeklemesi için gerekli tüm işlevselliği sağlamaktadır. Genel amaçlı, nesneye dayalı yetenekleri, çıkarsama ve öğrenme algoritmalarının kolayca gerçeklenmesini sağlar (akıllı etmenler için). Java'nın taşınabilir bytecode'u etmenlerin applet veya hareketli Java programları olarak paketlenmesine izin verir [19].

4. GÜVENLİ ETMEN SİSTEMİ (GES)

E-Borsa sistemi GES sistemi [7] kullanılarak gerçeklenmiştir.

GES, programcıya hareketli etmenler oluşturması için gerekli fonksiyonları sunan, etmenlerin birbirleriyle haberleşmesini ve hareket etmeleri için gerekli alt yapıyı oluşturan sistemin adıdır. GES, hareketli bir etmen sisteminden beklenen aşağıdaki isterleri yerine getirmektedir [20]:

- Sistem herhangi bir düğüm üzerinde birden çok etmenin çalışmasına olanak sağlar.
- Mimari, etmenlerin bulunduğu düğümden saydam olarak birbirleriyle haberleşmesine olanak sağlar.
- Mimari etmenlerin dağıtık ortamdaki düğümler arasında hareket etmesine olanak sağlar.
- Mimari gerekli gördüğü anda bir etmenin çalışmasını sona erdirebilir.
- Mimari etmenlerin birbirlerine doğrudan erişimini engeller.
- Mimari etmenlerin yaratılma, aktive edilme, mesajlaşma, hareket etme, yok edilme gibi etkinliklerini kaydeder.
- Mimari her etmene tekil bir kimlik numarası verir.
- Bütün haberleşmeler şifreli olarak gerçeklenir. Şifreleme standardı olarak SSL kullanılır.
- Etmen kodu kara kutu güvenliği ismi verilen yöntemle korunur. Sistem içinde yerini alacak olan etmen, güvenlik yöneticisi tarafından şifrelenir ve sisteme bu şekilde alınır. Etmen, düğümler üzerinde veya hareket halinde iken şifrelidir ve etmen kodu bu aşamada çalıştırılabilir bir kod değildir. Bir düğüm üzerinde aktive edilecek olan etmen, sadece belleğe yüklenme aşamasında kara kutu açılarak çalışmasına başlar. Etmen kodu sadece bellekte çalıştırılabilir kod parçası olarak görülmektedir.

4.1 Güvenli Etmen Sistemi Mimarisi

GES, heterojen platformlarda çalışılabilmesi için Java dilinde gerçeklenmiştir [20]. Etmen programcısı tarafından geliştirilecek etmenler de Java dili kullanılarak geliştirilmelidir.

GES mimarisinin ana bileşenleri şu şekildedir:

4.1.1 GES Sunucusu

GES mimarisinin ana bileşeni, etmen kabul edecek her düğüm üzerinde kurulu ve çalışır durumda olması gereken GES sunucusudur. GES sunucusu, etmen yaratma, çalıştırma, etmen haberleşmesi ve etmen göçü gibi temel fonksiyonları sağlar.

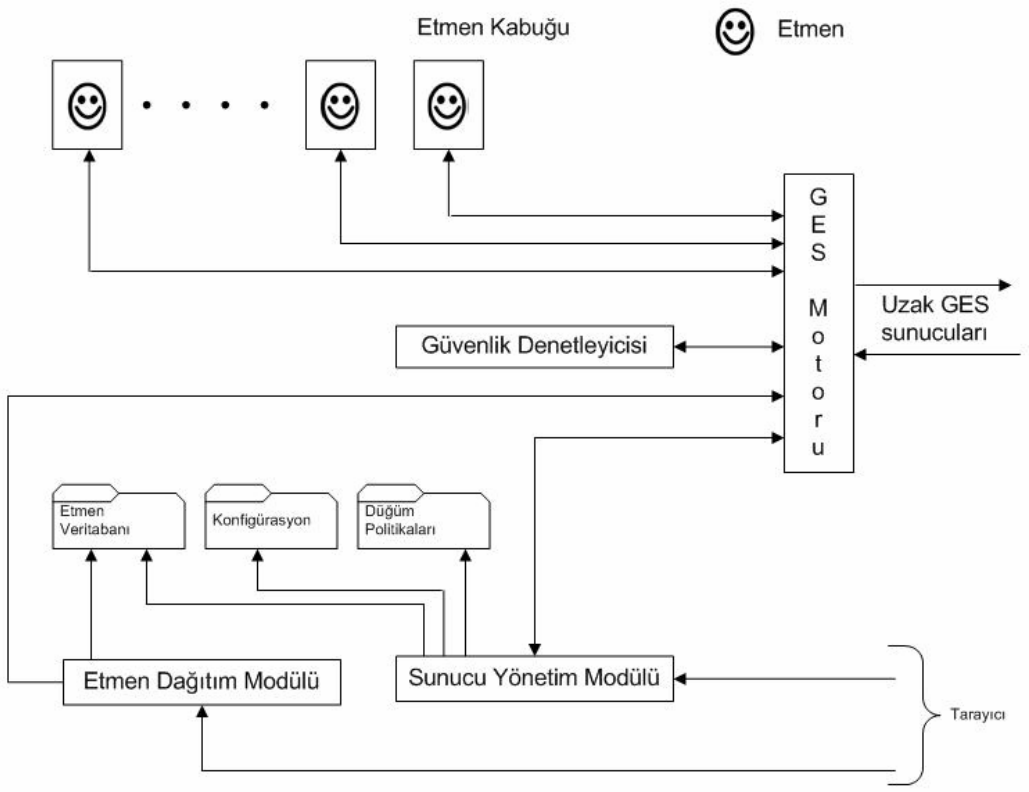
GES sunucusu, üç farklı düzende çalışabilir:

Standart GES Sunucusu (SGES): Görevi, etmen yaratma, çalıştırma, yok etme, etmen haberleşmesini sağlama ve etmen göçü gibi temel etmen fonksiyonlarını yerine getirmektir.

Güvenlik Yönetici GES Sunucusu (GYGES): Herhangi bir GES sunucu, başarılı bir şekilde başlayıp etmenlere ev sahipliği yapmadan önce kimlik denetiminden geçmek zorundadır. Bu kimlik denetimini kendisi için düğüm yönetici tarafından tanımlanmış olan GYGES'ten karşılar. Her GES sunucu için birden fazla GYGES tanımı yapılabilmesine rağmen aynı anda aktif olan sadece bir GYGES olabilir. Aktif olan GYGES'e ulaşılamaz ise GES sunucu güvenlik ihtiyaçları için tanımlanmış olan diğer GYGES sunuculara bağlanmaya çalışır. Kimlik denetiminden geçebilen GES sunucuları artık uzak sunucular ile iletişime geçebilir ve etmenlere ev sahipliği yapabilir. GES mimarisi sistemin hataya dayanıklı olması ve sürekliliği sağlamak üzere, sistemde birden çok GYGES'in etkin olmasına olanak sağlamaktadır.

Gözleyici GES Sunucusu (GGES): . Gözleyici düzende çalışan GES sunucunun görevi güncel "etmen-yer" bilgilerini tutmak ve istendiğinde bunu GES sunuculara iletmektir [7]. Herhangi bir GES sunucu, gözleyici olarak en az bir GES sunucuyu tanımlıyor olmalıdır. Birden fazla gözleyici GES tanımı olması durumunda, sunucu başarılı bir şekilde iletişim kuracağı gözleyici GES'i bulana kadar sırayla bağlantı kurmayı dener. Ancak, belirli bir anda, sadece bir gözleyici GES'i kullanıyor olabilir. Güvenlik yöneticisi düzende çalışan GES sunucular gibi Gözleyici düzende çalışan GES sunucular arasında da "ortaklık" ilişkisi vardır.

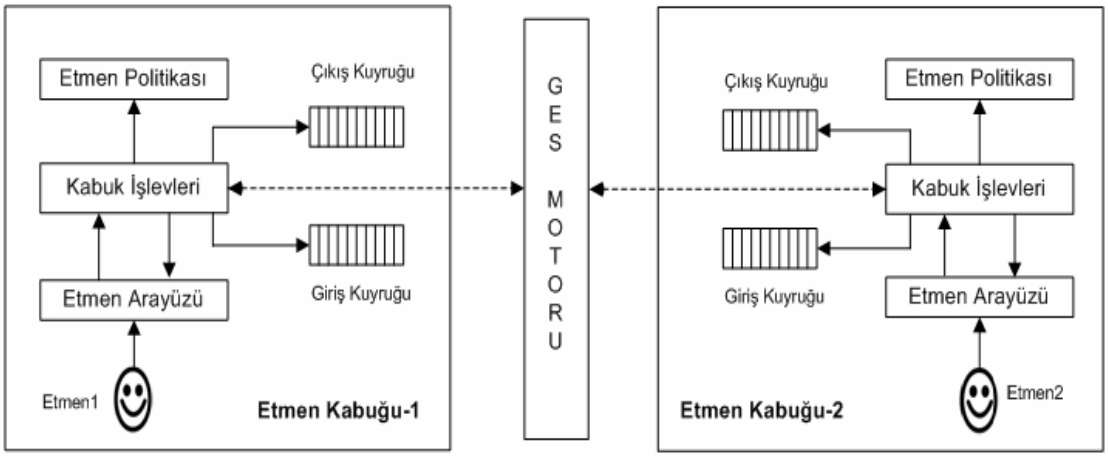
GES sunucusunun ana bileşeni GES motorudur. GES motoru, sunucunun diğer sunucular ile olan haberleşmesini sağlayan düşük seviyedeki temel fonksiyonları içerir. Şekil 4.1'de standart düzende çalışan bir GES sunucunun ana bileşenleri gösterilmektedir.



Şekil 4.1: GES Sunucusu Bileşenleri

Güvenlik Denetleyicisi; yürütülen etmen aktivitelerinin (mesajlaşma, göç etme, diske yazma, soket bağlantısı kullanma, vb.) güvenlik politikalarına uygun olup olmadığını kontrol eder.

Etmen Kabuğu; etmeni bir kabuk gibi sarmalayıp, diğer etmenlerden ya da yazılımlardan kaynaklanabilecek doğrudan erişimleri engelleyerek etmeni korur. Etmen Kabuğu etmenin dış dünya ile iletişimi için bir arayüz sunar. Ayrıca etmen giriş çıkış mesaj kuyruklarının yönetiminden de sorumludur. "Sarmalanmış Etmen Modeli" adı verilen bu modelde her etmen bir etmen kabuğu tarafından korunur. Sarmalanmış Etmen Modeli Şekil 4.2'de gösterilmiştir.



Şekil 4.2: Sarmalanmış Etmen Modeli

Etmen Veritabanı; etmenlerin saklanan son durumları, kodu ve politikasının tutulduğu disk alanıdır. Bu bilgilerin hepsi şifreli olarak disk üzerinde tutulur.

Konfigürasyon; etmen sunucusuna ait yapılandırma bilgisinin tutulduğu alandır. Örneğin sunucunun hangi düzende çalıştığı, ortaklık ilişkisi içinde olduğu diğer sunucuların adresleri, dinleyen TCP port numaraları gibi ayarlar burada bulunur.

Düğüm Politikaları; düğüm yöneticisinin düğümler için oluşturduğu politikaların tutulduğu disk alanıdır.

Sunucu Yönetim Modülü; sunucunun uzaktan web tarayıcı ile yönetilmesi için gerekli olan fonksiyonları barındırır. Etmen programcısı veya düğüm yöneticisi, etmenleri ve sunucuyu uzaktan bu bileşen aracılığı ile izleyebilmekte ve yönetebilmektedir.

Etmen Dağıtım Modülü; uzaktan etmen programcısının yazdığı etmeni tarayıcı aracılığı ile sunucu üzerine dağıtma görevini yerine getirir.

4.1.2 GES Etmenleri

GES etmenleri yaşam döngüleri boyunca belirli bir anda yaratılma, aktif, pasif, hareket halinde ve sonlanma durumlarından birinde olabilirler. Bu durumlar etmenin davranışını belirleyen durumlardır ve etmen programcısı, bu durumların her biri için, etmenin yapacağı işleri etmen kodu içerisinde belirtmelidir.

Etmen durumları şu şekilde sıralanır:

Yaratılma: Etmen yaşam döngüsü boyunca bu durumda sadece bir kez bulunur. Etmenlere yaratıldıkları anda bir kimlik bilgisi atanır. Bu kimlik bilgisi tekil olmayı sağlar.

Aktif Duruma Getirme: Daha önce yaratılmış olan bir etmenin çalıştırılabilir kod parçası belleğe alınır ve artık etmen bu aşamadan sonra çevresi ile etkileşime geçebilir. Aktif olan etmen mesaj gönderip alabilir, göç isteğinde bulunabilir, bir kopyasını yaratabilir, kendini çevresine bir isim ile duyurabilir veya pasif duruma geçme isteğinde bulunabilir.

Pasif Duruma Getirme: Pasif duruma geçirilen etmen artık etkin değildir, etmenin son durumu bilgisi GES sunucu diskinde saklanmıştır. Etmene ait çalıştırılabilir kod bellekten temizlenir. Etmen başka bir konağa göç ederken geçici bir süre pasif duruma geçirilir daha sonra tekrar aktif edilir. Pasif durumdaki bir etmen hiç bir şekilde çevresi ile etkileşime geçemez, mesaj gönderip alamaz [7].

Hareket Halinde Olma: Etmenin başka bir konağa taşınması sırasında ilk önce pasif duruma getirildiği daha sonra aktif edildiği durumdur.

Yok Edilme: Etmen için yok edilme isteği geldiğinde önce etmen pasif duruma geçirilir daha sonra, etmen kodu, durum bilgisi ve politikası diskten silinir.

Etmenlere yaratıldıklarında kendilerine tekil bir etmen kimliği atanır. Etmenlere bu kimlik bilgileri aracılığı ile erişilir.

Etmenler, GES sunucuları ile "Etmen Kabuğu" (AgentShield) sınıfının sağladığı "Etmen Arayüz" (AgentInterface) arayüzü aracılığı ile etkileşime geçer. Etmen bu arayüz aracılığı ile GES sunucuya, mesajlaşma, göç etme, kopya yaratma ya da pasif hale gelme gibi isteklerini iletir. Etmenin arayüz aracılığı yapabildiği çağrılar Tablo 4.1'de listelenmektedir.

Tablo 4.1: Etmen Çağrı Listesi

Metod	Kullanım Amacı
public String getAgentHostName()	Etmenin o an üzerinde çalıştığı düğümün adresini öğrenmesi sağlanır.
public void setVisibleOn(String strIdentifier)	Etmen bir isimle kendini ortama duyurur.
public void setVisibleOff()	Etmen kendini ortamdan gizler.
public Enumeration getAgentPointer(String strIdentifier)	Etmen, ismini verdiği bir diğer etmene bir işaretçi elde eder.
public Enumeration getAgentPointer()	Etmen sistemdeki tüm etmenleri gösteren bir liste elde eder.
public MessageID sendMessage(AgentPointer agentpointer, Message message)	Etmen diğer bir etmene mesaj yollar.
public MessagePacket receive()	Etmen giriş kuyruğundaki mesajı alır.
public boolean isMessageFailed(MessageID id)	Etmenin mesajının başarılı bir şekilde gönderilip gönderilmediğini öğrenir.
public void sendReply(MessagePacket packet,Object reply)	Etmen, bir mesajın yanıtını yollar.
public boolean isReplyReady(MessageID id)	Etmen, daha önce gönderdiği mesajın yanıtının hazır olup olmadığını öğrenir.
public boolean waitForReply(MessageID id,long ms)	Etmen gönderdiği mesajın yanıtı için belirli bir süre bekler.
public Object getReply(MessageID id)	Etmen giriş kuyruğundan daha önce gönderdiği bir mesajın yanıtını alır.
public boolean isAgentAlive()	Etmenin hala canlı olup olmadığı sorgulanır.
public boolean waitForMessage(long ms)	Etmen mesaj almak için belirli bir süre bekler.
public TransferResponse move(String strAgentHostName)	Etmen uzak düğümlere göç eder.
public void suspend(long suspendTime)	Etmen belirli bir süre uykuya geçer.
public AgentPointer createClone()	Etmen bir kopyasını yaratır.
public void setRepliedMessageHandler(String method)	Etmen, gönderdiği mesajın yanıtı geldiğinde çağrılmasını istediği metodu belirler.

4.2 Etmen Yazılımı Geliştirme Arayüzü

GES mimarisi programcıya, etmen davranışını programlayabileceği basit bir şablon ve oldukça esnek fonksiyonlar sunmaktadır [7]. Şablon, etmenin, bulunabileceği durumlara göre programlanabilmesi için gerekli metodları içerir.

Yazılacak etmen, "Agent" sınıfından türetilmeli ve sınıfın adı "Main" olmalıdır. Etmen şablonu aşağıdaki gibidir:

```
import agent.*;
public class Main extends Agent {
  public void OnMessageArrive() {
  }
  public void OnCreate() {
  }
  public void OnActivate() {
  }
  public void OnInactivate() {
  }
  public void OnEnd() {
  }
  public void OnTransfer() {
  }
}
```

"Main" içindeki metodların anlamları şu şekildedir:

OnMessageArrive: Etmene yeni bir mesaj geldiği zaman bu metod çalıştırılır. Çalıştırılan metod kodu ayrı bir iplik(thread) içinde çalıştırılır. Bu metodun çalıştırılabilmesi için etmenin aktif durumda olması gerekmektedir. Bu metod, etmene her yeni mesaj geldiğinde çalıştırılır, ancak eğer bir önceki mesajın işlenmesi devam ediyorsa çalışma bekletilir. Etmene ait aktif durum metodu (OnActivate) ayrı bir iplik içinde çalıştığı için, bazı durumlarda etmen iki ayrı iplik içinde (OnActivate ve OnMessageArrive) çalışıyor olabilir. Bu metodların içine kod yazılırken karşılıklı dışlama durumları dikkate alınmalıdır.

OnCreate: Etmenin sisteme alınıp, yeni bir kimlik bilgisi verilerek yaratıldığı anda çalıştırılan metoddur. Bu metod etmen yaşam döngüsü boyunca sadece bir kez çalıştırılır. Etmene ait bu metod çalıştırıldıktan sonra aktif durumu belirleyen "OnActivate" metodu çalıştırılır. Bu durumda etmen, kabuk tarafından kendisine atanan arayüz fonksiyonlarını henüz kullanamaz.

OnActivate: Etmen aktif olduğunda çalıştrılan metoddur.

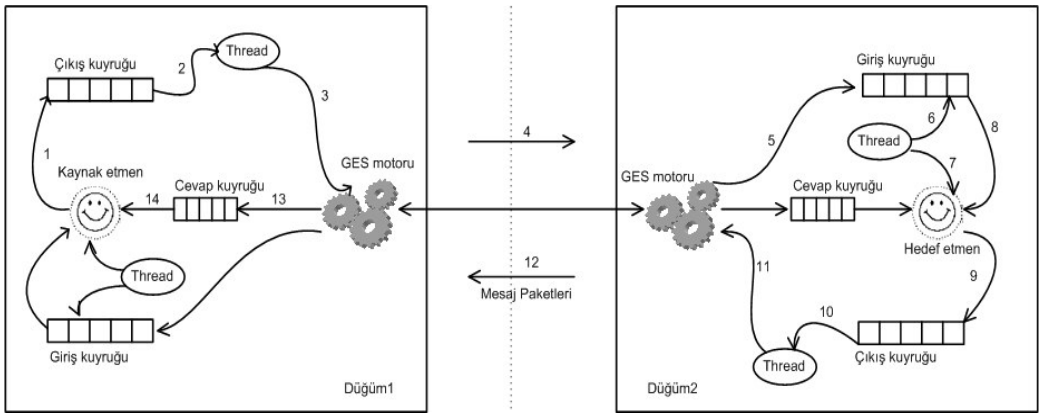
OnInactivate: Etmeni pasif duruma geçerirken çalıştırılan metoddur. Etmen pasif duruma geçerken bellekteki son durumu diske yazılarak saklanır. Pasif durumda olan bir etmen iletişim kuramaz, mesaj gönderip alamaz. Ayrıca, bu metod içinde mesaj göç etme istekleri de karşılanmaz.

OnEnd: Etmenin kendisini yok etmeye karar verdiği anda çalıştıran metoddur. "OnInactivate" metodundan sonra çalıştırılır. Bu metodun sonlanmasıyla etmen kodu, durum bilgisi ve politikası diskten silinmiş olur.

OnTransfer: Etmenin göç etmeden önce çalıştırdığı metoddur. Bu metod içinde mesaj alma ve gönderme mümkündür.

4.3 GES Haberleşme Altyapısı

GES mimarisi etmeni bloke etmeyen bir yapıya sahip olduğu için esnek bir yapıya sahiptir [7]. GES, SSL ve şifreleme tekniklerini kullanarak gizlilik, veri bütünlüğü ve kaynak kimliği denetimi gereksinimlerini; bütün iletişim aktivitelerinin tutulması ile de yalanlayamama gereksinimini karşılar.



Şekil 4.3: GES Mesajlaşma Mimarisi

Şekil 4.3'te mesaj gönderilmesi ve alınması sırasında gerçekleşen işlem adımları gösterilmiştir.

1- Gönderici (kaynak) etmen, mesaj göndereceği etmenin kimliğini elde eder. Mesajı gönderdikten sonra mesaj gönderici etmenin çıkış kuyruğuna yazılır.

2- Gönderici etmenin çıkış kuyruğunu gözleyen iplik yeni mesajdan haberdar olur.

3- Gönderici etmenin çıkış kuyruğunu gözleyen iplik, mesaj iletim isteğini GES motoruna aktarır.

4- GES motoru, alıcı etmenin bulunduğu düğüm üzerindeki GES motoru ile işbirliğine geçer.

5- Mesaj alıcı (hedef) etmenin giriş kuyruğuna yerleştirilir.

6- Alıcı etmenin giriş kuyruğunu gözleyen iplik, kuyruğa yeni bir mesajın geldiğinden haberdar olur.

7- Alıcı etmenin giriş kuyruğunu gözleyen iplik alıcı etmeni uyarır.

8- Alıcı etmen uyarıdan sonra giriş kuyruğundan mesajı çeker.

9- Alıcı etmen mesaja yanıtını (Mesaj yanıtları, mesaj paketinin sonuna özel bir işaret (ACK) konularak, mesaj gibi gönderilir.) çıkış kuyruğuna koyar(Her gelen mesaja yanıt verilmesi gerekmemektedir).

10- Alıcı etmenin çıkış kuyruğunu gözleyen iplik yanıttan haberdar olur.

11- Alıcı etmenin çıkış kuyruğunu gözleyen iplik, mesaj iletim isteğini GES motoruna aktarır.

12- GES motoru, gönderici etmenin bulunduğu düğüm üzerindeki GES motoru ile işbirliğine geçer.

13- Gönderici etmenin bulunduğu düğümdeki GES motoru mesaj yanıtını Cevap kuyruğuna koyar.

14- Etmen uygun gördüğü bir zamanda cevap kuyruğunu yoklayarak gönderdiği mesaja yanıt gelip gelmediğini kontrol edebilir.

Mesajlar ağ üzerinden RMI aracılığı ile iletildiği için "Serializable" olarak tanımlanmışlardır. Parametre listesi "Object" sınıfından türeyen herhangi bir nesne olabilir.

Mesaj nesneleri etmenler arasında mesaj paketleri olarak iletilir. Mesaj paketlerinin yapısı Şekil 4.4'te gösterilmiştir.

ID	Kaynak Düğüm	Kaynak Etmen	Hedef Düğüm	Hedef Etmen	ACK

Şekil 4.4: Mesaj Paketi Yapısı

4.4 GES Güvenlik Politikaları ve Yönetimi

Hareketli etmen sistemlerin güvenliği, etmen sistemlerinin kullanıldığı uygulamalar için kritik bir rol oynamaktadır. Güvenlik ile ilgili diğer bir konu; sistemlerin güvenlik ihtiyaçlarının dinamik olmasıdır. Örneğin, önceden kullanılması sistem için herhangi bir tehdit içermeyen kaynağın, çevre şartları değiştiği anda kullanılması sistemin güvenliği için sorun olabilir. Bu yüzden güvenlik yapısı değişen şartlara uyum gösterecek şekilde esnek olarak tasarlanmalıdır.

GES mimarisi, etmen ve düğümlere politikalar atanarak, etmen aktivitelerinin sınırlandırılabilmesine olanak sağlar. Düğümler de, düğüm politikaları oluşturularak korunur. Sistem bu desteği dinamik olarak vermektedir, dolayısı ile sistem aktif iken politikaları değiştirmek mümkündür. Bu çalışma modeli ile değişebilen çevresel şartlara göre etmenleri yeniden programlama gereksinimi de ortadan kalkmış olur [7].

GES, etmenler ve düğümler için olmak üzere iki ayrı politika tanımı sunar:

Etmen Politikaları: Etmen politikaları, etmen programcısının ya da düğüm yöneticisinin etmenlerin aktivitelerini sınırlandırması için kullanılır. Etmen politikaları şifreli XML dosyaları olarak saklanır ve etmen ağ üzerinde hareket ettikçe onunla birlikte taşınır. Etmen sahibi etmene ait politikayı çalışma anında görüntüleyebilir ve gerek duyarsa yürütme anında değiştirebilir.

Etmenlere politikalar aracılığı ile atanan izinler şu şekildedir:

- Mesaj gönderme izni.
- Mesaj alma izni.
- Başka bir düğüme göç etme izni.
- Etmenin kendini kopyalaması için gerekli izin.
- Etmenin diske yazması için gerekli izin.
- Etmenin diskten okuma yapması için gerekli izin.
- Etmenin ağ bağlantısı açması için gerekli izin.
- Etmenin ağ bağlantı alması için gerekli izin.
- Etmenin sistem değişkenlerine erişimi için gerekli izin.
- Etmenin sınıf yüklemesi için gerekli izin.

Etmen politikaları sadece etmenlere atanmamaktadır. Düğüm yöneticisi, isterse etmen politikalarını etmen sahiplerine göre düğümlere de atayabilir. Bu özellik sistemin esnekliğine büyük ölçüde katkı sağlamaktadır.

Düğüm Politikaları: Düğüm politikaları, düğüm kaynaklarının yetkisiz etmenler tarafından kullanılmasını engellemek için kullanılır. Sistem, yetkilendirme desteklediğini etmen kaynağı ve etmen sahibine göre verebildiği için oldukça esnek bir çalışma ortamı sunar [7]. Etmen sahibi etmenlerin yaratıldığı, etmen kaynağı etmenin son olarak çalıştığı GES sunucusunun adresidir.

5. BORSA YAPISI

Borsalar yatırımcıların menkul değerleri alıp sattıkları pazarlardır. Borsa ile ilgili açıklamalar aşağıda maddeler halinde sıralanmıştır.

5.1 Borsa Türleri

Broker Tabanlı Borsalar: Bu borsa yapısında brokerlar, yatırımcılar adına hisse senedi alıp satarlar. Borsa belirli bir konumda bulunur (örn. İMKB).

Satıcı Tabanlı Borsalar: Bu borsa tipinde alıcılar satıcılar ile ticaret yapar. Bu borsalar genellikle belirli bir konuma sahip değillerdir, alım ve satış işleri bilgisayar veya telefon aracılığı ile yapılır.

Teknolojideki ilerlemeler broker ve satıcı tabanlı borsalar arasındaki farkları giderek azaltmaktadır. Örneğin, Broker tabanlı borsalarda da, yatırımcılar telefon veya internet aracılığı alım ve satış yapabilmektedirler.

5.2 Gereksinim Duyulan Borsa Karakteristikleri

- Alım ve satış eldeki bilgiye dayanarak yapılır. Eski alım ve satış fiyatları, işlem hacmi ve güncel alım ve satış fiyatları yatırımcılara yardımcı olan bilgilerdir.
- Ticaret maliyeti düşük olmalıdır (Ticaret maliyeti, borsa sistemi veya brokerlara işlemler için verilen ücrettir.).
- Likidite: Yeni bir bilgi olmadığı sürece, hisseler daha önceki fiyatlarından çok farklı olmayan fiyatlardan, çok yüksek miktarlarda işlem görebilmelidir.

5.3 İstek Türleri

Pazar İsteği (Market Order): En çok kullanılan istek türüdür. Yatırımcılar pazardaki en iyi fiyattan alım veya satış yapmak isterler. En iyi fiyat, alım listesindeki en yüksek fiyatlı isteğin; satış listesinde ise en düşük fiyatlı isteğin fiyatıdır.

Limit İsteği: Yatırımcıların pazardaki en yüksek alım fiyatından daha düşük, pazardaki en düşük satış fiyatından daha yüksek bir fiyat ile yaptıkları isteklerdir. Yatırımcıların limit isteklerinin ne zaman karşılanacağı hakkında bir bilgisi yoktur.

5.4 Tacir Türleri

Aktif Tacirler: Genellikle pazar isteklerini kullanırlar. İvedi şekilde isteklerinin karşılanmasını isterler ve fiyatları kendi alışverişlerinin doğrultusuna çekerler.

Pasif Tacirler: Normalde limit isteklerini kullanırlar. Fiyatları dengede tutmaya çalışırlar. Satıcılar genellikle pasif tacirdir.

5.5 Borsada İsteklerin Eşleştirilmesi

Borsada isteklerin eşleştirilmesi sırasında Sürekli çift taraflı arttırma (Continuous double auction) metodu kullanılır.

Sürekli çifte arttırma metodunun özellikleri aşağıdaki gibidir:

- NASDAQ, NYSE ve İMKB gibi borsalarda kullanılan yöntemdir.
- Aynı anda hem alıcılar hem de satıcılar isteklerini bildirirler.
- İstekler arasındaki eşleştirmeler, sürekli devam eder.
- Eşleştirme alım listesindeki en yüksek fiyatlı ve satış listesindeki en düşük fiyatlı isteğe göre yapılır.

6. E-BORSA SİSTEMİ

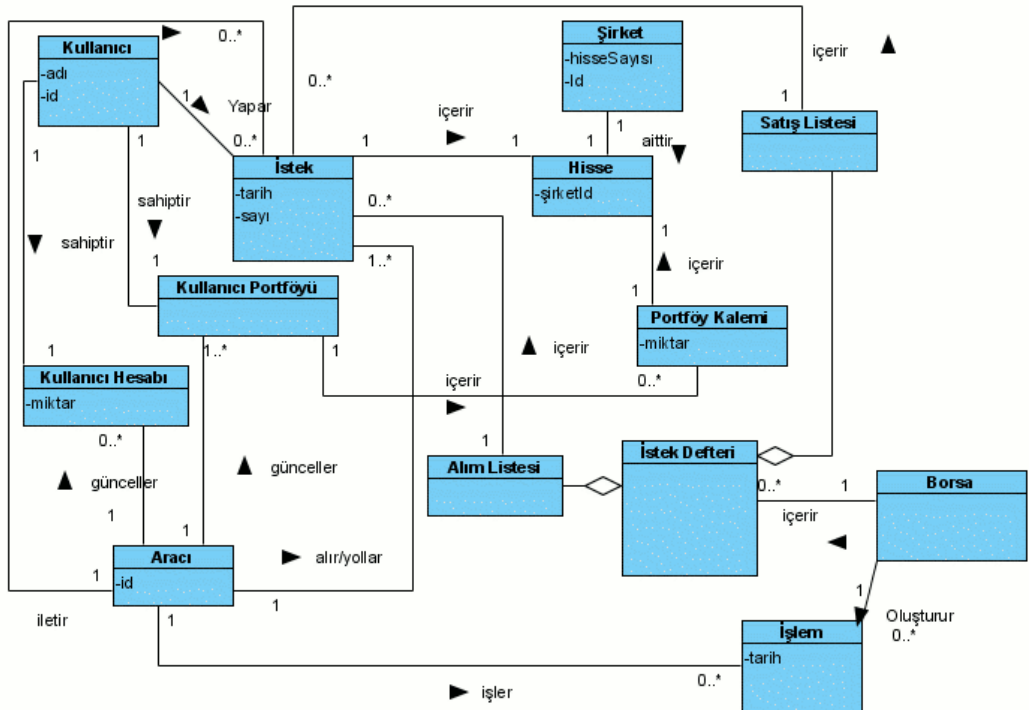
Tez kapsamında hareketli etmen sistemi ile gerçeklenen; dağıtık ortamlarda çalışabilme yeteneğine sahip olarak tasarlanan ve gerçeklenen elektronik borsa sistemi E-Borsa olarak adlandırılmıştır.

6.1 E-Borsa Tasarımı

Tasarıma sistem içindeki etmenlerin kullanacağı borsa sistemine ilişkin yapılar tasarlanarak başlanmıştır. Bu sınıflar "borsa" yazılım paketi içerisinde toplanmıştır.

6.1.1 Uygulama Etki Alanı (domain) Kavramsal Sınıfları

Uygulama etki alanında bulunan kavramsal sınıflar Şekil 6.1'de gösterilmiştir. "borsa" paketi içinde bulunan yazılım sınıfları bu sınıflardan yola çıkılarak yazılmıştır.



Şekil 6.1: Uygulama Etki Alanı Kavramsal Sınıfları

Kullanıcı:

Bu sınıf kullanıcıyı temsil etmektedir. Sistemde bulunan her kullanıcı, sahip olduğu menkul değerleri (hisse) barındıran bir portföye sahiptir. Ayrıca her kullanıcının hisse alımında kullanabileceği nakdi içeren bir kullanıcı hesabı vardır.

Kullanıcı Hesabı:

Kullanıcıya ait nakit miktarını tutan kavramsal sınıftır.

Kullanıcı Portföyü:

Kullanıcıya ait menkul değerleri tutan kavramsal sınıftır. Bir kullanıcı portföyü birden fazla Portföy Kalemi içerebilir.

Portföy Kalemi:

Bu kavramsal sınıf, portföyü oluşturan elemanları temsil eder. Her bir portföy kalemi, belirli bir şirkete ait hisse bilgisini tutar.

Hisse:

Menkul değeri temsil eden kavramsal sınıftır.

Şirket:

Menkul değerin ait olduğu şirketi temsil eder. E-Borsa sistemi, aynı anda birden çok şirkete ait hisselerin alınıp satılmasına imkan vermektedir.

İstek:

Kullanıcı tarafından oluşturulan ve menkul kıymet alım veya satış isteğini temsil eden kavramsal sınıftır.

Aracı:

Kullanıcılardan gelen istekleri borsaya ileten kavramsal sınıftır. Borsada isteklerin eşleşmesi sonucu oluşan işlemleri işler, işlem sonuçlarına göre işlemi oluşturmuş isteklere sahip kullanıcıların portföylerini ve hesaplarını günceller. Bu kavramsal sınıf sistemdeki Brokerlar'ı temsil etmektedir.

İşlem:

Kullanıcıların borsaya iletilen istekleri arasında bir eşleşme sonucunda oluşan ve alışverişin hangi miktar için ne kadardan, kimler arasında yapıldığını gösteren

kavramsal sınıftır. Daha önce de belirtildiği gibi, borsada oluşan bir işlem işlenmek üzere aracıya gönderilir.

Borsa:

Borsa mantığını oluşturan kavramsal sınıftır. Aracılar tarafından kullanıcılar adına gönderilen istekler burada eşleştirilir. Eşleşme olduğunda, alışveriş gerçekleşir ve alışveriş isteği yapan aracıya bildirilir.

İstek Defteri:

Bu kavramsal sınıf borsanın içinde bulunur. Gelen isteklerin karşılanıp karşılanamayacağı bu deftere bakılarak belirlenir. İstek karşılanıyorsa, işlem oluşturulur. Gelen isteği karşılayan ve defterde bulunan istekler güncellenir veya defterden kaldırılır. Yeni gelen istek karşılanmıyorsa veya kısmen karşılanmışsa (tamamı karşılanmamışsa) istek defterine eklenir. İstek defterinde Alım ve Satış listesi olmak üzere iki adet liste bulunur.

Alım Listesi:

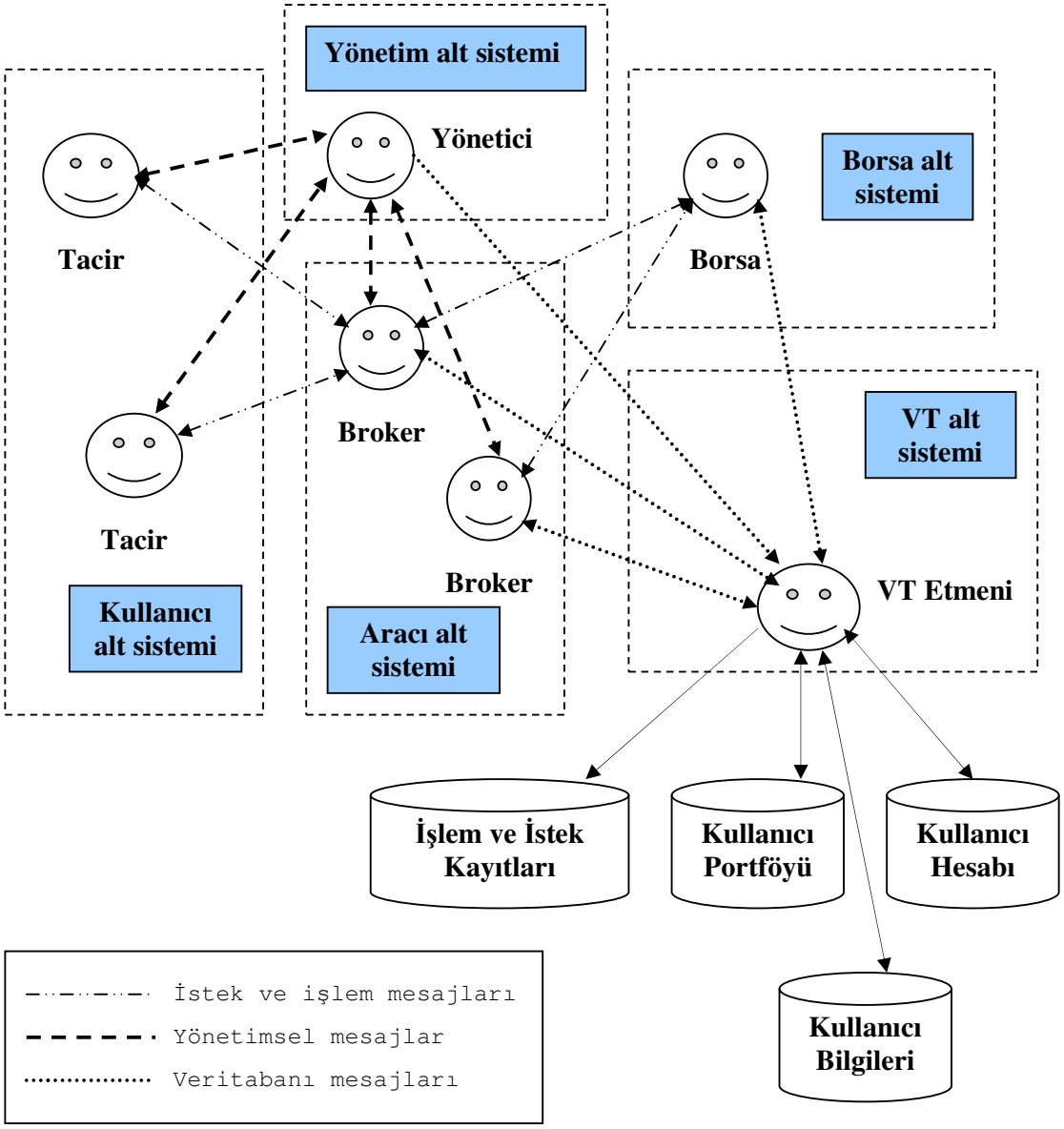
Alım isteklerinin tutulduğu listedir. Gelen satış istekleri alışveriş için bu listedeki istekler ile eşleştirilir. Tamamı veya belirli bir miktarı karşılanmayan alım istekleri de bu listeye yerleştirilir. Alım listesindeki istekler fiyata göre azalan sırada sıralanırlar. Listenin en tepesindeki istek, en yüksek fiyatı veren kullanıcıya ait istektir.

Satış Listesi:

Satış isteklerinin tutulduğu listedir. Gelen alım istekleri alışveriş için bu listedeki istekler ile eşleştirilir. Tamamı veya belirli bir miktarı karşılanmayan satış istekleri de bu listeye yerleştirilir. Satış listesindeki istekler fiyata göre artan sırada sıralanırlar. Listenin en tepesindeki istek, en düşük fiyatı veren kullanıcıya ait istektir.

Uygulama etki alanında bulunan sınıflar yazılım etki alanında gerçeklenirken ağ üzerinde taşınan nesnelerin yaratıldığı sınıflar “Serializable” arayüzünü gerçekleyecek şekilde yazılmıştır.

6.1.2 Sistemin GES Sistemi Kullanılarak Modellenmesi



Şekil 6.2: Etmen Tabanlı E-Borsa Modeli

Şekil 6.2'de de gösterildiği gibi E-Borsa sistemi 5 adet alt sistemden oluşmaktadır. Sistem hareketli etmen sisteminin dağıtık yapısını en etkin şekilde kullanabilmek için, farklı görevleri yerine getirecek alt birimlere ayrılmıştır. Her alt birim (sistem) içerisinde, alt birimin görevlerini yerine getirecek etmenler bulunmaktadır. E-borsa sisteminde bulunan alt sistemlerin tanımı ve görevleri aşağıda listelenmiştir:

Kullanıcı Alt Sistemi:

Bu alt sistem, kullanıcılardan isteklerini alan Tacir etmenlerden oluşmaktadır. Tacir etmenler sorumlu oldukları kullanıcılardan alım veya satış isteğini aldıktan sonra, kendilerine daha önceden bir Broker atanmışsa, aracı sistemdeki Broker etmeni ile etkileşime geçerek isteklerini Broker etmenine yollarlar. Tacir etmene daha önceden herhangi bir Broker etmen atanmamışsa, Tacir etmenler isteklerini hangi Broker etmene ileteceklerini öğrenmek için, Yönetim alt sisteminde bulunan Yönetici etmeni ile temasa geçerler, Yönetici etmen tarafından kendilerine atanan Broker etmenine isteklerini yollarlar. Kullanıcı etmenlerinin, Yönetici etmenine başvurmadan, sistemdeki Broker etmeni ile etkileşime geçmeleri ve mesajlaşmaya başlamaları mümkün değildir.

Aracı Alt Sistemi:

Bu sistem Tacir etmenler tarafından yollanan istekleri karşılar, değerlendirir ve isteğin karşılanabilir olduğuna karar verdiğinde işlemi borsa alt sistemine iletir.

Alt sistem Broker etmenlerinden oluşur. Broker etmenin görevi, Tacir etmenlerinin yolladıkları isteklerin karşılanabilir olup olmadığını denetlemek ve eğer istek karşılanabiliyorsa bu isteği Borsa etmenine yani borsa alt sistemine geçirmektir.

Yönetim Alt Sistemi:

Bu alt sistem Yönetici etmeni barındırır. Sistemde tek bir Yönetici etmen vardır, bu sayede sistem tek bir merkezden yönetilebilir. Yönetici etmenin görevi Tacir ve Broker etmenlerini yaratmak, Tacir etmenlerin isteklerini iletebilmeleri için Tacir-Broker atamalarını gerçekleştirmektir.

Borsa Alt Sistemi:

Bu sistem Aracı sistemden Broker etmeni aracılığı ile gelen istekleri karşılar ve birbiri ile eşler. Bir eşleşme olduğu takdirde ilgili eşleşmeyi, Broker etmeni aracılığı ile Aracı alt sisteme bildirir.

VT (Veritabanı) Alt Sistemi:

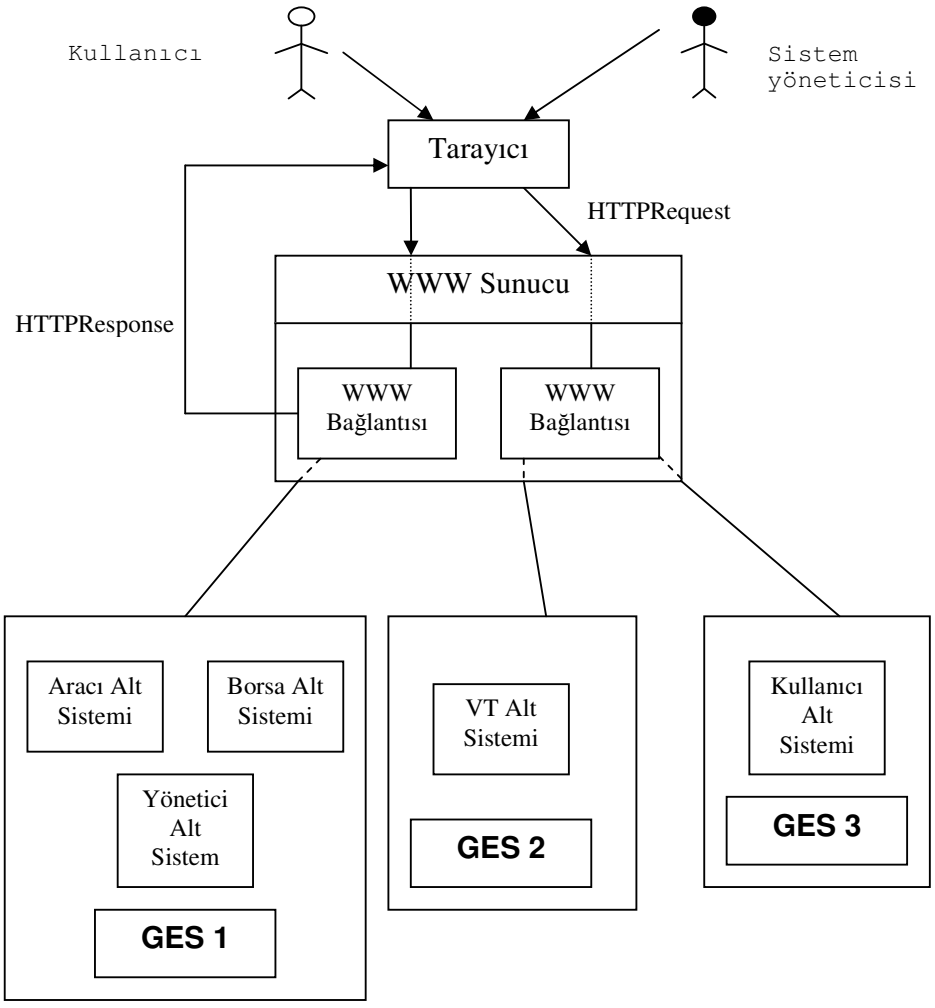
Bu sistem kullanıcı hesap ve portföy bilgilerinin güncellenmesi, borsaya gönderilen istekler ile borsada gerçekleşen işlemlerin izlenmesi ve yönetimsel amaçlar için kullanılmaktadır.

Borsa etmeni gerçekleşen işlemleri kaydettirmek için; Broker etmeni istekler sırasında kullanıcılara ait hesap ve portföy bilgilerini değiştirmek ve istekleri kaydetmek için; Yönetici etmeni ise kullanıcı yönetimi sırasında doğrulama işlemleri için Veritabanı etmenini kullanır.

Sistemde birden çok veritabanı etmeni bulunabilir. Sistemde bulunan tüm etmenler yalnızca VT etmenini kullanarak veritabanına erişebilirler.

Her alt sistem farklı GES sunucuları üzerinde çalışabileceği gibi, birden fazla alt sistem aynı GES sunucusu üzerinde de çalışabilir. Daha önce Bölüm 4.1.1'de de bahsedildiği gibi, GES sunucularının birbirleri ile iletişime geçmesi için bir adet GGES (Gözleyici GES) ve bir adet GYGES'e (Güvenlik Yöneticisi GES)ihtiyaç vardır. GGES ve GYGES olarak herhangi bir alt sistemin bulunduğu GES sunucusu veya üzerinde herhangi bir alt sistem bulunmayan bir GES sunucusu kullanılabilir.

Kullanıcılar isteklerini Tacir etmenlere web arayüzü ile ulaştırırlar. Ayrıca daha sonra ayrıntıları ile anlatılacak olan sistem yönetimine ilişkin işlemler de web arayüzü aracılığı ile yapılır. Kullanıcıların ve sistem yöneticisinin de sistemle etkileşimi göz önüne alındığı sistem mimarisi Şekil 6.3'te gösterilmiştir.



Şekil 6.3: E-Borsa Mimarisi

Şekil 6.3'te alt sistemler 3 adet GES sunucusuna dağıtılmıştır. Bu dağıtım örnek verme amacı ile yapılmıştır. Böyle bir yapılandırmanın gerçekleştirilmesi zorunlu değildir. Daha önce de belirtildiği gibi sistem oldukça esnek olduğu için, alt sistemlerin GES sunucular üzerine dağılımı, performansın en iyilenmesi göz önüne alınarak sistem yöneticisi tarafından düzenlenecektir.

6.2 E-Borsa Etmenleri

E-Borsa sisteminin temelini alt sistemlerde çalışan etmenler oluşturmaktadır. Etmenler kısım 4.2'de belirtilen etmen şablonu kullanılarak yaratılmıştır. Sistemde bulunan etmenler, bu etmenlerin görevleri ve yaratılma şekilleri şöyledir:

6.2.1 Tacir (Kullanıcı) Etmeni

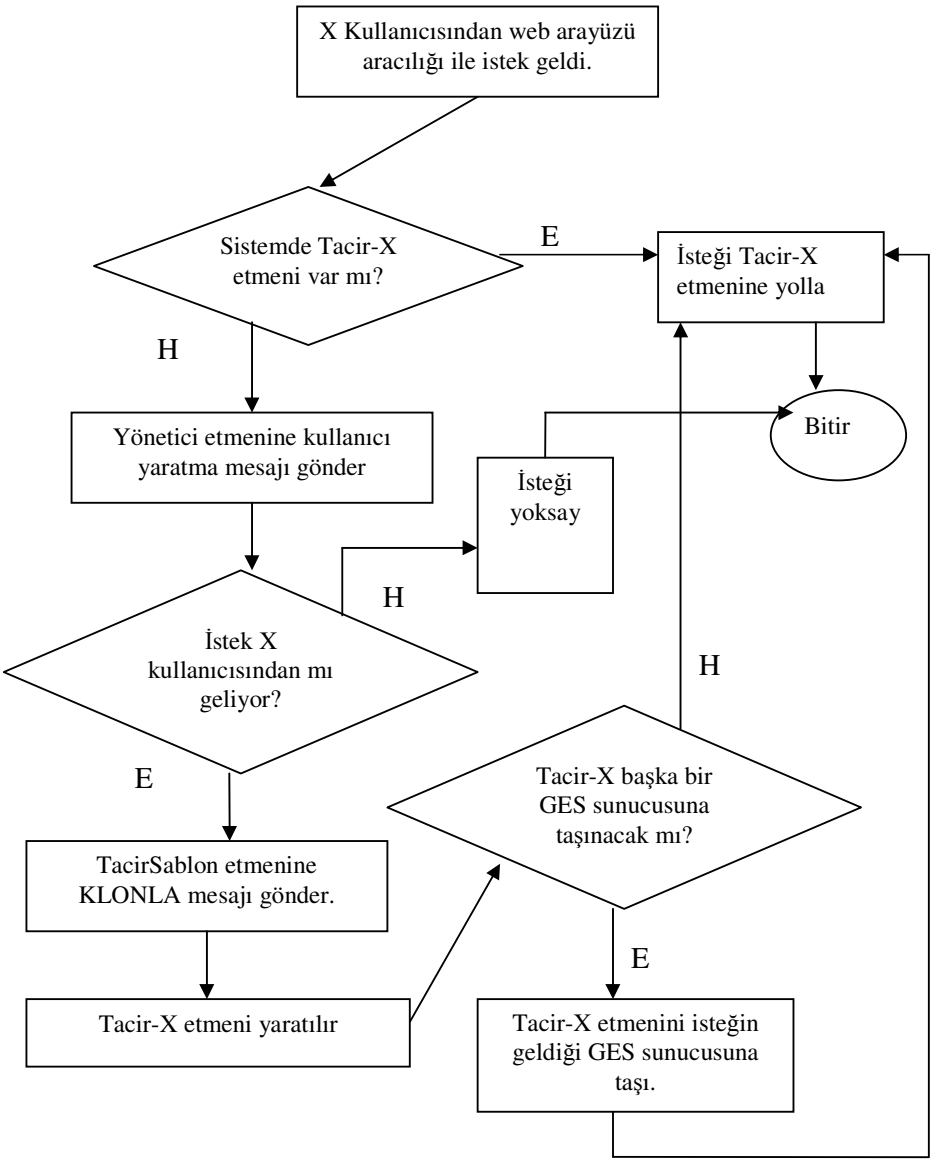
Bu etmenler sisteme kendilerini "Tacir-<kullanıcı adı>" ismi ile duyururlar. "kullanıcı adı" her bir kullanıcı için eşsiz olan özelliktir. Sistemde aynı anda bir kullanıcıya ait birden fazla Tacir etmen bulunmaz.

Görevi:

Tacir etmenler, kullanıcılardan web tarayıcısı aracılığı gelen istekleri alır ve Broker etmenine iletirler. Diğer bir deyişle sistemde kullanıcıları temsil ederler.

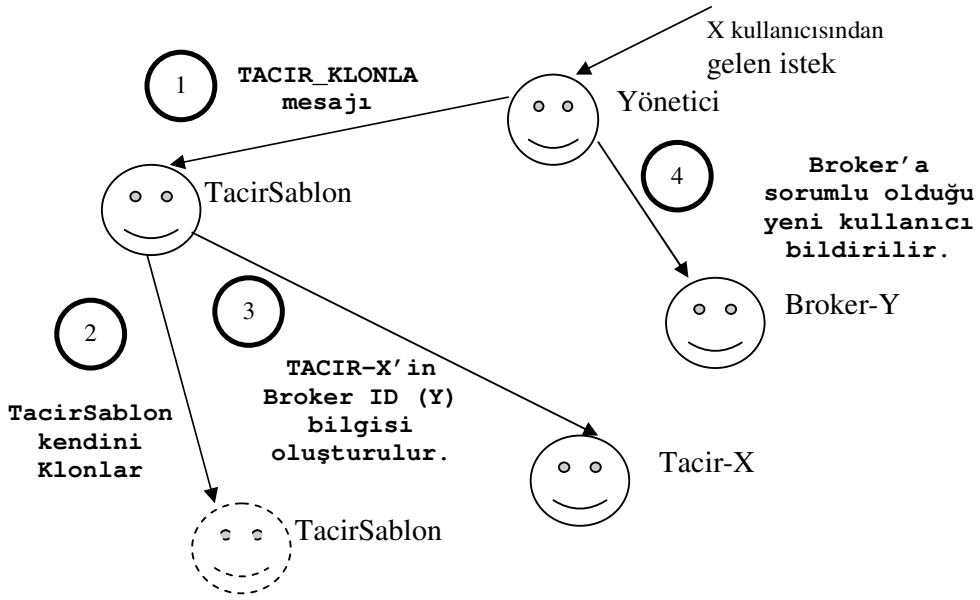
Yaratılması ve Çalışması:

Sistem ilk defa başlatıldığında, sistemde herhangi bir kullanıcı için Tacir etmen bulunmamaktadır. Sistemde başlangıçta "TacirSablon" ismi ile kendisini sisteme duyuran Tacir etmeni bulunur. Kullanıcı istek yapacağı anda sistemde o kullanıcıyı temsil eden bir Tacir etmen bulunmuyorsa, Yönetici etmeni uyarılır (Yönetici etmenine uyarı gönderilirken kendisi adına Tacir etmen yaratılacak kullanıcıya ait kullanıcı adı ve parola bilgileri yollanmalıdır. Ancak bu şekilde Yönetici etmen, isteğin yaratacağı kullanıcı etmenden geldiğini anlayabilir.); Yönetici etmeni "TacirSablon" etmenine yeni kullanıcı eklemek için gerekli mesajı yollar. Bu mesaja parametre olarak kullanıcı adı, -yine eşsiz bir değer olan- kullanıcı ID ve opsiyonel olarak kullanıcının istek yaptığı konağın adresi (Bu adres kullanıcının üzerinden istek yaptığı GES sunucusunun kendisini kaydettirdiği RMI adresidir.) gönderilir. Yeni bir Tacir etmen oluşturulurken adres gönderilmesi ile amaçlanan; her Tacir etmenin kullanıcısının isteklerini yaptığı GES sunucusu üzerinde çalışmasını sağlamaktır. Bu şekilde, Tacir etmenler GES sunucular arasında dağıtılacak; kullanıcı gruplarının farklı konaklar kullanarak sisteme giriş yapması sağlanarak, sistem üzerinde Tacir etmen aktivitesinden kaynaklanan yükün dağıtılması gerçekleştirilmiş olacaktır. Şekil 6.4'te sistemde yeni bir Tacir etmen oluşturulmasına ilişkin işlem adımları gösterilmiştir.



Şekil 6.4: Tacir Etmen Oluşturma Mekanizması

Tacir etmen oluşturulurken etmenler arasında oluşan mesajlaşmalar Şekil 6.5'te gösterilmiştir. 2. adımda sistemde mevcut olan TacirSablon etmenin kopyalanması ile oluşan TacirSablon etmeni, sistemdeki yeni TacirSablon olur. Eski TacirSablon ise TACIR_KLONLA mesaj parametreleri ile kendisine bildirilen kullanıcının Tacir etmeni (Tacir-<kullanıcı adı> şeklinde) olur.



Şekil 6.5: Tacir Etmen Yaratma İşlemleri

Tacir etmene ait "Hashtable" tipinden "Istekler" üyesi, etmenin göndermesi gereken istekleri tutar. Normal şartlarda, Tacir etmene gelen istek gelir gelmez, Broker etmene gönderilir. Ancak Tacir etmene Broker atanmamış olması durumu veya sistemde oluşabilecek herhangi bir hatadan ötürü alınan istek Broker'a iletilemezse, gelen istek daha sonra gönderilmek üzere "Istekler" tablosuna atılır. "Istekler" değişkeninin "Hashtable" olarak atanmasının nedeni, "Hashtable" sınıfının metodlarının "synchronized" olarak tanımlanmış olması ve bu sayede "Istekler" dizisine eş zamanlı erişimlere izin verilmesidir. "Istekler" dizisine, iki ayrı iplik içerisinde çalışan "OnActivate" ve "OnMeesageArrive" metodları içerisinde erişildiği için, oluşabilecek eş zamanlı erişimler için bu önlem alınmıştır.

Aldığı Mesajlar:

Tacir etmenin aldığı mesajlar Tablo 6.1'de listelenmiştir.

Tablo 6.1: Tacir Etmenin Aldığı Mesajlar

Mesaj Adı	Mesaj İşlevi	Gönderici
TACIR_KULLANICIDAN_ISTEK_AL	Tacir'e gelen bir isteği bildirir.	Web arayüzü aracılığı ile kullanıcı.
TACIR_KLONLA	TacirSablon etmenine kendisini kopyalamasını bildirir.	Yönetici Etmen.
TACIR_BROKER_KIMLIGI_AL	Tacir etmene atanan Broker'ı bildirir.	Broker Etmeni

Gönderdiği Mesajlar:

Tacir etmenin gönderdiği mesajlar Tablo 6.2'de listelenmiştir.

Tablo 6.2: Tacir Etmenin Gönderdiği Mesajlar

Mesaj Adı	Mesaj İşlevi	Alıcı	Mesaj Yanıtı
TACIR_ISTEK	Yapılan alım-satış isteğini Broker'a gönderir.	Tacir etmenden sorumlu Broker etmeni.	Alındı mesajı.
YONETICI_UYGUN_BROKER_VER	Yönetici etmeninden, kendisine yeni bir Broker atanmasını talep eder.	Yönetici Etmen.	Yok

Tacir etmenler yolladıkları isteklerin Brokera ulaşıp ulaşmadığını öğrenmek için beklemek zorunda değildir. Etmen arayüzünün (AgentInterface) sağladığı "setRepliedMessageHandler" metodu kullanılarak, gönderilen mesajların yanıtını işleyecek metod belirlenir. Tacir etmen için yanıtları işleyecek metod "isteklerDurumunuGuncelle" metodudur:

```
public void isteklerDurumunuGuncelle(MessageID mid)
{
try
{
BrokerIstekYaniti biy = (BrokerIstekYaniti)
      getAgentInterface().getReply(mid);
if(biy != null)
Istekler.remove(new Integer(biy.getIstekId()));
}
catch (ClassCastException e) {
//Gelen mesaj yanıtı geçerli değilse herhangi bir şey yapılmayacak.
}
}
```

Görüldüğü gibi yanıtları işleyen metod sadece TACIR_ISTEK mesajları için gelen yanıtları ("BrokerIstekYaniti" tipinden) işlemektedir.

6.2.2 Broker Etmeni

Bu etmenler sisteme kendilerini "Broker-<Broker ID>" ismi ile duyururlar. "Broker ID" değeri rasgele üretilen 14 haneli bir katardır. Her bir broker değeri için ID değeri eşsizdir. Sistemde aynı anda birden çok Broker etmeni bulunabilir. Daha önce bahsedildiği gibi Broker etmenleri aracı mantıksal alt sistemini oluşturur. Broker

etmenleri farklı konaklarda hizmet verebilirler. Broker etmenlerinin farklı konaklara taşınması sistem yöneticisi tarafından sağlanır. Broker etmenleri BrokerSablon etmenlerinin kendilerini kopyalaması ile oluşturulurlar. Broker etmenleri, sistem yöneticisi tarafından yönetim arayüzü kullanılarak yaratılır.

Görevi:

Broker etmenleri aşağıdaki görevleri yerine getirirler.

- Tacirlerden gelen istekleri alıp, kullanıcıların istekleri yapıp yapamayacağını değerlendirirler. Kullanıcı istek yapacak şartlara sahip ise isteği Borsa etmenine iletirler.
- Kullanıcının yaptığı istek için kullanıcı hesabını veya portföyünü bloke ederler. Bu işlemi VT etmeni ile mesajlaşarak gerçekleştirirler.
- Gerçekleşen ve gerçekleşemeyen (hesap bakiyesi yetersizliği veya portföy durumunun uygun olmaması nedenleriyle) isteklerin izlerinin tutulmasını sağlarlar. Bu işlem için de VT etmeni ile etkileşime geçerler.
- Borsada yaptıkları istek için bir işlem oluştuğunda borsadan gelen işleme göre, kullanıcının hesap ve bloke hesap bilgisini ve/veya portföy ve bloke portföy bilgisini güncellerler.

Yaratılması ve Çalışması:

Broker etmenleri, sistem yöneticisi tarafından yönetim arayüzü kullanılarak yaratılırlar. Broker etmenleri sistemde başlangıçta bulunan BrokerSablon etmenin kopyalanması aracılığı ile yaratılır. Yönetici etmen BrokerSablon etmenine "BROKER_KLONLA" mesajı ile birlikte, yarattığı 14 haneli eşsiz Broker ID değerini gönderir. BrokerSablon etmeni kendini kopyaladıktan sonra, ismini Broker-<Broker ID> olarak değiştirir. Sistemdeki yeni BrokerSablon etmeni, önceki BrokerSablon etmeninin kopyasıdır.

Aldığı Mesajlar:

Broker etmenin aldığı mesajlar Tablo 6.3'te listelenmiştir.

Tablo 6.3: Broker Etmenin Aldığı Mesajlar

Mesaj Adı	Mesaj İşlevi	Gönderici
BROKER_KULLANICI_EKLE	Broker'a yeni kullanıcısını (tacirini) bildirir.	Yönetici Etmen
BROKER_KULLANICI_CIKAR	Kullanıcı listesinden çıkacak kullanıcıyı bildirir.	Yönetici Etmen.
BROKER_ISLEM_AL	Borsada oluşan bir işlem Broker etmenine bildirilir.	Borsa Etmeni
BROKER_TASIN	Broker'a farklı bir konağa taşınması için istek yapar.	Yönetici Etmen
TACIR_ISTEK	Broker'a Tacir'den yapılan isteği bildirir.	Tacir Etmen
BROKER_KLONLA	BrokerSablon etmeninden yeni bir Broker etmeni yaratılması için istek bildirir.	Yönetici Etmeni

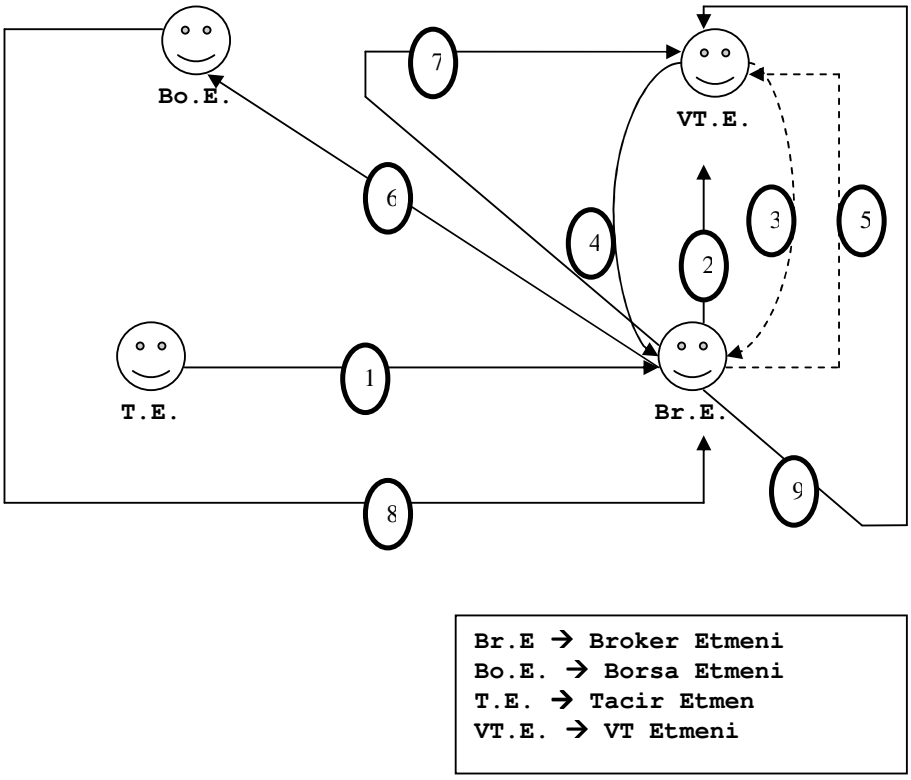
Gönderdiği Mesajlar:

Broker etmenin gönderdiği mesajlar Tablo 6.4'te listelenmiştir.

Tablo 6.4: Broker Etmenin Gönderdiği Mesajlar

Mesaj Adı	Mesaj İşlevi	Alıcı	Mesaj Yanıtı
BORSA_ISTEK	Tacir etmenden gelen isteği, Borsa etmenine iletir.	Borsa Etmeni	Alındı mesajı.
TACIR_BROKER_KIMLIGI_AL	Tacir etmene kullanıcıyı kullanıcılar listesine eklediğini bildirir.	Tacir Etmen	Yok
VT_BLOKE_ET	Tacir etmenden gelen istek üzerine, kullanıcı hesap ve/veya portföyünde istek için gerekli olan güncellemeleri bildirir.	VT Etmeni	"VTEtmeniYaniti" tipinden yanıt. Bu yanıt sayesinde Broker etmeni, yapılan bloke etme isteğinin başarılı olup olmadığını öğrenebilir.
VT_ISTEK_LOGLA	Başarı ile borsaya gönderilen istekler ile ilgili bilgileri VT etmenine bildirir.	VT Etmeni	Yok
VT_HESAP_PORTFOY_GUNCELLE	Borsa tarafından daha önce gönderilen istek için bir işlem oluştuğunda, oluşan işleme göre isteği yapan kullanıcının hesap portföy bilgisinin güncellenmesi için, işlem ile ilgili bilgileri VT etmenine bildirir.	VT Etmeni	Yok

Broker etmenine bir Tacir etmenden, istek geldiğinde gerçekleşen işlemler Şekil 6.6'da gösterilmiştir.



Şekil 6.6: İstek İşleme Döngüsü

1-2-4-6-7-8-9 adımları ile başarı ile gerçekleşmiş bir işlem gösterilmiştir. Adımların açıklamaları şu şekildedir:

1- Tacir etmen "TACIR_ISTEK" mesajı ile alım veya satış isteğini, kullanıcıdan sorumlu Broker etmenine iletir.

2- Broker etmen kendisine gelen isteğin karşılanabilir olup olmadığını öğrenmek ve istek karşılanabilir ise kullanıcıya ait hesap ve portföy bilgisinin güncellenmesi için "VT_BLOKE_ET" mesajını istek parametresi ile birlikte VT etmenine gönderir. Yapılan istek alım isteği ise, VT etmeni kullanıcının hesabından alım isteği için gerekli olan tutarı düşüp, hesaptan düşülen tutarı kullanıcının bloke hesabına ekler. Yapılan istek satış isteği ise; kullanıcının portföyünde bulunan istekle belirtilen şirkete ait, belirtilen miktardaki hisse, kullanıcı portföyünden bloke portföye aktarılır.

4- VT Etmeni isteğin geçerli olduğunu ve isteğin yapılması için gerekli bloke etme işleminin yapıldığını Broker etmene bildirir.

6- Broker etmeni "BORSA_ISTEK" mesajı ile yapılan isteği Borsa etmenine gönderir. Borsa etmeni gelen isteği istek defterine kaydeder.

7- Broker etmeni Borsa etmenine ulaştırılan istekle ilgili bilgileri, izleme tablosuna kaydedilmek üzere "VT_ISTEK_LOGLA" mesajı ile VT etmenine gönderir.

8- Borsa etmeni daha önce yollanan istek karşılandığı zaman, eşleşen istekleri yollayan Broker etmenlerini gerçekleşen işlemden haberdar eder. Borsa etmeni Broker etmenlerini yapılan işlemden haberdar etmek için Broker etmenlerine "BROKER_ISLEM_AL" mesajını yollar.

9- Broker etmeni, Borsa etmeni tarafından kendisine bildirilen işlemi inceleyerek, VT etmenine kullanıcı portföyü veya kullanıcı hesabını güncellemesi için "VT_HESAP_PORTFOY_GUNCELLE" mesajını yollar. Gerçekleşen işlem yapılan alım isteği için gerçekleşmişse, bloke kullanıcı hesabından alım için harcanan tutar düşürülür, alım yapılan hisse işlemde bildirilen miktar kadar kullanıcı portföyüne eklenir. Gerçekleşen işlem satış isteği için ise, bloke kullanıcı portföyünden işlem ile oluşan miktar kadar belirli hisse düşülür, işlem sonucu elde edilen tutar kullanıcı hesabına eklenir.

1-2-3-5 adımları ile başarı ile gerçekleşememiş bir işlem gösterilmiştir. Adımların açıklamaları şu şekildedir:

1- Tacir etmen "TACIR_ISTEK" mesajı ile alım veya satış isteğini, kullanıcıdan sorumlu Broker etmenine iletir.

2- Broker etmen kendisine gelen isteğin karşılanabilir olup olmadığını öğrenmek ve istek karşılanabilir ise kullanıcıya ait hesap ve portföy bilgisinin güncellenmesi için "VT_BLOKE_ET" mesajını istek parametresi ile birlikte VT etmenine gönderir.

3- VT etmeni istek karşılanamıyorsa bunu Broker etmenine bildirir. Yapılan istek alım isteği ise ve kullanıcı hesabında bu isteği karşılayacak kadar nakit bulunmuyorsa, o istek gerçekleştirilmez. Yapılan istek satış isteği ise ve kullanıcının portföyünde bu isteği karşılayacak kadar hisse bulunmuyorsa, o istek gerçekleştirilmez.

5- Broker etmeni gerçekleşmeyen istekle ilgili bilgileri, izleme tablosuna kaydedilmek üzere "VT_ISTEK_LOGLA" mesajı ile VT etmenine gönderir.

Broker etmeni yolladığı mesajların yanıtlarını beklemeden işlemlerine devam eder, Yapılan isteklerin ne durumda olduğunu (isteklerin hangi işlem adımında olduğu) , etmenin "OnActivate" metodu içinde bir döngü içerisinde kontrol edilir. Broker etmenin de Tacir etmenler gibi "setRepliedMessageHandler" metodunu kullanarak kendisine gelen yanıtların "istekleriYonet" metodu kullanılarak işlenmesi sağlanmıştır. Gelen her yanıt farklı bir iplik içinde işlendiği için eş zamanlı erişim için gerekli önlemler alınmıştır.

```
public void istekleriYonet(MessageID mid)
{
  try {
    Object gelenYanit = getAgentInterface().getReply(mid);
    if ( gelenYanit instanceof VTEtmeniYaniti )
    {
 VTEtmeniYaniti vtey =  (VTEtmeniYaniti)gelenYanit;
if(vtey.getMesajAdi().equalsIgnoreCase(Yardimci.MesajAdlari.VT_BLOKE
_ET) )
   blokeEdileniIsle(vtey);
    }
    else if ( gelenYanit instanceof Integer )
    {
 Integer biy = (Integer)gelenYanit;
 borsaIstekYanitiIsle(biy);
    }
   }
   catch (Exception e) { e.printStackTrace();
    }
}
```

Yukarıdaki kod bloğunda istekleri işleyen, "blokeEdileniIsle" ve "borsaIstekYanitiniIsle" metodları, karşılıklı dışlama gerektiren durumlar göz önüne alınarak "synchronized" olarak tanımlanmıştır.

6.2.3 Borsa Etmeni

Borsa etmeni, istek eşlemelerini yapan borsa mantığının bulunduğu etmendir. Sistemde bir anda bir tek Borsa etmeni bulunabilir.

Görevi:

Borsa etmeni, Broker etmenler tarafından kendisine gönderilen istekleri alır ve istek defterini kullanarak isteklerin eşleştirilmesini sağlar. Herhangi bir istek eşleşmesinde, oluşan işlemi, istekleri yapan Brokerlar'a bildirir.

Aldığı Mesajlar:

Borsa etmeninin aldığı mesajlar Tablo 6.5'te listelenmiştir.

Tablo 6.5: Borsa Etmeninin Aldığı Mesajlar

Mesaj Adı	Mesaj İşlevi	Gönderici
BORSA_ISTEK	Borsaya gelen bir isteği bildirir.	Broker Etmeni
BORSA_DEFTER_VER	Borsa istek defterinin son durumunu bildirir.	Web arayüzü aracılığı ile kullanıcı veya sistem yöneticisi.
BORSA_ISTEK_IPTAL_ET	ID'si verilen isteği istek defterinden çıkarır.	Web arayüzü aracılığı ile kullanıcı.

Gönderdiği Mesajlar:

Borsa etmeninin gönderdiği mesajlar Tablo 6.6'da listelenmiştir.

Tablo 6.6: Borsa Etmeninin Gönderdiği Mesajlar

Mesaj Adı	Mesaj İşlevi	Alıcı	Mesaj Yanıtı
BROKER_ISLEM_AL	Oluşan bir işlemi istekleri yapan Brokerlara bildirir.	Broker Etmeni	Yok
VT_ISLEM_LOGLA	Gerçekleşen bir işlem ile ilgili bilgileri, veritabanına kaydedilmesi için VT etmenine gönderir.	VT Etmeni	Yok
YONETICI_UYGUN_BROKER_VER	Daha önce istek bildiren bir Broker artık sistemde bulunamıyorsa, istek ile ilgili kullanıcıya ait güncel Broker'ı bulmak için kullanılır.	Yönetici Etmen	İsteğin sahibi olan kullanıcıya atanan yeni Boker'ın ID bilgisi.
VT_ISTEK_IPTAL_ET	İstek defterinden iptal edilerek çıkarılan isteği yapan kullanıcının hesap ve portföy bilgilerinin güncellenmesi için kullanılır.	VT Etmeni	Yok

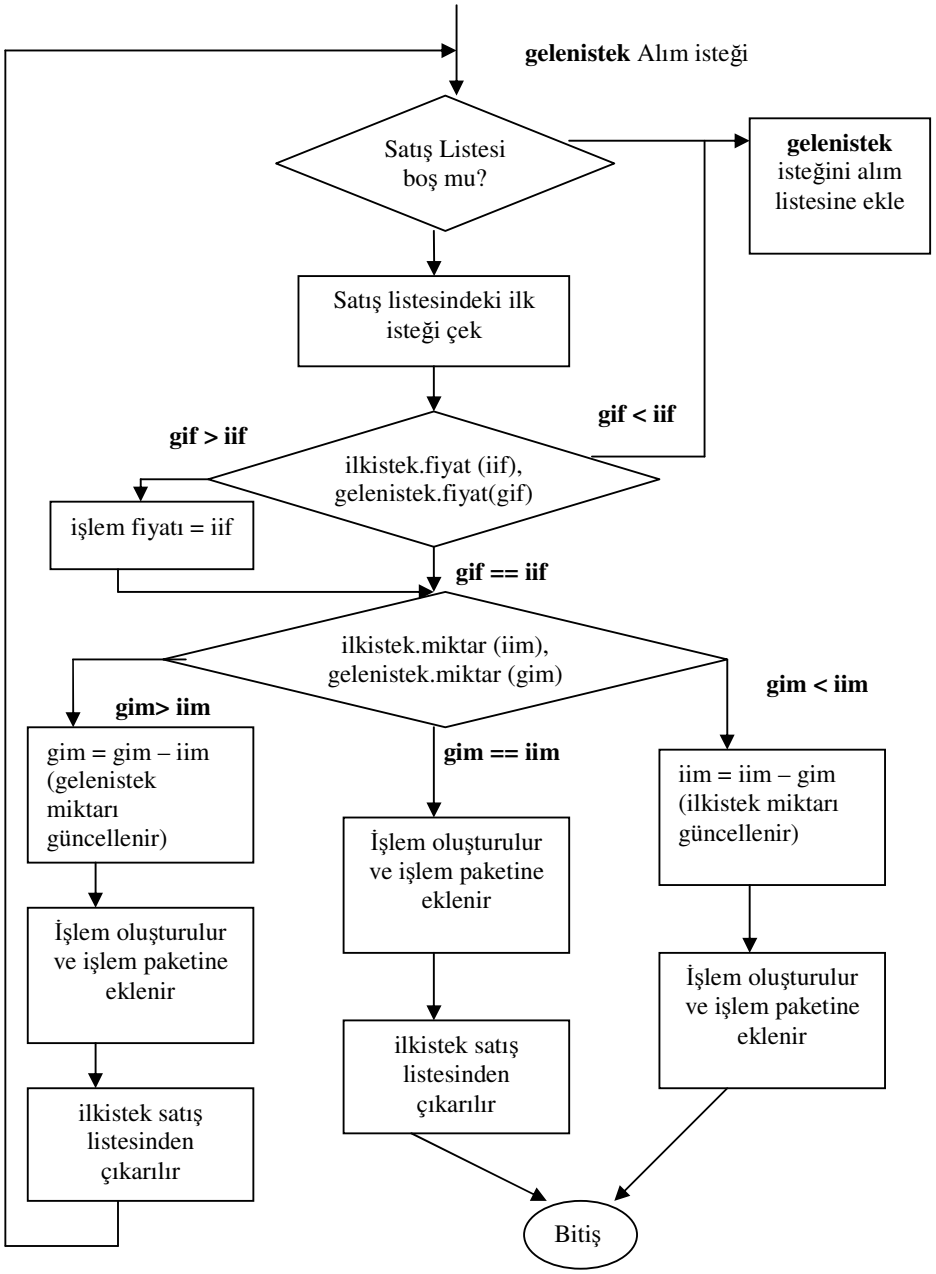
Borsa etmeni gelen istekleri istek defterinde tutar. Gelen satış istekleri, istek defterinin alım listesindeki istekler ile; gelen alım istekleri istek defterinin satış listesindeki istekler ile eşleştirilir. İstek eşleme yordamı özyinemeli (rekürsif) olarak tanımlanmıştır. Borsaya gelen alım ve satış istekleri için, istek eşleştirme yordamları sırası ile Şekil 6.7 ve Şekil 6.8'de açıklanmıştır. Diyagramlarda kullanılan "gelenistek", borsaya gelen isteği, "ilkistek" ise, alım ve satış listelerinin en başında bulunan isteği temsil etmektedir. Borsada her bir şirket için ayrı bir istek defteri tutulmaktadır. İstek defteri yazılım etki alanında "IstekDefteri" sınıfı ile temsil

edilmektedir. Bu sınıfın alım ve satış listesini temsil eden iki adet üyesi vardır. İsteklerin sıralanması için algoritma geliştirmemek için bu üyeler "TreeSet" olarak tanımlanmıştır. Bu şekilde kümeye yeni bir istek eklendiğinde isteklerin otomatik olarak sıralanması sağlanmıştır. "TreeSet" tipinden bir sınıfa eklenen elemanların sıralanabilmesi için elemanların bir sıralama şartına sahip olması gereklidir. Borsadaki istekler önce fiyatlarına göre sıralanırlar, fiyatların aynı olduğu durumda, istekler geliş zamanına göre sıralanırlar. En yüksek fiyata sahip alım isteği listenin en üst sırasında iken, en düşük fiyata sahip satış isteği listenin en üst sırasındadır. Aynı fiyatlı istekler için, önce gelen istek sonra gelen isteğe göre üst sırada yer alır. İsteklerin karşılaştırılabilir olması için "Istek" sınıfı "Comparable" olarak tanımlanmıştır. "Comparable" arayüzünü kullanan bir sınıfın bu arayüze ait "compareTo" işlevinin içinde karşılaştırma şartlarını tanımlaması gereklidir.

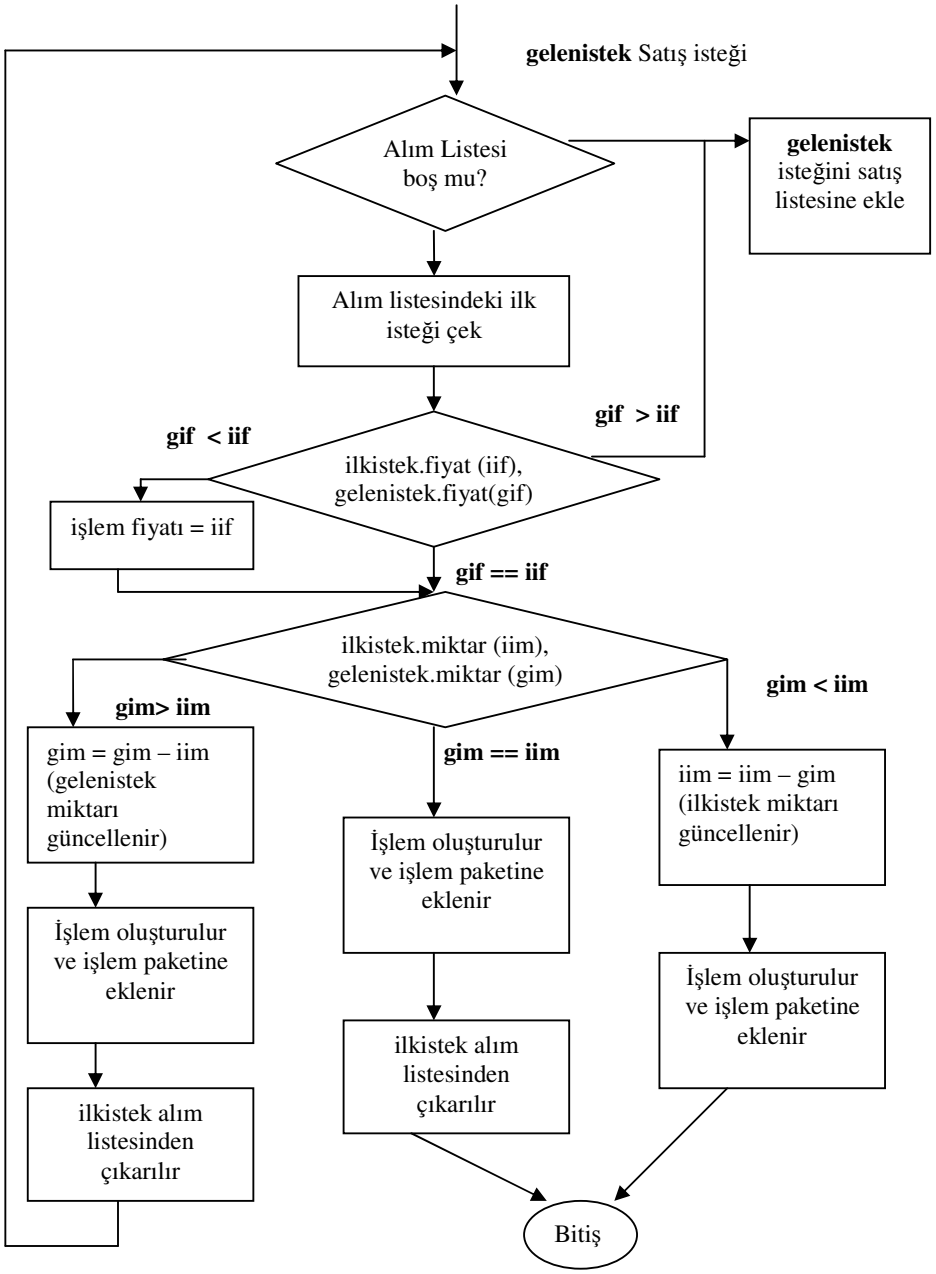
```
public int compareTo(Object kar)
{
  Istek istek1=(Istek)kar;
  Timestamp is1Zaman,zaman;
  double fiyat,is1Fiyat;
  is1Zaman=istek1.getTs();
  zaman=this.getTs();
  is1Fiyat=istek1.getFiyat();
  fiyat=this.getFiyat();
  byte tip=this.getTip();
  if(fiyat>is1Fiyat)
  {
    if(tip==Istek.ALIM)
     return -1;
    else if(tip==Istek.SATIS)
     return 1;
  }
  else if(fiyat<is1Fiyat)
  {
    if(tip==Istek.ALIM)
      return 1;
    else if(tip==Istek.SATIS)
      return -1;
  }
  else if(fiyat==is1Fiyat)
  {

   if(zaman.before(is1Zaman))
      return -1;
   else if(zaman.after(is1Zaman))
      return 1;
  }
  return 0;
}
```

Her bir eşleştirme işlem olarak tanımlanmaktadır, bir istek listede birden çok istekle eşleştirilebileceği için, işlemler işlem paketi (IslemPaketi yazılım sınıfı) isimli kapta tutulur.

Borsaya istek yapılırken alım istekleri için pazar fiyatının (satış listesindeki en düşük fiyat) üstünde bir değerde; satış istekleri için pazar fiyatının (alım listesindeki en yüksek) altında bir değerde istek yapılamaz. E-Borsa sisteminde bu kontrol işlem sırasında yapılmaktadır ve her işlem ile birlikte bir işlem fiyatı tutulmaktadır. İşlem fiyatı sayesinde istek ile ilgili düzeltme VT etmeni tarafından yapılmaktadır. Bu yaklaşım ile Borsa ve Broker etmenleri arasında pazar fiyatını öğrenmek için oluşacak fazladan mesajlaşma trafiği engellenmiştir.



Şekil 6.7: Alım İsteği Eşleştirme Yordamı



Şekil 6.8: Satış İsteği Eşleştirme Yordamı

6.2.4 Yönetici Etmeni

Yönetici etmen Broker ve Kullanıcı etmenlerin yönetimi ile ilgilenir. Sistemde herhangi bir zamanda sadece 1 tane Yönetici etmeni bulunabilir. Sistem başlatıldığında Yönetici etmeni sisteme dahil edilmiş olmalıdır.

Görevi:

Yönetici etmeninin görevleri şu şekilde sıralanır:

- Tacir etmenleri yaratır.
- Broker etmenleri yaratır.
- Kullanıcılardan gelen Broker atama isteklerini karşılar.
- Kullanıcıları mevcut Brokerlara adil bir şekilde dağıtır. Dağıtım sırasında basit bir algoritma kullanılır. Sisteme giriş yapan yeni bir kullanıcı, sistemde birden çok Broker etmeni varsa; yeni kullanıcıya en az sayıda kullanıcıdan sorumlu olan Broker etmeni atanır. Sistemde bulunan tüm Broker etmenlerine aynı sayıda kullanıcı atanmışsa; yeni kullanıcıya rasgele seçilen bir Broker etmeni atanır.
- Brokerlar'ın başka konaklara taşınmasını sağlar.

Aldığı Mesajlar:

Yönetici etmenin aldığı mesajlar Tablo 6.7'de listelenmiştir.

Tablo 6.7: Yönetici Etmenin Aldığı Mesajlar

Mesaj Adı	Mesaj İşlevi	Gönderici
YONETICI_SORUMLU_BROKER_VER	Tacir'den gelen Broker Atama isteğini bildirir.	Tacir Etmen
YONETICI_BROKER_YARAT	Yeni bir Broker etmen yaratılması isteğini bildirir.	Web arayüzü aracılığı ile sistem yöneticisi.
YONETICI_BROKER_KULLANICI_LISTESI_GETIR	Sistemde bulunan Brokerlar ve bu Brokerlara bağlı kullanıcıların listesini ister.	Web arayüzü aracılığı ile sistem yöneticisi.
YONETICI_UYGUN_BROKER_VER	Tacir etmene atanan Broker'ı öğrenmek için yapılan isteği bildirir.	Borsa Etmeni
YONETICI_KULLANICI_ETMENI_ATA	Kullancıya Tacir etmen atanması için yapılan isteği bildirir.	Web arayüzü aracılığı ile kullanıcı.
YONETICI_BROKER_LISTESI_VER	Yönetim arayüzüne, Broker taşıma betiğinde kullanılmak üzere Broker listesini bildirir.	Web arayüzü aracılığı ile sistem yöneticisi.
YONETICI_BROKER_TASI	Broker etmenin taşınması isteğini bildirir.	Web arayüzü aracılığı ile sistem yöneticisi.

Gönderdiği Mesajlar:

Yönetici etmenin gönderdiği mesajlar Tablo 6.8'de listelenmiştir.

Tablo 6.8: Yönetici Etmenin Gönderdiği Mesajlar

Mesaj Adı	Mesaj İşlevi	Alıcı	Mesaj Yanıtı
VT_KULLANICI_DENETLE	Kullanıcı için Tacir etmen yaratmadan önce, Kullanıcı bilgilerinin doğrulanması için kullanılır.	VT Etmeni	Kullanıcının eşsiz ID'si. Doğrulama yapılamazsa Yönetici etmene sıfırdan küçük bir değer döner.
TACIR_KLONLA	TacirSablon etmenine, kullanıcı için yeni bir Tacir etmen oluşturması için istek yapar.	TacirSablon Etmeni	Yok
BROKER_TASIN	Broker'a farklı bir konağa taşınması için istek yapar.	Broker Etmeni	Yok
BROKER_KULLANICI_EKLE	Broker'a yeni kullanıcısını (tacirini) bildirir.	Broker Etmeni	Yok
BROKER_KLONLA	BrokerSablon etmenine, sistemde yeni bir Broker etmen oluşturması için istek yapar.	BrokerSablon etmeni	Yok

6.2.5 VT (Veritabanı)Etmeni

E-Borsa sisteminde etmenlerin doğrudan veri kaynaklarına erişimi bulunmamaktadır. Etmenler veri kaynaklarına VT etmeni aracılığı ile ulaşırlar. Bu sayede VT etmeni dışında kalan etmenler veri kaynaklarının yeri ve tipi ile ilgilenmezler. Veri kaynağı üzerinde sadece kendileri için tanımlı işlemleri yapabilirler.

Görevi:

- Broker etmenlerinden gelen kullanıcı hesabı, portföyü ile ilgili istekleri gerçekleştirir.
- Broker etmenlerinden gelen istekler ve Borsa etmeninden gelen işlemler ile ilgili bilgileri veritabanına kaydeder.
- Yönetici etmenden gelen, kullanıcıların asıllama bilgilerinin doğruluğunu sınama isteklerini karşılar.
- Borsa etmeninden gelen istek iptali ile ilgili olarak kullanıcı portföy ve hesap bilgilerinde gerekli değişiklikleri yapar.

Aldığı Mesajlar:

VT etmeninin aldığı mesajlar 6.9'da listelenmiştir.

Tablo 6.9: VT Etmeninin Aldığı Mesajlar

Mesaj Adı	Mesaj İşlevi	Gönderici
VT_BLOKE_ET	Kullanıcı hesap ve/veya portföyünde, yapılan istek için gerekli olan güncellemeleri bildirir.	Broker Etmen
VT_HESAP_PORTFOY_GUNCELLE	Oluşan bir işlem sonucunda, kullanıcı hesap ve portföyü ile ilgili güncellemeleri bildirir.	Broker Etmen
VT_ISTEK_LOGLA	Başarı ile borsaya gönderilen bir isteğin iz (log) tablosuna kaydedilmesini için yapılan isteği bildirir.	Broker Etmeni
VT_ISLEM_LOGLA	Başarı ile gerçekleşen bir işlemi, iz tablosuna kaydedilmesi için yapılan isteği bildirir.	Borsa Etmeni
VT_KULLANICI_DENETLE	Kullanıcı kimlik asılaması isteğini bildirir.	Yönetici Etmen
VT_ISTEK_IPTAL_ET	İptal edilen bir isteği bildirir.	Borsa Etmeni

Gönderdiği Mesajlar:

VT Etmeninin gönderdiği herhangi bir mesaj bulunmamaktadır.

6.2.6 Etmen Politikaları

GES'in sağladığı dinamik politika yönetimi, E-Borsa etmenleri tarafından da kullanılmaktadır. Her etmene yalnızca ihtiyacı olan haklar -politikalar- atanmıştır. Etmenlere atanan politikalar aşağıdaki gibidir:

TacirSablonPolitikasi:

TacirSablon (Tacir etmenleri kendisini kopyalayarak yaratan etmen) etmeni için kullanılan politikadır. Kendisine bu politika atanmış etmen mesaj alıp gönderebilir, uzaktaki konaklara göç edebilir (Her Tacir etmene kullanıcısı atandıktan sonra Tacir etmen kullanıcısının bulunduğu GES sunucusuna göç eder.), kendini kopyalayabilir.

TacirPpolitikasi:

Kendilerine kullanıcıları atanmış Tacir etmenlere atanan politikadır. Bu politika TacirSablon → Tacir-<kullanıcı adı> dönüşümü sırasında, yönetici etmen tarafından Tacir-<kullanıcı adı> etmenine dinamik olarak atanır. Kendisine bu politika atanmış etmen mesaj alıp gönderebilir, sistem değişkenlerine ulaşabilir (Tacir etmenler

isteklerini oluştururken, istek zamanını belirlemek için sistem saatine ulaşmak zorundadır.)

BrokerPolitikasi:

Bu politika BrokerSablon (ve dolayısı ile Broker) etmenlerine atanır. Kendisine bu politika atanmış etmen mesaj alıp gönderebilir, uzaktaki konaklarda bulunan GES sunucularına göç edebilir, sistem değişkenlerine ulaşabilir.

BorsaPolitikasi:

Borsa etmenine atanan politikadır. Kendisine bu politika atanmış etmen mesaj alıp gönderebilir, sistem değişkenlerine (İşlemlere zaman etiketi yapıştırması için etmenin sistem saatine ulaşması gereklidir)ulaşabilir.

YoneticiPolitikasi:

Yönetici Etmenine atanan politikadır. Kendisine bu politika atanmış etmen mesaj alıp gönderebilir, sistem değişkenlerine ulaşabilir, ağ bağlantısı açabilir.

VTPolitikasi:

VT etmenine atanan politikadır. Kendisine bu politika atanmış etmen mesaj alıp gönderebilir, uzaktaki GES sunucularına taşınabilir, diskten okuma yapabilir (Veritabanı bağlantısı kurmak için gerekli bilgilerin olduğu ayar dosyasını okuyabilmek için), ağ bağlantısı yapabilir (Veritabanına ağ aracılığı ile bağlantı yapılıyorsa bu izin gereklidir.) ve sınıf dosyaları yükleyebilir(Veritabanı bağlantısı için etmenin, veritabanı sürücü sınıfını yüklemesi gerekmektedir).

6.3 E-Borsa Kullanıcı ve Yönetim Arayüzleri

Kullanıcı ve sistem yöneticileri, E-Borsa sisteminde bulunan etmenler ile GES sunucusu yönetimi için tasarlanan servlet'ler aracılığı ile (web tarayıcısı ile ulaştıkları) etkileşime geçerler.

Servlet yapısının iskeleti şu şekildedir:

```
public class <Servlet_adı> implements smasServlet {

  public void doGet(HttpRequest request, HttpResponse response)
  {

  }

  public void doPost(HttpRequest request, HttpResponse response)
  {
```

```
    }
}
```

"doGet" metodu GET ile yapılan HTTP isteklerini, "doPost" POST ile yapılan HTTP isteklerini karşılamaktadır. Sistem yönetimi ve kullanıcılar için yazılan servletler bu metodların içleri doldurularak gerçeklenmiştir.

6.3.1 Kullanıcı Arayüzü

E-Borsa sistemi, kullanıcılara borsada alım ve satış isteklerini gerçekleştirebilecekleri, yaptıkları işlemleri, istekleri ve isteklerinin durumlarını izleyebilecekleri, yaptıkları istekleri iptal edebilecekleri, borsada herhangi bir şirkete ait istek defterlerini görüntüleyebilecekleri (kullanıcılar, sistem yöneticilerinin aksine istek defterlerinde sadece istek miktarlarını ve fiyatlarını izleyebilirler), hesap ve portföy bilgilerini ve yaptıkları istekler sonucu hesap ve portföylerinden bloke edilen değerleri izleyebilecekleri bir arayüz sağlar.

Kullanıcıların sisteme giriş yaptıkları arayüz Şekil 6.9'da gösterilmiştir. Kullanıcı sisteme üye değilse, üye formunu (Şekil 6.10) kullanarak sisteme üye olabilir.

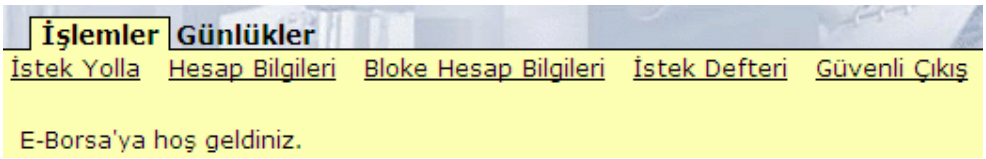
Şekil 6.9: E-Borsa Kullanıcı Giriş Ekranı

Şekil 6.10: E-Borsa Kullanıcı Kayıt Formu

Kullanıcı arayüzüne ait menüler (Şekil 6.11) aşağıda sıralanmıştır:

İşlemler Menüsü: "İstek Yolla" bağlantısı aracılığı kullanıcı isteklerini yollar. Kullanıcı, isteğini yollarken; listeden istek tipini (alım veya satış isteği), yine listeden istek yapmak istediği şirketi, alım veya satış yapacağı hisse miktarını ve fiyatını girer. "Hesap Bilgileri" ve "Bloke Hesap Bilgileri" bağlantıları ile kullanıcı sırası ile hesabında ve bloke hesabında bulunan nakit ile portföyünde ve bloke portföyünde bulunan hisseleri görüntüleyebilir. "İstek Defteri" bağlantısı ile kullanıcı, borsadaki diğer istekleri görüntüleyebilir. "Güvenli Çıkış" bağlantısı ile kullanıcı için açılan oturum kapatılarak sistemden çıkış yapılır.

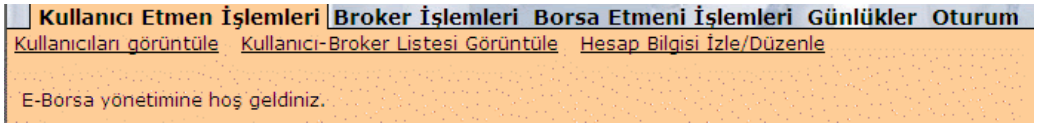
Günlükler: "İstek Günlüğü" bağlantısı ile kullanıcının yaptığı istekler, "İşlem Günlüğü" bağlantısı ile kullanıcının yaptığı istekler için oluşan işlemler izlenebilir.



Şekil 6.11: Kullanıcı Arayüzü Menüleri

6.3.2 Yönetim Arayüzü

Yönetim arayüzü, sadece sistem yöneticisi olan kullanıcılar tarafından kullanılabilir.



Şekil 6.12: Yönetim Arayüzü Menüleri

Yönetim arayüzüne ait menüler (Şekil 6.12) aşağıda sıralanmıştır:

Kullanıcı Etmen İşlemleri:

- "Kullanıcıları görüntüle" bağlantısı ile sistemde bulunan kullanıcılar ve bu kullanıcılara ait kullanıcı bilgileri görüntülenir.
- "Kullanıcı-Broker Listesi Görüntüle" bağlantısı ile sistemde bulunan Brokerlar ve bu Brokerlara bağlı Tacirler görüntülenir.
- "Hesap Bilgisi İzle/Düzenle" bağlantısı ile kullanıcıların hesap bilgileri izlenebilir ve hesap ve portföy bilgileri güncellenebilir.

Broker İşlemleri:

- "Broker Yarat" bağlantısı ile sistemde yeni bir Broker etmeni yaratılır.
- "Broker Taşı" bağlantısı ile sistem üzerinde bulunan bir Broker başka bir GES sunucu üzerine taşınabilir.

Borsa Etmeni İşlemleri:

- "İstek Defteri Göster" bağlantısı ile seçilen şirkete ait alım ve satış listelerinin bulunduğu istek defteri görüntülenir.

Günlükler:

- "İstek Günlüğü" bağlantısı ile sistemdeki kullanıcıların yaptıkları istekler izlenir.
- "İşlem Günlüğü" bağlantısı ile sistemde oluşan işlemler izlenebilir.

Oturum:

- "Güvenli Çıkış" bağlantısı ile oturum kapatılarak sistemden çıkış yapılır.

7. E-BORSA PERFORMANS ANALİZİ

7.1 Testlerin Planlanması

Testleri yapmak için Performans adında bir etmen yaratılmıştır. Performans etmeni 2 çeşit mesajı kabul etmektedir:

a-) Tacir Etmen Oluşturma Mesajı: Performans etmeni bu mesajı aldığında, kendisine mesaj parametresi aracılığı yollanan sayıda Tacir (Kullanıcı) etmenini yaratır.

b-) Tacir Etmenlere İstek Yollama Mesajı: Bu mesajın etmen sayısı ve istek sayısı olmak üzere 2 adet parametresi vardır. Performans etmeni bu mesajı aldığında, etmen sayısı kadar iplik yaratır ve bu iplikleri sıra ile başlatır.

```
public class EtmenIpligi extends Thread
{
   int kulid;
   int istekSayisi;
   public EtmenIpligi(int kulid,int istekSayisi)
   {
      this.kulid = kulid;
      this.istekSayisi = istekSayisi ;
   }
   public void run()
   {
     String adi = (String)kullaniciBilgileri.get(new
Integer(kulid));
     Enumeration tacirEtmenler =
getAgentInterface().getAgentPointer(Yardimci.Sabitler.TacirEtmenA
diOnEk+adi);
     if(tacirEtmenler.hasMoreElements())
     {
        AgentPointer etmen =
(AgentPointer)tacirEtmenler.nextElement();
        rastgeleIstekYolla(etmen,kulid,istekSayisi);
     }
   }
}
```

"EtmenIpligi" sınıfı tanımından da görülebileceği gibi, ipliğin "run" metodu içerisinde, "rastgeleIstekYolla" işlevi aracılığı ile istek sayısı kadar istek yaratılır. Yaratılan istekler, ilgili Tacir etmenine gönderilir.

İstekler 1'er saniye ara ile yaratılır.

Her test, borsada herhangi bir istek yokken yapılmıştır.

7.2 Testler

7.2.1 Tüm GES Sunucularının Aynı Konakta Olduğu Testler

Tablo 7.1: Testler İçin Kullanılan Kısaltmalar

İİBoUZ	İlk İsteğin Borsaya Ulaşma zamanı
SİBoUZ	Son İsteğin Borsaya Ulaşma zamanı
BoOİİşZ	Borsada Oluşan İlk İşlemin Zamanı
BoOSİşZ	Borsada Oluşan Son İşlemin Zamanı
TES	Tacir Etmen Sayısı
BrES	Broker Etmen Sayısı
EZİS	Eş Zamanlı İstek Sayısı
GİS	Gönderilen toplam İstek Sayısı
OişS	Oluşan toplam İşlem Sayısı

Test 1: Tüm şartlar eşit, değişik denemeler sonucu oluşan değerler.

Amaç: Tüm şartlar eşitken, elde edilen değerlerde belirgin sapmalar olup olmadığının tespit edilmesi.

Test sonuçları Tablo 7.2'de gösterilmiştir.

Tablo 7.2: Test 1 Sonuçları

Deneme	**İİBoUZ (sn)**	**SİBoUZ (sn)**	**BoOİİşZ (sn)**	**BoOSİşZ (sn)**	**EZİS**	**TES**	**BrES**	**GİS**	**OİşS**
1	2	41	3	37	10	10	2	100	53
2	4	44	6	40	10	10	2	100	59
3	2	50	3	42	10	10	2	100	57
4	2	44	2	35	10	10	2	100	63

Sonuç: Dört farklı deneme sonucunda borsaya iletilen son istek zamanlarının (SİBoUZ) ortalaması 44.75 sn olarak bulunmuştur. Dağılımın standart sapması 3.27 olarak bulunmuştur. Dört denemenin üçünde ortalamaya yakın değerler elde edilmiştir. Değişik denemelerde borsaya ulaşan son istek ve borsada oluşan son işlem zamanları için marjinal değişiklikler oluşmamıştır. Uygulama eşit şartlar için kararlı bir yapıya sahiptir.

Test 2: Tüm değişkenler sabit, istek sayıları değişken.

Amaç: Eşit şartlar altında istek sayıları değişiminin performansa etkisinin gözlemlenmesi.

Test sonuçları Tablo 7.3'te listelenmiştir.

Tablo 7.3: Test 2 Sonuçları

Deneme	İİBoUZ (sn)	SİBoUZ (sn)	BoOİİşZ (sn)	BoOSİşZ (sn)	EZİS	TES	BrES	GİS	OİşS
1	2	41	3	37	10	10	2	100	53
2	3	66	4	52	15	10	2	150	95
3	3	91	3	81	20	10	2	200	123
4	3	117	3	110	25	10	2	250	173
5	1	139	3	125	30	10	2	300	191
6	2	186	4	176	40	10	2	400	269

Sonuç: Sırası ile 100, 150, 200, 250, 300 ve 400 istek için yapılan denemeler sonucunda, SİBoUZ-İstek sayısı ilişkisi ile ilgili grafik Şekil 7.1'de gösterilmiştir. Grafikten de anlaşılacağı gibi, istek sayısının eşit miktarda (50 adet) arttırıldığı 100-300 aralığı için grafik, eğimi sabit bir doğru şeklinde elde edilmiştir. Bu grafik artan istek sayısı ile uygulamanın performansının düşmediğini -performansın sabit kaldığını- göstermektedir.

Şekil 7.1: Test 2 İçin SİBoUZ-İstek Sayısı İlişkisi

Test 3: Tüm şartlar eşit broker sayısı değişken.

Amaç: Şartlar eşit iken Broker etmen sayısının arttırılmasının performansa etkisinin ölçülmesi.

Sonuç: Dört farklı deneme sonucunda elde edilen borsaya iletilen son istek zamanlarının (SİBoUZ) ortalaması 87,5 sn, standart sapması 2,5 olarak bulunmuştur.

Test sonuçları Tablo 7.4'te listelenmiştir. Denemelerin tümünde ortalama değerden anlamlı sapmalar görülmemiş, ortalamaya yakın değerler elde edilmiştir. Elde edilen

verilerden yola çıkılarak, bu yapılandırma için; tüm şartlar eşitken broker sayısının artmasının performansa anlamlı bir etkisi olmadığı söylenebilir.

Tablo 7.4: Test 3 Sonuçları

Deneme	İİBoUZ (sn)	SİBoUZ (sn)	BoOİİşZ (sn)	BoOSİşZ (sn)	EZİS	TES	BrES	GİS	OİşS
1	3	91	3	81	20	10	2	200	123
2	2	87	4	70	20	10	3	200	138
3	2	84	4	74	20	10	4	200	114
4	3	88	4	78	20	10	5	200	111

Test 4: Tüm şartlar eşit kullanıcı sayısı değişken.

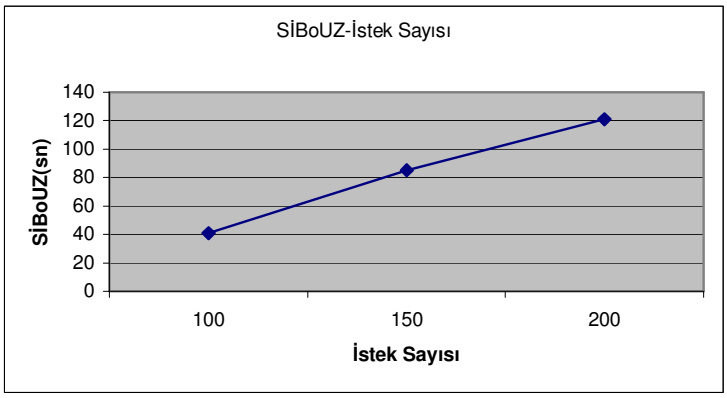
Amaç: Şartlar eşit iken kullanıcı sayısının arttırılmasının performansa etkisinin ölçülmesi.

Test 10, 15 ve 20 kullanıcının eş zamanlı 10 istekte bulunduğu denemelerden oluşmaktadır. Test sonuçları Tablo 7.5'te listelenmiştir.

Tablo 7.5: Test 4 sonuçları

Deneme	İİBoUZ (sn)	SİBoUZ (sn)	BoOİİşZ (sn)	BoOSİşZ (sn)	EZİS	TES	BrES	GİS	OİşS
1	2	41	3	37	10	10	2	100	53
2	4	85	7	79	10	15	2	150	83
3	11	121	11	100	10	20	2	200	126

Sonuç: Şekil 7.2'de de görüldüğü gibi, Tacir etmen sayısının artması, isteklerin Brokerlar'a ve borsaya gönderilmesi sırasında geçen süreyi doğrusal olarak arttırmıştır. Bunun yanında Tacir etmen kodu olabildiğince küçük boyutta tutulduğu için test sırasında sistem kaynaklarının kullanımı konusunda anlamlı performans farklılıkları gözlenmemiştir.



Şekil 7.2: Test 4 İçin SİBoUZ-İstek Sayısı İlişkisi

7.2.2 GES Sunucularının Farklı Konaklarda Olduğu Testler

Amaç: Bu testler aracılığı ile, E-Borsa sistemini oluşturan alt sistemlerin ağ üzerinde farklı makinelerde bulunduğu yapılandırmalar için sistemin toplam performans ve ağ yükünün belirlenmesi amaçlanmaktadır.

Konakların Sistem Özelikleri:

Konak 1: IP numarası: 10.0.0.4; İşlemci: Intel Centrino 1.6 GHz, Sistem Belleği: 752 MB

Konak 2: IP numarası: 10.0.0.21; İşlemci Intel P3 800 MHz, Sistem Belleği: 642 MB

Makineler arasındaki ağ hızı 10 Mbit'tir ve pasif HUB (ethernet topolojisinde bilgisayarları birbirlerine bağlamak için kullanılan ağ cihazı) üzerinden birbirlerine bağlıdırlar.

Birinci Yapılandırma (Test 5):

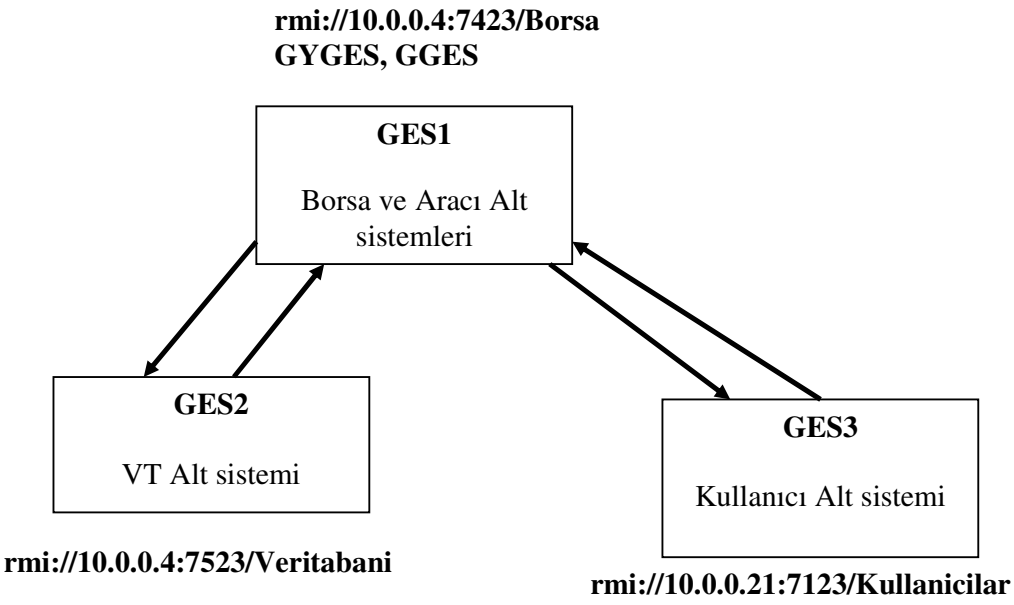
GES1: rmi://10.0.0.4:7423/Borsa

GES2: rmi://10.0.0.4:7523/Veritabani

GES3: rmi://10.0.0.21:7123/Kullanicilar

GYGES: GES1

GGES: GES1



Şekil 7.3: Birinci Yapılandırma İçin Sistem Yapısı

Şekil 7.3'te açıklanan sistem için yapılan testin sonucu Tablo 7.6'da listelenmiştir.

Tablo 7.6: Test 5 Sonuçları

Deneme	İİBoUZ (sn)	SİBoUZ (sn)	BoOİİşZ (sn)	BoOSİşZ (sn)	EZİS	TES	BrES	GİS	OİşS
1	3	100	3	94	10	10	2	100	51
2	2	142	4	132	15	10	2	150	84
3	2	202	2	200	20	10	2	200	114

İkinci Yapılandırma (Test 6) :

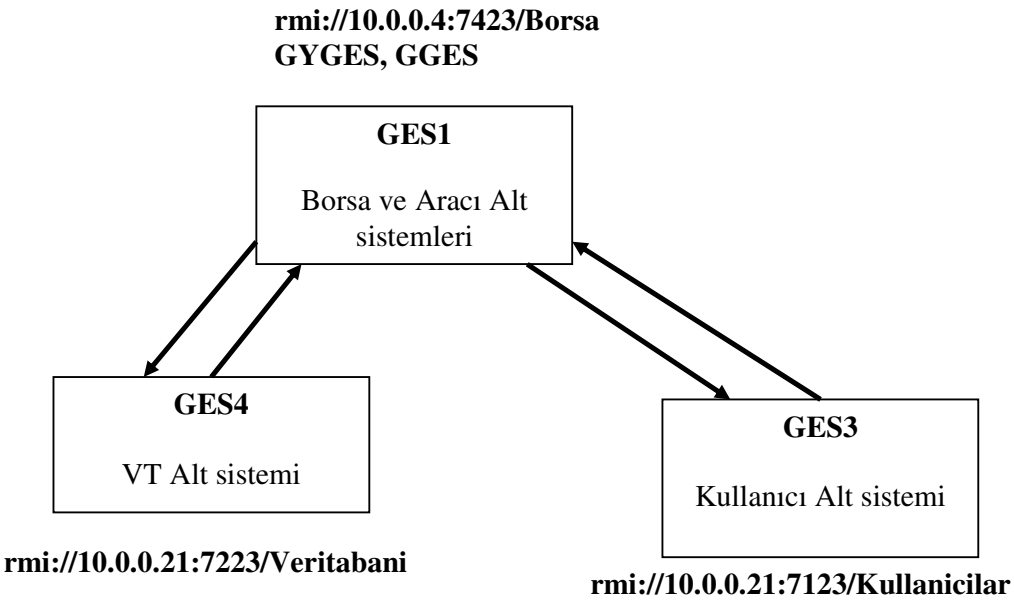
GES1: rmi://10.0.0.4:7423/Borsa

GES4: rmi://10.0.0.21:7223/Veritabani

GES3: rmi://10.0.0.21:7123/Kullanicilar

GYGES: GES1

GGES: GES1

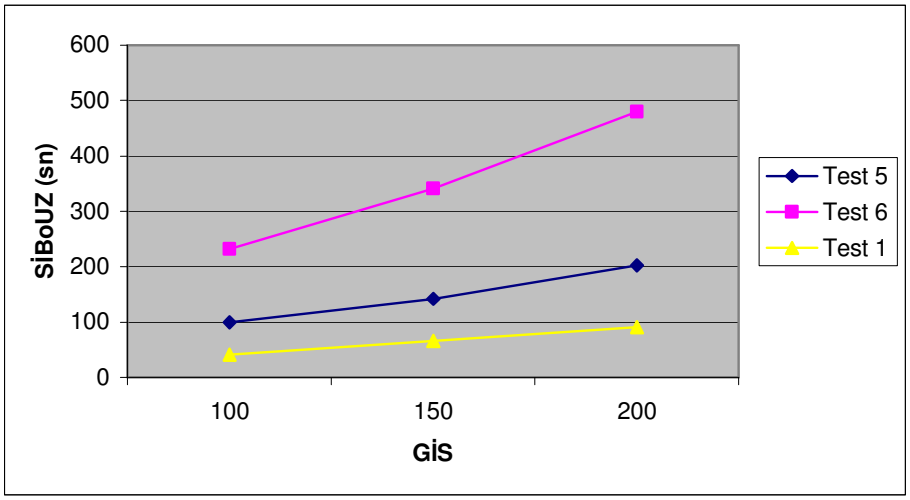


Şekil 7.4: İkinci Yapılandırma İçin Sistem Yapısı

Şekil 7.4'te açıklanan sistem için yapılan testin sonucu Tablo 7.7'de listelenmiştir.

Tablo 7.7: Test 6 Sonuçları

Deneme	İİBoUZ (sn)	SİBoUZ (sn)	BoOİİşZ (sn)	BoOSİşZ (sn)	EZİS	TES	BrES	GİS	OİşS
1	10	232	11	219	10	10	2	100	60
2	8	341	14	325	15	10	2	150	91
3	18	480	25	445	20	10	2	200	119



Şekil 7.5: Test 5 - Test 6 - Test 1 Karşılaştırması

Sonuçlar:

Şekil 7.5'ten de görüldüğü gibi, VT alt sisteminin Borsa ve Aracı alt sistemle farklı konaklarda bulunduğu Test 6'da elde edilen süreler ile VT alt sistemi ile Borsa ve Aracı alt sistemlerin aynı konakta bulunduğu Test 5'te elde edilen süreler arasında dramatik farklar vardır. VT etmeni sistemde darboğaza neden olmaktadır. Bunun nedeni Borsa ve Broker etmenleri ile VT etmeni arasındaki mesajlaşma trafiğinin çok yoğun olmasıdır. Alt sistemler GES sunucular arasında dağıtılırken bu konu göz önüne alınmalıdır.

Yine Şekil 7.5'ten görülebileceği gibi Test 1 ile Test 5'in sonuçları birbiri ile paralellik göstermektedir. Kullanıcıların farklı bir konaktan istek yaptıkları Test 5'te alınan sonuçlar kabul edilebilir sınırlar içerisindedir. Ayrıca Test 1 ve Test 5 sonucu elde edilen eğrilerin eğimleri birbiri ile uyumludur.

Test 5 ve Test 6 için elde edilen ağ trafiği ve sistem yükü ile ilgili grafikler, EkA'da verilmiştir. Şekil A.1 ve Şekil A.3 karşılaştırıldığı zaman, Test 5 için; Konak 1'deki sistem kaynaklarının daha yoğun olarak kullanıldığı görülmektedir. Bunun nedeni, Borsa ve Broker etmenleri tarafından yoğun olarak kullanılan VT alt sisteminin Konak 1'de bulunmasıdır. Benzer şekilde Şekil A.2 ve Şekil A.4'e bakıldığı zaman Konak 1'in ağ yükünün Konak 2'ye oranla daha fazla (indirme ve yükleme için yaklaşık 2 kat) olduğu görülmektedir. Şekil A.2, Şekil A.5 ve Şekil A.7 ile Şekil A.4, Şekil A.6 ve Şekil A.8'e dayanarak Test 5'teki farklı denemeler için sırası ile Konak 1 ve Konak 2'deki ağ trafiğinin zaman içerisinde sabit kaldığı söylenebilir. Test 6 için, Şekil A.9 ve Şekil A.11 karşılaştırıldığı zaman, -Test 5'te görüldüğü gibi- VT alt sisteminin bulunduğu Konak 2'deki sistem kaynaklarının yoğun olarak kullanıldığı anlaşılmaktadır. Şekil A.10 ve Şekil A.12 karşılaştırıldığı zaman; Konak 2'nin ağ yükünün Konak 1'e oranla daha yüksek olduğu görülmektedir. Bunun nedeni hem VT alt sisteminin, hem de Borsa alt sisteminin Konak 2'de bulunmasıdır. Şekil A.10, A.13 ve A.15 ile Şekil A.12, Şekil A.14 ve Şekil A.16'ya dayanarak Test 6'daki farklı denemeler için sırası ile Konak 1 ve Konak 2'deki ağ trafiğinin zaman içerisinde sabit kaldığı söylenebilir.

8. SONUÇLAR VE TARTIŞMA

E-Borsa sistemi, dağıtık çalışma kullanılarak GES'in sağladığı güvenlik, izleme, hareketlilik gibi özellikler kullanılarak gerçeklenmiştir. Sistem ölçeklenebilir, izlenebilir ve güvenli bir sistem olmuştur.

Sistem gerçeklenirken GES mimarisinin sunduğu özellikler azami şekilde kullanılmıştır. Bu özellikler sırasıyla etmen göçü, etmen kopyalanması ve etmen politikalarının dinamik olarak değiştirilmesidir. Güvenlik önlemleri uygulama düzeyinde alınmıştır (Tacir etmenler yaratılırken; Yönetici etmen, VT etmenine mesaj yollayarak isteğin kullanıcıdan gelip gelmediğini kullanıcı veritabanını sınayarak kontrol eder.), ağ güvenliği için tüm mesajlaşmaları, etmen kodunu ve etmen durumunu şifreleyerek saklayan GES sistemine güvenilmiştir.

Borsa gerçeklemesi sırasında sistem modüllere ayrıldığı için borsayı gerçeklemek için karmaşık bir algoritma üretilmesine gerek kalmamıştır. Buna ek olarak, modüler yapı sayesinde veritabanı erişimi olabildiğince azaltılmıştır (Örneğin, Borsa etmeninde bulunan istek defteri Borsa etmeninin bulunduğu makinenin sistem belleğinde bulunmaktadır, bu durum isteklerin çok daha hızlı şekilde işlenmesini sağlamaktadır.).

Çok sayıda eş zamanlı istek için yapılan performans testlerinde, elde edilen değerler kabul edilebilir sınırlar içerisindedir. Yine testlerden anlaşıldığı kadarı ile E-Borsa sistemini oluşturan alt sistemlerin yapılandırması -GES sunucularına dağılımı- performans için anahtar faktör konumundadır.

E-Borsa sistemi geliştirilmeye açık bir sistemdir. Örneğin, Tacir etmenler, kullanıcıları adına ticaret yapacak şekilde geliştirilebilirler. Böylece Tacir etmenler, izleyecekleri çeşitli ticaret metodları ile her an borsada olamayacak kullanıcılarından daha etkin işlemler yapabilecek duruma geleceklerdir.

E-Borsa sistemi, sanal borsa sistemlerine hareketli etmen tabanlı yeni bir yaklaşım getiren kullanılmaya değer bir e-ticaret sistemidir.

KAYNAKLAR

[1] **Subramanian, S. and Singhal, M**., 2000. A secure, real-time stock market protocol, *Netnomics*, **2**, 221-245.

[2] **Sierra, C., Wooldridge, M. and Sadeh, N.,** 2000. Agent research and development in Europe, *Internet Computing, IEEE*, **4**, 81-83.

[3] **Jennings, N. R. and Wooldridge, M.,** 1999. Agent-Oriented Software Engineering, *Proceedings of the 9th European Workshop on Modelling Autonomous Agents in a Multi-Agent World : Multi-Agent System Engineering (MAAMAW-99)*, Valencia, Spain, 30 June - 2 July 1999, 1-7.

[4] **Wooldridge, M.,** 2002. An Introduction to Multiagent Systems, John Wiley & Sons Ltd, London.

[5] **Wang, Y.**, 2002. Dispatching Multiple Mobile Agents in Parallel for Visiting E-Shops, *Proceedings of the Third International Conference on Mobile Data Management (MDM 2002)*, Singapore, January 8-11 2002, 61.

[6] **Corradi, A., Montanari, R. and Stefanelli, C.**, 2001. Security of mobile agents on the Internet, *Internet Research: Electronic Networking Application and Policy*, **11**, 84-95.

[7] **Uğurlu S.**, 2007. Güvenli Bir Hareketli Etmen Sisteminin Tasarımı ve Gerçeklenmesi, *Doktora Tezi,* İ.T.Ü. Fen Bilimleri Enstitüsü, İstanbul.

[8] **Pham, V. A., Karmouch, A.**, 1998. Mobile Software Agents: An Overview, *Communications Magazine, IEEE*, **36**, 26-37.

[9] **Mitsubishi Electric ITA**, 1997. Concordia: An Infrastructure for Collaborating Mobile Agents, *http://www.cis.upenn.edu/~bcpierce/courses/629/papers/Concordia-MobileAgentConf.html*

[10] **Jeon, H.,** 2000. JATLite Introductory FAQ, *http://www-cdr.stanford.edu/ProcessLink/papers/JATL.html*

[11] **Bryce, C., Vitek, J.**, 1999. The JavaSeal Mobile Agent Kernel, *First International Symposium on Agent Systems and Applications and Third International Symposium on Mobile Agents (ASA/MA'99)*, Palm Springs, CA, USA, 3-6 October 1999, 103-116.

[12] **Wang,C., Leung H.,** 2005. Mobile agents for secure electronic commerce transactions with privacy protection of the customers, *Proceedings of the 2005 IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE'05) on e-Technology, e-Commerce and e-Service,* Hong Kong, China, March 29 - April 01, 2005, 530-535.

[13] **Papazoglou, M. P.,** 2001. Agent-oriented technology in support of e-business, *Communications of the ACM*, **44**, 71-77.

[14] **Dogaç et al.,** 1998. METU-Emar: An Agent-Based Electronic Marketplace on the Web, *ECDL'98 LNCS*, **1513**, 777-790.

[15] **Boer et al.,** 2004. An Agent-Based Framework for Artificial Stock Markets, *Proceedings of 16th Belgian-Dutch Conference on Artificial Intelligence (BNAIC'04)*, Groningen, The Netherlands, 21-22 October 2004, 83-90.

[16] **LeBaron, B.,** 2002. Building the Santa Fe Artificial Stock Market, *http://people.brandeis.edu/~blebaron/wps/sfisum.pdf*

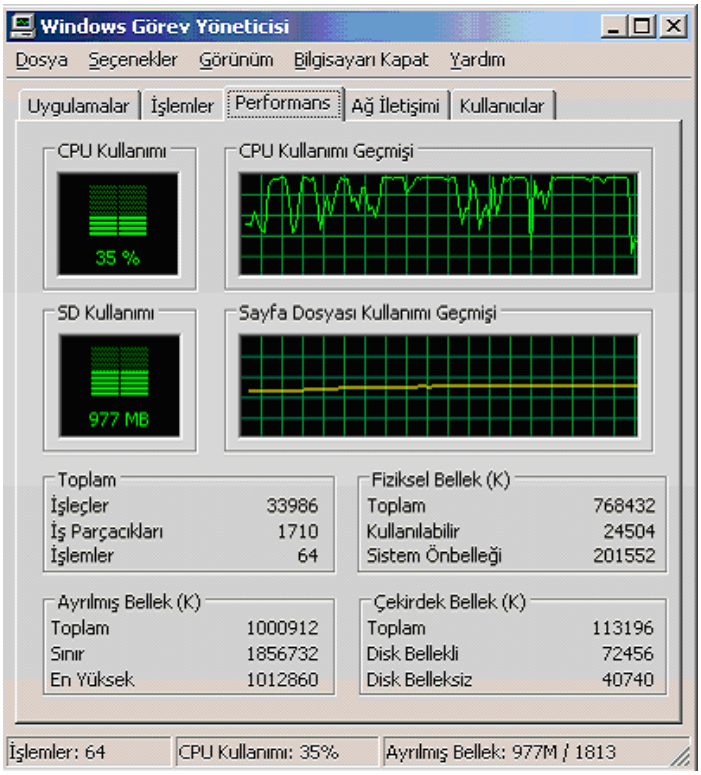
[17] **Phelps et al.**, 2002. JASA (Java Auction Simulator API), *http://www.csc.liv.ac.uk/~sphelps/jasa*

[18] **Wong, D., Paciorek, N. and Moore, D.,** 1999. Java-based Mobile Agents, *Communications of ACM*, **42**, 92-ff.

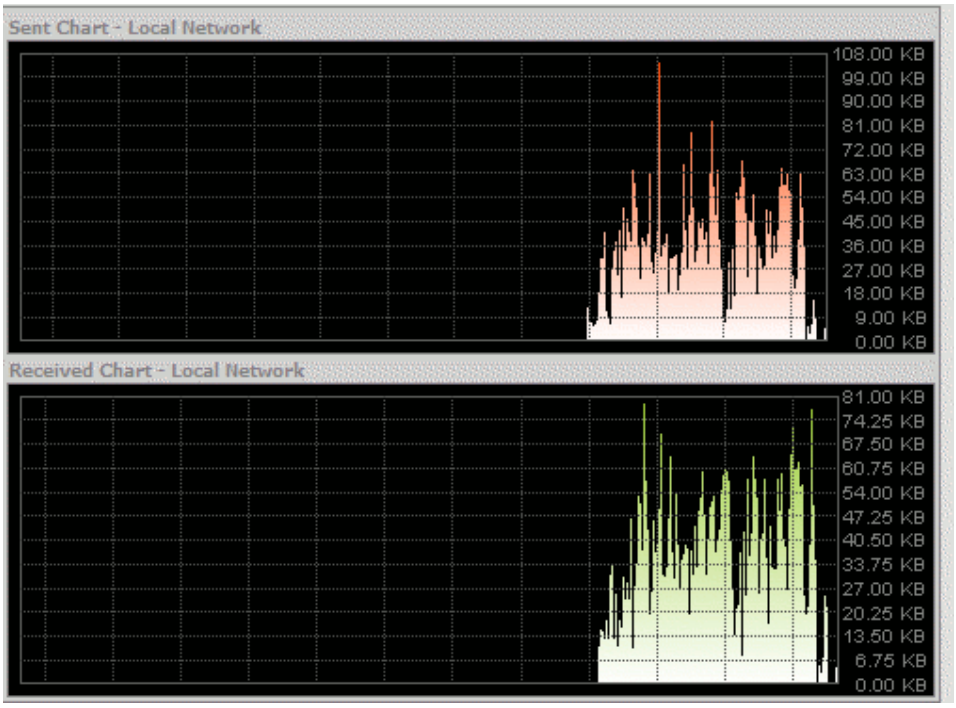
[19] **Bigus, J. and Bigus, J.,** 2001. Constructing Intelligent Agents Using Java, Second Edition, John Wiley & Sons Inc., New York.

[20] **Uğurlu, S.ve Erdoğan, N.**, 2004. Hareketli Bir Güvenli Etmen Sistemi, *Havacılıkta İleri Teknolojiler ve Uygulamaları Sempozyumu HİTEK'04*, İstanbul, Türkiye, Aralık 633-637.

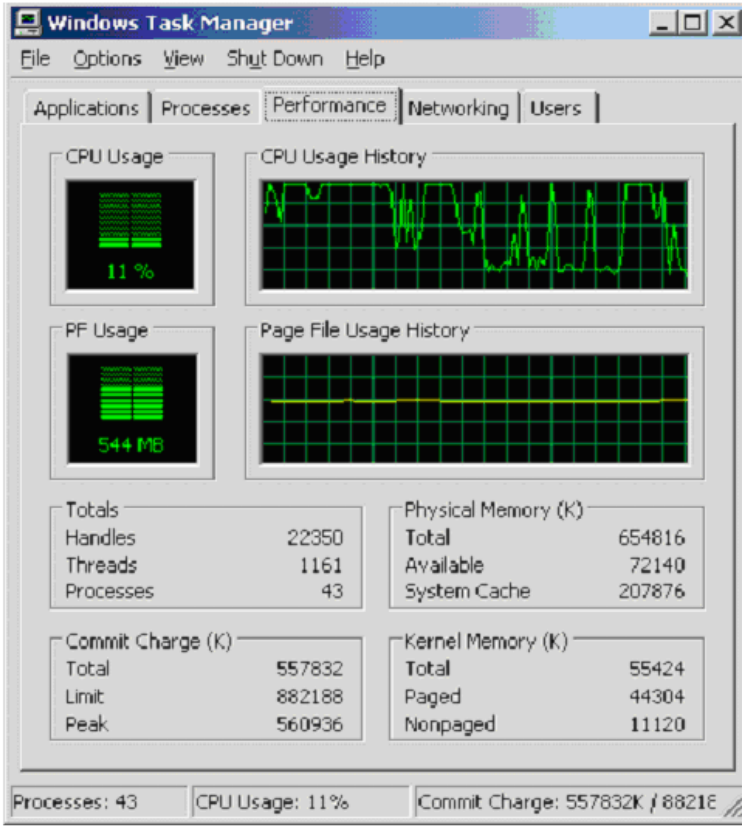
EK A: PERFORMANS İLE İLGİLİ GRAFİKLER



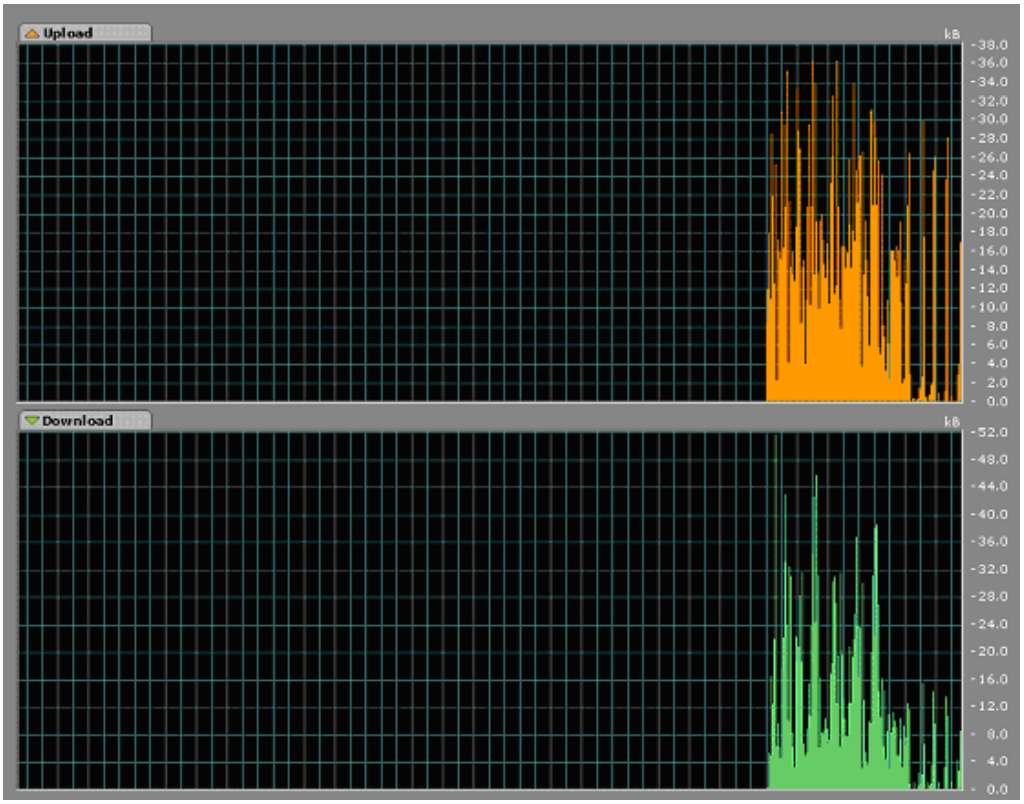
Şekil A.1: Test 5-1 Sırasında Konak 1'deki Sistem Kaynağı Kullanımı



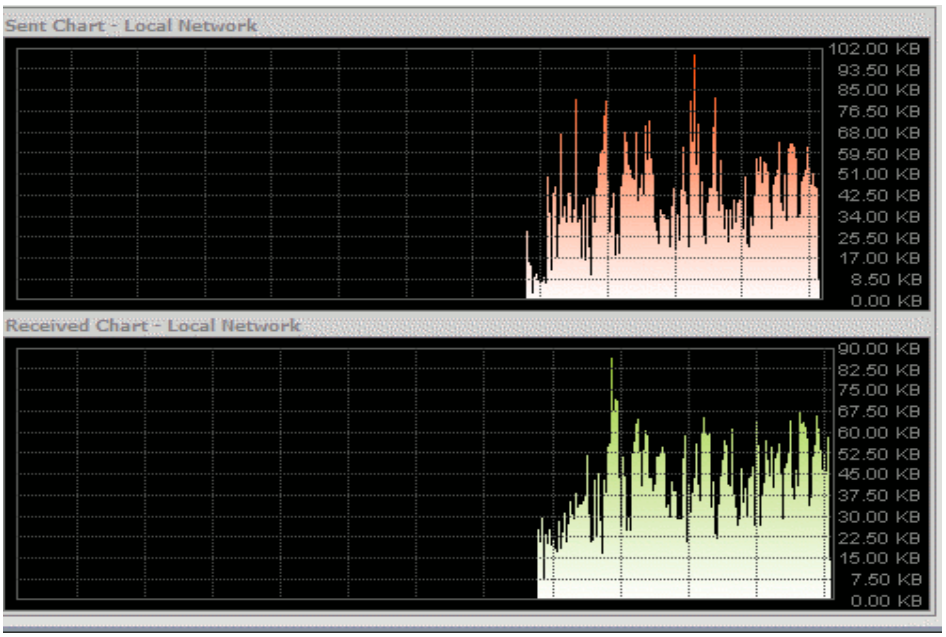
Şekil A.2: Test 5-1 Sırasında Konak 1'deki Ağ Trafiği



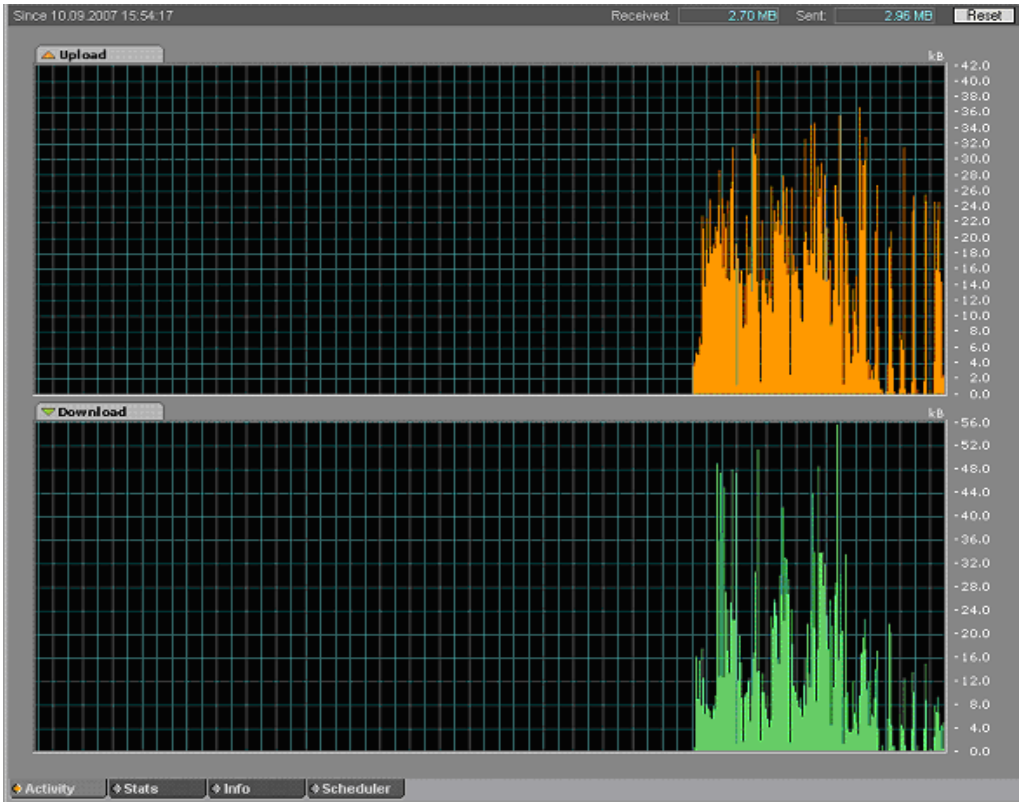
Şekil A.3: Test 5-1 Sırasında Konak 2'deki Sistem Kaynağı Kullanımı



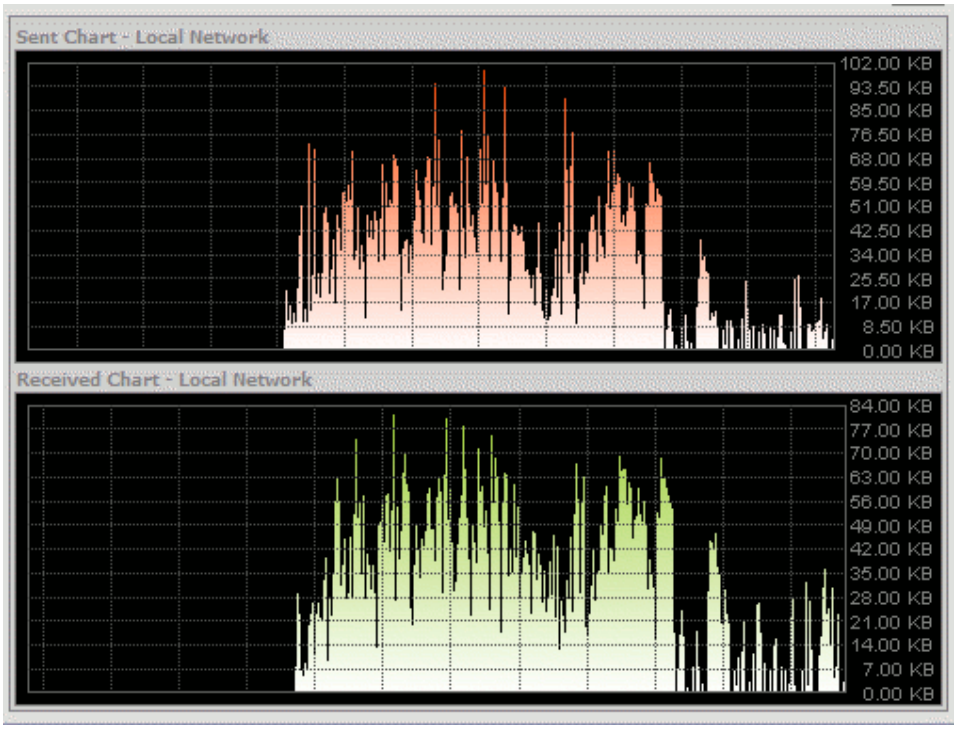
Şekil A.4: Test 5-1 Sırasında Konak 2'deki Ağ Trafiği



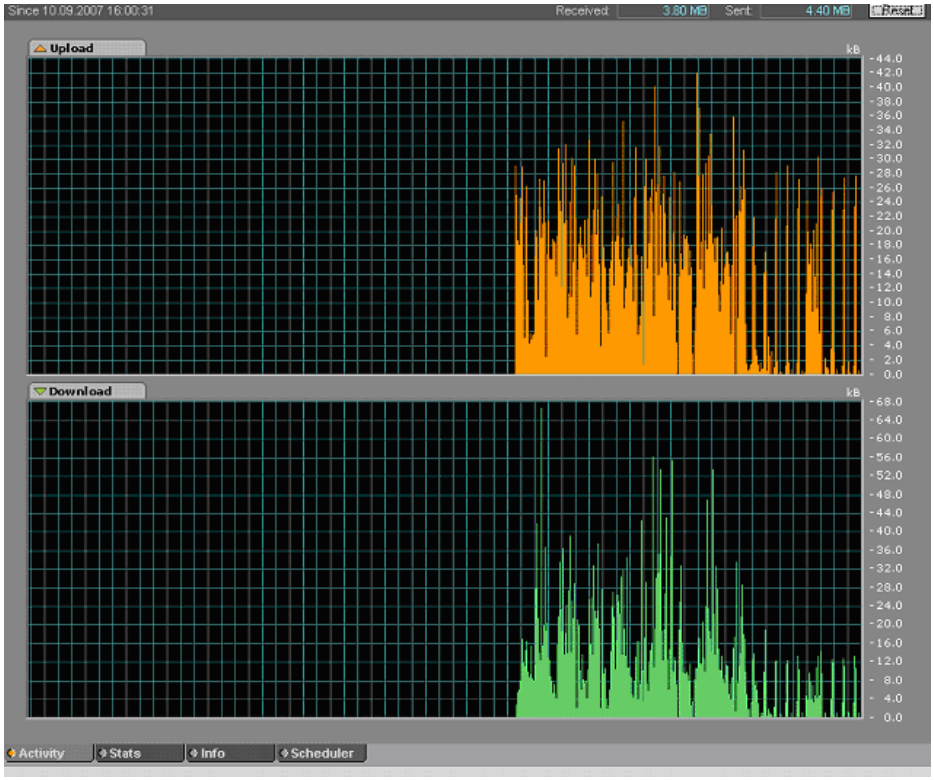
Şekil A.5: Test 5-2 Sırasında Konak 1'deki Ağ Trafiği



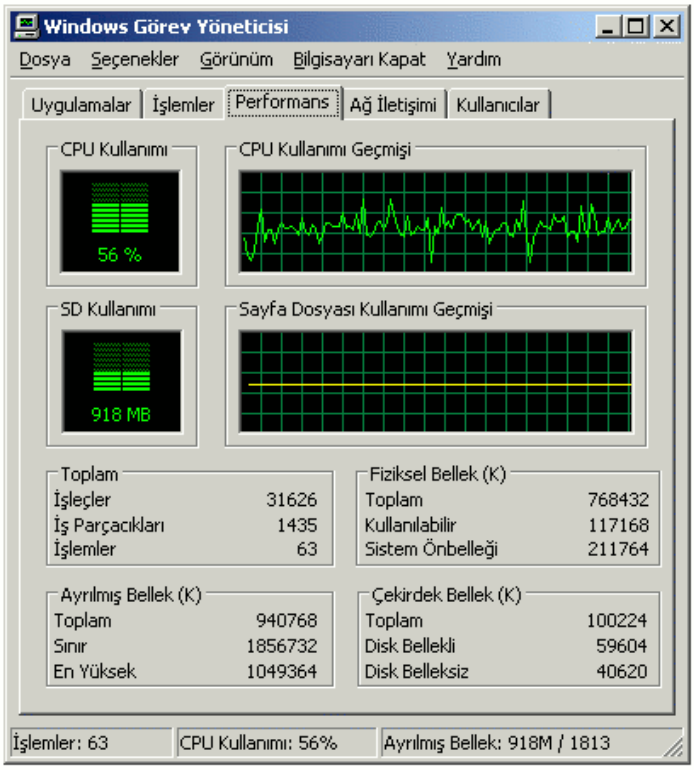
Şekil A.6: Test 5-2 Sırasında Konak 2'deki Ağ Trafiği



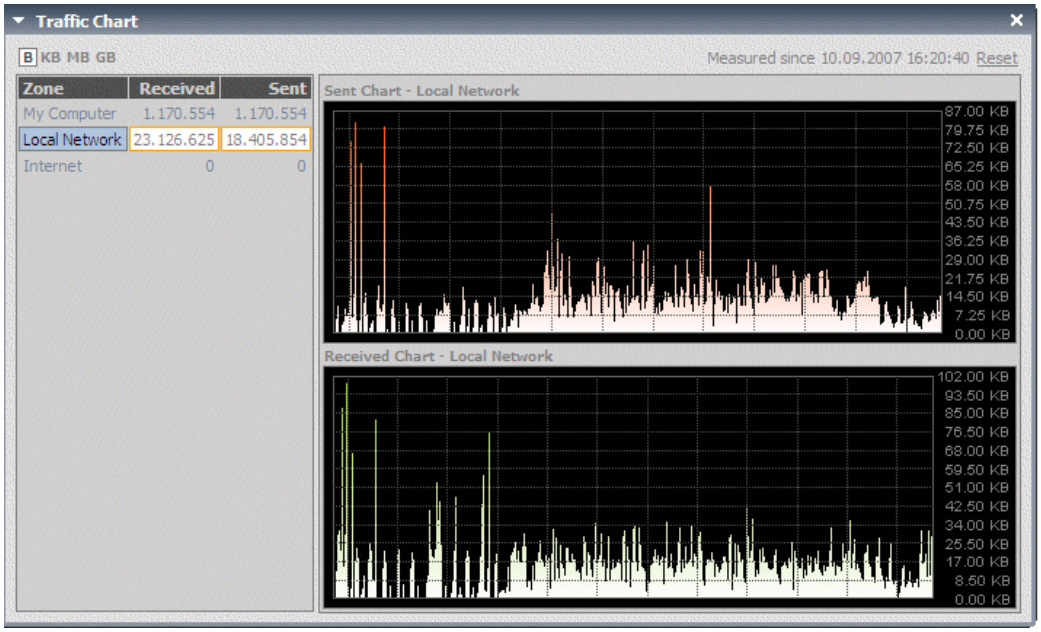
Şekil A.7: Test 5-3 Sırasında Konak 1'deki Ağ Trafiği



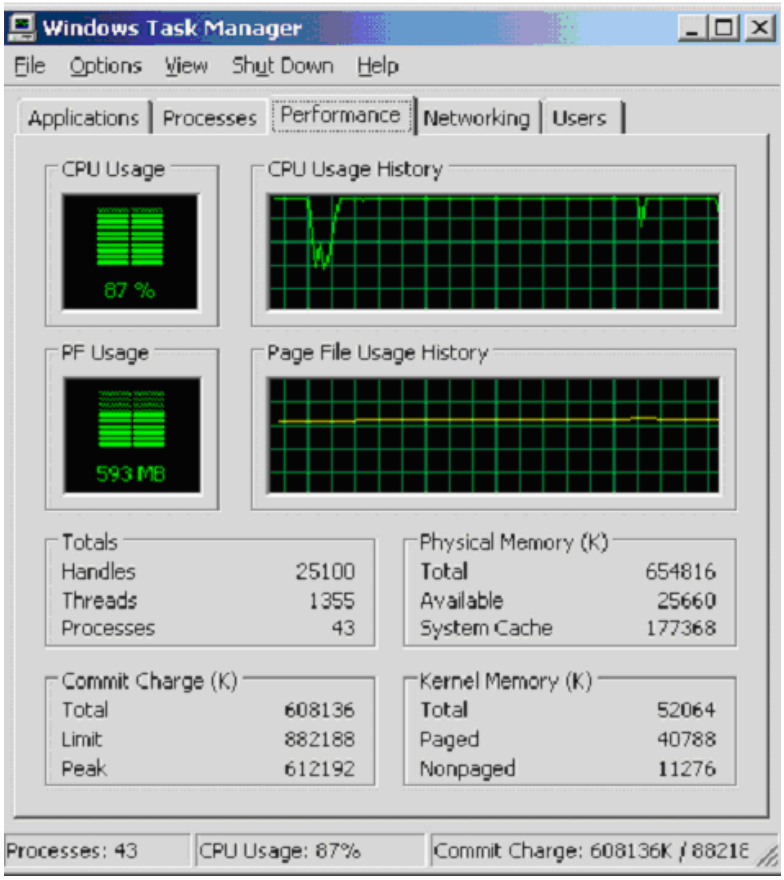
Şekil A.8: Test 5-3 Sırasında Konak 2'deki Ağ Trafiği



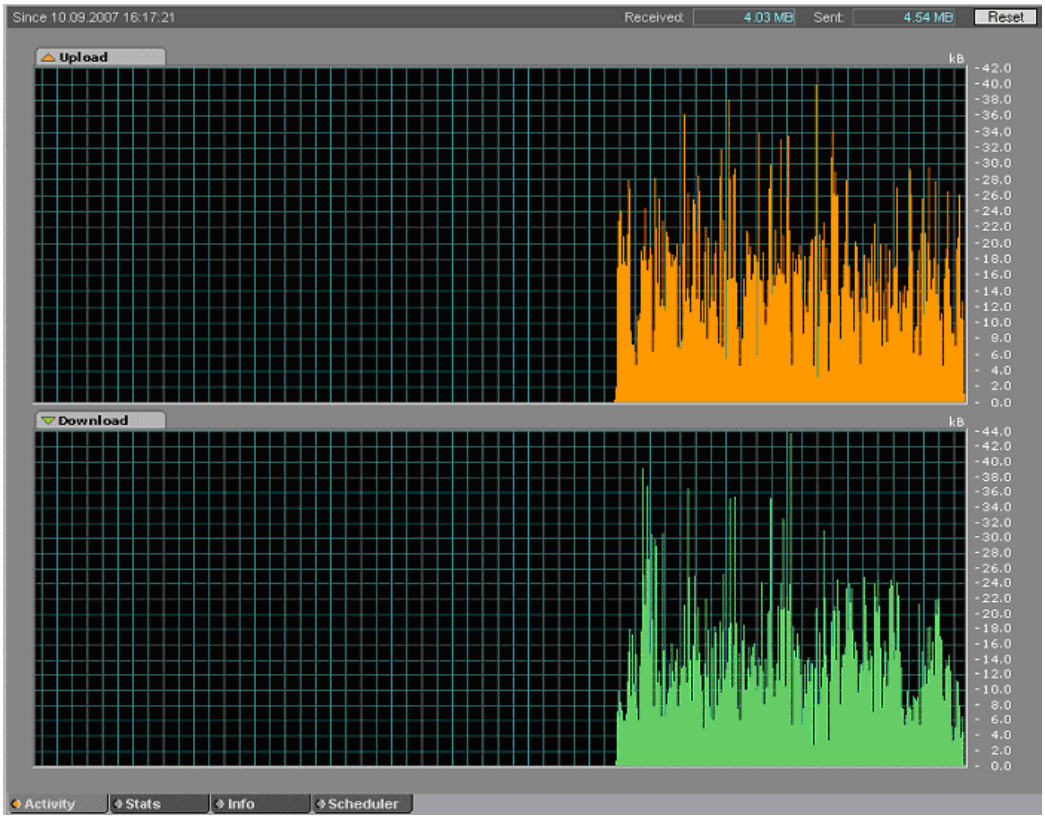
Şekil A.9: Test 6-1 Sırasında Konak 1'deki Sistem Kaynağı Kullanımı



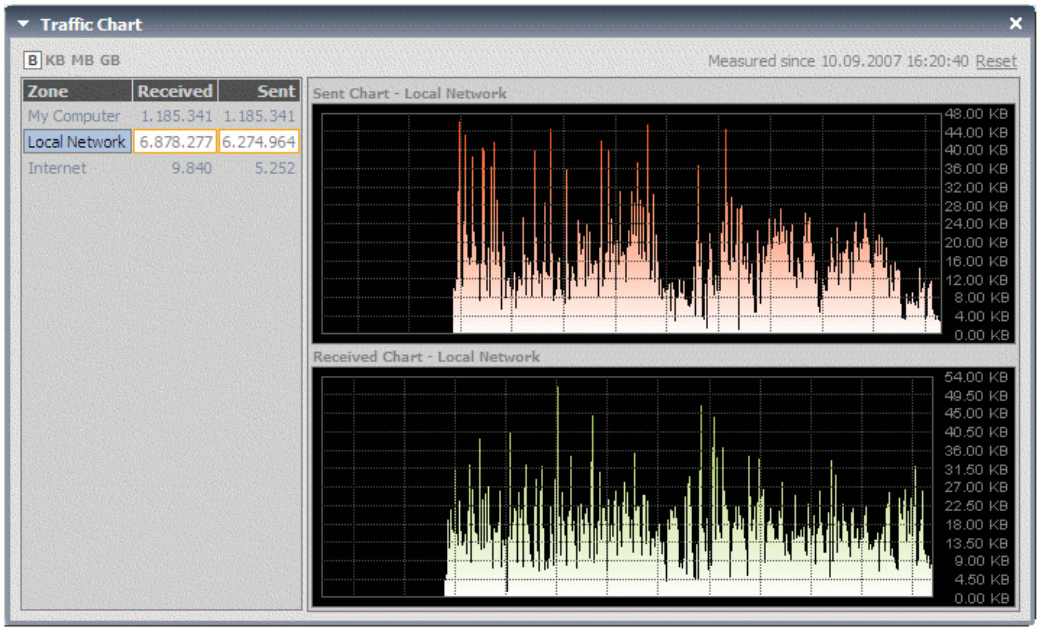
Şekil A.10: Test 6-1 Sırasında Konak 1'deki Ağ Trafiği



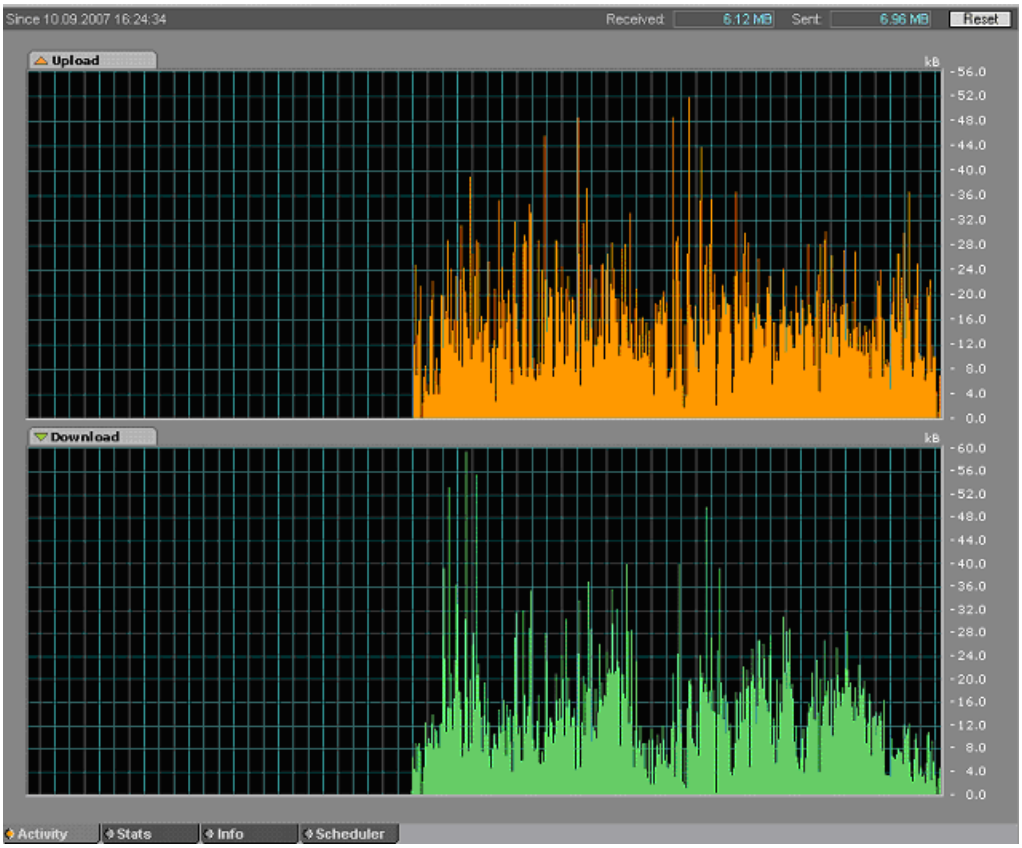
Şekil A.11: Test 6-1 Sırasında Konak 2'deki Sistem Kaynağı Kullanımı



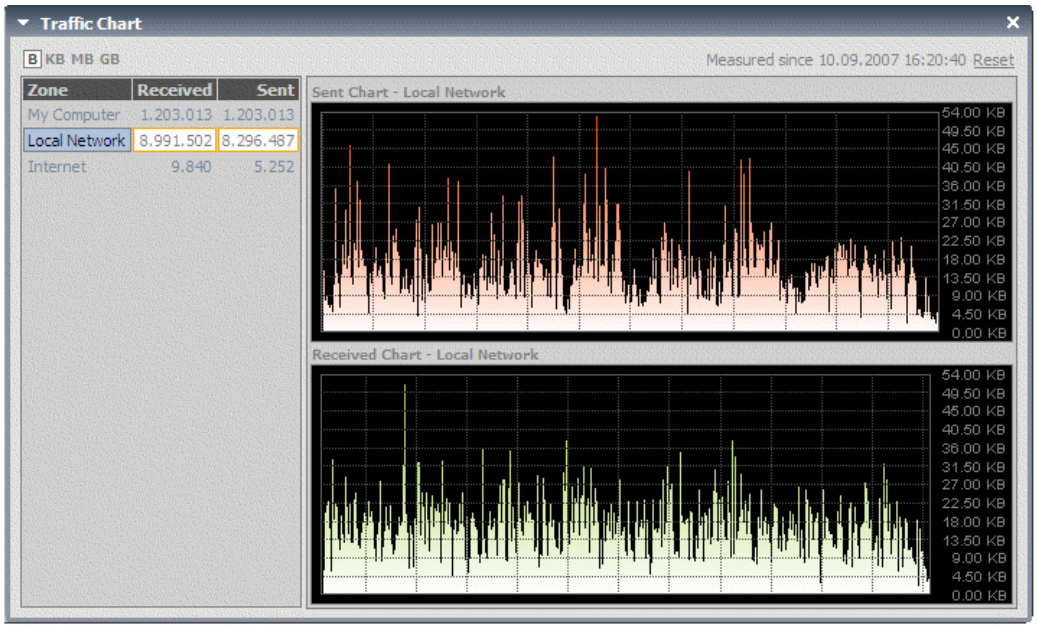
Şekil A.12: Test 6-1 Sırasında Konak 2'deki Ağ Trafiği



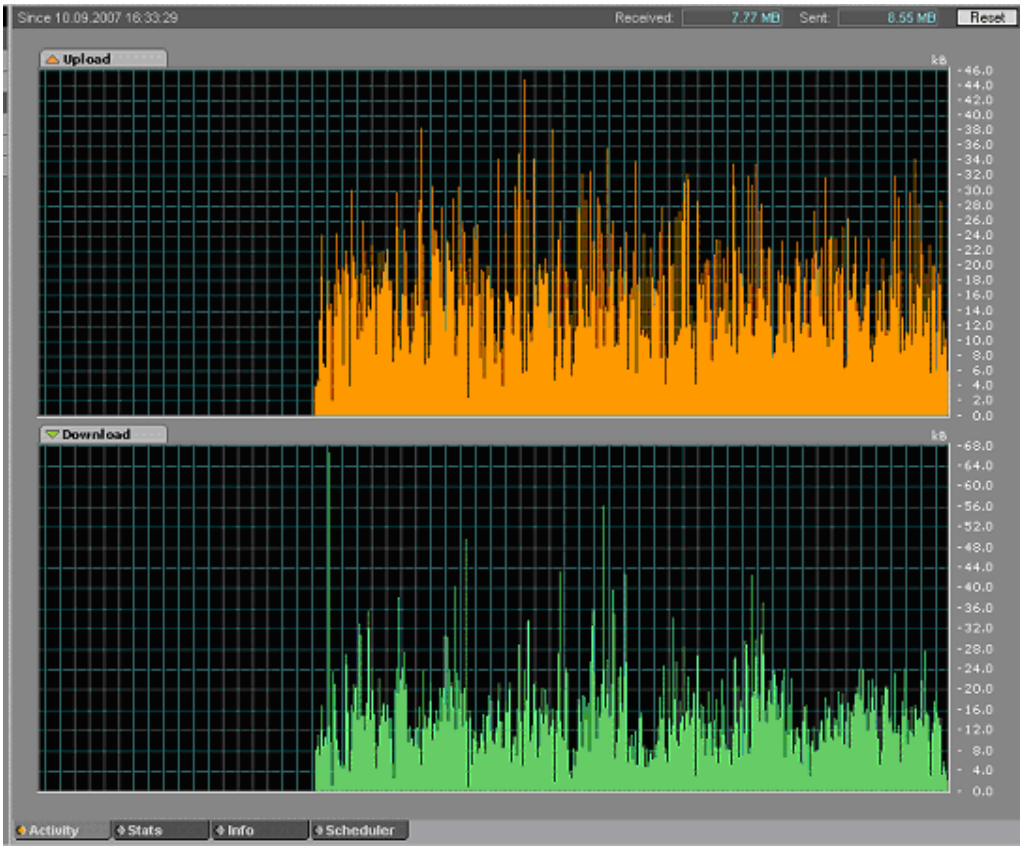
Şekil A.13: Test 6-2 Sırasında Konak 1'deki Ağ Trafiği



Şekil A.14: Test 6-2 Sırasında Konak 2'deki Ağ Trafiği



Şekil A.15: Test 6-3 Sırasında Konak 1'deki Ağ Trafiği



Şekil A.16: Test 6-3 Sırasında Konak 2'deki Ağ Trafiği

ÖZGEÇMİŞ

Cevher Cemal Bozkur 1981 yılında Mardin'de doğmuştur. 2004 yılında İstanbul Teknik Üniversitesi Bilgisayar Mühendisliği Bölümü'nden lisans derecesini almıştır.