

TC.
ÇANAKKALE ONSEKİZ MART ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
YÜKSEK LİSANS TEZİ

IPV6 DESTEKLİ VIDEO KONFERANS YAZILIMI
TASARIMI VE GELİŞTİRİLMESİ

Cem SÖNMEZ

Bilgisayar Mühendisliği Anabilim Dalı

Tezin Sunulduğu Tarih: 29.06.2010

Tez Danışmanları:

Doç. Dr. Mehmet Ali SALAHLI

Dr. Necdet YÜCEL

ÇANAKKALE

YÜKSEK LİSANS TEZİ SINAV SONUÇ FORMU

CEM SÖNMEZ tarafından **DOÇ. DR. MEHMET ALİ SALAHLI** ve **DR. NECDET YÜCEL** yönetiminde hazırlanan “**IPV6 DESTEKLİ VIDEO KONFERANS YAZILIMI TASARIMI VE GELİŞTİRİLMESİ**” başlıklı tez tarafımızdan okunmuş, kapsamı ve niteliği açısından bir Yüksek Lisans tezi olarak kabul edilmiştir.

Doç. Dr. Mehmet Ali SALAHLI

Danışman

Prof. Dr. İsmail TARHAN

Jüri Üyesi

Yrd. Doç. Dr. Ali Murat TIRYAKI

Jüri Üyesi

Sıra No :

Tez Savunma Tarihi : 29/06/2010

Prof. Dr. İsmail TARHAN

Müdür

Fen Bilimleri Enstitüsü

Hazırlanan bu Yüksek Lisans tezi TÜBİTAK tarafından 108G102 no'lu projeden desteklenmiştir.

İNTİHAL (AŞIRMA) BEYAN SAYFASI

Bu tezde görsel, işitsel ve yazılı biçimde sunulan tüm bilgi ve sonuçların akademik ve etik kurallara uyularak tarafımdan elde edildiğini, tez içinde yer alan ancak bu çalışmaya özgü olmayan tüm sonuç ve bilgileri tezde kaynak göstererek belirttiğimi beyan ederim.

Cem SÖNMEZ

TEŐEKKÜR

Bu tez alıőmasının hazırlanması süresince kendi bilgi ve deneyimlerini benden esirgemeyen tez danışmanlarım Do. Dr. Mehmet Ali SALAHLI ve Dr. Necdet YÜCEL'e, lisans hocam Okt. Kemal Ökmen'e, alıőma arkadaşım Öğr. Gör. Sedat Gündüz'e ve her zaman desteęini gördüğüm Figen Bozkurt'a içten teşekkürlerimi sunuyor, bu tez alıőmamı onlara ithaf ediyorum.

Cem SÖNMEZ

SİMGELER VE KISALTMALAR LİSTESİ

- AOP: Yaklaşım Tabanlı Programlama (Aspect-oriented Programming)
- API: Uygulama Programlama Arabirimi (Application Programming Interface)
- CPU: Merkezi İşlem Birimi (Central Processing Unit)
- DHTML: Dinamik HTML (Dynamic HTML)
- DOM: Belge Nesne Modeli (Document Object Model)
- DV: Dijital Video (Digital Video)
- EJB: Kurumsal Java Bean (Enterprise Java Bean)
- FI6EN: Özgür Ipv6 Video Konferans (Free Ipv6 Video Conference)
- FLV: Flash Video (Flash Video)
- HTTP: Hipermetin Aktarma İletişim Kuralı (Hypertext Transfer Protocol)
- HTTPS: Güvenli Hipermetin Aktarma İletişim Kuralı (HTTP Secure)
- IETF: İnternet Mühendisliği Görev Gücü (Internet Engineering Task Force)
- IP: İnternet Protokolü (Internet Protocol)
- IPv6: İnternet Protokol 6 (Internet Protocol 6)
- IPSec: İp Güvenliği (Ip Security)
- ISDN: Tümleşik Hizmetler Sayısal Şebekesi (Integrated Services Digital Network)
- J2EE: Java Platform Kurumsal Sürüm (Java Platform Enterprise Edition)
- JDBC: Java Veritabanı Bağlantısı (Java Database Connectivity)
- JPEG: Birleşik Fotoğraf Uzmanları Grubu (Joint Photographic Experts Group)
- LAN: Yerel Alan Ağı (Local Area Network)
- LPS: Laszlo Sunum Sunucusu (Laszlo Presentation Server)
- MAC: Ortam Erişim Kontrolü (Media Access Control)
- MVC: Model-Görünüm-Denetleyici (Model-View-Controller)
- NAT: Ağ Adres Dönüşümü (Network Address Translation)
- ORM: Nesne İlişkisel Eşleme (Object Relational Mapping)
- POJO: Basit Java Nesnesi (Plain Old Java Object)
- RIA: Zengin İnternet Uygulaması (Rich Internet Application)
- RMI: Uzak Metot Çağırımı (Remote Method Invocation)
- RTMP: Gerçek Zamanlı Mesajlaşma Protokolü (Real Time Messaging Protocol)
- RTMPT: Tünelenmiş Gerçek Zamanlı Mesajlaşma Protokolü (RTMP Tunneled)
- RTMPS: Güvenli Gerçek Zamanlı Mesajlaşma Protokolü (RTMP Secure)
- SAX: XML için Basit Uygulama Programlama Arabirimi (Simple API for XML)

SGML: Standart Genellenmiş İşaretleme Dili (Standard Generalized Markup Language)
SLAAC: Otomatik Adres Yapılandırması (Stateless Address Autoconfiguration)
SLF4J: Java için Basit Günlükleme Cephesi (Simple Logging Facade for Java)
SOAP: Basit Nesne Erişim Protokolü (Simple Object Access Protocol)
SQL: Yapısal Sorgulama Dili (Structured Query Language)
SSL: Güvenli Yuva Katmanı (Secure Socket Layer)
SWF: Ufak Web Formatı (Small Web Format)
SVN: Alt Versiyon (Subversion)
TCP: Veri İletim Kontrol Protokolü (Transmission Control Protocol)
URL: Birörnek Kaynak Konumlayıcı (Uniform Resource Locator)
XML: Genişletilebilir İşaretleme Dili (Extensible Markup Language)
W3C: Dünya Çapında Ağ Birliği (World Wide Web Consortium)
WAR: Web Uygulama Arşivi (Web Application Archive)
WSDL: Web Servisi Belirtme Dili (Web Services Description Language)

ÖZET

IPV6 DESTEKLİ VIDEO KONFERANS YAZILIMI TASARIMI VE GELİŞTİRİLMESİ

Cem SÖNMEZ

Çanakkale Onsekiz Mart Üniversitesi

Fen Bilimleri Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalı Yüksek Lisans Tezi

Danışmanlar:

Doç. Dr. Mehmet Ali SALAHLI

Dr. Necdet YÜCEL

29/06/2010, 45

Bu tez çalışmasının amacı; video konferans yazılımlarının özellikleri, kullanım alanları, tasarımı ve gerçekleştirilmesi konularında araştırma yapmak ve bir video konferans yazılımı geliştirmektir. Geliştirilmiş olan video konferans yazılımında IPv6 ağ protokolünün kullanımının ne gibi avantajlar sağlayacağı ve video konferans kalitesini ne yönde artırabileceği konusunda analizler yapılmıştır.

Yazılımının geliştirme sürecinde, ilk olarak yetenekli bir video konferans yazılımının ortaya çıkarılması hedeflenmiş olup, daha sonraki aşamada ise IPv6 bağlantı desteği eklenmiştir. IPv6 protokolünün beraberinde getirdiği kimi özelliklerin video konferans kalitesini artırabileceği öngörülmüştür. Bu nedenle; üzerinde IPv6 verimlilik testlerinin yapılabileceği bir video konferans yazılımı geliştirilmiştir.

Anahtar sözcükler : Video konferans, IPv6, Web konferans, Uzaktan eğitim

ABSTRACT

DEVELOPMENT OF IPV6 SUPPORTED VIDEO CONFERENCE SOFTWARE

Cem SÖNMEZ

Çanakkale Onsekiz Mart University

Graduate School of Science and Engineering

Chair for Computer Engineering Thesis of Master of Science

Advisors:

Assoc. Prof. Dr. Mehmet Ali SALAHLI

Dr. Necdet YÜCEL

29/06/2010, 45

The purpose of this thesis is to make a research about the features, area of usage, design and development of the video conference softwares and to develop a video conference software. Also in this study, some analysis has been performed about what kind of advantages may be provided by using IPv6 network protocol and how this will affect to the quality of the video conferencing during development of the software.

For the lifecycle of Fi6en development; the first aim is to create a skilled video conference software and at the second phase IPv6 connection support has been added. It is predicted that; some of the features that comes with IPv6 may increase the quality of video conferencing. Thus, IPv6 efficiency tests can be performed on a video conferencing software has been developed.

Keywords : Video conference, IPv6, Web conference, Distance education

İÇERİK	Sayfa
TEZ SINAVI SONUÇ BELGESİ.....	İİ
İNTİHAL (AŞIRMA) BEYAN SAYFASI	İİİ
TEŞEKKÜR.....	İV
SİMGELER VE KISALTMALAR.....	V
ÖZET.....	Vİİ
ABSTRACT.....	Vİİİ
BÖLÜM 1 – GİRİŞ.....	1
BÖLÜM 2 – VIDEO KONFERANS VE IPV6.....	2
2.1. Video Konferans.....	2
2.1.1. Video Konferans Yazılımı Kullanımının Avantajları	2
2.1.2. Mevcut Video Konferans Yazılımları	3
2.1.3. İyi Bir Video Konferans Yazılımının En Temel Özellikleri.....	4
2.2. İnternet Protokol 6 (IPv6).....	6
2.3. Video Konferans'ın IPv6 Ağı Üzerinde Gerçekleştirim Testi	7
BÖLÜM 3 – TASARIM	9
3.1. Fi6en Modül Yapısı.....	9
3.2. Fi6en Hedef ve Gereksinimleri	10
3.3. Fi6en Yazılım Mimarisi	14
3.4. Fi6en Paket ve Sınıf Hiyerarşisi.....	15
3.5. Kullanılan Teknoloji ve Araçlar	18
3.5.1. Openlaszlo.....	18
3.5.2. Red5	19
3.5.3. Spring.....	20
3.5.4. Hibernate	21
3.5.5. Genişletilebilir İşaretleme Dili (XML).....	25
3.5.6. Nesne İlişkisel Eşleme (ORM).....	25
3.5.7. Gerçek Zamanlı Mesajlaşma Protokolü (RTMP)	26
3.5.8. Fi6en Ses ve Görüntü Kodlayıcıları.....	26
3.6. Hata Raporlama ve Yapılacaklar Listesi (todo) Yönetimi	27

3.7. Java ve Ipv6.....	27
3.7.1. Java Uygulama Etkisi.....	27
3.7.2. İletişim Senaryoları	27
3.7.3. IPv6 Ağ Özellikleri ve JVM Parametreleri	28
3.8. Video Konferans Kalite Ölçütleri.....	29
BÖLÜM 4 – GERÇEKLEŞTİRİM.....	30
4.1. Günlükleme Desteği	30
4.2. Fi6en Mesajlaşma Modülü.....	32
4.3. Fi6en Kimlik Kanıtlama Modülü	34
4.3.1. Kullanılan Sınıflar ve Ayar Dosyaları	34
4.4. Fi6en Çoklu Dil Desteği	35
4.5. Fi6en Oda Tipi Video Konferans Modülü.....	37
4.6. Fi6en Kayıt Modülü	39
4.7. Demo Uygulaması	40
4.8. Fi6en (Free Ipv6 Video Conference) Kullanım Kılavuzu	40
BÖLÜM 5 – SONUÇLAR VE ÖNERİLER	43
KAYNAKLAR	44
Çizelgeler	I
Şekiller	II
Özgeçmiş.....	III

BÖLÜM 1**GİRİŞ**

Tez çalışmasının bu bölümünde tezin amacı ve önemine değinen bir giriş yapılmış; ikinci bölümde ise video konferans ve IPv6 konusunda yapılan ön çalışmalar anlatılmıştır. Üçüncü ve dördüncü bölümlerde ise yazılım tasarımı ve geliştirilmesi konusunda yapılan çalışmalar yer almaktadır.

Video konferans sistemi, yerel ve uzak konumlarda bulunan çoklu kullanıcıların haberleşmesi, eşzamanlı ve etkin olarak grup çalışması yapmasını yönetmek için kullanılan yazılımlardır. (Chang-Gen ve ark., 2009). Bilgisayar ağının ve multimedya teknolojisinin gelişmesi ile birlikte medya akım teknolojisi daha da popüler hâle gelmektedir. Bu teknolojinin bir gerçekleştirimi olan video konferans yazılımlarının kullanımı her geçen gün artmaktadır. Özellikle eğitim, iş, sağlık ve sosyal iletişim alanlarında, son zamanlarda sıklıkla tercih edilen bir yazılım türü haline gelmiştir. Piyasada pek çok video konferans yazılım alternatifi bulunmaktadır ancak, mevcut video konferans yazılımlarının hemen hepsi Genel Kamu Lisansı (GPL) (Anonim, 2010e) olmayan yazılımlardır. Açık kaynak kodlu olan video konferans yazılımlarının çoğu ise henüz yeni geliştirilmekte olan projelerdir. Bununla birlikte, daha önce bu konuda yapılmış bir Türkçe yazılım çalışması ve belgelendirmesi bulunmamaktadır. Bu nedenle, tamamı ile açık kaynak kodlu ve Türkçe belgelendirmesi olan bir video konferans yazılımı geliştirilmiştir.

Yeni nesil internet protokolü olan IPv6 ise, IPv4'ün eksik yönlerini gidermek amacı ile 1998 yılından bu yana geliştirilmektedir. Özellikle IPv4 adres uzayının gün geçtikçe yetersiz hâle gelmesi ile birlikte, IPv6 kullanımı gün geçtikçe artmaktadır. IPv6, IPv4'e göre çok daha büyük (2^{96} kat) bir adres uzayına sahiptir. Bunun dışında; gezicilik, otomatik yapılandırma, çoklu gönderim gibi özellikler IPv6'yı bir adım öne çıkarmaktadır.

Günümüzde pek çok kullanıcının video konferans yazılımı kullanımına ihtiyaç duyduğu ve IPv6'nın pek çok yeniliği beraberinde getirdiği göz önünde bulundurulacak olursa; IPv6 desteği olan bir video konferans yazılımının üretilmesi oldukça faydalı olacaktır. IPv6'nın bazı yeniliklerinin (Çoklu gönderim, gezicilik, servis kalitesi, hız vs.), aslında tam da bir video konferans yazılımının ihtiyaç duyabileceği özelliklerdir. Bu amaç doğrultusunda, verimli IPv6 testlerinin yapılması için kullanılacak bir video konferans yazılımının oluşturulması esas alınmıştır.

BÖLÜM 2**VIDEO KONFERANS VE IPV6****2.1. Video Konferans**

Video telekonferans olarak da bilinen iki veya daha farklı noktanın birbiriyle eşzamanlı olarak sesli ve görüntülü iletişimini sağlayan bir telekomünikasyon teknolojisidir. Video konferans teknolojisi aslında ses ve görüntü akımlarının sıkıştırılıp, paketlere bölündükten sonra gönderilmesi ve karşı taraftan ses veya görüntünün alınmasıdır. Bu gönderme IP veya ISDN hattı üzerinden yapılabilmektedir.

Tümleşik Hizmetler Sayısal Şebekesi (ISDN): ISDN numarasının kullanılması; bu numaranın sahibi ile gerçek zamanlı olarak video konferans yapabileceği anlamına gelir. ISDN, Telekom santrallerinden alınır. ISDN hatlarla yapılan görüşmelerde ücretlendirme kontör hesabına göre yapılmaktadır.

İnternet Protokolü (IP): Video konferans bilinen IP protokolü üzerinden yapılmaktadır

Bir video telekonferans sisteminde gerekli olan bileşenler:

- *Video Girişi:* Video veya web kamera
- *Video Çıkışı:* Bilgisayar, televizyon veya projeksiyon cihazı
- *Ses Girişi:* Mikrofon
- *Ses Çıkışı:* Genellikle görüntü cihazı ile bütünleşik olan hoparlör ya da telefon
- *Veri Alışverişi:* Analog ya da dijital telefon ağı, yerel alan ağı (LAN) ya da internet

2 tip video konferans vardır:

1-) Oda Tipi Video Konferans

Tüm gerekli bileşenler, genellikle uzaktan kumandalı video kamera gibi bir cihazın içerisinde bulunur. Bu sistemler pahalıdır ve çok fazla kullanılmamaktadır.

2-) Masaüstü Video Konferans

Video konferans için bilinen masaüstü bilgisayar sistemleri kullanılmaktadır. (Anonim,2010b)

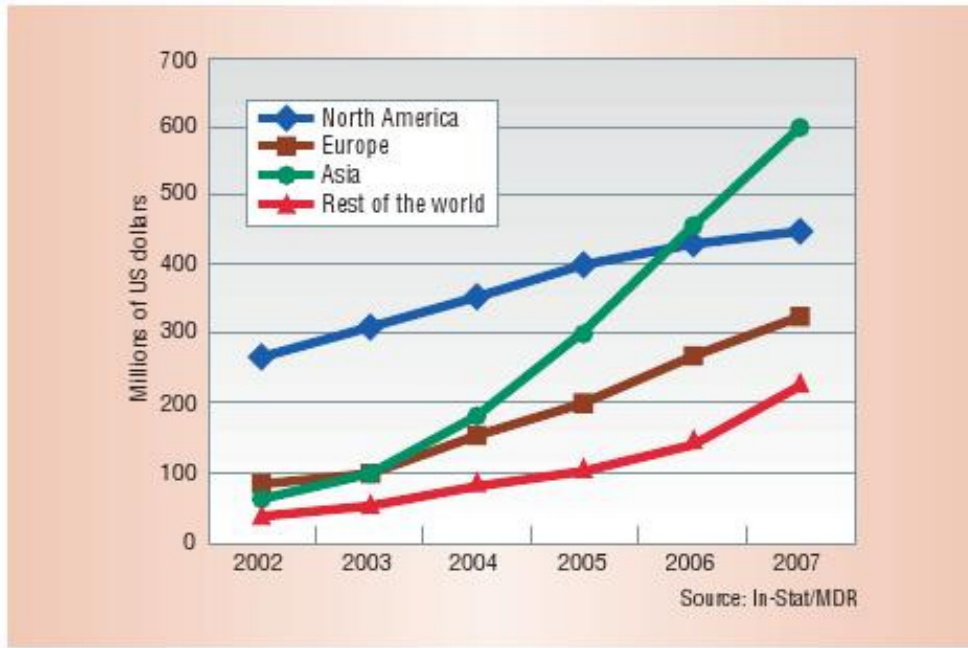
2.1.1. Video Konferans Yazılımı Kullanımının Avantajları

Video konferans yazılımlarının kullanımının beraberinde getirdiği pek çok avantaj vardır. Video konferans yazılımı kullanımı ile aynı yerde olmayan kullanıcılar, yol ve zaman masraflarını minimuma indirerek sesli veya görüntülü iletişim yapabilmektedir.

Ayrıca video konferans yazılımı, iş, eğitim, sağlık vs. alanlarındaki seminerlerde ve görüşmelerde de sıkça tercih edilen bir teknoloji hâline gelmiştir.

eDial, Juniper Network, Nortel Network ve Polycom gibi üreticiler, şirketlerin kendi iç ağlarında koşabilecek sunucu tabanlı web konferans yazılımlarını satmaya başlamışlardır.

Gerry Kaufhold'a göre; Şekil 1.'de görüldüğü üzere: 2002-2004 yılları arası web konferans yazılımı geliri %70 oranda artarak 450 milyon Dolar'dan 765 milyon Dolar'a ulaşmıştır. 2007'de ise bu değer 1,5 milyar Dolar'a ulaşmıştır. (Geer, 2005)



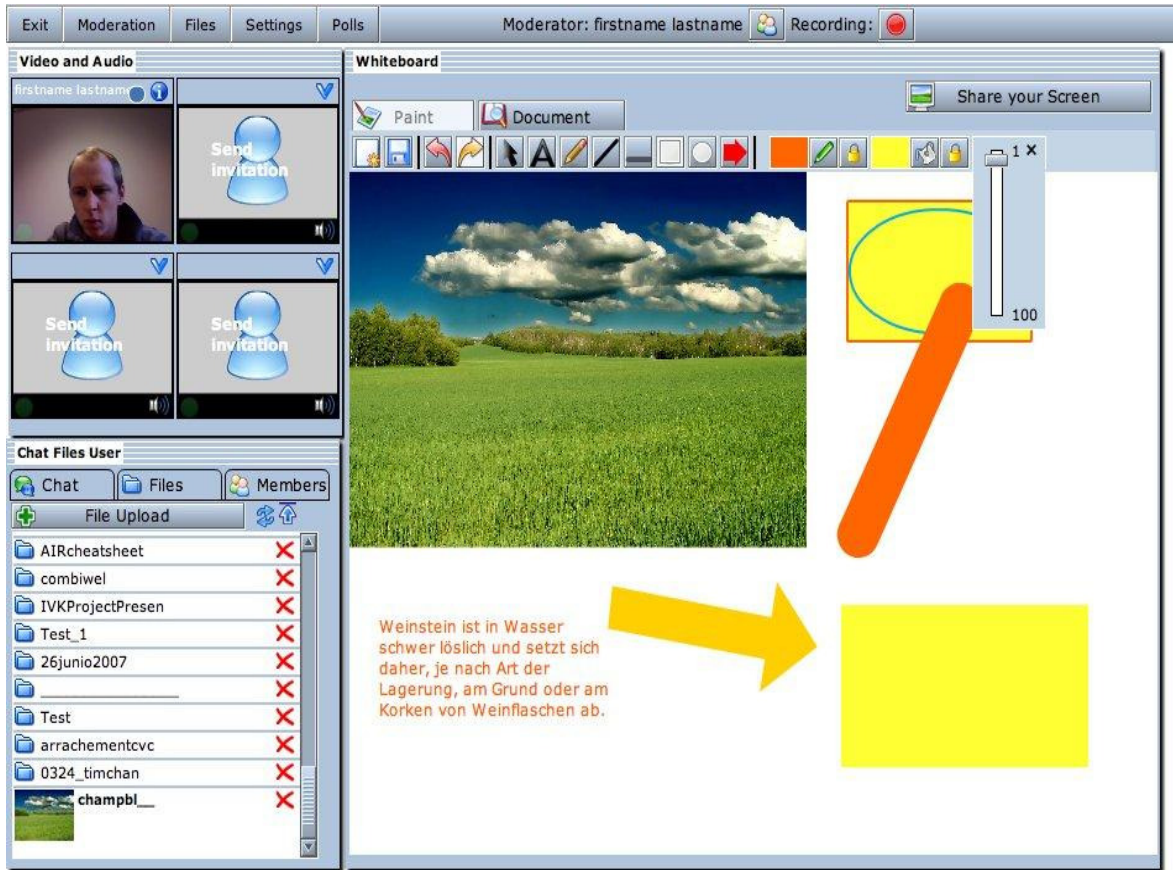
Şekil 1. Video konferans yazılımı kullanım gelirinin dağılımı.

(Geer, 2005)

2.1.2. Mevcut Video Konferans Yazılımları

Piyasada pek çok video konferans yazılımı bulunmaktadır. Bu yazılımlardan bazıları:

- Openmeetings (Anonim, 2009a)
- Microsoft NetMeeting (Anonim, 2009b)
- Camfrog (Anonim, 2009c)
- Dimdim (Anonim, 2009d)
- Big Blue Buton (Anonim, 2009e)



Şekil 2. Openmeetings ekran görüntüsü.

(Anonim, 2009a)

Belirtilen video konferans yazılımları içerisinde Microsoft NetMeeting, Camfrog ve Dimdim kapalı kaynak kodlu yazılımlardır. Openmeetings (Şekil 2.) ise; Eclipse Public License'a sahip, açık kaynak kodlu bir video konferans yazılımıdır. Openmeetings'in hâlihazırda IPv6 desteği bulunmaktadır. (Anonim, 2009a)

2.1.3. İyi Bir Video Konferans Yazılımının En Temel Özellikleri

Kaliteli bir video konferansın gerçekleştirilebilmesi için, kullanılacak olan yazılımın, kullanıcıların ihtiyaçlarına yanıt verebilir nitelikte olması gereklidir. Bu özellikler içerisinde, etkin bir video konferans yazılımının sahip olması gereken en temel özellikler şu şekildedir: (Yu ve Rui, 2008)

- *Kolay Kullanım*

Kolay kullanım ilkesi, tüm bilgisayar sistemlerinde olduğu gibi video konferans yazılımları için de her zaman en önemli tasarım ilkesi olmuştur. Kullanımı zor olan bir video konferans yazılımı ne kadar işlevse olursa olsun kullanıcılar tarafından

yeterli ilgiyi görmeyecektir. Bu nedenle video konferans yazılımı tasarımında, arayüz kısmı özenle üzerinde durulması gereken bir konudur. Yapılan araştırmalara göre, video konferans yazılımlarında kolay kullanım açısından konusunda iki sonuca varılmıştır :

1. Kullanıcıların Eski Alışkanlıklarını Takip Etmek

Bir kullanıcıya, yeni bir aracın ya da yazılımın kullanımını öğretmenin en kolay yolu; kullanıcının daha önce alışkın olduğu önceki araca benzer şekilde çalışacak bir ürün tasarlamaktır. Örneğin; video konferans sırasında kullanıcı bir sunu paylaşacak ise, video konferans yazılımının sunu paylaşma bölümü, var olan sunum programlarının (Powerpoint, OpenOffice.org Impress) arayüzüne benzer şekilde hazırlanır ise kullanıcı, zaten alışık olduğu bu görünüme anlamakta bir problem yaşamayacaktır.

2. Kullanılmayanı Gizle

Bir video konferans yazılımının sahip olacağı pek çok farklı özelliği vardır. Örneğin; mesajlaşma, sunu paylaşma ve randevu ayarlama bölümünün tümü birbirinden farklı işlemler yapan modüllerdir. Ayrıca bu bölümlerinin her birisinin de kendine özel işlemleri vardır. Tüm bu kısımları bir araya getirince, ortaya bir yığın işlem seçeneği çıkıyor. Bu noktada video konferans yazılımı tasarımında dikkat edilmesi gereken şey; birbirleri ile etkileşimi az olan işlemleri ana başlıklar halinde gruplandırmak ve kullanıcıyı sadece yapmak isteği alan ile ilgili olan eylemler ile karşı karşıya bırakmak olacaktır.

• *Özelleştirilebilirlik*

Video konferans yazılımının, kullanıcının ihtiyaçları doğrultusunda özelleştirilebilir bir yapıya sahip olması gerekir. Kullanıcı hoşuna giden servisleri video konferans sırasında aktif edebilmeli, istemediği servisleri ise devre dışı bırakabilmelidir.

• *Güvenilirlik*

Hiçbir kullanıcı, önemli bir görüşmesinin bir anda yarıda kesilmesini istemez. Bu nedenle, kararlı bir video konferans yazılımının tasarımında göz önünde bulundurulması gereken iki özellik vardır:

1. Bağımsız Servis Bileşeni

Uygulamanın çalışması esnasında herhangi bir hata oluşuyorsa, bunun nedeni genellikle belirli bir fonksiyondan kaynaklanıyordur. Bu nedenle, uygulamanın belirli bir modülünde hata oluşuyorsa, bu hata diğer modüllerin işleyişini engellememeli. Pek çok uygulamada, bir işlem yapılıyor iken, diğer işlemler

için arayüz etkisiz kılınıyor. Böyle bir durumda, mevcut işlemde bir hata olduğu zaman, eğer uygulama servislerinin çalışması birbirine bağımlı ise, uygulamanın tümü kullanılamaz duruma gelecektir. Bu tarz bir yaklaşımın yerine daha sağlam bir video konferans yazılımına için, birbirinden farklı servisler mümkün olduğunca ortak veriler üzerinde işlem yapabilmeli, birbirinden izole edilmiş bir şekilde çalışabilmelidirler.

2. Dağıtık Mimari

Merkezi bir sunucu genellikle problemin kaynağı durumundadır. Çünkü sürekli yüksek veri trafiği vardır ve saldırıların hedefi olmaktadır. Pek çok video konferans yazılımında, kullanıcı isteklerine cevap vermekle meşgul olan merkezi bir sunucu vardır. Bu durum güvenilir değildir, çünkü sunucu devre dışı olduğu anda, hiçbir konferans mümkün olmayacaktır. Bu sorunun aşılması için, sunucu hizmeti dağıtık bir mimariye sahip olmalıdır.

- *Güvenlik*

Video konferans sistemleri için güvenlik başlıca önemli konulardan biridir ve hemen hemen tüm sistemler güvenliğe oldukça önem verirler. Video konferans sistemlerinde sıkça kullanılan iki güvenlik mekanizması vardır :

1. Parola ile kimlik doğrulama
2. 128 bitlik SSL şifreleme

2.2. İnternet Protokol 6 (IPv6)

IPv6 (Internet Protocol Version 6) yeni nesil internet protokolüdür. Günümüzde pek çok kullanıcı, IP protokolü olarak IPv4 kullanmaktadır ve şu durumda bu protokol kullanıcıların ihtiyaçlarına yeteri derecede cevap vermektedir. Ancak zamanla Ipv4 protokolü kullanıcıların isteklerine yanıt vermeyecek düzeye geleceği öngörülmektedir. Bu nedenle Ipv4'ün eksik yönlerini gidermek amacıyla İnternet Mühendisliği Görev Gücü (IETF) tarafından 90'lı yıllarda IPv6 geliştirilmiştir (Deering ve Hinden, 1998). IPv6' ile gelen yenilikler şunlardır:

- *Daha Büyük Adres Uzayı*

Ipv4'de olan 32 bitlik adresleme IPv6' da 128 bite çıkarılmıştır. Bunun sebebi: şu an için mevcut IP yeterli olabilir, ancak ileride bu sayı yetmeyecektir. Bu nedenle Ipv4'den 2^{96} kat daha büyük adres uzayına sahip IPv6 protokolü geliştirilmeye başlanmıştır. Böylelikle ağ üzerindeki her makineye tekil bir IP verilebilmektedir.

- *Otomatik Yapılandırma (SLAAC)*

SLAAC sayesinde, istemcilerin ağ ayarları otomatik olarak yapılabilmektedir. Eğer IPv6 otomatik yapılandırması yoksa, istemciler ağ ayarlarını DHCPv6'dan veya elle alabilirler. IPv6' da ağ yapılandırması MAC adreslerine göre yapılmaktadır.

- *Çoklu Gönderim (Multicast)*

Grup adresleridir. Bu gruba üye olanların tamamı bu paketleri görür. Ipv4'deki broadcast yerine kullanılır. Böylece CPU'dan tasarruf sağlanmış olur.

- *Güvenlik*

Ipv4' de olduğu gibi IPv6'da da Ipsec desteklenmektedir.

Ipsec: Ipsec, IP seviyesinde yetkilendirme ve şifreleme sağlar. Belli bir rota (route) boyunca veri bozulmaması, değiştirilmemesi, bütünlüğü sağlanır.

Bu noktada Ipv6'nın güvenlik yönünden Ipv4'e göre bir artışı olmadığı düşünülebilir ama durum böyle değildir. Ipv4'de NAT dolayısıyla Ipsec'i çalıştırmak oldukça zordur. Sanal IP'ler sıkıntı yaratır.

- *Servis Kalitesi (Quality of Service)*

IP paketlerini önceliklendirme işlemi kolaylaşmıştır. Böylelikle gerçek zamanlı trafiğin kesinti olma olasılığı azaltılmıştır.

Ancak bundan daha önemlisi, paket başlığına “akış etiketi (flow label)” alanı eklenmiştir. Aynı akışa (stream) veya oturuma ait olan paketlerin ortak bir akış etiketi değerini paylaşmasıyla her paketin derinlemesine irdelenmeden tanınması kolaylaştırılmıştır. Akışın veya oturumun tanınması servis kalitesi mekanizmaları için yararlıdır.

- *Gezicilik (Mobility)*

IPv4' de olmayan aynı IP adreslerinin farklı ağlarda kullanımı IPv6 ile mümkün hale getirilmiştir.

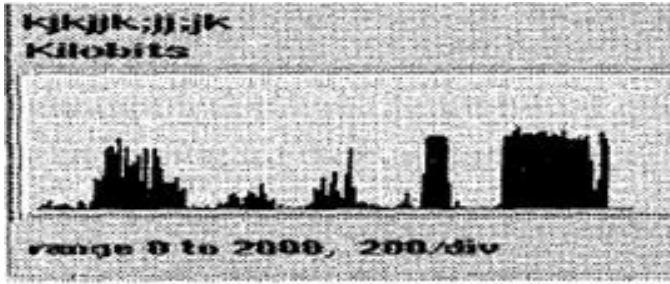
- *Hız*

IPv6' da olan trafik işgal edici paket başlıkları kaldırılarak bir hız artırımına gidilmiştir.

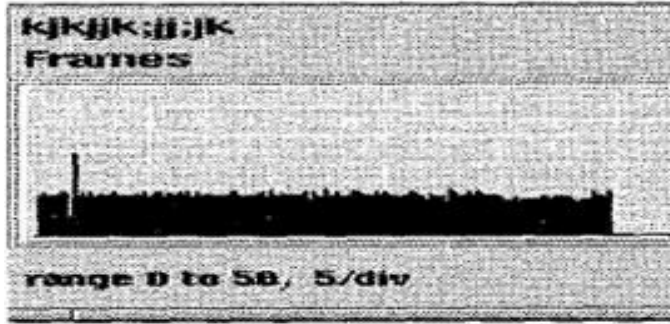
2.3. Video Konferans'ın IPv6 Ağı Üzerinde Gerçekleştirim Testi

Linux işletim sistemi üzerinde ve bir IPv6 ağı üzerinde video konferans gerçekleştirildiği koşullarda, iletim oranı (transmission rate) ve çerçeve oranı (frame rate)'nın gözlemlenmesi ile Şekil 3. ve Şekil 4.'teki sonuçlar elde edilmiştir (Chuan ve ark., 2002).

Bu şekiller dikkatle incelenecek olursa; çerçeve oranı konusunda IPv6 ile IPv4 arasında önemli derecede bir fark oluşmamaktadır(Şekil 3.-b, Şekil 4.-b). Ancak iletim oranı konusunda IPv6 ağındaki IPv4 ağına göre daha kararlı bir tablo görülmektedir(Şekil 3.-a, Şekil 4.-a). Video konferans gerçekleştirimi sırasında, IPv6 ağındaki kaynak iletimi IPv4 ağındaki kaynak iletimine göre daha az kesintiye uğramaktadır.

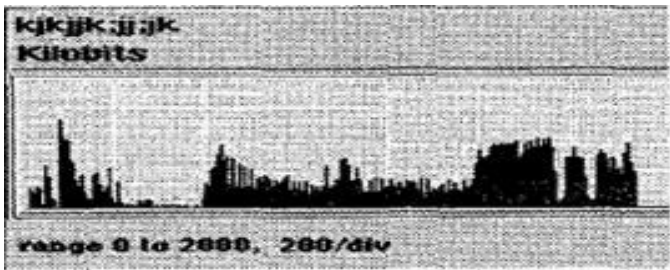


(a)



(b)

Şekil 3. Ipv4 ağı için (a) Kaynak iletim hızı ve (b) Çerçeve hızı.



(a)



(b)

Şekil 4. Ipv6 ağı için (a) Kaynak iletim hızı ve (b) Çerçeve hızı.

BÖLÜM 3

TASARIM

3.1. Fi6en Modül Yapısı

Fi6en modüller bir yapıya sahiptir. Her bir modül mümkün olduğunca birbirinden bağımsız olarak çalışmasını sürdürmektedir. Fi6en’de kullanılan ortak kaynaklar dışında her modül kendisine ait olan işlemleri yerine getirmektedir. Örneğin; mesajlaşma modülü sadece kullanıcılar arasındaki metin mesajlaşma ile ilgili olan işlemlerin yönetiminden sorumlu iken, ses/görüntü akım modülü video konferans sırasındaki ses/görüntü akışından sorumlu olan modüldür ve iki modülün de çalışması birbirini etkilememektedir.

Şekil 5.’te yer alan modüller Fi6en istemci uygulamasında bulunmaktadır. Bu modüllerden her birinin ne tür işlemleri yerine getirdiğinden bahsedilecek olursa:

Mesajlaşma Modülü: Kullanıcıların metin mesajlaşması ile ilgili işlemleri yürüten modüldür. Kullanıcının metin kutusuna yazmış olduğu mesajı sunucuya gönderir ve sunucudan aldığı mesajı diğer kullanıcılara gösterir.

Hata Modülü: Uygulamanın işleyişi sırasında, kullanıcıya gösterilecek olan hata mesajları bu modülden gönderilir. Bu hatalar, sunucunun çalışması sırasında ortaya çıkabilecek bağlantı ile ilgili aksaklıkları veya kullanıcının yanlış veya eksik bilgi girmesinden kaynaklanan mesajları içerebilir.

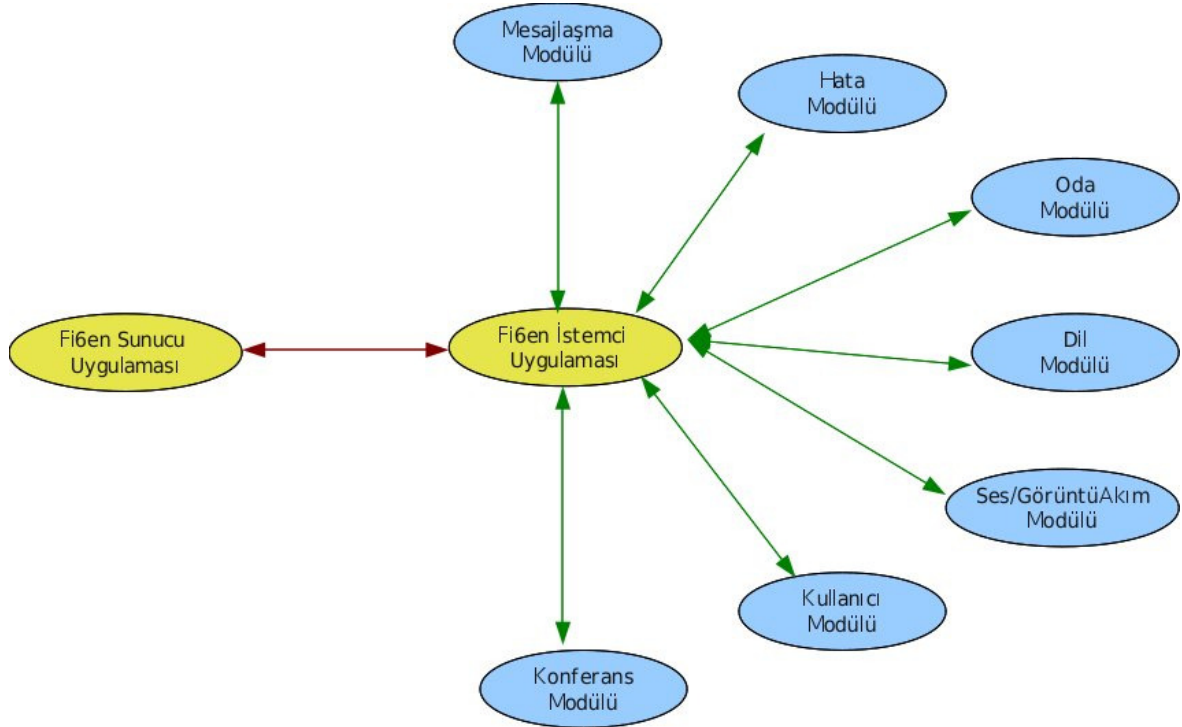
Oda Modülü: Oda tipi video konferans ile ilgili işlemler bu modül içerisinde gerçekleştirilir. Kullanıcının odaya giriş yapması, oda içerisindeki kullanıcıların yönetilmesi ve sadece o oda içerisindeki kullanıcılar tarafından görünür kılınabilecek paylaşılan nesnelerin yönetiminden sorumludur.

Dil Modülü: İstemci uygulaması arayüzündeki dil ile ilgili işlemler bu modül tarafından gerçekleştirilir. Çoklu dil desteğinin sağlanması için, dil değişikliklerinin uygulama esnasında uygulanabilir olması gene bu modül tarafından yürütülmektedir.

Ses/Görüntü Akım Modülü: Kullanıcı kamera görüntüsünün ve sesin alınıp sunucu tarafına gönderilmesi, paylaşılan bir akımın görüntülenmesi, kamera görüntüsünün kaydedilmesi, kayıtlı görüntülerin video oynatıcısında görüntülenmesi gibi işlemlerden sorumludur.

Kullanıcı Modülü: Kullanıcı kaydının oluşturulması, video konferansa giriş esnasında kimlik kanıtlanması, kullanıcı profilinin düzenlenmesi gibi işlemlerin yürütülmesinden sorumludur.

Konferans Modülü: Video konferans istemci uygulamasının çalışması sırasında, bahsedilen modüllerin yaptığı işler dışındaki genel işlemlerden sorumludur.



Şekil 5. Fi6en modül yapısı.

3.2. Fi6en Hedef ve Gereksinimleri

Mevcut video konferans yazılımları mimarileri ve alt yapısında kullanılan teknolojiler incelendikten sonra, proje kapsamında kullanılacak video konferans yazılımının aşağıdaki özelliklere sahip olması kararına varılmıştır. Yazılımın karmaşıklığını ve sonuçların kesin belli olmayışından dolayı yazılım geliştirilmesi süreci için evrimsel geliştirme modeli kullanılmıştır. Bu modele uygun olarak geliştirme sürecinde ürünün sürümleri-prototipleri geliştirilmiş ve incelenmiştir.



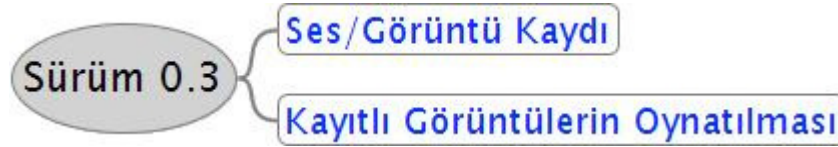
Şekil 6. Fi6en yol haritası - sürüm 0.1.

Yol haritasında, 0.1 sürümü ile Fi6en'e eklenilmesi öngörülen özellikler; metin mesajlaşma desteği, kullanıcı kimlik kanıtlaması ve iki kişi arasında sesli/görüntülü iletişimidir.



Şekil 7. Fi6en yol haritası - sürüm 0.2.

Yol haritasında, 0.2 sürümü ile Fi6en'e eklenilmesi öngörülen özellik ise iki daha veya fazla kullanıcının birbiri ile sesli/görüntülü iletişimidir.



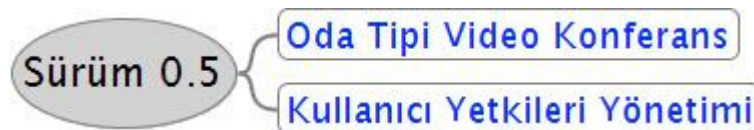
Şekil 8. Fi6en yol haritası - sürüm 0.3.

Yol haritasında, 0.3 sürümü ile Fi6en'e eklenilmesi öngörülen özellikler; kullanıcının kendi kamera görüntüsünü kaydedebilmesi ve daha önce kaydedilen görüntülere ulaşım Fi6en ile gömülü gelen oynatıcı üzerinde bu videoları oynatabilmesidir.



Şekil 9. Fi6en yol haritası - sürüm 0.4.

Yol haritasında, 0.4 sürümü ile Fi6en'e eklenilmesi öngörülen özellikler; kolay kullanım için kullanıcı arayüzü iyileştirilmesi ve arayüze Türkçe'nin yanı sıra İngilizce dil desteğinin eklenmesidir.



Şekil 10. Fi6en yol haritası - sürüm 0.5.

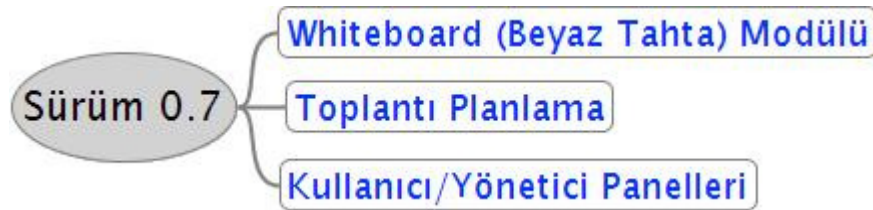
Yol haritasında, 0.5 sürümü ile Fi6en'e eklenilmesi öngörülen özellikler; belirli kullanıcıların özel görüşmeler yapabileceği oda tipi video konferans modülünün eklenmesi ve yaratılan bu odalar içerisinde her bir kullanıcıya yetkilendirme (oda sahibi, katılımcı, izleyici vs.) yapılmasıdır.



Şekil 11. Fi6en yol haritası - sürüm 0.6.

Yol haritasında 0.6 sürümü ile Fi6en'e eklenilmesi öngörülen özellikler şunlardır:

- İstemci uygulamasının bağlantı kuracağı sunucu olarak Ipv6 adresine sahip bir alan adı belirtilebilecek ve ilgili sunucu Ipv6 adresine sahip bir istemciden gelen bağlantı isteklerini kabul edebilecektir. Sunucu ilk olarak istemcinin Ipv6 adresi ile iletişim kurmaya çalışacaktır. Eğer istemci bir ipv6 adresine sahip değilse, sunucu bu istemci ile ipv4 adresini kullanarak iletişimde bulunacaktır.
- 0.4 sürümü ile birlikte iyileştirilmiş olan arayüz bir adım daha ileriye götürülecektir.
- Kullanıcıların uygulamayı sisteme nasıl kurabileceği ile ilgili olarak istemci tarafında arayüz hazırlanacaktır.

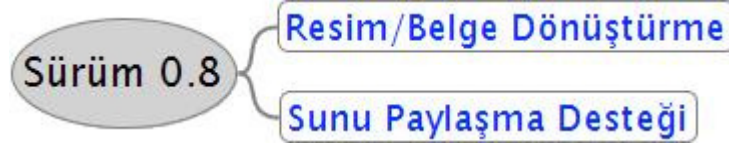


Şekil 12. Fi6en yol haritası - sürüm 0.7.

Yol haritasında 0.7 sürümü ile Fi6en'e eklenilmesi öngörülen özellikler şunlardır:

- Video konferansa katılan kullanıcıların tümü tarafından izlenebilecek olan, boş bir sayfa üzerindeki çizimler, işaretlemeler, metinler vs. içeren beyaz tahta modülü eklenecektir.

- İleri bir tarih için toplantı planlanabilecek ve toplantı tarihine gelindiğinde Fi6en, e-posta ile randevu katılımcılarına hatırlatma mesajı yollayacaktır.
- Normal kullanıcılar ile Fi6en sistem yöneticisi için ayrı ayrı paneller hazırlanacaktır.



Şekil 13. Fi6en yol haritası - sürüm 0.8.

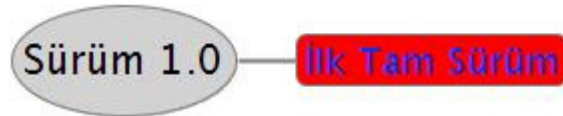
Yol haritasında 0.8 sürümü ile Fi6en'e eklenilmesi öngörülen özellikler şunlardır:

- Resim ve çeşitli dosyaların diğer dosya formatlarına dönüştürülüp kullanılabilmesi
- .pdf, .odp,.ppt gibi sunuların video konferans sırasında paylaşımı



Şekil 14. Fi6en yol haritası - sürüm 0.9.

Yol haritasında 0.9 sürümü ile Fi6en'e eklenilmesi öngörülen özellik; kayıtlı görüntülerin, yazışmaların vs. verileri yedekleme desteğidir.



Şekil 15. Fi6en yol haritası - sürüm 1.0.

1.0 sürümü Fi6en kararlı sürümüdür. Daha önceki sürümlerde belirtilen özelliklerin Fi6en'e eklenmesi ve mevcut hata ve açıkların giderilmesi ile birlikte tam sürüm duyurusu yapılacaktır.

3.3. Fi6en Yazılım Mimarisi

Video konferans yazılımını istemci-sunucu mimari modeli şeklinde geliştirilmektedir.

Sunucu tarafı

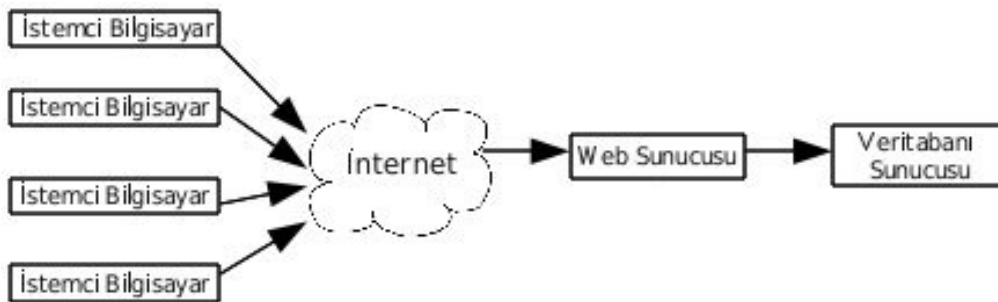
Video konferansı yönetecek ve kullanıcıların isteklerini işleyip yanıt gönderecek olan kısımdır. Arka tarafta çalışacak ve esas olarak tüm işlemlerin yapılacağı yer, sunucu tarafıdır. Bunlar; kamera görüntüsü ve sesin bir kullanıcıdan alınıp diğer kullanıcılara iletilmesi, paylaşılan ses/görüntü akımının kaydedilmesi, kayıtlı görüntü oynatımının yönetilmesi, metin mesajlaşmanın yönetilmesi, sunum vb. paylaşılan nesnelere kullanıcılara gösterilmesi vs. işlemleri olabilir.

İstemci tarafından Ipv6 bağlantısının gerçekleştirilebilmesi için gerekli yapılandırmalar bu kısımda yapılmaktadır.

İstemci tarafı

Video konferansı kullanacak olan kişilerin bulunduğu ortamdır. Yapılmak istenilen işlemlerle ilgili istekler sunucuya bu taraftan gönderilir. Yanıtlar da bu kısımda kullanıcıya sunulur. İstemci tarafındaki işlemler; kamera görüntüsünün görüntülenmesi, sesin iletimi, arayüz işlemleri, kayıtlı videoların görüntülenmesi, metin mesajlaşmanın görünümü ,sunuların paylaşılması vs. olabilir.

Fi6en, işletim sistemi ve donanımından bağımsız olarak tüm platformlarda çalışabilmesi için sunucu/(tarayıcı)istemci modelini kullanmaktadır. Sunucu/(tarayıcı)istemci modeli Şekil 16.'de görülmektedir.



Şekil 16. Sunucu/istemci modeli.

(Zhu, 2010)

Yukarıda anlatıldığı gibi; istemci ve sunucu bilgisayarlar internet veya yerel alan ağı içerisinde birlikte çalışabilmelidir. Video konferans sırasında istemci bilgisayar sunucu bilgisayardan bir istekte bulunur. Sunucu bilgisayar ise istemcinin isteğine göre, kendi sahip olduğu veya veritabanında bulunan bilgileri kullanarak kullanıcının isteğini yerine

getirir ve istemci bilgisayara sonucu geri döndürür. Örneğin; kullanıcı video konferansa giriş yapmak istediği sırada gerçekleşecek olan kimlik doğrulama senaryosu şu şekildedir:

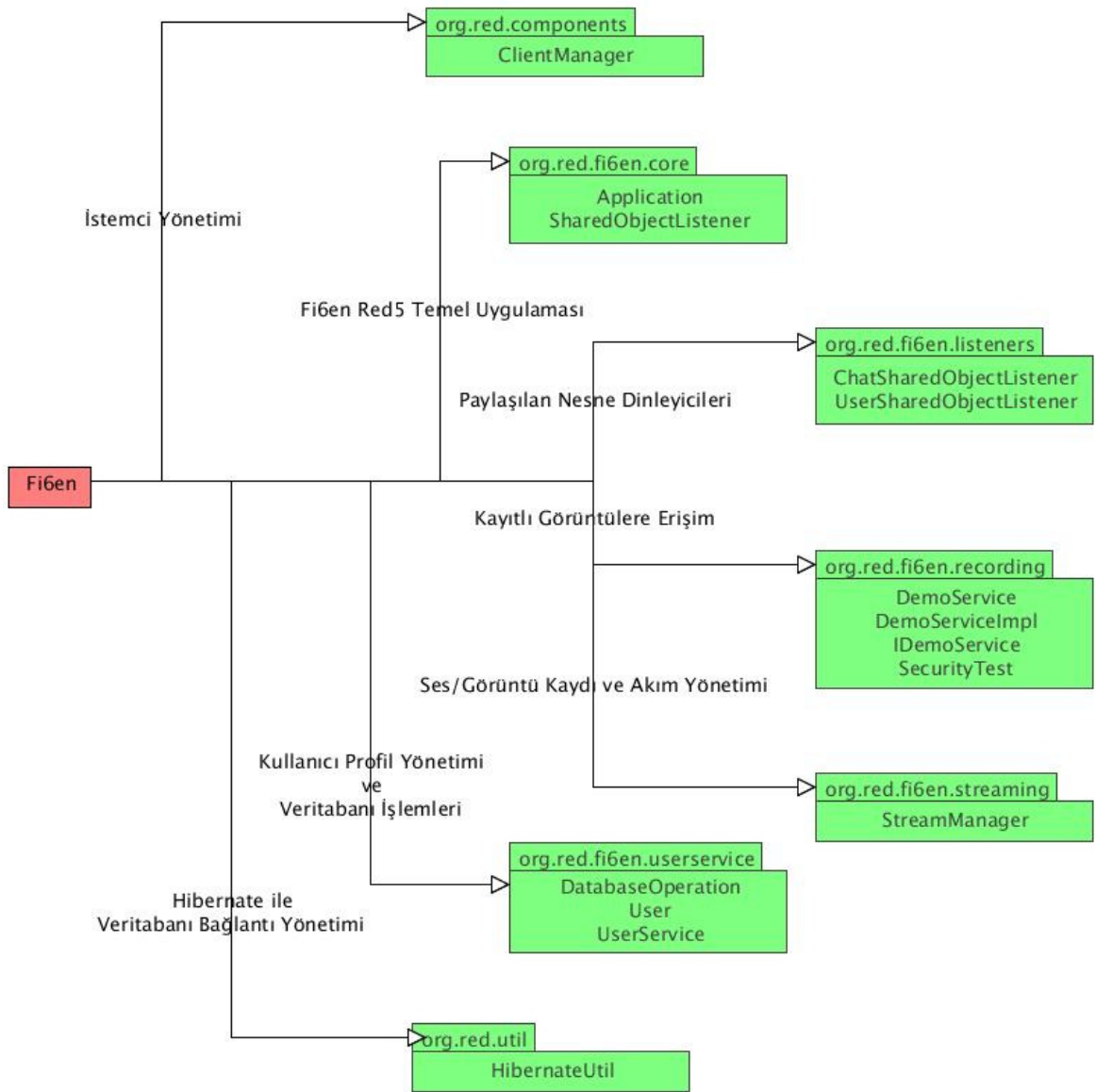
- İstemci bilgisayar sunucu bilgisayara kullanıcı adı ve parolasını gönderir.
- Sunucu bilgisayar, istemci bilgisayarın göndermiş olduğu kullanıcı adına ait kaydın bilgilerine erişim için veritabanı sunucusundan istekte bulunur.
- Veritabanı sunucusu, web sunucusundan gelen bu isteği işler ve istenilen bilgileri geri döndürür.
- Sunucu bilgisayar, veritabanından getirdiği kayıta bulunan kullanıcı parolası ile istemci bilgisayardan gelen parolayı karşılaştırır. Parolalar eşleşiyor ise; kullanıcıya kimlik doğrulamanın başarılı olduğuna dair bir mesaj gönderir. Parolaların eşleşmediği durumda ise; kimlik doğrulaması başarısız olduğunu istemci bilgisayara geri döndürür ve kullanıcı tekrar sisteme giriş yapmak istediğinde bu işlemler en baştan tekrarlanır.

Kimlik doğrulama başarılı olduğu takdirde; kullanıcı video konferansa giriş yapar ve benzer şekilde sunucu bilgisayardan yapmak istediği işlemler için isteklerde bulunabilir.

3.4. Fi6en Paket ve Sınıf Hiyerarşisi

Fi6en Sunucu (Red5) Uygulaması

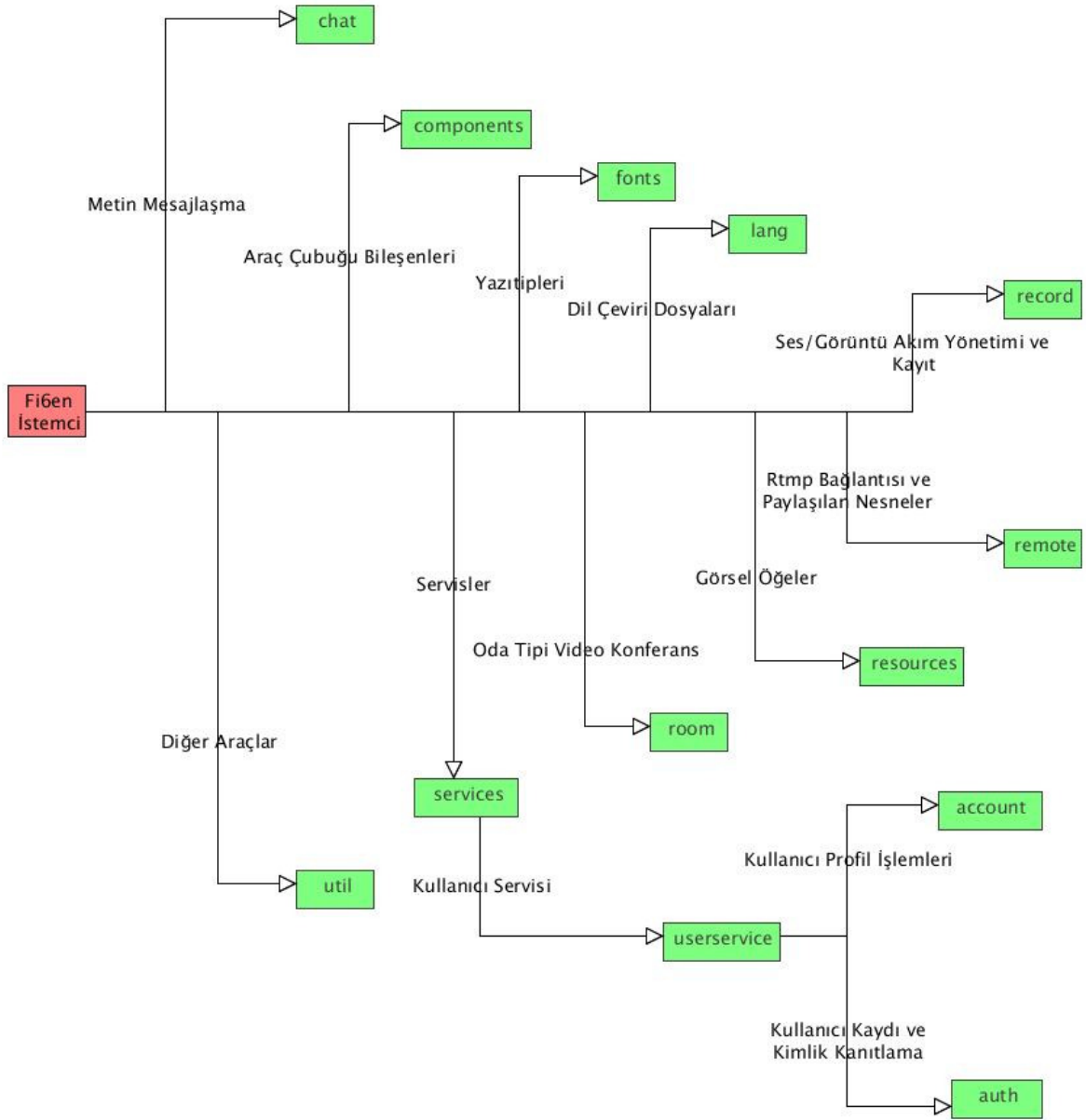
Fi6en sunucu uygulaması JAVA programlama dili kullanılarak yazılmıştır. JAVA dilinde kaynak dosya yolları paketler şeklinde ifade edilmektedir. Şekil 17.'de Fi6en sunucu kaynak kodundaki her bir paketin ne tür işler için özelleştirildiği ve hangi sınıf ve arayüzleri içerdiği görülmektedir.



Şekil 17. Fi6en sunucu uygulaması paket ve sınıf hiyerarşisi.

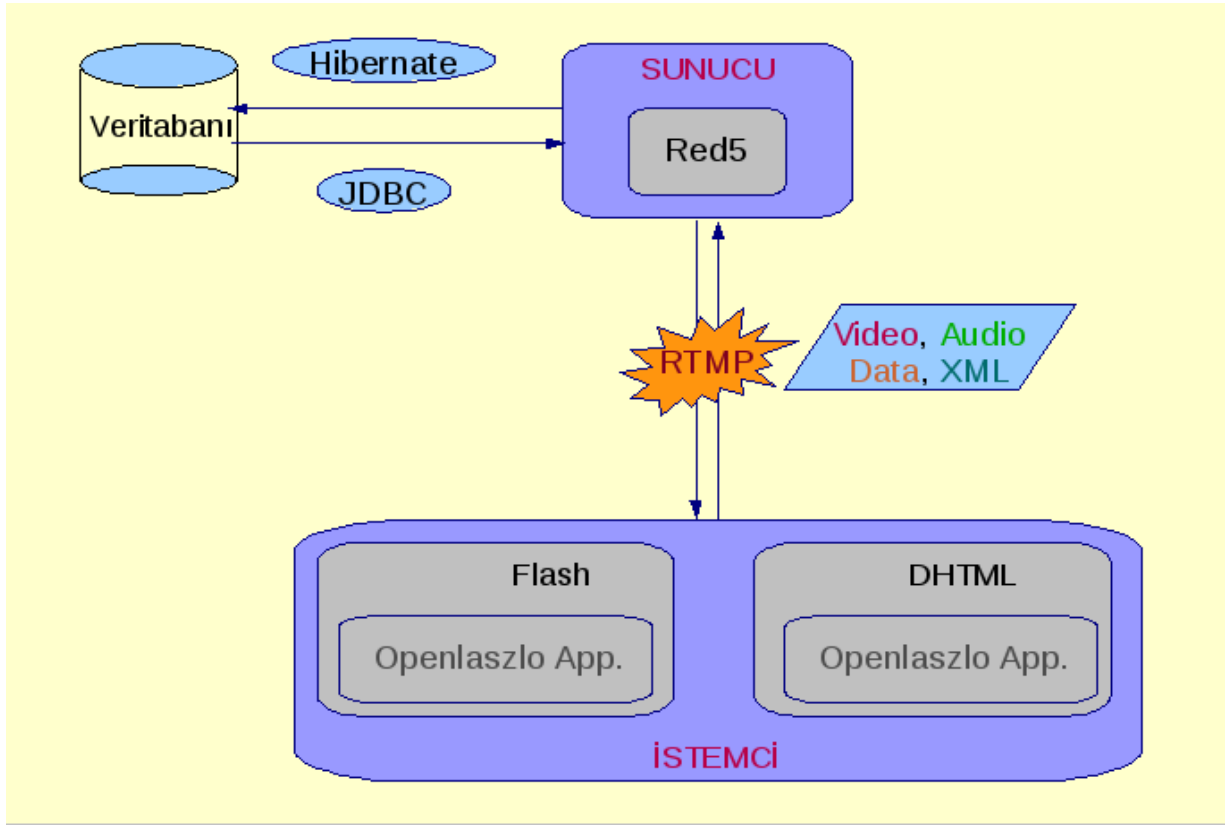
Fi6en İstemci (Openlaszlo) Uygulaması

İstemci uygulamasında kaynak dosyaları, sunucu uygulamasında olduğu gibi paketler şeklinde ifade edilmemektedir. İstemci tarafındaki sınıf gerçekleştirmelerinde, sunucu tarafındakinden farklı olarak, hangi pakete ait olduğu konusunda bir belirtim yoktur. Şekil 18.'de Fi6en istemci uygulaması kaynak kodu dizinleri ve bu dizinler içerisindeki sınıfların hangi işlemleri yerine getirdiği görülmektedir.



Şekil 18. Fi6en istemci uygulaması dizin hiyerarşisi.

3.5. Kullanılan Teknoloji ve Araçlar



Şekil 19. Fi6en çalışma yapısı.

3.5.1. Openlaszlo

FI6EN'in istemci uygulamasının oluşturulmasında kullanılacak olan zengin internet uygulaması (RIA) platformudur. İşletim sisteminden bağımsız olarak çalışan web tarayıcı tabanlı uygulamalar (browser-based) geliştirmede kullanılmaktadır.

RIA: Yeni nesil ağ uygulamalarından biridir. RIA uygulamalarının öne çıkan karakteristiklerinden bazıları şu şekildedir:

- Kolay masaüstü uygulaması yapısı
- En iyi kullanıcı arayüzü
- Hızlı ve düşük maliyetli yayınlama
- Gerçek zamanlı interaktif multimedya iletişimi

(Chang-Gen ve ark., 2009)

OpenLaszlo, Adobe Flex'in muadili olan, Common Public License lisansına sahip, RIA geliştirilmesinde kullanılan, açık kaynak bir platformdur.

Openlaszlo platformu OpenLaszlo sunucusundan (lps) ve LZX programlama dilinden oluşur.

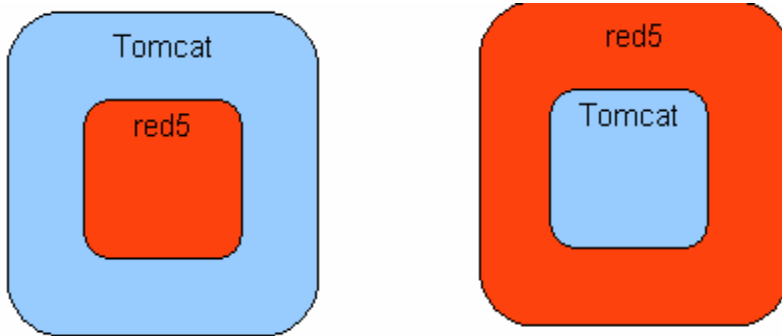
OpenLaszo sunucusu LZX dilinde yazılmış olan kodları derleyerek, ufak web formatı (SWF) ya da dinamik HTML (DHTML) olarak çalıştırılabilir hale dönüştürür. OpenLaszlo sunucusu aslında bir Java Servlet'tir. LZX ise XML ve Javascript'den oluşan bir dildir (Klein ve ark., 2008).

FiBen istemci uygulamasının çalıştırılabilir hâli SWF formatındadır ve uygulamasının çalıştırılabilmesi için kullanıcı çalışma ortamında bir Flash oynatıcısının kurulu olması gerekir. Flash oynatıcısı başlı başına bir uygulama veya tarayıcı üzerinde eklenti olarak kullanılabilir.

3.5.2. Red5

Red5, Java ile yazılmış açık kaynak kodlu bir Flash sunucusudur ve şunları destekler:

- Ses/Görüntü akışı (FLV ve MP3)
- İstemci yayınlarının kaydedilmesi (sadece FLV)
- Paylaşılan Nesneler
- Canlı akış yayınlama
- Dağıtık uygulamalar



Şekil 20. Red5 & servlet taşıyıcı.

Red5 bir Java Platform Kurumsal Sürüm (J2EE) taşıyıcı (Tomcat veya Jetty) içine web uygulama arşivi (WAR) olarak gömülebilir(solda) veya bir sistem servisi olarak çalışıp kendi içinde bir servlet taşıyıcı içerebilir (sağda). (Gong ve ark., 12 Nisan 2010)

FiBen sunucu uygulaması, Red5 kullanılarak geliştirilmiştir.

3.5.3. Spring

Spring Model-Görünüm-Denetleyici (MVC) web çatısı ile web tabanlı programlar hazırlamak mümkündür. Spring çatısı programcı için birçok işlemi basitleştirir ve kullanımını kolaylaştırır.

Spring üç ayaktan oluşur:

1) Spring basit ve sadeleştirilmiş bir uygulama programlama arabirimi (API) sunarak, birçok açık kaynak kodlu ürünün kullanımını ve entegrasyonunu kolaylaştırır. Son yıllarda Java platformunun (Java SE, Java EE) kullanımı, sunmuş olduğu detaylı API'lerden dolayı zorlaşmıştır. Spring kendi API'leri ile bu sorunu çözer.

2) Spring içinde bağımlılıkların enjekte edilmesi (Dependency Injection) metodu kullanılarak, nesnelere arası bağlar XML ayar dosyaları üzerinden otomatik olarak gerçekleştirilir. Örneğin ClassA isimindeki bir sınıf ClassB tipinde bir değişkene sahip ise, bu bağımlılık ClassA sınıfından bir nesne oluşturulurken Spring tarafından göz önünde bulundurulur. Spring otomatik olarak ClassB sınıfından bir nesne oluşturarak, ClassA sınıfından oluşturduğu nesneye enjekte eder. İki sınıf arasındaki bağ böylece Spring tarafından Dependency Injection metodu ile oluşturulmuş olur.

3) Spring yaklaşım tabanlı programlama (AOP) tarzı program yazılımını destekler. Genelde hareket yönetimi, günlükleme güvenlik gibi program parçaları kodun birçok yerinde kullanılır. AOP ile, aslında program mantığının bir parçası olmak zorunda olmayan bu metot ve modüller merkezi bir yerde toplanarak, programdan bağımsız bir şekilde gerçekleştirimi yapılır.

.....

Spring Kullanımının Avantajları

Spring, değişik bileşenlerin entegrasyonu için gerekli altyapı hizmetlerini sunar. Spring kullanımı sonucu elde edilen avantajları şu şekilde sıralayabiliriz:

- Basit Java sınıfları (POJO) kullanılarak yazılım yapılabilir. Bu sınıfların test edilmesi, hata ayıklama yöntemiyle hataların aranması ve kurulumu (deployment) daha kolaydır. Kurumsal Java Bean (EJB) teknolojisinde olduğu gibi bir EJB içerik sunucusuna ihtiyaç duyulmamaktadır.
- Spring Model-Görünüm-Denetleyici (MVC) ile web tabanlı programlar oluşturmak mümkündür. Web uygulamalarda bildirimsel hareketler

(declarative transaction) yönetimi, güvenlik ve bilgi bankası bağlantısı gibi temel altyapı ihtiyaçları Spring ve Spring MVC tarafından sağlanır.

- Spring modüller bir yapıya sahiptir. Proje gerekleri doğrultusunda belirli Spring modülleri kullanılabilir.
- Spring ile ayar dosyalarında yapılacak yapılandırmalar ile otomatik olarak tekil (Singleton) nesnel oluşturabilir. Bunun için özel Singleton sınıfların oluşturulması gereği ortadan kalkar.
- Spring arayüz sınıfları için değişik türde gerçekleştirim yapılmasını ve kullanım esnasında ayar dosyaları üzerinden gerçekleştirim sınıf seçimini kolaylaştırır. Bu açıdan bakıldığında tasarım şablonlarının kullanımı Spring aracılığıyla daha kolay hale gelmektedir.
- Sunduğu Servlet Mock sınıfları ile, Spring MVC ile oluşturulan web tabanlı programların Tomcat gibi bir Servlet taşıyıcısından bağımsız olarak test edilmesini kolaylaştırır (Acar, 2009).

Geliştirilen yazılımın istemci tarafından sunucu tarafındaki servislere erişimi için tanımlı olan Spring bean'leri kullanılarak gerçekleştirilmiştir. Spring bean'lerinin tanımlanması ise sunucu uygulamasındaki red5-web.xml dosyasında yapılmaktadır.

3.5.4. Hibernate

Hibernate 2002 yılında başlatılmış Java sistemler için açık kaynak kodlu bir projedir. Java platformunda yazılmış bir nesne ilişkisel eşleme (ORM) aracıdır. ORM; nesne odaklı (object oriented) dillerdeki nesnelere, ilişkisel veritabanlarındaki (relational databases) kayıtlara nasıl karşılık geldiğini yürüten bir teknolojidir.

Hibernate gibi ORM araçlarıyla, bir nesneyi veritabanına kaydetmek, yeni halini güncellemek ve sorgulama yapmak düz yapısal sorgulama dili (SQL) bağlantılarına göre çok daha kolaydır.

Örneğin, Java Veritabanı Bağlantısı (JDBC) ile veritabanına bir kayıt eklemek için şuna benzer bir kod yazılır:

```
stmt.executeUpdate( "INSERT INTO person VALUES ('Cem', "Sonmez",  
"cemosonmez@gmail.com")");
```

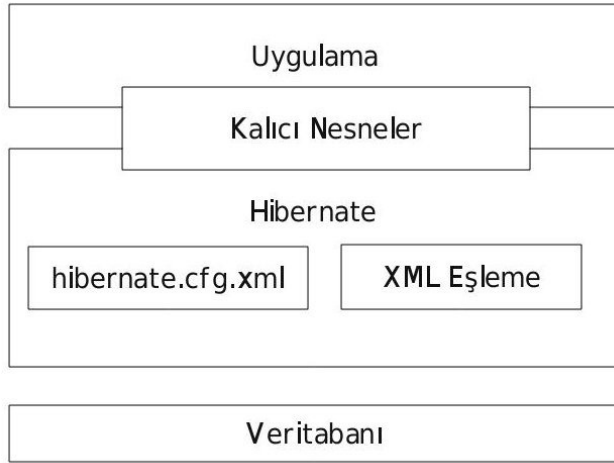
Burada, "person" tablosuna bir kayıt eklenmektedir. Böyle bir işlemi Hibernate ile yapmak için ise şu şekilde bir kullanım yeterli olacaktır:

```
session.save(person);
```

person, bildiğimiz düz bir java nesnesidir (POJO).

Hibernate gibi ORM araçlarının en önemli faydası, kod yazımını kısaltmak veya kolaylaştırmaktan öte, yazılım bakımını kolaylaştırmasıdır. Veritabanı temelli uygulamalarda, kodun 1/3'ü veritabanı erişimine yöneliktir. Veritabanındaki bir kolonunun tipinin değişmesi, yeni bir kolon eklenmesi gibi değişiklikler, bütün veri erişim kodunu tekrar gözden geçirmeyi gerektirir. Hibernate ile bu gözden geçirmeden çok yüksek oranda tasarruf edilir. Hibernate kullanılan yazılımlarda, veritabanındaki değişikliklerde yapılması gereken sadece nesnelere tabloların birbirine nasıl eşleştirildiğinin (mapping) gözden geçirilmesidir. (Anonim, 2010f)

Hibernate yalnızca Java sınıfları ile ilişkisel veritabanı tabloları arasındaki eşleştirmeden sorumlu olmayıp, ayrıca veri sorgulaması ve getirilmesi imkânlarını da sunmaktadır.



Şekil 21. Hibernate mimarisi.

(Xia ve ark., 2009)

Şekil 21.'de yer alan bileşenler:

- *Uygulama:* Veritabanı işlemlerinde veritabanı sunucusu ile iletişim kuracak olan web uygulamasıdır. Fi6en üzerinde ise bu bileşeni Fi6en Red5 sunucu uygulaması temsil etmektedir.
- *Kalıcı Nesnelere:* Veritabanı tablolarındaki satırların Uygulama üzerinde tanımlanmış sınıf türünden nesnelere şeklinde ifade edilmiş hâlidir. Fi6en üzerinde, User sınıfı bu bileşene örnek olarak verilebilir. User sınıfındaki her bir değişken, “fi6en” isimli veritabanı üzerindeki “users” tablosundaki her bir sütun ile eşleştirilmiştir.

- *Hibernate*: Uygulama ile veritabanı arasında köprü görevi gören teknolojidir. Veritabanı bağlantı işlemlerinin tümü ve uygulamada bulunan nesnelere ile veritabanı tablolarının eşleştirilmesi Hibernate üzerinde yapılmaktadır. Hibernate sınıf ve metodlarını Java Web uygulamalarında kullanabilmek için *hibernate3.jar* arşivini proje sınıf yoluna eklemek gerekir.
- *hibernate.cfg.xml*: Veritabanı bağlantı işlemlerinin yapıldığı Hibernate ayar dosyasıdır. Bu dosya FiBen sunucu uygulaması üzerinde *WEB-INF/classes* yolunda yer almaktadır.
- *XML Eşleme*: Nesnelere ile veritabanı tablolarının eşleştirilmesi işlemi bu kısımda yapılmaktadır. Bunun için iki seçenek bulunmaktadır: “*SınıfAdi.hbm.xml*” eşleme dosyası veya Hibernate Annotation kullanmaktır. Örneğin; “Kişi” sınıfına ait bir nesne ile “kisi” isimli veritabanı tablosu için eşleme dosyası aşağıdaki gibi olabilir:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate Mapping DTD
3.0//EN" "http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">

<hibernate-mapping>
  <class name="com.vizyon.demo.entity.Kisi" table="kisi">
    <id name="id" column="id">
      <generator class="increment"/>
    </id>

    <property name="adi" type="string" column="adi" length="25" />
    <property name="soyadi" type="string" column="soyadi" length="25" />
    <property name="evTelefonu" type="string" column="ev_telefonu" length="15"
/>
    <property name="cepTelefonu" type="string" column="cep_telefonu"
length="15" />
    <property name="mail" type="string" column="mail" length="50" />

  </class>
</hibernate-mapping>
```

(Örs, 2008)

Hibernate Annotation kullanımında ise, eşleme dosyası gibi ayrıca bir dosyaya ihtiyaç yoktur. Sınıf değişkenleri ile tablo sütunlarının eşleştirilmesi sınıf gerçekleştirimi içerisinde yapılmaktadır. Örneğin; “Uye” sınıfının “Uye” isimli veritabanı tablosu ile eşleştirilmesi Hibernate Annotation kullanılarak aşağıdaki gibi yapılabilmektedir:

```
package com.vizyon.arge.tirtil.entity;
import javax.persistence.Column;
```

```
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity
@Table(name="Uye", catalog="tirtil")
public class Uye {
    private int id;
    private String adi;
    private String mail;
    @Id
    @GeneratedValue(strategy=GenerationType.IDENTITY)
    @Column(name="id", nullable=false)
    public int getId() {
        return id;
    }
    public void setId(int id) {
        this.id = id;
    }
    @Column(name="adi", nullable=false , length= 30)
    public String getAdi() {
        return adi;
    }
    public void setAdi(String adi) {
        this.adi = adi;
    }
    @Column(name="e_mail", nullable=false, length= 100)
    public String getMail() {
        return mail;
    }
    public void setMail(String mail) {
```

```
this.mail = mail;  
}
```

Fi6en’de veritabanı işlemlerindeki “XML Eşleme” kısmında Hibernate Annotation’lar kullanılmaktadır. Yukarıdaki örneğe benzer şekilde; Fi6en’de “User” sınıfı veritabanı üzerindeki “users” tablo sütunları ile eşleştirilmiştir.

3.5.5. Genişletilebilir İşaretleme Dili (XML)

Hem insanlar hem bilgi işlem sistemleri tarafından kolayca okunabilecek dokümanlar oluşturmaya yarayan, W3C tarafından tanımlanmış bir standarttır. Bu özelliği ile veri saklamanın yanında farklı sistemler arasında veri alışverişi yapmaya yarayan bir ara format görevi de görür. SGML’in basitleştirilmiş bir alt kümesidir.

XML’in tasarımcısı, HTML’i de tasarlamış olan Tim Berners Lee'dir.

Günümüzde birçok yazılım, diğer yazılımlarla veri alışverişini XML formatı üzerinden yapmaktadır. Ayrıca XML’i esas format olarak kullanan uygulamalara, veri tabanlarına rastlamak mümkündür.

XML Dosyalarını İşleme Yöntemleri

XML için Basit Uygulama Programlama Arabirimi (SAX)

Olay tabanlı, sözcüksel işleme. Dosyada içerisindeki her düğüm bir geri-besleme (callback) fonksiyonu aracılığı ile istemci koda yansıtılır. Bu yapısı nedeniyle oldukça hızlı ve etkilidir, ancak XML dosyasından rastgele düğüm ulaşımı oldukça zordur; Hedef düğüme ulaşmak için, her defasında dosyayı en başından işlemek gerekmektedir.

Belge Nesne Modeli (DOM)

Bu teknik tamamen arayüz yönelimlidir. Her düğüm ve parçalarına arayüz aracılığıyla ulaşılır. Rastgele ulaşımında etkili olmasına rağmen, büyük boyutlu dosyalar söz konusu olduğunda oldukça hantaldır ve hafıza tüketiminde fazla talepkârdır. (Anonim, 2010c)

Fi6en istemci uygulaması üzerindeki “Dil Modülü” ve “Ses/Görüntü Akım” modüllerinde XML kullanılmaktadır. Dil modülünde kelime ve cümlelerin çevirileri XML yapısında tutulmaktadır. Ses/Görüntü akım modülünde ise; istemci uygulaması kayıtlı videoların listesine erişmek istediğinde sunucudan gelecek olan veri XML yapısındadır.

3.5.6. Nesne İlişkisel Eşleme (ORM)

Genel olarak, bilgisayar yazılımlarında ilişkisel veritabanlarındaki verilerin nesneye yönelik dillerdeki nesnelere ilişkilendirilmesine dayanan arabulucu bir programlama

teknigidir. ORM yaklaşımı ilk olarak Hibernate tarafından gerçekleştirilmiştir. (Xia ve ark., 2009)

Tabloları sınıflara, satırları nesnelere, kolonları değişkenlere eşleme işlemidir. Sınıf metotları tablo seviyesinde işlemleri, nesne metotları ise satır seviyesinde işlemleri yapmaktadır.

ORM yaklaşımı, FiBen Hibernate işlemlerinin alt yapısında kullanılmaktadır.

3.5.7. Gerçek Zamanlı Mesajlaşma Protokolü (RTMP)

Adobe Systems tarafından geliştirilen, bir Flash sunucusu ve istemcisi arasında ses, video ve verinin internet üzerinden iletiminde kullanılan protokoldür.

3 türü bulunur :

- 1) *RTMP*: 1935 no'lu portu kullanan ve TCP'nin en üstünde çalışan basit protokol
- 2) *Tünellenmiş Gerçek Zamanlı Mesajlaşma Protokolü (RTMPT)*: RTMP'nin, güvenlik duvarından geçiş için HTTP istekleri şeklindeki hâlidir (8088 no'lu port)
- 3) *Güvenli Gerçek Zamanlı Mesajlaşma Protokolü (RTMPS)*: RTMP'nin HTTPS' bağlantısı şeklinde çalışan halidir (443 no'lu port)

(Gong ve ark., 12 Nisan 2010)

Not: Red5 RTMPS'i 8443'ten sunuyor ancak, flash öntanımlı olarak 443'ten rtmps'yi denemektedir. Bu nedenle Red5 tarafında RTMPS portu \$red5-home/conf/red.properties içerisinde 443 olarak değiştirilmeli ya da flash tarafında bağlantı URL'sine RTMPS için kullanılacak olan port 8443 olarak belirtilmelidir.

FiBen istemci ve sunucu uygulaması birbirleri ile RTMP protokolünü kullanarak haberleşmektedir. Bu iki uygulama arasındaki mesajlaşmalar ve ses/görüntü trafiği RTMP protokolü üzerinden yürütülmektedir.

3.5.8. FiBen Ses ve Görüntü Kodlayıcıları

Kodlayıcı–Kodçözücü (Coder/Decoder): Ses ve görüntü akımlarının gerçek zamanda dijital olarak sıkıştırılmasına dayanan teknolojidir. Donanımın ya da yazılımın yapmış olduğu bu sıkıştırmaya ise *kodlama (codec)* denir.

FiBen uygulamasının çalışması esnasında eşzamanlı sesli/görüntülü iletişimde ses ve görüntünün sıkıştırılması ve açılması için kullanılan kodlayıcı-kodçözücüler şu şekildedir:

Ses Kodlayıcısı: Nellymoser (Flash Player 6 ve daha sonraki sürümleri tarafından desteklenmektedir).

Video Kodlayıcısı: Sorenson Spark (Flash Player 6 ve daha sonraki sürümleri tarafından desteklenmektedir).

3.6. Hata Raporlama ve Yapılacaklar Listesi (todo) Yönetimi

Hata raporlama ve yapılacaklar listesi yönetimi için “Trac” yazılımının etiket sistemi kullanılmaktadır. Eclipse geliştirme ortamının ilgili Trac eklentisi kurularak hata raporlama ve yapılacaklar listesi yönetiminin geliştirme ortamıyla tümleşik olması sağlanmıştır.

3.7. Java ve Ipv6

J2SDK/JRE 1.4 sürümü ile birlikte, Java Ağ uygulamalarına IPv6 desteği eklenmiştir. Ipv6'nın diğer özellikleri olan tünelleme, adres otomatik yapılandırması, gezici IP vs. gibi seçenekler, JAVA API seviyesinde desteklenmemektedir. Bu özellikler, otomatik olarak işletim sistemi altında ele alınmaktadır.

Linux'ta 2.1.12 çekirdeğinden sonra Ipv6 desteği eklenmiştir.

3.7.1. Java Uygulama Etkisi

- IPv6 için, Java uygulama kodlarında bir değişiklik yapılmasına gerek yoktur
- Tüm adres ya da soket tip bilgileri Java ağ API'sinde kapsüllenmiştir.
- Sistem özelliklerini değiştirme yolu ile adres ve/ya soket tiplerinde ayarlamalar yapılabilir.
- Ipv6'ya özel yeni uygulamalar için yeni sınıf ve API'ler kullanılabilir.

3.7.2. İletişim Senaryoları

Tablo 1. IPv6 iletişim senaryoları

(Nodes)	V4 Only	V4/V6	V6 Only
V4 Only	⊗	⊗	
V4/V6	⊗	⊗	⊗
V6 Only		⊗	⊗

Tablo 1.'de bulunan, soldaki ve yukarıdaki sütunlar, birbirleriyle iletişim kurmaya çalışan düğüm yapılarını temsil etmektedir. X karakteri ise, bu iki düğümün birbiri ile haberleşebileceğini gösterir. Örneğin ikinci satırın birinci sütunu ele alınacak olursa; sol kısımda bulunan makine (V4/V6) IPv4 ve IPv6 adresine sahiptir. Böylelikle bu makine, IPv4 veya IPv6 adresine sahip başka makineler ile haberleşebilecektir. Üst kısımdaki makine (V4 Only) ise sadece IPv4 adresine sahiptir. Bunun anlamı ise; bu makine sadece IPv4 adresini kullanarak diğer makinelerle iletişim kurabilecektir. Bu bilgilerden yola çıkılarak şu sonuca varılabilir: Sol kısımda bulunan makine (V4/V6) üst kısımdaki makine (V4 only) birbirleri ile IPv4 protokolünü kullanarak iletişim kurabilirler.

3.7.3. IPv6 Ağ Özellikleri ve JVM Parametreleri

java.net.preferIPv4Stack (default: false)

Eğer IPv6 desteği işletim sisteminde mevcut ise, altta bulunan temel soket bir IPv6 soketi olacaktır. Bununla birlikte, Java (tm) uygulamaları hem Ipv4 hem de IPv6 konaklarından gelen bağlantıları kabul edecektir.

Eğer bir uygulamanın, sadece Ipv4 soketlerini kullanacağı ayarlanacak ise, bu özellik “true” yapılır. Bu durumda uygulama IPv6 konakları ile iletişim kuramayacaktır.

java.net.preferIPv6Addresses (default: false)

Eğer IPv6 desteği işletim sisteminde mevcut ise, öntanımlı ayar IPv6 üzerinden Ipv4 adresini kabul etmek olacaktır. Bu durum geriye doğru uyum amacı ile yapılmaktadır. Örneğin; sadece Ipv4 servisine erişimine bağımlı olan uygulamalar ya da “%d.%d.%d.%d” biçiminde temsil edilen bir IP adresine bağımlı olan uygulamalar için. Bu özellik, IPv6 servislerine bağlanması beklenen uygulamanın test edilmesi ve yayımlanmasına imkân sunar (Anonim, 2010a).

Fi6en sunucu uygulamasında, Red5 sunucu başlatma betiğinde JAVA parametreleri şu şekilde eklenmiştir:

`-Djava.net.preferIPv4Stack=false -Djava.net.preferIPv6Addresses=true`

Parametreler bu şekilde belirtildiği için; Fi6en sunucusu istemciden gelen bağlantı isteklerinde, eğer istemci makinesi bir IPv6 adresine sahipse, bağlantı için IPv6 adresini kullanacaktır. Eğer istemci makinesi IPv6 adresine sahip değilse sunucu ve istemci uygulaması birbirleri ile IPv4 adresini kullanarak iletişim kuracaktır.

IPv4 veya IPv6 bağlantıları, daha önce bahsedilmiş olan RTMP protokolü üzerinden gerçekleştirilmektedir.

3.8. Video Konferans Kalite Ölçütleri

Video konferans kullanımı sırasında video kalitesine etki eden üç faktör vardır :

- 1) *Frame Rate*: Bir saniye içerisinde oynatılan çerçeve (frame) sayısıdır. Video için fps (frame per seconds) olarak ifade edilir.
- 2) *Frame Resolution*: Bir çerçeve içerisindeki piksel sayısıdır. Yüksek frame resolution'a sahip olan çerçevelerde daha çok detay bulunur.
- 3) *Frame Quantization*: JPEG, DV gibi blok tabanlı sıkıştırma algoritmalarında kullanılan ayırık kosinüs değişimlerinin quantization faktörünü ifade eder.

Rader ve ark. (2006) yaptığı çalışmalar sonucunda; video kalitesine etki eden en önemli faktörün frame rate olduğunu öne sürmüştür.

Sonuçlar:

- Video konferans yazılımını aynı anda 1-2 kişi kullandığı zaman, orta düzey bantgenişliği kabul edilebilir bir kalite sunar.
- Sadece surat bölgesinin olduğu görüntülerde düşük bantgenişliği kullanılabilir.
- Yüksek bantgenişliği, daha çok akışın çok olduğu hareketli videolarda kullanışlı hâle geliyor.
- Yüksek Bantgenişliği= Yüksek Kalite
Düşük Bantgenişliği= Düşük Kalite (Rader ve ark., 2006)

BÖLÜM 4

GERÇEKLEŞTİRİM

4.1. Günlükleme Desteği

FiBen günlükleme desteği sayesinde sunucu tarafında oluşan olayları izlemek mümkün kılınmıştır. Sunucunun hizmet verdiği sırada oluşabilecek herhangi bir aksaklığın nedeninin belirlenmesi ve çözüm üretilmesi konusunda günlükleme kullanımı oldukça etkili bir yöntemdir. Günlükleme kullanımı ile birlikte, oluşan hataların izlenebilmesinin yanı sıra, geliştirme sürecinde program akışı da izlenebilmektedir.

Günlükleme (Logging) sistemi için SLF4J (Simple Logging Facade for Java) çatısı kullanılmaktadır. Bu çatı, Log4J ve Apache Commons günlükleme işlemlerinin her ikisinin de özelliklerini bir araya getiren bir sistemdir. Red5'in bağımlılıklarında Apache Commons ve Log4J'in her ikisi de kullanılmıştır. SLF4J ise, farklı çatılar için bir vekil görevi görür ve bir günlükleme gerçekleştirimini sunar.

Günlükleme gerçekleştirimi için, Log4J ve Apache commons'un etrafından dolaşabilmek için logback tercih edilmiştir. Logback'i kullanabilmek için build path'e eklenmesi gereken kütüphaneler şunlardır:

slf4j-api: The core API

logback-core: Current Logback core library

logback-classic: Logback support library

log4j-over-slf4j: Log4j proxy/bridge

jcl-over-slf4j: Apache commons logging proxy/bridge

jul-to-slf4j: java.util.logging proxy/bridge

Proxy/bridge olarak kullanılan kütüphanelerin (*log4j-over-slf4j*, *jcl-over-slf4j*) görevi; ilgili gerçekleştirimde olan günlükleme çağrılarını dinlemek ve bunların etrafından dolanıp SLF4J'i kullanmaktır. Logback, Log4J'in varisidir, Log4J ve SLF4J'in yaratıcısı tarafından ortaya çıkarılmıştır.

FiBen logging için neler yapıldığı konusuna değinilecek olursa; *log4j*'nin kullanılmasını etkisiz hale getirmek için *web.xml* dosyasından aşağıdaki satırlar çıkartılır.


```
<context-param>
  <param-name>log4jConfigLocation</param-name>
  <param-value>/WEB-INF/log4j.properties</param-value>
</context-param>
```

Bunun yerine aşağıdaki satırlar web.xml dosyasına yazılır:

```
<listener>
  <listener-class>org.red5.logging.ContextLoggingListener</listener-class>
</listener>
<filter>
  <filter-name>LoggerContextFilter</filter-name>
  <filter-class>org.red5.logging.LoggerContextFilter</filter-class>
</filter>
<filter-mapping>
  <filter-name>LoggerContextFilter</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>
```

Bu kısımda dinleyici sınıf tanımlamaları yapılmıştır.

Günlükleme dosyasının tanımlanması için WEB-INF/classes içerisinde "logback-fi6en.xml" isimli dosya oluşturulmuştur. Bu dosya içerisine aşağıdaki satırlar eklenmiştir.

```
<configuration>
  <contextName>fi6en</contextName>
  <jmxConfigurator contextName="fi6en" />
  <appender name="FI6ENLOG"
class="ch.qos.logback.core.FileAppender">
    <File>log/fi6en.log</File>
    <Append>>false</Append>
    <Encoding>UTF-8</Encoding>
    <BufferedIO>>false</BufferedIO>
    <ImmediateFlush>>true</ImmediateFlush>
```

```

        <layout class="ch.qos.logback.classic.PatternLayout">
            <Pattern>
                %-4relative [%thread] %-5level %logger{35} - %msg%n
            </Pattern>
        </layout>
    </appender>
</root>
    <appender-ref ref="FI6ENLOG" />
</root>
<logger name="org.red5.core">
    <level value="INFO" />
</logger>
</configuration>

```

Bu dosya kaydedilecek günlük dosyasının sahip olacağı özellikleri tanımlıyor. (nereye kaydedileceği, günlükte yer alan mesajlarda hangi özelliklerin yer alacağı vs.)

Bu işlemler yapıldıktan sonra, red5 sunucusunu yeniden başlatınca \$red5-home/log dizini içerisinde fi6en.log isiminde bir dosyanın oluştuğunu ve eğer herhangi bir günlükleme işlemi yaptırıldı ise bu mesajların o dosyaya kaydedildiği görülecektir.

Günlüklemenin Gerçekleştirimi?

UserService sınıfını için yapılan günlüklemeye bakılacak olursa; öncelikle

```
import org.slf4j.Logger;
```

```
import org.red5.logging.Red5LoggerFactory; sınıfları dahil edilir.
```

ve

Logger log= Red5LoggerFactory.getLogger(UserService.class,"fi6en"); şeklinde nesne oluşturulur. Burada "fi6en", o bağlam için günlükleme yapılacağını göstermektedir. Daha sonra şu şekilde günlük mesajı yazılabilir:

```
log.info("UserService constructor ");
```

4.2. Fi6en Mesajlaşma Modülü

Mesajlaşma desteğinin gerçekleştirilmesi için; Fi6en sunucu uygulaması tarafında, tüm istemciler tarafından görülebilecek bir paylaşılan nesne (shared object) yaratılması gerekir. Oluşturulan paylaşılan nesne sayesinde Red5 uygulamasına bağlanan tüm istemciler aynı şeyi görüyor olacaktır.

Sunucu tarafı

Sunucu tarafında paylaşılan nesne şu şekilde oluşturulmaktadır:

```

public boolean connect (IConnection conn, IScope scope, Object []params) {
    log.info("Connected to server application - Client : "+
conn.getRemoteAddress());
    appScope= scope;
    createSharedObject(appScope, "chat", true);
    sharedObjectChat = getSharedObject(appScope, "chat");
    ISharedObjectListener listenerSOChat = new SharedObjectListener();
    sharedObjectChat.addSharedObjectListener(listenerSOChat);
    return true;
}

```

Yukarıda belirtildiği gibi; paylaşılan nesnenin (shared object) yaratılması işlemi *connect* methodunda yapılmaktadır. İstemci uygulamaya bağlandığı zaman "chat" isminde bir paylaşılan nesne yaratılıyor. Daha sonra, *ISharedObjectListener* interface'inin gerçekleştirimini yapan *MyCustomListener* sınıfından bir nesne yaratılıyor ve bu nesneyi yaratılmış olan paylaşılan nesneye dinleyici olarak ekleniyor. Böylelikle "chat" paylaşılan nesnesi üzerinde yapılan tüm işlemleri dinleyip, günlükleme yapılabilmektedir.

İstemci tarafı

Paylaşılan nesne ile ilgili olarak, istemci tarafında *sharedObject.lzx* sınıfında temel bir paylaşılan nesnede olması gereken özellik ve metotlar vardır. (paylaşılan nesneye bağlan, paylaşılan nesneden bir değer al, paylaşılan nesneyi sil, temizle vs.). *sharedObjectChat.lzx* sınıfı ise *sharedObject* sınıfını miras alıyor ve bu sınıfta da chat paylaşılan nesnesi ile ilgili olarak yapılabilecek işlemler yer alıyor.

Bahsi geçen *sharedObjectChat.lzx* kaynak dosyasının bir bölümü şu şekildedir:

```

<library>
    <class name="sharedObjectChat" extends="sharedObject">
        <method name="writeMessage" args="username, msg"><![CDATA[
            varmessageArea=
            chatScreen.mainChatView.chat.messageAreaView.messageArea;

```

```

        messageArea.addText("<b><font color='${ color }>"+username + ' :
'+"/font></b>"+msg+"\n");
    ]]></method>
    <method name="writeSharedObject" args="message">
        this.send("writeMessage",loginScreen.getUsername(),message);
    </method>
    <!--this method writes the messages to the debug window which will be
shown to all users-->
    <method name="writeInfo" args="message">
        Debug.info("%w",message);
    </method>
</class>
</library>

```

4.3. Fi6en Kimlik Kanıtlayıcı Modülü

Bu kısımda kimlik kanıtlayıcı ile giriş desteği eklenirken hangi sınıfların kullanıldığından bahsedilecektir. Kimlik kanıtlayıcı modülünün eklenmesi ile birlikte, katılımcıların video konferansa giriş yapabilmeleri için giriş ekranında istenilen kullanıcı adı ve parolasını doğru bir şekilde girmeleri şart koşulur. Böylelikle kimlik kanıtlayıcı başarılı olan bir katılımcı video konferans sistemine giriş yapabilecek, başarısız olan bir kullanıcının ise sisteme girişine izin verilmeyecektir.

4.3.1. Kullanılan Sınıflar ve Ayar Dosyaları

İstemci Tarafta (Openlaszlo)

(services/userservice/auth yolunda)

loginWindow.lzx: Uygulama başladığı zaman kullanıcıyı karşılayan penceredir. Bu pencerede “giriş” ve “kayıt” gibi düğmeler bulunur. Sınıfın temel olarak yaptığı iş: kayıt işlemi için bir kayıt penceresi açmak ve giriş işlemi için ise girilen kullanıcı adını ve parolanın şifrelenmiş halini veritabanındaki kayıtlar ile karşılaştırma yapması için sunucu tarafına göndermektir.

registerNewUser.lzx: Kayıt işlemi için kullanıcının karşısına çıkan penceredir. Kayıt olacak kullanıcı için gerekli bilgiler bu pencereden alınmakta ve kayıt düğmesine tıkladığı zaman, girilen değerlerin kontrolü yapıldıktan sonra (bkz: “validation.lzx”) kayıt

için uygun ya da uygun olmadığı belirtilmektedir. Bu sınıfa ait bir nesne “loginWindow” sınıfı içerisinde yaratılmaktadır.

validation.lzx: Kayıt penceresindeki değerlerin formata uygunluğu ile ilgili değerlendirmeyi yapan sınıftır. Bu işlemler: tüm alanların dolu olması, email formatı, parolaların eşleşmesi gibi kontrollerdir.

md5.js: Giriş penceresinde, giriş düğmesine tıklandığında ve kayıt sırasında kullanıcı bilgileri veritabanına kaydedilirken, kullanıcının parolası sunucuya şifrelenerek gönderilmektedir. Şifreleme için “loginWindow” ve “registerNewUser” sınıflarının kullandığı script ise “md5.js” javascript dosyasıdır. Girilen stringi (fi6en için parola) md5 algoritmasına göre şifreler ve şifrelenmiş halini geri döndürür.

Sunucu Tarafı (Red5)

(org/red5/fi6en/userservice yolunda)

DatabaseOperation.java: Kayıt için gelen parametreleri veritabanına kaydetme ve giriş sırasında kullanıcı adı-parolaların karşılaştırılması gibi metotları içeren sınıftır.

User.java: Veritabanında users tablosunu, java tarafında nesne olarak temsil edecek olan sınıftır. Sütunlara karşılık gelen kullanıcı değişkenleri bu sınıfta bulunur.

(org/red5/fi6en/userservice yolunda)

HibernateUtil.java: Veritabanı bağlantısı için hibernate oturumu oluşturan sınıftır. (org/red5/fi6en/util yolunda)

hibernate.cfg.xml: HibertnateUtil sınıfının kullanacağı, hibernate veritabanı bağlantısı için kullanılacak olan ayar dosyasıdır. veritabanı sunucusu türü, kullanıcı adı, parola vs. bilgiler bu dosya içerisinde bulunur.(WebContent/WEB-INF yolunda)

red5-web.xml: İstemci tarafından kullanıcı kaydı, giriş gibi işlemler için çağrılacak olan metotlara sahip sınıflar, service olarak bu ayar dosyasında belirtilir. Örneğin;

```
<bean name="databaseservice.service"
class="org.red5.fi6en.userservice.DatabaseOperation"
singleton="true" />
```

4.4. Fi6en Çoklu Dil Desteği

Çoklu dil desteğinin eklenmesi için, ilk olarak dil dosyaları oluşturulmuştur. Dil dosyaları, kelime ve cümlelerin orjinal halinin kullanılacak dildeki çevirilerini içerir. Uygulamanın dil dosyalarını sorunsuz işleyebilmesi için dil dosyalarının tümü aynı yapıda

olmalıdır. Dil dosyalarının bir standart oluşturması için bu dosyalar XML yapısında oluşturulmuştur.

Örneğin:

turkish.xml

```
<language>
  <string id="1" name="submit">
    <value>Gönder</value>
  </string>
  <string id="2" name="login">
    <value>Giriş</value>
  </string>
  ...
</language>
```

english.xml

```
<language>
  <string id="1" name="submit">
    <value>Submit</value>
  </string>
  <string id="2" name="login">
    <value>Log in</value>
  </string>
  ...
</language>
```

Dil dosyalarındaki stringlerin ayırt edilmesi id değerine göre yapılır.

(Bu dosyalar client/lang/ yolundadır)

Mevcut olan bu dil dosyalarını kullanabilmek için ise, istemci (openlaszlo) tarafında bu XML'i işleyebilecek bir yapı kullanılmıştır. XML üzerinde işlem yapabilmek için kullanılan Openlaszlo etiketi dataset'tir. dataset etiketi XML kaynağını aşağıdaki şekilde alır.

```
<dataset name="dsetLang" id="dsetLang" src="lang/turkish.xml">
</dataset>
```

languageSelector.lzx: Bu sınıf combobox'tan türetilmiştir. Bu sınıf sayesinde combobox'taki seçili dile ait olan dil dosyası, uygulamanın çalışma anında dinamik olarak değiştirilebilmektedir. (client/util yolunda)

Örneğin metni İngilizce “submit” - Türkçe “gönder” olan, id'si 1 olan bir button şu şekilde oluşturulur:

```
<button name="loginButton"
datapath="dsetLang:/language/string[ @id='1']"
text="$path{'value/text()}'">
</button>
```

Buradaki \$path{'value/text()}' ifadesi ile birlikte, uygulama sırasında dil dosyası kaynağı değişse bile, o id'ye sahip olan değer button'un text'i içerisine gömülmüş durumdadır.

4.5. Fi6en Oda Tipi Video Konferans Modülü

Oda tipi video konferans, normal video konferansın dışında, herhangi bir kullanıcı tarafından oluşturulan ve grup olarak özel görüşmelerin yapılabileceği, ana uygulama üzerindeki sanal bölümlerdir. Oda'ya katılan her kullanıcı, sadece o oda içerisindeki değişikliklerden haberdar olabilir. Örneğin; sadece odadaki kullanıcılarla mesajlaşabilme, sesli/görüntülü iletişim, kullanıcı listesinden haberdar olmak vs. mümkündür.

Oda modülünün eklenmesi için, sunucu (fi6en red5 uygulaması) ve istemci (fi6en istemci uygulaması) taraflarında birtakım değişiklikler yapılmıştır. Bu iki kısımda yapılan değişiklikler:

Sunucu tarafı

Oda aslında, ana red5 uygulaması faaliyet alanını (scope) üzerinde başka bir faaliyet alanının oluşturulup kullanılmasıdır.

Örneğin:

Ana uygulama URL'si:

rtmp://127.0.0.1:1935/fi6en

ise; bu uygulama üzerindeki "room" isimli odanın URL'si

rtmp://127.0.0.1:1935/fi6en/room şeklinde olacaktır.

Red5 uygulama kodu üzerinde yapılan değişikliklerde ise; “ApplicationAdapter” sınıfından türetilen “Application” isimli sınıf açıklaması içerisinde, ana sınıfa ait room* isimli metotların geçersiz kılınması (overriding) gereklidir.

roomStart: İlgili odanın oluşturulması esnasında çağrılacak olan metottur. Kullanıcı fi6en üzerinde "room" isimli bir odaya bağlanmak isterse, ilk olarak “roomStart” metodu çağrılır ve faaliyet alanının (scope) adı "room" olarak belirlenir. Bu metot içerisinde, oluşturulacak her oda için kullanılacak ve bu odaya katılan her kullanıcı tarafından ortak olarak görülecek olan "roomChat" (mesajlaşma için) ve "roomUserlist" (oda içerisindeki kullanıcı listesi için) paylaşılan nesnelere (shared object) oluşturulur.

roomStop: Odanın yok edilmesi esnasında çağrılacak olan metottur. Odaya bağlı kullanıcılardan sonuncusunun da odadan ayrılması ile birlikte bu metodun çağırımı gerçekleştirilir. Bu metodun gerçekleştirilmesinde, “roomStart” içerisinde yaratılan paylaşılan nesnelere (shared object) yok edilmesi işlemi gerçekleştirilmektedir.

roomJoin: Daha önce oluşturulmuş olan bir odaya herhangi bir kullanıcı bağlandığı zaman çağrılacak olan metottur. Bu metodun gerçekleştirilmesinde, “roomStart” içerisinde oluşturulmuş olan paylaşılan nesnelere (shared object) var olup olmadığının kontrolü yapılır. Eğer yok ise yeni bir tane oluşturulur, var ise bir işlem yapılmaz.

roomLeave: Odadan herhangi bir kullanıcının ayrılması esnasında çağrılacak olan metottur.

roomConnect: Oda ile bağlantının kurulması esnasında çağırımı yapılacak olan metottur.

roomDisconnect: Oda ile bağlantının kesilmesi esnasında çağrılacak olan metottur.

Yukarıda bahsedilen metotların tümü içerisinde günlükleme (logging) yapılarak, gerçekleşen tüm durumların izlenmesi mümkün hale getirilmiştir.

İstemci tarafı

Fi6en istemci tarafında yapılacak olan işlemler; kullanıcının oda oluşturabilmesi, odaya giriş yapabilmesi ve oda içerisinde ilgili işlemlerinin sağlanmasıdır. Bu işlemlerin yapılabilmesi için ilk olarak, oda işlemleri için sunucu tarafı ile gereken bağlantının kurulması arka tarafta gerçekleştirilmelidir. İstemci tarafında, oda modülünün eklenmesi için eklenen sınıflar şunlardır:

roomButton.lzx: Kullanıcının odaya giriş yapılabilmesine görsel olarak imkan sağlayacak olan sınıftır.

roomView.lzx: Oda ekran sınıfıdır. Kullanıcı odaya giriş yaptığında, uygulama ile etkileşimde bulunacağı görsel bileşenleri içerisinde barındıran sınıftır.

sharedObjectChatRoom.lzx: Oda içerisinde mesajlaşmaya imkan sağlayacak olan, sunucu tarafında oluşturulmuş olan "roomChat" isimli paylaşılan nesneye bağlantıyı sağlayan sınıftır.

sharedObjectRoomList.lzx: Lobi üzerinde, herhangi bir kullanıcı oda yarattığı zaman, ilgili odanın yaratıldığından diğer kullanıcıları haberdar edecek olan paylaşılan nesne sınıfıdır.

4.6. Fi6en Kayıt Modülü

Kayıt modülü kullanımı ile birlikte kullanıcı, kamera görüntüsünü daha sonra görüntüleyebileceği şekilde kaydedebilmektedir. Bu modül sayesinde kamera görüntüsü sunucu tarafındaki red5 uygulamasındaki "streams" isimli bir dizin içerisinde .flv uzantılı olarak kaydedilmektedir. Bu modülün fi6en'e eklenmesi sırasında sunucu ve istemci tarafında bir dizi işlem yapılmıştır. Bunlar:

Sunucu tarafı

İstemci uygulamasından sunucuya kamera görüntüsünün kaydedilmesi üzerine bir istek geldiği zaman, sunucunun yapması gereken; bu istek ile birlikte gelen parametreleri almak, ilgili ekran görüntüsünün akımına bağlantı kurup ve kayıt için tekil bir isim tanımlayıp, dosya sistemine kaydetmek olacaktır.

StreamManager.java

Bahsedilen kamera görüntüsü ile bağlantı kuran ve .flv (flash video) uzantılı bir dosya şeklinde kayıt işlemini gerçekleştiren sınıftır.

public void recordShow(String username): Kayıt isteğinde bulunan kullanıcının kullanıcı adını alır ve mevcut zaman değerini de `username+ "_" + String.valueOf(System.currentTimeMillis())` şeklinde ekleyerek, kayıt için tekil bir isim oluşturur. Daha sonra ilgili kamera görüntüsü akımını alır ve belirlemiş olduğu ismi kullanarak kayıt işlemini gerçekleştirir.

public void stopRecordingShow(String username): Kaydı durdurma isteğinde bulunan kullanıcının bu isteğini ve kullanıcı adı parametresini alır ve mevcut kayıt işlemini sonlandırır. Kullanıcı adı'nın gönderilmesinin nedeni; her kullanıcının kamera görüntüsünün akımının ismi kullanıcı adı'nın değeri olarak yayınlanmasıdır.

İstemci Tarafı

Kullanıcıdan kamera görüntüsünün kaydedilmesi isteğini alma, bu isteğini fi6en red5 sunucu uygulamasına bildirme ve oluşan durumlardan kullanıcıyı haberdar edilmesi gibi işlemler bu tarafta gerçekleştirilir. Kayıtın başlatılması, durdurulması vs. gibi kullanıcı ile etkileşimde kullanılacak görsel bileşenler de bu kısımda bulunmaktadır.

publisher.lzx

Bu sınıf içerisinde kamera görüntüsünün gösterileceği bir video alanı, "kaydet" ve "kayı durdur" düğmeleri bulunur. Kullanıcı kamera görüntüsünü paylaşmak istediği zaman, kamera görüntüsü video alanı üzerinde gösterilir ve "kaydet", "kayı durdur" düğmeleri görünür kılınır. Daha sonra, kullanıcı eğer kamera görüntüsünü paylaşmak isterse, ilgili düğmelere tıklar ve istemci uygulaması da yukarıda bahsedilen, sunucu tarafındaki kayıt ile ilgili olan işlemler için ilgili metot çağırımını gerçekleştirir.

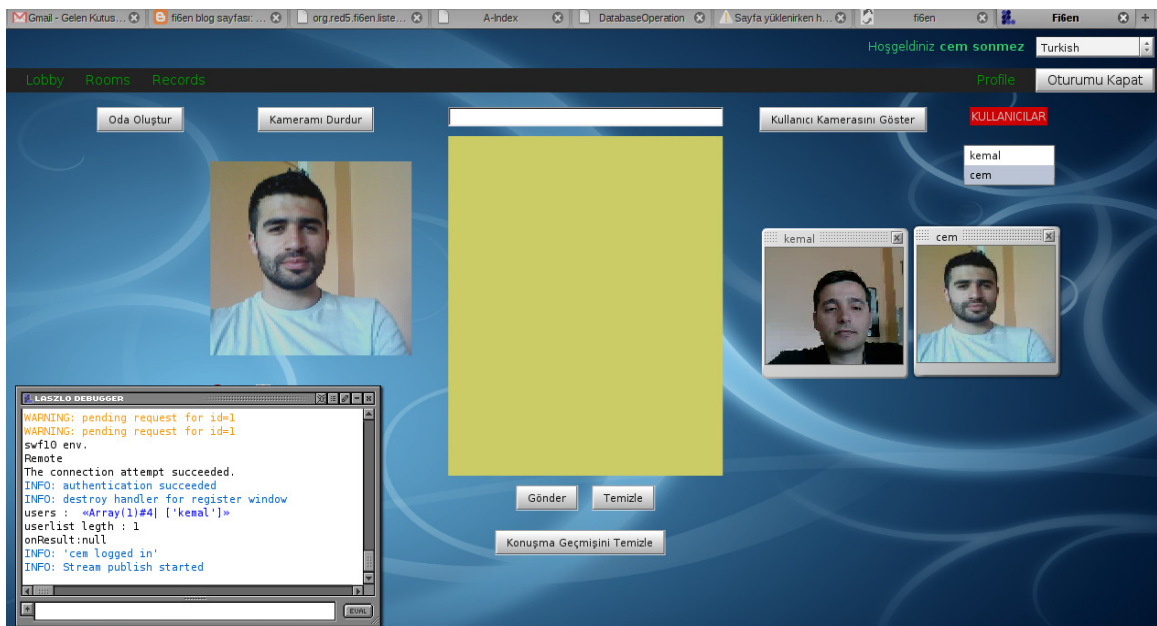
4.7. Demo Uygulaması

Yazılımın başarımının test edilmesi için güncel hali bir web sunucu bilgisayarda yayınlanmaktadır

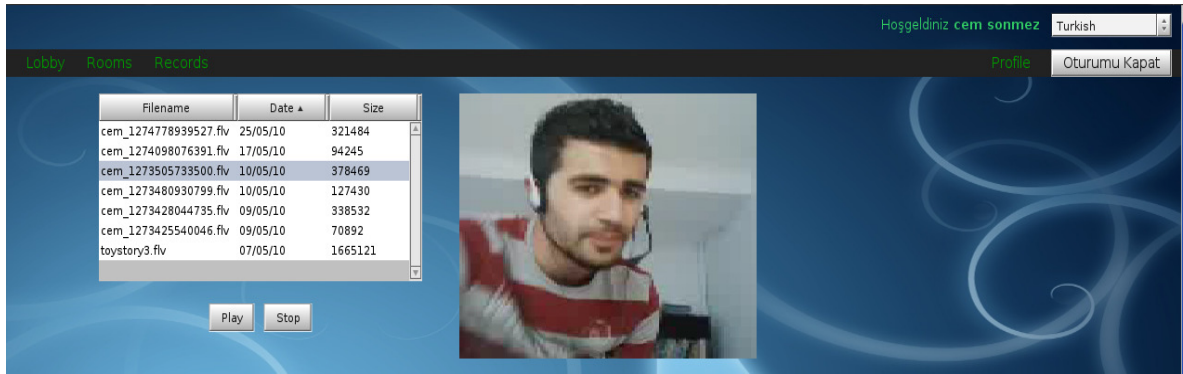
<http://fi6en.comu.edu.tr>

4.8. Fi6en (Free Ipv6 Video Conference) Kullanım Kılavuzu

Bu bölümde Fi6en'in nasıl kullanılacağı ile ilgili bilgileri içeren kullanım kılavuzu yer almaktadır. Kullanıcının Fi6en'i kullanımı sırasında yazılıma yabancılığını en aza indirmek adına bu kılavuz hazırlanmıştır.



Şekil 22. Fi6en ana ekranı.



Şekil 23. Fi6en kayıt listesi ekranı.

- Kullanıcı uygulama arayüzünü farklı bir dilde görüntülemek isterse; sağ üst kısımda bulunan açılır listeden dil seçimini yapar.
- Giriş ekranında (ilk ekran), “Kaydol” düğmesine tıklanır.
- Kayıt penceresinde kullanıcı, kendisi ile ilgili olan tüm bilgileri doğru ve eksiksiz bir biçimde doldurur. Bu sırada kendisine ait bir kullanıcı adı ve parola belirler.
- Kayıt başarılı ise; "Laszlo Debugger" ekranında "Username is available" şeklinde bir mesaj görünür, kayıt gerçekleştirilir ve otomatik olarak giriş tekrar geri dönülür.
- Kaydın başarılı olmaması ya da yanlış bilgi girilmesi durumunda oluşacak olan hata mesajına göre, bilgiler doğru bir şekilde girilerek kayıt girişimi tekrarlanır.
- Kullanıcı, kayıt ekranında kendisi için belirlemiş olduğu kullanıcı adı ve parolasını giriş ekranında girer ve video konferansa giriş yapar.
- Kullanıcı video konferanstaki tüm kullanıcılar ile mesajlaşmak isterse; ekranın ortasında bulunan metin kutusuna mesajını yazar ve aşağıda bulunan "Gönder" düğmesine tıklar. Böylelikle göndermiş olduğu mesaj, metin kutusunun altındaki metin alanında tüm kullanıcılar tarafından görünür kılınır.
- Kullanıcı kamera görüntüsünü paylaşmak isterse; mesajlaşma alanının sol tarafında bulunan "Kameramı Göster" isimli düğmeye tıklar. Böylelikle kullanıcı, kamera görüntüsünün video konferansa katılan tüm kullanıcılar tarafından görülebilmesine izin vermiş olur.
- Kullanıcı herhangi bir kullanıcının paylaşmış olduğu kamera görüntüsünü görüntülemek isterse; ekranın sağ tarafında bulunan “KULLANICILAR” listesinden, görüntülemek istediği kullanıcıyı seçer ve mesajlaşma alanının sağ tarafındaki “Kullanıcı Kamerasını Göster” düğmesine tıklar.

- Kullanıcı ses ve görüntüsünü kaydetmek isterse; daha önce nasıl yapılacağı belirtildiği şekilde kamera görüntüsünü paylaşır ve kamera görüntüsü altındaki “kaydet” (kırmızı yuvarlak düğme) tıklayarak kaydı başlatır. Kaydı durdurmak istediğinde ise kaydet; düğmesinin yanındaki “kayı durdur” (sarı düğme) tıklar.
- Kullanıcı kayıtlı görüntüleri oynatmak isterse; üst kısımdaki menüyü kullanarak “Records” sayfasına geçiş yapar. Bu sayfada (Şekil 23.), daha önce kaydedilen kayıtların bir listesi görüntülenir. Kullanıcı oynatmak istediği kaydı listeden seçer ve alt kısımdaki “Play” düğmesine tıklayarak seçili kaydı oynatma işlemini gerçekleştirir.
- Kullanıcı özel görüşmelerin yapılabileceği “oda” yaratmak isterse; ana ekranın (Şekil 22.) sol tarafında bulunan “Oda Oluştur” düğmesinin üzerine fare imlecini getirir, açılan kutucuğa oluşturmak istediği odanın ismini yazar ve düğmeye tıklar.
- Kullanıcı daha önce yaratılan bir odada görüşme yapmak istediğinde; “Oda Oluştur” düğmesinin altında yer alan odalardan, giriş yapmak istediğine tıklar ve odaya giriş yapar. Oda içerisinde yapılan mesajlaşmalar, sadece o oda içerisinde bulunan kullanıcılar tarafından görülebilmektedir.
- Kullanıcı tekrar ana ekrana geri dönmek istediğinde; üst kısımdaki menüden “Lobby”e tıklar ve geçiş yapar.
- Kullanıcı konferanstan güvenli çıkış yapmak isterse; Üst kısımdaki menu çubuğunun sağ tarafında bulunan “Oturumu Kapat” düğmesine tıklar.

BÖLÜM 5**TARTIŞMA VE SONUC**

Günümüzde insanlar toplantılarını ve iş görüşmelerini olabildiğince seyahat etmeden, bilgisayar başında yapmaya gayret etmektedir. Bu ihtiyaçtan dolayı, farklı noktalarda bulunan kişilerin zaman, yolculuk vb. masraflarından kaçınarak yüz yüze iletişim kurmasında, iş, eğitim ve sağlık görüşmelerinde vs. kullanılacak olan bir video konferans yazılımı, kullanıcıların işlerini oldukça kolaylaştıracaktır.

Bilindiği gibi ağdaki cihazların büyük bir kısmı IPv4 protokolünü kullanarak internete çıkıyorlar. Zamanla internetin büyüme hızı oldukça arttı ve ileride adresleme sıkıntısının ortaya çıkacağı aşikârdır. Öngörülen bu adres sıkıntısının ve Ipv4'ün bazı konularda yetersiz kalmasından dolayı, yeni bir internet protokolüne ihtiyaç duyulmuştur. Bu ihtiyaçtan dolayı 90'lı yılların başında yeni nesil IP protokolü olan IPv6 protokolü geliştirilmeye başlanmıştır.

Sonuç olarak; ortaya çıkan bu video konferans yazılımı ihtiyacından ve IPv6'nın beraberinde getirmiş olduğu avantajların kullanımının faydalı olabileceği düşünüldüğünden, IPv6 desteği veren, Türkçe, açık kaynak kodlu olan bir video konferans yazılımı (Fi6en) geliştirilmektedir.

Bölüm 3.1. FI6EN Hedef Ve Gereksinimleri'nde belirtilmiş olan yol haritasına bağlı kalarak, şu anki durumda Fi6en 0.5 sürümüne ulaşmış durumdadır ve şu özelliklere sahiptir:

Fi6en Özellikleri

- Sesli/Görüntülü iletişim
- Anında mesajlaşma
- Ses ve görüntünün kaydı
- Kayıtlı videoları oynatma
- Oda tipi video konferans
- Çoklu dil desteği (Türkçe, İngilizce)
- IPv6 bağlantı desteği

Bundan sonra yapılacak çalışmalar ise, IPv6 protokolünün kullanımının video konferans kalitesi ve başarımı üzerindeki etkisinin araştırılmasına yönelik testlerin yapılması ve Fi6en'in çok daha etkin bir video konferans yazılım ürünü hâline getirilmesine yönelik olmalıdır.

KAYNAKLAR

- Chang-Gen J., Ming-He H. ve Jie T., 2009. Design and implementation of video conference system based on Flex and J2EE, *IASP 2009. International Conference on*, 184-186.
- Deering S. ve Hinden R., 1998. Internet Protocol, Version 6 (IPv6) Specification - RFC 2460, Retrieved January 10, 2010, from <http://www.ietf.org/rfc/rfc2460.txt>.
- Rader E.J., Hofer E. ve Finholt T., 2006. High-Bandwidth Video Conferencing Systems: When Is the Quality Worth the Cost?.
- Chuan L.L., Jumari K. ve İsmail M., 2002. Implementation of video conferencing over IPv6 on the Linux platform, *SCORED 2002. Student Conference on*, 375-377.
- Geer D., 2005. Developments advance Web conferencing, *Computer*, 38 (2):14-17.
- Yu B. ve Rui Y., July 2008. Towards good web conferencing system design for ubi-media collaboration, *2008 First IEEE International Conference on*, 213-218.
- Zhu J., 2010. Study on E-learning Education Model Based on Web Intelligence, *IC4E '10. International Conference on*, 554-557.
- Klein N., Carlson M. ve MacEwen G., 2008. The Basics, *Laszlo in Action*, Manning Publication Co., Greenwich. 1-5.
- Gong S., Gregorie P. ve Rossi D.,(n.d.). *Red5 - Reference Documentation Version 0.8*, Retrieved April 12, 2010, from <http://red5.org/wiki/Documentation/UsersReferenceManual>.
- Xia C., Yu G. ve Tang M., 2009. Efficient Implement of ORM (Object/Relational Mapping) Use in J2EE Framework: Hibernate, *CiSE 2009. International Conference on*, 1-3.
- Acar Ö., (2009). *Spring Framework*. 10 Nisan 2010, <http://www.kurumsaljava.com/2009/03/11/spring-framework/>.
- Bieringer P., (2009). *Linux IPv6 HOWTO*, Retrieved March 21, 2010, from <http://www.tldp.org/HOWTO/Linux+IPv6-HOWTO/>.
- Örs K., (2008). *Hibernate'e Giriş*, 25 Aralık 2009, http://www.ceviz.net/programlama/hibernatee-giris_a1215.html
- Anonim, 2009a. <http://code.google.com/p/openmeetings/>.
- Anonim, 2009b. <http://technet.microsoft.com/en-us/library/bb457173.aspx>.
- Anonim, 2009c. <http://www.camfrog.com/>.
- Anonim, 2009d. <http://www.dimdim.com/>.

Anonim, 2009e. <http://bigbluebutton.org/>.

Anonim, 2010a. http://java.sun.com/j2se/1.4.2/docs/guide/net/ipv6_guide/index.html.

Anonim, 2010b. <http://www.unimedya.net.tr/videokonferans.aspx>.

Anonim, 2010c.

http://tr.wikipedia.org/wiki/Geni%C5%9Fletilebilir_i%C5%9Faretleme_dili.

Anonim, 2010d. <http://trac.red5.org/wiki/Documentation/Tutorials/LoggingSetup>.

Anonim, 2010e. <http://www.gnu.org/licenses/gpl-3.0.html>

Anonim, 2010f. <http://tr.wikipedia.org/wiki/Hibernate>

TABLÖLAR LİSTESİ

Sayfa No:

Tablo 1. IPv6 iletişim senaryoları	27
--	----

ŞEKİLLER LİSTESİ

Sayfa No:

Şekil 1. Video konferans yazılımı kullanım gelirinin dağılımı.....	3
Şekil 2. Openmeetings ekran görüntüsü.....	4
Şekil 3. Ipv4 ağı için (a) Kaynak iletim hızı ve (b) Çerçeve hızı.	8
Şekil 4. Ipv6 ağı için (a) Kaynak iletim hızı ve (b) Çerçeve hızı.	8
Şekil 5. Fi6en modül yapısı.	10
Şekil 6. Fi6en yol haritası - sürüm 0.1.....	10
Şekil 7. Fi6en yol haritası - sürüm 0.2.....	11
Şekil 8. Fi6en yol haritası - sürüm 0.3.....	11
Şekil 9. Fi6en yol haritası - sürüm 0.4.....	11
Şekil 10. Fi6en yol haritası - sürüm 0.5.....	11
Şekil 11. Fi6en yol haritası - sürüm 0.6.....	12
Şekil 12. Fi6en yol haritası - sürüm 0.7.....	12
Şekil 13. Fi6en yol haritası - sürüm 0.8.....	13
Şekil 14. Fi6en yol haritası - sürüm 0.9.....	13
Şekil 15. Fi6en yol haritası - sürüm 1.0.....	13
Şekil 16. Sunucu/istemci modeli.....	14
Şekil 17. Fi6en sunucu uygulaması paket ve sınıf hiyerarşisi.	16
Şekil 18. Fi6en istemci uygulaması izin hiyerarşisi.	17
Şekil 19. Fi6en çalışma yapısı.	18
Şekil 20. Red5 & servlet taşıyıcı.....	19
Şekil 21. Hibernate mimarisi.	22
Şekil 22. Fi6en ana ekranı.	40
Şekil 23. Fi6en kayıt listesi ekranı.	41

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı: Cem SÖNMEZ

Doğum Yeri: Gebze

Doğum Tarihi: 01.10.1985

EĞİTİM DURUMU

Lisans Öğrenimi: ÇOMÜ, Bilgisayar Mühendisliği

Yüksek Lisans Öğrenimi: ÇOMÜ, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği

Bildiği Yabancı Diller: İngilizce

BİLİMSEL FAALİYETLERİ

a) Yayınlar -SCI -Diğer: ---

b) Bildiriler-Uluslararası-Ulusal: ---

c) Katıldığı Projeler: TUBİTAK tarafından desteklenen 108G102 no'lu “Ulusal IPv6 Protokol Altyapısı Tasarımı ve Geçiş Projesi”

İŞ DENEYİMİ

Çalıştığı Kurumlar ve Yıl: ÇOMÜ, Bilgi İşlem Daire Başkanlığı, Bilgisayar Mühendisi (2010-Bugün)

İLETİŞİM

E-posta Adresi: cem@comu.edu.tr

cemosonmez@gmail.com