T.C.
Radyo ve Televizyon Üst Kurulu

LINUX/UNIX TABANLI AÇIK KAYNAK YAZILIMLARA GEÇİŞ SÜREÇLERİ VE RTÜK İÇİN GÖÇ MODELİ

UZMANLIK TEZİ

Sefa YILMAZ

ANKARA
HAZİRAN / 2021

T.C.
Radyo ve Televizyon Üst Kurulu

LINUX/UNIX TABANLI AÇIK KAYNAK YAZILIMLARA GEÇİŞ SÜREÇLERİ VE RTÜK İÇİN GÖÇ MODELİ

UZMANLIK TEZİ

Sefa YILMAZ

DANIŞMAN
Üst Kurul Uzmanı
Fatih ÖZTÜRK

ANKARA
HAZİRAN / 2021

Radyo ve Televizyon Üst Kurulu Başkanlığına

Bu çalışma, Tez Değerlendirme Komisyonu tarafından oy birliği / oy çokluğu ile Uzmanlık Tezi olarak kabul edilmiştir.

		Adı Soyadı	İmza
Başkan	:	……	……
Üye	:	……	……
Üye	:	……	……
Üye	:	……	……
Üye	:	……	……

ONAY
… /… /2021

Ebubekir ŞAHİN
Üst Kurul Başkanı

TEZ TESLİM TUTANAĞI VE DOĞRULUK BEYANI

Radyo ve Televizyon Üst Kurulunda görev yapan Üst Kurul Uzman Yardımcısı olarak, Radyo ve Televizyon Üst Kurulu Tez Hazırlama Yönergesi'ne uygun şekilde hazırlamış olduğum uzmanlık tezi ilişikte sunulmuştur. Bu uzmanlık tezindeki bütün bilgileri, akademik kurallara ve etik davranış ilkelerine uygun olarak toplayıp sunduğumu; ayrıca, bu kural ve ilkelerin gereği olarak, çalışmada bana ait olmayan tüm veri, düşünce ve sonuçları andığımı ve kaynağını gösterdiğimi beyan ederim.

Bilgilerinizi ve gereğini arz ederim. … /… /2021

Sefa YILMAZ
Üst Kurul Uzman Yardımcısı

Uzmanlık Tezinin Adı:
LINUX/UNIX Tabanlı Açık Kaynak Yazılımlara Geçiş Süreçleri ve RTÜK İçin Göç Modeli

EK:
- Tez (6 Adet)
- CD (1 Adet)

Tezi Teslim Alan

Daire Başkanı
… /… /2021

İÇİNDEKİLER

İÇİNDEKİLER I
KISALTMALAR V
TABLOLAR DİZİNİ VII
ŞEKİLLER DİZİNİ VIII
GİRİŞ 1
BİRİNCİ BÖLÜM 6
ÖZGÜR / AÇIK KAYNAK YAZILIM 6
1.1. ÖZGÜR / AÇIK KAYNAK TANIMI VE GELİŞİMİ 6
1.1.1. UNIX'in Geliştirilmesi 7
1.1.2. BSD UNIX 10
1.1.3. NetBSD, FreeBSD, OpenBSD 12
1.1.4. BSD Lisans Süreci 14
1.1.5. Özgür Yazılım 15
1.1.6. GNU Projesi 17
1.1.7. Özgür Yazılım Vakfı 18
1.1.8. Linux 'un Geliştirilmesi 19
1.1.9. Açık Kaynak Yazılım Hareketi 21
1.1.10. Özgür ve Açık Kaynak Yazılım Konsepti (ÖAKY) 22
1.2. ÖAK OLMAYAN YAZILIMLAR VE EULA LİSANSLARI 23
1.3. ÖZGÜR VE AÇIK KAYNAK LİSANSLAR 26
1.3.1. GNU Lisansları ve Copyleft 28
1.3.2. BSD Lisansları 31
1.3.3. Mozilla Kamu Lisansı 33
İKİNCİ BÖLÜM 34
ÖZGÜR / AÇIK KAYNAK YAZILIMLARA GÖÇ 34
2.1. DÜNYADA ÖAK YAZILIMLARIN KURUMSAL KULLANIMI 34
2.1.1. Avrupa 'da ÖAK Yazılımların Kurumsal Kullanımı 34

2.1.2. Avrupa Dışında ÖAK Yazılımların Kurumsal Kullanımı 38
2.1.3. Dünyada Kurumsal Açık Kaynak Girişimlerinin Başarı Durumu 40
2.2. ÖZGÜR VE AÇIK KAYNAK YAZILIMLARA GÖÇ 43
2.3. ÖAK YAZILIMLARA GÖÇÜN AVANTAJLARI 44
2.3.1. Ekonomik Fayda ve Düşük Maliyet 44
2.3.2. Güvenlik 45
2.3.3. Kararlılık ve Güvenirlilik 46
2.3.4. Ücretsiz Destek ve Hızlı Debug 47
2.3.5. Açık Standartlar ve Daha Az Satıcı Bağımlılığı 47
2.3.6. Yerel Yazılım Kültürünün Desteklenmesi 50
2.4. AÇIK KAYNAK GÖÇÜNÜN RİSK ANALİZİ 51
2.4.1. Kullanılabilirlik 51
2.4.2. Dokümantasyon ve Eğitim 53
2.4.3. Mevcut Ticari/Tescilli Yazılımlarla Çalışabilirlik 54
2.4.4. Bakım ve Destek Süreçleri 54
2.4.5. Sistemin Devamlılığı Riski 55
2.4.6. Daha Az Rekabet 57
2.4.7. Ek Maliyet Yükü 58
2.4.8. Teknolojik Beklentiyi Karşılamama ve İşleyişin Aksaması 59
2.4.9. Güvenlik Açıklarının Oluşması 60
2.5. AÇIK KAYNAK GÖÇÜNÜN FİNANSAL ANALİZİ 60
2.5.1. Ekonomik Fayda Örnekleri 63
2.5.2. Ekonomik Zarar Örnekleri 65
2.5.2.1. IRS Açık Kaynak Göçü 65
2.5.2.2. Münih Açık Kaynak Göçü 66
2.6. AÇIK KAYNAK GÖÇÜNÜN TEKNİK ANALİZİ 67
2.6.1. Göç Süreç Hızı Modelleri 67
2.6.1.1. Big Bang Geçiş 68
2.6.1.2. Kademeli Geçiş 68
2.6.1.3. Kullanıcı Bazlı Geçiş 69
2.6.2. Göç Kapsam Modelleri 69
2.6.2.1. Kısmi / Hibrit Göç Modeli 70

2.6.2.2. Tam Göç Modeli 70
2.6.3. Göç Entegrasyon Araçları 71
2.6.3.1. SAMBA 71
2.6.3.2. NSS ve PAM 72
2.6.3.3. OpenLDAP 73
2.6.3.4. GNU/Linux SMBFS 73
2.6.4. Açık Kaynak Göç Projeksiyonu 74
2.6.5. Uygulama Göçü 75
2.6.5.1. Açık Kaynak Eşdeğeri Olan Uygulamalar 75
2.6.5.2. Açık Kaynak Platformlarda Çalışabilecek Uygulamalar 75
2.6.5.3. Uzaktan Erişim Yöntemi 76
2.6.5.4. Uygulama Sanallaştırma Yöntemi 76
2.6.5.5. Uygulamayı ÖAK Platformda Yeniden Derleme 77
2.6.6. Alternatif Göç Teknikleri 77
2.7. TÜRKİYE'DE ÖAK YAZILIMLARA KURUMSAL İLGİ VE BAŞARILI GÖÇ OPERASYONLARI 79
2.7.1. Millî Savunma Bakanlığı 80
2.7.2. Afet ve Acil Durum Yönetimi Başkanlığı (AFAD) 81
2.7.3. Radyo ve Televizyon Üst Kurulu (RTÜK) 81
2.7.3.1. Sayısal Kayıt Arşiv ve Analiz Sistemi (SKAAS) 82
2.7.4. Belediyeler 85

ÜÇÜNCÜ BÖLÜM 86
RTÜK GÖÇ MODELİ 86
3.1. MEVCUT ALTYAPI 86
3.2. RTÜK GÖÇ MODELİ TABLOSU 87
3.3. ALTERNATİF SİSTEM, YAZILIM VE UYGULAMA TABLOSU 88
3.3.1. Kimlik Doğrulama ve DNS Hizmetleri İçin FreeIPA 89
3.3.2. HTTP, FTP, Web Sunucu Hizmetleri İçin Apache 90
3.3.3. Mail Sunucu Hizmetleri İçin Zimbra 91
3.3.4. Veritabanı Yönetimi İçin PostgreSQL 93
3.4. DİĞER SUNUCU HİZMETLERİ İÇİN CENTOS VEYA PARDUS 96
3.5. YAZILIM EŞDEĞERLİĞİ 99

SONUÇ VE DEĞERLENDİRME 100
KAYNAKÇA 106
ÖZET 117
ABSTRACT 118
ÖZGEÇMİŞ 119

KISALTMALAR

A.g. : Adı Geçen
AKY : Açık Kaynak Yazılım
ASM : Assembly Dili
ASAL : Askeralma Genel Müdürlüğü
AT&T : American Telephone and Telegraph Company
BSD : Berkeley Software Distribution
C. : Cilt
CACM : Communications of the Association for Computing Machinery
CIFS : Common Internet File System
CSIS : Center for Strategic and International Studies
CSV : Comma Separated Values
DC : Domain Controller
DFSG : Debian Free Software Guidelines
DHCP : Dynamic Host Configuration Protocol
DNS : Domain Name System
EPDK : Enerji Piyasası Düzenleme Kurumu
EULA : End User Licence Agreement
FOSS : Free and Open Source Software
FSF : Free Software Foundation
FTP : File Transfer Protocol
GCC : GNU Compiler Collection
glibc : GNU C Library
GNU : GNU's Not Unix
GNU GPL : GNU General Public Licence
GPL : General Public Licence
GUI : Graphical User Interface
IRS : Internal Revenue Service
IT : Information Technology
ITS : Incompatible Timesharing System

ITU : International Telecommunication Union
İng. : İngilizce
LDAP : Lightweight Directory Access Protocol
LGPL : The GNU Lesser General Public License
MEA : Microsoft Enterprise Agreement
MIT : Massachusetts Institute of Technology
MPL : Mozilla Public License
MSB : Millî Savunma Bakanlığı
NSS : Name Service Switch
OSI : Open Source Initiative
OSS : Open Source Software
ÖAK : Özgür ve Açık Kaynak
ÖAKY : Özgür ve Açık Kaynak Yazılım
RTÜK : Radyo ve Televizyon Üst Kurulu
s. : Sayfa
S. : Sayı
SaaS : Software As A Service
SKAAS : Sayısal Kayıt Arşiv ve Analiz Sistemi
SMB : Server Message Block
SMBFS : SMB File System
TÜBİTAK : Türkiye Bilimsel ve Teknolojik Araştırma Kurumu
UCB : University of California Berkeley
USL : Unix System Laboratories
VKS : Video Kayıt Sistemi
VNC : Virtual Network Computing
VPN : Virtual Private Network
ZPL : Zimbra Public Licence

TABLOLAR DİZİNİ

Tablo 2.1: Kurumsal Açık Kaynak Girişimlerinin Yıllara Göre Dağılımı 40
Tablo 2.2: Kurumsal Açık Kaynak Girişimi Sayısı 41
Tablo 2.3: Kurumsal Açık Kaynak Girişimlerinin Ulusal Kurumlar ile Yerel Yönetimler Arasındaki Dağılımı 42
Tablo 2.4: Kurumsal Açık Kaynak Girişimlerinin Bölgesel Dağılımı 42
Tablo 3.1: RTÜK, Göç Modeli Tablosu 87
Tablo 3.2: RTÜK, Alternatif ÖAK Yazılımlar..... 88
Tablo 3.3: Windows – Linux Eşdeğer Yazılımlar..... 99

ŞEKİLLER DİZİNİ

Şekil 1.1: EULA Lisansı Örneği 25
Şekil 1.2: EULA Lisansının Coğrafi Sınırlandırma Örneği 25
Şekil 1.3: Microsoft'un Enterprise Sürümü için Gösterdiği Lisans Bildirimi ... 26
Şekil 2.1: Ticari/tescilli ve ÖAK Yazılım Kullanım Oranı Projeksiyonu 62
Şekil 2.2: ÖAK / SaaS Yazılımların Maliyet/Zaman Grafiği 63
Şekil 2.3: Açık Kaynak Sisteme Göç Projeksiyonu 74
Şekil 2.4: İki Sistemin Birbirine Entegre Edildiği Geçiş Modeli 78
Şekil 2.5: İki Sistemin Ayrı (Paralel) Olduğu Geçiş Modeli 79

GİRİŞ

Geçtiğimiz yüzyılın ortalarından başlayıp günümüze kadar devam eden süre içerisinde, dijitalleşme insan yaşamını şekillendiren en önemli unsur olmuştur. 1947'de Bell Laboratuvarlarında geliştirilen ilk transistörle başlayan dijitalleşme serüveni, kısa süre içerisinde savunmadan ticarete, eğitimden iletişim teknolojilerine kadar birçok alanı yeniden şekillendirmeyi başarmıştır. Günümüzde insanlara, şirketlere ve kurumlara ait tüm veriler, modern bilgisayar ve sistemlerin oluşturduğu dijital bir evrende kayıt altına alınıyor, transfer edilip gerektiğinde işleniyor. Bu modern bilgisayarlar ve sistemler ile bunları çalıştıran yazılım ve işletim sistemleri ise sürekli gelişerek içinde bulunduğumuz dijital çağın temel itici gücünü oluşturmuştur. Bu sebeple yazılım ve işletim sistemleri, kişi ve kurumlar için bir seçenekten çok, gereklilik ve bağımlılık ağı oluşturan temel teknolojik ihtiyaçlar olarak bu dijital evrende yerlerini almıştır.

Dünya çapında aktif internet kullanıcı sayısının 4 milyara[1] ulaştığı bu günlerde, küreselleşme ve sanayileşme dijitalleşme eğilimini hızla artırmış, yazılım ve işletim sistemlerine olan talep de buna paralel olarak artmıştır. Şirketler ve kurumlar, sadece bilgi sistem altyapılarında sahip oldukları verilerin işlenmesi işini değil; aynı zamanda üretim, sanayi, enerji, iletişim ve savunma gibi kritik alanların altyapısında yer alan sistem, cihaz, makine ve araçların yönetimi işini de bu yazılım ve işletim sistemlerine bırakmıştır. Bu yazılımların ve işletim sistemlerinin bu denli yaygın ve yoğun kullanımı, şirket

[1] ITU, (2020). *"Measuring Digital Development Facts and Figures"*, s.7, İsviçre. https://www.itu.int/en/ITU-D/Statistics/Documents/facts/FactsFigures2020.pdf (09.01.2021)

ve kurumların bu yazılımlardan beklentilerini ve kriterlerini de sürekli değişime uğratmıştır. Yazılım ve işletim sistemlerinin veri işleme politikaları, algoritmaları, sürdürülebilirlikleri, sistem esneklikleri, güvenirlikleri ve maliyetleri de veri sahibi kişi ve kurumlar için bir tercih unsuru haline gelmiştir. Beklenti ve kriterlerin çeşitlenmesi, yazılım ve işletim sistemlerinin de çeşitlenmesine ortam hazırlamıştır. Bu değişim ve dönüşüm süreci, yazılım geliştirme süreçlerinde iki farklı akım oluşturmuştur: Açık Kaynak Yazılımlar ve Açık Kaynak Olmayan Yazılımlar.

Açık Kaynak Yazılımlar, genellikle herhangi bir lisans bedeli ödemeden temin edilebilen; kaynak kodlarının kullanılmasına, incelenmesine, değiştirilmesine ve üçüncü taraflarla paylaşılmasına imkân veren yazılımlardır. Açık Kaynak Olmayan Yazılımlar ise lisans şartları taraflar arasında yapılan sözleşmeye bağlı olarak değişse de genellikle bu imkânları sunmayan yazılımlardır. Kişi, kurum ve şirketler; ihtiyaçlarına, beklentilerine ve hedeflerine göre bu iki ana akımdaki yazılımları da tercih edebiliyorlar.

Günümüzde açık kaynak olmayan yazılımlar sektörde daha fazla pazar payına sahip olsa da son zamanlarda açık kaynak yazılımlar da pazar paylarını büyük ölçüde artırmıştır. Ayrıca son yıllarda, şirketler ve kurumlar artan yazılım ihtiyaçlarını karşılamak için özellikle ekonomik gerekçelerle hızla açık kaynak yazılımlara yönelmektedirler.

Şirketler ve kurumlar, açık kaynak yazılımları sistem ve altyapılarının ilk planlamasında tercih ettikleri gibi; açık kaynak olmayan lisanslı yazılımlarla çalışan mevcut sistemlerini de sonradan açık kaynak yazılımlı sistemlere dönüştürebiliyorlar. Veri sahibi tarafın, sahip olduğu sistemin altyapısında yapacağı bu kapsamlı değişim “Göç” olarak adlandırılmaktadır. Açık kaynaklı sistemlere yapılacak bir göç ise, kapsamlı bir maliyet analizi ve planlama

gerektiren önemli bir süreçtir. İyi planlanmamış, kısmen ya da tamamen başarısız olmuş göç operasyonları; ek maliyet yükü, teknolojik beklentilerin karşılanamaması, güvenlik açıklarının oluşması ve genel sistem işleyişinin aksaması gibi riskler taşır.

Tezin Konusu ve Amacı

Aslında “Göç” her iki yönde de geçişi ifade eder. Yani bu kavram, açık kaynak olmayan sistemden açık kaynak sisteme geçişi ifade ettiği gibi tersi durumu da ifade eder. Ancak bu çalışmayla, özellikle kamu kurumlarında, açık kaynak yazılımların kullanımı ve bu sistemlere geçiş konusunda, gerekli motivasyonun oluşmasına katkı sunulması hedeflendiğinden; bu tezde, sadece açık kaynak yazılımlara geçiş süreçlerinin analizi yapılmıştır. Yapılan analizler sonucu elde edilen verilerle, fayda / risk analizi yapılarak, muhtemel bir RTÜK-Açık Kaynak Göçü için gerekli bilgi birikimine katkı sağlanması amaçlanmaktadır.

Tezin Önemi

Açık Kaynak Yazılımların kamu kurum ve kuruluşlarında kullanımını konu alan birçok çalışma olmasına rağmen bu yazılımlara geçiş süreçlerini irdeleyen, bu geçiş süreçlerinin riskleri ve maliyeti konusunda analiz yapan bir çalışma yoktur. Muhtemel göç süreçlerini planlayan kamu kurumlarının, potansiyel risk ve maliyet kestiriminde bu çalışmadan yararlanabilecekleri düşünülmektedir. Ancak bu çalışmadan sağlanabilecek fayda, içeriğinin güncelliğine bağlıdır.

Tezin Yöntemi

Bu tez çalışması hazırlanırken; konu ile ilgili raporlar, kurumsal web siteleri, kitaplar, makaleler ve tezlerden elde edilen veriler kullanılmış olup elde edilen bu veriler, bilimsel etik kurallarına uygun biçimde teze dâhil edilmiştir. Konuyla ilgili olaylar, bütüncül bir biçimde gözlemlenebilir, ölçülebilir ve sayısal olarak ifade edilebilir bir şekilde ortaya koyulmuştur. Ayrıca deneysel ve olgusal tümevarım yöntemi ile ele alınan ikincil türde verilerle tezin genel yapısı oluşturulmuştur.

Kapsam ve Sınırlılıklar

Çalışmanın ilk bölümünde, özgür ve açık kaynak yazılımların tanımları ve tarihsel gelişim süreçleri incelenerek kavramsal çerçeve çizilmiş; Linux ve GNU gibi açık kaynak projelerin gelişim süreçleri ile mevcut lisansları hakkında genel bilgi verilmiş; diğer özgür ve açık kaynak yazılım lisansları hakkında da genel bilgi verilerek özgür ve açık kaynak hareketinin mevcut durumu açıklanmıştır. Ayrıca bu bölüm altında tescilli / ticari / özel mülk yazılımlardan ve EULA lisanslarından da kısaca bahsedilmiştir.

İkinci bölümde ise göç kavramının genel tanımı yapılmış; açık kaynak yazılımlara göç süreçleri ve bu girişimlerin başarı istatistikleri analiz edilmiş; Dünyada, özgür ve açık kaynak yazılımlar ile işletim sistemlerinin kullanım yoğunlukları irdelenmiştir. Devamında ise özgür ve açık kaynak yazılımlara yapılacak muhtemel bir göçün sağlayacağı fayda ve avantajlar verilmiş; yapılması planlanan muhtemel bir açık kaynak göçünün taşıdığı riskler anlatılmış; Türkiye'deki mevcut geçişler araştırılmış ve son olarak bir göç operasyonunun sahip olması gereken teknik altyapı hakkında bilgi verilmiştir.

Üçüncü ve son bölümde ise muhtemel bir RTÜK-Açık Kaynak Göçü için basit ve temel düzeyde bir geçiş modeli önerilmiştir. Bilgi güvenliği gerekçesiyle önerilecek modelde, RTÜK'ün mevcut bilişim altyapısı hakkında ayrıntılı teknik bilgiye yer verilmemiş olup bu durum önerilecek modeli sınırlandırmıştır. Bu çalışmanın amaçları doğrultusunda; tescilli / ticari / özel mülk yazılımlar ve bu sistemlere geçiş süreçlerine bu çalışmada yer verilmemiştir. Ayrıca tez kapsamında konu edilen açık kaynak yazılımlar ve işletim sistemlerinin ana omurgasını, LINUX/UNIX tabanlı yazılım ve işletim sistemleri ile bunların türevleri oluşturmaktadır.

Veri Toplama ve Veri Analiz Teknikleri

Bu tez çalışması, kaynak tarama ve örnek olayların incelenmesi yöntemleri kullanılarak hazırlanmıştır. Kurumsal rapor ve kılavuzlarda yer alan bilgiler ile daha önce gerçekleştirilmiş göç süreçleri incelenerek elde edilen veriler derlenerek içeriğe eklenmiştir.

BİRİNCİ BÖLÜM

ÖZGÜR / AÇIK KAYNAK YAZILIM

1.1. ÖZGÜR / AÇIK KAYNAK TANIMI VE GELİŞİMİ

Açık Kaynak Yazılımlar (*İng.: Open Source Software-OSS*), genelde herhangi bir lisans bedeli ödemeden temin edilebilen; kaynak kodlarının kullanılmasına, incelenmesine, değiştirilmesine ve üçüncü taraflarla paylaşılmasına imkân veren yazılımlardır. AKY için bu tanım genel bir görüş olarak kabul edilebilir. Ancak bu tanımı kısmen ya da tamamen karşıladığı kabul edilen başka bir kavram daha var; Özgür Yazılım (*İng.: Free Software*). Peki bu iki kavram aynı şeyleri mi ifade ediyor? Ya da farklı değerleri temsil ediyorlarsa nerede ayrılıyorlar? Aslında bu soruların basit bir cevabı yok. Çünkü bu iki kavramın nerede başlayıp nerede bittiği, nerede buluşup nerede ayrıştıkları konusunda günümüzde bile genel bir kabul söz konusu değil.

AKY ile Özgür Yazılım kavramlarının aynı değerleri temsil ettiği görüşü ya da tamamen farklı akımlar olduğu görüşü gibi birçok farklı bakış açısı ve tartışma mevcut. Tüm bu farklı görüşlere rağmen iki kavramın da benzer temel yaklaşımları benimsedikleri söylenebilir. İki kavram arasında belirgin bir ayrım yapabilmek için ise, bu kavramların ortaya çıkış ve gelişim süreçlerini ele alarak kavramsal çerçeveyi oluşturmak gerekir.

1.1.1. UNIX'in Geliştirilmesi

AKY ve Özgür Yazılım kavramlarının gelişimi UNIX işletim sisteminin ilk geliştirildiği yıllara kadar uzanır. UNIX, 1970'li yıllarda, Ken Thompson ve Dennis Ritchie, ile B. Kernighan, D. McIlroy ve J. Ossanna tarafından AT&T-Bell Laboratuvarlarında geliştirilmiştir. Sistem, zaman paylaşımlı, çoklu görev ve çoklu kullanıcı yapısında taşınabilir bir işletim sistemi olarak C ve Assembly dillerinde yazılmıştı. Ken Thompson ve Dennis Ritchie UNIX'i, PDP-11 ve Interdata 8/32 bilgisayarları için geliştirilmiş; genel amaçlı, çok kullanıcılı, taşınabilir, etkileşimli, sökülebilir birimler içeren hiyerarşik bir dosya sistemine sahip, asenkron işlemleri başlatabilen bir işletim sistemi olarak tanımlamışlardı.[2] Ancak ilk geliştirdikleri versiyonda bu özelliklerin birçoğu mevcut değildi. UNIX'in geliştirilen ilk orijinal versiyonu, PDP-11 bilgisayarının donanımına uygun olarak Assembly dilinde yazılmıştı. Bu sebeple ilk versiyon taşınabilir değildi. Ayrıca çoklu görev yapısına da uygun tasarlanmamıştı. Ancak UNIX için en önemli dönüm noktası, işletim sistemi çekirdeği Kernel'in C'de yeniden yazıldığı 1973'te gerçekleşti.[3] Yeniden C dilinde yazılan bu versiyon UNIX'in dördüncü versiyonu olmuştu. Ama hala PDP-11 bilgisayar donanımına bağımlı kodları bünyesinde barındırıyordu. Bu sebeple taşınabilir bir özellik kazansa da tam olarak "porting" işlemine hâlâ uygun değildi. İlk tam taşınabilir UNIX, 1978 yılında geliştirilmiş olup taşınamaz kod bloklarının tespitine yardımcı olan bazı araçlarla birlikte geliştirilmişti. Bu araçlarla birçok program, geliştirildikleri PDP-11'den diğer makinelere taşınmıştı. Özellikle, UNIX işletim sistemi ve yazılımlarının birçoğu Interdata 8/32 bilgisayarına taşınmıştı.[4] Böylece artık bu versiyonla UNIX; uyumlu dosya, cihaz ve işlemler arası giriş/çıkış işlemleri yapabilen, kullanıcı bazında seçilebilen sistem komut diline sahip, Fortran ve Pascal dahil bir düzine dili içeren yüzden fazla alt sistemi bünyesinde barındıran taşınabilir bir işletim sistemi haline gelmişti.[2]

---

[2] Dennis M. Ritchie ve Ken Thompson, (1978). *"The UNIX Time-Sharing System"*, The Bell System Technical Journal, AT&T Bell Laboratories, ABD.
[3] Dennis M. Ritchie, (1984). *"The Evolution of the Unix Time-sharing System"*, The Bell System Technical Journal, AT&T Bell Laboratories, ABD.
[4] Stephen C. Johnson, Dennis M. Ritchie, (1978). *"Portability of C Programs and the UNIX System"*, The Bell System Technical Journal, AT&T Bell Laboratories, ABD.

Ayrıca işletim sistemi çekirdeğinin geliştirilmesi sürecinde Dennis Ritchie'nin C dilini geliştirmesi yazılım tarihi açısından da önemli bir dönüm noktası olmuştur.

1974 yılında Ritchie-Thompson ikilisi CACM'de bir makale yayınlayarak UNIX'i resmi olarak duyurmuşlardı. Yayınlanan makale sonrası dünyanın birçok yerinden araştırma laboratuvarları ve üniversiteler UNIX'i denemek için sıraya girmişti. Ayrıca altıncı versiyonu ile UNIX, kazandığı özellikler bakımından da ticari değere sahip olabilecek potansiyele erişmişti. Bunun üzerine AT&T, UNIX'i lisanslayıp ticari bir ürüne dönüştürmek istiyordu. Ancak AT&T'nin telefon ve diğer iletişim hizmetleri dışında bir alana girmesi yasal olarak engellenmişti.[5] Bu sebeple UNIX'i lisanslayıp bir ticari ürüne dönüştüremiyor; bu telefon dışı teknolojisini, isteyen herkese lisanslaması gerekiyordu. Bu arada Ken Thompson ise, talepleri sessizce yanıtlamaya ve her birine -efsaneye göre- "Sevgiler, Ken" imzalı bir notla birlikte kasetler ve diskler göndermeye başlamıştı bile.[5] AT&T de, mevcut yasal engeller sebebiyle UNIX'i, isteyen herkese lisanslayıp kaynak kodlarını da paylaşıyordu. AT&T, beşinci versiyonu o zamana göre çok düşük bir ücret karşılığında üniversitelere lisanslıyordu. Altıncı versiyonu ise ticari kullanıcılara daha yüksek bir fiyatla lisanslanıyordu. Böylece AT&T'nin beklentisinin aksine UNIX, ticari bir sır olarak değil isteyen herkesin kaynak kodlarına erişebildiği, değiştirip yeniden programlayabildiği açık kaynak bir sistem haline gelmişti.

Nitekim UNIX'in kaynak kodlarına ve diğer yazılım araçlarına erişim imkânı bulan üniversiteler ve araştırmacılar UNIX'in kaynak kodlarını değiştirip yeniden programlıyorlardı. Özellikle üniversitelerde geliştirilen ve yeniden programlanan versiyonlar hızlıca yayılıp UNIX evreninin sınırsız genişlemesine yardımcı oluyordu. Bu bilgi paylaşımı ve kolektif çalışma kültürü, özgür yazılım / açık kaynak hareketinin de habercisiydi.

[5] Eric S. Raymond, (2003). *"The Art of Unix Programming: Exodus: 1971–1980"*, s.56, ABD.

Ancak AT&T elindeki bu yazılımın ticari potansiyelini değerlendirme fikrinden vazgeçmiyordu. AT&T avukatları, UNIX'i ticari bir sır olarak korumanın yollarını arıyor, patent ve lisanslama ile ilgili yasal engelleri farklı yorumluyorlardı. Nihayetinde, 1979'da sürüm yedinin yayınlanmasıyla da birlikte, UNIX kaynak kodlarının ve lisanslarının üniversitelerdeki derslerde kullanımı yasaklandı. Ancak bu hamle, UNIX'in kaynak kodlarına sahip araştırmacıların sistemi yeniden kodlamasına engel olamadı. Dünyanın birçok yerinden araştırmacılar UNIX'i yeniden programlıyor, açık kaynak olarak kullanmaya ve paylaşmaya devam ediyordu.

Yale Üniversitesindeki araştırmacılar UNIX'in kabuğunu değiştirdiler. Purdue Üniversitesi Elektrik Bölümü ise hem UNIX'in daha fazla kullanıcıyı destekleyen bir sürümünü hem de ilk UNIX bilgisayar ağını geliştirdiler. California Üniversitesindeki (Berkeley) araştırmacılar UNIX için birçok araç ve kabuk geliştirdi. Vrije Üniversitesinden Andrew S. Tanenbaum ise, Intel 80286 işlemcisi için bir UNIX klonu olan Minix'i geliştirdi. Bell System, UNIX'in yedinci versiyonunu çıkardığında, önceki versiyonlarda var olan birçok yapısal problemi bu geliştirmelerden yararlanarak çözmüştü.[6]

Avustralya'daki New South Wales Üniversitesinden John Lions, derslerinde kullanmak üzere altıncı versiyon UNIX çekirdeğinin kaynak kodlarını içeren ders notlarını kullanıyordu. Lions'ın notlarından derlediği kitabı, yıllarca Bell Laboratuvarları dışında yayınlanmış tek kapsamlı UNIX Kernel dokümanıydı. Ancak UNIX'in yedinci versiyonu ile lisans ve telif hakları konusunda AT&T daha katı bir duruş sergiliyordu. Bu sebeple telif hakkı ve benzeri nedenlerle kitap ABD'de yayınlanamadı. Buna rağmen kitabın binlerce legal ya da illegal kopyası tüm dünyaya yayılmıştı bile. Bu sayede UNIX çekirdeğine ait kaynak kodları tüm dünya için ulaşılabilir olmuştu.

[6] Eric S. Raymond, (2003). *"The Art of Unix Programming: TCP/IP and the Unix Wars: 1980-1990"*, s.58, ABD.

1.1.2. BSD UNIX

BSD (*İng.: Berkeley Software Distribution*) ya da BSD Unix; Berkeley'deki California Üniversitesi araştırmacıları tarafından geliştirilmiş Research Unix[7] temelli bir işletim sistemidir. İlk BSD UNIX, Üniversitede yüksek lisans öğrencisi olan Bill Joy ve ekibi tarafından 1977'de geliştirilmiştir. İlk geliştirilen BSD sürümü, yeni versiyon bir UNIX benzeri (UNIX-like[8]) işletim sisteminden çok sadece bazı araç ve protokollerin eklendiği güncellenmiş bir Research UNIX idi.

Ken Thompson, 1975 yılında Berkeley'e misafir profesör olarak gidip UNIX'in altıncı versiyonunun laboratuvardaki bir PDP-11 bilgisayarına kurulmasına yardım etti. Ayrıca sistem için Pascal temelli bir uygulama geliştirdi. Bill Joy ve ekibi bu uygulamayı geliştirip kabukta değişiklikler yaptılar ve ilk BSD sürümü olarak 1978'de yayınladılar. Daha sonra, "vi" metin editörü ve bazı araçları 1BSD'ye ekleyerek BSD'nin ikinci versiyonunu geliştirdiler ve 1979'da yayınladılar. Berkeley'de yüksek lisans öğrencisi Özalp Babaoğlu, BSD için sanal bellek eklentileri geliştirmiş; bu sanal bellek desteği ile kernel güncellemelerini alan BSD, üçüncü sürüm olarak yayınlanmıştır. Ayrıca 4.2BSD ile TCP/IP, 4.3BSD ile de "cross-platform" desteği BSD sürümlerine gelmiştir.

Artık UCB'nin elinde TCP/IP, cross-platform ve sanal bellek gibi destekleri almış 32-bit bir UNIX-like işletim sistemi vardı. Kısa zamanda geliştirilen bu versiyonların kopyaları BSD lisansı altında, disket ve kaset maliyetleri karşılığında satılmaya başlanmıştı. Ancak 1984 yılında, AT&T aleyhine olan yasal engellerin kalkmasıyla UNIX artık ticari bir ürün olarak

[7] Research Unix: UNIX 'in ilk versiyonları için kullanılan terimdir. Genellikle ilk sekiz versiyon için kullanılır.
[8] UNIX-like: UNIX türevi, UNIX üzerine geliştirilmiş

pazarlanabilir olmuştu. AT&T, agresif bir lisanslama çabasına girişti ve UNIX ve UNIX benzeri yazılımların büyük ölçüde AT&T lisansına bağımlı hale gelmesini sağladı. Bu sayede 4.3BSD'nin çıkışına kadar, BSD'yi temin etmek isteyenlerin önce bir AT&T kaynak lisansı alması gerekiyordu. Bunun nedeni BSD sistemlerinin Berkeley tarafından hiçbir zaman "binary" formatında yayınlanmamasıydı. Dağıtımlar daima sistemin tüm parçalarının kaynak kodlarını içeriyordu. Bu da BSD'yi, sürekli maliyeti artan AT&T kaynak lisanslarına bağımlı kılıyordu.

BSD kullanarak PC pazarı için bağımsız TCP/IP tabanlı ağ ürünleri geliştirmek isteyen satıcılar, Berkeley'den ağ kodunu ve yardımcı programları BSD sürümlerinden çıkarmasını ve bunları AT&T kaynak lisansı gerektirmeyen lisans koşulları altında sağlamasını istediler.[9] Böylece BSD kaynaklı ağ kodu ile destekleyici araçları içeren ve serbestçe yeniden dağıtılabilen ilk BSD sürümü "Networking Release 1" 1989'da piyasaya sürüldü. Daha sonra UCB araştırmacıları, çekirdeğin büyük bir kısmını yeniden programlayıp 4.3BSD/Networking Release 2'yi 1991'de yayınladılar. Bu sürümde altı tane çekirdek dosyası hariç, AT&T kodlarının büyük bir kısmı BSD lisansına uygun olarak yeniden yazıldı.[9]

Sonraki altı ay içinde ise UCB araştırmacıları William (Bill) Jolitz ve eşi Lynne Jolitz eksik altı dosyayı yeniden yazdı. Jolitz, 386BSD adını verdiği 386 tabanlı PC mimarisi için tamamen derlenmiş ve önyüklenebilir bir sistemi 1992'de yayınladı. Jolitz, geliştirdiği sistemi tamamen açık kaynak olarak bir FTP sunucuda yayınlayıp herkesin ücretsiz bir şekilde indirmesine izin vermişti.[9] 386BSD, bugün dahi kullanılan tüm BSD'lerin atası olarak kabul edilir. 386BSD'nin yayınlanmasından sonraki süreçte ise 386BSD için gerekli hata düzeltmeleri ve sistem iyileştirmeleri yapmak üzere bir grup 386BSD

[9] Chris DiBona, Sam Ockman, Mark Stone, (1999). *"Open Sources: Voices from the Open Source Revolution: A Brief History of Hackerdom By Eric S. Raymond"*, s.19, O'Reilly, ABD.

kullanıcısı, sistemin iyileştirilmesine ve daha sonra geliştirilmesine yardımcı olmak için NetBSD grubunu oluşturdu.

1.1.3. NetBSD, FreeBSD, OpenBSD

NetBSD, 386BSD dağıtımı üzerinde geliştirilmiş, açık kaynak bir Unix benzeri işletim sistemidir. 386BSD'nin yayınlanmasından sadece birkaç ay sonra, gerekli hata düzeltmeleri, sistem iyileştirmeleri ve ilgili araç setlerinin geliştirilmesi ile bunların açık kaynak olarak paylaşımı için NetBSD grubu kurulmuştur. Yayınladıkları yamalar ve BSD türevi işletim sistemleri de NetBSD olarak adlandırılmıştır. NetBSD'nin kernel çekirdeğinin tamamı ve kaynak kodunun büyük bir bölümü, BSD Lisansı altında yayınlanmıştır. Böylece, açık kaynak olarak kullanması, değiştirilmesi, yeniden dağıtılması veya satılması için bir engel yoktur. Bu sebeple NetBSD tabanlı ürünlerin geliştirilmesi, kaynak kodunda genel değişiklikler yapmak zorunda kalmadan mümkündür.

NetBSD gibi FreeBSD ve OpenBSD de açık kaynak yazılımlar olup UNIX-like işletim sistemleridir. 4.4BSD-Lite ve 386BSD'den türetilen birçok işletim sistemi olmasına rağmen bu üç BSD türevi, en çok bilinen ve yaygın olarak kullanılan sistemlerdir. Hem NetBSD hem de FreeBSD, 1993 yılında 386BSD üzerinde geliştirildi. Ancak 1994'te 4.4BSD-Lite kod tabanına geçildi.[10] OpenBSD, 1995 yılında NetBSD'den ayrılarak gelişmiştir. Mevcut BSD işletim sistemlerinin çoğu açık kaynak kodludur ve BSD Lisansı altında ücretsiz olarak indirilebilir ve değiştirilebilirler. Ancak bunun en önemli istisnaları macOS ve iOS'dur.

[10] Chris DiBona, Sam Ockman, Mark Stone, (1999). *"Open Sources: Voices from the Open Source Revolution: A Brief History of Hackerdom By Eric S. Raymond"*, s.19, O'Reilly, ABD.

FreeBSD, bir işletim sistemini herhangi bir amaç için kullanılabilir hale getirmeyi amaçlayan bir yaklaşımla geliştirilmiştir. Çok çeşitli uygulamaları çalıştırması, kullanımı kolay olması, en son özellikleri içermesi ve çok yüksek oranda ölçeklenebilir olması amaçlanmıştır.[11]

FreeBSD tamamen ücretsiz olup kaynak kodları değiştirilip yeniden dağıtılabilir. Hatta kaynak kodlarını değiştirip yeniden dağıtmak ve satmak isteyen ticari kullanıcılar için bile ücretsizdir.

OpenBSD ise, NetBSD ekibinden 1994'te ayrılan Theo de Raadt tarafından kurulmuştur. OpenBSD ilk BSD türevi işletim sistemini 1996'da yayınlamıştır. Diğer sürümlere göre daha çok güvenlik ve kriptografi protokolleri üzerine yoğunlaşarak geliştirilmiş bir sistemdir.

Günün sonunda, bu üç BSD temelli Unix-like işletim sistemleri, açık kaynak olmaları, ücretsiz olmaları ve değiştirilip yeniden dağıtılabilir olmaları sebebiyle günümüzde Özgür ve Açık Kaynak Yazılımlar tarafından yaygın olarak kullanılan sürümler olmuşlardır. Ayrıca FreeBSD grubu, Linux binary dosyalarının FreeBSD platformunda çalışmasına izin veren bir Linux emülatörü geliştirmiştir. Bu sayede kullanıcılar, FreeBSD sisteminin sağlamlığını, güvenilirliğini ve performansını elde ederken, Linux için geliştirilmiş olan ve gittikçe sayıları artan uygulamaları kullanmaları da mümkün olmuştur.[12] Böylece FreeBSD, diğer BSD türevlerine göre daha geniş bir kullanıcı kitlesine sahip olmuştur.

---

[11] The FreeBSD Foundation. *"FreeBSD Documents, Chapter 1: Introduction Part I. Getting Started"*, https://docs.freebsd.org/en/books/handbook/introduction/ (16.01.2021)

[12] Chris DiBona, Sam Ockman, Mark Stone, (1999). *"Open Sources: Voices from the Open Source Revolution:Twenty Years of Berkeley Unix, From AT&T-Owned to Freely Redistributable By Marshall Kirk McKusick"*, s.26 O'Reilly, ABD.

1.1.4. BSD Lisans Süreci

1992'de USL/AT&T, BSD ile ilgili lisans ihlalleri nedeniyle UCB'ye dava açtı. UCB ise AT&T'nin, BSD'nin yaptığı birçok iyileştirme ve geliştirmeyi onayı olmadan kendi UNIX sürümlerine dahil ettiğini öne sürdü.[12] USL davada, UCB'nin büyük ölçüde USL kodu içeren NR/2'yi[13] yayınlayarak; USL'nin yazılım lisans sözleşmelerini ve UNIX üzerindeki telif haklarını ihlal ettiğini, USL'nin UNIX'e dair ticari sırlarını kötüye kullandığını iddia etti. Bu gerekçelerle USL, mahkemeden, dava karara bağlanana kadar BSD'nin NR/2 sürümünü dağıtmasını engelleyecek bir ihtiyati tedbir talebinde bulundu. Bu dönemde bazı UCB programcıları, BSD ile ilişkili tüm AT&T kodlarını yeniden yazmak için bir proje başlattı ve AT&T/USL kodlarından tamamen arındırılmış 4.4BSD-Lite sistemi geliştirildi.[14] Dava ise 1994'te taraflar arasında varılan anlaşmayla sonuçlandı. Anlaşmaya göre; [15]

- 4.4BSD-Lite, tartışmalı dosyalar içermeyecek şekilde piyasaya sürülecek ve UCB, lisans sahiplerini NR/2'den 4.4BSD-Lite'a geçiş yapmaya teşvik edecek.
- UCB, belirli dosyaların dağıtımını durduracak.
- USL, belirli dosyaların ücretsiz dağıtımına izin verecek.

Berkeley dağıtımındaki 18.000 dosyadan sadece üçünün kaldırılması ve 70'inin USL telif hakkı bildirimlerini göstermek için değiştirilmesi gerekiyordu. Anlaşmanın en önemli maddesi ise, USL'nin çıkacak olan 4.4BSD-Lite sürümünün kullanıcıları ve dağıtımcılarına karşı başka dava açmamasıydı.[15]

[13] NR/2: Network Release 2, BSD, UNIX-like
[14] 386BSD ve 4.3BSD NR/1 Projesi
[15] Nokia Bell Labs. (1994). *"UNIX System Laboratories Inc. vs BSDi Settlement Agreement"*, https://www.bell-labs.com/usr/dmr/www/bsdi/USLsettlement.pdf (20.01.2021)

Bu anlaşma UCB'ye, BSD'nin kendi lisans koşulları altında dağıtımını yasal olarak yapmasına imkân veriyordu. Ancak UCB kısa zaman sonra BSD'ye desteğini kesti. Buna rağmen BSD türevi olan NetBSD, FreeBSD ve OpenBSD gibi dağıtımların ortaya çıkışı açık ve özgür kaynak yazılım kültürünün artık geliştiğini gösteriyordu. Günümüzde ise BSD lisansı içinde farklı yaklaşımlar oluşmuştur. 4 maddeli lisans, 3 maddeli lisans, 2 maddeli ya da 0 maddeli lisans bildirimi gibi. Dört maddeli lisans orijinal BSD lisansı kabul edilir. Üç maddeli lisans ise günümüzde kullanılan BSD lisansıdır. Bu lisans maddeleri BSD'nin kullanımı ve yeniden dağıtımı için herhangi bir ücretlendirme getirmez ya da kullanıcının kaynak kodları üzerinde sahip olduğu hakları sınırlamaz. Özellikle günümüzde kullanılan üç maddeli BSD lisansı, özgür ve açık kaynak lisansları ile GPL lisanslarına da uyumludur.

1.1.5. Özgür Yazılım

Özgür Yazılım konsepti ilk olarak 1983'te Richard Stallman tarafından açılımı "GNU UNIX Değildir" (*İng.: GNU's Not UNIX*) olan GNU projesiyle ortaya atılmıştır. Stallman, 1985 yılında ise kâr amacı gütmeyen "Özgür Yazılım Vakfı'nı kurarak ortaya çıkardığı bu konseptin gelişerek yayılmasını desteklemiştir.

Stallman, 1971 yılında MIT'de yapay zekâ laboratuvarında çalışmaya başladığında aynı zamanda bir yazılım paylaşım topluluğunun da bir üyesi olmuştu. Bu toplulukta kaynak kodlarını görüp, okuyup ve değiştirip kullanabilecekleri programları paylaşabiliyorlardı.[16] Ayrıca burada PDP-10 bilgisayarı için ASM dilinde bir Zaman Paylaşımlı İşletim Sistemi (ITS)

[16] Chris DiBona, Sam Ockman, Mark Stone, (1999). *"Open Sources: Voices from the Open Source Revolution:The GNU Operating System and the Free Software Movement By Richard Stallman"*, s.32, O'Reilly, ABD.

geliştirmişlerdi. Diğer üniversitelerden ya da firmalardan bu yazılımın kaynak kodlarını kullanmak isteyenlere de karşılıksız bir şekilde izin veriyorlardı. Stallman ve arkadaşları, geliştirdikleri bu yazılımı Özgür Yazılım olarak adlandırmamışlardı çünkü henüz böyle bir kavram gelişmemişti.[17]

İlerleyen yıllarda MIT'deki bu topluluk dağıldı. Ayrıca yapay zekâ laboratuvarında kullanılan bir yazıcı kontrol programının kaynak kodlarının kendisi ile paylaşılmaması da Stallman'ı oldukça sinirlendirmişti.[18] Lisanslı yazılımların dünyasına girmeyi istemeyen ve yazılımların kullanıcılar tarafından ne yapmak istedikleri konusunda özgür olması gerektiğini düşünen Stallman, 1983 yılında GNU projesini duyurdu.

Stallman, MIT'nin GNU'nun özgür yazılım olarak dağıtımına müdahale edememesi ve kendi dağıtım şartlarını dayatmaması için 1984'te MIT'deki işinden ayrıldı ve GNU'yu yazmaya başladı.[19] Ortaya attığı "Free Software" fikri, yazılımların ücretlendirilmemesi ile ilgili bir şartı öne sürmüyordu. Stallman'ın ortaya attığı konseptle ilgili düşüncesi de "fiyatla hiçbir ilgisi yok tamamen özgürlükle alakalı" şeklindeydi.[19]

Özgür yazılım konsepti; kullanıcıların yazılımı çalıştırma, kopyalama, dağıtma, inceleme, değiştirme ve geliştirme özgürlüğüne sahip olduğu fikrine dayanıyordu. Yani odak noktası, yazılım alıcısının yazılımla ne yapmasına izin verildiği ile alakalıydı. Bir yazılımın ücretli ya da ücretsiz olup olmaması özgür yazılım için bir parametre değildi.

---

[17] Richard M. Stallman, (2015). *"Free Software, Free Society: Selected Essays of Richard M. Stallman: The GNU Project"*, s.9, ABD.
[18] Chris DiBona, Sam Ockman, Mark Stone, (1999). *"Open Sources: Voices from the Open Source Revolution:The GNU Operating System and the Free Software Movement By Richard Stallman"*, s.32, O'Reilly, ABD.
[19] Richard M. Stallman, (2015*). "Free Software, Free Society: Selected Essays of Richard M. Stallman: Selling Free Software"*, s43., ABD.

Stallman, bir yazılımın özgür olması için şu ilkelere uyması gerektiğini söylemiştir; [20]

- Yazılımın herhangi bir amaç için istenildiği şekilde çalıştırılabilmesi özgürlüğü. (0 numaralı özgürlük).
- Yazılımın nasıl çalıştığını öğrenebilme ve onu ihtiyaçlar doğrultusunda değiştirebilme özgürlüğü (1 numaralı özgürlük). Yazılımın kaynak koduna erişim şartı ile.
- Kopyaları ücretli ya da ücretsiz yeniden dağıtabilme özgürlüğü (2 numaralı özgürlük)
- Yazılımın değiştirilmiş sürümlerini de dağıtabilme özgürlüğü. (3 numaralı özgürlük)

1.1.6. GNU Projesi

Stallman'ın 1984'te geliştirmeye başladığı GNU projesinin amacı, tüm yazılım setleri de dahil tamamen özgür yazılım bir işletim sisteminin geliştirilmesiydi. Stallman, geliştireceği sistemin taşınabilir ve UNIX ile uyumlu olmasını istiyordu. Böylece, o günlerde oldukça popüler olan UNIX ve UNIX benzeri işletim sistemlerini kullanan kullanıcıların GNU'ya geçişi daha kolay olacaktı. Proje kapsamında geliştirilen yazılım setlerinin büyük bölümü C dilinde ve gönüllüler tarafından yazılıyordu. Ayrıca Stallman, açık kaynak olan başka yazılım setlerini de GNU'ya uyarlayıp sistem üzerinde kullanıyordu. TeX'i metin editörü olarak, X Window sistemini ise grafik arayüz altyapısı olarak kullanmıştı.[21]

[20] Richard M. Stallman, (2015*). "Free Software, Free Society: Selected Essays of Richard M. Stallman: What Is Free Software?"*, s.3, ABD.
[21] Richard M. Stallman, (2015*). "Free Software, Free Society: Selected Essays of Richard M. Stallman: GNU Software and the GNU System"*, s.12, ABD.

GNU sistemi için çoklu platform ve çoklu dil desteği sunan GCC derleyicisi de 1987 yılında yayınlandı. Böylece sistem cross-platform yeteneği kazandı ve donanım kısıtlaması olmadan çalıştırılabilir yazılım setleri geliştirmeye uygun bir ortam da sunulmuş oldu. Ayrıca GNU C kütüphanesi, GNU Debugger, GNU Emacs gibi yazılım setleri de proje kapsamında geliştirilmiştir.

GCC derleyicisi, GNU Genel Kamu Lisansı (GNU GPL) kapsamında özgür yazılım olarak dağıtılmakta ve GNU/Linux için standart derleyici konumundadır. GCC, günümüzde en büyük açık kaynak derleyicilerden biri olup bugün dahi açık kaynak ve özgür yazılımın büyümesinde önemli bir rol oynamaktadır. GCC, BSD-like[22] işletim sistemleri için de standart derleyici olarak kullanılmaktadır. İlk sürümlerinde sadece C dilini destekleyen GCC, geliştiren birçok Front-End sayesinde C++, Fortran, Java gibi birçok dili de destekleyebilir duruma gelmiştir. Ayrıca GCC, "libgcc" adlı, C ve işlemciye özgü makine kodu ile yazılmış düşük seviyeli bir kütüphaneyi de içerir. Böylece derleyicinin gömülü sistemlerde de kullanılabilmesi mümkündür.

1.1.7. Özgür Yazılım Vakfı

Özgür Yazılım Vakfı (FSF), 1985 yılında Richard Stallman tarafından özgür yazılım hareketini desteklemek üzere kurulmuş kâr amacı gütmeyen bir kuruluştur. Vakıf, GNU kapsamında geliştirilen yazılım setlerinin büyük bir bölümünü copyleft[23] lisansı altında dağıtmaktadır. FSF, GNU kapsamında sahip olduğu telif haklarında ihlal meydana geldiğinde, GNU Genel Kamu Lisansının (GPL) copyleft gereksinimlerini uygulama yetkisine de sahiptir. Bu kapsamda da FSF, 1991 yılından itibaren IBM ve Cisco dahil birçok tarafa karşı

[22] NetBSD, FreeBSD, OpenBSD vb. BSD türevi.
[23] 1.3.1. bölümünde anlatılmıştır.

GPL lisans ihlalleri nedeniyle dava açmıştır. Bu topluluk günümüzde de faaliyetlerine devam etmekte olup Özgür Yazılım hareketinin kurumsallaşmasına önemli katkıda bulunmaktadır.

1.1.8. Linux ‘un Geliştirilmesi

Linux, 1991 yılında Linus Torvalds tarafından geliştirilmiş açık kaynak bir işletim sistemidir. Sistem, UNIX-like olup, yine Torvalds tarafından yazılan Linux çekirdeği üzerinde geliştirilmiştir.

Torvalds, 1991’de Linux çekirdeğinin ilk sürümünü yayınlamış, yazılımı denemek isteyenlerin indirebilmesi için de Finlandiya Üniversitesindeki bir araştırma ağının FTP sunucusuna yüklemişti. Ancak sistemin ilk sürümünü çalıştırabilmek için kodlarını derleyecek Minix’e ihtiyaç duyuluyordu. Birkaç ay sonra Torvalds, Linux’un ikinci sürümünü yayınladı ve sistemin GCC derleyicisi ile çalışmasını sağladı.

Devam eden süreçte, Özgür Yazılım hareketinden bazı geliştiriciler de dâhil olmak üzere birçok kişi Linux’un geliştirilmesine katkıda bulundu. O zamanlar GNU topluluğu, açık kaynak birçok işletim sistemi aracı geliştirmesine rağmen çalıştırılabilir bir kernel geliştirememişti. Bu açığı kapatmak için GNU-Hurd adlı Microkernel projesini başlatmışlardı. Ancak proje çok yavaş ilerliyor ve birçok yapısal sorun barındırıyordu. Bu sebeple Linux, ilk sürümlerinin sınırlı işlevselliğine rağmen GNU-Hurd’dan daha önce geliştirilmesinin meyvelerini alıyor, hızla yaygınlaşıp GNU araçlarıyla beraber en çok kullanılan açık kaynak işletim sistemi haline geliyordu.

Linux'un yaygınlaşmasıyla Torvalds, 0.12 sürümünü yayınladığında, ticari dağıtıma izin vermeyen kendi belirlediği lisans taslağından vazgeçip, GNU Genel Kamu Lisansı versiyon 2'yi (GPLv2) benimsedi.[24] Böylece UNIX'in aksine, Linux'un tüm kaynak dosyaları ve aygıt sürücüleri, dünyanın farklı yerlerinden binlerce yazılımcının özgür ve açık kaynak olarak geliştirdiği, ücretsiz olarak kullanıp dağıtabildiği bir işletim sistemi çekirdeği haline geldi.

Linux Kernel sürüm 4.1, dünyanın birçok yerinden yaklaşık 13.708 yazılımcının katkılarıyla 19,5 milyondan fazla kod satırı içerecek şekilde Torvalds tarafından 2015'te yayınlanmıştır.[25] Linux Kernel'in son sürümü ise 5.11 olarak 2021 yılında yayınlanmıştır.

Günümüzde Linux olarak adlandırılan işletim sistemleri, aslında Linux Kernel ile GNU araçlarının beraber kullanılmasıyla ortaya çıkan dağıtımları ifade eder. Bu sebeple Stallman, Linux'dan bahsederken GNU/Linux dağıtımları diye bahsetmekte ve sadece bu kullanımın doğru olduğunu savunmaktadır. Torvalds ise bu yaklaşımın Özgür ve Açık Kaynak hareketini böldüğünü iddia etmektedir.

İkili arasındaki bu tartışma bugün dahi devam etmektedir. Aslında GNU projesinin çekirdek eksikliği, tam bir özgür yazılım işletim sisteminin var olmadığı anlamına geliyordu. Torvalds'ın Linux Kernel'i geliştirilmesi bu son boşluğu kapatmıştı. Aynı şekilde Linux Kernel'de sürücü, derleyici, kabuk vs. gibi önemli işletim sistemi araçlarından mahrumdu. İki sistemin bir araya getirilmesiyle, ilk kez tam anlamıyla özgür ve açık kaynak bir işletim sistemi ortaya çıkmıştı.

---

[24] Linus Torvalds, (2007). *"Release Notes for Linux v0.12"*, kernel.org, https://mirrors.edge.kernel.org/pub/linux/kernel/Historic/old-versions/RELNOTES-0.12 (29.01.2021)

[25] Michael Larabel, (2015). *"Linux Kernel At 19.5 Million Lines Of Code, Continues Rising"*, https://www.phoronix.com/ (29.01.2021)

1.1.9. Açık Kaynak Yazılım Hareketi

Açık Kaynak Yazılım konsepti ya da Açık Kaynak Hareketi (OSI), Özgür Yazılım Hareketi'ne alternatif olma düşüncesiyle 1998 yılında Eric S. Raymond ve diğer yazılım geliştiricileri tarafından ortaya atılmıştır. Çünkü onlara göre Özgür Yazılım ilkeleri, ticari yazılım dünyasına tam olarak uyarlanabilir değildi.

Açık Kaynak Yazılım fikrinin oluşması ise o zaman popüler olan Netscape Communicator adlı internet yazılım setinin özgür yazılım olarak yayınlanmasının ardından gerçekleşti. Eric S. Raymond'ın 1997'de yayınladığı "The Cathedral and Bazaar" adlı makalesinin Netscape'i bu yönde teşvik ettiği iddia edilmektedir.[26] Daha sonra kitap haline getirilen bu eserde, iki farklı özgür yazılım geliştirme modeli ele alınmaktadır.

Netscape'in bu kararı Raymond ve diğer açık kaynak destekçilerini ticari yazılım dünyasına özgür yazılım ilkelerini ve faydalarını nasıl getirecekleri konusunda düşünmeye teşvik etti. Bu kapsamda Özgür Yazılım ilkelerinden çok fazla kopmadan yeni bir kavram üzerinde çalışmaya başladılar.

Nihayetinde büyük bölümü Özgür Yazılım hareketi içinde olan yazılımcıların katıldığı bir toplantı California ABD'de gerçekleştirildi ve Açık Kaynak kavramı benimsendi. Katılımcılar, Netscape'i kaynak kodlarını yayınlamaya motive eden gerekçelerin, yazılımların kullanıcıları ve geliştiricileriyle etkileşimde bulunmanın ve onları paylaşımcı bir topluluğa katarak kaynak kodu oluşturmaya ve iyileştirmeye ikna etmenin değerli bir

[26] Christopher M. Kelty, (2008). *"The Cultural Significance of Free Software: Forking Free Software, 1997–2000"*, s.108, Duke University Press, ABD.

yolunu gösterdiğine inanıyorlardı.[27] Devam eden süreçte, Açık Kaynak Hareketini desteklemek ve geniş kitlelere duyurmak üzere 1998 yılının Şubat ayında Bruce Perens ve Eric S. Raymond tarafından Açık Kaynak Girişimi (OSI) kuruldu.

Özgür Yazılım Hareketinin öncüsü Richard Stallman ise, Açık Kaynak Kavramını ilk başlarda desteklese de devam eden süreçte Özgür Yazılım ilkeleriyle bağdaşmadığı gerekçesiyle desteğini geri çekmiştir. Sonrasında ise Stallman ve FSF, açık kaynak hareketine karşı keskin bir cephe almıştır.

Buna rağmen Özgür Yazılım kavramından ziyade Açık Kaynak Yazılım kavramı daha geniş kitleler tarafından kabul gördü ve yazılım dünyası dışında da daha yaygın olarak kullanıldı. Neticede bu iki hareketin farklı yaklaşımlar benimsedikleri söylense de benzer sonuçlara ulaştığı söylenebilir. Bu sebeple ve doğal olarak iki kavramı da aynı çatı altında toplayan yeni bir kavram ortaya çıktı: ÖAKY *(İng.: FOSS).*

1.1.10. Özgür ve Açık Kaynak Yazılım Konsepti (ÖAKY)

Özgür Yazılım ve Açık Kaynak Yazılım hareketlerini oluşturan felsefelerin aslında aynı lisans ve yazılım setlerine atıfta bulunduğu kabul edilebilir ancak özellikle bu iki hareketin öncülerinin bu konudaki fikir ayrılığı yazılım dünyasında kafa karışıklığına da sebep olmuştur. Tüm bu karışıklık Özgür ve Açık Kaynak Yazılım *(İng.: FOSS-Free and Open Source Software)* kavramının gelişmesine ortam hazırlamıştır.

[27] Christopher M. Kelty, (2008). *"The Cultural Significance of Free Software: A Movement?"*, s.113, Duke University Press, ABD.

ÖAKY, AKY ve Özgür Yazılım gibi kesin bir ortaya çıkış tarihi ve kronolojik gelişim sürecine sahip değildir. Tam olarak ne zaman ve nasıl ortaya çıktığıyla ilgili kesin bir kabul de söz konusu değildir.

Açık Kaynak fikri daha çok güçlü, verimli, kararlı ve açık kaynak yazılım geliştirme gibi teknik değerlere odaklanırken Özgür Yazılım daha çok etik ve ahlaki değerlere atıfta bulunur. Diğer bir deyişle Özgür Yazılım kavramı daha sosyal ve ahlaki bir felsefe olarak özgürlüğü şart koşmakta iken Açık Kaynak kavramı ise özgürlük konusunda çok keskin uçlara sahip değildir. Bu sebeple ticari yazılım dünyası Açık Kaynak fikrine açık kapı bırakırken, Özgür Yazılım fikrine ise genellikle karşı çıkmıştır.

Özgür ve Açık Kaynak Yazılımlar gibi ÖAK yazılımları da herhangi bir şekilde kullanmak, kopyalamak, değiştirmek ve dağıtmak hiçbiri kısıtlamaya tabi olmadan mümkündür. Günümüzde sahipli/lisanslı yazılımların tam karşısında ÖAKY'nin konumlandığı kabul edilmektedir.

1.2. ÖAK OLMAYAN YAZILIMLAR VE EULA LİSANSLARI

Açık Kaynak olmayan, Özgür olmayan, sahipli / lisanslı / tescilli / özel mülk *(İng.: Proprietary Software or Closed Source Software)* gibi terimlerle ifade edilen yazılımlardır.[28] Belirli lisans koşulları altında dağıtılan bu tür yazılımlar, yazılımın kullanılmasıyla ilgili kullanıcısına katı kısıtlamalar getirir. Lisans şartları, yazılım sahibi ile kullanıcı arasındaki lisans sözleşmesine göre değişse de genellikle kullanıcıya yazılımın kaynak kodlarını görme imkânı

[28] Bundan sonraki bölümlerde FOSS için *"Açık Kaynak, ÖAKY ya da ÖAK Yazılım"* terimleri, Proprietary Yazılımlar için *"ÖAKY olmayan, ÖAK Olmayan Yazılım, Özel Mülk Yazılım ya da Tescilli/Ticari Yazılım"* terimi kullanılacaktır.

vermez. Ayrıca kullanıcının yazılımı değiştirmesini, dağıtmasını ve üçüncü taraflarla paylaşmasını ciddi şekilde kısıtlar.

Bu sahipli yazılımların ticari bir sır olarak tutulabilmesi için birçok lisans mekanizması mevcuttur: yazılım patentleri, telif hakları, lisans şartları, yazılımın tescili ve yazılım dağıtım standartları gibi.

ÖAK olmayan yazılımlar, genellikle yazılımın geliştiricisinin, yazılımın tüm mülkiyetini elinde tuttuğunu kullanıcıya bildiren özel lisanslara tabidir. Bu mülkiyeti ve lisans şartlarını da genelde son kullanıcıya, Son Kullanıcı Lisans Anlaşması (EULA) ile bildirir. EULA lisans sözleşmesi genelde pazarlığa açık olmayan kesin sözleşmelerdir ve GNU GPL lisansının tam tersi ilkeleri bünyesinde barındırır. Bu sözleşmeler kullanıcı tarafa yazılımın kullanımı ve dağıtımı ile ilgili sorumluluklar da yükler. EULA sözleşmesi genellikle yaygın yazılım dağıtımlarında kullanılan bir yöntemdir. Büyük şirketler, devlet kurumları ve üniversiteler genellikle yazılım sahibiyle özel lisans ve dağıtım anlaşmaları yapar. EULA sözleşmeleri genellikle uzun, karışık, teknik ve yasal bir dille yazılır. Bu da çoğu ortalama kullanıcının sözleşmeleri okumadan onay vermesine sebep olabilir.

GNU Genel Kamu Lisansının aksine özel mülk olan EULA lisansları, kullanıcıya sadece yazılımın bir kopyasını kurup çalıştırma iznini verir. Kopyanın yeniden dağıtılmasına veya üçüncü taraflarla paylaşılmasına izin vermez. Ayrıca yazılımın kaynak kodlarının incelenip değiştirilmesi de mümkün değildir. Şekil 1.1’de, ticari/tescilli yazılım sektöründeki en büyük oyunculardan biri olan Microsoft firmasının, Office yazılımına ait EULA lisans bildirimi görülebilir. Bildirime göre, lisans kapsamında kullanıcıya sadece yazılımı kurup çalıştırma izni verildiği ifade ediliyor.

How can I use the software that is provided as part of the service? We do not sell our software or your copy of it – we only license it. Under our license we grant you the right to install and run that one copy of the software on one licensed device (the first licensed device) for use by one person at a time, but only if you comply with all the terms of this Supplement. The user whose Microsoft account is associated with the software license for the first licensed device is the "licensed subscriber." Provided that you comply with all the terms of this Supplement, you may install and run copies of the software on licensed devices (including on the first licensed device) as follows:

Office 365 Home: On five PCs/Macs and five tablets, for use only by members of the same household as the licensed subscriber. [1]

Office 365 University: On one PC/Mac or tablet and one additional PC/Mac or tablet, for use only by the licensed subscriber. [2]

Office 365 Personal: On one PC/Mac and one tablet, for use only by the licensed subscriber. [1]

Şekil 1.1: EULA Lisansı Örneği

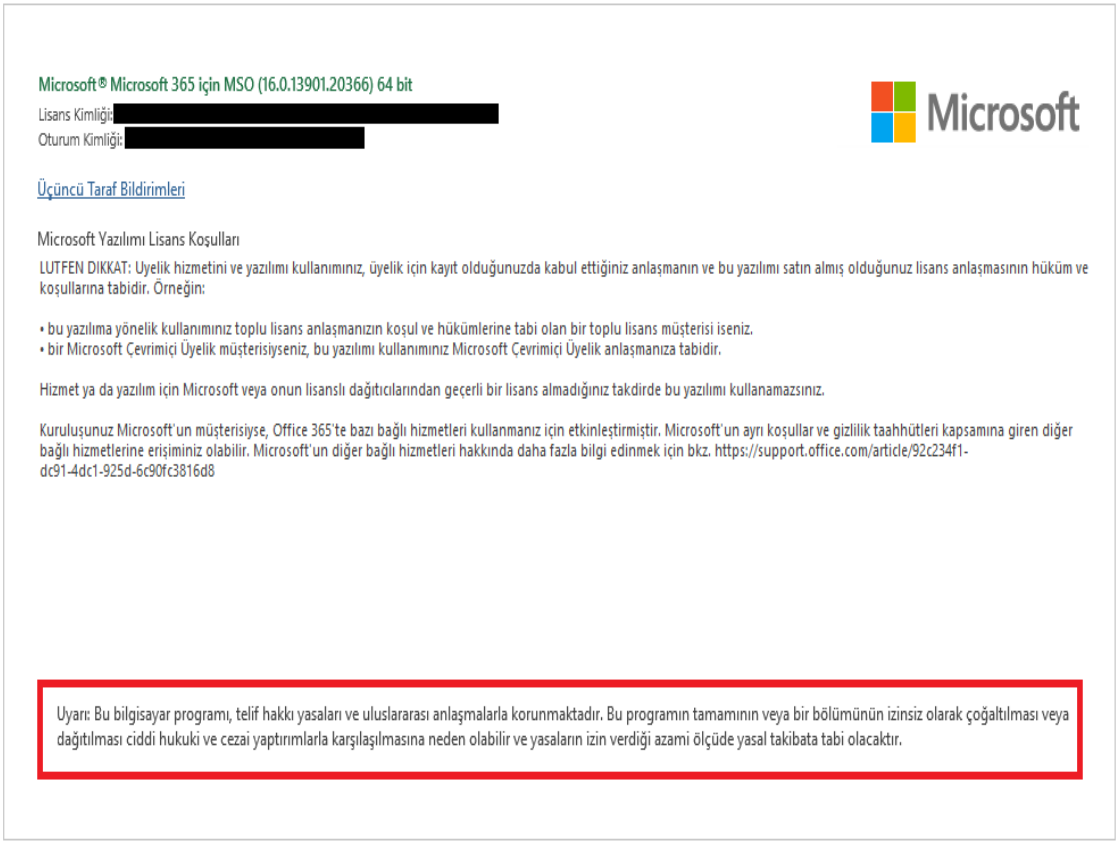
Ek olarak GNU Genel Kamu Lisansları dünya çapında kabul edilen tek lisans setine ve bildirimine sahip olmasına karşın, Microsoft gibi bazı ticari/tescilli yazılım geliştiricileri, kullanıcılara sağladığı yazılım lisanslarını coğrafi olarak da sınırlandırma yolunu tercih edebiliyor. Şekil 1.2'de ilgili yazılım setinin coğrafi sınırlandırmaya tabi olabileceği belirtiliyor.

E. GEOGRAPHIC AND EXPORT RESTRICTIONS

You may activate the service/software only in the region in which it was acquired. Microsoft's software and services are subject to U.S. and other jurisdictions' export and technology laws and you agree to comply with all such applicable laws and regulations that apply to the software and/or services.

Şekil 1.2: EULA Lisansının Coğrafi Sınırlandırma Örneği

Ayrıca Microsoft gibi büyük kuruluşlar, MEA – Microsoft Kurumsal Sözleşmesi altında toplu lisanslama paketi sunmasına rağmen; sadece anlaşma yapılan kuruluşu değil aynı zamanda bu toplu lisanslama paketi kapsamındaki yazılımların kopyalarını kullanan son kullanıcılara da yasal sorumluluklar yükleyebilmektedir.



Şekil 1.3: Microsoft'un Enterprise Sürümü için Gösterdiği Lisans Bildirimi

1.3. ÖZGÜR VE AÇIK KAYNAK LİSANSLAR

Özgür ve Açık Kaynak Lisanslar, bir yazılımın belirli şartlar altında kullanılmasına, kaynak kodlarının incelenip değiştirilmesine ve üçüncü taraflara dağıtılıp paylaşılmasına izin veren lisanslardır. Buradaki bahsedilen şartlar, kullanıcının yazılım üzerindeki tasarruflarını sınırlayıcı tedbirler içermez.

Açık kaynak lisanslar da ÖAK olmayan yazılımlar gibi belirli hüküm ve koşulları şart koşabilir. Ancak bu şartlar yazılımın özgür ve açık kaynak olmasını engelleyecek hususlar içeremez. Genelde değiştirilen yazılımın

yeniden açık kaynak olarak dağıtımını konu alan lisans maddeleri ile geliştiricinin sorumluluk reddi beyanını içerir.

Yine de Açık Kaynak Lisanslar, ÖAKY ilkelerine aykırılık barındırmadan geliştiricinin inisiyatifinde bazı kısıtlamalara sahip olabilir. Örneğin geliştiricilerin isimleri ve notlarının lisans bildirimine dahil edilmesi, ya da orijinal yazılımın değiştirilen sürümlerinin lisanslarının da copyleft olarak devam ettirilmesi gibi. Ancak açık kaynak lisansların dikkate alınması ve bunların yasal takibinin yapılması oldukça zor. Hatta günümüzde bu lisans bildirimlerinin yasal geçerliliği de tartışılmaktadır.

Açık kaynak lisanslarda yazılımın ücretlendirilmesi konusu ise geliştiricilerin tercihine bırakılmıştır. Ancak günümüzde kullanılan ÖAK yazılımların büyük bölümü bir ücretlendirmeye tabi değildir. Geliştiriciler ÖAKY lisans şartlarına uygun olarak yazılımlarını yayınlayabildikleri gibi kendi özgür ve açık kaynak lisans setlerini de oluşturabiliyor.

Örneğin Debian dağıtımı, Debian projesi altında geliştirilen yazılımları kendi belirlediği lisans bildirimine (DFSG) uygun olarak yayınlamaktadır. Debian Özgür Yazılım Kılavuzu (DFSG), bir yazılım lisansının özgür bir yazılım lisansı olup olmadığını belirlemek ve daha sonra bu yazılımın Debian Projesine dahil edilip edilemeyeceğine karar vermek için kullanılan bir kılavuzdur.[29]

Günümüzde ise yüzlerce ÖAKY lisansı bulunmaktadır. En yaygın kullanılanları ise GNU GPL, BSD ve Mozilla lisanslarıdır.

[29] Bruce Perens, (1997). *"Debian's "Social Contract" with the Free Software Community"*, Debian-Announce Mailing List, https://lists.debian.org/debian-announce/1997/msg00017.html (01.02.2021)

1.3.1. GNU Lisansları ve Copyleft

GNU; Genel Kamu Lisansı (GPL), Kısıtlı Genel Kamu Lisansı (LGPL), Affero Genel Kamu Lisansı (AGPL), Özgür Belgelendirme Lisansı (GNU FDL) gibi çeşitli lisans setlerine sahiptir.[30] Ancak GNU kapsamında geliştirilen yazılımların büyük bir bölümü GPL lisansı altında yayınlanmaktadır.

GNU GPL son kullanıcıya, yazılımı çalıştırma, inceleme, değiştirme ve yeniden dağıtabilme özgürlüğünü garanti eder. Yaygın olarak kullanılan bir özgür yazılım lisansı olup tamamı copyleft lisanslardır.

GNU GPL, özgür yazılımlarla ilgili lisans süreçlerini doğru yönetebilmek adına günümüze kadar sürekli güncellenmiştir. Ancak GNU GPL, geliştiriciye GPLv2'den sonraki sürümlerin kullanılmasını şart koşmaz. GNU GPL lisansının ilk sürümü olan GNU GPLv1 1989'da yayınlanmış olup bu sürümde iki önemli düzenleme yapılmıştır. İlki Özgür Yazılım ruhunun temel prensibi olan kullanıcıların kaynak kodunu görebilme özgürlüğü ile ilgiliydi. Buna göre GNU GPL lisansı altında yayınlanan yazılımların sadece binary formatta yayınlaması engellenmişti.[30] Böylece binary dosyaların yanında kullanıcıların anlayabileceği bir formatta yazılımın kaynak kodlarının da yayınlanması şart koşulmuştu. İkinci önemli düzenleme ise geliştiricinin sahipli yazılımlarla Özgür Yazılımları bir arada kullanarak yazılımın dağıtımı konusunda kısıtlamalar getirmesini engellemeye yönelikti. Bunu engellemek için, GPLv1, değiştirilmiş sürümleri bir bütün olarak ele alır ve yazılımın bütün parçalarının GPL lisansına uygun olmasını şart koşar. Yani, GPLv1 lisansı altında dağıtılan yazılım, yine GPLv1 lisansına uygun olan yazılımlarla birleştirilebilir. Bununla birlikte, GPLv1 kapsamında yayınlanan yazılım, GPL ilkeleriyle çelişen bir lisans altında dağıtılan yazılımlın parçalarıyla da birleştirilemez.

[30] Free Software Foundation Inc. *"Licences"*, GNU Projesi, https://www.gnu.org/licenses/licenses.en.html (02.02.2021)

GNU GPL'nin ikinci sürümü GPLv2 1991 yılında yayınlanmıştır. Bu sürümdeki en önemli değişiklik ise lisansın bir bütün olarak ele alınması ile ilgiliydi. Yani GPL lisansı altında yayınlanan yazılımın lisansın tüm şartlarını sağlaması gerekiyordu. Lisansın tüm şartları yerine birkaçını sağlayan yazılım GPL lisansı altında yayınlanamayacaktı. GNU GPL'nin üçüncü ve son sürümü GPLv3 2007 yılında yayınlandı. Bu sürümde genellikle yazılım patenti, lisans ihlalleri, telif hakları ve bunların Özgür Yazılımlara nasıl entegre edileceği ele alınmıştır.

Linux'a ilginin giderek artması ve yaygın kullanımı ile beraber, ÖAK olmayan yazılımların Linux'a eklenmesi konusunda yoğun baskı oluştu. Ancak Linux, GPL lisansı altında yayınlanmıştı ve ÖAK olmayan yazılımların, uygulamaların veya kodların Linux'a entegre edilmesi mümkün görünmüyordu. Bunun sonucunda Kısıtlı Genel Kamu Lisansı (LGPL) yayınlandı. LGPL, ÖAK olmayan bir kod bloğunun GNU C kütüphanesi glibc'ye bağlanmasına izin veriyordu. Böylece LGPL lisansı altında yayınlanan kütüphanenin kaynak kodlarını yayınlamaya gerek kalmıyordu.

Bir yazılımı GPL lisansı altında yayınlamanın avantajları veya dezavantajları olabilir. Yayınlanacak yazılımın hangi amaçla kullanılacağı, hedef kitlesi ve ücretlendirme konusu geliştiricinin göz önünde bulundurması gereken hususlardır. GPL lisansının sunacağı avantajlar veya dezavantajlar, kullanıcıya veya geliştiriciye göre farklılıklar gösterebilir;

- Yazılımı dağıtmak, desteklemek veya dokümante etmek için ücret talep edilebilir, ancak yazılımın kendisini satılamaz. Bu durum kullanıcıya ekonomik anlamda avantaj sağlayıp geliştiriciye de özgürlük sunar ancak yazılımın ticari değerini de kısıtlar.

- GPL lisansı altında yayınlanan bir yazılımın tüm parçaları, araçları ve kaynak kodları gibi GPL lisansına tabi olmalıdır. Bu durum ÖAK olmayan yazılımların veya kodların GPL yazılımlara eklenmesini engeller. Bu durum GPL lisansına tabi yazılımların özgürce dağıtımı ve paylaşımını mümkün kılarken GPL lisanslı yazılımların yaygın olarak kullanılan bazı araç ve yazılım setlerinden mahrum kalmasına sebep olur. Her ne kadar LGPL ile bu sorun kısmen çözülmüş olsa da GPL için bu engel hala mevcut.

GNU GPL kapsamında lisanslanan yazılımlar aynı zamanda copyleft lisanslardır. Copyleft, ilkeler bakımından olduğu gibi sembol, slogan ve kelime itibari ile de Copyright'ın tam tersi konsepte sahiptir. Copyleft, bir programın (veya başka bir çalışmanın) özgür yazılım haline getirilmesi, programın tüm değiştirilmiş ve geliştirilmiş sürümlerinin de özgür yazılım olması için genel bir yöntemdir.[31] Copyleft, program üzerinde telif hakkını kullanmanın bir yoldur. Telif hakkından vazgeçmek anlamına gelmez; gerçekte, telif hakkından vazgeçmek copyleft'i imkansızlaştıracaktır.

"Copyleft" kelimesindeki "left" (sol); İngilizce'deki "to leave" (terketmek, bırakmak) fiiline bir atıf değildir; sadece "right" (sağ) yönünün tersini temsil etmektedir.[31] Copyleft; Özgür ve Açık Kaynak bir yazılımın tamamen kamu malı kapsamına girmesini ya da herhangi bir telif hakkına tabi olmayan anonim değere dönüşmesini engeller. Ayrıca ÖAKY bağlamından kopmadan geliştiricisine yazılımın kullanımına belirli kısıtlamalar getirmesine de imkân verir.[32] Copyleft tarafından yazılımın dağıtım metoduna getirilen ana kısıtlamalardan biri, yazılımın değiştirilen sürümlerinin de uyumlu bir copyleft lisansı kapsamında dağıtılmasıdır.

---

[31] Free Software Foundation Inc. *"What is Copyleft?"*, GNU Projesi, https://www.gnu.org/licenses/copyleft.en.html (08.02.2021)

[32] John C. Newman, (2011). *"Copyright and Open Access at the Bedside"*, New England Journal of Medicine, ABD.

1.3.2. BSD Lisansları

GNU GPL'den sonra ÖAKY dünyasında en yaygın kullanılan ikinci lisans seti BSD lisansıdır. BSD, yazılımın nasıl kullanılacağına, değiştirilebileceğine ve yeniden dağıtılacağına ilişkin çok az kısıtlamalara sahip, genellikle garanti ve sorumluluk reddi beyanı içeren bir ÖAKY lisansıdır. Günümüzde kullanılan BSD lisansları, ilk yayınlanan orijinal BSD lisanslarından farklıdır.

BSD lisansları, genellikle BSD lisans bildirimlerini BSD lisansı altında yayımlanan yazılımların tüm orijinal ve değiştirilmiş sürümlerine eklenmesini gerektirir. Ayrıca bazı BSD lisansları, GNU GPL lisansının aksine, yazılımın kaynak kodunun binary dosyalar ile birlikte yayınlanmasını şart koşmaz. Bu sebeple bu tarz BSD lisansları altında yayınlanan yazılımların ÖAKY olacağı tartışmalıdır.

BSD lisansı tek bir lisans setine sahip değildir. Bu sebeple bu başlık altında da BSD lisansları diye bahsedilmiştir. BSD lisansları lisans maddelerine göre farklı gruplara ayrılmaktadır: 4 maddeli BSD lisansı, 3 maddeli BSD lisansı, 2 maddeli BSD lisansı vs.

4 maddeli BSD lisansı, ilk kullanılan orijinal BSD lisansı idi. Bu lisansa tabi yayımlanan tüm yazılımların reklam ve tanıtım materyallerinde telif hakkı sahibinin belirtilmesi şartı vardı. Daha sonra bu şart kaldırıldı ve 1999'da 3 maddeli BSD lisansı yayımlandı. ÖAKY standartlarına en çok uyduğu kabul edilen bu lisans aynı zamanda Yeni BSD lisansı olarak kabul edilmektedir. Daha sonra orijinal geliştiricinin adının, izni olmadan kullanılmamasını belirtilen madde çıkarılarak 2 maddeli BSD lisansı yayınlanmıştır. Böylece lisans seti daha da basite indirgendi. Bu lisans setinin de ÖAKY standartlarına uyduğu

kabul edilmektedir. Ayrıca FreeBSD ve NetBSD de bu 2 maddeli BSD lisansından türettikleri lisansları kullanmaktadırlar. Yeni BSD lisansı genel olarak yazılımlar ve kaynak kodu ile ne yapılacağına müdahale etmez. Bu anlamda geniş özgürlük sunmuş olur. Ancak yazılımın kullanılması ile ilgili doğabilecek hasarlardan da kimsenin sorumlu tutulamayacağı ve geliştiricilerin bu konuda bir yükümlülüğünün bulunmadığı bir lisanslama yöntemidir. Bu anlamda BSD lisansları son kullanıcılara ve geliştiricilere hem avantaj hem dezavantaj sunabilir;

- BSD lisansının GPL'ye karşı en büyük kozu hiç şüphesiz ticari yazılım dünyasına daha uyarlanabilir olmasıdır. GNU GPL, copyleft olan GPL lisansına tabi bir yazılımın değiştirilmiş sürümlerinin de yine copyleft olarak dağıtılması şartı BSD için bir parametre değildir. Böylece BSD lisansı altında dağıtılan bir yazılım, değiştirilip ticari ürün olarak ve minimum yasal yükümlülükle pazarlanabilir.

- BSD'nin en önemli dezavantajı ise; BSD lisanslı yazılımların geliştiricisi ve sahibi belli olmasına rağmen, anonim ve "Public Domain" yazılım olarak algılanabilmesidir. Bu durum BSD lisansı altında yazılımını yayınlamayı düşünen geliştiricilerin motivasyonunu olumsuz etkilemektedir.

- BSD ile geliştirici kaynak kodlarını yayınlamak zorunda değildir, sadece binary dosyaları yayınlayabilir. Bu anlamda BSD'nin GPL'ye en çok ters düştüğü nokta budur. Bu durum geliştiricinin kodlarını ticari bir sır olarak değerlendirmesine imkân tanır.

- Binary dosyalara sahip herhangi bir kullanıcı, hiçbir yasal engele takılmadan, geliştirici kadar dağıtma ve kopyalama hakkına sahip olacaktır.

- Kullanıcı, ihtiyaçları doğrultusunda ilerde gerçekleştirmeyi planladığı tüm geliştirmelerde BSD lisanslı kodu ya da yazılımı diğer sahipli kodlara entegre edip çalıştırabilir. Sistemin tamamını yeniden tasarlaması gerekmez. Böylece maliyetleri düşürmüş olur.

1.3.3. Mozilla Kamu Lisansı

Mozilla Kamu Lisansı (MPL), Mozilla vakfı tarafından geliştirilen özgür ve açık kaynaklı bir yazılım lisansı olup şu anda MPL 2.0 sürümü kullanılmaktadır. MPL; copyleft, GPL ve BSD lisanslarının ortasında konumlanmıştır. Yani son kullanıcı, MPL altında yayınlanmış bir yazılımın kodlarını aynı GPL'de olduğu gibi; kullanabilir, görebilir, değiştirebilir ve yeniden dağıtabilir. Hatta kullanıcı isterse belirli şartlar altında MPL lisansı altında olan orijinal bir sürümü değiştirerek MPL ile uyumsuz bir lisans altında tekrar yayınlayabilir. [33] Ayrıca BSD gibi, GPL'nin aksine MPL lisanslı kodlara ÖAK olmayan yazılım kodları entegre edilebilir. Bu sebeple MPL ile, GPL ve BSD'nin sunduğu avantajlardan birlikte yararlanmak mümkündür. [34]

MPL, kodun BSD Lisansı gibi özel projelerde kullanılmasına izin verir, ancak çekirdek dosyaların MPL altında kalmasını sağlar. Çekirdek dosyaların MPL altında kalması şartı, geliştiricilere çekirdek işlevselliğinin özelliklerini iyileştirme ve geliştirme yönünde teşvik eder.[34] MPL ve GPL'nin ayrıştığı en önemli nokta, GPL'nin kısmen GPL kodundan türetilen yazılımların değiştirilmiş sürümlerini dahi GPL altında tutma konusunda daha katı olmasıdır. Ancak MPL, MPL kodlarının ya da yazılımların ÖAK olmayan kodlarla ve yazılımlarla entegre edilmesini mümkün kılar.[34]

---

[33] Mozilla Foundation. *"Mozilla Public License, Version 2.0"*, https://www.mozilla.org/en-US/MPL/2.0/ (13.02.2021)

[34] Keith J. O'Hara ve Jennifer S. Kay, (2003*). "Open Source Software and Computer Science Education"*, Rowan University, ABD.

İKİNCİ BÖLÜM

ÖZGÜR / AÇIK KAYNAK YAZILIMLARA GÖÇ

2.1. DÜNYADA ÖAK YAZILIMLARIN KURUMSAL KULLANIMI

ÖAK yazılımlar, ÖAK olmayan diğer yazılımlar gibi hem dünyada hem de Türkiye'de yaygın olarak kullanılmaktadır. Şirketler, devlet kurumları ve bireysel kullanıcılar, başta ekonomik gerekçeler olmak üzere sistemlerinde güvenlik ve esnekliği sağlamak için her geçen gün daha fazla ÖAK yazılımları tercih etmektedirler.

2.1.1. Avrupa 'da ÖAK Yazılımların Kurumsal Kullanımı

Dünyanın dört bir yanından hükümetler, ÖAK yazılımların geliştirilmesi yönünde politikalar gütmeye başlamıştır. ÖAKY çözümlerini öneren çok sayıda rapor ve teknik incelemenin yanı sıra, dünya çapında ÖAK yazılımları zorunlu kılan veya teşvik eden yasa önerileri veya raporlar da mevcuttur. Özellikle Kuzey Avrupa ülkeleri başta olmak üzere Avrupa Birliği'nin genelinde ÖAK yazılımları teşvik edici birçok düzenleme ve yasa mevcuttur.

Avrupa Komisyonu, 21 Ekim 2020 tarihinde, 2020-2023 dönemi için Açık Kaynak Yazılım Stratejisini onaylamıştır. Buna göre bilişim alanı dışındaki alanlarda da açık kaynak yazılım kullanımını artırmayı hedefleyen bir strateji belirlendi.[35] Komisyon, belirlediği bu strateji kapsamında; diğer Avrupa kamu sektörü kuruluşlarıyla birlikte, açık kaynak yazılım çözümleri ile bu konudaki bilgi ve uzmanlığın paylaşılmasını teşvik ederek, toplumu zenginleştiren ve maliyetleri azaltan daha iyi hizmetler sunmayı taahhüt etmiştir.[35]

Avrupa Komisyonu bu strateji belgesinde; Açık kaynak yazılımların, bilgi ve becerilerin paylaşımına dayalı yeniliğe izin verdiği için kamu hizmetlerine olan güveni artıracağını belirtmiştir.[35] Beklenmedik sorunların çözümlerine herkesin katkıda bulunmasını sağlayarak karmaşık teknolojik sorunların çözülmesine yardımcı olunacağı; inceleme özgürlüğüne sahip olan açık kaynak yazılımların güvenliği artırmak için de daha fazla seçenek sunacağı belirtilmiştir. Bağımsız denetimlere ve kod incelemelerine izin verdiği için de harcanan zaman ve çabanın ihtiyaca göre dengelenebilir olduğu ifade edilmiştir.[35]

Belgede, Komisyonun büyük ölçüde açık kaynak ilkelerine dayanan çalışma kültürüne katkıda bulunacağı ifade etmiştir.[35] Temel strateji olarak ise şu ifade kullanılmıştır: *"Açık kaynak kod, algoritmaları şeffaf hale getirir ve bağımsız denetimlere ve yeniden üretilebilir yapılara izin verir. Strateji, uzantı olarak, Komisyonun süreçlerini, verilerini, bilgilerini ve teknolojisini güvende ve kontrol altında tutmasına yardımcı olur."*[35] Ayrıca belgede; ÖAK yazılımların kodlarının herkes tarafından kullanılabilir ve değiştirilebilir olmasının; yapay zekâ ve makine öğrenimi modellerine erişim için birlikte çalışabilen, güvenli ve şeffaf prosedürler oluşturulmasına da katkı sunacağı belirtilmiştir.

[35] European Commission, (2020). *"Communication To The Commission: Introduction"*, s.3, Open Source Software Strategy 2020 – 2023, Brussels.

Avrupa Komisyonu bu Açık Kaynak stratejisi ile sağlanacak asıl faydayı şöyle özetlemiştir:[36] *"Açık Kaynak Yazılımların, Avrupa'nın dijital bağımsızlığına önemli etkisi olacaktır. Açık kaynak Avrupa'ya kendi bağımsız dijital yaklaşımını oluşturma ve sürdürme imkânı ile bilgisini ve teknolojisini kontrol etme şansı verebilir."* Komisyon sağlanacak bu faydayı şu gerekçelere bağlamıştır:[36]

- ÖAK Yazılımlar, şirketlerden ve ülkelerden bağımsızdır. Böylece satıcıya bağlanma ile siyasi ve ticari risklere maruz kalma ihtimalini en aza indirir.

- ÖAKY, önemli bir modeldir ve doğası gereği blockchain, yüksek performanslı bilgi işlem ve yapay zekadan nesnelerin internetine kadar birçok alanda yazılım geliştirme için en esnek platformu sağlar.

- Avrupa, ÖAK yazılımlardan yararlanarak, bulut bilişim ve hizmet yazılımı konularında avantajlarını ve risklerini dengeleyen bir yaklaşım geliştirebilir.

- Açık olmak, Birliğe ve kurumlarına olan güveni arttırır.

- ÖAKY kullanarak Avrupa, bu teknoloji alanında kendi bağımsız ve rekabetçi konumuna yatırım yapmış olur. Böylece Avrupalı aktörleri ve mali adaleti de destekleyecek açık, merkezi olmayan bir bilgi toplumuna odaklanabilir.

- Açık kaynak, bağımsız olarak denetlenebilen merkezi olmayan, birleşik hizmetleri mümkün kılar. Böylece Avrupa'nın, Avrupa'yla uyumlu çözümler üretmesini mümkün kılar.

[36] European Commission, (2020*). "Communication To The Commission: Progress Towards Digital Autonomy"*, s.5-6, Open Source Software Strategy 2020 – 2023, Brussels.

Açık Kaynak Yazılımların kullanımı bu strateji belgesinden önce de Avrupalı devletler tarafından yaygın olarak kullanılmakta idi. Örneğin Almanya'da Alman Federal Meclisi bilgi işlem altyapısındaki sunucularında Linux kullanmakta iken Alman Maliye Bakanlığı ise kurum içi ağında Linux/Apache temelli açık kaynak sistemler kullanmaktadır.[37] Yine Alman İçişleri Bakanlığı, Açık Kaynak Yazılım için göç ve entegrasyon yönergelerini içeren bir yazılım geçiş kılavuzu yayınlamıştır.[37]

Diğer bir Avrupa ülkesi Fransa'da Savunma, Kültür ve Ekonomi Bakanlıkları ÖAKY işletim sistemlerini kullanmaktadır. Savunma Bakanlığı 2004 yılında, güvenlik gerekçesiyle Linux tabanlı açık kaynak bir işletim sistemi geliştirmek için konsorsiyum oluşturmuştur.[38]

Hollanda ise çok daha erken dönemde; 2003 yılında, Kamu yönetiminde Açık Kaynak Standartların ve ÖAK yazılımların tanıtımı için bir eylem planı oluşturmuştur.[38] Yine aynı yıl, Hollanda hükümeti, Hollanda devlet kurumlarını yazılımlarında açık kaynak standartları kullanmaya teşvik etmek ve onları açık kaynaklı yazılım hakkında bilgilendirmek için Açık Standartlar ve Açık Kaynak Yazılım (OSSOS) adlı bir program başlatmıştır. Hollanda İçişleri Bakanlığı, 2007 yılında, iki yıl içinde tüm devlet bakanlıklarında açık kaynaklı yazılımın kullanılması çağrısında bulunmuştur. ÖAK olmayan yazılımların kullanımında hâlâ ısrar eden tüm kurumların ise, bu ısrarlarını gerekçelendirmelerini istemiştir.[38]

---

[37] Kenneth Wong ve Phet Sayo, (2004). *"Free/Open Source Software: A General Introduction: FOSS Success Stories"*, s.19-20, United Nations Development Programme's Asia-Pacific Development Information Programme (UNDP-APDIP), Malaysia.

[38] James A. Lewis, (2008). *"Government Open Source Policies: National Open Source Policies"*, Center for Strategic and International Studies CSIS, ABD.

2.1.2. Avrupa Dışında ÖAK Yazılımların Kurumsal Kullanımı

Avrupa dışındaki ülkeler de açık kaynak yazılımların kullanımının teşvik edilmesi yönünde kararlar almıştır. Bu konuda en az istekli ülke ise ABD'dir. Ancak ABD dışında, Güney Amerika, Asya ve Asya Pasifik ülkelerinde ÖAK yazılımlara olan kurumsal ilgi her geçen gün artmaktadır.

Brezilya hükümeti 2005 yılında, 300.000 devlet bilgisayarını Microsoft Windows işletim sisteminden Linux gibi açık kaynaklı yazılım işletim sistemlerine geçireceğini açıkladı. Ayrıca dönemin devlet başkanı Lula da Silva, onaylanması halinde federal devlet kurumlarında ÖAK yazılımların kullanımını zorunlu hale getirecek bir kararname tasarısı üzerinde çalıştıklarını belirtmiştir.[39]

Rusya Federasyonu, 2007 yılında, 2009 yılı sonuna kadar tüm okulların bilgisayarlarına yerel olarak geliştirilen GNU / Linux açık kaynak yazılımları kurarak yabancı ticari yazılıma olan bağımlılığın azaltılmasını öngören planı onaylamıştır.[40]

Asya pasifik ülkeleri Japonya, Çin ve Güney Kore gibi ülkeler de açık kaynak yazılımların devlet kurumlarında kullanımını düzenleyen birçok rapor ve plan yayınlamıştır. Ayrıca ÖAK yazılımların okullarda öğretilmesini teşvik edici planalar hazırlamışlardır.

[39] Steve Kingstone, (2005). *"Brazil Adopts Open-Source Software"*, BBC News, Sao Paulo, Brezilya. http://news.bbc.co.uk/2/hi/business/4602325.stm (17.02.2021)
[40] James A. Lewis, (2008). *"Government Open Source Policies: National Open Source Policies"*, Center for Strategic and International Studies CSIS, ABD.

Güney Kore'de İletişim Bakanlığı, ÖAK olmayan yazılımlardan ÖAK yazılımlara geçiş yapan kurumlara toplam 3 milyar Won teklif ederek hükümette açık kaynak kullanımını teşvik edecek bir planı 2005 yılında onaylamıştır.[41] Ayrıca Eğitim Bakanlığı, tamamen açık kaynak modeli üzerine inşa edilen Linux temelli Ulusal Eğitim Bilgi Sistemi Projesini 2006 yılında başlatmıştır.[41]

Japonya'da İçişleri ve İletişim Bakanlıkları, Microsoft ürünlerine bağımlılığını azaltmak için kritik hükümet sistemlerini Linux gibi açık kaynak sistemlere geçirme planlarını 2005'te duyurmuştur.[41] Ayrıca Japonya, Microsoft Windows yazılımındaki güvenlik sorunları ve işletim maliyetleri sebebiyle e-devlet ve e-bordro sistemlerini GNU/Linux temelli açık kaynak platformlara taşımayı düşündüğünü belirtmiştir.[42]

Dünyanın dört bir yanından devlet kurumları sistemlerinde açık kaynak yazılımları tercih ettikleri gibi belediyeler gibi yerel yönetimler de bilgi sistem altyapılarında ÖAK yazılımları tercih etmektedirler. Brezilya, Avusturalya, Belçika, Almanya, İtalya, Tayland, Çin gibi ülkelerdeki yerel belediyeler açık kaynak sistemleri tercih etmişlerdir. Hatta federal hükümet olarak ÖAK yazılımlara mesafeli olan ABD'de de Oklahoma, Oregon, Massachusetts gibi eyaletlerdeki yerel yönetimlerde açık kaynak kullanımını teşvik edici planlar yayınlanmıştır.

---

[41] James A. Lewis, (2008). *"Government Open Source Policies: National Open Source Policies"*, Center for Strategic and International Studies CSIS, ABD.
[42] Kenneth Wong ve Phet Sayo, (2004*). "Free/Open Source Software: A General Introduction: FOSS Success Stories: Japan"*, s.24, United Nations Development Programme's Asia-Pacific Development Information Programme (UNDP-APDIP), Malaysia.

2.1.3. Dünyada Kurumsal Açık Kaynak Girişimlerinin Başarı Durumu

Hem Avrupa'da hem de Dünya genelinde ÖAK sistemlere geçişle ilgili teşvik edici raporlar, planlar ve projeler her geçen gün artmaktadır. Ancak bu girişimlerin tamamının başarılı olduğunu söylemek pek mümkün değil.

Ticari/tescilli ürünler geliştiren büyük yazılım şirketlerinin, siyasi ve diplomatik kanallardan yönetimleri veya kurumları etkilemesiyle, göç projeleri başarısızlıkla sonuçlanabiliyor ya da rafa kaldırılabiliyor.

Bazen de maddi sebepler veya kötü planlama sonucu da bu girişimler başarısız olabiliyor. Bu kapsamda, Stratejik ve Uluslararası Çalışmalar Merkezi CSIS adlı düşünce kuruluşunun 2008 yılını kapsayan; dünya genelindeki devlet kurumlarında ya da belediyelerde açık kaynak kullanımını teşvik eden planlar, kararlar, yasal düzenlemeler ve projelere ait girişim sayısı ile bunların başarı durumlarını gösteren sayısal verilere Tablo 2.1, 2.2, 2.3 ve 2.4'ten ulaşılabilir.

Tablo 2.1: Kurumsal Açık Kaynak Girişimlerinin Yıllara Göre Dağılımı [43]

Yıllara Göre Açık Kaynak Girişimi (Dünya Geneli)				
	2004	**2005**	**2007**	**2008**
Tavsiye Niteliğinde	%37,2	%34,0	%31,6	%30,7
Tercihe Bırakılan	%31,0	%31,0	%31,6	%30,7
Ar-Ge	%30,2	%33,8	%33,4	%32,9
Zorunlu	%1,6	%1,3	%3,4	%5,5

[43] *James A. Lewis, (2008). "Government Open Source Policies", s.2, CSIS, ABD.*

Tablo 2.1'den de görüldüğü üzere, Zorunlu tutulan açık kaynak girişimleri / geçişleri yıllar içerisinde önemli artış göstermiştir. Özellikle Avrupa ülkelerinin bu konuda daha teşvik edici ve zorlayıcı tedbirler almasının bu durumun oluşmasında etkili olduğu söylenebilir.

Tablo 2.2: Kurumsal Açık Kaynak Girişimi Sayısı [44]

Açık Kaynak Girişim Sayısı (Dünya Geneli, 2008)				
	Onaya Sunulan	Onaylanan	Başarısız	**Toplam**
Tavsiye Niteliğinde	16	56	3	**75**
Tercihe Bırakılan	35	56	4	**95**
Ar-Ge	7	60	1	**68**
Zorunlu	18	10	9	**37**
Toplam	**76**	**182**	**17**	**275**

Tabla 2.2'de, Zorunlu Tutulan Açık Kaynak Girişimlerinin[45] büyük bölümünün başarısızlıkla sonuçlandığı görülmektedir. Bu sebeple kurumsal zorunluluk getirmenin başarıyı garantilemediği söylenebilir. Ayrıca 2008 yılındaki açık kaynak girişimlerinin %91'lik gibi yüksek bir oranının başarıya ulaştığı görülebilmektedir.

Tablo 2.3 ve 2.4'te kurumsal açık kaynak girişimlerinin ulusal/yerel ile Dünya genelindeki bölgesel dağılımları görülebilir. Tablo 2.4'te, açık kaynak girişimlerde Avrupa'nın büyük bir farkla önde olduğu görülebilir.

[44] James A. Lewis, (2008). *"Government Open Source Policies"*, s.2, CSIS, ABD.
[45] Açık Kaynak Girişimi: İlgili ülkenin ya da kurumun, açık kaynak kullanımı ile ilgili oluşturduğu plan, proje, teşvik, rapor ya da yasal düzenlemeler ile başarılı göçleri ifade eder.

Tablo 2.3: Kurumsal Açık Kaynak Girişimlerinin Ulusal Kurumlar ile Yerel Yönetimler Arasındaki Dağılımı [46]

Ulusal ve Yerel Açık Kaynak Girişimleri (Dünya Geneli, 2008)		
	Ulusal	Yerel
Tavsiye Niteliğinde	62	13
Tercihe Bırakılan	47	48
Ar-Ge	51	17
Zorunlu	26	11
Toplam	**186**	**89**
Toplam Girişim: 275		

Tablo 2.4: Kurumsal Açık Kaynak Girişimlerinin Bölgesel Dağılımı [47]

Kurumsal Açık Kaynak Girişimlerinin Bölgesel Dağılımı, 2008				
	Onaya S. (Sunulan)	Onaylanan	Başarısız	**Toplam**
Avrupa	33	95	7	**135**
Asya	22	47	1	**70**
Latin Amerika	13	20	4	**37**
Kuzey Amerika	6	9	4	**19**
Afrika	1	4	0	**5**
Orta Doğu	1	7	0	**8**

[46] James A. Lewis, (2008). *"Government Open Source Policies"*, s.2, CSIS, ABD.
[47] James A. Lewis, (2008). *"Government Open Source Policies"*, s.2, CSIS, ABD.

2.2. ÖZGÜR VE AÇIK KAYNAK YAZILIMLARA GÖÇ

Göç, kuruluşların bilgi işlem altyapılarında ve hizmetleri sağlama biçiminde birçok değişikliğe neden olur. Açık kaynak göçünde kurulum maliyeti ya yoktur ya da çok azdır. Burada asıl sorun nereden ve ne ölçüde destek alınacağıdır. Açık kaynak yazılım dağıtıcılarının yanı sıra üçüncü taraflardan da destek alınabilir. Altyapısında açık kaynak göçü planlayan kuruluş, herhangi bir ÖAK yazılımla ilgili işler ters gittiğinde bir yasal sorumlu aramayı düşünüyorsa açık kaynak bir sistem o kuruluş için doğru seçenek olmayabilir. Bu bağlamda, ÖAK yazılımların dinamiklerinin iyi anlaşılması ve ona göre bir projeksiyon çizilmesi gerekir.

ÖAK yazılımların ekonomik avantajlar sağlaması ve ÖAK olmayan yazılımlara göre daha esnek bir alternatif olması, açık kaynağa göçün motivasyonunu oluşturmaktadır. Ancak muhtemel bir göçün başarıya ulaşması, ÖAK yazılımların ne kadar benimsendiği, kurumsal iş ihtiyaçlarını ne ölçüde karşıladığı, yazılımın kalitesi ve özellikleri, yazılım desteğinin sürdürülebilirliği ve kullanılabilirliği, ekonomik fayda ve son kullanıcıların ÖAK yazılımlara yönelik tutumu gibi kritik faktörlere bağlıdır.

Açık kaynak göçünün başarıyla gerçekleştirilmesi de tek başına yeterli değildir. Sistemin sürdürülebilirliğinin sağlanamadığı durumlarda ters göç ihtimali vardır. Açık kaynak sistemle ilgili yeterli teknik destek ve eğitimlerin sağlanamaması, yüksek bakım ve ek maliyetlerin oluşması, yazılımların yeterli güvenliği ve kaliteyi sağlayamaması, son kullanıcıların ihtiyaçlarının yeterli düzeyde karşılanamaması gibi sebeplerle göç girişimi başarısızlıkla sonuçlanabilir.

2.3. ÖAK YAZILIMLARA GÖÇÜN AVANTAJLARI

ÖAK yazılımlara göç, getirdiği birçok faydanın yanında bazı dezavantajları da bünyesinde barındırabilir. Geçişi planlayan veri sahibi taraf, ihtiyaçlarını, beklentilerini ve hedeflerini tam olarak belirlediği takdirde muhtemel bir göçün dezavantajlarından kaçınabilir.

2.3.1. Ekonomik Fayda ve Düşük Maliyet

ÖAK yazılımların büyük bir bölümü ücretsizdir. ÖAK yazılımların dağıtıcıları bile genellikle yazılım paketlerinin kaynak kodunu içeren ücretsiz sürümlere sahiptir. Çoğu açık kaynak yazılım, özellikle işletim sistemleri "live" olarak mevcut olup bu sayede zaman/kurulum maliyetlerinden de tasarruf edilmesi mümkündür. Ayrıca özel mülk yazılımlar için ayrılması gereken bütçe, ÖAK yazılımlar tercih edildiği takdirde, diğer donanım ve araçlar için harcanabilir.

Özellikle gelişmekte olan ülkelerin ÖAKY sistemlerini benimsemeleri için en önemli motivasyon, ÖAKY lisanslarının ya sıfır ya da çok düşük maliyetli olmasıdır. Gelişmekte olan ülkelerdeki ÖAK olmayan yazılımların neredeyse tamamı ithal edildiğinden, döviz kurları ve cari açık üzerindeki etkisi olumsuzdur. ÖAK olmayan yazılımların ülke ekonomisi üzerindeki olumsuz etkisinin yanı sıra kurum ve kuruluşlar için de önemli bir gider kalemidir. Üstelik ilk lisanslama ücretinin yanında bakım destek sözleşmeleri kapsamında da astronomik ücretler talep edilebiliyor. Ticari/tescilli yazılımın mimarisine ve kaynak koduna erişimi olmayan kurum/kuruluş ise bu bakım destek sözleşmelerine yıllarca mahkûm edilebiliyor. ÖAK yazılımlar ise özel mülk yazılımların aksine bu konuda kullanıcıya neredeyse tam bağımsızlığı garanti eder.

2.3.2. Güvenlik

Tamamen güvenli bir işletim sistemi veya yazılım olmasa da açık standartlar, geliştirme yöntemi ve program mimarisi gibi faktörler bir sistemin güvenliğini büyük ölçüde etkileyebilir. ÖAKY sistemlerin bu açıdan ÖAKY olmayan sistemlerden daha üstün olduğu söylenebilir. ÖAK yazılımların bu üstünlüğü sağlamasındaki en önemli unsur hiç şüphesiz açık standartlarda geliştirilmiş olmasıdır. Çünkü açık kaynak olarak geliştirilen bir yazılım güvenlik açıklarının tespiti açısından daha avantajlıdır. ÖAK olmayan yazılımlarda kaynak koda erişim olmadığından, muhtemel bir güvenlik açığının tespiti sadece o yazılımı geliştiren firmanın geliştirici ekibinin bakış açısına tabidir. Oysa ÖAK yazılımlarda ortaya çıkabilecek herhangi bir güvenlik açığının, yüz binlerce geliştiricinin olduğu bir topluluğun gözünden kaçması daha zordur.

ÖAKY sistemleri için kaynak kodunun görünebilirliği, geliştiricilerin ve kullanıcıların, genellikle bir güvenlik açığının art niyetli kullanılmasından önce keşfedilmesini kolaylaştırır. ÖAK yazılımlardaki güvenlik açıklarının birçoğu, periyodik denetimler sırasında keşfedilen ve herhangi bir kötüye kullanım olmaksızın giderilen hatalardı. Bu sebeple ÖAKY sistemler, reaktif denetim mekanizmaları yerine proaktif denetim mekanizmalarını kullanır.[48]

Ayrıca ÖAKY geliştirici toplumu genelde kullanıcı dostu arayüzler ya da kullanım kolaylığı gibi konulara yoğunlaşmak yerine güvenlik ve kararlılık gibi konulara odaklanır. Herhangi bir ÖAKY yayımlanmadan önce sistem güvenliği konusunda diğer geliştiricilerden de onay alınır.

[48] Kenneth Wong ve Phet Sayo, (2004). *"Free/Open Source Software: A General Introduction: Security"*, s.13, United Nations Development Programme's Asia-Pacific Development Information Programme (UNDP-APDIP), Malaysia.

Hiç şüphesiz ticari / tescilli / özel mülk yazılım kullanıcılarının güvenlikle ilgili en büyük endişesi “Backdoor”[49]’lardır. Yazılımın geliştiricisinin genellikle kasıtlı olarak yazılımda bıraktığı bu güvenlik açığı sayesinde sistem veya yazılım, yetkisiz erişim ve işlemlere açık hale gelir. Geliştirici bu açığı kendi çıkarı doğrultusunda her amaç için kullanabilir ya da üçüncü taraflara satabilir. Backdoor’ların tespit edilmesi çok zor olup kaynak koda erişimi olmayan kullanıcının bu açığı tespit etmesi de pek mümkün değildir. ÖAK yazılımlarda ise böyle bir durum söz konusu değildir. Açık kaynak kodların herkes tarafından görülebilmesi ÖAK yazılımlarda, Backdoor gibi kasıtlı güvenlik açıklarının bırakılmasını büyük oranda engeller.

2.3.3. Kararlılık ve Güvenirlilik

ÖAK yazılımların sağladığı önemli avantajlardan biri de sistem kararlılığıdır. ÖAK olmayan yazılımların karmaşık yapısı ve kullanışsız Debug prosedürlerinden arındırılmış olan ÖAK yazılımlar, yıllarca sistem kararlılığı ve güvenirliliğini sağlayabilir.

Özel mülk yazılımların aksine daha basit ve verimli yapısı sayesinde ÖAK yazılımlardaki sistem çökme sorunları daha azdır. Örneğin, 1999 yılında RedHat Linux, OpenLinux ve Microsoft Windows NT Server sistemleri 10 aylık bir bağımsız kararlılık testine tabi tutuldu. Üç sistem de tamamen aynı donanım sistemleri üzerinde çalıştırıldı ve baskı, web hizmeti ile dosya sunma işlevlerini gerçekleştirmesi beklendi. Sonuç olarak Windows NT Server 6 haftada bir çöktü ve düzeltilmesi ortalama 30 dakika sürdü. Ancak GNU / Linux tabanlı sistemler bu 10 aylık test sürecinde hiçbir zaman sistem çökmesi yaşamadı.[50]

[49] Backdoor: Yazılımın geliştiricisinin genellikle kasıtlı olarak yazılıma eklediği/eklemediği bir yazılım açığıdır. Yazılıma yetkisiz erişime izin verir.
[50] David A. Wheeler, (2015*). "Why Open Source Software / Free Software (OSS/FS, FLOSS, or FOSS)? Look at the Numbers!"*, https://dwheeler.com/oss_fs_why.html (22.02.2021)

2.3.4. Ücretsiz Destek ve Hızlı Debug

ÖAKY sahibi son kullanıcı ya da kuruluş, bir ÖAKY dağıtımını ister ticari bir yazılıma entegre ediyor olsun isterse de tek başına kullanma niyetinde olsun hiçbir mali yükün altına girmeden ve hiçbir taahhüt vermeden yüz binlerce üyesi bulunan açık kaynak topluluklarına bu konuda soru sorabilir ya da tavsiye alabilir. Vakıflar, forumlar, haber grupları, yazılım paylaşım platformları aracılığıyla sorunlara çözüm bulanabilir. Ayrıca çok iyi dokümante edilmiş hata ayıklama raporları, “datasheet” ler, “errata” lar, ve “troubleshooting” planları her an her yerde ulaşılabilir durumdadır.

Açık kaynak olarak geliştirilen yazılımların kaynak kodları, onları inceleyen milyonlarca göz olabileceğinden, çok daha küçük bir geliştirme kadrosuna sahip ticari/tescilli yazılım satıcısının koduna kıyasla daha iyi bir hata ayıklama sürecinden geçer.[51] Ayrıca, açık kaynak toplulukları genellikle çok hızlı şekilde hata düzeltmelerini uygular veya geçici bir çözüm bildirir. Ek olarak, kaynak kodu yazılımla birlikte geldiğinden, son kullanıcılar istedikleri zaman kendi yamalarını uygulamakta özgürdürler.

2.3.5. Açık Standartlar ve Daha Az Satıcı Bağımlılığı

Açık standartlara göre geliştirilen yazılımlar hükümetlere, son kullanıcılara, kurum ve kuruluşlara esneklik ve özgürlük sunar. Yani ÖAK yazılımların kullanıcıları farklı yazılım paketleri, platformlar ve satıcılar arasında geçiş yapma özgürlüğüne sahiptir. Tescilli/ticari olarak gizli standartlara göre geliştirilmiş yazılımlar ise kullanıcılarını sadece kendi tedarik zincirinde tutmaya yönelik katı metotlar geliştirmişlerdir. ÖAK olmayan

[51] Optimus Information, (2015). *"Open-Source vs. Proprietary Software Pros and Cons: Fewer Bugs and Faster Fixes"*, s.4, Canada.

yazılımlar kullanıcılarını sadece ilk satın alma lisansı ile kendine bağlamaz. Aynı zamanda kullanım lisansı ile de kullanıcıyı sınırlar. Yani ÖAK olmayan bir yazılımın dağıtıcısı, kullanıcının ayrı bir lisans altında yayımlanmış farklı bir yazılımı, kendi yazılımıyla birlikte kullanmasını genellikle yasaklar ya da kısıtlar. Böylece farklı yazılım ihtiyaçlarını karşılamayı planlayan kullanıcıları yine kendi yazılım setlerine yönlendirmiş olur. Ayrıca ticari/tescilli yazılımlarını lisanslarken algoritmasını ve kaynak kodunu paylaşmaz ve muhtemel müdahaleleri de yasaklar. Böylece kullanıcının kendi yamalarını geliştirmesini de engellemiş olur. Kullanıcı da ÖAK olmayan bu sistemi işletebilmek için ÖAK olmayan yazılımın sahibiyle bakım ve destek sözleşmeleri yapmak zorunda kalır. Yani son kullanıcı, sadece ilk satın alma lisansına değil aynı zamanda o firmanın ÖAK olmayan tüm yazılım ekosistemine kökten bağlanmış olabilir.

ÖAK yazılımlar ise tüm bu bağımlılık gerektiren yazılım ekosistemini son kullanıcıya dayatmaz. Son kullanıcı isterse farklı yazılım paketleri, platformlar ve satıcılar arasında geçiş yapabilir. Ek olarak bu yazılımları birbirine entegre edebilir, kodları parçalayıp ihtiyacına göre yeniden yazıp kendi yamalarını geliştirebilir. Hatta bazı açık kaynak lisanslarına uymak üzere; açık kaynak yazılımlarını diğer tescilli/ticari yazılımlara entegre edebilir. Ayrıca ÖAK yazılımların çoğu, neredeyse her tür bilgisayarda çalışmak üzere tasarlandığından, eski donanımınızın ömrünü uzatmanıza ve arada sırada bunları değiştirme ihtiyacını ortadan kaldırmanızı da sağlar.

Açık kaynak toplulukları ÖAK olmayan standartlar konusunda temkinli davranmakta, bunun yerine iletişim protokolleri ve veri formatları etrafında açık standartlara bağlı kalmayı tercih etmektedir. Bu özellik, açık kaynak ve ÖAK olmayan yazılımların birlikte çalışmasına ortam hazırlar.[52]

[52] Optimus Information, (2015). *"Open-Source vs. Proprietary Software Pros and Cons: Open Standarts-Fewer Bugs and Faster Fixes"*, s.4, Canada.

Tüm bu avantajların yanı sıra, ÖAK yazılımlar açık standartları sayesinde tersine mühendisliği de mümkün kılar. Böylece son kullanıcı algoritmayı ihtiyacına göre yeniden programlayabilir veya kod bloklarını ihtiyacına göre parçalayabilir. Ancak ÖAK olmayan yazılımlarda böyle bir durum söz konusu dahi değildir. ÖAK yazılımın geliştiricisi kasıtlı olarak kaynak kodun görülmesini engelleyecek tedbirler alır. Yazılımlarını binary formatta yayımlayabilir, gizleyici kod bloklarını yazılıma ekleyebilir.

ÖAK yazılım lisansları dünya çapında geçerli tek lisans setine sahiptir. Yani GNU GPL altında yayınlanmış bir yazılım, dünyanın neresinde olursanız olun aynı GPL lisans bildirimlerine tabidir. Ancak ticari/tescilli yazılım geliştiricileri son kullanıcıya verdiği yazılım lisanslarını bölgesel olarak sınırlandırabiliyor. Örneğin Türkiye'de lisansını aldığınız bir yazılımı, Almanya'da kullanamayabilirsiniz. Hatta böyle bir kullanım, son kullanıcının ciddi yasal yaptırımlarla karşılaşmasına sebep olabilir. Çünkü özel mülk yazılım sahipleri, bir ülkenin ekonomik ortamına, döviz kurlarına veya siyasi ilişkilerine bakarak o ülke için farklı yazılım ücretlendirme politikası belirleyebiliyor. Ya da bir ülkede belirli bir alandaki yazılım sektörüne ilk kez giriş yapıyorsa piyasa payını artırmak için farklı lisans ücretleri belirleyebilir.

Örneğin ticari/tescilli bir yazılım, A ülkesinde 350 $'a satılırken, daha az gelişmiş B ülkesinde 250 $'a satılabiliyor. A ülkesindeki bir son kullanıcının B ülkesinden yazılımın lisansını edinip A ülkesinde kullanmasını istemeyen lisans tedarikçisi, B ülkesinde verdiği yazılım lisansına, B ülkesiyle sınırlı kalacak şekilde coğrafi sınırlandırma getirebiliyor. Ancak ÖAK yazılımlar için böyle bir durum söz konusu değildir. BSD, GNU GPL, Mozilla, LGPL vs. gibi açık kaynak lisanslarda coğrafi sınırlandırma gibi son kullanıcının yazılım üzerindeki tasarruflarını kısıtlayan tedbirler yoktur. BSD veya GPL lisans bildirimlerinin mevcut sürümlerinde olduğu gibi eski sürümlerinde de yazılım lisansının coğrafi sınırlandırılmasına yönelik bir madde olmamıştır.

2.3.6. Yerel Yazılım Kültürünün Desteklenmesi

Geliştirmeyi ve yeniden dağıtımı teşvik eden ÖAK yazılımlara erişmek, kullanmak ve öğrenmek hem kolay hem de maliyetsizdir. Bu sayede bir yazılımın birlikte çalışma kültürüyle geliştirilmesi ve paylaşılması mümkündür. Bu paylaşım kültürünün devamı için kurulan yazılım toplulukları da bu sürece katkı sunmaktadır. Bu yazılım topluluklarının gelişmesi ise o ülkedeki ÖAK yazılımların ne kadar yaygın kullanıldığına bağlıdır. Açık kaynak yazılım kültürünün geliştiği bir ortamda üniversiteler, araştırmacılar, öğrenciler ile mühendisler ve serbest geliştiriciler de bu kültüre katkı sunmak için bir araya gelecektir. Böylece ülke içinde geliştirilen ÖAK yazılım ürünlerinin hem sayısı hem de kalitesi artacaktır. Bu açık kaynak kültürünün oluşması, o ülkenin dijital dönüşümüne ve bu dönüşümü daha güvenli milli yazılımlarla gerçekleştirmesine katkı sunacaktır.

Tescilli/ticari yazılımlar ise, yalnızca kaynak kodunun sınırlı kullanılabilirliği nedeniyle değil, lisanslama, patent ve telif hakkı sınırlamaları nedeniyle çok daha kısıtlayıcı olma eğilimindedir.[53] Bu sebeple bu yazılımlarla ilgili kaynaklara, dokümanlara ve eğitimlere ulaşmak biraz daha zordur. Bu zorluk, özel mülk yazılımların üzerine bir yazılım eğitim modeli ya da yazılım kültürü oluşturulmasını zorlaştırmaktadır.

ÖAK yazılımların açık ve işbirlikçi doğası, bir öğrencinin topluma neredeyse hiçbir mali yük getirmeksizin yazılım konseptlerini incelemesine ve denemesine olanak tanır. Benzer şekilde, bir öğrenci, gerekli teknik bilgi ve dokümanlara özgürce açık kaynak platformlardan erişebilir. Ancak herhangi bir son kullanıcının veya bir öğrencinin açık kaynak kültürünü benimseyip bu

[53] Kenneth Wong ve Phet Sayo, (2004). *"Free/Open Source Software: A General Introduction: Developing Local Software Capacity"*, s.15-16, United Nations Development Programme's Asia-Pacific Development Information Programme (UNDP-APDIP), Malaysia.

ekosistem içinde bir yazılım geliştiricisi olması için teşvik edici bir ortamın oluşması gerekir. Bunun için, hükümetler ÖAK yazılımların kullanımını teşvik edici yasal düzenlemeler yapmalı, açık kaynak yazılımlara geçiş için plan ve projeksiyonlar oluşturmalıdır. Aynı şekilde şirketler, kurum ve kuruluşlar da sistemlerinde açık kaynak yazılımlara yer vererek ÖAKY geliştiricilerine, geliştirecekleri yazılımlara ticari anlamda ilgi duyulduğu mesajını vermelidir.

Tüm bunların yanında ülke içinde oluşacak yerel bir yazılım kültürü, o ülkenin daha güvenli, bağımsız ve düşük maliyetli yazılımlara sahip olması anlamına gelir. Böylece milli güvenliğini tehlikeye atacak Backdoor şüphesi taşıyan özel mülk yazılımlar yerine kendi kapasitesiyle geliştirdiği bir yazılım ekosistemi oluşturabilir.

2.4. AÇIK KAYNAK GÖÇÜNÜN RİSK ANALİZİ

ÖAK yazılıma göç etmenin getirdiği avantajların yanında, bir açık kaynak göçü her duruma uygun olmayabilir. Bunun için açık kaynak göçü planlayan kurum ya da kuruluş ÖAKY sistemden beklentilerini, gelecek hedeflerini iyi belirlemelidir. Aksi takdirde açık kaynak göçünün taşıdığı risklerden ve dezavantajlardan daha fazla etkilenebilir.

2.4.1. Kullanılabilirlik

ÖAK yazılım ve işletim sistemleri, ticari/tescilli yazılımların aksine kullanıcı pazarını hedef alan yazılımlar değillerdir. ÖAKY geliştiricileri genelde, kullanıcı dostu arayüzler geliştirmeye yoğunlaşmak yerine daha stabil ve güvenli sürümler geliştirmeye odaklanmıştır. Özellikle BSD ailesinden işletim

sistemleri, son kullanıcılarının yüksek profil olduğu sistemlerdir. Düşük profilli son kullanıcılar bu bakımdan ÖAK işletim sistemlerini kullanmakta zorlanabilirler. Yazılımın açık kaynak olması ve kodlarının incelenebilir olması da son kullanıcı için bir şey ifade etmeyebilir. Ayrıca kodu incelemek bir yana bir yazılım paketini derleyip sisteme kurmak bile son kullanıcıların istediği son şey olabilir. Bu durum, üretkenliği ve iş akışışını da etkileyebilir.

Son kullanıcı bilgisayarlarında yapılacak kapsamlı bir göçte, son kullanıcıların ÖAK yazılımları benimsemesi önemli bir parametredir. Son kullanıcı bilgisayarlarında kullanılan ticari/tescilli işletim sistemlerinin büyük bölümünün Microsoft Windows ya da Mac OS olduğu söylenebilir. ÖAK yazılım tarafında ise Linux ve türevleri bu sistemlerin en önemli rakibidir. Özellikle son yıllarda Linux ve diğer Linux türevi dağıtımlar için kullanıcı dostu yazılım araç setleri geliştirilmiştir. Hatta son kullanıcıların daha kolay geçiş sağlaması için Mac OS ya da Windows'a neredeyse birebir benzeyen tema ile yazılım araçlarını içeren yazılım paketleri de geliştirilmiştir. Bu kapsamda personel tepkisinin iyi ölçülmesi, gerekirse pilot uygulamalarla geçişin kademelendirilmesi ve ÖAK yazılımların kullanımı konusunda verilecek eğitimlerin iyi planlanması göçün başarısı açısından önem arz etmektedir.

Son kullanıcıdan bağımsız olarak, sistem altyapısında gerçekleştirilecek bir göç operasyonunda ise kullanılabilirlik veya kolay kullanım çok önemli bir parametre değildir. Çünkü sistem yöneticileri genelde yüksek profilli kullanıcılar olmaktadır. Bu sebeple sistem entegrasyonu sağlandığı sürece altyapıda kullanılacak bir ÖAK yazılımın kullanıcı dostu olması çok anlam ifade etmeyebilir. Altyapıda yapılması planlanan bir göç operasyonunda bir ÖAK yazılımın kolay kullanım sunması sistemi işletecek olan sistem yöneticileri için önemli ise, ÖAK yazılımlara kolayca entegre edilebilecek ÖAK olmayan operatör yazılımlarla da bu imkân sağlanabilir.

2.4.2. Dokümantasyon ve Eğitim

Bazı ÖAK yazılımlar, ticari/tescilli yazılımlardaki gibi kapsamlı belgelere ve kullanım kolaylığına aynı düzeyde sahip olmayabilir. İlk ÖAKY geliştiricilerinin odak noktası işlevsellikti. İyi çalışan ve stabil bir program oluşturmak, kullanım kolaylığından çok daha önemliydi.

Yüksek kaliteli dokümantasyon eksikliğinin yanı sıra, Kullanıcı Grafik Arayüzü GUI ile ilgili de sorunlar vardı. Bu sebeple günümüzde var olan çoğu ÖAKY sistemindeki GUI öğesi tek bir öğe değil, birbirine iliştirilmiş farklı projelerden oluşan bir koleksiyon gibidir. Verileri kaydetme komutu bile, Mac OS veya Microsoft Windows gibi tescilli masaüstü işletim sistemlerinden oldukça farklı olarak, bir programdan diğerine farklılık gösterebilir. Farklı programlar arasında kesmek ve yapıştırmak gibi son derece basit işlemler bile dağıtımlar arasında farklılık gösterebilir. ÖAK yazılımlarla ilgili sertifikalı ve kaliteli eğitimler bulmanın da zorluğuyla kullanım kolaylığı sağlanamayabilir.

Tüm bu uyumsuzluk ve dağınık yapı sebebiyle ÖAKY çalışmalarını dokümante etmek geliştiriciler için zor olmuştur. Bu sebeple bazı ÖAK yazılımlar iyi organize edilmiş kaliteli ve detaylı dokümanlara sahip olmayabilir. Dolayısıyla ÖAK yazılımlarla ilgili bir şeyler ters gittiğinde çözüm için kapsamlı bir dokümana ulaşmak bazen zor olabilir. Bahse konu bu dokümantasyon eksikliği ise genellikle küçük geliştirici grupların yazdığı uygulamalarda söz konusudur.

Bu durum açık kaynak göçü planlayan kurum ve kuruluşların dikkate alması gereken bir risktir. Riski azaltmak için sistemde kullanılacak ÖAK yazılımla ilgili ön araştırma yapmak ve dokümantasyon kaynaklarını araştırmak iyi bir strateji olabilir.

2.4.3. Mevcut Ticari/Tescilli Yazılımlarla Çalışabilirlik

Özellikle son kullanıcıya yönelik geliştirilen ÖAK yazılımlar, ticari/tescilli yazılım ve sistemlerle genellikle uyumlu değildir. Birçok tescilli donanım türü de genellikle yalnızca ekipman üreticisinden temin edilebilen açık kaynak programları çalıştırmak için özel sürücülere ihtiyaç duyabilir. Bu durum, potansiyel olarak göç operasyonunun maliyetini artırabilir. Açık kaynaklı bir sürücü mevcut olsa bile, yazılımınızın ticari/tescilli sürücüler ile çalışmasının garantisi yoktur. Ticari/tescilli uygulamalara ve veri depolama formatlarına halihazırda büyük miktarda para yatırmış olan kuruluşlar için, ÖAKY çözümlerini entegre etmeye çalışmak çok daha pahalı olabilir.[54] Genellikle alternatif çözümlerle entegrasyonu sağlamaya çalışmak daha mantıklıdır. Program sanallaştırma, yazılım sanallaştırma ya da web tabanlı uygulamalara geçiş gibi. Son kullanıcı tarafında açık kaynak yazılım ve özel mülk yazılım entegrasyonu konusu bu tarz alternatif yollarla çözüme kavuşturulabilse de sistem altyapısında yapılması gereken entegrasyonlar daha zorlayıcı olabilir. Böyle bir durumda ise göç planlamasında ilk aşamada hibrit yöntemler takip edilebilir. Yani özel mülk ve açık kaynak yazılımları birbirine entegre etmek yerine bağımsız işleri yerine getirmeleri istenebilir.

2.4.4. Bakım ve Destek Süreçleri

ÖAKY yazılımlara geçiş yapan bir kurum ya da kuruluş, özel mülk yazılımlarda aldığı düzeyde bir bakım destek hizmetini ÖAKY sistemlerde de alabilir. Bunun için açık kaynak geçişlerini planlayan ve devamında işletme desteği veren artık birçok firma mevcut. Göçten sonra böyle bir bakım destek

[54] Kenneth Wong ve Phet Sayo, (2004). “Free/Open Source Software: A General Introduction: Interoperability with proprietary systems”, s.17, United Nations Development Programme’s Asia-Pacific Development Information Programme (UNDP-APDIP), Malaysia.

hizmeti alınmasa dahi, açık kaynak yazılımlar üzerinde kuruluş bünyesinde geliştirmeler ya da yamalar yapabilir. Ancak bu anlamda yetişmiş insan gücü ve teknik kapasiteye sahip olmayan bir kuruluş, açık kaynak sistemlerini mevcut bilgi, belge ve dokümanlar yardımıyla işletmesi zor olabilir. Özellikle kod hataları, tasarım kusurları ve entegrasyon zorlukları gibi sorunları bünyesinde barındırma ihtimali sebebiyle ÖAKY sistemlerinde bakım destek hizmeti alınması bazen zorunluluk olabilir. Bununla birlikte, çoğu ÖAKY geliştiricisi ya da dağıtıcısı, belirli bir ücret karşılığında destek ve bakım hizmetlerini de sunmaktadır. Ayrıca açık kaynak yazılım topluluklarından da sorunları ve açıkları düzeltme konusunda yardım alınabilir. Ancak unutulmamalıdır ki, açık kaynak toplulukları sorunlara hızlı bir şekilde yanıt vermekte başarılı olsa da hiç kimse yasal olarak bunu yapmak zorunda değildir.

2.4.5. Sistemin Devamlılığı Riski

ÖAK yazılımlar genelde stabil, kararlı, güvenilir ve uzun dönem ihtiyaçları karşılayabilecek sistemlerdir. Ancak zaman içerisinde, ÖAK yazılımın kullanıcısı, sisteminde iyileştirmeler ya da geliştirmeler yapmak isteyebilir. Ya da ihtiyacına göre özel mülk yazılımlarla entegrasyon sağlamak isteyebilir. Bu konularda yetişmiş insan kaynağına da sahip değilse, sisteminde kullandığı ÖAK yazılımın sağlayıcısı, dağıtıcısı veya geliştiricisi ile bu işlemi gerçekleştirmek isteyecektir. Bu noktada, ÖAKY satıcısı/geliştiricisinin işin devamlılığını sağlaması kritik önemdedir.

Red Hat gibi, finansal açıdan güçlü ve büyük açık kaynak yazılım geliştiricisi çok azdır. Açık kaynak göçü gerçekleştirilirken küçük ama yetenekli ve hızlı geliştiriciler bazen tercih edilebilir. Böylece göç süreci ile ilgili hızlı karar alma, hızlı hata ayıklama, uzun prosedürlerden kaçınma gibi avantajlar elde

edilebilir. Ancak devam eden süreçte, sistemde kullanılan bir ÖAK yazılımla ilgili herhangi bir sorunda muhatap bulamama riski de yüksektir. Yani geliştiricilerin projeden ayrılma ve yazılımı resmi olarak "yetim"[55] bırakma ihtimali vardır. Bu sebeple göç ile ilgili takvimin iyi belirlenmesi; göç ve bakım destek hizmeti sürecinde destek alınacak firmaların özenle seçilmesi sistemin işleyişi açısından kritik öneme sahiptir.

Sistemin devamlılığı riski ÖAKY sistemler için var olduğu gibi özel mülk yazılımlar için de söz konusudur. Bu ÖAKY sistemler açısından ticari/tescilli yazılımlara oranla ekstra bir dezavantaj oluşturmaz. Sadece ÖAKY geliştiren, finansal açıdan güçlü ve büyük firma sayısı ticari/tescilli yazılım geliştiren firma sayısına göre daha azdır. Bu sebeple hangi açık kaynak geliştiricisinin hangi yazılımının kullanılacağı son kullanıcının sorumluluğundadır.

ÖAK olmayan yazılımları tercih edenler, ürünü kimin tasarladığını, geliştirdiğini ve dağıttığını tam olarak bildikleri için; programın düzgün çalışmaması veya donanımlarına zarar vermesi durumunda kimi sorumlu tutabilecekleri konusunda net bir fikirleri vardır. Ancak ÖAK yazılımlar için bunu söylemek zor. ÖAK yazılımlar, genellikle çok sayıda geliştirici tarafından geliştirildiğinden, son kullanıcının işler ters gittiğinde direk sorumlu tutabileceği bir kişi ya da şirket bulması mümkün olmayabilir.

Ancak, özel mülk yazılım geliştiricileri de bu konuda karşı tarafa garanti vermemektedir. Hatta EULA lisans bildirimlerinde, genellikle tüm yükümlülüklerini reddettiklerini ve yazılımın kullanım sorumluluğunun tam olarak son kullanıcıya ait olduğunu ifade eden sorumluluk reddi beyannamelerini görmek mümkündür. Bu nedenle, sonuç alınamayacak

[55] "Yetim" Yazılım: Geliştiricilerin ilgisini kaybetmesi, projenin sonlandırılması, geliştirilen dilin eskimesi, donanım bağımlılığı varsa ilgili donanımın artık üretilmemesi gibi sebeplerden ötürü geliştiricileri tarafından terk edilen yazılımlar için kullanılan ifadedir.

davalar açmak, büyük şirketlerin peşine düşmek ve tüm bunlar için zaman, para ve enerji harcamak yerine, ÖAK yazılımları kullanmak ve bunların düşük maliyetli ve kolay erişilebilirliğinin bazı risklerle birlikte geldiğini kabul etmek daha mantıklı olabilir.

2.4.6. Daha Az Rekabet

Bir ÖAK yazılımının geliştirilmesi ve dağıtımı, genellikle yüksek gelir modellerine uymaz. Açık kaynak olarak, genellikle birden fazla yazılımcının geliştirmesi, ÖAK yazılımları ticari bir sır olarak tutmayı imkânsız kılmaktadır. Zaten böyle bir şey özgür ve açık kaynak olmanın doğasına da aykırıdır.

Taraflar arasında imzalanan sözleşmeye göre lisans şartları değişse de bu sistemlerden ticari/tescilli yazılımlarda olduğu gibi çok yüksek gelir elde etmek pek mümkün değildir. Bu sebeple daha yükse gelir vadeden özel mülk yazılım ekosistemi daha fazla nitelikli yazılımcıyı kendine çekmektedir. Daha fazla yetişmiş insan kapasitesi, daha fazla para, daha fazla nüfuz ve daha fazla imkân sunan özel mülk yazılım dünyası kendi platformu üzerinde daha fazla yazılım firmasının da çıkmasına ortam hazırlamıştır. Ticari/tescilli yazılım dünyasında hızla artan firma sayısı, rekabeti de beraberinde getirmiştir. Böylece daha fazla imkana, yetişmiş insan kapasitesine ve paraya erişim imkânı bulan firmalar, daha kaliteli, daha organize ve daha çeşitli yazılımlar geliştirmişlerdir.

Ancak ÖAKY ekosisteminde aynı düzeyde rekabet ortamının olduğunu söylemek pek mümkün değil. Her ne kadar aynı kapasitede yetişmiş insan gücüne sahip olsa da geliştiriciler arasındaki organizasyon eksikliği, tam birlikteliğin sağlanamaması, geliştirilen ürünlerdeki dağınık yapı, bu insan

gücünün tam kapasiteyle çalışması önündeki engellerden sadece birkaçı. Bu dağınık yapı ve kısıtlı imkânlar sebebiyle; ÖAK hareketi içinden çıkan, güçlü finansal yapıya ve büyüklüğe erişmiş yazılım firmalarının sayısı da oldukça azdır. ÖAKY geliştiren nitelikli firmaların sayısının az olması geliştirilen yazılım ürünlerinin çeşitliliğinin ve kalitelerinin yeterli seviye de olmasına engel olmaktadır.

Ayrıca sektördeki rekabet eksikliği göç planlayan kurum ve kuruluşlar için de önemli bir risktir. Çünkü altyapısını tamamen açık kaynak yazılımlara geçirmiş bir kuruluş, gelecek vizyonunu da açık kaynak sistemlere göre oluşturmuş demektir. Yani sonradan bu ekosistem içinden çıkacak yeni sürümlere, yazılımlara, sürücülere, güncellemelere ve geliştirmelere bağımlıdır. Yukarda bahsedilen rekabet eksikliğinden kaynaklı bu geliştirmelerde ortaya çıkabilecek kalitesizlik ya da yetersizlik göçün sürdürülebilirliğini olumsuz etkileyebilir. Ayrıca firma sayısının azlığı, hem sonraki dönemde ortaya çıkabilecek sorunlar için muhatap bulmayı zorlaştırabilir hem de satıcılar arasındaki ücret rekabetini sınırlandırabilir.

2.4.7. Ek Maliyet Yükü

İyi planlanmamış ya da kısmen başarısız olmuş göç operasyonları, göçü gerçekleştiren kurum ya da kuruluşa ciddi ek maliyetler getirebilir. Daha da kötüsü tamamen başarısız olmuş bir göçten sonra eski sisteme dönmek gerekebilir. Böyle bir durum yüksek ek maliyet yükü getirebileceği gibi sistemin tamamen durmasına ve güvenlik açıklarının oluşmasına da sebep olabilir. Bu hasarı telafi etmek için ise fazladan hem zaman hem para hem de enerji harcamak gerekebilir.

Böyle bir durumun önüne geçmek için göç planlamasının iyi yapılması, gerekli teknik altyapının oluşturulması gerekir. Eğer veri sahibi, teknik altyapıyı sağlayacak gerekli bilgi ve insan kapasitesine sahip değilse, açık kaynak göçü konusunda bilgi ve tecrübe sahibi firmalardan destek alması önemlidir. Ayrıca göç kararı veren taraf, geleceğe yönelik hedeflerini, planlarını ve projelerini iyi belirlemelidir. Bu kapsamda ÖAK sistemlerden beklentilerini, geçiş takvimini, geçişin mali ve risk analizini bu kıstaslara göre önceden hazırlamalıdır. Gerekirse olumsuz durumlar için ikincil ve üçüncül planlar hazırlanmalıdır. Geçişin hibrit bir model mi ya da tam bir geçiş mi olacağı ile mevcut ticari/lisanslı yazılımlara karşı hangi ÖAKY alternatiflerinin kullanılacağı konularında detaylı planlar hazırlanmalıdır. Duruma ve takvime göre pilot uygulamalarla geçişin kademelendirilmesi sağlanarak sistemin ve son kullanıcıların tepkisi ölçülüp göçün ek maliyet yükü getirmesi kısmen ya da tamamen önlenebilir.

2.4.8. Teknolojik Beklentiyi Karşılamama ve İşleyişin Aksaması

Göç sonrasında ÖAKY sistemlerde kullanılan yazılımların yeterli geliştirme ve güncellemeleri alamaması nedeniyle ilgili yazılım, günün teknolojik gelişimlerinin gerisinde kalabilir. Böyle bir durumda sistemde; entegrasyon problemleri, verilen işleri gerçekleştirememe ve son kullanıcı ihtiyaçlarını karşılayamama gibi sorunlar ortaya çıkabilir. Özellikle “yetim” bırakılan yazılımlar, copyleft lisansı altında yayımlanan yazılımlar ve anonim olarak geliştirilen yazılımlarda bu tarz durumlar daha fazla ortaya çıkabiliyor. Bunu önlemek için ise; genellikle yaygın olarak kullanılan, geniş bir geliştirici toplum tarafından yazılmış, mümkünse arkasında köklü ve güçlü finansal yapıya sahip firma ya da firmaların bulunduğu ÖAK yazılımlar tercih edilmelidir. Tercih edilecek yazılımın ya da işletim sistemlerinin aldığı güncellemeler ile sürümler detaylıca incelenmeli, sistemde kullanılıp kullanılmayacağına buna göre karar verilmelidir.

2.4.9. Güvenlik Açıklarının Oluşması

Kısmen ya da tamamen başarısız olmuş bir göç, sistem genelinde güvenlik açıklarına sebep olabilir. Özellikle sistem üzerindeki yazılımların yeterli güncelleme, geliştirme ve hata düzeltmelerini alamaması o yazılım özelinde bir güvenlik açığı oluşturabileceği gibi tüm sistemi tehdit eden bir açığa da dönüşebilir. Hibrit bir göç sonucunda, Ticari/tescilli yazılımlar ile açık kaynak yazılımlar arasındaki entegrasyonun tam sağlanamaması sonucunda da güvenlik açıkları oluşabilir. Ayrıca bir yazılımın ÖAKY olması o yazılımın tamamen güvenilir olduğu anlamına gelmez. Bu genel kanının aksine veri sahibinin ÖAK yazılımları seçerken Backdoor ve diğer Truva yazılımlara karşı dikkatli olması gerekir.

2.5. AÇIK KAYNAK GÖÇÜNÜN FİNANSAL ANALİZİ

Bütün ÖAK yazılımların ücretsiz olduğu konusundaki genel kanı doğru değildir. Her ne kadar ÖAK yazılımların büyük bölümü ücretsiz olsa da ücretli ÖAK yazılımlar da mevcuttur. Ancak ÖAK yazılımların hem çok azı bir ücretlendirmeye tabidir hem de bu ücretlendirme tescilli/ticari yazılımlar kadar yüksek değildir. Örneğin bir tarayıcıya girip Debian'ı hiçbir ücret ödemeden indirip cihazınıza kurabilirsiniz. Hatta ÖAK işletim sistemleri, kurma işini bile kullanıcının tercihine bırakır. Kullanıcı isterse "live" bir sistem olarak USB sürücüye kurup USB den işletim sistemini çalıştırabilir.

ÖAK yazılımlar genellikle özel mülk yazılımlardan daha ucuzdur. Ancak, ilk satın alma lisans maliyetleri bir ÖAK yazılım paketinin tek maliyeti değildir. Yani altyapısında muhtemel bir göç planlayan kurum ya da kuruluş sistemde tamamen ÖAK yazılımlar kullanarak ilk satın alma maliyetlerini kısabilir hatta sıfırlayabilir. Fakat bu çıktıya dayanarak, ÖAKY bir sistemin tamamen

maliyetsiz olduğu fikrine kapılmamalıdır. Çünkü sistemi işletme maliyetleri, personel giderleri, harici donanım ihtiyacı ve personelin eğitim maliyetleri de hesaba katılmalıdır. Ayrıca göç sürecinde alınacak teknik destek ile devamındaki bakım destek hizmetleri de maliyet hesabında dikkate alınmalıdır.

İlk lisans edinim ücretinin olmaması, bir göç operasyonunun her zaman ekonomik fayda sağlamasını garanti etmez. İyi organize edilmemiş bir göç operasyonunun toplam maliyeti, lisans ücretlerinden yapılan tasarrufları gölgede bırakabilir. Nihayetinde ÖAK yazılımlarda lisans ücretinin olmaması, açık kaynak kodlu bir sistemin genel mali değerlendirmesinde yalnızca bir gider kalemini temsil eder. Bir açık kaynak göçünün mali hesaplamasını ve yatırım getirisini doğru hesaplayabilmek için, göç sürecinin ve sonrasının da dahil edildiği; ilgili tüm giderlerin hesaba katıldığı kapsamlı bir finansal değerlendirme yapmak gerekir.

ÖAK yazılımlı bir sisteme geçişte sağlanacak fayda sistemin ilk yılında çok belirgin olmayabilir. Genellikle ÖAK bir sistemin işletme ömrü lisanslı/tescilli yazılımların kullanıldığı bir sisteme göre çok daha uzun olup bakım ve destek ihtiyacı ise daha azdır.[56]

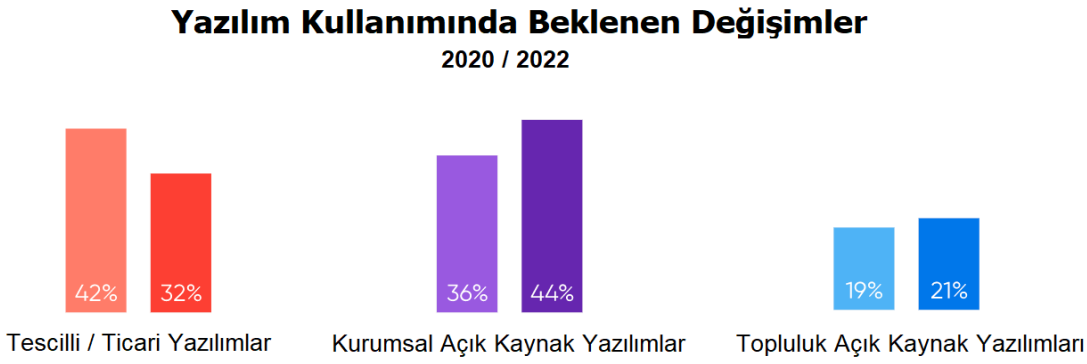
Ayrıca genellikle basit yapıları sayesinde uzun yıllar eskiyen donanımlarda verimli bir şekilde çalışabilirler. Bu sebeple bir açık kaynak göçünün sağlayabileceği ekonomik faydayı hesaplarken uzun dönem projeksiyonlar dikkate alınmalıdır.

[56] OpenLogic Inc. ve Navica (2006). *"Open Source Return On Investment: Achieving the Financial Promise of Open Source Software"*, ABD.

Bir açık kaynak göçü ile ilgili finansal analiz yapılırken şu giderleri göz önüne almak gerekir:

- Göç sonrası bakım destek hizmeti maliyetleri
- Ekstra donanım maliyetleri
- Göç sürecinde alınacak teknik destek hizmeti giderleri
- Eğitim ve dokümantasyon maliyetleri
- Daha sonra talep edilebilecek geliştirme ve iyileştirme giderleri

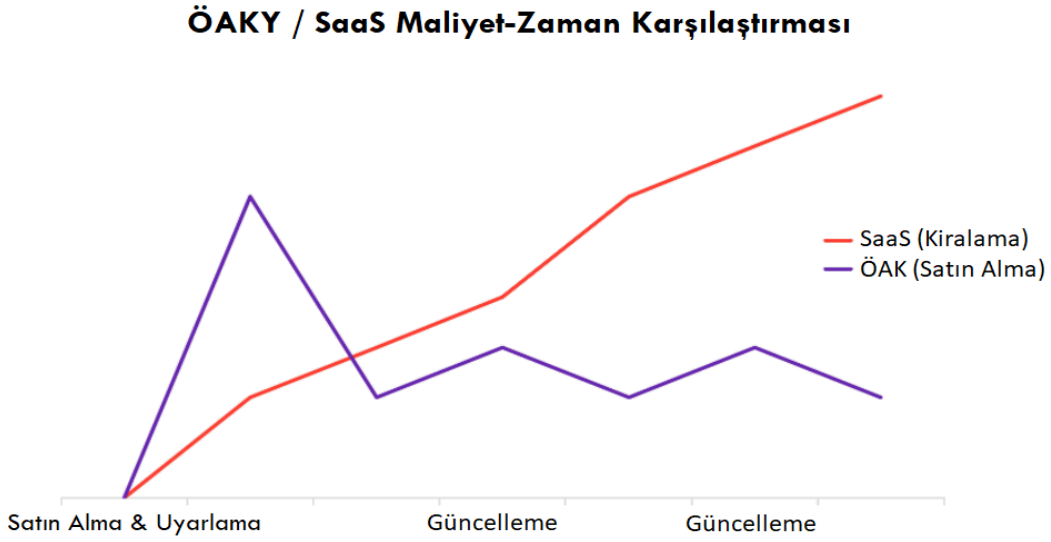
Her ne kadar tüm bu gider kalemleri açık kaynak bir sistem için fazla gibi gelse de aslında bu giderler ticari/tescilli yazılımlar için de geçerlidir. Hem de daha yüksek ücretlere tabi olarak. Ayrıca tüm bunların yanında her yıl (sözleşmeye göre değişebilir) lisans yenileme ücretleri de ticari/tescilli yazılım kullanıcıları için en büyük gider kalemi olmaktadır. Özel mülk yazılım satıcısı, sistemin yazılıma bağımlılığı oranına göre yüksek lisans yenileme ücretlerini kullanıcıya dayatabilir. Tescilli/ticari yazılımların bu tekelci ve bağlayıcı doğası, bu ürünlerin popülerliğinin her geçen gün azalmasına sebep olmaktadır. Bu kapsamda, Şekil 2.1 de 2020 ve 2022 yılları için ticari/tescilli ve ÖAK yazılım kullanım oranlarında beklenen değişimi görebiliriz.



Şekil 2.1: Ticari/tescilli ve ÖAK Yazılım Kullanım Oranı Projeksiyonu [57]

[57] Pimcore, https://pimcore.com/en/resources/insights/why-open-source (23.03.2021)

ÖAK yazılımlar uzun yaşam döngüsüne sahip ürünlerdir. Kararlı ve güvenli yapıları sayesinde uzun yıllar güncelleme almadan dahi sistemleri çalıştırabilirler. Bu sayede ÖAK yazılımların son kullanıcıya sağlayacağı ekonomik fayda genellikle uzun dönem faydalarıdır. Şekil 2.2'de verilen maliyet/zaman grafiğinden bu durum gözlemlenebilir.



Şekil 2.2: ÖAK / SaaS[58] Yazılımların Maliyet/Zaman Grafiği [59]

2.5.1. Ekonomik Fayda Örnekleri

ÖAK yazılımlı sistemlere göç, doğru adımlar izlenerek gerçekleştirildiğinde, kullanıcı tarafa kısa ve uzun vadede ekonomik fayda sağlar. Örneklendirmek gerekirse;

- Wisconsin, ABD'de Waukesha adlı bir internet hizmeti sağlayıcısı, veritabanı sunucularında çalışan ticari/tescilli yazılımların lisansını yenilemek için yılda yaklaşık 700.000 $ harcıyordu. Yani 10 milyon

[58] Hizmet olarak yazılım (SaaS), kullanıcıların bulut tabanlı ticari/tescilli yazılımları internet üzerinden bağlanarak kullanmasını sağlayan bir lisans dağıtım/kiralama modelidir.
[59] Pimcore, https://pimcore.com/en/resources/insights/why-open-source (23.03.2021)

doların biraz altında olan yıllık gelirinin %7'sinden fazlasını, yalnızca halihazırda sahibi olduğu yazılımların lisansını yenilmeye harcıyordu. Bu şirket şu anda açık kaynaklı platformlara geçiş sürecinde ve şimdiden lisans ücretlerinde tasarruf etmekte.[60]

- California Franchise Vergi Kurulu, yardım masası çalışanlarının sorunlu cihazlara uzaktan erişmek için kullandığı programın açık kaynak alternatifi olan VNC'ye geçiş yapmaya karar verdi. Bu ÖAK yazılımın ticari/tescilli muadili ise 66 $ idi. 5000'den fazla kullanıcıyı destekleyen sistemle elde edilen tasarruf miktarı yaklaşık 330.000 $ ı buldu.[60]

- Amazon, 2002 yılında Windows'tan Linux'a geçmeye karar verdi. E-ticaret devinin, Menkul Kıymetler ve Borsa Komisyonu'na yaptığı bildirime göre, 2002 yılı içinde teknoloji harcamalarını 71 milyon dolardan 54 milyon dolara düşürerek yaklaşık %25 oranında tasarruf ettiğini ifade etti.[61]

- 2019 yılında Güney Kore Hükümeti, sistemlerinde kullandıkları Windows'ların destek süresinin bitmesinin ardından 655 milyon dolarlık dev bir Linux göç planı hazırladıklarını duyurdu. Bu sayede lisans giderlerinin %72 oranında azalacağı öngörüldü.[62]

- Standish Group International adlı uluslararası bir IT danışmanlık şirketi, 2008 yılında, dünyada ÖAK sistemlere geçiş eğilimleri araştırması yaptı. Araştırma sonucuna göre; ÖAK yazılımlara geçiş oranının her geçen yıl arttığı, bu eğilimin tescilli/ticari yazılım şirketlerine dünya genelinde yıllık yaklaşık 60 milyar dolara mal olduğu açıklandı. Bu, yıllık

---

[60] Jerry Gartner, (2019). *"What Is Open Source Software And How Much Can It Save Us?"*, Gartner Technology LLC., https://gartnertechnology.com/blog/2019/01/26/open-source-software-much-can-save-us/ (25.03.2021)
[61] Stephen Shankland, (2002). *"How Linux Saved Amazon Millions"*, CNET, https://www.cnet.com/news/how-linux-saved-amazon-millions/ (25.03.2021)
[62] James Sanders, (2019). *"South Korean Government Planning Linux Migration as Windows 7 Support Ends"*, TechRepublic, https://www.techrepublic.com/article/south-korean-government-planning-linux-migration-as-windows-7-support-ends/ (25.03.2021)

1 trilyon doları bulan IT pazarı için yüzde 6'lık bir kayıp anlamına geliyordu. Yani açık kaynak yazılımlar özel mülk yazılım dünyasını yıkmıyor, ancak artık onlar için büyük ölçüde tehdit oluşturuyordu.[63]

2.5.2. Ekonomik Zarar Örnekleri

Şirketler, kurum ve kuruluşlar ÖAK sistemlere geçerek büyük ekonomik fayda sağlayabiliyorlar. Lisans ücretlerinin büyük bölümünden muaflık, lisans yenileme ücretlerinin olmaması ya da bakım destek giderlerinin daha az olması IT departmanları için önemli tasarruf kalemleri olabiliyor. Ancak kısmen ya da tamamen başarısız olmuş, hatta ters göçe sebep olabilen başarısız açık kaynak dönüşümleri, geçiş yapan tarafın daha vahim ekonomik tablolarla karşılaşmasına sebep olabiliyor.

2.5.2.1. IRS Açık Kaynak Göçü

ABD'nin vergi kurumu IRS, 2014 yılında hazırladığı bir plana göre; bir sisteminde kullandığı 190 yazılımdan 141 tanesini Linux tabanlı açık kaynak sistemlere geçireceğini açıkladı. Böylece tek bir sistemin açık kaynak dönüşümüyle bile, lisans ücretlerinden 5 yılda yaklaşık 12 milyon dolar tasarruf edileceğini belirtti. Ancak iyi planlanmayan bu göç operasyonu başarısızlıkla sonuçlandı. 2018 yılında, Hazine Bakanlığı müfettişlerinin bu başarısız göç operasyonuna yönelik hazırladıkları rapora göre; yetersiz yönetim, zayıf planlama, yeterli teknik desteğin alınmaması ve geçişi gerçekleştirecek IRS personelinin yeterli teknik bilgi ve eğitimden yoksun olması sebebiyle

[63] Dave Rosenberg, (2008). *"Study Finds Free Open Source Software Is Costing Vendors $60 Billion"*, CNET, https://www.cnet.com/news/study-finds-free-open-source-software-is-costing-vendors-60-billion/ (25.03.2021)

operasyonun başarısız olduğu ifade edilmiştir. Yine aynı rapora göre; 141 yazılımın açık kaynak dönüşümünden sadece 8'inin yapıldığı, bu sebeple kurumun planlanan 12 milyon dolar tasarruftan mahrum kaldığı belirtilmiştir. Ayrıca başarısız operasyon sebebiyle, dönüşüm ve entegrasyon hatalarının düzeltilmesi kapsamında alınan bakım destek hizmetleri nedeniyle IRS'in 814.272$'lık zarara uğratıldığı ifade edilmiştir.[64]

2.5.2.2. Münih Açık Kaynak Göçü

Münih'in göç hikayesi ise tam bir fiyasko olması ile ün kazanmıştır. 2003 yılında, Almanya'da Münih Belediye Meclisi, Microsoft'un yoğun lobi çalışmalarına rağmen, LiMux adlı Ubuntu-Linux türevi bir işletim sistemine ve diğer açık kaynaklı yazılımlara geçmeye karar verdi. 10 yıl olarak planlanan projeye; donanım güncellemeleri ve eğitim masrafları dahil 30 milyon € bütçe ayrıldı. Geçiş süreci 2006 yılında resmen başlatıldı. İlk başta ticari/tescilli yazılımlara göre 5 milyon Euro tasarruf edileceği öngörülmüştü. Ancak 2012 yılında hazırlanan bir rapora göre aslında 11.6 milyon € tasarruf edildiği açıklandı.[65] Ancak 2017 yılında belediye meclisi, açık kaynak sistemin yeteri kadar ekonomik fayda sağlamadığı, verimli olmadığı ve personelin sistemi benimseyemediği gerekçeleriyle tekrar Windows ve ürünlerine ters göç yapılmasını öngören planı onayladı. Plan kapsamında; 29.000 bilgisayarın, Linux'dan Windows 10 sistemine geçirilmesi ve LibreOffice yerine de 6000 Microsoft Office lisansının tedarik edilmesi isteniyordu.[66] Bu tersine göçün toplam maliyetinin, lisanslama ve diğer göç masraflarıyla beraber 86.1 milyon €'yu bulduğu belirtilmiştir.[66] 2020 yılında ise Münih Belediyesindeki partiler, bu

[64] Treasury Inspector General For Tax Administration. (2018). *"The Solaris to Linux Migration Project Was Delayed and Needs Improved Governance"*, ABD.
[65] Ed Bott, (2014*). "After a 10 Year Linux Migration, Munich Considers Switching Back to Windows and Office"*, ZDNet, https://www.zdnet.com/article/after-a-10-year-linux-migration-munich-considers-switching-back-to-windows-and-office/ (27.03.2021)
[66] Nick Heath, (2017). *"End of an open source era: Linux pioneer Munich confirms switch to Windows 10"*, TechRepublic, https://www.techrepublic.com/article/end-of-an-open-source-era-linux-pioneer-munich-confirms-switch-to-windows-10/ (27.03.2021)

sefer tekrar açık kaynak sisteme geri dönüş için yeni bir koalisyon anlaşmasını imzaladılar. Anlaşma kapsamında, mevcut ticari/tescilli yazılım lisanslarının süresi biter bitmez tekrar ÖAKY sistemlere geçilmesi üzerinde mutabakata varıldı.[67] Bu geri dönüşün ne kadara mal olacağı ve ne kadar süre alacağı ise henüz belli değil. Son olarak Münih'teki bu ters göçlerin, göçün teknik standartlarıyla bağdaşmadığı ve alınan kararların siyasi olduğu iddia edilmektedir. Özellikle 2017'de ki ters göç kararının, Microsoft'un siyasiler üzerindeki etkisi ile alındığı iddia edilmektedir.

2.6. AÇIK KAYNAK GÖÇÜNÜN TEKNİK ANALİZİ

Bilgi sistem altyapılarında göç, bir işletim ortamından, çoğu durumda daha iyi olduğu düşünülen başka bir işletim ortamına geçme sürecidir. Geçiş, yeni donanıma, yeni yazılıma veya her ikisine yükseltmeyi içerebilir.[68] Ayrıca IT altyapısında, kapsamlı tam bir geçiş süreci yürütebileceği gibi kademelendirilmiş ya da hibrit geçiş modelleri benimsenebilir.

2.6.1. Göç Süreç Hızı Modelleri

Göç süreciyle ilgili olarak planlanması gereken konulardan biri göç sürecinin hangi geçiş hızı yöntemiyle sürdürüleceğidir. Başlıca hız belirleme yöntemleri üç başlıkta toplanabilir.[69]

---

[67] Cathrin Schaer, (2020). *"Linux not Windows: Why Munich is shifting back from Microsoft to open source – again"*, ZDNet, https://www.zdnet.com/article/linux-not-windows-why-munich-is-shifting-back-from-microsoft-to-open-source-again/ (28.03.2021)
[68] Ben Cole, (2016). *"Migration"*, TechTarget, https://searchcio.techtarget.com/definition/migration (29.03.2021)
[69] Netproject Ltd. (2003). *"The IDA Open Source Migration Guidelines: Management Guidelines"*, s.19-20, İngiltere.

2.6.1.1. Big Bang Geçiş

Bu senaryoda, tüm altyapı ve son kullanıcılar aynı günde veya çok kısa bir zaman aralığında eski sistemden yeni sisteme geçirilir. Geçiş süreci genellikle sistem yoğunluğunun düşük olduğu tatil günlerinde yapılır. Bu yöntemin avantajı, farklı erişim yöntemlerinin aynı anda kullanılması ihtiyacının olmamasıdır. Eski sisteme dair tüm protokol ve yazılım politikaları bu yöntemde genellikle göz ardı edilebilir. Ayrıca geçişi gerçekleştirecek personelin iki sistem arasında harcadıkları enerji minimumdur. Bu yöntemin avantajları olduğu gibi dezavantajları da vardır. En önemli dezavantaj ise bu geçiş hızı yönteminin sistem için çok daha riskli olmasıdır. Aynı anda kontrol edilmesi gereken çok daha fazla değişkenin olmasının yanı sıra sistemi kısa sürede ayağı kaldırma ihtiyacı ve zaman baskısı hata yapma riskini artırır.

Bu göç hızı yöntemi genelde küçük işletmeler için uygundur. Özellikle çok büyük ölçekli göç operasyonlarında kaçınılması gereken bir yöntemdir. İyi planlama yapıldığı takdirde küçük ölçekli geçişler için uygun bir yöntem olabilir.

2.6.1.2. Kademeli Geçiş

Bu yöntemde göç, fazlar halinde tekrarlanır. Altyapı ve son kullanıcılar gruplara ayrılarak kademeli olarak yeni sisteme geçirilir. Veri paylaşımını ve grup çalışma sorunlarını en aza indirmek için tüm işlevsel grupların birlikte hareket ettirilmesi muhtemeldir. Uygun grup boyutları seçilerek riskler kontrol altına alınabilir ve kaynaklar yönetilebilir.[70]

[70] Netproject Ltd. (2003). *"The IDA Open Source Migration Guidelines: Management Guidelines: Phased Transition In Groups"*, s.20, İngiltere.

2.6.1.3. Kullanıcı Bazlı Geçiş

Esasen grup geçiş seçeneğiyle aynıdır, ancak burada grup tek kişilik kullanıcı ya da sistemi içerir. Bu yöntemin düşük bir kaynak gereksinimi vardır, ancak verimsizdir.[71] Büyük ölçekli göç operasyonu gerçekleştirecek işletmelere hitap etme olasılığı düşüktür. Yine de pilot göç uygulamalar için uygun olabilir. Özellikle kapsamlı geçişten önce sistem ve son kullanıcı tepkisini ölçmek için uygulanacak küçük ölçekli pilot göç süreçleri için kullanılabilecek bir yöntemdir.

Hangi geçiş hızı yöntemi kullanılırsa kullanılsın, eski ve yeni sistemlerin bir süre yan yana çalışması gerekebilir. Geçiş döneminde sistem işleyişinin aksamaması için eski ve yeni sistemlerin birlikte çalışmasını sağlayan bir geçiş stratejisine sahip olmak önemlidir.[72]

2.6.2. Göç Kapsam Modelleri

Bir kurum ya da kuruluşun, altyapısında gerçekleştirilecek olan göçün kapsamı isteğe göre ölçeklendirilebilir. Sistemin hangi parçalarında geçiş işleminin gerçekleştirileceği, o sistemin yapısına, mimarisine ve ihtiyaçlarına göre belirlenir. Ayrıca son kullanıcıların beklenti ve alışkanlıkları da bu konuda belirleyici olabilir. Sistem altyapısında halihazırda kullanılmakta olan diğer tescilli/ticari yazılımların ÖAK yazılımlarla göstereceği uyumluluk davranışları da göçün kapsam modelini etkileyen önemli hususlardandır. Tüm bu

[71] Netproject Ltd. (2003). *"The IDA Open Source Migration Guidelines: Management Guidelines: User by User Transition"*, s.20, İngiltere.
[72] Netproject Ltd. (2003). *"The IDA Open Source Migration Guidelines: Management Guidelines"*, s.20, İngiltere.

parametrelere göre göçün kapsamı değişkenlik gösterebilir. Bu değişkenlik iki farklı göç kapsam modeli ortaya çıkarmıştır.

2.6.2.1. Kısmi / Hibrit Göç Modeli

Bu modelde sistem, ÖAK yazılımlar ile ticari/tescilli yazılımları aynı anda bünyesinde bulunduracak şekilde dizayn edilir. Yani sistemin ve son kullanıcıların beklentilerine göre bazı özel mülk yazılımlar sistemde bırakılarak ÖAK dönüşümleri yapılmaz. Bunun sebebi; işletme bünyesinde yürütülen işlere özgü geliştirilmiş olan yazılımların ve bu yazılımların bağımlı olduğu işletim sistemlerinin varlığı olabilir. Ya da altyapıda kullanılan herhangi bir donanım, özel bir ticari yazılıma göre tasarlanmış olabilir. Ayrıca son kullanıcıların genel kullanım alışkanlıkları da bu özel mülk yazılımların bazılarını sistemde tutmayı zorunlu kılabilir. Tüm bu sebeplerden dolayı, sistemin o kısmı için ÖAK dönüşümü yapılmayıp ÖAK yazılımlara entegrasyonu sağlanır. Ya da ÖAKY ve ticari/tescilli yazılım entegrasyonu da yapılmayıp sistemin işleyişini bozmayacak şekilde ayrı görevleri işleyen tamamen bağımsız sunucu blokları oluşturulabilir. Böyle bir geçiş modeli kısmi ya da hibrit model olarak adlandırılır.

2.6.2.2. Tam Göç Modeli

Sistemde kullanılan istisnasız tüm yazılımların, işletim sistemlerinin, sunucu yazılımlarının, yazılım araç setlerinin, son kullanıcı uygulamalarının ve veritabanı yazılımlarının ÖAKY sistemlere geçirildiği modele tam geçiş modeli denir. Böyle bir göç işleminden sonra sistemde hiçbir özel mülk yazılıma yer verilmez. Dolayısıyla entegrasyon süreci yoktur. Tam göç modelinde sistemde uyumsuzluk gibi sorunların oluşma ihtimali ya yoktur ya da azdır. Ayrıca hibrit

modele göre iş yükü daha az olduğundan daha hızlı ve verimli bir model olup sağlayacağı ekonomik fayda da daha fazladır. Genellikle kademeli faz geçiş yöntemi ile beraber benimsenir. Özellikle büyük ölçekli göçlerde hem Big Bang hem de Tam Geçiş Modeli aynı anda kullanılmaz. Aksi takdirde bu tarz bir göç modelinin başarı şansı azdır.

2.6.3. Göç Entegrasyon Araçları

Göç sürecinde ve devamında, geçiş ve entegrasyon sürecinin kolaylaştırılması ve muhtemel sorunların azaltılması için bazı açık kaynak yazılım entegrasyon araçları kullanılabilir. Özellikle hibrit modellerde, ÖAK yazılımlar ile özel mülk yazılımlar arasında gerekli iletişimi ve uyumu sağlamak için bu çözümlerin kullanılması yararlı olabilir.

2.6.3.1. SAMBA

Samba, GNU GPL lisansı altında yayınlanmış, UNIX tabanlı açık kaynak işletim sistemleri ile Windows NT gibi ticari/tescilli işletim sistemleri arasındaki iletişimi sağlayan özgür ve açık kaynak bir ağ sunucu uygulamasıdır. Samba, UNIX benzeri bir sisteme Windows makinelerin erişebilmesini mümkün kılar. SMB ağ protokolünün özgür ve açık kaynak olarak yeniden tanımlanmış hali olup Windows istemcileri için dosya ve yazdırma hizmetlerini sağlar. Bir Etki Alanı Denetleyicisi (DC) gibi çalışabilir ve Microsoft Windows Server DC'si ile entegre olabilir. Dördüncü sürümü ile sonraki sürümleri, Active Directory ve Microsoft Windows NT gibi etki alanlarını da destekler. Samba, MacOS sunucusu ve istemcisi (Mac OS X 10.2 ve üstü) dahil olmak üzere Linux, Solaris, AIX ve BSD varyantları gibi çoğu UNIX-like

sistemde çalışır.[73] Samba, Linux'un neredeyse tüm dağıtımlarında standarttır ve diğer Unix tabanlı işletim sistemlerinde de temel bir sistem aracı olarak kullanılır. Genel olarak açık kaynak sistemler için bir dosya ve yazdırma sunucusu olarak kullanılmasına rağmen Microsoft'un SMB protokolünü uygulayarak bir Windows SMB sunucusunun fonksiyonlarının tamamını da yerine getirebilir.

Samba, Windows NT Etki Alanı Denetleyicisi olarak da hareket edebildiği için DC verilerini LDAP yardımıyla erişilen bir dizinde depolayabilir. Samba ayrıca, SMB ağlarıyla ilgili sorunları teşhis ederken, Windows sunucularından toplu dosya geçişi yaparken çok yararlı olan, komut dosyası oluşturmaya uygun SMB istemci araçlarını da sağlar.[74] Tüm bu sahip olduğu özellikler bakımından Samba, özellikle hibrit göç modellerinde sistem yönetimi, entegrasyon, uyum ve kolay kullanım için tercih edilebilecek önemli göç yazılım aracıdır.

2.6.3.2. NSS ve PAM

NSS, UNIX benzeri sistemlerde, kullanıcı adları ve grup adları gibi veriler aratılırken farklı alan adı hizmetlerinin birlikte kullanılmasına izin verir. Ayrıca NSS, altyapıdaki sistemleri ortak veri tabanlarına ve DNS mekanizmaları kaynaklarına bağlayabilir. PAM ise düşük seviyeli kimlik doğrulama yöntemlerini, yüksek seviyeli bir uygulama programlama arayüzüne entegre edebilen bir mekanizmadır. PAM de NSS gibi, UNIX-like işletim sistemlerinde yaygın olarak kullanılır. PAM, kimlik doğrulama ve yetkilendirme sürecini yapılandırmada büyük esneklik sağlar.

---

[73] Samba Org. *"Samba 4.12 Features Added / Changed"*, SambaWiki https://wiki.samba.org/index.php/Samba_4.12_Features_added/changed (08.04.2021)
[74] Netproject Ltd. (2003). *"The IDA Open Source Migration Guidelines: Tools"*, s.70, İngiltere.

2.6.3.3. OpenLDAP

OpenLDAP, BSD türevi bir lisans altında yayınlanmış olan LDAP protokolünün özgür ve açık kaynak bir uygulamasıdır.[75] Platformdan bağımsız bir protokol olup birçok Linux dağıtımı, LDAP desteği için OpenLDAP yazılımını kullanır. BSD-like sistemlerin yanı sıra AIX, Android, HP-UX, macOS, Solaris, Microsoft Windows gibi sistemler üzerinde de çalışabilir. Böylece farklı platformlar arası LDAP hizmetini sistem üzerinde yürütebilir.

2.6.3.4. GNU/Linux SMBFS

Samba, açık kaynak bir sistemin Windows istemcilerine dosya hizmeti sağlamasına izin verir. SMBFS ise tam tersi şekilde, bir ÖAK sisteminin Windows sunucuları üzerindeki dosyalara erişmesine izin verir.[76] Windows dosya sistemleri tarafından kullanılan erişim denetimi modeli, GNU / Linux ve diğer açık kaynak sistemler tarafından kullanılandan modelden farklıdır. Bu nedenle farklı platformlar arasında kullanılacak SMBFS ile elde edilebilecek fayda sınırlıdır.

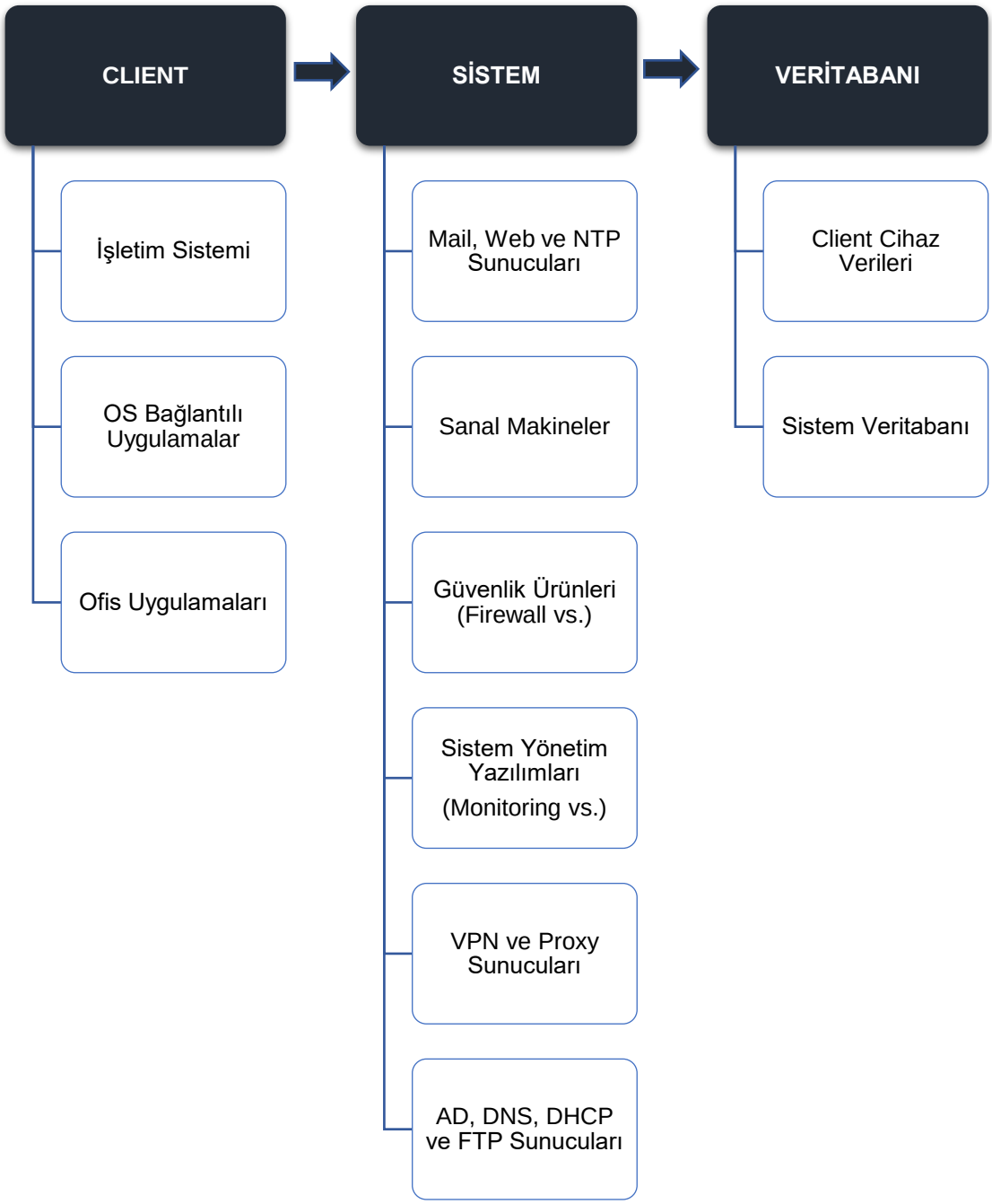
SMBFS dosya sistemi, Linux uyumlu bir SMB dosya sistemidir. Başka herhangi bir sistemde çalışmaz. Ayrıca SMBFS son birkaç yıldır geliştiricilerden güncelleme almadı. Bunun yerine, kernel üzerindeki CIFS protokolünün başka bir uygulamasına odaklanıldı.[77]

---

[75] OpenLDAP Foundation. (2003). *"The OpenLDAP Project Overview"*, ABD. https://www.openldap.org/ (03.04.2021)
[76] Netproject Ltd. (2003). *"The IDA Open Source Migration Guidelines"*, s.70, İngiltere.
[77] Samba Org. *"The SMBFS Filesystem"*, https://www.samba.org/samba/smbfs/ (05.04.2021)

2.6.4. Açık Kaynak Göç Projeksiyonu

Bir açık kaynak göçünün kapsamı ve süreci Şekil 2.3'te verilmiştir.



Şekil 2.3: Açık Kaynak Sisteme Göç Projeksiyonu

2.6.5. Uygulama Göçü

Son kullanıcı cihazları ile sistem cihazlarında kullanılan yazılım ve uygulamaları, ÖAK muadilleri ile değiştirmek her zaman mümkün olmayabilir. Sistem ve veri sahibi tarafın ihtiyaçları, donanım bağımlılığı ve hizmete özgü uygulamalar sebebiyle bazen ÖAK olmayan yazılımların sistemde tutulması gerekebilir. Bu durum uygulama göçü ile ilgili farklı yaklaşımlar benimsenmesini gerektirir.

2.6.5.1. Açık Kaynak Eşdeğeri Olan Uygulamalar

Windows üzerinde çalışan Microsoft Office, Internet Explorer, Photoshop, AutoCad, Windows Media Player gibi bazı ticari/tescilli uygulamaların, OpenOffice, LibreOffice, Mozilla Firefox, Gimp, FreeCAD, VLC gibi açık kaynak çözümleri vardır. İhtiyaçları karşıladığı takdirde bu ticari/tescilli uygulamalar ÖAK alternatiflerine geçirilebilir. Bu yaklaşım hem daha fazla ekonomik fayda sağlar hem de satıcı bağımlılığını en aza indirir.

2.6.5.2. Açık Kaynak Platformlarda Çalışabilecek Uygulamalar

Bazı özel mülk uygulamaların açık kaynak platformlarda da çalışan sürümleri olabilir. Açık kaynak sistemler üzerinde çalışabilen ancak açık kaynak olmayan bir yazılımın, ihtiyaçları tam olarak karşılayabilecek şekilde özgür ve açık kaynak bir alternatifi yoksa bu yazılım, göç sonrası sistemde tutulabilir. Ancak bu uygulamanın açık kaynak bir sistemde tam verimle çalışabileceği garanti edilmez.

2.6.5.3. Uzaktan Erişim Yöntemi

Bu yöntemde, ÖAK alternatifi olmayan ve son kullanıcı ya da sistem tarafından ihtiyaç duyulan uygulamaları sistemde tutmak hedeflenir. Açık kaynak olmayan uygulama yine açık kaynak olmayan bir işletim sisteminin koştuğu uzak bir sunucuya kurulur.

Uygulamayı kullanmak isteyen son kullanıcı, bu sunucuya uzak masaüstü bağlantısı ile bağlanıp kendi donanım kaynaklarını kullanmadan uygulama üzerinde işlemlerini gerçekleştirebilir. Bu yöntem ile hem son kullanıcıların ihtiyaçları karşılanmış olur hem de lisans ücretlerinden tasarruf edilir. Ancak bu metot hem ağ üzerindeki trafiği hem de donanım ihtiyacını artırabilir.

2.6.5.4. Uygulama Sanallaştırma Yöntemi

Uygulama sanallaştırma, uygulamaların bir paket haline getirilerek sunucu üzerinden son kullanıcıların bilgisayarlarına dağıtılmasına denir. Bu yöntemde uygulamalar, son kullanıcıların cihazlarına yüklenmez sadece önbelleklerinde tutulurlar.[78] Uygulama yine uzak bir sunucuda tutulur ve buradan istemci cihazlara atanır. Uzak masaüstü erişiminin aksine, bu yöntemde sunucunun kaynakları yerine son kullanıcı cihazının donanım kaynakları kullanılır. Ayrıca uzaktan erişim yönteminin ağ üzerinde oluşturduğu trafik yoğunluğu, uygulama sanallaştırmada daha kabul edilebilir düzeydedir. Sistem üzerinde ticari/tescilli yazılım ve ÖAKY entegrasyonu yapma ihtiyacı daha azdır. Daha az entegrasyon demek daha az hata, daha az entegrasyon yazılımı ve daha kolay sistem yönetimi demektir. Üstelik uygulama sanallaştırma sunucuları, geleneksel uzak masaüstü sunucularına göre daha

[78] Mehmet Parlakyiğit, *"Microsoft App-V Uygulama Sanallaştırmaya Giriş"*, s.2.

az kaynak ve daha az bant genişliğine ihtiyaç duyar. Bu yöntem daha az cihaz, daha az kurulum gerektirdiğinden sistem yöneticilerinin iş yükünü de azaltır. Uygulamalar, kurulmasına gerek kalmadan taşınabilir ortama veya bilgisayarlar arasında aktarılabilir ve taşınabilir bir yazılım haline de gelir.[79]

2.6.5.5. Uygulamayı ÖAK Platformda Yeniden Derleme

Kurum ya da kuruluşun, altyapısında kullandığı ticari/tescilli uygulamanın kaynak kodlarına erişimi varsa, bu kodlar uygun ÖAK bir platformda tekrar derlenip çalıştırılabilir. Burada önemli olan kodun yazıldığı dilin açık kaynak platformda da desteklenmesidir. Derlemeyle ilgili asıl sorun ise kod bloklarının ve kütüphanelerinin işletim sistemine çok fazla bağımlı kod içermesidir. Bu bağımlı kodlar, işletim sisteminin grafik arayüzü gibi daha ayırt edilebilir kütüphanelerine de atıf yapabilir; iletişim protokolleri gibi daha temel araçlara da atıf yapabilir. Her iki durumda da bu yöntem çok fazla enerji ve zaman gerektiren bir süreçtir. Büyük ölçekli yazılım setleri için kullanılmayan bu yönteme göre, uzak erişim ve uygulama sanallaştırma daha iyi bir alternatiftir.

2.6.6. Alternatif Göç Teknikleri

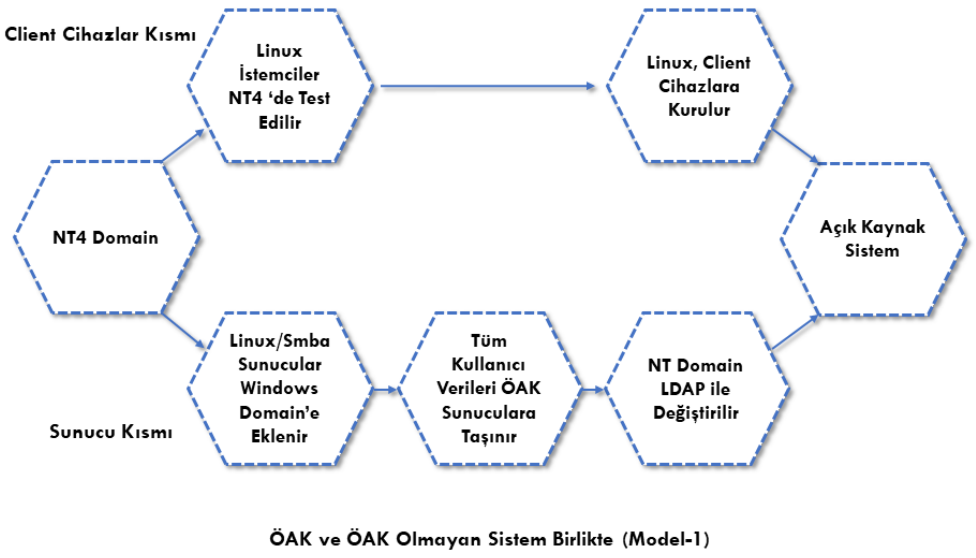
Bir açık kaynak göçünü, sistemin genel yapısını bozmadan gerçekleştirmek için iki ayrı yol izlenebilir. Hangi yolun tercih edileceği, sistemin fiziksel altyapısına, büyüklüğüne, mimarisine ve ağ yapısına göre değerlendirilmelidir.

---

[79] Credit Suisse Securities LLC. (2007). *"Desktop Virtualization Comes Of Age"*, Equity Research, ABD.

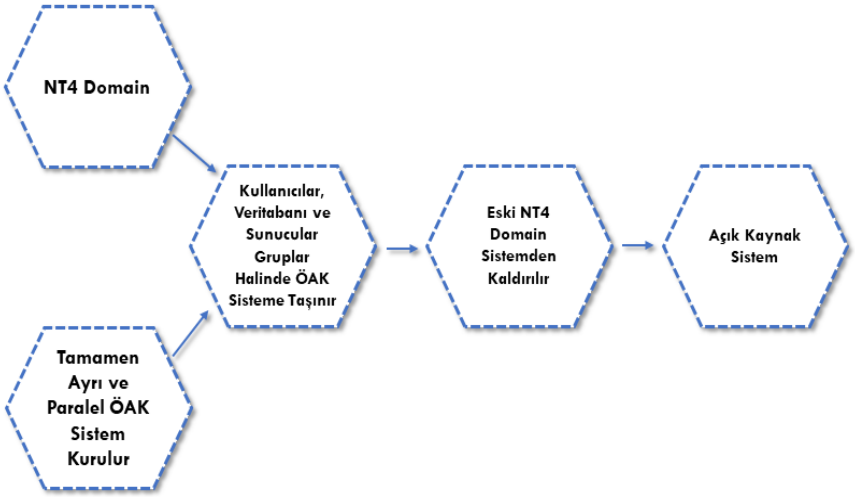
İlk yöntemde; açık kaynak makineler mevcut Windows etki alanlarına eklenir. Daha sonra sunucular ve kullanıcılar kademeli olarak sisteme dahil edilirken eski sunucular ve istemci cihazlar sistemden çıkarılır. Bu aşamada Samba kullanılarak, mevcut ortamda herhangi bir değişiklik yapmadan Windows sistemi yerine kullanılabilen güçlü ve düşük maliyetli bir dosya ve yazdırma sunucu hizmeti sağlanmış olur. Bu yöntemde hata riski azdır ancak iki ayrı platformu aynı anda ayakta tutmak yönetim açısından zordur ve fazladan emek gerektirir. Ayrıca, geçiş sırasında eski ve yeni sistemler arasında daha sıkı entegrasyon gerektireceğinden, önemli ölçüde daha fazla planlama ve zaman gerektirir.

İkinci yöntemde; mevcut sisteme tamamen paralel bir açık kaynaklı yapı oluşturulur. Sistemdeki sunucular, kullanıcılar ve veritabanı hiçbir entegrasyona tabi tutulmadan gruplar halinde bu sisteme taşınır. Bu ilk yönteme göre çok daha basit bir yoldur. Daha az zaman ve emek gerektirip Big Bang geçiş yöntemiyle beraber kullanılabilir. Şekil 2.4 ve 2.5'te bu iki yöntemin şeması verilmiştir.



Şekil 2.4: İki Sistemin Birbirine Entegre Edildiği Geçiş Modeli [80]

[80] Netproject Ltd. (2003). *"The IDA Open Source Migration Guidelines"*, s.65.



Şekil 2.5: İki Sistemin Ayrı (Paralel) Olduğu Geçiş Modeli [81]

2.7. TÜRKİYE'DE ÖAK YAZILIMLARA KURUMSAL İLGİ VE BAŞARILI GÖÇ OPERASYONLARI

Türkiye'de son yıllarda ÖAK yazılımlara olan kurumsal ilgi, özellikle Pardus 'un da etkisiyle oldukça artmıştır. Yerel yazılım kültürünün geliştirilmesine yönelik verilen destekler de Türkiye'de açık kaynak kültürünün oluşmasına ortam hazırlamıştır. Her ne kadar Türkiye'de özgür ve açık kaynak ekosistemi her geçen gün gelişse de hâlâ sahipli/ticari/tescilli yazılımları geliştiren/dağıtan firmalar, sektördeki ağırlıklarını büyük ölçüde korumaktadır. Bu günlerde ÖAK sistemlere geçiş konusunda pozitif hava oluşmuş olsa da gerekli yasal düzenlemelerin eksikliği, kurumların açık kaynak göçü konusunda çekimser kalmasına sebep olabiliyor. Ancak açık kaynak hareketini teşvik edecek girişimler de bu durumu tersine çevirebilir. Örneğin; Sanayi ve Teknoloji Bakanı Mustafa VARANK, Türkiye Açık Kaynak Platformunun düzenlediği çevrim içi seminerin açılışında yaptığı konuşmada, Türkiye Açık Kaynak Platformunun iki yıllık çalışmalarına kalkınma ajanslarının 30 milyon

[81] Netproject Ltd. (2003). *"The IDA Open Source Migration Guidelines"*, s.65.

lira katkı sunacağını ifade etmiştir.[82] Ayrıca altyapılarında özel mülk yazılım yerine, özgür ve açık kaynak yazılımlar kullanmayı tercih eden kamu kurumları da mevcuttur.

2.7.1. Millî Savunma Bakanlığı

MSB, ASAL ve şubelerinde 2007 yılından beri işletim sistemi olarak Pardus kullanılmaktadır. MSB, gerçekleştirmiş olduğu bu açık kaynak göçü ile Pardus 'un ilk kurumsal kullanıcısı olmuştur.[83] Bu geçişle beraber sağlanan toplam tasarruf miktarının 2 milyon $ olduğu ifade edilmiştir.[84] Göç kapsamında; ASAL'ın, Ankara ve Türkiye çapında tüm şubelerindeki toplam 625 sunucu ile 4 bin 500 son kullanıcı bilgisayarının açık kaynak dönüşümü yapılmıştır.[83]

2020 yılında ise, Havelsan ve MSB arasında imzalanan iş birliği protokolüne göre, 2009'dan beri kullanılan Pardus cihazlara ilave 10 bin cihazın daha Pardus sistemine geçirilmesi konusunda anlaşmaya varılmıştır. Protokol kapsamında; mevcut mimarinin güncellenmesi, yaygınlaştırılması ve güvenlik hizmetlerinin verilmesi de kararlaştırılmıştır. Ayrıca projenin üç yıl sürmesi ve cihazların tamamının Pardus Kurumsal Sürümünü kullanması kararlaştırılmıştır.[85]

---

[82] Sevgi C. Gökkoyun, (14.04.2020). *"Bakan Varank: Türkiye Açık Kaynak Platformu çalışmalarına kalkınma ajanslarımız 30 milyon lira katkı sunacak"*, Anadolu Ajansı, https://www.aa.com.tr/tr/ekonomi/bakan-varank-turkiye-acik-kaynak-platformu-calismalarina-kalkinma-ajanslarimiz-30-milyon-lira-katki-sunacak/1804321 (14.04.2021)

[83] İbrahim Eser, (2011). *"Kamuda Özgür Yazılım Kullanımı: Dünyada Ve Türkiye ' deki Örnekleri Üzerinden Bir Değerlendirme"*, s.108, RTÜK, Ankara.

[84] NTV. (2009). *"MSB, Pardus ile 2 milyon dolar tasarruf etti"*, https://www.ntv.com.tr/turkiye/msb-pardus-ile-2-milyon-dolar-tasarruf-etti,rE0SKD34VkGRgE_uv3votw (14.04.2021)

[85] Göksel Yıldırım, (21.11.2020). *"Millî Savunma Bakanlığı bilgisayarları yerli işletim sistemine geçiyor"*, Anadolu Ajansı, https://www.aa.com.tr/tr/turkiye/milli-savunma-bakanligi-bilgisayarlari-yerli-isletim-sistemine-geciyor/2051148# (14.04.2021)

2.7.2. Afet ve Acil Durum Yönetimi Başkanlığı (AFAD)

2020 yılında, Afet ve Acil Durum Yönetimi Başkanı Mehmet Güllüoğlu, Kamu Bilişim Derneğince dijital ortamda düzenlenen Kamu Bilişimi Dijital Zirvesi'nde yaptığı konuşmada, 28 il müdürlüğünde Pardus sistemine geçişin yapıldığını açıkladı.[86]

AFAD'ın bilişim alanında gururla anlatabileceği birçok çalışması olduğunu aktaran Güllüoğlu ayrıca diğer alanlarda da Pardus ile uyumlu yazılımları tercih ettiklerini aktardı.[86] Ayrıca bu kapsamda, AFAD'da kullanılan her yazılımın Linux tabanlı olmasına önem verdiklerini, Pardus ile uyumlu bir şekilde çalışacağına inanılan yazılımları tercih ettiklerini, bu sebeple de veritabanı için açık kaynak çözümleri ve uygulamaları kullanmaya başladıklarını ifade etmiştir.[86] Güllüoğlu ayrıca, kamu bütçelerinin yazılım lisanslarına değil bu konuda yetişmiş insan kaynağına aktarılması gerektiğini ifade etmiştir.[86]

2.7.3. Radyo ve Televizyon Üst Kurulu (RTÜK)

Radyo ve Televizyon Üst Kurulu, 2008 yılından önce radyo ve televizyon yayınlarını video kasetler vasıtasıyla arşivlemekteydi. Video Kayıt Sisteminin (VKS) gelişen teknolojinin çok gerisinde kalması, arıza sıklığı, arşiv yönetim zorluğu ve yayın kayıpları gibi sebeplerden ötürü yeni bir sisteme ihtiyaç duyulmuş, bu kapsamda tamamen açık kaynak yazılım ve işletim sistemlerinin kullanılacağı Sayısal Kayıt ve Arşiv Sistemi Projesi (SKAAS-I)

[86] Orhan O. Gemici, (20.05.2020). *"AFAD Başkanı Güllüoğlu: 28 İl Afet ve Acil Durum Müdürlüğünde Milli İşletim Sistemi Pardus 'a Geçildi"*, Anadolu Ajansı, https://www.aa.com.tr/tr/turkiye/afad-baskani-gulluoglu-28-il-afet-ve-acil-durum-mudurlugunde-milli-isletim-sistemi-pardusa-gecildi/1847557 (15.04.2021)

2006 yılında RTÜK ile TÜBİTAK arasında imzalanan sözleşme ile başlatılmıştır. Söz konusu proje, 2008 yılında kabul testlerinin tamamlanması ile de hizmete alınmıştır.

2.7.3.1. Sayısal Kayıt Arşiv ve Analiz Sistemi (SKAAS)

Üst Kurulun görev, yetki ve sorumluluklarının yerine getirilmesi kapsamında radyo ve televizyon yayınlarının arşiv ve analiz süreçleri 2008 yılından önce standart Video Kayıt Sistemleri (VKS) marifeti ile yürütülmekte idi. Bu sistemde; yayınlar bir personel tarafından manuel olarak video kasetlere kaydedilip arşivlenmekteydi. Video Kayıt Sistemlerinin gelişen teknolojinin gerisinde kalması, arıza sıklığı, arşiv yönetim zorluğu ve yayın kayıpları gibi sebeplerden ötürü yeni bir sisteme ihtiyaç duyulmuştur.

Kurumun yayın izleme ve denetleme yükümlülüğünün daha etkin bir şekilde yerine getirilmesi amacıyla; yeni sistem ihtiyacını karşılamak üzere Sayısal Kayıt ve Arşiv Sistemi Projesi (SKAAS-I) 2006 yılında RTÜK ile TÜBİTAK arasında imzalanan sözleşme ile başlatılmış olup 2008 yılında kabul testlerinin tamamlanması ile sistem hizmete alınmıştır.

SKAAS'ın ilk versiyonu olan SKAAS-I projesi ile 110 TV ve 80 Radyo yayını sisteme kaydedilip arşivlenebilmekte idi. SKAAS Projesi zaman içerisinde kapasite artırımı, yeni teknolojilerin eklenmesi, daha fazla platformdan yayın takibi ve analiz yeteneği gibi önemli teknolojileri bünyesine katarak günümüzde üçüncü versiyonu olan SKAAS-III olarak hizmet vermektedir.

Bugün SKAAS-III Sistemi ile yaklaşık 300 uydu TV, 90 radyo, 90 karasal TV olmak üzere yaklaşık 480 yayın sisteme kaydedilmekte olup bu yayınlar sistemde 1 yıl süreyle tutulmaktadır. Sisteme alınan yayınların arşivlenmesi, analizi, raporlanması, raporların üst kurula sunulmasını içeren tüm süreç entegre bir şekilde SKAAS-III sistemi üzerinde yapılmaktadır. Ayrıca sisteme sonradan dahil edilen 8 ayrı analiz modülü ile de yayınların; video, ses, reklam, anahtar kelime ve anahtar kare yönünden analiz edilmesine imkân tanınmıştır.

SKAAS sistemi ile birçok kaynaktan yayınlar alınabilmekte ve sisteme kaydedilmektedir. Halihazırda TURKSAT, EUTELSAT, HOTBIRD karasal/analog, Kablo TV ve IPTV ortamlarından yapılan yayınlar sisteme alınıp kaydedilmektedir. Yayın alış sistemi ile diğer sistem arayüzleri de yine açık kaynak platformlar üzerinde geliştirilmiştir.

SKAAS Projesi, günümüzün bilişim teknolojileriyle, millî bir proje olarak tasarlanıp geliştirilmiştir. Kullanıcı bilgisayarları ve veritabanı dâhil tüm sistem altyapısında, açık kaynak yazılımlar ve açık kaynak işletim sistemleri kullanılmıştır. Bu sebeple yazılımlar için herhangi bir lisans bedeli veya lisans yenileme ücreti ödenmemektedir.

SKAAS siteminde, 300 ‘e yakın son kullanıcı cihazı Linux kullanmakta olup kullanıcı arayüzleri de yine Linux uyumlu olarak tamamen açık kaynak platformlar üzerinde geliştirilmiştir. Aynı şekilde sistem üzerinde, mail sunucusu olarak Zimbra, kimlik doğrulama ve DNS hizmetleri için ise FreeIPA kullanılmaktadır. Sunucu makinelerde CentOS’un kullanıldığı sistemde veritabanı yönetimi için ise açık kaynak PostgreSQL kullanılmaktadır.

SKAAS sisteminde yine açık kaynak platformlar üzerinde geliştirilen birçok analiz modülü de bulunmaktadır:

- Video Klip Yakalama (VKY)
- Video Klip Yakalama-Ses (VKY-Ses)
- Video Program Analizi (VPA)
- Video Çekim Analizi (VÇA)
- Ses Seviye Analizi (SSA)
- Radyo Reklam Analizi (RRA)
- Video Metin Okuma (VMO)
- Anahtar Kelime Yakalama (AKY)

SKAAS ağı ile RTÜK'ün mevcut bilgi işlem ağı birbirinden tamamen bağımsızdır. SKAAS, kapalı network bir sistem olup sadece uçtan uca yerel yayıncı stüdyolarındaki kayıt cihazlarına bağlıdır. Bu ağın izlenmesi ve yönetimi de yine ÖAK bir yazılım olan Nagios ile sağlanmakta olup bu kayıt cihazları da Linux ile çalışmaktadır.

SKAAS sistemi ölçeklenebilir ve genişletilebilir bir altyapı mimarisine sahiptir. Böylece mevcut 650 radyo televizyon yayınını sisteme alma kapasitesi ile 3PB'lık arşiv kapasitesini artırmak, sistemin mimarisini değiştirmeye gerek duymadan da mümkündür. Açık kaynak yapısı sayesinde ise sistemde yapılacak güncelleme ve bakım hizmetleri, ilgili kurum personeli tarafından da gerçekleştirilebilmektedir.

Bu bağlamda SKAAS sistemi, büyük bölümünde özgür ve açık kaynak yazılım ve işletim sistemlerinin kullanıldığı, yaklaşık 13 yıldır 7 gün 24 saat kesintisiz hizmet veren Türkiye'deki en önemli açık kaynak projelerden biri olmuştur.

2.7.4. Belediyeler

Kamu kurumlarında olduğu gibi bazı belediyeler de açık kaynak sistemleri altyapılarında kullanmaya başlamıştır. Bunlara, Eyüpsultan Belediyesi, Pendik Belediyesi ve Ankara Büyükşehir Belediyesi örnek olarak verilebilir.

Ankara Büyükşehir Belediyesi, Alo153 Mavi Masa Hattında kullanılan cihazlar Pardus ile çalışmakta olup buradaki sunucular da açık kaynak yazılımlarla çalışmaktadır. [87] Otobüs sefer sorgulama hizmetleri ve kamera güvenlik sistemleri de yine Pardus ile çalışmaktadır.[87]

Pendik Belediyesi altyapısında Pardus ile beraber; Microsoft Exchange yerine açık kaynak Zimbra, sunucularda Linux ve Lider Ahenk gibi diğer açık kaynak yazılımlara ve işletim sistemlerine yer vermiştir.[88]

2015 yılında, Eyüpsultan Belediyesi, açık kaynaklı yazılımlara göç etmeye başladı. Göç kapsamında, son kullanıcı bilgisayarlarında Pardus, e-posta sunucusu olarak açık kaynak Zimbra, veritabanında ise PostgreSQL kullanmaya başladı.[89] Belediyeden yapılan açıklamaya göre, Eyüpsultan Belediyesi sistem altyapısında ve son kullanıcı bilgisayarlarında Pardus ve açık kaynak kullanarak 2 milyon $ tasarruf edildi.[90]

---

[87] Bilgi Teknolojileri ve İletişim Kurumu BTK. Havelsan. Tübitak ULAKBİM. (13.03.2019). *"Teknoloji Sohbetleri 3 / Açık Kaynak Yazılımlar: Tugay ŞAHİN"*, s.24, Ankara. https://www.aciklab.org/s/2501/i/TS3.pdf (15.04.2021)

[88] Bilgi Teknolojileri ve İletişim Kurumu BTK. Havelsan. Tübitak ULAKBİM. (13.03.2019). *"Teknoloji Sohbetleri 3 / Açık Kaynak Yazılımlar: Üstün Murat YILDIZ"*, s.25, Ankara. https://www.aciklab.org/s/2501/i/TS3.pdf (15.04.2021)

[89] Hüseyin Güç, (31.08.2020). *"How A Local Government Migrated To Open Source"*, opensource.com, https://opensource.com/article/20/8/linux-government (15.04.2021)

[90] Ayşe B. Erkeç, (13.08.2018). *"Eyüpsultan, Yerli Yazılımla 2 Milyon Dolar Tasarruf Etti"*, Anadolu Ajansı, İstanbul. https://www.aa.com.tr/tr/sirkethaberleri/belediye/eyupsultan-yerli-yazilimla-2-milyon-dolar-tasarruf-etti/645993 (15.04.2021)

ÜÇÜNCÜ BÖLÜM

RTÜK GÖÇ MODELİ

3.1. MEVCUT ALTYAPI

Radyo ve Televizyon Üst Kurulu bünyesinde, SKAAS sistemi ve bilgi işlem olmak üzere iki ayrı bilgi sistem altyapısı bulunmaktadır. Daha önce de bahsedildiği üzere SKAAS sistemi zaten tamamen özgür ve açık kaynak bir sistemdir. Yani bu bölümde bahsedilecek olan açık kaynak göçünün kapsamı, ÖAK olmayan bilgi işlem altyapısıyla sınırlıdır.

RTÜK bilgi işlem altyapısı, yaklaşık 800 son kullanıcıya hizmet veren, diğer kurumlara göre nispeten küçük yapıda bir bilgi işlem altyapısıdır. RTÜK'ün İstanbul, İzmir ve Diyarbakır dışında çok dağınık ve yaygın bir ağ altyapısı yoktur. Bu kapsamda muhtemel bir göçün, fazla enerji gerektirmeyecek bir geçiş olma olasılığı yüksektir.

Son kullanıcı cihazlarında Windows işletim sisteminin kullanıldığı sistemde ofis uygulamaları için ise Microsoft Office dağıtımı kullanılmaktadır. Sistem altyapısında ise sanallaştırma ürünü olarak VMware, sunucu cihazlarda ise ağırlıkla Windows Server kullanılmaktadır. Ayrıca mail sunucusu

olarak Microsoft Exchange, etki alanı denetleyicisi olarak ise Active Directory kullanılmaktadır. Aynı şekilde sistemde kullanılan web sunucusu, veri tabanı yönetimi yazılımı ve güvenlik ürünleri de yine ÖAK olmayan yazılımlardır.

3.2. RTÜK GÖÇ MODELİ TABLOSU

Radyo ve Televizyon Üst Kurulunun mevcut bilgi işlem altyapısının büyüklüğüne, son kullanıcı sayısına ve mevcut kullanılmakta olan yazılım tiplerine göre hazırlanan göç model önerisi Tablo 3.1 de görülebilir.

Tablo 3.1: RTÜK, Göç Modeli Tablosu

	MODEL	**GEREKÇE**
Göç Süreç Hızı Modeli	Big Bang Göç	1- Geçiş Yapılacak Sistemin Küçük Ölçekli Olması
Göç Kapsam Modeli	Kısmi / Hibrit	1- Sistem Bağımlı Uygulamaların Varlığı
Göç Entegrasyon Araçları	Samba, Citrix, NSS	1- Farklı Platformlar Arası İletişim
Uygulama Göç Modeli	Uygulama Sanallaştırma	1- Hizmete Özel Uygulamaları Sistemde Tutma İhtiyacı 2- Ağ Yükünü Azaltma
Göç Yöntemi	Ayrı (Paralel) Sistem Yöntemi	1- Entegrasyondan Kaçınma 2- Küçük Ölçekli Sistem

3.3. ALTERNATİF SİSTEM, YAZILIM VE UYGULAMA TABLOSU

RTÜK bilgi işlem altyapısında ve son kullanıcı cihazlarında kullanılan işletim sistemi, yazılım ve uygulamaların özgür ve açık kaynak alternatifleri Tablo 3.2'de verilmiştir.

Tablo 3.2: RTÜK, Alternatif ÖAK Yazılımlar

	RTÜK - MEVCUT	**ÖAK ALTERNATİF**
İşletim Sistemi	Microsoft Windows	Pardus
Ofis	Microsoft Office	LibreOffice, Open Office
Sunucular	Windows Server	CentOS, Pardus
Mail Server	Microsoft Exchange	Zimbra
Directory Service	Active Directory	OpenLDAP, FreeIPA
Web Server	Microsoft IIS	Apache
Siber Güvenlik	- -	Ahtapot (Birçok Bileşen İçerir)
Sanallaştırma	VMware	WMware (ÖAK Değil)
Veritabanı	Microsoft SQL Server	PostgreSQL, MySQL (ÖAK Sürümleri)

3.3.1. Kimlik Doğrulama ve DNS Hizmetleri İçin FreeIPA

Halihazırda SKAAS sisteminde de kullanılmakta olan FreeIPA, kimlik ve yetkilendirme verilerini depolayan, merkezi kimlik doğrulama hizmeti sunan, Kerberos ve DNS hizmetleri için etki alanı kontrolü ve yetkilendirme politikaları oluşturan entegre bir merkezi kimlik yönetim ve doğrulama yazılımıdır.

Temel olarak FreeIPA, Linux ve Unix makineleri için bir etki alanı denetleyicisidir. FreeIPA, kontrol edilen sunucuları ve istemci makineleri kullanarak etki alanını tanımlar. Bu, merkezi bir yapı sağlar. Ancak bu temel işlevi dışında sistem için diğer entegre çözümleri de sunar.

FreeIPA, PAM, LDAP, Kerberos, DNS, NTP, NFS ve diğer sertifika hizmetleri dahil olmak üzere tüm ortak ağ hizmetlerinin yönetimi ve yetkilendirmesini yapabilir ve Linux sistemlerin etki alanı denetleyicileri olarak hizmet etmesine olanak tanır.

FreeIPA ile DNS blokları oluşturulabilir, ağdaki Linux istemcileri ve sunucular bu bloklara ayrılarak tek bir merkezi konumdan yönetilebilir. Daha sonra bu client ve sunucular için servis ve hizmet yetkilendirmesi yapılabilir, tek kaynaktan Kerberos kimlik doğrulama ve yetkilendirme ilkeleri tanımlanabilir.

Ayrıca FreeIPA, Microsoft Active Directory ile ortak bir yetkilendirme havuzu oluşturabilir. Böylece RTÜK altyapısında hibrit bir göç modeli benimsenmesine olanak tanınmış olur.

3.3.2. HTTP, FTP, Web Sunucu Hizmetleri İçin Apache

RTÜK bilgi işlem altyapısında; web sunucu, HTTP ve FTP hizmetlerini bir arada sunan Microsoft IIS'nin en önemli ÖAK muadili olan Apache HTTP sunucusu, Linux ve FreeBSD gibi birçok UNIX-like sistemin üzerinde çalıştığı gibi Windows ve MacOS gibi ticari/tescilli sistemlerin üzerinde de çalışabilmektedir. Bu sebeple bu alternatif de hibrit göç modellerine uygundur.

Ağdaki bir IIS sunucu üzerindeki web sitesini, UNIX/Linux üzerinde koşan bir Apache sunucuya taşırken, dikkat edilmesi gereken hususlardan biri dosya sisteminin Apache sunucuda "case-sensitive"[91] olmasıdır. IIS'de böyle bir durum yoktur.

Diğer bir entegrasyon problemi de dosya sisteminde dizin adresleme ayracı ile ilgilidir. IIS hem "/" hem de "\" kabul etmektedir. Bu sebeple bir dosya dizini Apache için farklı konumları temsil edebilir.

Bazı web sitelerinde bulunan dinamik sayfaların bağlı olduğu veri tabanları ve scriptlerin de Apache sunucuya taşınması gerekebilir. Bu veri tabanı için genellikle kullanılan Access veya SQL Server ile scriptlerin taşınma sürecinde bazı program veya "script converter" lardan yardım alınabilir.

Statik bir website, Apache sunucuya taşındıktan ve name server, IP vs. ayarları yapıldıktan sonra FTP gibi hizmetler bu sunucu üzerinden verilebilir.

[91] Case-sensitive: Yazılımın büyük/küçük harf duyarlılığı olması

3.3.3. Mail Sunucu Hizmetleri İçin Zimbra

SKAAS sisteminde de aktif olarak kullanılmakta olan Zimbra Mail Server; istemci posta kutularını, mesaj içeriklerini, kişileri, takvimi, adres defterlerini ve diğer posta dosyalarını yöneten özel bir sunucudur. Zimbra özellikle ÖAK olmayan muadili Microsoft Exchange yerine tercih edilebilecek en önemli alternatiflerden biridir. Zimbra, ilk sürümünün 2005 yılında yayınlanmasından bu yana birçok kez el değiştirmiştir. Bu sebeple farklı lisanslar altında yayınlanmış sürümleri bulunmaktadır. ÖAK olmayan Zimbra Network Edition EULA lisansının yanı sıra, versiyon 8.5'ten sonra ZPL ve GNU GPL gibi ÖAK lisanslar altında yayınlanmış sürümleri de bulunmaktadır.

Hem sunucu hem de istemci bileşenlerini bünyesinde barındırabilen Zimbra birçok Linux dağıtımı üzerinde çalıştırılabilir. Ayrıca sanal bir makine kullanılarak bir Windows sunucusu üzerinde de çalıştırılabilir.

Mesajlaşma için CalDAV, CardDAV ve SMTP; dizin hizmetleri için ise LDAP ve Microsoft Active Directory desteği vardır. Zimbra, MTA işlevi için Postfix'i kullanır. Kötü amaçlı yazılımlardan koruma özellikleri için ClamAV, SpamAssassin ve DSPAM; e-posta imzalama ve şifreleme için ise S/MIME teknolojisini destekler. OS X Sunucu desteği ise ZCS 7.0 sürümüyle kaldırılmıştır.

Zimbra'nın bazı ticari/tescilli sürümlerinde; postalar, kişiler ve takvim öğeleri, Mozilla Thunderbird gibi açık kaynaklı mail istemcileriyle ya da Microsoft Outlook veya Apple Mail gibi ÖAK olmayan istemcilerle ActiveSync protokolü kullanılarak senkronize edebilir.

Microsoft Exchange'in tüm işlevleri, genellikle kolay bir şekilde ÖAK muadilleri ile değiştirilebilir. Asıl sorun, Exchange ve Outlook arasındaki iletişim mekanizması ticari/tescilli olduğundan, bunları Outlook istemcilerine sorunsuz bir şekilde bağlamaya çalışırken ortaya çıkar. Outlook, belirli açık standartlara dayalı hizmetlere erişme yeteneğine sahiptir, ancak bazı durumlarda kullanıcı deneyimi, özel protokolü kullanırken farklılık gösterebilir. Sonuç olarak, istemci cihazlar Outlook'a bağlı kalsalar bile bazı değişiklikleri ortaya çıkacağı göz önüne alındığında, sunucu geçişi yapılırken aynı anda bir ÖAK istemci paketine geçiş yapılıp yapılmayacağına başlangıçta karar vermek doğru bir tercih olacaktır.

Tüm Exchange kullanıcılarının sistemde kayıtlı kullanıcı adları ve şifreleri Active Directory ile yönetilir. Bu nedenle ÖAK tabanlı sunucular kayıt verilerine LDAP aracılığıyla erişebilir, böylece yeni sunucular mevcut Active Directory sunucusunu kullanabilir veya veriler OpenLDAP gibi ÖAK tabanlı bir veri deposuna taşınabilir.

Mail sunucularında yapılacak dönüşümdeki en büyük zorluk kullanıcıların hem kişisel hem de diğer grup üyeleriyle paylaşılan önemli miktarda depolanmış postaya ve eklerine sahip olmasıdır. ÖAK tabanlı posta hizmeti sistemine bir geçiş planlarken, depolanan tüm bu verilerin yerini belirlemek ve geçişten sonra da erişilebilir olacağından emin olmak önemlidir. Exchange, Windows gruplarını dağıtım listeleri olarak kullanabilir, bunlar Windows'un erişim denetimi için kullandığı gruplarla aynıdır. Bu, bir ÖAK ortamında dağıtım listelerini korumanın genel bir yöntemi değildir, ancak istenirse desteklenebilir. Outlook bir posta istemcisi olarak tutuluyorsa, posta kutularına yerel erişim yerine IMAP kullanacak şekilde yeniden yapılandırılması gerekir. Exchange'in dışa aktarma olanağı yoktur, bu nedenle veri geçişi bir istemci bağlantısı üzerinden yapılmalıdır.

Outlook kullanıcıları, ileti gönderip alırken otomatik olarak kişisel bir adres defteri oluşturulur. Ayrıca, Exchange sunucusu kullanılıyorsa, istemcilerin bir veya daha fazla paylaşılan adres defterine erişimleri de vardır. Bu adres defterlerinin içeriği, Zimbra tarafından okunabilir bir formata dönüştürülmelidir. Kişisel adres defterleri, birçok posta istemcisi tarafından anlaşılan ve gerektiğinde diğer biçimlere dönüştürülmek üzere komut dosyaları tarafından ayrıştırılabilen vCard biçiminde de dışa aktarılabilir. Benzer şekilde, paylaşılan adres defterleri dışa aktarılabilir ve ardından bir LDAP deposuna yüklenebilir.

3.3.4. Veritabanı Yönetimi İçin PostgreSQL

SKAAS sisteminde de aktif olarak kullanılmakta olan PostgreSQL, SQL uyumlu ilişkisel bir veri tabanı yönetim sistemidir. PostgreSQL, orijinal Berkeley kodunun açık kaynaklı bir sürümü üzerine geliştirilmiş olup SQL standardının büyük bir bölümünü destekler ve birçok modern özellik sunar:[92]

- Karmaşık sorgular
- Yabancı anahtarlar
- Tetikleyiciler
- Güncellenebilir ön izlemeler
- İşlem bütünlüğü
- Çok sürümlü eşzamanlılık denetimi vb.

PostgreSQL, tek bir makineli sistemden veri merkezleri gibi çok büyük boyutlu sistemlere kadar web hizmetleri dâhil çeşitli iş yüklerini işlemek için tasarlanmıştır. MacOS sunucuları için varsayılan veri tabanı yönetim sistemidir. Ayrıca Windows, Linux, FreeBSD ve OpenBSD desteği vardır.

[92] PostgreSQL.org. *"What Is PostgreSQL?"*, https://www.postgresql.org/docs/current/intro-whatis.html (21.08.2021)

Büyük veri tabanı sistemleri söz konusu olduğunda, veri tabanını mevcut platformda çalışır durumda bırakmak ve sadece istemciyi ÖAK platforma taşımak en iyisi seçenek olabilir. Bu, özellikle sistem yöneticilerinin mevcut veri tabanı için önemli miktarda “know-how”’ sahibi olması durumunda göz önünde bulundurulmalıdır. Ağ üzerinden veri tabanlarına bağlanmanın standart yolları vardır. Bu nedenle platform seçimi, veri tabanı ve istemci uygulamaları için farklı olabilir. Ayrıca, Microsoft'a ait olmayan veri tabanlarının çoğu GNU/Linux ve Unix platformlarında mevcuttur: Bu sebeple işletim sistemini tamamen yeni bir veri tabanı öğrenmek zorunda kalmadan değiştirmek mümkündür.

Verileri MS SQL Server platformundan ÖAK platforma geçirmenin birkaç yolu vardır. Basit veri kümeleri için belki de en kolay yol, tabloları MS SQL’den CSV dosyaları olarak dışa aktarmak ve ardından bunları yeni sunucuya almaktır. Bu yöntem, tabloların yeni sunucuda önce elle oluşturulmasını gerektirir, ancak herhangi bir özel yazılıma ihtiyaç duyulmaz.

İkinci yöntemde ise; tabloları başka bir veri tabanı yönetim sisteminde yeniden oluşturmak için gerekli verileri eksiksiz bir şekilde MS SQL Server veri tabanından dışarı aktarabilen bazı script ve programlar kullanılır. Bunlardan bazıları yeni platforma kopyalanacak dosyalar üretirken, diğerleri doğrudan ağ üzerinden bağlanır ve değişiklikleri yeni veri tabanına işler.

Basit veri tabanları için verileri ortak bir formatta dışa aktarmak ve ardından yeni veri tabanına aktarmak genellikle yeterlidir. Saklı yordamlar ve tetikleyiciler dâhil daha karmaşık veri tabanları daha fazla çaba gerektirecektir. Bu durumlarda geçiş sürecine yardımcı olacak çeşitli araçları ve scriptleri kullanmak faydalı olacaktır. Bunlardan bazıları ÖAK bazıları ise ticari/tescilli ürünlerdir. Bunlardan bazıları: PGAdmin, SQLPorter, SQLWays vb.

PGAdmin aslında PostgreSQL veri tabanlarını yönetmek için kullanılan ücretsiz bir “tool” yazılımdır. Ancak diğer veri tabanı platformlarından veri geçişini işleyen eklentileri de bünyesinde sunmaktadır. Özellikle MS SQL Server – PostgreSQL dönüşümlerinde kullanılabilecek oldukça yararlı bir araçtır.

Veri tabanında ÖAK dönüşümünde birçok sorun çıkabilir. Verileri taşımak işin kolay kısmı olabilir, ancak verilere ağ üzerinden basit SQL stili tablolar olarak erişilecekse, yapılacak fazla bir şey olmayabilir. Veri tabanı ÖAK dönüşümünde sorunların büyük olasılıkla mevcut veri tabanını çevreleyen yardımcı programlardan ve komut dosyası dillerinden gelmesi muhtemeldir. SQL’in kendisi standartlaştırılmıştır, ancak neredeyse tüm veri tabanı yönetim yazılımı geliştiricileri onu geliştirmiş ve kullanıcıları standart olmayan uzantıları kullanmaya teşvik etmiştir.

Ayrıca, SQL'de belirli bir arama sonucu elde etmenin genellikle birkaç farklı yolu vardır ve hangisinin en verimli olduğu seçimi bir veri tabanından diğerine değişebilir. Birçok veri tabanı uygulaması, uygulama oluşturucular veya form oluşturucular ile oluşturulmuştur. Bunlar, lisanslandığı veri tabanları dışındaki diğer veri tabanı yönetim yazılımlarıyla çalışmayabilir.

3.4. DİĞER SUNUCU HİZMETLERİ İÇİN CENTOS VEYA PARDUS

Mevcut sistemde; Proxy, VPN, FTP, DHCP vb. gibi hizmetleri veren sunucular Windows Server ile çalışmaktadır. Bu sunucularda yapılacak bir göç senaryosu için en önemli ÖAK alternatif CentOS ya da yerli işletim sistemi Pardus 'tur.

CentOS, bir Linux dağıtım olup sekizinci sürümünden itibaren x86, x64 ve ARM64 mimarilerini destekler. CentOS, Windows Server sisteme göre daha hızlı, siber tehditlere daha dirençli ve donanım kaynaklarını daha verimli kullanan bir sistemdir. Ve tabi ki ÖAK ve ücretsiz bir yazılımdır.

Pardus 'un ise, sunucu makinelere yönelik Pardus Sunucu sürümü yayınlanmış olup bu sistem de donanım kaynaklarını verimli kullanan kararlı ve hızlı bir sunucu işletim sistemidir.

Sunucu tarafta yapılacak ÖAK dönüşümde, sunucular disk imaj dosyaları kullanılarak tek tek CentOS ya da Pardus'a geçirilebilir. RTÜK altyapısında çalışan sunucu sayısı düşünüldüğünde bu tarz bir geçiş yönteminin belirlenmesi yanlış olmayacaktır. Ancak böyle bir geçiş yöntemi çok fazla zaman ve enerji gerektirecektir.

Aynı işi yapan ya da daha limitli görevleri gerçekleştiren sunucu veya sunucu bloklarında gerçekleştirilecek dönüşümde, bu sunucuların toplu halde ve tek bir kaynaktan yüklenmesi daha doğru bir tercih olacaktır. Ayrıca network üzerinden işletim sistemi imajlarının klonlanması, disklerin veya disk bölümlerinin yedeklenmesi veya geri yüklenmesi için de Clonezilla gibi ÖAK bir yazılım sistemde kullanılabilir.

Toplu dönüşüm için öncelikle ağ üzerinde kaynak olarak bir FTP sunucusu (vsftpd) kurulmalıdır. Bu, tek bir kaynak sunucu noktasından birden fazla makinaya CentOS kurulabilmesine olanak tanır. Bunun çalışması için, ağınıza bağlı bir makinede CentOS kurulumunuzun olması gerekir. Ancak, CentOS DVD ISO görüntüsünü bağlayacağınız FTP, HTTP veya NFS sunucusunun kurulu olduğu diğer CentOS sürümlerini veya hatta diğer Linux dağıtımlarını da kullanabilirsiniz.

FTP ağ kaynaklarını kullanarak birden çok CentOS sunucusu kurmak için şu adımlar takip edilebilir:[93]

İlk adım, aşağıdaki komut kullanılarak CentOS sunucunuza vsftpd sunucusunu kurmak olacaktır.

- *yum install vsftpd*

Vsftpd paketi sisteminize yüklendikten sonra, hizmetin başlatın, etkinleştirin.

- *systemctl start vsftpd*
- *systemctl enable vsftpd*
- *systemctl status vsftpd*

Daha sonra uzak bir konumdan ağ kaynaklarınıza erişmeniz gerekeceğinden IP adresinizi kaydedin.

- *ifconfig*

[93] TecMint.com, *"How to Install Multiple CentOS/RHEL Servers Using FTP Network Sources"*, https://www.tecmint.com/install-multiple-centos-servers-using-ftp/ (28.08.2020)

Vsftpd sunucusunu dış bağlantılara açık hale getirmek için, aşağıdaki komutu kullanarak 21 nolu portu açmak üzere sisteminize bir güvenlik duvarı kuralı ekleyin ve kuralı uygulamak için güvenlik duvarını yeniden başlatın.

- *firewall-cmd --add-service=ftp --permanent*
- *systemctl restart firewalld*

CentOS disk görüntüsünü sunucu makinenin (fiziksel ya da sanal makine olabilir) DVD-ROM/USB sürücüsüne yerleştirin ve vsftpd sunucu yoluna salt okunur özniteliklere sahip bir döngü olarak “mount” edin.

- *mount -o loop,ro /dev/sr0 /var/ftp/pub/*
- *mount -o loop,ro path-to-isofile /var/ftp/pub/*

Şimdiye kadarki sonucu görmek için uzak bir konumdan bir tarayıcı açın ve FTP protokolünü kullanarak ftp://system_IP/pub/ adresine gidin. Eğer işlemler doğru yapıldıysa imaj dosyaları ağaç şeklinde görüntülenecektir.

Bu aşamalar tamamlandıktan sonra, yapılandırılmış sunucuyu FTP kaynağı olarak gören sunucu makinelere CentOS’un kurulması ile devam edilir. CentOS kurulumunun gerçekleştirileceği sistemde önyüklenebilir ISO görüntüsü DVD-ROM/USB sürücüsüne yerleştirilir. Önyüklenebilir USB sürücü oluşturmak için Unetbootin Bootable veya Rufus araçlarından biri kullanılabilir.

Ağ kartı aktif ve çalışır hale geldikten sonra, kurulum özeti menüsünden kurulum kaynağı seçilir. FTP protokolünü kullanan ağ kurulum kaynakları seçilir ve FTP sunucusu IP adresi ve yolu ile daha önce yapılandırılan kaynaklar eklenir.

3.5. YAZILIM EŞDEĞERLİĞİ

Windows sistemi üzerinde çalışan ve yaygın kullanılan uygulamaların Linux dağıtımındaki açık kaynak alternatifleri Şekil 3.1'de gösterilmiştir.

Tablo 3.3: Windows – Linux Eşdeğer Yazılımlar

Windows Tabanlı	Linux Eşdeğeri
Internet Explorer / Microsoft Edge	Firefox / Opera
Microsoft Office	LibreOffice / Apache OpenOffice
Adobe Photoshop	Gimp
Paint	Pinta / Gimp
Media Player	VLC
Movie Maker	OpenShot
Adobe PDF Reader	Adobe PDF Reader (Linux)
Outlook	Thunderbird / Evolution
MS IIS (FTP)	FileZilla (FTP)
AutoCAD, Solidworks, CATIA	FreeCAD
Norton Ghost, Symantec Ghost	CloneZilla
WinRAR	7-Zip
Microsoft RDP	XRDP / Remmina
Notepad	Gedit / Vim

SONUÇ VE DEĞERLENDİRME

Bu çalışmayla, ÖAK yazılımların kullanımı ile bu sistemlere geçiş süreci konularında, özel ve kamu kesiminde gerekli bilinç ve motivasyonun oluşmasına katkıda bulunulması hedeflenmiştir.

Özgür ve açık kaynak yazılımlara geçiş süreçlerinin finansal, teknik ve risk analizi yapılarak muhtemel göç operasyonları süresince ortaya çıkabilecek olumsuz durumların yönetilebilmesine katkıda bulunulacağı düşünülmektedir. Yapılan analizler sonucu elde edilen verilerle, fayda / risk analizi de yapılarak, muhtemel bir RTÜK Açık Kaynak Göç'üne katkıda bulunulacağı değerlendirilmektedir.

Açık kaynaklı sistemlere yapılacak bir göç, kapsamlı bir maliyet analizi ve planlama gerektiren önemli bir süreçtir. İyi planlanmamış, kısmen ya da tamamen başarısız olmuş göç operasyonları; ek maliyet yükü, teknolojik beklentilerin karşılanamaması, güvenlik açıklarının oluşması ve genel sistem işleyişinin aksaması gibi riskler taşır. Muhtemel göç süreçlerini planlayan kamu kurumlarının, potansiyel risk ve maliyet kestiriminde bu çalışmadan yararlanabilecekleri düşünülmektedir.

Çalışmadan çıkarılacak sonuçlardan biri, açık kaynak göçünün her zaman fayda sağlayacağının garantisinin olmamasıdır. Geçişten maksimum fayda sağlayabilmek için detaylı planlamaların yapılması, mevcut sistemin

imkânlarının iyi araştırılması, sistem ve son kullanıcı beklentilerinin iyi belirlenmesi gerekir.

Bir kurum ya da kuruluşun, altyapısında gerçekleştirilecek olan göçün kapsamı isteğe göre ölçeklendirilebilir. Sistemin hangi bölümlerinde geçiş işleminin gerçekleştirileceği, o sistemin yapısına, mimarisine ve ihtiyaçlarına göre belirlenir. Ayrıca son kullanıcıların beklenti ve alışkanlıkları da bu konuda belirleyici olabilir. Sistem altyapısında halihazırda kullanılmakta olan diğer tescilli/ticari yazılımların ÖAK yazılımlarla göstereceği uyumluluk davranışları da göçün kapsam modelini etkileyen önemli hususlardandır. Tüm bu kriterler göz önünde bulundurularak, göç hızı modeli, göç kapsam modeli ve göç geçiş yönteminin detaylı çalışması yapılarak sağlıklı bir göç süreci izlenmelidir.

Göç sürecinde ve devamında, geçiş ve entegrasyon sürecinin kolaylaştırılması ve muhtemel sorunların azaltılması için bazı açık kaynak yazılım entegrasyon araçları kullanılabilir. Özellikle hibrit modellerde, ÖAK yazılımlar ile özel mülk yazılımlar arasında gerekli iletişimi ve uyumu sağlamak için bu çözümlerin kullanılması yararlı olabilir.

Muhtemel bir göçün başarısında birçok etken rol oynar. Geçişten sonra son kullanıcıların ve sistem yöneticilerinin açık kaynak sistemi ne ölçüde benimsediği bu etkenlerden sadece biridir. Bu noktada başarıyı ölçecek olan kıstaslarda çok önemlidir. Göçü gerçekleştiren kuruluş, başarı kriteri olarak geçişin ekonomik getirisini, sağladığı sistem esnekliğini ya da güvenlik ve kararlılık gibi parametreleri göz önünde bulundurabilir. Eğer beklentiler arasında ekonomik fayda daha ön plana çıkıyorsa, doğru bir değerlendirme için hemen hükme varmamak doğru bir yaklaşım olacaktır. Çünkü ÖAK yazılımlı bir sisteme geçişte sağlanacak ekonomik fayda sistemin ilk yılında çok belirgin olmayabilir. Genellikle ÖAK bir sistemin işletme ömrü

lisanslı/tescilli yazılımların kullanıldığı bir sisteme göre çok daha uzun olup bakım ve destek ihtiyacı ise daha azdır. Ayrıca genellikle basit yapıları sayesinde uzun yıllar eskiyen donanımlarda bile verimli bir şekilde çalışabilirler. Bu sebeple bir açık kaynak göçünün sağlayabileceği ekonomik faydayı hesaplarken uzun dönem projeksiyonların dikkate alınması daha doğru bir yaklaşım olacaktır.

ÖAK yazılıma göç etmenin getirdiği avantajların yanında, bir açık kaynak göçü her duruma ve her sisteme her zaman uygun olmayabilir. Bunun için açık kaynak göçü planlayan kurum ya da kuruluş ÖAKY sistemden beklentilerini ve gelecek hedeflerini iyi belirlemelidir. Aksi takdirde açık kaynak göçünün taşıdığı risklerden ve dezavantajlardan daha fazla etkilenebilir. Örneğin, bir kuruluş, sisteminde çok fazla işletim sistemi bağımlılığı olan yazılım barındırıyorsa ya da son kullanıcı tarafında geçiş için çok fazla direnç varsa, açık kaynak göçü o kuruluş için doğru çözüm olmayabilir.

ÖAK yazılımlara göçün sağlayacağı geniş avantajlar sebebiyle bazı dezavantajlar göz ardı edilebilir. Hatta geçişi planlayan kuruluş ihtiyaçlarını, beklentilerini ve hedeflerini tam olarak belirlediği takdirde bu dezavantajlardan etkilenmeyebilir.

Kurum ve kuruluşların ÖAK sistemlere yapacağı göç operasyonları, Türkiye'deki yerel ÖAK yazılım kültürünün gelişmesi için gerekli motivasyonun oluşmasına katkı sunacaktır. Böylece Türkiye, kendi kaynakları ve kapasitesiyle geliştirdiği bir yazılım ekosistemine sahip olabilir. Bu ekosistem içinden çıkacak daha güvenli, millî, bağımsız ve düşük maliyetli yazılım ürünleri ile de ülkenin dijital dönüşümüne ve cari açığına önemli katkıda bulunulabilir.

Açık kaynak sistemlere göç konusu, RTÜK bağlamında değerlendirildiğinde;

- Bu geçişin RTÜK'e en önemli getirisi hiç şüphesiz ekonomik fayda olacaktır. Özellikle ilk edinme lisans maliyetleri yanında lisans yenileme maliyetleri de RTÜK bilgi işlem giderlerinde en önemli kalemlerden biridir. Muhtemel bir açık kaynak göçü ile, özellikle orta ve uzun vadede büyük tasarruf sağlanabilir. Buradan elde edilecek tasarruf ile donanım yenileme gibi diğer projelere kaynak aktarılabilir.

- Özellikle ticari/tescilli lisans sağlayıcı firmaların müşterilerini kendi yazılım ekosisteminde tutma eğilimleri de kurum ve kuruluşları uzun yıllar bu birkaç firmanın ürünlerini kullanmaya mahkûm bırakabiliyor. Bazen lisans sağlayıcının dayattığı donanım-yazılım bağımlılığı, yazılım entegrasyon kısıtları gibi sebeplerle kurum ve kuruluşların ticari/tescilli yazılım ekosisteminden çıkmaları her geçen yıl daha da zorlaşır. ÖAK sistemlerde ise böyle bir durum söz konusu değildir. İşte bu noktada açık kaynak göçü ile RTÜK'e kazandırılacak satıcı bağımsızlığı kozu önemli faydalardan biri olacaktır.

- ÖAK yazılımlar, uzun yıllar hiçbir güncelleme almadan ve bakım destek ihtiyacı olmaksızın sistem ve çalışma kararlılığını sağlayabilirler. Özellikle RTÜK bilgi işlem altyapısında bulunan ticari/tescilli yazılımların ÖAK muadilleri özel mülk rakiplerine göre çok daha stabil çalışırlar. Ayrıca ticari/tescilli rakiplerine göre çok daha az donanım güncellemesi gerektirirler. Bu da donanım güncelleme maliyetlerinden de tasarruf edebilme fırsatı sunar.

- Bakım destek hizmeti maliyetleri de RTÜK için önemli gider kalemlerindendir. ÖAK sistemlerde de bu ihtiyaç vardır. Ancak tescilli yazılım ekosisteminde olduğu gibi yüksek maliyetli, sık ve bağlayıcı değildir. Hatta bu ihtiyaç ücretsiz kaynaklardan da karşılanabilir.

- RTÜK için sağlanacak tüm bu faydaların yanında dikkat edilmesi gereken hususlar da vardır. Örneğin, RTÜK bilgi işlem altyapısında bulunan hizmete özel geliştirilmiş sistem bağımlı uygulamaların varlığı gibi. Bu sorun ise, uygulama sanallaştırma veya uzak sunucu bağlantısı gibi yöntemlerle aşılabilir.

- Açık kaynak göçünün başarısı ile ilgili önemli parametrelerden biri de son kullanıcıların client cihazlara kurulacak olan ÖAK yazılım ve işletim sistemlerini ne düzeyde benimseyecekleridir. Ancak RTÜK bu konuda avantajlı konumdadır. Çünkü 2009'dan beri hizmette olan ve son kullanıcı cihazlarında Pardus ve Linux Mint kullanılan SKAAS sistemi, kurum personelinin büyük bir bölümü tarafından yaklaşık 12 yıldır yoğun olarak kullanılmaktadır. Bu bağlamda ÖAK yazılımların benimsenmesi konusunda dirençle karşılaşma olasılığı düşüktür. Ancak geçişin kademelendirilmesi, özellikle ilk aşamada aktif olarak SKAAS sistemine aşina personellerin cihazlarının ÖAK dönüşümlerinin yapılması veya göç için pilot dairelerin belirlenmesi gibi yöntemlerle direnç oranı test edilip riskler minimize edilebilir.

- Son kullanıcı tarafında yapılacak ÖAK dönüşümünün aksine sistem altyapısında yapılacak dönüşüm için ise bir direnç söz konusu değildir. Gerekli teknik kapasiteye sahip personelin varlığı ile söz konusu personele verilecek eğitimler ile hem geçiş süreci hem de devamında ÖAK sistemin işleyişi garanti altına alınabilir.

- Ayrıca tamamen ÖAK bir sistem olan SKAAS sisteminin kuruluşunda görev alan personel vasıtasıyla ÖAK sistemler konusunda oluşturulmuş olan kurumsal hafıza ve tecrübe, muhtemel bir göçün daha kolay ve daha organize gerçekleştirilmesine katkı sağlayacaktır.

- Bu çalışmanın son kısmında bahsedilen RTÜK göç modelinde; göç modeli tablosu, alternatif sistem tablosu ve son kullanıcı eşdeğer yazılım tablosu önerilmiş; kimlik doğrulama ve DNS hizmetleri, HTTP/FTP/Web sunucu hizmetleri, mail sunucu hizmetleri, veri tabanı yönetimi hizmeti ile diğer sunucu hizmetleri için dönüşüm projeksiyonu hazırlanmıştır. Ancak burada önerilen model ve projeksiyonların uygulanabilirliğinin, çalışma içeriğinin güncelliğine bağlı olduğu unutulmamalıdır.

KAYNAKÇA

ITU, (2020). *“Measuring Digital Development Facts and Figures”*, s.7, Switzerland.
https://www.itu.int/en/ITU-D/Statistics/Documents/facts/FactsFigures2020.pdf (09.01.2021)

Ritchie, D.M. ve Thompson, K. (1978). *“The UNIX Time-Sharing System”*, The Bell System Technical Journal, AT&T Bell Laboratories, ABD.

Ritchie, D.M. (1984). *“The Evolution of the Unix Time-sharing System”*, The Bell System Technical Journal, AT&T Bell Laboratories, ABD.

Johnson, S.C., Ritchie, D.M. (1978). *“Portability of C Programs and the UNIX System”*, The Bell System Technical Journal, AT&T Bell Laboratories, ABD.

Raymond, Eric S. (2003). *"The Art of Unix Programming: Exodus: 1971–1980"*, s.56, ABD.

Raymond, Eric S. (2003). *"The Art of Unix Programming: TCP/IP and the Unix Wars: 1980-1990"*, s.58, ABD.

DiBona, C. Ockman, S. Stone, M. (1999). *"Open Sources: Voices from the Open Source Revolution: A Brief History of Hackerdom By Eric S. Raymond"*, s.19, O'Reilly, ABD

FreeBSD, The FreeBSD Foundation. *"FreeBSD Documents, Chapter 1: Introduction Part I. Getting Started"*, https://docs.freebsd.org/en/books/handbook/introduction/ (16.01.2021)

DiBona, C. Ockman, S. Stone, M. (1999). *"Open Sources: Voices from the Open Source Revolution:Twenty Years of Berkeley Unix, From AT&T-Owned to Freely Redistributable By Marshall Kirk McKusick"*, s.26 O'Reilly, ABD

Nokia Bell Labs. (1994). *"UNIX System Laboratories Inc. vs BSDi Settlement Agreement"*, https://www.bell-labs.com/usr/dmr/www/bsdi/USLsettlement.pdf (20.01.2021)

DiBona, C. Ockman, S. Stone, M. (1999). *"Open Sources: Voices from the Open Source Revolution:The GNU Operating System and the Free Software Movement By Richard Stallman"*, s.32, O'Reilly, ABD

Stallman, Richard M. (2015). *"Free Software, Free Society: Selected Essays of Richard M. Stallman: The GNU Project"*, s.9, ABD

DiBona, C. Ockman, S. Stone, M. (1999). *"Open Sources: Voices from the Open Source Revolution:The GNU Operating System and the Free Software Movement By Richard Stallman"*, s.32, O'Reilly, ABD

Stallman, Richard M. (2015). *“Free Software, Free Society: Selected Essays of Richard M. Stallman: Selling Free Software”*, s43., ABD

Stallman, Richard M. (2015). *“Free Software, Free Society: Selected Essays of Richard M. Stallman: What Is Free Software?”*, s.3, ABD.

Stallman, Richard M. (2015). *“Free Software, Free Society: Selected Essays of Richard M. Stallman: GNU Software and the GNU System”*, s.12, ABD.

Torvalds, L. (2007). *"Release Notes for Linux v0.12"*, kernel.org, https://mirrors.edge.kernel.org/pub/linux/kernel/Historic/old-versions/RELNOTES-0.12 (29.01.2021)

Larabel M. (2015). *“Linux Kernel At 19.5 Million Lines Of Code, Continues Rising”*, https://www.phoronix.com/ (29.01.2021)

Kelty, Christopher M. (2008). *"The Cultural Significance of Free Software: Forking Free Software, 1997–2000"*, s.108, Duke University Press, ABD.

Kelty, Christopher M. (2008). *"The Cultural Significance of Free Software: A Movement?",* s.113, Duke University Press, ABD.

Perens B. (1997). *"Debian's "Social Contract" with the Free Software Community"*, Debian-Announce Mailing List, https://lists.debian.org/debian-announce/1997/msg00017.html (01.02.2021)

Free Software Foundation Inc. *“Licences”*, GNU Projesi, https://www.gnu.org/licenses/licenses.en.html (02.02.2021)

Free Software Foundation Inc. *“What is Copyleft?”*, GNU Projesi, https://www.gnu.org/licenses/copyleft.en.html (08.02.2021)

Newman, J. (2011). *"Copyright and Open Access at the Bedside"*, New England Journal of Medicine, ABD.

Mozilla Foundation. *“Mozilla Public License, Version 2.0”*, https://www.mozilla.org/en-US/MPL/2.0/ (13.02.2021)

O’Hara Keith J. ve Kay, Jennifer S. (2003). *“Open Source Software and Computer Science Education”*, Rowan University, ABD.

European Commission, (2020). *“Communication To The Commission: Introduction”*, s.3, Open Source Software Strategy 2020 – 2023, Brussels.

European Commission, (2020). *“Communication To The Commission: Progress Towards Digital Autonomy”*, s.5-6, Open Source Software Strategy 2020 – 2023, Brussels.

Wong, K. ve Sayo P. (2004). *“Free/Open Source Software: A General Introduction: FOSS Success Stories”*, s.19-20, United Nations Development Programme’s Asia-Pacific Development Information Programme (UNDP-APDIP), Malaysia.

Lewis, James A. (2008). *"Government Open Source Policies: National Open Source Policies"*, Center for Strategic and International Studies CSIS, ABD.

Kingstone. S. (2005). *"Brazil Adopts Open-Source Software"*, BBC News, Sao Paulo, Brezilya. http://news.bbc.co.uk/2/hi/business/4602325.stm (17.02.2021)

Wong, K. ve Sayo P. (2004). *“Free/Open Source Software: A General Introduction: FOSS Success Stories: Japan”*, s.24, United Nations Development Programme’s Asia-Pacific Development Information Programme (UNDP-APDIP), Malaysia.

Optimus Information, (2015). *"Open-Source vs. Proprietary Software Pros and Cons: Free to Try Before You Buy"*, s.3, Canada.

Wong, K. ve Sayo P. (2004). *“Free/Open Source Software: A General Introduction: Security”*, s.13, United Nations Development Programme’s Asia-Pacific Development Information Programme (UNDP-APDIP), Malaysia.

Wheeler, David A. (2015). *"Why Open Source Software / Free Software (OSS/FS, FLOSS, or FOSS)? Look at the Numbers!"*, https://dwheeler.com/oss_fs_why.html (22.02.2021)

Optimus Information, (2015). *"Open-Source vs. Proprietary Software Pros and Cons: Fewer Bugs and Faster Fixes"*, s.4, Canada.

Optimus Information, (2015). *"Open-Source vs. Proprietary Software Pros and Cons: Open Standarts-Fewer Bugs and Faster Fixes"*, s.4, Canada.

Wong, K. ve Sayo P. (2004). *"Free/Open Source Software: A General Introduction: Developing Local Software Capacity"*, s.15-16, United Nations Development Programme's Asia-Pacific Development Information Programme (UNDP-APDIP), Malaysia.

Wong, K. ve Sayo P. (2004). *"Free/Open Source Software: A General Introduction: Interoperability with proprietary systems"*, s.17, United Nations Development Programme's Asia-Pacific Development Information Programme (UNDP-APDIP), Malaysia.

OpenLogic Inc. ve Navica (2006). *"Open Source Return On Investment: Achieving the Financial Promise of Open Source Software"*, ABD.

Gartner, J. (2019). *"What Is Open Source Software And How Much Can It Save Us?"*, Gartner Technology LLC., https://gartnertechnology.com/blog/2019/01/26/open-source-software-much-can-save-us/ (25.03.2021)

Shankland, S. (2002). *"How Linux Saved Amazon Millions"*, CNET, https://www.cnet.com/news/how-linux-saved-amazon-millions/ (25.03.2021)

Sanders, J. (2019). *"South Korean Government Planning Linux Migration as Windows 7 Support Ends"*, TechRepublic, https://www.techrepublic.com/article/south-korean-government-planning-linux-migration-as-windows-7-support-ends/ (25.03.2021)

Rosenberg, D. (2008). *"Study Finds Free Open Source Software Is Costing Vendors $60 Billion"*, CNET, https://www.cnet.com/news/study-finds-free-open-source-software-is-costing-vendors-60-billion/ (25.03.2021)

Treasury Inspector General For Tax Administration. (2018). *“The Solaris to Linux Migration Project Was Delayed and Needs Improved Governance”*, ABD.

Bott, E. (2014). *"After a 10 Year Linux Migration, Munich Considers Switching Back to Windows and Office"*, ZDNet, https://www.zdnet.com/article/after-a-10-year-linux-migration-munich-considers-switching-back-to-windows-and-office/ (27.03.2021)

Heath, N. (2017). *"End of an open source era: Linux pioneer Munich confirms switch to Windows 10"*, TechRepublic, https://www.techrepublic.com/article/end-of-an-open-source-era-linux-pioneer-munich-confirms-switch-to-windows-10/ (27.03.2021)

Schaer, C. (2020). *"Linux not Windows: Why Munich is shifting back from Microsoft to open source – again"*, ZDNet, https://www.zdnet.com/article/linux-not-windows-why-munich-is-shifting-back-from-microsoft-to-open-source-again/ (28.03.2021)

Cole, B. (2016). *"Migration"*, TechTarget, https://searchcio.techtarget.com/definition/migration (29.03.2021)

Netproject Ltd. (2003). *"The IDA Open Source Migration Guidelines: Management Guidelines"*, s.19-20, İngiltere.

Netproject Ltd. (2003). *"The IDA Open Source Migration Guidelines: Management Guidelines: Phased Transition In Groups"*, s.20, İngiltere.

Samba Org. *"Samba 4.12 Features Added / Changed"*, SambaWiki https://wiki.samba.org/index.php/Samba_4.12_Features_added/changed (08.04.2021)

OpenLDAP Foundation. (2003). *"The OpenLDAP Project Overview"*, ABD. https://www.openldap.org/ (03.04.2021)

Samba Org. *“The SMBFS Filesystem”*, https://www.samba.org/samba/smbfs/ (05.04.2021)

Netproject Ltd. (2003). *"The IDA Open Source Migration Guidelines: Tools: Windbind"*, s.71, İngiltere.

Parlakyiğit, Mehmet. *"Microsoft App-V Uygulama Sanallaştırmaya Giriş"*, s.2.

Credit Suisse Securities LLC. (2007). *"Desktop Virtualization Comes Of Age"*, Equity Research, ABD.

Gökkoyun, S. Ceren. (14.04.2020). *"Bakan Varank: Türkiye Açık Kaynak Platformu çalışmalarına kalkınma ajanslarımız 30 milyon lira katkı sunacak"*, Anadolu Ajansı, https://www.aa.com.tr/tr/ekonomi/bakan-varank-turkiye-acik-kaynak-platformu-calismalarina-kalkinma-ajanslarimiz-30-milyon-lira-katki-sunacak/1804321 (14.04.2021)

Eser, İbrahim. (2011). *“Kamuda Özgür Yazılım Kullanımı: Dünyada ve Türkiye ' deki Örnekleri Üzerinden Bir Değerlendirme”*, s.108, RTÜK, Ankara.

NTV. (2009). *"MSB, Pardus ile 2 milyon dolar tasarruf etti"*, https://www.ntv.com.tr/turkiye/msb-pardus-ile-2-milyon-dolar-tasarruf-etti,rE0SKD34VkGRgE_uv3votw (14.04.2021)

Yıldırım, G. (21.11.2020). *"Milli Savunma Bakanlığı bilgisayarları yerli işletim sistemine geçiyor"*, Anadolu Ajansı, https://www.aa.com.tr/tr/turkiye/milli-savunma-bakanligi-bilgisayarlari-yerli-isletim-sistemine-geciyor/2051148# (14.04.2021)

Gemici, O. Onur. (20.05.2020). *"AFAD Başkanı Güllüoğlu: 28 İl Afet ve Acil Durum Müdürlüğünde Milli İşletim Sistemi Pardus 'a Geçildi"*, Anadolu Ajansı, https://www.aa.com.tr/tr/turkiye/afad-baskani-gulluoglu-28-il-afet-ve-acil-durum-mudurlugunde-milli-isletim-sistemi-pardusa-gecildi/1847557 (15.04.2021)

Bilgi Teknolojileri ve İletişim Kurumu BTK. Havelsan. Tübitak ULAKBİM. (13.03.2019). *"Teknoloji Sohbetleri 3 / Açık Kaynak Yazılımlar: Tugay ŞAHİN"*, s.24, Ankara. https://www.aciklab.org/s/2501/i/TS3.pdf (15.04.2021)

Bilgi Teknolojileri ve İletişim Kurumu BTK. Havelsan. Tübitak ULAKBİM. (13.03.2019). *"Teknoloji Sohbetleri 3 / Açık Kaynak Yazılımlar: Üstün Murat YILDIZ"*, s.25, Ankara. https://www.aciklab.org/s/2501/i/TS3.pdf (15.04.2021)

Güç, Hüseyin. (31.08.2020). *"How A Local Government Migrated To Open Source"*, opensource.com, https://opensource.com/article/20/8/linux-government (15.04.2021)

Erkeç, A. Büşra. (13.08.2018). *"Eyüpsultan, Yerli Yazılımla 2 Milyon Dolar Tasarruf Etti"*, Anadolu Ajansı, İstanbul. https://www.aa.com.tr/tr/sirkethaberleri/belediye/eyupsultan-yerli-yazilimla-2-milyon-dolar-tasarruf-etti/645993 (15.04.2021)

Vohra, T. (2004). *"Migration from Windows to Linux for a Small Engineering Firm A&G Associates"*, Rochester Institute of Technology, s.24, ABD.

PostgreSQL.org. *"What Is PostgreSQL?"*, https://www.postgresql.org/docs/current/intro-whatis.html (21.08.2021)

TecMint.com, *“How to Install Multiple CentOS/RHEL Servers Using FTP Network Sources”*, https://www.tecmint.com/install-multiple-centos-servers-using-ftp/ (28.08.2020)

ÖZET

Ücretsiz veya düşük lisans maliyetleri, yüksek kararlılık, artırılmış güvenlik, esneklik ve satıcı bağımsızlığı gibi, özgür ve açık kaynak sistemlere göç ederek elde edilebilecek birçok fayda vardır. Bu çalışma, açık kaynak sistemlere geçiş yapmayı planlayan kuruluşlar için yararlı olabilir. Zira bu çalışma, kuruluşların Özgür ve Açık Kaynak Yazılım platformlarına geçerken kullanabilecekleri geçiş modelleri ve yollarını açıklamayı amaçlamaktadır.

İlk bölümde, Unix, BSD, Linux ve GNU gibi işletim sistemlerinin erken dönem gelişiminin kısa bir geçmişi anlatılmıştır. Daha sonra açık kaynak lisanslar ve göç süreçleri açıklanmıştır. Son olarak ise, temel bir göç modeli önerilmiştir. Bu çalışmada aynı zamanda son kullanıcıların özgür ve açık kaynak yazılımları benimsemeleri üzerinde etkili olan faktörler araştırılmıştır.

Anahtar Kelimeler: Açık Kaynak Yazılım, AKY, Özgür Yazılım, ÖAKY, Açık Kaynak Göçü, Açık Kaynak Lisansları.

ABSTRACT

There are many benefits to be gained from migrating to the FOSS systems, including free or low licence costs, higher stability, increased security, flexibility and no vendor lock-in. This study can be helpful for organizations who are planning a migration to open source systems. Becasue this study sets out to describe the migration models and routes that organisations can take when migrating towards the Free and Open Source Software platforms.

A brief history of the early development of the operating systems like Unix, BSD, Linux and GNU are presented in the first chapter. Afterwards, open source licenses and migration processes were explained. Finally, a basic migration model has been proposed. This study also explored various factors that may effect free and open source software adoption by clients.

Key Words: Open Source Software, OSS, Free Software, FOSS, Open Source Migration, Open Source Licenses.

ÖZGEÇMİŞ

1990 yılında Erzurum'da doğan Sefa YILMAZ, ilköğrenimini Erzurum, Ankara ve Kırıkkale'de çeşitli okullarda, orta öğrenimini ise Adnan Menderes Anadolu Lisesinde tamamladı. Lisans eğitimini Karadeniz Teknik Üniversitesi Elektrik Elektronik Mühendisliği Bölümünde tamamlayarak 2015 yılında mezun oldu. 2016-2018 yılları arasında Gömülü Sistem Mühendisi olarak özel bir firmada çalışmaya başladı. 2018 yılında Radyo ve Televizyon Üst Kurulunda Üst Kurul Uzman Yardımcısı olarak göreve başlamış olup yüksek lisans eğitimine ise, Ankara Üniversitesi Elektrik Elektronik Mühendisliği Ana Bilim Dalında hâlen devam etmektedir.