


T.C.
RADYO VE TELEVİZYON ÜST KURULU

# NESNE DEPOLAMA TEKNOLOJİLERİ VE RTÜK'E UYARLANABİLİRLİĞİ

UZMANLIK TEZİ

HAZAL KILIÇ

ANKARA
OCAK, 2025


RTÜK
RADYO VE TELEVİZYON ÜST KURULU

**T.C.**
**RADYO VE TELEVİZYON ÜST KURULU**

# NESNE DEPOLAMA TEKNOLOJİLERİ VE RTÜK'E UYARLANABİLİRLİĞİ

**UZMANLIK TEZİ**

**HAZAL KILIÇ**

**DANIŞMAN: ÜST KURUL UZMANI İBRAHİM ESER**

**ANKARA**
**OCAK, 2025**

Radyo ve Televizyon Üst Kurulu Başkanlığına,

Hazal KILIÇ adlı Üst Kurul Uzman Yardımcısının hazırladığı "Nesne Depolama Teknolojileri ve RTÜK'e Uyarlanabilirliği" başlıklı bu çalışma Tez Değerlendirme Komisyonu üyelerince oy birliği / oy çokluğu ile **Uzmanlık Tezi** olarak kabul edilmiştir.

| Komisyon Üyeleri | | İmza |
|---|---|---|
| Başkan | Ebubekir ŞAHİN<br>Üst Kurul Başkanı | …… |
| Üye | Dr. Deniz GÜÇER<br>Üst Kurul Üyesi | …… |
| Üye | Deniz GÜLER<br>Üst Kurul Başkan Yardımcısı | …… |
| Üye | Murat ELLİALTI<br>Daire Başkanı | …… |
| Üye | İbrahim ESER<br>Üst Kurul Uzmanı (Tez Danışmanı) | …… |

ONAY

Bu tez Radyo ve Televizyon Üst Kurulu Uzmanlığı Yönetmeliği'nin ilgili maddeleri uyarınca …/…/20… tarihinde kabul edilmiştir.

……
Ebubekir ŞAHİN
Üst Kurul Başkanı

# ETİK İLKELERE UYGUNLUK BEYANI

Bu tez çalışmasında;

Tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi,

Tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu,

Tez çalışmasında yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi,

Kullanılan verilerde ve ortaya çıkan sonuçlarda herhangi bir değişiklik yapmadığımı,

Sunduğum çalışmanın özgün olduğunu bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim. 23/01/2025

Hazal KILIÇ

# TEŞEKKÜR

"Nesne Depolama Teknolojileri ve RTÜK'e Uyarlanabilirliği" başlıklı tezin hazırlanması ve sunulması sürecinde, başta Üst Kurul Başkanı Sayın Ebubekir ŞAHİN olmak üzere; Üst Kurul Üyelerine ve Tez Değerlendirme Komisyonu Üyelerine teşekkürlerimi sunarım.

Tez çalışması sürecinde, bilgi ve tecrübeleriyle rehberlik eden ve değerli geri dönüşleriyle tezime katkıda bulunan tez danışmanım Üst Kurul Uzmanı Sayın İbrahim ESER'e teşekkürlerimi borç bilirim.

Her zaman olduğu gibi, bu süreçte de desteğini eksik etmeyen kıymetli aileme ve arkadaşlarıma sonsuz teşekkür ederim.

Ankara, 2025
Hazal KILIÇ

# ÖZET

## NESNE DEPOLAMA TEKNOLOJİLERİ VE RTÜK'E UYARLANABİLİRLİĞİ

KILIÇ, Hazal
Uzmanlık Tezi
Tez Danışmanı: Üst Kurul Uzmanı İbrahim ESER
Ocak, 2025, 91 sayfa

Veri depolama, modern bilgi ve iletişim sistemlerinin temel yapı taşlarından biridir. Bu alandaki teknolojik ilerlemeler ile veri miktarındaki artış, veri depolama çözümlerinin sürekli olarak evrim geçirmesini zorunlu kılmaktadır. Bu bağlamda, nesne depolama teknolojileri, özellikle büyük veri setlerinin etkili ve verimli bir şekilde yönetilmesi açısından öne çıkmaktadır. İlk ortaya çıktığı günden bu yana önemli gelişmeler kaydeden nesne depolama teknolojileri, son yıllarda yaygınlaşarak veri depolama teknolojileri arasında önemli bir rol oynamaktadır.

Bu tez çalışması kapsamında, veri depolama teknolojileri ele alınmış, depolama ve veri yönetimi ile depolama güvenliği konularına değinilmiştir. Nesne depolama teknolojileri teknik açıdan incelenmiş, avantajları ve dezavantajlarına yer verilmiş, kullanım alanları örneklerle açıklanmış ve diğer veri depolama biçimleriyle karşılaştırılmıştır. Ayrıca, nesne depolama teknolojilerinin çeşitli uygulama sınıfları ve bu sınıflara ait örnek çözümlerin çalışma prensipleri analiz edilmiştir.

Tezin sonunda, ele alınan konular ışığında nesne depolama teknolojilerinin Radyo ve Televizyon Üst Kurulu (RTÜK) bilişim sistemlerinde uygulanabilirliği değerlendirilmiş ve öneriler sunulmuştur. Bu değerlendirme, nesne depolamanın RTÜK'ün bilgi yönetim ihtiyaçlarını nasıl karşılayabileceği ve mevcut bilişim altyapısına nasıl katkı sağlayabileceğine dair bir bakış açısı sunmaktadır. Çalışma, veri depolama teknolojilerinin gelişen ihtiyaçlara yanıt verme yeteneğini ve nesne depolamanın bu süreçteki rolünü anlamak açısından katkı sağlamayı amaçlamaktadır.

**Anahtar Sözcükler:** veri depolama, veri depolama teknolojileri, nesne depolama, nesne depolama uygulamaları

# ABSTRACT

## OBJECT STORAGE TECHNOLOGIES AND APPLICABILITY TO RTUK

KILIÇ, Hazal
Specialization Thesis
Thesis Advisor: Supreme Council Expert İbrahim ESER
January, 2025, 91 pages

Data storage constitutes a fundamental component of contemporary information and communication systems. Technological advancements in this domain, coupled with the exponential growth in data volume, compel the continual evolution of data storage solutions. In this context, object storage technologies emerge as particularly significant, especially in terms of their efficacy and efficiency in managing large datasets. Since their inception, object storage technologies have experienced notable advancements and have increasingly established a crucial role among data storage solutions in recent years.

This thesis undertakes a comprehensive examination of data storage technologies, addressing issues related to storage and data management as well as data storage security. A technical analysis of object storage technologies is conducted, encompassing their advantages and disadvantages, while various application domains are elucidated through illustrative examples and comparative analyses with alternative data storage methods. Furthermore, the thesis analyzes different application categories of object storage technologies, detailing the operational principles of exemplary solutions within these categories.

In the conclusion of the thesis, the applicability of object storage technologies in the information systems of the Radio and Television Supreme Council (RTÜK) has been evaluated, and recommendations have been provided. This evaluation provides insights into how object storage can meet RTÜK's information management needs and contribute to the existing IT infrastructure. The study aims to contribute to understanding the ability of data storage technologies to respond to evolving needs and the role of object storage in this process.

**Key Words:** data storage, data storage technologies, object storage, object storage implementations

# İÇİNDEKİLER

**Sayfa**

ETİK İLKELERE UYGUNLUK BEYANI ..... iii
TEŞEKKÜR ..... iv
ÖZET ..... v
ABSTRACT ..... vi
İÇİNDEKİLER ..... vii
TABLOLAR DİZİNİ ..... x
ŞEKİLLER DİZİNİ ..... xi
KISALTMALAR ..... xii
BÖLÜM 1 ..... 1
GİRİŞ ..... 1
Problem ..... 1
Amaç ..... 1
Önem ..... 2
Sınırlılıklar ..... 2
Tanımlar ..... 2
YÖNTEM ..... 4
Araştırmanın Modeli ..... 4
Evren ve Örneklem ..... 4
Verilerin Toplanması ..... 5
Verilerin Çözümlenmesi ..... 5
BÖLÜM 2 ..... 6
VERİ DEPOLAMA ..... 6
2.1. Veri ..... 6
2.1.1. Yapılandırılmış Veri ..... 7
2.1.2. Yapılandırılmamış Veri ..... 8
2.1.3. Yarı-Yapılandırılmış Veri ..... 8
2.2. Veri Depolamanın Tarihsel Gelişimi ..... 9
2.3. Veri Depolama Teknolojileri ..... 11
2.3.1. Veri Depolama Medyaları ..... 11
2.3.2. Doğrudan Bağlı Depolama (DAS) ..... 12
2.3.3. Ağ Bağlı Depolama (NAS) ..... 13

2.3.4. Depolama Alan Ağı (SAN) ... 13
2.3.5. Blok Depolama ... 14
2.3.6. Dosya Depolama ... 15
2.3.7. Nesne Depolama ... 17
2.3.8. İçerik Adresli Depolama (CAS) ... 19
2.3.9. Yüksek Düzeyli Veri Erişim Hizmetleri ... 19
2.3.10. Yazılım Tanımlı Depolama (SDS) ... 20
2.3.11. Depolama Sanallaştırması ... 20
2.3.12. Sanal Sunucular ve Konteynerler İçin Depolama ... 21
2.3.13. Bütünleşik ve Hiper-Bütünleşik Depolama ... 23
2.3.14. Bulut Depolama ... 24
2.4. Depolama ve Veri Yönetimi ... 24
2.4.1. Depolama Kaynaklarının Yapılandırılması ve Kaynak Yönetimi ... 25
2.4.2. Veri Sınıflandırılması ve Kategorizasyonu ... 25
2.4.3. Veri Temizleme ... 26
2.4.4. Veri Saklama ... 27
2.4.5. Veri Koruma ... 27
2.4.6. Veri Azaltma ... 29
2.5. Depolama Güvenliği ... 30
2.5.1. Tehditler, Riskler ve Saldırı Yüzeyleri ... 30
2.5.1.1. Tehditler ... 31
2.5.1.2. Riskler ... 32
2.5.1.3. Saldırı Yüzeyleri ... 33
2.5.2. Önlemler ... 35
BÖLÜM 3 ... 41
NESNE DEPOLAMA ... 41
3.1. Nesne Depolamaya İlişkin İlk Çalışmalar ... 41
3.2. Nesne Depolamanın Önemi ... 42
3.3. Nesne Depolama Sınıfları ... 43
3.4. Nesne ... 44
3.4.1. Nesne Türleri ... 44
3.4.1.1. Kök Nesne ... 44
3.4.1.2. Bölüm ... 45
3.4.1.3. Koleksiyon ... 45

3.4.1.4. Kullanıcı Nesnesi ..... 46
3.4.2. Öznitelikler ..... 46
3.4.2.1. OSD Tarafından Belirlenen Öznitelikler ..... 47
3.4.2.2. Statik Öznitelikler ..... 47
3.4.2.3. Oturum Öznitelikleri ..... 48
3.4.2.4. Genişletilmiş Öznitelikler ..... 49
3.5. Nesne Depolama Cihazı (OSD) ..... 49
3.5.1. Temel Nesne Depolama Cihazı Komutları ..... 52
3.5.2. Nesne Depolama Cihazında Güvenlik ..... 54
3.6. Nesne Depolamanın Avantajları ve Dezavantajları ..... 57
3.6.1. Avantajları ..... 57
3.6.2. Dezavantajları ..... 59
3.7. Nesne Depolamanın Kullanım Alanları ve Örnekleri ..... 60
3.8. Nesne Depolamanın Diğer Depolama Biçimleri ile Karşılaştırılması ..... 61
3.9. Nesne Depolama ve Zekâ ..... 62
BÖLÜM 4 ..... 64
NESNE DEPOLAMA UYGULAMALARI VE ÇÖZÜMLERİ ..... 64
4.1. Nesne Tabanlı Dosya Sistemleri ..... 64
4.1.1. Red Hat Ceph ..... 64
4.2. Bulut Nesne Depolama ..... 66
4.2.1. Amazon Simple Storage Service (S3) ..... 66
4.2.2. EMC Elastic Cloud Storage (ECS) ..... 69
4.3. Arşiv Depolama ..... 70
4.3.1. Hitachi Data Systems-Hitachi Content Platform (HCP) ..... 71
4.4. Nesne Depolama Sistemleri ..... 74
4.4.1. NetApp StorageGRID WebScale ..... 74
4.4.2. OpenStack Swift ..... 76
BÖLÜM 5 ..... 78
SONUÇLAR VE ÖNERİLER ..... 78
Sonuçlar ..... 78
Öneriler ..... 79
Radyo ve Televizyon Üst Kurulu Kapsamında Öneriler ..... 81
KAYNAKLAR ..... 84
ÖZGEÇMİŞ ..... 91

# TABLOLAR DİZİNİ

**Tablo** **Sayfa**

Tablo 1 Temel OSD Komutları……52
Tablo 2 Yeni Bir Dosya Oluştururken Kullanılan OSD Komutları…….53
Tablo 3 Nesneye Veri Aktarımı İşlemi Sırasında Kullanılan OSD Komutları…… 53
Tablo 4 Dosya Güncelleme Sürecinde Kullanılan OSD Komutları…….54
Tablo 5 Dosya – Blok – Nesne Depolamanın Karşılaştırılması……...62

# ŞEKİLLER DİZİNİ

**Şekil** **Sayfa**

Şekil 1 Blok Depolamada Veri Yapısı…….15
Şekil 2 Dosya Depolamada Veri Yapısı……...16
Şekil 3 Nesne Depolamada Veri Yapısı……...17
Şekil 4 Geleneksel Depolama Modeli ve Nesne Depolama Modeli……....51
Şekil 5 Nesne Depolama Cihazında Güvenlik Mimarisi……. 55
Şekil 6 Red Hat Ceph Depolama Kümesi Bileşenleri……..65
Şekil 7 Amazon S3 Çalışma Diyagramı……...68
Şekil 8 EMC ECS Mimarisinin Katmanları……70
Şekil 9 HCP'de Nesne Yapısı……...71
Şekil 10 StorageGRID Webscale Temel Bileşenleri……75

# KISALTMALAR

| Kısaltma | Açıklama |
|---|---|
| ACL | Erişim Kontrol Listesi (Access Control List) |
| API | Uygulama Programlama Arabirimi (Application Programming Interface) |
| AWS | Amazon İnternet Hizmetleri (Amazon Web Services) |
| BLOB | İkili Formatta Büyük Nesneler (Binary Large Objects) |
| CAS | İçerik Adresli Depolama (Content-Addressed Storage) |
| CCPA | Kaliforniya Tüketici Gizliliği Yasası (California Consumer Privacy Act) |
| CD | Kompakt Disk (Compact Disc) |
| CDP | Sürekli Veri Koruma (Continuos Data Protection) |
| CIFS | Ortak İnternet Dosya Sistemi (Common Internet File System) |
| CLI | Komut Satırı Arabirimi (Command Line Interface) |
| DAS | Doğrudan Bağlı Depolama (Direct Attached Storage) |
| DVD | Dijital Video Disk (Digital Video Disc) |
| EC | Silme Kodlaması (Erasure Coding) |
| ECS | Elastic Cloud Storage |
| FC | Fiber Kanal (Fibre Channel) |
| FCoE | Ethernet Üzerinden Fiber Kanal (Fiber Channel over Ethernet) |
| FTP | Dosya Transfer Protokolü (File Transfer Protocol) |
| GDPR | Genel Veri Koruma Yönetmeliği (General Data Protection Regulation) |
| HCI | Hiper-Bütünleşik Altyapı (Hyper Converged Infrastructure) |
| HCP | Hitachi Content Platform |
| HDD | Sabit Disk Sürücüsü (Hard Disk Drive) |
| HIPAA | Sağlık Sigortası Taşınabilirlik ve Sorumluluk Yasası (Health Insurance Portability and Accountability Act) |
| HTTP | Hiper Metin Transfer Protokolü (Hyper-Text Transfer Protocol) |
| HTML | Hiper Metin İşaretleme Dili (Hyper Text Markup Language) |
| IBM | International Business Machines |

| | |
|---|---|
| IDC | Ulusal Veri Şirketi (International Data Corporation) |
| ILM | Bilgi Yaşam Döngüsü Yönetimi (Information Lifecycle Management) |
| IP | İnternet Protokolü (Internet Protocol) |
| ISO | Uluslararası Standart Organizasyonu (International Standart of Organization) |
| iSCSI | İnternet Küçük Bilgisayar Sistemleri Arabirimi (Internet Small Computer Systems Interface) |
| JSON | JavaScript Nesne Gösterimi (JavaScript Object Notation) |
| LUN | Mantıksal Birim Numarası (Logical Unit Number) |
| MCA | Music Corporation of America |
| MIT | Massachusetts Teknoloji Enstitüsü (Massachusetts Institute of Technology) |
| NAS | Ağ Bağlı Depolama (Network Attached Storage) |
| NASD | Ağa Bağlı Güvenli Diskler (Network Attached Secure Disks) |
| NIST | Ulusal Standartlar ve Teknoloji Enstitüsü (National Institute of Standard and Technology) |
| NFS | Ağ Dosya Sistemi (Network File System) |
| NSIC | Ulusal Depolama Endüstri Konsorsiyumu (National Storage Industry Consortium) |
| NTP | Ağ Zaman Protokolü (Network Time Protocol) |
| NVMe | Hızlı Geçici Olmayan Bellek (Non-Volatile Memory Express) |
| NVMe-oF | Ağ Üzerinden Hızlı Geçici Olmayan Bellek (Non-Volatile Memory Express over Fabrics) |
| OBS | Nesne Tabanlı Depolama (Object-Based Storage) |
| OSD | Nesne Depolama Cihazı (Object Storage Device) |
| OSS | Nesne Depolama Sistemleri (Object Storage Systems) |
| PATA | Paralel Gelişmiş Teknoloji Bağlantısı (Parallel Advanced Technology Attachment) |
| PCI DSS | Ödeme Kartı Endüstrisi Dijital Güvenlik Standardı (Payment Card Industry Data Security Standard) |
| PHI | Korunan Sağlık Bilgisi (Protected Health Information) |
| PII | Kişisel Olarak Tanımlanabilir Bilgi (Personally Identifiable Information) |

| | |
|---|---|
| PKI | Açık Anahtar Yapısı (Public Key Infrastructure) |
| RDMA | Uzak Doğrudan Bellek Erişimi (Remote Direct Memory Express) |
| REST | Temsili Durum Aktarımı (Representational State Transfer) |
| RPO | Kurtarma Noktası Hedefi (Recovery Point Objective) |
| RTO | Belirli Kurtarma Süresi Hedefi (Recovery Time Objective) |
| RTÜK | Radyo ve Televizyon Üst Kurulu |
| SAN | Depolama Alan Ağı (Storage Area Network) |
| SCSI | Küçük Bilgisayar Sistemi Arayüzü (Small Computer Systems Interface) |
| SDS | Yazılım Tanımlı Depolama (Software-Defined Storage) |
| SIEM | Güvenlik Bilgi ve Olay Yönetimi (Security Information and Event Management) |
| SMB | Sunucu Mesaj Bloğu (Server Message Block) |
| SKAAS | Sayısal Kayıt, Arşiv ve Analiz Sistemi |
| SNIA | Depolama Ağ Endüstrisi Derneği (Storage Networking Industry Association) |
| SNMP | Basit Ağ Yönetim Protokolü (Simple Network Management Protocol) |
| SSD | Katı Hâl Sürücüsü (Solid State Drive) |
| SQL | Yapılandırılmış Sorgu Dili (Structured Query Language) |
| S3 | Simple Storage Service |
| TCP | Aktarma Kontrol Protokolü (Transmission Control Protocol) |
| TÜBİTAK | Türkiye Bilimsel ve Teknik Araştırma Kurumu |
| t.y. | tarih yok |
| USB | Evrensel Seri Veri Yolu (Universial Serial Bus) |
| vb. | ve benzeri |
| vd. | ve diğerleri |
| VM | Sanal Sunucu (Virtual Machine) |
| VPC | Sanal Özel Bulut (Virtual Private Cloud) |
| QoS | Erişim Hizmet Kalitesi (Quality of Services) |
| XML | Genişletilebilir İşaretleme Dili (Extensible Markup Language) |

# BÖLÜM 1

# GİRİŞ

Bu başlık kapsamında; araştırmanın problemi, hedeflenen amaçlar, önemi ve sınırlılıkları ele alınmıştır. Ayrıca, araştırmada kullanılan kavramlara ilişkin tanımlamalar yapılmıştır.

## Problem

Veri depolama teknolojileri, dijital çağın hızla değişen gereksinimlerine yanıt verebilmek için sürekli olarak evrim geçirmektedir. Bu evrimin bir parçası olarak, nesne depolama teknolojisi son yıllarda önemli bir rol oynamıştır. Özellikle büyük veri setlerinin depolanmasında tercih edilen nesne depolama teknolojisi, veri yönetimi ve veriye erişim performansını optimize etme potansiyeli nedeniyle dikkat çekmektedir. Bu bağlamda, nesne depolamanın veri depolama teknolojilerindeki yerini ve rolünü anlamak, hem akademik hem de pratik açıdan kritik bir öneme sahiptir. Ancak, mevcut literatürde nesne depolamanın veri depolama teknolojilerindeki rolü yeterince kapsamlı bir şekilde incelenmemiştir.

Bu çalışmada, nesne depolamanın veri depolama teknolojilerindeki rolünü anlamak amacıyla aşağıdaki sorulara yanıt aranacaktır:

- Mevcut veri depolama teknolojileri nelerdir?
- Nesne depolama nasıl çalışır? Nesne depolamanın avantajları ve dezavantajları nelerdir? Nesne depolamanın kullanım alanları nelerdir?

## Amaç

Bu çalışmanın amacı, mevcut araştırmalardan yararlanarak belirtilen sorulara yanıt aramak ve nesne depolama konusunu kapsamlı bir şekilde incelemektir. Böylece,

araştırmanın problemini çözmeye yönelik bu alana akademik bir katkı sağlanması hedeflenmektedir.

## Önem

Son yıllarda veri üretimi ve tüketimi hızla artarken, büyük veri yığınlarının etkin ve verimli bir şekilde depolanması giderek daha kritik bir hâle gelmiştir. Nesne depolama teknolojisi, yüksek hacimli verilerin yönetimi ve saklanmasında önemli bir çözüm sunmaktadır. Bu çalışma, nesne depolamanın veri depolama teknolojilerindeki rolünü ve bu alandaki katkılarını inceleyerek, veri yönetimi ve depolama stratejileri üzerinde genel bir anlayış kazandırması bakımından önemlidir.

## Sınırlılıklar

Bu çalışma kapsamında;

Veri depolama teknolojilerinin geniş bir yelpazesine yer verilmiş olsa da, tüm mevcut teknolojilere dair kapsamlı bir inceleme sunmamaktadır. Özellikle, bazı niş teknolojiler çalışmanın dışında kalmıştır.

Nesne depolama teknolojilerinin temel özellikleri, avantajları ve dezavantajları üzerine odaklanmıştır. Ancak, bu teknolojilerin entegrasyon süreçleri derinlemesine incelenmemiştir.

## Tanımlar

Büyük veri: Teknolojide, büyük veri kümeleri için kullanılan bir terimdir. Büyük veri, geleneksel olarak üç temel "V" ile tanımlanır; bunlar, hacim, hız ve çeşitliliktir. Hacim, bu tür veri kümelerinin büyük boyutunu; hız, bu verilerin üretildiği ve analiz edildiği hızını; çeşitlilik ise, metin, ses, video veya diğer biçimlerde olabilen çok çeşitli veri türlerini ifade eder.

Yapay zekâ: Yapay zekâ, bir bilgisayarın veya bilgisayar kontrollü robotun, genellikle akıllı varlıklarla ilişkilendirilen görevleri yerine getirme yeteneğidir. Yapay zekâ, makinelerin insan benzeri düşünme ve karar verme yetenekleri göstermesini sağlar.

Makine öğrenmesi: Yapay zekâ alanında, otonom olarak öğrenebilen bilgisayar yazılımlarının geliştirilmesiyle ilgili bir disiplindir. Makine öğrenmesi, bilgisayar sistemlerinin verilerden öğrenmesini ve deneyimlere dayanarak performanslarını artırmasını sağlar.

Sanallaştırma: Sunucu, depolama, ağ ve diğer fiziksel makinelerin sanal temsillerini oluşturmak için kullanılan bir teknolojidir. Sanallaştırma, kaynakların daha verimli kullanılmasını ve yönetimini sağlar, fiziksel altyapının üzerinde birden fazla sanal ortam oluşturur.

Hipervizör: Tek bir fiziksel makinede birden fazla sanal makineyi çalıştırmak için kullanılan bir yazılımdır. Hipervizör, sanal makinelerin kaynaklara erişimini yönetir ve sanal ortamların performansını optimize eder.

Bulut bilişim: Uygulama yazılımlarını çalıştırma ve ilgili veri merkezi bilgisayar sistemlerinde depolama ile müşterilere veya diğer kullanıcılara internet üzerinden erişim sağlama yöntemidir. Bulut bilişim, kaynakların internet üzerinden erişilebilmesini sağlar ve genellikle ölçeklenebilir ve maliyet etkin bir çözüm sunar.

Ana bilgisayar (Host): Diğer bilgisayarların bir ağ veya modem aracılığıyla erişebileceği veri veya programlar içeren bir bilgisayardır. Ana bilgisayar, veri ve uygulama hizmetleri sunarak ağ üzerindeki diğer cihazlara destek sağlar.

Sunucu: Ağ üzerinde bilgisayar programları veya bir istemciden gelen istekleri işleyen aygıttır. Sunucular, istemcilerden gelen veri taleplerini karşılamak ve çeşitli hizmetler sunmak için yapılandırılır.

Protokol: Bilgisayar biliminde, bilgisayarlar ve diğer elektronik cihazlar arasında veri iletimi için belirlenmiş kurallar ve prosedürler kümesidir. Protokoller, veri iletişiminin doğru ve güvenli bir şekilde gerçekleşmesini sağlar.

Veri tabanı: Bir bilgisayar tarafından hızlı arama ve erişim için özel olarak düzenlenmiş veri veya bilgi koleksiyonudur. Veri tabanları, verilerin düzenli bir biçimde saklanmasını ve hızlı bir şekilde erişilmesini sağlayan sistemlerdir.

Meta veri: Verilerin, bilgisel yönleriyle ilgili verilerdir. Meta veriler, verilerin kolayca alınmasını, yönetilmesini ve kullanılmasını sağlar.

# YÖNTEM

Bu başlık kapsamında; araştırmanın modeli, çalışmanın evren ve örneklemi açıklanmıştır. Ayrıca, çalışma boyunca elde edilen verilerin toplanması ve verilerin çözümlenmesi konularına değinilmiştir.

## Araştırmanın Modeli

Bu çalışma süresince yapılan araştırmalarda çeşitli yöntemler ve analizler kullanılmıştır:

- Veri depolama teknolojileri ve nesne depolama hakkında mevcut literatür taranarak, bu alandaki temel kavramlar, teknolojiler ve yöntemler belirlenmiştir. Bu aşamada, akademik makaleler, teknik raporlar, endüstri raporları ve ilgili literatür kaynakları analiz edilmiştir.
- Veri depolama, teorik bir çerçevede ele alınmıştır. Bu çerçevede, veri türleri, veri depolama teknolojileri, veri yönetimi ve depolama güvenliği gibi konular detaylandırılmıştır.
- Nesne depolamanın farklı yönleri, avantajları, dezavantajları incelenmiştir. Nesne depolamanın tarihi, kullanım alanları ve örnekleri ele alınmıştır.
- Nesne depolama ve diğer depolama biçimleri arasında karşılaştırmalı bir analiz gerçekleştirilmiştir.
- Nesne depolama uygulamaları ve çözümleri üzerinde durulmuş ve örnek çözümlerin özelliklerine değinilmiştir.
- Elde edilen bilgiler doğrultusunda nesne depolama teknolojilerinin RTÜK'e uyarlanabilirliği değerlendirilmiş ve öneriler sunulmuştur.

## Evren ve Örneklem

Bu çalışmada evren, geniş bir veri kümesi olan veri depolama teknolojilerini ifade eder. Bu kapsamda, veri depolama teknolojilerinin tarihi gelişimi, mevcut çeşitleri, kullanım alanları ve uygulama örnekleri ile ilgili bilgileri içeren bir evren söz konusudur.

Örneklem ise nesne depolama teknolojileri ve uygulamaları üzerine yapılmış çeşitli raporlar, akademik çalışmalar ve sektör analizlerinden elde edilen verilerden oluşmaktadır. Ayrıca, belirli nesne depolama uygulamaları, bu teknolojilerin avantajları, dezavantajları ve kullanım alanları ile ilgili örnekler de örneklem olarak değerlendirilmiştir.

## Verilerin Toplanması

Bu çalışmanın kapsamına uygun olarak, yazılmış kitaplar, makaleler ve raporlar fiziksel ve dijital kütüphanelerden temin edilmiştir. DergiPark, Google Akademik ve YÖK Tez Merkezi gibi akademik veri tabanları kullanılarak, makale, tez ve bildiriler gibi yapılmış araştırmalara ulaşılmıştır. Konu hakkındaki standartlar araştırılmış, raporlar ve teknik analizler incelenmiştir. Teknoloji firmalarının ve veri depolama sağlayıcılarının resmi internet siteleri, ürün belgeleri ve kılavuzları gibi çevrimiçi kaynaklardan yararlanılmıştır.

## Verilerin Çözümlenmesi

İnternet ortamından, dijital ve fiziksel kütüphanelerden elde edilen elektronik, yazılı ve basılı kaynaklar, raporlar, dergi makaleleri ve diğer akademik kaynaklar kullanılarak kapsamlı bir veri seti oluşturulmuştur. Toplanan veriler, geçerlilik ve güvenilirlik açısından incelenmiş ve gerekiyorsa temizlenmiştir. Geriye kalan veriler, mevcut teorik çerçeve ve literatür bilgileri doğrultusunda değerlendirilmiştir.

# BÖLÜM 2

# VERİ DEPOLAMA

Bilgi teknolojilerinin temel yapı taşlarından biri olan veri depolama, verilerin kalıcı veya geçici olarak muhafaza edilmesi sürecini ifade etmektedir. Verilerin güvenli bir şekilde saklanması, veri kaybının önlenmesi ve veri erişiminin hızlı ve verimli bir şekilde sağlanması, hem kurumlar hem de bireyler için büyük önem taşımaktadır. Veri depolama sistemlerinin etkinliği, veri analizi ve yönetimi gibi süreçleri doğrudan etkileyebilmektedir.

Tarih boyunca, teknolojinin evrimi ve toplumsal ihtiyaçların değişimi doğrultusunda veri depolama teknolojileri sürekli olarak değişmiş ve gelişmiştir. Bu bağlamda, veriler veri depolama teknolojilerinin gelişiminde belirleyici bir rol oynamıştır.

Bu bölümde, öncelikle veri tanımsal bir yaklaşım çerçevesinde ele alınmış ve yapılarına göre veri türlerine yer verilmiştir. Ardından, veri depolamanın tarihsel gelişimi ve çeşitli veri depolama teknolojileri incelenmiştir. Ayrıca, depolama ve veri yönetimi ile depolama güvenliği konularına da değinilmiştir.

## 2.1. Veri

Günümüz bilgi çağının en değerli unsurlarından biri veridir. Veri, bireylerin günlük yaşantılarında, üretimde, sağlıkta, çalışma hayatında vb. pek çok faaliyette değişime neden olabilecek dijital dönüşümün hammaddesidir (Boz Eravcı, 2020). Kapsamlı bir terim olan veri, geniş bir tanım yelpazesine sahiptir ve çeşitli akademik perspektifler tarafından ele alınmaktadır.

Türk Dil Kurumu (TDK), veriyi en genel hâliyle “Bir araştırmanın, tartışmanın, bir muhakemenin temeli olan ana öge, muta, done” ve “gözlem ve deneye dayalı elde edilen sonuçlar” olarak tanımlamıştır. Kalseth ve Cummings (2001), veriyi yorumsuz ve içeriksiz şekiller ve olgular olarak değerlendirmiştir.

Albayrak ve Koltan Yılmaz (2009), ham verinin kendi başına değersiz olduğunu ve bilgisayar sistemlerinde belirli bir amaca yönelik işlenerek bilgiye dönüştüğünü belirtmiştir. Yılmaz (2009), veriyi tek başına anlam ifade etmeyen veya kullanılamayan,

ancak enformasyon ve bilgiye temel oluşturan ham bilgiler olarak tanımlamış ve bu bilgilerin ilişkilendirilme, gruplandırılma, yorumlanma, anlamlandırılma ve analiz edilme gereksinimini vurgulamıştır. Rotella'nın (2012) aktardığına göre, Ann Winblad veriyi "yeni petrol" olarak tanımlamış, Michael Palmer ise veriyi "ham petrol" olarak nitelendirip, işlenmediği takdirde değerinin ortaya çıkamayacağını ifade etmiştir.

Doğan ve Arslantekin (2016) verinin, araştırmalar, gözlemler, internet, sosyal medya, sensörler gibi çeşitli ortamlardan elde edilebileceğini belirtmişlerdir. Şeker (2013) ise veriyi, herhangi bir işleme tabi tutulmadan, gözlem veya ölçüm yöntemleriyle ortamdan elde edilen her türlü değer olarak tanımlamıştır. Bu tanımlamalar ışığında, veri gözlem veya ölçüm yöntemleriyle çeşitli kaynaklardan elde edilen ve işlendiğinde anlamlı sonuçlar çıkarılabilecek ham değer olarak ifade edilebilir.

Veriler, homojen bir nitelik taşımaz ve farklı yapılarda bulunabilmektedir. Veri yapısı, bilgilerin özümlenme ve erişim hızını etkileyen önemli bir faktördür. Yapılarına göre veriler; yapılandırılmış veri, yapılandırılmamış veri ve yarı-yapılandırılmış veri olmak üzere üç temel kategoriye ayrılmaktadır (Mansouri vd., 2017).

### 2.1.1. Yapılandırılmış Veri

Yapılandırılmış veri, belirli bir formata ve boyuta sahip olan veriyi ifade etmektedir. Bu veriler, organizasyonel kullanım için bilgi edinimi sağlamak amacıyla sorgulara yanıt verebilecek şekilde tanımlanabilir bir yapıda organize edilmektedir (Doan vd., 2009). Sayılar, tarihler ve kelimelerden oluşan gruplar, örneğin bir kişinin adı ve adresi gibi bilgiler, yapılandırılmış veri örnekleri arasında yer almaktadır (Hurwitz vd., 2013).

Yapılandırılmış veri, son derece düzenli bir yapıya sahip olduğundan, okunması, anlaşılması ve işlenmesi kolaydır. Ayrıca, makine öğrenme algoritmaları tarafından kolaylıkla çözülebilmektedir (IBM Cloud Education, 2021). Bu tür veriler, ilişkisel veri tabanlarında depolanabilir ve Yapılandırılmış Sorgu Dili (SQL) kullanılarak sorgulanabilmektedir (Hurwitz vd., 2013).

Yapılandırılmış veri, mevcut veri miktarının yaklaşık %20'sini oluşturmaktadır. Yapılandırılmış verinin temel iki kaynağı, insanlar ve makinelerdir. İnsanlar tarafından üretilen veriler, insanların makinelerle etkileşimleri sonucunda oluşmaktadır. İnsanlar tarafından üretilen bazı yapılandırılmış veri örnekleri arasında girdi verisi, tıklama verisi ve

oyun bağlantılı veri sayılabilir. Makineler tarafından üretilen veriler ise insan müdahalesi olmadan elde edilmektedir. İnternet log verisi, alışveriş verisi ve finans verisi makine tarafından üretilen yapılandırılmış veri örnekleridir (Dokuz ve Çelik, 2017).

### 2.1.2. Yapılandırılmamış Veri

Belirli bir formata veya boyuta sahip olmayan veri, yapılandırılmamış veri olarak tanımlanmaktadır (Hurwitz vd., 2013). Günümüzde organizasyonlardaki verilerin çoğu yapılandırılmamıştır ve bu miktar gün geçtikçe artmaktadır (Das ve Mohan Kumar, 2013; Eberendu, 2016). Uzmanlar, mevcut verilerin yaklaşık %80'inin yapılandırılmamış veri olduğunu öngörmektedir (Dialani, 2020). Organizasyonlar, yapılandırılmamış verilerin analiz edilmesi ve karar alma süreçlerine dâhil edilmesi durumunda elde edilecek potansiyelin farkına vardıklarından, yapılandırılmamış verilerin kullanımı gerçek dünya uygulamalarında büyük bir artış göstermektedir (Maluf ve Tran, 2008).

Genellikle nitel veri içeren yapılandırılmamış veri, geleneksel veri araçları ve yöntemleriyle işlenememektedir. Bu tür veriler, tanımlanmış bir veri modeline sahip olmadıklarından dolayı genellikle ilişkisel olmayan veri tabanlarında veya veri göllerinde depolanmaktadır (IBM Cloud Education, 2021).

Yapılandırılmamış veri, insanlar ve makineler olmak üzere iki ana kaynaktan elde edilmektedir. İnsanlar tarafından üretilen bazı yapılandırılmamış veri örnekleri arasında özel metin verisi, sosyal medya verisi, mobil veri ve internet sitesi içerikleri sayılabilir. Makineler tarafından üretilen yapılandırılmamış veri örnekleri arasında ise uydu fotoğrafları; sismik, atmosferik ve yüksek enerjili fizik verisi; radar ve sonar verisi; videolar yer almaktadır (Dokuz ve Çelik, 2017).

### 2.1.3. Yarı-Yapılandırılmış Veri

Yarı–yapılandırılmış veri, yapılandırılmış ve yapılandırılmamış veri türlerinin aynı kayıt içinde bir arada bulunmasıyla ortaya çıkan bir veri türüdür. Bu veri türü, bazı özellikleri itibarıyla yapılandırılmış veri karakteristiklerini taşırken, diğer özellikleri ise yapılandırılmamış veri niteliklerini yansıtmaktadır (Dokuz ve Çelik, 2017). Yarı-

yapılandırılmış veri modeli, farklı ancak ilişkili özelliklere sahip çeşitli kaynaklardan elde edilen bilgilerin bir bütün içinde birleştirilmesine olanak tanımaktadır (Hanig vd., 2010). Genişletilebilir İşaretleme Dili (XML), Hiper Metin İşaretleme Dili (HTML) ve JavaScript Nesne Gösterimi (JSON) yarı-yapılandırılmış veri örnekleridir (Karabay ve Ulaş, 2017).

## 2.2. Veri Depolamanın Tarihsel Gelişimi

Yaklaşık 64.000 yıl öncesine dönüp bakıldığında, insanların bilgiyi saklamak ve iletmek için yalnızca resimler ve oymalar gibi ilkel veri depolama yöntemlerine sahip olduğu görülmektedir. Bu dönemde yazılı bir metin sistemi mevcut olmadığından, veriler yazılı olarak kaydedilememiştir. Ancak, zaman içerisinde yazılı metinler oluşturulmuş ve veriler yazılı formatlarda saklanmaya başlanmıştır (Sivakumar vd., 2023).

Yaklaşık 1.000 yıl önce, baskı teknolojisinin icadıyla birlikte verilerin kâğıda basılması mümkün hâle gelmiştir. Bu gelişme, bilginin daha geniş kitlelere ulaşmasını sağlamış ve yazılı kaynakların korunmasını kolaylaştırmıştır. Ancak, zamanla veri depolama gereksinimleri artmış ve dönemin teknolojisi bu gereksinimleri karşılamakta yetersiz kalmıştır (Sivakumar vd., 2023).

Son birkaç yüzyılda, bilim insanları ve mucitler, geleneksel yöntemlere göre çok daha kullanışlı ve güvenilir dijital veri depolama çözümleri geliştirmişlerdir. Bu dijital çözümler, büyük miktarda verinin güvenli bir şekilde saklanmasına ve hızlı bir şekilde erişilmesine olanak tanımıştır. Veri depolamada meydana gelen bazı önemli gelişmeler ve teknolojiler aşağıda özetlenmiştir (Sivakumar vd., 2023):

- Delikli Kartlar ve Delikli Kâğıt Şeritleri (1890): Delikli kartlar ve delikli kâğıt şeritleri, mekanize bilginin ilk depolama biçimleri olarak kabul edilmektedir. 1890 yılında, istatistikçi Herman Hollerith, sayım verilerini toplamak amacıyla 960 bit kapasiteye sahip delikli kart tabulasyon makinesini icat etmiştir. Bu gelişme, yarı otomatik veri işleme sistemlerinin başlangıcı olarak değerlendirilmiştir. Delikli kartlar ve delikli kâğıt şeritleri, erken dönem bilgisayarların giriş ve çıkış cihazları olarak kullanılmış olup, veriler ikili kodlara dönüştürülmüştür; delikler 1, delik olmayan yerler ise 0 olarak kabul edilmiştir (Utmel, 2021).
- Williams-Kilburn Tüpü (1947): Manchester Üniversitesi'nde Freddie Williams ve Tom Kilburn tarafından geliştirilen Williams-Kilburn tüpü, 1947 yılında test edilmiştir. Bu tüp, ilk yüksek hızlı ve tamamen elektronik bellek sistemi olarak

kabul edilmektedir. Katot ışınlı tüp kullanarak bitler, ekran yüzeyinde noktalar olarak depolanmış ve veriler sürekli olarak yenilenmiştir. Veriler, elektriksel yük değişikliklerini algılayan bir metal alıcı plakayla okunmuştur (Computer History Museum, t.y.).

- Manyetik Bant (1951): Mühendis Fritz Pfleumer tarafından icat edilen manyetik bant, analog sinyalleri depolama kapasitesine sahip bir ses bandıdır. Bu bantlar, ezilmiş manyetik parçaların bir kâğıt şeridine yapıştırılmasıyla hazırlanmıştır. Sıralı erişim sağlamakta olup, bu özellikleriyle, disk veri depolama teknolojilerinden ayrılmaktadır. 1951 yılında bilgisayarlarda kullanılmaya başlanmıştır (Computer History Museum, t.y.).
- Sabit Disk Sürücüsü (HDD) (1956): International Business Machines (IBM) tarafından 1956 yılında üretilen ilk sabit disk sürücü, IBM Model 350 Disk Dosyası olarak bilinmektedir. Bu disk dosyası, IBM 305 RAMAC bilgisayarları için tasarlanmış olup, toplamda 5 milyon karakter depolama kapasitesine sahiptir. Veriler, 50 adet 24 inç çapında manyetik kaplı metal diskte saklanmıştır (Computer History Museum, t.y.).
- Manyetik Disk (1978): Music Corporation of America (MCA) ve Philips tarafından ortaklaşa geliştirilen ve "Discovision" olarak tanıtılan manyetik disk, optik disk depolama teknolojisinin ilk örneklerinden biridir. Manyetik disk, 1980'lerin büyük bir kısmında önemini korumuş ve birkaç yıl içinde oldukça popüler hâle gelen Kompakt Disk (CD) ve Dijital Video Disk (DVD) teknolojilerinin temelini oluşturmuştur (Computer History Museum, t.y.).
- Katı Hâl Sürücüsü (SSD) (1992): SanDisk tarafından üretilen ticari flaş tabanlı katı hâl sürücüsü, dizüstü bilgisayarlar için geliştirilmiştir. Bu sürücülerde dönen diskler yerine uçucu olmayan bellek yongaları kullanılmıştır. 1998 yılında Paralel Gelişmiş Teknoloji Bağlantısı (PATA) arabirimli cihazlar için 2.5 inç ve 3.5 inç form faktörü sürücüler tanıtılmıştır (Computer History Museum, t.y.).
- Evrensel Seri Veri Yolu (USB) Flaş Sürücüler (2000): USB flaş sürücüler, USB arabirimine sahip flaş bellek teknolojisini kullanmaktadır. Dosyaları depolama, yedekleme veya cihazlar arasında transfer yapma amaçlarıyla kullanılabilmektedir. Küçük boyut, yüksek hız ve büyük depolama kapasitesi gibi avantajlar sunmaktadır. Uygun fiyatlı olmaları ve yüksek dayanıklılıkları sayesinde, masaüstü

ve dizüstü bilgisayar üreticileri disket ve CD yuvaları yerine USB bağlantı noktalarını tercih etmektedir (Computer History Museum, t.y.).

- Bulut Depolama (2006): Bulut depolama, veri depolama hizmetlerinin "hizmet olarak" sunulmaya başlanmasıyla popülerlik kazanmıştır. Amazon İnternet Hizmetleri (AWS) Simple Storage Service (S3) ile başlayan bu süreç, verilerin mantıksal havuzlar şeklinde dijital olarak depolanmasını mümkün kılmıştır. Bu veriler, bulut hizmet sağlayıcıları tarafından erişilebilirlik ve kullanılabilirlik açısından dış mekânda saklanmaktadır. Bulut tabanlı depolamanın avantajları arasında bakım süresinin azalması ve daha düşük sermaye harcaması bulunmaktadır (Computer History Museum, t.y.).

## 2.3. Veri Depolama Teknolojileri

Veri depolama teknolojileri, bilgisayar verilerini saklamak için kullanılan cihazları, nesneleri (örneğin, depolama elemanları, depolama kümeleri, depolama ağ anahtarları veya depolama medyaları) ve süreçleri (örneğin, protokoller ve arayüzler) kapsamaktadır. Bu teknolojiler, çeşitli sınıflarda değerlendirilmektedir (Chandramouli ve Pinhas, 2020).

### 2.3.1. Veri Depolama Medyaları

Veri depolama medyası, verileri depolayabilen ve geri alabilen her türlü cihazın parçasıdır (Law Insider, t.y.). Veri depolama teknolojilerinin en çok gelişim gösterdiği alanlardan biri veri depolama medyalarıdır. Veri depolama medyaları, çeşitli teknolojileri kapsamaktadır (Chandramouli ve Pinhas, 2020). Yaygın olarak kullanılan veri depolama medya seçenekleri aşağıda sıralanmıştır (Blumzon ve Panescu, 2019):

- Manyetik Medya: Manyetik medya, veriyi özel bir yüzeyin manyetizasyon desenleri kullanarak depolar. Bu tür medyalara örnek olarak HDD ve manyetik bantlar verilebilir.
- Optik Medya: Optik medya, veriyi bir lazer diyodu tarafından aydınlatıldığında okunabilen dairesel bir yüzeydeki deformasyonlar şeklinde depolar. Bu medya türüne örnek olarak CD ve DVD gösterilebilir.

- Yarı İletken Medya: Yarı iletken medya, veriyi yarı iletken bazlı entegre devreler kullanarak depolar. Geleneksel olarak, bu tür teknolojiler verileri geçici olarak saklar; elektrik sağlanmadığında veri kaybolur. Ancak, yarı iletken medya örneği olan SSD, tüketici bilgisayarlarında yer almakta olup, manyetik sürücülerden daha hızlı erişim sağlayan kalıcı veri depolama çözümleri sunar.

### 2.3.2. Doğrudan Bağlı Depolama (DAS)

Depolama sistemi mimarisinin tarihine bakıldığında, dijital depolama altyapısının ilk biçimlerinden birinin Doğrudan Bağlı Depolama (DAS) olduğu görülmektedir (Chandramouli ve Pinhas, 2020). DAS, bir ağ aracılığına ihtiyaç duymadan, depolama öğesinin veya cihazının doğrudan ana bilgisayara bağlandığı bir dijital depolama yöntemini ifade etmektedir (Süß ve Süß, 2024). Depolama cihazının ana bilgisayara göre konumuna bağlı olarak DAS, iki kategoriye ayrılmaktadır (Somasundaram vd., 2009):

- İç DAS: İç DAS mimarilerinde, depolama cihazı ana bilgisayara seri veya paralel bir veri yolu aracılığıyla dâhili olarak bağlanır. Fiziksel veri yolunun mesafe sınırlamaları bulunmakta olup, yüksek hızlı bağlantılar yalnızca kısa mesafelerde sürdürülebilmektedir. Ayrıca, çoğu iç veri yolu sınırlı sayıda cihazı destekleyebilir ve ana bilgisayarın içinde büyük bir alan kaplar; bu da diğer depolama bileşenlerinin bakımını zorlaştırmaktadır.
- Dış DAS: Dış DAS, sunucunun doğrudan dış bir depolama cihazına bağlandığı bir mimaridir. Ana bilgisayar ile depolama cihazı arasındaki iletişim genellikle Küçük Bilgisayar Sistemi Arayüzü (SCSI) veya Fiber Kanal (FC) protokolleri üzerinden gerçekleştirilir. İç DAS ile karşılaştırıldığında, dış DAS mesafe ve cihaz sayısı sınırlamalarını aşar ve depolama cihazlarının merkezi yönetimini sağlamaktadır.

DAS, sunucuya özel depolama kaynakları, basit yapılandırma ve uygun maliyetli kurulum avantajları sunmaktadır. Ancak, sunucular arasında veri paylaşımının yapılamaması, depolama kaynaklarının birleştirilememesi, düşük kullanılabilirlik ve performans gibi dezavantajlara sahiptir (Surianarayanan ve Chelliah, 2023).

### 2.3.3. Ağ Bağlı Depolama (NAS)

Depolama Ağ Endüstrisi Derneği (SNIA) sözlüğünde Ağ Bağlı Depolama (NAS), "Bir ağa bağlanan ve bilgisayar sistemlerine dosya erişim hizmetleri sağlayan depolama aygıtlarını ifade etmek için kullanılan bir terim" olarak tanımlanmaktadır. NAS cihazları genellikle, dosya hizmetlerini sağlayan bir makineden ve verilerin depolandığı bir veya daha fazla aygıttan oluşmaktadır. NAS, dağıtılmış, bağımsız verileri büyük, merkezi olarak yönetilen veri merkezlerinde bir araya getirerek farklı ana bilgisayarlar ve uygulama sunucuları tarafından erişilebilir hâle getirmektedir (SNIA, 2024a).

NAS, dosya düzeyinde erişim sağlar ve dosyalama ile depolama işlevlerini gerçekleştirmek için ağ ve dosya paylaşım protokollerini kullanmaktadır. Bu protokoller arasında veri iletimi için Aktarma Kontrol Protokolü (TCP) / İnternet Protokolü (IP), uzak dosya hizmetleri için Ortak İnternet Dosya Sistemi (CIFS) ve Ağ Dosya Sistemi (NFS) bulunmaktadır. NAS, hem UNIX hem de Microsoft Windows kullanıcılarının aynı verileri sorunsuz bir şekilde paylaşabilmesini sağlamaktadır. Veri paylaşımı için NAS, genellikle UNIX için NFS, Microsoft Windows için CIFS ve her iki ortam için Dosya Transfer Protokolü (FTP) protokollerini kullanır (Somasundaram vd., 2009).

NAS'ın sunduğu avantajlar arasında ölçeklenebilirlik, gelişmiş esneklik, yüksek performans, uygun maliyet ve kullanışlılık bulunmaktadır. Bununla birlikte, ağ bant genişliği kullanımı, hizmet garanti eksiklikleri ve verimlilik sorunları NAS'ın dezavantajları arasında sayılabilir (Surianarayanan ve Chelliah, 2023).

### 2.3.4. Depolama Alan Ağı (SAN)

Depolama Alan Ağı (SAN), depolama aygıtlarına ağ erişimi sağlayan özel, yüksek hızlı bir ağ sistemidir (SNIA, 2024b). SAN sistemi, uzak depolama cihazlarını, depolama istemci yazılımının çalıştığı ana bilgisayara yerel olarak bağlıymış gibi sunmaktadır. SAN, yüksek hızlı bir ağ kullanarak blok düzeyinde erişim sağlamaktadır (Chandramouli ve Pinhas, 2020).

SAN tabanlı sistemlerin temel bileşenleri; ana bilgisayarlar (istemciler), anahtarların dağıtımını içeren topoloji ve depolama cihazları/kümeleridir. Bu üç bileşen çeşitli ağ yığınları aracılığıyla birbirine bağlanır. Ağ yığınları, depolama özel protokollerinden oluşabilir veya bu protokollerin üzerinde çalışabilmektedir. SAN

sistemleri çeşitli protokoller kullanmaktadır. Yaygın olarak kullanılan bazı SAN protokolleri şunlardır (Chandramouli ve Pinhas, 2020):

- FC SAN: FC SAN, beş katmandan oluşan fiber kanal protokolünü kullanan bir ağ yığınıdır. FC SAN tarafından adreslenen depolama kaynağı, Mantıksal Birim Numarası (LUN) ile adlandırılır.
- IP SAN: IP kullanarak çalışan bir ağ yığınıdır. İnternet Küçük Bilgisayar Sistemleri Arabirimi (iSCSI), IP SAN'a bir örnek olarak verilebilir.
- Ethernet Üzerinden Fiber Kanal (FCoE): Fiber Kanal çerçevelerinin Ethernet paketlerine kapsüllendiği bir protokoldür.
- Fiber Üzerinden Hızlı Geçici Olmayan Bellek (NVMe-oF) Protokolleri: Hızlı Geçici Olmayan Bellek (NVMe) komutlarının ağ üzerinden iletilmesini sağlayan bir protokol arayüzüdür ve ilgili uzantıları tanımlamaktadır. NVMe-oF, NVMe uygulamasını yerel bir bilgisayardan uzak bir bilgisayara genişleterek ölçeklenebilir bir NVMe depolama sistemi sağlar. NVMe-oF protokolleri arasında Uzak Doğrudan Bellek Erişimi (RDMA), FC ve TCP bulunmaktadır. RDMA, uygulama belleği arasında veri okuma ve yazma işlemlerine olanak tanımaktadır. Infiniband, yüksek performanslı bilgi işlem ortamlarında RDMA'yı mümkün kılan endüstri standardı bir bağlantıdır.

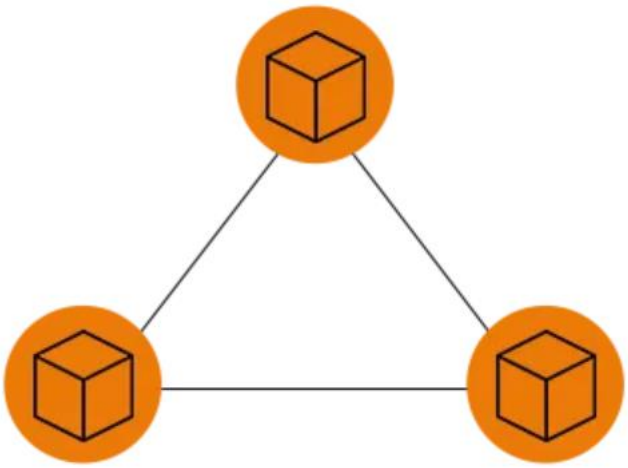
SAN'ın sunduğu avantajlar arasında yüksek performans ve ölçeklenebilirlik, merkezi yönetim ve felaket kurtarma yetenekleri bulunmaktadır. Ancak, kurulum ve bakımın karmaşıklığı ile önemli maliyet hususları SAN'ın dezavantajları arasında yer almaktadır (Singh, 2024).

### 2.3.5. Blok Depolama

Blok depolama, en eski depolama biçimlerinden biridir. Bu yöntemde veriler, Şekil 1'deki gibi sabit boyutlu parçalar veya bloklar hâlinde depolanmaktadır. Blok depolama sistemlerinde, veriler bağımsız veri birimleri olarak saklanır ve her blok kendine ait bir adrese sahiptir. Blok depolama sistemi, dosyaları tanımlamak için bu adresleri kullanır (SNIA, 2018a).

**Şekil 1**

*Blok Depolamada Veri Yapısı* File storage, block storage, or object storage. Red Hat, 2018.



Blok depolama, genellikle yüksek bant genişliği ve düşük gecikme süresi ile blok düzeyinde depolama cihazlarına erişim sağlayan bir arayüz sunar. Verilere, blok düzeyinde FCoE ve iSCSI gibi yüksek performanslı protokoller aracılığıyla erişilmektedir (Perry, 2020). Her depolama cihazı, bir blok düzeyinde depolama sisteminde bireysel bir sabit disk olarak kontrol edilebilmektedir. Bloklar, ana işletim sistemi tarafından yönetilmektedir. SAN, Hiper-Bütünleşik Depolama ve Bulut Blok Depolama Hizmeti, blok depolama türleri arasında yer almaktadır (Chandramouli ve Pinhas, 2020).

Blok depolamanın avantajları arasında yüksek performans, esneklik ve ölçeklenebilirlik, dosya değişikliğinin kolaylığı ve işletim sistemi önyüklenebilirliği bulunmaktadır. Bununla birlikte, blok depolamanın dezavantajları arasında meta veri sınırlamaları, sunucu bağımlılığı ve maliyetler yer almaktadır (Jyoti vd., 2021).

Blok depolama, en hızlı depolama biçimi olarak yüksek performans ve düşük gecikme süreleri gerektiren uygulamalarda yaygın olarak kullanılmaktadır. Sanal makineler ve veri tabanları, blok depolamanın tipik kullanım senaryoları arasında yer almaktadır (Domonech, 2018).

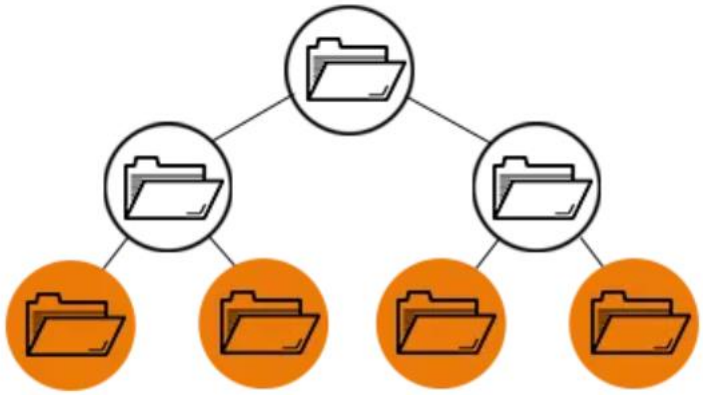
### 2.3.6. Dosya Depolama

Dosya depolama, dosya tabanlı depolama olarak da bilinmektedir. Bu yöntemde veriler, Şekil 2'deki gibi dosyalar ve klasörler hiyerarşisi şeklinde düzenlenmektedir.

Dosya depolama, geleneksel bir ağ sistemi gibi çalışır; yapılandırması genellikle kolaydır ancak verilere erişim yalnızca belirli yollar üzerinden gerçekleştirilebilmektedir. Dosya tabanlı depolama, ortak dosya düzeyinde protokoller kullandığı için farklı sistemler arasında kullanımı sınırlı olabilir (Paul, 2021). Veriler, genellikle TCP/IP protokolü üzerinden taşınır; Linux ortamında NFS protokolü, Windows ortamında ise Sunucu Mesaj Bloğu (SMB) protokolü tercih edilmektedir (Chandramouli ve Pinhas, 2020). Dosyalar, kullanıcı uygulamalarına güvenli veri paylaşımı sağlayan yüksek seviyeli bir depolama soyutlaması sunmaktadır (Mesnier vd., 2003).

**Şekil 2**

*Dosya Depolamada Veri Yapısı* File storage, block storage, or object storage. Red Hat, 2018.



Küresel bir dosya sistemi tarafından desteklenen dosya tabanlı depolama, verileri depolamak için dizinler ve alt dizinler kullanmaktadır. Dosya ve dizinler üzerinde kullanıcı ve grup izinleri ile güvenlik sağlanır; kullanıcılar ve gruplar, dosyalara erişim haklarına göre düzenlenmektedir (Samundiswary ve Dongre, 2017). Dosya sistemi; dizin konumu, erişim tarihi, dosya türü, dosya boyutu, oluşturma ve değiştirme gibi meta verileri yönetmekle sorumludur. Dosya tabanlı depolama genellikle yapılandırılmış verilerin yönetimi için en uygun çözümü sunmaktadır. Ancak, veri miktarının artmasıyla birlikte artan kaynak talepleri ve yapısal sorunlar, dosya tabanlı depolamanın etkinliğini zorlaştırabilmektedir (Paul, 2021).

Paylaşımlı dosya depolama genellikle NAS olarak adlandırılmaktadır. NAS, birden fazla makinede dosya paylaşımını sağlar ve yüksek hızlı flaş disklerin kullanımı ile performansı artırmaktadır. Büyük NAS çözümlerinde, meta veri erişim gecikmeleri, meta

veri tarama ve arama sürelerinin artması performans sorunlarına yol açabilmektedir. NAS sistemlerinde bağlantılar durumsaldır ve düzenli isteklerin gönderilmesi gerekebilir. İsteklerin hedefe ulaşamaması durumunda, bağlantılar geçersiz hâle gelebilmektedir (Samundiswary ve Dongre, 2017).

Dosya tabanlı depolama; basit depolama sistemi, dosya paylaşımı, güvenlik ve tanıdık protokollerin kullanımı gibi avantajlar sunmaktadır. Ancak, veri yedekliliği ve dosya arama gibi dezavantajları da vardır (Jyoti vd., 2021).

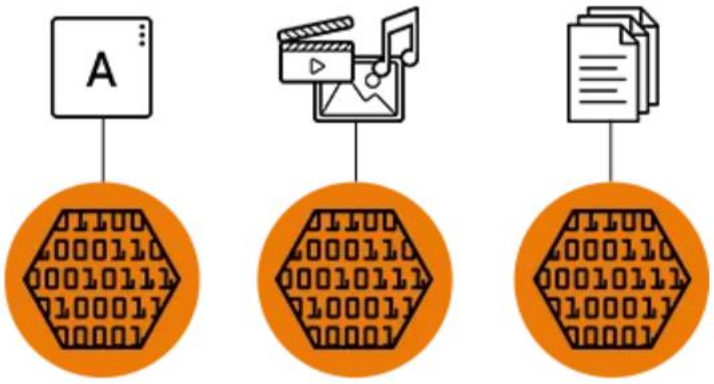
Yerel arşivleme ve dosya paylaşımı, dosya tabanlı depolamanın örnek kullanım senaryoları arasında yer almaktadır. NAS, ölçeklenebilir NAS (Scale-out NAS) ve Bulut Dosya Depolama Hizmeti, dosya tabanlı depolama türleri arasında bulunmaktadır (Domenech, 2018).

### 2.3.7. Nesne Depolama

Nesne depolama, veri kümelerini tekil, benzersiz bir şekilde tanımlanabilen ve bölünemez öğeler veya nesneler koleksiyonları olarak depolama ve daha sonra geri alma biçimidir. Bu sistem, Şekil 3'te gösterildiği üzere her türlü veri biçiminin bir nesne olarak yapılandırılabilmesini sağlar. Nesneler, atomik bir birim olarak değerlendirilir. Bu nedenle, bir dosya sisteminde yer alan dizin hiyerarşisine karşılık gelen bir yapı bulunmaz; her nesne, sistemde benzersiz bir nesne tanımlayıcısı ile tanımlanmaktadır (SNIA, 2024c).

**Şekil 3**

*Nesne Depolamada Veri Yapısı* File storage, block storage, or object storage. Red Hat, 2018.



Nesne depolama sistemlerinde bir nesne oluşturulduğunda, tüm veri kümesi, alt parçalara bakılmaksızın bütün olarak işlenmektedir. Okuma işlemi sırasında, kullanıcı tüm nesneyi okuyabilir veya belirli kısımlarını seçebilir. Ancak genellikle, nesne veya nesnenin bazı bölümleri üzerinde doğrudan güncelleme yapmak mümkün değildir; genellikle tüm nesnenin yeniden yazılması gerekmektedir. Çoğu nesne depolama sistemi, nesnelerin silinmesine izin vermektedir (SNIA, 2024c).

Nesne depolama, meta verileri desteklemektedir. Meta veriler, nesnenin kendisine ek olan, nesne kimliğine ve verilere dair ek bilgilerdir ve genellikle öznitelik-değer çiftleri şeklinde ifade edilir. Meta veriler, nesne kimliklerine göre ayrı ayrı adreslenebilen nesne koleksiyonlarının, belirli kimlikler hakkında bilgi sahibi olmaksızın gruplar hâlinde aranmasına, filtrelenmesine ve okunmasına olanak tanımaktadır (SNIA, 2024c).

Depolama sistemi, nesnelerin içeriğine bağlı kalmadan çalışır; nesneler basit veri kümeleri, dosyalar, tam dosya sistemleri, videolar, sanal makineler veya veri tabanları gibi çeşitli biçimlerde olabilmektedir. Depolama sistemi, nesneyi yalnızca bir veri parçası ile ilişkilendirilmiş bir nesne kimliği olarak görür ve yönetir (SNIA, 2024c).

Nesnelerin nasıl depolandığı, uygulama açısından önemli değildir. Birçok sistem dosya veya blok tabanlı depolama kullanırken, bazı sistemler nesneleri doğrudan destekleyen özel cihazlarla tasarlanmıştır. Her durumda, nesne depolama sistemleri, nesneleri oluşturmak, okumak ve silmek ve meta verileri aramak için basit Uygulama Programlama Arabirimi (API) sağlamaktadır (SNIA, 2024c).

Nesne depolama sistemleri, kapasite, işleme ve ağ kaynaklarını yatay olarak ölçeklendirebilen “scale-out” sistemlerdir. Yüksek ölçeklenebilirliği sayesinde yüksek hacimli veri içeren kullanım durumlarıyla başa çıkabilmektedir (Yiğit, 2016).

Nesne tabanlı depolama, dosyaların ve blokların avantajlarını bir arada sunmaktadır. Bloklar gibi, nesneler de doğrudan depolama cihazına erişim sağlayabilir, bu da performans avantajları sunar. Aynı zamanda, dosyalar gibi, nesneler de bir arayüz aracılığıyla erişilir ve bu arayüz, nesneyi depolamak için gerekli olan meta verileri depolama uygulamalarından soyutlar. Bu özellikler, nesne tabanlı depolamanın performans ve kullanım kolaylığı açısından önemli avantajlar sunduğunu göstermektedir (Mesnier vd., 2003).

### 2.3.8. İçerik Adresli Depolama (CAS)

İçerik Adresli Depolama (CAS), belgelerin içerik özetlerini depolamak amacıyla geliştirilmiş bir nesne tabanlı depolama teknolojisidir. CAS, kullanıcıların içeriklere, belgelerin gerçek verilerinin yerini veya kopya sayısını bilmeden erişimini sağlar. Bu teknoloji, verileri kriptografik özet (hash) fonksiyonları aracılığıyla özetleyerek saklar ve belgelerin kısa ve orta vadeli saklama ihtiyaçlarını karşılar (Chandramouli ve Pinhas, 2020). CAS, sabit içeriklerin güvenli bir şekilde çevrimiçi saklanması ve erişilmesi için optimize edilmiştir; ayrıca veri çoğaltmasını ortadan kaldırarak maliyetleri düşürür (Somasundaram vd., 2009).

### 2.3.9. Yüksek Düzeyli Veri Erişim Hizmetleri

Temel depolama biçimlerinden (blok, dosya, nesne) daha yüksek bir soyutlama düzeyinde veri sağlayan veri erişim hizmetleri bulunmaktadır. Bu hizmetler, yalnızca aynı soyutlama düzeyinde veri almak için özel olarak yapılandırılmış istemciler aracılığıyla erişilebilir. Yüksek düzeyli veri erişim hizmetleri, hem kurumsal veri merkezlerinde hem de bulut ortamlarında uygulanabilmektedir. Bazı örnekleri aşağıda verilmiştir (Chandramouli ve Pinhas, 2020):

- NoSQL Veritabanı Hizmetleri**:** Yapılandırılmamış verilerin depolanmasını sağlar. NoSQL veritabanı hizmetleri arasında anahtar-değer deposu, çok modelli veritabanı ve grafik veritabanı yer almaktadır.
- SQL Veritabanı Hizmetleri**:** Genellikle tablo biçiminde (ilişkisel tablolar olarak da adlandırılır) yapılandırılmış verilerin depolanmasında kullanılır. Ayrıca, XML, JSON ve İkili Formatta Büyük Nesneler (BLOB) gibi veri yapıları da desteklenmektedir.
- Mesajlaşma Kuyruğu Depolama Hizmetleri**:** Mesajlaşma kuyrukları altyapısında veri depolamak amacıyla özel olarak tasarlanmıştır. Bu altyapılar, bileşenlerin mesajlaşma kuyruk sistemine abone olarak asenkron iletişim kurduğu dağıtılmış uygulamalar tarafından kullanılmaktadır.

### 2.3.10. Yazılım Tanımlı Depolama (SDS)

Yazılım Tanımlı Depolama (SDS), veri depolama yönetimini fiziksel donanımdan bağımsız hâle getirerek yazılım aracılığıyla gerçekleştiren bir teknolojidir. Bu yaklaşım, depolama kaynaklarının soyutlanmasını sağlayarak esneklik, verimlilik ve hızlı ölçeklenebilirlik kazandırmaktadır. SDS, çeşitli donanımların birlikte çalışmasını kolaylaştırarak otomatik kaynak dağıtımı ve yönetim süreçlerini hızlandırmaktadır (Chandramouli ve Pinhas, 2020).

SDS'nin temel özellikleri arasında otomasyon, programlanabilirlik ve standardize edilmiş arayüzler bulunmaktadır. Bu özellikler, depolama kaynaklarının etkin bir şekilde yönetilmesini ve hizmet seviyelerinin otomatik olarak sağlanmasını mümkün kılmaktadır. Ayrıca, meta veri kullanımıyla hizmet seviyelerinin belirlenmesi ve yönetilmesi, depolama yönetimini daha şeffaf ve ölçeklenebilir hâle getirmektedir (Carlson vd., 2015).

SDS'nin avantajları arasında donanım satıcılarına bağımlılığı azaltması, programlanabilirlik ve otomasyon ile hızlı değişiklikler yapılabilmesi ve verimliliği artırması yer almaktadır. SDS'nin çeşitli türleri mevcuttur; bunlar arasında hipervizör tabanlı, konteyner tabanlı ve dağıtılmış dosya sistemleri gibi çeşitler bulunmaktadır (NetApp, 2024).

SDS, veri merkezi modernizasyonu ve hibrit bulut uygulamaları gibi birçok kullanım senaryosunda uygulanabilmektedir. Bu teknoloji, depolama kaynaklarını tek bir mantıksal birim olarak yönetmeye olanak tanır ve bu sayede veri yönetimini daha basit ve etkili hâle getirmektedir (Vmware, t.y.).

### 2.3.11. Depolama Sanallaştırması

Depolama sanallaştırması, bir depolama sisteminin iç işlevlerini uygulamalardan, bilgi işlem sunucularından veya genel ağ kaynaklarından soyutlamak amacıyla kullanılan bir tekniktir. Bu teknoloji, depolama cihazlarının veya dizilerinin kapasitesini soyutlayarak tek bir mantıksal varlık olarak yönetmeyi sağlamaktadır. Sanallaştırma, geniş ağlarda ve veri merkezlerinde fiziksel depolama cihazları arasında mantıksal kaynakları toplamak ve yönetmek için kullanılmaktadır. Bu yaklaşım, fiziksel depolama kaynaklarının detaylarını

gizleyerek, depolamanın uygulama ve ağdan bağımsız bir şekilde yönetilmesini mümkün kılar (Chandramouli ve Pinhas, 2020).

Sanallaştırmanın faydaları arasında ölçeklenebilirlik, performans iyileştirmeleri, yedeklilik ve artan depolama kaynak kullanım oranı bulunmaktadır. Ayrıca, sanallaştırma, birden fazla fiziksel disk sürücüsünü tek bir mantıksal hacim olarak sunabilme, erişim desenlerindeki değişikliklere yanıt verebilme ve büyük ölçekli veri göçleri gerçekleştirme gibi kullanım senaryolarını desteklemektedir (Chandramouli ve Pinhas, 2020).

Depolama sanallaştırmasının uygulanması, fiziksel depolama kaynaklarının mantıksal bir görünümünü sunarak, fiziksel depolamanın sunucuya doğrudan bağlıymış gibi görünmesini sağlar. Bu süreç, depolamanın eklenmesi veya silinmesi işlemlerini uygulama kullanılabilirliğini etkilemeden yapabilmeyi ve kesintisiz veri göçlerini mümkün kılmaktadır (Somasundaram vd., 2009).

SNIA tarafından geliştirilen taksonomi, depolama sanallaştırmasını sistematik olarak sınıflandırır ve sanallaştırmanın ne, nerede ve nasıl yapılacağını tanımlamaktadır. Bu taksonomi, sanallaştırma türlerini, seviyelerini ve metodolojilerini detaylandırır, böylece sanallaştırma stratejilerinin etkin bir şekilde uygulanmasını sağlamaktadır (Bunn vd., 2004).

Depolama sanallaştırmasının avantajları, donanım bağımsızlığı ve esneklik sağlayarak, farklı ağlardan ve veri merkezlerinden gelen depolama donanımlarını tek bir mantıksal görünümde birleştirmeyi içermektedir. Sanallaştırma, depolama yönetimini daha esnek hâle getirir ve bu süreçte performans, ölçeklenebilirlik ve yönetim zorlukları gibi temel sorunlara çözüm sunmaktadır. Ayrıca, depolama sanallaştırması, yüksek maliyetler, performans sorunları ve yönetim zorlukları gibi engelleri aşarak, büyük veri havuzlarının yönetimini kolaylaştırmaktadır (Datacore, 2024).

### 2.3.12. Sanal Sunucular ve Konteynerler İçin Depolama

Sanal sunucular (VM) ve konteynerler, modern bilişim altyapılarında yaygın olarak kullanılan iki farklı sanallaştırma teknolojisidir. Bu teknolojiler, fiziksel kaynakları sanallaştırarak verimlilik ve esneklik sağlar, ancak farklı işlevler ve avantajlar sunmaktadır (Chandramouli ve Pinhas, 2020).

Sanal sunucular, bir fiziksel sunucunun üzerinde birden fazla işletim sisteminin ve uygulamanın aynı anda çalışmasını sağlayan teknolojilerdir. Her sanal sunucu, fiziksel

sunucunun kaynaklarını izole bir şekilde paylaşır ve kendi işletim sistemi ile bağımsız olarak çalışmaktadır. Sanal sunucular, fiziksel donanım kapasitelerinden tam olarak yararlanılmasını sağlar, ancak kaynak paylaşımı nedeniyle yüksek erişilebilirlik ve performans gereksinimlerini karşılamada sınırlamalar yaşanabilmektedir (Somasundaram vd., 2009). Sanal makineler, genellikle test etme, geliştirme, yalıtım, bulut bilişim ve felaket kurtarma gibi çeşitli kullanım senaryolarında tercih edilir (Google Cloud, t.y.).

Konteynerler, uygulamaları ve bağımlılıklarını içeren hafif, taşınabilir ve kendi kendine yeten yazılım paketleridir. Bu teknoloji, uygulamaların çeşitli ortamlarda tutarlı bir şekilde çalışmasını sağlar ve mikro hizmet mimarileri ile bulut bilişimde popülerdir. Konteynerler, sanal makinelerden daha hafif ve taşınabilir olup, uygulama dağıtımında hızlılık ve verimlilik sunmaktadır. Konteynerler, genellikle Kubernetes gibi düzenleme platformları kullanılarak yönetilir ve ölçeklenebilirlik, sürekli entegrasyon ve sürekli teslimat gibi senaryolarda avantaj sağlamaktadır (Google Cloud, t.y.).

Konteynerler, ana işletim sisteminin çekirdeğini paylaşırken, sanal makineler kendi çekirdeklerine sahiptir. Bu, konteynerlerin daha hızlı başlatılması ve daha az kaynak kullanılması anlamına gelmektedir. Konteyner görüntüleri genellikle megabayt biriminde ölçülen daha küçük dosyalardır, bu da taşınabilirliklerini artırmaktadır. Sanal makine görüntüleri ise gigabayt biriminde olup daha büyüktür ve daha fazla kaynak tüketmektedir. Konteynerler, uygulama dağıtımında daha hızlı ve verimli bir yaklaşım sunarken, sanal makineler daha kapsamlı yalıtım ve güvenlik gereksinimleri için tercih edilmektedir (AWS, 2023).

Sanal sunucular için özel olarak tasarlanmış depolama altyapısı, VM-bilincine sahip depolama olarak adlandırılmaktadır. Bu altyapı, hipervizör aracılığıyla VM düzeyinde depolama kaynaklarının yönetimini sağlar ve uygulama veri Erişim Hizmet Kalitesini (QoS) iyileştirmektedir. Konteynerler ise depolama ihtiyaçlarını ana bilgisayara yerel dizinler oluşturarak veya harici SAN/NAS cihazlarına bağlanarak karşılamaktadır (Chandramouli ve Pinhas, 2020).

Sanal makineler ve konteynerler, farklı ihtiyaçlar ve senaryolar için uygun teknolojilerdir. Sanal makineler geniş kapsamlı yalıtım ve çoklu işletim sistemi desteklerken, konteynerler daha hafif, taşınabilir ve hızlı dağıtım çözümleri sunmaktadır. Her iki teknoloji de modern bilgi teknolojileri altyapısının önemli bileşenleridir ve genellikle birlikte kullanılarak ikisinin de avantajlarından yararlanılmaktadır (AWS, 2023).

### 2.3.13. Bütünleşik ve Hiper-Bütünleşik Depolama

Bütünleşik bir mimaride, depolama, bellek, ağ ve sanallaştırma yazılımları, hızlı dağıtım için tek bir kutuda (örneğin, bir sunucu rafında bir veya daha fazla fiziksel ana bilgisayar, depolama kaynakları (DAS veya depolama dizileri) ve ağ bileşenleri içeren bir yapı) önceden yapılandırılmış ve kurulmuş olarak sunulmaktadır (Azeem ve Sharma, 2017). Hiper-bütünleşik bir mimari ise soyutlama seviyesini bir adım daha ileriye taşımaktadır. Bu mimaride, fiziksel ana bilgisayarlarla ilişkili bireysel depolama bileşenleri sanallaştırılır ve tüm sanal makineler veya konteynerler arasında paylaşılan ortak bir depolama havuzu oluşturulmaktadır. Böylece, bir fiziksel ana bilgisayarda barındırılan bir sanal makine veya konteynerin, farklı bir fiziksel ana bilgisayara bağlı depolamayı kullanabilmektedir. Bu yetenekler, uzak disk erişimi için bir depolama soyutlama katmanı sunmaktadır (Chandramouli ve Pinhas, 2020).

Hiper-Bütünleşik Altyapıda (HCI), bilgi işlem, ağ ve depolama için gereken donanım sıkı bir şekilde birleştirilir. Tüm temel depolama yönetim işlevleri, HCI tedarikçisinin yönetim yazılımı katmanı ve/veya donanımı aracılığıyla bilgi işlem sağlama ile birlikte sunulmaktadır. Donanımın sıkı entegrasyonu, HCI tedarikçilerinin depolama cihazı üreticileri ile birlikte çalışarak yazılım yığınına uyacak şekilde özelleştirilmiş bir depolama çözümü oluşturmasını sağlamaktadır. HCI çözümlerinden beklenen özellikler şunları içermektedir (Chandramouli ve Pinhas, 2020):

- Opsiyonel veri azaltma özellikleri
- Merkezi bir panel aracılığıyla yönetim kontrolü
- Uygulama ihtiyaçlarına dayalı QoS depolama gereksinimleri

HCI sistemde, bilgi işlem için kullanılan bazı işlemci kaynaklarının depolama erişimi ve yönetim işlevlerini yerine getirmek için paylaşılması gerekebilir. Genel yönetim yazılımı yığını, dağıtım ortamına bağlı olarak (örneğin, sanallaştırılmış altyapı, yapılandırılmamış veri depoları, yüksek performanslı bilgi işlem) bir bilgi işlem düğümü, bir hipervizör ve SDS yazılımını içerebilmektedir (Chandramouli ve Pinhas, 2020).

### 2.3.14. Bulut Depolama

Bulut depolama, bulut bilişimin bir parçası olarak veri saklama ve yönetiminde önemli bir yenilik sunmaktadır (Surianarayanan ve Chelliah, 2023). Bu sistem, çeşitli türdeki depolama cihazlarını bir araya getirir ve bu cihazları ağlar üzerinden entegre ederek veri erişim hizmetleri sunar (Huawei Technologies Co., 2023). Bulut depolama, genellikle dört ana katmandan oluşmaktadır (Surianarayanan ve Chelliah, 2023):

- Depolama katmanı: Depolama katmanı, çeşitli fiziksel depolama cihazlarını içerir ve bu cihazlar genellikle farklı coğrafi konumlarda bulunur.
- Temel yönetim katmanı: Temel yönetim katmanı, depolama cihazlarının sanallaştırma, yedeklilik ve güvenlik gibi yönetim işlevlerini sağlar.
- Uygulama arayüzü katmanı: Uygulama arayüzü katmanı, veri erişim türleri için çeşitli API'ler sunar, bu da farklı uygulama ihtiyaçlarına göre özelleştirilmiş veri erişimi sağlar.
- Erişim katmanı: Erişim katmanı, kullanıcıların bulut hizmetlerine erişimini ve çeşitli erişim yöntemlerini yönetir.

Bulut depolama sistemleri, işbirliği, veri entegrasyonu, analitik yetenekler ve gelişmiş veri koruma hizmetleri sunmaktadır. İşbirliği yeteneği, dosya paylaşımı ve sürümleme gibi özellikleri içerirken, veri entegrasyonu ve analitik yetenekleri, verilerin entegre edilmesi ve analiz edilmesini sağlar. Ayrıca, gelişmiş veri koruma hizmetleri, verilerin çoğaltılması, yansıtılması ve şifrelenmesi gibi güvenlik önlemleri sunar (Chandramouli ve Pinhas, 2020).

Bulut depolama çözümleri; özel, genel ve hibrit bulut seçenekleri sunarak esneklik sağlamaktadır. Kurumsal kullanıcılar için bu sistemler, veri erişimini kolaylaştırırken maliyetleri düşürür ve güvenliği artırır. Küçük işletmeler ve bireysel kullanıcılar için ise bulut depolama, depolama alanını basitleştirir ve erişilebilirliği artırmaktadır. Bulut depolama, veri yönetimini modernleştiren, ölçeklenebilir ve güvenilir bir çözüm olarak öne çıkmaktadır (SNIA, 2024d).

## 2.4. Depolama ve Veri Yönetimi

Depolama yönetimi, depolama kaynaklarının dayanıklılığını, performansını ve güvenliğini sağlamak amacıyla yönetim araçları ve süreçlerinin kullanılmasıyla ilgili tüm

faaliyetleri ifade etmektedir. Bu faaliyetler arasında depolama kaynaklarının yapılandırılması ve kaynak yönetimi, veri sınıflandırılması ve kategorizasyonu, veri temizleme, veri saklama, veri koruma ve veri azaltma bulunmaktadır (Chandramouli ve Pinhas, 2020).

### 2.4.1. Depolama Kaynaklarının Yapılandırılması ve Kaynak Yönetimi

Depolama kaynaklarının yapılandırılması ve yönetimi, depolama altyapısının tüm yaşam döngüsü boyunca yönetimini kapsamaktadır. Depolama altyapısının yaşam döngüsü boyunca gerçekleştirilen işlemler aşağıda sıralanmıştır (Chandramouli ve Pinhas, 2020):

- Fiziksel depolama cihazlarının yönetimi ve kontrolü (depolama dizileri, SAN anahtarları, yazılım güncellemeleri, cihaz yapılandırılması, cihaz eklenmesi/çıkarılması işlemleri gibi)
- Birden fazla varlık arasında değişiklik yönetimi
- Depolama kaynaklarının yönetimi (blok cihazı, dosya sistemleri, depolama havuzu gibi) ve bunların ana bilgisayara atanması, çoğaltılması, veri taşınması ve veri katmanlama
- Performans yönetimi ve optimizasyonu
- Kapasite yönetimi ve optimizasyonu
- Envanter yönetimi
- Olay yönetimi

### 2.4.2. Veri Sınıflandırılması ve Kategorizasyonu

Veri sınıflandırması, kuruluşların veri varlıklarını hassasiyet, önem ve düzenleyici gerekliliklere göre sistematik bir şekilde düzenlemelerine yardımcı olur. Bu süreç, çeşitli veri türlerini belirli kategorilere ayırarak, uygun güvenlik önlemlerinin uygulanmasını sağlamaktadır. Verilerin sınıflandırılması, hassas ve hassas olmayan veriler arasında ayrım yaparak güvenlik kontrollerinin etkinliğini artırır (SNIA, 2018b).

Kurumsal veriler, çeşitli boyutlara göre sınıflandırılabilir. SNIA Veri Koruma En İyi Uygulamalar Beyaz Kitabı, veri sınıflandırmasını basit bir şemayla özetlemektedir. Bu

şema, üretim ve üretim dışı ortamlar ile hassas ve hassas olmayan verileri dikkate alır ve veri hassasiyetine göre önceliklendirme yapmaktadır. Bu yaklaşım, verilerin düzenleyici standartlara uyumunu destekler ve güvenlik kontrollerinin belirlenmesinde başlangıç noktası sağlamaktadır (SNIA, 2018b).

Veri sınıflandırması, Kişisel Olarak Tanımlanabilir Bilgi (PII), Korunan Sağlık Bilgisi (PHI) ve finansal veriler gibi hassas bilgileri tanımlamak ve korumak için kritik bir adımdır. Bu sınıflandırma, Genel Veri Koruma Yönetmeliği (GDPR), Sağlık Sigortası Taşınabilirlik ve Sorumluluk Yasası (HIPAA) ve Ödeme Kartı Endüstrisi Dijital Güvenlik Standardı (PCI DSS) gibi düzenleyici standartlara uyum sağlamada temel bir rol oynamaktadır. Sınıflandırma, veri türlerine uygun güvenlik önlemlerinin belirlenmesine yardımcı olur ve veri koruma stratejilerinin etkinliğini artırmaktadır (Paloalto, 2024).

Veri sınıflandırma süreci, verilerin türüne bağlı olarak uygun güvenlik önlemlerini belirler. Bu önlemler arasında şifreleme, erişim kontrolü, veri yedekleme ve uyumluluk bulunmaktadır. Veri sınıflandırması, risk değerlendirmesi, erişim kontrolleri ve şifreleme stratejilerinin optimize edilmesini sağlar ve bu sayede, veri güvenliğini büyük ölçüde güçlendirmektedir (Paloalto, 2024).

### 2.4.3. Veri Temizleme

Veri temizleme, daha önce yazılmış verilerin geri alınamaz hâle getirilmesi sürecidir; böylece verilerin erişilemeyeceği veya yeniden yapılandırılamayacağı konusunda mâkul bir güvence sağlanmaktadır. Temizleme için üç yöntem bulunmaktadır (Chandramouli ve Pinhas, 2020):

- Temizleme (Clear): Mevcut verilerin üzerine yazma
- Silme (Purge): Manyetik ortamlar için güçlü bir manyetik alan kullanma, şifrelenmiş veriler için kriptografik silme
- Yok etme (Destruct): Ortamın fiziksel olarak yok edilmesi (yakma, öğütme gibi)

Uygun temizleme türünü belirleyen faktörler; ortamda bulunan verinin kategorisi, medyanın doğası (katı hâl, manyetik veya optik) ve medyanın yeniden kullanım planlarıdır. Temizleme, bireysel depolama medyalarına veya mantıksal verilere (örneğin, bulut) uygulanabilmektedir. Temizlemenin amacı gerçekleştirilmek isteniyorsa, medya verisinin türü ve özellikleri göz önünde bulundurulmalıdır. Örneğin, manyetik disklerde

veri üzerine yazma etkili olabilirken, flaş bellek bulunan katı hâl sürücülerde cihazlar verileri yerinde üzerine yazmadığından veri üzerine yazma etkili değildir.

Veri temizleme, hassas bilgilerin korunması amacıyla yapılmaktadır. Veri temizlemenin faydaları arasında veri güvenliğinin artırılması, düzenleyici uyumluluğun artırılması, veri ihlâllerinin önlenmesi ve çevresel sürdürülebilirliğin teşvik edilmesi yer almaktadır. Ayrıca, veri kaynaklarının optimize edilmesine de katkı sağlamaktadır (Brook, 2024).

### 2.4.4. Veri Saklama

Veri saklama, genellikle verilerin bazı yedekleme ortamlarında bir kopyasının tutulmasıyla sağlanmaktadır. Belirli verilere kısa, orta (yani 10 yıldan az) veya uzun vadeli erişimin korunması gereken durumlar olabilmektedir. Bu gereklilikler, operasyonel, yasal, düzenleyici veya gerekçelerden kaynaklanabilmektedir (SNIA, 2018b).

### 2.4.5. Veri Koruma

Veri koruma; verilerin erişilebilir, kullanılabilir, bozulmamış ve yetkilendirilmiş tüm amaçlar için kabul edilebilir bir performans seviyesinde mevcut olmasını sağlayan tüm faaliyetleri kapsamaktadır. Ayrıca, veri koruma, tesadüfi veya yetkisiz ifşaya, değiştirmeye veya yok etmeye karşı güvence sağlamak için fiziksel, idari ve teknik yöntemleri içerir ve uyum gerekliliklerıyle uyumlu olarak ele alınmaktadır (SNIA, t.y.).

Veri koruma, tüm depolama faaliyetleri ve mekanizmalarını içermektedir. Bu aşamalar şunlardır (Chandramouli ve Pinhas, 2020):

- Veri dinlenmede/son noktada: Bir sunucu veya istemci cihazda
- Veri aktarımda: Depolama cihazları arasında, istemci ile sunucu arasında veya sunucu ile sunucu arasında
- Veri kullanımda: Görüntüleme, değiştirme veya cihazlar arasında senkronize etme sırasında
- Güvenlik çevresinin dışına çıkan veri: İndirmeler, fiziksel medya gönderimi vb. sırasında

Veri koruma faaliyetlerinin kapsamı ve ilişkili etkinlikler, veri koruma faaliyetlerini depolama, gizlilik ve bilgi güvencesi/güvenliği olmak üzere üç açıdan sınıflandırmak için bir taksonomi sağlar. Gizlilikle ilgili yasalar ve düzenlemeler ülkeler ve topluluklar arasında farklılık göstermektedir. Bilgi güvencesi/güvenliği ile ilgili faaliyetler çoğunlukla teknik kontrollerden oluşur ve her birinin detaylarını tartışmak için özel bir bölümü vardır. Bu başlıkta yalnızca depolama ile ilgili veri koruma faaliyetleri ele alınmaktadır. Bu kontrol kategorisi şunları içermektedir (Chandramouli ve Pinhas, 2020) :

- Veri yedekleme ve kurtarma: Yedekleme, depolama cihazında saklanan verilerin üretim sistemleri tarafından erişilebilmesi ve belirli aralıklarla başka bir set depolama cihazına kopyalanmasını içerir. Veri içeriğinin değişken doğası nedeniyle daha önce alınan bir yedekleme, daha sonra alınan bir yedekleme ile geçersiz hâle gelebilir. Yedeklemeler, "dosya yedeklemeleri" (genellikle dosyalar, dizinler, bir veritabanı şeması altındaki veriler gibi mantıksal veri yapıları temelinde seçilmiş bir veri kısmını yedekler) veya "görüntü yedeklemeleri" (örneğin, bireysel bir LUN) olabilir (Chandramouli ve Pinhas, 2020).
- Arşivleme: Arşivleme, verilerin uzun süreli saklanmasını içerir (Nelson, 2011). Yedekleme ve arşivleme için ayrı teknikler ve çözümler kullanılabilir, ancak arşivleme genellikle yedekleme kopyalarının daha uzun süreli saklanmasını ve ek sınıflandırmayı içerdiği için ikisi arasında yakın bir ilişki vardır. Arşivleme; bilgi yaşam döngüsü yönetimini, kayıt arama ve veri keşfini, veri koruma, saklama ve yasal koruma için düzenleyici gereksinimlerini karşılanmasını destekler.
- Veri çoğaltma: Veri çoğaltma, aynı veriyi en az iki ayrı konumda yazma sürecidir (Pandey, 2023). Çoğaltma genellikle veri kurtarma sürecinin bir parçası olarak kullanılır. Veriyi bir yerden diğerine kopyalamayı içerir. Senkron ve asenkron olmak üzere iki tür çoğaltma bulunmaktadır. Senkron çoğaltma, veriyi A noktasından (örneğin, bir üretim platformu) B noktasına (örneğin, özel olarak belirlenmiş bir felaket kurtarma sitesi) gerçek zamanlı olarak kopyalar. Asenkron çoğaltma, zaman gecikmesi içerir ve veriyi A noktasından B noktasına yazmak için belirli bir sıklıkta veya sürekli olarak yapılabilir. Zaman gecikmesi ve sıklık, işletmenin felaket kurtarma politikası tarafından belirlenir ve belirli Kurtarma Süresi Hedefi (RTO) ve Kurtarma Noktası Hedefi (RPO) ile tanımlanır (Chandramouli ve Pinhas, 2020).

- Değişmezlik: Değişmezlik, verinin oluşturulduktan sonra kilitlenme yeteneğini içerir. Böylece, verinin değiştirilmesi veya silinmesi engellenir (Chandramouli ve Pinhas, 2020).
- Sürekli veri koruma: Sürekli Veri Koruma (CDP), ince ayrıntılı kurtarma ve geliştirilmiş RPO destekleyen bir yedekleme türüdür. Geleneksel yedeklemenin aksine, CDP'de değişen bloklar sürekli olarak hedef depolama ortamında iletilir ve bu değişiklikler zamanla yakalanır ve günlüklenir. CDP, bu açıdan çoğaltmaya benzese de genellikle kopyalanmış verilerin geçmiş noktalara oynatılmasını sağlar (örneğin, bayt bayt önceden işaretlenmiş yerler, önceki sürümler). Ek teknolojiler, dosya ve nesne sürümleme veya günlükleme, veri tabanı günlük gönderimi gibi sürekli veri koruma biçimleri olarak değerlendirilebilir (Chandramouli ve Pinhas, 2020).
- Zaman noktasında kopyalar ve anlık görüntüler: Zaman noktasında kopyalar, genellikle kaynak veriyi saklayan ilgili unsur (depolama dizisi, dosya sistemi, veri tabanı gibi) tarafından oluşturulur. Hızlı kurtarma ve çeşitli diğer kullanım senaryoları (örneğin, üretim verilerini test amaçlı kopyalama) için tasarlanır. Zaman noktasında kopyalar, uzak çoğaltmalardan da alınabilir (bu nedenle "uzak, zaman noktasında kopyalar" olarak adlandırılabilir) (Chandramouli ve Pinhas, 2020). Anlık görüntü, zaman noktasında kopyaların depolama açısından verimli bir biçimidir ve yalnızca bir noktadaki veri ile kaynak verinin geri kalan kısmı arasındaki değişiklikleri depolar. Kaynak verinin kullanılmaz hâle gelmesi durumunda anlık görüntüler de kullanılamaz (SNIA, 2018b).

### 2.4.6. Veri Azaltma

Veri azaltma, maliyetleri azaltma ve verimliliği artırma amacıyla depolanan ve/veya iletilen veri miktarını azaltma sürecidir. Veri azaltmanın iki yaygın yaklaşımı tekilleştirme ve sıkıştırmadır.

- Veri tekilleştirme: Veri tekilleştirme, birden fazla kopyası bulunan verilerin yalnızca bir kopyasını tutarak depolanan veriyi azaltmayı amaçlar. Örneğin, bir depolama sistemi 500 adet aynı veriyi içeriyorsa depolama dizisi sadece bir kopya depolayacak ve 499 kopyayı saklamaktan kaçınacaktır. Tekilleştirme; depolama

cihazı düzeyinde, iletim aşamasında veya dosya sistemi düzeyinde gerçekleştirilebilir (Stanek, 2018).

- Veri sıkıştırma: Veri sıkıştırma, veriyi bilinen bir algoritma ile kodlayarak veri miktarını azaltmayı amaçlar. Veri sıkıştırma algoritması, bir girdi nesnesini sıkıştırılmış bir çıktı sembolü dizisine dönüştürür; bu diziden, eşleşen bir açma algoritmasıyla orijinal girdi geri kazanılabilir. Sıkıştırılmamış temsilinden daha az depolama alanı kullanılmasını sağlar. Veri sıkıştırma, birçok yerde kullanılabilir, ancak en yaygın olarak yedekleme bantlarında ve uzaktan veri çoğaltma sırasında ağ geçitlerinde kullanılır. Bu sayede, felaket kurtarma ve iş sürekliliği operasyonları için bant genişliği gereksinimlerini azaltır. Sıkıştırma ve ardından açma işlemi farklı varlıklar tarafından gerçekleştirilebilir; bu durumda birlikte çalışabilirlik önemli bir gereksinimdir (Steinruecken, 2014).

## 2.5. Depolama Güvenliği

Bilgi güvenliği alanında depolama güvenliği, veri ve bilgi varlıklarının korunması amacıyla uygulanan stratejileri ve yöntemleri ifade etmektedir. Depolama güvenliği, hem fiziksel hem de dijital ortamda veri koruma ihtiyaçlarını kapsamaktadır. Bu doğrultuda, Ulusal Standartlar ve Teknoloji Enstitüsü (NIST) tarafından yayınlanan Depolama Altyapısı İçin Güvenlik Kılavuzları'nda (2020) yer verilen depolama güvenliğinde karşılaşılabilecek bazı tehditler, riskler, saldırı yüzeyleri ve bunların karşısında alınabilecek önlemler ele alınmıştır (Chandramouli ve Pinhas, 2020).

### 2.5.1. Tehditler, Riskler ve Saldırı Yüzeyleri

Veri depolama sistemlerinin güvenliği, veri kaybı, yetkisiz erişim ve diğer siber tehditlere karşı korunma açısından büyük bir önem taşımaktadır. Bu başlıkta, veri depolama sistemlerinde karşılaşılabilecek başlıca tehditler, riskler ve saldırı yüzeyleri incelenmiştir.

#### 2.5.1.1. Tehditler

Tehdit, istenmeyen bir olayın potansiyel nedenidir ve bir sisteme veya kuruluşa zarar verebilmektedir. Depolama altyapısı ile ilgili bazı tehditler şunlardır (Chandramouli ve Pinhas, 2020):

- Kimlik bilgilerinin çalınması veya tehdit edilmesi: Kimlik bilgilerinin çalınması en çok karşılaşılan siber tehditlerden biridir (Ceballos, 2016). Kullanıcı kimlik bilgileri çeşitli araçlar ve tekniklerle çalınabilmektedir. En yaygın saldırılar şifrelerin tahmin edilmesi veya çalınması ile gerçekleştirilmektedir.
- Şifrelemenin kırılması: Şifreleme verileri korur, ancak zayıf şifreleme algoritmaları ve kötü yönetilen anahtarlar, şifrelemenin kırılmasına yol açabilmektedir. Anahtarların güçlü ve güvenli bir şekilde yönetilmesi önem arz etmektedir.
- Kötü amaçlı yazılım ve fidye yazılımlar: Kötü amaçlı yazılım, sistemlere zarar verebilir, verileri çalabilir veya sistemleri bozulmasına neden olabilmektedir. Fidye yazılımları, verileri şifreleyerek veya engelleyerek erişim için ödeme talep etmektedir (Lemmou vd., 2020).
- Arka kapılar ve yamalanmamış güvenlik açıkları: Arka kapılar, genellikle kasıtlı olarak yazılıma eklenen gizli erişim yollarıdır. Güvenlik açıkları, yazılımın eksik test edilmesi sonucu ortaya çıkar ve düzeltilmezse risk oluşturmaktadır.
- Ayrıcalık yükseltme: Güvenlik açıklıklarından yararlanarak, kullanıcıları veya uygulamaların daha yüksek erişim hakları elde etmesine yol açmaktadır. Veri kaybı veya bozulması gibi riskler taşımaktadır.
- İnsan hatası ve kasıtlı yanlış yapılandırma: Kullanıcı hataları veya kasıtlı sabotajlar güvenlik açıklıklarına yol açabilmektedir. Bu durum; yapılandırma hatalarından, bilgi eksikliğinden veya iletişim sorunlarından kaynaklanabilmektedir.
- Depolama ortamlarının fiziksel olarak çalınması: Veriler, fiziksel ortamlarda saklandığından, bu ortamların çalınabilmesi olasıdır.
- Ağ dinlenmesi: Verilerin iletim sırasında dinlenmesi veya değiştirilmesi, iletim yolundaki bileşenlerin tehlikeye girmesiyle gerçekleşebilmektedir. Şifreleme, ağ dinlenmesi ihtimalini tamamen ortadan kaldıramamaktadır.
- Güvensiz görüntüler, yazılımlar ve sabit yazılımlar: Saldırganlar, yazılım güncellemeleri veya kurulum süreçlerini bozarak kötü niyetli veya hatalı kodları

yayabilmektedir. Yazılım güncelleme süreçleri, çeşitli aşamalarda hedef alınabilmektedir.

Bahsi geçen tehditler, depolama altyapısının güvenliğini tehdit eden önemli risklerdir. Her bir tehdit için uygun önlemler alınması, ciddi güvenlik sorunlarıyla karşılaşmamak adına önemlidir.

#### 2.5.1.2. Riskler

Güvenlik riski, bir varlığın potansiyel durum veya olay tarafından ne kadar tehdit altında olduğu şeklinde tanımlanmaktadır. Risk, genellikle durum veya olay gerçekleşirse ortaya çıkacak olumsuz etkiler ve olasılık gibi etkenlerin fonksiyonudur. Bilgi sistemi ile ilgili güvenlik riskleri, bilgi veya bilgi sistemlerinin gizliliğinin, bütünlüğünün ve erişilebilirliğinin kaybından kaynaklanmaktadır. Bu risklerden bazıları şunlardır (Chandramouli ve Pinhas, 2020):

- Veri ihlâli ve veri maruziyeti: Veri ihlâli, hassas bilgilerin yetkisiz kişiler tarafından kopyalanması, iletilmesi veya açığa çıkması anlamına gelmektedir. Bu ihlâller, dış kaynaklardan (örneğin, bir bilgisayar korsanı) veya iç kaynaklardan (örneğin, kötü niyetli bir çalışan) gerçekleşebilir ve ciddi güvenlik sorunlarına yol açabilmektedir. Veri ihlâlinin temel nedenleri arasında yetersiz şifreleme, yazılım hataları ve yanlış erişim kısıtlamaları bulunmaktadır.
- Yetkisiz veri değişikliği ve ekleme: Yetkisiz veri değişikliği, verilerin yetkisiz kişiler tarafından değiştirilmesidir. Bu değişiklikler, genellikle gizli veya tanımlanabilir kaynaklardan gelir ve sistemin işleyişini olumsuz etkileyebilmektedir.
- Veri bozulması: Veri bozulması, verilerin depolama veya işleme sırasında hasar görmesidir. Veri bozulması, sistemde beklenmeyen sonuçlara yol açabilir ve dosyaların açılmasını veya kullanılmasını engelleyebilmektedir. Kötü amaçlı yazılımlar, verileri bozarak bu tür sorunlara neden olabilmektedir.
- Yedekleri tehlikeye atma: Yedekleme, veri kaybı durumunda bilgilerin geri kazanılmasını sağlar. Ancak, yedeklerin bozulması veya zehirlenmesi, verilerin geri alınmasını zorlaştırabilmektedir. Ayrıca, yedekleme süreçlerinin hedef alınması, gelecekteki verilerin de tehlikeye atılmasına yol açabilmektedir.

- Kötü niyetli veri gizleme ve şifreleme: Fidye yazılımları gibi kötü niyetli yazılımlar, verileri şifreleyerek erişimi engelleyebilir ve tekrar erişim sağlanabilmesi karşılığında ödeme talep edebilmektedir. Bu tür saldırılar, genellikle dış kaynaklardan gelir ve verilerin geri alınmasını zorlaştırabilmektedir.
- Veri erişilmezliği ve hizmet reddi: Erişilebilirlik kesintileri, veri erişim yollarına zarar verilmesi sonucu oluşmaktadır. Kesintiler, fiziksel veya mantıksal nedenlerden kaynaklanabilir ve uzun süreli hizmet kesintilerine yol açabilmektedir. Hizmet reddi saldırıları, sistemleri aşırı yükleyerek erişimi engelleyebilmektedir.
- Depolama ile ilgili günlük ve denetim verilerinin bozulması: Günlük verilerinin bozulması, saldırganların saldırıları gizlemek veya araştırmaları yanıltmak için günlükleri değiştirmesi anlamına gelmektedir. Bu tür verilerin bozulması, güvenlik sistemlerinin saldırı tespit etmesini zorlaştırmaktadır.
- Depolama yazılımı ve sistemlerinin tehlikeye atılması: Depolama yazılımı ve sistemlerinin tehlikeye atılması, veri erişiminde değişiklikler veya yanlış raporlamalar gibi çeşitli istenmeyen sonuçlara yol açabilmektedir. Bu saldırılar, sistemlerin davranışı değiştirerek gizli ve tespit edilmesi zor saldırılar için fırsat yaratabilmektedir.

#### 2.5.1.3. Saldırı Yüzeyleri

Saldırı yüzeyi, "yetkisiz bir kullanıcının (saldırgan) bir ortama veri girmeyi veya veriyi çıkarmayı deneyebileceği farklı noktaların toplamı" olarak tanımlanmaktadır. Depolama altyapısıyla ilgili yaygın olan dijital ve fiziksel saldırı yüzeyleri şunlardır (Chandramouli ve Pinhas, 2020):

- Fiziksel erişim: Depolama altyapısına fiziksel erişim; veri merkezine, çevresine, iletişim altyapısına veya fiziksel nesnelere (örneğin, sunucular, sabit diskler) yapılan müdahaleleri içermektedir. Fiziksel erişim çeşitli yollarla gerçekleştirilebilmektedir:
    - Açık erişim: Saldırgan, temizlik görevlisi gibi bir rol üstlenerek güvenlik önlemlerini aşabilir.
    - Kapıdan geçiş (Tailgating): Bir saldırgan, bir çalışanı takip ederek kısıtlı alanlara giriş yapabilir.

        - Medya ve verileri çalma veya zarar verme: Saldırgan, medya ve verileri çalabilir, kopyalayabilir veya yok edebilir. Ayrıca, fiziksel dinleme cihazları kurarak uzaktan erişim sağlayabilir.

- Depolama işletim sistemi erişimi: Depolama cihazlarının işletim sistemlerine güvenlik açıklarından yararlanarak erişim sağlamak mümkün olabilmektedir. Depolama işletim sistemleri, genellikle özel Linux/Unix sürümleridir ve güvenlik açıkları içerebilmektedir. Saldırgan, yerel veya uzaktan oturum açarak bu sistemlere erişebilmektedir.
- Yönetim ana bilgisayarlarına erişim: Depolama bileşenleri, genellikle yönetim ana bilgisayarları aracılığıyla kontrol edilmektedir. Saldırgan, kötü amaçlı yazılım veya güvenlik açıkları yoluyla bu ana bilgisayarlara erişebilir ve veri bozulması, veri kaybı gibi riskler oluşturabilmektedir.
- Yönetim arayüzleri, yönetim yazılımı ve bant içi (in-band) yönetimi: Depolama bileşenleri yönetim yazılımları, arayüzler ve yönetim protokolleri aracılığıyla kontrol edilmektedir. Saldırgan, bu arayüzleri taklit ederek yönetim yetkilerine erişim sağlayabilmektedir. Ayrıca, bant içi erişim kullanarak veri ve yönetim komutları arasında veri iletimi gerçekleştirebilmektedir.
- Depolama istemcileri: Depolama istemcileri, depolama sistemlerinde veri okuma ve yazma işlemleri gerçekleştirmektedir. Tehlikeye giren istemciler, verileri okuma, yazma veya şifreleme gibi işlemler için hedef olabilmektedir. Ayrıca, depolama istemcisini taklit eden saldırgan, yönetim komutları gönderebilmektedir.
- Depolama ağı: Depolama verileri, ağ bileşenleri aracılığıyla iletilmektedir. Bu bileşenler (ana bilgisayar adaptörleri, anahtarlar, kablolar) tehlikeye girerse, saldırgan veri yolunu dinleyebilir, verileri çalabilir veya değiştirebilmektedir. Ortadaki adam (Man-in-the-middle) saldırıları da bu bileşenler üzerinde gerçekleştirilebilmektedir.
- Anahtar kişilerin bilgisayar ortamı: Anahtar kullanıcılar, uzak depolama yönetimi için çeşitli bilgisayar ortamlarını kullanmaktadır. Dizüstü bilgisayarlar, ev ağları gibi ortamlar kötü amaçlı yazılım saldırıları için hedef olabilmektedir. Saldırgan, bu ortamlara kötü amaçlı yazılım yerleştirerek hassas bilgileri çalabilmektedir.
- Elektrik ağı ve diğer kamu hizmetleri: Depolama altyapısı, elektrik ağına bağlıdır ve elektrik ağındaki sorunlar (örneğin, voltaj dalgalanmaları, yıldırım) veri hasarına

neden olabilmektedir. Ayrıca, güç hatlarını kullanarak veri sızdırma yöntemleri veya diğer kamu hizmetleri (ısıtma, havalandırma, iklimlendirme, yangın söndürme sistemleri gibi) potansiyel saldırı yüzeyleri olabilmektedir.

### 2.5.2. Önlemler

Depolama güvenliğini hedef alan tehditler, riskler ve saldırı yüzeyleri karşısında alınabilecek bir takım önlemler bulunmaktadır. Bu önlemler aşağıda ele alınmıştır (Chandramouli ve Pinhas, 2020):

- Fiziksel depolama güvenliği: Fiziksel depolama güvenliği, bilgi teknolojileri altyapısının korunmasında temel bir unsurdur. Depolamaya fiziksel erişim, yazılım tabanlı güvenlik kontrollerini tehlikeye atabilmektedir. Bu bağlamda, alınabilecek önlemler aşağıda sıralanmıştır:
    - Medya güvenliği, NIST ve Uluslararası Standart Organizasyonu (ISO) gibi standartlara uygun olarak sağlanmalıdır. Medyanın satın alınması, yaşam döngüsü yönetimi ve hassas verilerin uzak yedeklenmesi önemlidir. Medyanın envanteri tutulmalı, içeriği izlenmeli ve hassas medya için gelişmiş takip sistemleri (RFID, GPS) kullanılmalıdır.
    - Hassas iş istasyonları ve yönetim ekipmanları, fiziksel güvenlik kontrolleri ve çevresel faktörler tarafından korunmalıdır.
    - Veri temizleme, depolama altyapısındaki tüm görünmeyen bileşenleri (kalıcı bellekler, önbellekler, yazılım ayarları gibi) kapsamalıdır ve uygun politikalarla desteklenmelidir.
- Veri koruma: Veri koruma önlemleri, depolama sisteminin güvenliğini artırmaktadır. Depolama sistemlerinde alınabilecek veri koruma önlemlerine aşağıda yer verilmiştir:
    - Veri koruma planları, farklı yedekleme katmanları, medya türleri ve şifreleme gereksinimlerini içermelidir. Ayrıca, veri

yaşam döngüsünü yönetmeli ve düzenleyici gereksinimlere uygun kurtarma prosedürlerini belirlemelidir.

- İkincil depolamalarda birincil veri koruma önlemleri uygulanmalı, veri gizliliği ve bütünlüğü sağlanmalı ve eski kopyalar kaldırılmalıdır. Senkron çoğaltma ve veri şifrelemesi gereklidir.
- Anlık görüntüler uygun şekilde yapılandırılmalı ve eski kopyalar temizlenmelidir. Bu, kurtarma hedeflerini karşılamak ve saldırı yüzeyini azaltmak için önemlidir.
- Sürekli veri koruma teknikleri, hassas verilerin adli analizi için yararlı olabilir, ancak bu yöntemler zaman alıcı olabilir.

- Kimlik doğrulama ve veri erişim kontrolü: Depolama altyapı sistemleri, bu sistemlere erişim sağlamak için çeşitli hesaplar kullanan yetkilendirilmiş kullanıcılar tarafından yönetilmektedir. Yönetici kullanıcılar ve onların yönetim sunucuları önemli bir saldırı yüzeyi oluşturur ve saldırganlar tarafından istismar edilebilmektedir. Depolama sistemlerini ve altyapısını yöneten kişilerin sahip olduğu sistem yönetim ayrıcalıklarının uygunsuz kullanımı, depolama sistemlerinin başarısızlıkları veya ihlâllerinde önemli rol oynayabilmektedir. Bu durumu önleyebilmek için alınabilecek önlemlere aşağıda sıralanmıştır:
    - Hesaplar ve yönetim kullanıcıları, "En Az Ayrıcalık Modeli"ne göre yönetilmelidir. Dört temel rol (Güvenlik Yöneticisi, Depolama Yöneticisi, Güvenlik Denetçisi, Depolama Denetçisi) belirlenmeli ve her rolün erişim hakları sınırlandırılmalıdır. Benzersiz tanımlayıcılar ve çok faktörlü kimlik doğrulama kullanılmalıdır.
    - Şifreler uzun, karmaşık ve düzenli olarak değiştirilmelidir. Şifreler düz metin olarak saklanmamalı, hizmet hesapları için de güvenli politikalar uygulanmalıdır.
    - Kullanıcılarla ilişkilendirilmeyen hesaplar devre dışı bırakılmalı ve acil durumlar için güvenli yerel hesaplar kullanılmalıdır. Varsayılan hesaplar kaldırılmalı ve minimumda tutulmalıdır.

- Ayrıcalıklar rol atama ve görev ayrımı ilkesine uygun olarak yönetilmeli, tüm oturumlar gerekli kimlik doğrulama güvenlik seviyelerinde tutulmalıdır.
- SAN erişim kontrolü bölgeleme (zoning) ve maskeleme (masking) ile sınırlandırılmalı, varsayılan bölge "tümünü reddet" olarak ayarlanmalıdır. Dosya ve nesne erişimi en az ayrıcalık ilkesine göre kontrol edilmeli ve hassas bilgiler kötü amaçlı yazılımdan koruma (anti-malware) araçlarıyla taranmalıdır.

- Denetim günlüğü: Denetim günlüğü, sistemlerin güvenlik olaylarını ve işlemlerini izleyerek, sistemlerin izlenebilirliğini ve sorgulanabilirliğini artırmaktadır. Etkili ve güvenli denetim günlüğü süreç yönetimi sağlamak için alınabilecek önlemler aşağıda sıralanmıştır:
  - Denetim günlükleri, güvenli protokoller kullanılarak tüm depolama bileşenlerinde etkinleştirilmelidir.
  - Ağ Zaman Protokolü (NTP) tüm cihazlarda etkin olmalı, olayların doğru zaman damgalarıyla kaydedilmesi sağlanmalıdır.
  - Günlükler merkezi sunucularda toplanmalı ve merkezi günlükleme sürekli izlenmelidir.
  - Günlük verileri yeterince uzun süre saklanmalı, arşivlenmiş veriler değiştirilemez şekilde korunmalı ve erişim şifreleme ile güvence altına alınmalıdır.
  - Destekleniyorsa, günlükler Güvenlik Bilgi ve Olay Yönetimi (SIEM) yazılımı ile entegre edilmelidir.
- Veri olay müdahale ve siber kurtarma hazırlığı: Veri olay müdahale ve siber kurtarma, siber güvenlik programının kritik bir parçasıdır. NIST çerçevesine göre, etkin bir olay müdahale planı izolasyon, kök neden analizi, yanıt planı oluşturma, test etme ve süreç gözden geçirme adımlarını içermektedir. Bu doğrultuda alınabilecek önlemlere aşağıda yer verilmiştir:
  - Depolama dizileri, yedekleme sistemleri ve diğer bileşenler için yanıt planları oluşturulmalıdır.
  - Kurtarma kopyaları olay yönetimi sırasında izole tutulmalıdır.

- Kurtarılan bileşenlerin enfeksiyonlardan arındırılması sağlanmalıdır.

- Ağ yapılandırma: Depolama ile ilgili ağ konuları; veri erişim kontrolü, yönetim erişimi, şifreleme gibi konularla örtüşmektedir. Depolama güvenliği için ağ yapılandırma konusunda alınabilecek bazı önlemler aşağıda sıralanmıştır:
  - FC SAN ve NVMeoF: Kimlik doğrulama, Açık Anahtar Yapısı (PKI) kullanımı, bölgelendirme, maskeleme ve ağ yönetimi detaylandırılmalıdır.
  - IP depolama ağı: Trafik ayrımı, yönetim portlarının izole edilmesi, erişim kontrolü ve Basit Ağ Yönetim Protokolü (SNMP) güvenliği sağlanmalıdır.
  - Protokoller: Güvensiz protokoller engellenmeli ve güvenlik artırılmalıdır.
- İzolasyon: Veri kurtarma senaryolarında, kötü niyetli olmayan durumlarda veri kopyaları kullanılmalı, siber saldırı durumlarında ise güçlendirilmiş ve izole edilmiş kopyalar tercih edilmelidir. İzolasyon konusunda alınabilecek önlemlere aşağıda yer verilmiştir:
  - Kurtarma kopyaları ayrı sistemlerde tutulmalıdır.
  - Kurtarma sistemleri, üretim ortamından bağımsız yönetilmelidir.
  - Erişim, sadece yetkili kişilere veya gruplara verilmelidir.
- Kurtarma güvencesi: Başarılı veri kurtarma için yalnızca bir süreç oluşturmak yeterli değildir; tüm kritik veri bileşenlerinin korunduğundan ve eksiksiz bir şekilde geri yüklenebildiğinden emin olunmalıdır. Bu doğrultuda alınabilecek önlemler aşağıda sıralanmıştır:
  - Tüm kritik veri ve bileşenler yedeklenmeli ve korunmalıdır.
  - Gerekli tüm yazılım ve donanım bileşenleri yedeklenmeli ve geri yüklenebilir olmalıdır.
  - Yedeklerin geri yüklenebilirliği düzenli olarak test edilmelidir.
  - Veri kurtarma hedefleri belirlenmeli ve yedekleme frekansı bu hedeflerle uyumlu olmalıdır.
  - Kritik bilgiler kötü amaçlı yazılım taramalarına tabi tutulmalıdır.

- Yedekleme ve kurtarma süreçleri yılda en az bir kez gözden geçirilmelidir.

- Şifreleme: Şifreleme, hassas verileri korunmak için kritik bir yöntemdir ve depolama sistemlerinde uygulanmalıdır. Depolama sistemlerinde şifrelemenin uygulanmasına ilişkin önlemlere aşağıda yer verilmiştir:
  - Veriler, depolama ortamlarında şifrelenmelidir.
  - Depolama elemanları ve ağlar üzerindeki veri iletimi şifrelenmelidir.
  - Yönetim protokolleri ve API erişimleri şifrelenmelidir.
- Yönetim erişimi: Yönetim erişimi, bilgi teknolojileri altyapısının kontrolü için kritik bir öneme sahiptir. Yönetim erişimini güvence altına almak için önerilen önlemler aşağıda sıralanmıştır:
  - Yönetim portlarına sadece yetkili cihazlar ve yöneticiler erişim sağlamalıdır.
  - Yönetim yeteneklerine sahip cihazlar sınırlı olmalı ve gereksiz olanlar kaldırılmalıdır.
  - Yönetim hakları en az ayrıcalık ilkesine göre verilmelidir.
  - Hizmet hesaplarının erişimi, sadece gerekli okuma ve meta veri erişimi ile sınırlandırılmalıdır.
  - Komut Satırı Arabirimi (CLI) ve API erişimleri kimlik doğrulama ve yetkilendirme süreçlerine tâbi olmalıdır.
  - CLI yerine API erişimi tercih edilmeli ve CLI güvenli protokollerle kullanılmalıdır.
  - Yönetim internet arayüzleri, diğer internet uygulama standartlarını karşılamalı veya aşmalıdır.
  - Host yönetim erişimi sadece gerekli öğeler ve eylemlerle sınırlı olmalıdır.
  - Komut cihazları ve geçitler en aza indirilmelidir.
  - Uzak erişim devre dışı bırakılmalı veya güvenli bir şekilde sınırlandırılmalıdır.
- Yapılandırma yönetimi: Yapılandırma yönetiminin amacı, depolama varlıklarının yaşam döngüsü boyunca ayarlarına, davranışlarına ve fiziksel ile mantıksal özelliklerine yönelik görünürlük ve kontrol sağlamaktır. Yapılandırma yönetimine ilişkin önerilere aşağıda yer verilmiştir:

        - Depolama cihazlarının detaylı envanteri oluşturulmalıdır.
        - Depolama bileşenleri ve yapılandırmaları kapsamlı şekilde listelenmelidir.
        - Depolama güvenlik politikası oluşturulmalı ve düzenli olarak güncellenmelidir.
        - Yapılandırma standartlarına uyum periyodik olarak kontrol edilmelidir.
        - Yapılandırma değişiklikleri planlanmalı ve onaylanmalıdır.
        - Yetkisiz değişiklikler için bir tespit süreci oluşturulmalıdır.
        - Yazılım güncellemeleri ve yamalar düzenli olarak yapılmalıdır.
        - Depolama ağ topolojisi güncel tutulmalıdır.
        - FC SAN yapılandırmaları yılda en az bir kez denetlenmelidir.
- Eğitim: Depolama güvenliği için eğitim; bilgi güvenliği profesyonelleri, depolama yöneticileri ve yöneticiler için özel olarak tasarlanmalıdır. Eğitimler; güvenlik prensipleri, politikalar ve veri koruma temellerini kapsamalıdır.

# BÖLÜM 3

# NESNE DEPOLAMA

Günümüzde dijital verilerin üretim ve tüketim hızında belirgin bir artış gözlemlenmektedir ve bu artışın devam etmesi beklenmektedir. (Qader vd., 2020) Özellikle, yapılandırılmamış verilerdeki artış, yapılandırılmış verilere kıyasla çok daha hızlı ve belirgin bir şekilde gerçekleşmektedir. Veri miktarındaki bu artış, veri depolama ve yönetimi konusunda çeşitli zorlukları da beraberinde getirmektedir. Geleneksel veri depolama yöntemleri, dosya depolama ve blok depolama gibi yaklaşımlar, büyük veri hacimlerinin etkili bir şekilde depolanmasında yeterli verimliliği sağlayamamaktadır.

Dosya depolama ve blok depolama sistemlerinden daha yeni bir teknoloji olan nesne depolama, ilk ortaya çıktığı günden bu yana önemli bir gelişim göstermiştir ve özellikle büyük hacimli, yapılandırılmamış verilerin depolanmasında tercih edilen bir çözüm hâline gelmiştir. Önemli kuruluşlar, nesne depolama teknolojisini doğrudan veya diğer depolama sistemleri aracılığıyla kullanarak bu teknolojinin sağladığı avantajlardan faydalanmaktadır. Bu bağlamda, nesne depolamanın yaygınlaşma oranı gün geçtikçe artmaktadır.

Bu bölümde, nesne depolamaya ilişkin ilk çalışmalar, nesne depolamanın önemi, nesne depolama sınıfları, nesne, nesne depolama cihazı, nesne depolamanın diğer depolama biçimleriyle karşılaştırılması, nesne depolamanın avantaj ve dezavantajları, kullanım alanları ve örnekleri ele alınmıştır. Ayrıca, nesne depolama ile zekâ arasındaki ilişkiye de değinilmiştir.

## 3.1. Nesne Depolamaya İlişkin İlk Çalışmalar

Nesne depolama kavramı, 1990'ların başlarında araştırma topluluğu tarafından tanıtılmıştır (Factor vd., 2005). Ancak, nesne depolamanın ilkel biçimleri, 1980'lerde nesne yönelimli işletim sistemlerinin erken çalışmalarında ortaya çıkmıştır. Carnegie Mellon Üniversitesi'nden Hydra OS ve Intel'den iMAX-432OS gibi işletim sistemleri, diskte değişken boyutlu nesneleri kullanarak işlem durumu, talimatlar ve veriler dâhil olmak

üzere sayfalanabilir tüm varlıkları ve dosyaları depolamak için geliştirilmiştir. Bu işletim sistemleri, nesne depolamanın temel özelliklerinden biri olan yetenek tabanlı güvenliğin temellerini atmada önemli bir rol oynamıştır. 1980'lerde Massachusetts Teknoloji Enstitüsü'nden (MIT) Swallow Projesi, erken dosya sunum mimarilerinin bir öncüsü olarak kabul edilir ve dağıtılmış nesne depolamayı ilk uygulayan sistemlerden biri olmuştur (Mesnier vd., 2003).

Nesne depolamanın temel çalışması, Carnegie Mellon Üniversitesi'nin Paralel Veri Laboratuvarı tarafından yürütülen Ağa Bağlı Güvenli Diskler (NASD) projesi ile gerçekleştirilmiştir (Gibson vd., 1997). Bu projede, özellikle ağ, güvenlik ve temel alan yönetimi işlevleri için tekil disklerde işlem gücünün maliyet etkin bir şekilde artırılmasına odaklanılmıştır. Araştırma, Ulusal Depolama Endüstri Konsorsiyumu (NSIC) tarafından desteklenen büyük bir endüstri projesine dönüşmüştür. Birçok depolama şirketi bu projeye katılmış ve NASD, ağ bağlantılı depolama cihazlarına genelleştirilmiştir. Tekil sürücüler, izinler, kontrolörler ve cihazlar arayüz değişiminden yararlanmıştır. Bu çalışma, SCSI protokolüne nesne depolama için standart bir eklenti yapılmasıyla sonuçlanmıştır (Mesnier vd., 2003).

Standartların gelişmeye devam etmesiyle birlikte, sektör nesne depolama teknolojisini uygulamaya başlamıştır. IBM, yeni nesil StorageTank için nesne depolama üzerine araştırmalar yapmış; National Laboratories ve Hewlett-Packard, temel depolama arayüzü olarak nesne depolamayı kullanarak yüksek derecede ölçeklenebilir Lustre dosya sistemini kurmuştur. Ayrıca, BlueArc, DataDirect ve Panasas gibi dönemin küçük ve yeni kurulan şirketleri nesne depolamayı kullanan cihazlar üretmiştir (Mesnier vd., 2003). EMC Centera projesi, referans verilerin arşivlenmesi için "bir kere yazılan" medya uygulaması için nesne depolama ilkelerini kullanmıştır (EMC, 2003).

### 3.2. Nesne Depolamanın Önemi

Nesnelerin interneti, yapay zekâ ve bulut bilişim gibi teknolojilerin günlük yaşamın ayrılmaz bir parçası hâline gelmesiyle, veri üretiminde önemli bir patlama yaşanmaktadır (Sivakumar vd., 2023). Özellikle yapılandırılmamış veri miktarındaki hızlı artış, yapılandırılmamış veri depolama pazarının hızla gelişmesine yol açmaktadır. Bu bağlamda, nesne depolama temelli altyapı ürünleri, yapılandırılmamış veri depolama

ihtiyaçlarını karşılamak üzere hızla ölçeklenmekte ve kapasite açısından genişlemektedir (Shankar ve Yadav, 2022).

Ulusal Veri Şirketi (IDC) tarafından yayınlanan raporlar, işletmelerin petabayt ölçeğindeki veri depolama ihtiyaçlarına yönelik ilerlemeleriyle birlikte, nesne depolama çözümlerinin ölçeklenebilirlik, karmaşıklık ve maliyet dengelemesi açısından ideal bir seçenek olarak öne çıktığını vurgulamaktadır. Nesne depolamanın yüksek veri iletimi, ölçeklenebilirlik ve düşük maliyet gibi avantajları, büyük miktardaki yapılandırılmamış verilerin etkili bir şekilde depolanmasını sağlamaktadır (Potnis, 2019). Ayrıca, depolama sistemleri üzerindeki araştırmaların giderek nesne depolamaya yönelmesi ve endüstrinin büyük veri depolamada nesne depolama çözümlerine doğru bir eğilim göstermesi, nesne depolamanın ağ depolama teknolojisinin bir sonraki evrimi olarak kabul edileceğinin göstergesi olarak değerlendirilmektedir (Liu vd., 2006).

### 3.3. Nesne Depolama Sınıfları

Nesne depolama, depolama cihazları ve internet hizmetleri olmak üzere iki ana sınıfa ayrılmaktadır. Depolama cihazları, donanım tabanlı çözümler sunarken, internet hizmetleri ise internet üzerinden erişilebilen daha soyut çözümler sağlamaktadır. Bu iki nesne depolama sınıfı, veri erişimi ve yönetimi konusunda farklı yaklaşımlar sunmaktadır (Welch, 2013):

- Depolama cihazlarında, veri yönetimi ve erişimi gibi akıllı işlevler doğrudan cihaz üzerinde gerçekleştirilmektedir. Bu cihazlar, genellikle yüksek bir soyutlama düzeyinde veri yönetimi ve erişim sunar; böylece kullanıcıların veya sistemlerin veriye erişimini basitleştirmektedir. Depolama cihazları, veri güvenliğini sağlamak amacıyla çeşitli güvenlik özellikleri ve protokolleri içermektedir. Örneğin, SCSI, depolama cihazları ile iletişim kurmak için kullanılan bir komut setidir ve bu model üzerinden cihazlar komut alıp yanıt verebilmektedir. Ayrıca, nesne depolama cihazlarına özgü olarak tasarlanmış bir komut seti de mevcuttur. Bu komut seti, veri yönetiminde ve erişiminde nesne tabanlı bir yaklaşım sunmaktadır.
- İnternet hizmetleri, depolama soyutlamasını internet tabanlı sistemlere entegre etmektedir. Bu entegrasyon sayesinde, kullanıcıların veya uygulamaların internet üzerinden veri depolama ve erişim işlemleri daha kolay hâle gelmektedir. İnternet

hizmetleri, veri güvenliğini sağlamak için çeşitli güvenlik protokolleri ve mekanizmaları kullanmaktadır. Temsili Durum Aktarımı (REST) iletişim modeli, internet hizmetleri ile etkileşimde bulunmayı sağlar ve bu hizmetler REST prensiplerine göre yapılandırılmıştır. Veri transferinde kullanılan temel protokol Hiper Metin Transfer Protokolü (HTTP) olup, veri iletimini gerçekleştirir ve REST tabanlı sistemlerde yaygın olarak kullanılmaktadır.

## 3.4. Nesne

Nesne depolamada veri, "nesne" adı verilen yapılar hâlinde depolanmaktadır. Nesne, depolama cihazı içinde sıralı baytlar kümesi olarak tanımlanır ve değişken boyutlarda olabilmektedir. Her nesne, veriyi ve bu veriyle ilişkili öznitelikleri içerir ve kendisine atanmış benzersiz bir tanımlayıcıya (ID) sahiptir (Milligan, 1999). Veri, bir aile fotoğrafından yüzlerce sayfalık bir rapora kadar geniş bir yelpazede olabilmektedir. Öznitelikler, veri hakkında bilgi sağlayan veri setleri olarak tanımlanmaktadır. Benzersiz tanımlayıcı ise, her nesneye özgü bir adres işlevi görerek, nesnelerin fiziksel yerini bilmeye gerek olmaksızın erişimi kolaylaştırmaktadır. Bu özellikler, nesne tabanlı depolama sistemlerinin etkinliğini ve veri yönetiminin esnekliğini artırmaktadır (Lu vd., 2017).

### 3.4.1. Nesne Türleri

Nesneler, farklı türlerde olabilmektedir. Kök nesne, bölüm, koleksiyon ve kullanıcı nesnesi olmak üzere toplam dört tür nesne bulunmaktadır (Maddi, 2009).

#### 3.4.1.1. Kök Nesne

Depolama cihazındaki tüm ayarlamaları kök nesnenin öznitelikleri kontrol etmektedir. Her mantıksal birim, sadece bir adet kök nesne içermektedir. Teorik olarak, mantıksal birimler kök nesne olarak temsil edilmektedir ve kök nesnenin öznitelikleri, mantıksal birimin öznitelikleriyle (örneğin, mantıksal birimin toplam kapasitesiyle) aynı

olmaktadır. Kök nesne, kök nesnede bulunan tüm kullanıcı nesneleri için geçerli olan özniteliklerle birlikte atanır ve kök nesnede bulunan tüm bölümlerin listesini de tutmaktadır. Ancak, kök nesne, okuma/yazma veri alanı içermemektedir. Cihaz sunucusu, kök nesneye gönderilen tüm OKUMA (READ) ve YAZMA (WRITE) komutlarını sonlandırmaktadır (Maddi, 2009).

#### 3.4.1.2. Bölüm

Bölüm nesnesi, bir uygulama istemcisinden gelen belirli komutlar aracılığıyla oluşturulmaktadır. Bölüm, ortak güvenlik gereksinimlerini ve öznitelikleri paylaşan bir dizi koleksiyon ve kullanıcı nesnesini içerir; bu öznitelikler varsayılan güvenlik yöntemi, kapasite kotası gibi parametreleri kapsamaktadır. Bazı bölüm öznitelikleri için varsayılan değerler, kök nesnede belirtilen özniteliklerden kopyalanmaktadır. Her bölüm, kendisine ait koleksiyon ve kullanıcı nesnelerinin listesini tutar ve "CREATE PARTITION" komutu ile oluşturulmaktadır. Bölüm, okuma/yazma veri alanı içermez ve bu nedenle bölüme gönderilen tüm OKUMA (READ) ve YAZMA (WRITE) komutları sonlandırılmaktadır (Maddi, 2009).

#### 3.4.1.3. Koleksiyon

Koleksiyonlar, benzer özniteliklere sahip nesnelerin gruplandırılması amacıyla kullanılmaktadır. Bu yapı, kullanıcı nesnelerinin hızlı bir şekilde dizinlenmesini sağlamaktadır. Koleksiyonlar, bir bölüm içinde yer alır ve bir bölüm, sıfır veya daha fazla koleksiyon içerebilmektedir. Bir kullanıcı nesnesi, aynı anda sıfır veya daha fazla koleksiyonun üyesi olabilmektedir. Koleksiyonların desteklenmesi isteğe bağlıdır. Koleksiyon öznitelikleri için bazı varsayılan değerler, koleksiyonun bulunduğu bölümün belirtilen özniteliklerinden kopyalanmaktadır. Her koleksiyon, bu koleksiyona dâhil olan kullanıcı nesnelerinin bir listesini içermektedir. Bu liste, "LIST COLLECTION" komutu ile elde edilebilmektedir (Maddi, 2009).

#### 3.4.1.4. Kullanıcı Nesnesi

Kullanıcı nesnesi, son kullanıcı verilerini içeren bir yapıdır; bu veriler dosya veya veri tabanı verileri gibi çeşitli biçimlerde olabilmektedir. Kullanıcı nesnesinin öznitelikleri, kullanıcı verilerinin mantıksal boyutunu ve verinin oluşturulma, erişim ve değiştirilme zaman damgalarını içermektedir. Ayrıca, bazı kullanıcı nesnesi özniteliklerinin varsayılan değerleri, bulunduğu bölümün belirtilen özniteliklerinden kopyalanmaktadır (Maddi, 2009).

### 3.4.2. Öznitelikler

Öznitelikler, nesnelerle ilişkili özellikler olup, nesne erişimi sırasında istenen davranışları veya etkileri belirlemek ve nesnelere meta veri olarak içsel özellikler atamak için kullanılmaktadır. Nesneler veri içerecek şekilde tasarlandığından, öznitelikler veri üzerindeki işlemleri de etkileyebilmektedir (Milligan, 1999).

Her öznitelik, bir öznitelik numarası ile tanımlanır ve bir öznitelik uzunluğu ile ilişkilendirilmektedir. Öznitelik uzunluğu, özniteliğin tanımlı mı yoksa tanımlanmamış mı olduğunu belirlemek için kullanılmaktadır. Öznitelik uzunluğu sıfır olmayan bir değere sahip olduğunda, bu öznitelik tanımlı öznitelik olarak adlandırılmaktadır. Öznitelik uzunluğu sıfır olan öznitelikler ise tanımlanmamış öznitelikler olarak kabul edilmektedir (Maddi, 2009).

Öznitelikler, depolama sistemlerinde verinin tanımlanmasına olanak tanıyan ortak bir bilgi kümesi sağlamaktadır. Ayrıca, depolama cihazlarının nesnelerin nasıl erişildiğine dair farkındalık kazanmasını sağlayan anahtar bir işlev görmektedir. Benzer özniteliklere sahip nesneler, bir depolama cihazında grup olarak düzenlenir ve bu sayede verimli bir şekilde yönetilmesi sağlanır. Bu düzenleme, nesnelerin erişim ve yönetim süreçlerinin daha etkin bir şekilde yürütülmesine olanak tanımaktadır (Mesnier vd., 2003).

Öznitelikler, nesne depolamanın önemli bir ayırt edici özelliğidir ve bu öznitelikler, belirli bir düzen içinde organize edilmektedir. Her öznitelik, belirli bir sayfada gruplanıp, bu sayfalarda öznitelikler benzer kaynaklara veya kullanımlara sahip özniteliklerle bir araya getirilmektedir (Maddi, 2009).

Öznitelikler; Nesne Depolama Cihazı (OSD) tarafından belirlenen öznitelikler, statik öznitelikler, oturum öznitelikleri ve genişletilmiş öznitelikler olmak üzere dört sınıfa ayrılmaktadır (Milligan, 1999).

#### 3.4.2.1. OSD Tarafından Belirlenen Öznitelikler

Cihazlar, nesneler üzerinde gerçekleştirilen işlemler sonucunda, OSD tarafından belirlenen öznitelikleri ayarlamaktadır. Bu öznitelikler, politika bazlı bakım süreçlerinde kullanışlı olup, aşağıdaki gibi tanımlanabilmektedir (Milligan, 1999):

- Nearby Object: Nesnenin mümkün olduğunca yakınında bulunması gereken bir nesnenin nesne ID'si
- Depending Object: Nesnenin bağımlı olduğu bir nesnenin nesne ID'si
- Copied Object: Nesnenin bir kopyası olduğu nesnenin ID'si
- Object Logical Length: Bir READ talebinde veri döndüren en yüksek bayt adresi

#### 3.4.2.2. Statik Öznitelikler

Statik öznitelikler, nesneyi oluşturan işlem tarafından belirlenen ve kalıcı niteliklere sahip özelliklerdir. Bu öznitelikler, ana bilgisayar dosya sistemleri tarafından kullanılmak üzere tasarlanmıştır ve aşağıdaki meta verileri içermektedir (Milligan, 1999):

- OBJECT_SIZE: OSD'nin nesneye atadığı depolama alanı miktarı
- CREATED TIME: Nesnenin oluşturulduğu zaman (OSD'nin saati temel alınarak)
- DATA_MODIFIED_TIME: Nesnenin en son yazıldığı veya düzenlendiği zaman (OSD'nin saati temel alınarak)
- DATA_ACCESS_TIME: Nesnenin en son okunduğu veya yazıldığı zaman (OSD'nin saati temel alınarak)
- ATTRIBUTE_MODIFIED_TIME: Nesnenin özniteliklerinin en son değiştirildiği zaman (OSD'nin saati temel alınarak)
- EXPIRATION_TIME_STAMP: Nesnenin artık gerekli olmadığı zaman
- INDELIBILITY: Değiştirilmez bir işaretçi (indicator)

- TRANSIENT_OBJECT: Dosya sisteminin, güç kaybı durumlarında nesnenin hayatta kalmasını beklemediğini belirten öznitelik
- FILE SYSTEM: Nesnenin oluşturulduğu dosya sisteminin kimliği
- FILE SYSTEM SPECIFIC ATTRIBUTES: OSD tarafından yorumlanmayan, dosya sistemiyle ilgili özel bilgiler

#### 3.4.2.3.Oturum Öznitelikleri

Oturum öznitelikleri, erişim süreci tarafından doğrudan değiştirilebilen özellikleri ifade etmektedir. Bu öznitelikler, bir oturum oluşturulduğunda belirlenir ve sürecin belirli koşullar altında değişiklik göstermesine olanak tanımaktadır. Oturum öznitelikleri, süreç-nesne dinamik özniteliklerinin değer setinin diğer süreç-nesne çiftlerinden bağımsız olarak saklanmasını sağlamaktadır. Aynı nesne üzerinde yapılan farklı süreçler için OSD, her süreç-nesne çiftine özgü bir oturum kimliği ile bu değer setini ilişkilendirmelidir. Oturum öznitelikleri, şunları içermektedir (Milligan, 1999):

- TIME TO INITIAL ACCESS (TIA): Nesneden ilk veri baytı teslim edilene kadar tolere edilebilecek ortalama zaman gecikmesi (milisaniye cinsinden)
- SUSTAINED ACCESS RATE (SAR): Verilerin nesneden okunması veya yazılması gereken devam eden ortalama veri hızı (saniye başına bayt cinsinden)
- FREQUENCY OF ACCESS (IOR): Nesneden veri okumak veya nesneye veri yazmak için beklenen ortalama istek sayısı (saniye başına)
- BANDWIDTH OF ACCESS (BOA): Nesneden okunacak veya nesneye yazılacak verinin ortalama istek boyutu

Oturum öznitelikleri bazı durumlarda ayrı ayrı belirtilebilmektedir. Bu durumlar aşağıda verilmiştir (Milligan, 1999):

- READ VS. WRITE BIAS: Belirlemenin okuma istekleri için mi yoksa yazma istekleri için mi olduğunu gösterir.
- DIRECT VS. SEQUENTIAL ACCESS: Belirlemenin doğrudan erişim istekleri için mi yoksa ardışık istekler için mi olduğunu gösterir.

#### 3.4.2.4.Genişletilmiş Öznitelikler

Genişletilmiş öznitelikler, daha fazla özellik ve/veya yüksek seviyede kontrol imkânı sağlamak amacıyla kullanılan ek özniteliklerdir. Bu öznitelikler, nesnelerin yönetimini ve kontrolünü optimize etmek için tasarlanmış komutlar aracılığıyla belirlenir, alınır, değiştirilir, silinir ve listeleme işlemleri yapılır. Aşağıda, genişletilmiş özniteliklerle ilgili ana komutlar ve işlevleri açıklanmaktadır (Milligan, 1999):

- SET ATTRIBUTE: Belirli bir özniteliğin değerini ayarlamak için kullanılan komuttur. Nesneye özgü özniteliklerin belirli bir değerle güncellenmesini sağlar.
- GET ATTRIBUTE: Bir nesnenin mevcut özniteliklerini almak için kullanılan komuttur. Nesnenin mevcut özniteliklerinin ve değerlerinin sorgulanmasını sağlar.
- MODIFY ATTRIBUTE: Mevcut bir özniteliğin değerini değiştirmek için kullanılan komuttur. Önceden ayarlanmış bir özniteliğin güncellenmesini sağlar.
- DELETE ATTRIBUTE: Bir özniteliği nesneden kaldırmak için kullanılan komuttur. Nesneden belirli özniteliğin silinmesini sağlar.
- LIST ATTRIBUTE: Bir nesne üzerindeki tüm mevcut öznitelikleri listelemek için kullanılan komuttur. Nesnenin tüm özniteliklerini ve değerlerini görüntülemeyi sağlar.

## 3.5. Nesne Depolama Cihazı (OSD)

Verilerin nesneler hâlinde düzenlendiği ve erişildiği depolama cihazları "nesne depolama cihazı" olarak adlandırılmaktadır. Nesne depolama cihazları, tek bir disk sürücüsünden bir dizi sürücüye sahip bir depolama denetçisine kadar birçok farklı formda olabilmektedir. Bu cihazlar, yalnızca rasgele erişimli veya yazılabilir cihazlarla sınırlı olmayıp, bant sürücüleri ve optik medyalar gibi çeşitli medyalar üzerinde de nesneleri depolamak için kullanılabilmektedir (Mesnier vd., 2003).

Nesne depolama cihazlarının geliştirilmesinde, NSIC ve NASD çalışma grubu, depolama alt sistemlerinin tüketicilerine değer sağlayacak bir mimari tasarlamayı amaçlamıştır. Bu mimarinin amacı, daha yüksek kaliteli depolama yönetimi, daha az ana bilgisayar çabası, veri paylaşımının daha verimli yönetimi ve performans gibi faydalar sağlamaktır. Verileri nesne olarak depolamak ve bu verileri en verimli şekilde almak

amacıyla nesne depolama cihazları bazı hizmetlerin yönetimini gerçekleştirmektedir. Bu hizmetler şunlardır:

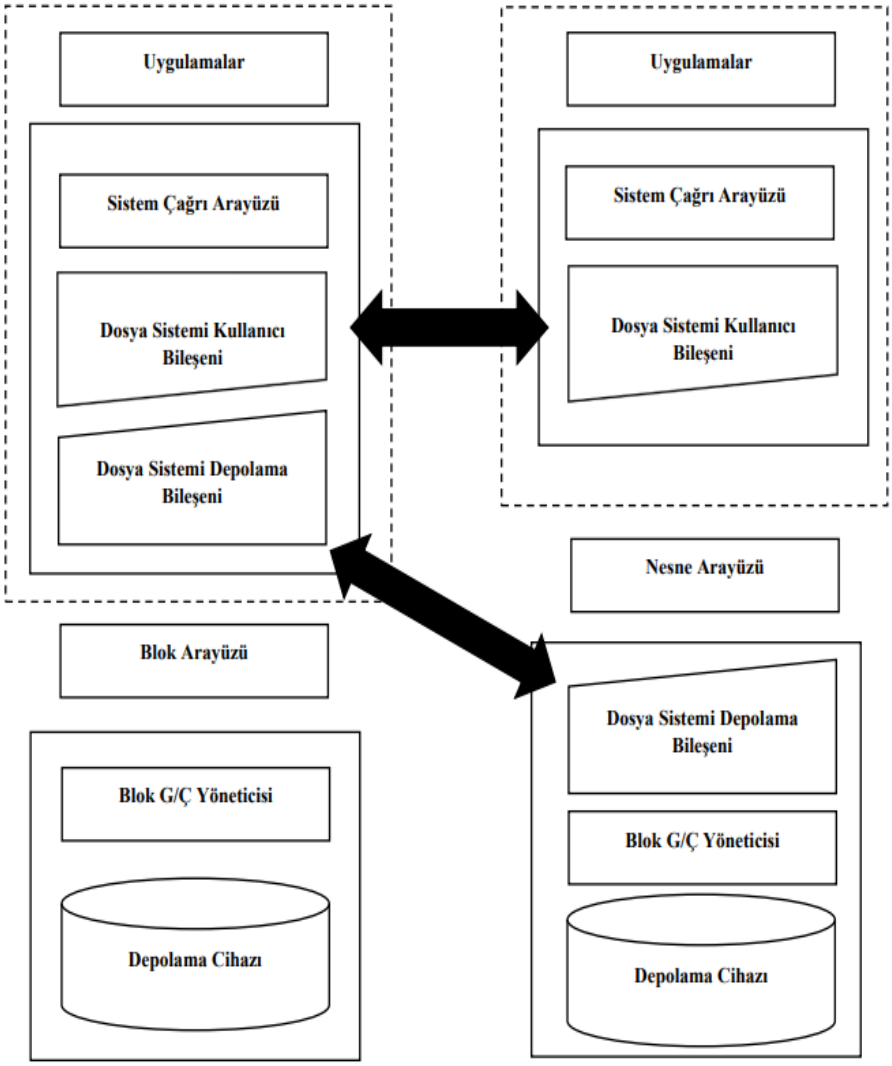
- Boş alan yönetimi: OSD, iç alan yönetimini üstlenir ve uygun bakım için her nesneye ait sistem meta verilerini saklamaktadır. Verilerin verimli bir şekilde depolanabilmesi için boş alan bilgisinin düzgün bir şekilde yönetilmesi gereklidir. Ayrıca, nesneleri tutmak için uygun veri yapıları kullanılması gerekmektedir. Tipik bir nesne depolama sisteminde veriler, parçalara ayrılarak nesneler adı verilen ayrık birimlere dönüştürülmektedir. Bu nesneler, tek bir depolama alanında tutulmaktadır. Nesne depolama mimarisi, verilerin parçalara ayrıldığı ve donanım arasında yayılmış bir düz yapı gibi görünmektedir (Caceres Gutierrez, 2020).
- Güvenlik: Güvenlik, OSD'nin temel işlevlerinden biridir. Erişim yetkilendirme, nesnelere erişim sağlamak için kullanılır. OSD, yetki kontrolü yaparak yetkisiz istemcilerin depolama alanına erişimini engellemektedir (Schlosser ve Iren, 2005).
- Komut yorumlayıcı: Nesne komutlarını depolama sistemi için kullanılan bir formata dönüştürmektedir. Bu süreçte, bir komutu dosya sistemi çağrısına çevirmektedir. Yorumlayıcı süreci, doğrulama, ön işleme, yürütme ve son işleme aşamalarını içermektedir. Doğrulama aşamasında, yorumlayıcı komutun nesne üzerinde erişim ve yürütme yetkisi olup olmadığını doğrulamaktadır. Her komut, hedef nesnenin özniteliklerini alır ya da ayarlar. Ön işleme aşamasında, komutun türü bir grup içinden kontrol edilir ve özniteliği alıp alamayacağı ya da ayarlayıp ayarlayamayacağı belirlenmektedir. Yürütme aşamasında, komutun hedef nesne üzerindeki işlenmesi gerçekleşir ve nesne komutu depolama sistemi özel sistem çağrısına dönüştürülmektedir. Son işleme aşamasında ise son güncellemeler yapılır ve komutla ilgili tüm öznitelikler değiştirilmektedir (Mondal vd., 2024).

Nesne depolama cihazları, büyük veri ve ölçeklenebilirlik gereksinimlerini karşılamak için gelişmiş bir çözüm sunmaktadır. Geleneksel depolama sistemleri ve nesne depolama sistemleri arasındaki en büyük fark dosya bileşenlerindedir. Geleneksel depolama dosya sistemi, kullanıcı bileşeni ve depolama bileşeni olarak ikiye ayrılmaktadır. Kullanıcı bileşeni; hiyerarşi yönetimi, isimlendirme, kullanıcı erişim kontrolü gibi işlevleri yerine getirmektedir. Depolama bileşeni ise dosya sisteminin mantıksal yapılarını fiziksel depolama ortamına yerleştirmeye odaklanmaktadır. Verinin konumunu etkileyebilmektedir. Örneğin, bir nesnenin başka bir nesneye yakın konumda veya daha yüksek performans gösteren mevcut alanın bir bölümünde (bir disk sürücüsünün dış bölgesi gibi) konumlanmasını yönlendirebilmektedir. Bu sayede daha yüksek veri hızı elde

edilmektedir. Nesne depolama cihazı mimarisinde, kullanıcı bileşeni geleneksel depolamadaki gibiyken, depolama bileşeni OSD'ye devredilmektedir. Sorumluluk ayrımı, yönetimi depolama cihazına aktarmayı kolaylaştırır ve bu da nesne depolamanın amaçlanan etkisidir (Mesnier vd., 2003).

**Şekil 4**

*Geleneksel Depolama Modeli ve Nesne Depolama Modeli* Object-based Storage. Mesnier vd., 2003, s.86.




Şekil 4'te görüldüğü üzere, kullanıcı bileşeni değişmeden kalmakta, ancak depolama bileşeni ve dolayısıyla meta veriler depolama cihazına aktarılmaktadır. Bu

durum, cihaz arayüzünün bloktan nesneye dönüştürülmesine olanak tanımaktadır. Geleneksel depolama modelinde, meta verilerin yönetimi tamamen depolama uygulamasının yetki alanına girmektedir. Dosya sistemleri, veriyi düzenleme ve disk üzerindeki meta veri yapılarını koruma hususunda kendine özgü yöntemler kullanmaktadır. Bu bağımlılıklar, depolama cihazına erişilmeden önce meta veri yapıları ve disk düzeni hakkında bilgi sahibi olunmasını gerektirdiğinden, verilerin doğrudan uygulamalar arasında paylaşılmasını zorlaştırmaktadır. Verilerin meta verilerini depolama cihazına taşıma işlemi, nesnelerin meta verilerle depolama uygulamaları arasındaki bağımlılığı ortadan kaldırır ve böylece farklı depolama uygulamaları arasında veri paylaşımını mümkün kılar.

### 3.5.1. Temel Nesne Depolama Cihazı Komutları

Kullanıcılar, doğrudan OSD'ye özgü komutları kullanarak nesne depolama cihazına erişebilir ve nesneleri oluşturma, okuma, değiştirme gibi işlemleri gerçekleştirebilmektedir. Nesne depolama cihazlarında kullanılan başlıca komutlar Tablo 1'de özetlenmiştir (Milligan, 1999).

**Tablo 1**

*Temel OSD Komutları*

| Komut | İşlevi |
|---|---|
| **CREATE** | Yeni bir nesne oluşturur ve statik ile dinamik özniteliklerini ayarlar. |
| **DELETE** | Nesneyi ve nesnenin tüm özniteliklerini siler. |
| **GET ATTRIBUTES** | Nesnenin mevcut özniteliklerini alır. Tüm öznitelikler ve karşılık gelen değerler döndürülür. |
| **SET ATTRIBUTES** | Nesnenin statik ve dinamik özniteliklerini ayarlar. |
| **OPEN** | Nesneye erişim sağlar ve dinamik özniteliklerini ayarlar. |
| **CLOSE** | Açık olan nesnenin oturumunu kapatır ve kilidini açar. |
| **READ** | Nesneden veri okur. Okuma işlemi belirli bir öznitelik veya veri aralığına dayanabilir. |
| **WRITE** | Nesneye veri yazar. Yazma işlemi belirli bir öznitelik veya veri aralığına dayanabilir. |
| **LOCK** | Nesneyi belirli bir süre ya da işlem tamamlanana kadar kilitler, böylece diğer işlemler erişim sağlayamaz. |
| **UNLOCK** | Kilitlenmiş olan nesnenin kilidini açar. |

Tablo 2'de, nesne depolama cihazında bir dosya oluşturma örneğinde kullanılan OSD komutlarının sıralaması verilmiştir. Bu örnekte, bir dosya sisteminin basit bir dosya yapısı oluşturacağı ve bunu OSD üzerinde gerçekleştireceği varsayılmıştır. Basit bir PC/UNIX benzeri dosya sistemi olduğu kabul edilmiştir. Dosya sistemi, "baba/oğul" olarak adlandırılan tek bir alt dizinde bulunan bir dosyadan oluşmaktadır. "Baba" dizinin adı, "oğul" ise dosyanın adıdır. OSD ve nesne grubunun bilindiği varsayılmıştır.

**Tablo 2**

*Yeni Bir Dosya Oluştururken Kullanılan OSD Komutları*

| Adım | Komut | Grup, Nesne | İşlem |
|---|---|---|---|
| **1** | GET ATTRIBUTE | n, 1 | Kök nesnenin id'si alınır; "r" |
| **2** | READ | n, r | "baba" dizininin mevcut olup olmadığı kontrol edilir |
| **3** | CREATE | n | "oğul" dosyasını tutacak nesne "s" döner |
| **4** | CREATE | n | "baba" dizinini tutacak nesne "f" döner |
| **5** | WRITE | n, s | "oğul" dosyasını içeriğini yazar, bir veya birden fazla yazma işlemi içerebilir |
| **6** | WRITE | n, f | "baba" dizininin içeriğini yazar, bir veya birden fazla yazma işlemi içerebilir |
| **7** | WRITE | n, r | Kök nesne, "baba" dizinini içerecek şekilde yeniden düzenlenir |

Object Based Storage Devices Command Set Proposal. Milligan, 1999, s.10

Tablo 3'te oluşturulan yeni nesneye veri aktarımı işlemi gerçekleştirilmektedir. Bu süreçte, veri aktarımı işlemi "baba" veya "oğul" için ayrı ayrı WRITE işlemlerine gerek duymamaktadır. Bunun yerine, tüm veriler tek bir WRITE işlemi ile işlenebilmektedir.

**Tablo 3**

*Nesneye Veri Aktarımı İşlemi Sırasında Kullanılan OSD Komutları*

| Adım | Komut | Grup, Nesne | İşlem |
|---|---|---|---|
| **1** | GET ATTRIBUTE | n, 1 | Kök nesnenin id'si alınır; "r" |
| **2** | READ | n, r | "baba" dizininin mevcut olup olmadığı kontrol edilir |

(Devam ediyor)

**Tablo 3 (Devam)**

| Adım | Komut | Grup, Nesne | İşlem |
|---|---|---|---|
| **3** | CREATE | n | "oğul" dosyasını tutacak nesne "s" döner |
| **4** | CREATE | n | "baba" dizinini tutacak nesne "f" döner |
| **5** | WRITE | n, r | Kök nesne, "baba" dizinini içerecek şekilde yeniden düzenlenir |

Object Based Storage Devices Command Set Proposal. Milligan, 1999, s.10

Tablo 4'te ise "baba" dizini güncellenmektedir. Güncelleme işlemi, OPEN ve CLOSE eylemlerini içermektedir.

**Tablo 4**

*Dosya Güncelleme Sürecinde Kullanılan OSD Komutları*

| Adım | Komut | Grup, Nesne | İşlem |
|---|---|---|---|
| **1** | GET ATTRIBUTE | n, 1 | Kök nesnenin id'si alınır; "r" |
| **2** | OPEN | n, r | Oturum id'sini döner, QoS "lock" nesneyi kilitler |
| **3** | READ | n, r | "baba" dizininin mevcut olup olmadığı kontrol edilir |
| **4** | CREATE | n | "oğul" dosyasını tutacak nesne "s" döner |
| **5** | CREATE | n | "baba" dizinini tutacak nesne "f" döner |
| **6** | WRITE | n, s | "oğul" dosyasının içeriği yazılır, bir veya birden fazla yazma işlemi içerebilir |
| **7** | WRITE | n, f | "baba" dizininin içeriği yazılır, bir veya birden fazla yazma işlemi içerebilir |
| **8** | WRITE | n, r | Kök dizin, "baba" dizinini içerecek şekilde yeniden düzenlenir |
| **9** | CLOSE | n, r | Kök nesnenin kilidi açılır |

Object Based Storage Devices Command Set Proposal. Milligan, 1999, s.11

### 3.5.2. Nesne Depolama Cihazında Güvenlik

Nesne depolama cihazların temel güvenlik yapı taşı yetkilerdir. Yetki, istemcinin hangi erişimlere izin verildiğini açıklayan mekanizmalardır. Nesne depolama cihazlarında güvenli erişimi sağlamak için her komutun belirli bir nesneyi ve belirtilen nesneye karşı

gerçekleştirilebilecek işlemler listesini tanımlayan kriptografik olarak güvenli bir yetki ile birlikte olması gerekmektedir. Yetkiler, cihaz güvenliğini sağlamakla birlikte bireysel nesnelere ayrıntılı güvenli erişimi de sağlamaktadır (Factor vd., 2005).

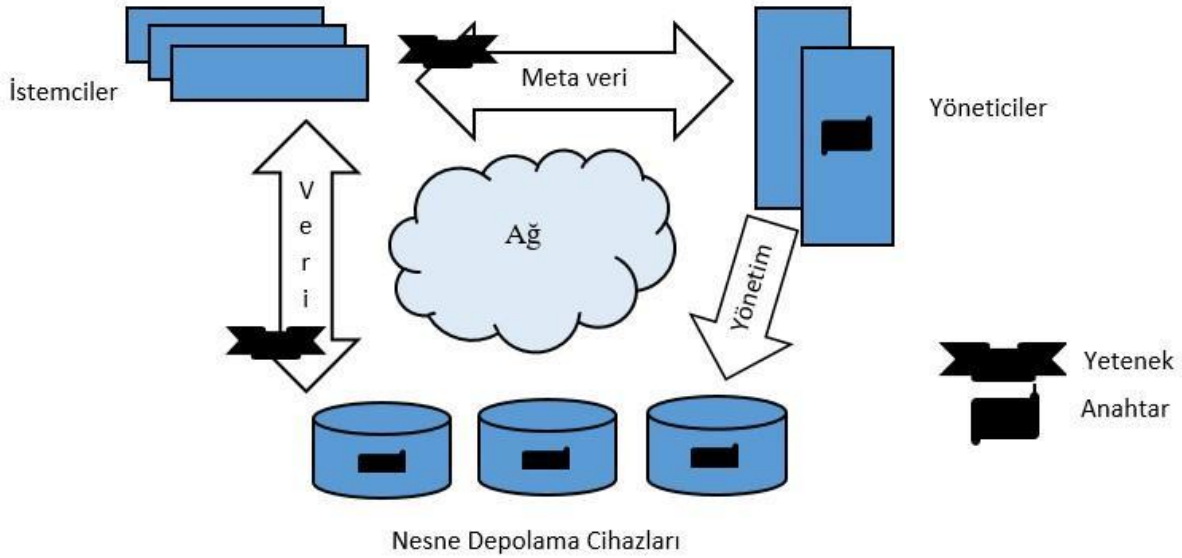
Şekil 5'te gösterildiği üzere nesne depolama cihazı, güvenlik yöneticisi ve istemciden oluşan bir kimlik doğrulama tabanlı erişim sisteminde, bir nesneye ilk erişimden önce, istemci, istenen işlem için güvenlik yöneticisinden bir yetki talep eder. İstemcinin nesneye erişim yetkisi olduğunu varsayarsak, güvenlik yöneticisi uygun bir kimlik bilgisi döndürür; bu kimlik bilgisi şunları içermektedir:

- Erişim haklarını tanımlayan yetki
- Güvenlik yöneticisi ile OSD arasında paylaşılan yetki anahtarı

Her komutta, istemci bu yetkiyi nesne depolama cihazına talebinin bir parçası olarak gönderir ve talebi yetki anahtarı kullanarak kriptografik olarak güvence altına almaktadır. Nesne depolama cihazı talebi doğrular, yetkinin bozulmadığından, istemcinin yetkiyi haklı olarak elde ettiğinden ve talep edilen işlemin yetki tarafından izin verildiğinden emin olur.

**Şekil 5**

*Nesne Depolama Cihazında Güvenlik Mimarisi* Object-based Storage. Mesnier vd., 2003, s.88.



Depolama için kapsamlı bir güvenlik sağlamak amacıyla, OSD ile istemci ve OSD ile güvenlik yöneticisi arasında güvenlik protokolü tanımlanmaktadır. OSD güvenlik protokolü, güvenli ağ ortamlarında, yalnızca erişim kontrolü güvenliğini sağlamaktadır; güvensiz ağlarda çalışırken, hem ağ güvenliğini hem de erişim kontrolünü sağlamaktadır. Her iki ortamı da verimli şekilde desteklemek için, protokolün ağ güvenliğini mekanizmalarının erişim kontrolü mekanizmaları ile birleştirilebileceği bazı yöntemler bulunmaktadır (Factor vd., 2005):

- NOSEC (Güvenliksiz) : Diğer yöntemler ile aynı mesaj yapısı kullanılır, ancak OSD, bir işlemi gerçekleştirmeden önce yetkinin doğruluğunu kontrol etmez.
- CAPKEY (Yetki bütünlüğü) : Erişim kontrolü güvenliği, tüm yöntemler için ortak bir bileşendir. CAPKEY, nesne depolama cihazı ile istemci arasındaki kanalın dışarıdan güvence altına alındığı durumlarda faydalıdır. CAPKEY, istemcinin bir kimliği istemcinin bir kimliği sahteleştirmesini, değiştirmesini veya kimliği başka bir kimlik doğrulama kanalı üzerinden tekrar kullanılmasını önlemek için bir mekanizma sağlar. Ayrıca, istemcinin sunduğu kimliği haklı olarak elde ettiğini doğrular.
- CMDRSP (Komut ve argümanların bütünlüğü): Komut ve argümanların bütünlüğü, OSD ile istemci arasındaki kanal dışarıdan güvence altına alınmadığında ve büyük veri transferlerinin bütünlüğünü sağlamak çok maliyetli olduğunda faydalı olmaktadır. CMDRSP, CAPKEY'e ek olarak, aksi bir durumda güvenli bir kanal tarafından sağlanacak argümanların bütünlüğünü ekler. Kötü niyetli bir istemcinin komut parametrelerini yeniden oynatmasını veya değiştirmesini engeller. Ancak, istemci ile OSD arasındaki mesajların veri kısmında ağ saldırılarına karşı koruma sağlamaz.
- ALLDATA (Veri bütünlüğü): ALLDATA, CAPKEY ve CMDRSP'nin işlevselliğini içerir. Kimlik doğrulama yapılmış bir kanal üzerinden çalıştırıldığında CAPKEY ile benzer güvenlik sağlar. İkisi arasındaki kesin karşılaştırma, sağlanan ağ güvenliği seviyesine bağlıdır. ALLDATA yöntemi, OSD protokolünün ayrılmaz bir parçası olarak ağı güvence altına alır ve böylece depolama katmanında uçtan uca bir çözüm tanımlanır. Güvensiz ağlar için idealdir.

Güvenlik, nesne depolamayı blok tabanlı depolamadan ayıran en önemli özelliklerinden biridir. Güvenlik, SAN'larda genellikle cihaz ve yapı düzeyinde sağlanırken (örneğin, cihazlar bir güvenli giriş gerektirebilir ve anahtarlar bölgeleme uygulanmasını engelleyebilir) nesne tabanlı sistemlerde büyük cihazların veya yapıların çoklu güvenli

alanlara bölünmesine izin verilmektedir. Her bir alanın erişim politikası, depolama uygulaması tarafından ayrı ayrı belirlenmektedir. Nesne tabanlı güvenlik mimarisi, her erişimi yetkilendirir ve yetkilendirme, veri yolunu yavaşlatabilecek merkezi bir otoriteyle iletişim kurmadan gerçekleştirilmektedir. Bu bağlamda, güvenlik nesne depolamanın ayırt edici bir özelliğidir (Nami ve Sakar, t.y.).

## 3.6. Nesne Depolamanın Avantajları ve Dezavantajları

Depolama sistemleri, farklı kullanıcı gruplarına ve farklı iş yüklerine sahip olup, çeşitli özellikler ve önceliklere sahiptir. Her depolama sistemi tasarımı gereği kendine özgü avantajlar ve dezavantajlar sunmaktadır.

### 3.6.1. Avantajları

Nesne depolama, tasarımı gereği çeşitli avantajlar sağlamaktadır. Sağladığı avantajlar arasında depolama yönetimi, zekâ, ölçeklenebilirlik, güvenlik, dinamik yeniden yapılandırma ve farklı bilgisayar sistemleri ile kullanım yer almaktadır (Tauro vd., 2012):

- Depolama yönetimi**:** Nesne depolama, depolamanın kendi kendini yönetebilmesini mümkün kılmaktadır. Depolama alanı, nesne depolamanın öznitelik yönetimine erişimi olduğunda etkili bir şekilde yönetilebilmektedir. Örneğin, sistem yedeklemeleri genellikle hafta sonları yapıldığında, zamanla bu süreç sürekli hâle gelip büyük miktarda veri sürekli olarak yedeklenecektir. Nesne Tabanlı Depolama (OBS), nesneye atanan öznitelikler aracılığıyla bir nesne, yedekleme işlemi için uygun duruma geldiğinde yedekleme işlevini bilgilendirebilmektedir. Veri büyüdükçe, tüm dosyaların yedeği belirli bir süre boyunca alınıp diğerleri güncellenmeye devam ettiğinden veri bütünlüğü önemli bir endişe hâline gelmektedir. OSD'nin depolama yönetim bileşeni, mekânsal nesneleri fiziksel organizasyona eşler ve bireysel veri varlıkları için depolama kapasitesi vermeyi ve boş alanı yönetmeyi sağlamaktadır (Li ve Yu, 2006).
- Zekâ: Nesne depolama, depolama cihazlarının çalıştıkları ortamların önemli özelliklerini aktif olarak öğrenmelerine olanak tanımaktadır. Nesnelerle depolama

cihazları, cihaz üzerindeki bloklar arasındaki bazı ilişkileri anlayabilir ve bu bilgiyi veriyi daha iyi organize etmek için kullanabilmektedir. Cihaz üzerindeki bloklar arasındaki ilişkiyi anlamada nesnenin statik, dinamik, depolama uygulamasına özgü ve özelleştirilebilir özniteliklerinden faydalanılabilmektedir. Nesneler, depolama cihazlarının bulundukları ortamları anlamalarını ve kaynakları daha iyi tahsis etmelerini sağlamaktadır (Tauro vd., 2012).

- Ölçeklenebilirlik: Büyük hacimli verileri depolamak için birden fazla soyutlama ve yönetim seviyesi gereklidir; her biri ayrı ayrı yapılandırılmalı ve işletilmelidir. Meta verinin sistem genelinde dağıtılması, yönetimi zorlaştırmaktadır. Nesne depolama cihazı arayüzü sayesinde, meta veri doğrudan her nesneyle ilişkilendirilebilir ve katmanlar arasında otomatik olarak taşınmaktadır. Depolama cihazları, hangi blokların hangi nesnelere ait olduğunu yönetir ve meta veri katmanlarını birleştirerek sistemin ölçeklenebilirliğini artırmaktadır. Bu sayede, tek bir sunucunun daha fazla sayıda cihazı yönetmesini sağlar ve çok büyük kurumlar için depolama kümeleri oluşturmaya veya daha ekonomik sistemler kurmaya olanak tanımaktadır (Tauro vd., 2012).
- Güvenlik: OSD arayüzü, düşük sistem seviyelerinde, örneğin disk sürücülerinde savunma derinliği güvenliğini sağlamaktadır. OSD arayüzüyle, bireysel disk sürücüsü her okuma ve yazma isteğini doğrulayabilmektedir. Nesne düzeyinde güvenliğin artan granülaritesi, kullanıcı verilerinin ve depolanmanın korunmasını sağlamaktadır. Her istek için, kullanıcı verilerine erişim sağlamak için mantıksal bir kontrol yapılmaktadır. Sistem tarafından verilen yetki belgesi, OSD cihazlarına erişmek için doğrulanmalıdır. Bu kontrol, hem istenmeyen istemci hem de yetkisiz istekler için güvenliği sağlamaktadır. MIT araştırmacıları tarafından yapılan bir test, 100'e yakın disk sürücüsü etrafında veri erişiminin neredeyse imkânsız olduğunu ortaya koymuştur. Örneğin, SCSI ve ATA gibi çeşitli disk sürücüleri, hassas uygulamalar için formatlanmış ve nadiren veri geri alınabilmiştir. Geleneksel ağ depolama mimarisi, tüm veri ve meta veriye meta veri sunucu aracılığıyla erişim sağlar ve veri üzerinde merkezi güvenlik kontrolü yapılmaktadır (Mesnier vd., 2003). OBS mimarisindeki istemci ve OSD ayrımı; ağ güvenliğini saldırılardan korunmasını, istemci ve OSD arasında karşılıklı kimlik doğrulanmasını, istemci ve OSD arasındaki iletişimin şifrelenmesi sağlamaktadır (Li ve Yu, 2006).

- Dinamik yeniden yapılandırma: Meta veri, depolama sistemleri arasında taşınır ve OSD arayüzü aracılığıyla veri nesnesiyle ilişkilendirilmektedir. Sistem katmanı, bireysel nesneler için özniteliklerin parçalanmasını sağlamaktadır. Bir nesne, sistem katmanından geçirildiğinde sistemle ilişkili özniteliklerin değerlerine dayalı olarak işlem yapılmaktadır. Değişmemiş diğer öznitelikler için herhangi bir işlem yapılmamaktadır. Böylece, yüksek güvenilirlik olarak işaretlenen nesnelerin geçici olarak işaretlenen nesnelerden farklı şekilde işlenebilmektedir. Bir sistem, depolama cihazları veri nesnelerini ve yönetim özniteliklerini bildiğinden veri nesnelerini depolama cihazları arasında taşıyabilir veya veriye özel işlevler sağlayabilmektedir. Göç sırasında meta veri, cihazlar arasında sağlam kalmaktadır. Anlamsal meta verinin doğrudan nesnelerle ilişkilendirilmesi, depolama cihazlarının veriyi performans ve güvenilirlik için esnek bir şekilde yönetilmesini mümkün kılmaktadır (Tauro vd., 2012).
- Farklı bilgisayar sistemleri ile kullanım: Nesne depolamanın amacı, heterojen bir işlemci kümesinde depolamanın paylaşılmasını sağlamaktır. İşletim sistemi meta veri yapıları benzersizdir; bu nedenle, OBS'nin herhangi bir işletim sisteminin dosya sistemini üzerine koyabileceği ortak bir platforma sahip olma olasılığı vardır. Böylece, nesne depolamanın ana bilgisayar bağımsızlığı kavramını garanti etmektedir (Tauro vd., 2012).

### 3.6.2. Dezavantajları

Nesne depolamanın sağladığı avantajların yanında bazı dezavantajları da bulunmaktadır. Geleneksel veri tabanları ile daha az uyumluluk ve veri bloklarını değiştirememe nesne depolamanın dezavantajları arasında sayılmaktadır (Jyoti vd., 2021).

- Geleneksel veri tabanları ile daha az uyumluluk: Nesne depolama, yapılandırılmamış verileri işlemek üzerine odaklanmaktadır. Bu nedenle, yapılandırılmış verileri depolayan geleneksel veri tabanlarına geri dönmek uyumluluk sorunlarına yol açabilmektedir.
- Veri bloklarını değiştirememe: Nesne depolama, bir nesnenin belirli bir parçasını değiştirmeyi engellemektedir, nesnenin tümünün okunması gerekmektedir. Nesnenin tamamını işlemek ek bir yük getirir ve ek gecikmelere yol açmaktadır. Bu

nedenle, nesne depolama yüksek performans gerektiren kullanım durumlarında verimli olamamaktadır.

### 3.7. Nesne Depolamanın Kullanım Alanları ve Örnekleri

Ölçeklenebilirliği, esnekliği ve kullanım kolaylığı gibi yetenekleri sayesinde nesne depolamadan faydalanılan birçok kullanım alanı bulunmaktadır. En yaygın bazı kullanım alanları ve örnekleri şöyledir (Paul, 2021):

- Medya ve eğlence: Nesne depolama, büyük ölçekli medya dosyalarını ve multimedya varlıklarını depolayabilmektedir. Medya dosyalarına ait meta verilerin kullanımı sayesinde dosyaların tanımlanması ve erişilmesi kolaylaşmaktadır. Film ve video yapımları büyük veri dosyalarını nesne depolarında saklayabilmektedir. Ayrıca, müzik ve ses dosyalarının arşivlenmesi ve dağıtımında da nesne depolama tercih edilen bir depolamadır. Birçok akış platformu, veri depolama ihtiyaçları için nesne depolamayı kullanmaktadır. Netflix, PrimeVideo, Youtube ve Spotify gibi büyük şirketler, veri depolamada nesne depolama çözümlerinden faydalanmaktadır (Tso, 2017).
- Büyük veri: Ölçeklenebilirliği sayesinde petabaytlarca verinin depolanması ve işlenmesinde, makine öğrenimde eğitim modelleri için gerekli büyük veri kümesinin depolanmasında nesne depolama kullanılabilmektedir. Veri gölleri ve analitik platformlarda büyük veri kümelerinin saklanması, makine öğrenimi ve yapay zekâ projeleri için veri hazırlığı ve model eğitimi bu kullanıma örnek olarak verilmektedir.
- Sağlık**:** Hasta verilerinin saklanmasında GDPR ve Kaliforniya Tüketici Gizliliği Yasası (CCPA) gibi düzenlemelere uyumluluk sağlayabilen güvenli ve uyumlu veri depolama kullanılabilmektedir. Görüntüleme ve tarama sonuçları gibi uzun süreli verileri arşivleyebilmektedir.
- Yoğun veri depolama: Büyük miktarda müşteri verisinin ve dosyalarının depolanmasında ve erişilmesinde nesne depolamadan faydalanılabilmektedir. CRM sistemlerinin müşteri verilerini yönetmesi yoğun veri depolama kullanım alanı örnekleri arasında sayılmaktadır.
- Hizmet olarak depolama: S3 protokolü veya API destekli depolama çözümü olarak kullanılabilmektedir. Çoklu kiracı desteği, hizmet kalitesi kontrolü gibi özellikler

sağlayabilmektedir. Veri depolama platformları ve bulut servis sağlayıcıların sunduğu hizmetler (yazılım olarak hizmet gibi) bu kullanıma örnek olarak verilmektedir.

- Yedekleme ve kurtarma: Verilerin farklı veri merkezlerinde yedeklenmesinde tercih edilebilmektedir. Bir kez yaz, çok kez oku (WORM) özelliği ile veri kaybını önleyebilmektedir. Örneğin, kurumsal veri yedekleme çözümü olarak ve felaket kurtarma senaryoları için veri yedeklemede kullanılabilmektedir.
- Soğuk depolama**:** Etkin olmayan, nadiren erişilen verilerin depolanmasında nesne depolama düşük maliyetli bir çözüm sunmaktadır. Uzun süreli arşivleme ve uyumluluk gereksinimleri için depolama, işletmelerin eski verilerini depolaması soğuk depolama kullanım örnekleri arasında yer almaktadır.

Nesne depolamanın sağladığı ölçeklenebilirlik, esneklik ve geniş veri yönetimi kapasitesi sayesinde birçok alanda etkili ve verimli çözümler sunmaktadır. Ancak, bazı uygulamalar için nesne depolama uygun bir çözüm olmayabilir. Nesne depolamanın uygun olmadığı durumlarda, genellikle blok depolama ya da daha düşük gecikme süreleri ve yüksek performans sağlayabilen diğer depolama çözümleri tercih edilmektedir. Bu çözümler, bu alanlarda nesne depolamaya göre daha verimli olabilmektedir (Patil vd., 2017):

- Çok düşük gecikme süresi gerektiren uygulamalar
- Yüksek G/Ç operasyonu gerektiren uygulamalar
- Veri tabanları
- İşlem verileri
- Sanallaştırma ve sanal makineler

### 3.8. Nesne Depolamanın Diğer Depolama Biçimleri ile Karşılaştırılması

Nesne, blok ve dosya depolama, verileri depolamada farklı yöntemler kullanan depolama biçimleridir. Veri; nesne depolamada nesne, blok depolamada blok, dosya depolamada ise dosya olarak depolanmaktadır. Nesne depolama düz bir yapıya sahiptir, blok depolama son derece yapılandırılmıştır, dosya depolama ise hiyerarşik olarak yapılandırılmıştır. Bu üç depolama biçimi, farklı tasarımları gereği değişik yeteneklere sahiptir.

**Tablo 5**

*Dosya – Blok – Nesne Depolamanın Karşılaştırılması*

| Parametre | Dosya Depolama | Blok Depolama | Nesne Depolama |
|---|---|---|---|
| Ölçeklenebilirlik | Sınırlı Ölçeklenebilirlik | Yüksek | Oldukça Yüksek |
| Performans | Düşük Performans | Yüksek | Sadece Geri Almada Yüksek Performans |
| Maliyet | Düşük | Yüksek | Düşük |
| Güvenlik | İşletim Sistemine Bağlı | Zayıf | Yüksek |

A Survey on Different Storage Architectures. Jyoti vd., 2021, s. 795.

Tablo 5'te dosya, blok ve nesne depolama biçimleri karşılaştırılmaktadır. Depolama biçimlerinin üçü de incelendiğinde, tüm kullanım senaryoları için en iyi depolama mimarisinin mevcut olmadığı görülmektedir. Her mimarinin kendine özgü avantajları ve dezavantajları bulunmaktadır ve belirli bir dizi probleme hitap edecek şekilde tasarlanmışlardır. Uygun depolama seçeneğini seçmeden önce, depolama seçeneğinin gereksinimleri karşılayıp karşılamadığını değerlendirmek gerekmektedir. Eğer NAS cihazları ile veriler düzenli bir şekilde saklanmak isteniyorsa, dosya depolama muhtemelen en iyi seçenektir. Verilere sık erişim gerekiyorsa, blok depolama iyi bir tercih olabilir. Büyük miktarda depolama gereksinimi bulunması ve verilerin gelecekte ölçeklenmesi beklenmesi durumda ise en verimli sonucu nesne depolama sunabilir.

### 3.9. Nesne Depolama ve Zekâ

Yeni teknolojilerin ortaya çıkışı ve veri miktarındaki hızlı artış, depolama sistemlerinin giderek daha karmaşık hâle gelmesine neden olmaktadır. Depolama sistemlerinin karmaşıklığının artması, bu sistemlerin yönetimini zorlaştırmaktadır. Bu sorunu çözmek ve depolama sistemleri yönetimini optimize etmek amacıyla yapay zekâ tabanlı akıllı depolama sistemlerinin geliştirilmesi faydalı olabilir (Amyrosiadis vd., 2018).

Akıllı depolama cihazları, çalışma ortamlarını anlayarak kaynakları daha etkin bir şekilde tahsis edebilir ve performanslarını artırabilir. Nesne depolama, depolama cihazlarının çalışma ortamlarını öğrenme ve bu ortama uygun hareket etme potansiyeline sahiptir. Geleneksel depolama mimarileri, cihazlar, kullanıcılar ve uygulamalar hakkında

sınırlı bilgi sunarken, nesne depolama cihazları veriyi yönetme ve düzenleme konusunda daha kapsamlı bilgilere erişim sağlamaktadır (Mesnier vd., 2003).

Nesne öznitelikleri, depolama sistemlerinin veriyi akıllı bir şekilde düzenlemesine ve yönetmesine olanak tanımaktadır (Jain, 2007). Beklenen okuma/yazma oranı, en olası erişim desenleri (ardışık veya rasgele) ve nesnenin beklenen ömrü gibi bilgiler, depolama sisteminin performansını iyileştirmede kullanılabilir. Örneğin, düzenli yedeklemeler gerçekleştiren bir veri tabanı için yedekleme zamanları gibi özellikler, depolama cihazlarının bu desenleri öğrenmesine ve optimizasyon yapmasına yardımcı olabilir. Ayrıca, bir nesnenin oluşturulma zamanını bilen bir depolama cihazı, bu nesnenin büyüklüğünü tahmin ederek kaynaklarını buna göre tahsis edebilir. Akademik araştırmalar, nesne depolamanın davranış sınıflandırmada önemli bir rol oynayabileceğini göstermiştir (Mesnier vd., 2003).

Schlosser ve Iren (2005), veri tabanı depolama yöneticileri ile depolama alt sistemleri arasında nesne depolama arayüzünün kullanılmasını önermiştir; çalışmada, geleneksel depolama mimarilerindeki sistemlerin yeterince bilgiye sahip olmadığı ve kurallara dayalı çalıştığı belirtilmektedir. Veri tabanı yöneticileri ile depolama alt sistemleri arasında daha iyi bir iletişim sağlamak için nesne depolama cihazı arayüzlerinin daha fazla bilgiye erişim sağladığı ifade edilmektedir. Zeng vd. (2004) tarafından yapılan çalışmada, nesne özniteliklerinin akıllı nesne depolama sistemlerinde bir nesnenin nasıl erişileceği ve yönetileceği konusunda güçlü ipuçları sunduğu gösterilmiştir. Yue vd. (2007) tarafından yapılan çalışmada ise akıllı rol tabanlı nesne yönetimi ele alınmış; farklı görevlerin belirli nesneler tarafından gerçekleştirilmesi gerektiği vurgulanmıştır.

# BÖLÜM 4

# NESNE DEPOLAMA UYGULAMALARI VE ÇÖZÜMLERİ

Nesne depolama, geniş bir kullanım yelpazesine sahip olup çeşitli uygulama alanlarına sahiptir. Nesne tabanlı dosya sistemleri, bulut depolama, arşiv depolama ve nesne depolama sistemleri, bu kullanım alanları arasında yer almaktadır. Bu bölümde, nesne depolamanın uygulama alanları ve bu alanlarla ilgili örnek çözümler ele alınmıştır.

## 4.1. Nesne Tabanlı Dosya Sistemleri

Nesne depolamadaki dosya sistemleri, "nesne tabanlı dosya sistemi" olarak tanımlanmaktadır. Dosya sistemleri, genellikle dağıtık dosya sistemleri ve kümelenmiş dosya sistemleri olarak iki ana kategoriye ayrılmaktadır (Mondal vd., 2024).

Kümelenmiş dosya sistemlerinde, ortak dosya depolama sunucuları arasında veri dağıtılmakta ve tüm sunucular tek bir dosya sistemi gibi işlev görmektedir. Bu sistemlerde, depolama alanı sunucular arasında paylaşılır ve bu durum, daha verimli bir depolama alanı kullanımı ile sistem performansını artırmaktadır (Mondal vd., 2024).

Dağıtık dosya sistemlerinde ise veri dosyaları, farklı coğrafi konumlarda bulunan ancak tek bir dosya sistemi gibi davranan sunucular arasında dağıtılmaktadır. Sunucular, internet aracılığıyla birbirine bağlanmakta ve uzak makine çağrıları ile erişim sağlanmaktadır. Bu yaklaşım, ölçeklenebilirlik ve hata toleransı gibi avantajları mümkün kılmaktadır. Modern teknolojiler, büyük verilerin etkili bir şekilde depolanmasını desteklemektedir. Ceph, dağıtık dosya sistemlerine bir örnek olarak verilebilir (Mondal vd., 2024).

### 4.1.1. Red Hat Ceph

Red Hat Ceph depolama, performans, güvenilirlik ve ölçeklenebilirlik sağlamak amacıyla tasarlanmış dağıtılmış bir nesne depolama sistemidir ve açık kaynak kodlu bir

yapıya sahiptir. Ceph'in temel bileşenlerini içeren dağıtımı, Ceph Depolama Kümesi olarak adlandırılır ve Şekil 6'da da görüldüğü üzere bu küme üç ana bileşenden oluşmaktadır (Red Hat, 2024):

- Ceph OSD (Object Storage Daemon): Ceph OSD'leri verilerin depolanmasından sorumlu olup, aynı zamanda replikasyon, silme kodlaması (EC), yeniden dengeleme, kurtarma, izleme ve raporlama işlevlerini yerine getirir.
- Ceph monitör: Ceph monitör, küme haritasının güncel durumunu takip eder ve Paxos protokolü kullanarak küme üzerindeki değişikliklerin çoğunluk tarafından onaylanmasını sağlar.
- Ceph yöneticisi: Ceph yöneticisi, Ceph monitörün işlevlerini destekler ve küme performansını artırır. Ayrıca, RESTful izleme API'si sunarak küme durumu hakkında bilgi sağlar.

**Şekil 6**

Red *Hat Ceph Depolama Kümesi Bileşenleri* Red Hat Ceph Storage 6 Architecture Guide. Red Hat, 2024, s.3.




Ceph, güvenlik amacıyla Cephx kimlik doğrulama sistemini kullanmaktadır. Cephx, kimlik doğrulama için paylaşılan gizli anahtarlar kullanır ve hem istemciler hem de monitörler gizli anahtarın bir kopyasına sahiptir. Kimlik doğrulama protokolü, iki tarafın da anahtarın bir kopyasına sahip olduğunu kanıtlamasını sağlar, ancak anahtar doğrudan ifşa edilmez. İstemciler, Ceph monitöründen bir küme haritası alarak nesneler üzerinde giriş/çıkış işlemleri gerçekleştirebilmektedir. Küme haritası, tüm monitörler ve OSD'lerin durumları hakkında bilgi verir; ancak, nesnelerin konumları hakkında bilgi içermemektedir (Red Hat, 2024).

Ceph Depolama Kümesi, verileri "havuzlar" (pools) olarak adlandırılan mantıksal bölümlerde saklar. Ceph yöneticileri, belirli veri türleri için havuzlar oluşturabilmektedir. Yerleşim grupları, Ceph'te nesne depolamayı kolaylaştırmak ve daha verimli bir yönetim

sağlamak amacıyla kullanılmaktadır. Yerleşim grubu, bir havuzun alt kümesidir ve bir nesne koleksiyonu içermektedir. Ceph, havuzlara CRUSH kural setleri atar ve havuzun CRUSH kural seti ile yerleşim gruplarının sayısı, verilerin nasıl yerleştirileceğini belirleyen temel faktörlerdir (Weil, 2007).

Ceph, veri güvenliğini replikasyon ve silme kodlaması yöntemleri ile sağlamaktadır. Replikasyon, verilerin birden fazla kopyasını oluşturarak donanım arızalarına karşı korumaktadır. Silme kodlaması ise verileri parçalara ayırarak bu parçaları birden fazla OSD'de saklar ve veri kaybı riskini azaltmaktadır (Red Hat, 2024).

## 4.2. Bulut Nesne Depolama

Bulut depolama sağlayıcıları, nesne depolama mimarisini arka planda soyutlayan platformlar sunar; ancak yalnızca bazı bulut sağlayıcıları nesne depolama hizmeti sağlamaktadır. Verilere erişim, genellikle kullanıcı arayüzü veya REST API aracılığıyla gerçekleştirilmektedir. Ayrıca, bazı Java kütüphaneleri de bulutta depolanan nesnelere erişim sağlamak için kullanılmaktadır. Amazon S3 ve EMC Elastic Cloud Storage (ECS), bulut tabanlı nesne depolama çözümlerine örnek olarak verilebilir (Mondal vd., 2024).

### 4.2.1. Amazon Simple Storage Service (S3)

Amazon Simple Storage Service, ölçeklenebilirlik, veri erişilebilirliği, güvenlik ve yüksek performans sunan bir nesne depolama hizmetidir. Amazon S3, veri gölleri, mobil uygulamalar, internet siteleri, yedekleme ve geri yükleme, kurumsal uygulamalar, arşivleme, büyük veri analitiği ve nesnelerin interneti cihazları gibi çeşitli kullanım durumlarını desteklemektedir. Bu hizmet, kullanıcıların iş ve uyumluluk gereksinimlerini karşılamak için verilere erişimi yapılandırmalarını, optimize etmelerini ve düzenlemelerini sağlamaktadır (Perry, 2021).

S3'te veriler, "kova (bucket)" olarak adlandırılan konteynerlerde nesne olarak depolanmaktadır. Nesne, S3'ün temel birimi olup veri ve meta verileri içermektedir. Kovalar, nesnelerin depolandığı konteynerler olduğu için, kullanıcıların nesneleri depolamadan önce kovaları oluşturması gerekmektedir. Kullanıcılar, bir veya birden fazla

kova oluşturabilir ve kova oluştururken kovanın coğrafi konumunu seçebilmektedir. Kovalar oluşturulduktan sonra, kullanıcılar bu kovaya verileri içeren nesneleri ekleyebilmektedir. Her nesnenin, S3 ortamında onu benzersiz şekilde tanımlamak için kullanılan bir anahtarı vardır. Nesnelere erişim, HTTP protokolünü destekleyen herhangi bir araç veya tarayıcı aracılığıyla sağlanabilmektedir (AWS, t.y.a).

Nesneler, ilişkili oldukları depolama sınıfına göre depolanmaktadır. S3'ün sunduğu depolama sınıfları şunlardır (AWS, t.y.a):

- S3 Akıllı Katmanlama (S3 Intelligent-Tiering): Bilinmeyen veya değişken erişim düzenlerine sahip verilerde otomatik maliyet tasarrufu sağlar.
- S3 Standart (S3 Standard): Sık erişilen veriler için uygundur.
- S3 Hızlı Tek Alan (S3 One Zone-IA): Kurumsal olarak sık erişilen veriler için tasarlanmıştır.
- S3 Standart – Seyrek Erişim (S3 Standard-IA) ve S3 Tek Alan – Seyrek Erişim (S3 One Zone-IA): Daha az sıklıkla erişilen veriler için uygundur.
- S3 Glacier Anında Veri Alma (S3 Glacier Instant Retrieval): Anında erişim gerektiren arşiv verileri için tasarlanmıştır.
- S3 Glacier Esnek Veri Alma (S3 Glacier Flexible Retrieval): Anında erişim gerektirmeyen, nadiren erişilen uzun süre saklanan veriler için uygundur.
- Amazon S3 Glacier Derin Arşiv (S3 Glacier Deep Archive): Bulutta düşük depolama maliyetleriyle uzun süreli arşivleme ve dijital olarak korunması gereken veriler için tasarlanmıştır.
- S3 Glacier Derin Arşiv (S3 Glacier Deep Archive): En düşük maliyetli S3 sınıfıdır ve veriler en az 12 saatte geri alınabilir.

Veriler, yöneticinin tanımladığı kurallara göre bir sınıftan diğerine taşınabilmektedir. Örneğin, S3 Standart'taki veriler belirli bir süre sonra S3 Standart-IA'ya, daha sonra S3 Glacier'e taşınabilir ve istenirse Glacier'de belirli bir süre tutulan veriler silinebilir (Kingsley, 2023).

S3, güvenlik sağlamak amacıyla verileri hem dinlenme hem de iletim sırasında şifrelemektedir. İki tür şifreleme yöntemi mevcuttur (Surianarayanan ve Chelliah, 2023):

- Sunucu tarafı şifreleme (Server-Side Encryption): Veri, depolanmadan önce şifrelenir ve veriler kullanıcıya sunulmadan önce şifre çözülür.
- İstemci tarafı şifreleme (Client-Side Encryption): Şifreleme ve şifre çözme sorumluluğu kullanıcıya aittir.

**Şekil 7**

*Amazon S3 Çalışma Diyagramı* Amazon S3 Nasıl Çalışır? AWS, t.y.




Şekil 7, verilerin Amazon S3'e nasıl taşınacağını, Amazon S3'te depolanan verilerin nasıl yönetileceğini ve diğer hizmetlerle nasıl analiz edileceğini göstermektedir. Şekilde soldan sağa üç bölüm görüntülenmektedir. Birinci bölüm, "verileri taşıma" bölümüdür; veriler (analiz verileri, günlük verileri, uygulama verileri, video ve resimler, yedek ve arşiv vb.) bulunduğu yerden (buluttan, uygulamalardan veya şirket içinden) Amazon S3'e taşınmaktadır. İkinci bölüm, "Amazon S3" bölümüdür; her yerden, her boyutta veri almak ve depolamak için oluşturulmuş bir nesne depolama hizmetini içermektedir. Bu depolama hizmetinin verilere erişimi denetleme, depolama sınıflarıyla maliyeti optimize etme, verileri herhangi bir bölgeye çoğaltma, şirket içinden veya Sanal Özel Bulut (VPC)'tan erişme, verileri koruma ve güvenceye alma, depolama alanına yönelik görünürlük sağlama özellikleri bulunmaktadır. Üçüncü bölüm ise "verileri analiz etme" bölümüdür; öngörüler elde etmek için yapay zekâ, gelişmiş analiz ve makine öğrenimi teknolojileri aracılığıyla verileri analiz etmek üzere AWS ve üçüncü taraf hizmetler kullanılabilmektedir. (AWS, t.y.b)

Mevcut nesne depolama sektörü, nesne depolama protokollerini sağlayan tedarikçiler ve bu protokolleri kullanan uygulamalar gibi tüketiciler olmak üzere iki ana grupta değerlendirilmektedir. Amazon S3 API, nesne depolama için standart bir protokol haline gelmiştir. Yaygın olarak kullanılan bu protokol, geniş bir geliştirici topluluğu oluşturmuş ve kurumsal veri merkezlerinde nesne depolama platformları kuran satıcılar tarafından benimsenmiştir.

### 4.2.2. EMC Elastic Cloud Storage (ECS)

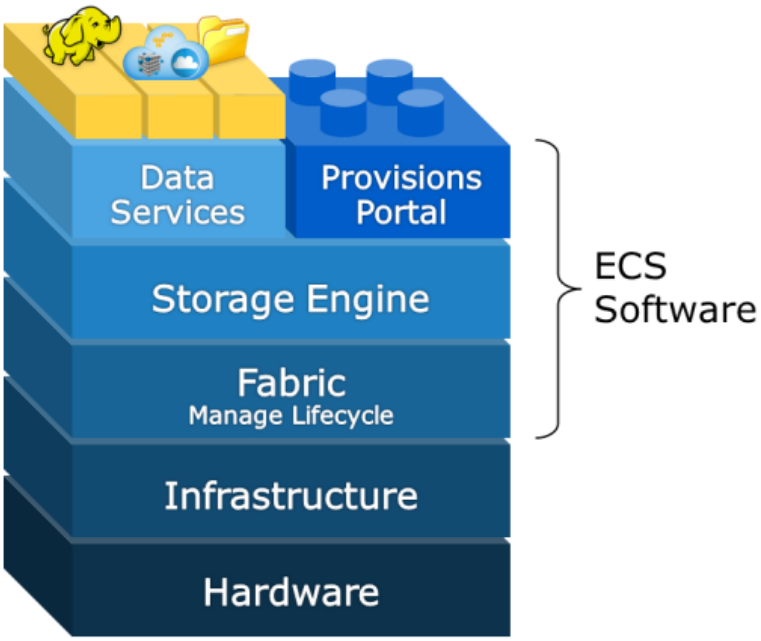
EMC Elastic Cloud Storage, yazılım tanımlı, ölçeklenebilir ve bulut tabanlı bir nesne depolama platformudur. ECS, tek bir küresel ad alanı altında küresel olarak dağıtılmış bir depolama altyapısını yönetme yeteneği sağlamaktadır. ECS'nin temel bileşenleri, esneklik ve dayanıklılığı artırmak amacıyla katmanlı bir mimari ile tasarlanmıştır. Her katman soyutlanmıştır ve bağımsız olarak ölçeklenebilmektedir. Depolama servislerine erişim, RESTful API'ler aracılığıyla sağlanmaktadır (Dell Technologies, 2024).

ECS mimarisinin tasarımında, küresel ad alanı ile tutarlılık, ölçeklenebilirlik, güvenli çoklu kiracılık ve hem küçük hem de büyük nesneler için yüksek performans gibi temel ilkeler uygulanmaktadır. ECS, nitelikli endüstri standardı donanım setleri veya teslim edilmeye hazır depolama cihazları üzerine dağıtılabilmektedir. Şekil 8'de ECS mimarisinin katmanları gösterilmektedir (Dell Technologies, 2024):

- ECS portal ve sağlama hizmetleri: ECS düğümlerinin otomasyon, raporlama ve yönetim işlemlerini gerçekleştiren arayüzdür. Bu katman, lisanslama, kimlik doğrulama, çoklu kiracılık ve ad alanı oluşturma gibi sağlama hizmetlerini de yönetir.
- Veri hizmetleri: Bu bileşen, nesne ve dosya erişimini desteklemek amacıyla kullanılan araçlar ve API'leri içerir.
- Depolama altyapısı: Veri depolama ve alma işlemlerinden sorumlu olan temel hizmettir. İşlemleri yönetir, verileri yerel olarak ve site'lar arasında korur ve çoğaltır.
- Yapı: Sistemin sağlık durumu, yapılandırılması, yükseltilmesi ve uyarılar için kullanılan bileşendir.
- Altyapı: Linux işletim sistemleri kullanılarak sağlanır.
- Donanım: Teslim edilmeye hazır cihazlar veya nitelikli endüstri standardı donanımlardan oluşur.

**Şekil 8**

*EMC ECS Mimarisinin Katmanları* ECS Overview and Architecture. Dell Technologies, 2024, s.9




ECS tasarımı, çeşitli kullanım durumları için optimize edilmiştir. Modern uygulamalar, ikincil depolama, coğrafi korumalı arşiv, küresel içerik deposu, nesnelerin interneti için depolama, veri gölleri ve analitik uygulamalar bu kullanım durumlarına örnek olarak verilmektedir (Dell Technologies, 2024).

### 4.3. Arşiv Depolama

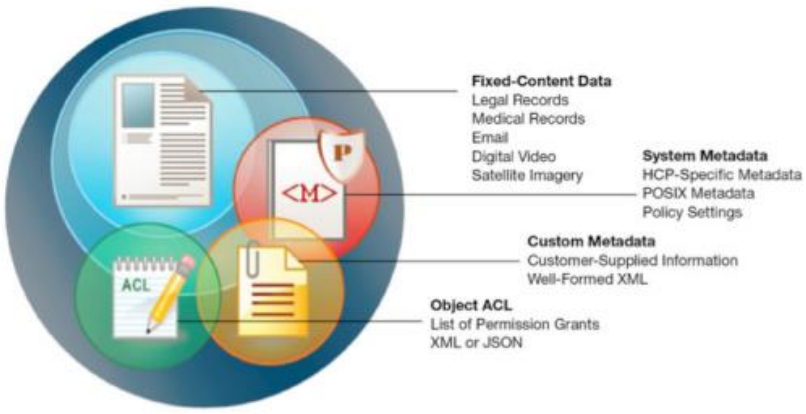
Arşiv tipi depolamada, büyük veri hacminin uzun süreli ve belirsiz bir süre boyunca yönetimine odaklanılmaktadır. Bu tür bir depolama sisteminde, zaman ölçeği sabit olduğundan, donanım yapılandırması ve mimarisi, mümkün olan en düşük maliyetle gerçekleştirilmektedir. Performans metrikleri, ölçeklenebilirlik kolaylığı, saklama verimliliği, maliyet etkinliği, replikasyon mekanizması ve güvenlik politikası gibi çeşitli faktörlere dayanmaktadır. Bu gruba örnek olarak Hitachi Data Systems tarafından sunulan Hitachi Content Platform (HCP) gösterilebilir (Mondal vd., 2024).

### 4.3.1. Hitachi Data Systems-Hitachi Content Platform (HCP)

Hitachi Content Platform (HCP), bir nesne depolama platformu olarak işlev görmektedir. HCP, geleneksel dosya depolama yöntemlerine kıyasla daha büyük veri setlerini yönetme ve işleme kapasitesine sahiptir. HCP yazılımı, yapılandırılmamış verileri REST protokolü aracılığıyla alarak nesne şeklinde depolamaktadır. Tüm nesneler, ad alanlarına atanmış politikalara uygun olarak kullanılabilir depolama havuzlarına dağıtılmaktadır. Nesneler, standart dizin yapısındaki bir dosya kümesi olarak ya da kullanıcılar tarafından HTTP veya HTTPS protokolü aracılığıyla erişilebilen URL'ler olarak sunulmaktadır. Şekil 9'da gösterildiği üzere HCP'de nesne, sabit içerikli veri ve bu veriyi tanımlayan meta verilerden oluşmaktadır. Meta verilerin tamamı görüntülenebilirken, yalnızca bir kısmı kullanıcı tarafından değiştirilebilmektedir. Meta verilerin görüntülenmesi ve değiştirilmesi, ad alanının yapılandırılmasına, veri erişim protokolüne ve meta veri türüne bağlıdır. HCP meta veri türleri; sistem meta verileri, opsiyonel özel meta veriler ve erişim kontrol listesi (ACL) olarak sınıflandırılmaktadır (Hitachi Vantara, 2020).

**Şekil 9**

*HCP'de Nesne Yapısı* Hitachi Content Platform Architecture Fundamentals. Hitachi Vantara, 2020, s.19.




Sabit içerikli veriler, yazılı bir dosyanın bir hash algoritması (MD5, SHA-1, SHA-256 (varsayılan), SHA384, SHA-512 veya RIPMD160) kullanılarak oluşturulan "parmak

izi" dijital kopyası olarak tanımlanmaktadır. Dosyalar, sanal depolama havuzuna kaydedildikten sonra değiştirilememektedir. Eğer nesne saklama altında ise, saklama süresi dolmadan silinememektedir. Versiyonlama etkinleştirilmişse, bir dosyanın birden fazla versiyonu saklanabilmektedir (Hitachi Vantara, 2020).

Sistem meta verisi, 28 özelliği içermektedir. Bu özellikler arasında nesnenin ad alanına eklendiği tarih ve saat (alım zamanı), nesnenin son değiştirildiği tarih ve saat (değişim zamanı), nesnenin kriptografik hash değeri ve bu değeri oluşturmak için kullanılan ad alanı hash algoritması, nesnenin alındığı protokol değeri yer almaktadır. Ayrıca, nesnenin politika ayarlarını da içerir; bu ayarlar arasında yedek kopya sayısı, saklama, imha, dizinleme ve versiyonlama gibi özellikler bulunmaktadır. POSIX meta verisi, bir kullanıcı kimliği (user ID), grup kimliği (group ID), POSIX izin değeri ve POSIX zaman özelliklerini içermektedir (Hitachi Vantara, 2020).

Özel meta veri, genellikle iyi yapılandırılmış XML formatında sağlanan, kullanıcı tarafından eklenen açıklayıcı bilgilerdir ve isteğe bağlıdır. Bu meta veri, nesne hakkında daha fazla açıklayıcı detay eklemek için kullanılmaktadır. Kullanıcılar ve uygulamalar, özel meta veriyi nesne içeriğini anlamak ve yeniden kullanmak için kullanabilmektedir. HCP, her nesne için birden fazla özel meta veri alanını desteklemektedir (Hitachi Vantara, 2020).

Nesne Erişim Denetim Listesi'si (ACL), kullanıcılara veya kullanıcı gruplarına nesne üzerinde işlem yapma yetkisi veren bir dizi izin içeren kullanıcı tarafından sağlanan meta veridir ve isteğe bağlıdır. ACL'ler, veri erişimini bireysel nesne düzeyinde kontrol etmektedir (Hitachi Vantara, 2020).

HCP, sabit içerikli dosya verilerini meta verilerden ayrı olarak ve paralel veri yapılarında depolamaktadır. HCP erişim düğümleri, içeriğin nerede depolandığını bilmekten sorumlu olup, ölçeklenebilirlik amacıyla HCP düğümleri arasında dağıtılmış bir hash indeksini de sürdürmektedir. Hash indeksi, verileri bulmak için içerik adreslenebilir arama işlevini sağlamaktadır. Her düğüm, veriyi ve meta veriyi bulmak için belirli bir bölgeyi takip etmektedir. Yeni bir dosya alındığında, herhangi bir düğüm, atanmış hizmet alanı planına göre sabit içerik dosyasını sahip olduğu depolama alanına yazabilmektedir. Ardından, dosya yolunun bir hash'ini hesaplayarak bunu nesnenin meta verilerine ekler ve nesnenin hash indeks bölgesini takip eden düğüme iletmektedir. HCP, indeksini ikinci seviye meta veri koruma (MDLP2) ile korur; bu, iki kopyanın saklanacağı anlamına gelir ve kopyalar farklı düğümlerde tutulmaktadır. Bir yetkili kopya ve en az bir yedek kopya

bulunmaktadır. Gerçek dosya, kiracı yönetici tarafından tanımlanan bir depolama havuzunda saklanmaktadır (Hitachi Vantara, 2020).

HCP yazılımı, sistemin genel sağlığını iyileştirmek, verimliliği optimize etmek ve saklanan verilerin bütünlüğünü ve erişilebilirliğini korumak amacıyla arka plan hizmetlerini uygulamaktadır. Bu hizmetler; sürekli olarak, belirli bir takvime göre periyodik olarak veya belirli olaylara yanıt olarak çalıştırılabilmektedir. Bu hizmetlerden bazıları şunlardır (Hitachi Vantara, 2020):

- Kapasite dengeleme: Sistemdeki tüm erişim düğümleri arasında kullanılabilir birincil depolama kapasitesini dengede tutmayı amaçlar. Düğümler arasındaki depolama kullanımı geniş bir farkla çalışıyorsa, hizmet nesneleri taşıyarak düğümleri dengeler.
- Sıkıştırma: Nesne verilerini fiziksel depolama alanını daha verimli kullanacak şekilde sıkıştırır.
- İçerik doğrulama: Depo nesnelerinin veri bütünlüğünü, bir dosyanın dijital hash imzası ile eşleşmesini doğrulayarak sağlar. Hash eşleşmezse, HCP nesneyi onarır ve meta veri tutarsızlıklarını tespit edip düzeltir.
- Yinelenme önleme: Depodaki yinelenen nesneleri tanımlar ve kopya verileri birleştirerek alan boşaltır.
- İmha: Süresi dolmuş nesnelerin otomatik olarak temizlenmesini sağlar.
- Hızlı nesne kurtarma: Kullanılamayan nesnelerin kurtarıldığında durumlarını kullanılabilir olarak değiştirir.
- Çöp toplama: Silinmek için işaretlenmiş nesnelerin gizli veri ve meta verilerini temizleyerek depolama alanını geri kazandırır. Ayrıca, imha işlemlerini de gerçekleştirebilir.
- Taşıma: Verileri seçilen düğümlerden veya Hitachi veri dizinlerinden taşır.
- Koruma: Veri koruma seviyesi politika uyumunu uygular ve her nesnenin sistemde uygun sayıda kopyasının bulunduğundan emin olur.
- Yinelenme: Bir HCP sisteminden diğerine bir veya daha fazla kiracıyı kopyalayarak veri erişilebilirliğini sağlar ve felaket kurtarma yeteneği sunar.
- Depolama katmanlama: Bir nesne için hangi depolama katmanlama stratejisinin geçerli olduğunu belirler.
- Coğrafi dağıtılmış silme kodlaması (Geo-EC): HCP, üç veya daha fazla siteyi kapsadığında coğrafi dağıtılmış silme kodlaması uygulayabilir. Bu teknoloji, tüm

site seviyesindeki kesintilere rağmen veri erişilebilirliğini sağlar. Geo-EC dağıtımları, basit yansıtma yinelenmesi ile dağıtılan sistemlerden %25-40 daha az depolama tüketir.

## 4.4. Nesne Depolama Sistemleri

Nesne Depolama Sistemleri (OSS), organizasyonlar tarafından kendi kurumsal altyapılarında veya çeşitli bulut sağlayıcılarının platformlarında kullanılan yazılım çözümleridir. OSS, diğer bulut depolama sağlayıcılarıyla aynı yüksek performans ve verimliliği sunmayı hedeflemektedir. Bu tür sistemler, verilerin ölçeklenebilir bir şekilde depolanması, yönetilmesi ve erişilmesini sağlamaktadır. NetApp StorageGRID Webscale ve OpenStack Swift, nesne depolama sistemlerine örnek olarak verilebilir (Mondal vd., 2024).

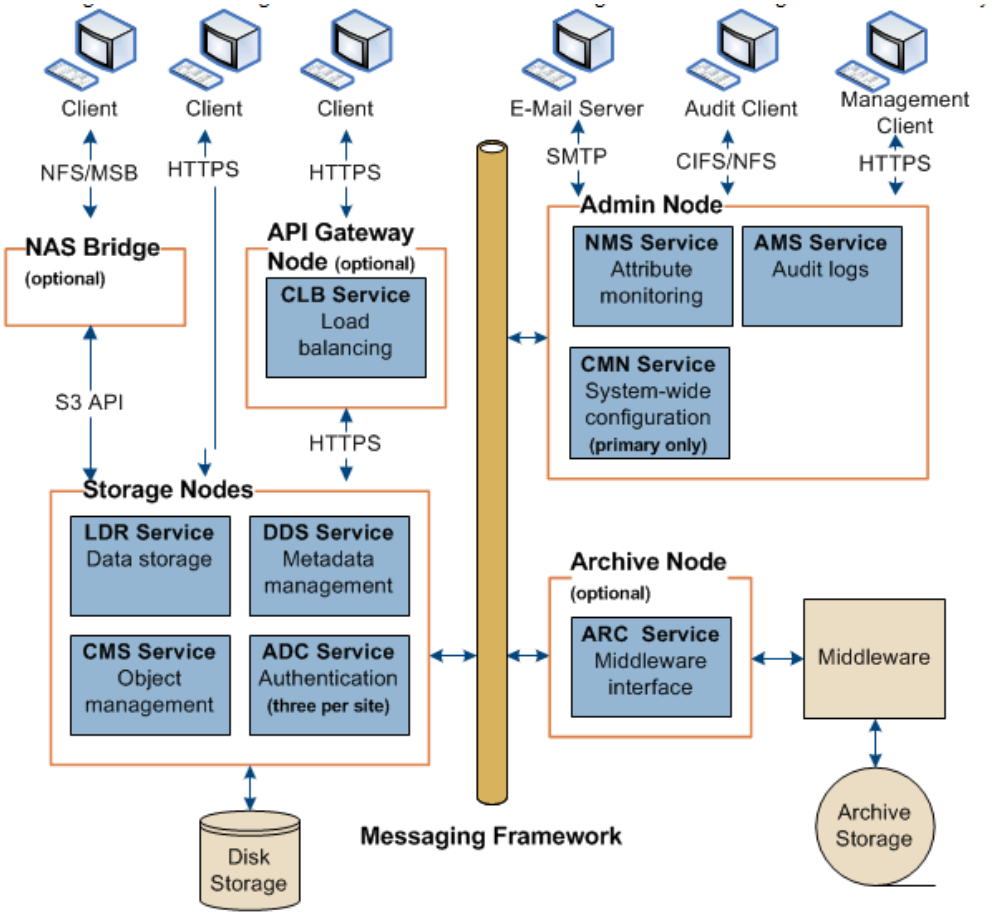
### 4.4.1. NetApp StorageGRID WebScale

StorageGRID Webscale, sabit içerikli verileri depolayan ve koruyan dağıtık bir nesne depolama sistemidir. Bu sistem, verilerin kopyalarını küresel ölçekte dağıtarak yüksek derecede güvenilirlik sağlar ve verilerin sürekli olarak erişilebilir durumda kalmasını temin etmektedir. StorageGRID Webscale sisteminin temel bileşenleri, Şekil 10'da da görüldüğü üzere dört farklı türde düğümden oluşmaktadır. Her düğüm türü, belirli hizmetleri barındırır ve özel bir işlem seti gerçekleştirmektedir (NetApp, 2016):

- Yönetici düğümü: Sistem yapılandırması ve izleme gibi yönetim hizmetlerini sağlar.
- Arayüz geçidi düğümü: Uygulamaların sisteme bağlanabilmesi için yük dengeleme arayüzü sunar ve isteğe bağlıdır.
- Arşiv düğümü: Nesne verilerinin uzun vadeli depolama için harici arşivleme sistemleri (bant veya bulut) ile arşivlenmesini sağlayan bir arayüz sunar.
- Depolama düğümü: Nesne verilerini ve meta verilerini yönetir, ayrıca veri kaybına karşı koruma işlevi görür.

**Şekil 10**

*StorageGRID Webscale Temel Bileşenleri* What grid nodes and services are. NetApp, 2016.




Sistem, nesneleri yönetmek için meta veri tabanlı Bilgi Yaşam Döngüsü Yönetimi (ILM) kurallarını kullanmaktadır. ILM kuralları, bir nesnenin verilerinin sisteme alındıktan sonraki sürecini belirler; verilerin nerede depolanacağı, nasıl korunacağı ve ne kadar süreyle saklanacağı gibi kriterleri içermektedir. Nesnelere erişim, nesne depolama API'leri (S3, OpenStack Swift, CDMI), geleneksel dosya sistemi arayüzleri (NFS, SMB) veya her ikisi aracılığıyla sağlanabilmektedir (NetApp, 2016).

StorageGRID Webscale sistemi, donanım bağımsız yazılım tanımlı depolama yönetimi sunar ve donanım bağımlılığı bulunmamaktadır. Standart nesne depolama ve dosya depolama protokollerini desteklemektedir. Politika bazlı katmanlı depolama sağlar, böylece genel depolama maliyetlerini azaltabilmektedir. Ayrıca, uygulamaları temel depolama altyapısındaki değişikliklerden izole etmek amacıyla siteler ve katmanlar arasındaki depolamayı sanallaştırmaktadır (NetApp, 2016).

### 4.4.2. OpenStack Swift

OpenStack Swift, büyük miktarda yapılandırılmamış veriyi uygun maliyetle depolamak üzere tasarlanmış bir nesne depolama sistemidir. Küçük işletmelerden büyük ölçekli servis sağlayıcılarına ve araştırma kuruluşlarına kadar geniş bir kullanıcı yelpazesine hitap eden açık kaynak kodlu bir çözümdür. Swift, nesneleri ve nesnelere ait meta verileri konteynerlerde depolar ve bu verilere erişim, RESTful API'ler aracılığıyla sağlanmaktadır (Openstack, 2024).

Swift'in veri modeli, hesap, konteyner ve nesne olmak üzere üç ana bileşene sahiptir. Bu bileşenlerin biri veya daha fazlası kullanılarak veriler için benzersiz bir depolama yeri sistem tarafından oluşturulmaktadır (Openstack, 2024):

- Hesap: Hesaplar, veriler için kök depolama alanıdır ve sistemde benzersiz olarak adlandırılır. Her hesap, hem hesap meta verilerini hem de hesap içindeki tüm konteynerlerin listesini barındırır.
- Konteyner: Konteynerler, kullanıcı tarafından tanımlanan hesap içindeki depolama alanlarıdır. Her konteyner, konteyner hakkındaki meta verileri ve içindeki nesnelerin listesini içerir.
- Nesne: Nesne, verilerin ve meta verilerin depolandığı alandır. Bu bileşenler birleştirilerek konumlar oluşturulur, bu nedenle konteyner ve nesne adlarının küme içinde benzersiz olması gerekmez.

Swift verileri (hesaplar, konteynerler, nesneler), fiziksel donanım üzerinde saklanan kaynaklardır. Swift süreçlerini çalıştıran fiziksel makineler “düğüm” olarak adlandırılır ve bu düğümler, Swift kümesinin dağıtılmış bir depolama sistemi olarak işlev görmesi için gerekli tüm süreçleri ve hizmetleri kolektif olarak çalıştırmaktadır (Openstack, 2024).

Dayanıklılığı sağlamak ve arızaları izole etmek amacıyla, bir kümedeki düğümler bölgeler ve kullanılabilirlik alanları olarak organize edilebilmektedir. Bölgeler, fiziksel olarak ayrı olan küme parçalarıdır ve genellikle coğrafi sınırlarla tanımlanmaktadır. Örneğin, bir bölge, kümedeki diğer düğümlerden uzak, yüksek gecikmeli bir dış mekânda yerleştirilmiş birkaç sunucu rafı (düğüm) için tanımlanabilir. Arızaları izole etmek için bölgelerin içinde kullanılabilirlik alanları yapılandırılabilmektedir. Bir kullanılabilirlik alanı, diğer alanlarda izole edilmiş bir donanım kümesi ile tanımlanmaktadır. Örneğin, büyük bir dağıtımda, veri merkezindeki ayrı tesisler (güvenlik duvarıyla ayrılmış veya farklı enerji sağlayıcıları tarafından beslenmiş olabilir), kullanılabilirlik alanı olarak

tanımlanabilir. Tek bir veri merkezi dağıtımında ise kullanılabilirlik alanları farklı raflar olabilmektedir (Openstack, 2024).

Swift'in temel bileşenlerinden biri olan Ring, varlık isimlerini fiziksel konumlara haritalamaktan sorumludur. Swift sisteminde her varlık türü için ayrı bir ring bulunur ve her nesne için bir ring mevcut olup, nesnenin konumunu belirlemede kullanılmaktadır (Openstack, 2024).

Swift, verilerini yönetmek için çeşitli süreçler çalıştırır ve her düğüm bu süreçlerden bazılarını veya hepsini çalıştırabilmektedir. Yedeklilik ve dayanıklılık için genellikle her hizmetin birden fazla düğümde çalışması istenmektedir. Temel Swift sunucu süreçleri; proxy, hesap, konteyner ve nesnedir. Bir kümedeki düğümlerde çalışan toplu sunucu süreçlerine "sunucu süreçleri katmanı" denmektedir (örneğin, proxy katmanı, hesap katmanı) (Openstack, 2024).

- Proxy katmanı: Proxy sunucu süreci, Swift kümesinin dış istemcilerle iletişim kuran tek parçasıdır. İstemcilerden gelen okuma ve yazma taleplerini koordine eder ve sistemin okuma ve yazma garantilerini uygular.
- Hesap katmanı: Hesap sunucu süreci, bireysel hesaplar için meta verileri ve bir hesap içindeki konteynerlerin listesini sağlar. Hesap verilerinin kopyaları yapılır ve küme genelinde dağıtılır.
- Konteyner katmanı: Konteyner sunucu süreci, konteyner meta verilerini ve her konteyner içindeki nesnelerin listesini yönetir.
- Nesne katmanı: Nesne sunucu süreci, nesneleri düğümün sürücülerinde saklayabilen, alabilen ve silebilen bir BLOB depolama hizmeti sağlar.

# BÖLÜM 5

# SONUÇLAR VE ÖNERİLER

Tezin son bölümünde, araştırmanın ortaya koyduğu sonuçlar ele alınmış ve konuya ilişkin öneriler sunulmuştur. Ayrıca, nesne depolama teknolojilerinin RTÜK bilişim altyapısına uyarlanabilirliği ve sağlayabileceği katkılar değerlendirilmiştir.

## Sonuçlar

Tarih boyunca veri depolama teknolojileri, çağın şartlarına ve teknolojik ilerlemelere bağlı olarak köklü değişiklikler göstermiştir. Bu değişim süreci, verilerin doğrudan etkisi altında şekillenmiş ve mevcut depolama çözümleri, verilerin artan miktarına ve çeşitliliğine uyum sağlamak için evrim geçirmiştir. Özellikle büyük veri çağının başlangıcıyla birlikte, veri depolama teknolojileri daha dinamik ve ölçeklenebilir çözümler sunmak zorunda kalmıştır. Bu bağlamda, nesne depolama teknolojileri, yapılandırılmamış ve büyük miktardaki verileri verimli bir şekilde depolama kapasitesiyle önemli bir yer edinmiştir.

Bu tez, nesne depolama teknolojilerinin temel prensiplerini ve bu teknolojilerin veri depolama alanındaki evrimini incelemektedir. Nesne depolama, veri yönetiminde sunduğu esneklik, ölçeklenebilirlik ve yüksek performans özellikleri ile büyük veri ve bulut bilişim alanlarında önemli bir rol oynamaktadır. Tez kapsamında gerçekleştirilen araştırmalar, nesne depolamanın veri yönetimindeki avantajlarını ve kullanım senaryolarını ortaya koymaktadır.

Geleneksel dosya ve blok depolama sistemlerinden farklı olarak, nesne depolama, verileri nesne tabanlı bir yapıda yönetir ve her veriyi meta verilerle birlikte tek bir birim olarak saklamaktadır. Bu yapı, veri üzerinde hızlı erişim, yüksek ölçeklenebilirlik ve düşük maliyet avantajları sağlamaktadır. Özellikle büyük veri setlerinin ve yapılandırılmamış verilerin yönetiminde, nesne depolama teknolojisinin sağladığı bu özellikler, sistem performansını ve veri yönetim verimliliğini önemli ölçüde artırmaktadır.

Nesne depolama teknolojileri, özellikle veri yoğun uygulamalarda ve bulut hizmetlerinde önemli avantajlar sunmaktadır. Nesne depolama sistemleri, verilerin yatay olarak ölçeklenmesini sağlar ve bu sayede büyük veri kümeleri üzerinde etkili bir şekilde işlem yapılabilmektedir. Özellikle dinamik veri ortamlarında sistem performansını artırırken, maliyetleri düşürme noktasında da önemli bir katkı sağlamaktadır.

Nesne depolamanın sağladığı diğer bir önemli avantaj ise veri erişimindeki esnekliktir. Veriler ve ilişkili meta veriler tek bir nesne içinde saklandığından veri üzerinde kapsamlı sorgulama ve analiz yapmak daha kolay hâle gelmektedir. Bu özellik, büyük veri uygulamalarında veri entegrasyonu ve analizi süreçlerinde verimliliği artırmaktadır. Ayrıca, veri güvenliği ve erişim kontrolü konularında sunduğu çözümler, kullanıcıların verilerini daha güvenli bir şekilde yönetmelerine yardımcı olmaktadır.

Nesne depolama teknolojileri, veri depolama alanında önemli bir yenilik ve gelişim sürecini temsil etmektedir. Bu teknolojinin sağladığı yüksek verimlilik, ölçeklenebilirlik ve esneklik özellikleri, büyük miktardaki veri yönetimi ve işlenmesi konusunda büyük avantajlar sunmaktadır. Nesne depolamanın, özellikle bulut bilişim ve büyük veri uygulamalarındaki rolü, veri yönetimi ve depolama stratejilerinin geleceğini şekillendireceği değerlendirilmektedir.

## Öneriler

Tezde ele alınan konu ve kavramlar neticesinde yapılan değerlendirmeler sonucunda sunulan öneriler ve gelecek çalışmalara yönelik tavsiyeler, aşağıda sıralanmıştır:

- Yapay zekânın depolama teknolojilerinde kullanımı araştırılabilir ve bu alandaki potansiyel etkileri incelenebilir. Yapay zekânın veri yönetimi, otomasyon, güvenlik ve performans optimizasyonu konularındaki uygulamaları, depolama teknolojilerinin gelecekteki gelişimine önemli katkılarda bulunabilir. Dolayısıyla, yapay zekânın depolama sistemleri üzerindeki etkilerini anlamak, mevcut sistemlerin iyileştirilmesi ve yeni çözümlerin geliştirilmesi için önemli bir araştırma alanı olarak değerlendirilmektedir.
- Nesne depolamanın tasarım özellikleri, yapay zekâ teknolojileri ile entegre edilmesine uygun bir altyapı sunmaktadır. Nesne depolama sistemlerinin yapay

zekâ ile optimize edilmesinin, veri yönetimi ve analiz süreçlerinde önemli avantajlar sağlayabilir.

- Nesne depolama sistemlerinde meta veri yönetiminin performans üzerinde belirgin bir etkisi vardır. Meta veri yönetiminin optimize edilmesi, veri erişim hızını ve sistem verimliliğini artırma potansiyeline sahip olduğundan, bu alanın incelenmesi, nesne depolama çözümlerinin etkinliğini ve performansını önemli ölçüde iyileştirebilir.
- Nesne depolama teknolojilerinin büyük hacimli verileri depolamada oldukça verimli olduğu ancak küçük boyutlu verilerin depolanmasında yeterince etkili olmadığı görülmüştür. Nesne depolama sistemlerinin küçük veri kümeleriyle ilgili performansını artırmak üzerine yapılacak çalışmalar, nesne depolama teknolojilerinin genel verimliliğini ve kapsamını genişletebilir.
- Önde gelen dijital yayıncılık platformlarının, depolama altyapılarında nesne depolama teknolojilerinden yararlandığı ve bu teknolojilerin sektördeki öneminin giderek arttığı görülmüştür. Gelecekte, nesne depolama teknolojilerinin dijital yayıncılık sektöründe daha fazla yaygınlaşacağı öngörülmektedir. Nesne depolamanın bu sektördeki kullanımının incelenmesi, bu alandaki teknolojik gelişmeleri ve uygulama potansiyelini anlamak açısından araştırılabilir.
- Teknolojinin ilerlemesiyle birlikte veri yapılarının karmaşıklığının arttığı ve bu durumun veri depolama ve yönetim süreçlerini zorlaştırdığı görülmüştür. Bu bağlamda, etkin bir veri depolama ve yönetimi sağlamak amacıyla, kurumların teknolojik altyapılarında yeni teknolojilerle birlikte iyileştirmeler yapmasının yararlı olabilir.
- Nesne depolama teknolojilerinin arşivleme sistemlerinde yüksek verimlilik sağladığı gözlemlenmiştir. Nesne depolamanın, veri yönetiminde sunduğu esneklik ve ölçeklenebilirlik özellikleri, büyük miktardaki verilerin etkin bir şekilde saklanmasını ve yönetilmesini mümkün kılmaktadır. Kurumların arşivleme sistemlerinde nesne depolama teknolojilerini entegre etmeleri, arşivleme süreçlerinin verimliliğini artırabilir. Nesne depolamanın sunduğu avantajlar, özellikle uzun vadeli veri saklama gereksinimlerine sahip kurumlar için kritik öneme sahiptir. Bu teknoloji, veri bütünlüğünü korurken aynı zamanda maliyet etkin çözümler sunarak arşivleme maliyetlerini azaltabilir ve veri erişimini hızlandırabilir. Dolayısıyla, kurumların arşivleme stratejilerini optimize etmek ve

veri yönetiminde daha etkili sonuçlar elde etmek için nesne depolama teknolojilerinden faydalanılabilir.

- Nesne depolama, yedekleme ve felaket kurtarma sistemlerinde etkin bir şekilde çalışabilmektedir. Nesne depolama teknolojileri, yüksek hacimli veri yedekleme ve hızlı felaket kurtarma süreçleri için uygun bir çözüm sunduğundan bu alanlarda kullanımı değerlendirilebilir.
- Kapasite yönetimi, depolama sistemlerinde kritik bir öneme sahiptir. Bu konuda, nesne depolama teknolojilerinin esnek yapısı ve ölçeklenebilirliği sayesinde kapasite yönetimi süreçlerinde önemli avantajlar sağlayabilir.
- Nesne depolama teknolojileri, depolama sistemlerinin güvenliğini sağlama konusunda önemli katkılar sunabilir. Özellikle, nesne depolamanın sağladığı veri bütünlüğü ve erişim kontrol mekanizmaları, verilerin güvenliğini artırmakta ve veri kaybı ile yetkisiz erişim risklerini azaltmaktadır. Kurumlar, nesne depolamanın potansiyelinden yararlanarak veri güvenliğini güçlendirebilir ve siber saldırılara karşı daha dayanıklı bir yapı oluşturabilir.

## Radyo ve Televizyon Üst Kurulu Kapsamında Öneriler

Nesne depolama teknolojilerinin incelenmesi sonucunda, bu teknolojilerin düşük gecikme ve yüksek performans gerektiren kullanım durumları için uygun olmadığı, ancak ölçeklenebilirlik, esneklik ve yüksek veri yönetimi kapasitesi gerektiren kullanım senaryolarında etkili ve verimli olabildiği sonucuna varılmıştır. Bu bağlamda, nesne depolamanın Radyo ve Televizyon Üst Kurulu kapsamında, kurumsal verilerin yedeklenmesi ve Sayısal Kayıt, Arşiv ve Analiz Sistemi (SKAAS) üzerindeki uygulanabilirliği ele alınmıştır.

Kurumsal verilerin yedeklenmesi, iş sürekliliğinin sağlanması, veri kaybının önlenmesi ve düzenleyici gereksinimlere uyum açısından kritik bir öneme sahiptir. Günümüzde, işletmelerin veri yedekleme stratejileri, veri güvenliğini sağlamak, operasyonel kesintileri minimize etmek ve hızlı veri kurtarma kapasitelerini artırmak amacıyla kapsamlı sistemler içermektedir. Kurumsal veri yedekliliğini sağlamak için kullanılan mevcut sistemler, genellikle veri kaybı ve erişim performansı ile ilgili çeşitli zorluklarla karşılaşabilmektedir. Geleneksel yedekleme yöntemleri, büyük veri

kümeleriyle başa çıkmada, esneklik ve ölçeklenebilirlik açısından sınırlamalar sergileyebilmektedir. Ayrıca, veri erişim hızları, maliyetler ve veri güvenliği konularında iyileştirme gereksinimleri söz konusu olabilmektedir. Bu bağlamda, nesne depolama teknolojilerinin, mevcut kurumsal veri yedekleme süreçlerinin daha etkili ve verimli bir şekilde yönetilmesine olanak tanıyabileceği değerlendirilmektedir. Nesne depolamanın mevcut veri yedekleme sistemlerine entegrasyonu, yedekleme süreçlerinde önemli iyileştirmeler sağlayabilir. Ancak, bu entegrasyonun başarılı olabilmesi için mevcut sistemlerin ve veri yönetim politikalarının gözden geçirilmesi gerekmektedir. Nesne depolamanın maliyet etkileri, yönetim zorlukları ve teknik gereksinimlerin değerlendirilmesi, etkili bir entegrasyon stratejisinin geliştirilmesinde kritik rol oynamaktadır. Bu teknolojilerin entegrasyonunun, veri yedekleme stratejilerinin iyileştirilmesine ve kurumsal veri yönetiminin güçlendirilmesine katkıda bulunabileceği değerlendirilmektedir.

Ülkemizde radyo ve televizyon yayınlarının izlenmesi ve denetlenmesi sorumluluğu, 6112 Sayılı Kanun kapsamında Radyo ve Televizyon Üst Kurulu'na verilmiştir. RTÜK, ilgili yasanın kendisine tanıdığı yetki ve sorumluluk çerçevesinde, radyo ve televizyon kuruluşlarının yayınlarını izleyerek, bu yayınların yasa ve yönetmeliklerle belirlenen standartlara uygunluğunu Sayısal Kayıt, Arşiv ve Analiz Sistemi ile denetlemektedir. SKAAS, 2006-2008 yılları arasında Türkiye Bilimsel ve Teknik Araştırma Kurumu (TÜBİTAK) tarafından gerçekleştirilen bir araştırma ve geliştirme projesi olup, ülke genelindeki uydu, kablo ve karasal ortamda yapılan ulusal, bölgesel ve yerel televizyon yayınları ile ulusal radyo yayınlarının kaydedilmesi, arşivlenmesi ve analiz edilmesini sağlamaktadır. Sistem, yıl boyunca her gün 24 saat kesintisiz olarak faaliyet göstermektedir. Günümüz yayıncılık ve bilişim teknolojilerindeki ilerlemeler, RTÜK'ten yayın izni alan televizyon yayıncı kuruluşlarının sayısının artması ve 27.03.2018 tarihinde Resmî Gazete'de yayımlanan 7103 sayılı Kanun ile düzenlenen internet ortamında yapılan yayınların RTÜK tarafından izlenebilmesi ve denetlenebilmesi gerekliliği, SKAAS'ın veri depolama kapasitesinde artışa yol açmıştır. Bu bağlamda, özellikle medya ve içerik yönetimi alanında önemli avantajlar sağlayan nesne depolama teknolojilerinin SKAAS'ın artan veri depolama ihtiyacını karşılamada ve sistemin çeşitli yönlerini iyileştirmede faydalı olabileceği değerlendirilmektedir. Nesne depolama teknolojilerinin, SKAAS'ta kullanılabilirliğine ilişkin değerlendirmeler ele alınmıştır:

- SKAAS sistemi, medya içeriklerinin hızla büyüyen hacmiyle başa çıkmak durumundadır. Nesne depolama, yüksek ölçeklenebilirlik sunan bir teknoloji

olduğundan, büyük veri kümesine sahip arşivlerin yönetimini kolaylaştırabilir. Nesne depolamanın, SKAAS'ın sürekli artan veri ihtiyaçlarını karşılamada ve gelecekteki veri büyümesini verimli bir şekilde yönetmede fayda sağlayabilir.

- Nesne depolama, verilerin indekslenmesini ve meta verilerle ilişkilendirilmesini destekleyerek, arama ve erişim süreçlerini hızlandırır. Bu özelliği, SKAAS sisteminde depolanan içeriklere daha hızlı ve etkili bir şekilde ulaşılmasında etkili olabilir.
- Nesne depolamada meta veriler de asıl veriler ile birlikte saklandıklarından video verileri için ayrıca üretilecek indeksler ve üst veriler ile anlamsal arama (semantic search), video sahne anlamlandırma, otomatik ihlâl tespiti, veri madenciliği gibi uygulamalar geliştirilebilir.
- Nesne depolama sistemleri, veri güvenliği ve bütünlüğü açısından çeşitli özellikler sunar. Verilerin şifrelenmesi, yedeklenmesi ve erişim kontrolü gibi güvenlik önlemleri, SKAAS sisteminin veri güvenliğini artırabilir.
- Nesne depolama, uzun vadeli veri saklama için maliyet etkin bir çözüm sunar. Depolama alanı ihtiyacı arttıkça, geleneksel depolama çözümleri yüksek maliyetler getirebilir. Nesne depolama, daha düşük maliyetlerle büyük veri hacimlerini yönetme imkânı sunduğundan SKAAS sisteminin işletme maliyetlerini azaltabilir ve bütçe optimizasyonuna fayda sağlayabilir.
- Nesne depolama teknolojisi, veri arşivleme süreçlerini optimize eder**.** SKAAS sistemi, medya içeriklerinin uzun vadeli saklanmasını ve arşivlenmesini gerektirir. Nesne depolama, bu süreçleri destekleyerek içeriklerin uzun süreli saklanmasını ve yönetimini kolaylaştırabilir. Ayrıca, veri sınıflandırması ve kategorize edilmesi süreçlerini geliştiren nesne depolama, arşivleme verimliliğini artırabilir.

# KAYNAKLAR


Albayrak, A. S. ve Koltan Yılmaz, Ş. (2009). Veri madenciliği: Karar ağacı algoritmaları ve İMKB verileri üzerine bir uygulama. Süleyman Demirel Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi, 14(1), 31-52.

Amvrosiadis, G., Butt, A. R., Tarasov, V., Zadok, E., Zhao, M., Ahmad, İ. vd. (2019). Data storage research vision 2025 report. https://www.ece.iastate.edu/~mai/docs/papers/2018_DataStorage2025.pdf. Erişim tarihi: 12.07.2024.

AWS (2023). Container'lar ile sanal makineler arasındaki fark nedir? https://aws.amazon.com/tr/compare/the-difference-between-containers-and-virtual-machines/. Erişim tarihi: 12.07.2024.

AWS (t.y.a). Amazon S3 depolama sınıfları. https://aws.amazon.com/tr/s3/storage-classes/?nc=sn&loc=3. Erişim tarihi: 14.07.2024.

AWS (t.y.b). Amazon S3 nasıl çalışır? https://aws.amazon.com/tr/s3/?nc=sn&loc=1 Erişim tarihi: 14.07.2024.

Azeem, S. A. ve Sharma, S. K. (2017). Study of converged infrastructure & hyper converge infrastructre as future of data centre. International Journal of Advanced Research in Computer Science, 8(5), 900-903.

Blumzon, C. F. I. ve Pănescu, A. T. (2019). Data storage. Bespalov, A., Michel, M. ve Steckler, T. (Eds.). Good research practice in non-clinical pharmacology and biomedicine. Handbook of experimental pharmacology 257 içinde (277-299). Springer. DOI: 10.1007/978-3-030-33656-1

Boz Eravcı, D. (2020). Kurumların dijital dönüşümü: Büyük veri. Çalışma İlişkileri Dergisi, 11(1), 90-112.

Brook, C. (2024). What is data sanitization? How it protects your organization. https://www.digitalguardian.com/blog/what-data-sanitization-how-it-protects-your-organization. Erişim tarihi: 14.07.2024.

Bunn, F., Simpson, N., Peglar, R. ve Nagle, G. (2004). Storage virtualization. https://www.snia.org/sites/default/files/sniavirt.pdf. Erişim tarihi: 12.07.2024.

Caceres Gutierrez, F. J. (2020). Towards an S3-based, DataNode-less implementation of HDFS. KTH Royal Institute of Technology School of Electrical Engineering and Computer Science.

Carlson, M., Yoder, A., Schoeb, L., Deel, D., Pratt, C., Lionetti, C. ve Voigt, D. (2015). Software defined storage. Storage Networking Industry Association.

Ceballos, J. (2016). A security strategy against steal-and-pass credential attacks. International Journal of Network Security & Its Applications (IJNSA), 8(1), 45-57.

Chandramouli, R. ve Pinhas, D. (2020). Security guidelines for storage infrastructure. NIST Special Publication 800-209. DOI: 10.6028/NIST.SP.800-209

Computer History Museum (t.y.). Timeline of computer history. https://www.computerhistory.org/timeline/memory-storage/. Erişim tarihi: 12.07.2024.

Das, T.K. ve Mohan Kumar, P. (2013). Big data analytics: A framework for unstructured data analysis. International Journal of Engineering and Technology (IJET), 5(1), 155-158.

Datacore (2024). Storage virtualization. https://www.datacore.com/storage-virtualization/. Erişim tarihi: 12.07.2024.

Dell Technologies (2024). ECS overview and architecture. https://infohub.delltechnologies.com/en-us/t/ecs-overview-and-architecture/. Erişim tarihi: 01.07.2024.

Dialini, P. (2020). The future of data revoloution will be unstructured data. https://www.analyticsinsight.net/insights/the-future-of-data-revolution-will-be-unstructured-data. Erişim tarihi: 12.07.2024

Doan, A., Naughton, J.F., Baid, A., Chai, X., Chen, F., Chen, T., Chu, E., DeRose, P., Gao, B., Gokhale, C., Huang, J., Shen, W. ve Vuong, B. (2009). The case for a structured approach to managing unstructured data. 4th Biennial Conference on Innovative Data Systems Research (CIDR 2009), Asimolar, CA, USA.

Doğan, K. ve Arslantekin, S. (2016). Büyük veri: Önemi, yapısı ve günümüzdeki durum. Ankara Üniversitesi Dil ve Tarih-Coğrafya Fakültesi Dergisi, 56(1), 15-36.

Dokuz, A. Ş. ve Çelik, M. (2017). Bulut bilişim sistemlerinde verinin farklı boyutları üzerine derleme. Niğde Ömer Halisdemir Üniversitesi Mühendislik Bilimleri Dergisi, 6(2), 316-338.

Domenech, J. S. (2018). Enhancing the programmability of cloud object storage [Yayımlanmamış doktora tezi, Universitat Rovira Virgili].

Eberendu, A. C. (2016). Unstructured data: An overview of the data of big data. International Journal of Computer Trends and Technology (IJCTT), 38(1), 46-50.

EMC (2003). EMC Centera Content Addressed Storage. https://course.ccs.neu.edu/csg389/2005/readings/centera_guide.pdf. Erişim tarihi: 12.07.2024

Factor, M., Nagle, D., Naor, D., Riedel, E. ve Satran, J. (2005). The OSD security protocol; Third IEEE International Security in Storage Workshop (SISW'05), San Francisco, CA, USA.

Factor, M., Meth, K., Naor, D., Rodeh, O. Ve Satran, J. (2005). Object storage: The future building block for storage systems. 2005 IEEE International Symposium on Mass Storage Systems and Technology, Sardinia, Italy.

Gibson, G. A., Nagle, D. F., Amiri, K., Chang, F. W., Feinberg, E. M., Gobioff, H., Lee, C., Ozceri, B., Riedel, E., Rochberg, D. ve Zelenka, J. (1997). File server scaling with network-attached secure disks. ACM International Conference on

Measurement and Modeling of Computer Systems (Sigmetrics ’97), Seattle, Washington, USA.

Google Cloud (t.y.). Containers vs VMs (virtual machines): What are the differences? https://cloud.google.com/discover/containers-vs-vms. Erişim tarihi: 12.07.2024.

Hanig, C., Schierle, M. Ve Trabold, D. (2010). Comparison of structured vs. unstructured data for industrial quality analysis. Proceeding of the World Congress on Engineering and Computer Science (WCECS 2010), San Francisco, USA.

Huawei Technologies Co. (2023). Cloud computing technology. Posts & Telecom Press.

Hurwitz, J., Nugent, A., Halper, F. ve Kaufman, M. (2013). Big data for dummies. John Wiley & Sons, Inc.

IBM Cloud Education, (2021). Structured vs unstructured data. https://www.ibm.com/think/topics/structured-vs-unstructured-data. Erişim tarihi: 14.07.2024.

Jain, K. (2007). Object-based storage. CSE 598D-Storage Systems Survey. https://www.cse.psu.edu/~buu1/teaching/spring07/598d/_assoc/0F9F9F1F33E2420396C5B60D90F5600B/kanishk.pdf. Erişim tarihi: 14.07.2024.

Jyoti, R., Kulkarni, S. ve Wanjale, K. (2021). A survey on different storage architectures. Journal of University of Shanghai for Science and Technology, 23(5), 791-796.

Kalseth, K. ve Cummings, S. (2001). Knowledge management: Development strategy or business strategy? Information Development, 17(3), 163-172.

Karabay, B. ve Ulaş, M. (2017). Büyük veri işlemede yaygın kullanılan araçların karşılaştırılması. 8th International Advanced Technologies Symposium (IATS’17), Elazığ, Türkiye.

Kingsley, M. S. (2023). Cloud technologies and services. Springer International Publishing. DOI: 10.1007/978-3-031-33669-0.

Law Insider (t.y.). Storage media definition. https://www.lawinsider.com/dictionary/storage-media. Erişim tarihi: 14.07.2024.

Lemmou, Y., Lanet, J. L. ve Souidi, E. M. (2020). A behavioural in-depth analaysis of ransomware infection. IET Information Security, 15(1), 38-58. John Wiley & Sons Ltd.

Li, Z. ve Yu, Z. (2006). A credential-based security mechanism for object-based storage. 2006 International Conference on Communications, Circuits and Systems, 1610-1614. Guilin, China. DOI: 10.1109/ICCCAS.2006.284981

Liu, Q., Feng, D., Ling-jun, Q. ve Ling-Fang, Z. (2006). A framework for accessing general object storage. 2006 International Workshop on Networking, Architecture and Storages (IWNAS 2006), Shenyang, China.

Lu, Y. F., Kuo, C. F., Chou, S. C., Li, J. S. ve Lai, Y. W. (2017). Cost-aware software-defined hybrid object-based storage system. 18th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT), 477-482.

Maddi, A. (2009). Performance analysis and implementation of object based storage. [Yayımlanmamış yüksek lisans tezi, Wichita State University].

Maluf, D.A. ve Tran, P. B. (2008). Managing unstructured data with structured legacy systems. 2008 IEEE Aerospace Conference, Big Sky, MT, USA. DOI: 10.1109/AERO.2008.4526666

Mansouri, Y., Toosi, A. N. ve Buyya, R. (2017). Data storage management in cloud environments: Taxonomy, survey, and future directions. ACM Computing Surveys, 50(6), Article No: 91, 1-51. DOI: 10.1145/3136623

Mesnier, M., Ganger, G. R. ve Riedel, E. (2003). Object-based storage. IEEE Communications Magazine, 41(8), 84-90.

Milligan, G. E. (1999). Object based storage devices. Command Set Proposal; 10/26/1999; T10/ 99-315R0.

Mondal, A. S., Sanyal, M., Barua, H. B., Chattopadhyay, S. ve Mondal, K. C. (2024). Comparative analysis of object-based big data storage systems on architectures and services: A recent survey. Journal of The Institution of Engineers (India) Series B, 105, 685-700.

Nami, B. Ve Sakar, G. (t.y.). Nesne tabanlı depolama ve yazılım tabanlı veri merkezleri. https://ab.org.tr/ab15/kitap/yyy/243.pdf. Erişim tarihi: 12.07.2024

Nelson, S. (2011). Pro data backup and recovery. Apress.

NetApp (2024). What is software-defined storage? https://www.netapp.com/data-management/what-is-software-defined-storage/. Erişim tarihi: 14.07.2024.

NetApp (2016). What grid nodes and services are. https://library.netapp.com/ecmdocs/ECMLP2412002/html/GUID-559D8264-3034-48C7-8E4C-92147C348B1C.html. Erişim tarihi: 12.07.2024.

Openstack (2024). Documentation for 2024.2. https://docs.openstack.org/. Erişim tarihi: 12.07.2024.

Paloalto (2024). What is data classification? https://www.paloaltonetworks.com /cyberpedia/data-classification. Erişim tarihi: 14.07.2024.

Pandey, H. (2023). Data replication in dbms. https://www.geeksforgeeks.org/data-replication-in-dbms/. Erişim tarihi: 12.07.2024.

Patil, A., Rangarao, D., Seipp, H., Lasota, M., dos Santos, R. M., Markovic, R. vd. (2020). Cloud object storage as a service: IBM cloud object storage from theory to practice-for developers, IT architects and IT specialists. IBM Redbooks.

Paul, S. (2021). What is object storage? It's crucial for managing cloud data. https://www.g2.com/articles/object-storage. Erişim tarihi: 14.07.2024.

Perry, Y. (2020). What is block storage? https://bluexp.netapp.com/blog/cvo-blg-what-is-block-storage-pros-cons-and-comparisons. Erişim tarihi: 12.07.2024.

Perry, Y. (2021). S3 depolama: Eksiksiz kılavuz. https://bluexp.netapp.com/blog/aws-cvo-blg-s3-storage-the-complete-guide. Erişim tarihi: 14.07.2024.

Potnis, A. (2019). IDC marketspace: Worldwide object-based storage 2019 vendor assessment. https://www.netapp.com/media/7614-idc-marketscape-worldwide-object-based-storage-2019-vendor-assessment.pdf. Erişim tarihi: 10.07.2024.

Red Hat (2018). File storage, block storage, or object storage. https://www.redhat.com/en/topics/data-storage/file-block-object-storage. Erişim tarihi: 18.07.2024.

Red Hat (2024). Red hat ceph storage 6 architecture guide. https://docs.redhat.com/en-us/documentation/red_hat_ceph_storage/6/pdf/architecture_guide/Red_Hat_Ceph_Storage-6-Architecture_Guide-en-US.pdf. Erişim tarihi: 02.09.2024

Rotella, P. (2012). Is data the new oil?. https://www.forbes.com/sites/perryrotella/2012/04/02/is-data-the-new-oil/. Erişim tarihi: 12.07.2024.

Qader, W. A., Ameen, M.M. ve Ahmed, B.M. (2020). Big data characteristics, architectures, technologies and applications. Journal of Computer Science, 16(6), 817-824.

Samundiswary, S. ve Dongre, N. M. (2017). Object storage architecture in cloud for unstructured data. International Conference on Inventive Systems and Control (ICISC-2017). DOI: 10.1109/ICISC.2017.8068716

Shankar, N. ve Yadav, K. (2022). Objectscale: Next evolution in object storage. Dell Inc. https://learning.dell.com/content/dam/dell-emc/documents/en-us/2022KS_Shankar-ObjectScale_Next_evolution_in_object_storage.pdf. Erişim tarihi: 14.07.2024.

Singh, A. (2024). Storage area network advantages and disadvantages. https://storagearea.network/storage-area-network-advantages-and-disadvantages/. Erişim tarihi: 12.07.2024.

Sivakumar, R., Pai, S. S. ve Rajeev, A. (2023). DNA: Future of data storage. Dell Inc. https://learning.dell.com/content/dam/dell-emc/documents/en-us/2023KS_Sivakumar-DNA_Future_of_Data_Storage.pdf. Erişim tarihi: 14.07.2024.

SNIA (2018a). File vs block vs object storage. https://www.snia.org/educational-library/file-vs-block-vs-object-storage-2018. Erişim tarihi: 16.07.2024.

SNIA (2018b). Storage security: Data protection. Storage Networking Industry Association. https://www.snia.org/sites/default/files/security/SNIA-Data-Protection-TechWhitepaper.pdf. Erişim tarihi: 14.07.2024.

SNIA (2024a). What is nas. https://www.snia.org/education/what-is-nas. Erişim tarihi: 12.07.2024.

SNIA (2024b). What is a storage area network (SAN)? https://www.snia.org/education/storage_networking_primer/san/what_san. Erişim tarihi: 14.07.2024.

SNIA (2024c). What is object storage? https://www.snia.org/education/what-is-object-storage. Erişim tarihi:10.07.2024.

SNIA (2024d). What is cloud storage? https://www.snia.org/education/what-is-cloud-storage. Erişim tarihi: 12.07.2024.

SNIA (t.y.). What is data protection? https://www.snia.org/education/what-is-data-protection. Erişim tarihi: 12.07.2024.

Somasundaram, G., Shrivastava, A. ve EMC education services. (2009). Information storage and management: Storing, managing, and protecting digital information. Wiley Publishing.

Stanek, J. (2018). Secure data deduplication in cloud storage services. [Yayımlanmamış doktora tezi, Czech Technical University in Prague].

Steinruecken, C. (2014). Lossless data compression. [Yayımlanmamış doktora tezi, University of Cambridge].

Schlosser, S. W. ve Iren, S. (2005). Database storage management with object-based storage devices. Workshop on Data Management on New Hardware (DaMoN 2005), Baltimore, MD, USA.

Surianarayanan, C. ve Chelliah, P. R. (2023). Essentials of cloud computing: A holistic, cloud-native perspective (2. Baskı). Springer.

Süβ, R. ve Süβ, Y. (2024). IT infrastructure: Security and resilience solutions. Apress.

Şeker, S. E. (2013). İş zekâsı ve veri madenciliği. Cinius yayınları.

Tauro, C. J., Ganesan, N. ve RS, A. K. (2012). A study of benefits in object based storage systems. International Journal of Computer Applications, 42(8), 28-30.

Tso, M. (2017). Why object storage is a blockbuster hit for media and entertainment. https://www.forbes.com/sites/forbestechcouncil/2017/11/07/why-object-storage-is-a-blockbuster-hit-for-media-and-entertainment/. Erişim tarihi: 14.07.2024.

Utmel (2021). The evolution history of storage devices. https://www.utmel.com/blog/categories/memory%20chip/the-evolution-history-of-storage-devices. Erişim tarihi: 14.07.2024.

Vantara, H. (2020). Hitachi content platform architecture Fundamentals. https://cms.neva.gov.in/FileStructure_LS/AssemblyFiles/17/5/1.pdf. Erişim tarihi: 12.07.2024.

Vmware (t.y.). What is software-defined storage (SDS)? https://www.vmware.com /topics/software-defined-storage. Erişim tarihi: 14.07.2024.

Weil, S. A. (2007). Ceph: Reliable, scalable, and high-performance distributed storage. [Yayımlanmamış doktora tezi, University of California].

Welch, B. (2013). Object storage technology. https://www.snia.org/sites/ default/education/tutorials/2013/spring/file/BrentWelch_Object_Storage_Technolo gy.pdf. Erişim tarihi: 01.07.2024.

Yılmaz, M. (2009). Enformasyon ve bilgi kavramları bağlamında enformasyon yönetimi ve bilgi yönetimi; Ankara Üniversitesi Dil ve Tarih-Coğrafya Fakültesi Dergisi, 49(1), 95-118.

Yiğit, S. A. (2016). Nesne-tabanlı depolama birimleri. https://tr.linkedin.com/pulse/nesne-tabanl%C4%B1-depolama-birimleri-object-based-storage-asiye-yigit. Erişim tarihi: 12.07.2024.

Yue, Y., Wang, F. ve Feng, D. (2007). iRBO: Intelligent role-based object for object-based storage device. 2007 International Conference on Computational Intelligence and Security (CIS 2007), 257-261. IEEE.

Zeng, L. F., Feng, D. ve Qin, L. J. (2004). SOSS: Smart object-based storage system. Proceedings of 2004 International Conference on Machine Learning and Cybernetics (IEEE Cat. No. 04EX826), 5. Cilt, 3263-3266. IEEE.

# ÖZGEÇMİŞ

**Kişisel Bilgiler**

**Adı ve Soyadı** **:** Hazal KILIÇ

**E-Posta Adresi:** **:** hazalkilic@rtuk.gov.tr

**İş Deneyimi** **:**

| **Unvan** | **Görev Yeri** | **Yıl** |
|---|---|---|
| Üst Kurul Uzman Yardımcısı | Radyo ve Televizyon Üst Kurulu | 2021-… |

**Akademik Bilgiler**

**Öğrenim Durumu:**

| **Derece** | **Böliim/Program** | **Üniversite** | **Yıl** |
|---|---|---|---|
| Lisans | Tarih | İstanbul Üniversitesi | 2021-… |
| Lisans | Bilgisayar Mühendisliği | Ankara Üniversitesi | 2013-2018 |