



Red Hat Jboss Data Grid: Uygulamalarınıza Süper Güç Katın!

M. Aykut BULGU
Middleware Consultant - Red Hat
@systemcraftsman

Red Hat Forum İstanbul - 2018

Ben Kimim?



Evli ve bir çocuk babasıyım

Çocukluğum 80'lerin sonu 90'ların başında geçti

Teknoloji aşığı

Yaklaşık 12 yıldır teknoloji/yazılım sektöründe

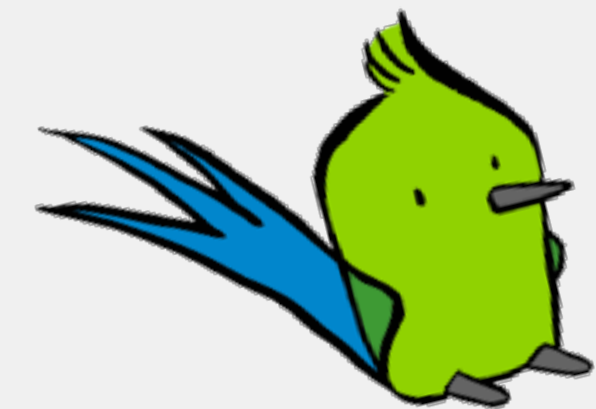
Yazılım Zanaatkarlığı olgusu sıkı destekçisi

Software Craftsmanship Turkey co-organizatör

Red Hat'te Middleware Consultant



RED HAT® JBOSS® MIDDLEWARE





RED HAT 3SCALE
API MANAGEMENT

RED HAT JBOSS
ENTERPRISE
APPLICATION PLATFORM

RED HAT
PROCESS AUTOMATION
MANAGER

RED HAT JBOSS
DATA GRID

RED HAT® JBOSS® MIDDLEWARE

RED HAT
DECISION
MANAGER

RED HAT
FUSE

RED HAT JBOSS
DATA VIRTUALIZATION

RED HAT JBOSS
WEB SERVER

RED HAT
AMQ

HIZLANDIR



RED HAT®
OPENSIFT
Application Runtimes

RED HAT® 3SCALE®
API MANAGEMENT

RED HAT® JBOSS®
ENTERPRISE
APPLICATION PLATFORM

RED HAT® JBOSS®
DATA GRID

RED HAT® JBOSS®
WEB SERVER

Uygulama geliştirme, ölçeklenebilirlik ve performans

RED HAT® JBOSS® MIDDLEWARE

OTOMATİZE ET

RED HAT®
PROCESS AUTOMATION
MANAGER

RED HAT®
DECISION
MANAGER

İş kuralları ve süreç yönetimi

ENTEĞRE ET

RED HAT®
FUSE

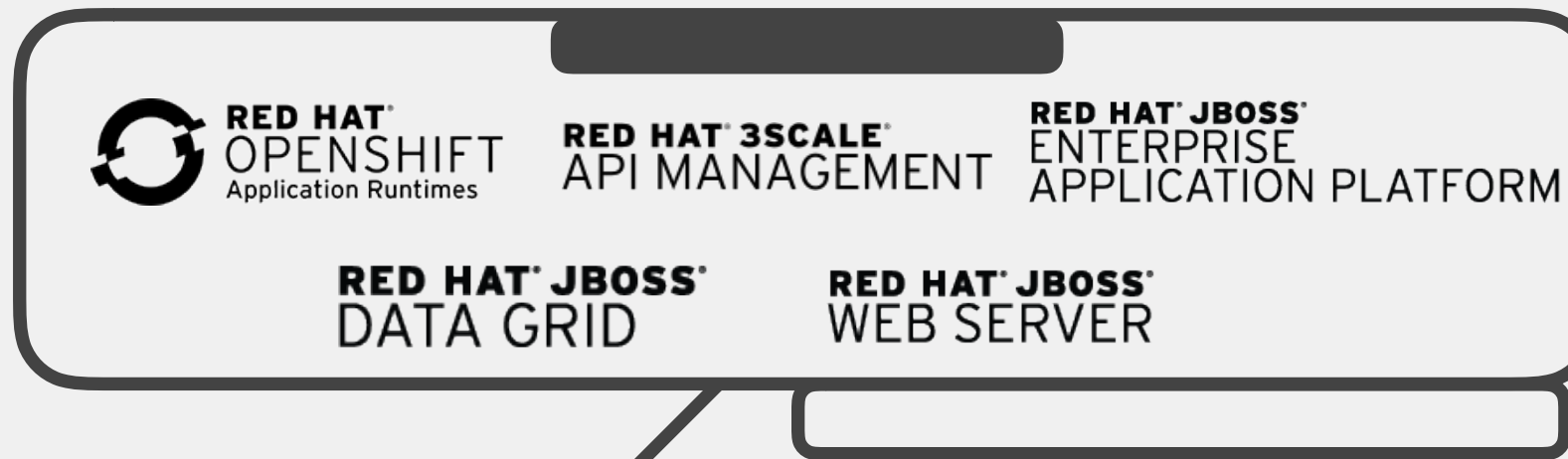
RED HAT®
AMQ

RED HAT® JBOSS®
DATA VIRTUALIZATION

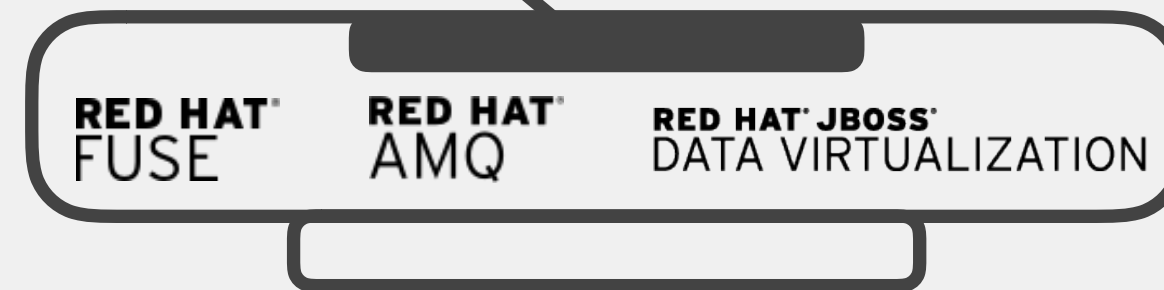
Uygulamalar, veri ve araçları

PUBLIC

PRIVATE



RED HAT JBOSS MIDDLEWARE



HYBRID

Enterprise Uygulamaların Karşılaştığı Genel Zorluklar



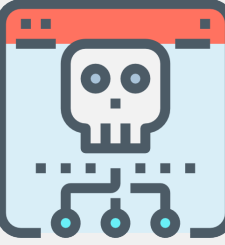
Uygulama çok yavaş

Zaman ve güven ve/veya iş kayıpları çünkü *uygulama çok yavaş*



Eşzamanlı istek limitleri

Belirli bir limit üzerindeki verilerin *eşzamanlı isteklerle* işlenememesi



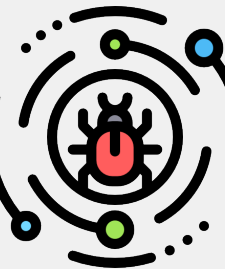
Giderek artan veri depolama ihtiyacı

Zaman geçtikçe uygulamaların genelde lineer olmayan, *giderek artan veri depolama ihtiyacı*



Latency problemleri

Uygulama ve back-end'e bağlı veri deposu uygulama ile birlikte sadece bir coğrafi bölgede bulunmakta ve farklı bölgeler ile *latency* problemleri yaşanmakta



Single Point of Failure

Uygulamanın veri deposuyla ilgili *single point of failure* riski



Veri toplama problemleri

Enterprise müşterinin aktivitelerine ait *analiz ve iç-görüleri* toplayamamak, bu verileri veri analizi, iş zekası, *büyük veri ve makina öğrenmesi* içeren işlerde kullanamamak



Eski Yöntem Çözümler

Çözüm	Dezavantajı
Performans iyileştirmesi için uygulama sunucusuna daha fazla hardware eklemek	Pahalı Yavaş veri akışına doğru adreslenmiş bir çözüm değil
Depolama kapasitesine arttırmak adına daha fazla disk eklemek	Pahalı Geçici çözüm Hala bir <i>Single Point of Failure</i> durumu söz konusu
“Data contention” problemini farklı zamanlardaki veri okuma/yazma zamanlamasını planlayarak çözmeye çalışmak	Geçici çözüm Arkaplandaki veri yükü probleminin altında yatan ana sebebi çözmekte
“High Availability”ye bir alternatif olarak veri tabanı çöktüğünde “disaster recovery” yedeklerine güvenmek	Pahalı donanım Bir zero-downtime çözümü değil İş süreçlerine olumsuz etki
Veri erişimini bölgeler arası artırmak için özelleştirilmiş veri deposu replikasyonlarını kullanmak	Karışık ve zaman alıcı Uyumsuz veri bütünlüğü ihtimali

Çözüm

Performans iyileştirmesi için uygulama sunucusuna daha fazla hardware eklemek

Depolama kapasitesine arttırmak adına daha fazla disk eklemek

“Data contention” problemini farklı zamanlardaki veri okuma/yazma zamanlamasını planlayarak çözmeye çalışmak

“High Availability”ye bir alternatif olarak veri tabanı çöktüğünde “disaster recovery” yedeklerine güvenmek

Veri erişimini bölgeler arası artırmak için özelleştirilmiş veri deposu replikasyonlarını kullanmak



Dezavantajı

Pahalı

Yavaş veri akışına doğru adreslenmiş bir çözüm değil

Pahalı

Geçici çözüm

Hala bir *Single Point of Failure* durumu söz konusu

Geçici çözüm

Arkaplandaki veri yükü probleminin altında yatan ana sebebi çözmekte

Pahalı donanım

Bir zero-downtime çözümü değil

İş süreçlerine olumsuz etki

Karışık ve zaman alıcı

Uyumsuz veri bütünlüğü ihtimali

Çözüm

Performans iyileştirmesi için uygulama sunucusuna daha fazla hardware eklemek

Depolama kapasitesine arttırmak adına daha fazla disk eklemek

“Data contention” problemini farklı zamanlardaki veri okuma/ yazma zamanlamasını planlayarak çözmeye çalışmak

“High Availability”ye bir alternatif olarak veri tabanı çöktüğünde “disaster recovery” yedeklerine güvenmek

Veri erişimini bölgeler arası artırmak için özelleştirilmiş veri deposu replikasyonlarını kullanmak



Dezavantajı

Pahalı

Yavaş veri akışına doğru adreslenmiş bir çözüm değil

Pahalı

Geçici çözüm

Hala bir *Single Point of Failure* durumu söz konusu

Geçici çözüm

Arkaplandaki veri yükü probleminin altında yatan ana sebebi çözmekte

Pahalı donanım

Bir zero-downtime çözümü değil

İş süreçlerine olumsuz etki

Karışık ve zaman alıcı

Uyumsuz veri bütünlüğü ihtimali

Çözüm

Performans iyileştirmesi için uygulama sunucusuna daha fazla hardware eklemek

Depolama kapasitesine arttırmak adına daha fazla disk eklemek

“Data contention” problemini farklı zamanlardaki veri okuma/ yazma zamanlamasını planlayarak çözmeye çalışmak

“High Availability”ye bir alternatif olarak veri tabanı çöktüğünde “disaster recovery” yedeklerine güvenmek

Veri erişimini bölgeler arası artırmak için özelleştirilmiş veri deposu replikasyonlarını kullanmak



Dezavantajı

Pahalı

Yavaş veri akışına doğru adreslenmiş bir çözüm değil

Pahalı

Geçici çözüm

Hala bir *Single Point of Failure* durumu söz konusu

Geçici çözüm

Arkaplandaki veri yükü probleminin altında yatan ana sebebi çözmekte

Pahalı donanım

Bir zero-downtime çözümü değil

İş süreçlerine olumsuz etki

Karışık ve zaman alıcı

Uyumsuz veri bütünlüğü ihtimali

Çözüm

Performans iyileştirmesi için uygulama sunucusuna daha fazla hardware eklemek

Depolama kapasitesine arttırmak adına daha fazla disk eklemek

“Data contention” problemini farklı zamanlardaki veri okuma/ yazma zamanlamasını planlayarak çözmeye çalışmak

“High Availability”ye bir alternatif olarak veri tabanı çöktüğünde “disaster recovery” yedeklerine güvenmek

Veri erişimini bölgeler arası artırmak için özelleştirilmiş veri deposu replikasyonlarını kullanmak



Dezavantajı

Pahalı

Yavaş veri akışına doğru adreslenmiş bir çözüm değil

Pahalı

Geçici çözüm

Hala bir *Single Point of Failure* durumu söz konusu

Geçici çözüm

Arkaplandaki veri yükü probleminin altında yatan ana sebebi çözmekte

Pahalı donanım

Bir zero-downtime çözümü değil

İş süreçlerine olumsuz etki

Karışık ve zaman alıcı

Uyumsuz veri bütünlüğü ihtimali



Çözüm

Performans iyileştirmesi için uygulama sunucusuna daha fazla hardware eklemek

Depolama kapasitesine arttırmak adına daha fazla disk eklemek

“Data contention” problemini farklı zamanlardaki veri okuma/ yazma zamanlamasını planlayarak çözmeye çalışmak

“High Availability”ye bir alternatif olarak veri tabanı çöktüğünde “disaster recovery” yedeklerine güvenmek

Veri erişimini bölgeler arası artırmak için özelleştirilmiş veri deposu replikasyonlarını kullanmak

Dezavantajı

Pahalı

Yavaş veri akışına doğru adreslenmiş bir çözüm değil

Pahalı

Geçici çözüm

Hala bir *Single Point of Failure* durumu söz konusu

Geçici çözüm

Arkaplandaki veri yükü probleminin altında yatan ana sebebi çözmekte

Pahalı donanım

Bir zero-downtime çözümü değil

İş süreçlerine olumsuz etki

Karışık ve zaman alıcı

Uyumsuz veri bütünlüğü ihtimali



Çözüm

Performans iyileştirmesi için uygulama sunucusuna daha fazla hardware eklemek

Depolama kapasitesine arttırmak adına daha fazla disk eklemek

“Data contention” problemini farklı zamanlardaki veri okuma/ yazma zamanlamasını planlayarak çözmeye çalışmak

“High Availability”ye bir alternatif olarak veri tabanı çöktüğünde “disaster recovery” yedeklerine güvenmek

Veri erişimini bölgeler arası artırmak için özelleştirilmiş veri deposu replikasyonlarını kullanmak

Dezavantajı

Pahalı

Yavaş veri akışına doğru adreslenmiş bir çözüm değil

Pahalı

Geçici çözüm

Hala bir *Single Point of Failure* durumu söz konusu

Geçici çözüm

Arkaplandaki veri yükü probleminin altında yatan ana sebebi çözmekte

Pahalı donanım

Bir zero-downtime çözümü değil

İş süreçlerine olumsuz etki

Karışık ve zaman alıcı

Uyumsuz veri bütünlüğü ihtimali

Çözüm

Performans iyileştirmesi için uygulama sunucusuna daha fazla hardware eklemek

Depolama kapasite arttırmak için daha fazla disk eklemek

“Data contention” problemlerini zamanlardaki veri okuma/ yazma zamanlarını azaltarak çözmeye çalışmak

“High Availability” için alternatif veri tabanı çöktüğünde “disaster recovery” yeteneklerine güvenmek

Veri erişimini bölgeler arası artırmak için özelleştirilmiş veri deposu replikasyonlarını kullanmak

Dezavantajı

Pahalı

Yavaş veri akışına doğru adreslenmiş bir çözüm değil

Pahalı

Geçici çözüm

Hala bir *Single Point of Failure* durumu söz konusu

Geçici çözüm

Arkaplandaki veri yükü probleminin altında yatan ana sebebi çözmekte

Pahalı donanım

Bir zero-downtime çözümü değil

İş süreçlerine olumsuz etki

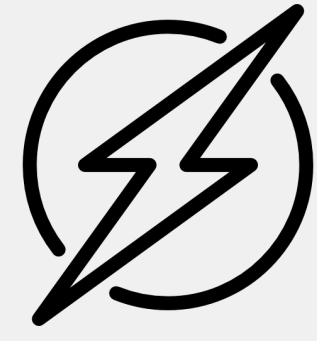
Karışık ve zaman alıcı

Uyumsuz veri bütünlüğü ihtimali



Gerçek ihtiyalar?





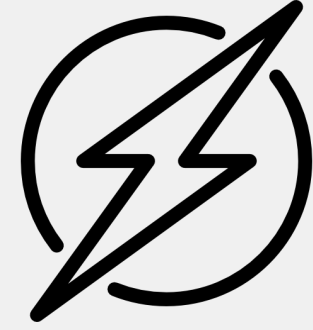
Her bir salisenin bile önemli olduđu bir ortamda hız mı?



Her bir salisenin bile önemli olduđu bir ortamda hız mı?



Mevcut donanımınız üzerinde çalışan uygulamalarınızda ölçeklenebilirlik mi?



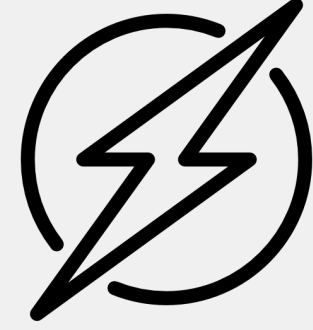
Her bir salisenin bile önemli olduđu bir ortamda hız mı?



Mevcut donanımınız üzerinde çalışan uygulamalarınızda ölçeklenebilirlik mi?



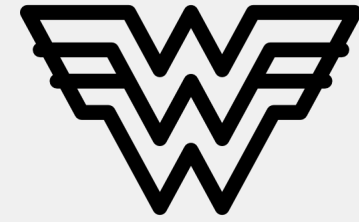
Özel bir donanım ihtiyacı olmadan hataya tolere bir yapı mı?



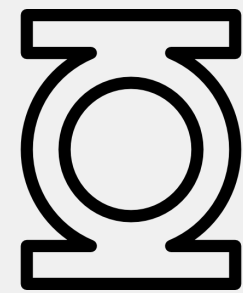
Her bir salisenin bile önemli olduđu bir ortamda hız mı?



Mevcut donanımınız üzerinde çalışan uygulamalarınızda ölçeklenebilirlik mi?



Özel bir donanım ihtiyacı olmadan hataya tolere bir yapı mı?



Veri deęişimleri uyarılarını gösteren bir sistem mi?



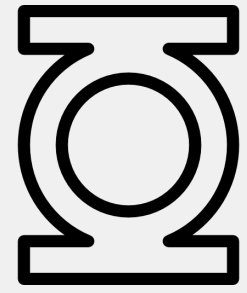
Her bir salisenin bile önemli olduđu bir ortamda hız mı?



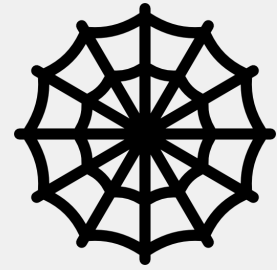
Mevcut donanımınız üzerinde çalışan uygulamalarınızda ölçeklenebilirlik mi?



Özel bir donanım ihtiyacı olmadan hataya tolere bir yapı mı?



Veri deđişimleri uyarılarını gösteren bir sistem mi?



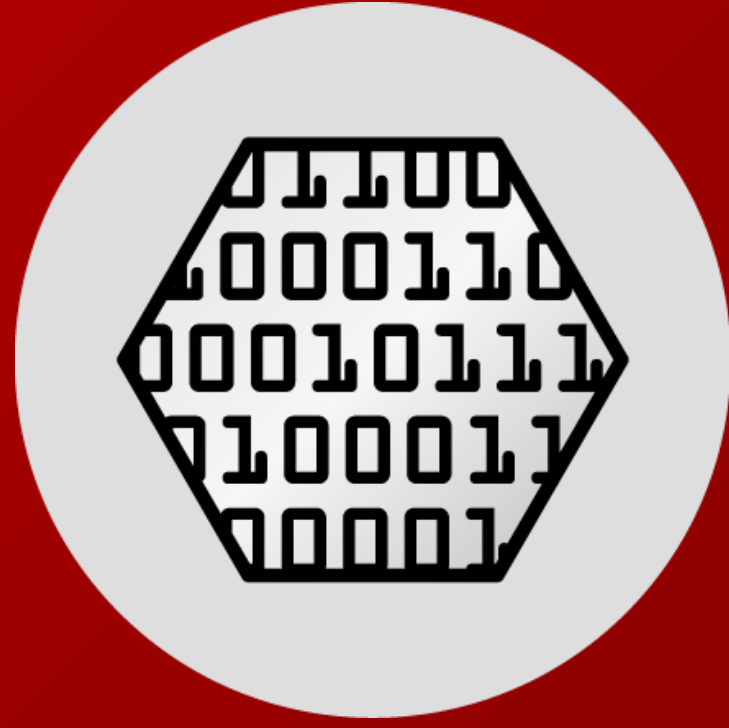
Mevcut analiz sisteminizle veri deđiştikçe tekrar tekrar aynı analizleri yapabilmek mi?

Çözüm:

Red Hat Jboss Data Grid



Red Hat Jboss Data Grid



Veri üzerinde daha hızlı aksiyon

alma

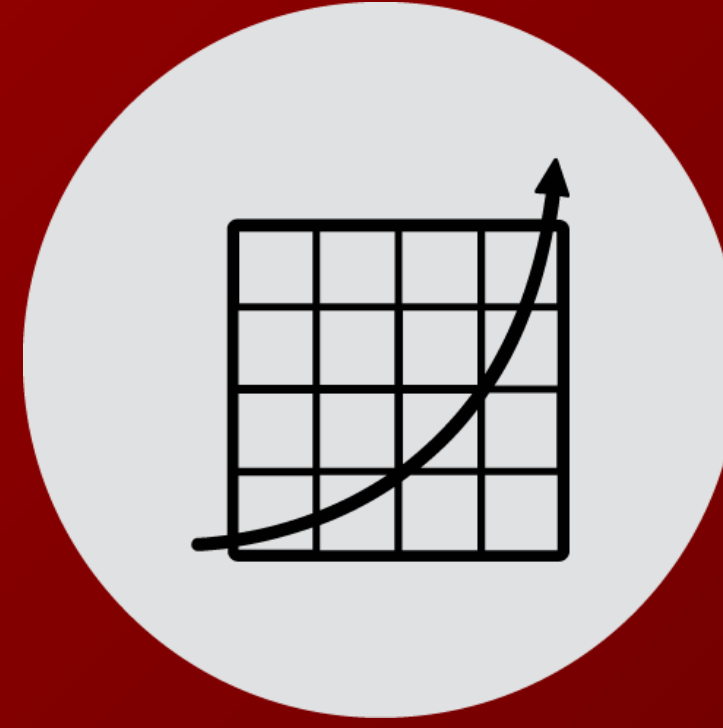
Veri güvenliği



Red Hat Jboss Data Grid



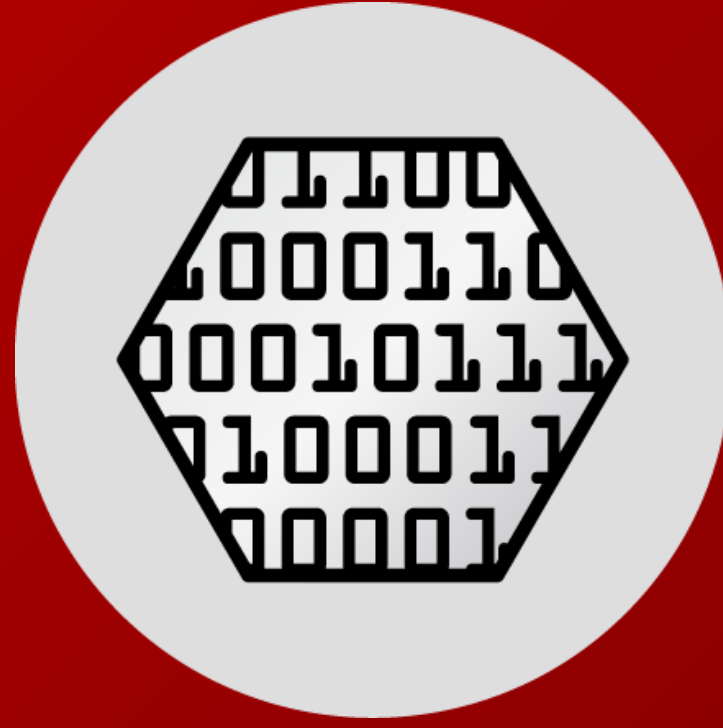
Veri üzerinde daha hızlı aksiyon
alma
Veri güvenliği



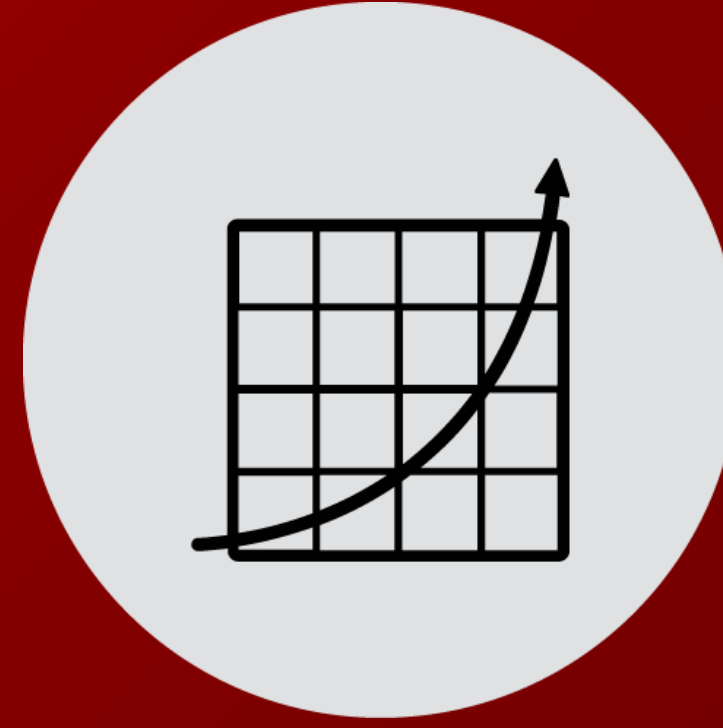
Veri ölçekleme
Verimlilik artışı



Red Hat Jboss Data Grid



Veri üzerinde daha hızlı aksiyon
alma
Veri güvenliği



Veri ölçekleme
Verimlilik artışı



Her zaman mevcut veri (Highly
Available)
Hata toleranslı veri





4 Ana Özellik

Dağıtık Önbellek

En sık erişilen dataları tutabilmek için sağlanan in-memory veri deposu

Geçişken, kısa-zamanlı veri depolama imkanı



4 Ana Özellik

Dağıtık Önbellek

En sık erişilen data ları tutabilmek için sağlanan in-memory veri deposu

Geçişken, kısa-zamanlı veri depolama imkanı

NoSQL Veritabanı

NoSQL Key-Value veri deposu

Konfigüre edilebilir ACID transaction desteği



4 Ana Özellik

Dağıtık Önbellek

En sık erişilen data ları tutabilmek için sağlanan in-memory veri deposu

Geçişken, kısa-zamanlı veri depolama imkanı

NoSQL Veritabanı

NoSQL Key-Value veri deposu

Konfigüre edilebilir ACID transaction desteği

Olay Dinleyiciler

Veri üzerindeki CRUD işlemlerini dinleme ve tepki verme yeteneği

Süregelen sorgular ile her zaman aynı sonuç kümesi dönmesi sağlanır



4 Ana Özellik

Dağıtık Önbellek

En sık erişilen dataları tutabilmek için sağlanan in-memory veri deposu

Geçişken, kısa-zamanlı veri depolama imkanı

NoSQL Veritabanı

NoSQL Key-Value veri deposu

Konfigüre edilebilir ACID transaction desteği

Olay Dinleyiciler

Veri üzerindeki CRUD işlemlerini dinleme ve tepki verme yeteneği

Süregelen sorgular ile her zaman aynı sonuc kümesi dönmesi sağlanır

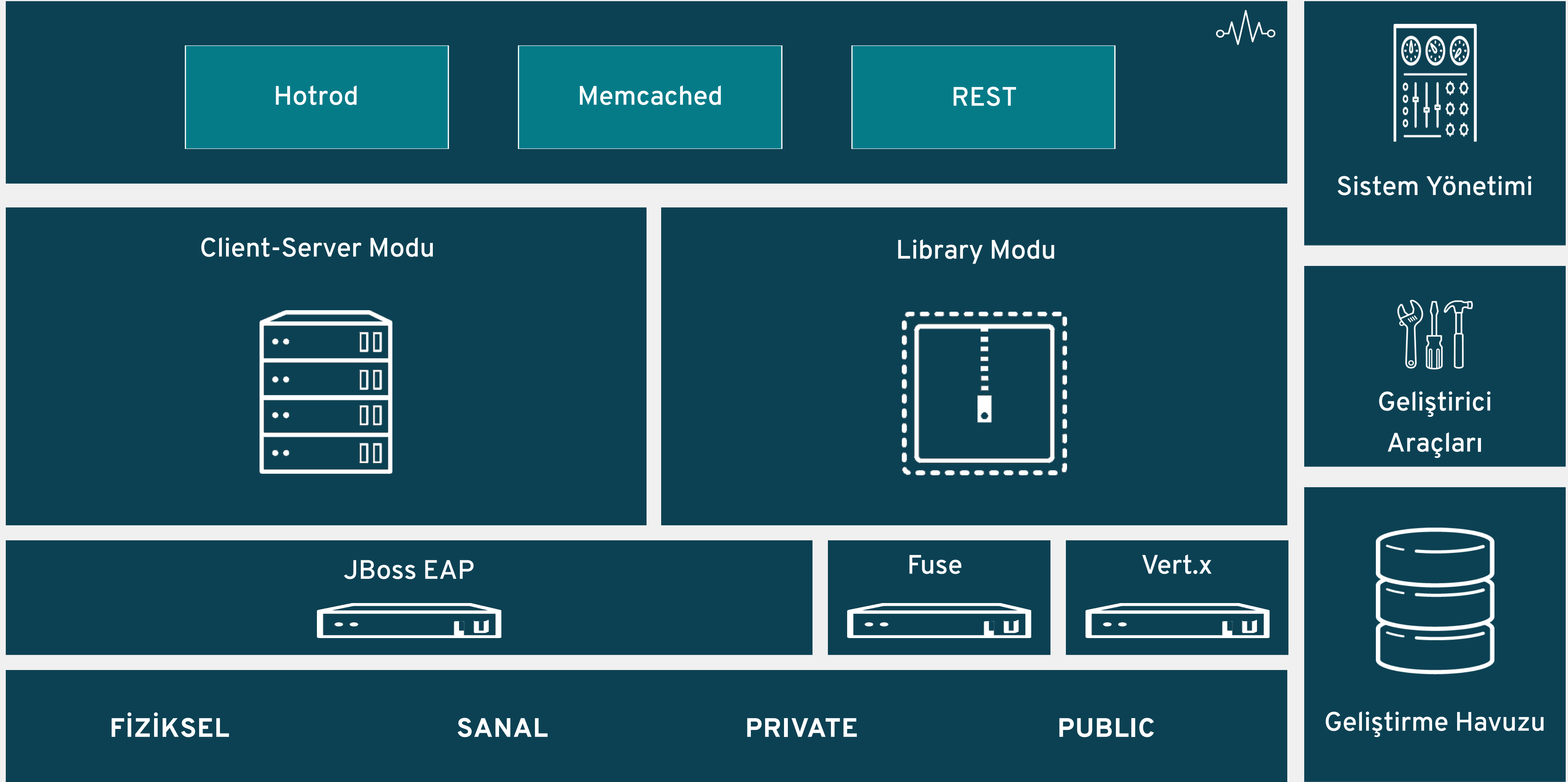
Büyük Veri/İoT Analitiği

Java Stream API ile basitleştirilmiş MapReduce işlemi

Spark ve Hadoop ile entegrasyon



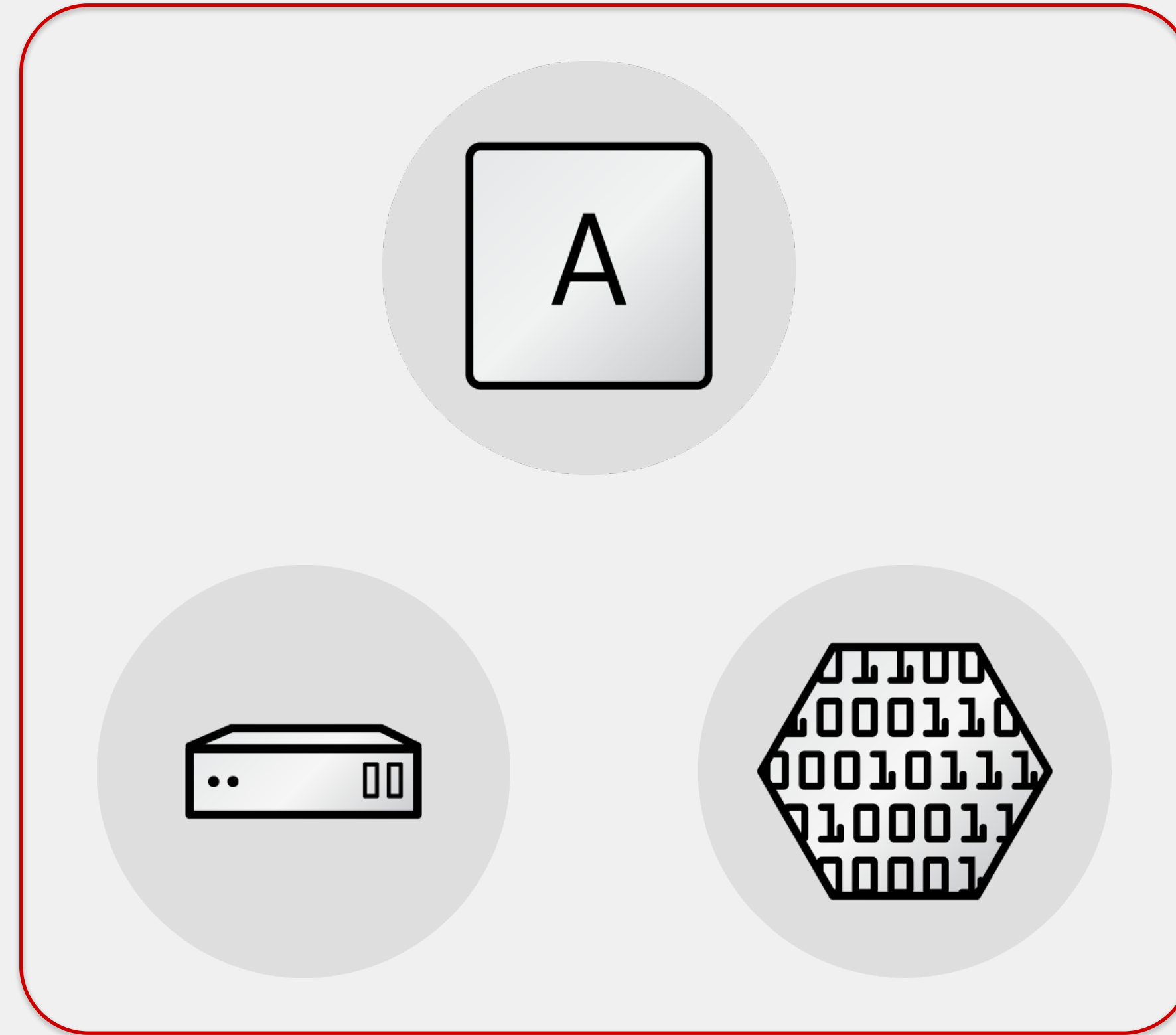
Mimariye Genel Bakış





Mimariye Genel Bakış

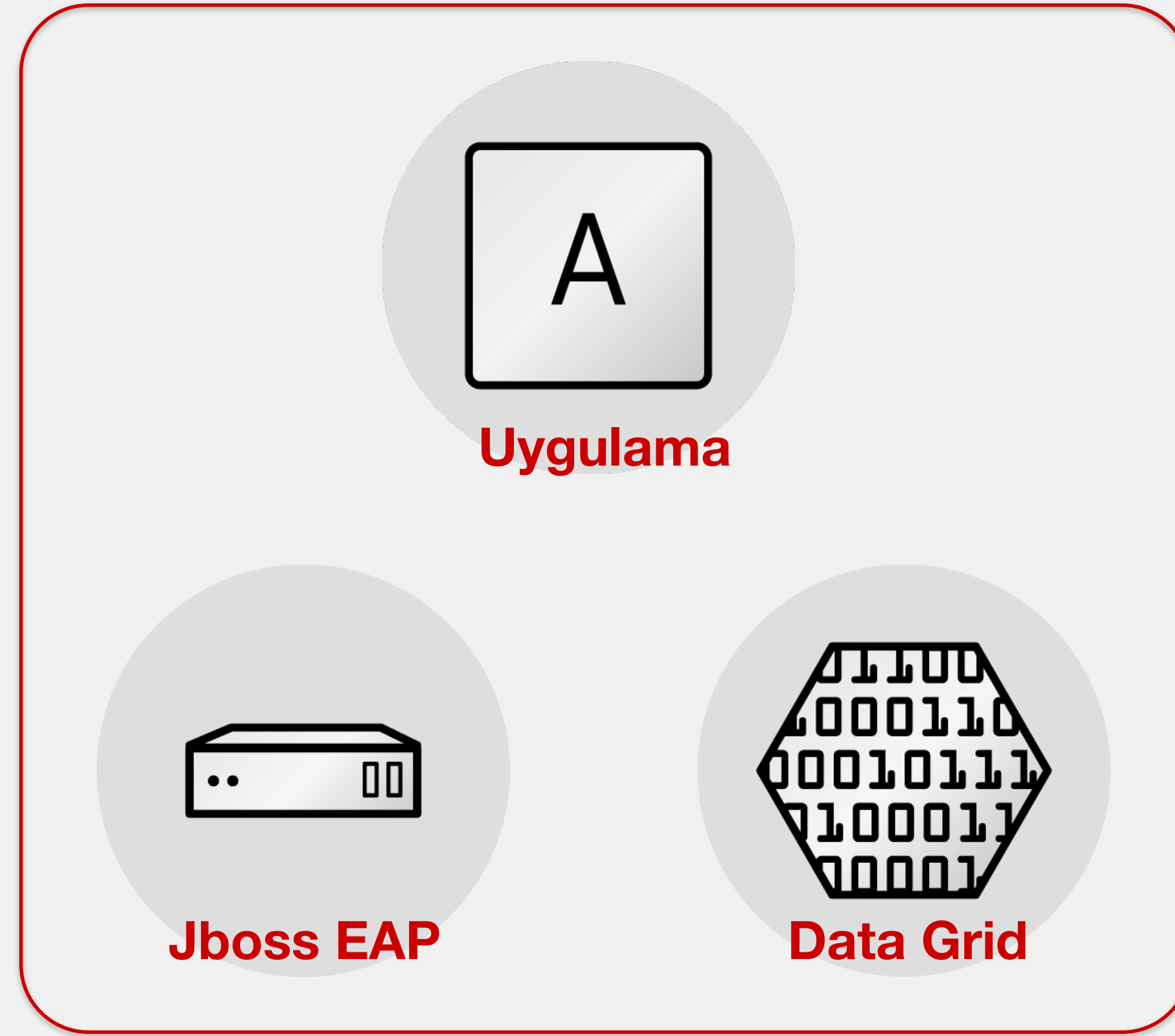
Standalone Çalışma Modu





Mimariye Genel Bakış

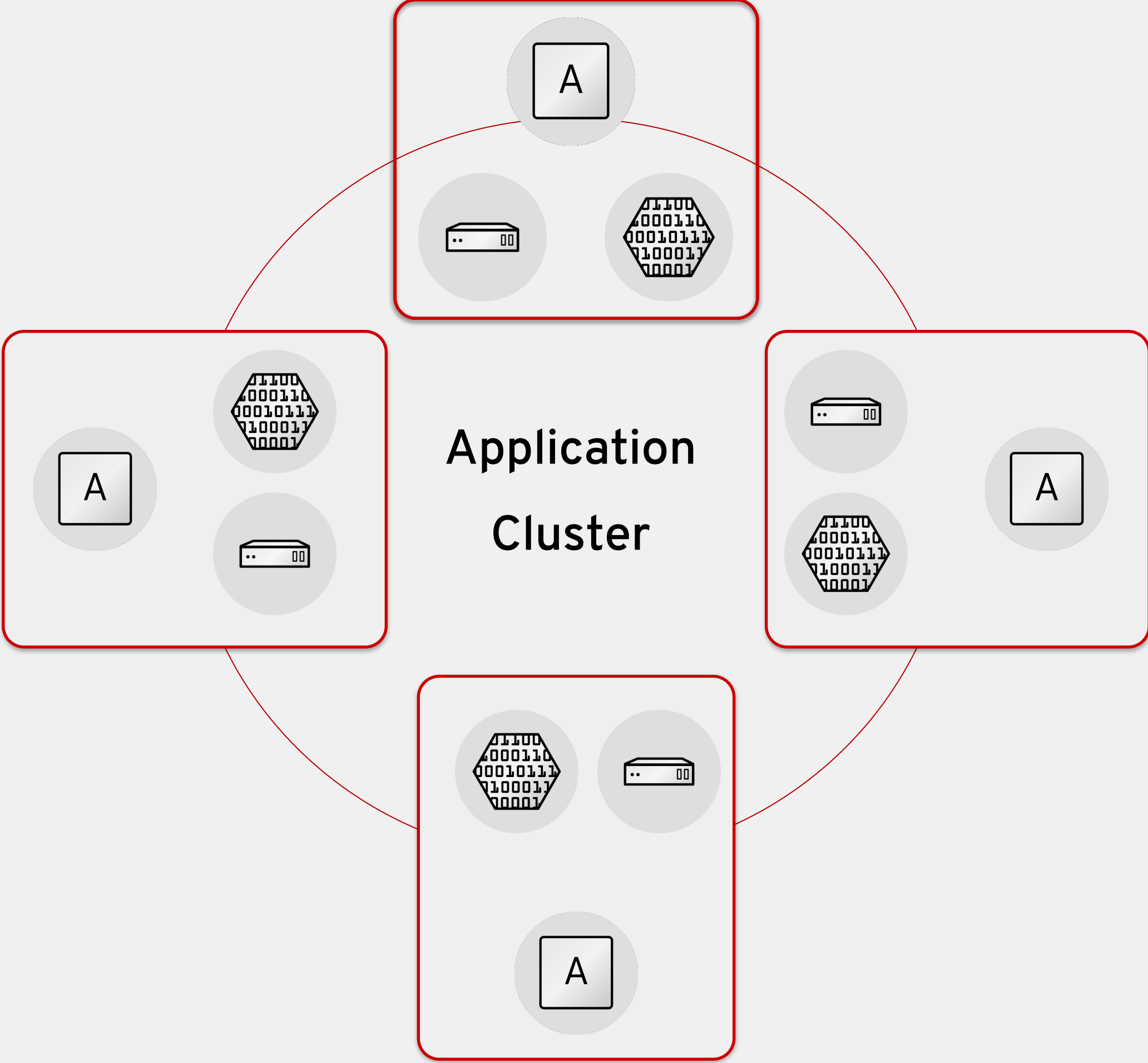
Standalone Çalışma Modu





Mimariye Genel Bakış

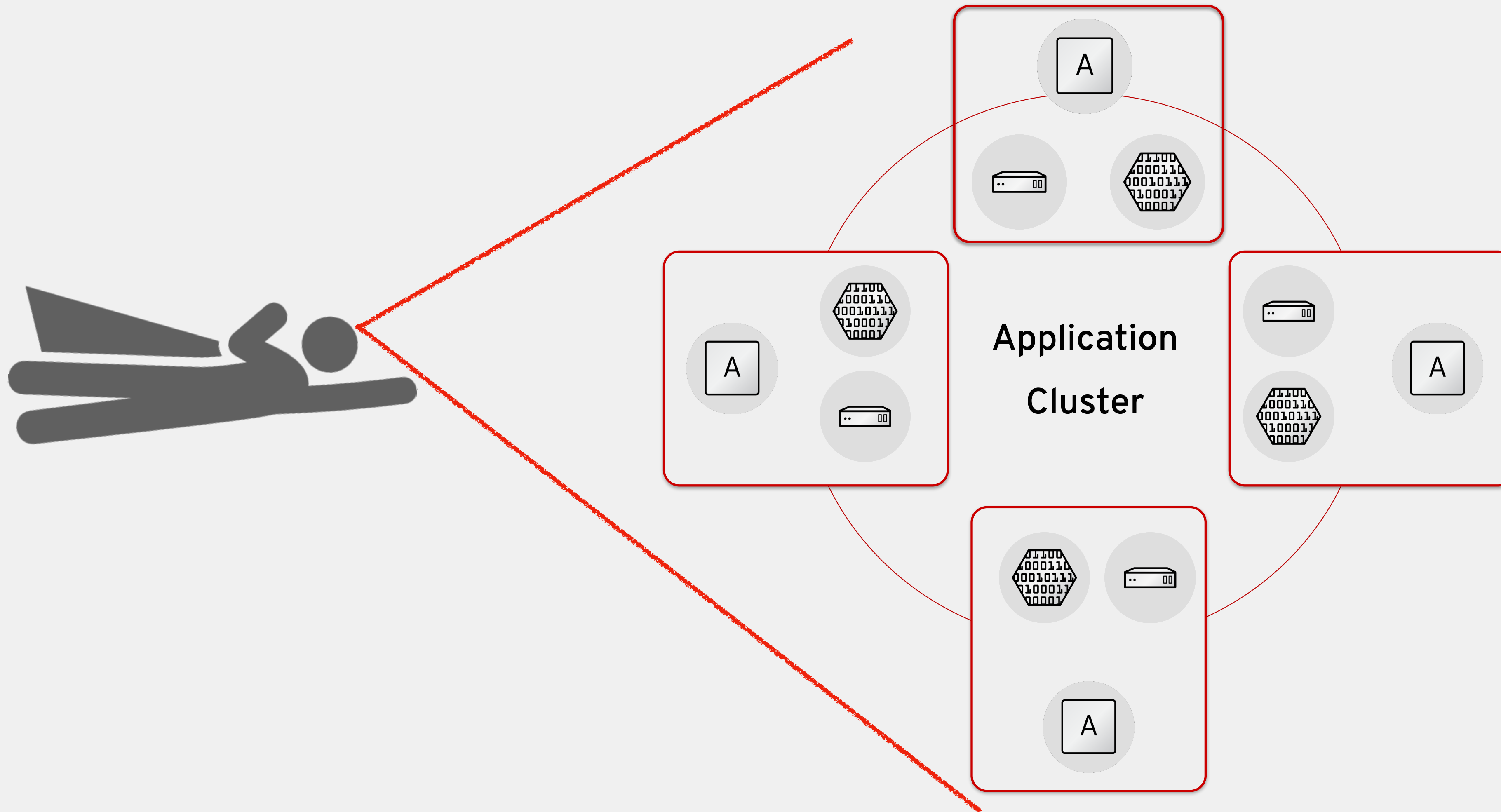
Library Modu





Mimariye Genel Bakış

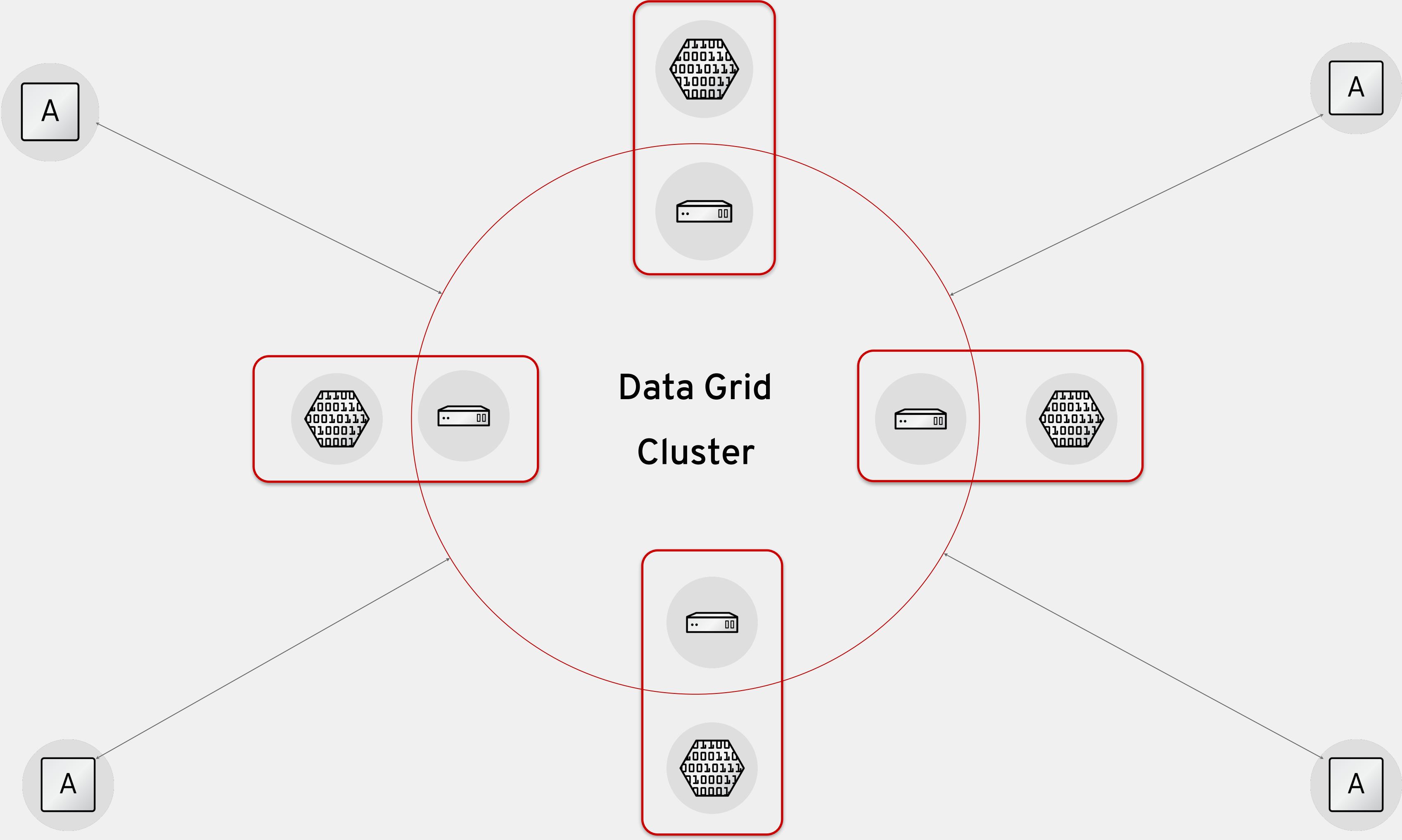
Library Modu





Mimariye Genel Bakış

Remote Client-Server Modu





Remote Client-Server Modu

Protokol ve Dil Desteđi

Client Protokolü	Client Dili	Format	Akıllı yönlendirme	Load balancing ve failover
Hot Rod	Java, C++, C#, Node.js	Binary deđer	Yes	Dinamik
REST	Herhangi biri	Text deđer	No	Herhangi bir HTTP load balancer
Memcached	Herhangi biri	Text deđer	No	Öntanımlı server listesi

Data Grid Kullanan müşterilerin %90'dan fazlası Remote Client-Server modunda erişim için gözle görülür biz hız sağlayan Hot Rod protokolünü tercih etmekte



Red Hat Jboss Data Grid

4 Ana Özellik

Dağıtık Önbellek

En sık erişilen dataları tutabilmek için sağlanan in-memory veri deposu

Geçişken, kısa-zamanlı veri depolama imkanı

NoSQL Veritabanı

NoSQL Key-Value veri deposu

Konfigüre edilebilir ACID transaction desteği

Olay Dinleyiciler

Veri üzerindeki CRUD işlemlerini dinleme ve tepki verme yeteneği

Süregelen sorgular ile her zaman aynı sonuç kümesi dönmesi sağlanır

Büyük Veri/IoT Analitiği

Java Stream API ile basitleştirilmiş MapReduce işlemi

Spark ve Hadoop ile entegrasyon



Red Hat Jboss Data Grid

Dağıtık Önbellek

NoSQL Veritabanı

Olay Dinleyiciler

Büyük Veri/loT Analitiği



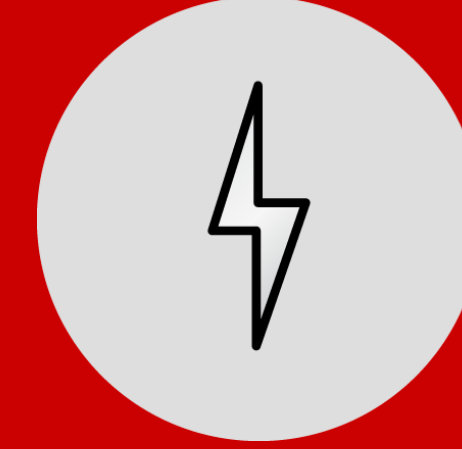
Red Hat Jboss Data Grid

4 Ana Özellik

Dağıtık Önbellek



Olay Dinleyiciler



NoSQL Veritabanı

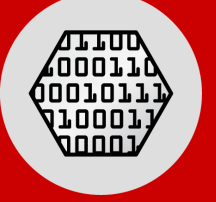


Büyük Veri/IoT Analitiği

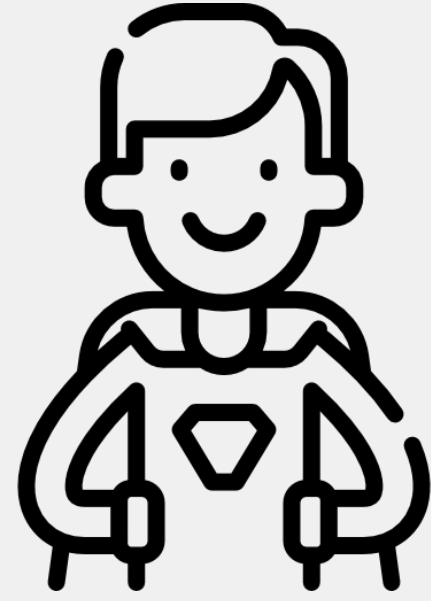




Dağıtık
Önbellek



Dağıtık Önbellek

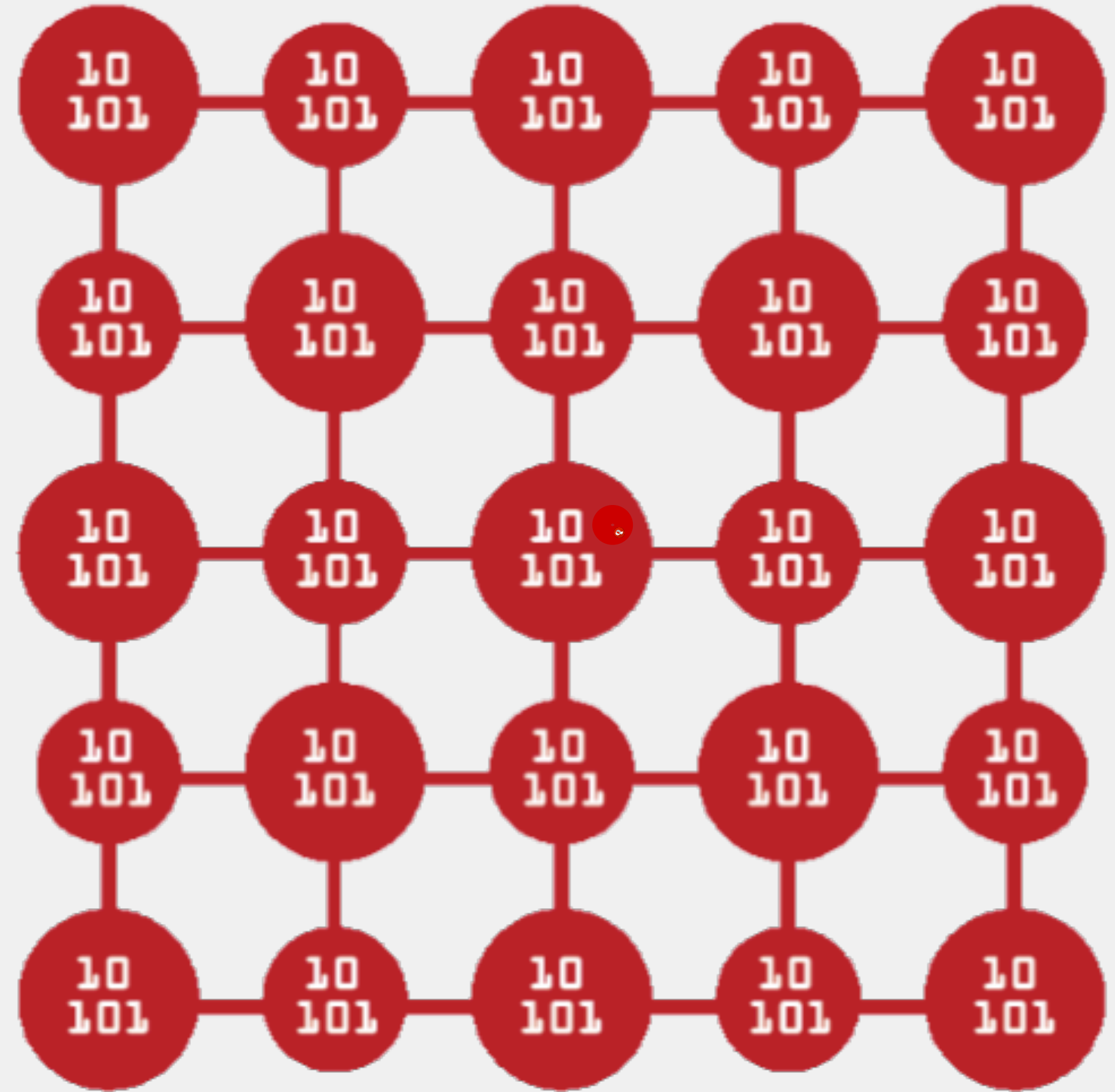


Tamamıyla Peer-to-Peer, Master/Slave yok

Single Point of Failure yok

Lineer olarak yüzlerce node'a ölçeklenebilir

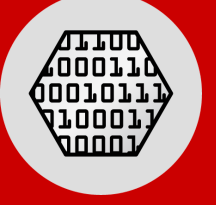
Basit donanım ile çalışabilir





Önbellek (Caching) Modları

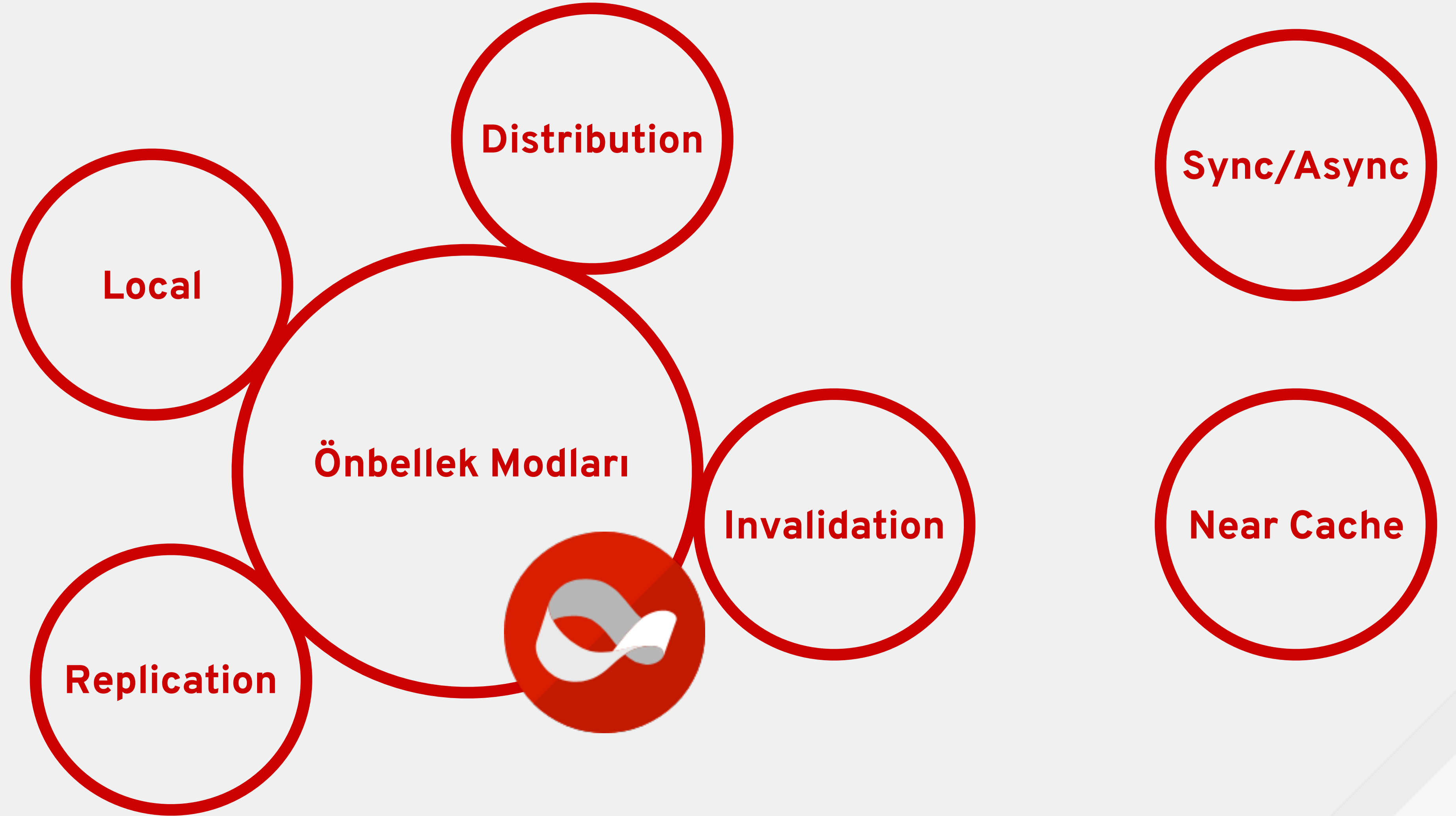
Dağıtık
Önbellek





Önbellek (Caching) Modları

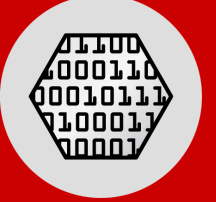
Dağıtık
Önbellek



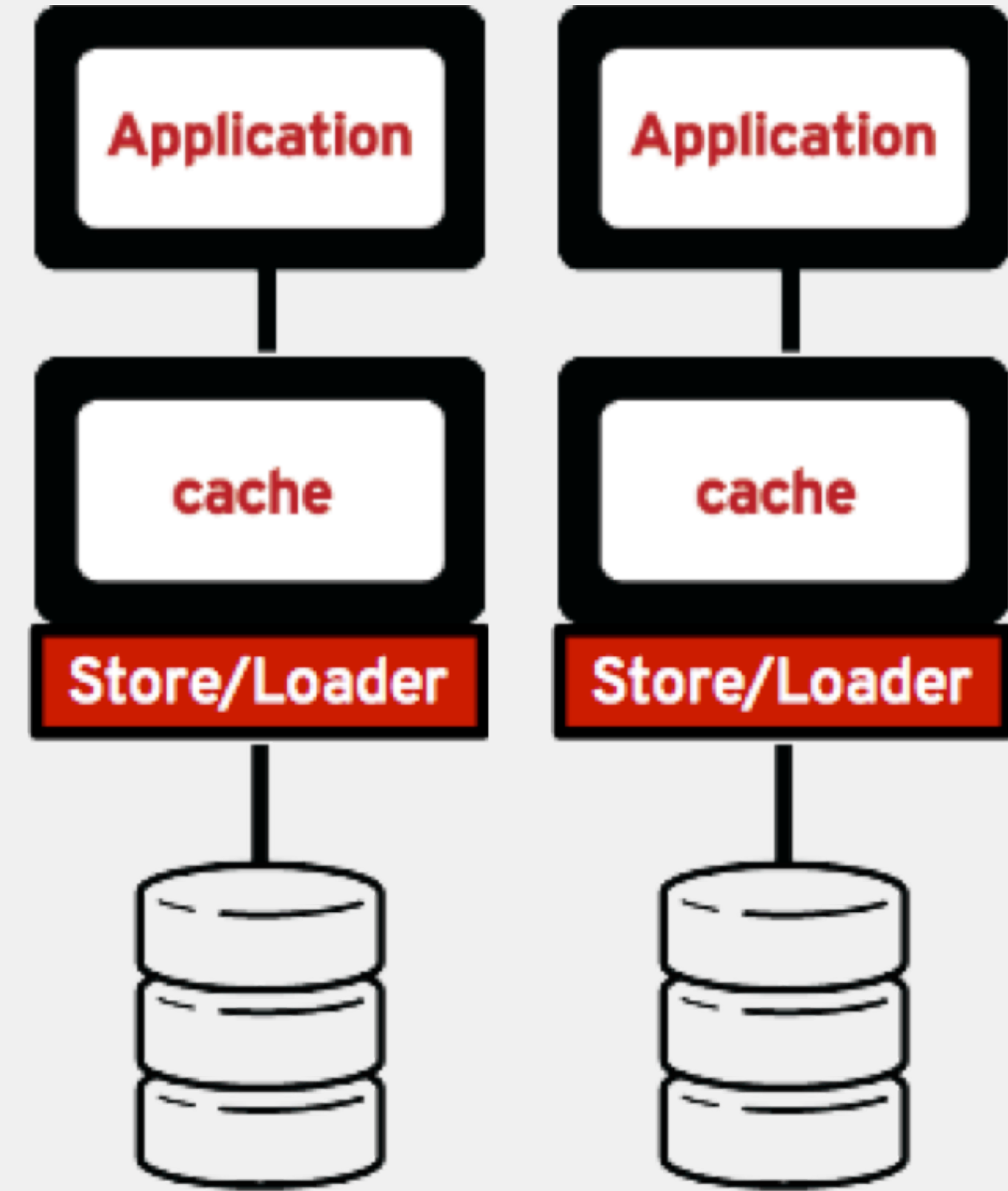


Önbellek (Caching) Modları

Dağıtık
Önbellek



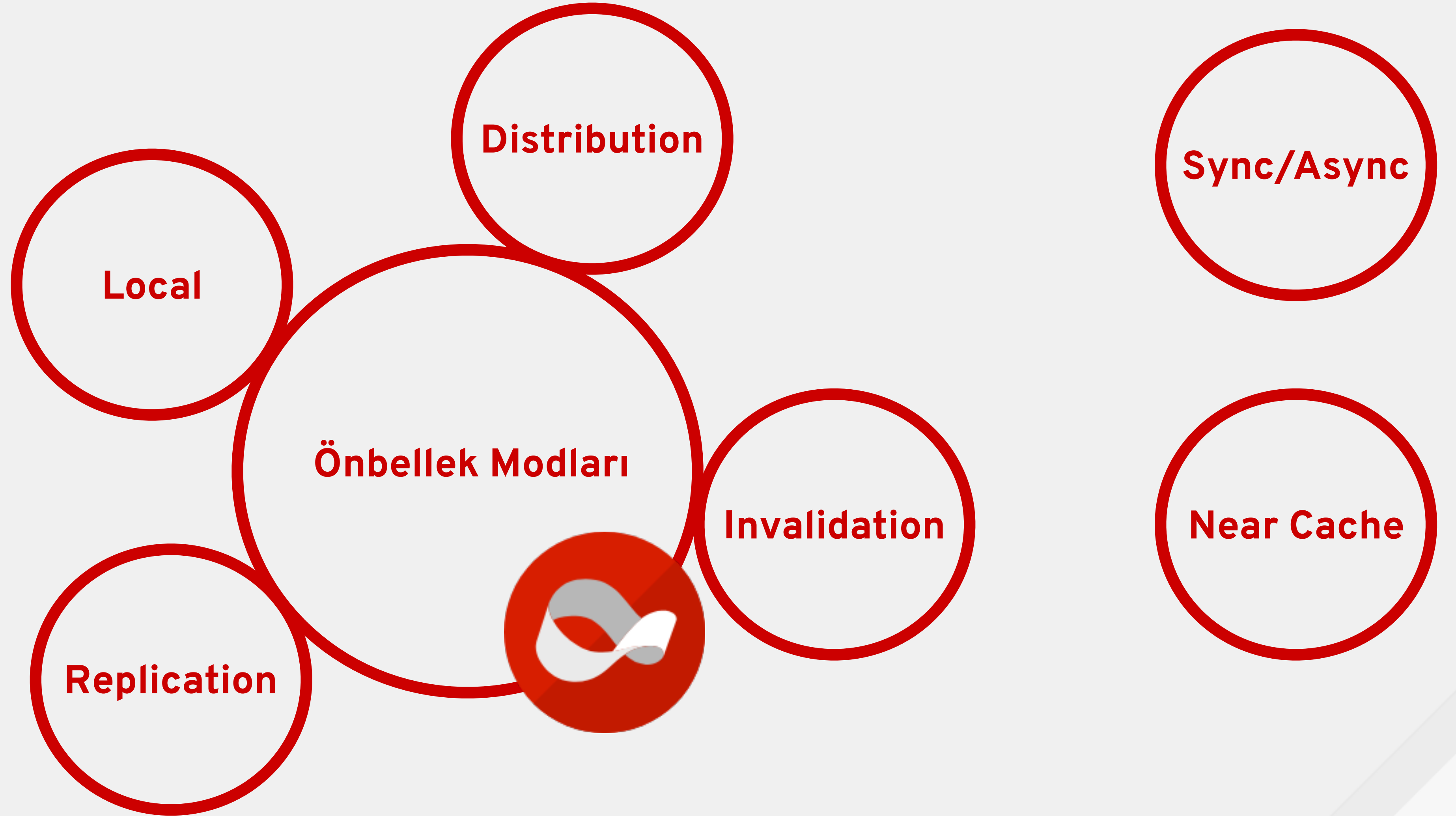
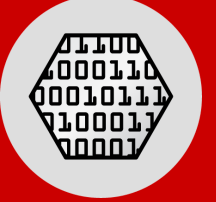
Basit in-memory, cluster içinde olmayan önbellek





Önbellek (Caching) Modları

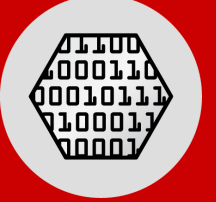
Dağıtık
Önbellek



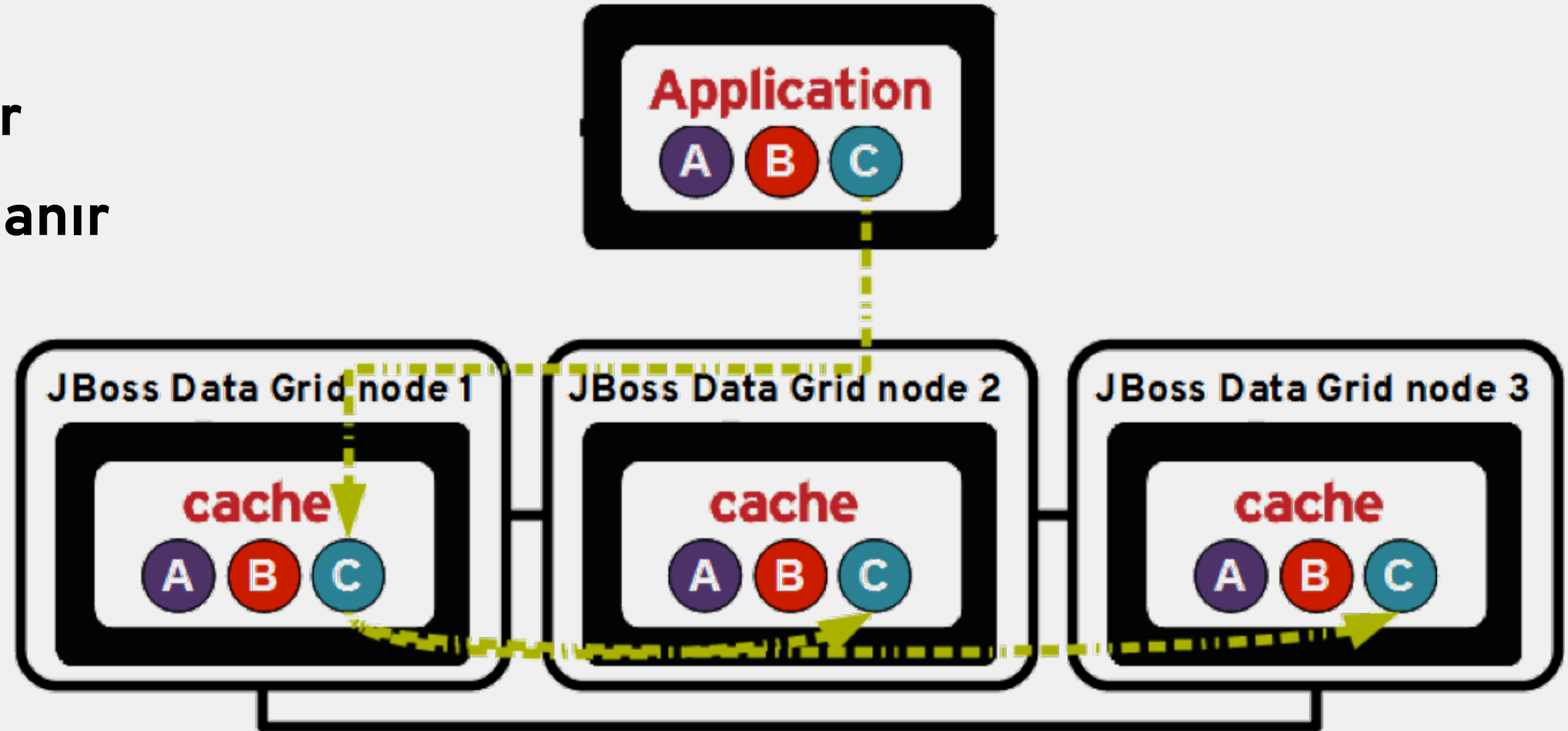


Önbellek (Caching) Modları

Dağıtık
Önbellek



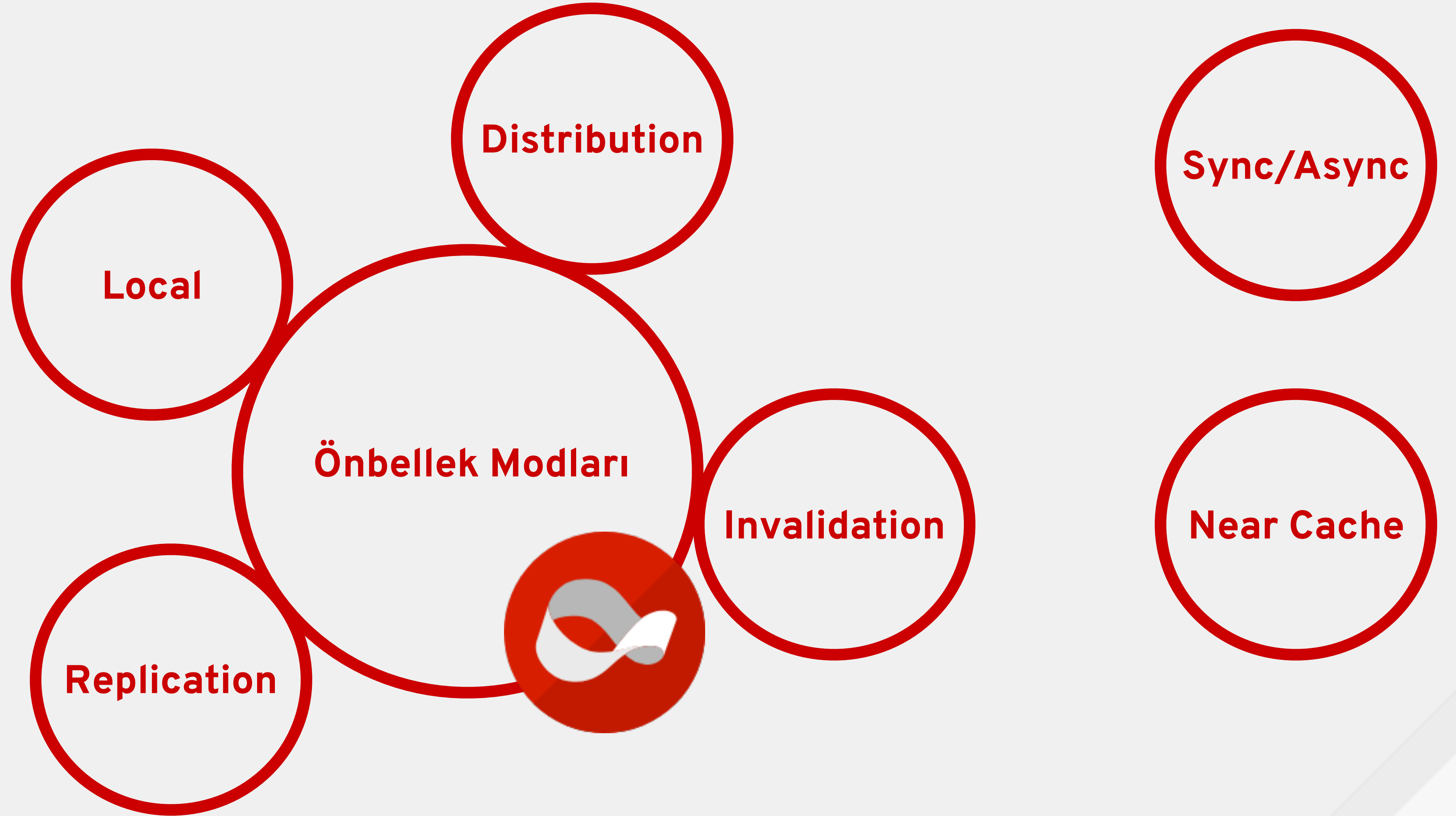
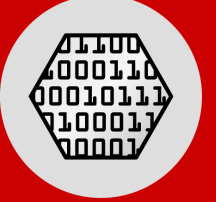
Bütün önbellek değerleri cluster
üzerindeki tüm nodelara kopyalanır





Önbellek (Caching) Modları

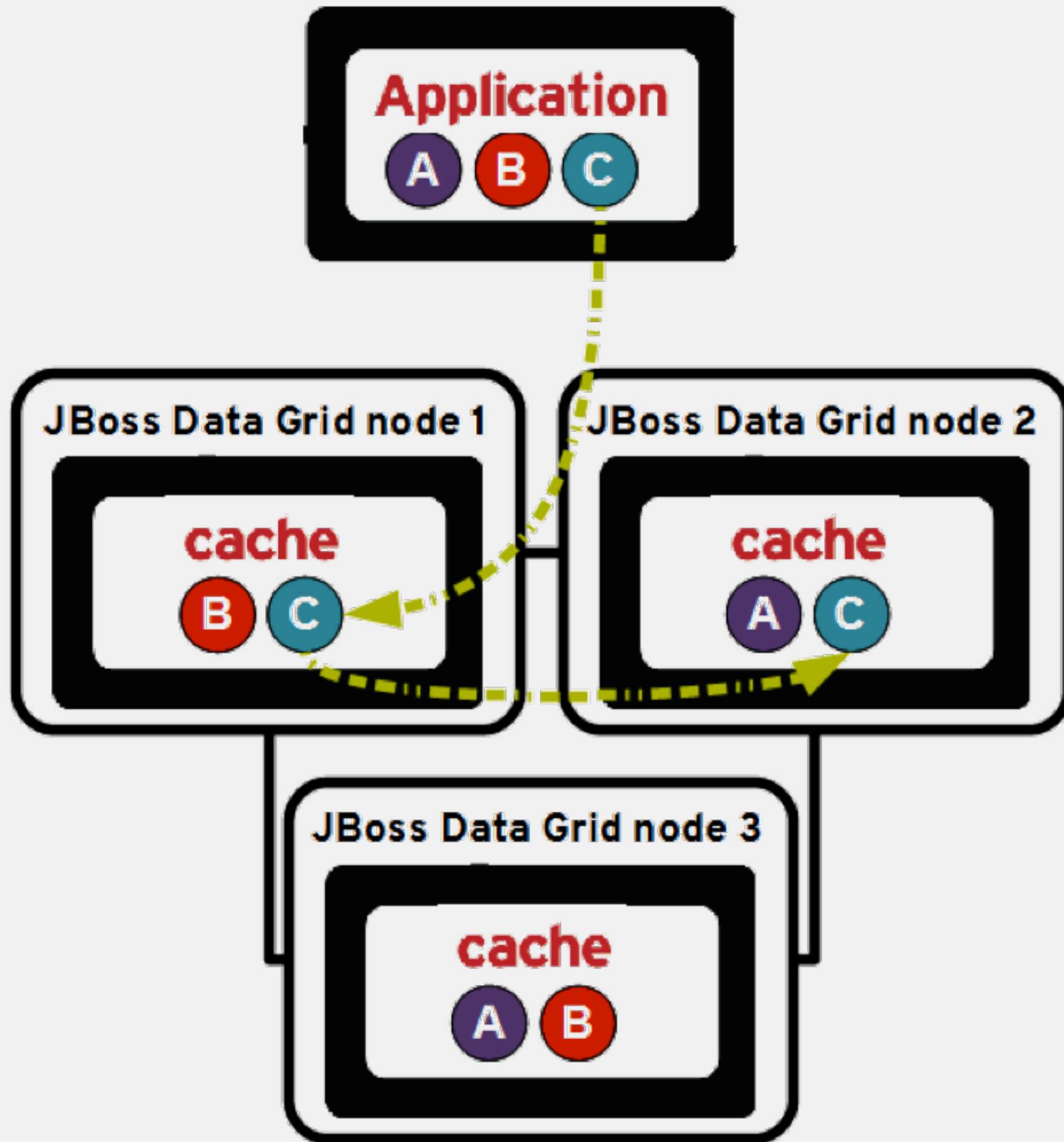
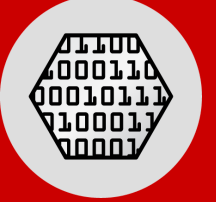
Dağıtık
Önbellek





Önbellek (Caching) Modları

Dağıtık
Önbellek



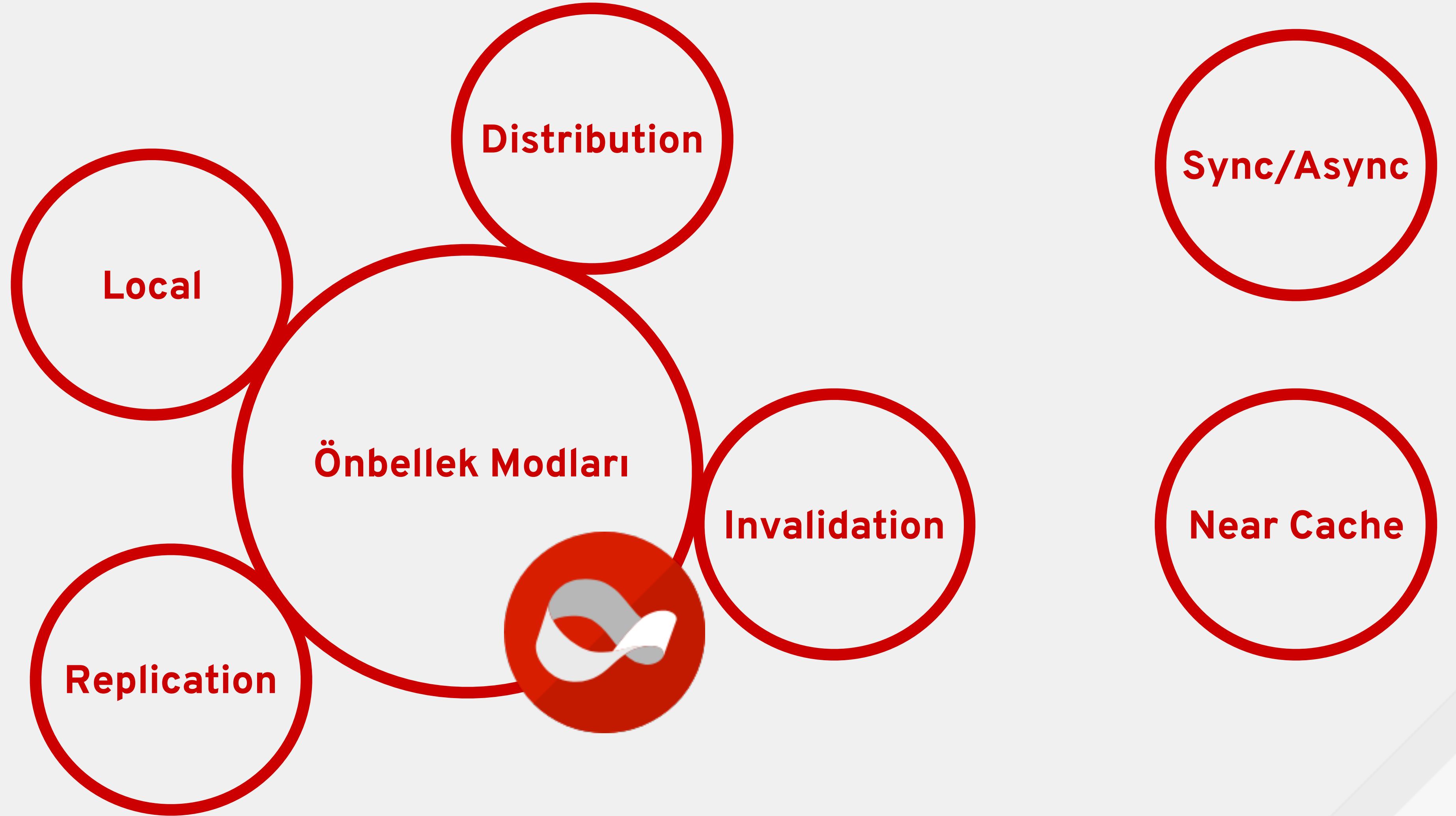
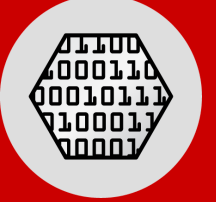
Distribution

Nodlar arası önbellek verisi dağılarak tutulur.
Hata toleranslı ve ölçeklenebilirdir



Önbellek (Caching) Modları

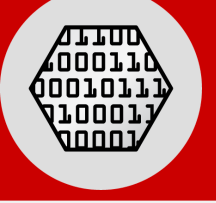
Dağıtık
Önbellek





Önbellek (Caching) Modları

Dağıtık
Önbellek



Paylaşılan bir önbellek değeri yok.

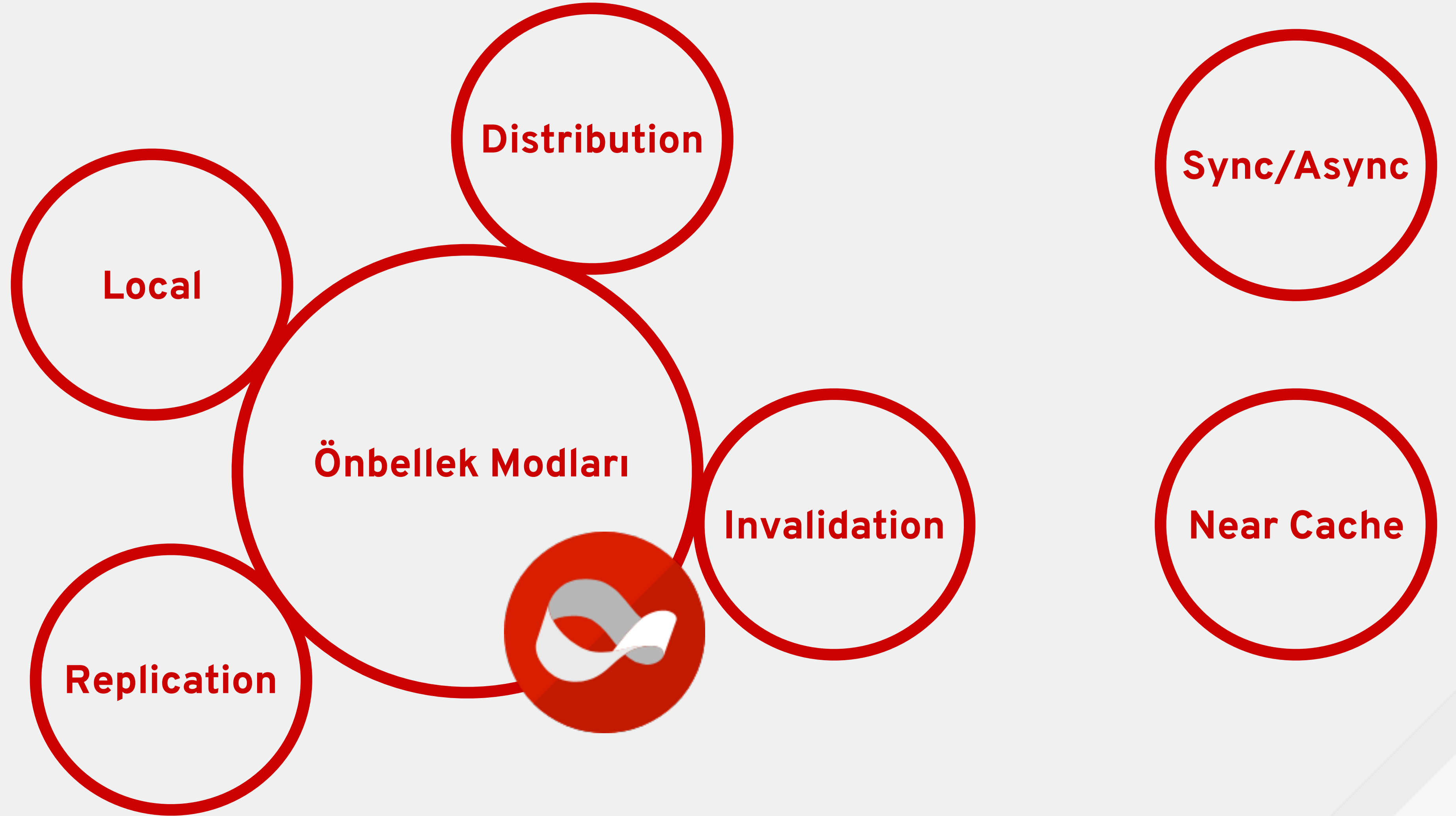
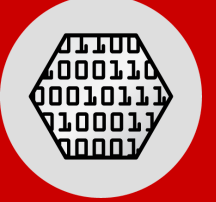
Verilerin nodelar arası invalidasyonunu sağlar

Invalidation



Önbellek (Caching) Modları

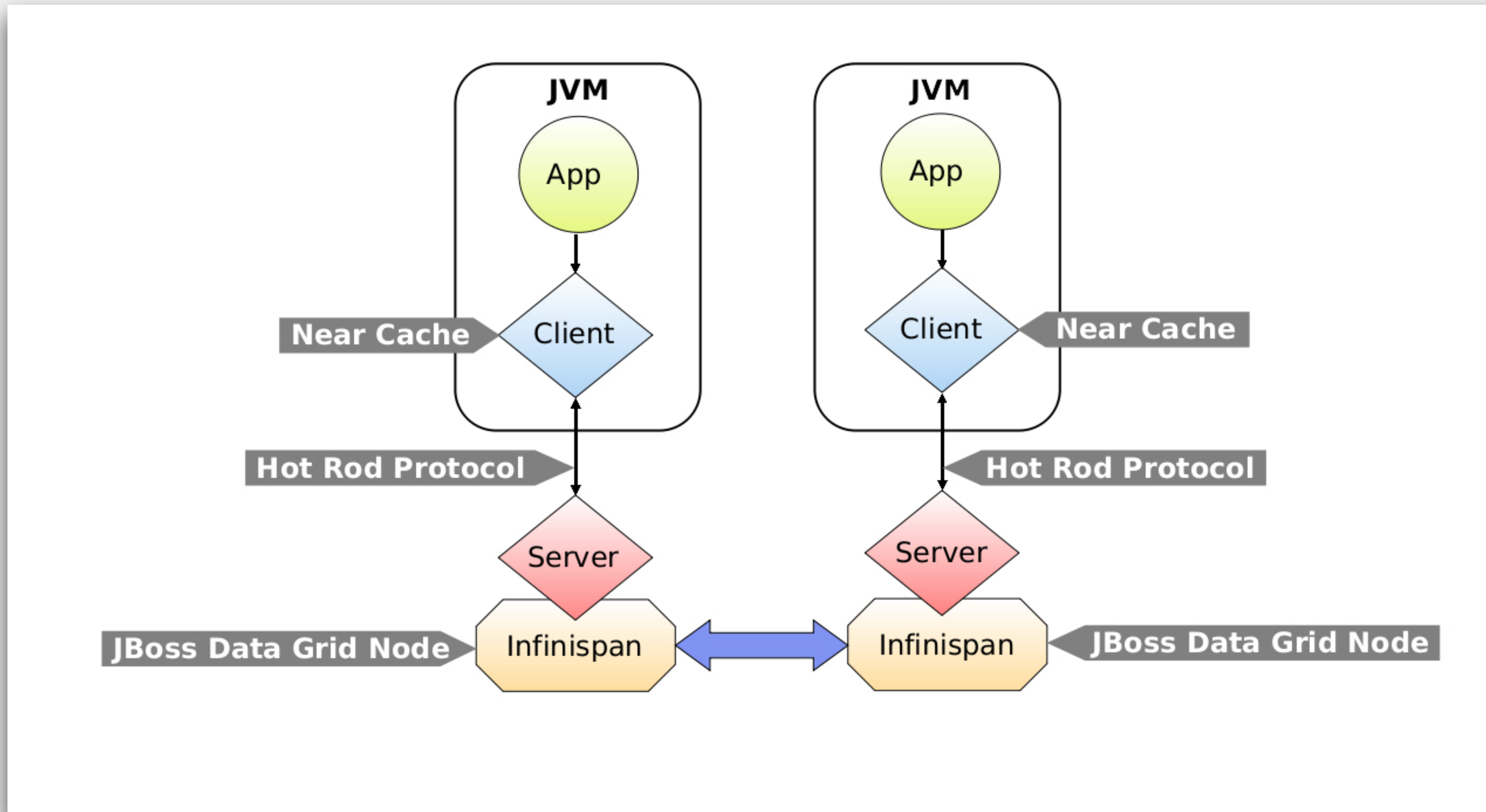
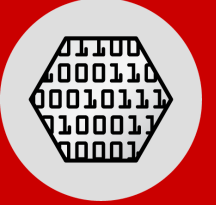
Dağıtık
Önbellek





Önbellek (Caching) Modları

Dağıtık
Önbellek

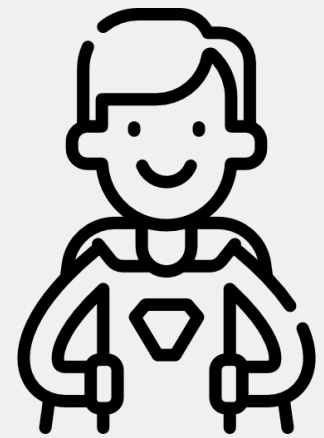
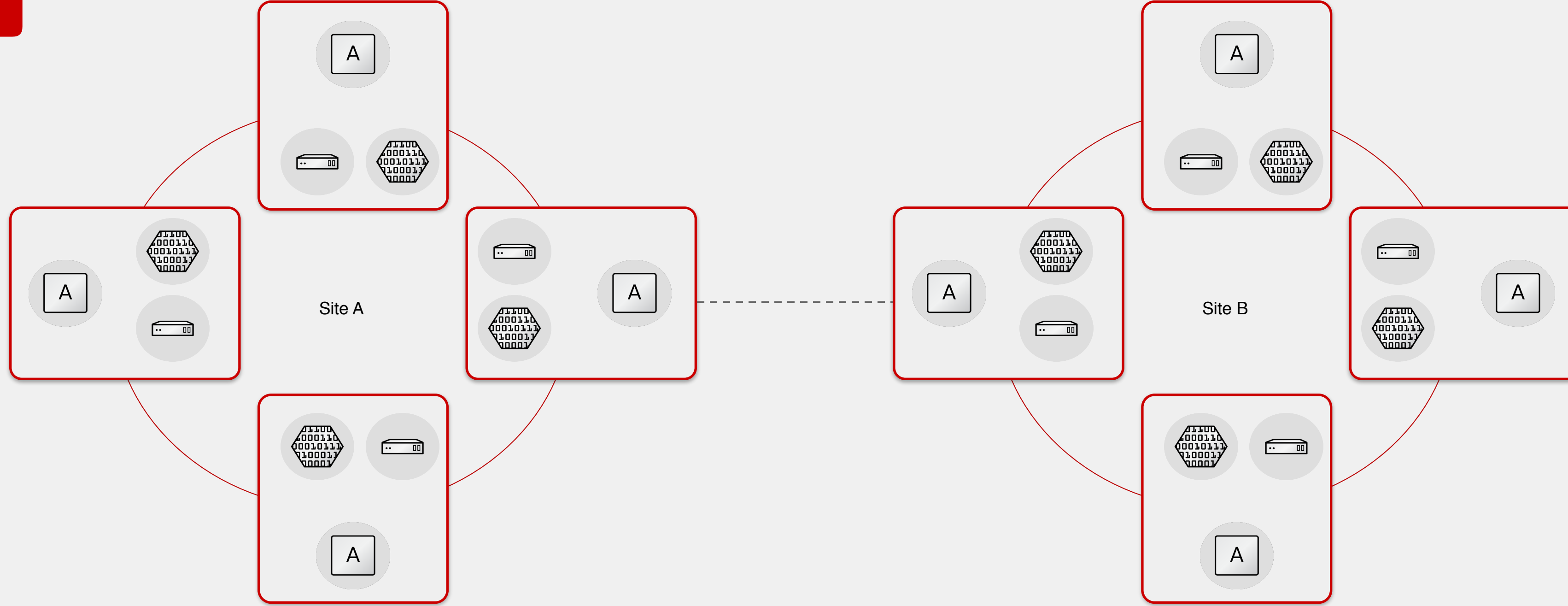
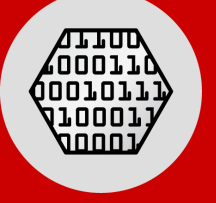


<https://access.redhat.com>



Veri Merkezleri Arası Replikasyon

Dağıtık
Önbellek



Coğrafyalar arası Aktif-Aktif replikasyon

Hem senkron hem de asenkron replikasyon desteği

Veri merkezinin offline olma durumunda online olduktan sonra re-sync olabilme özelliği

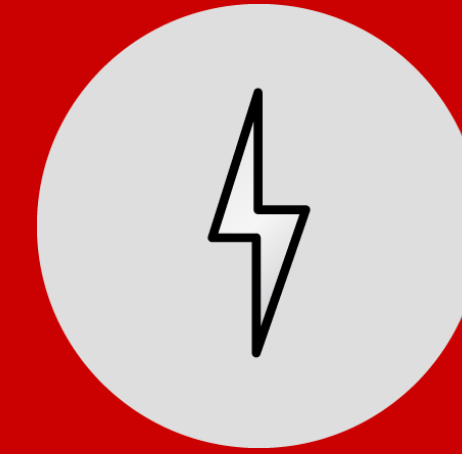


Red Hat Jboss Data Grid

Dağıtık Önbellek



Olay Dinleyiciler



NoSQL Veritabanı

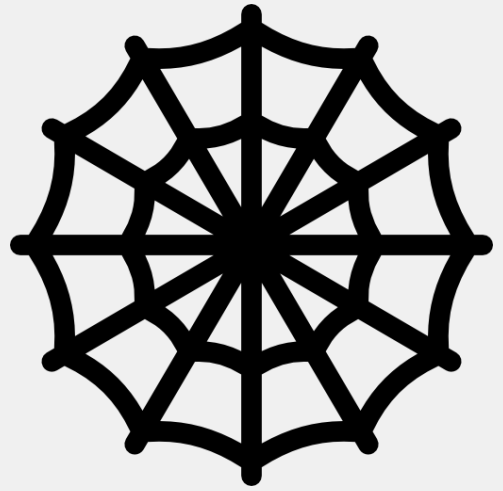


Büyük Veri/IoT Analitiği





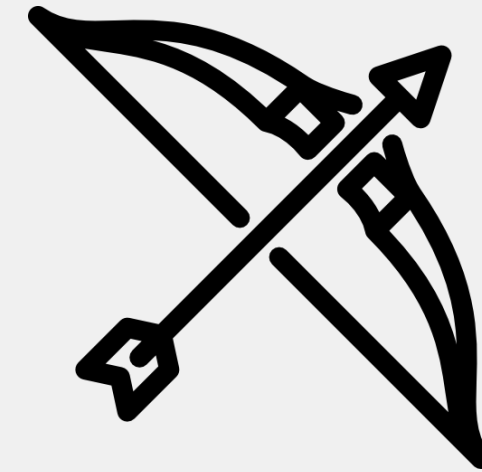
NoSQL Veritabanı



EXPIRATION & EVICTION

Ölümsüz ve ölümlü veriler

Girdi sayısı veya kullanılan belleğe göre



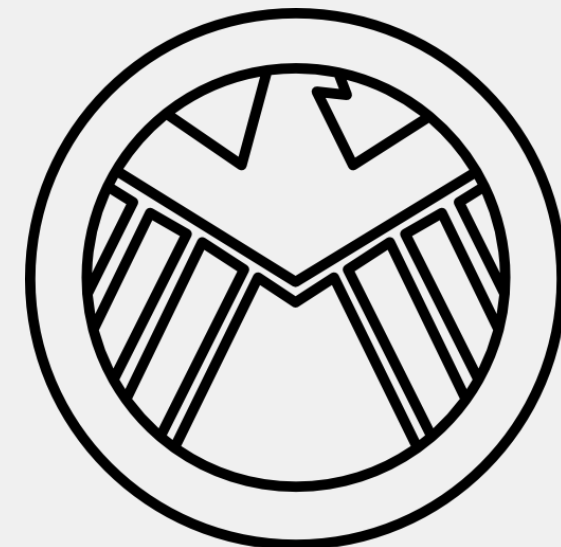
TRANSACTIONS & PERSISTENCE

XA desteği, Recovery ve konfigüre edilebilir

ACID transactions

Seçimli pasivasyon

Dosya, JDBC, JPA, LevelDB...vb.

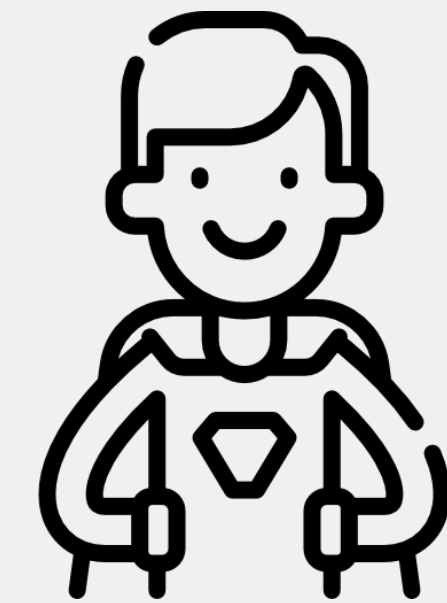


GÜVENLİK & OTORİZASYON

Coarse-grained

JAAS

Güvenlik Poliçeleri



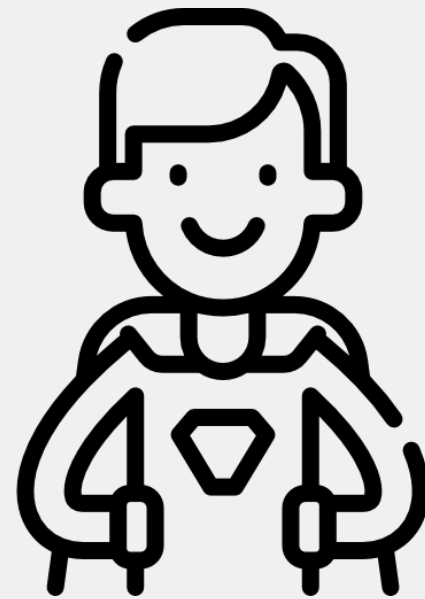


Sorgular



EMBEDDED SORGULAR

Lucene & Hibernate aramayı kullanır
Lucene Sorguları ve Query DSL
Aggregate'ler, gruplama, sıralama
Dinamik indeksleme



REMOTE SORGULAR

Çapraz dil uyumu için protobuf
şemaları kullanımı
Query DSL

```
QueryBuilder qb = Search.getSearchManager(cache)
    .buildQueryBuilderForClass(Book.class).get();

org.apache.lucene.search.Query query = qb
    .keyword().onFields("title", "author")
    .matching("Java rocks!").createQuery();
CacheQuery cacheQuery =
    Search.getSearchManager(cache).getQuery(query);

List list = cacheQuery.list();
```



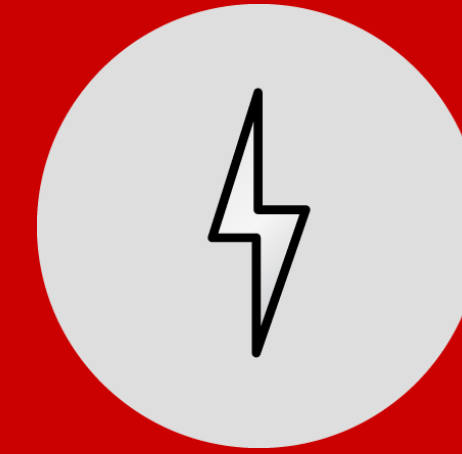


Red Hat Jboss Data Grid

Dağıtık Önbellek



Olay Dinleyiciler



NoSQL Veritabanı



Büyük Veri/IoT Analitiği





Olay Dinleyiciler



Olay
Dinleyiciler



LOCAL/CLUSTER DİNLEYİCİLERİ

Local veya clusterdaki değişiklikleri algılar
CDU olayları, Expiration olayları



REMOTE DİNLEYİCİLER

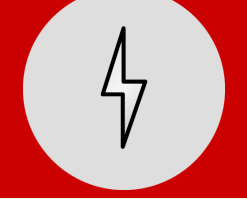
Remote serverlardaki değişiklikleri algılar
Özelleşmiş server-side filtre veya çeviriciler

```
@Listener
public class PrintWhenAdded {
    @CacheEntryCreated
    public void print(CacheEntryCreatedEvent e) {
        System.out.printf(
            "New entry %s created in local cache",
            e.getKey());
    }
}
```

```
@ClientListener
public class RemotePrintWhenAdded {
    @ClientCacheEntryCreated
    public void print(ClientCacheEntryCreatedEvent
e) {
        System.out.printf(
            "New entry %s created in remote cache",
            e.getKey());
    }
}
```



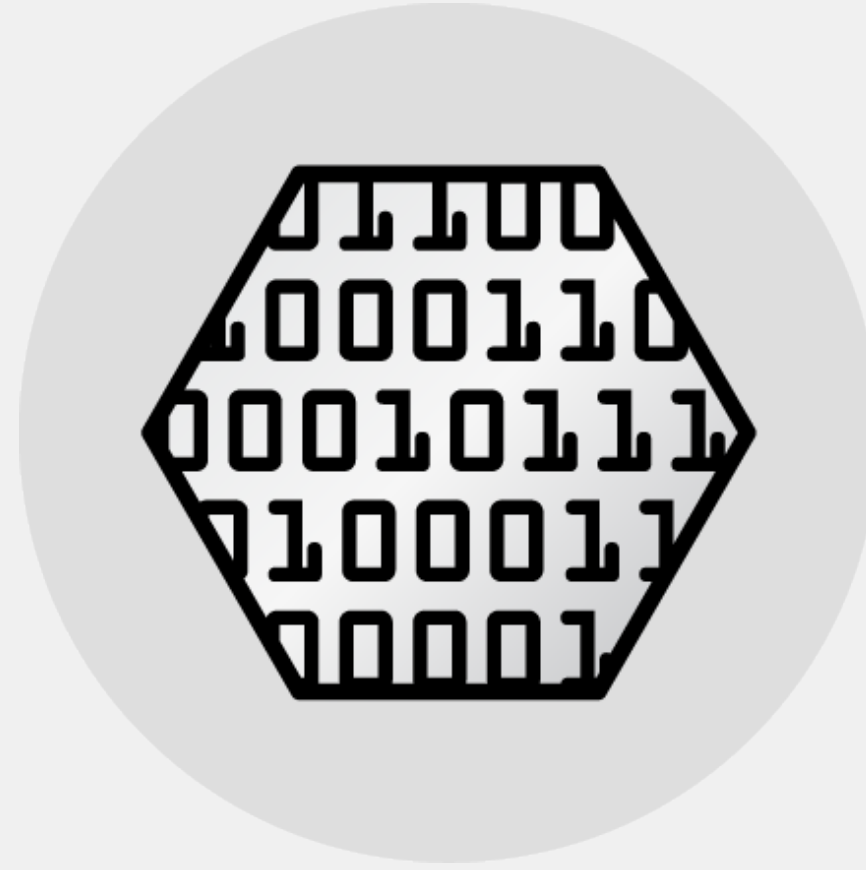
Süregelen Sorgular



Olay
Dinleyiciler



Sorgu oluřtur



Sonucu elde et



Olay sonucu dönünce:
Veri seti deęiřir



Süregelen Sorgular



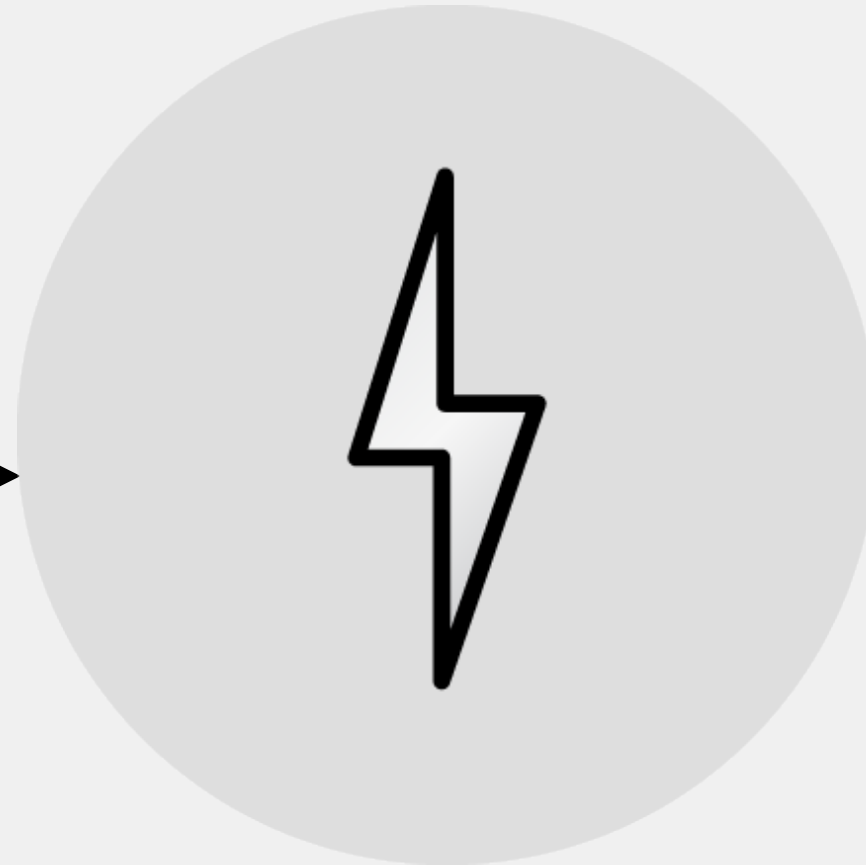
Olay
Dinleyiciler



Sorgu oluştur



Sonucu elde et



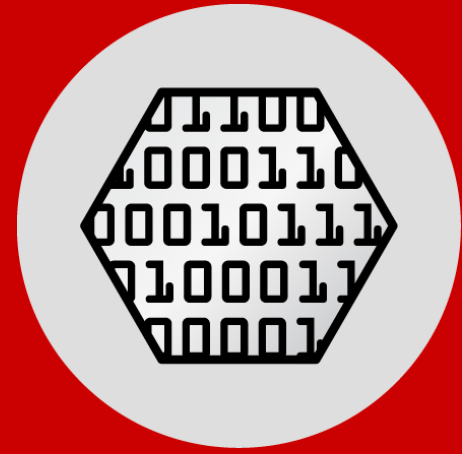
Olay sonucu dönünce:
Veri seti değişir

Embedded ve remote sorguların ikisi de desteklenmekte

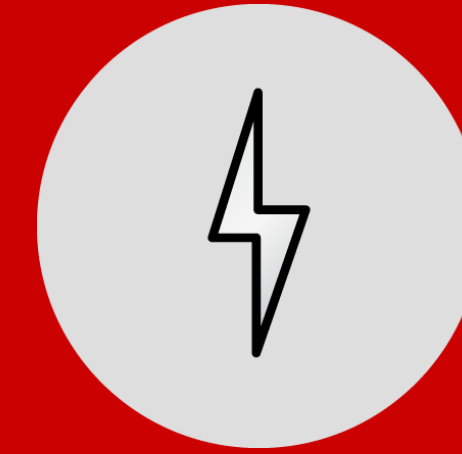


Red Hat Jboss Data Grid

Dağıtık Önbellek



Olay Dinleyiciler



NoSQL Veritabanı

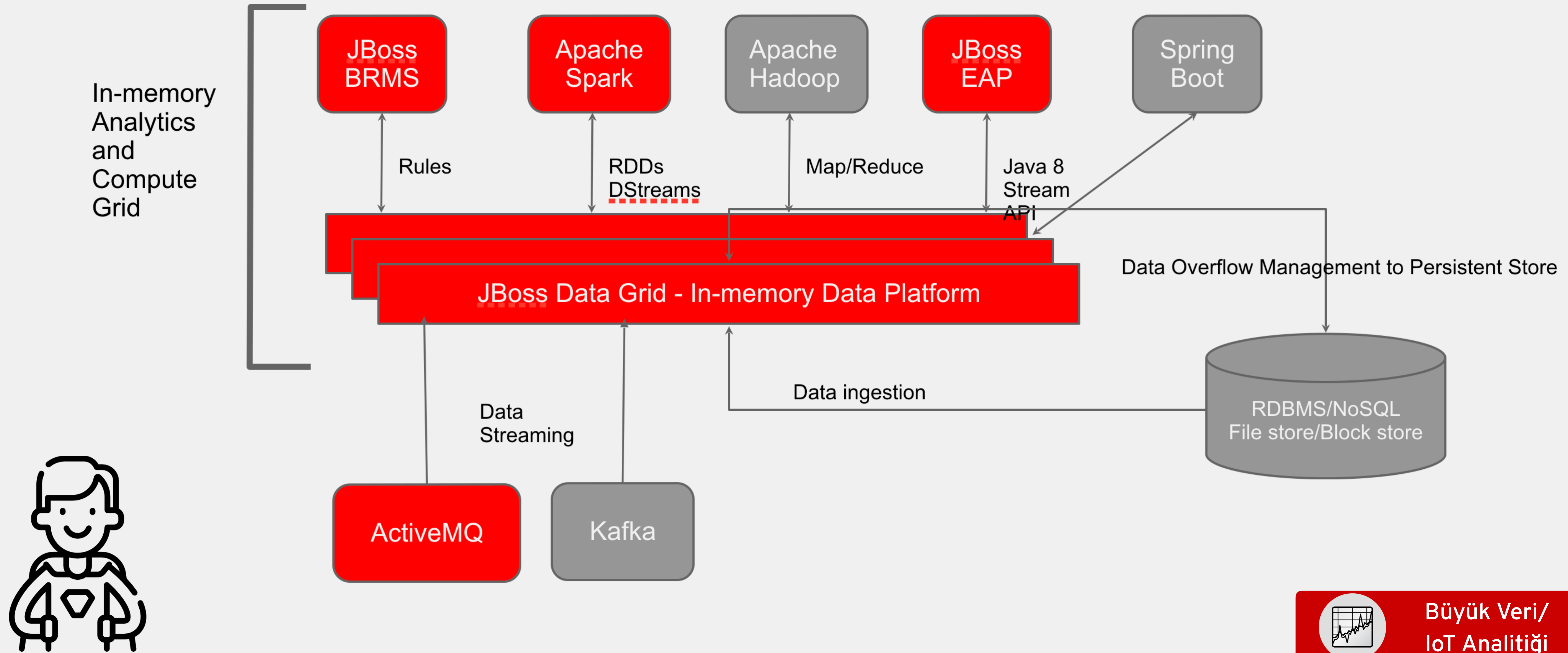


Büyük Veri/loT Analitiği



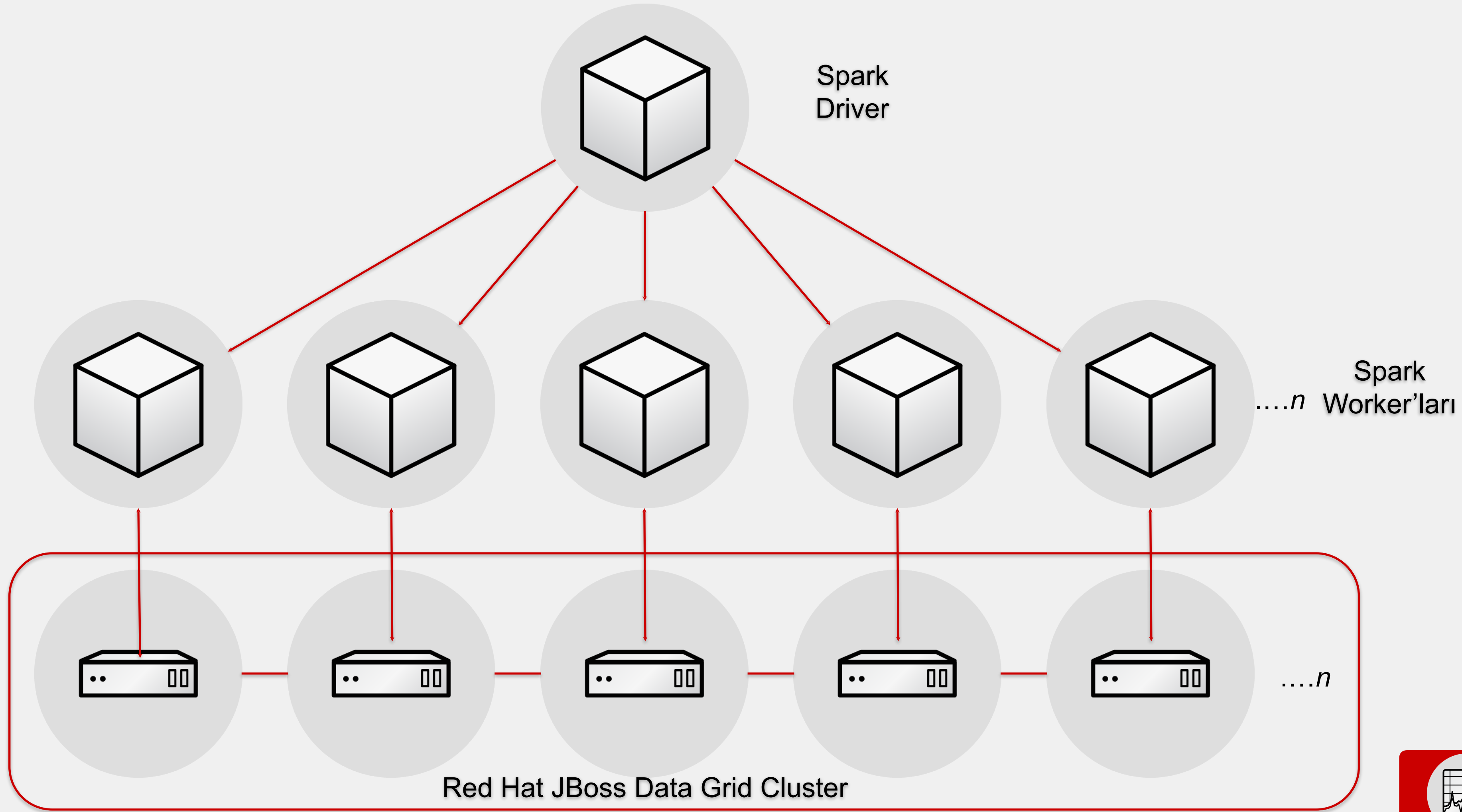


Büyük Veri/IoT Analitiği





Apache Spark Entegrasyonu



Büyük Veri/
IoT Analitiği

Kullanım Alanları



Web, E-Ticaret



**Eski Uygulama
Performans İyileřtirme**

Web, E-Ticaret



Telco

Finans

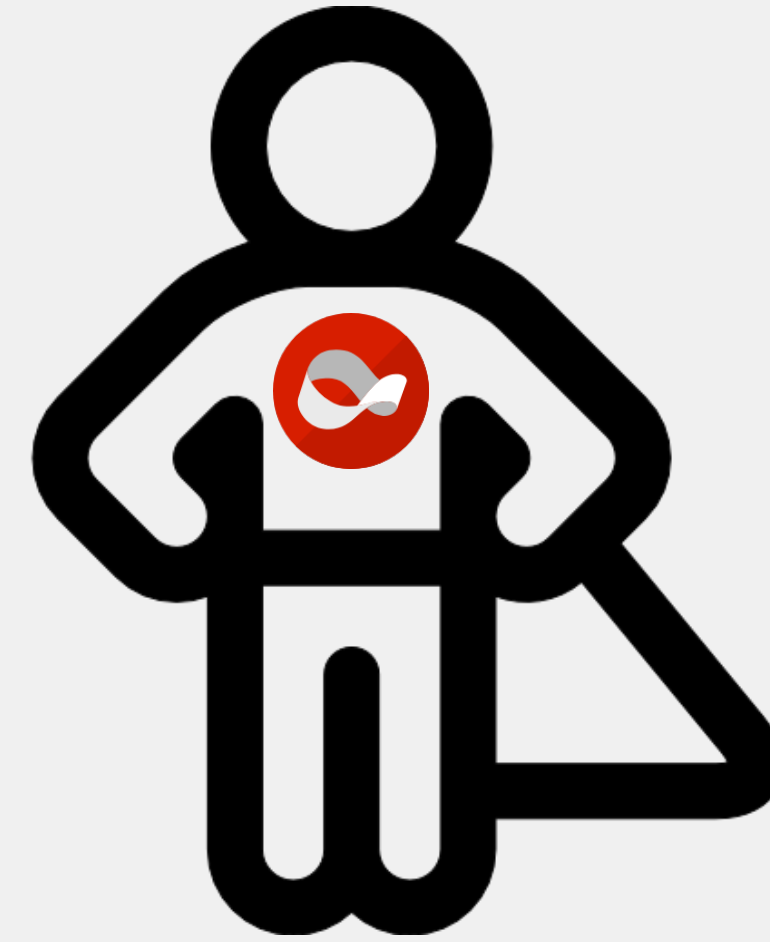
Sigortacılık

Savunma

Seyahat

**Eski Uygulama
Performans İyileştirme**

Web, E-Ticaret





RED HAT® JBOSS®
DATA GRID



Teşekkürler

@systemcraftsman