

Hochverfügbare Keycloak-Cluster

Digitale Woche Dortmund
11.11.2021

CONCISO.

✉ sven-torben.janus@conciso.de



@sventorben

Sven-Torben Janus

Partner, Principal Software Architekt



Agenda

Keycloak Overview

Cluster Basics

Distributed Caches

Pitfalls: Load Balancers

Session Revocation & Retention

Cross Data Center Replication

Cluster Updates & Deployments

Outlook: Keycloak.X



GENERAL

- Open source IAM solution
- Apache 2.0 license
- Under development since 2013
- First production-ready release in September 2014
- JBoss community project since 2018 under supervision of RedHat
- Upstream project for RH SSO
- Current release version: 15.0.2

PROTOCOLS

- OpenID Connect (OIDC)
- OAuth2
- SAML 2.0

CORE COMPONENTS

- Keycloak Server
- Keycloak Adapter
- Keycloak Operator

INTEGRATION

- LDAP
- Active Directory
- diverse relationale Datenbanken

MISC

- User self-registration
- Social login
- Multitenancy (Realms)
- Themes

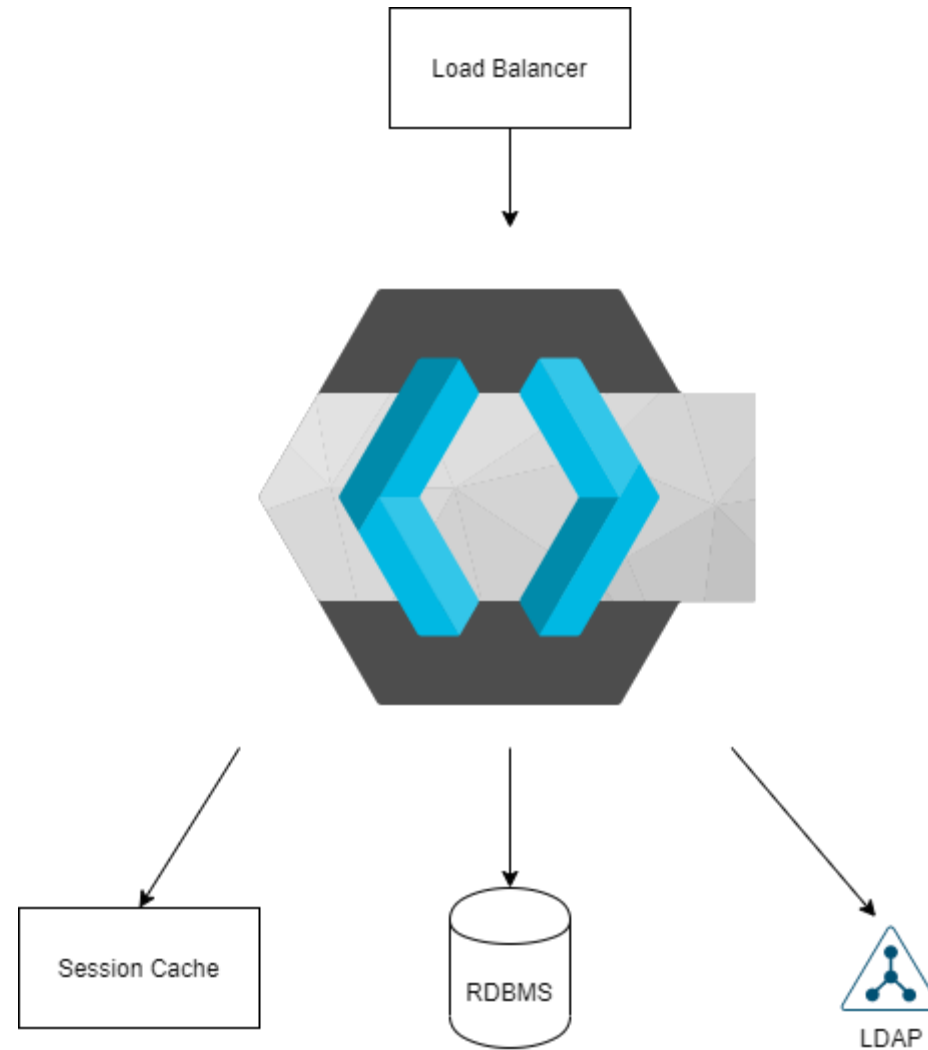
AUTHENTIFIZIERUNG

- Cookie
- Username / Password
- Kerberos
- 2FA (H/TOTP)
- WebAuthN
- Identity Brokering

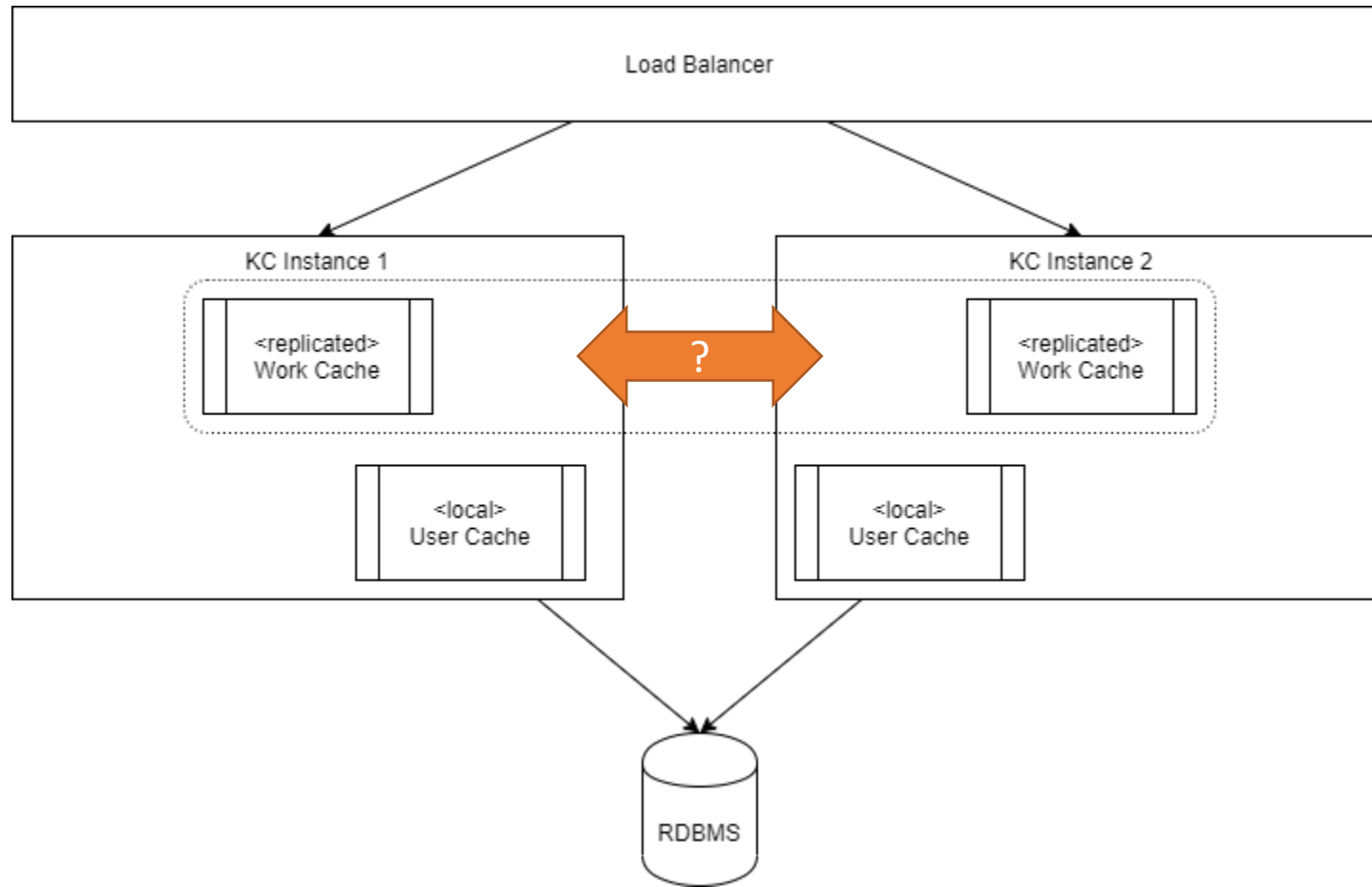


Cluster Basics

System Context



Simple Cluster



JGroups Initial Member Discovery

PING, TCPPING

FILE_PING

JDBC_PING

S3_PING, AWS_PING

GOOGLE_PING

DNS_PING

... and many others

JGroups Protocol Stack

Transport

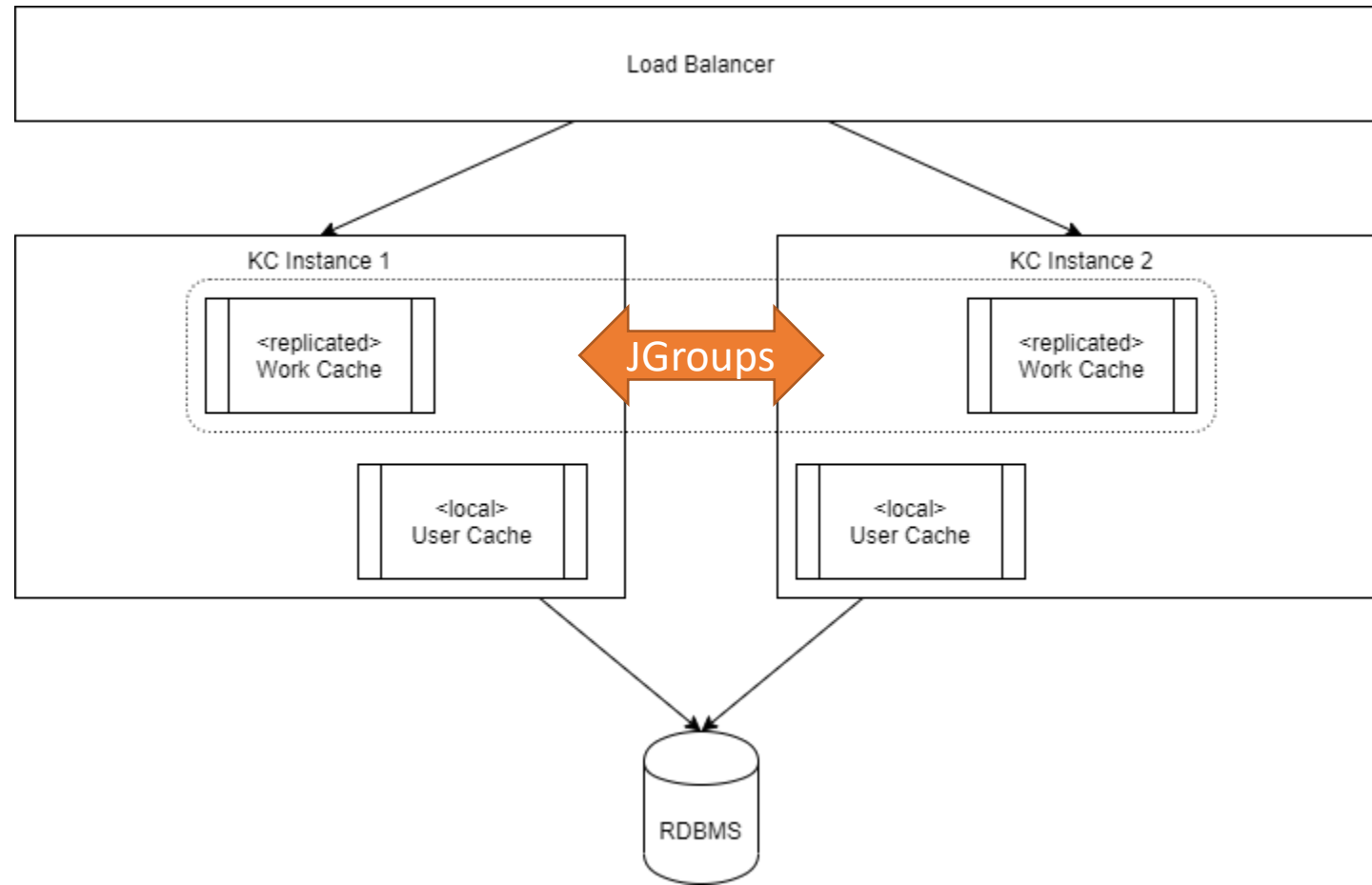
Failure Detection

Merging after partition

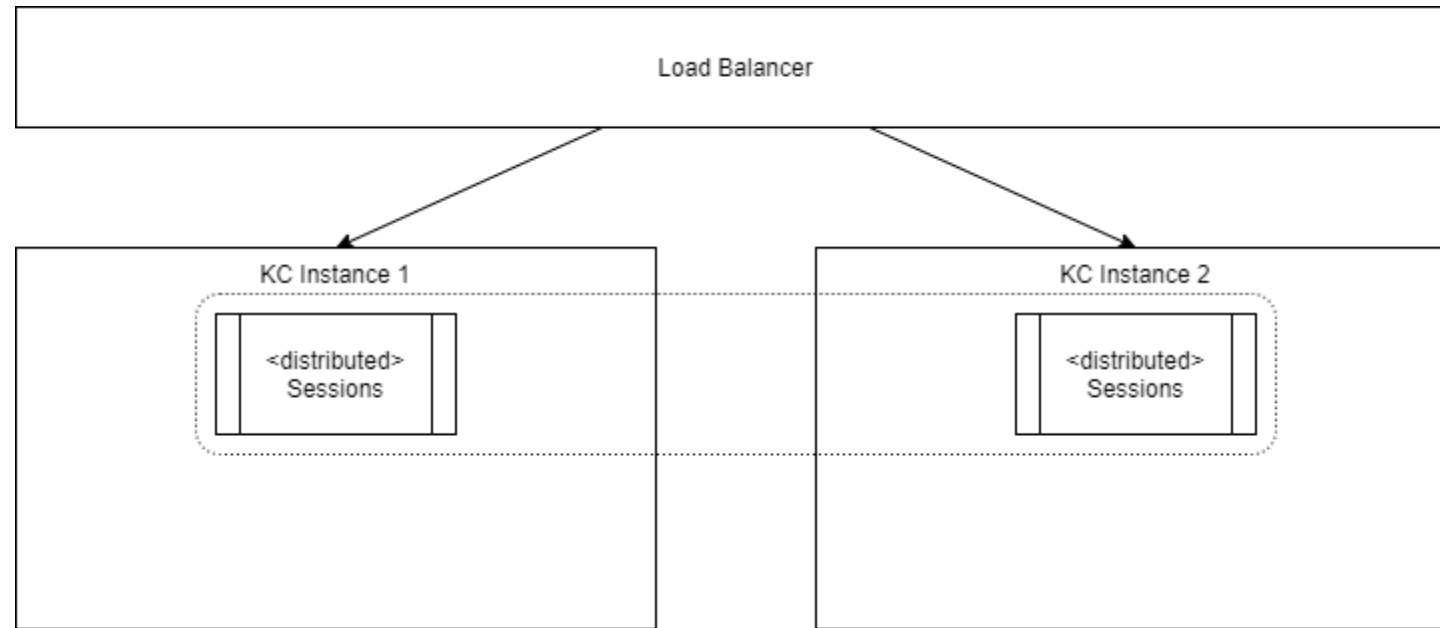
Group Membership

... and many others

Simple Cluster



Simple Cluster – Sessions



Sessions

Sessions



Resource
Owner



Client

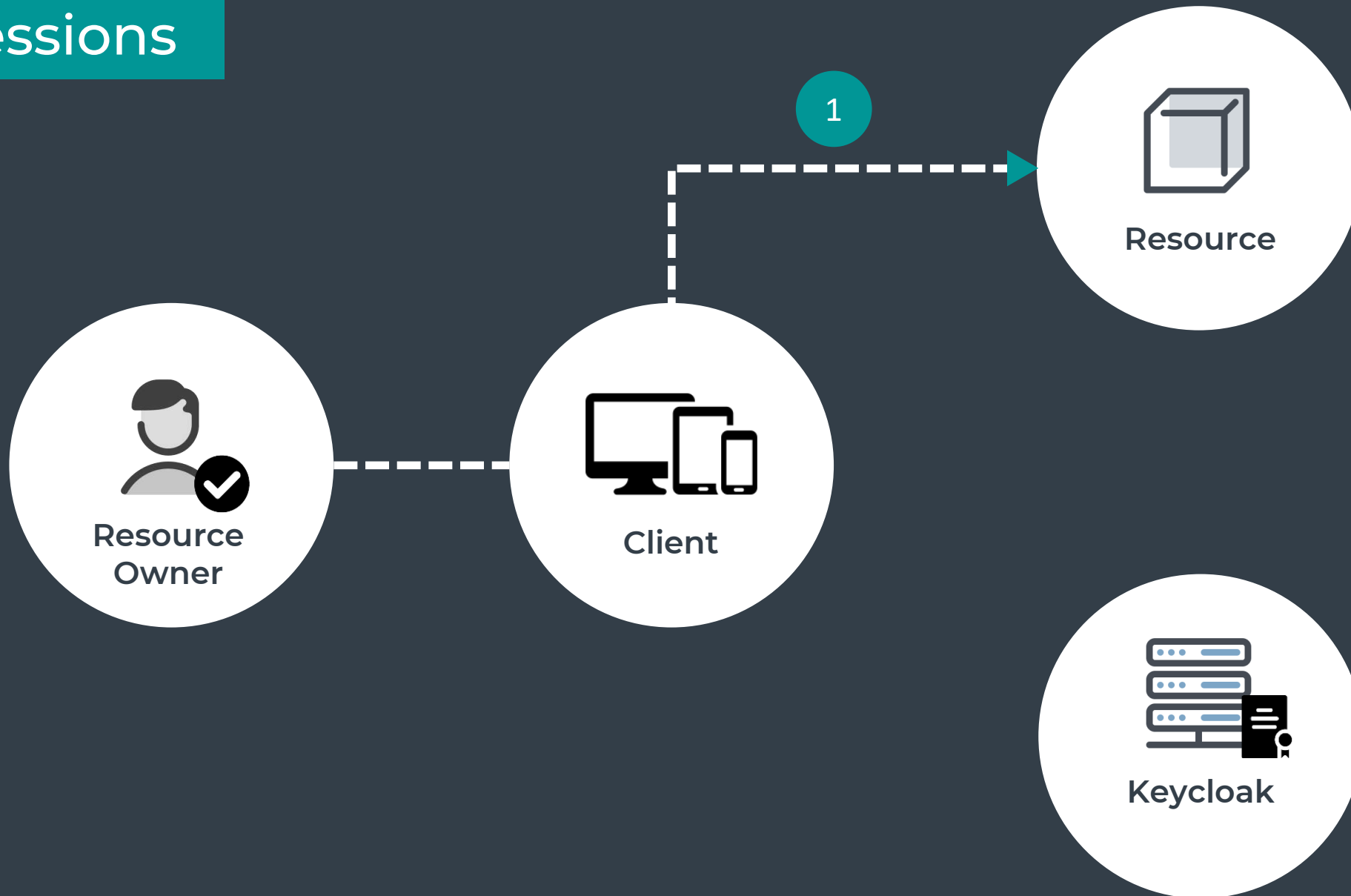


Resource

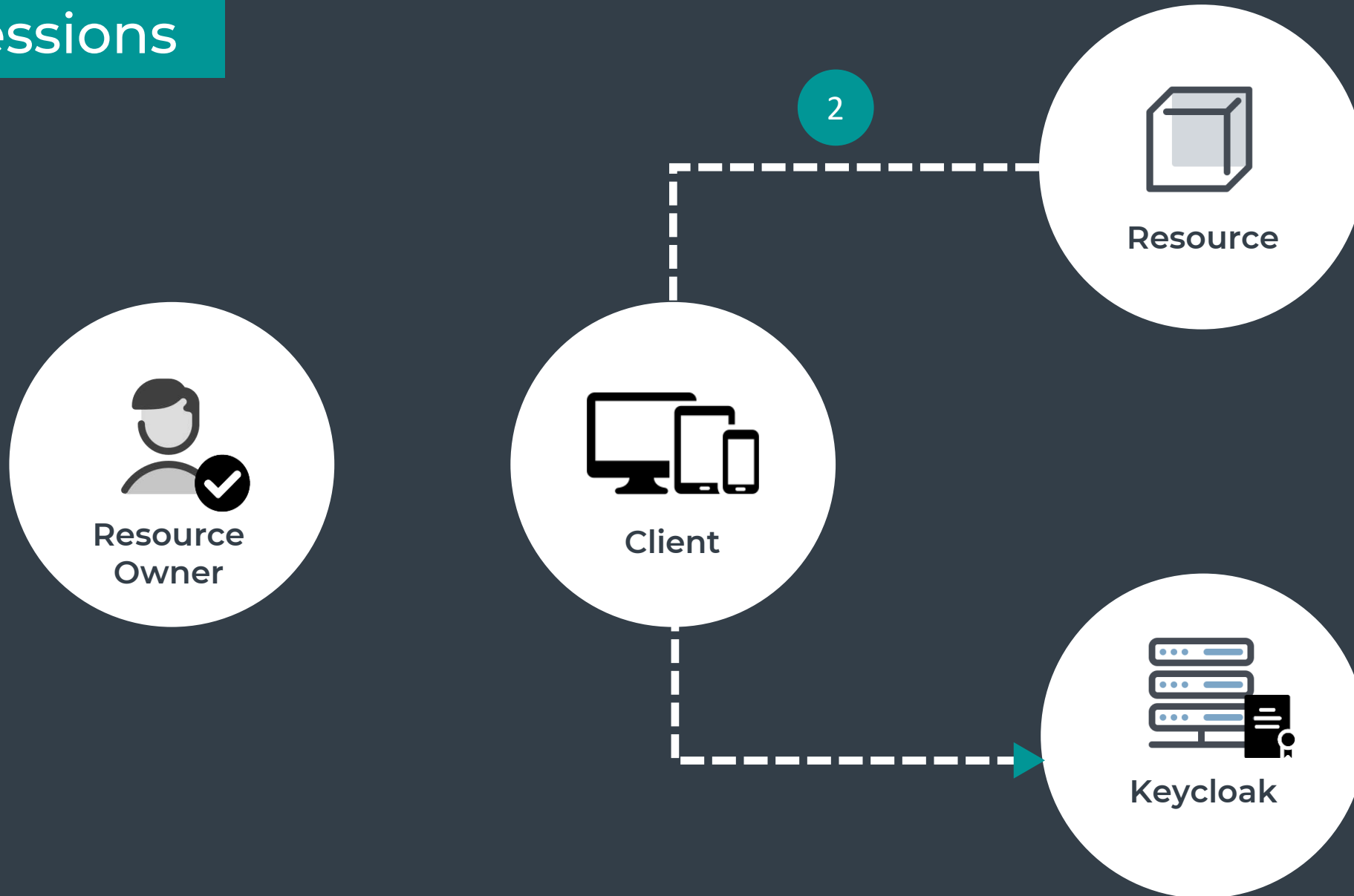


Keycloak

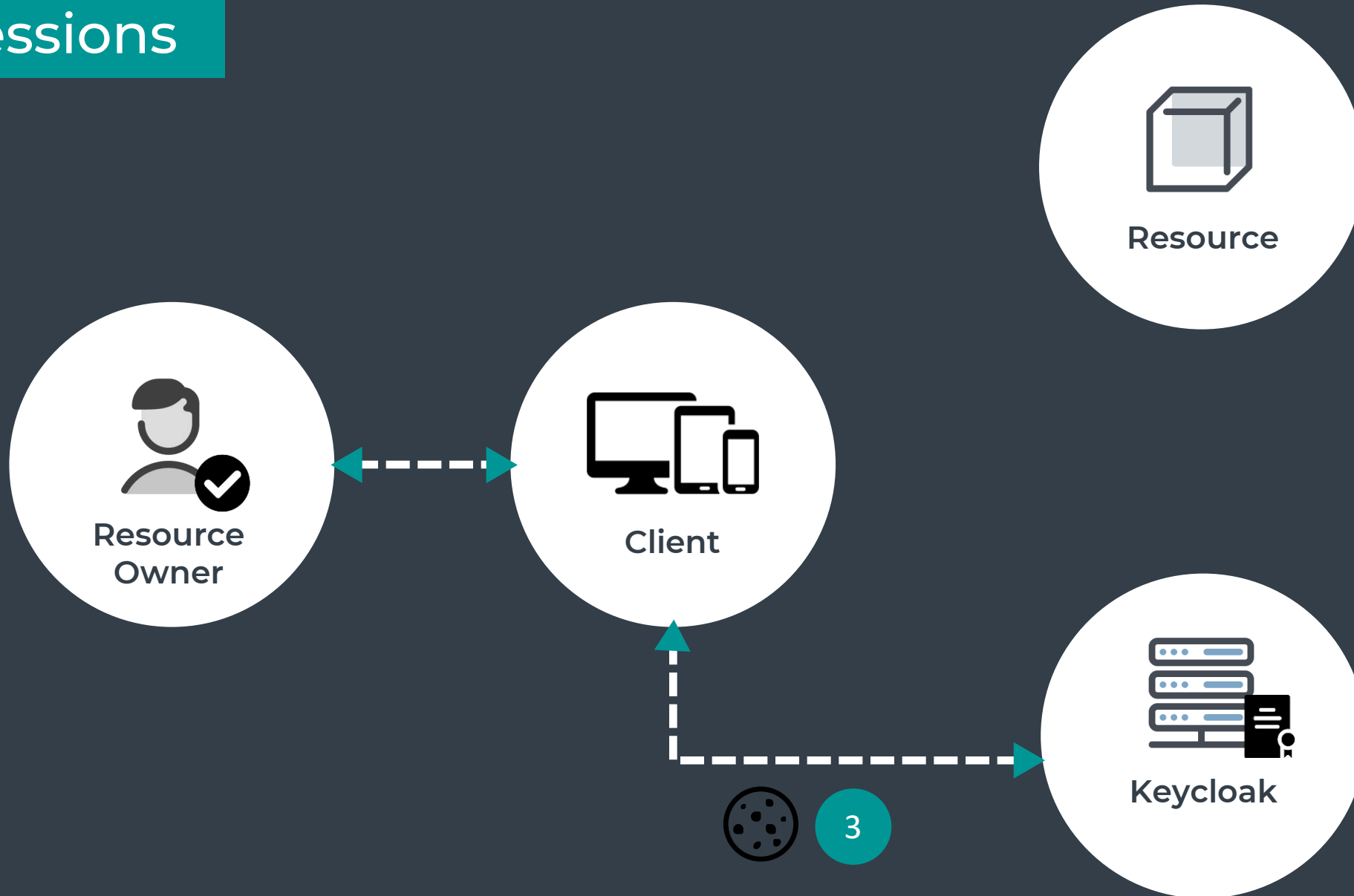
Sessions



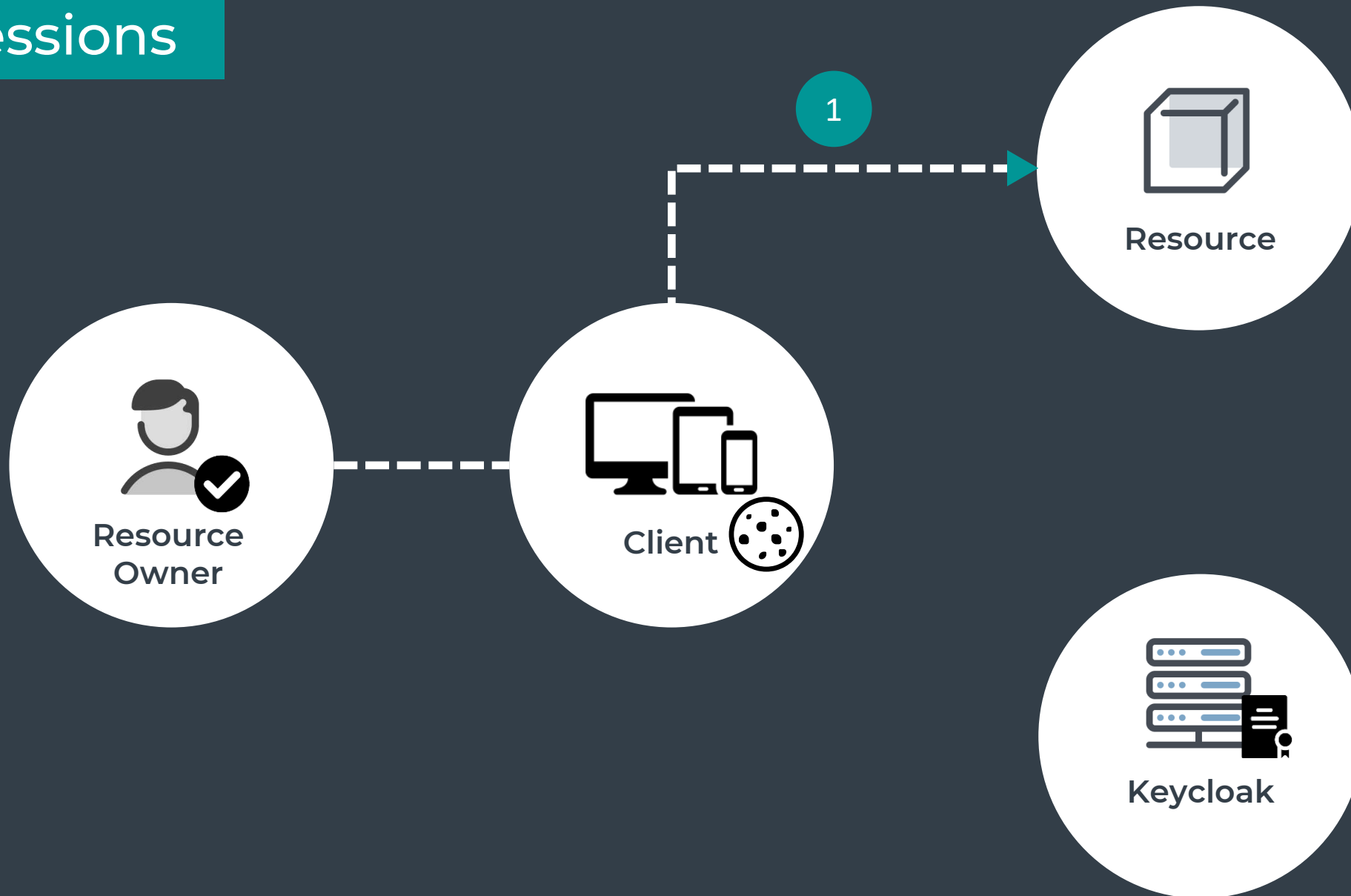
Sessions



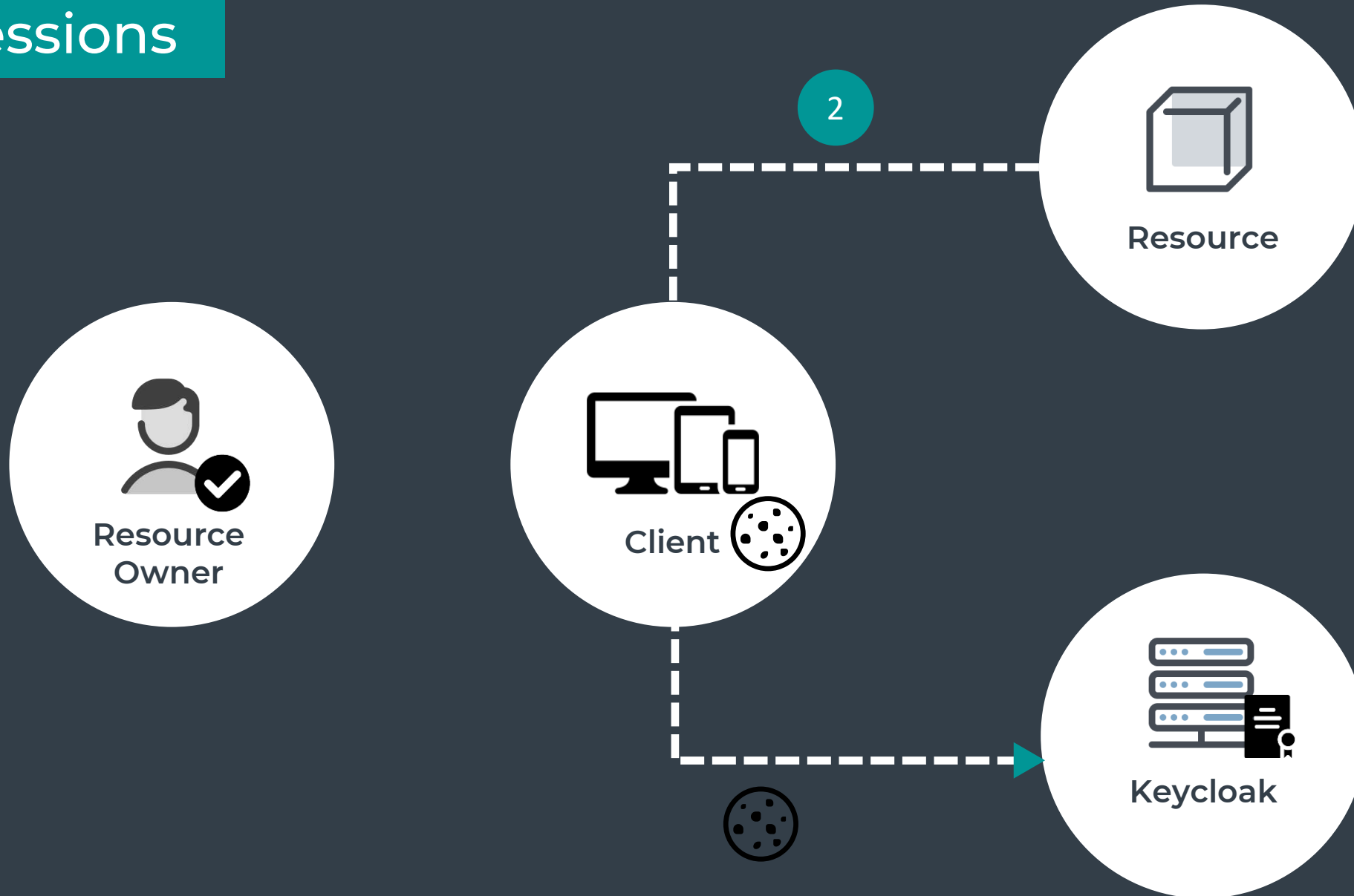
Sessions



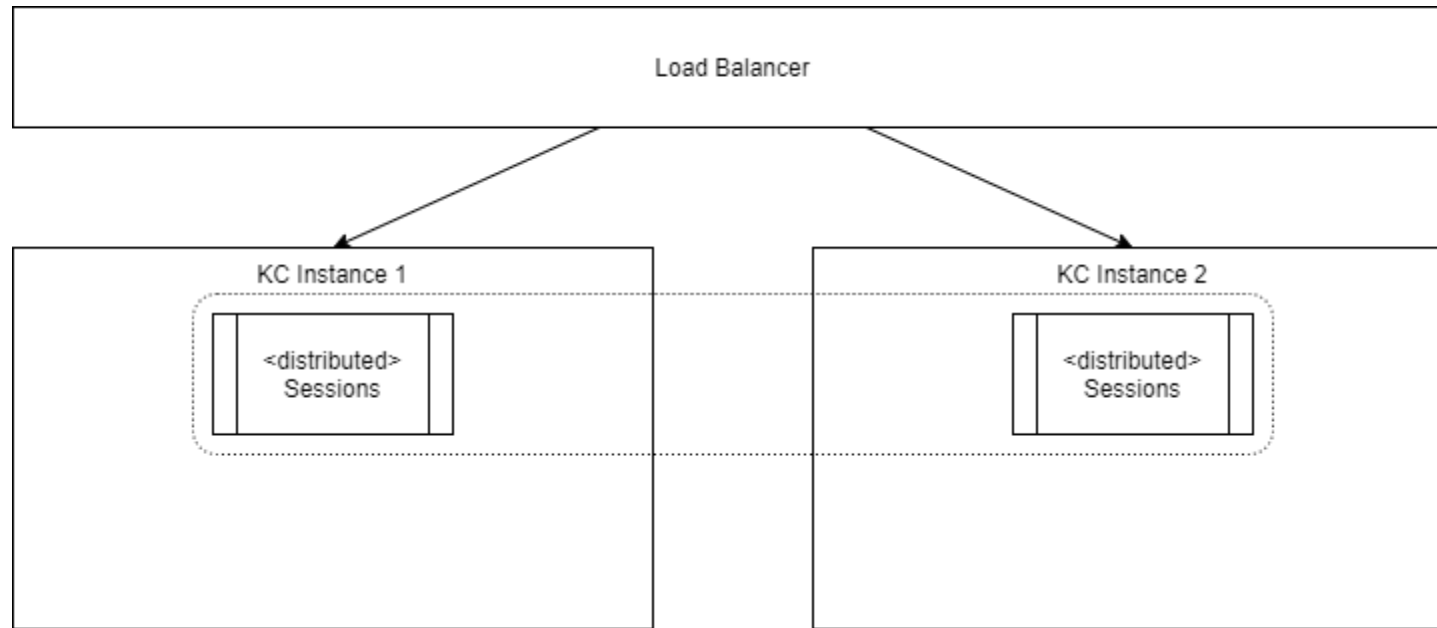
Sessions



Sessions



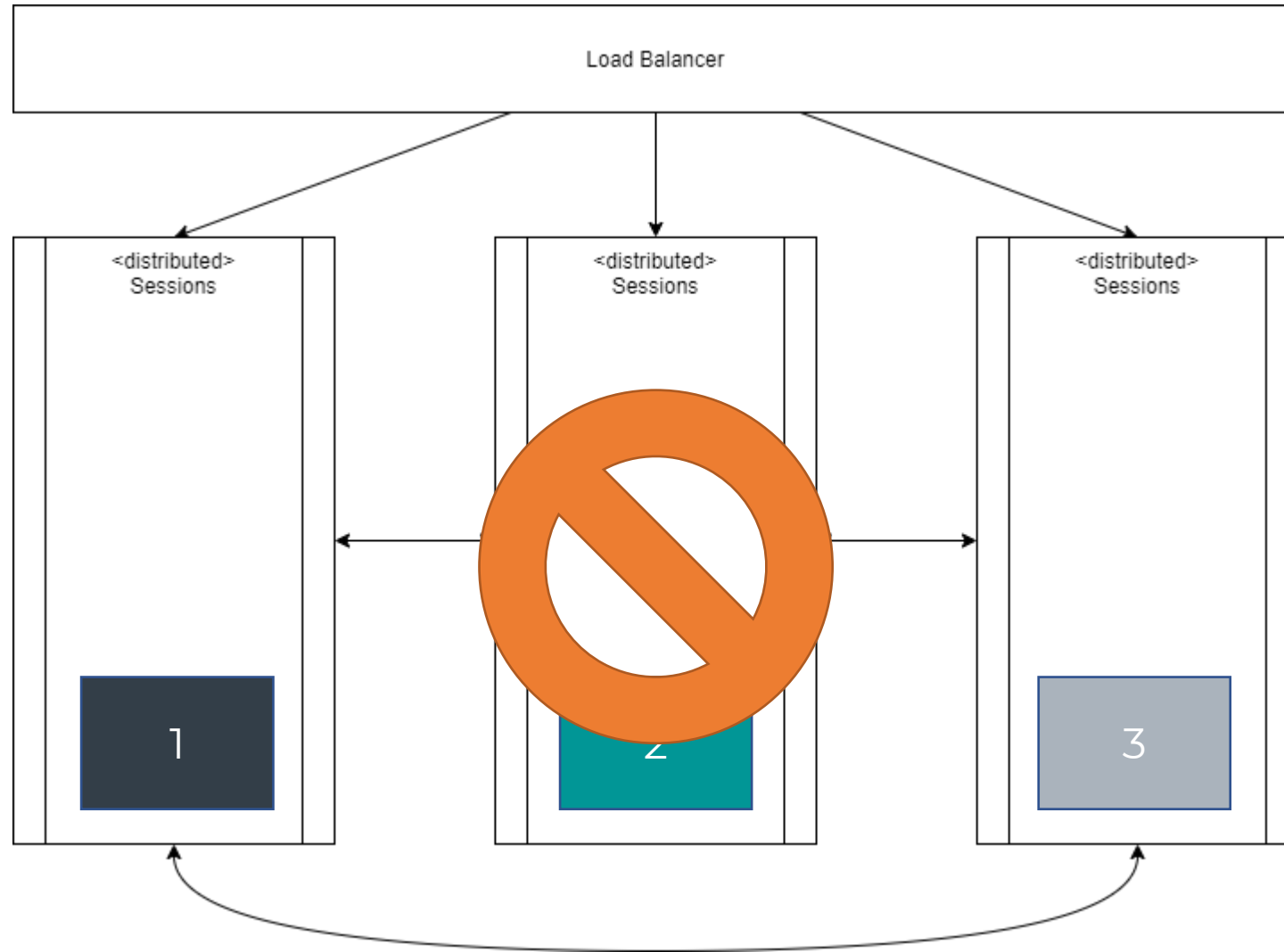
Simple Cluster – Sessions



Authentication Sessions
User Sessions
Client Sessions

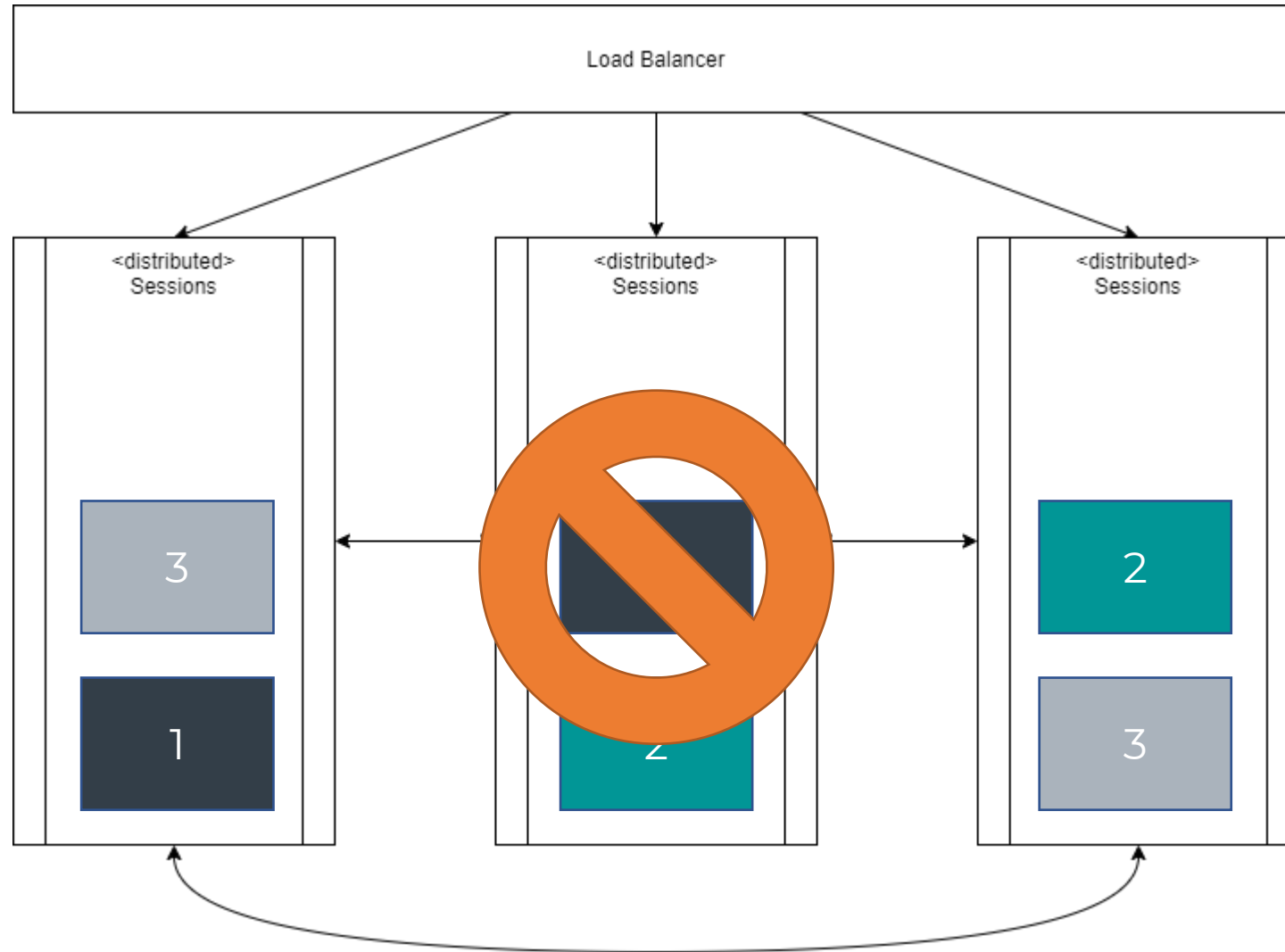
Distributed Caches

Distributed Caches



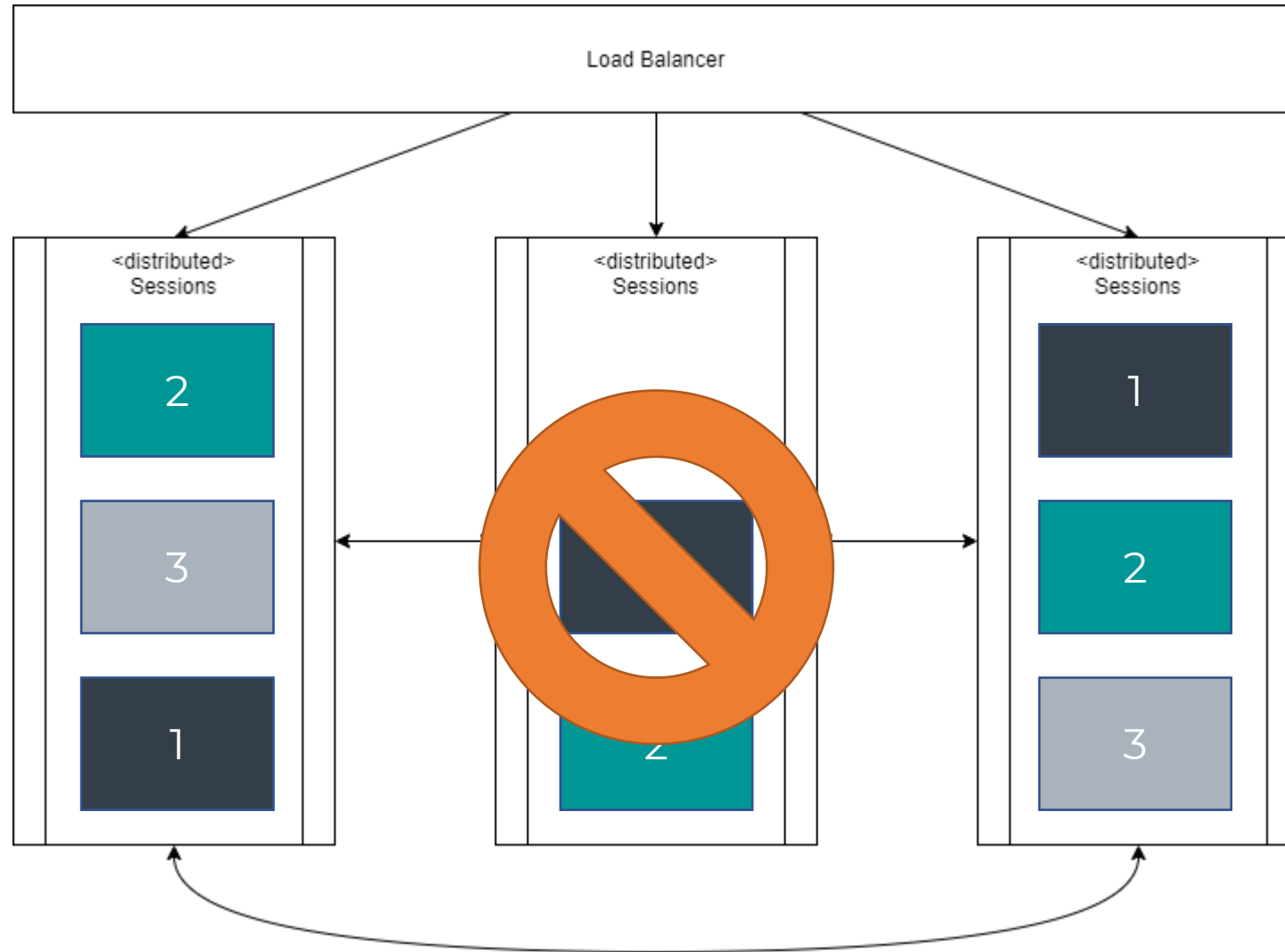
number of owners = 1

Distributed Caches



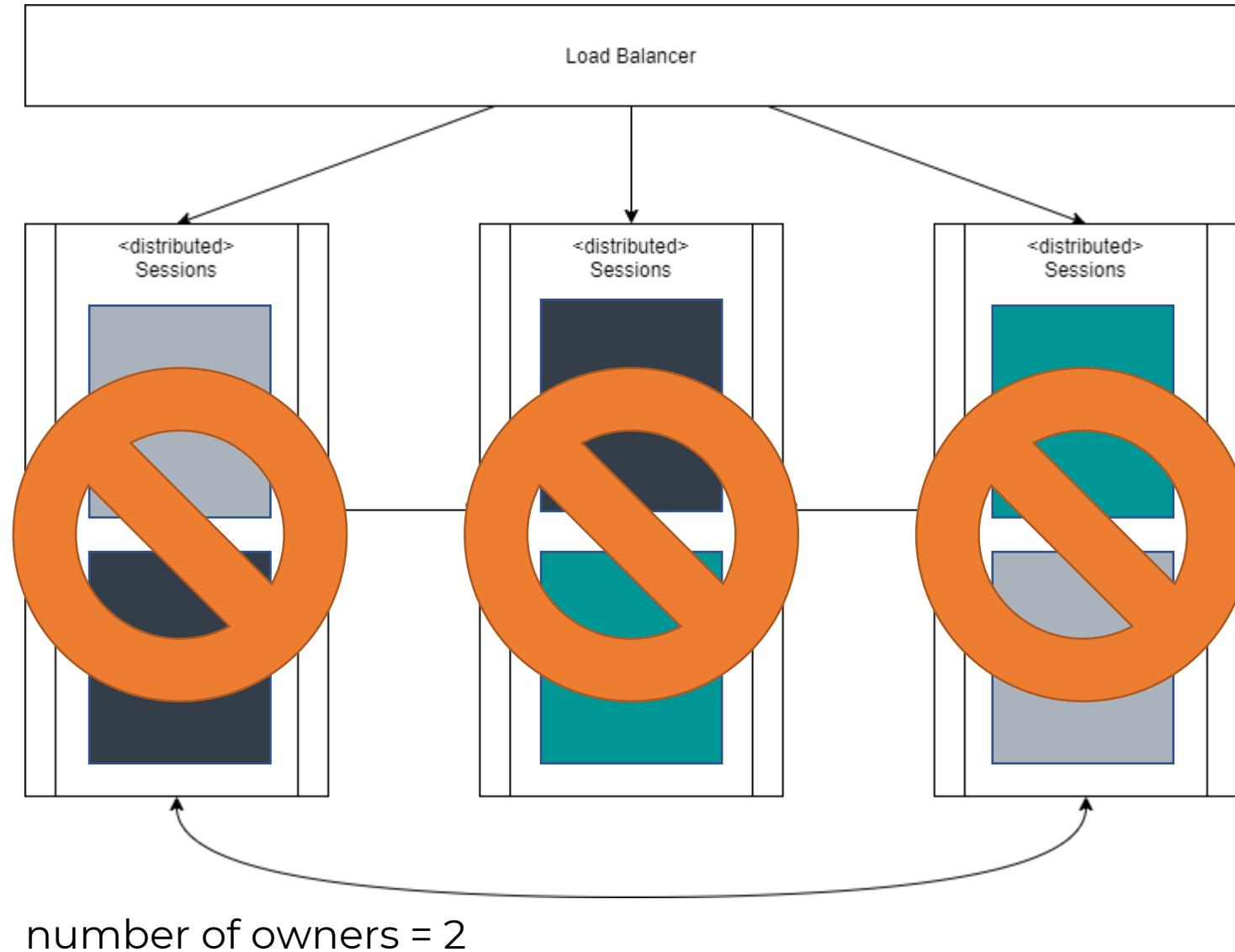
number of owners = 2

Distributed Caches



number of owners = 2

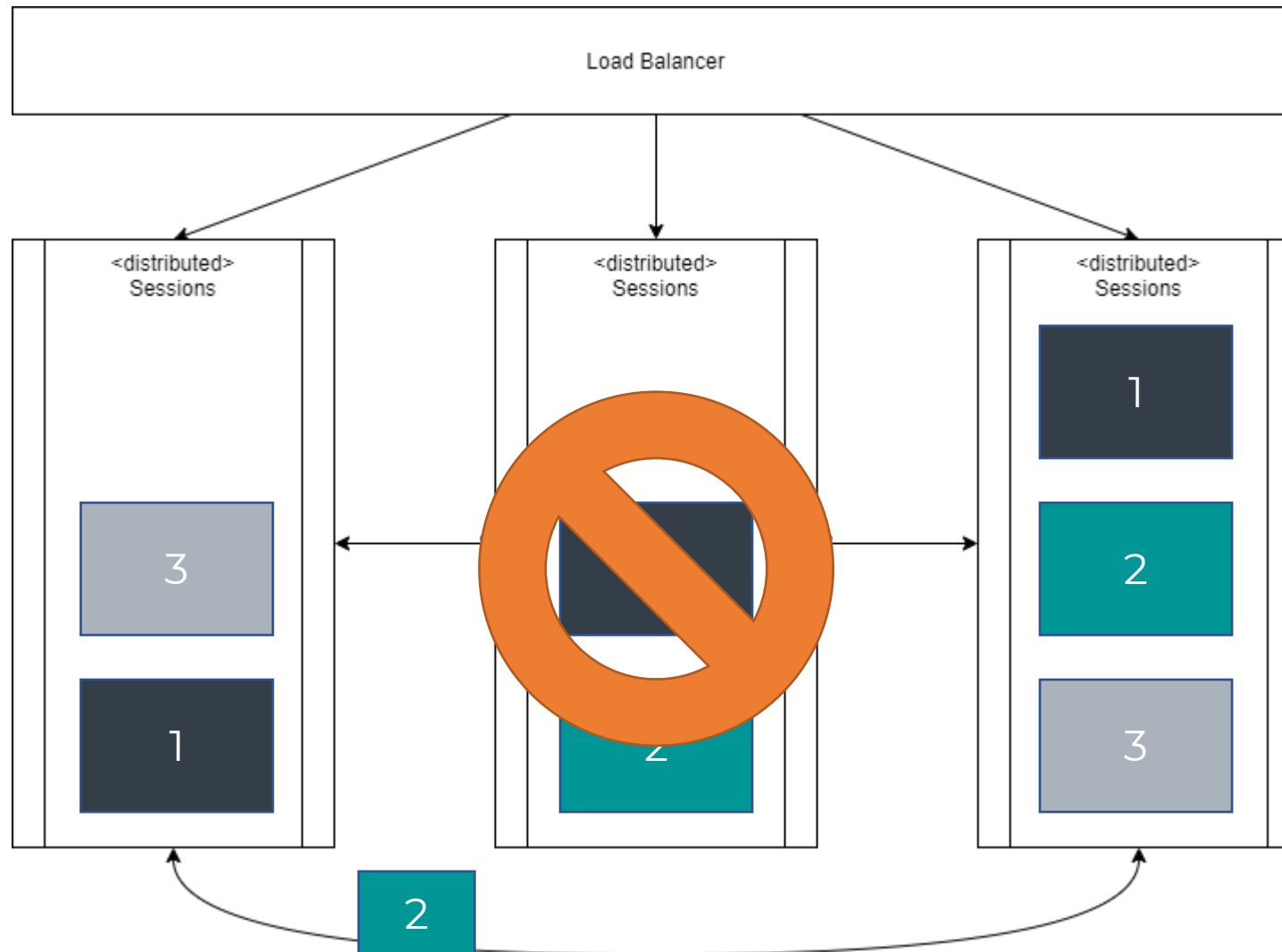
Distributed Caches – Memory Limits



Load Balancer Pitfalls

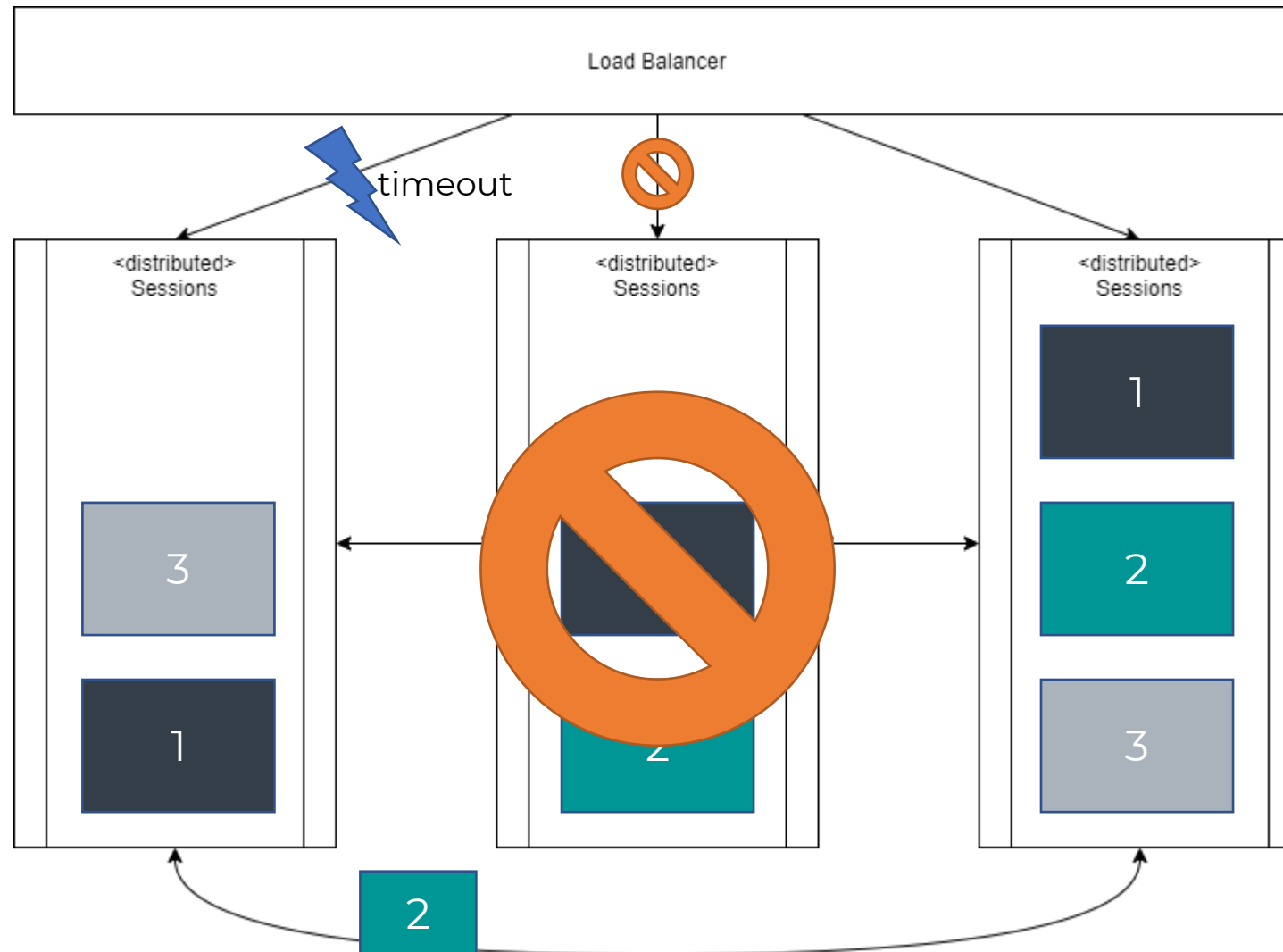
Load Balancers

Beware of passive health checks



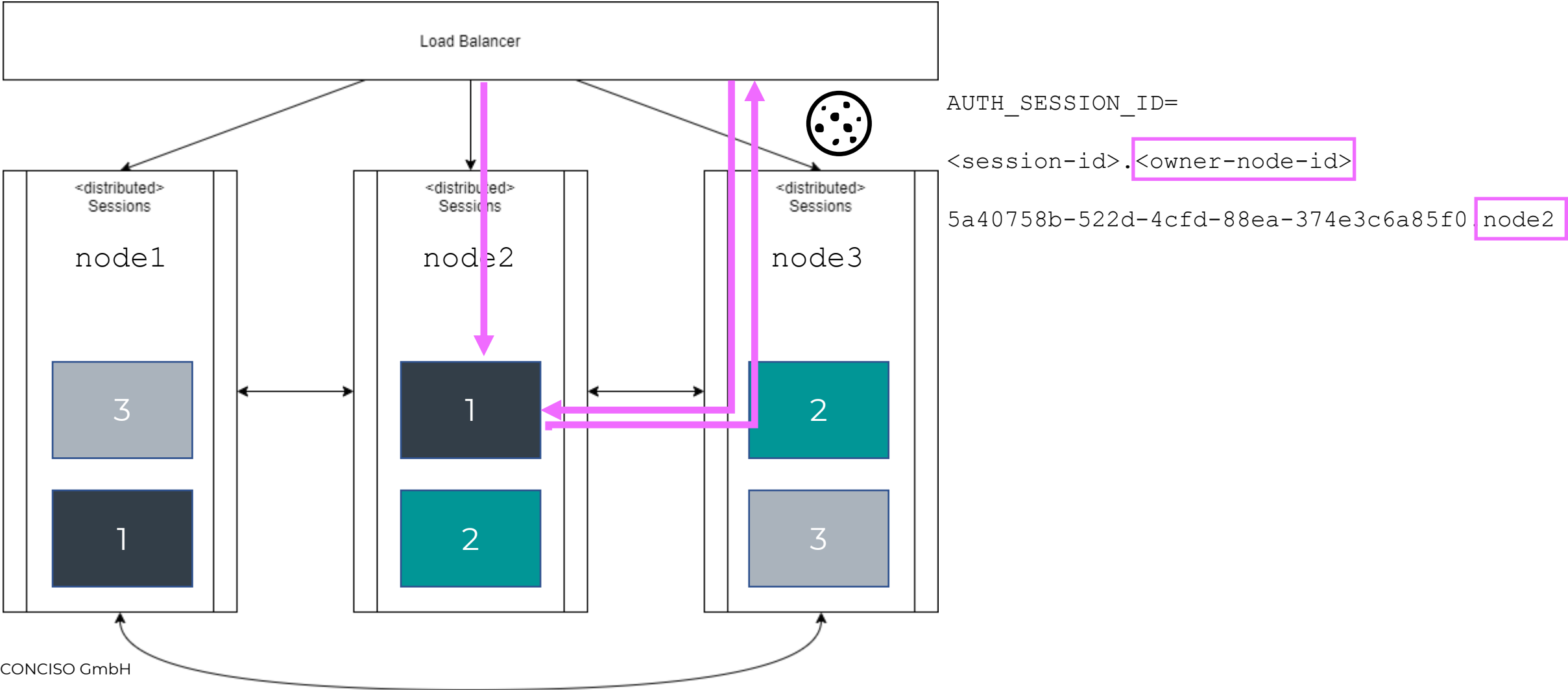
Load Balancers

Beware of passive health checks



Load Balancers

Understand session stickiness!



Load Balancers

Apache mod_proxy_balancer

```
<Proxy "balancer://keycloak-cluser">
    BalancerMember "http://198.51.100.50:80" route=node1
    BalancerMember "http://198.51.100.51:80" route=node2
    BalancerMember "http://198.51.100.52:80" route=node3
    ProxySet stickysession=AUTH_SESSION_ID
</Proxy>
ProxyPass          "/auth" "balancer://keycloak-cluser"
ProxyPassReverse   "/auth" "balancer://keycloak-cluser"
```

Session Revocation

Token/Session revocation

RFC 7009 – OAuth 2.0 Token Revocation:

„Depending on the authorization server's revocation policy, the revocation of a particular token may cause the revocation of related tokens and the underlying authorization grant.”

```
// return a stream that 'wraps' the infinispán cache stream so that the cache stream's elements are read one by one
// and then filtered/mapped locally to avoid serialization issues when trying to manipulate the cache stream directly.
return StreamSupport.stream(cache.entrySet().stream().spliterator(), true)
    .filter(predicate)
    .map(Mappers.userSessionEntity())
    .map(entity -> this.wrap(realm, entity, offline));
```

java.util.Spliterators\$ArraySpliterator @ 0x6dff7c8a8 Unreachable	0.00 MB	46.15 MB	4.40%
java.util.Spliterators\$ArraySpliterator @ 0x6e4a06370 Unreachable	0.00 MB	43.18 MB	4.11%
java.util.Spliterators\$ArraySpliterator @ 0x76fa630a0 Unreachable	0.00 MB	40.32 MB	3.84%
java.util.Spliterators\$ArraySpliterator @ 0x76d48bf88 Unreachable	0.00 MB	37.34 MB	3.56%
java.util.Spliterators\$ArraySpliterator @ 0x76a8b02f0 Unreachable	0.00 MB	34.14 MB	3.25%
java.lang.Object[23552] @ 0x76a8b0310	0.09 MB	34.14 MB	3.25%
org.infinispan.container.entries.TransientMortalCacheEntry @ 0x76bda2da8	0.00 MB	0.00 MB	0.00%
org.infinispan.container.entries.TransientMortalCacheEntry @ 0x76c7bbb50	0.00 MB	0.00 MB	0.00%
org.infinispan.container.entries.TransientMortalCacheEntry @ 0x76baa2ca0	0.00 MB	0.00 MB	0.00%
org.infinispan.container.entries.TransientMortalCacheEntry @ 0x76b145ac0	0.00 MB	0.00 MB	0.00%
org.infinispan.container.entries.TransientMortalCacheEntry @ 0x76baefa38	0.00 MB	0.00 MB	0.00%
org.infinispan.container.entries.TransientMortalCacheEntry @ 0x76aab5420	0.00 MB	0.00 MB	0.00%
org.infinispan.container.entries.TransientMortalCacheEntry @ 0x76c8ed218	0.00 MB	0.00 MB	0.00%
org.infinispan.container.entries.TransientMortalCacheEntry @ 0x76b1e9120	0.00 MB	0.00 MB	0.00%
org.infinispan.container.entries.TransientMortalCacheEntry @ 0x76ba7be90	0.00 MB	0.00 MB	0.00%

→ KEYCLOAK-18981 Keycloak cluster fails occasionally due to high memory usage

Transient Sessions

Keycloak supports transient sessions for Client Credential Grant

✓ OpenID Connect Compatibility Modes

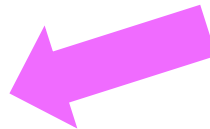
Exclude Session State From
Authentication Response 

☐ OFF

Use Refresh Tokens 

☒ ON

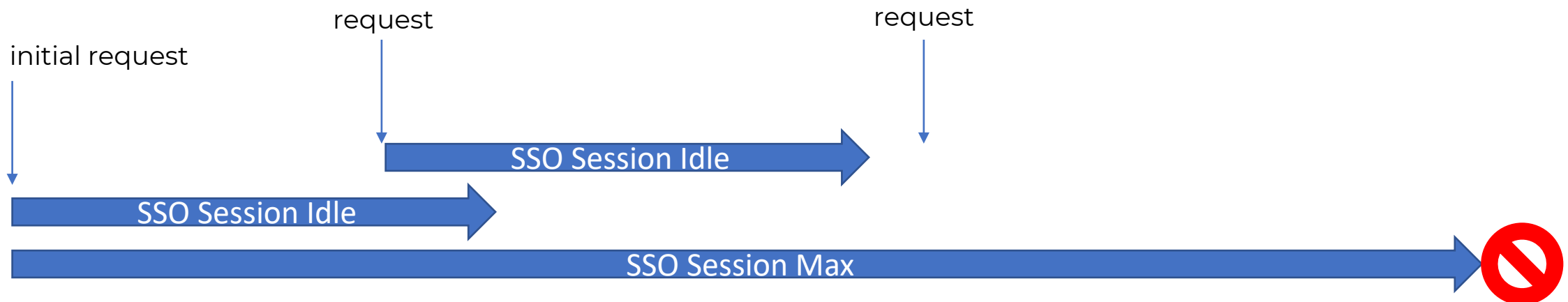
Use Refresh Tokens For Client
Credentials Grant 

☐ OFF

Session Retention Time

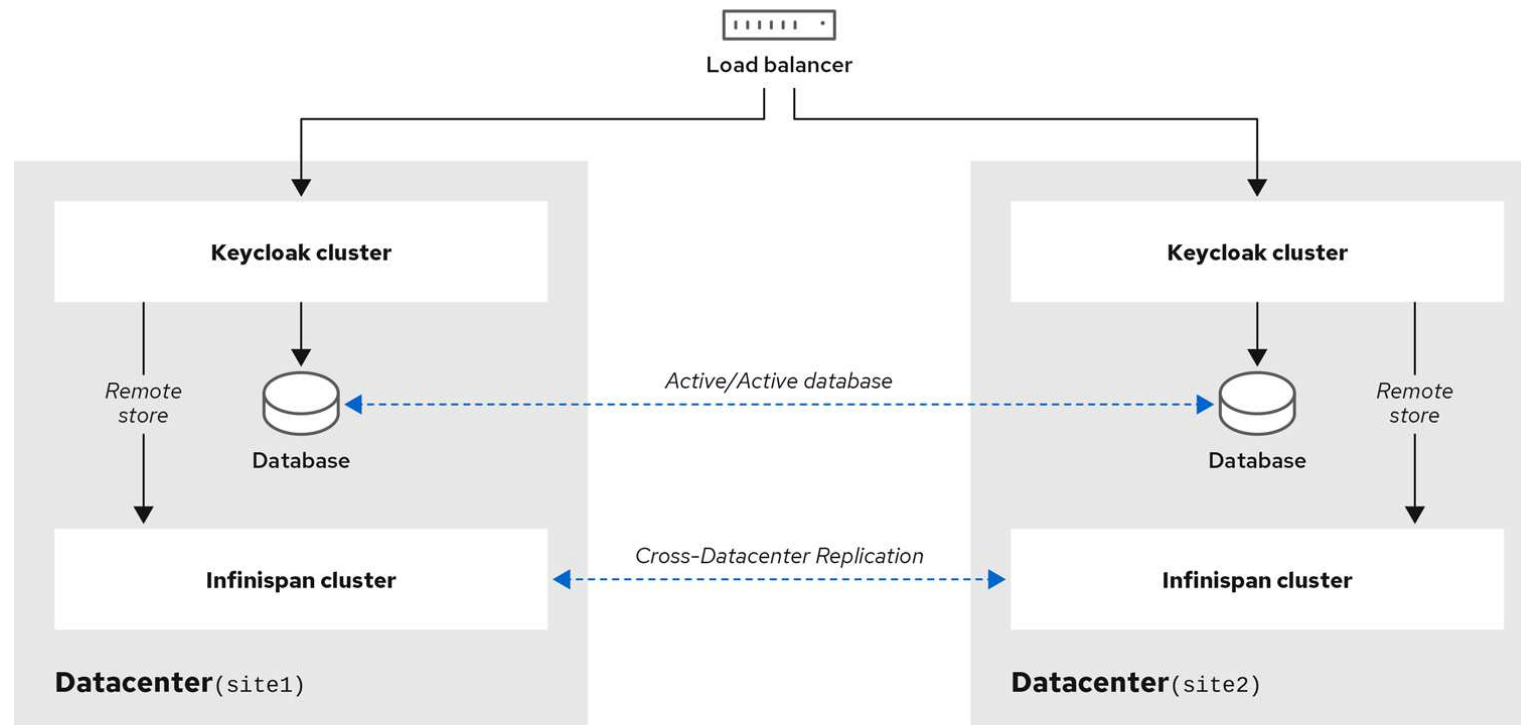
Keycloak relies on Infinispan cache entry expiration

```
public void removeAllExpired() {  
    // Rely on expiration of cache entries provided by infinispan. Just expire entries from persister is needed  
    // TODO: Avoid iteration over all realms here (Details in the KEYCLOAK-16802)  
    session.realms().getRealmsStream().forEach(this::removeExpired);  
}  
  
@Override  
public void removeExpired(RealmModel realm) {  
    // Rely on expiration of cache entries provided by infinispan. Nothing needed here besides calling persister  
    session.getProvider(UserSessionPersisterProvider.class).removeExpired(realm);  
}
```



Cross-DC Replication

Cross Data Center



Source: Keycloak Server Installation Documentation (<https://keycloak.org>)

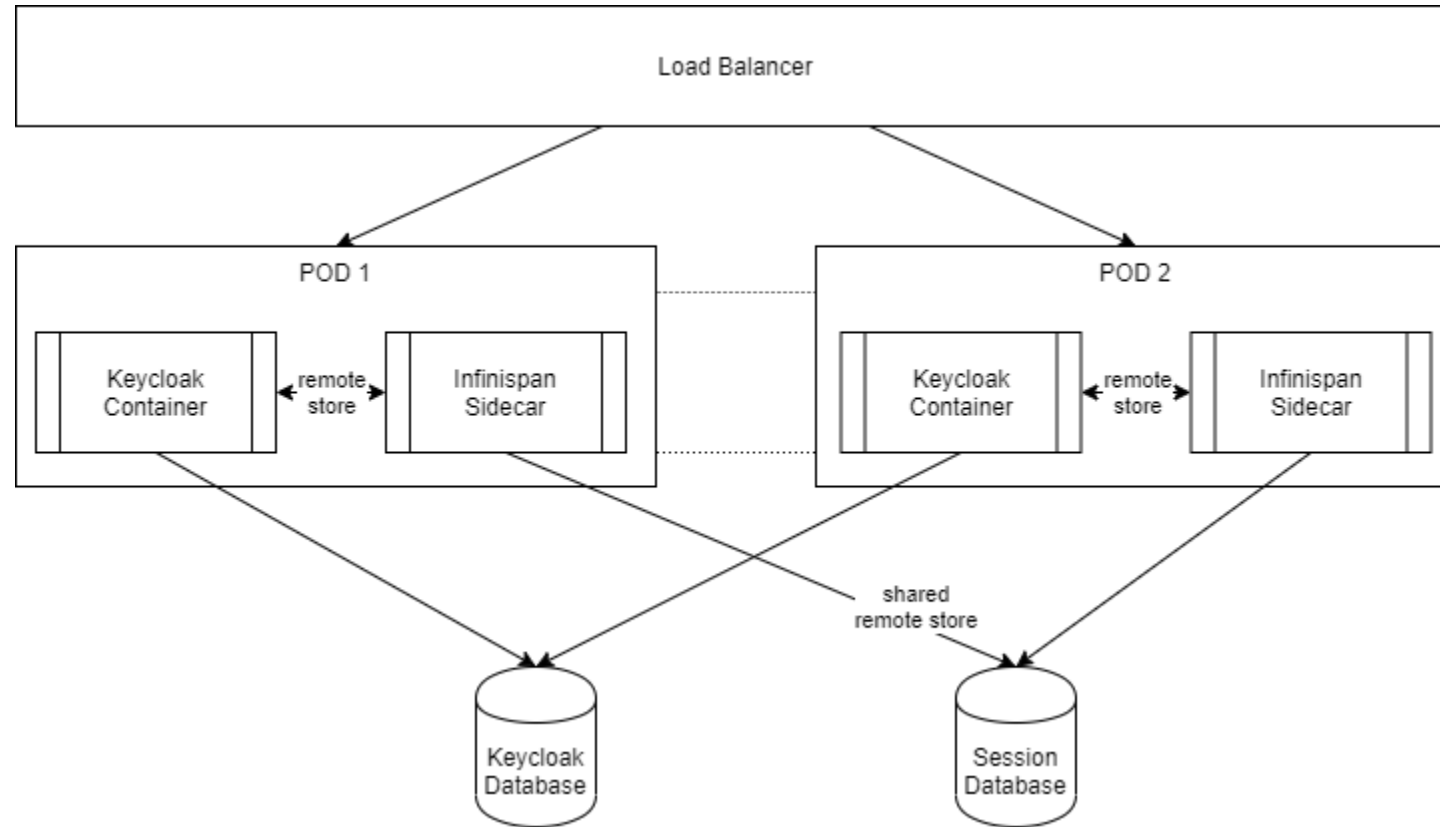
70_RHSSO_0420

Cross Data Center - Replication

Replication recommendation:

Cache	Active/Active	Active/Passive
work	SYNC	SYNC
actionTokens	SYNC → better security ASYNC → better performance	ASYNC
loginFailures	NO BACKUP	NO_BACKUP or ASYNC
sessions/clientSessions	SYNC	ASYNC
offline(Client)Sessions	SYNC	ASYNC
authenticationSessions	NO BACKUP → use sticky sessions	NO BACKUP → re-login ASYNC

External Session Storage with Infinispan Sidecar (experimental)



Updates & Deployments

Keycloak Updates

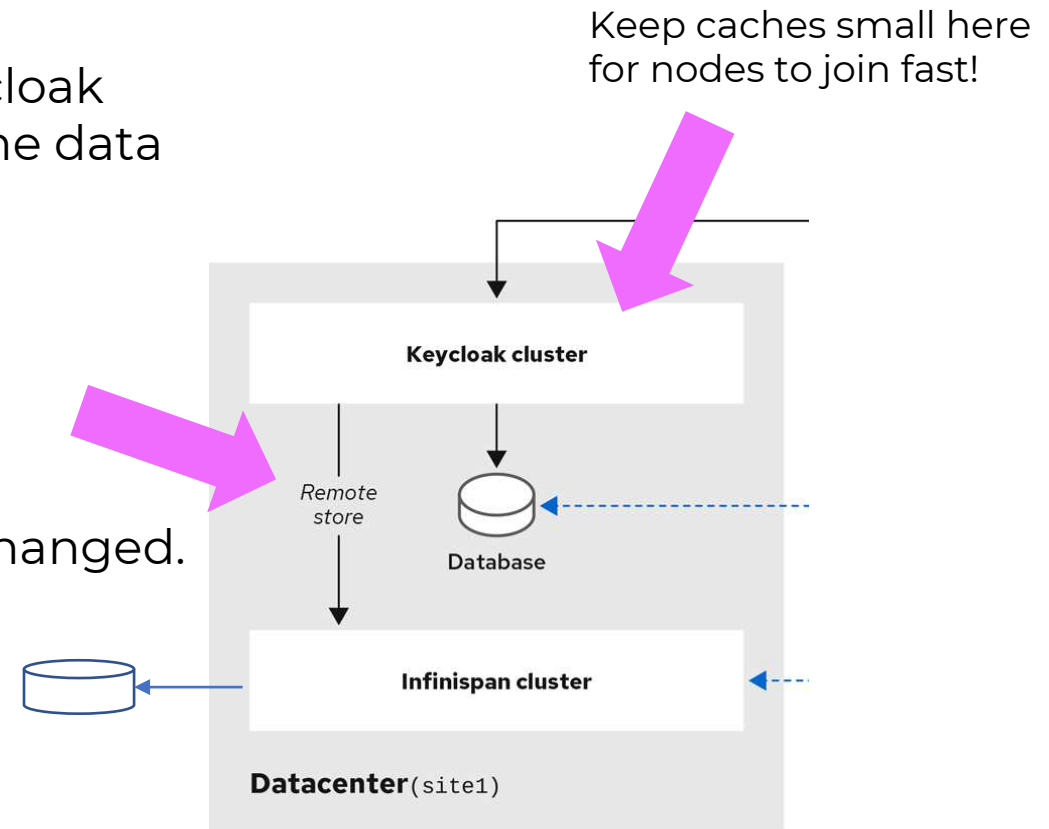
Keycloak does not support 0-downtime deployments!

Practitioner hints:

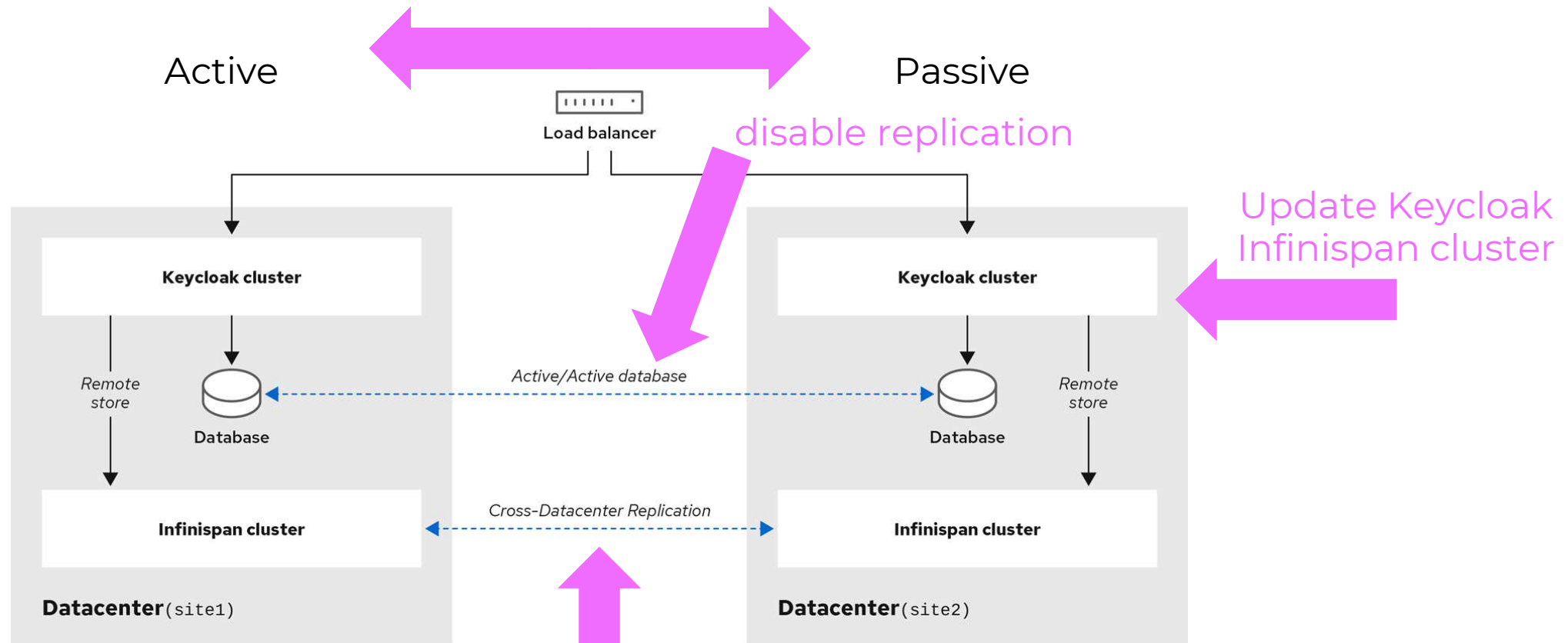
External Infinispan clusters allow to update Keycloak without losing session information as long as the data format/schema has not changed

Allows to update Infinispan Cluster if sessions are stored to external storage.

Supports rolling updates if database schema is compatible and infinispan format/schema not changed.



Kecloak Updates (Active/Passive - Blue/Green)



Source: Keycloak Server Installation Documentation (<https://keycloak.org>)

70_RHSSO_0420

switch failure-policy
from FAIL to WARN/IGNORE

Outlook Keycloak.X

Keycloak.X WILL support 0-downtime deployments!

- Quarkus-based instead of Wildfly
- Faster startup times, lower resource footprint
- New storage system, focus on scalability and availability
- Observability – Metrics, logging, health-checks
- GitOps-friendly configuration

Roadmap:

- Keycloak 16 last preview for Keycloak.X
- End of 2021: Keycloak 17 will deprecate Wildfly distribution
- March 2022: Preview of new storage system
- June 2022: First release with Quarkus-only support

<https://www.keycloak.org/2021/10/keycloak-x-update>

THANK

YOU.