STORAGE DEVELOPER CONFERENCE

SDC 23

BY Developers FOR Developers

Re-thinking security in a distributed storage system

Apache Ozone Security

Istvan Fajth - Apache Ozone PMC - Cloudera Inc.

A brief project overview

Security in Apache Ozone

Tokens

Public Key Infrastructure

Shopee

Apache Ozone

2003

Papers:

- GFS
- Mapreduce

2006

2009

Sub-projects:

- Common
- HDFS
- Mapreduce
- YARN

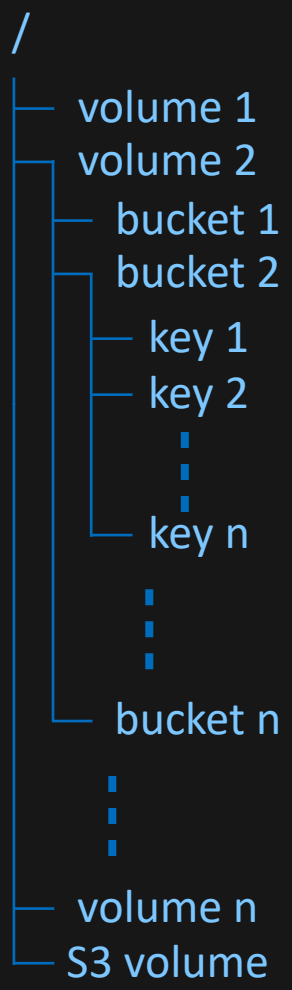
2014

HDFS-7240

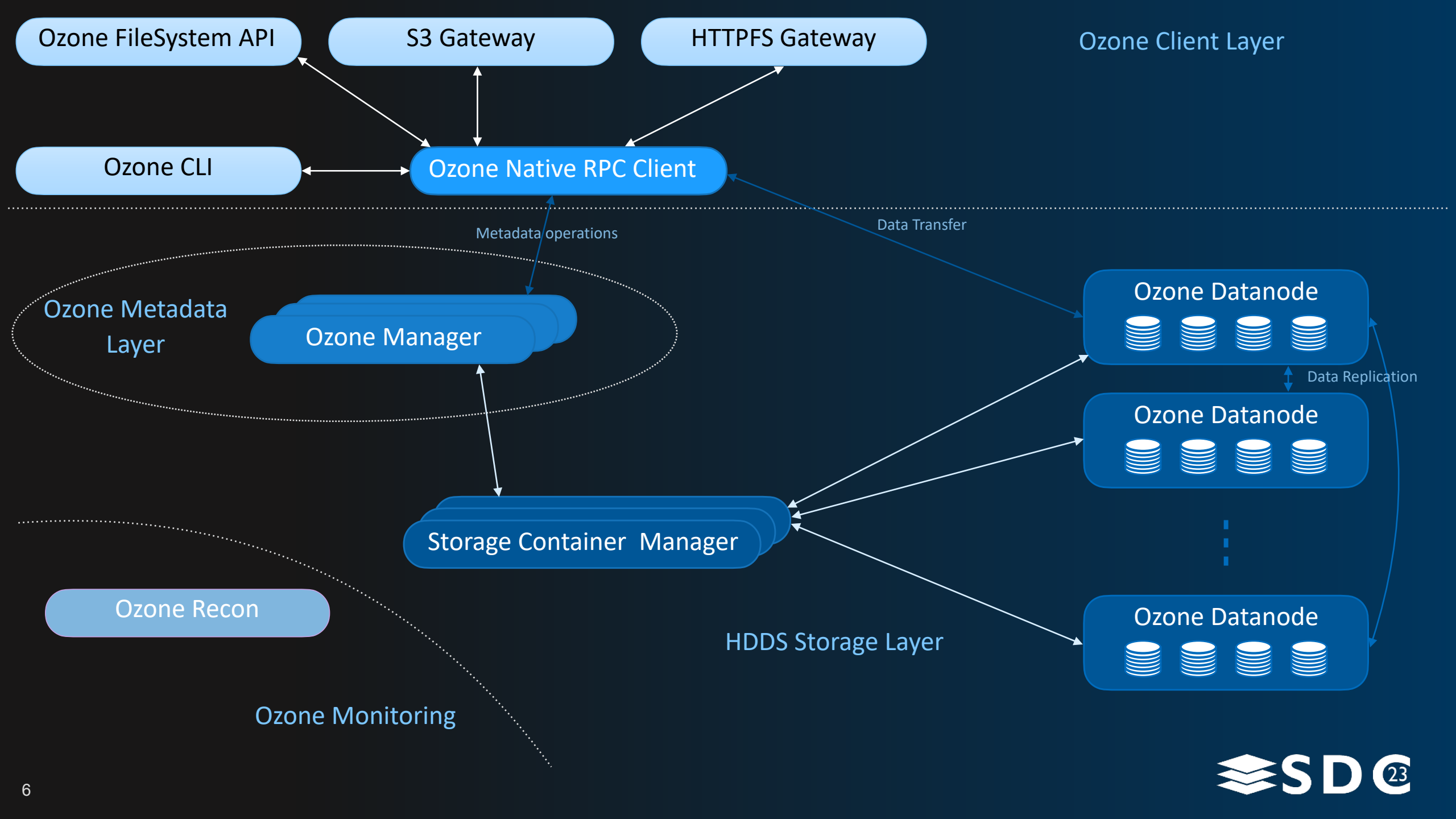
Scaling HDFS

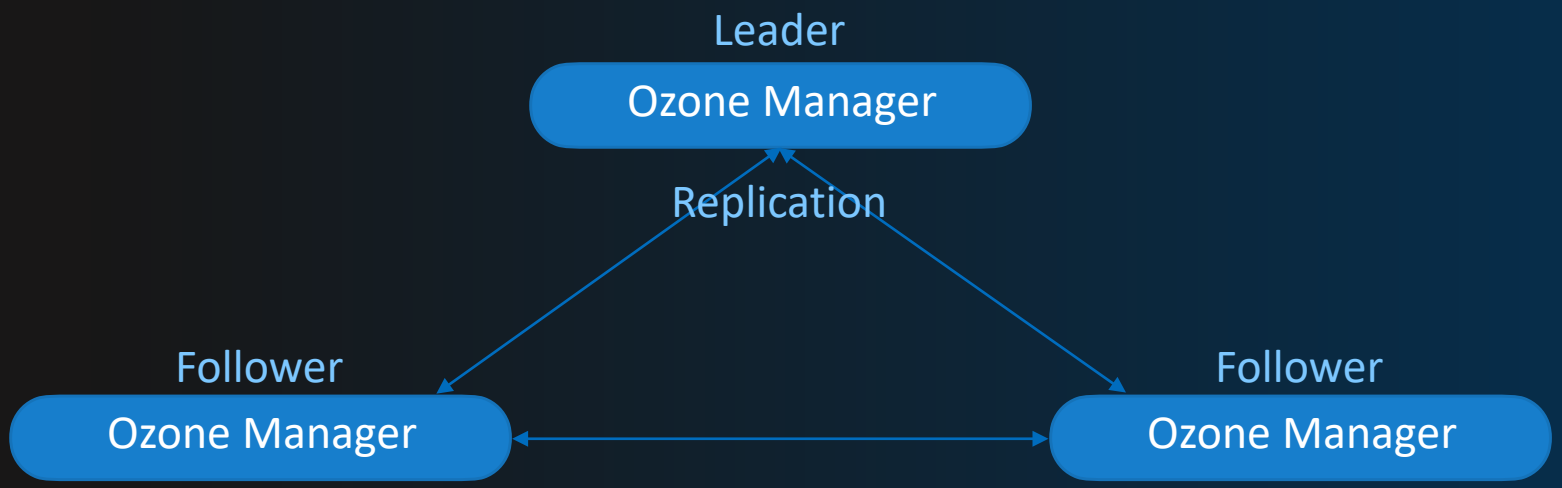
2018

Apache Ozone is a highly scalable, distributed storage for Analytics, Big data and Cloud Native applications. Ozone supports S3 compatible object APIs as well as a Hadoop Compatible File System implementation. It is optimized for both efficient object store and file system operations.



	Object Store	FileSystem
Volume	A bucket group	Top level directory
Bucket	A bucket	Directory in a top level directory
Key	A key	Directory or file in a bucket or directory


Ozone FileSystem API
S3 Gateway
HTTPFS Gateway
Ozone Client Layer
Ozone CLI
Ozone Native RPC Client
Metadata operations
Data Transfer
Ozone Metadata Layer
Ozone Manager
Ozone Datanode
Data Replication
Ozone Datanode
Storage Container Manager
Ozone Recon
Ozone Datanode
HDDS Storage Layer
Ozone Monitoring


Leader
Ozone Manager
Replication
Follower
Ozone Manager
Follower
Ozone Manager

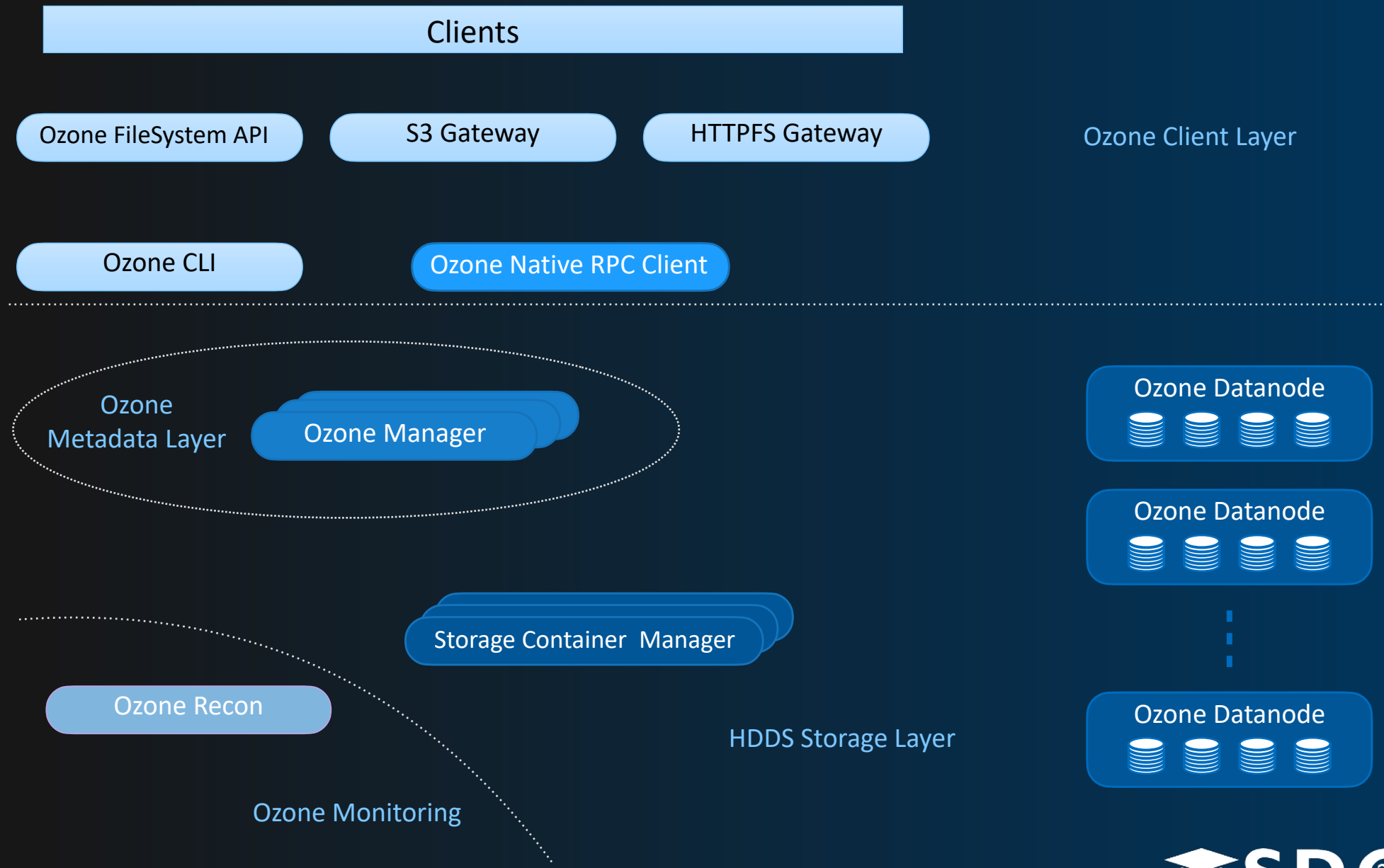
A brief project overview

Security in Apache Ozone

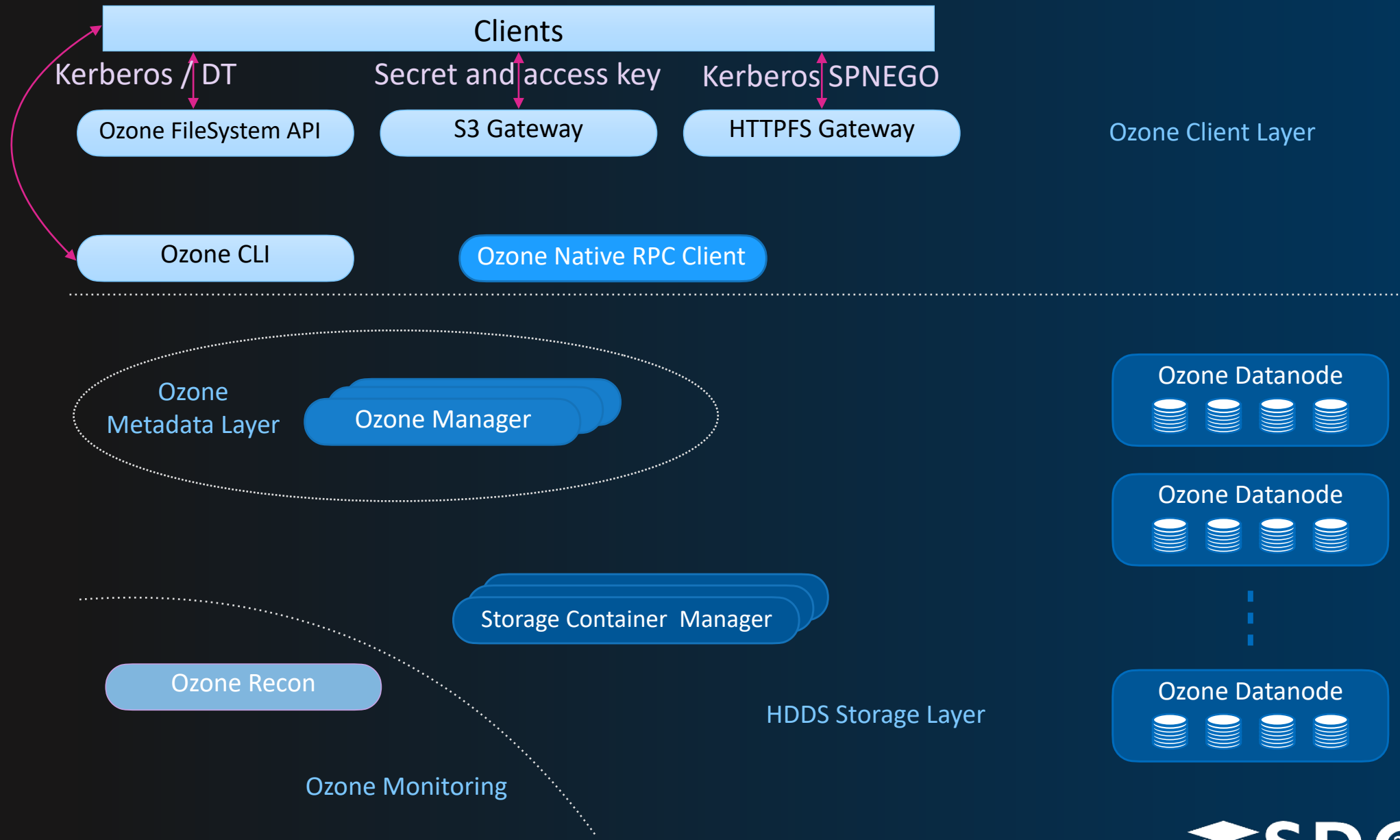
Tokens

Public Key Infrastructure

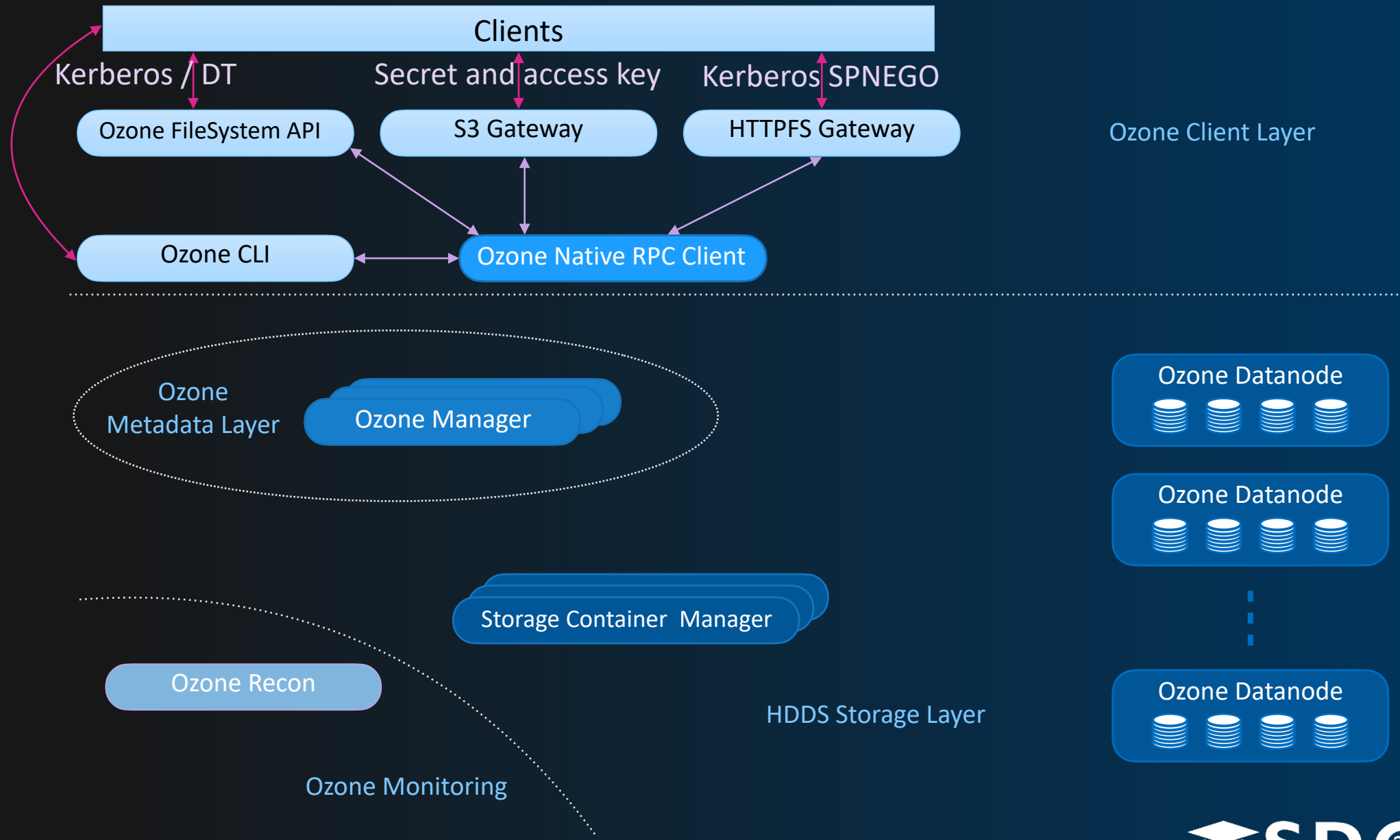
Authentication



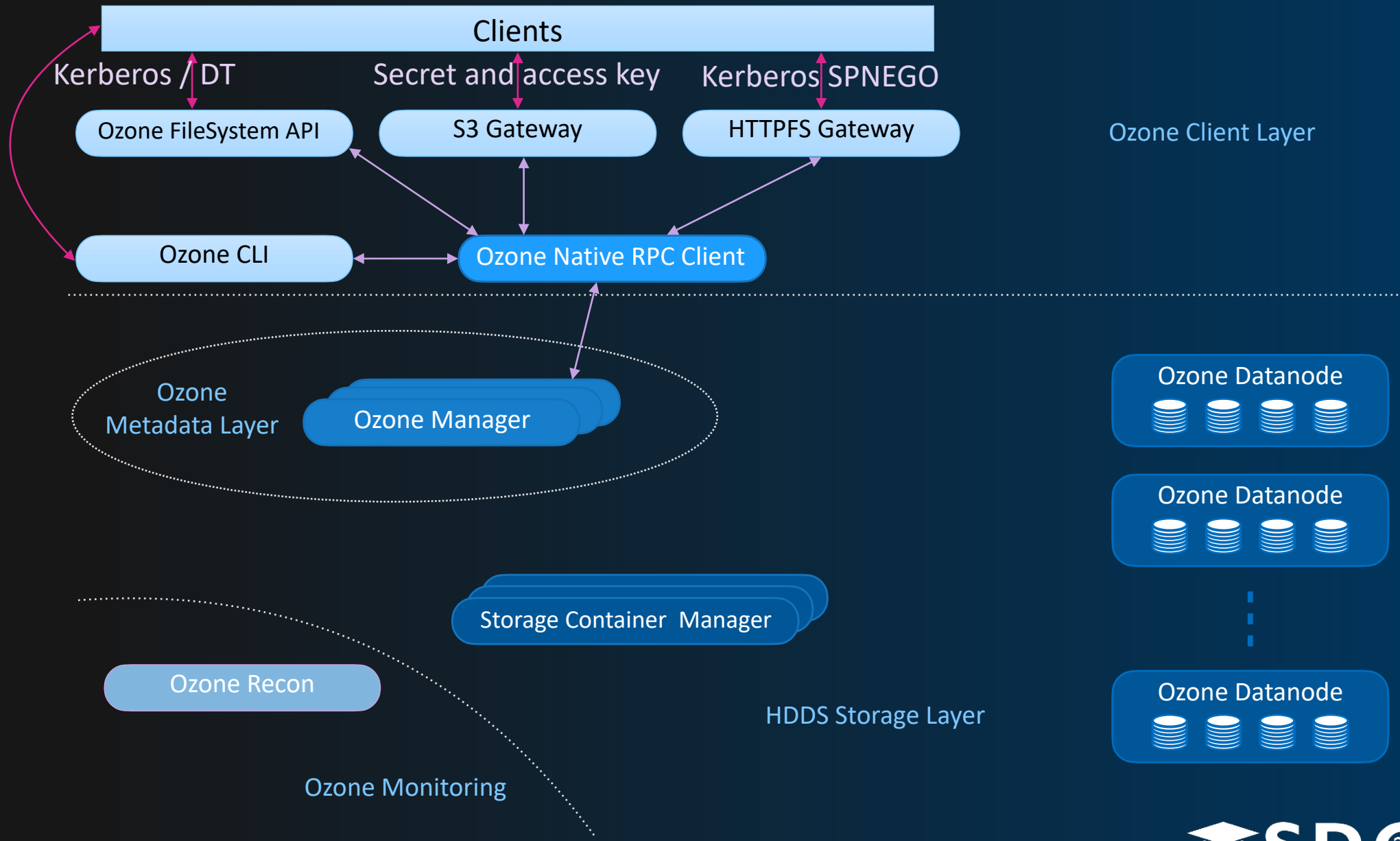
Authentication



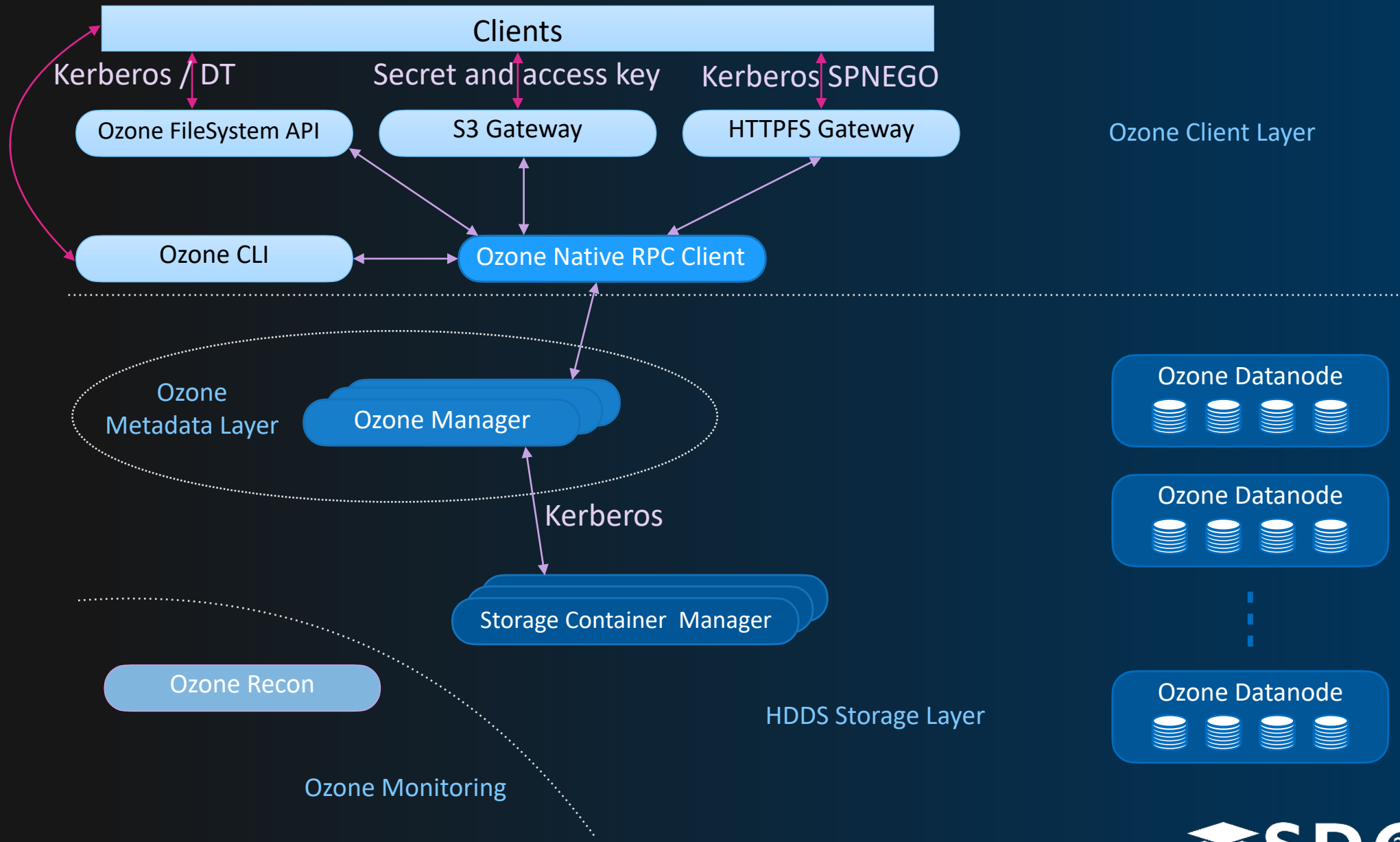
Authentication



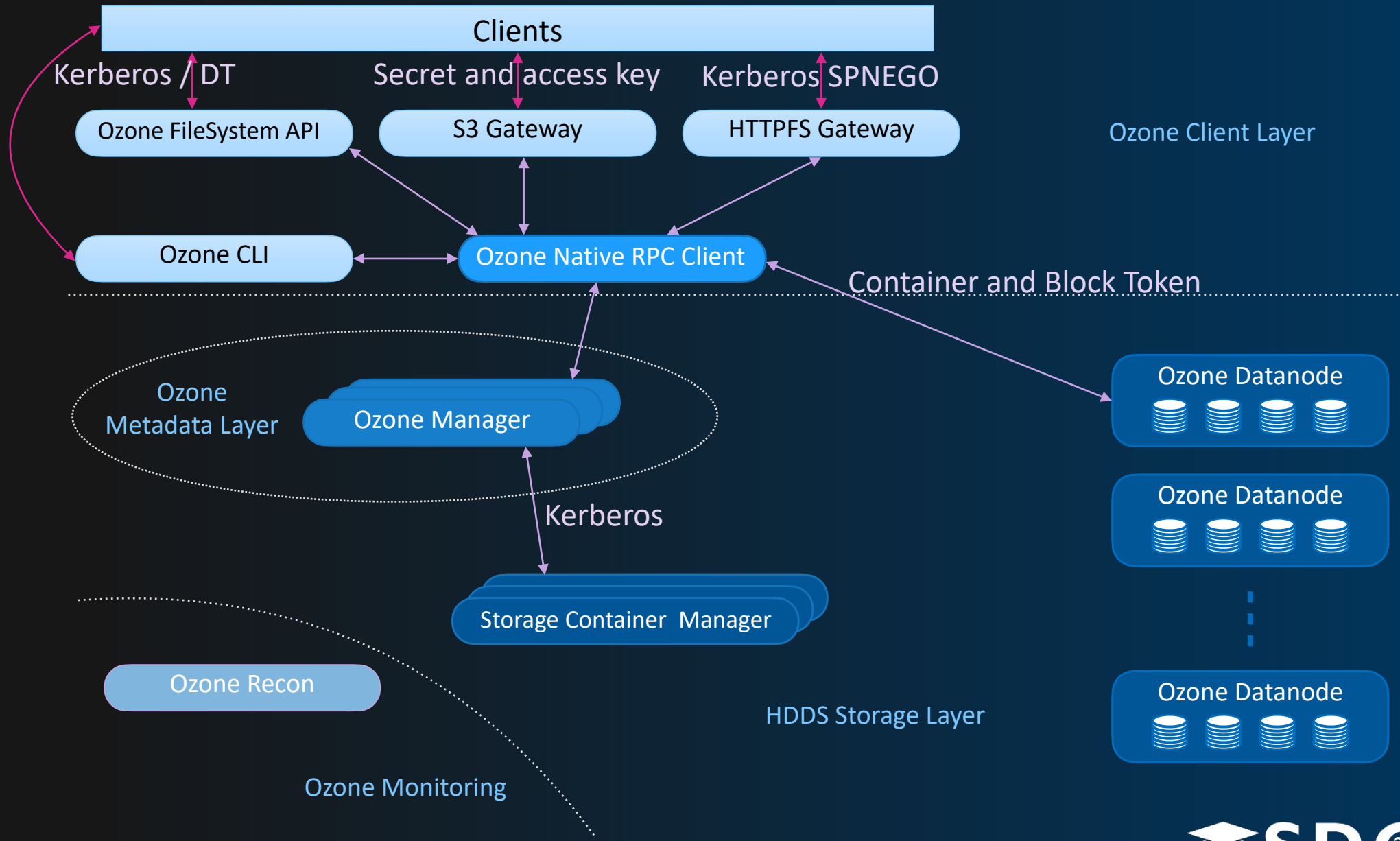
Authentication



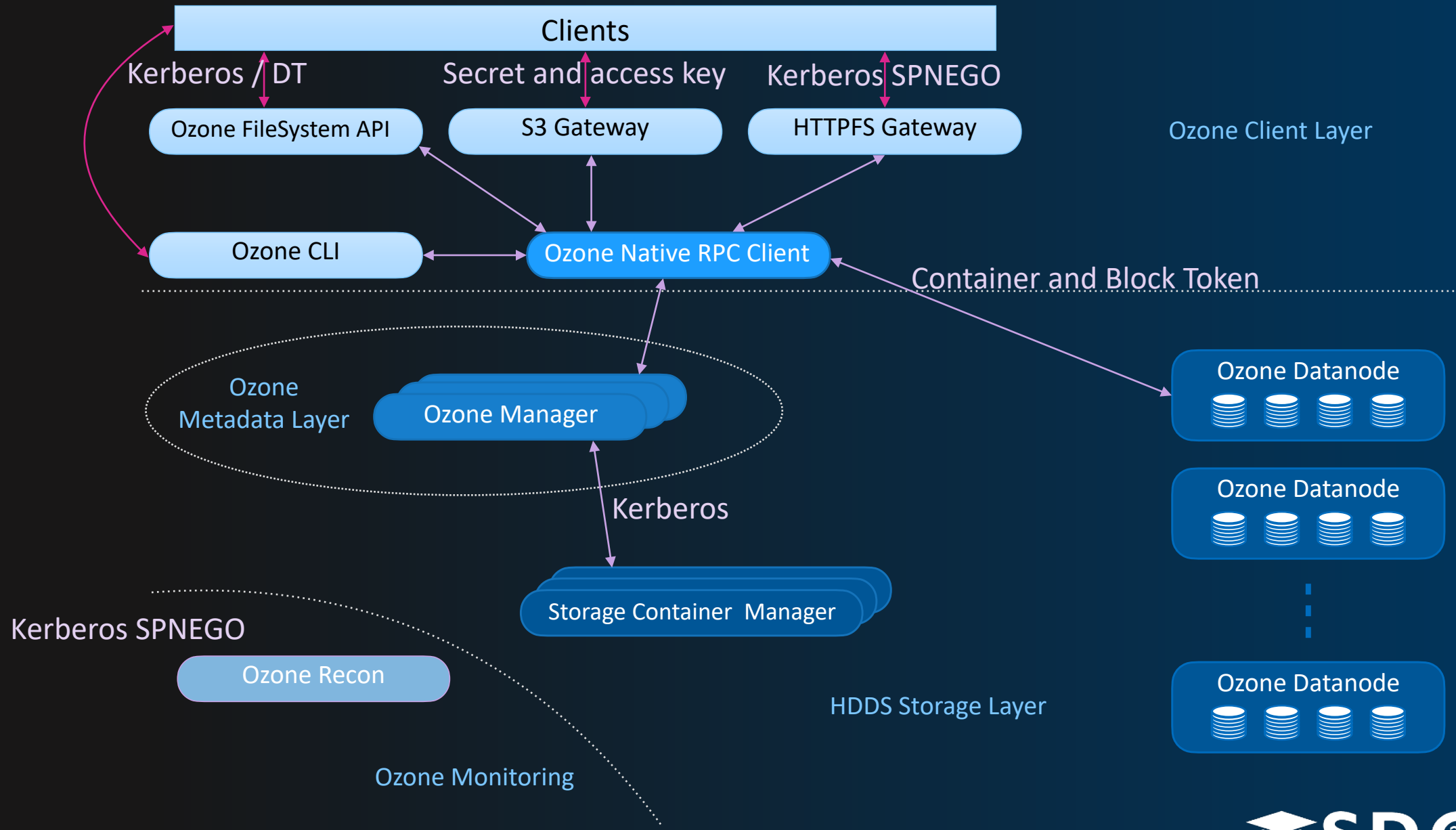
Authentication



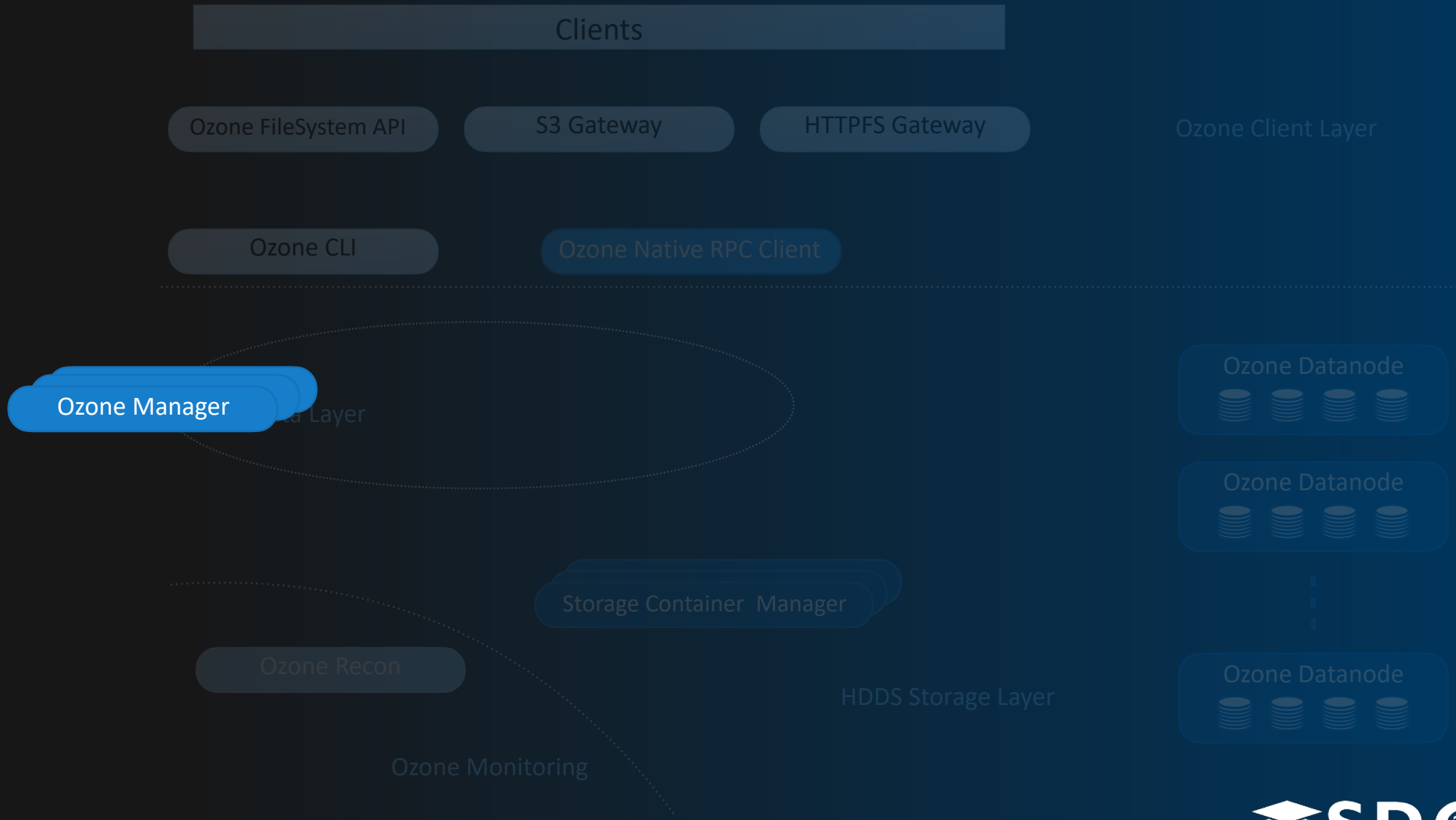
Authentication



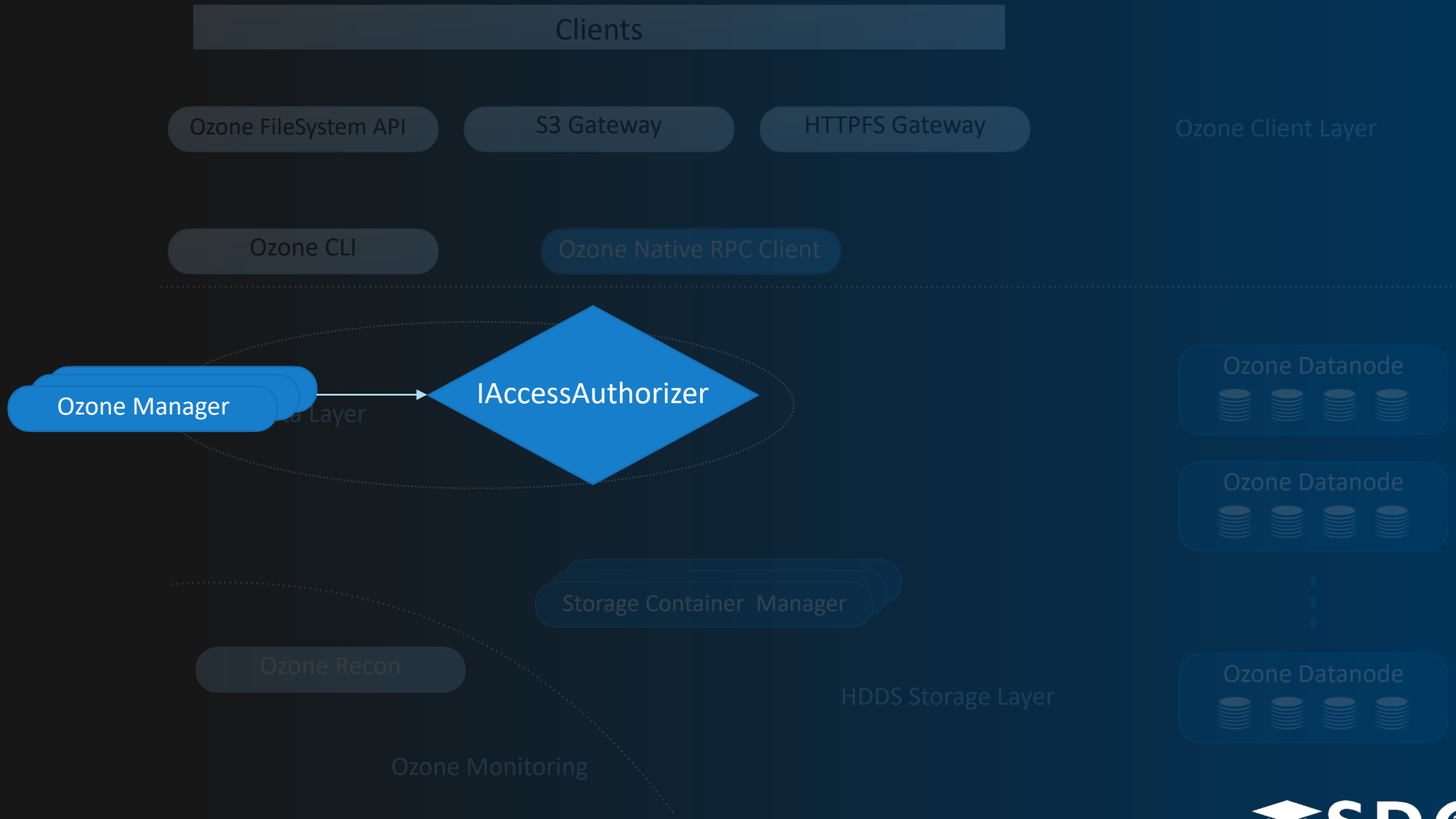
Authentication



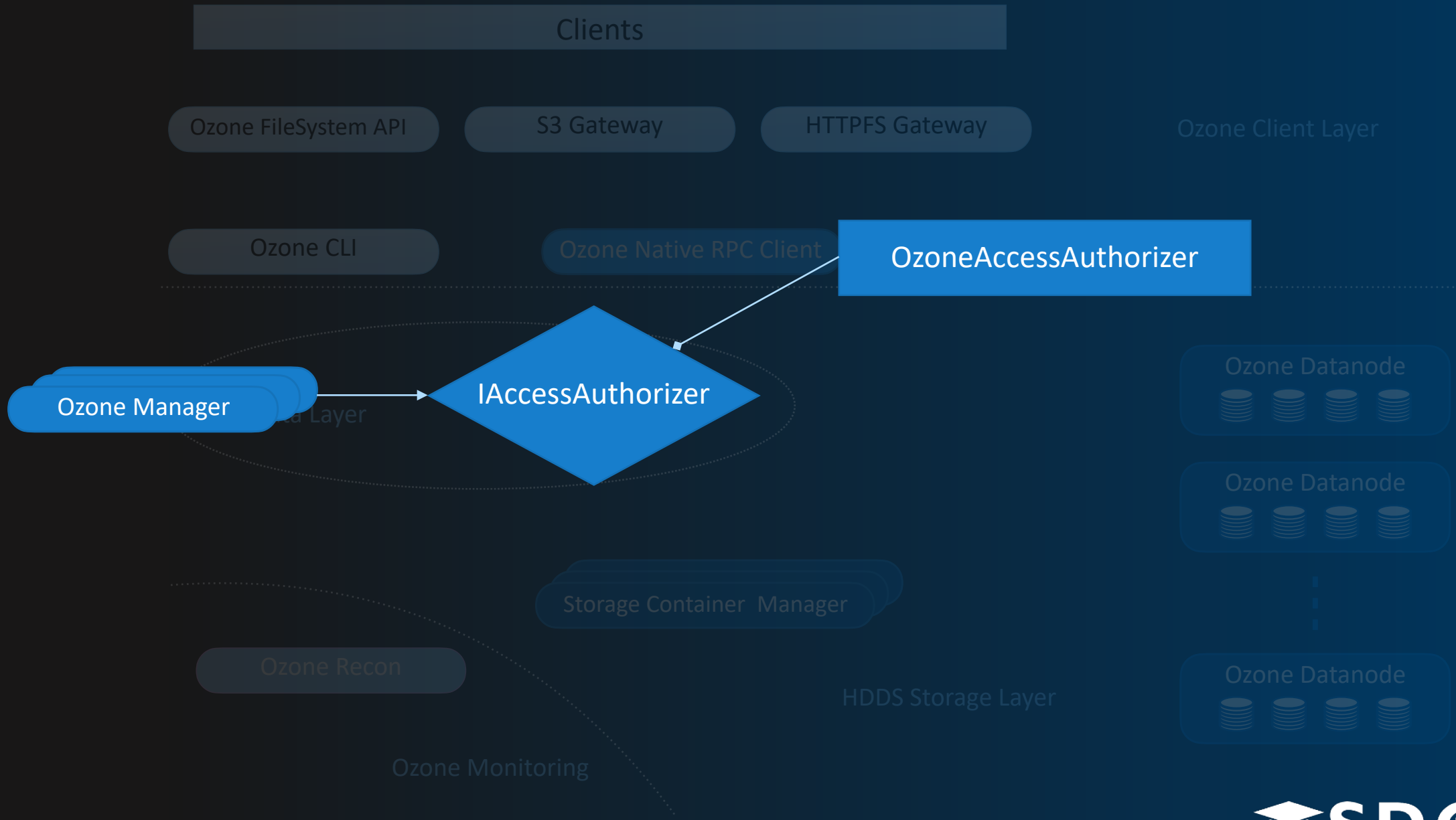
Authorization



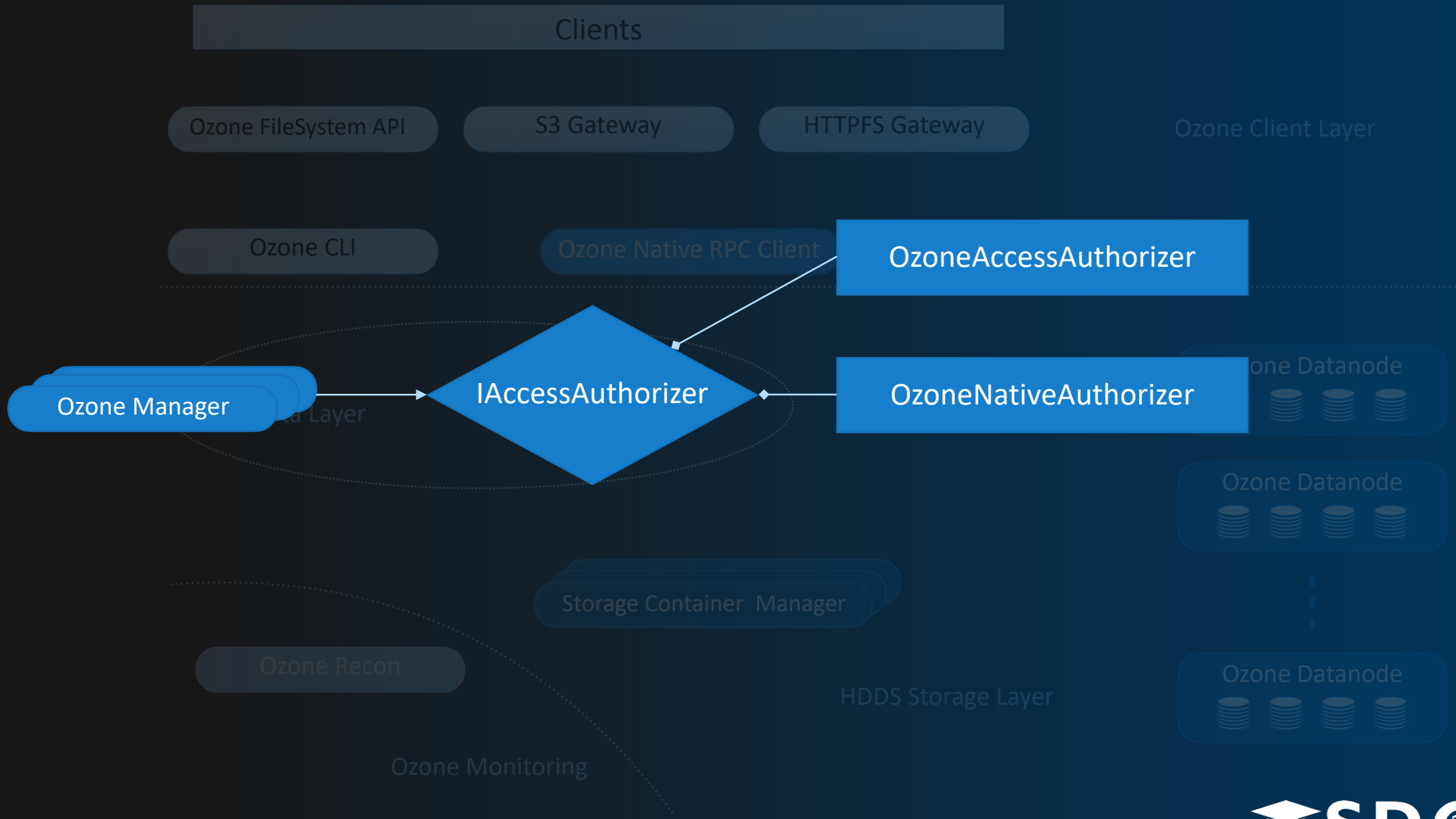
Authorization



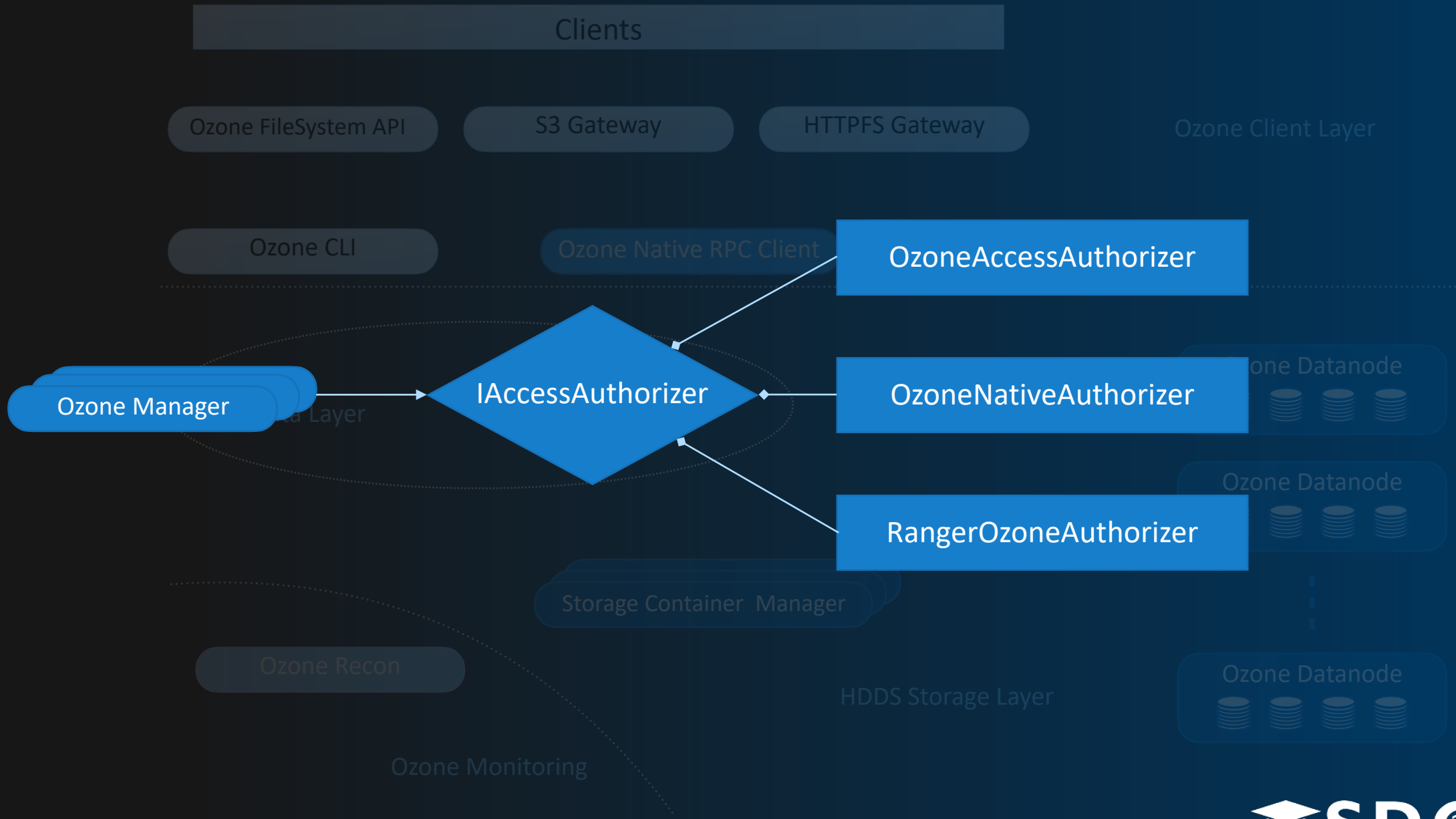
Authorization



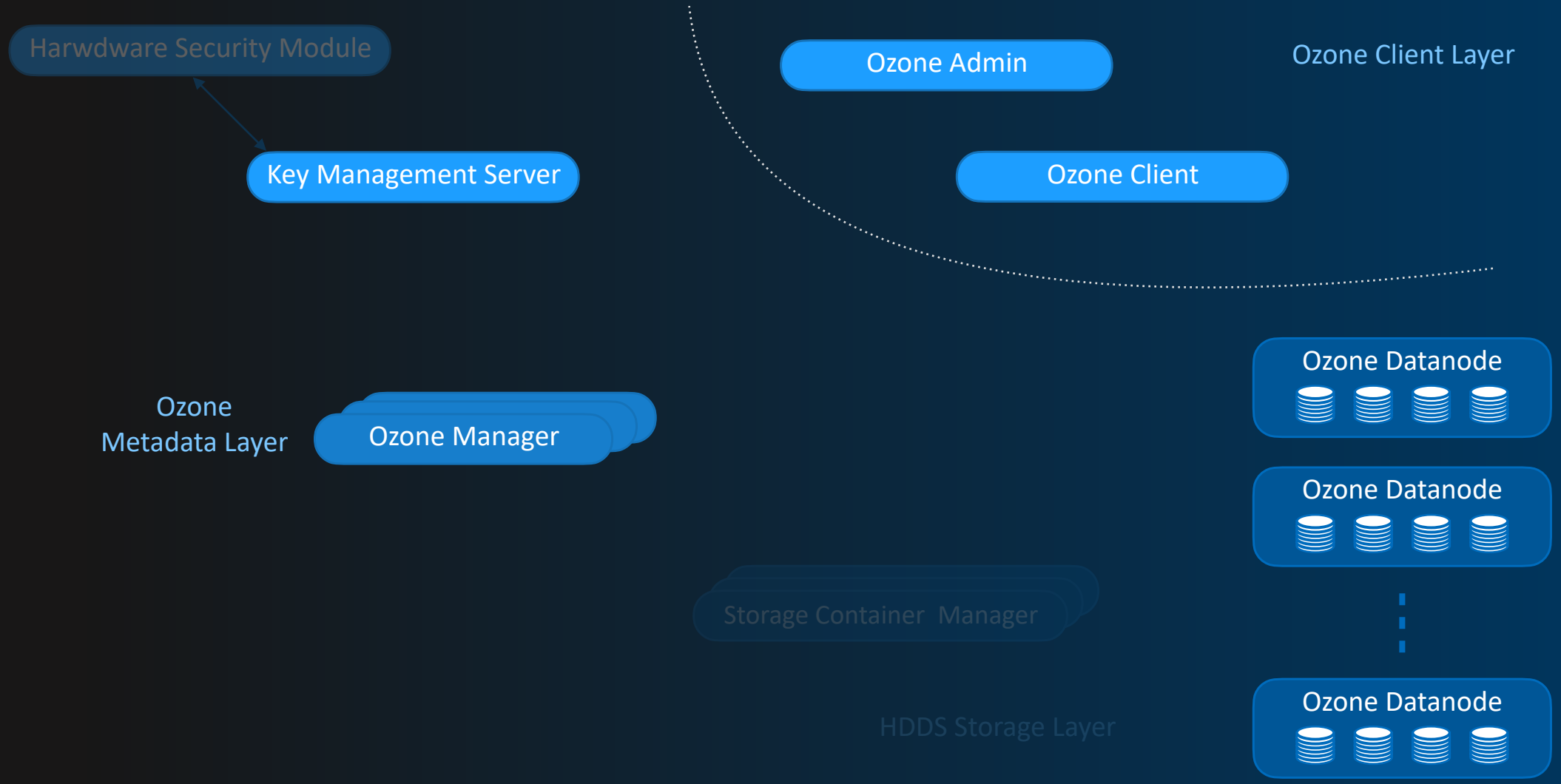
Authorization



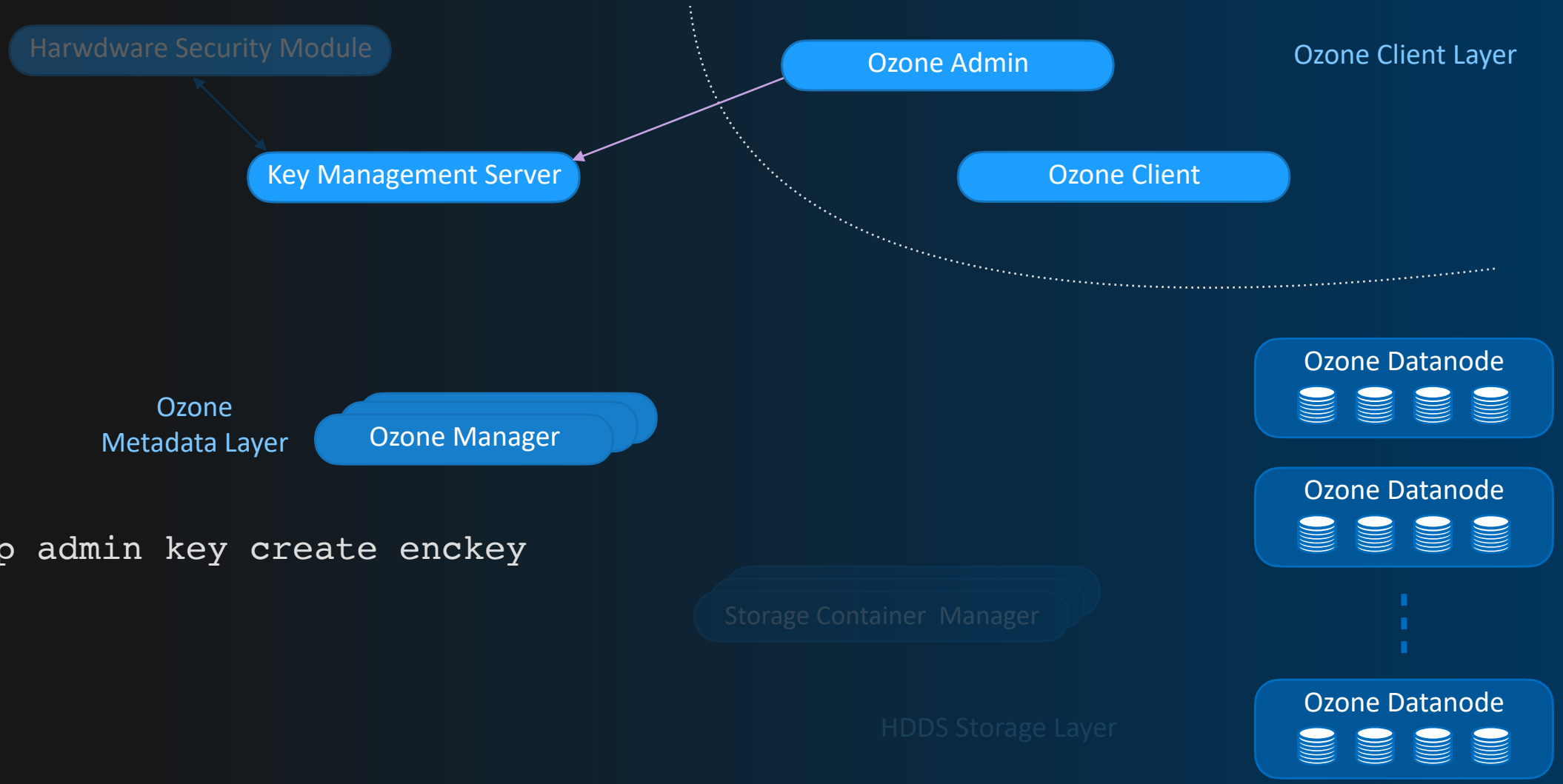
Authorization



At rest encryption - write

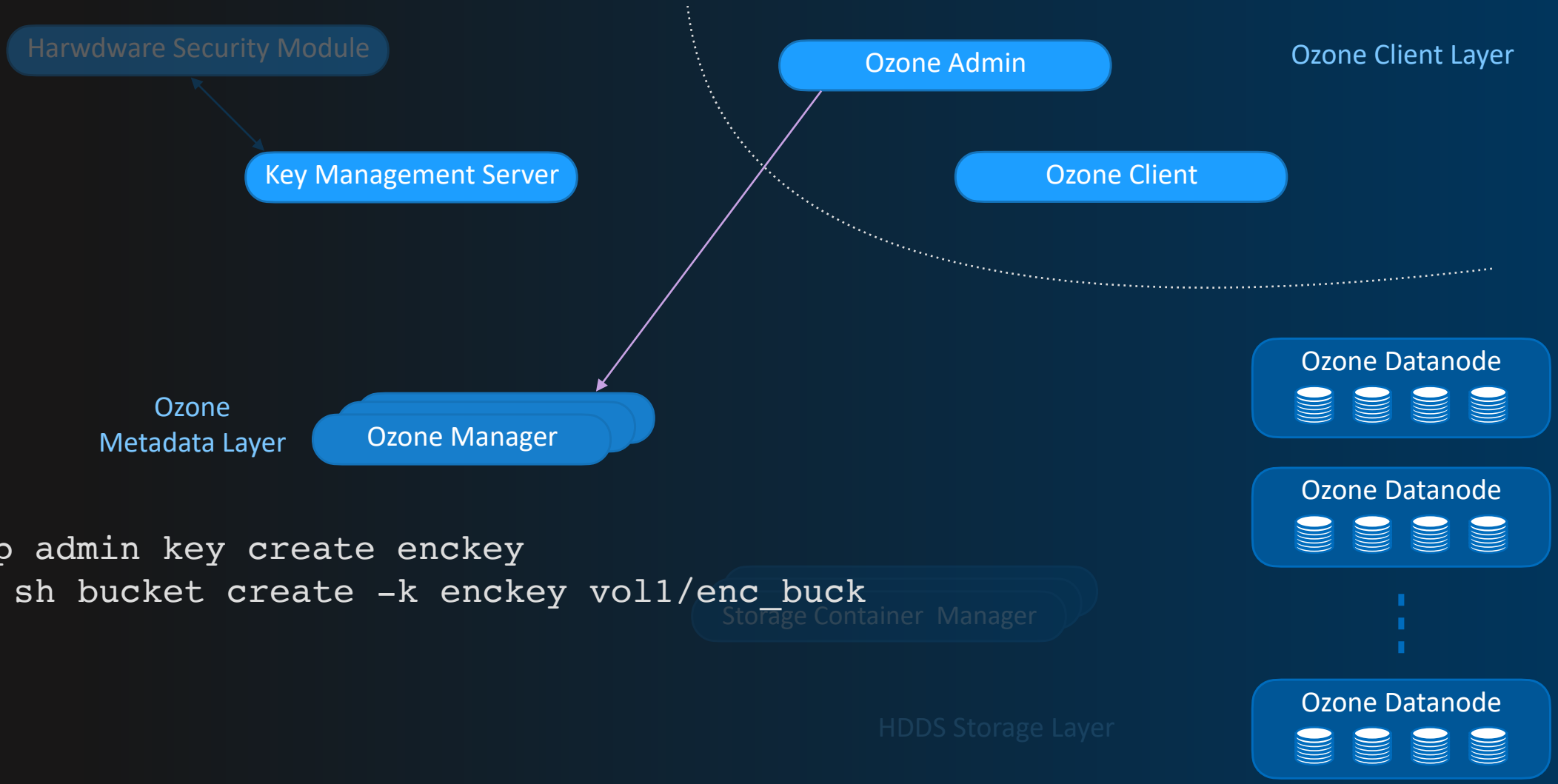


At rest encryption - write



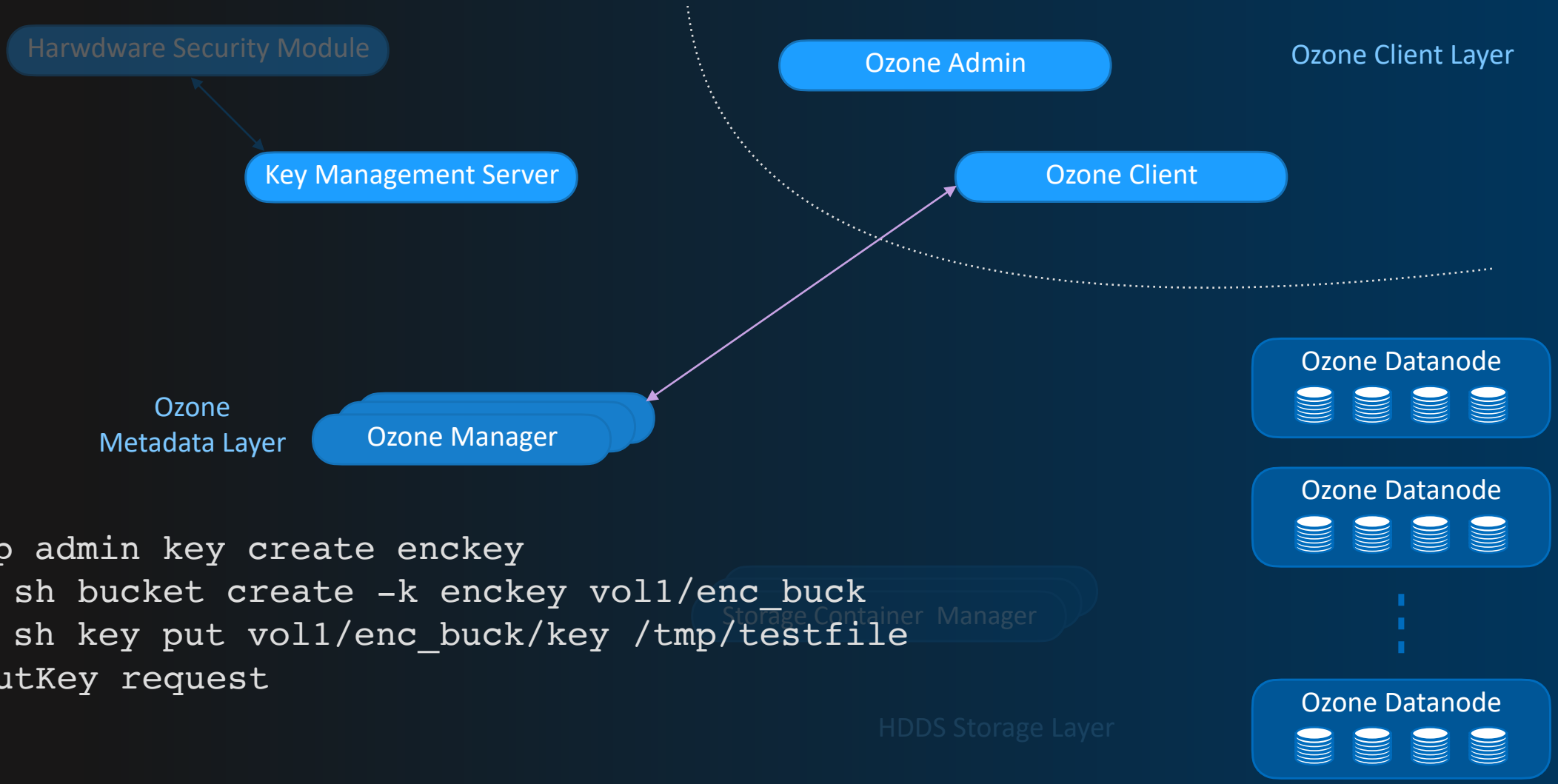

```
1. hadoop admin key create enckey
```

At rest encryption - write



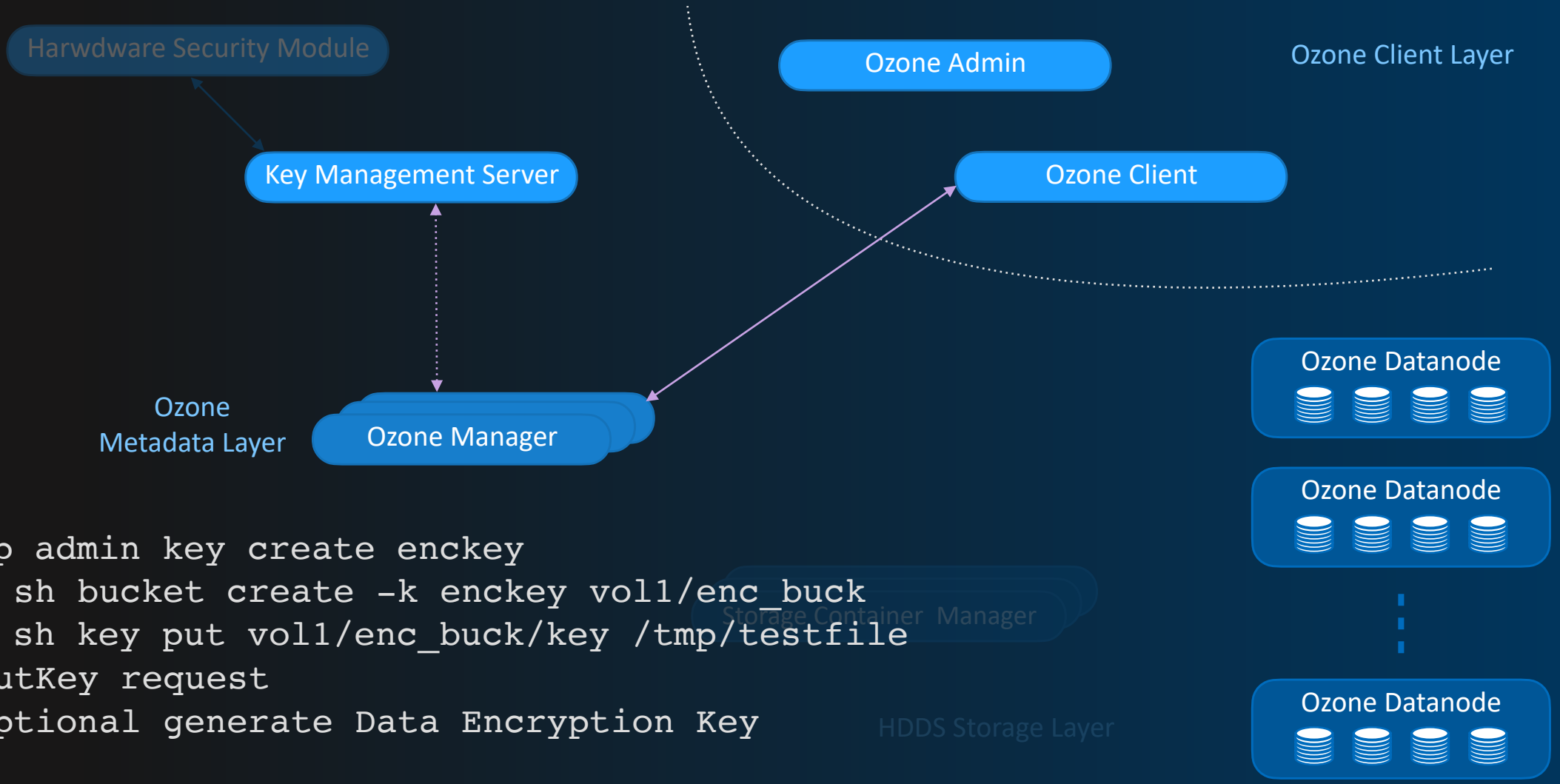
1. `hadoop admin key create enckey`
2. `ozone sh bucket create -k enckey vol1/enc_buck`

At rest encryption - write



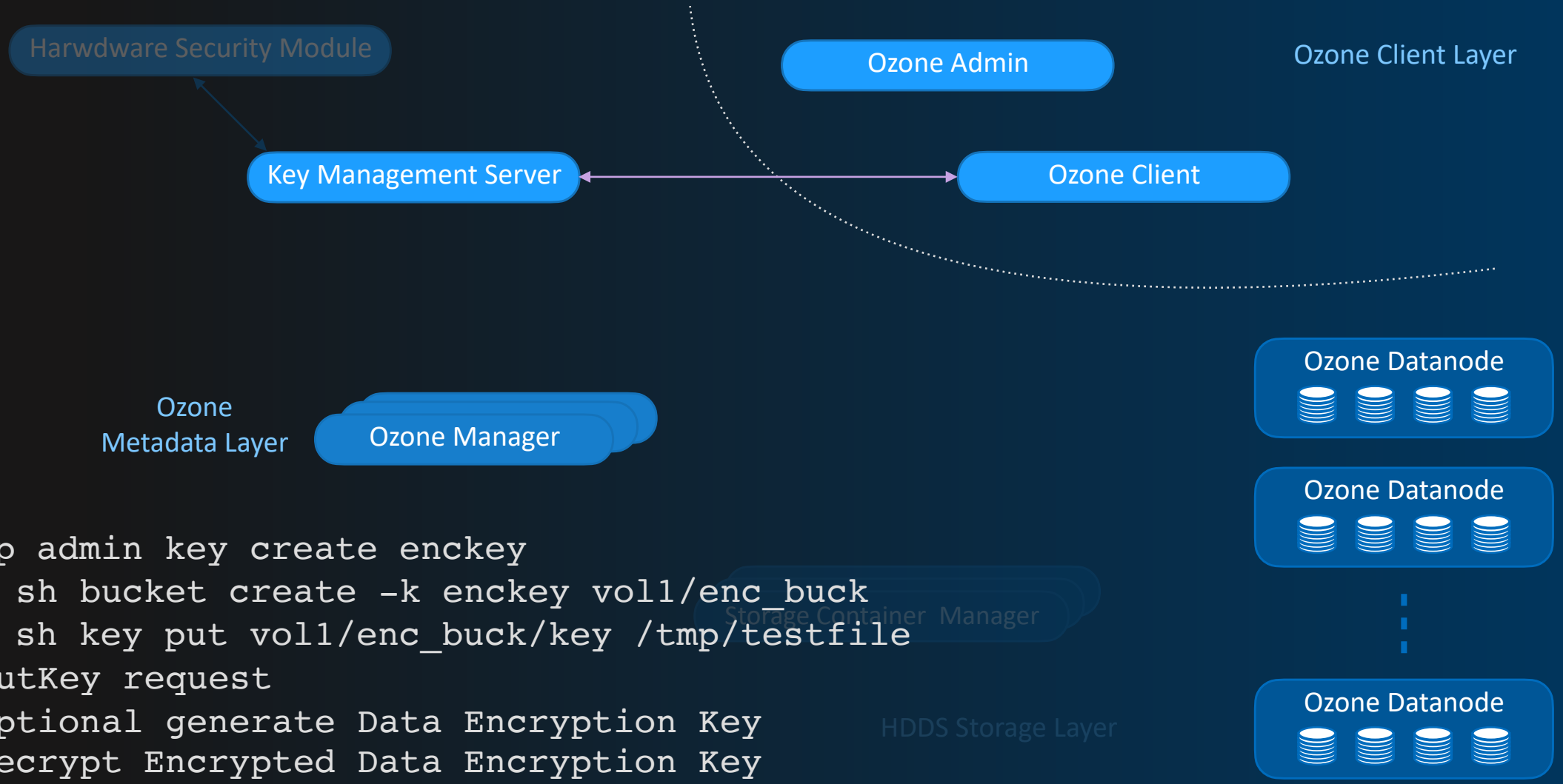

```
1. hadoop admin key create enckey
2. ozone sh bucket create -k enckey vol1/enc_buck
3. ozone sh key put vol1/enc_buck/key /tmp/testfile
  3.1. putKey request
```

At rest encryption - write



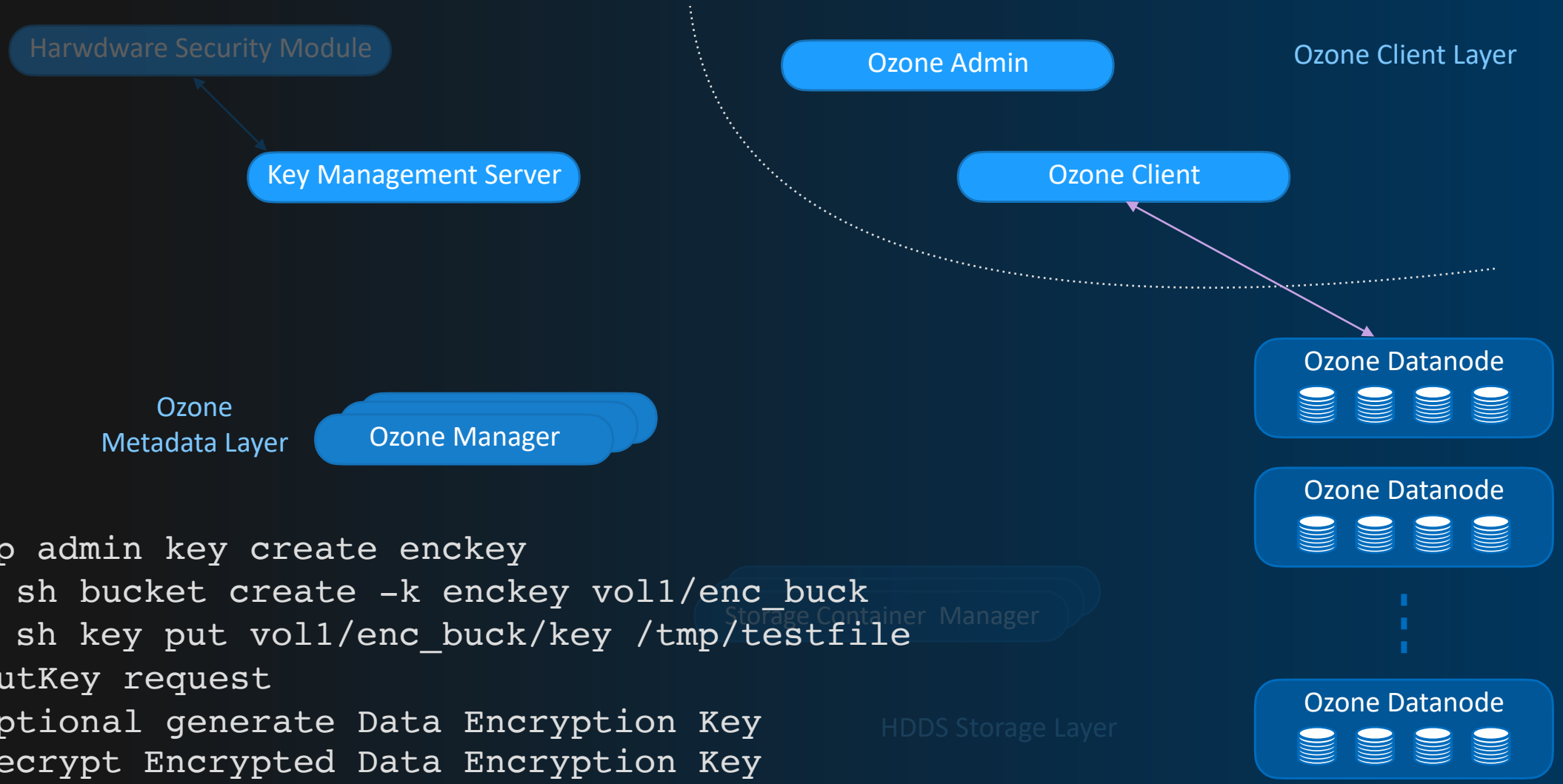

```
1. hadoop admin key create enckey
2. ozone sh bucket create -k enckey vol1/enc_buck
3. ozone sh key put vol1/enc_buck/key /tmp/testfile
   3.1. putKey request
   3.2. optional generate Data Encryption Key
```

At rest encryption - write



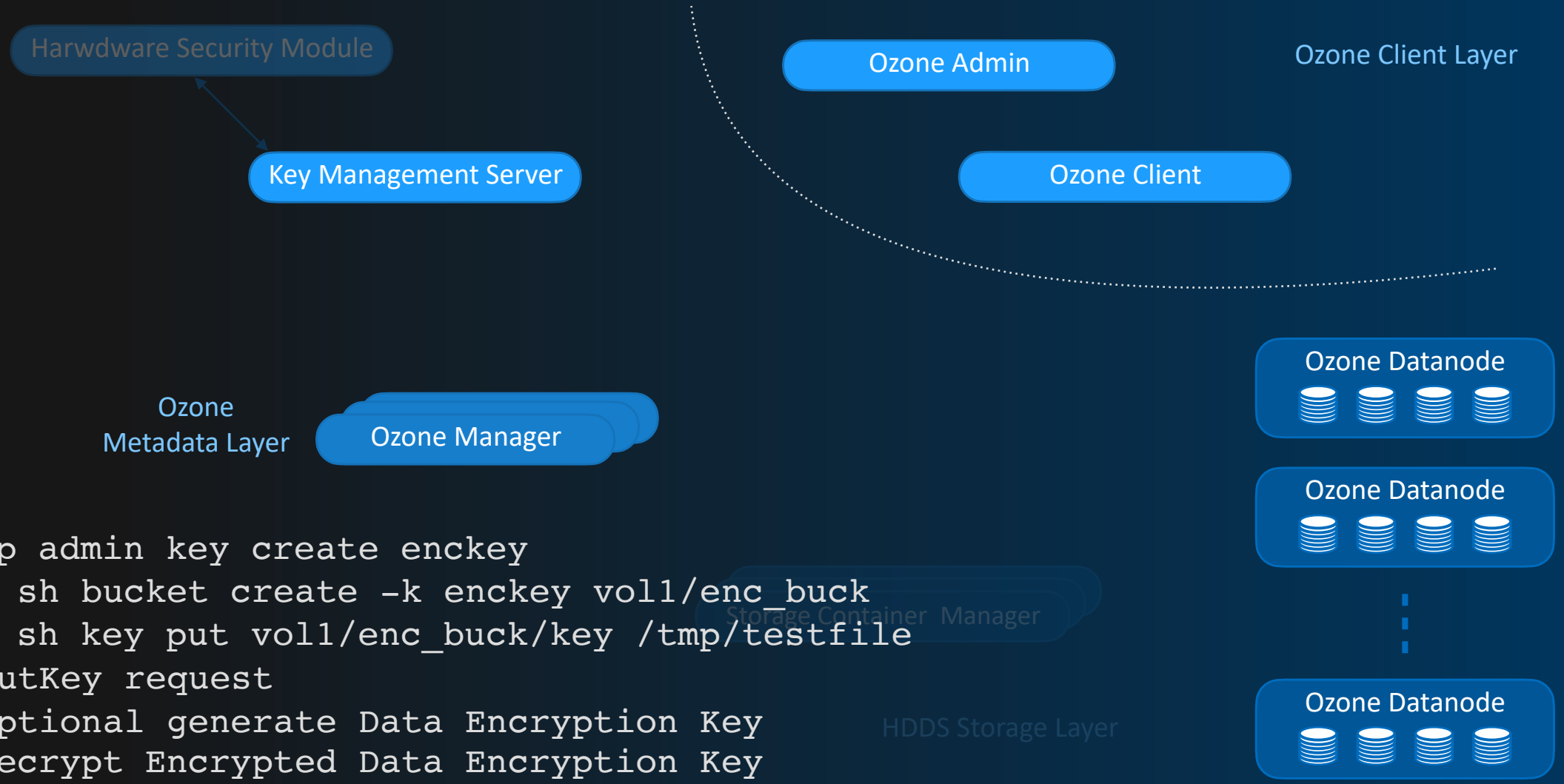

```
1. hadoop admin key create enckey
2. ozone sh bucket create -k enckey vol1/enc_buck
3. ozone sh key put vol1/enc_buck/key /tmp/testfile
   3.1. putKey request
   3.2. optional generate Data Encryption Key
   3.3. decrypt Encrypted Data Encryption Key
```

At rest encryption - write



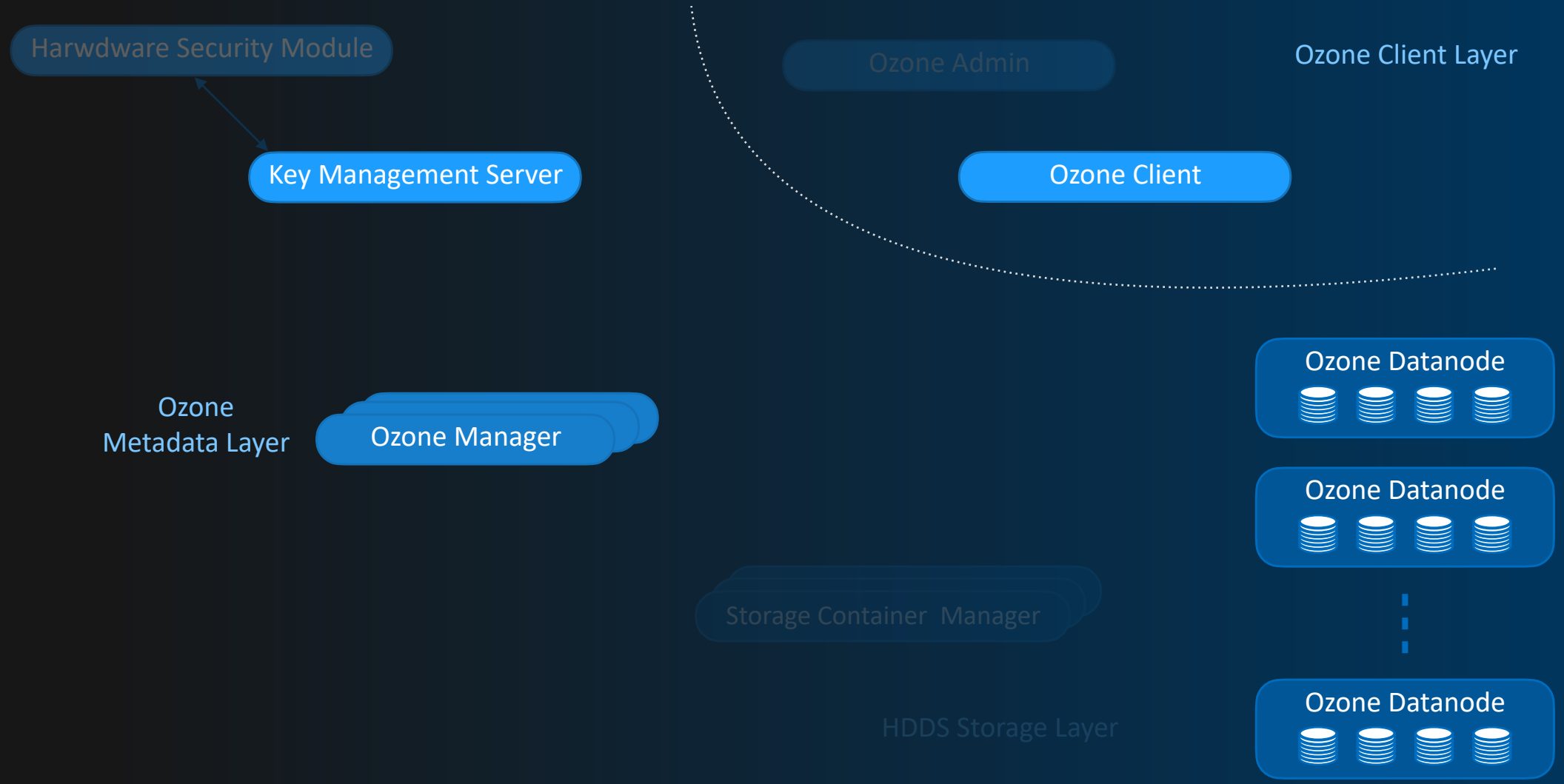
1. hadoop admin key create enckey
2. ozone sh bucket create -k enckey vol1/enc_buck
3. ozone sh key put vol1/enc_buck/key /tmp/testfile
   3.1. putKey request
   3.2. optional generate Data Encryption Key
   3.3. decrypt Encrypted Data Encryption Key
   3.4. send encrypted data

At rest encryption - write

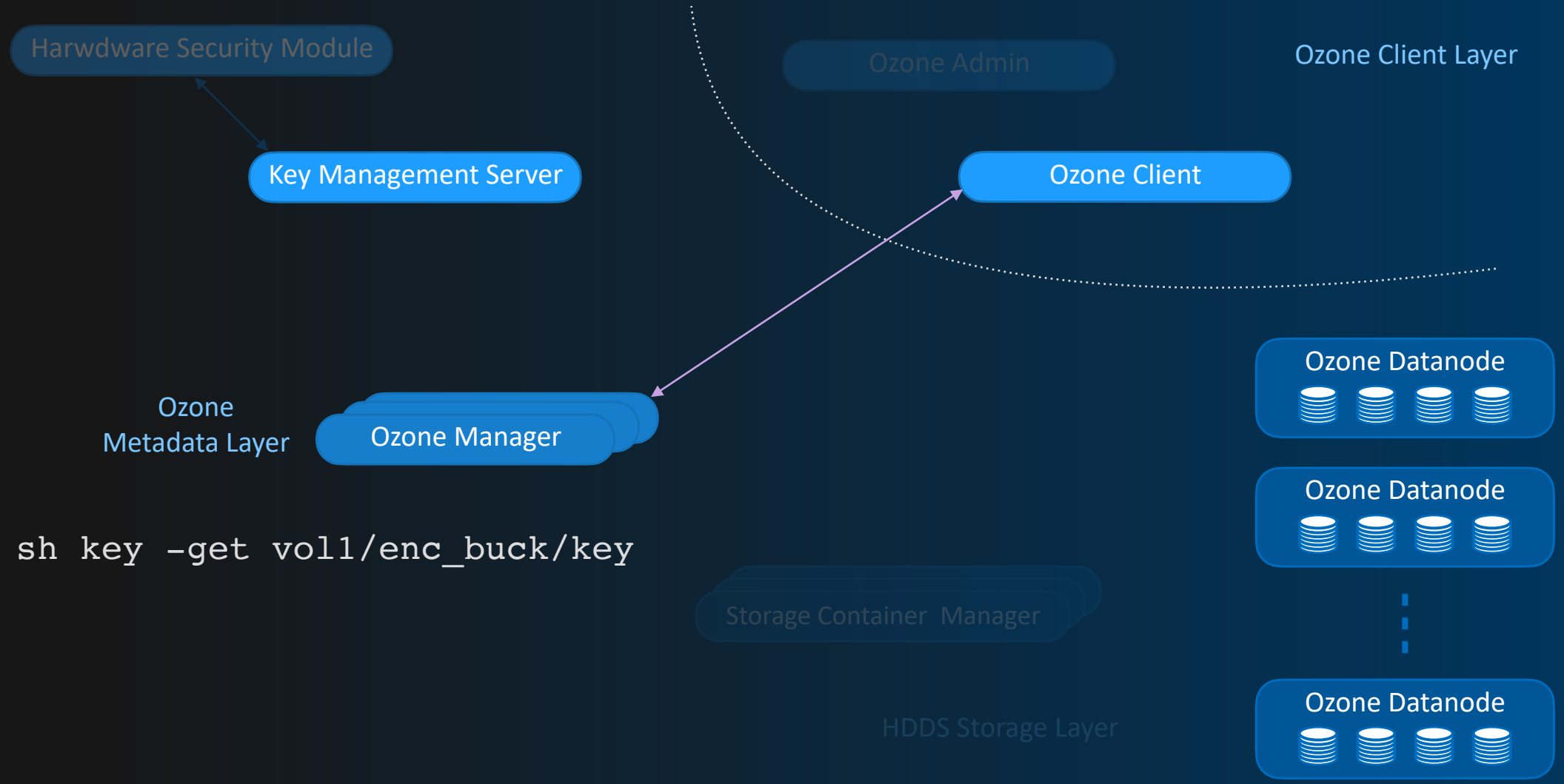


1. hadoop admin key create enckey
2. ozone sh bucket create -k enckey vol1/enc_buck
3. ozone sh key put vol1/enc_buck/key /tmp/testfile
   - 3.1. putKey request
   - 3.2. optional generate Data Encryption Key
   - 3.3. decrypt Encrypted Data Encryption Key
   - 3.4. send encrypted data
   - 3.5. commitKey request

At rest encryption - read

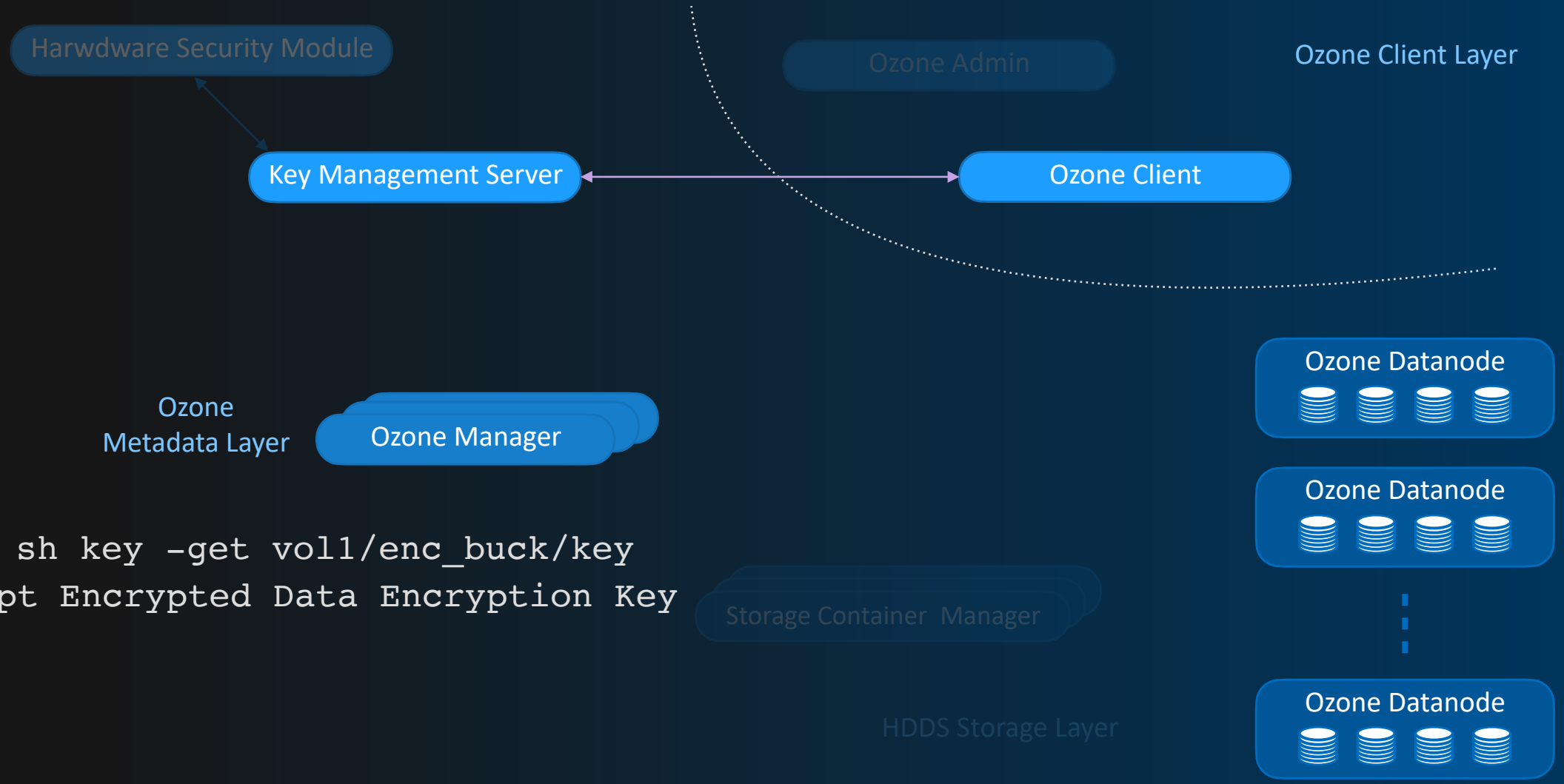


At rest encryption - read



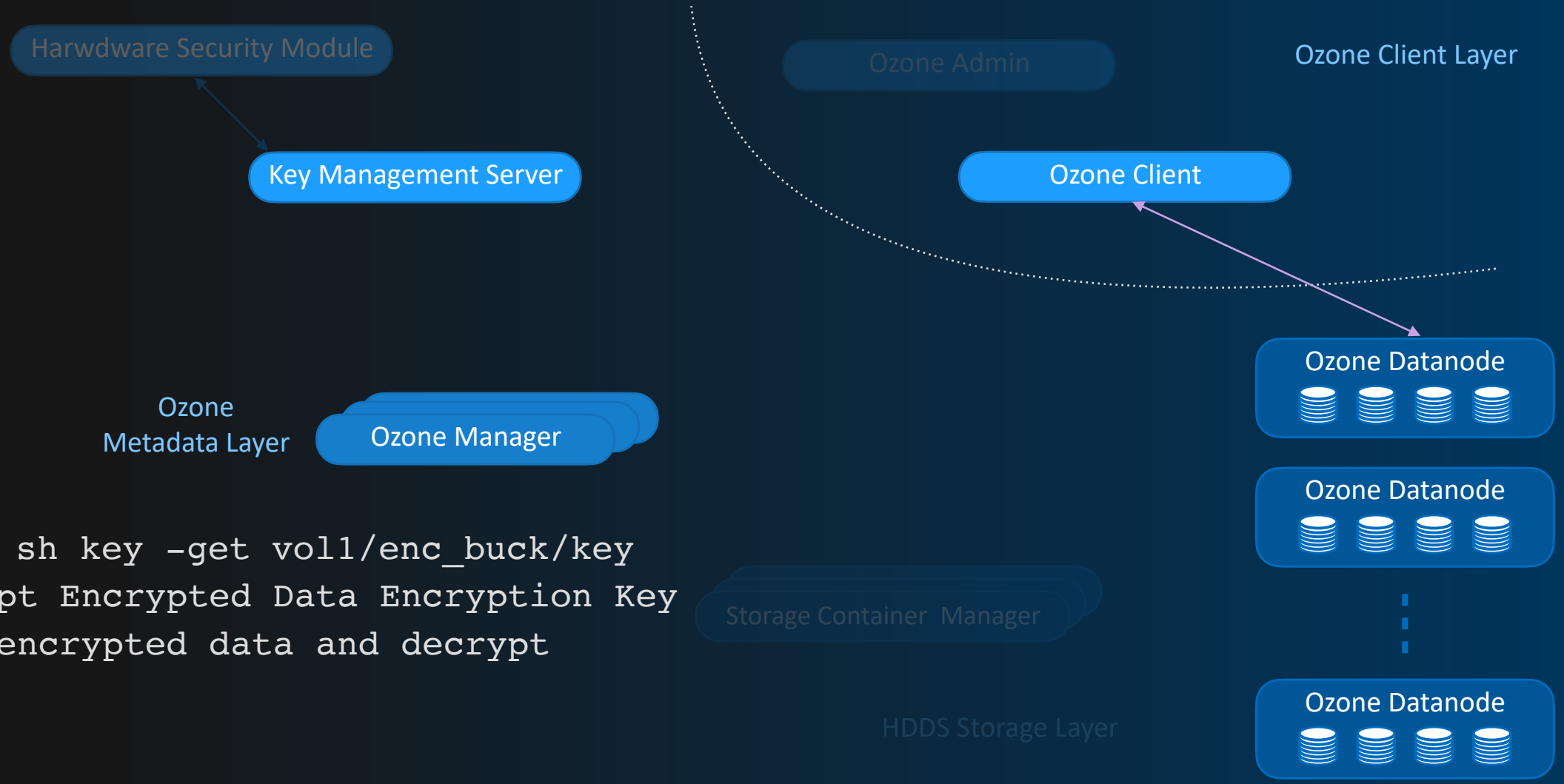

```
1. ozone sh key -get vol1/enc_buck/key
```

At rest encryption - read



1. `ozone sh key -get vol1/enc_buck/key`
2. `decrypt Encrypted Data Encryption Key`

At rest encryption - read



1. `ozone sh key -get vol1/enc_buck/key`
2. `decrypt Encrypted Data Encryption Key`
3. `read encrypted data and decrypt`

A brief project overview

Security in Apache Ozone

Tokens

Public Key Infrastructure

Token

- TokenIdentifier
- Password
- Kind
- Service
- Renewer

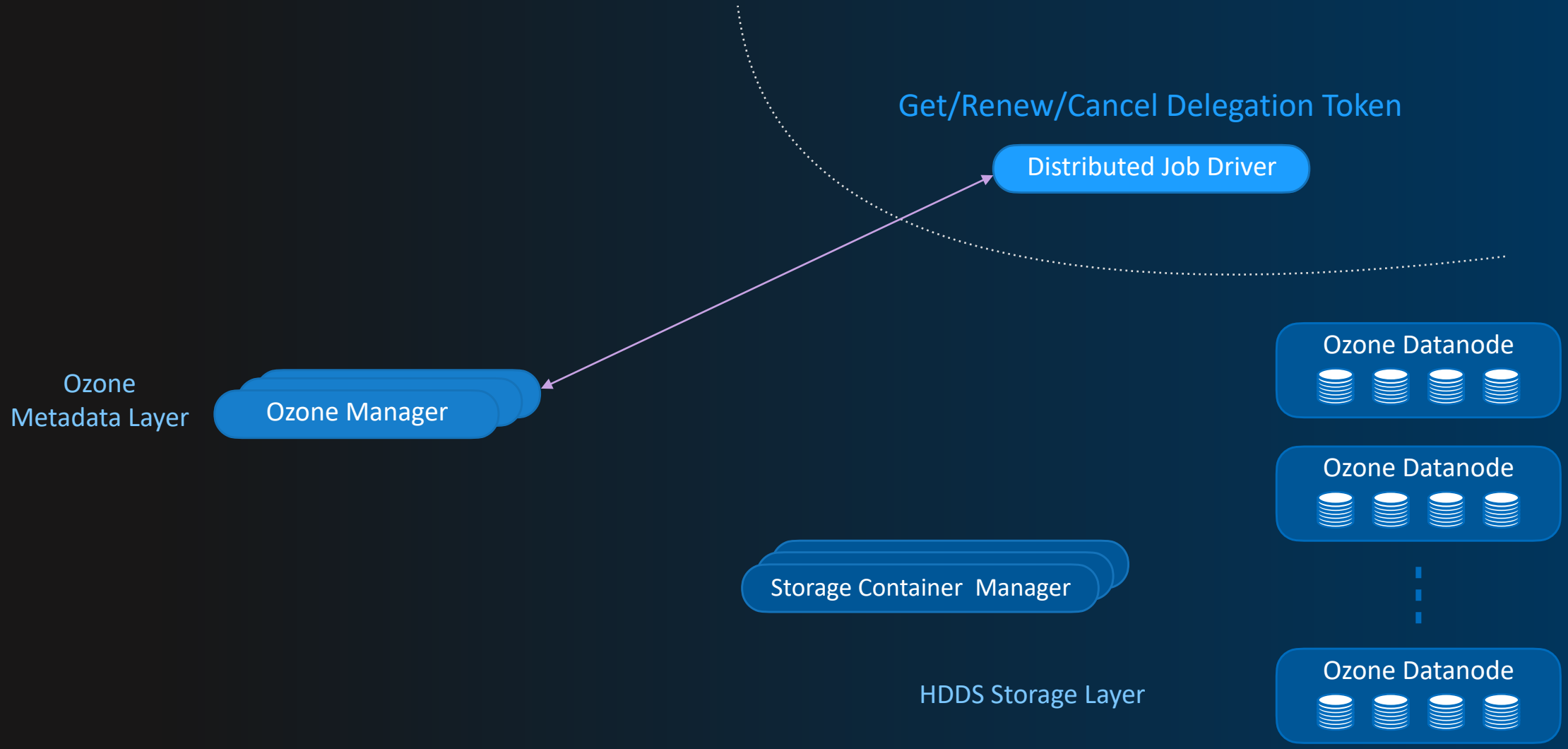
Token

- Password
- Kind
- Service
- Renewer

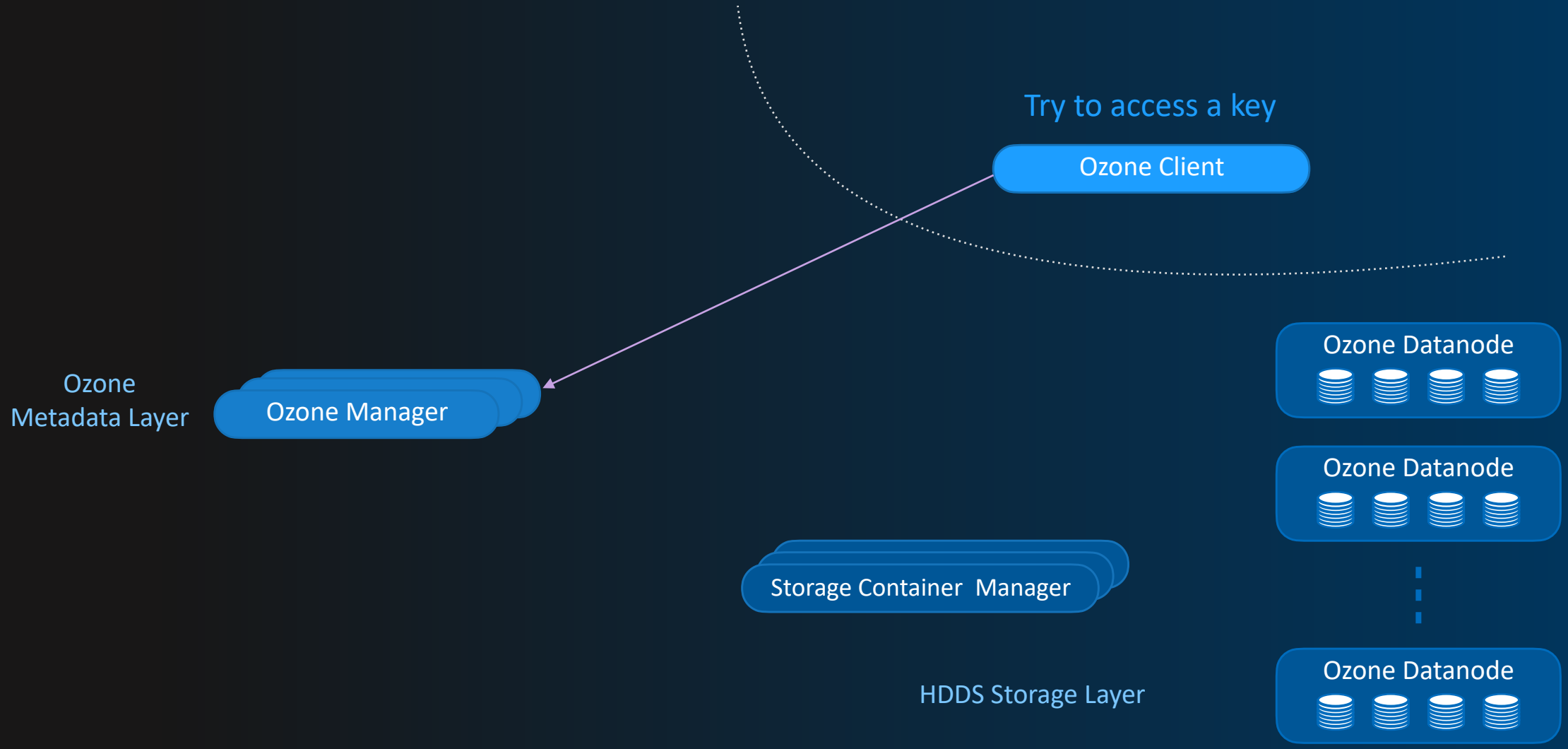
TokenIdentifier

- Kind
- User
- TrackingID
- Implementation dependent identification information

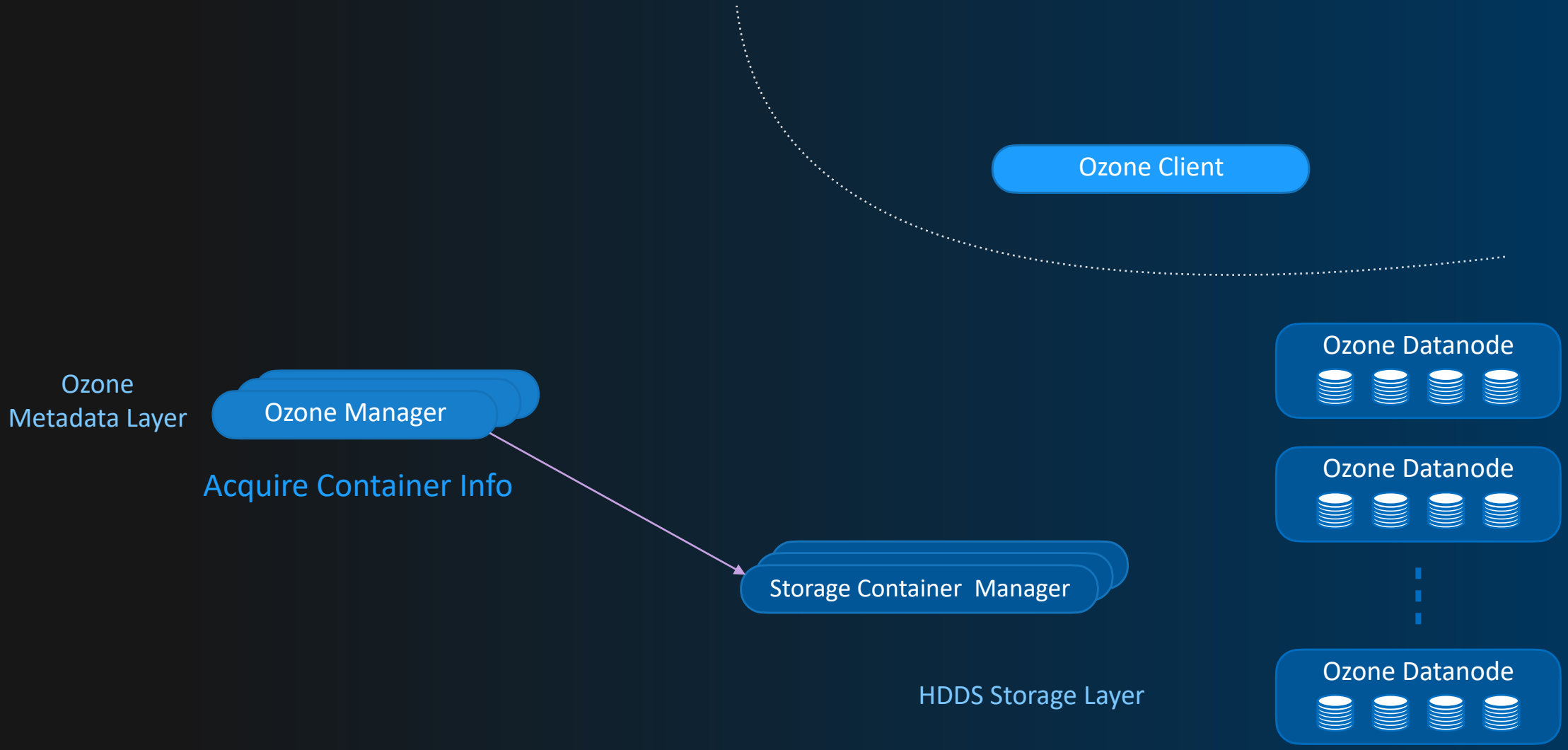
Tokens in communication



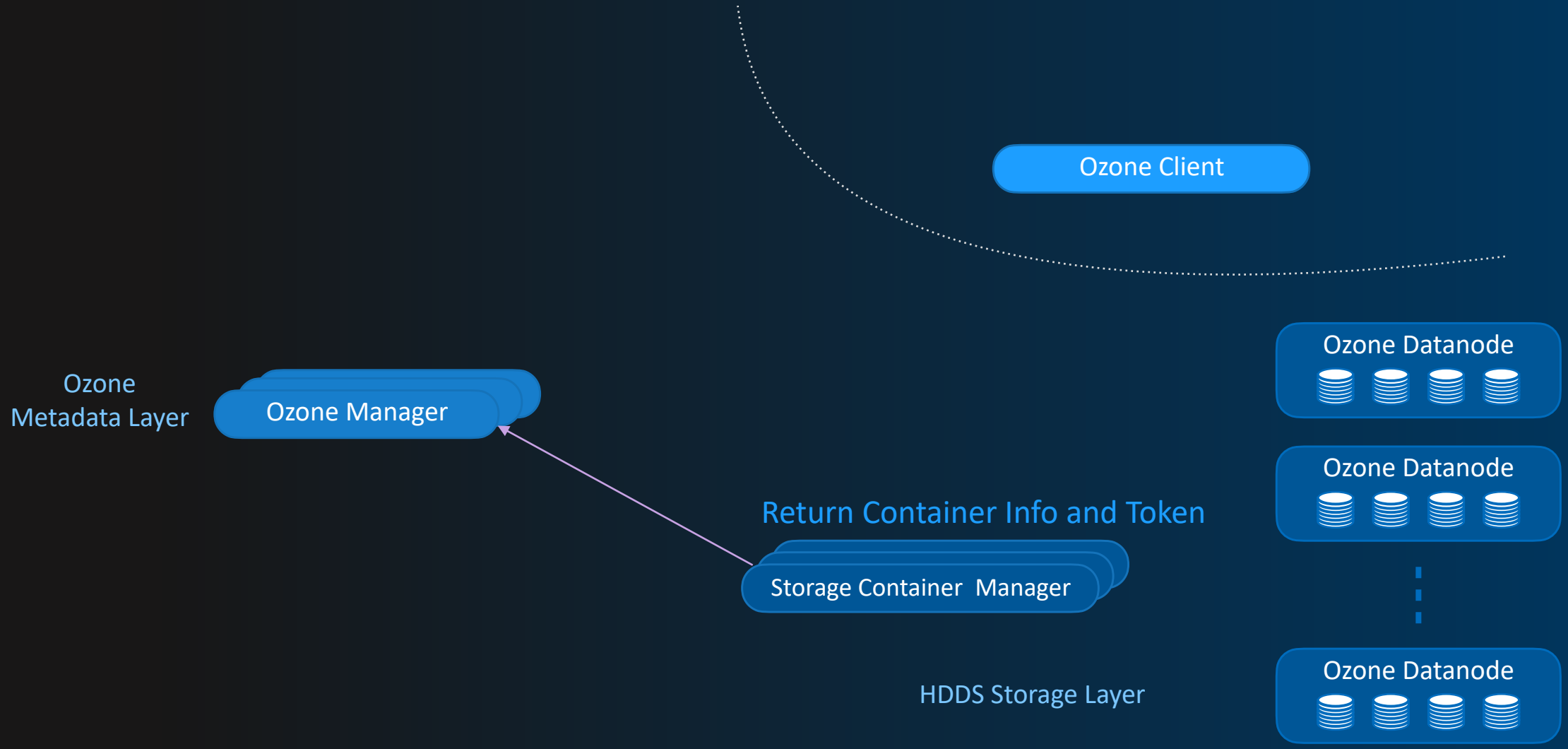
Tokens in communication



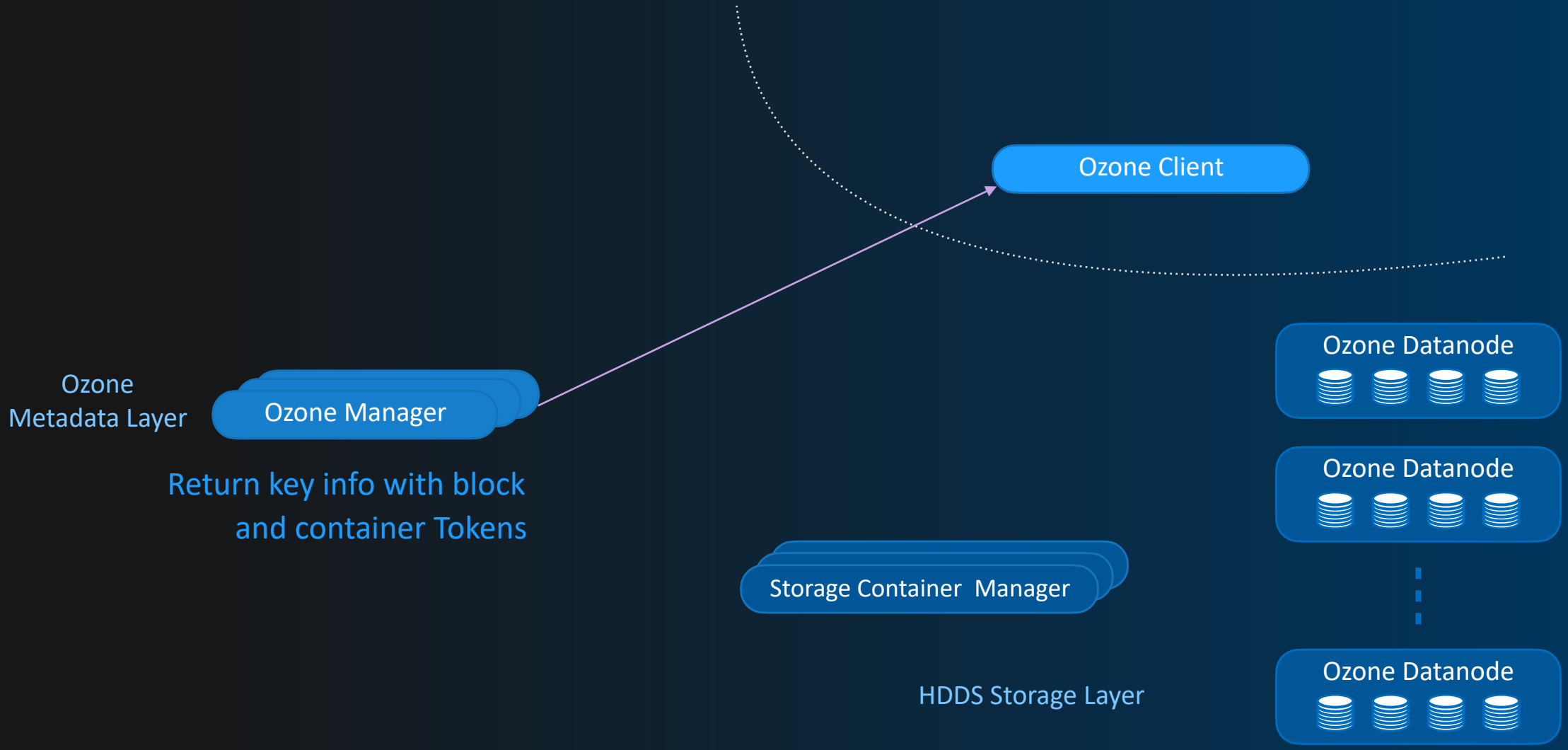
Tokens in communication



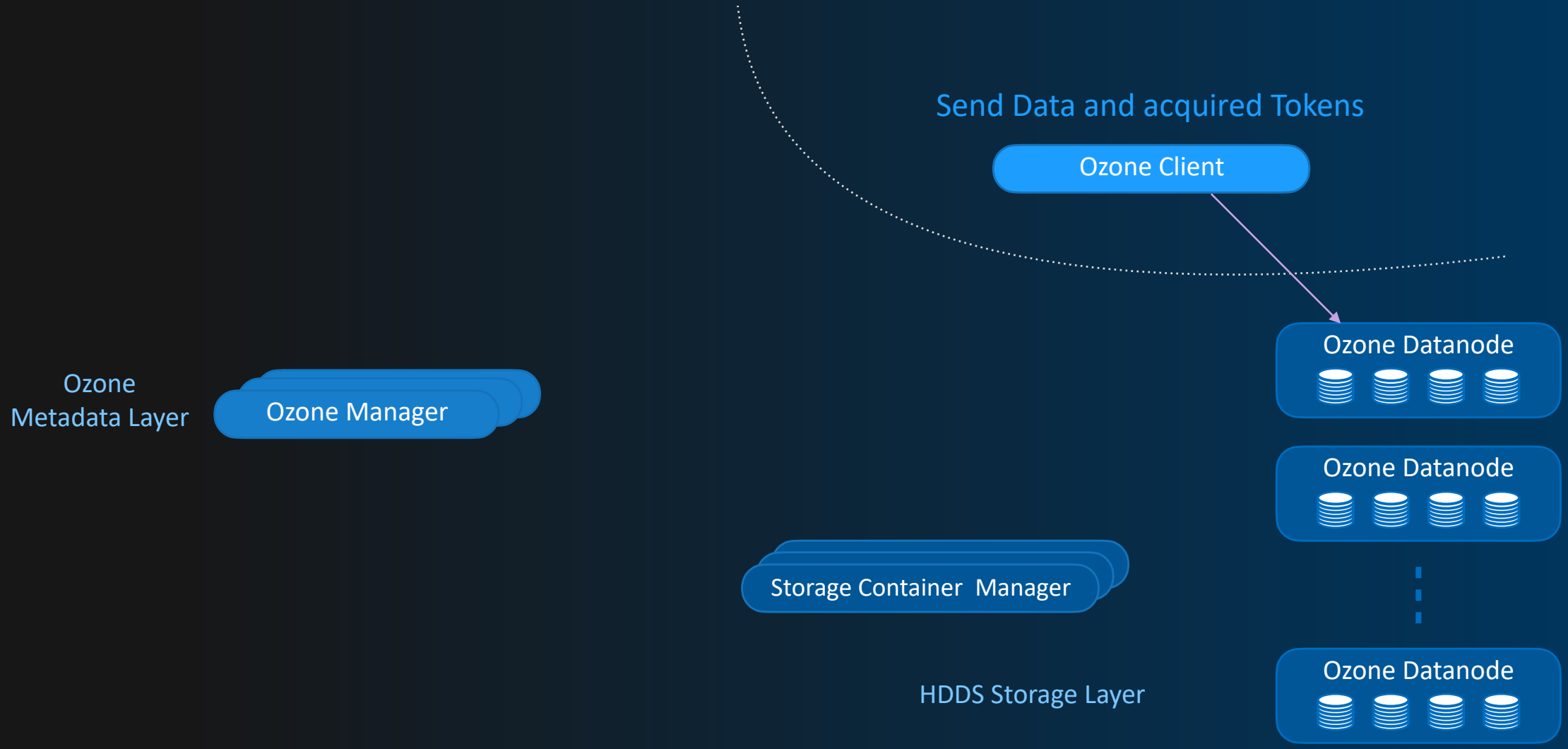
Tokens in communication



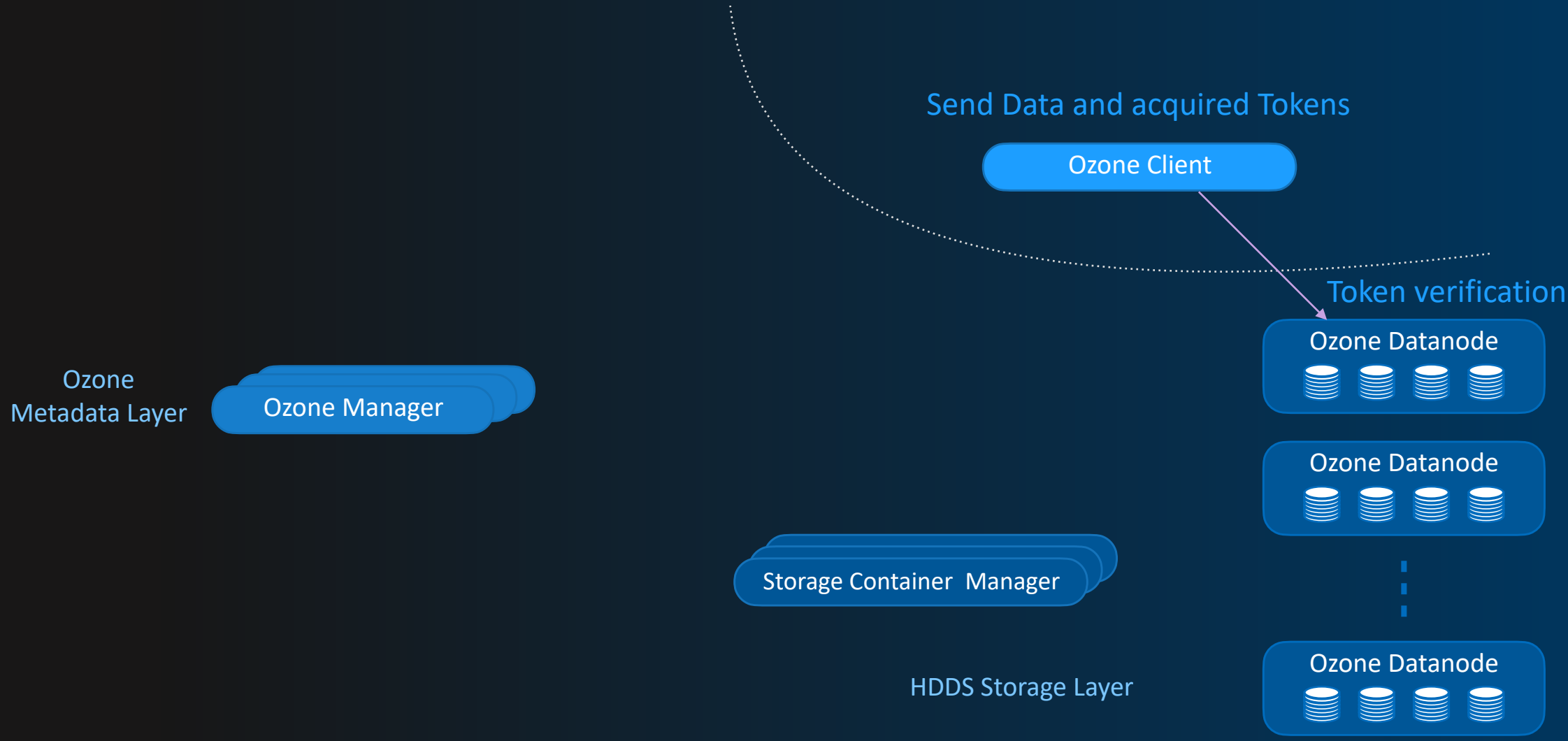
Tokens in communication



Tokens in communication



Tokens in communication



100x speedup in request processing? Really??

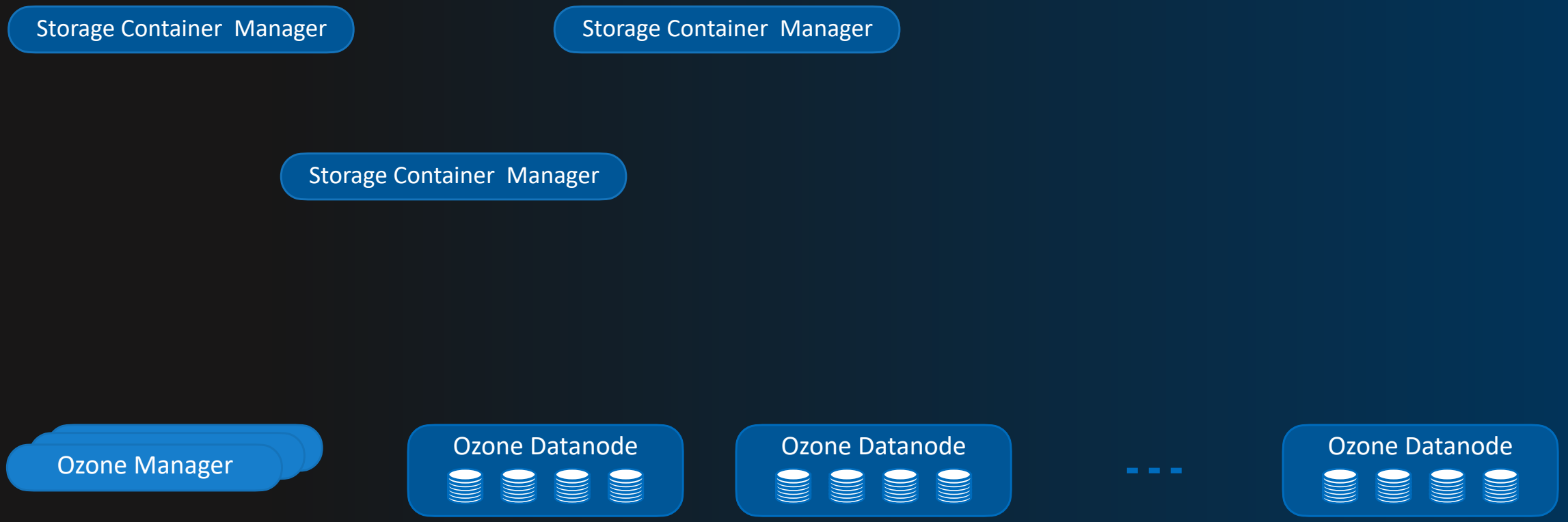
Asymmetric key signatures are expensive

Ozone tokens were signed by a 2048 bit RSA key used in PKI
RSA signature denominated the affected RPCs (~80% runtime share)

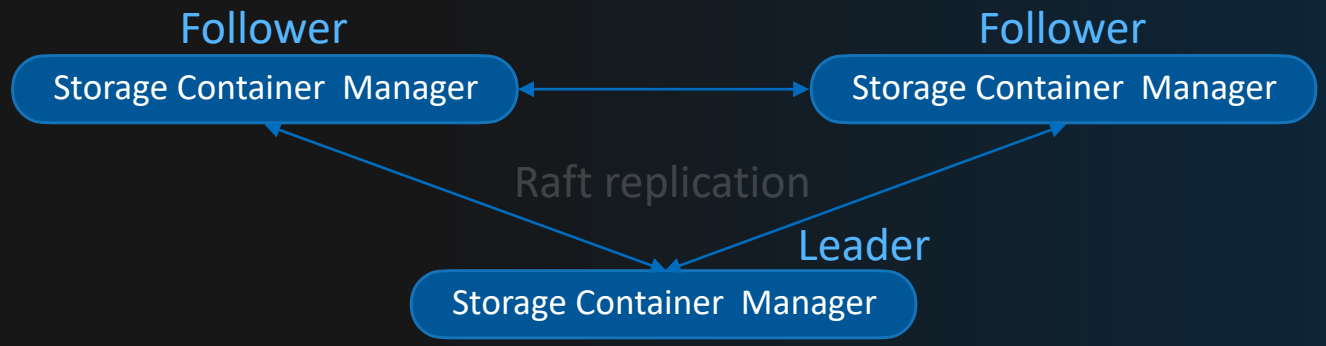
Introducing symmetric encryption helped
Signature generation is down from the 1-2ms range to 10-30 µs range

But how we can have a shared secret securely distributed?

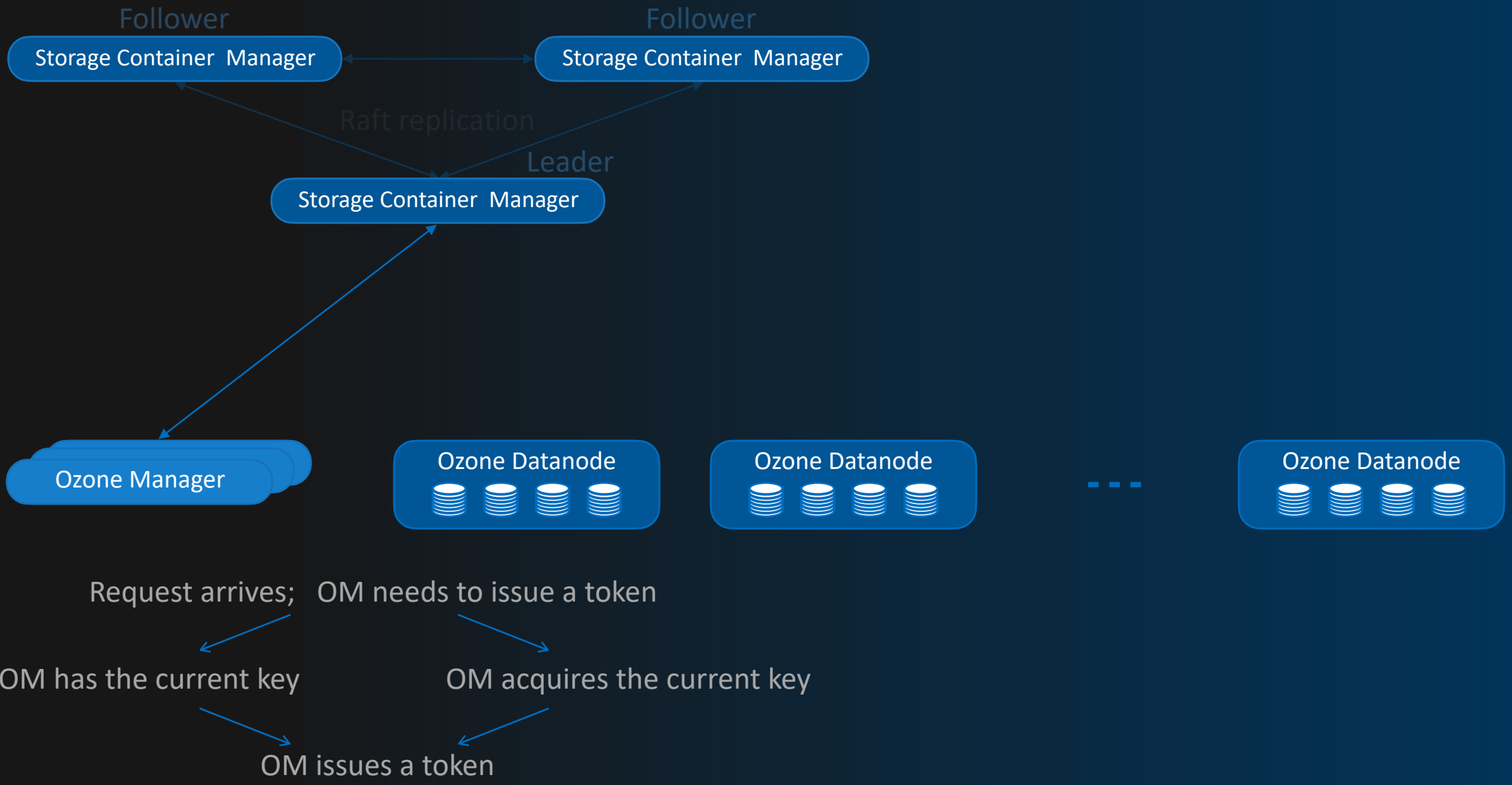
Symmetric Key Distribution



Symmetric Key Distribution

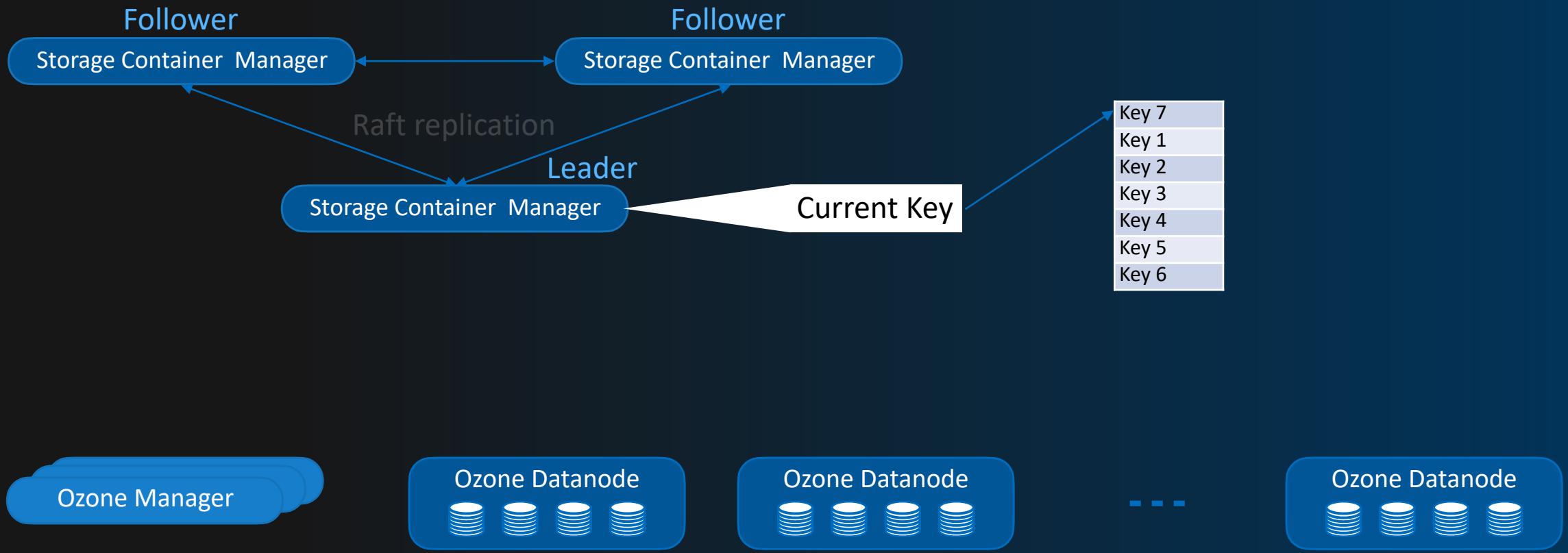


Symmetric Key Distribution



Symmetric Key Distribution

Symmetric Key Rotation



A brief project overview
Security in Apache Ozone
Tokens
Public Key Infrastructure

Internal PKI system - bootstrap

Ozone Manager

Ozone Manager

Ozone Client

Ozone Manager

Ozone Datanode

Ozone Datanode

Storage Container Manager

Storage Container Manager

Storage Container Manager

Ozone Datanode

Ozone Recon

Internal PKI system - bootstrap

Ozone Manager

Ozone Manager

Ozone Client

Ozone Manager

Ozone Datanode

Ozone Datanode

Storage Container Manager

Storage Container Manager

Primordial node - init

Storage Container Manager

Ozone Datanode

Ozone Recon

Internal PKI system - bootstrap

Ozone Manager

Ozone Manager

Ozone Client

Ozone Manager

Ozone Datanode

rootCA server

Ozone Datanode

Storage Container Manager

Storage Container Manager

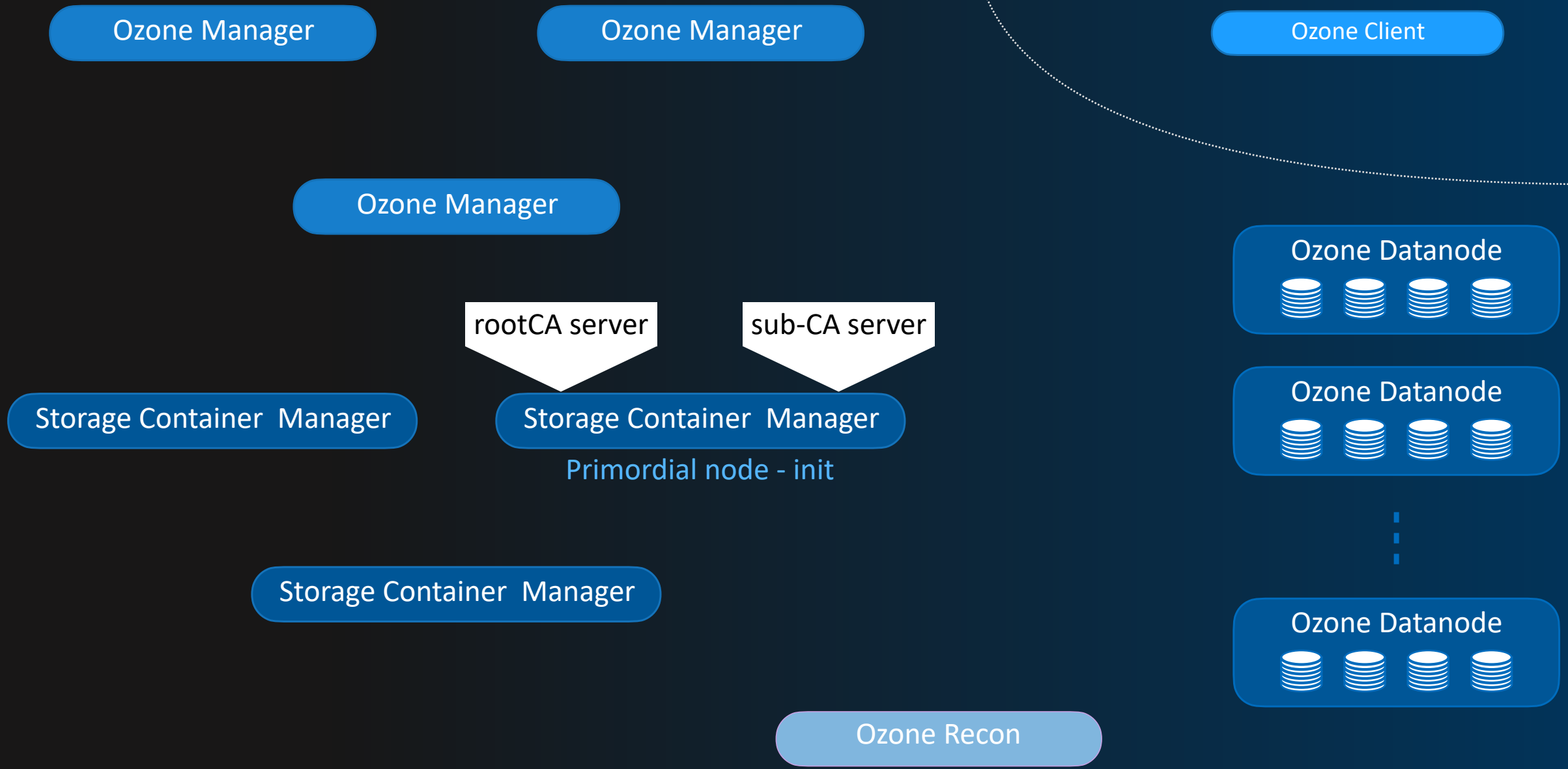
Primordial node - init

Storage Container Manager

Ozone Datanode

Ozone Recon

Internal PKI system - bootstrap



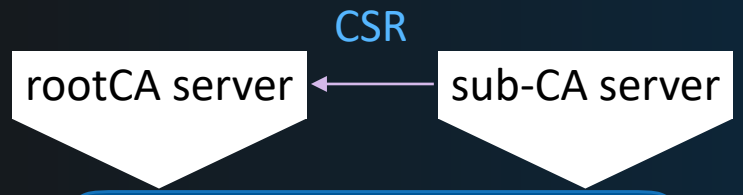
Internal PKI system - bootstrap

Ozone Manager

Ozone Manager

Ozone Client

Ozone Manager



Ozone Datanode

Ozone Datanode

Storage Container Manager

Storage Container Manager

Primordial node - init

Storage Container Manager

Ozone Datanode

Ozone Recon

Internal PKI system - bootstrap

Ozone Manager

Ozone Manager

Ozone Client

Ozone Manager

x509 certificate

rootCA server → sub-CA server

Storage Container Manager

Storage Container Manager

Primordial node - init

Storage Container Manager

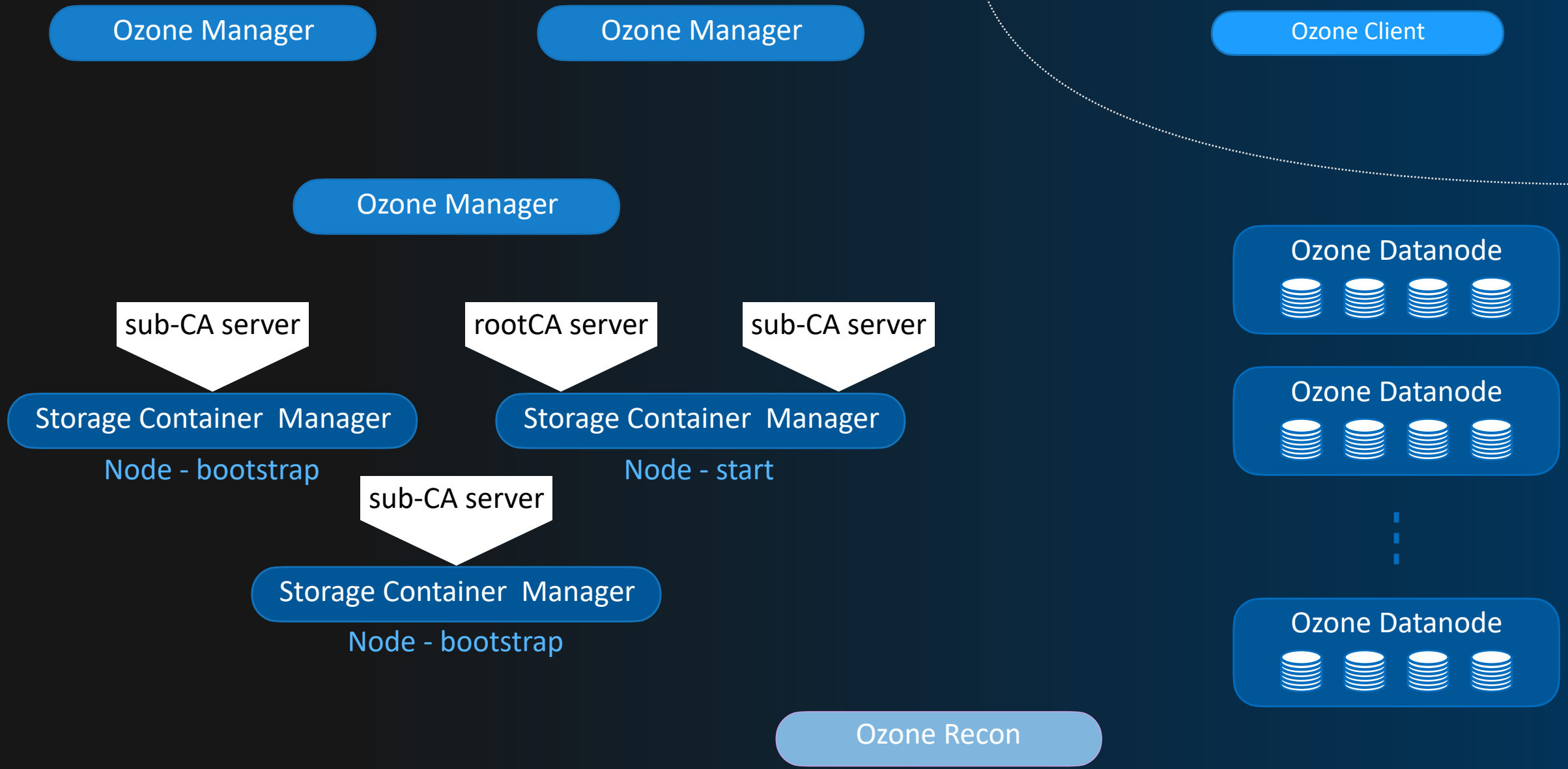
Ozone Datanode

Ozone Datanode

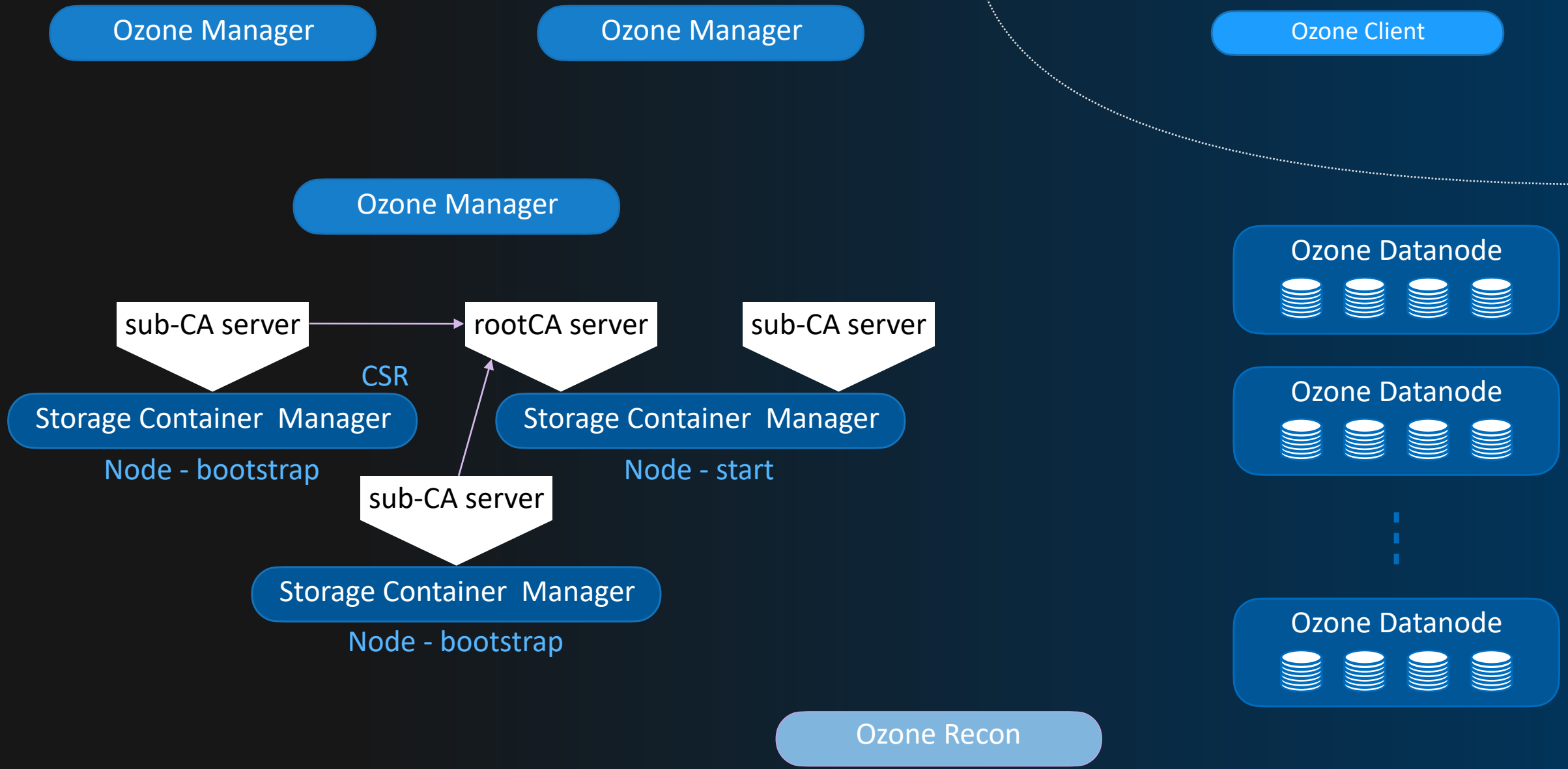
Ozone Datanode

Ozone Recon

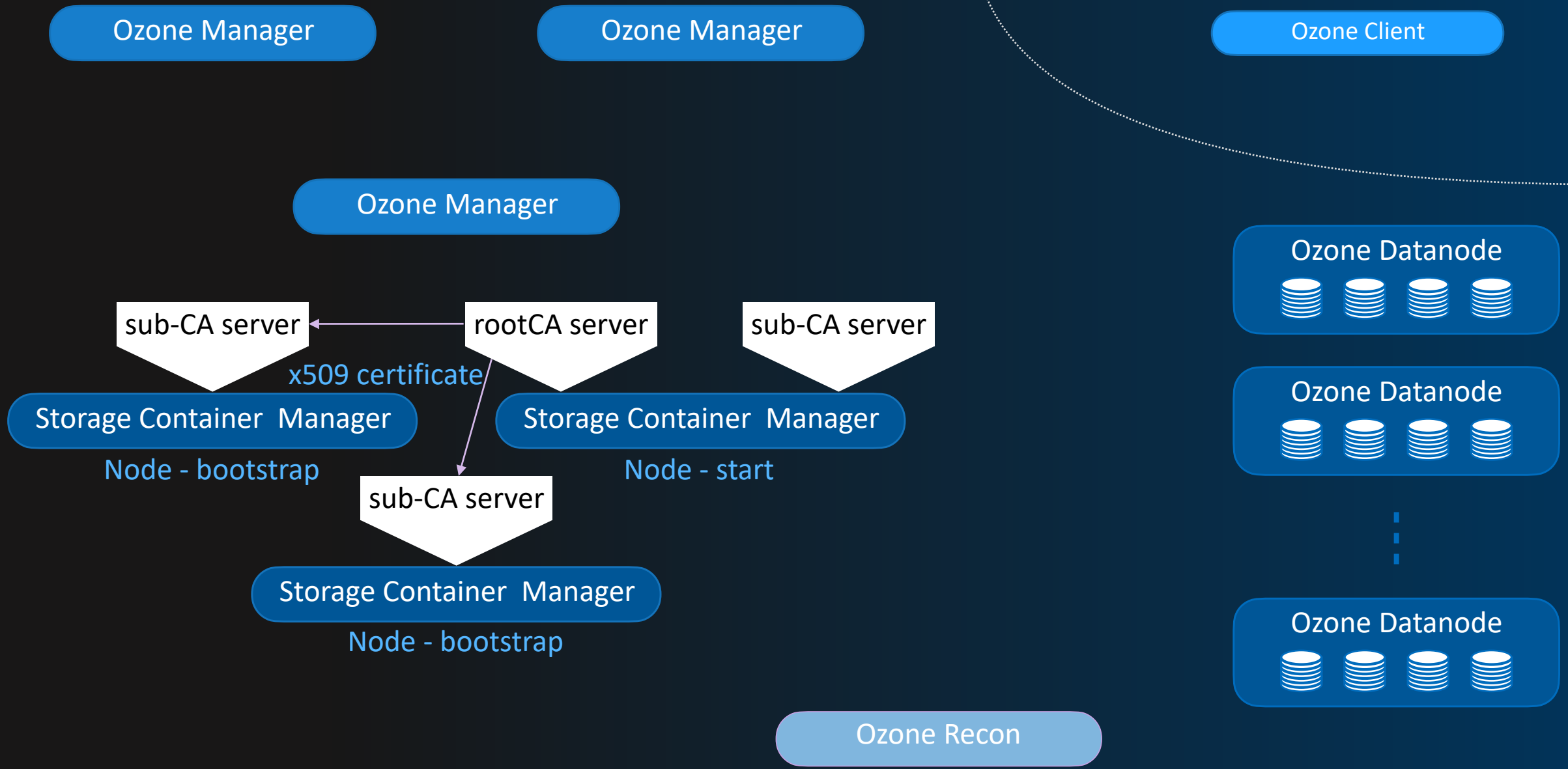
Internal PKI system - bootstrap



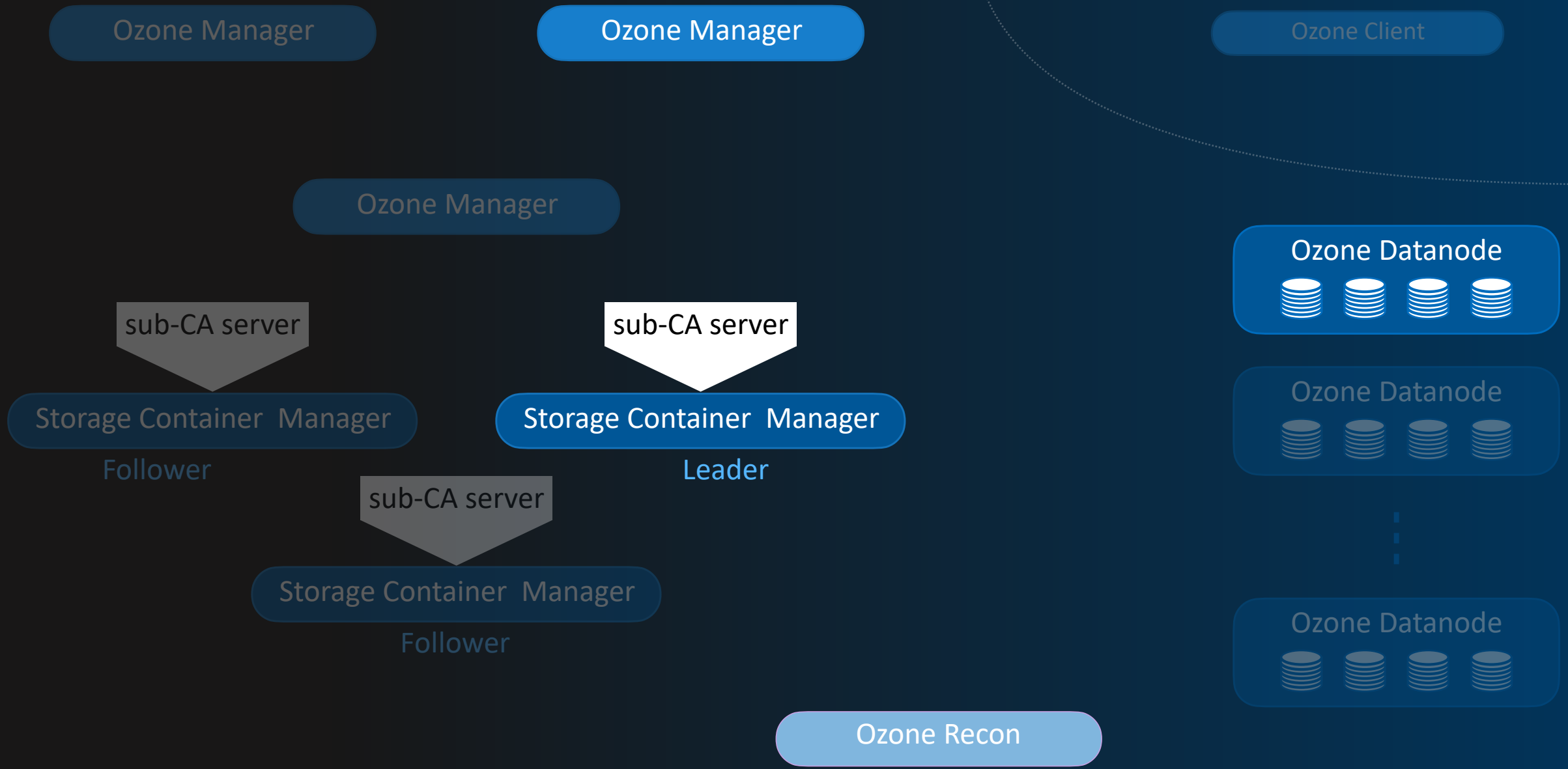
Internal PKI system - bootstrap



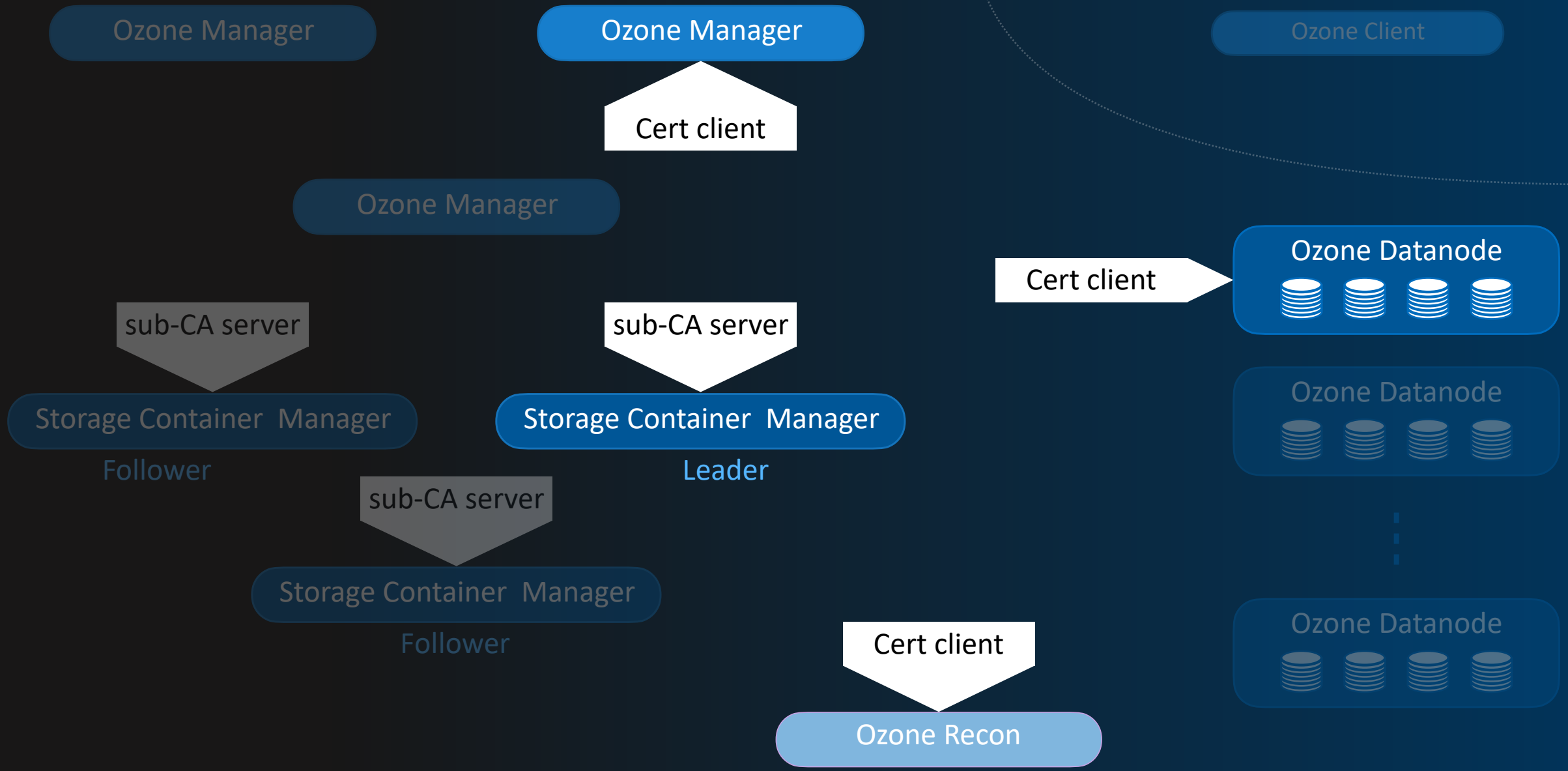
Internal PKI system - bootstrap



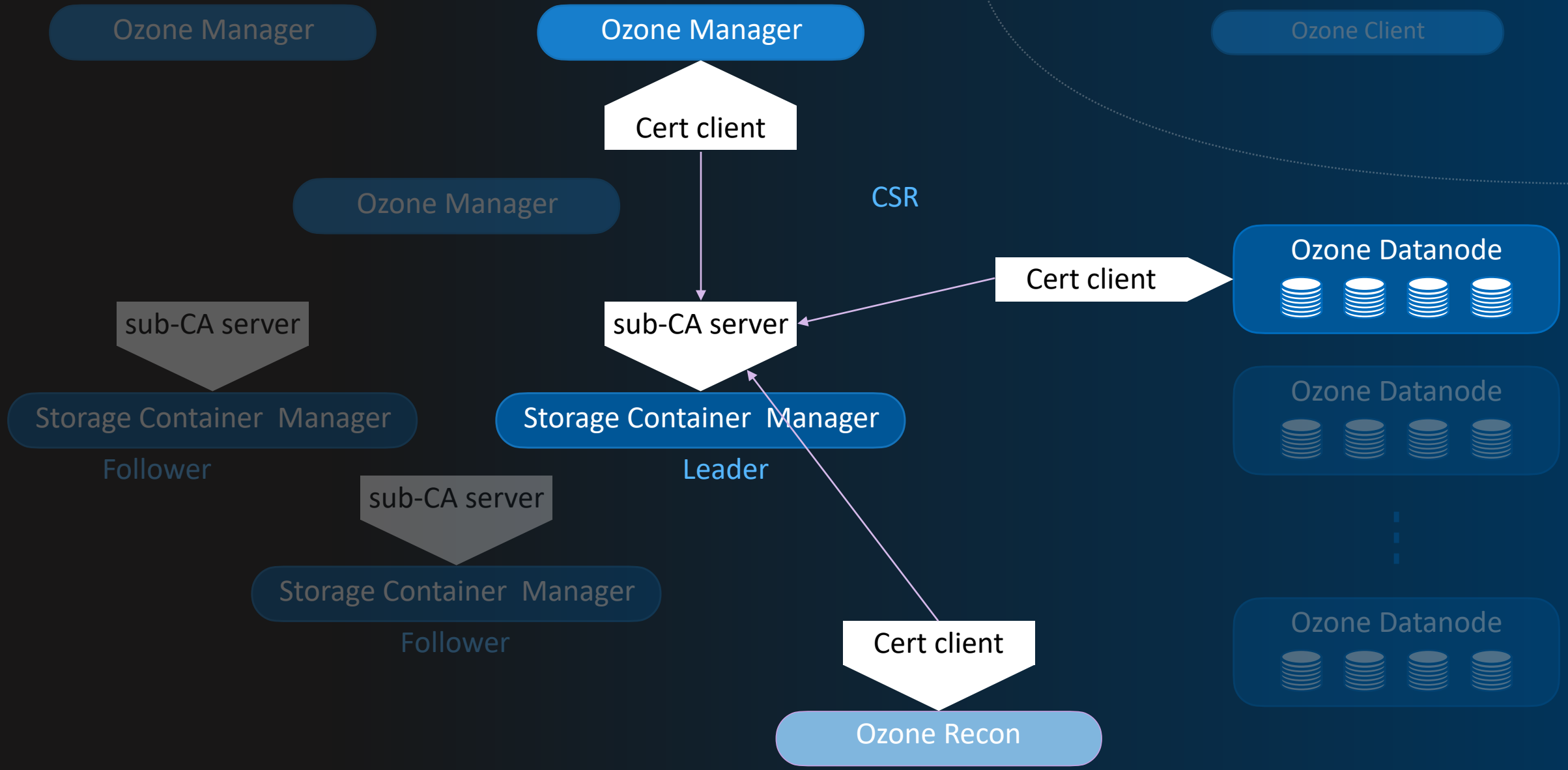
Internal PKI system - bootstrap



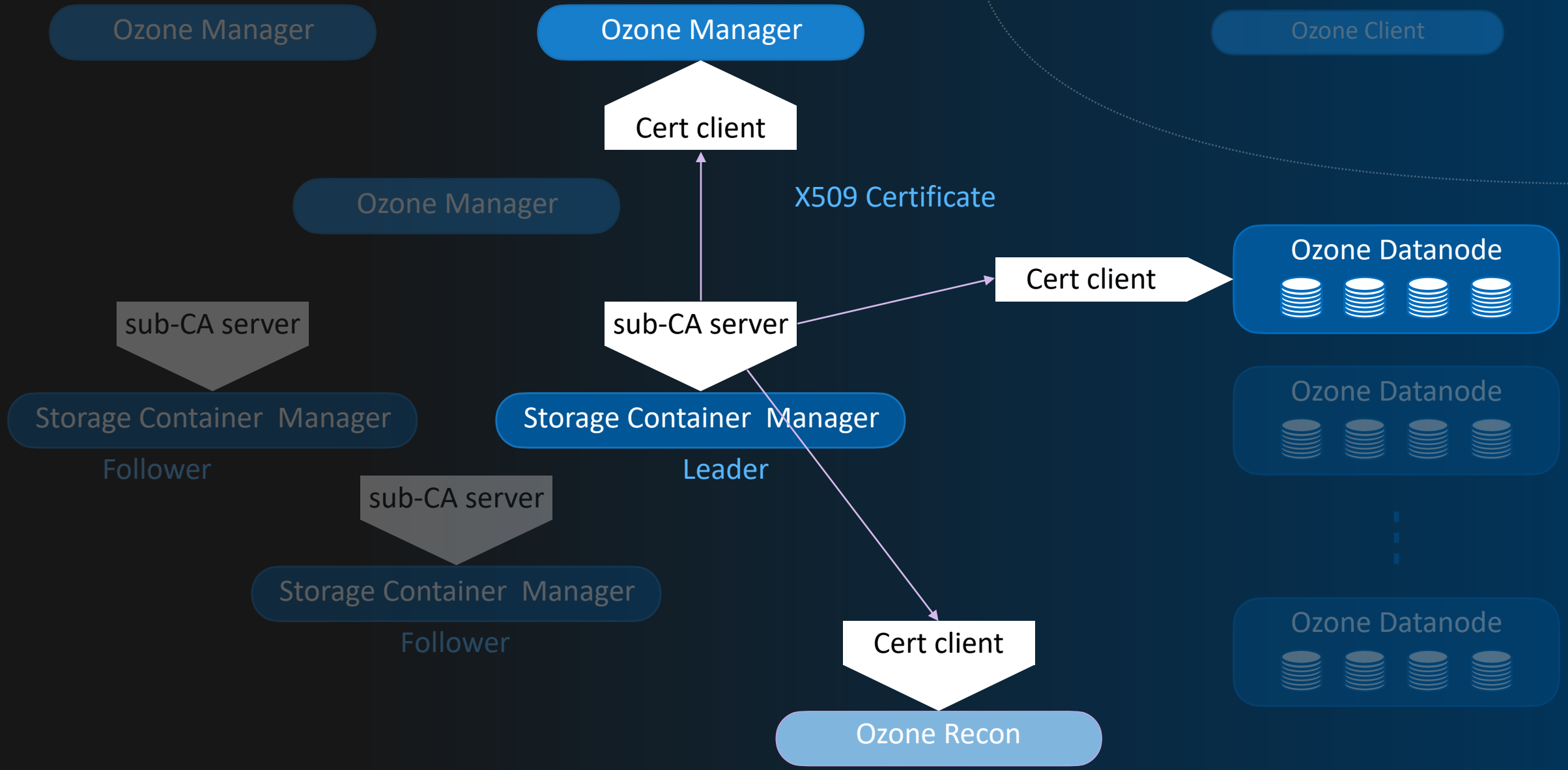
Internal PKI system - bootstrap



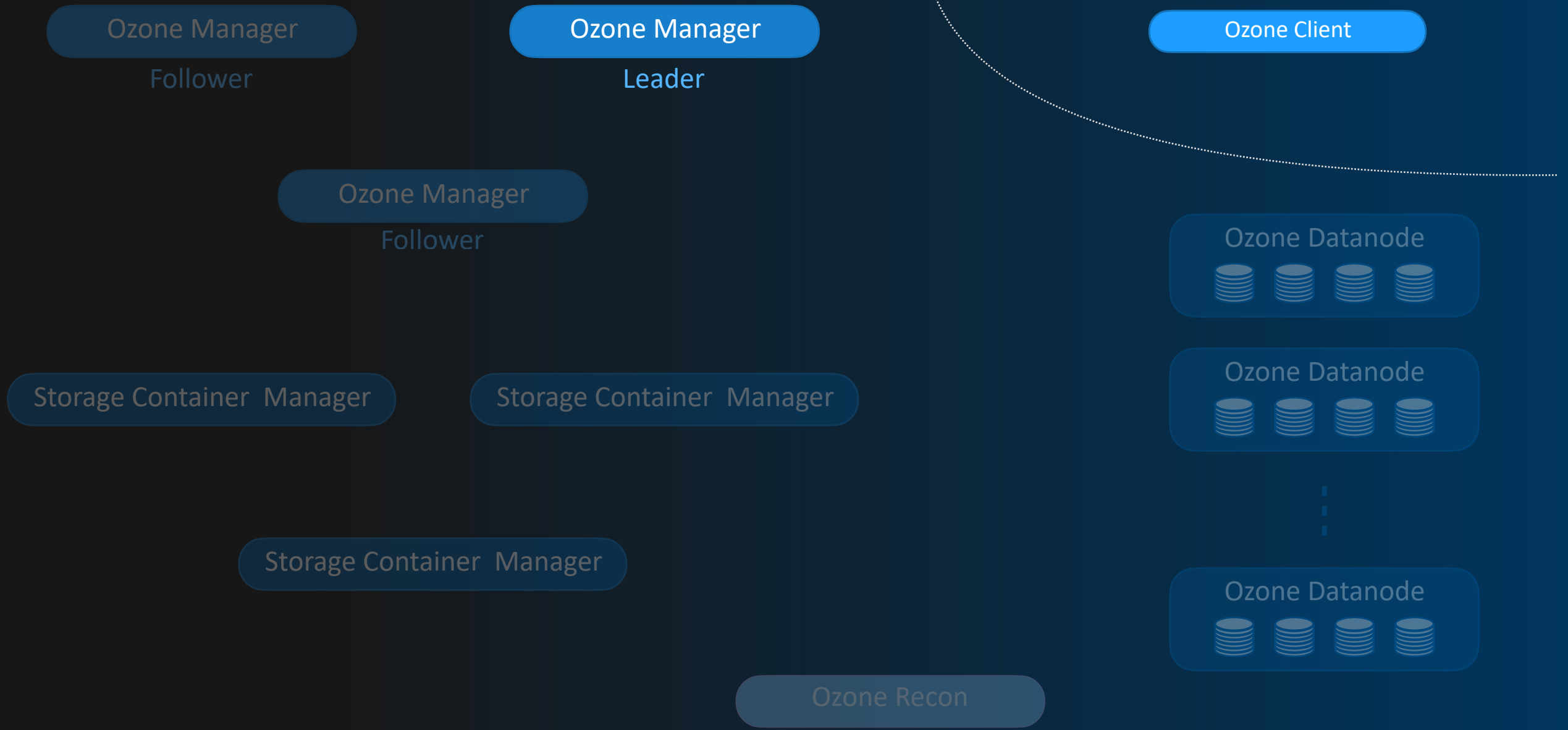
Internal PKI system - bootstrap



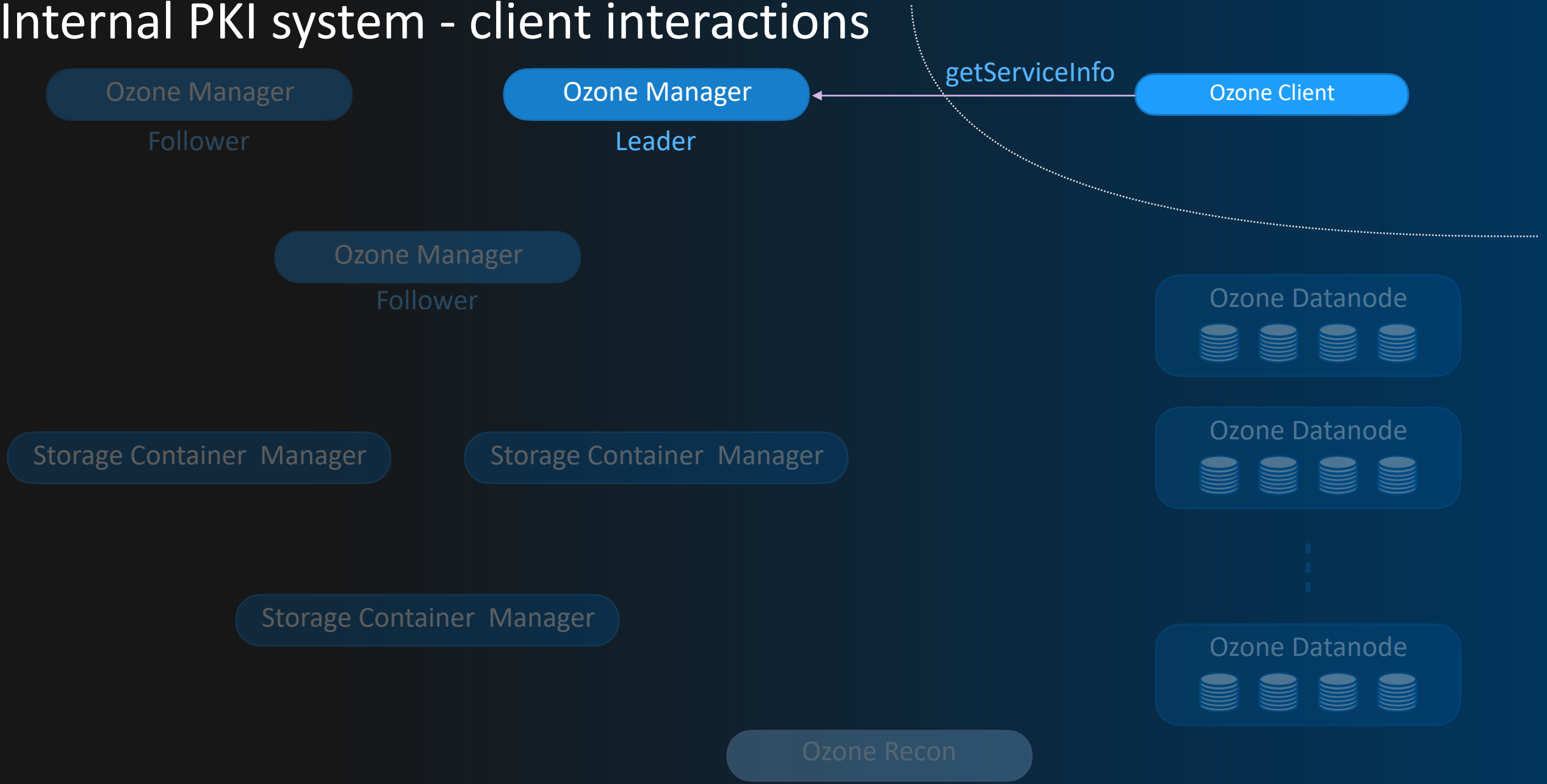
Internal PKI system - bootstrap



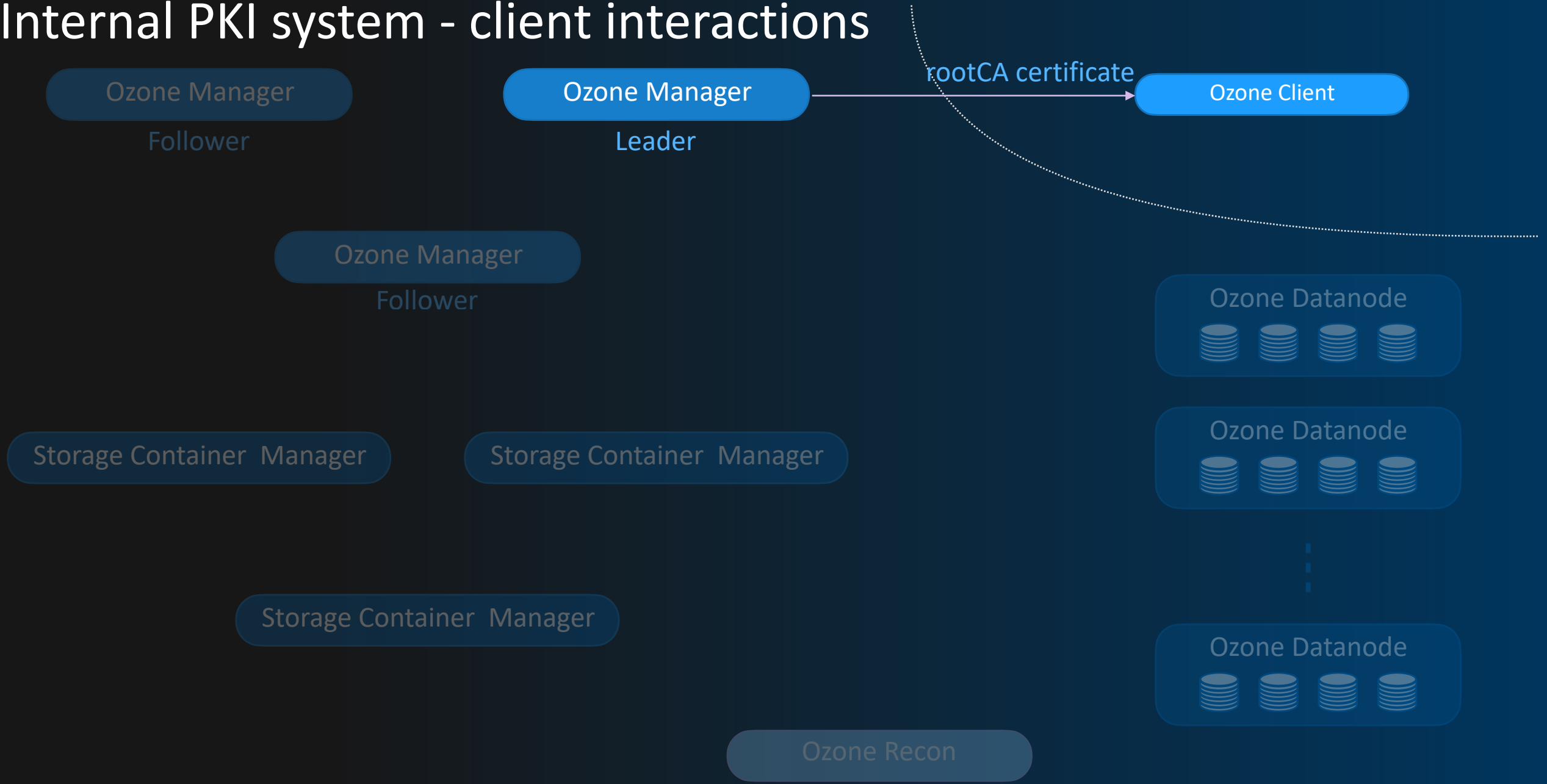
Internal PKI system - client interactions



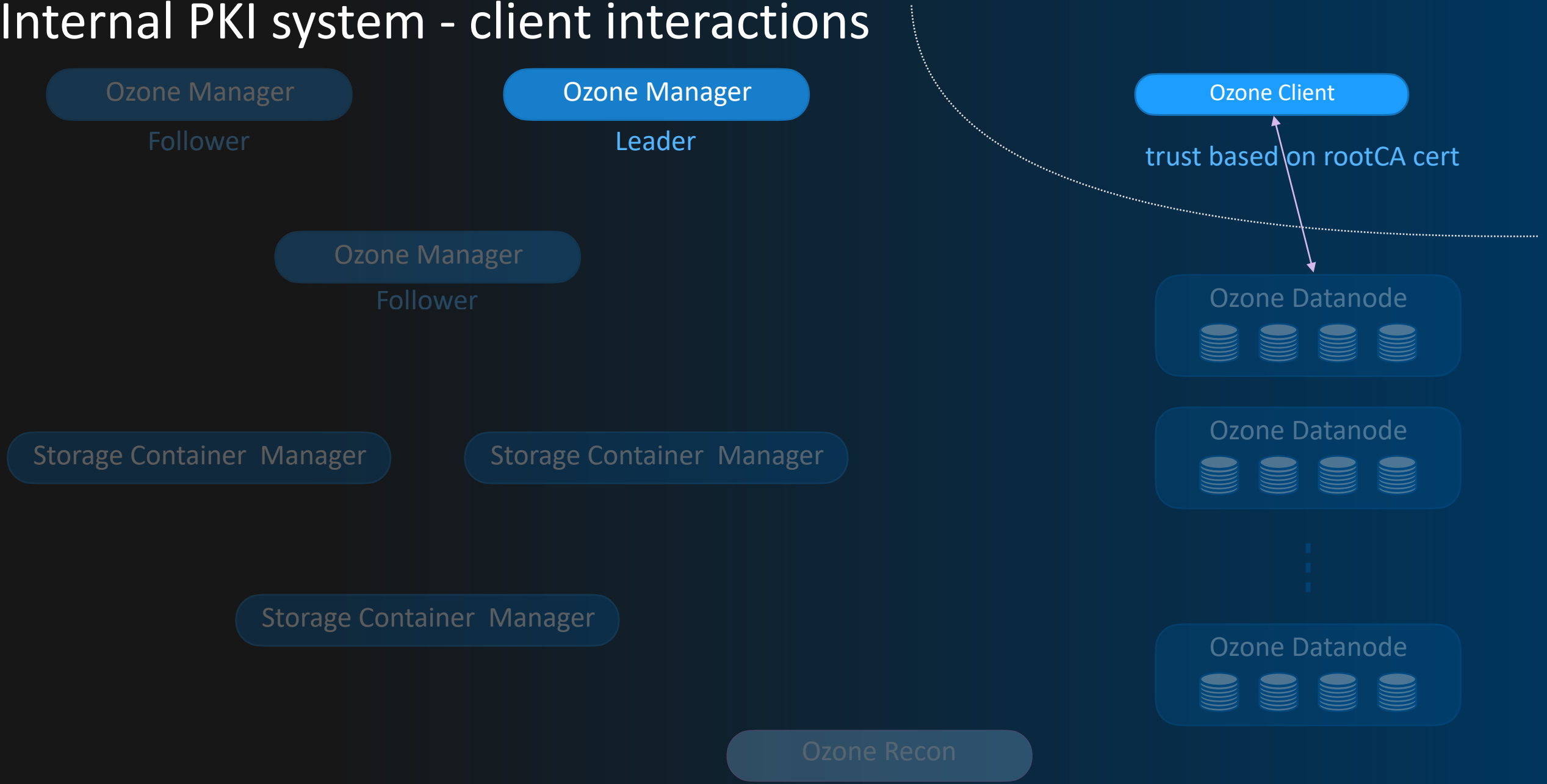
Internal PKI system - client interactions



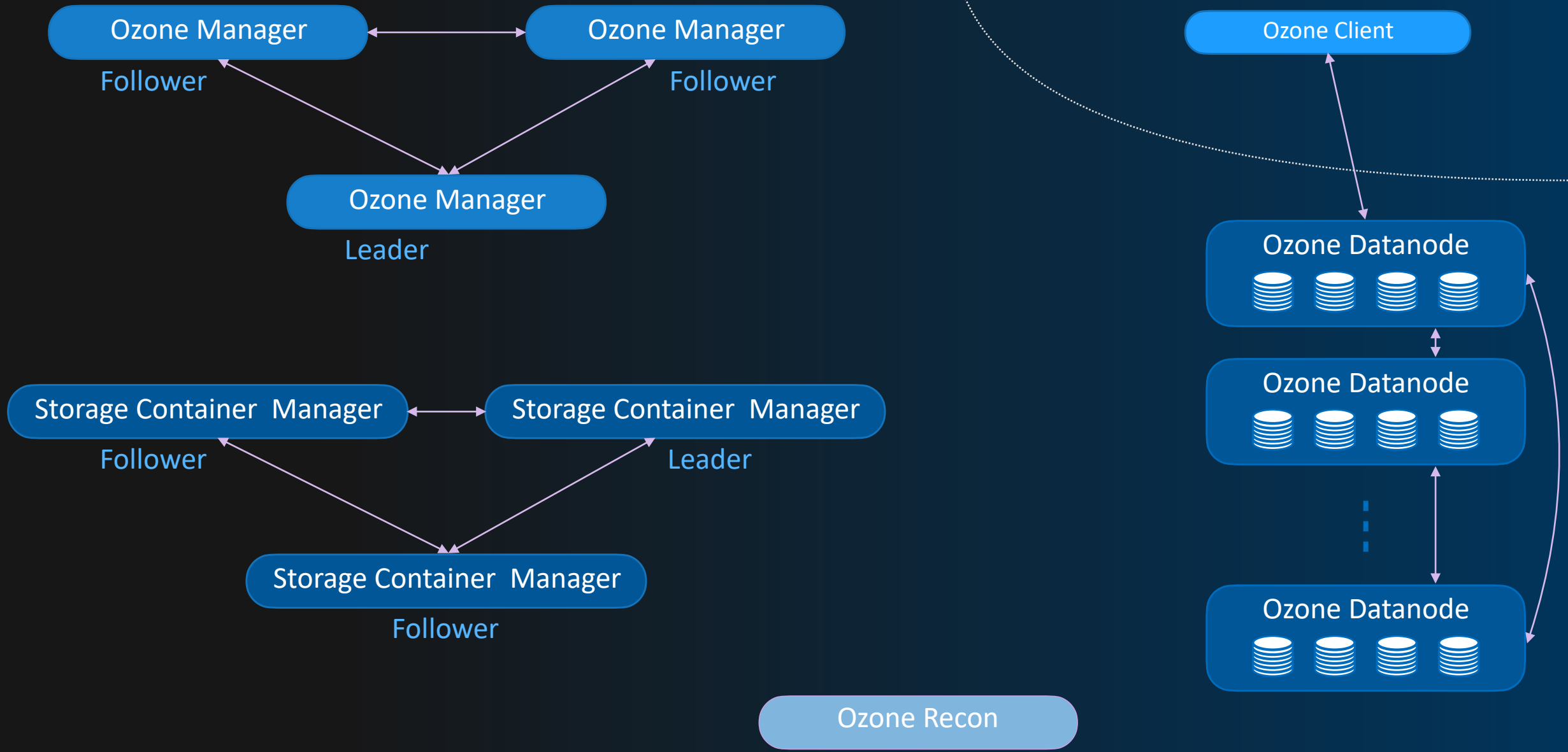
Internal PKI system - client interactions



Internal PKI system - client interactions



Communication using mTLS



Certificate Client responsibilities

Creation of key material and certificate

Rotation of certificate upon expiration

Download and store rootCA certificate

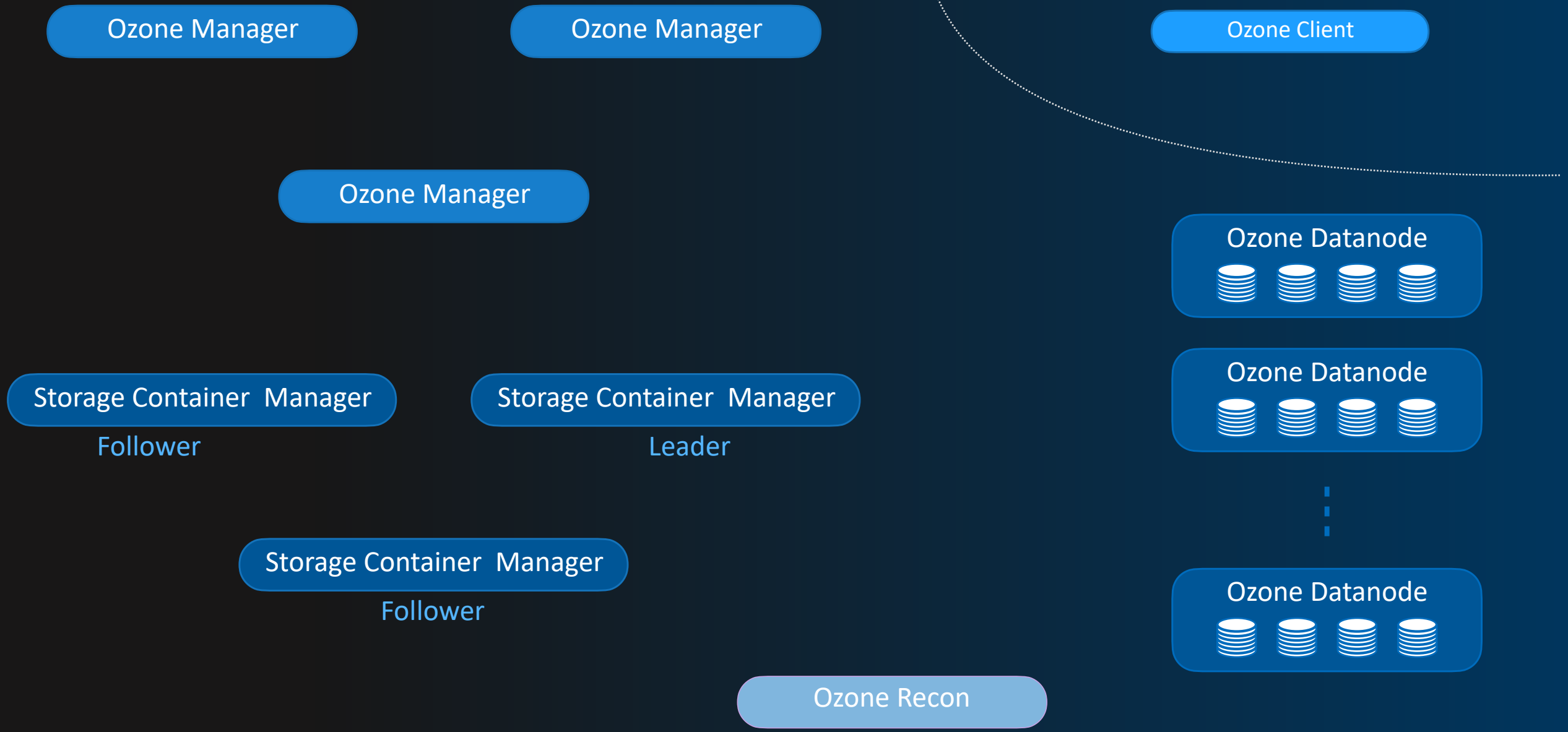
Refresh rootCA certificate upon renewal

Provide TLS setup information for connections

Initialize and refresh custom Java keystore

Initialize and refresh custom Java truststore

Rotating rootCA certificate



Rotating rootCA certificate

Ozone Manager

Ozone Manager

Ozone Client

Ozone Manager

1st step:
Leader SCM creates
a new rootCA certificate

Ozone Datanode

Storage Container Manager

Follower

Storage Container Manager

Leader

Ozone Datanode

Storage Container Manager

Follower

Ozone Datanode

Ozone Recon

Rotating rootCA certificate

Ozone Manager

Ozone Manager

Ozone Client

Ozone Manager

2nd step:
Leader SCM initiates
rootCA rotation via Raft

Ozone Datanode

Ozone Datanode

Storage Container Manager

Follower

Storage Container Manager

Leader

Storage Container Manager

Follower

Ozone Datanode

Ozone Recon

Rotating rootCA certificate

Ozone Manager

Ozone Manager

Ozone Client

Ozone Manager

3rd step:
Followers create new CSR
and get back their certs signed
by the new rootCA

Ozone Datanode

Ozone Datanode

Storage Container Manager

Follower

Storage Container Manager

Leader

Storage Container Manager

Follower

Ozone Datanode

Ozone Recon

Rotating rootCA certificate

Ozone Manager

Ozone Manager

Ozone Client

Ozone Manager

4th step:
Followers ack the rootCA rotation

Ozone Datanode

Ozone Datanode

Ozone Datanode

Storage Container Manager

Follower

Storage Container Manager

Leader

Storage Container Manager

Follower

Ozone Recon

Rotating rootCA certificate

Ozone Manager

Ozone Manager

Ozone Client

Ozone Manager

5th step:
Leader commits the rotation to Raft.
New rootCA certificate is available to download for any clients

Ozone Datanode

Ozone Datanode

Storage Container Manager

Follower

Storage Container Manager

Leader

Storage Container Manager

Follower

Ozone Datanode

Ozone Recon

Rotating rootCA certificate

Ozone Manager

Ozone Manager

Ozone Client

Ozone Manager

Ozone Datanode

Storage Container Manager

Follower

Storage Container Manager

Leader

Ozone Datanode

Storage Container Manager

Follower

Ozone Datanode

Ozone Recon

Rotating rootCA certificate

Ozone Manager

Ozone Manager

Ozone Client

Ozone Manager

6th step:
CertificateClients poll for new rootCA certificate

Ozone Datanode

Ozone Datanode

Storage Container Manager

Follower

Storage Container Manager

Leader

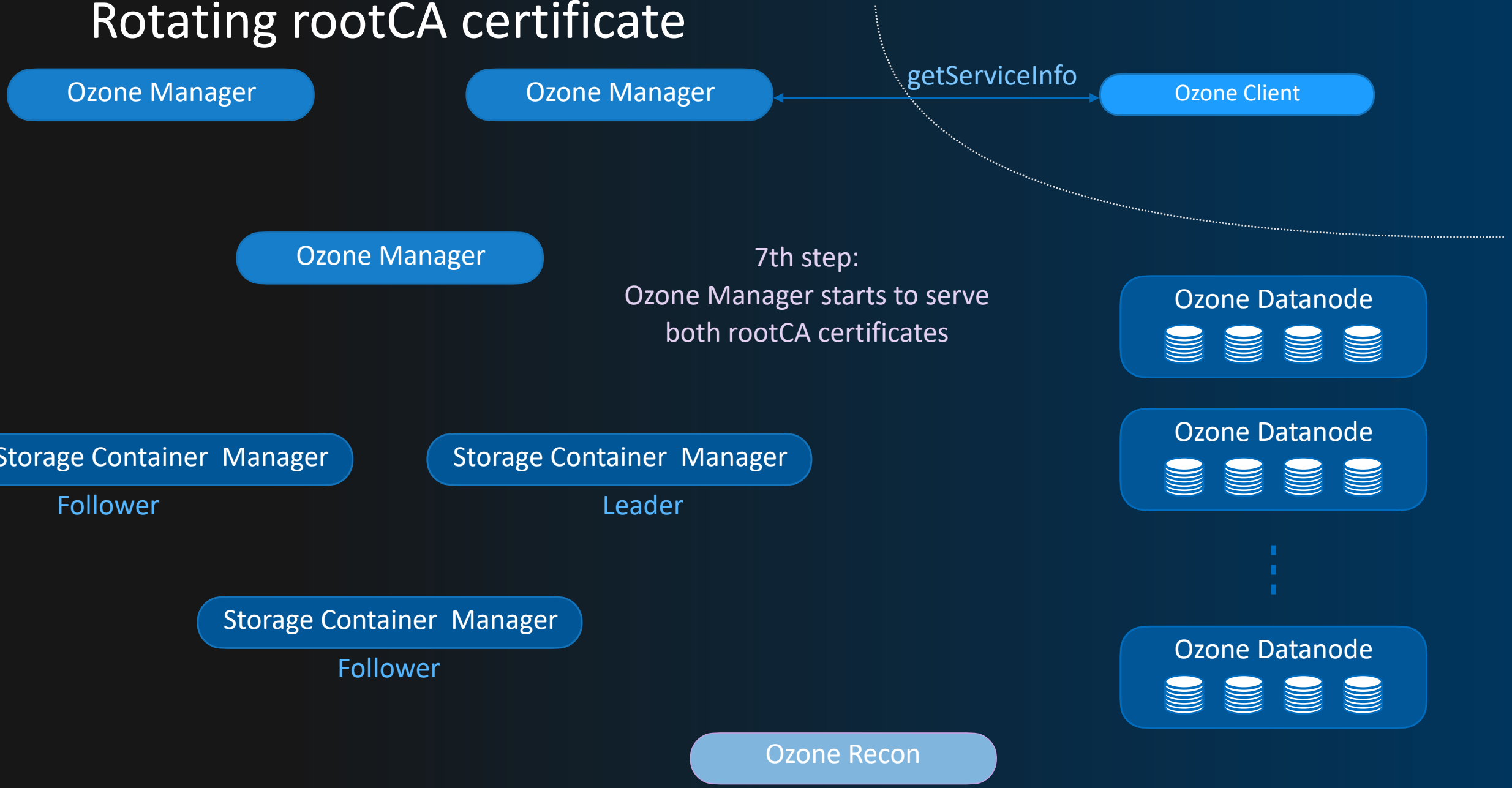
Storage Container Manager

Follower

Ozone Datanode

Ozone Recon

Rotating rootCA certificate



Rotating rootCA certificate

Future

Streamlined certificate revocation

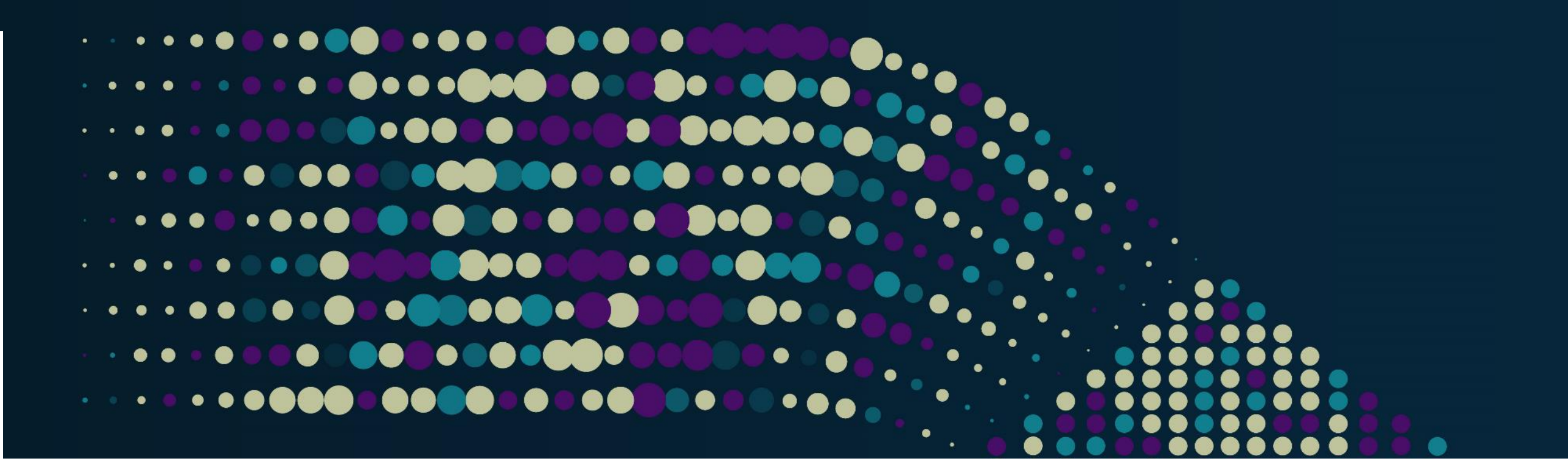
Prescriptive documentation on secure Ozone setup

Pluggable storage mechanism for keys and certificates

Simple tools to ensure the "right to be forgotten"

Security implications of clusters stretching over multiple data centers

Questions?



Please take a moment to rate this session.

Your feedback is important to us.