



**Resilient and Fast Persistent Container Storage  
Leveraging Linux's Storage Functionalities**

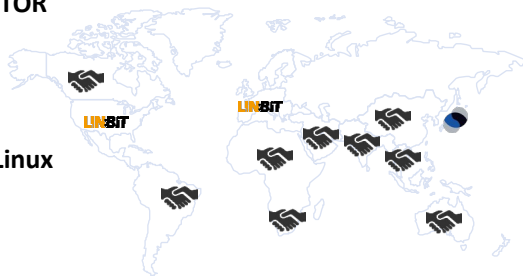
Philipp Reisner, CEO LINBIT

# Leading Open Source OS based SDS



## COMPANY OVERVIEW

- Developer of DRBD and LINSTOR
- 100% founder owned
- Offices in Europe and US
- Team of highly experienced Linux experts
- Exclusivity Japan: SIOS

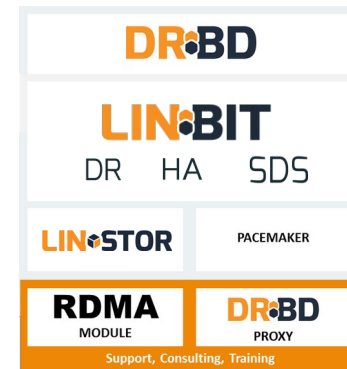


## REFERENCES



## PRODUCT OVERVIEW

- Leading Open Source Block Storage (included in Linux Kernel (v2.6.33))
- Open Source DRBD supported by proprietary LINBIT products / services
- OpenStack with DRBD Cinder driver
- Kubernetes Driver
- Install base of >2 million



## SOLUTIONS

### LINBIT SDS

Since 2016

Perfectly suited for SSD/NVMe high performance storage

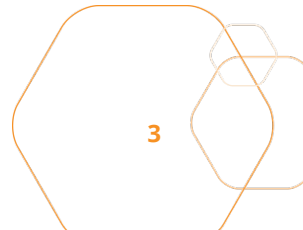
### DRBD HA, DRBD DR

Market leading solutions since 2001, over 600 customers

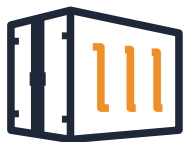
Ideally suited to power HA and DR in OEM appliances (Cisco, IBM, Oracle)

# **LINBIT SDS**

**When? Why? What?**



# When is LINBIT SDS a fit?



## Persistent Volumes

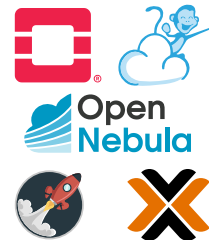
...for Containers

- Kubernetes
- Nomad
- Docker



## Virtualization

- OpenStack
- CloudStack
- OpenNebula
- XCP-ng
- Proxmox



## Transaction Processing

- Oracle DB
- PostgreSQL
- MariaDB
- Message queuing systems



## Analytic Processing

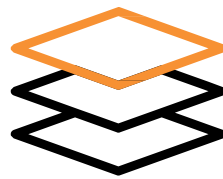
- DB2 Warehouse
- And similar read intensive workloads

# Why is LINBIT SDS so fast?



## In Kernel data-path

- Reduce number of context switches
- Saving on CPU/memory resources
- Minimal latency for block-IO operations
- Optional load-balancing for READs



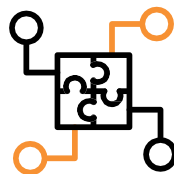
## Layout at volume allocation

- All participating machines have full replicas, which machines participate determined when creating a volume.
- Be faster at IO submission time
- Saving on CPU/memory



## Build on existing components

- DRBD, LVM, ZFS, LUKS, VDO, ...
- Help day2 operations by leveraging on the operation teams prior knowledge
- Build on the shoulders of giants



## Hyper-Converged

Very well suitable for hyper-converged deployment

- Reduced network load for reads
- Reduces latency
- LINBIT SDS' Low resource consumption leaves most of CPU and memory for workload. About 0.5% of a single core are consumed by DRBD under heavier IO load (measured with an analytics DB)

# What is LINBIT SDS doing?



## Storage Allocation

- 3 to 1000s of nodes
  - Multiple tiers
  - Multi tenancy
  - Complex policies
- Chassis - rack - room



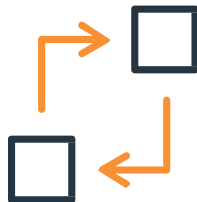
## Data Replication

- Persistence & availability
- Sync / async
- 2,3 or more replicas
- Consistency groups
- Quorum



## Network

- Multiple NICs per server
- Multiple networks
- RDMA
- TCP



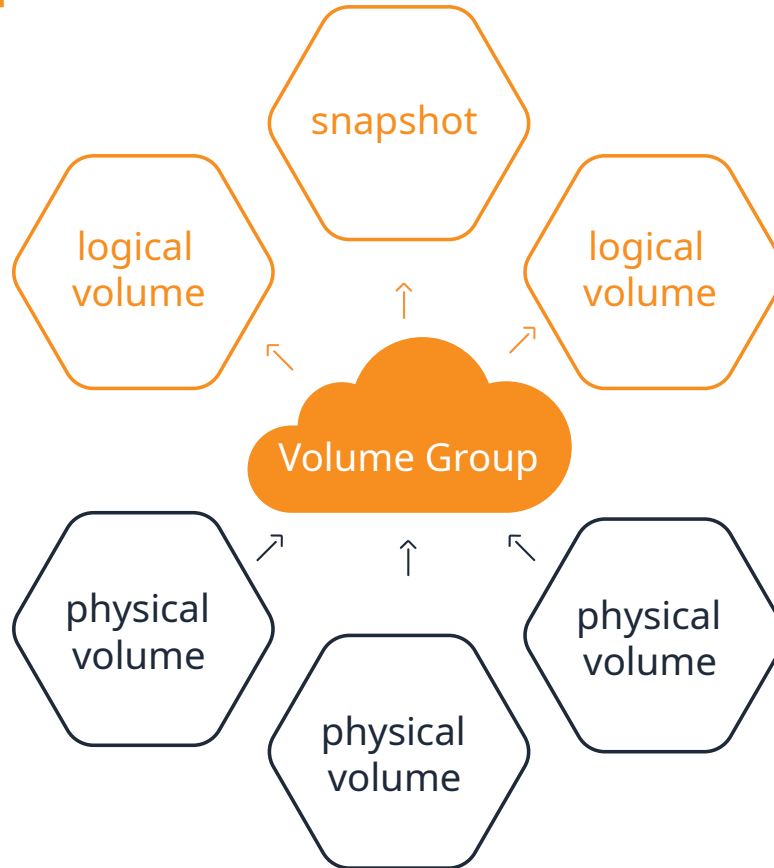
## Business continuity

- Continuous data protection
  - Multiple sites
  - Backups
- SSD - disk - cloud

# Linux Storage Gems

LVM, RAID, SSD cache tiers, deduplication, targets & initiators

# Linux's LVM



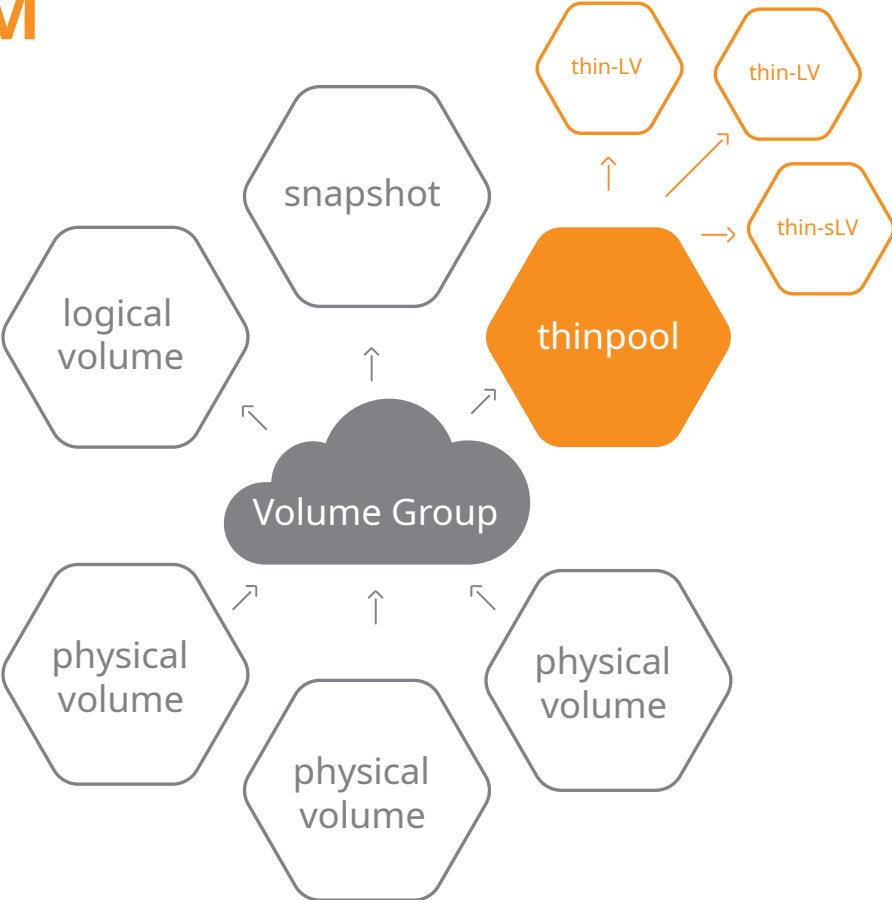


# Linux's LVM

- based on device mapper
- original objects
  - PVs, VGs, LVs, snapshots
  - LVs can scatter over PVs in multiple segments
- thinlv
  - thinpools = LVs
  - thin LVs live in thinpools
  - multiple snapshots became efficient!

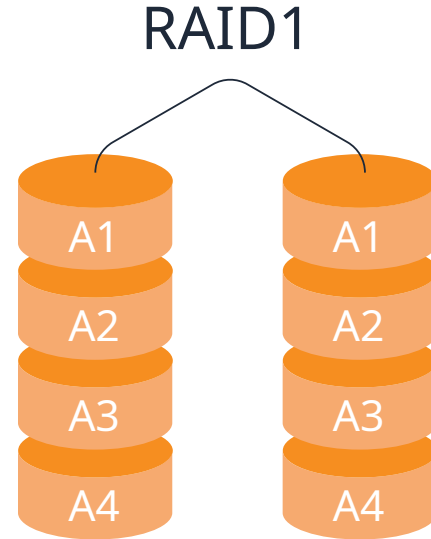


# Linux's LVM



# Linux's RAID

- original MD code
  - `mdadm` command
  - Raid Levels: 0,1,4,5,6,10
- now available in LVM as well
  - device mapper interface for MD code
  - do not call it 'dmraid'; that is software for hardware fake-raid
  - `lvcreate --type raid6 --size 100G VG_name`



# Linux's DeDupe

- Virtual Data Optimizer (VDO) since RHEL 7.5
  - Red hat acquired Permabit and is GPLing VDO
- Linux upstreaming is in preparation
- in-line data deduplication
- kernel part is a device mapper module
- indexing service runs in user-space
- async or synchronous writeback
- recommended to be used below LVM

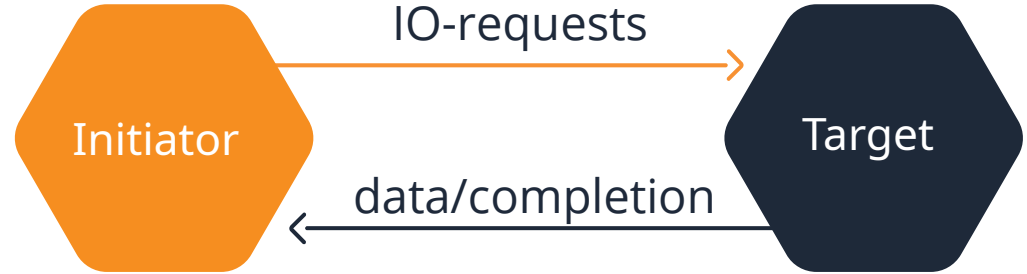
# SSD cache for HDD

- dm-cache
  - device mapper module
  - accessible via LVM tools
- bcache
  - generic Linux block device
  - slightly ahead in the performance game
- dm-write-cache
  - for combining PMEM & NVMe drives



# Linux's targets & initiators

- Open-ISCSI initiator
- letd, STGT, SCST
  - mostly historical
- **LIO**
  - iSCSI, iSER, SRP, FC, FCoE
  - SCSI pass through, block IO, file IO, user-specific-IO
- NVMe-OF & NVMe/TCP
  - target & initiator



# ZFS on Linux

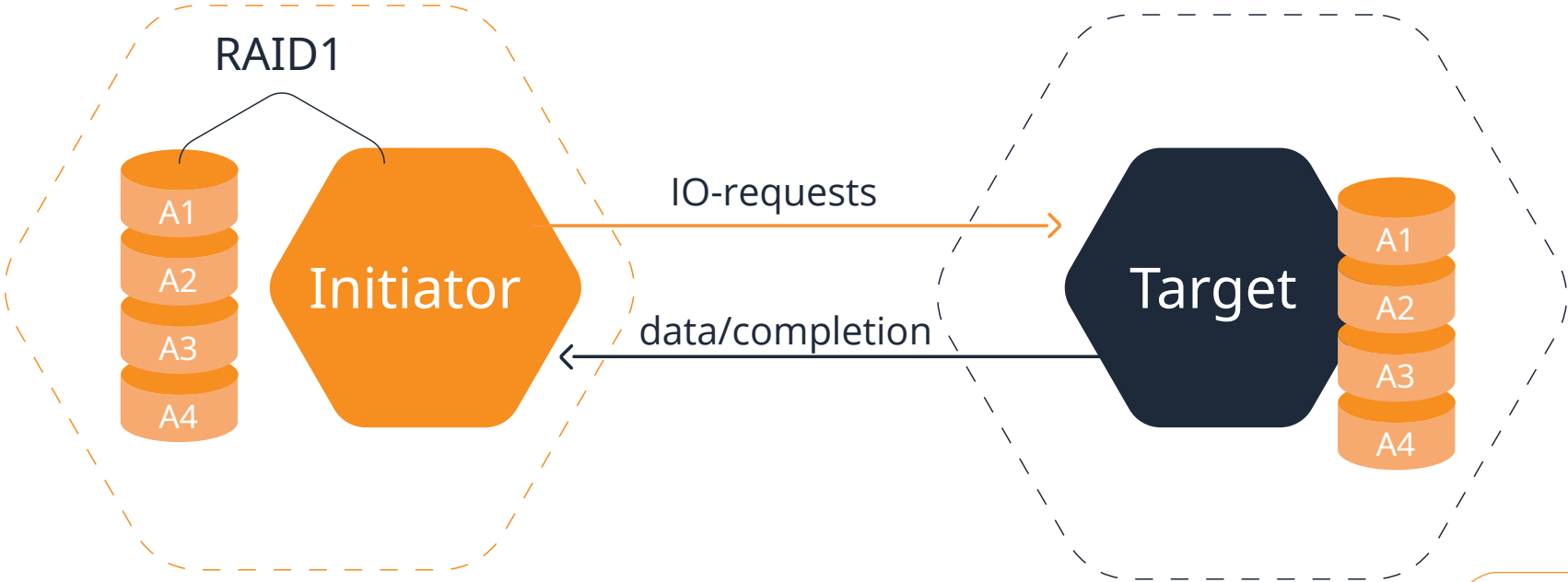
- Ubuntu eco-system only
- has its own
  - logic volume manager (zVols)
  - thin provisioning
  - RAID (RAIDz)
  - caching for SSDs (ZIL, SLOG)
  - and a file system!

**DR:BD**

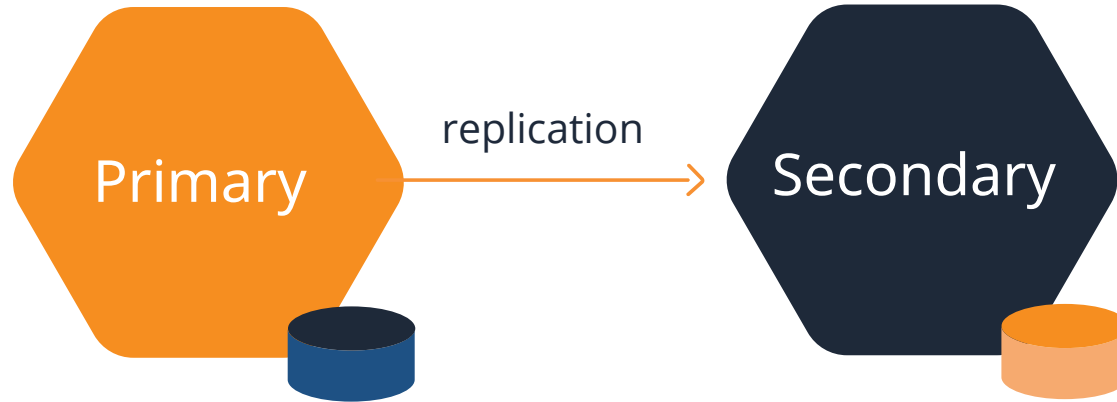
Put in simplest form



# DRBD - think of it as ...

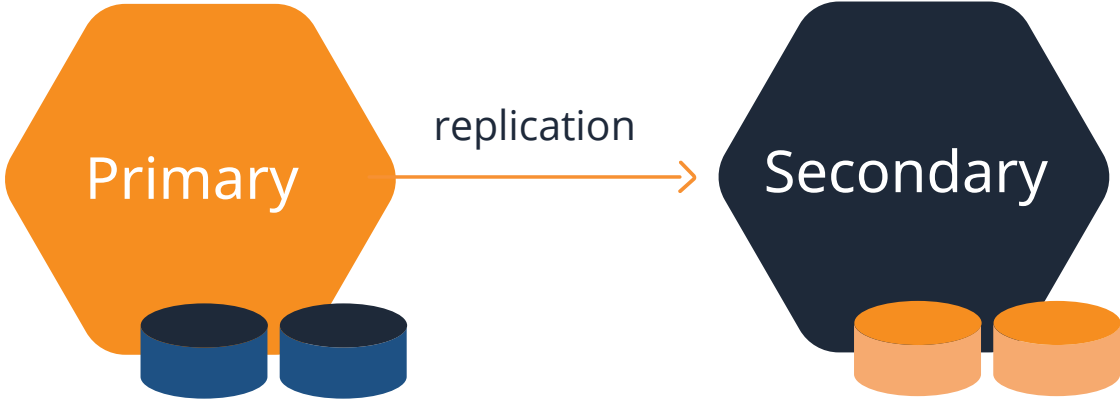


# DRBD Roles: Primary & Secondary



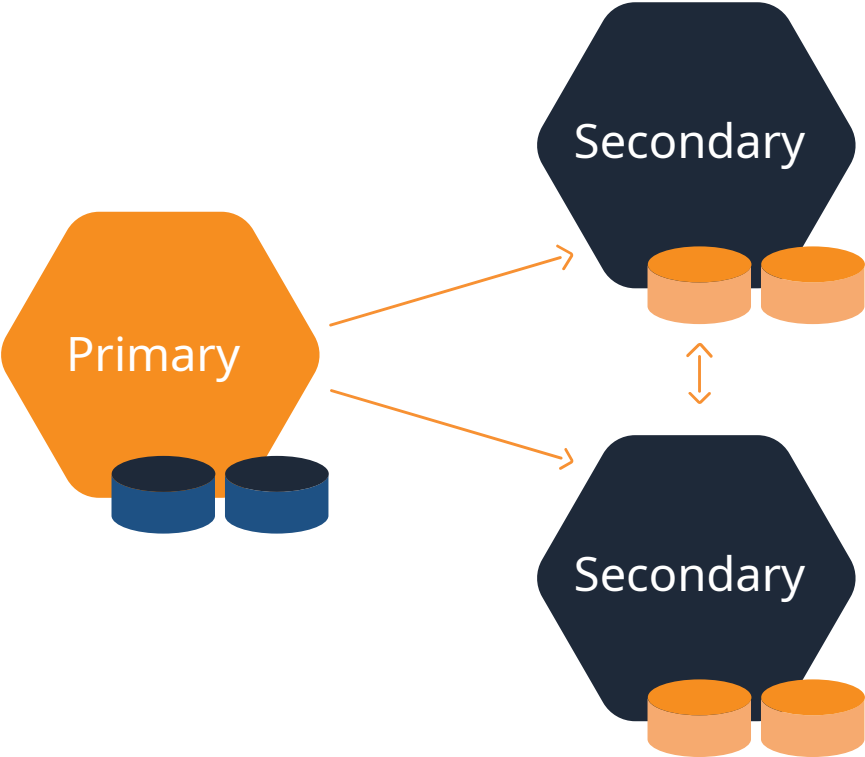
# DRBD - multiple Volumes

- consistency group



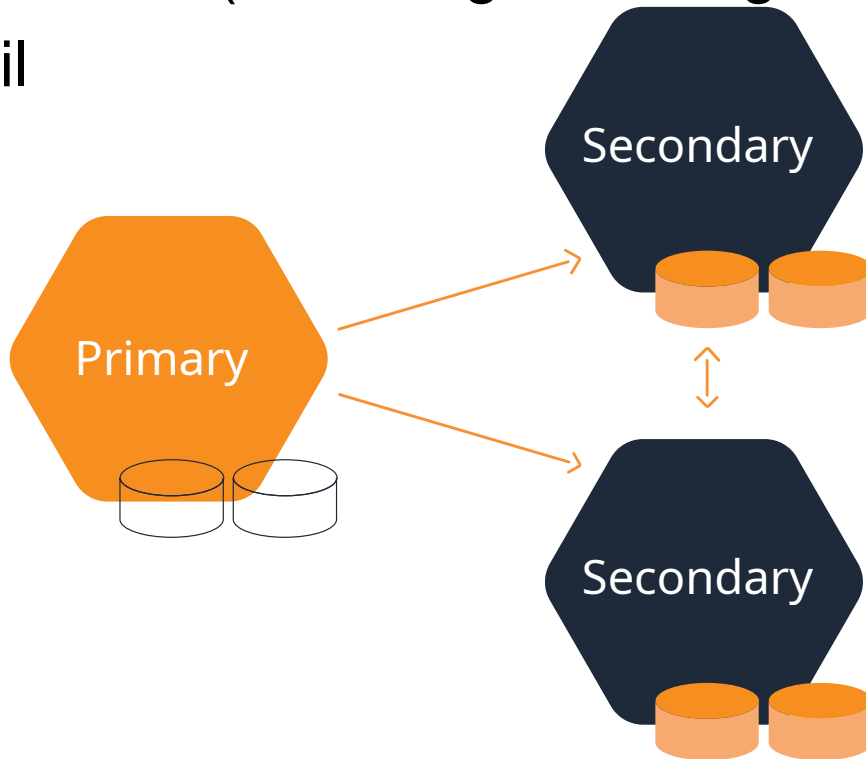
# DRBD - up to 32 replicas

- each may be synchronous or async



# DRBD - Diskless nodes

- intentional diskless (no change tracking bitmap)
- disks can fail



## DRBD - more about

- a node knows the version of the data it exposes
- automatic partial resync after connection outage
- checksum-based verify & resync
- split brain detection & resolution policies
- fencing
- quorum
- multiple resources per node possible (1000s)
- dual Primary for live migration of VMs only!

- Recent
  - meta-data on PMEM/NVDIMMS
  - improved, fine-grained locking for parallel workloads
  - Eurostars grant: DRBD4Cloud
  - started DRBD-9.1
- ROADMAP
  - performance optimizations
  - replace „stacking”
  - production release of WinDRBD

# **LIN:STOR**

**The combination is more than the sum of its parts**



# LINSTOR - goals

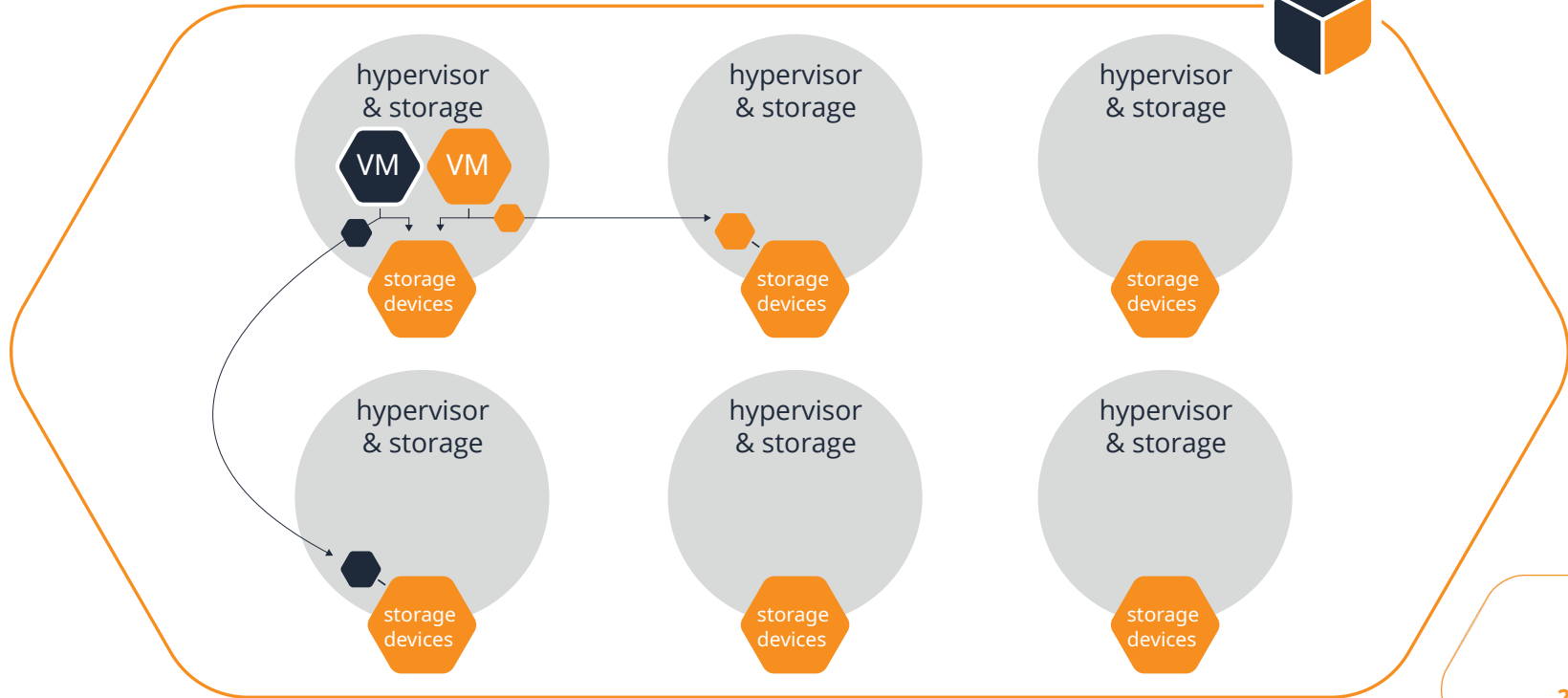
- storage build from generic Linux nodes
- for SDS consumers (K8s, OpenStack, OpenNebula)
- building on existing Linux storage components
- multiple tenants possible
- deployment architectures
  - distinct storage nodes
  - hyperconverged with hypervisors / container hosts
- LVM, thin LVM or ZFS for volume management (stratis later)
- **Open Source, GPL**

# **LIN** **STOR**

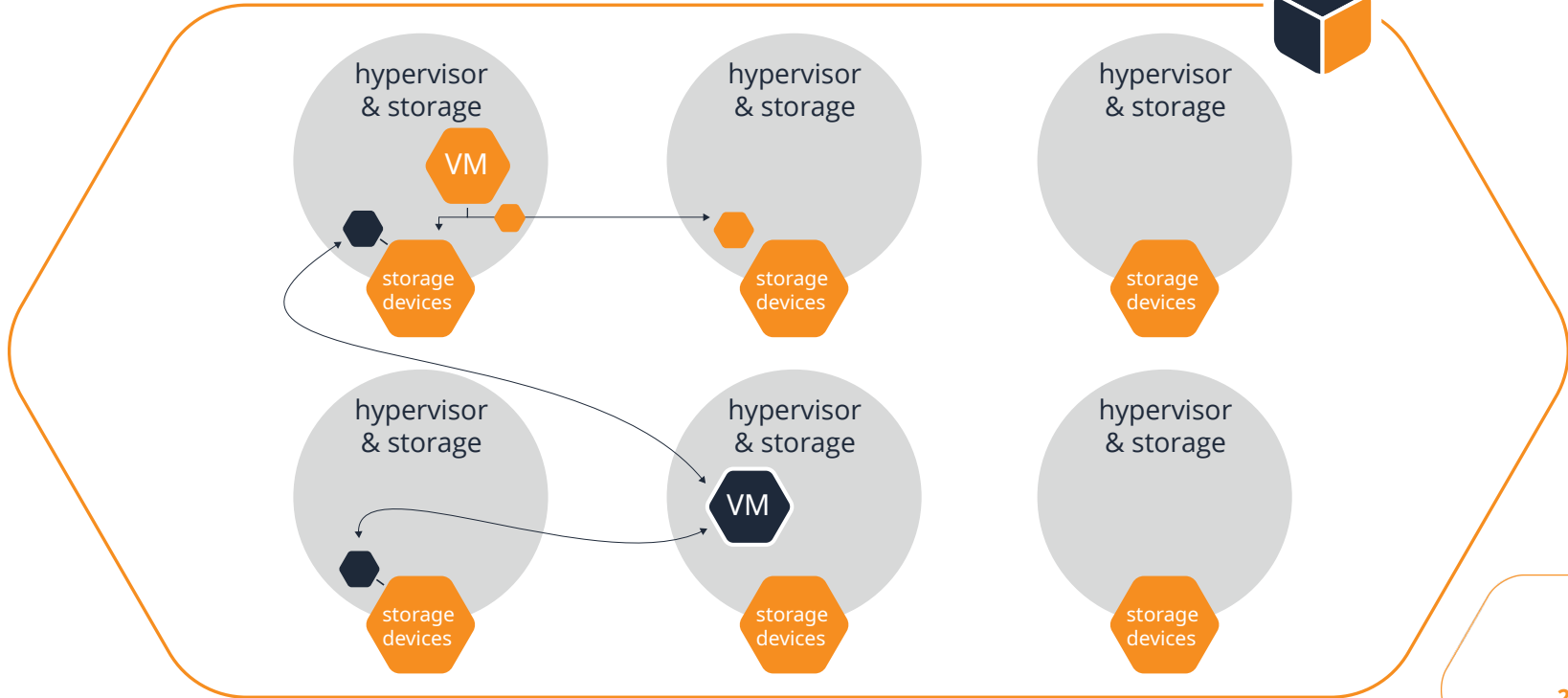
**Example**



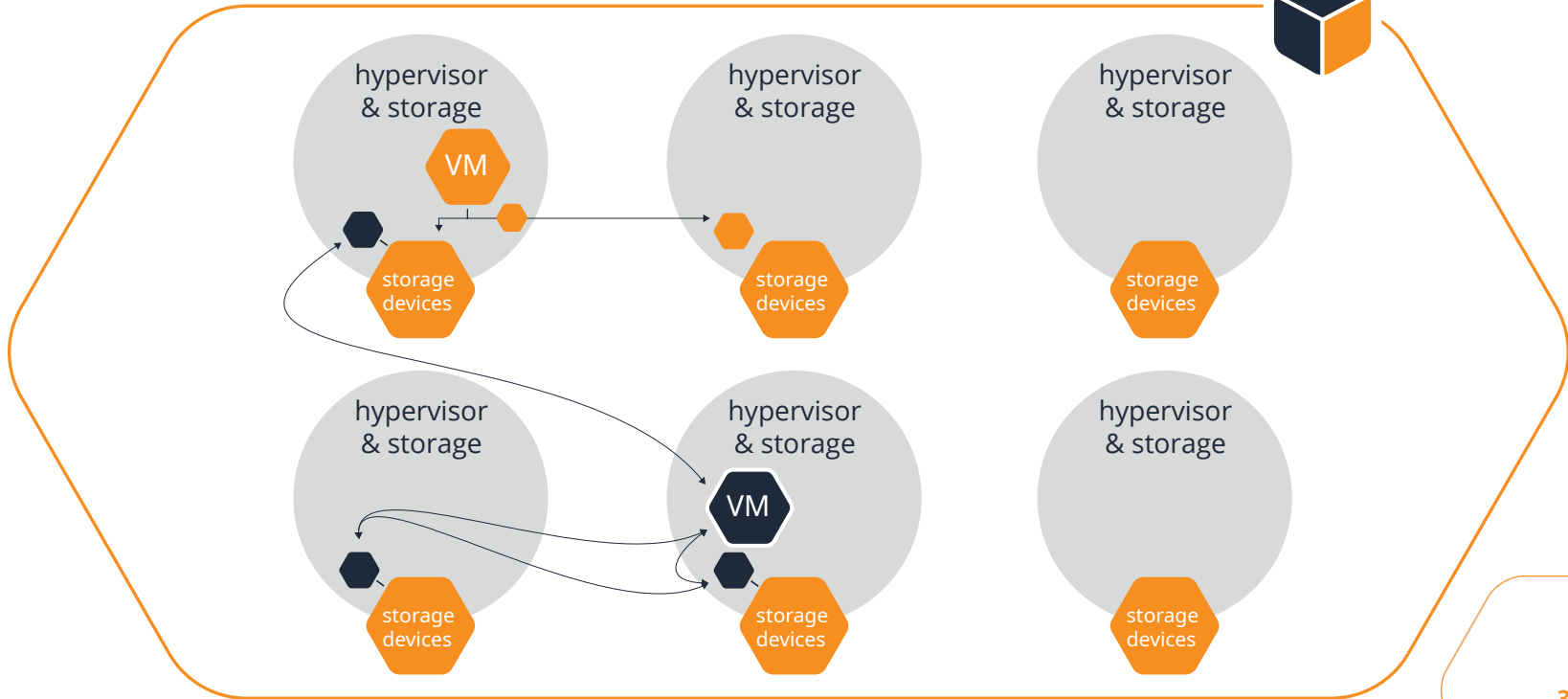
# LINSTOR - Hyperconverged



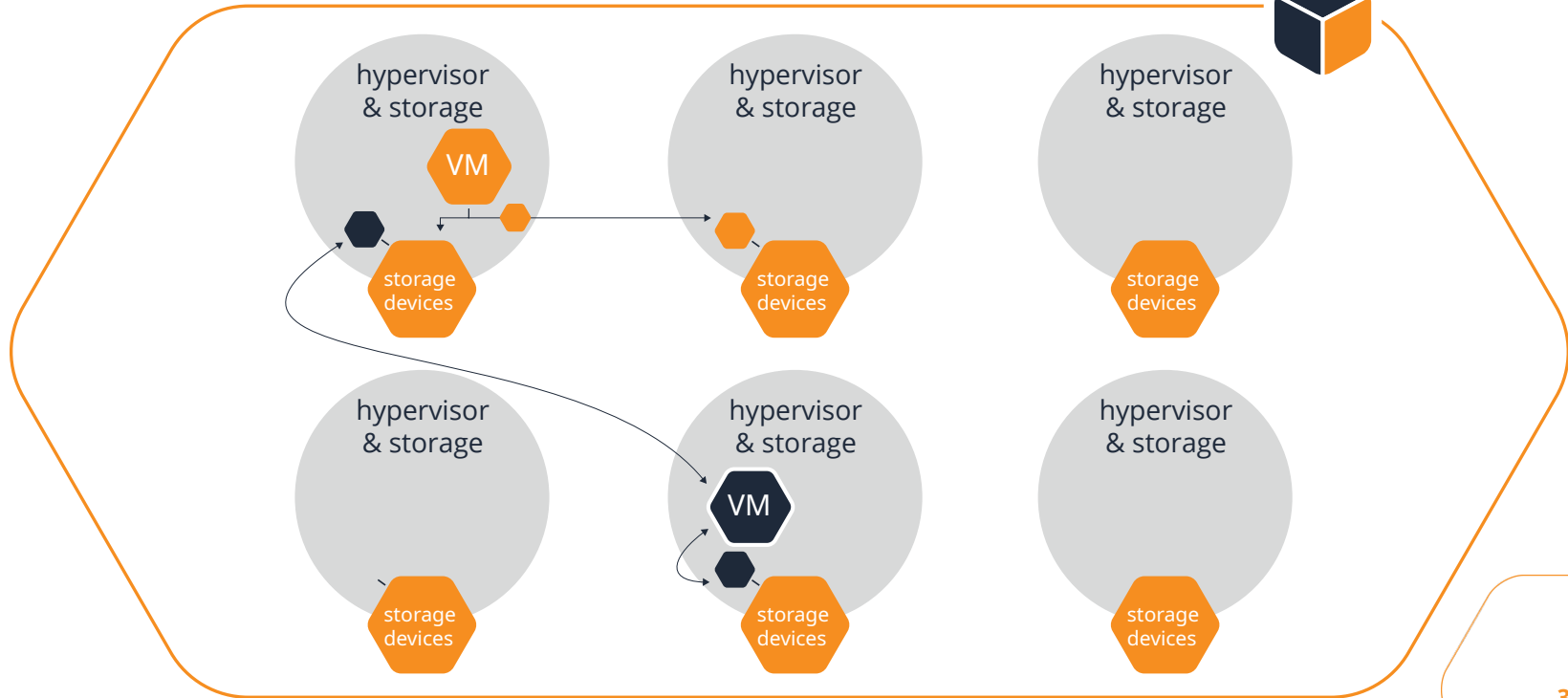
# LINSTOR - VM migrated



# LINSTOR - add local replica

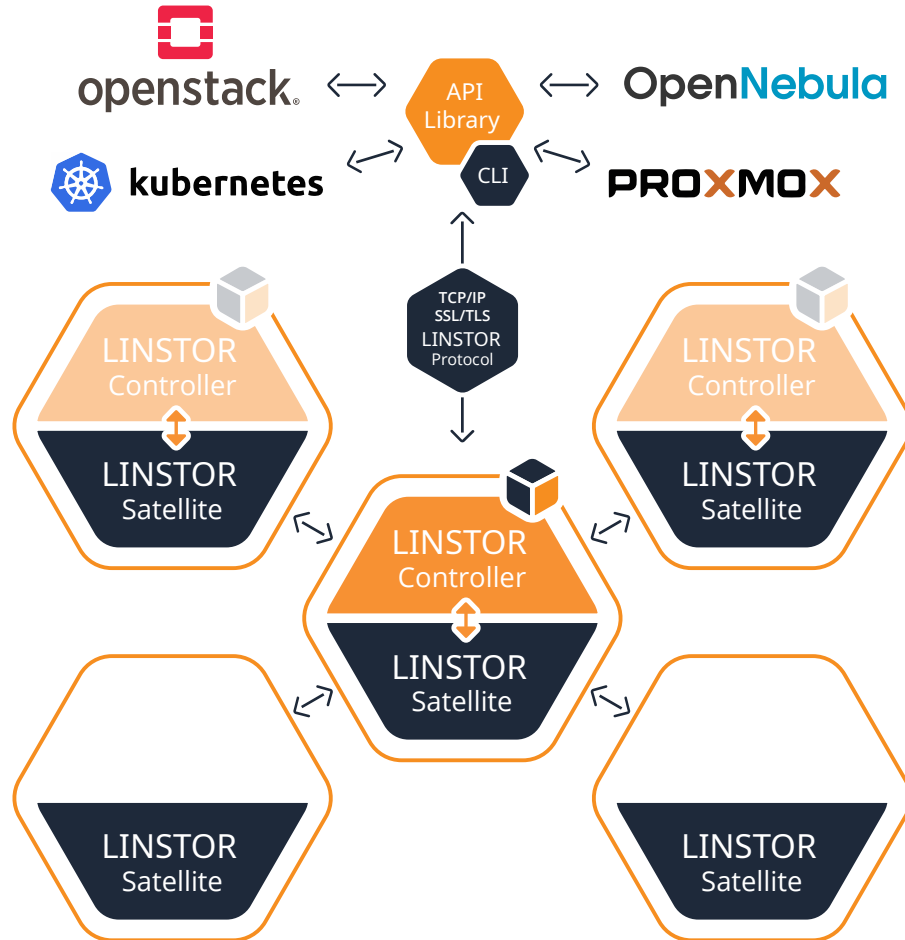


# LINSTOR - remove 3<sup>rd</sup> copy



# **LIN** **STOR**

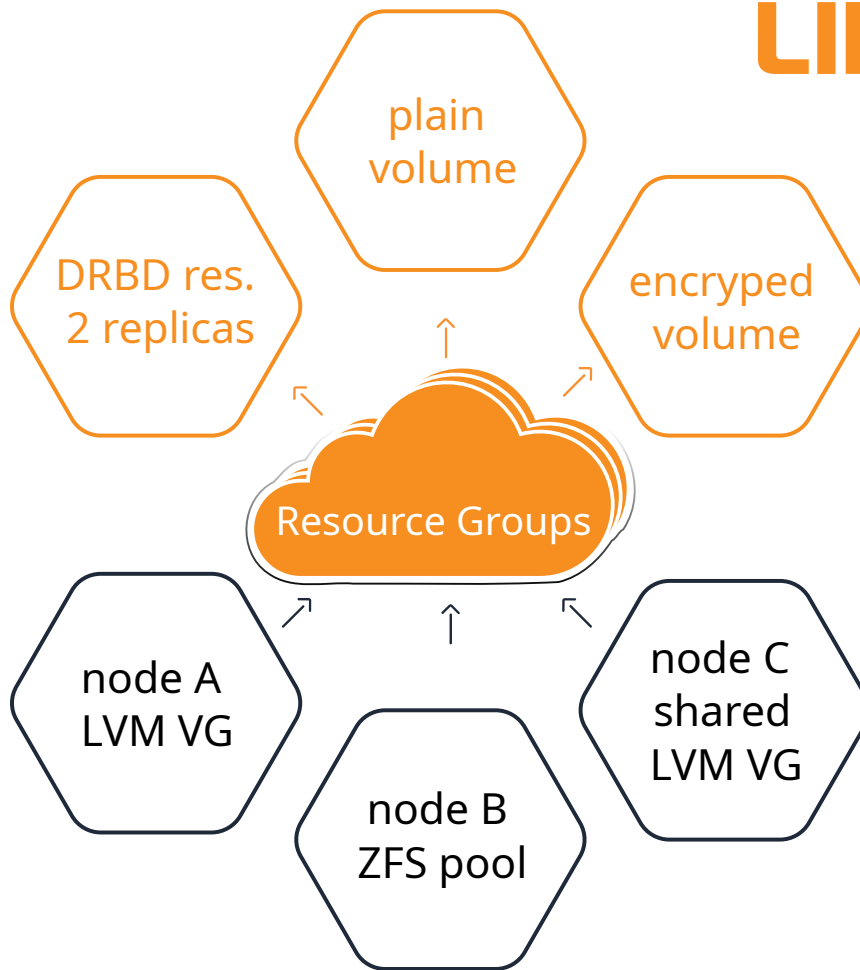
**Architecture, objects and functions**





# LINSTOR Objects

- Nodes
- Resources
  - Volumes
  - Snapshots
  - Storage Pools
  - shared
- Resource groups
- Properties
  - Aux properties



# LINSTOR Storage Layers

Top (opt)

DRBD

Mid (opt & multiple)

LUKS (encryption), caches, NVMe target & initiator

Bottom (required)

LVM VG, ZFS zPool, Exos, OpenFlex, SPDK, shared LVM VG

Below (opt)

VDO (deduplication), software RAID

# LINSTOR data placement

- arbitrary tags on nodes
  - require placement on equal/different/named tag values
- prohibit placements with named existing volumes
  - different failure domains for related volumes

## Example policy

3 way redundant, where two copies are in the same rack but in different fire compartments (synchronous) and a 3<sup>rd</sup> replica in a different site (asynchronous)

## Example tags

rack = number  
room = number  
site = city



# LINSTOR network path selection

- a storage pool may preferred a NIC
  - express NUMA relation of NVMe devices and NICs
- DRBD's multi pathing supported
  - load balancing with the RDMA transport
  - fail-over only with the TCP transport

# **LIN:STOR**

**in the Software Ecosystem**

# LINSTOR connectors



Kubernetes: CSI-driver, Operator, Stork, HA, YAMLs, kubectl plugin



Nomad: CSI-driver (verification pending)



OpenStack: Cinder-driver since “Stein” (April 2019)



OpenNebula: Storage Driver



Proxmox VE: storage plugin

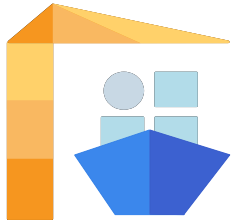


XCP-ng (in preparation)



Apache CloudStack (in preparation)

# Piraeus Datastore



- CSI-driver, Operator, Stork, HA, helm-chart, kubectl
- Publicly available containers of all components
- Joint effort of LINBIT & DaoCloud
- CNCF onboarding to Sandbox in progress




<https://piraeus.io>

<https://github.com/piraeusdatastore>



# LINSTOR SDS & Piraeus Datastore



	LINBIT SDS	Piraeus Datastore
Container base Img	Red hat UBI 	Debian 
Available	<a href="https://drbd.io">drbd.io</a> LINBIT customers only	<a href="https://dockerhub.com">dockerhub</a> , <a href="https://quay.io">quay.io</a> publicly
Support	✓ Enterprise, incl 24/7	Community only
OpenShift/RHCOS	✓ 	n.a.
DRBD driver	Pre-compiled for RHEL/SLES kernels	Compile from source
Contains	LINSTOR, DRBD, operator, CSI-driver, Stork, HA, helm-chart, kubectl	



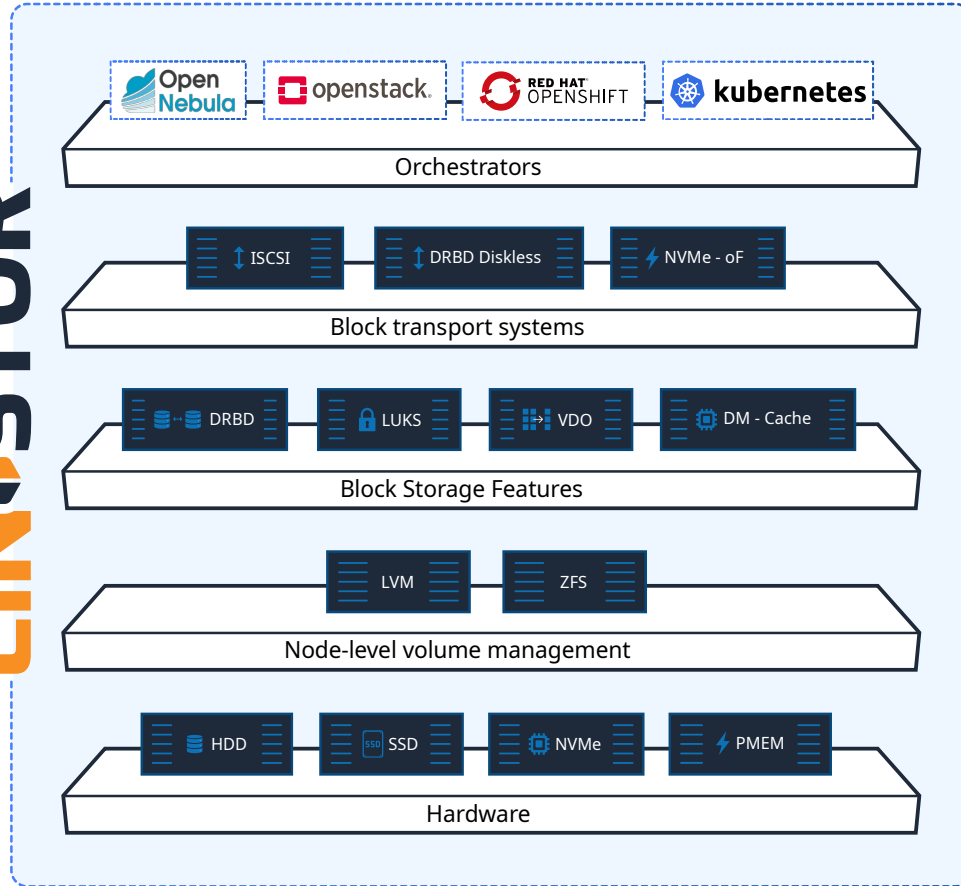


# Translation Matrix

LINSTOR	Resource Group	Resource/Volume
Kubernetes	storageClass	+ file system → persistentVolume
Nomad		
OpenNebula	Datastore	Image
OpenStack	Volume Type	Volume
Proxmox	Storage Pool	Volume
XCP-ng	Storage Repository (SR)	Virtual disk Image (VDI)
CloudStack	Primary storage	Volume

“Naming is hard” – *Phil Karlton*

# Summary





**Thank you**

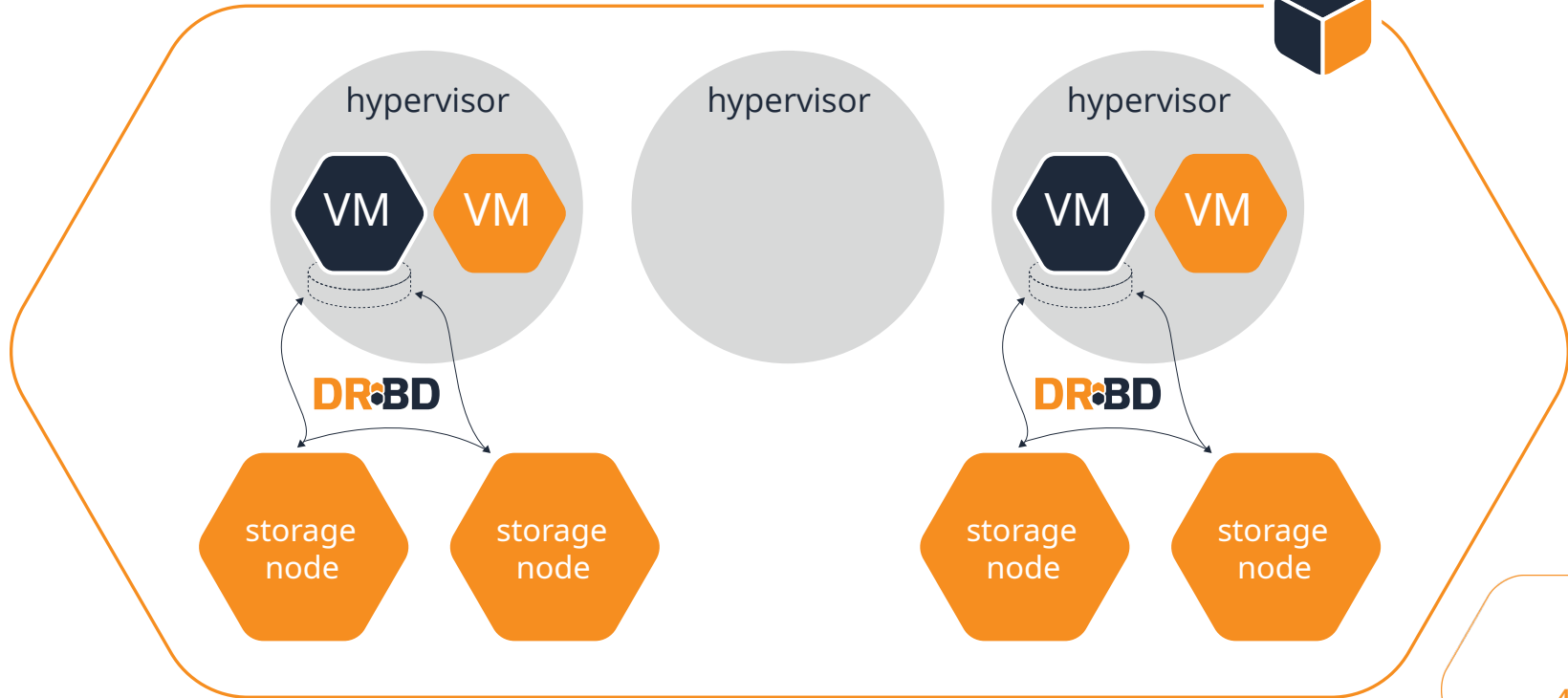
<https://www.linbit.com>

# **LIN:STOR**

## **Appendix Slides: Example Disaggregated Architecture**

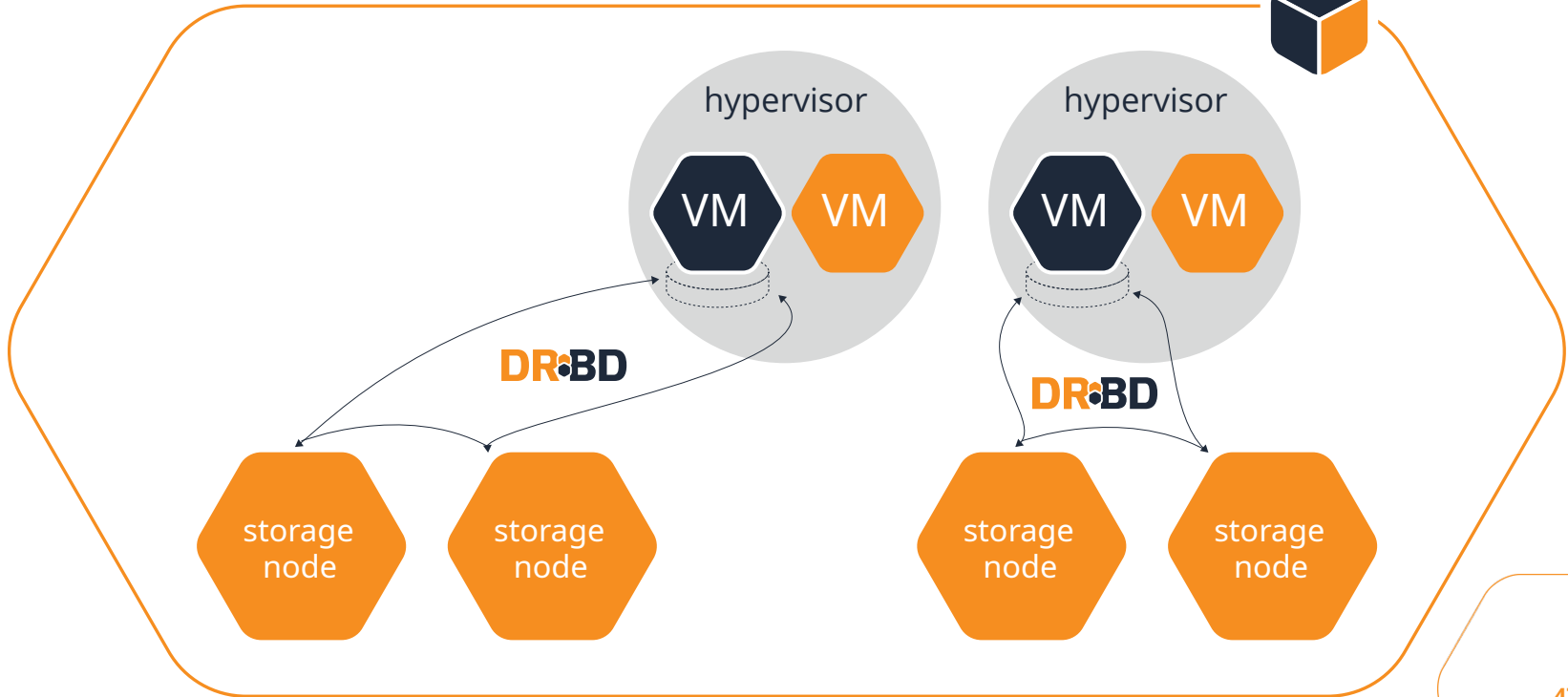


# LINSTOR – disaggregated stack

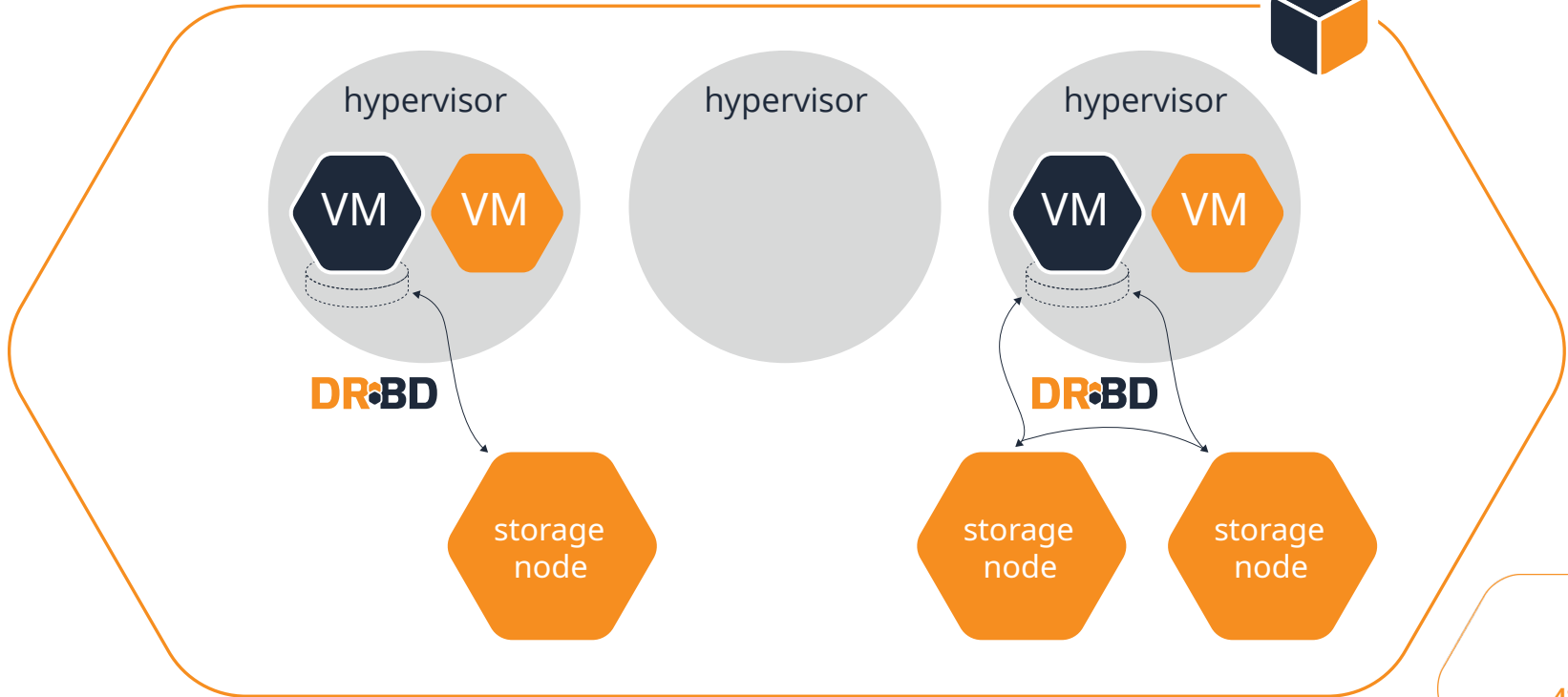


# LINSTOR / failed Hypervisor

LINSTOR



# LINSTOR / failed storage node



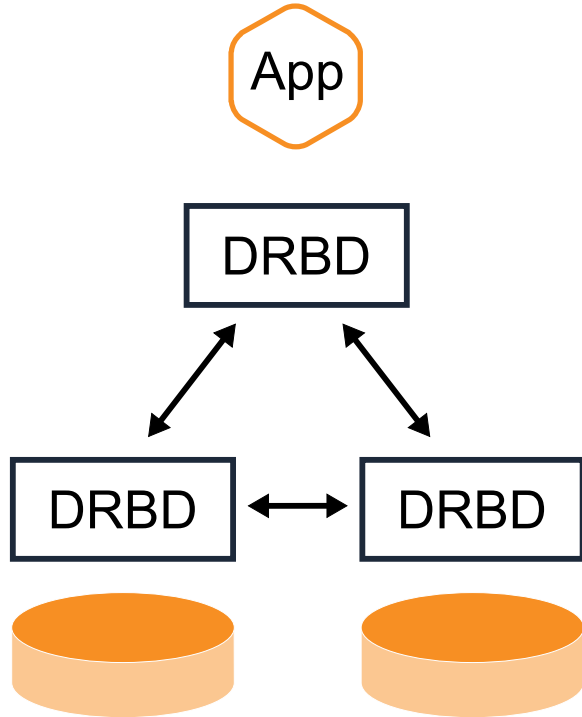
# **LIN:STOR**

## **Appendix Slides: Possible Storage Stacks**



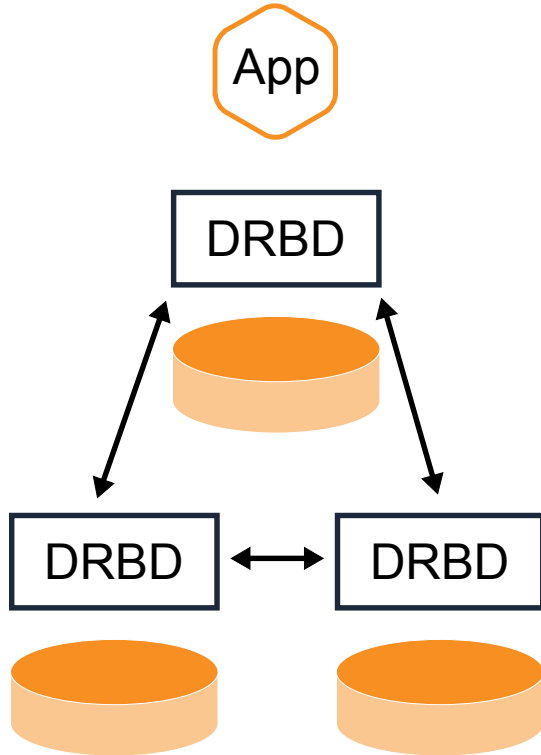


# LINSTOR Storage Stacks



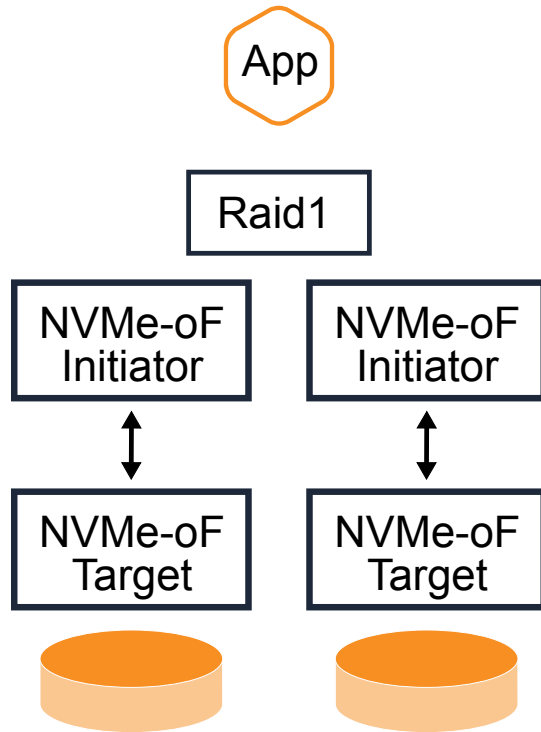
- Disaggregated Storage
- Classic enterprise workloads
  - Data bases
  - Message queues
- Typical Orchestrators
  - OpenStack, OpenNebula
  - Kubernetes
- Flexibly redundancy (1-n)
- HDDs, SSDs, NVMe SSDs

# LINSTOR Storage Stacks



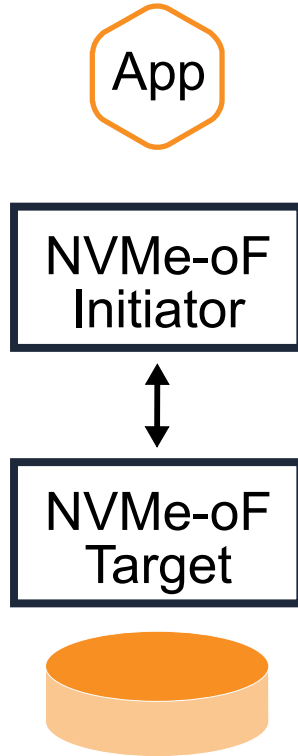
- Hyperconverged
- Classic enterprise workloads
  - Data bases
  - Message queues
- Typical Orchestrators
  - OpenStack, OpenNebula
  - Kubernetes
- Flexibly redundancy (1-n)
- HDDs, SSDs, NVMe SSDs

# LINSTOR Storage Stacks



- Disaggregated
- Classic enterprise workloads
  - Data bases
  - Message queues
- Typical Orchestrators
  - OpenStack, OpenNebula
  - Kubernetes
- NVMe SSDs, SSDs

# LINSTOR Storage Stacks



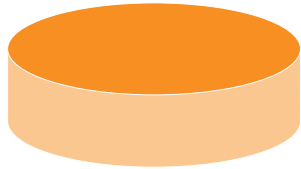
- Disaggregated
- Cloud native workload
  - Ephemeral storage
- Typical Orchestrator
  - Kubernetes
- Application handles redundancy
- Best suited for NVMe SSDs

# LINSTOR Storage Stacks



- Hyperconverged
- Cloud native workload
  - Ephemeral storage
  - PMEM optimized data base
- Typical Orchestrator
  - Kubernetes
- Application handles redundancy
- PMEM, NVDIMMs

# LINSTOR Slicing Storage



- LVM or ZFS
- Thick – pre allocated
  - Best performance
  - Less features
- Thin – allocated on demand
  - Overprovisioning possible
  - Many snapshots possible
- Optional
  - Encryption on top
  - Deduplication below



Setup - WinDRBD version windrbd-0.8.18-signed

**License Agreement**

Please read the following important information before continuing.

Please read the following License Agreement. You must accept the terms of this agreement before continuing with the installation.

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.  
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA

Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

Preamble

- I accept the agreement  
 I do not accept the agreement

Next &gt;

Cancel

WinDRBD

# WinDRBD



- in public beta
  - <https://www.linbit.com/en/drbd-community/drbd-download/>
- Windows 7sp1, Windows 10, Windows Server 2016
- wire protocol compatible to Linux version
- driver tracks Linux version with one day release offset
- WinDRBD user level tools are merged into upstream