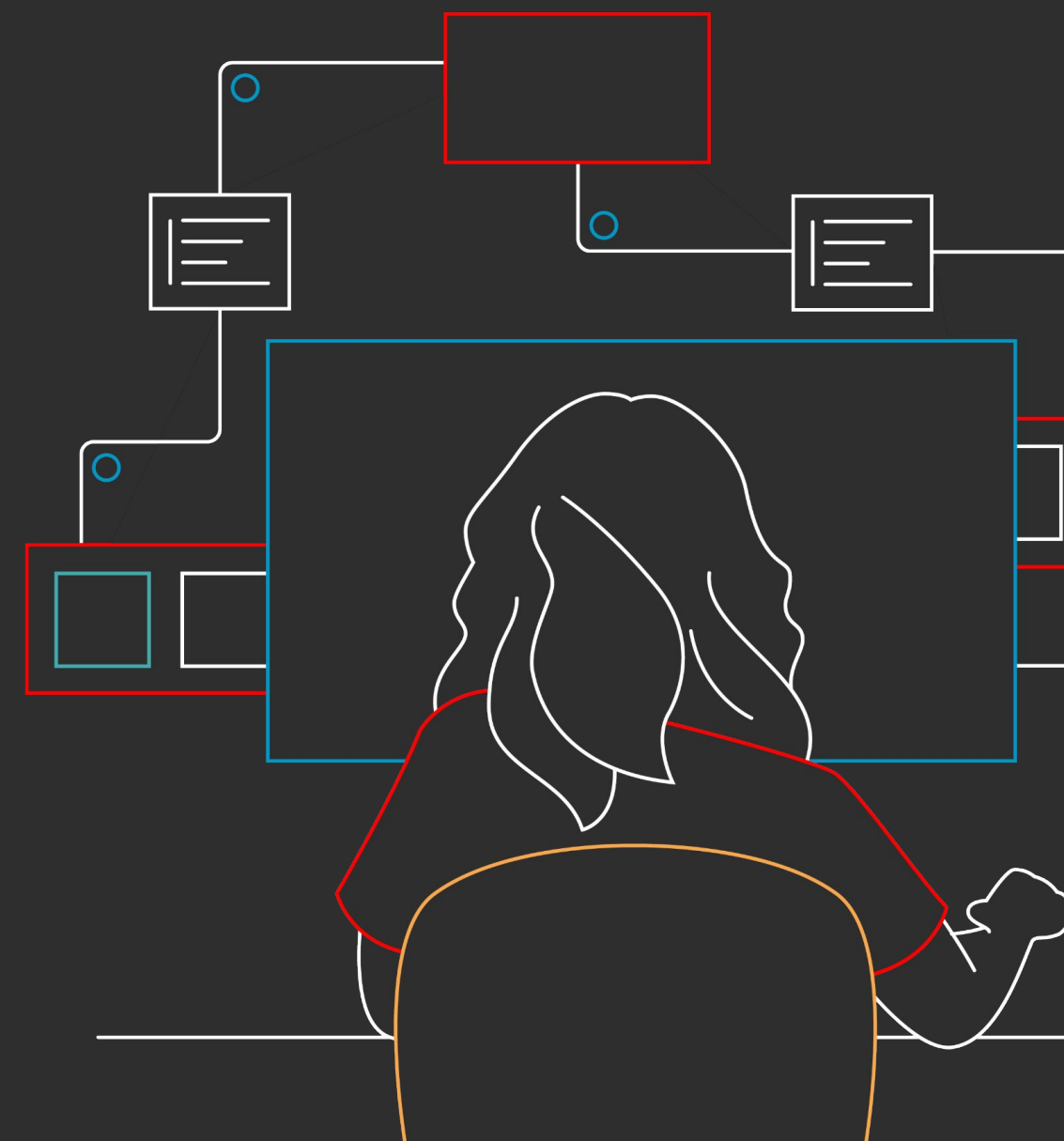


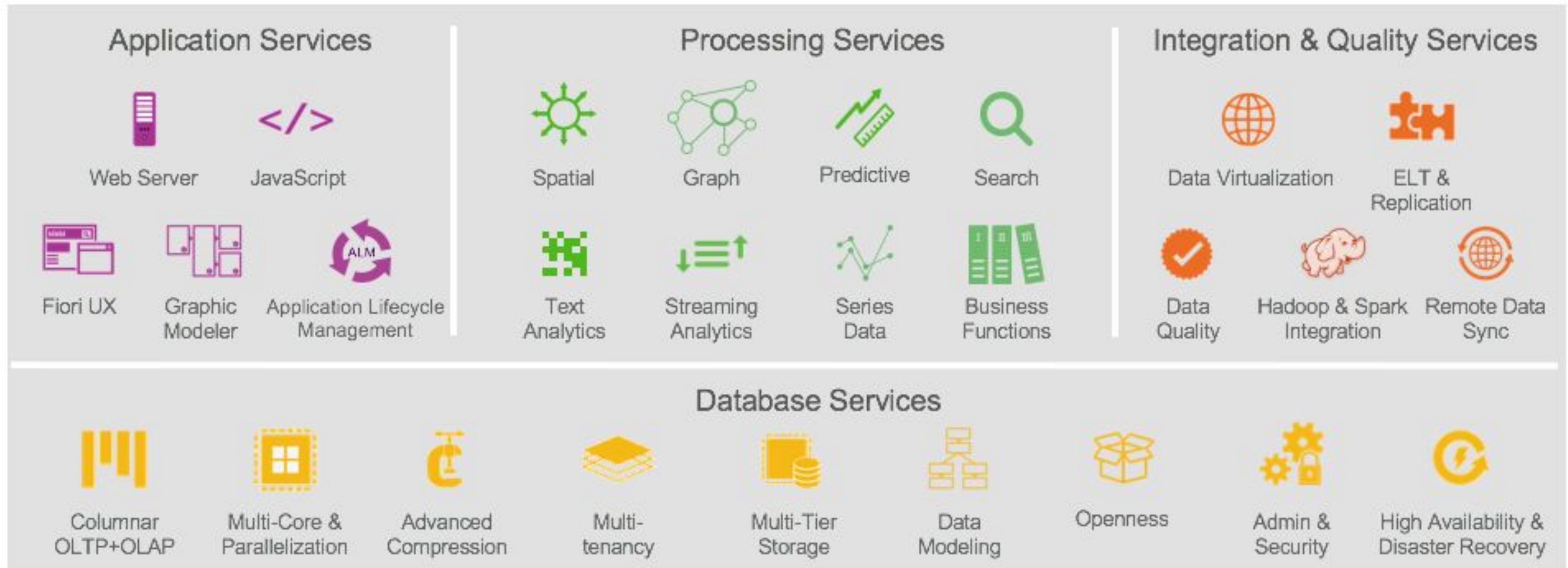
Introducing **SAP** Architecture

Introducing SAP HANA



SAP HANA platform

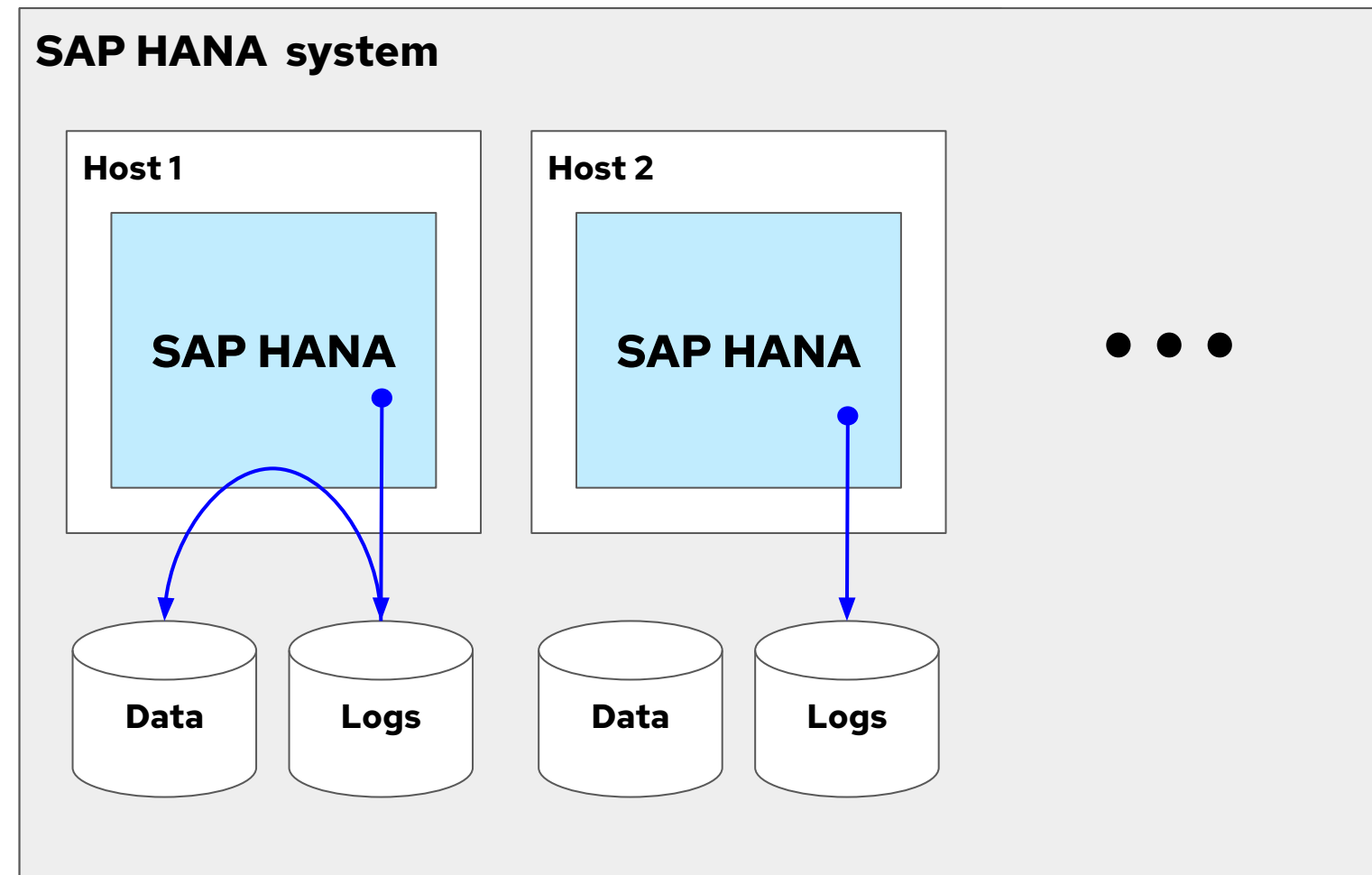
on-premise | cloud | hybrid



Key features of SAP HANA

- In-Memory Computing
- Column-Oriented Data Storage
- Unified Data Model
- Real-Time Analytics
- High Availability and Scalability

SAP HANA architecture



Role details: install SAP HANA

**redhat.sap_install.
sap_hana_install**

This role creates the configuration file for an unattended install of SAP HANA with hdb1cm and kicks off the installation process.

It automatically detects the hdb1cm binary or the installation files in the given installation directory

A minimal configuration example is on the right.

More information about configuration options can be found in the documentation

```
sap_hana_install_software_directory: /sap-software
sap_hana_install_master_password: "*****"
sap_hana_install_sid: 'RHA'
sap_hana_install_instance_number: "00"
```

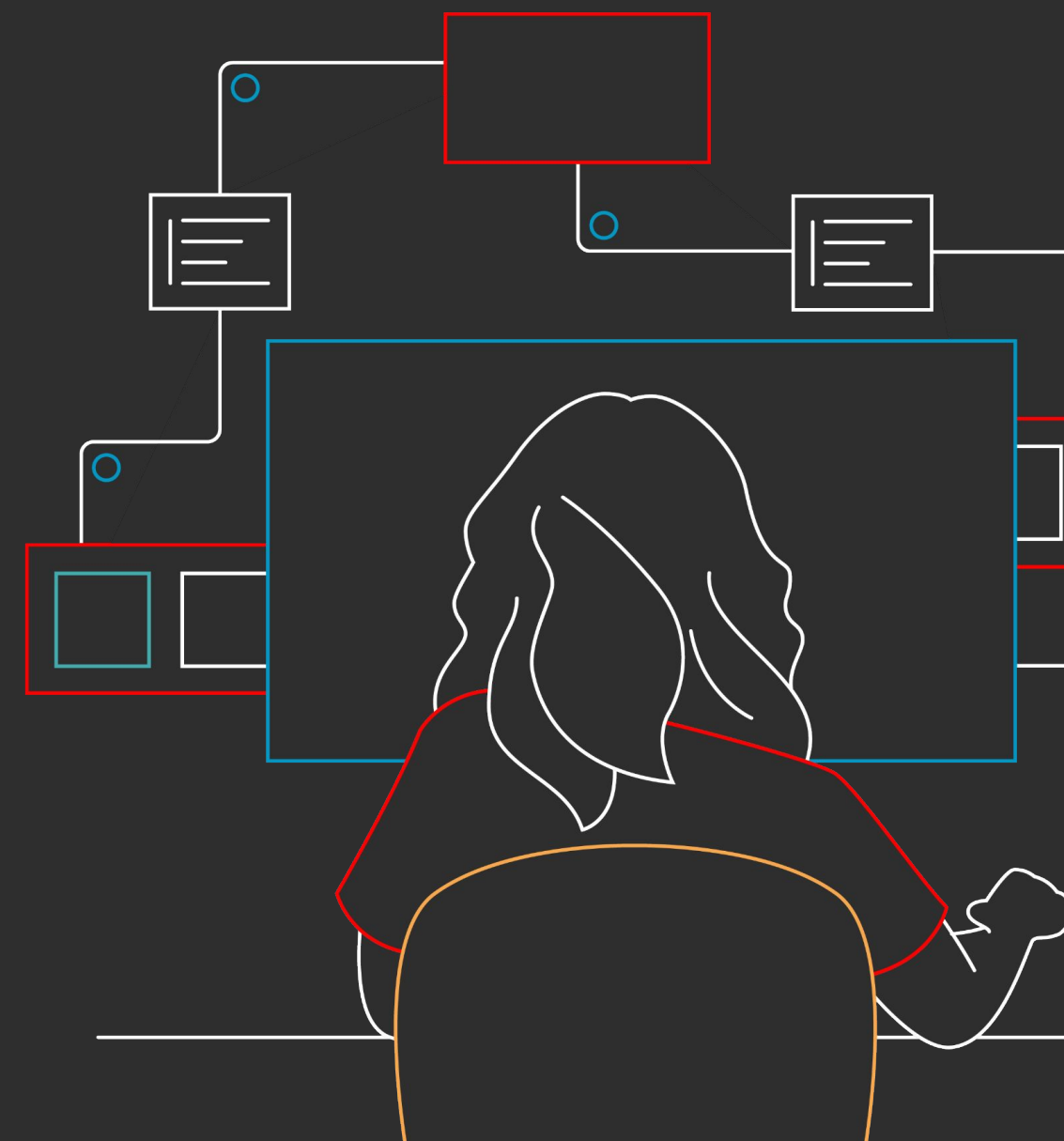
Optional:

```
sap_hana_install_software_extract_directory: /my_local/inst_dir
```

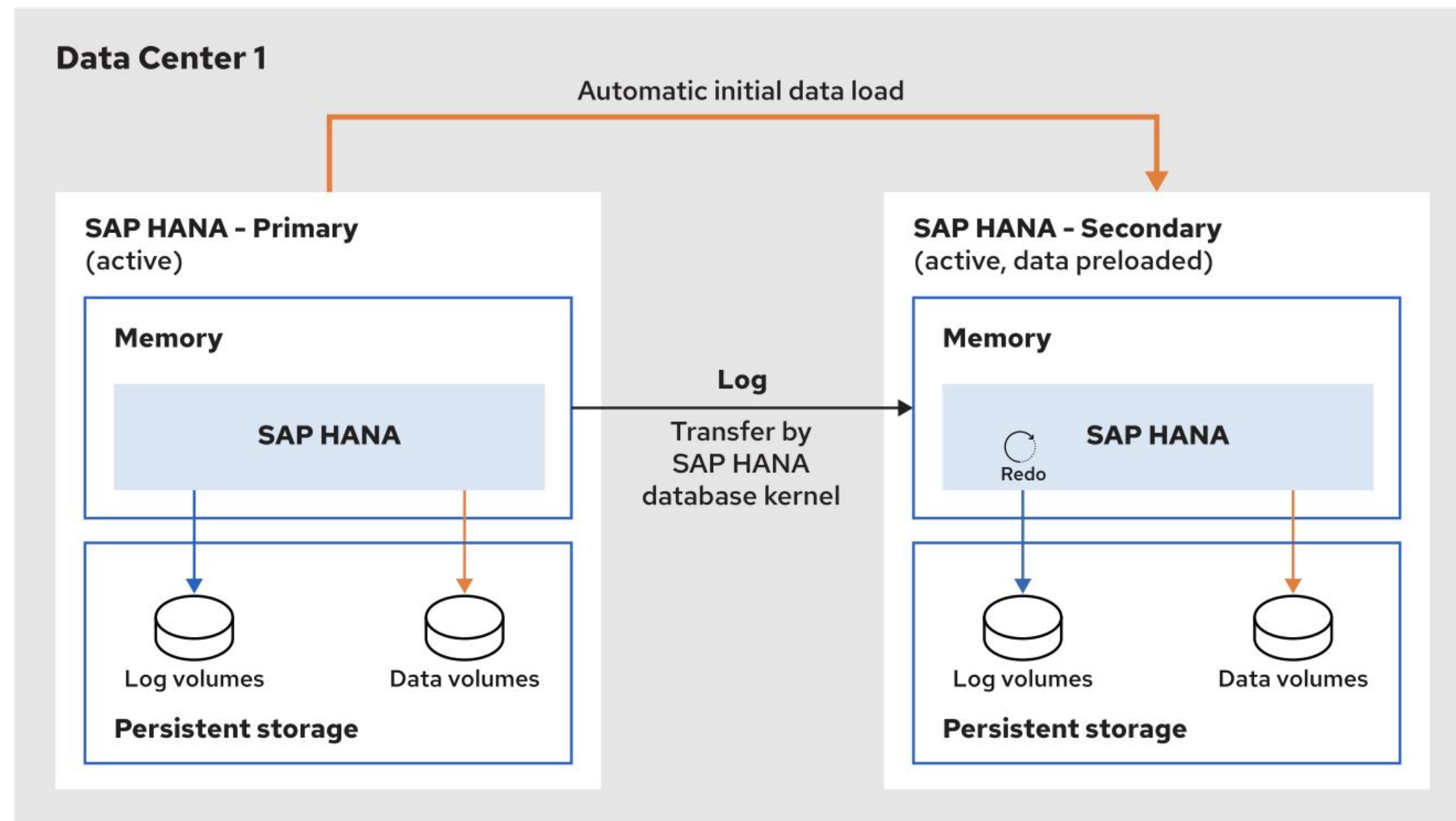
LAB Install HANA

Introducing **SAP** Architecture

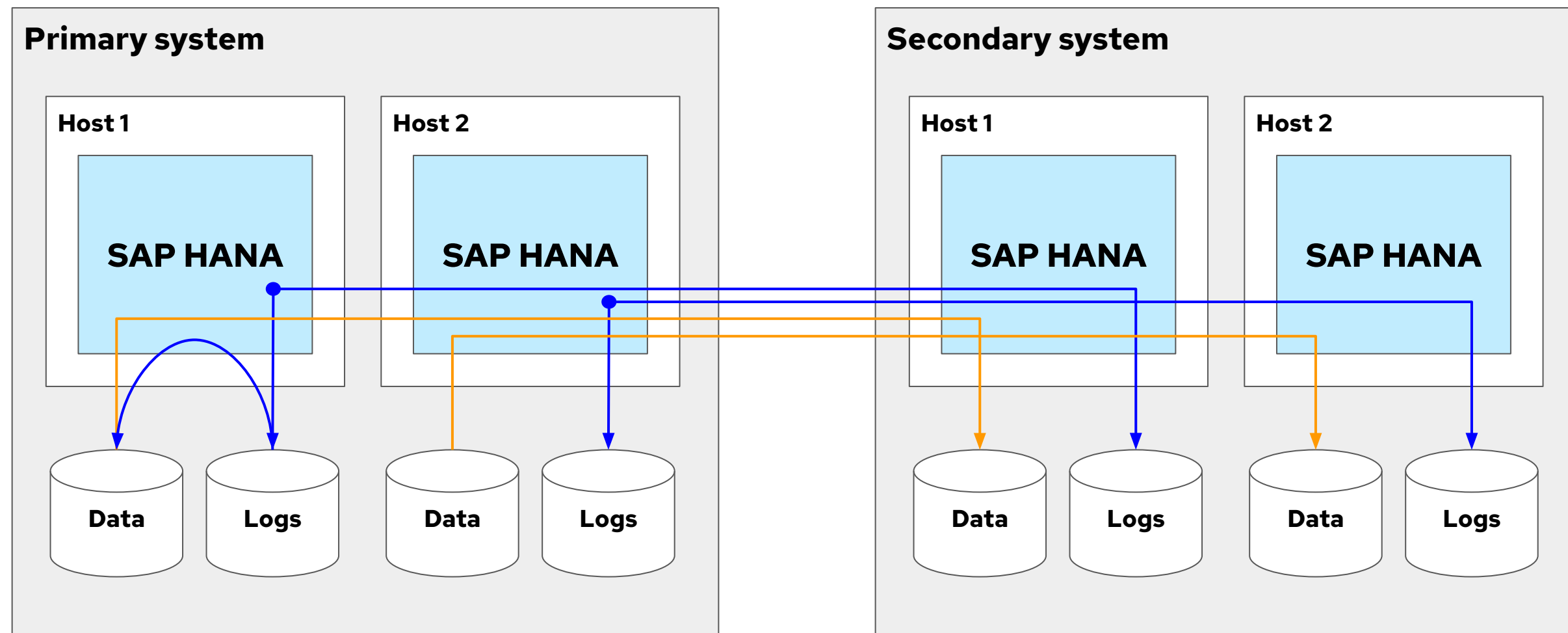
SAP HA concepts



General SAP HSR architecture

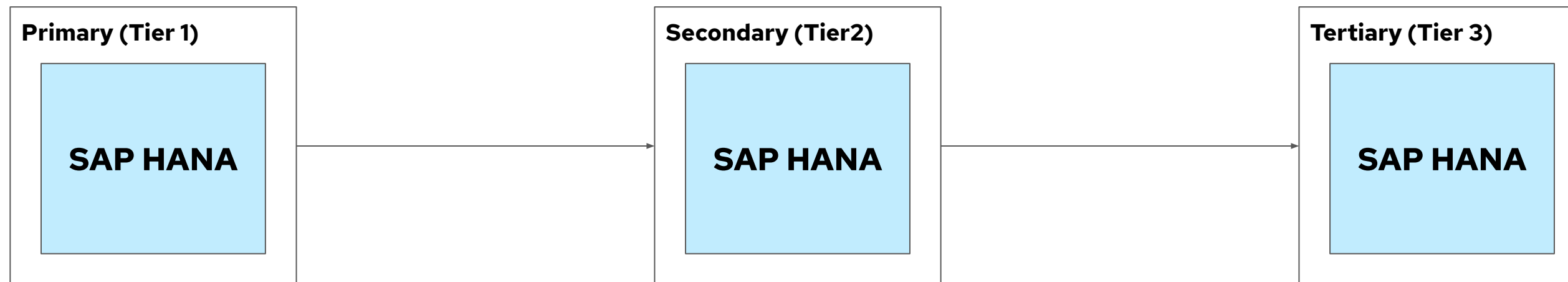


SAP HANA system replication



SAP HANA System Replication Scenarios

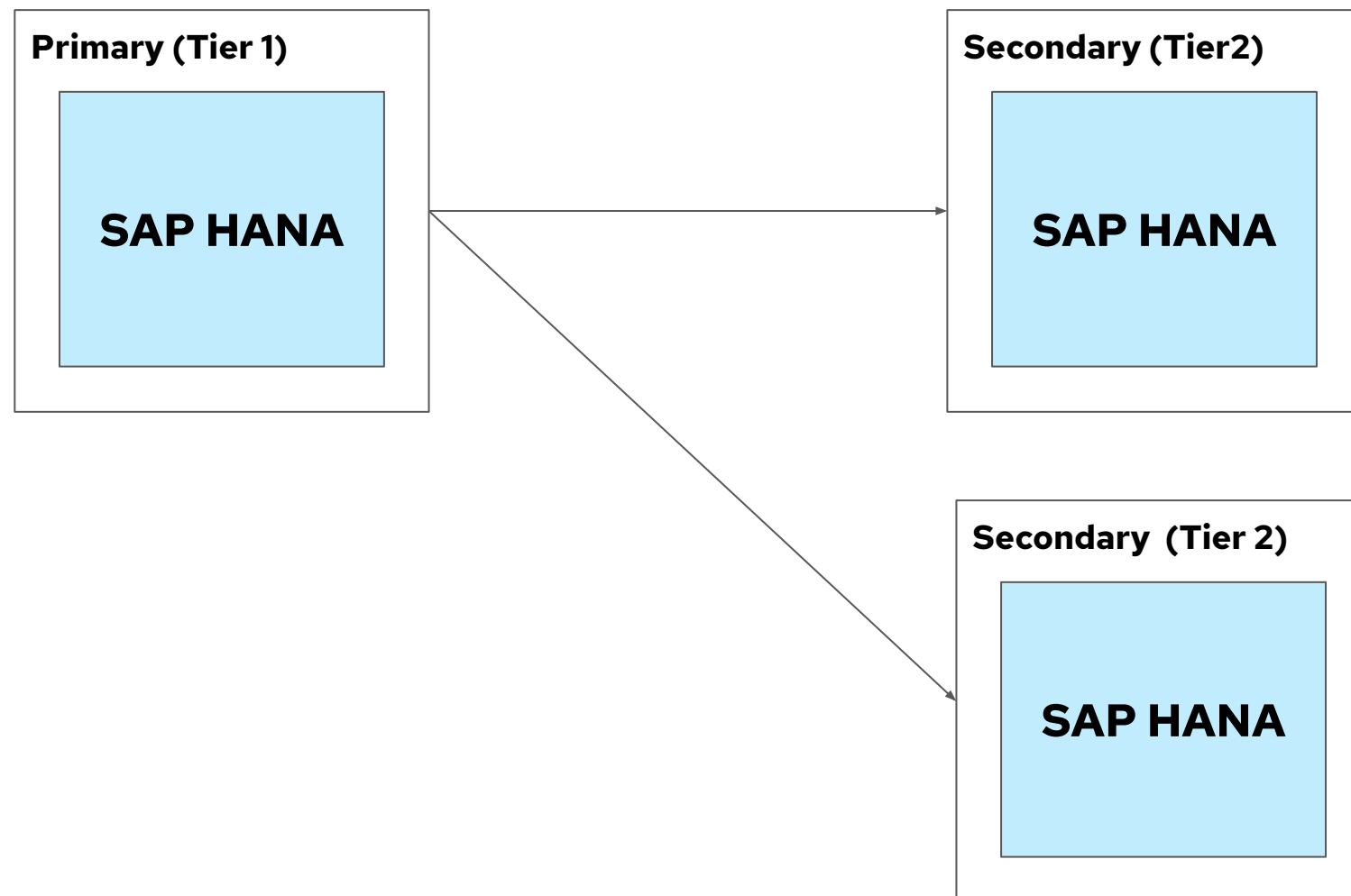
multi-tier replication



Replication Chain

SAP HANA System Replication Scenarios

multi-target replication



Parallel Replication

RHEL High Availability Solutions for SAP



	HDB hdbindexserver crash restart	2-node HA cluster with HANA System Replication (HSR)	2-node HA cluster with multi-tier HSR and manual failover	2-node HA cluster with multi-target HSR and manual failover	3+ node HA cluster with multi-target HSR and auto-failover	cost-optimized HA with auto-evac & failover (multi-SID)
SAP HANA scale-up	Supported ¹	Supported ¹	Supported ³	Supported ³	planned ⁴	Supported ¹
SAP HANA scale-out	Supported ¹	Supported ¹	Supported ³	Supported ³	Supported ¹	planned ⁴

Performance Optimized

- RTO (recovery time objective), RPO (recovery point objective)
- cluster support guidelines:

<https://access.redhat.com/articles/2912891>

<https://access.redhat.com/articles/3397471>

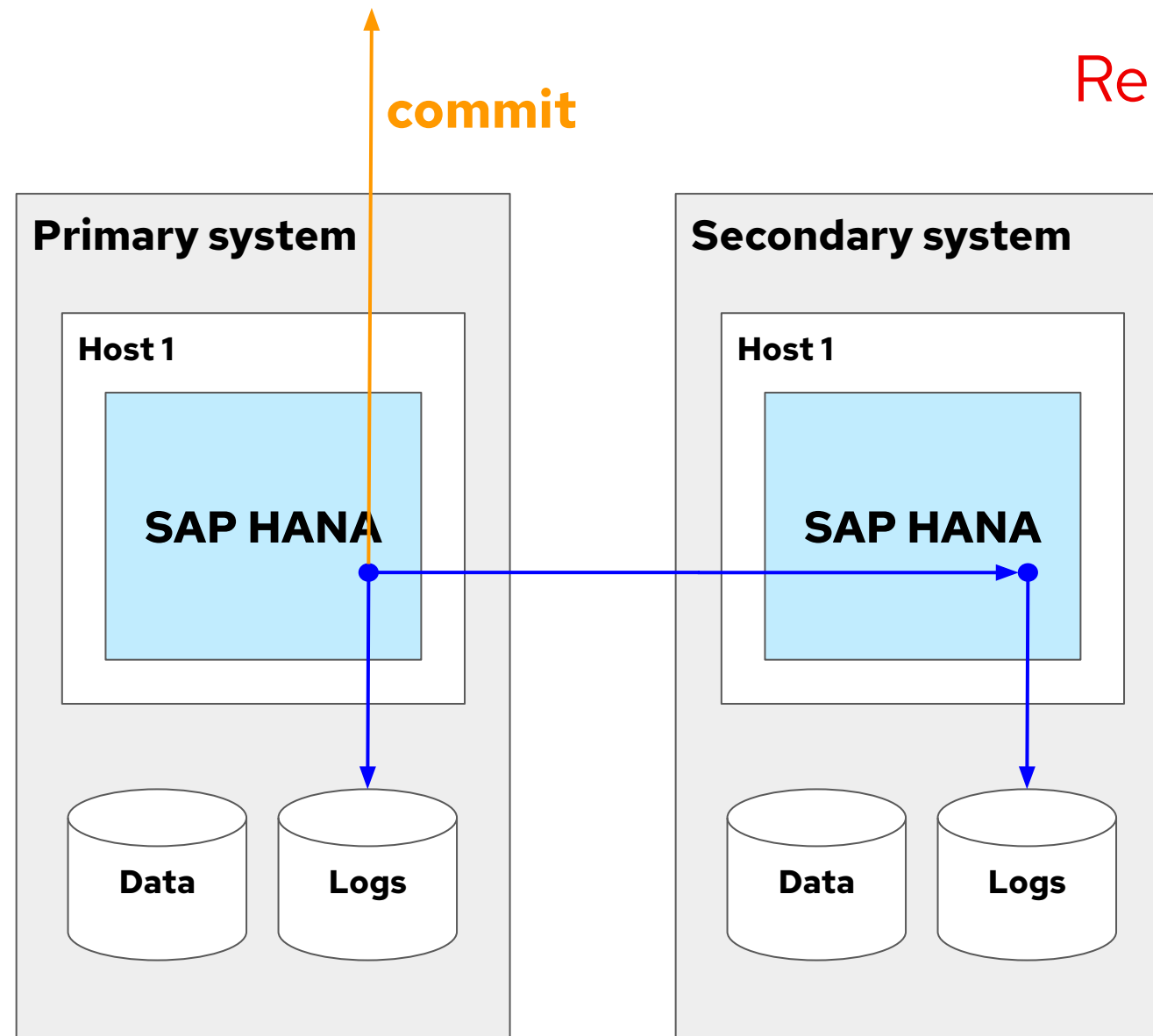
1,2) See [3397471](#), [3569681](#) and [3974941](#) for further details on Support Policies for RHEL HA Cluster Management of SAP HANA, SAP NetWeaver and S/4HANA.

3) RHEL for SAP Solutions provides HA resource agents to automate the takeover from a primary to secondary SAP HANA instance. Customers may also configure multi-target and multi-tier SAP HANA system replication in conjunction with RHEL HA solutions for SAP. In such scenario, the RHEL HA solution will remain functional & supported within its defined scope, whereas failover to such additional targets are not taken care of by the HA cluster solution.

4) Current planning. Actual timelines / scope may change.

SAP HANA system replication

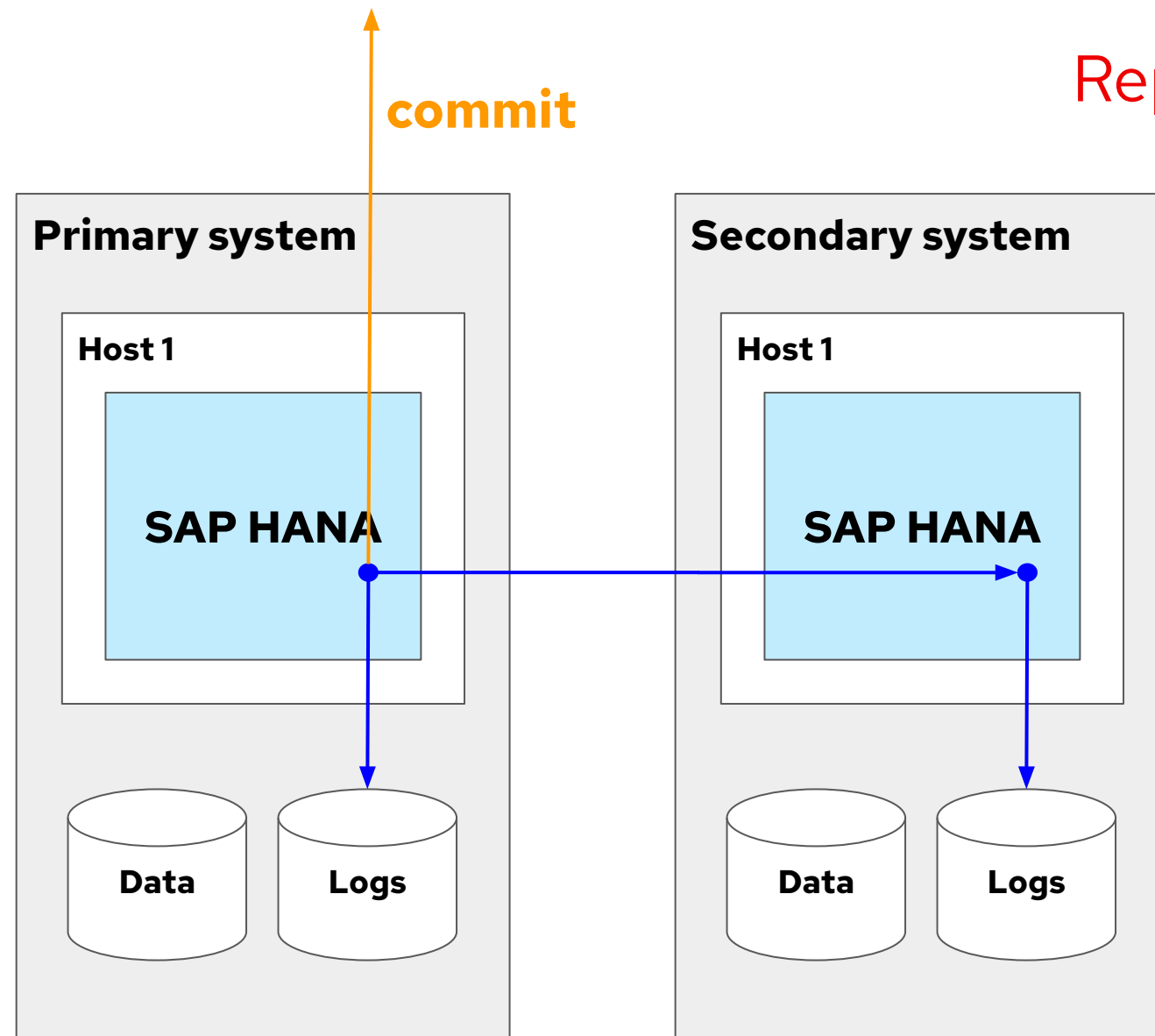
Replication modes



- **syncmem** (default)

SAP HANA system replication

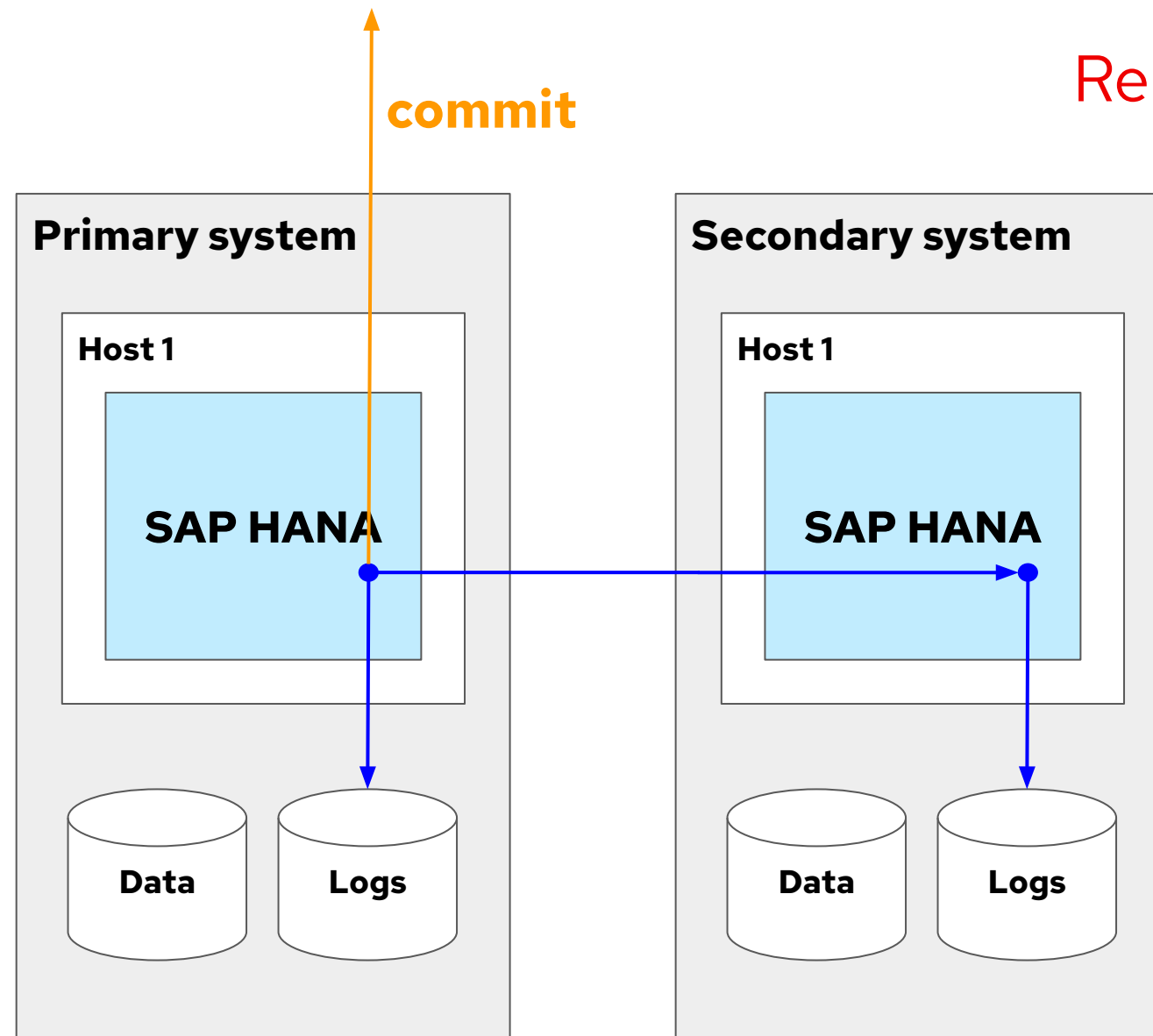
Replication modes



- syncmem (default)
- **sync**

SAP HANA system replication

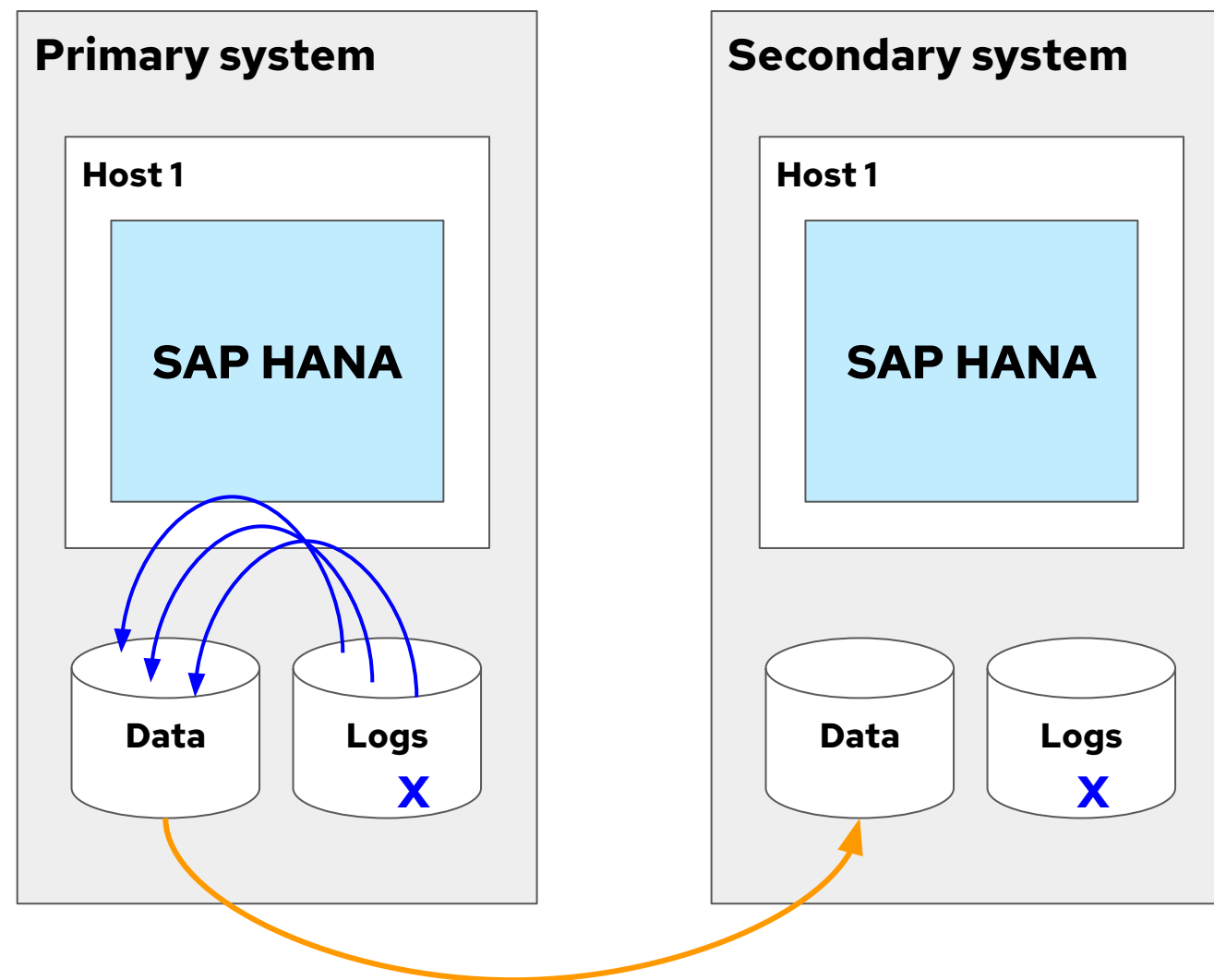
Replication modes



- syncmem (default)
- sync
- **async**

SAP HANA system replication

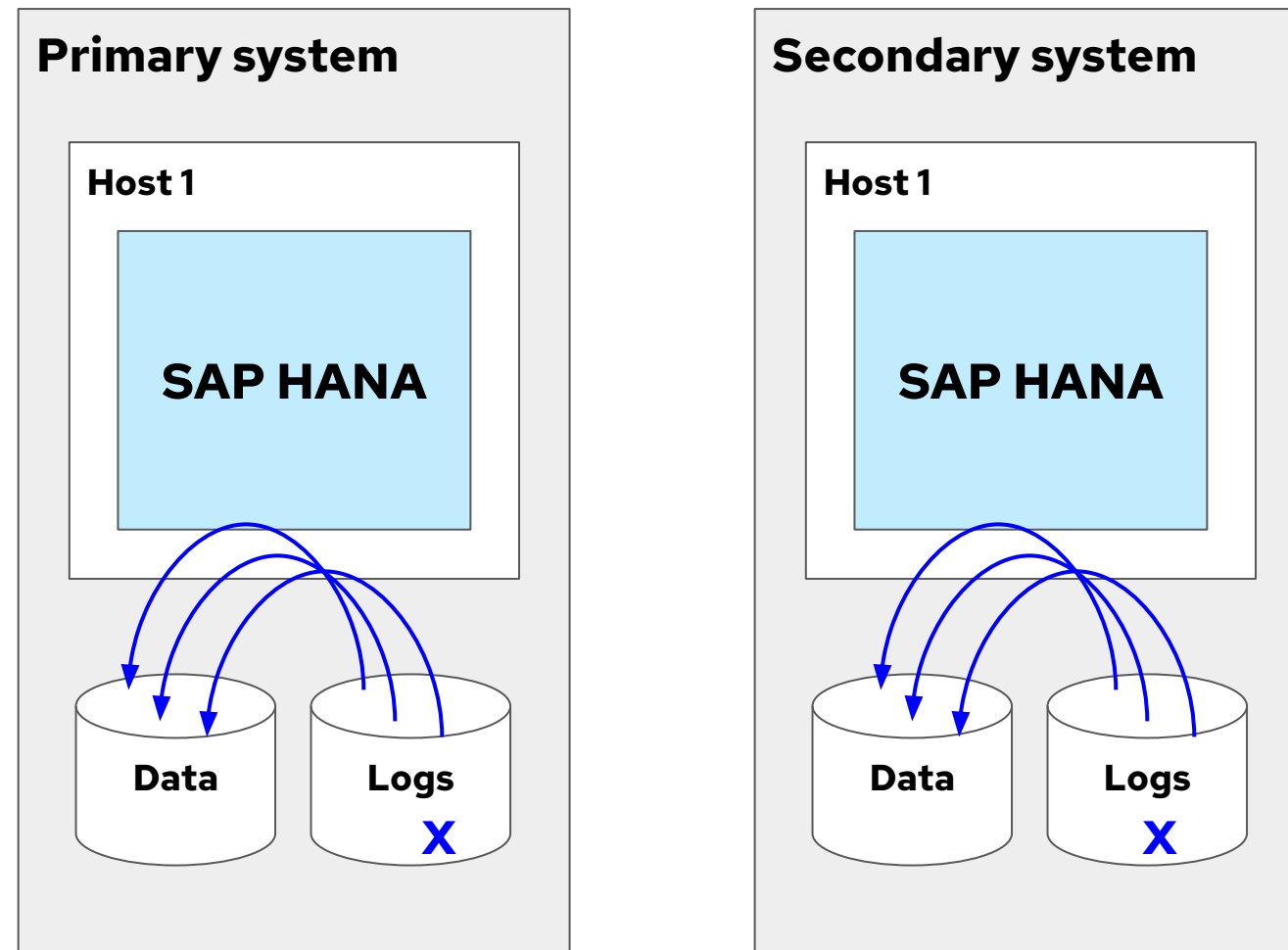
operation modes



- **delta_datashipping**
 - default: every 10 minutes
 - differential backup triggered by secondary
 - log only replayed at takeover

SAP HANA system replication

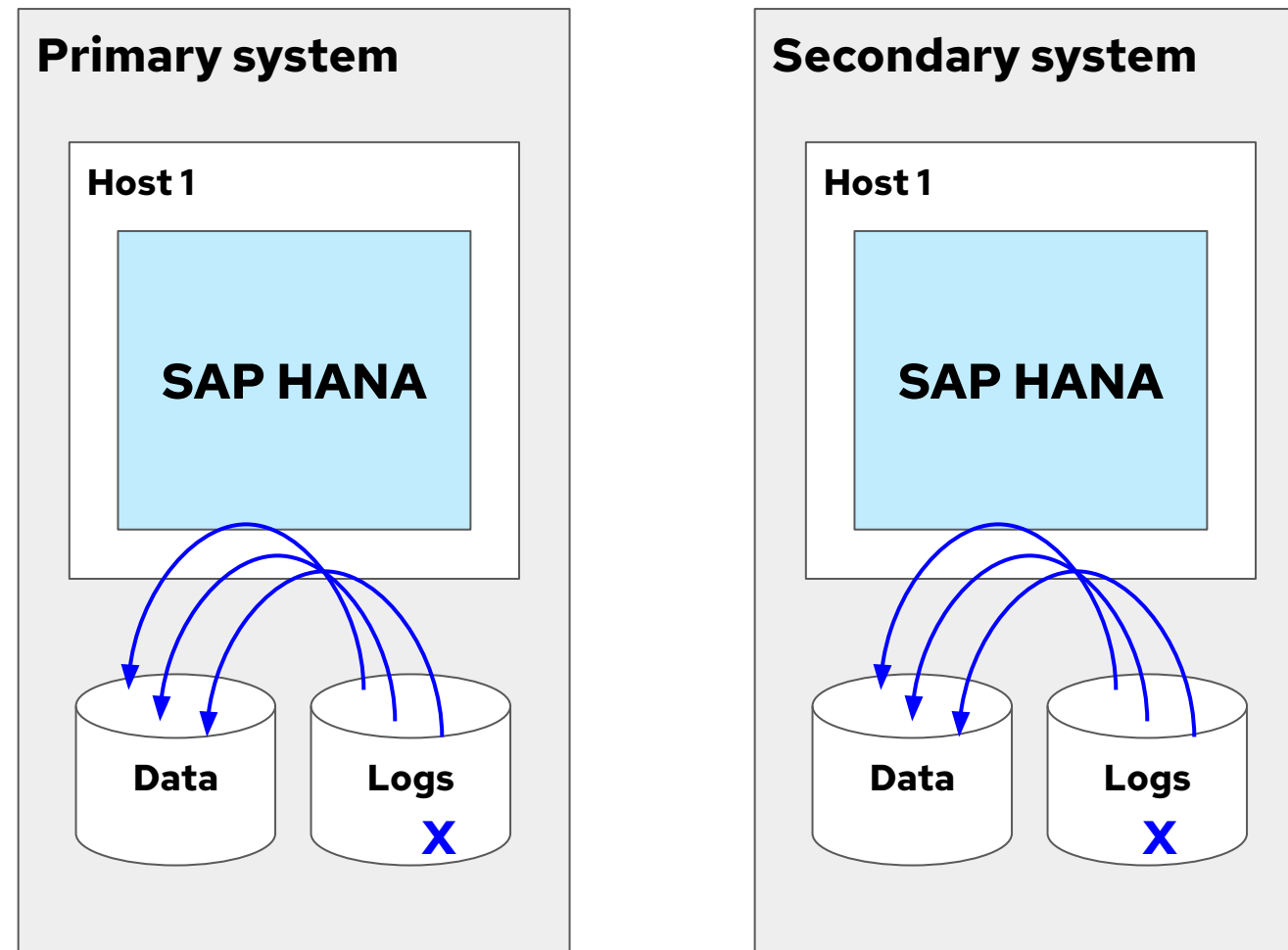
operation modes



- delta_datashipping
- **logreplay**
 - persistent log shipping and replay
 - retention period in case of network interruption
 - after retention full data snapshot is transferred again
- logreplay_readaccess

SAP HANA system replication

operation modes



- delta_datashipping
- logreplay
- **logreplay_readaccess**
 - identical to logreplay
 - enables read_access on the secondary

Manual installation steps

without ansible

1. Set up an operating system instance (100% identical on 2 systems).
2. Create file systems.
3. Check the subscription and repositories.
4. Prepare the operating system with preconfigured system roles.
5. Verify and update sync hostname resolution.
6. Obtain and download the SAP HANA installation package.
7. Install HANA and use the same SID (System Identifier), instance number, and UID (User ID).
8. Back up the primary database server.
9. Copy PKI files from the primary to the secondary database server.
10. Run the `sr_enable` command on the primary database server.
11. Run the `sr_register` command on the secondary database server

Overview of manual steps

Verify Logmode:

```
hdbsql -u system -p $HANA_SYSTEM_PASSWORD -i 00 "select value from "SYS"."M_INIFILE_CONTENTS" where key='log_mode'"
```

Required commands to change log mode:

```
hdbsql -U HDB_SYSTEMDB -i 00
```

```
ALTER SYSTEM ALTER CONFIGURATION ('global.ini', 'SYSTEM') SET ('persistence', 'log_mode') = 'normal' WITH RECONFIGURE;
```

```
ALTER SYSTEM ALTER CONFIGURATION('global.ini','HOST','{{ ansible_hostname }}') SET ('persistence','log_mode') = 'normal' WITH RECONFIGURE;
```

List Databases for backup

```
SELECT * FROM M_DATABASES
```

Back up the primary database on hana01

```
hdbsql -i 00 -u system -p $HANA_SYSTEM_PASSWORD -d SYSTEMDB "BACKUP DATA USING FILE ('/tmp/foo')"
```

```
hdbsql -i 00 -u system -p $HANA_SYSTEM_PASSWORD -d RH1 "BACKUP DATA FOR RH1 USING FILE ('/tmp/foo-RH1')"
```

Initialize replication on the hana01 primary node:

```
sudo su - rh1adm
```

```
hdbnsutil -sr_enable --name=DC1
```

On secondary node:

```
sudo su - rh1adm -c "HDB stop" # stop HANA
```

```
# copy the keys
```

```
scp root@hana01:/usr/sap/RH1/SYS/global/security/rsecssfs/key/SSFS_RH1.KEY /usr/sap/RH1/SYS/global/security/rsecssfs/key/SSFS_RH1.KEY
```

```
scp root@hana01:/usr/sap/RH1/SYS/global/security/rsecssfs/data/SSFS_RH1.DAT /usr/sap/RH1/SYS/global/security/rsecssfs/data/SSFS_RH1.DAT
```

```
sudo su - rh1adm
```

```
hdbnsutil -sr_register --remoteHost=node01 --remoteInstance=00 --replicationMode=syncmem --name=DC2
```

```
HDB start
```

```
hdbnsutil -sr_state
```

```
cdpy
```

```
python systemReplicationStatus.py
```

Role details: configure SAP HANA system replication

**redhat.sap_install.
sap_ha_install_hsr**

If you have used the `sap_hana_install` role to set up two identical instances, you can use this role to easily set up SAP HANA system replication between these instances.

Example:

Common variables that need to be used on both hosts:

```
# Already defined
sap_domain: domain.name
sap_hana_sid: RHA
sap_hana_install_instance_number: "00"
sap_hana_install_master_password: "*****"

# Optional
sap_ha_install_hana_hsr_rep_mode: sync
sap_ha_install_hana_hsr_oper_mode: logreplay
```

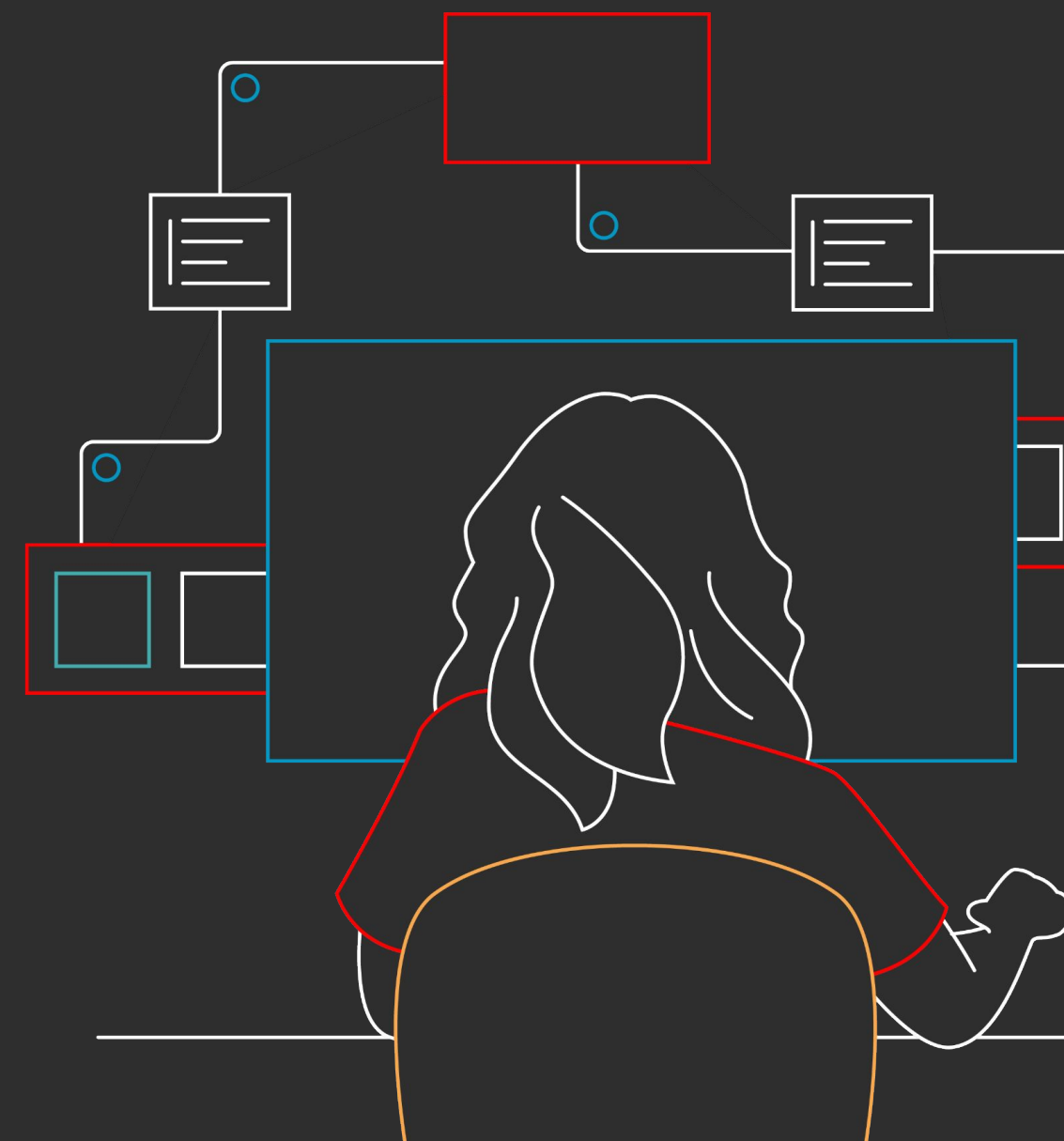
```
sap_hana_cluster_nodes:
  - node_name: "hana1"
    node_ip: "1.2.3.4"
    node_role: primary
    hana_site: DC01

  - node_name: "hana2"
    node_ip: "hostvars['hana2']['private_ip']"
    node_role: secondary
    hana_site: DC02
```

LAB: Configure HSR

Explaining Red Hat Enterprise Linux HA **for SAP** Solutions

Pacemaker Architecture and Components



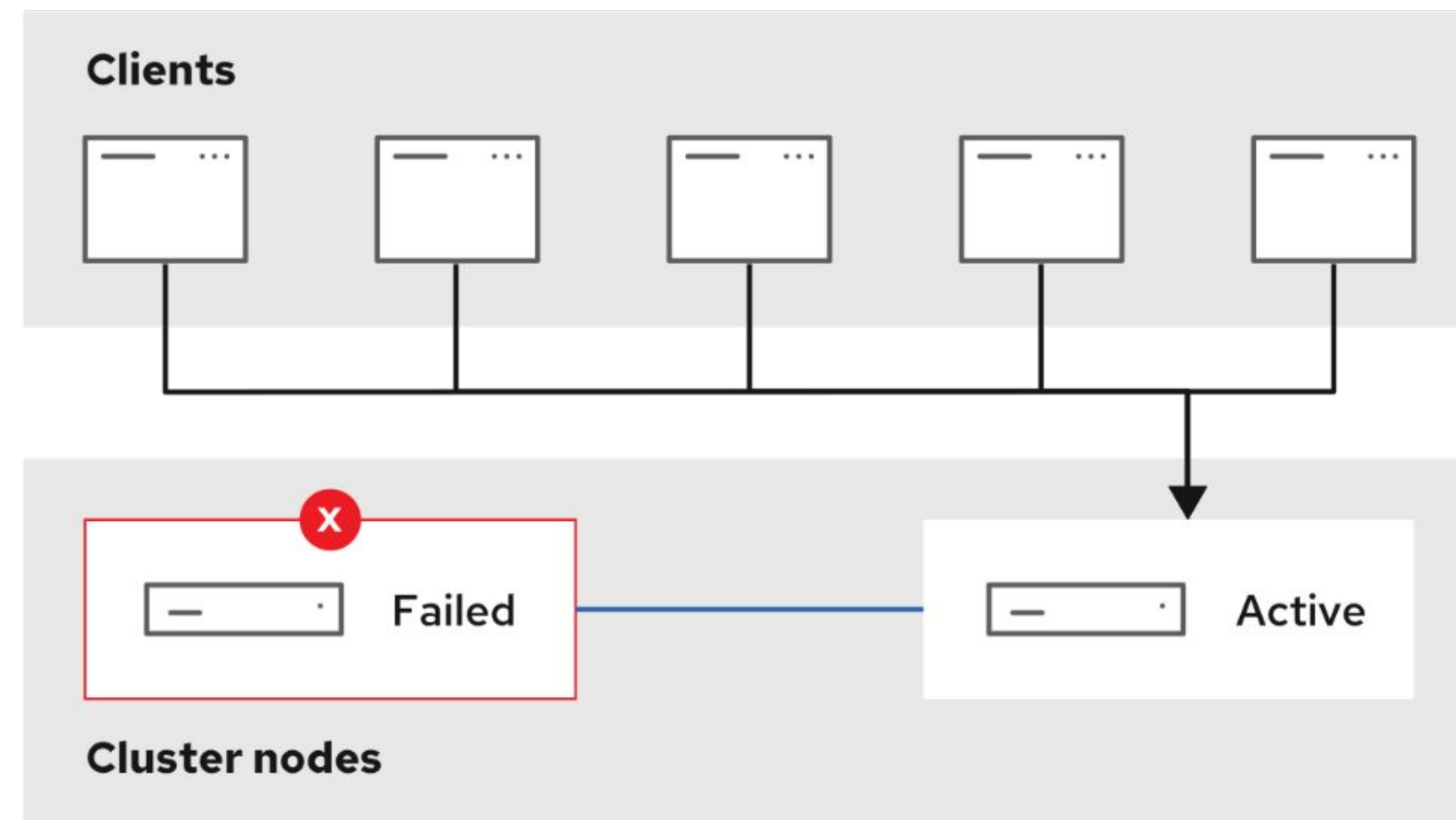
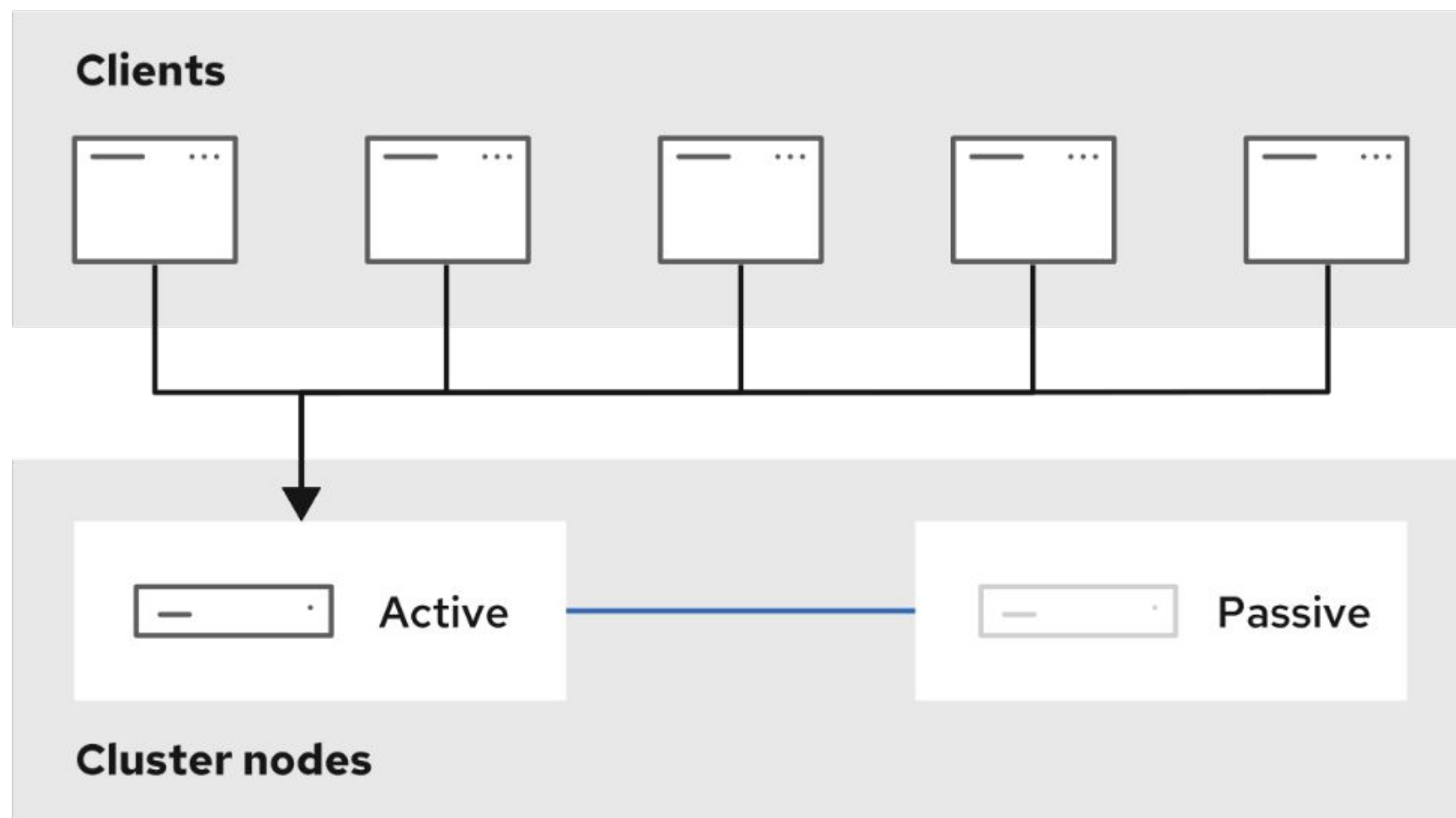
Definition Cluster

A **cluster** is a set of computers that work together on a single task. Which task is performed, and how that task is performed, differ from cluster to cluster.

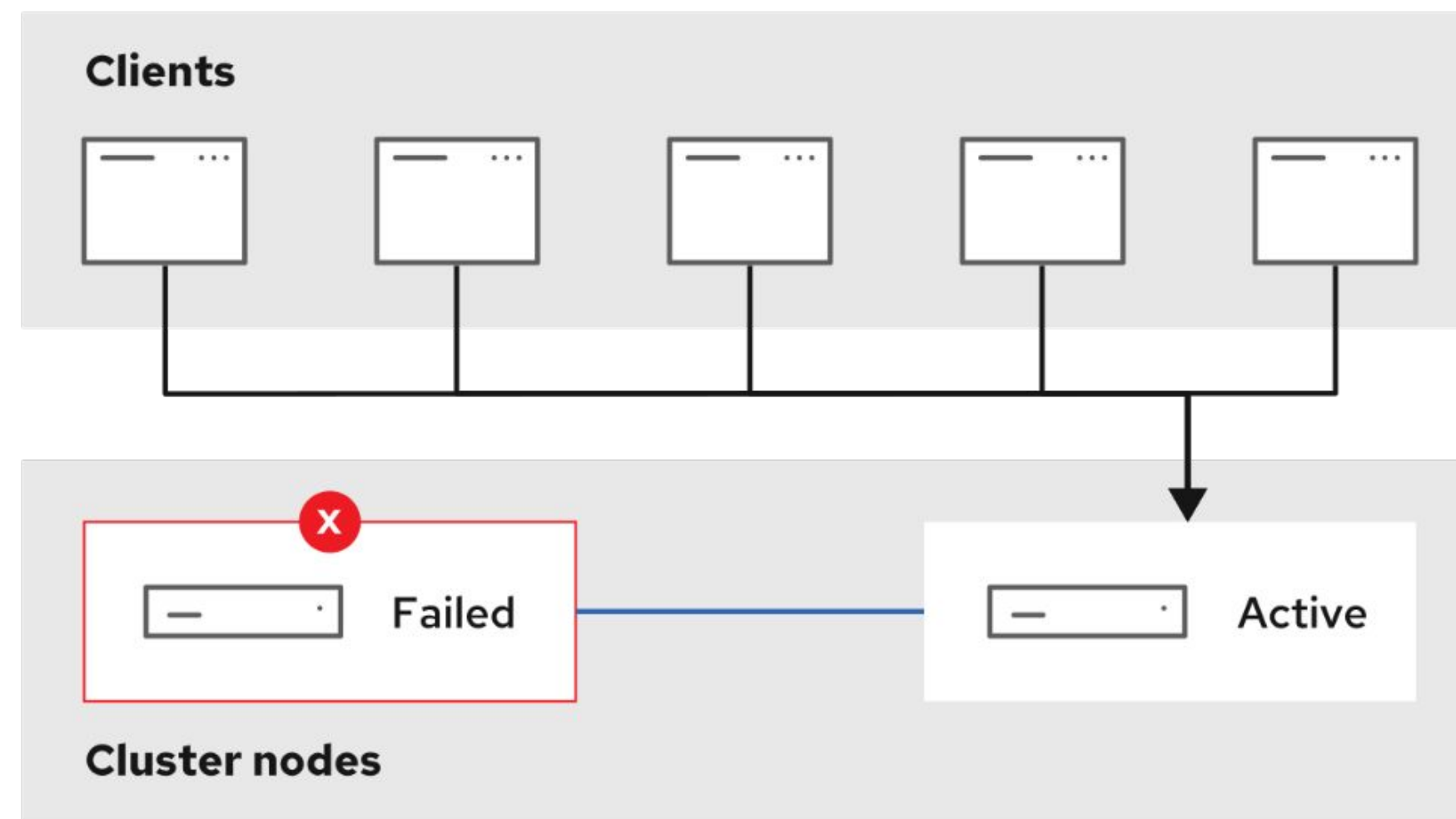
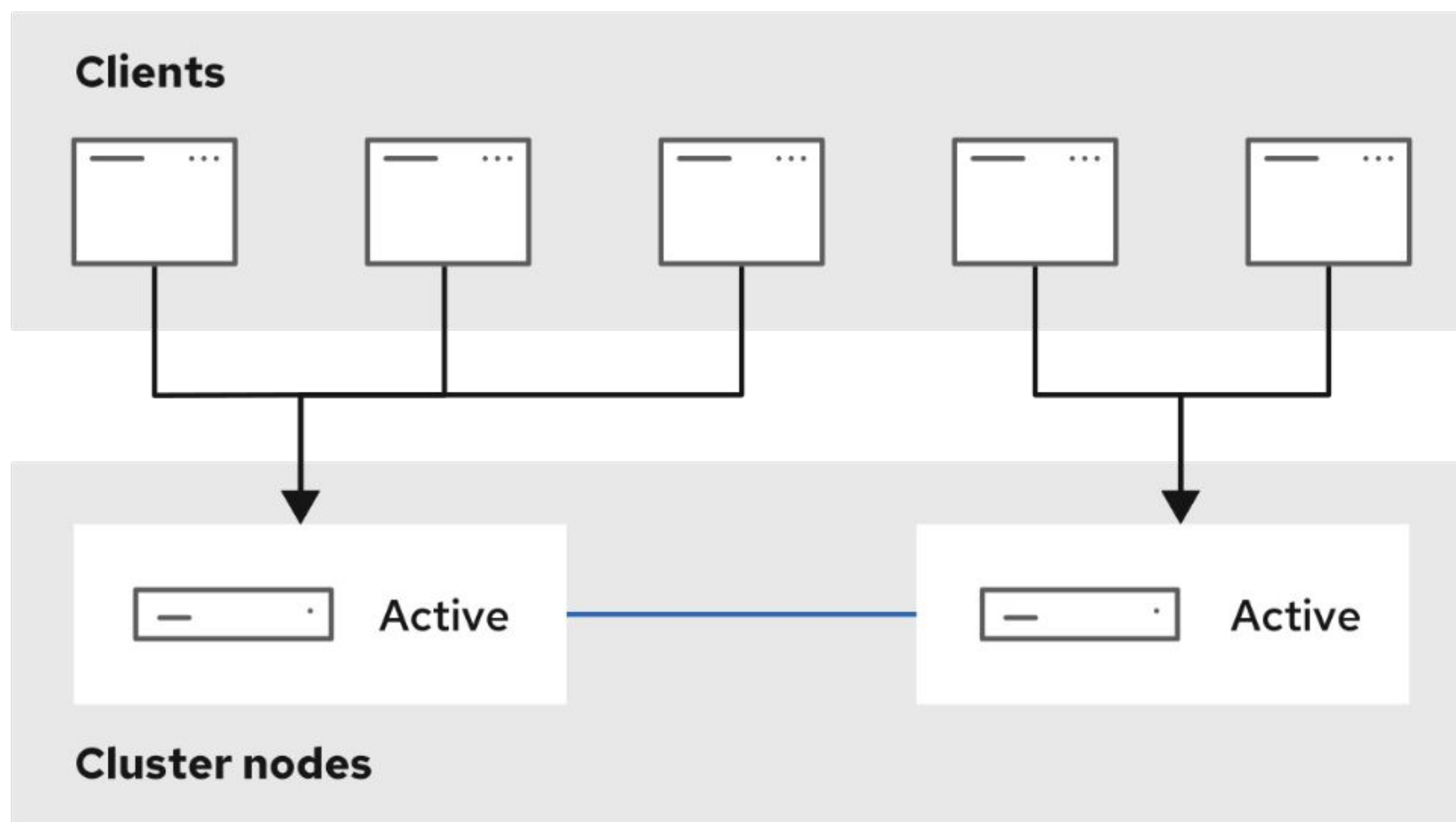
High Availability Cluster:

- keep running a service as available as possible
- no bottlenecks , no single point of failure
- keep service alive by moving it to a "healthy node"

Active-Passive HA-Cluster



Active-Active HA Cluster



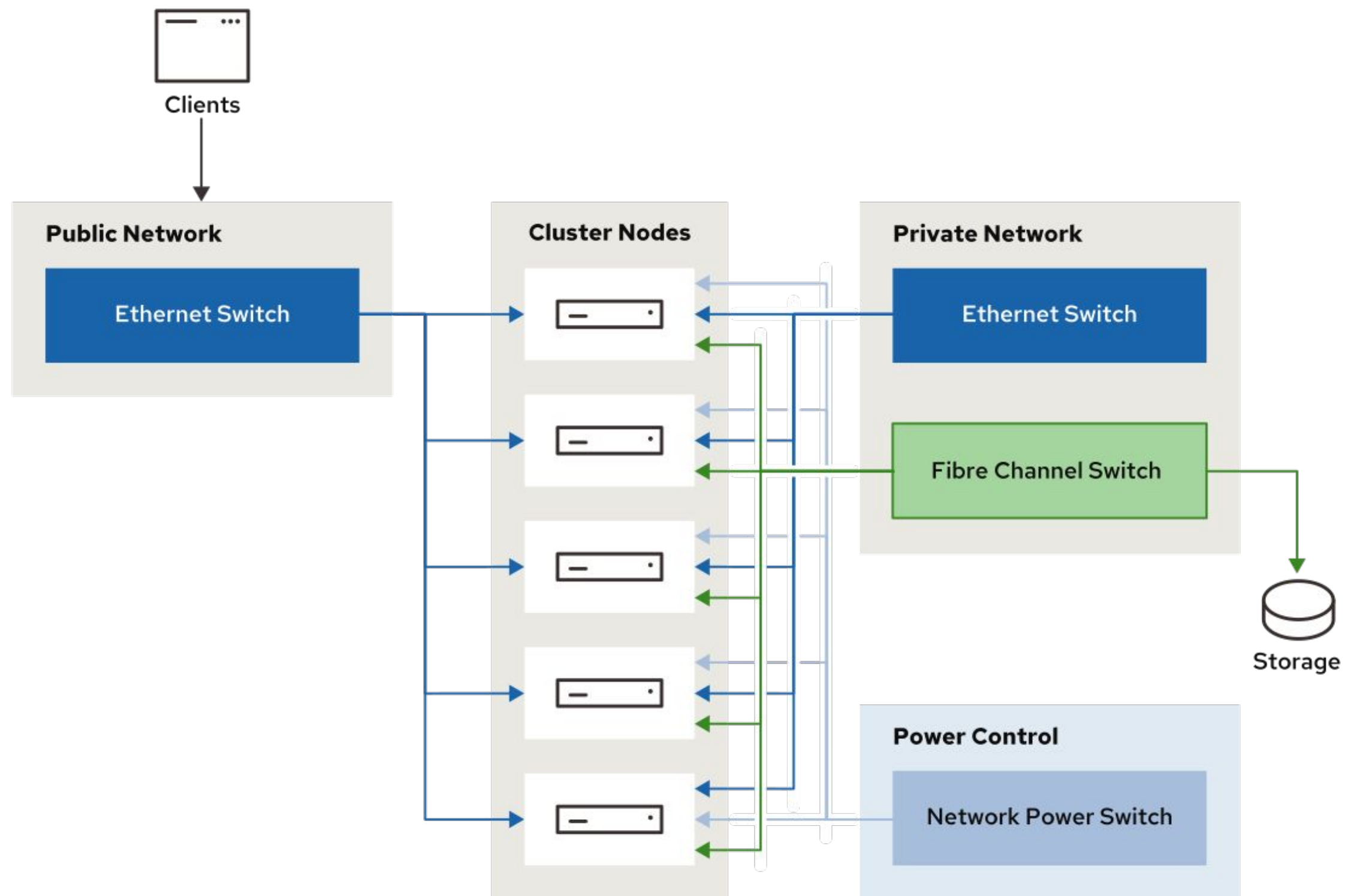
When to Use the High Availability Add-On for Clustering?

- what is the ability requirement?
- will the availability increase, when a cluster is used ?
 - does the service has built-in HA (such as DNS, LDAP)?
 - does the service w/o built-in HA benefit (e.g. NFS) ?
- Not every problem can be solved by a cluster (e.g. network, application flaw)
- a cluster involves risk management

Basic Components and Terminology

- Resources and Resource Groups
- Failover
- Fencing
- Shared Storage
- Quorum

Hardware Configuration of an HA Cluster



Software Components of Red Hat HA Cluster

- corosync
- pacemaker
 - cluster information base (CIB)
 - cluster resource management daemon (CRMd)
 - shoot the other node in the head (STONITH)
- pcs – pacemaker cluster shell
 - pcs – command line interface
 - pcsd – web frontend

Requirements and Recommendations

- number of nodes
- Single Site, Multisite, and Stretch or Geo Clusters
- Fencing
- Virtualized and Cloud Environments
- SE Linux Support

Planning for Failures

no SPOFs

Hardware Single Points of Failure

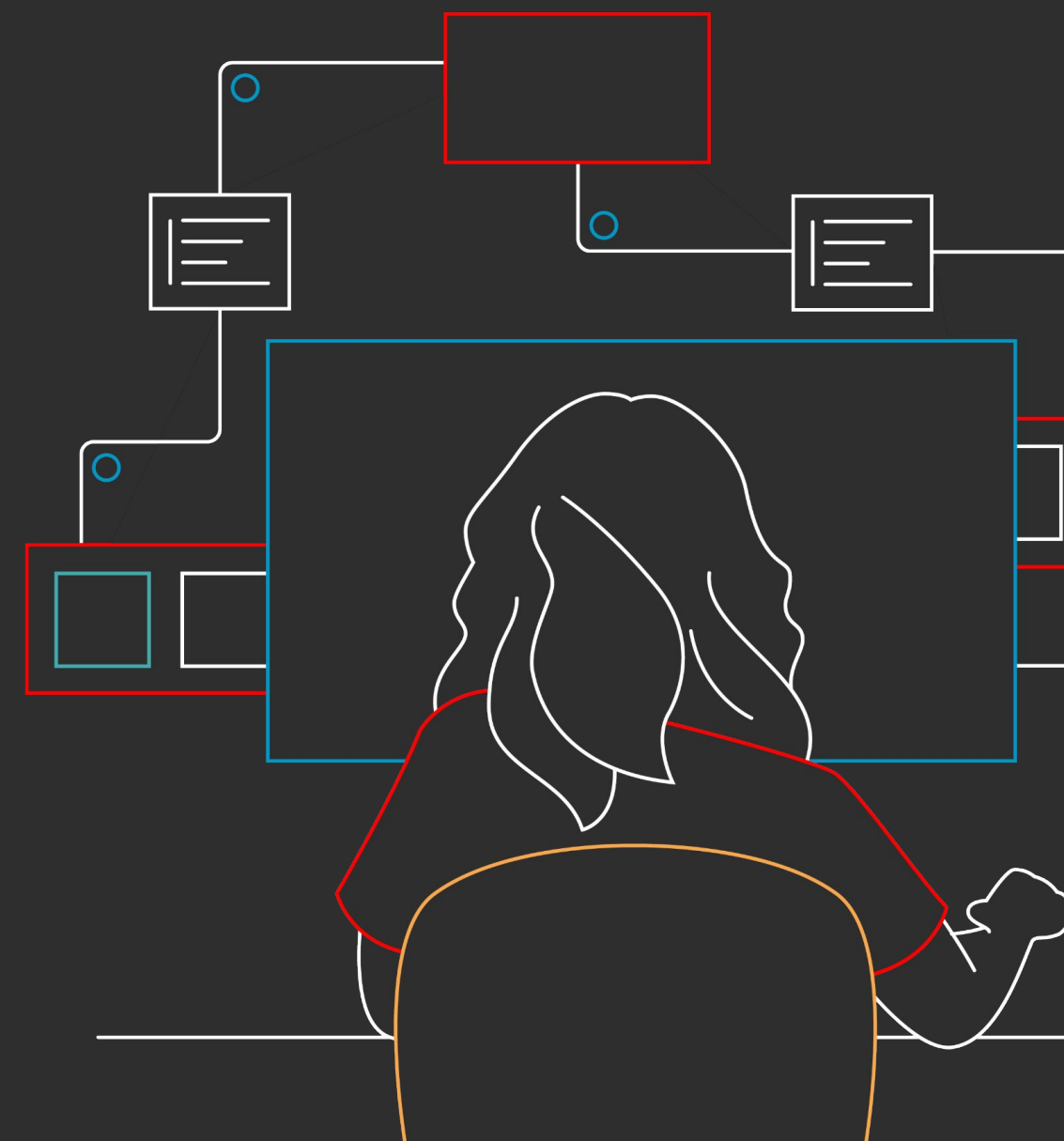
- Power supply
- Local storage
- Network interfaces
- Network switches
- Fencing software

Software Single Points of Failure

- Cluster communications
- Shared storage connection
- Software fencing configuration

Operation, Update, and **Monitoring**

**Configuring Pacemaker Cluster HA for
SAP HANA and SAP Netweaver**



Overview

step by step installation instructions - cluster framework

1. Install the cluster software on **all** nodes

```
[root@node ~]# yum install pcs fence-agents-all
```

Overview

step by step installation instructions - cluster framework

1. Install the cluster software on **all** nodes
2. Disable or configure the firewall for cluster communication

```
[root@node ~]# firewall-cmd --permanent --add-service=high-availability
```

```
[root@node ~]# firewall-cmd --reload
```

Overview

step by step installation instructions - cluster framework

1. Install the cluster software on **all** nodes
2. Disable or configure the firewall for cluster communication on **all** nodes
3. Enable pacemaker and corosync on the **all** nodes

```
[root@node ~]# systemctl enable --now pcsd
```

Overview

step by step installation instructions - cluster framework

1. Install the cluster software on **all** nodes
2. Disable or configure the firewall for cluster communication on **all** nodes
3. Enable pacemaker and corosync on the **all** nodes

```
[root@node ~]# systemctl enable --now pcsd
```

```
[root@node ~]# echo redhat | passwd --stdin hacluster
```

Overview

step by step installation instructions - cluster framework

1. Install the cluster software on **all** nodes
2. Disable or configure the firewall for cluster communication on **all** nodes
3. Enable pacemaker and corosync on the **all** nodes
4. Authenticate the cluster nodes on **one** node

```
[root@node ~]# pcs host auth node1.example.com node2.example.com
Username: hacluster
Password: redhat
node1.example.com: Authorized
node2.example.com: Authorized
```

Overview

step by step installation instructions - configure basic cluster communication

1. Set up the cluster

```
[root@node ~]# pcs cluster setup mycluster --start \  
> node1.example.com \  
> node2.example.com
```

Overview

step by step installation instructions - configure basic cluster communication

1. Set up the cluster
2. Enable auto rejoin after reboot

```
[root@node ~]# pcs cluster enable --all
```


Overview

step by step installation instructions - configure basic cluster communication

1. Set up the cluster
2. Enable auto rejoin after reboot
3. Verify the cluster status

```
[root@node ~]# pcs cluster status
```

Cluster Status:

Cluster Summary:

- * Stack: corosync
- * Current DC: node2.example.com (version 2.0.4-6.el8-2deceaa3ae) - partition with quorum
- * Last updated: Fri Mar 5 12:23:08 2021
- * Last change: Fri Mar 5 12:22:57 2021 by root via cibadmin on node1.example.com
- * 2 nodes configured
- * 0 resource instances configured

Node List:

- * Online: [node1.example.com node2.example.com]

PCSD Status:

node1.example.com: Online
node2.example.com: Online

Overview

step by step installation instructions - configure fencing

1. Select and test proper fence device and fencing method
 - a. supported fence devices: <https://access.redhat.com/articles/2881341>
 - b. how to test a fence device: <https://access.redhat.com/solutions/18803>

Overview

step by step installation instructions - configure fencing

1. Select and test proper fence device and fencing method
2. Configure fencing device

```
[root@node ~]# pcs stonith create fence_device_name fence_ipmilan \  
> pcmk_host_list=node_private_fqdn \  
> ip=node_IP_BMC \  
> username=username \  
> password=password
```

Overview

step by step installation instructions - configure fencing

1. Select and test proper fence device and fencing method
2. Configure fencing device
3. Display Status of fencing device

```
[root@node ~]# pcs stonith status
```

```
* fence_nodea (stonith:fence_ipmilan): Started node1.example.com
```

```
* fence_nodeb (stonith:fence_ipmilan): Started node2.example.com
```

Overview

step by step installation instructions - setting up HA for SAP HANA

1. Ensure HANA SR is configured and working properly

SAP HANA Topolgy Resource Agent

Required Parameters

- SID
- InstanceNumber

Responsibilities

- Gathers information about the status and configuration of the SAP HANA System Replication on each node.
- Starts and monitors the local SAP HostAgent, which is required for starting, stopping, and monitoring the SAP HANA instances.

Overview

step by step installation instructions - setting up HA for SAP HANA

1. Ensure HANA SR is configured and working properly
2. Configure SAP Hana topology clone resource

```
[root@node ~]# pcs resource create SAPHanaTopology_<SID>_<InstanceNumber> \  
> SAPHanaTopology SID=<SID> InstanceNumber=<InstanceNumber> \  
> op start timeout=600 op stop timeout=300 \  
> op monitor interval=10 timeout=600 \  
> clone clone-max=2 clone-node-max=1 interleave=true
```

SAP HANA Resource Agent (Scale Up)

Parameters

- SID (required)
- InstanceNumber (required)
- PREFER_SITE_TAKEOVER
- AUTOMATED_REGISTER
- DUPLICATE_PRIMARY_TIMEOUT

Responsibilities

- managing HANA instances and monitors HSR
- can trigger an SR takeover

Overview

step by step installation instructions - setting up HA for SAP HANA

1. Ensure HANA SR is configured and working properly
2. Configure SAP Hana topology clone resource
3. Create Master/Slave SAPHana resource

```
[root@node ~]# pcs resource create SAPHana_<SID>_<InstanceNumber> SAPHana \  
> SID=<SID> InstanceNumber=<InstanceNumber> \  
> PREFER_SITE_TAKEOVER=true \  
> DUPLICATE_PRIMARY_TIMEOUT=7200 AUTOMATED_REGISTER=true \  
> op start timeout=3600 op stop timeout=3600 op monitor interval=61 \  
> role="Slave" timeout=700 op monitor interval=59 \  
> role="Master" timeout=700 op promote timeout=3600 op demote timeout=3600 \  
> promotable meta notify=true clone-max=2 clone-node-max=1 interleave=true
```

Overview

step by step installation instructions - setting up HA for SAP HANA

1. Ensure HANA SR is configured and working properly
2. Configure SAP Hana topology clone resource
3. Create Master/Slave SAPHana resource
4. Create Virtual IP Address Resource

```
[root@node ~]# pcs resource create vip_<SID>_<InstanceNumber> \  
> IPaddr2 ip="192.168.0.15"
```

Overview

step by step installation instructions - setting up HA for SAP HANA

1. Ensure HANA SR is configured and working properly
2. Configure SAP Hana topology clone resource
3. Create Master/Slave SAPHana resource
4. Create Virtual IP Address Resource
5. Create Constraints

```
# pcs constraint order SAPHanaTopology_<SID>_<InstanceNumber>-clone \  
> then SAPHana_<SID>_<InstanceNumber>-clone symmetrical=false  
# pcs constraint colocation add vip_<SID>_<InstanceNumber> \  
> with master SAPHana_<SID>_<InstanceNumber>-clone 2000
```

Overview

step by step installation instructions - additional config for Active/Active HANA SR setup

1. Creating the resource for managing the secondary virtual IP address

```
[root@node ~]# pcs resource create vip2_<SID>_<InstanceNumber> \  
> IPaddr2 ip="192.168.1.11"
```

Overview

step by step installation instructions - additional config for Active/Active HANA SR setup

1. Creating the resource for managing the secondary virtual IP address
2. Create constraints

```
[root@node ~]# pcs constraint location vip2_<SID>_<InstanceNumber> \  
> rule score=INFINITY hana_<sid>_sync_state eq SOK and hana_<sid>_roles \  
> eq 4:S:master1:master:worker:master  
[root@node ~]# pcs constraint location vip2_<SID>_<InstanceNumber> \  
> rule score=2000 hana_<sid>_sync_state eq PRIM and hana_<sid>_roles eq \  
> 4:P:master1:master:worker:master
```

Role details: configure pacemaker for SAP HANA

**redhat.sap_install.
sap_hana_ha_pacemaker**

This role configures pacemaker on two SAP HANA systems that have properly configured SAP HANA system replication deployment on a RHEL 8.x systems.

Example variables to be used for both hosts:

```
sap_hana_sid: RH1
sap_instance_number: "00"
ha_cluster_cluster_name: hanaccluster
ha_cluster_hacluster_password: 'S3cr3tP@ssw0rd' # notsecret
sap_hana_vip:
  primary: 10.0.0.202
sap_ha_pacemaker_cluster_stonith_custom:
  - name: "fence_with_hmc"
    agent: "stonith:fence_lpar"
    options:
      [...]
```

This role is TechPreview, hence parameters are subject to change

Role update: configure pacemaker for SAP HANA

**redhat.sap_install.
sap_hana_ha_pacemaker**

There has been an interface change in the current role.

The cluster Setup now needs to look like this:

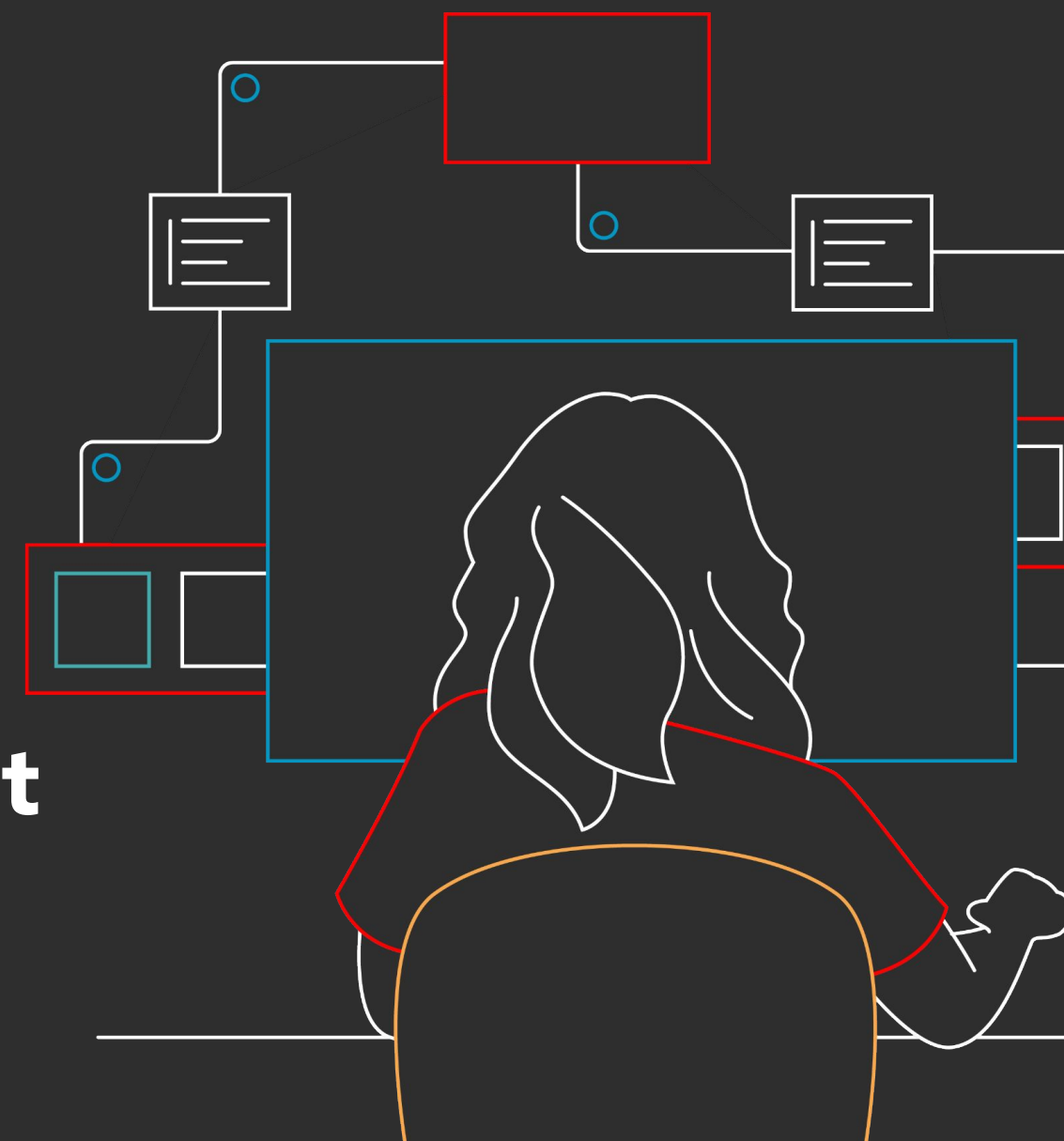
```
sap_hana_sid: RH1
sap_instance_number: "00"
ha_cluster_cluster_name: hanaccluster
ha_cluster_hacluster_password: 'S3cr3tP@ssw0rd' # notsecret
sap_ha_pacemaker_cluster_vip_hana_primary_ip_address: 10.0.0.202
sap_ha_pacemaker_cluster_stonith_custom:
  - name: "fence_with_hmc"
    agent: "stonith:fence_lpar"
    options:
      [...]
```

See also:
<https://access.redhat.com/solutions/3786791>

LAB: Configure SAP Cluster

Explaining SAP HANA **System Replication**

Installing the SAP HANA Scale-out Resource Agent

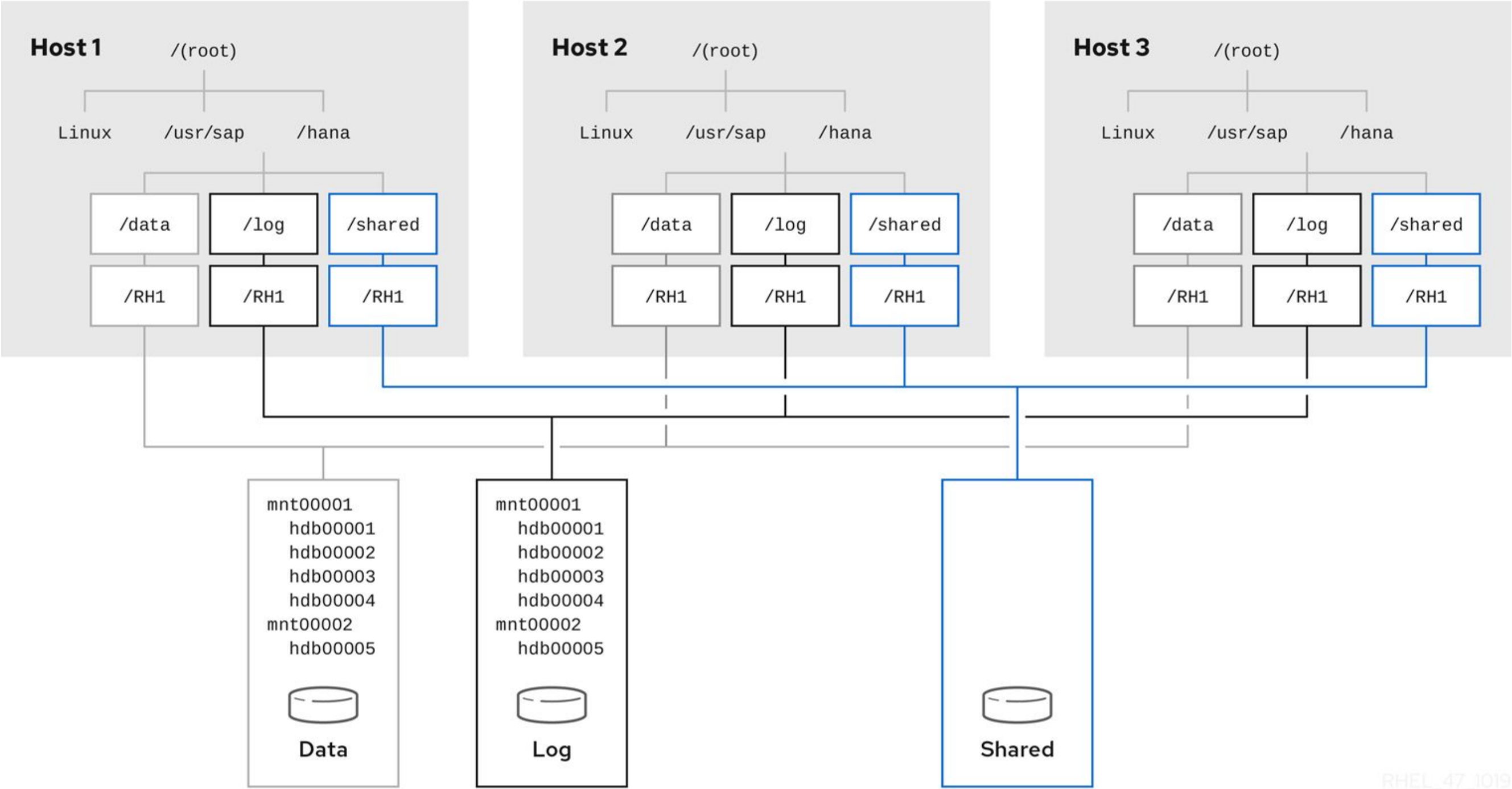


Overview

differences to scale-up HSR

- more than 4-nodes in a cluster
- additional shared mount points
- additional majority make node
- additional constraints on the majority maker node

HANA Scale-Out Shared Storage Architecture



RHEL_47_1019

Pacemaker Integration

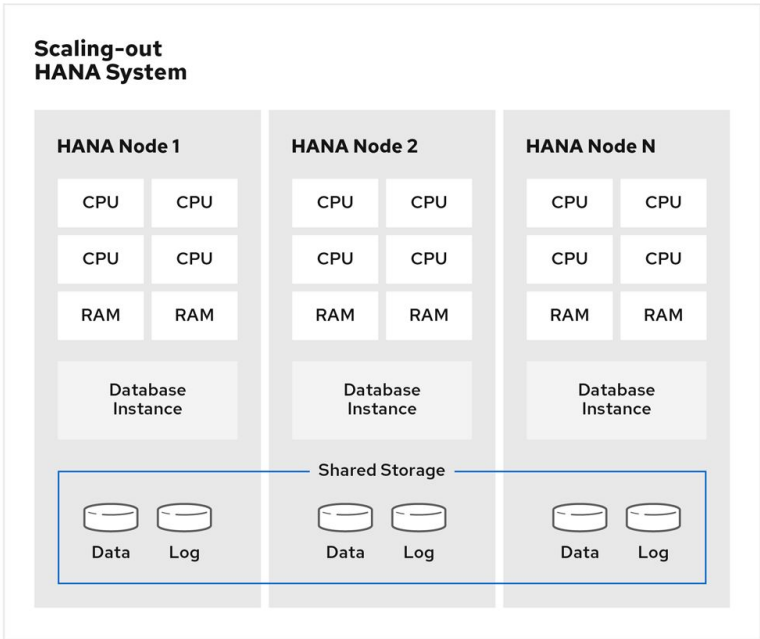
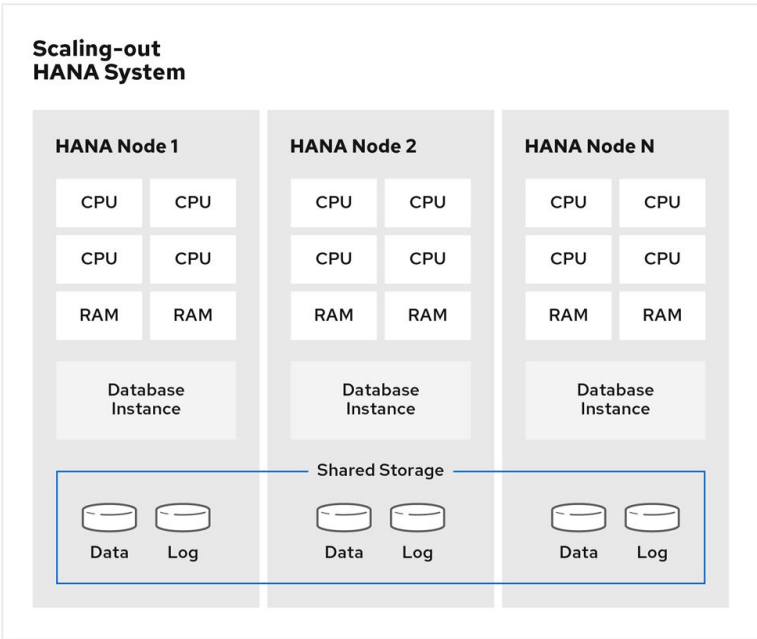
Hana Installed and SR established

DC1

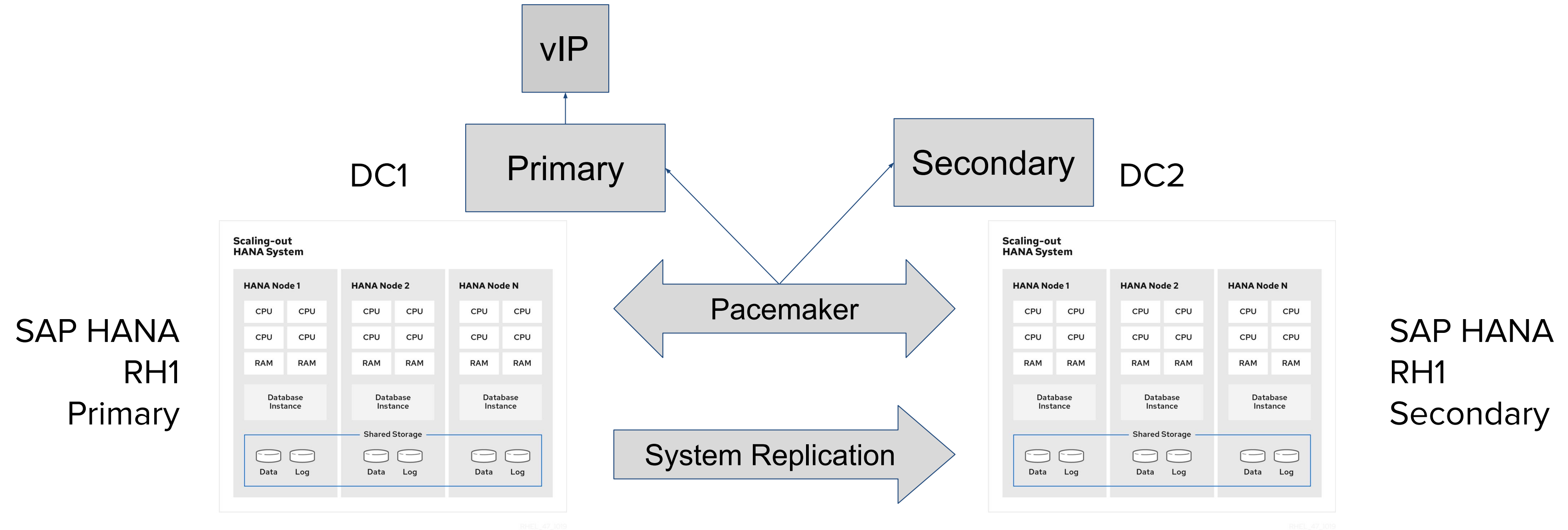
DC2

SAP HANA
RH1
Primary

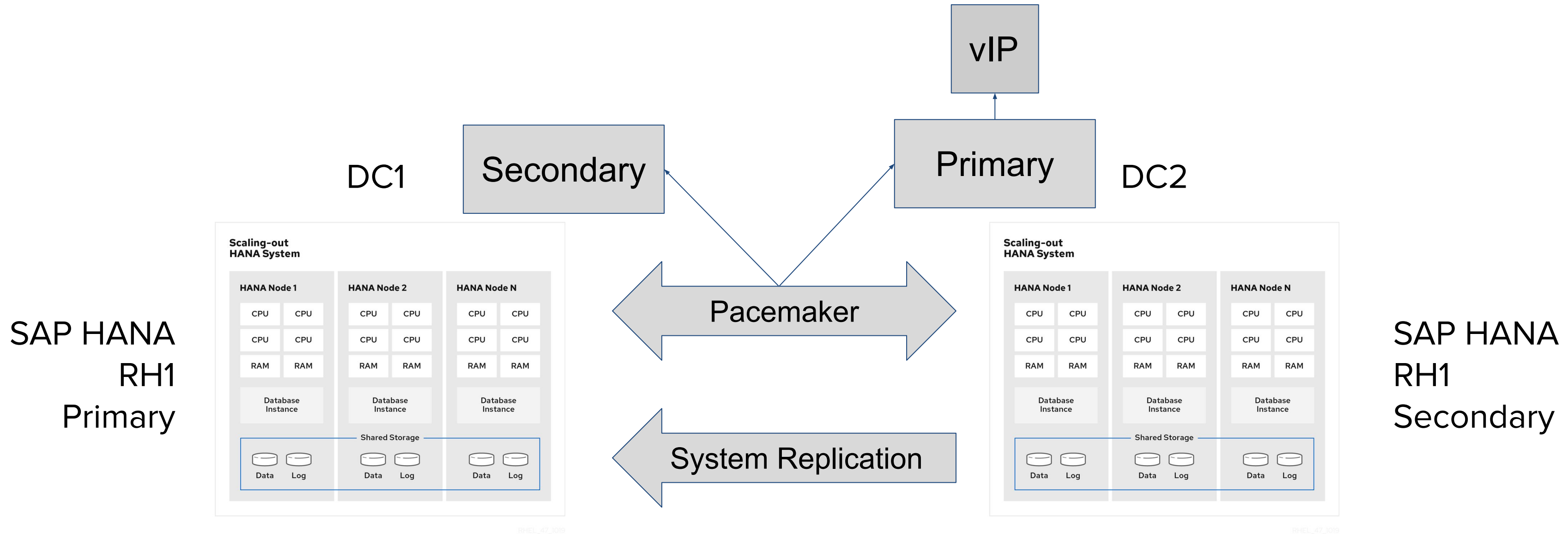
SAP HANA
RH1
Secondary



Pacemaker Integration



Pacemaker Integration

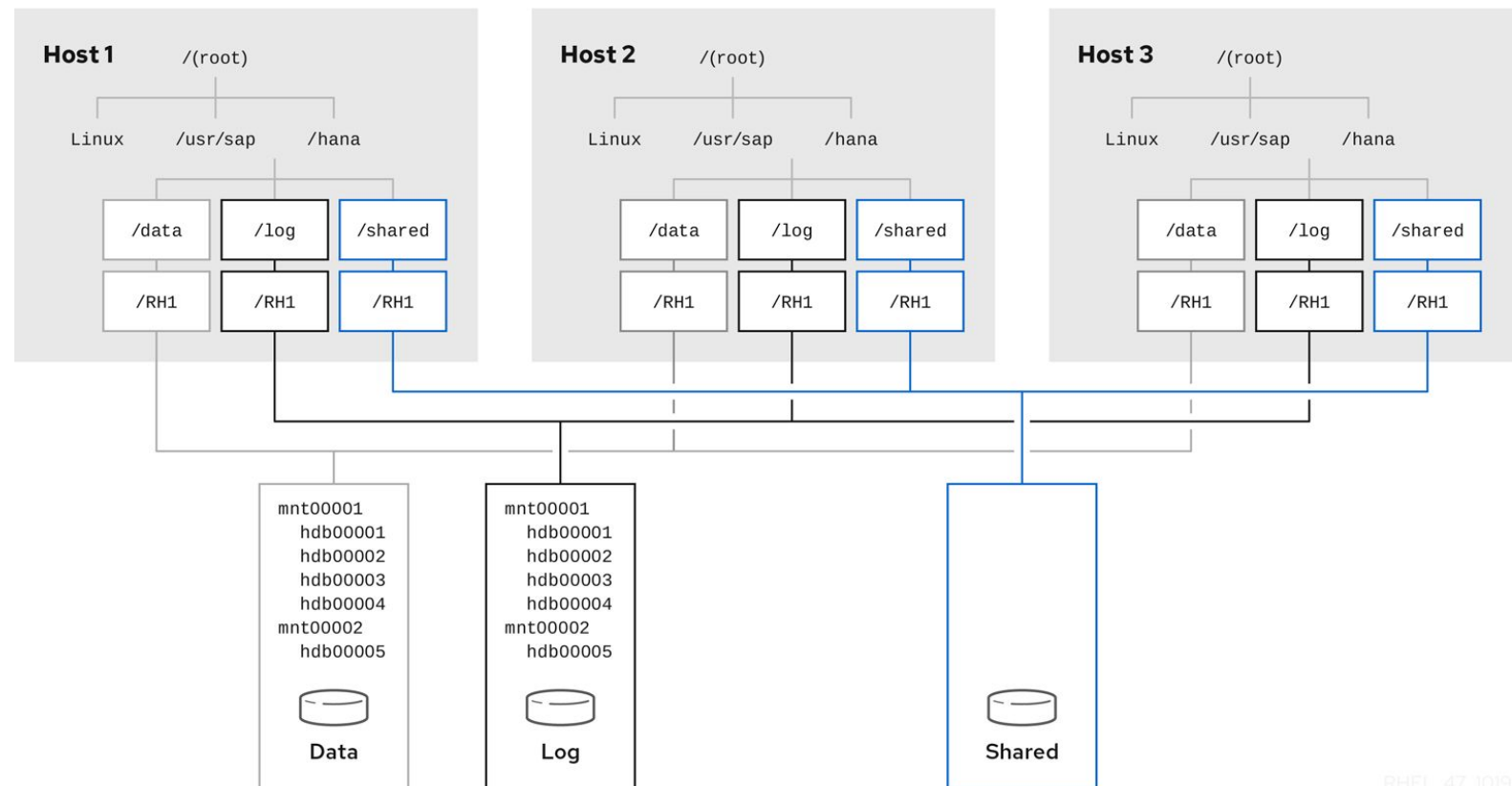


SAP HANA Scale-Out Pacemaker Agents

- Resource Agents
 - SAPHanaTopology
 - Gathering all needed Information
 - Set the Scoring for the Cluster configuration
 - Clone Resource
 - SAPHanaController
 - Set the State of the Environment
 - IP, System Replication, Change State
 - Master/Slave Resource

SAP HANA Types

- Master
 - Master “Nameserver”
 - up to three
- Worker
 - Could be a Master node
 - Name- & Index-Server
- Standby
 - Waiting for orders
 - Has no Storage Attachment



SAPHanaSR hook

```
root@dc1hana01:/hana/shared
Datei Bearbeiten Ansicht Suchen Terminal Hilfe
[root@dc1hana01 shared]# SAPHanaSR-showAttr
Global cib-time          prim sec srHook sync_state
-----
global Fri Jan 24 17:23:30 2020 DC1  DC2 SOK      SOK

Sit lpt          lss mns          srr
-----
DC1 1579904610 4    dc1hana01 P
DC2 30          4    dc2hana01 S

Hosts          clone_state node_state roles          score  site
-----
dc1hana01      PROMOTED   online   master1:master:worker:master 150    DC1
dc1hana02      DEMOTED   online   slave:slave:worker:slave     -10000 DC1
dc1hana03      DEMOTED   online   master2:slave:worker:slave    110    DC1
dc1hana04      DEMOTED   online   master3:slave:standby:standby 115    DC1
dc2hana01      DEMOTED   online   master2:master:worker:master  100    DC2
dc2hana02      DEMOTED   online   slave:slave:worker:slave     -12200 DC2
dc2hana03      DEMOTED   online   master3:slave:worker:slave    80     DC2
dc2hana04      DEMOTED   online   master1:slave:standby:standby 80     DC2
majoritymaker  online
```

```
[root@dc1hana01 shared]#
```

Landscape Overview

Each Data Center has its own Overview

```
root@dc1hana01:/hana/shared
Datei Bearbeiten Ansicht Suchen Terminal Hilfe
rh1adm@dc1hana01:/usr/sap/RH1/HDB00> HDBSettings.sh landscapeHostConfiguration.py
```

Host	Host Active	Host Status	Failover Status	Remove Status	Storage Config Partition	Storage Actual Partition	Failover Config Group	Failover Actual Group	NameServer Config Role	NameServer Actual Role	IndexServer Config Role	IndexServer Actual Role	Host Config Roles	Host Actual Roles	Worker Config Groups	Worker Actual Groups
dc1hana01	yes	ok			1	1	default	default	master 1	master	worker	master	worker	worker	default	default
dc1hana02	yes	ok			3	3	default	default	slave	slave	worker	slave	worker	worker	default	default
dc1hana03	yes	ok			2	2	default	default	master 2	slave	worker	slave	worker	worker	default	default
dc1hana04	yes	ignore			0	0	default	default	master 3	slave	standby	standby	standby	standby	default	-

```
overall host status: ok
rh1adm@dc1hana01:/usr/sap/RH1/HDB00>
```


System Replication Overview

Complete Overview of all replication states available on primary Environment

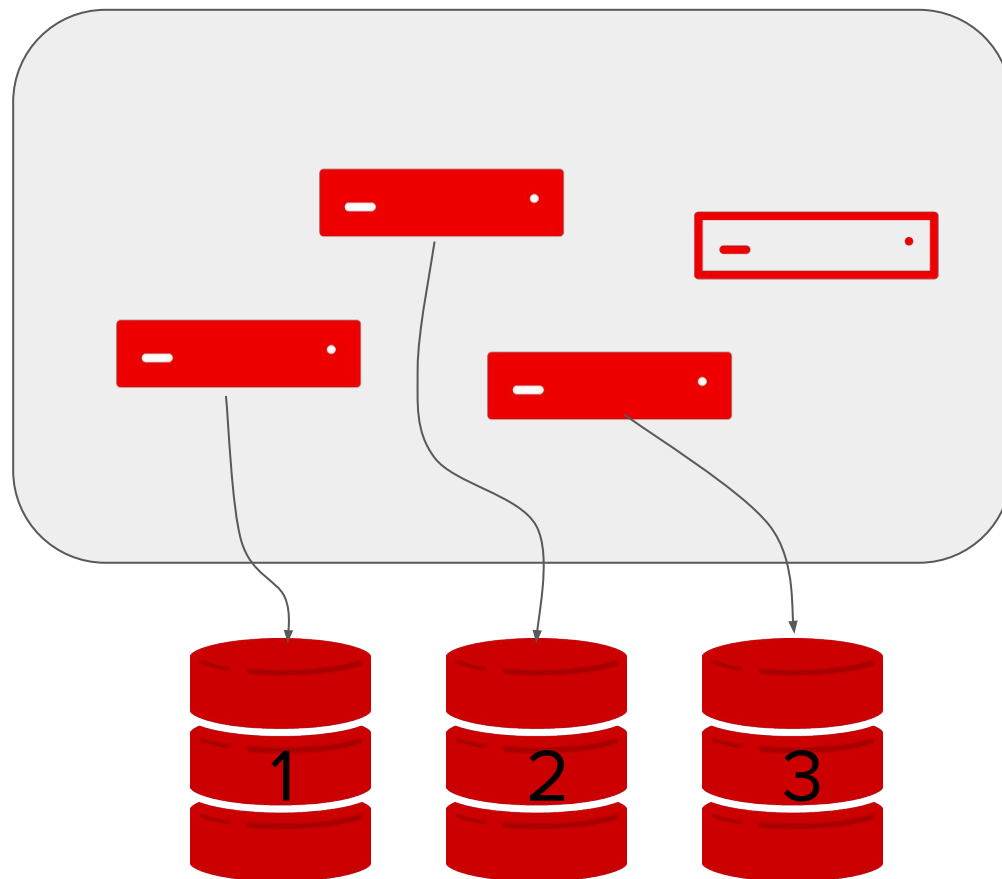
```
root@dc1hana01:/hana/shared
Datei Bearbeiten Ansicht Suchen Terminal Hilfe
rh1adm@dc1hana01:/usr/sap/RH1/HDB00> HDBSettings.sh systemReplicationStatus.py
| Database | Host | Port | Service Name | Volume ID | Site ID | Site Name | Secondary Host | Secondary Port | Secondary Site ID | Secondary Site Name | Secondary Active Status | Replication Mode | Replication Status | Replication Status Details |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| SYSTEMDB | dc1hana01 | 30001 | nameserver | 1 | 1 | DC1 | dc2hana01 | 30001 | 2 | DC2 | YES | SYNC | ACTIVE |
| RH1 | dc1hana01 | 30007 | xsengine | 2 | 1 | DC1 | dc2hana01 | 30007 | 2 | DC2 | YES | SYNC | ACTIVE |
| RH1 | dc1hana01 | 30003 | indexserver | 3 | 1 | DC1 | dc2hana01 | 30003 | 2 | DC2 | YES | SYNC | ACTIVE |
| RH1 | dc1hana03 | 30003 | indexserver | 4 | 1 | DC1 | dc2hana03 | 30003 | 2 | DC2 | YES | SYNC | ACTIVE |
| RH1 | dc1hana02 | 30003 | indexserver | 5 | 1 | DC1 | dc2hana02 | 30003 | 2 | DC2 | YES | SYNC | ACTIVE |

status system replication site "2": ACTIVE
overall system replication status: ACTIVE

Local System Replication State
~~~~~
mode: PRIMARY
site id: 1
site name: DC1
rh1adm@dc1hana01:/usr/sap/RH1/HDB00>
```

SAP HANA Scale-Out explained

Worker and Stand-by Nodes



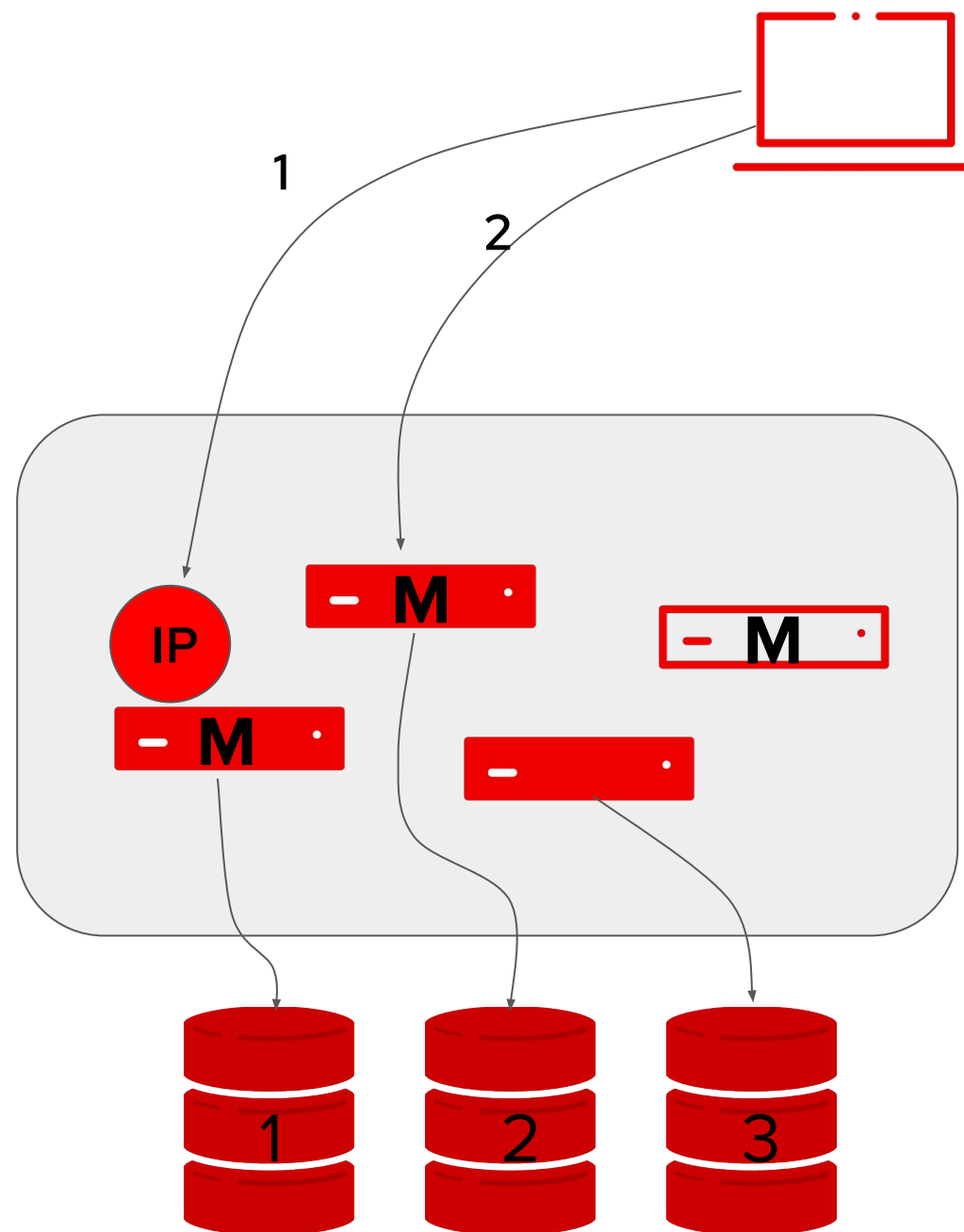
An SAP HANA Scaleout database consists of multiple nodes and SAP HANA instances.

Each worker node has it's own data partition.

Standby nodes do not have a data partition

SAP HANA Scale-Out explained

Master and Slave Nodes



An SAP HANA Scaleout database consists of several services such as master nameserver (M).

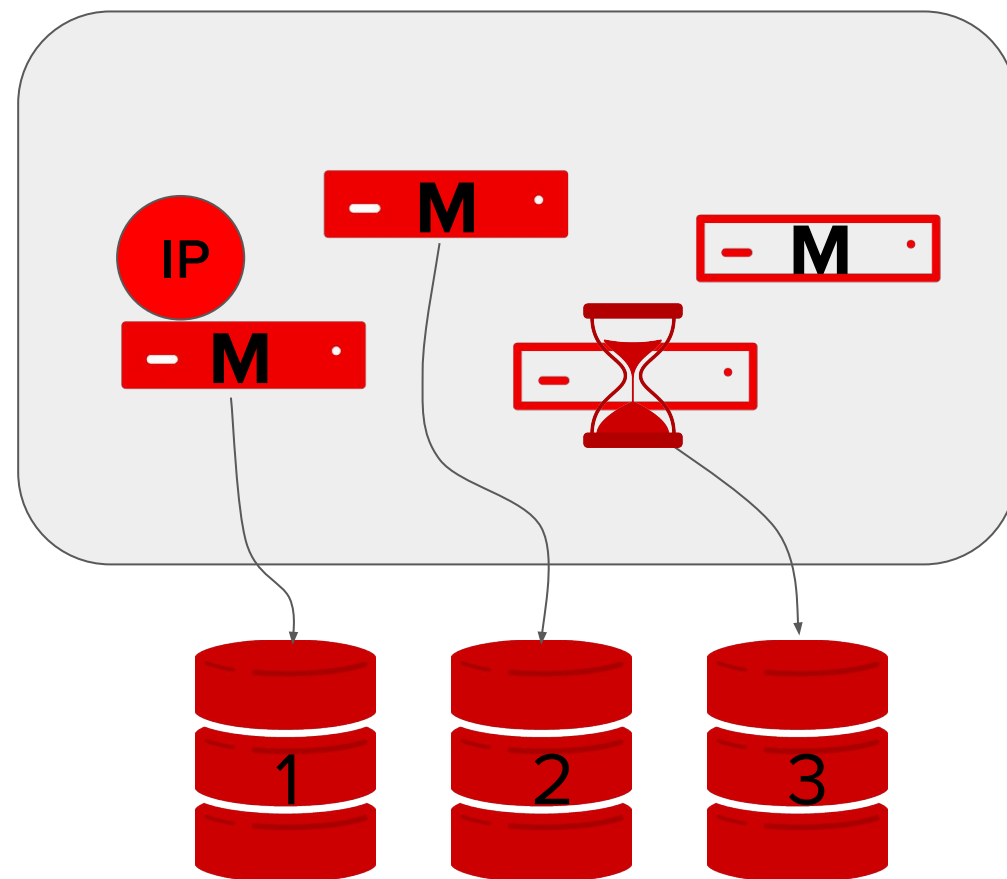
The active master nameserver takes all client connections and redirect the client to the proper worker node. It always has data partition 1.

Master candidates could be worker or standby nodes.

Typically there are 3 nodes which could get active master nameserver

SAP HANA Scale-Out - Worker Node Failure

Failing Worker Node or Instance



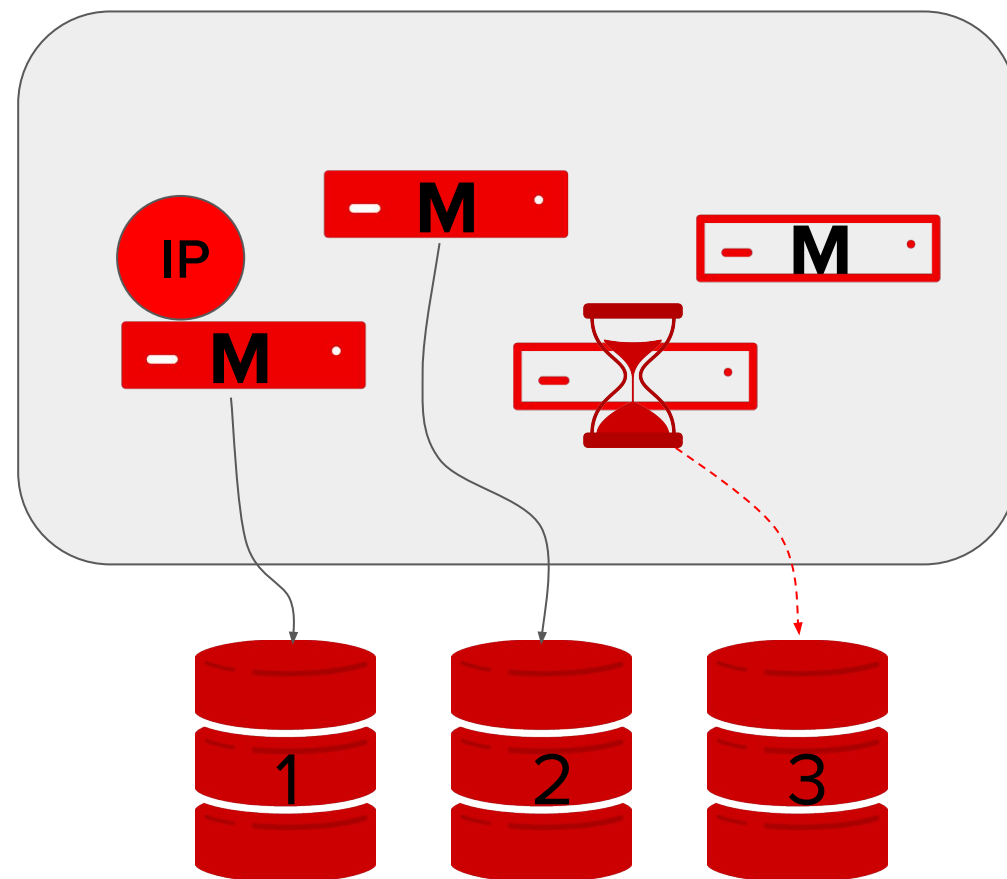
if a normal worker node failed, client could still connect to the SAP HANA database.

However answers which need data of the failed node could not be processed

SAP HA tries to repair this situation using a standby node.

SAP HANA Scale-Out - Worker Node Failure

Failing Worker Node or Instance

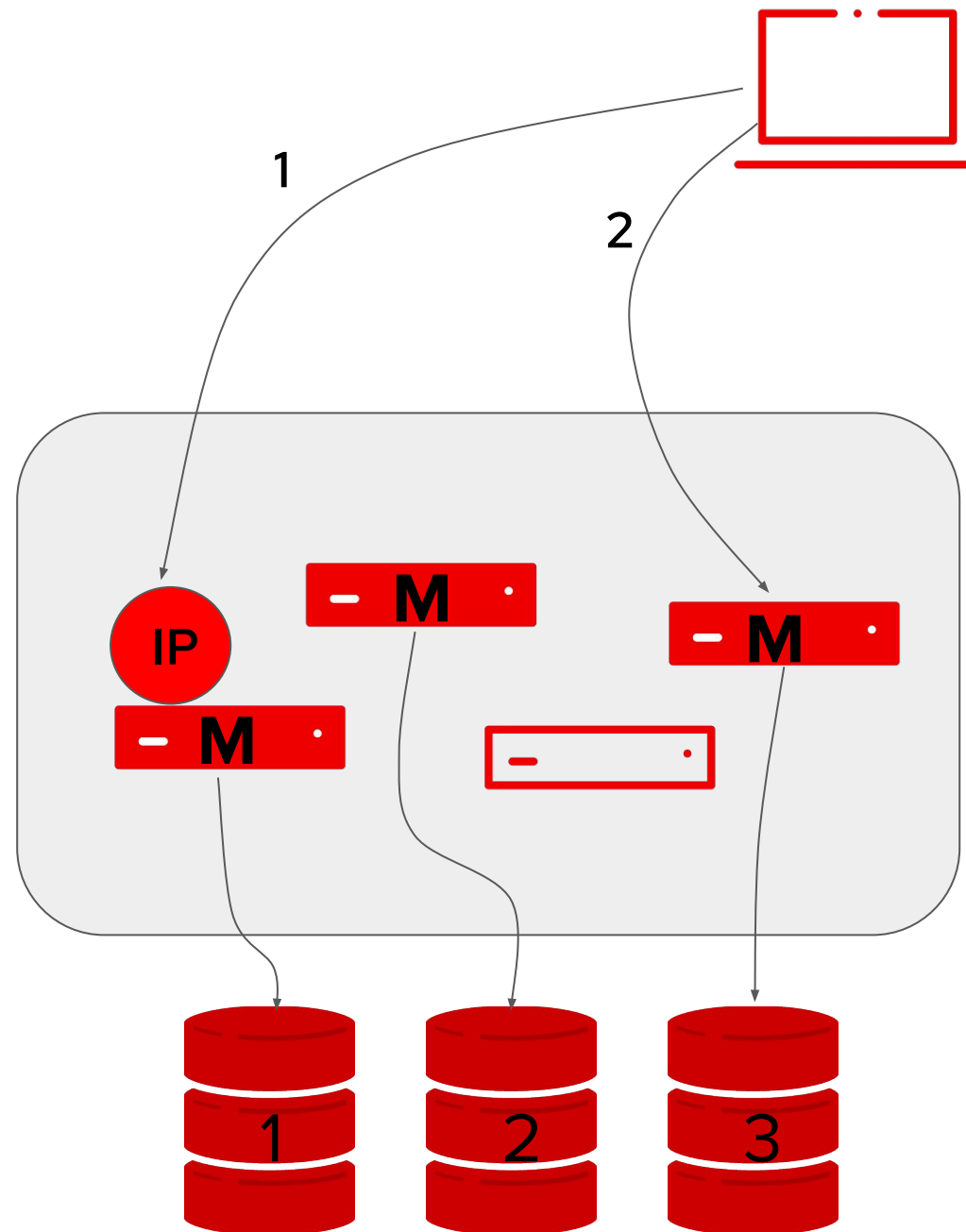


first of all the SAP HANA HA storage API must guarantee, that the old node does not longer have access to the data (SAP STONITH)

After the data partition is "free" the failover could be processed

SAP HANA Scale-Out - Worker Node Failure

Failing Worker Node



Any available standby node could take the “lost” data partition

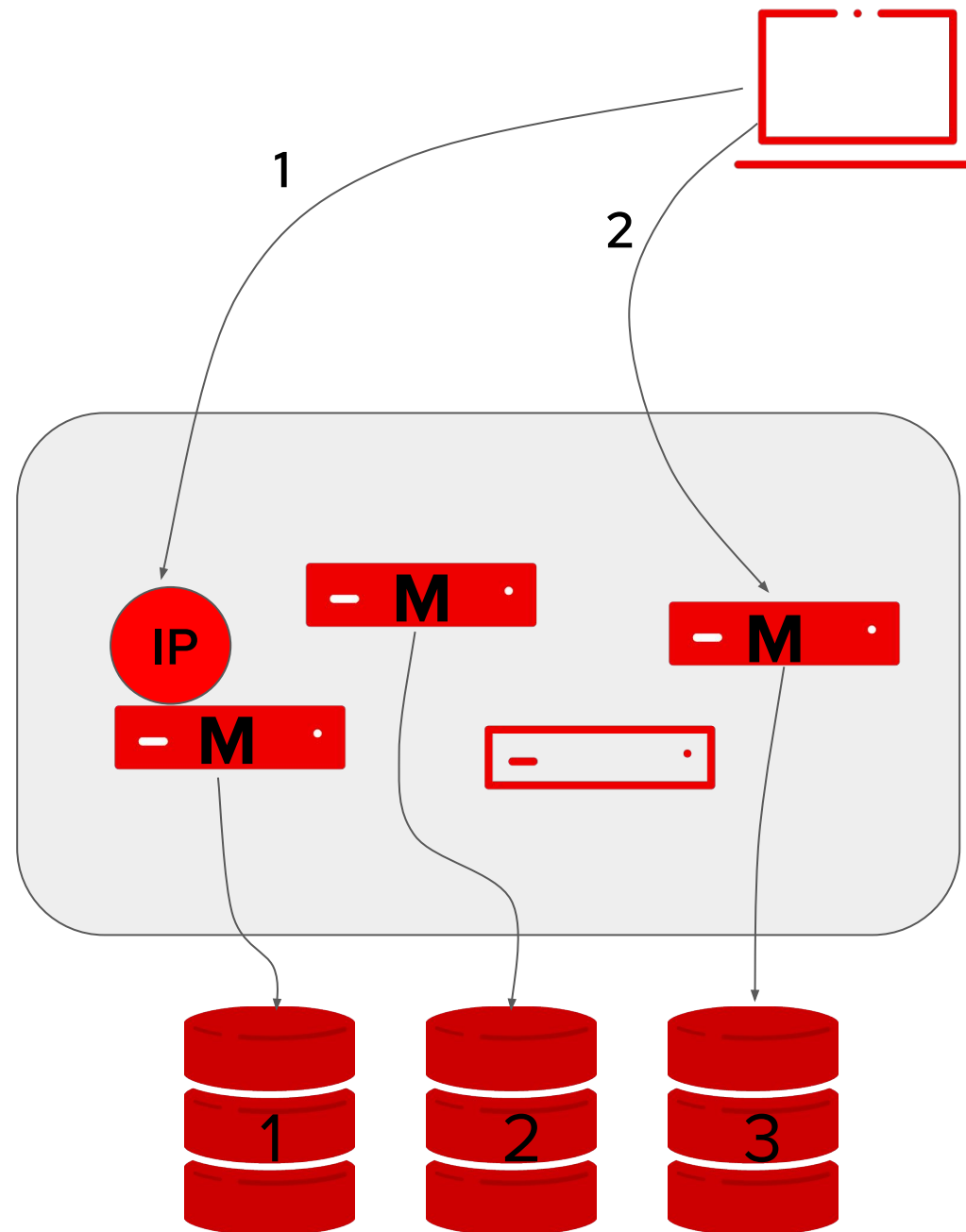
The standby node is now a worker node and loads the data.

The active master nameserver will now redirect clients to the new node.

the old worker will be a standby node once available again

SAP HANA Scale-Out - Worker Node Failure

Failing Worker Node



Summary:

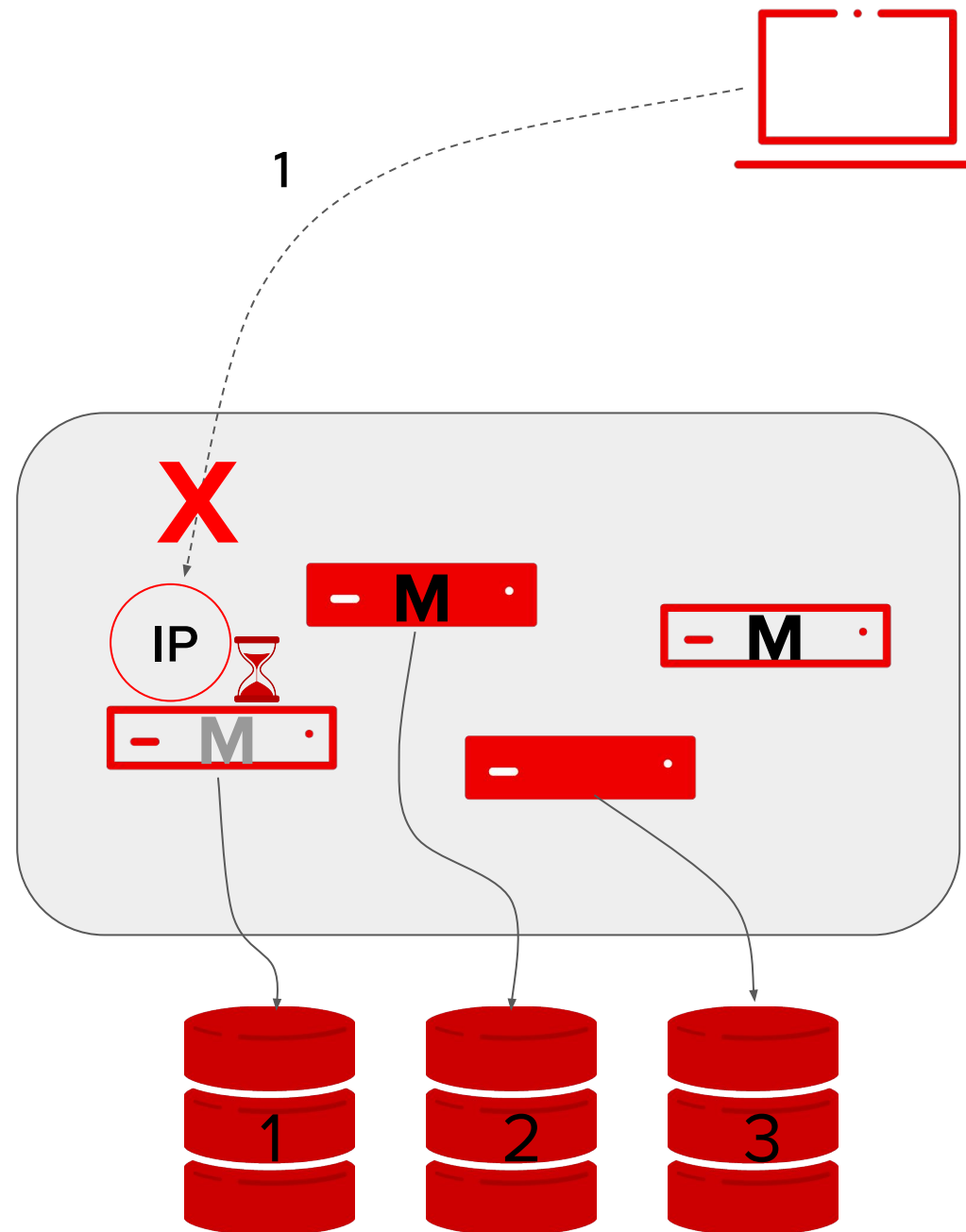
SAPHanaSR detects all failovers of worker nodes

SAPHanaSR checks the overall landscape status of the SAP HANA database

SAPHanaSR "follows" the decision of the SAP HA and checks, if the failover is successful

SAP HANA Scale-Out - Master Node failure

Failing Master node

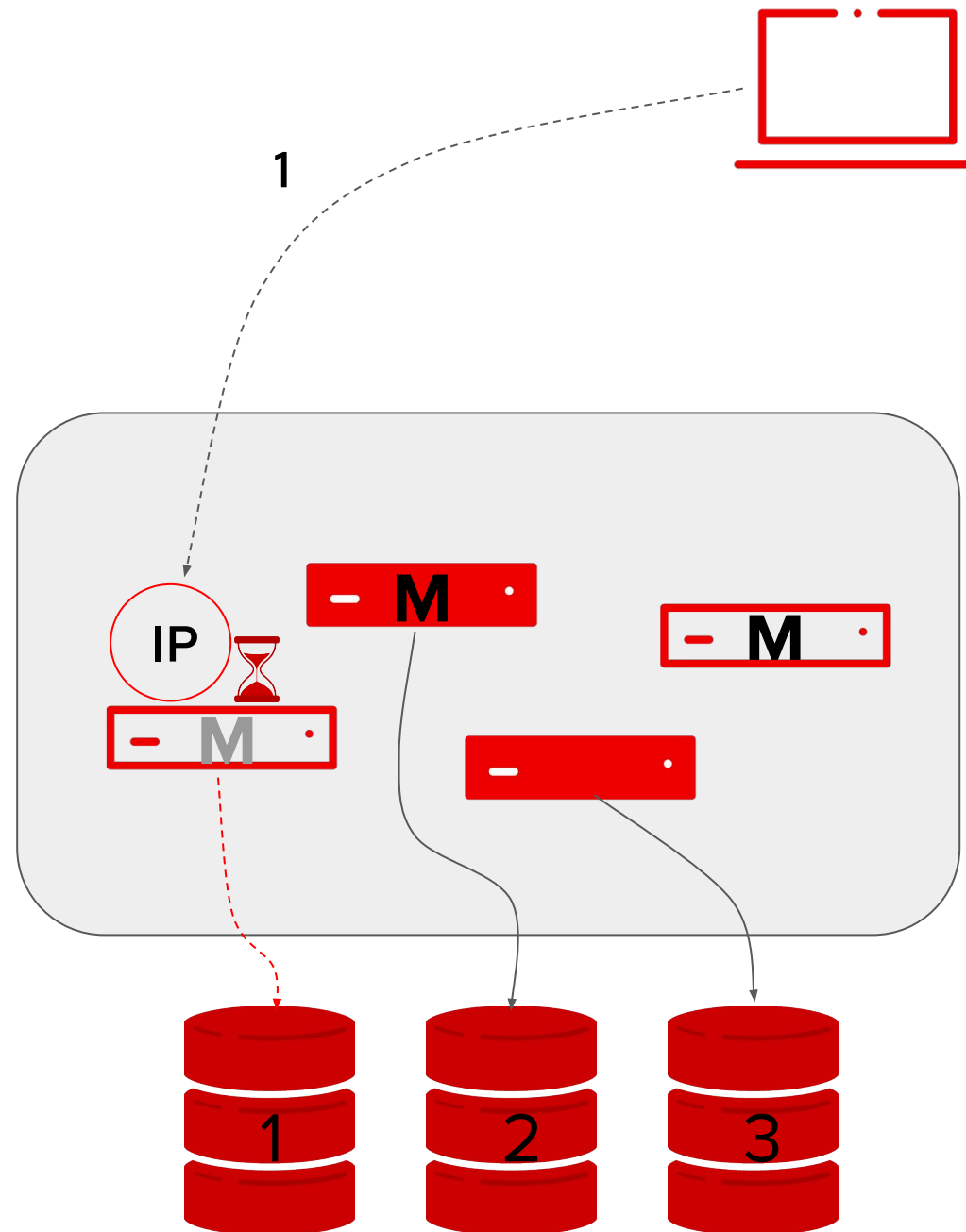


The active master nameserver is failing. All client connections are blocked.

As the active master nameserver is also a worker node SAP HA needs to failover the active master role including the worker part.

SAP HANA Scale-Out - Master Node failure

Failing Master node



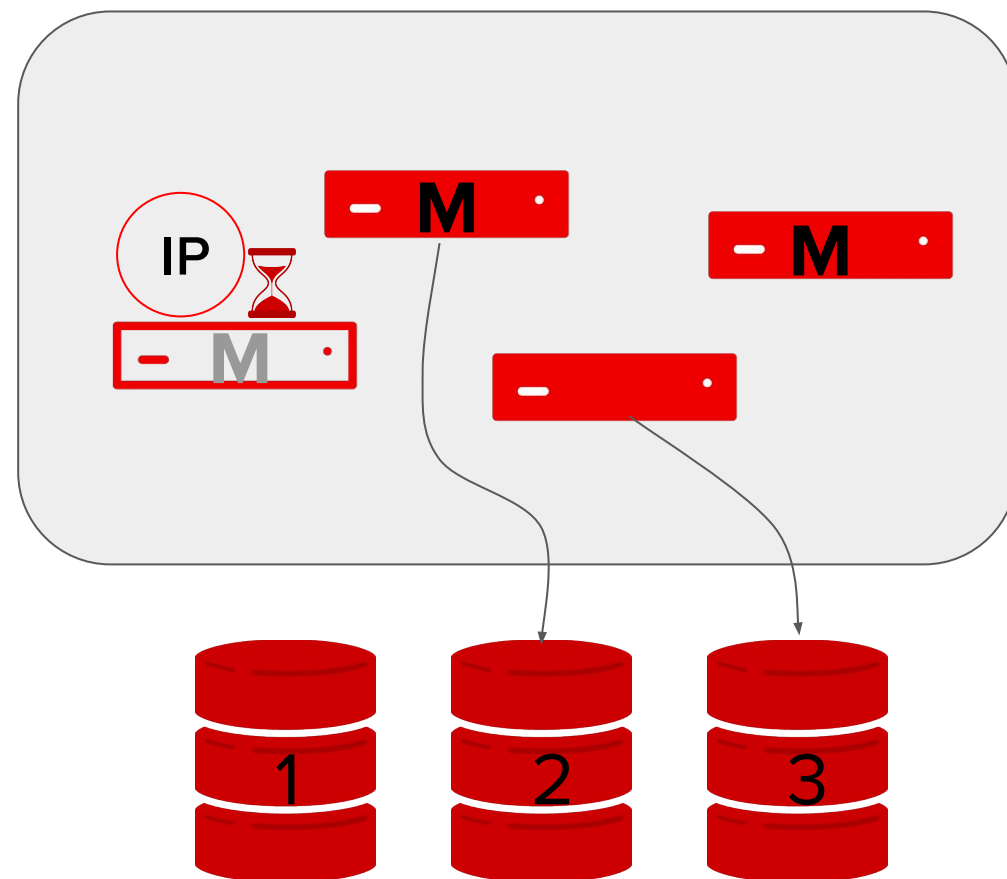
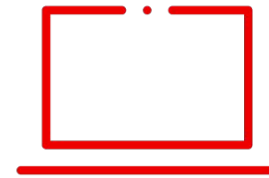
The data partition 1 needs to be released (SAP STONITH).

One of the master nameserver candidates try to failover the active master nameserver role

In best case this should be a standby node because otherwise it's data partition would need to failover, too.

SAP HANA Scale-Out - Master Node failure

Failing Master node



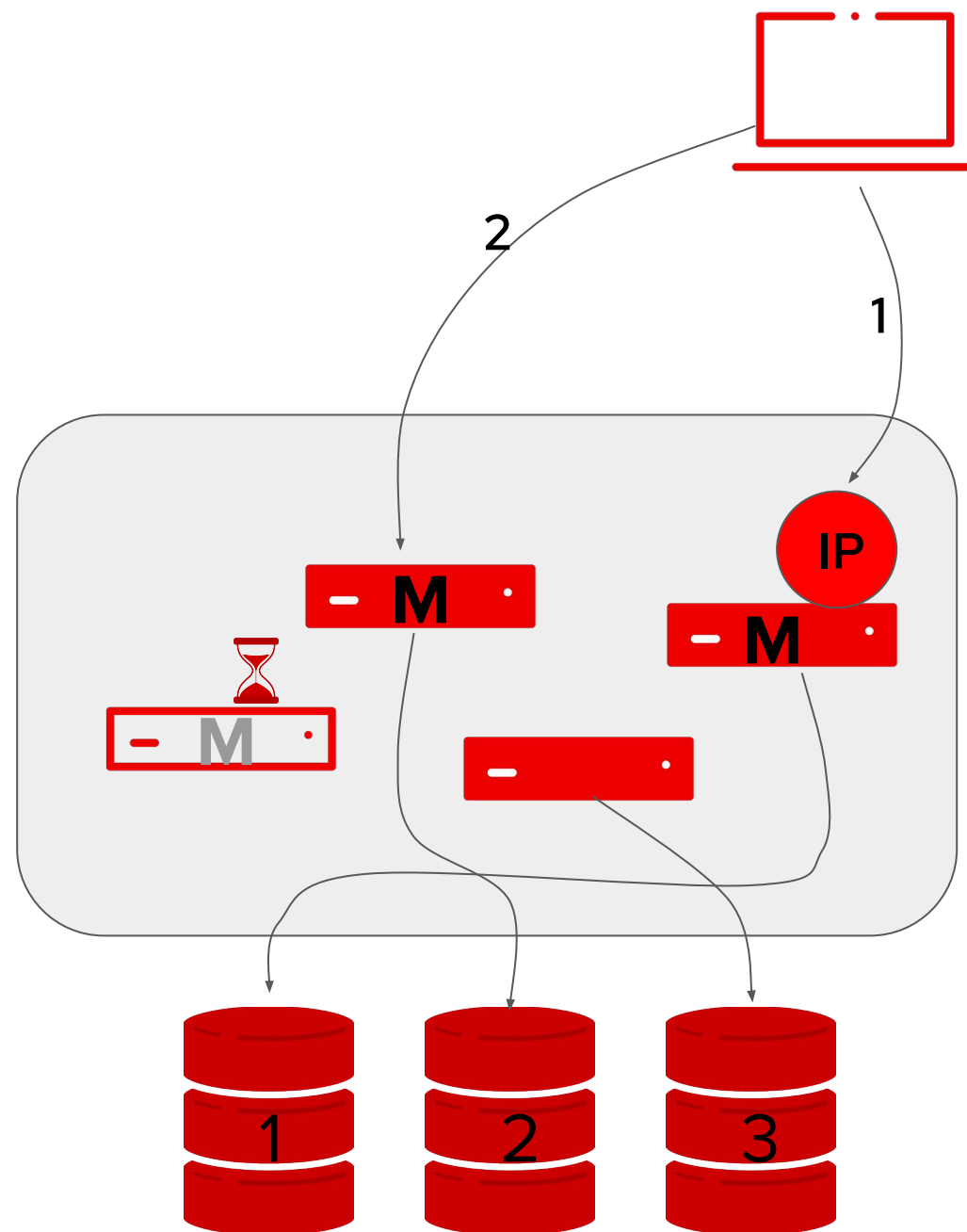
The data partition 1 needs to be released (SAP STONITH).

One of the master nameserver candidates try to failover the active master nameserver role

In best case this should be a standby node because otherwise it's data partition would need to failover, too.

SAP HANA Scale-Out - Master Node failure

Failing Master node



Summary:

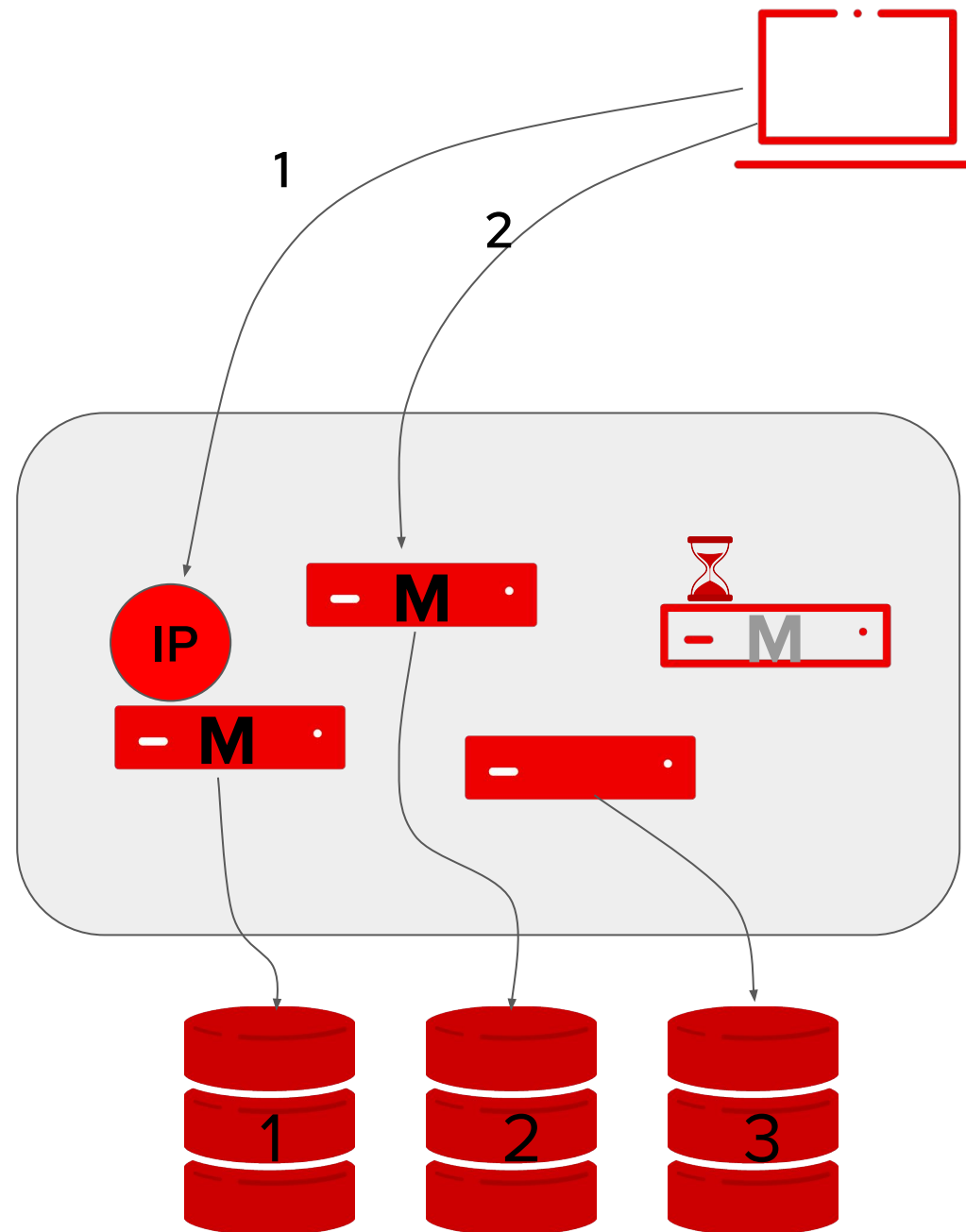
SAPHanaSR detects all failovers of the active master nameserver and migrates the virtual IP address to the standby node

SAPHanaSR allows clients to process a transparent reconnect and do not need to be configured for multiple access addresses

SAPHanaSR enables also high availability for software which is not able to connect to different IP addresses

SAP HANA Scale-Out - Standby failure

Failing standby node or instance



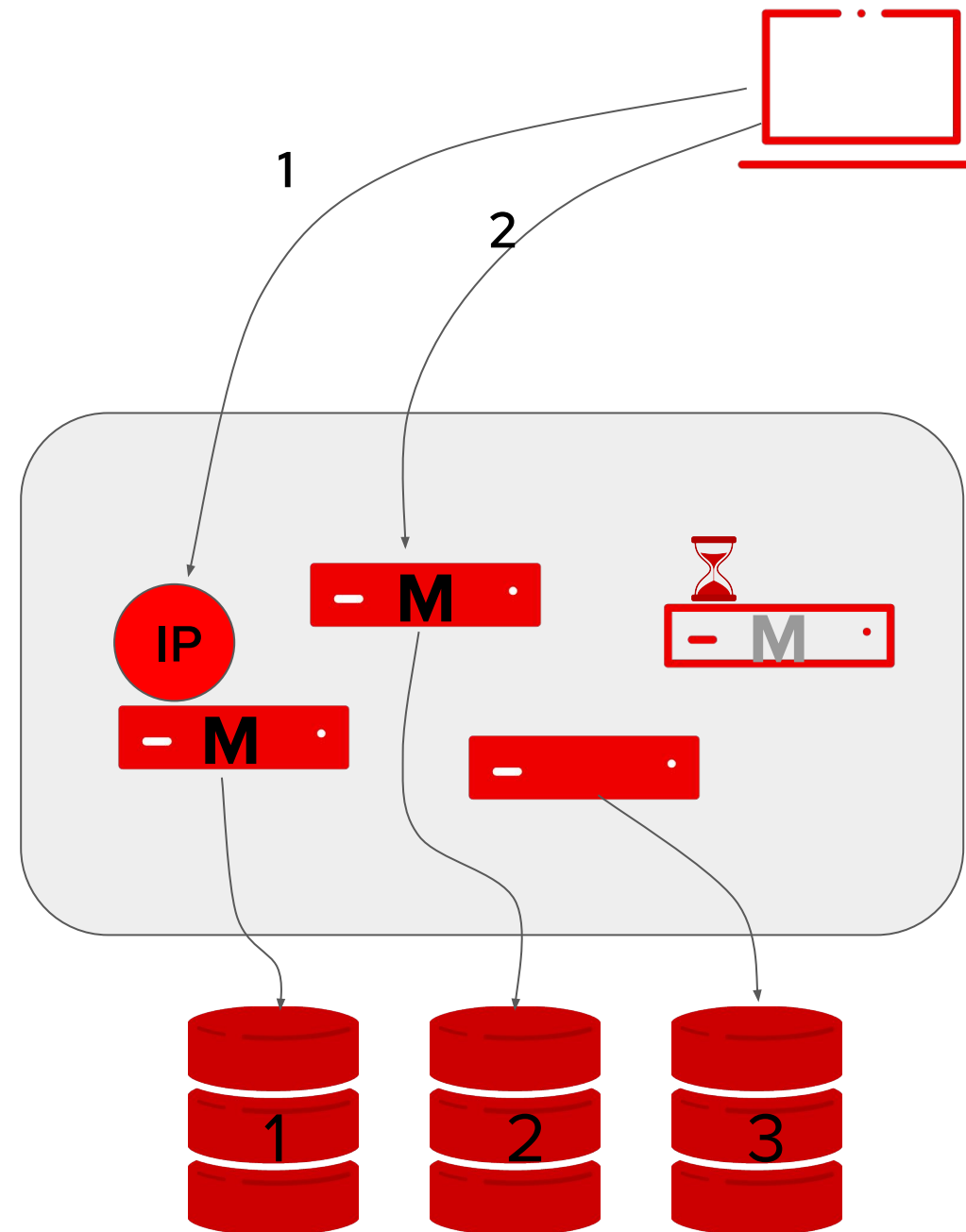
A SAP HANA standby node could be either a master nameserver candidate or a “plain” standby node.

SAP HA does typically **not** repair this situation

The running SAP HANA database is not directly influenced, but HA capacity of the site gets degraded.

SAP HANA Scale-Out - Standby failure

Failing standby node or instance



Summary:

SAPHanaSR detects the outage of the SAP HANA standby node or instance

SAPHanaSR restarts the failed SAP HANA standby instance, if the node is still part of the pacemaker cluster or rejoining the cluster

SAPHanaSR takes care of the SAP HA failover "capacity" and increases the build-in SAP high availability

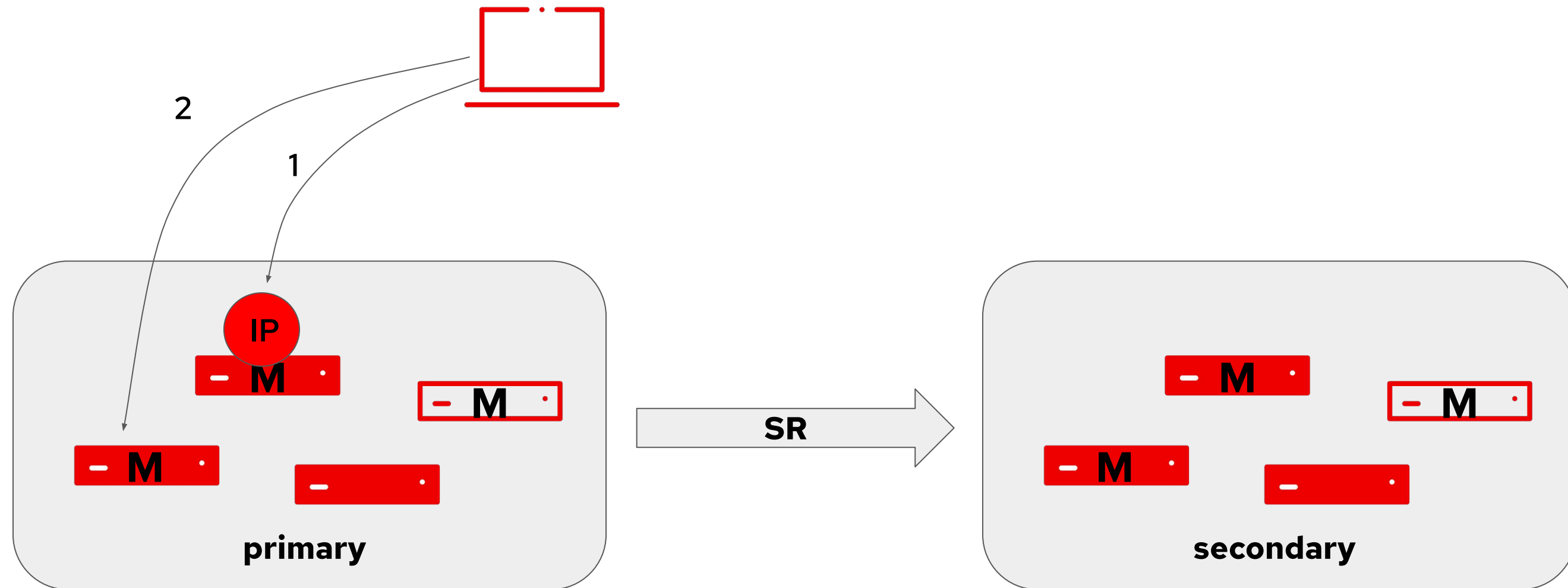
SAPHanaSR checks, if the situation allows a restart of the standby node or not

SAPHanaSR Typical Failures and Reactions

<i>Failure</i>	<i>SAPHanaSR</i>
Worker fails - node or instance	SAP HA processes failover. If SAP HA fails, SAPHanaSR processes a takeover or restart.
Active master nameserver fails - node or instance	Like the worker failure. In addition SAPHanaSR migrates the virtual IP address to the new active master nameserver.
Standby fails - node or instance	SAPHanaSR processes a instance restart to reestablish the full SAP HA capacity.
Primary site fails	SAPHanaSR processes a takeover on secondary or restart of the failed primary depending on configuration and system replication status.
Standby site fails	SAPHanaSR processes a database system restart to re-establish SAP HANA system replication.

SAP HANA Scale-Out System Replication

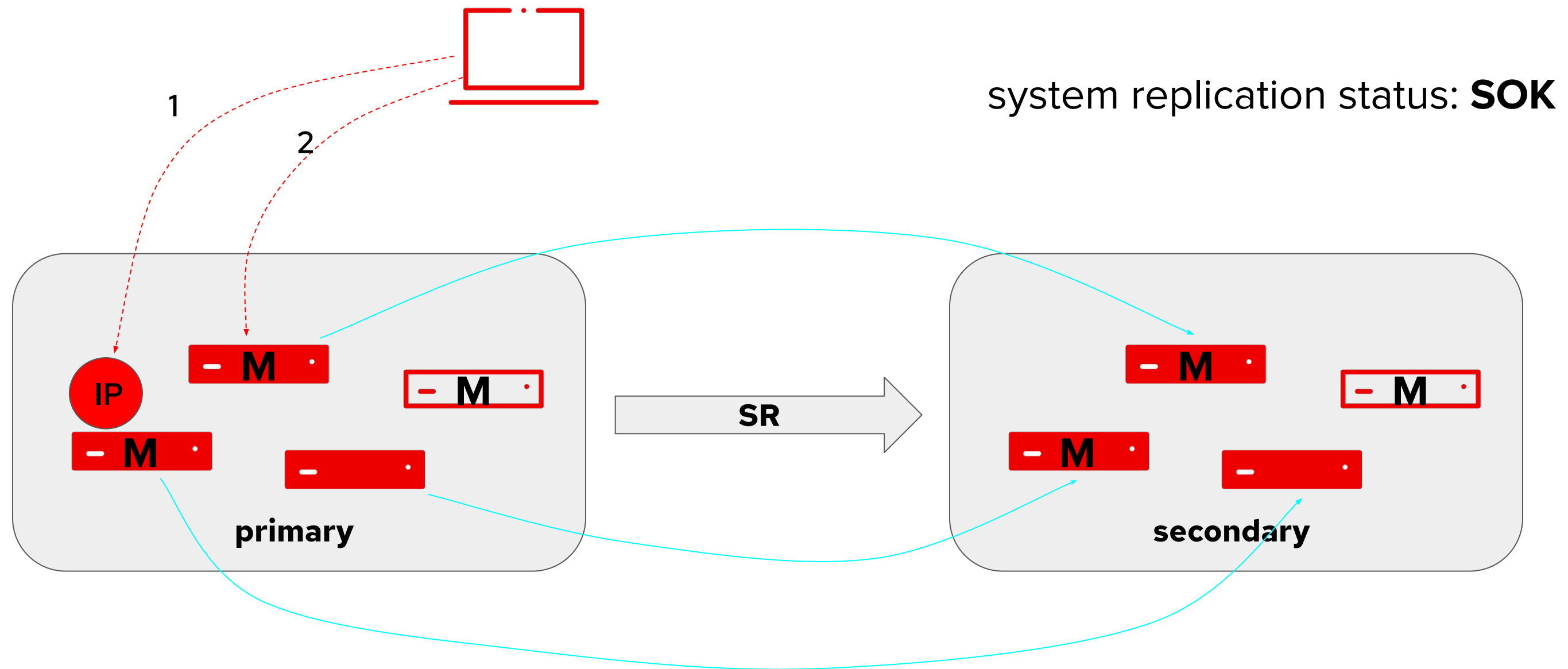
Scale-Out with System Replication (SR)



A Scale-Out SR scenario consists of two SAP HANA Scale-Out database systems

SAP HANA Scale-Out System Replication

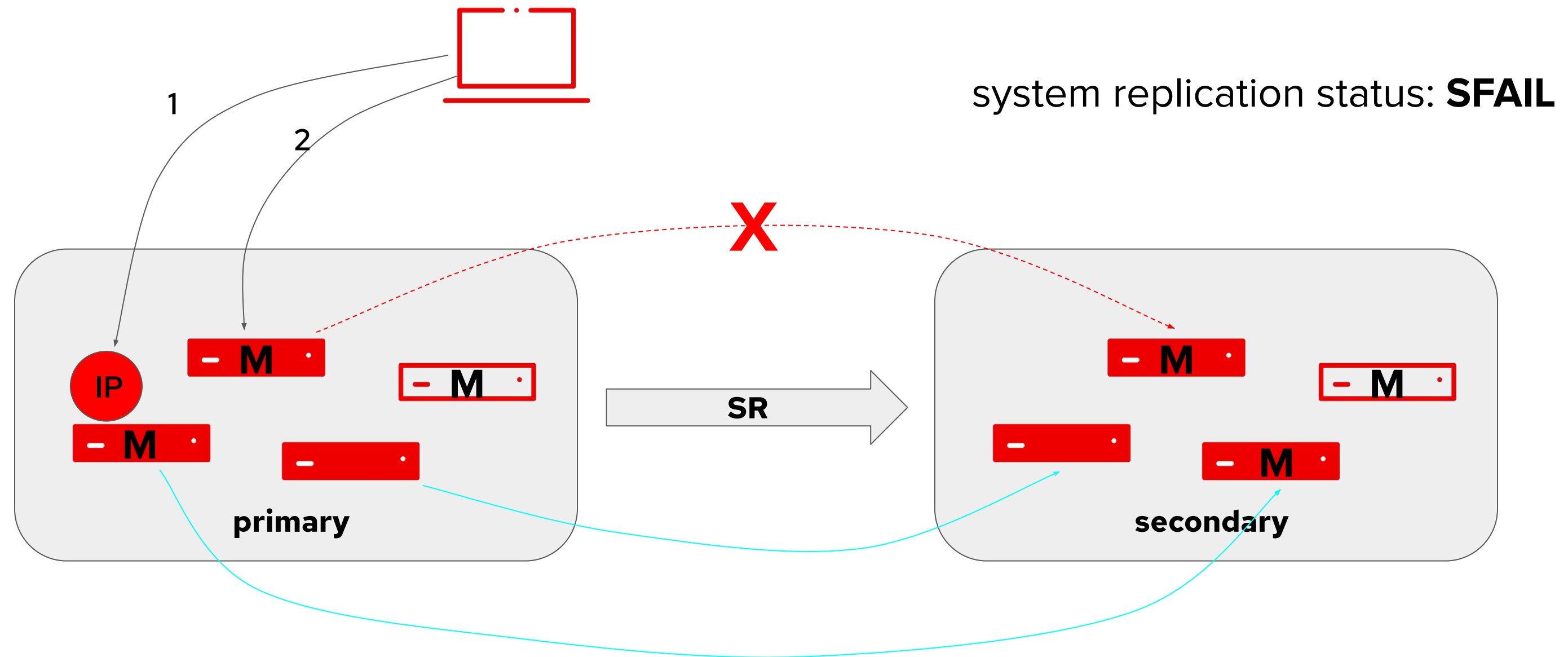
Failing primary



Synchronisation of Scale-Out is done in pairs by all worker nodes and services like tenants

SAP HANA Scale-Out System Replication

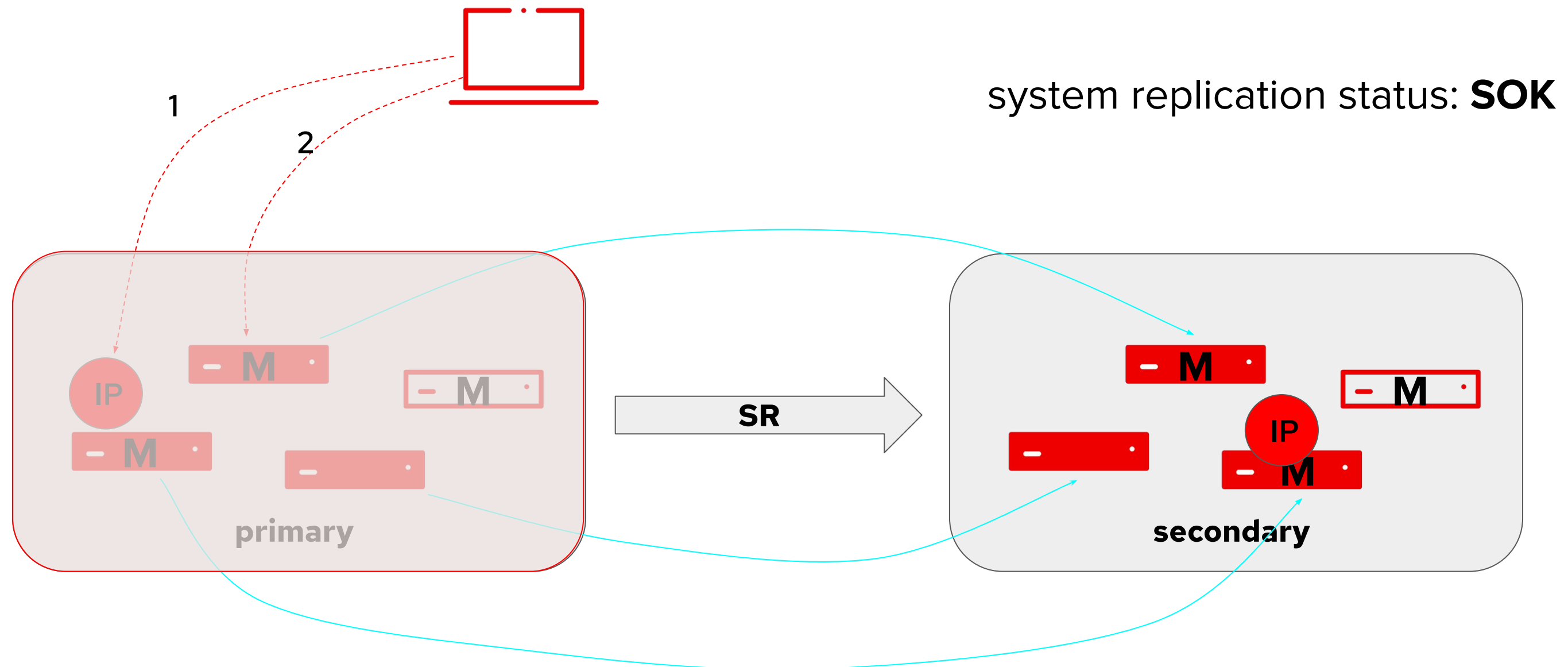
Failing Synchronization



Each single replication could fail
SAPHanaSR detects such failures and excludes
the secondary from site takeover

SAP HANA Scale-Out System Replication

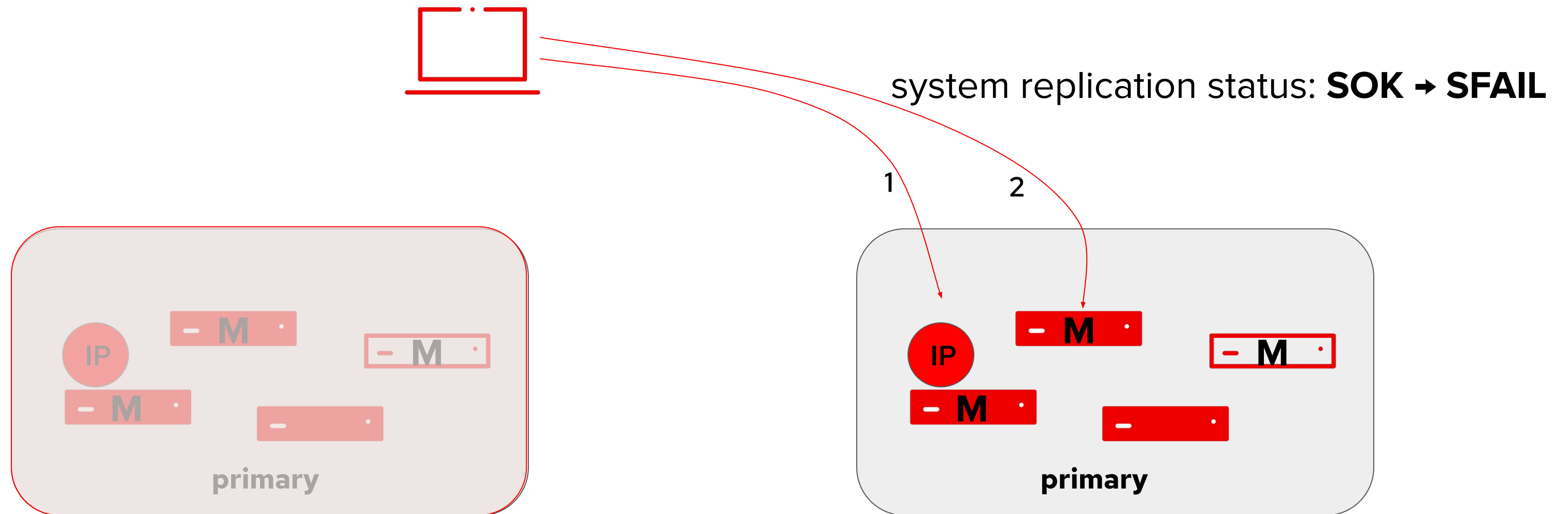
Failing primary



SAPHanaSR detects the failing primary.
Depending on the configuration and the system
replication status a takeover is processed

SAP HANA Scale-Out System Replication

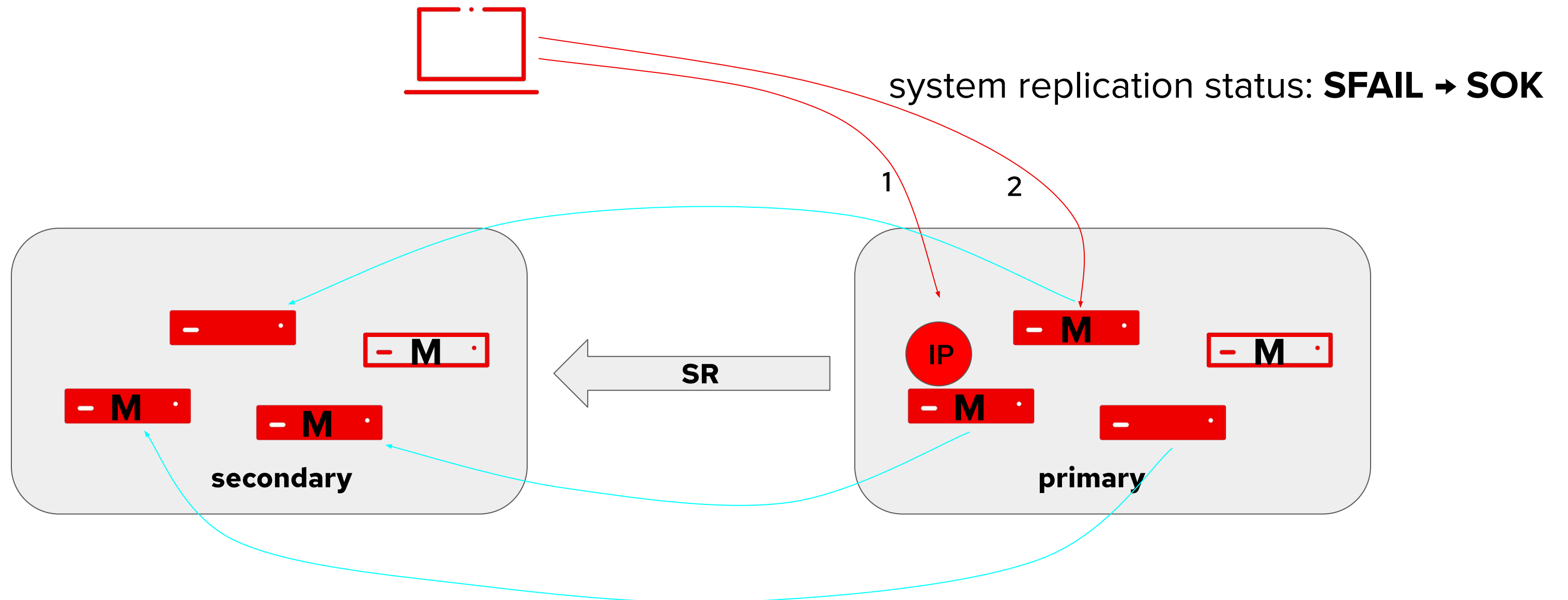
Failing primary



SAPHanaSR processes the takeover to the secondary site
and switches the virtual IP address
so clients could transparently reconnect

SAP HANA Scale-Out System Replication

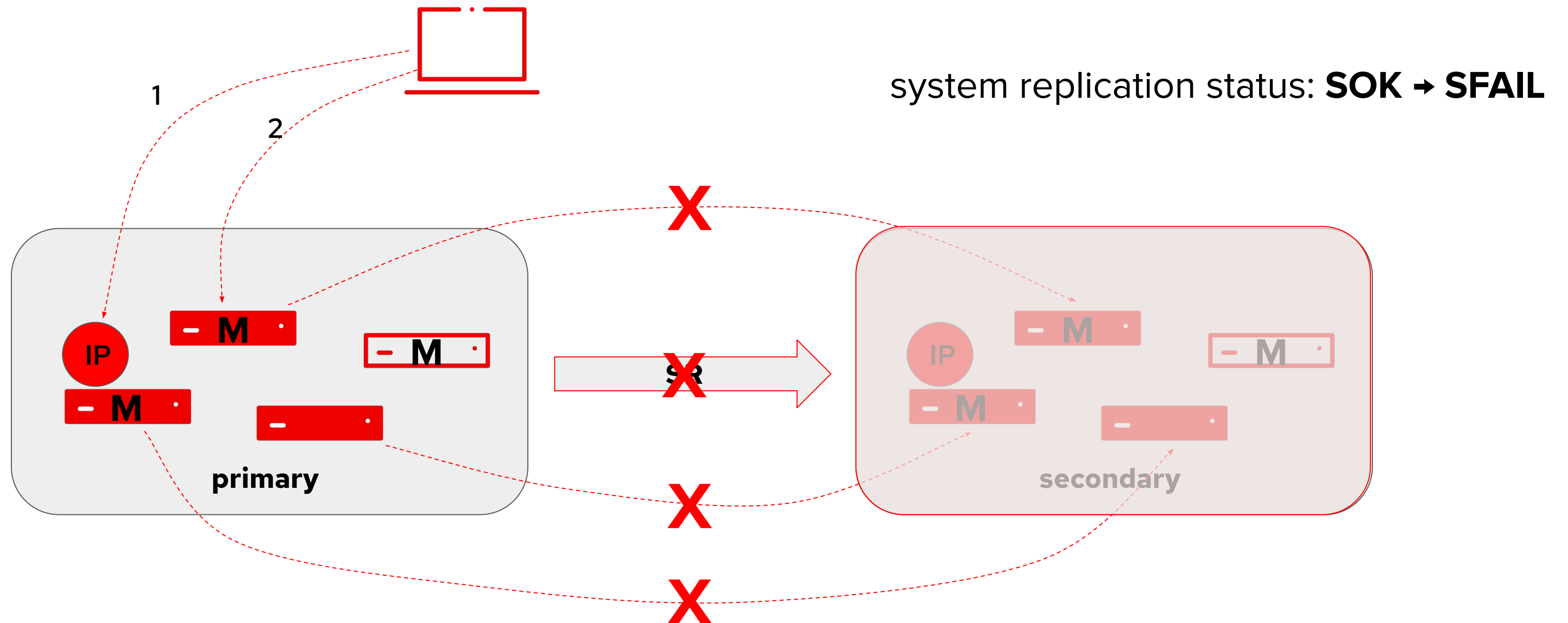
Failing primary



SAPHanaSR could process a registration of the failed primary, depending in the configuration and checks if the new SR pair gets in sync

SAP HANA Scale-Out System Replication

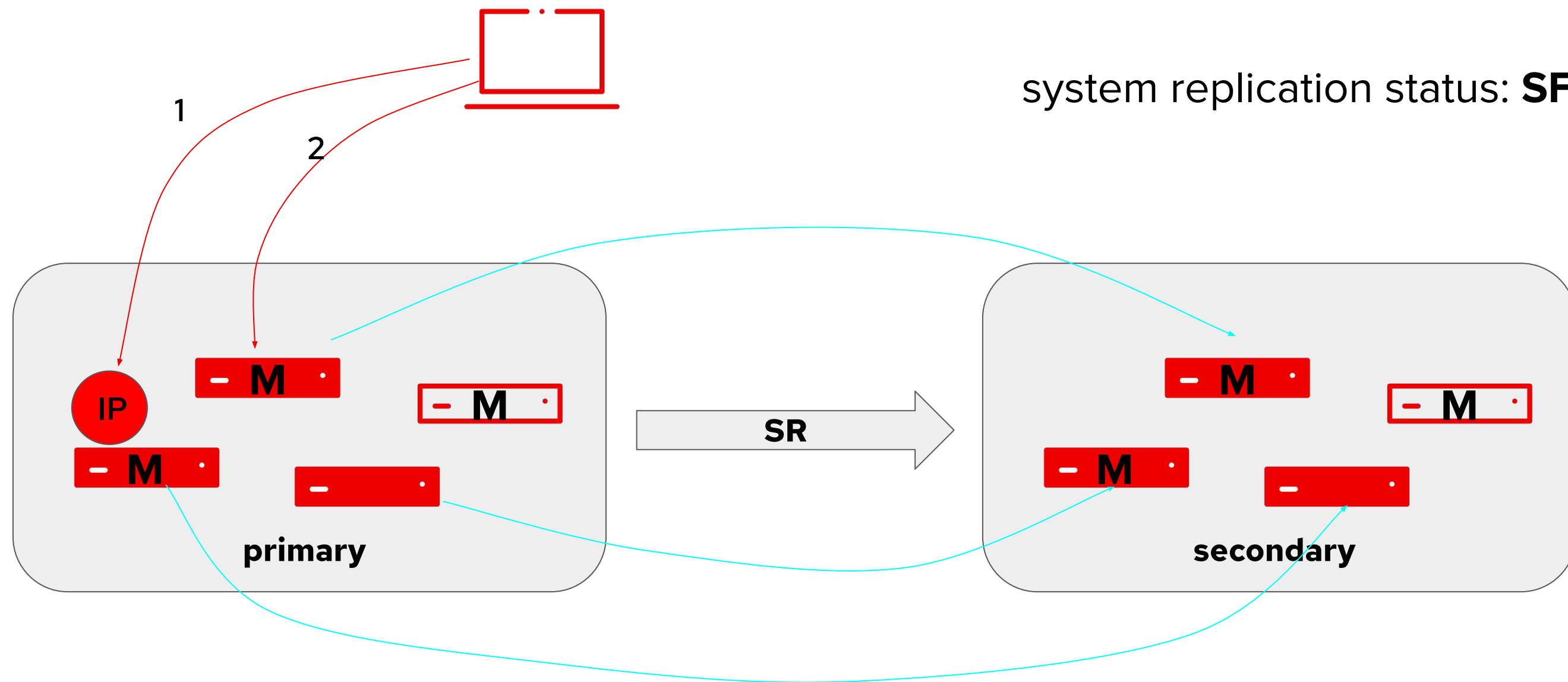
Failing secondary



SAPHanaSR detects failing secondary sites and handles the tracking of the system replication status to prevent sub-optimal takeovers

SAP HANA Scale-Out System Replication

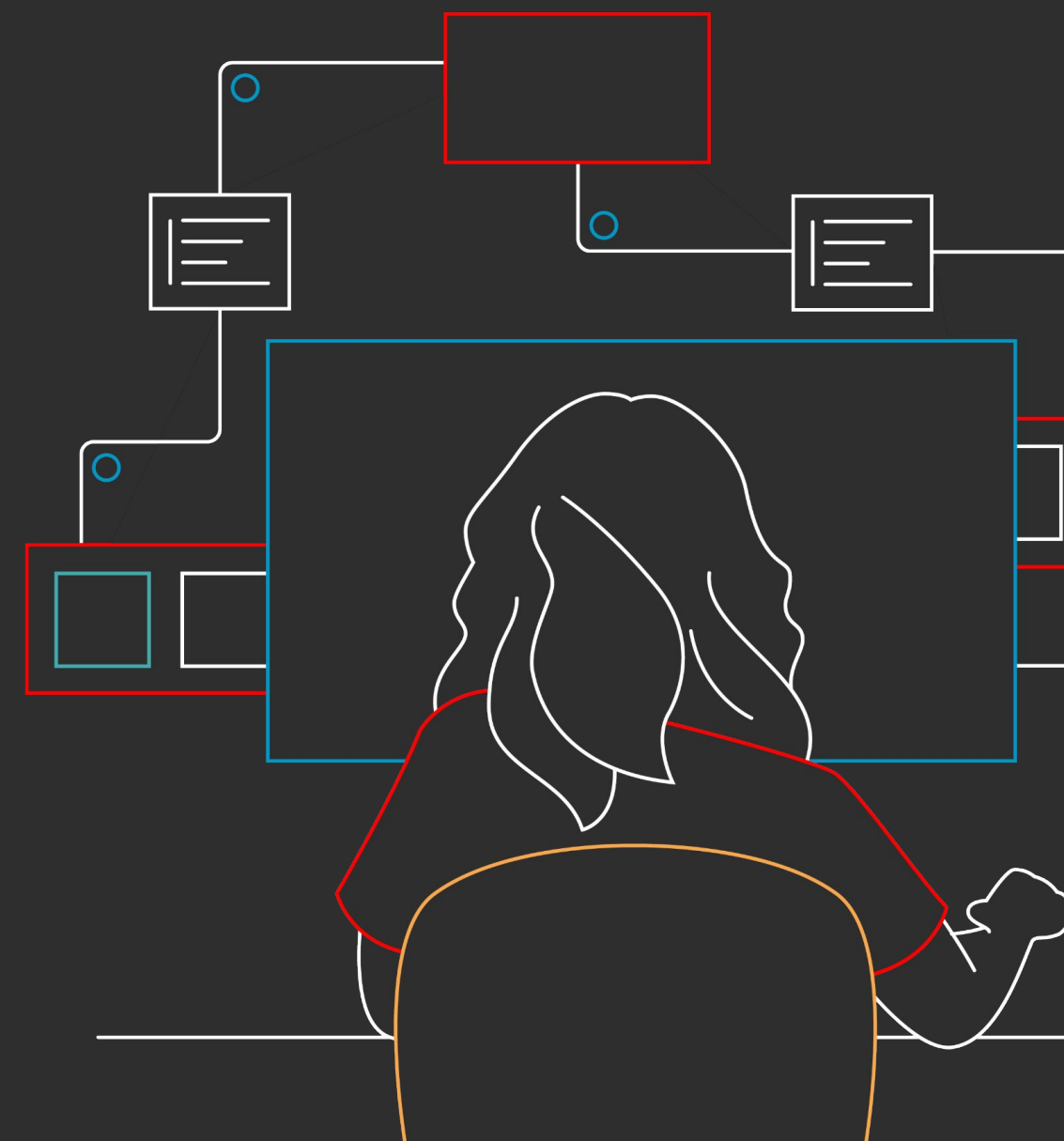
Failing secondary



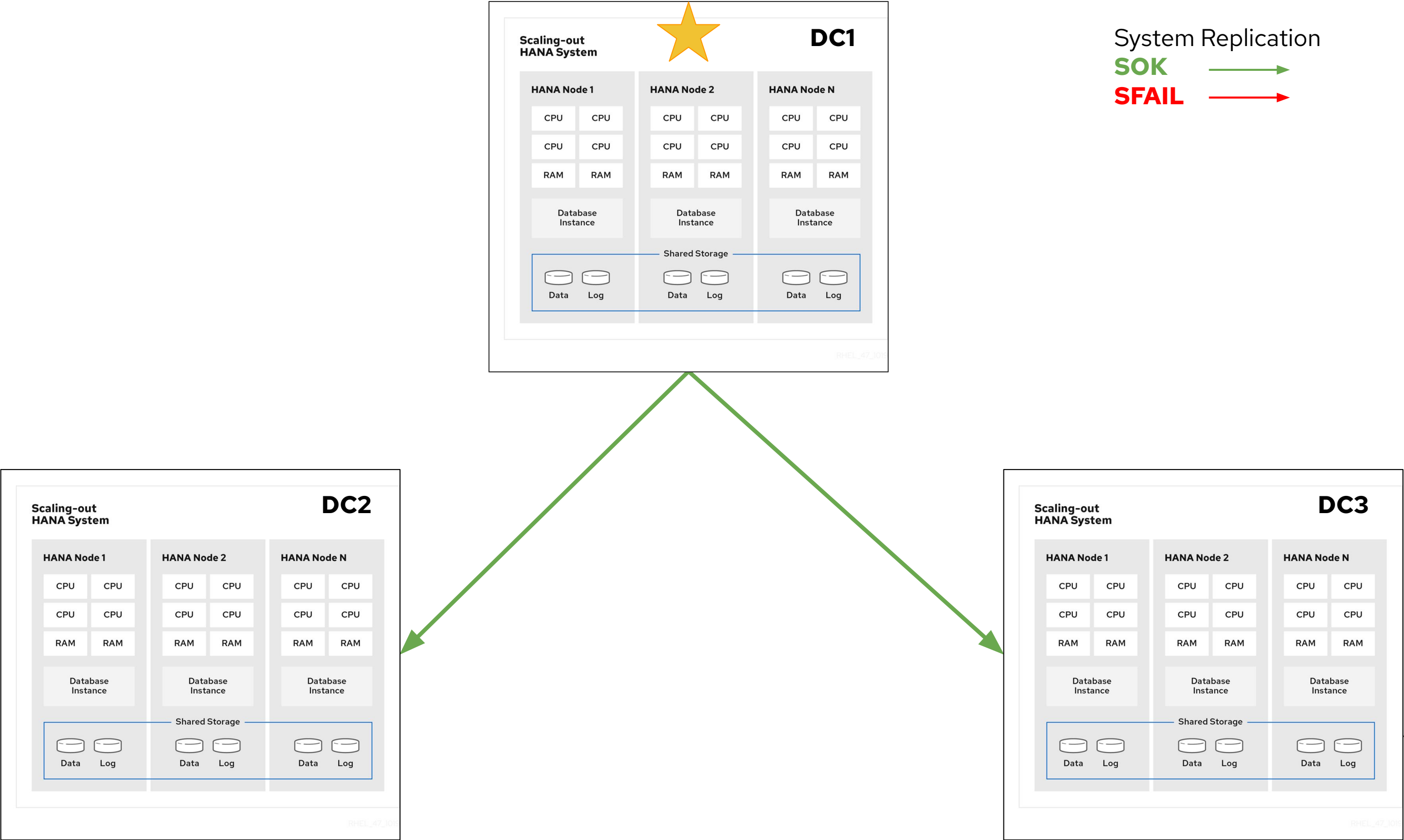
SAPHanaSR processes the restart of the secondary site and checks the system replication status to allow optimal takeovers.

Explaining SAP HANA **System Replication**

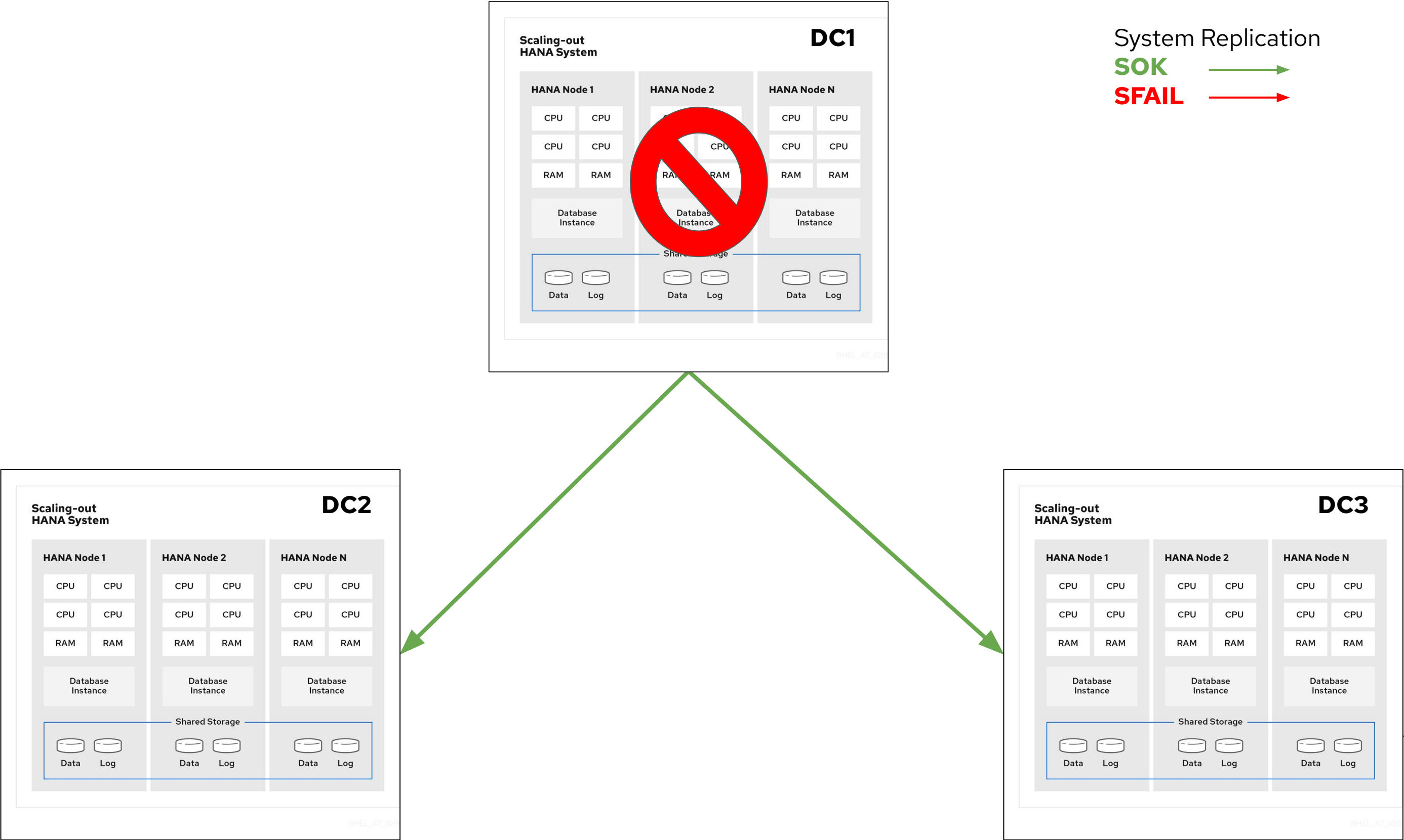
**Explaining Failover in the
Multitarget System Replication Environment**



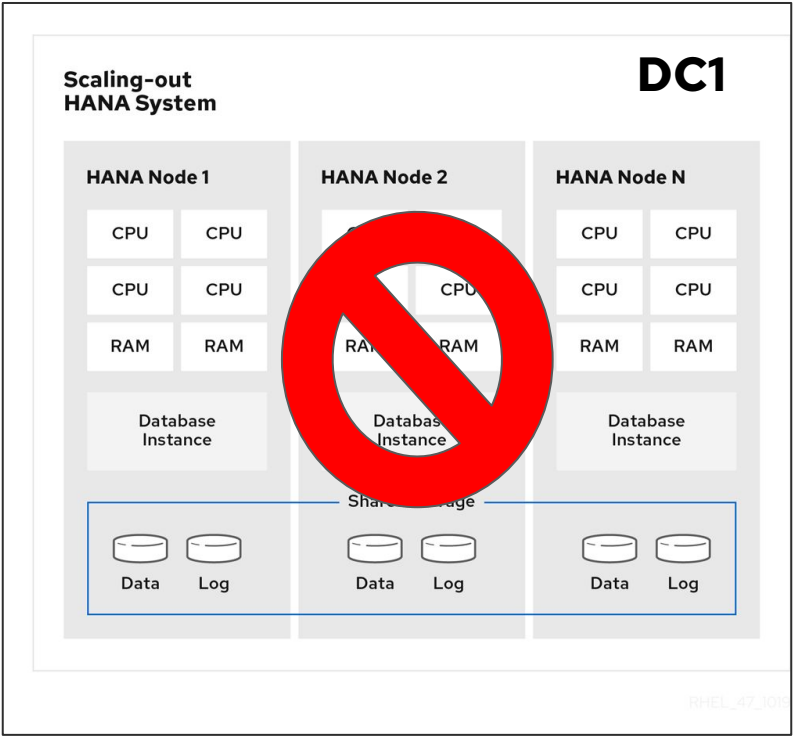
Normal Operation



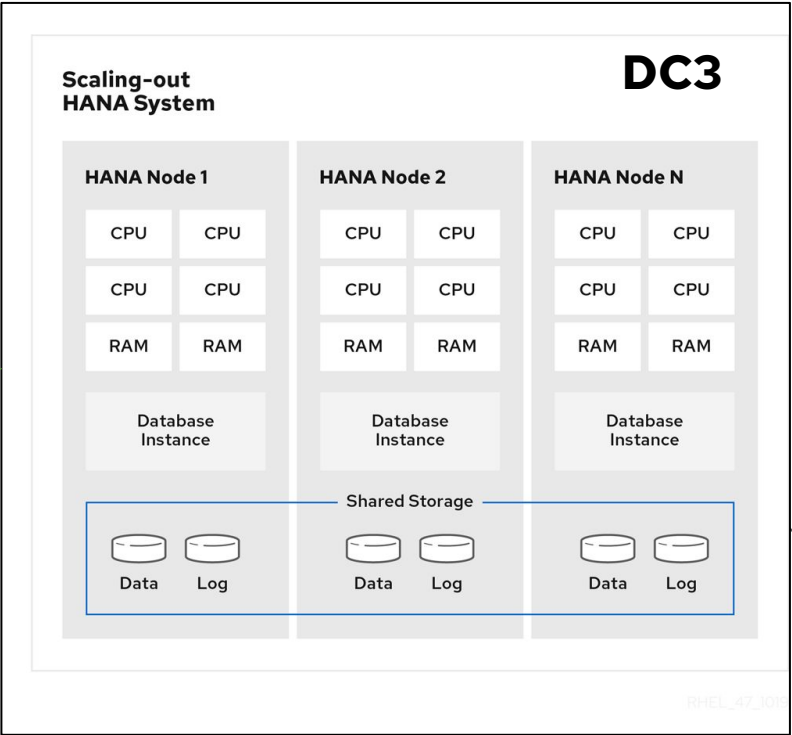
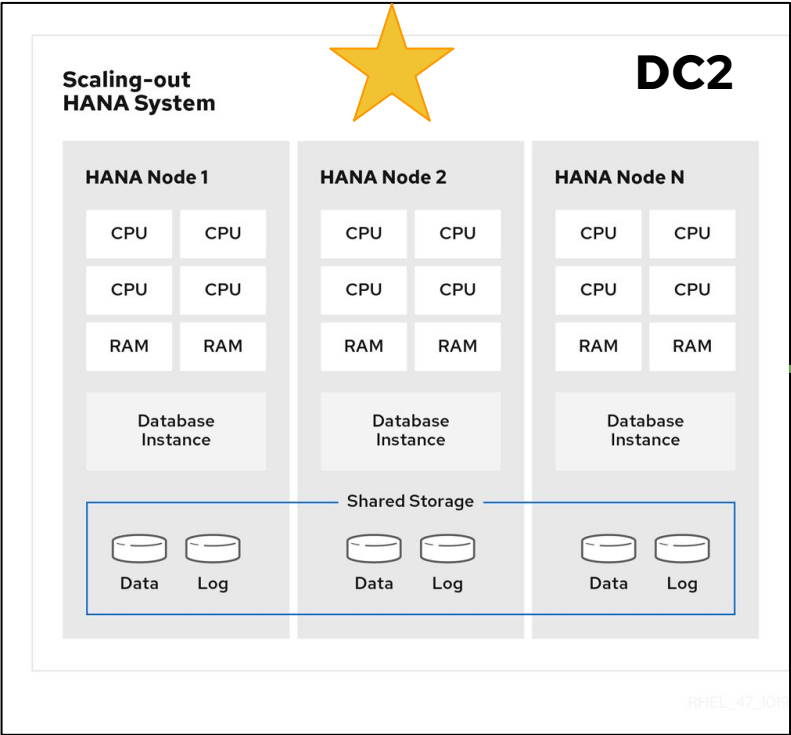
Primary in DC1 fails



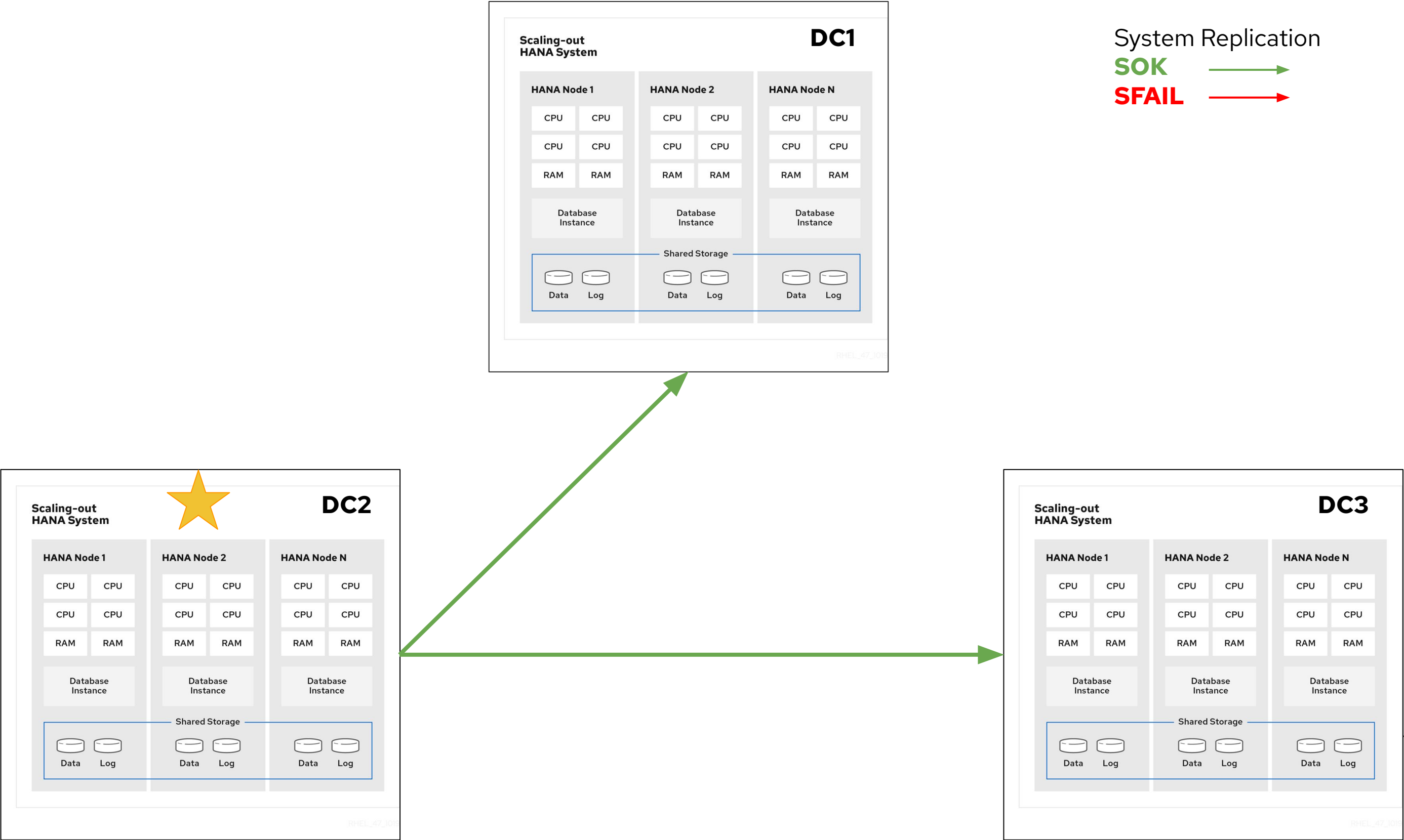
DC2 takes over and DC3 is re-registered to DC2



System Replication
SOK →
SFAIL →



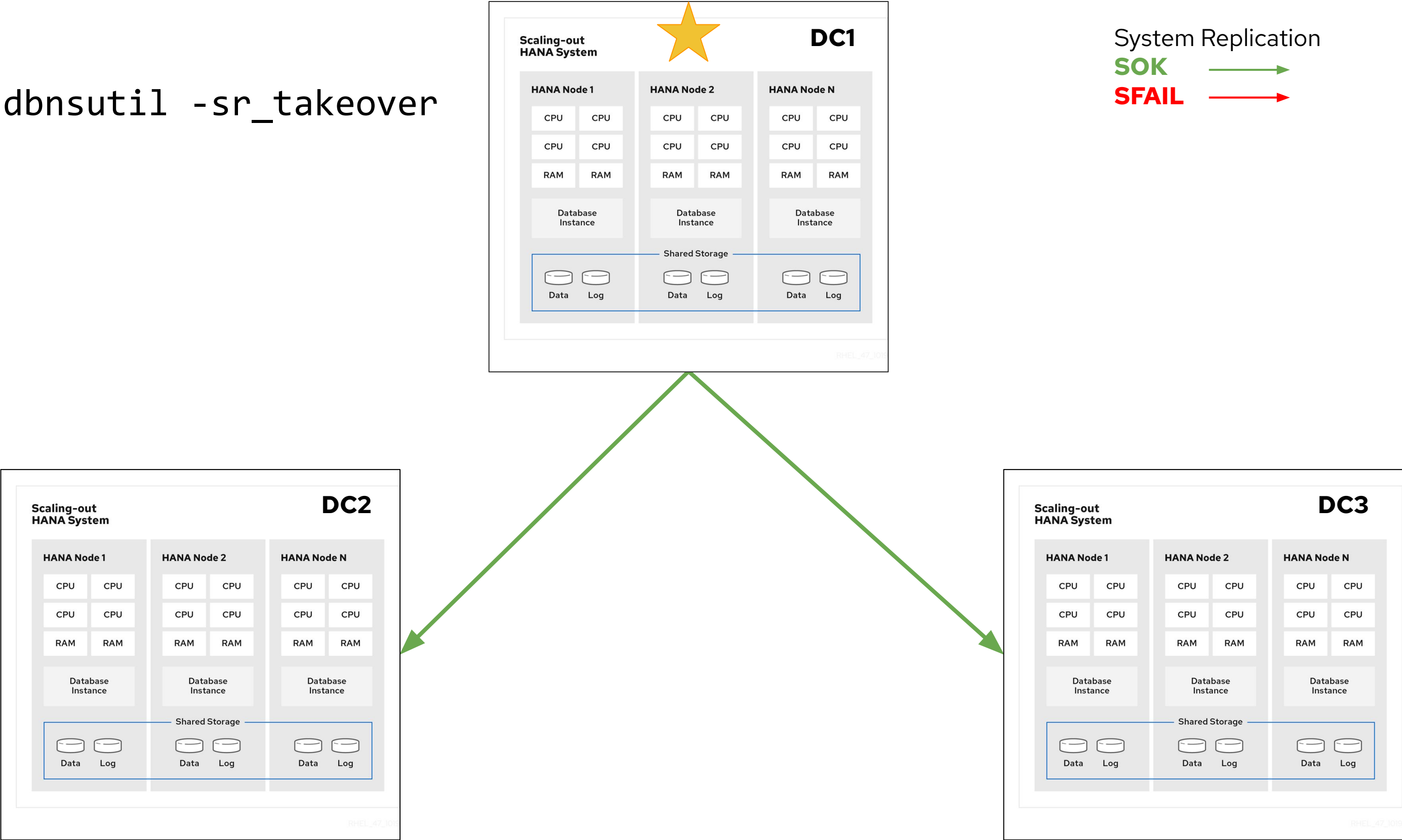
DC1 is re-registered to DC2



Back to Normal Operation

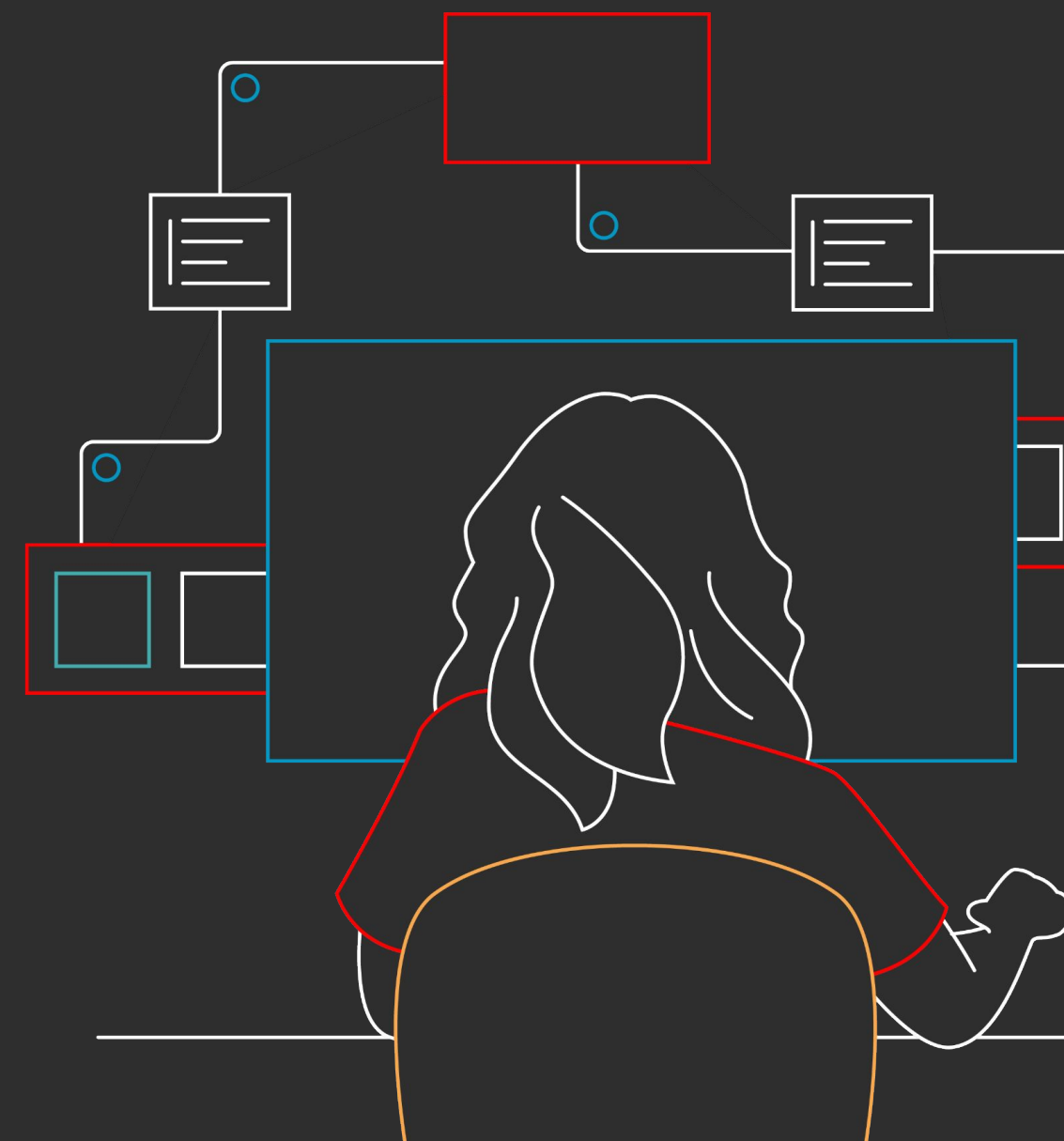
```
nsdc1# hdbnsutil -sr_takeover
```

System Replication
SOK →
SFAIL →

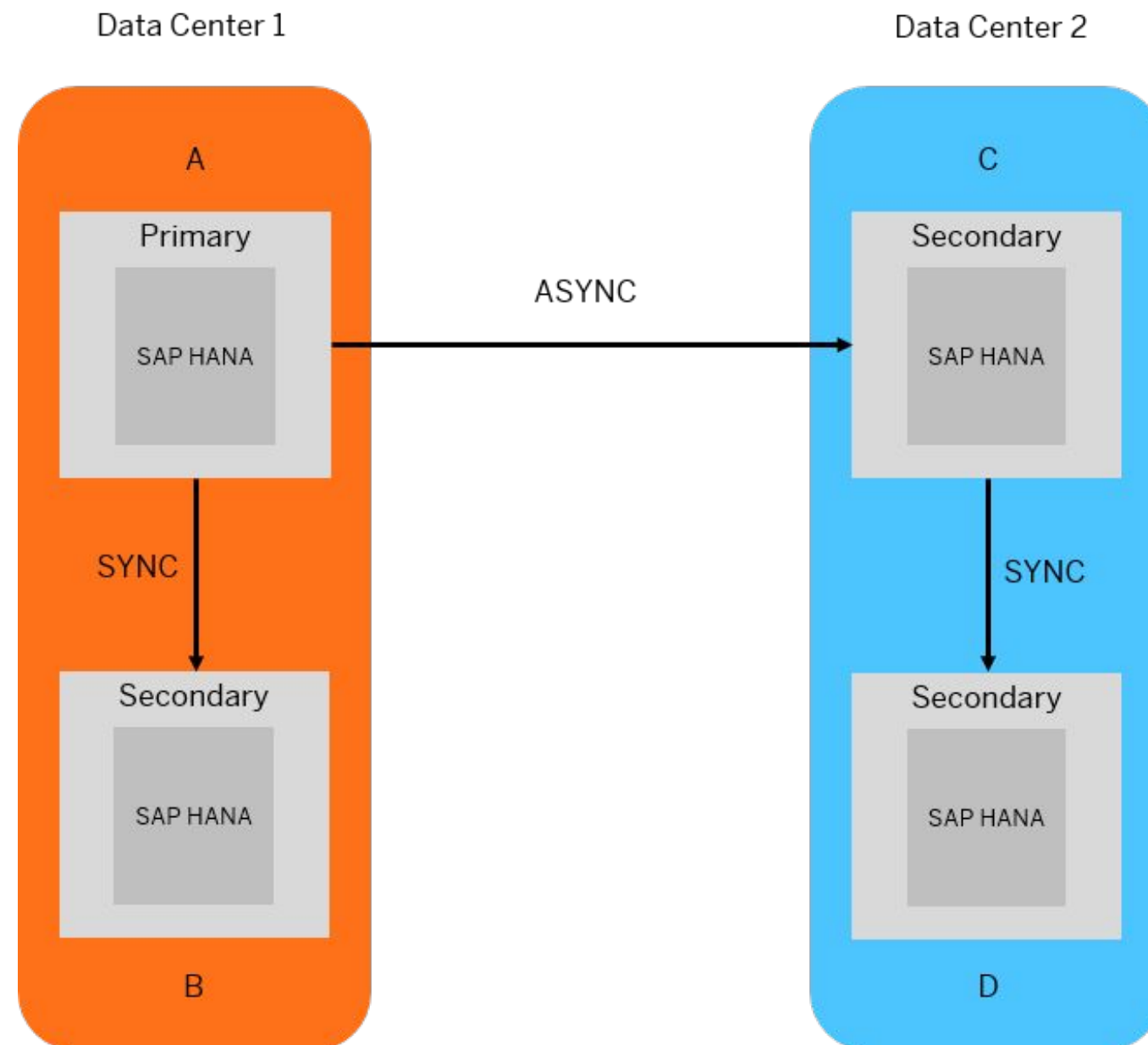


Explaining SAP HANA **System Replication**

**Installing SAP HANA Scale-out
Multitarget System Replication**



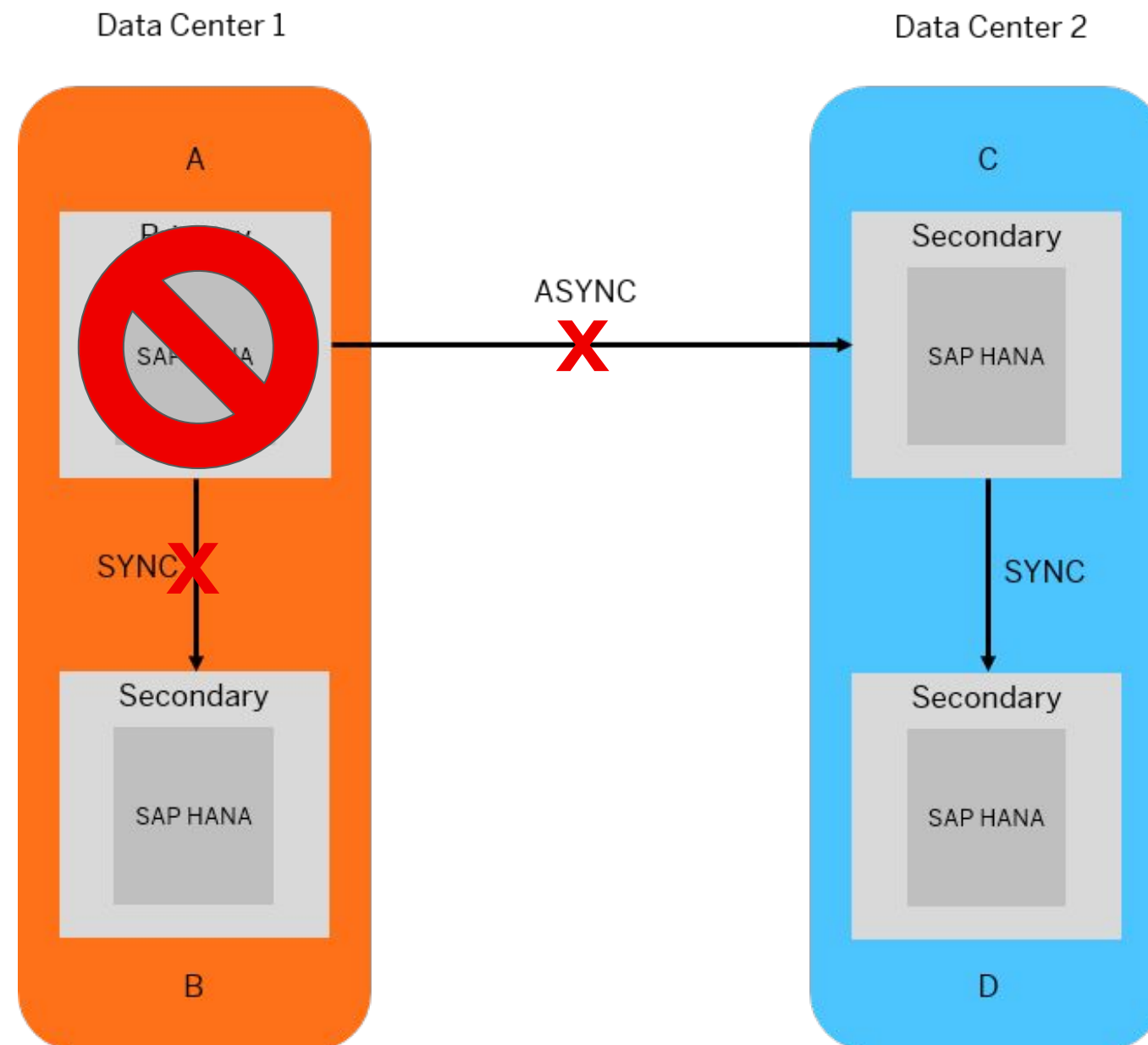
SAP Hana Multi Target Replication



"Multi-Target" System Replication is a new feature in HANA 2.0 SPS04

The primary system can replicate data changes to more than one secondary system

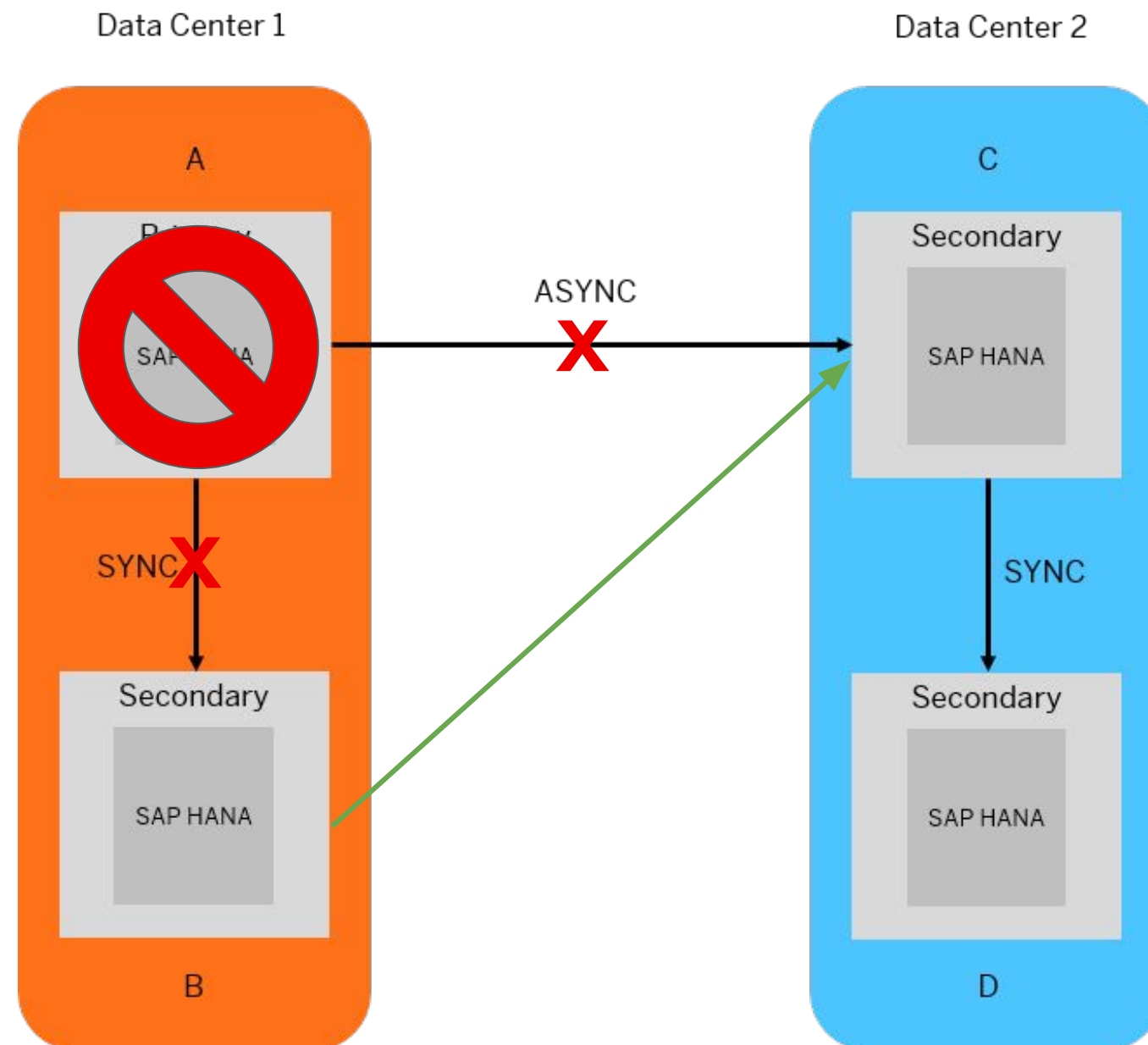
SAP Hana Multi Target Replication



"Multi-Target" System Replication is a new feature in HANA 2.0 SPS04

The primary system can replicate data changes to more than one secondary system

SAP Hana Multi Target Replication



"Multi-Target" System Replication is a new feature in HANA 2.0 SPS04

The primary system can replicate data changes to more than one secondary system

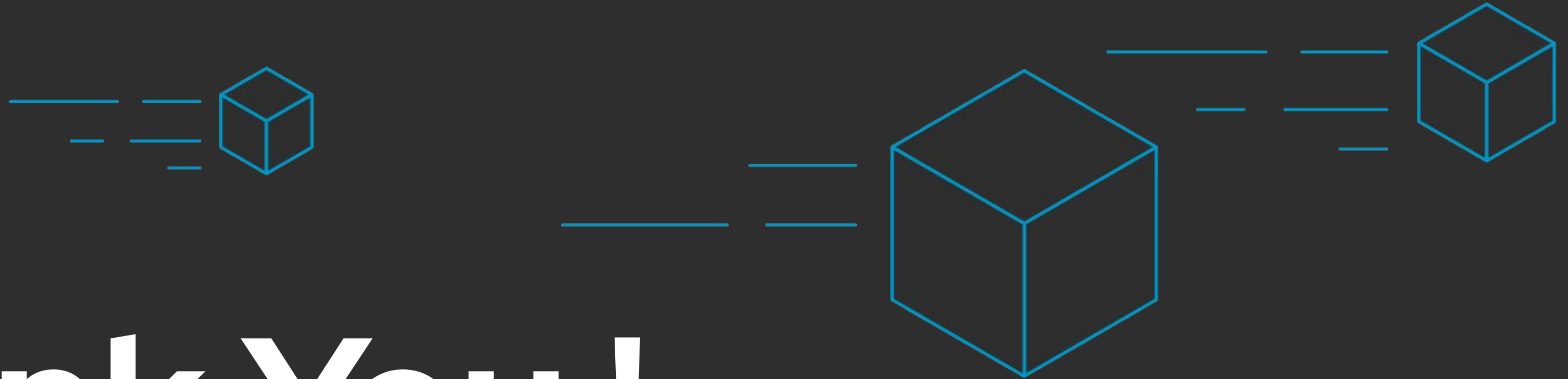
non-failed secondaries are re-registered after fail-over

Installation Steps

- Set `register_secondaries_on_takeover` parameter in the SAP HANA `global.ini`
- SAP HANA installation on a third site.
- Register the third site as a secondary SAP HANA replication server.

Optional:

- Add the nodes of the third site to the cluster.
- Create constraints to avoid SAP resource agent resources running on nodes of the third site.



Thank You!



<https://linkedin.com/company/Red-Hat>



<https://facebook.com/RedHatinc>



<https://youtube.com/user/RedHatVideos>



<https://twitter.com/RedHat>

