HIGH AVAILABILITY AND DISASTER RECOVERY FOR IMDG

VLADIMIR KOMAROV, MIKHAIL GORELOV

SBERBANK OF RUSSIA

ABOUT SPEAKERS

Vladimir Komarov

Enterprise IT Architect

vikomarov@sberbank.ru

in Sberbank since 2010. He realized the concepts of operational data store (ODS) and retail risk data mart as a part of enterprise data warehouse. In 2015 performed the testing of 10+ distributed in-memory platforms for transaction processing. Now responsible for grid-based core banking infrastructure architecture including high availability and disaster recovery.

Mikhail Gorelov

Operations expert & manager

magorelov@sberbank.ru

in Sberbank since 2012. He is responsible for building the infrastructure landscape for the major mission-critical applications as core banking and cards processing including new grid-based banking platform. Now he acts as both expert and project manager in “18+” core banking transformation program.

ABOUT SBERBANK

SBERBANK

The largest bank in Russian Federation

- 16K+ offices in Russia, 11 time zones
- 110M+ retail clients
- 1M+ corporate clients
- 90K+ ATMs & POS terminals
- 50M+ active web & mobile banking users

OUR GOALS

$$Availability = \frac{Total\ time - Downtime}{Total\ time} \times 100\ \%$$

Availability	Yearly downtime
99 %	3d 15:39:29.5
99.9 %	8:45:57.0
99.99 %	0:52:35.7
99.999 %	0:05:15.6
99.9999 %	0:00:31.6

target for 2018

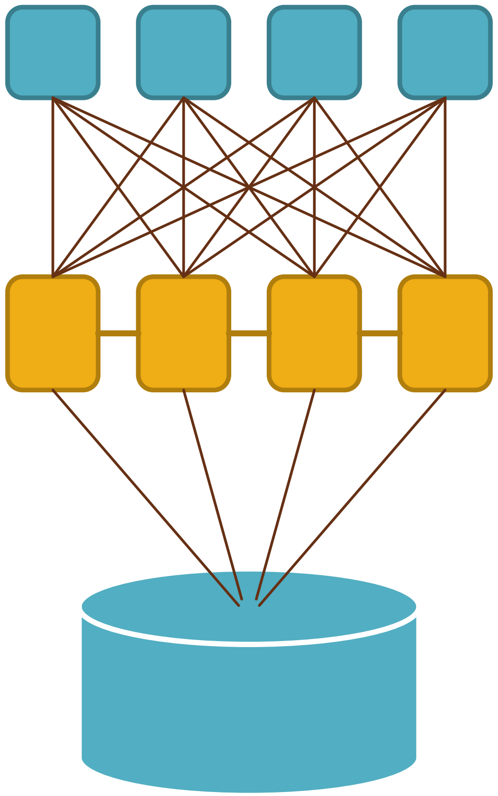
OUR METHODS

- additional **control and checking tools;**
- **monitoring** improvement:
  - new metrics design;
  - new visualizations;
- continuous **testing**:
  - operational acceptance tests;
  - performance tests;
  - 45+ scenarios of destructive testing;
- keeping **incident response plan** up-to-date.

THREATS AND FACILITIES

	Datacenter loss	DC interconnect failure	Application bugs, admin errors	User data corruption	HW/OS/JVM failures
On-disk data persistence					✓
Data redundancy	✓				✓
Distributed cluster	✓	✓			
Data snapshots			✓	✓	
Point-in-time recovery			✓	✓	
Health self-check					✓
Data replication	✓	✓		✓	

THE LEGACY GRID-ENABLED ARCHITECTURE



Application servers
compute

In-memory data grid
caching & temporary storage

Relational DBMS
persistence & compute

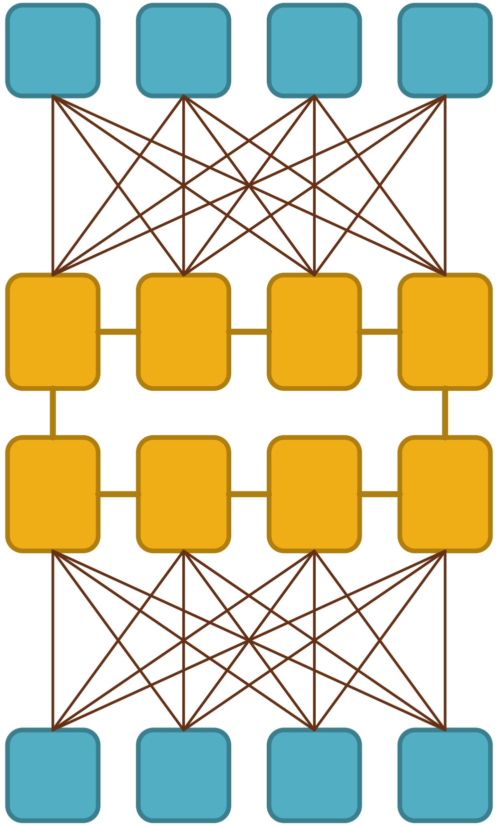
Strengths

- Robust and stable persistence layer
- A grid hasn't to be highly available

Weaknesses

- The write performance is limited by database
- The persistence layer is not horizontally scalable
- Data need to be converted from object representation to relational model
- Database and grid can become inconsistent if data is changed directly in the database
- The database requires high-end hardware

SBERBANK CORE BANKING PLATFORM ARCHITECTURE



Application servers
compute

In-memory data grid
compute & data persistence

GridGain

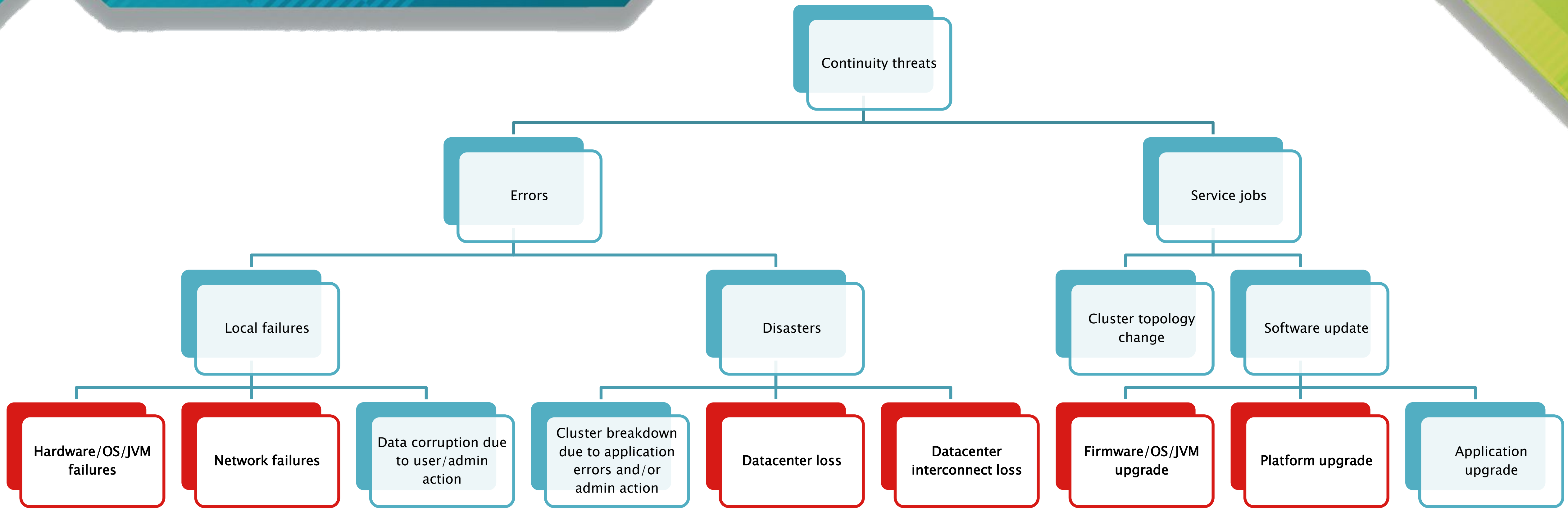
Opportunities

- Fully horizontally scalable architecture on commodity hardware
- The data is stored as objects, no conversion required
- The only instance of the data

Challenges

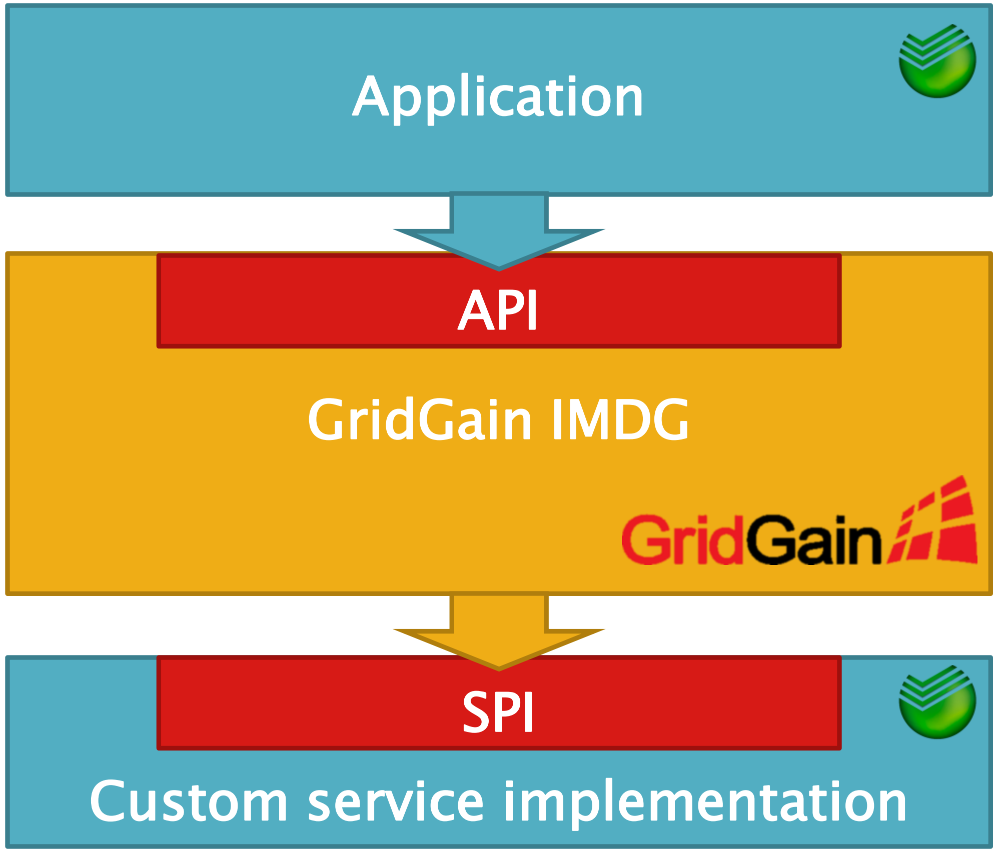
- The grid has to persist the data
- The grid has to be fault tolerant

SERVICE CONTINUITY THREATS



- The above tree does not consider security issues
- Application and user issues cannot be solved at platform level
- **Let's focus on system issues!**

THE CONCEPT OF SERVICE PROVIDER INTERFACE (SPI)



API vs. SPI

	API	SPI
Defined by	Platform	Platform
Implemented by	Platform	System software (custom code)
Called by	Application (custom code)	Platform

Sberbank implements GridGain SPI:

`TopologyValidator`

`AffinityFunction`

THE CONCEPT OF AFFINITY

AffinityFunction SPI

`nodeFilter`

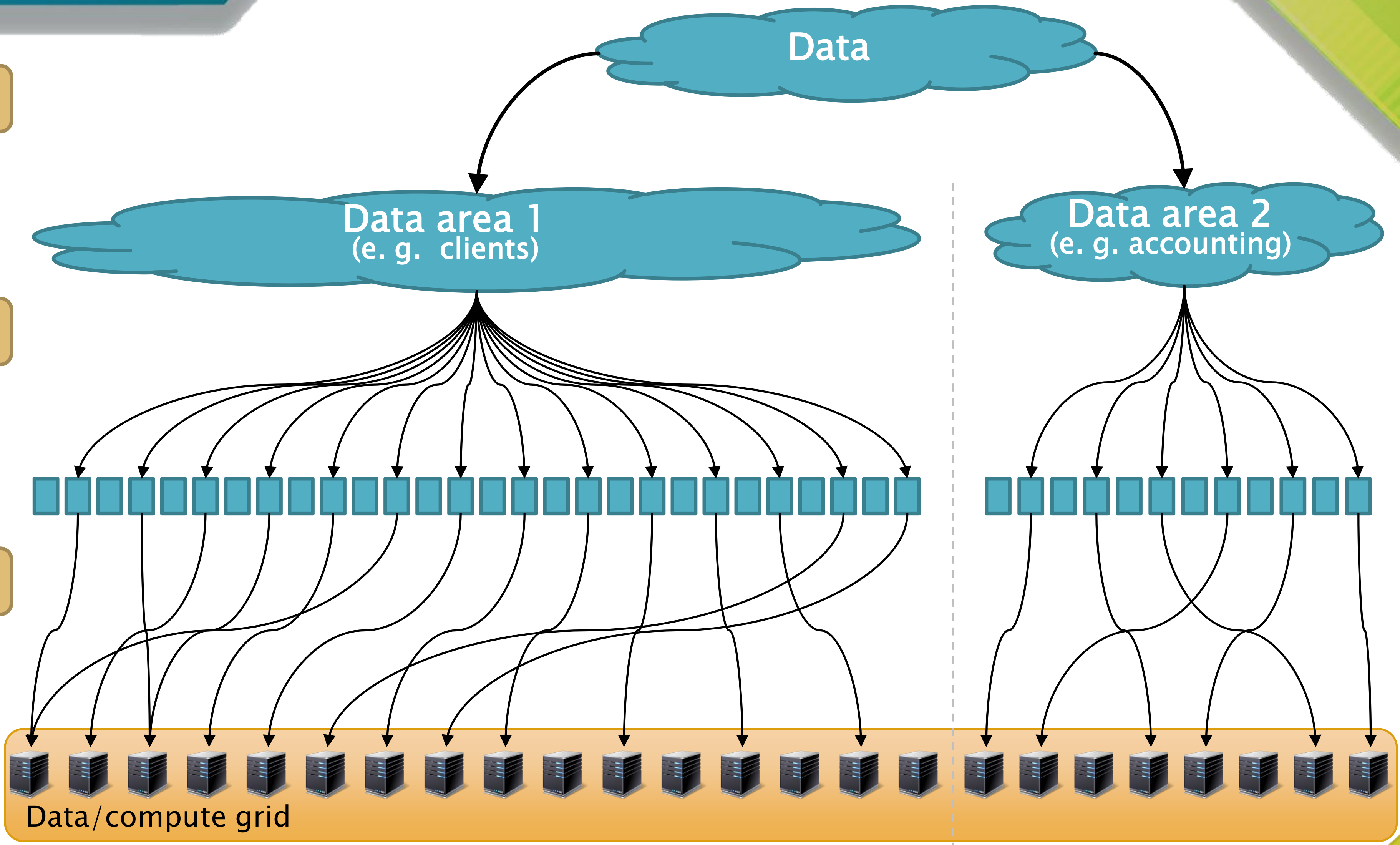
the property of the cache that defines the set of nodes where the cache's data can reside

`partition()`

the fast, simple and deterministic function usually division reminder mapping object to the partition (chunk)

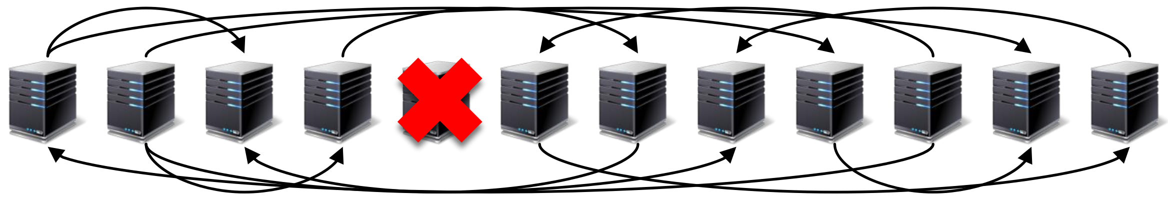
`assignPartitions()`

the function distributing partitions (chunks) across the nodes



THE CONCEPT OF CELL; NEW AFFINITY FUNCTION

Broken node:



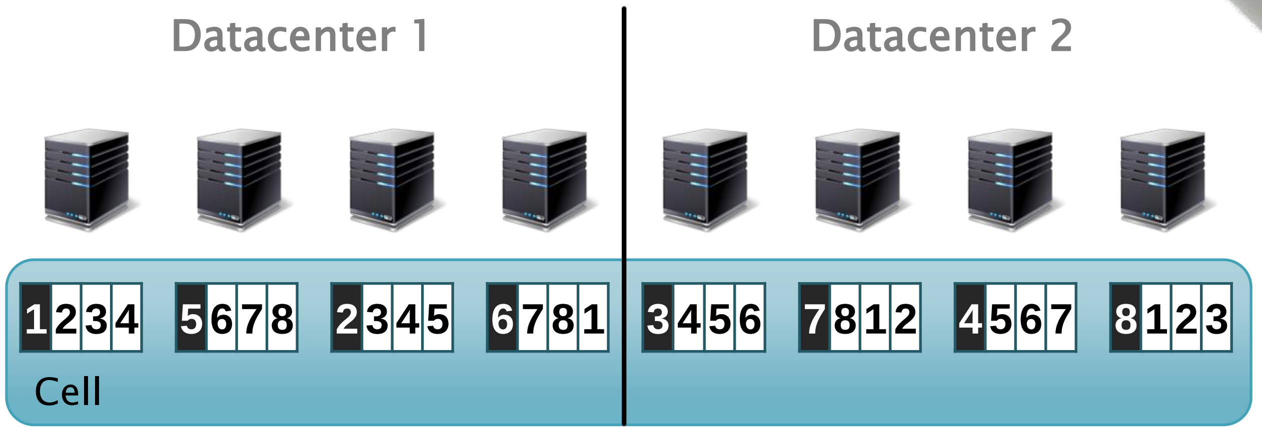
more nodes in the cluster → faster recovery

Semi-broken node:

more linked nodes → stronger performance impact

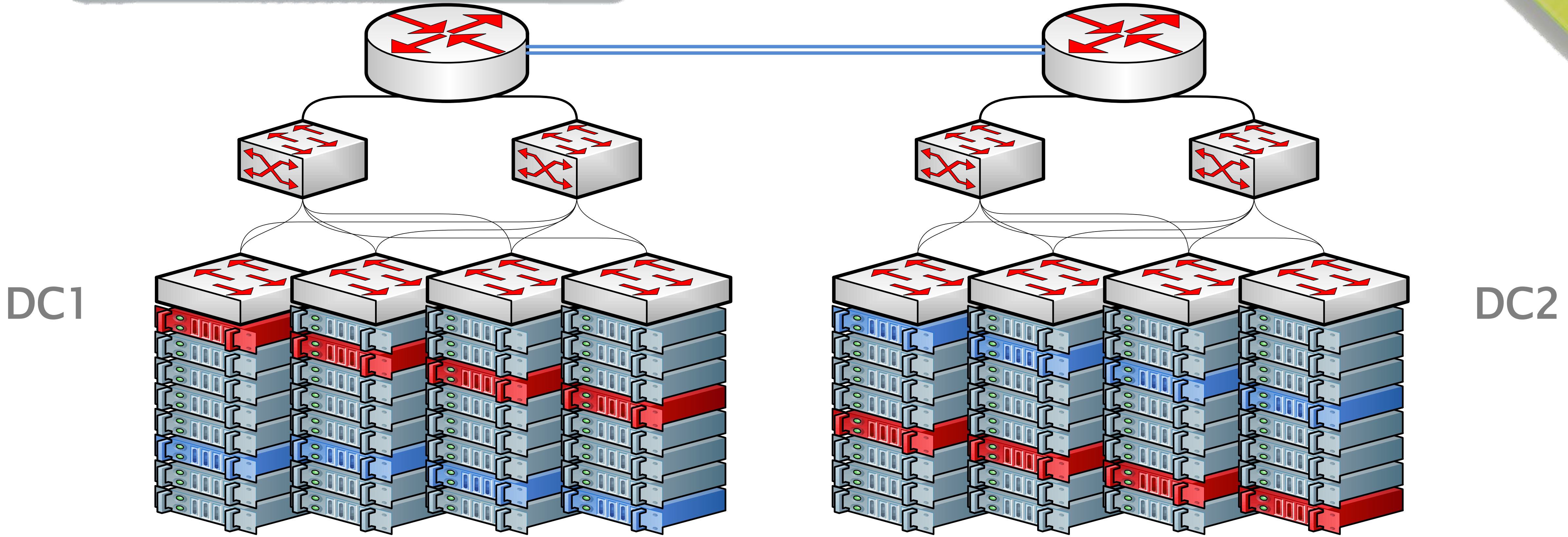
Find a balance!

Sberbank's affinity implementation



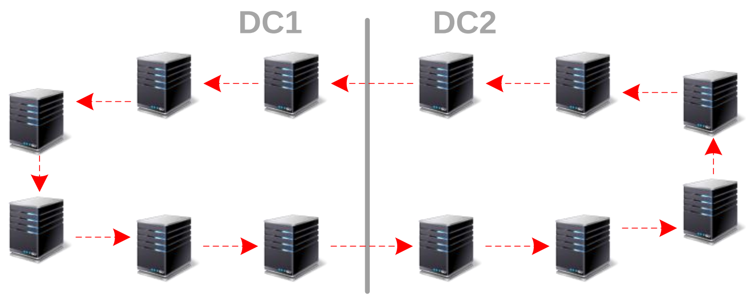
- The grid is distributed across 2 datacenters.
- Data connectivity is **limited to 8 nodes (a cell).**
- Every partition has the master copy and 3 backups.
- Each datacenter has 2 copies of a partition.
- Both datacenters are active.

SBERBANK CORE BANKING INFRASTRUCTURE

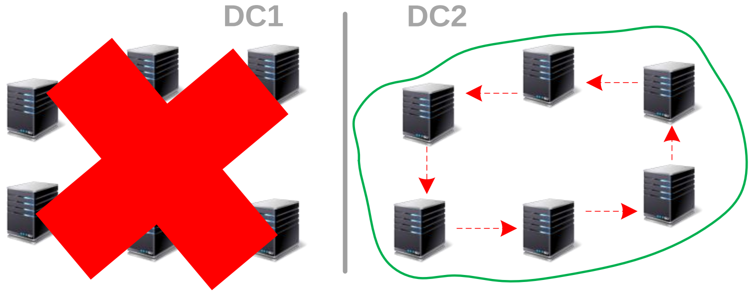


- Nodes of a cell reside in **different racks.**
- **Clos network** provides stable high-speed connectivity.
- **Doubled datacenter interconnect** reduces split-brain probability.
- Every server contains **NVMe flash and HDDs**.

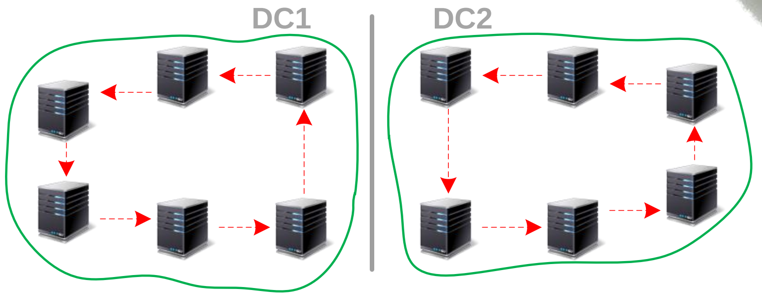
LET'S SPEAK ABOUT NETWORK FRAGMENTATION…



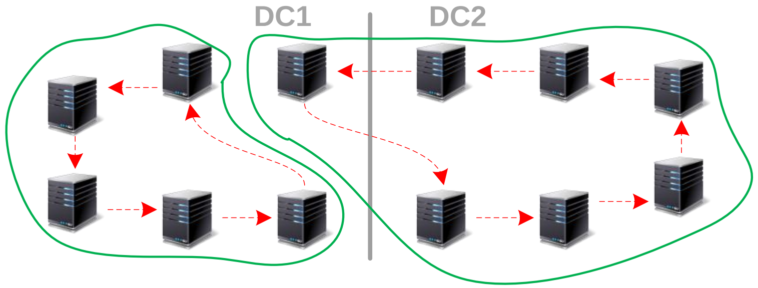
Regular operation



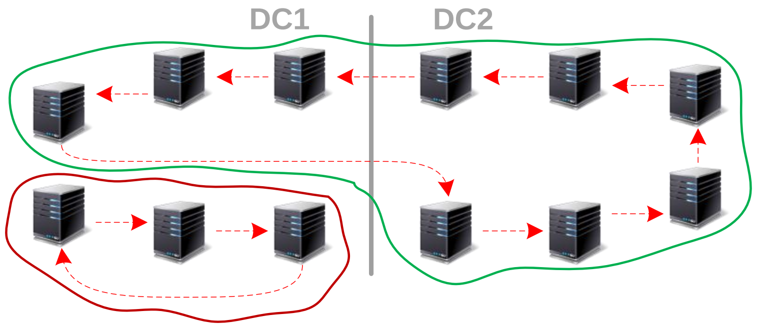
Datacenter loss



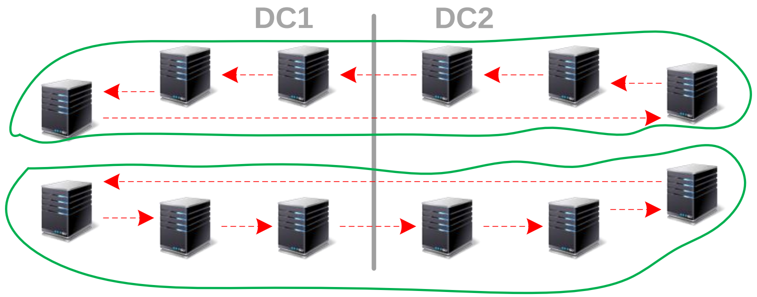
DC interconnect loss



Fragmentation type 1

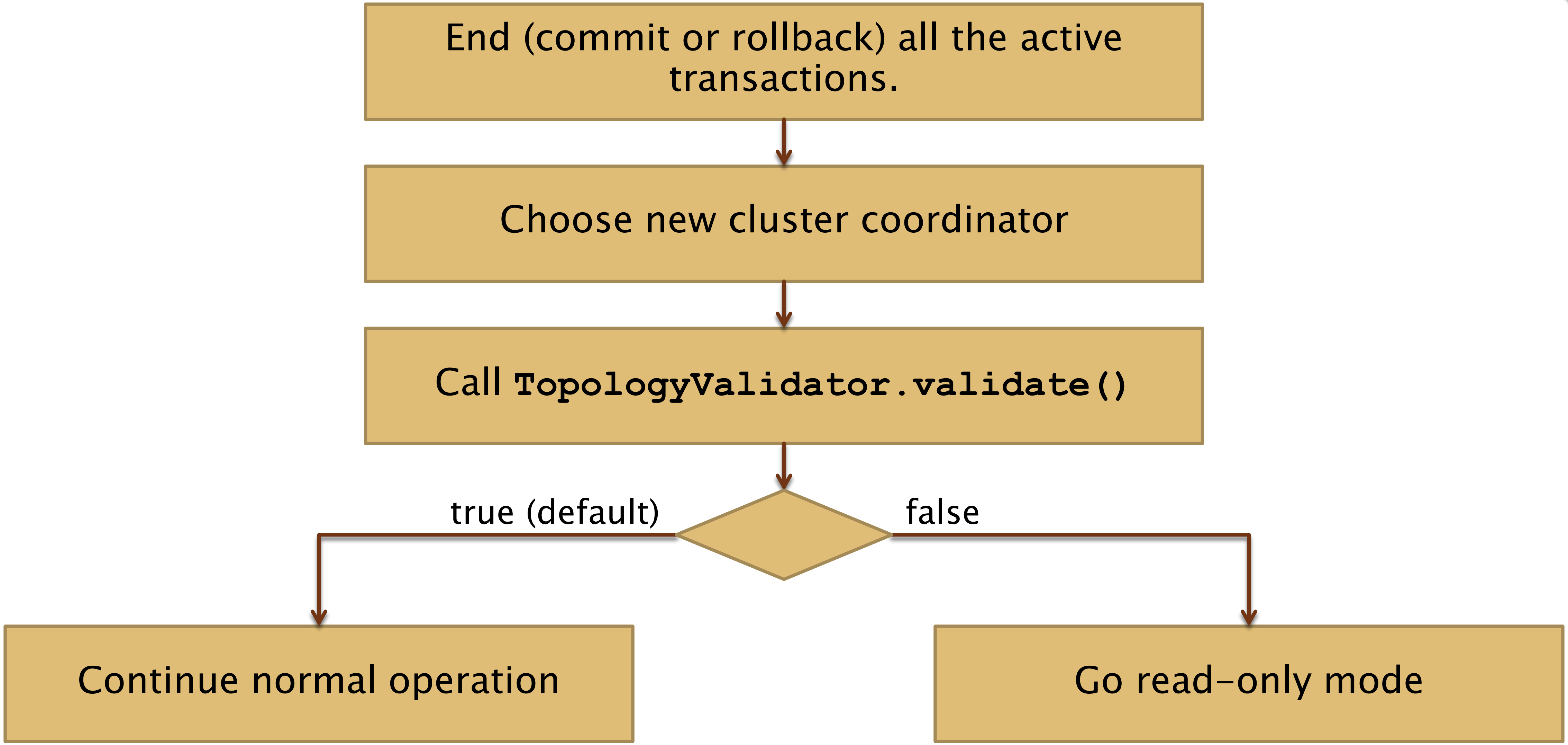


Fragmentation type 2



Fragmentation type 3

HOW DOES GRIDGAIN RECOVER A BROKEN CLUSTER?



LET'S OVERRIDE DEFAULT TOPOLOGY VALIDATOR!

Check if
- ❑ the previous topology was valid
- ❑ either the new nodes appear
  or not more than N nodes lost

yes

RW

no

Check

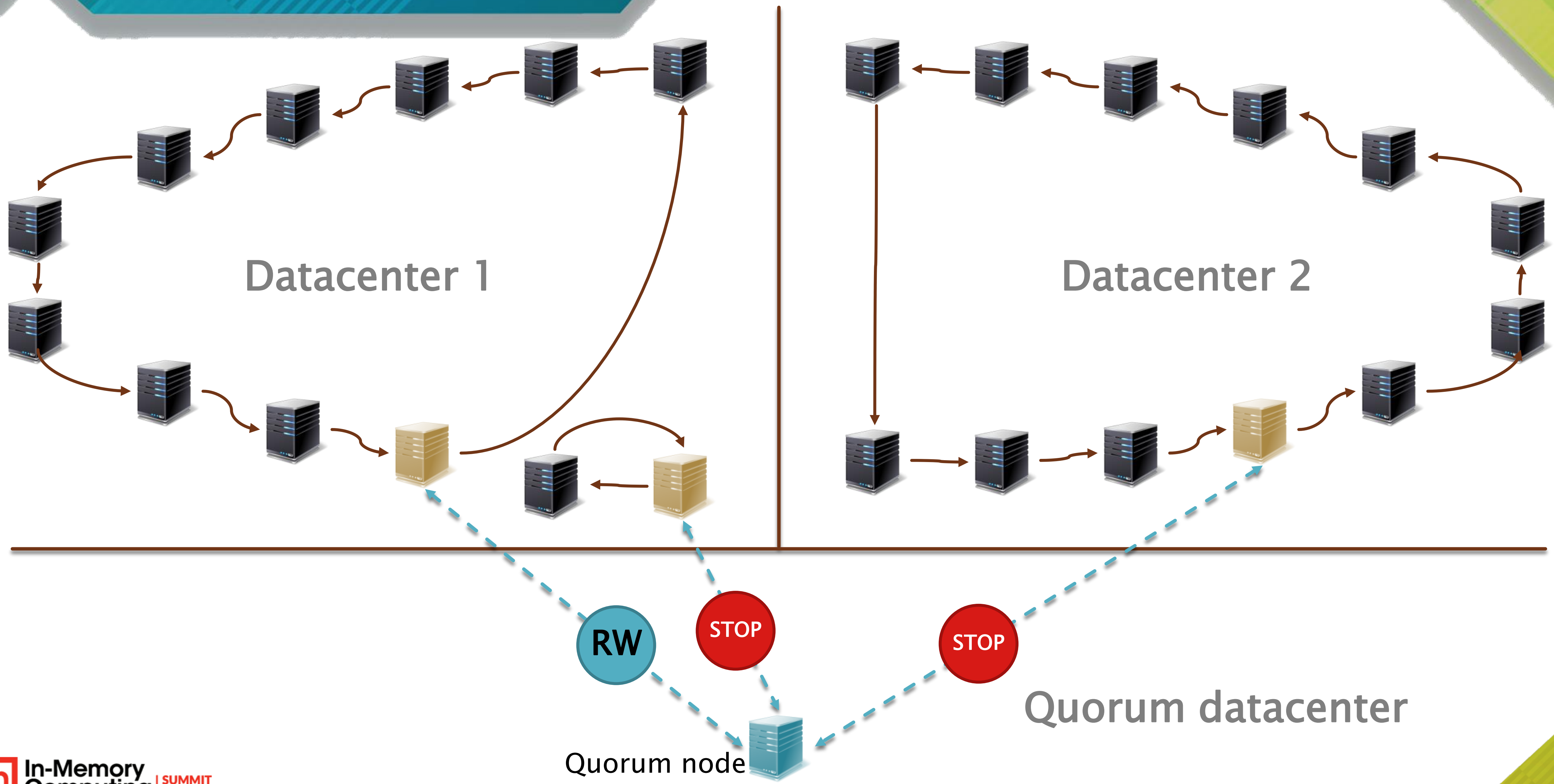
1. **there are nodes from DC1**
   - All
   - Partial
   - None
2. **there are nodes from DC2**
   - All
   - Partial
   - None
3. **data is integral** (no partition loss happens)
   - Yes
   - No

Decisions possible:

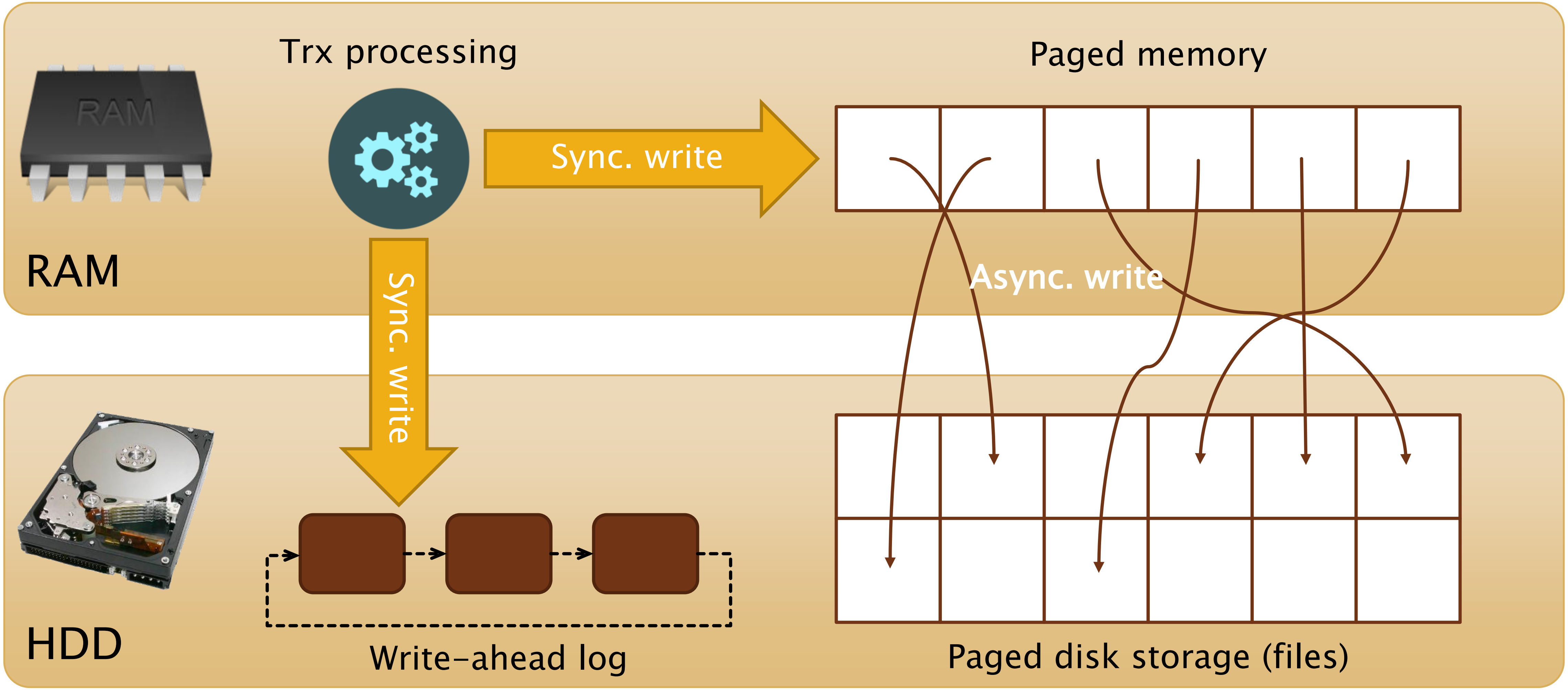
- RW (read-write): continue normal operation
- AW (admin wait): freeze the cluster and wait for admin interaction

DC1	DC2	Data	Decision
All	Partial	✓	RW
~~All~~	~~Partial~~	~~✗~~	
All	None	✓	RW
~~All~~	~~None~~	~~✗~~	
Partial	All	✓	AW

DECISION AUTOMATION USING QUORUM NODE



LOCAL FILE STORE (LFS)



BACKUP SUBSYSTEM

Current

- **Snapshot** to local disk (full/incremental/differential)
- Snapshot **catalog** inside the data grid
- Copying to NAS **using NFS**
- Restoring on **arbitrary grid topology**

Future

- **Point-in-time recovery** using snapshot and WAL;
- **External backup catalog** in relational DBMS;
- Copying to SDS using **S3/SWIFT**;
- ...and more!

THANK YOU!

Vladimir Komarov <vikomarov@sberbank.ru>

Mikhail Gorelov <magorelov@sberbank.ru>