Global Resource Management in Oracle RAC

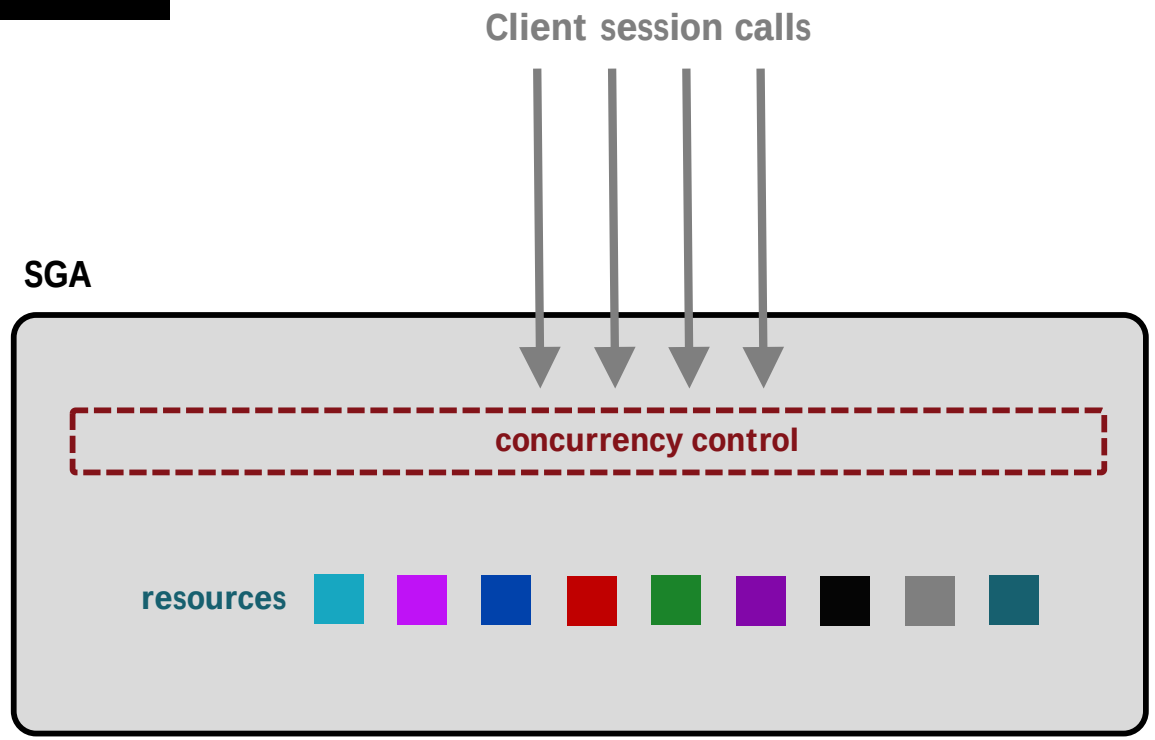
By Ahmed Baraka

Objectives

In this lecture, you learn to understand the following concepts:

- Global Concurrency Control
- Global Resource Directory (GRD)
- Mastering and shadowing instances
- Global Cache Management scenarios for single block access

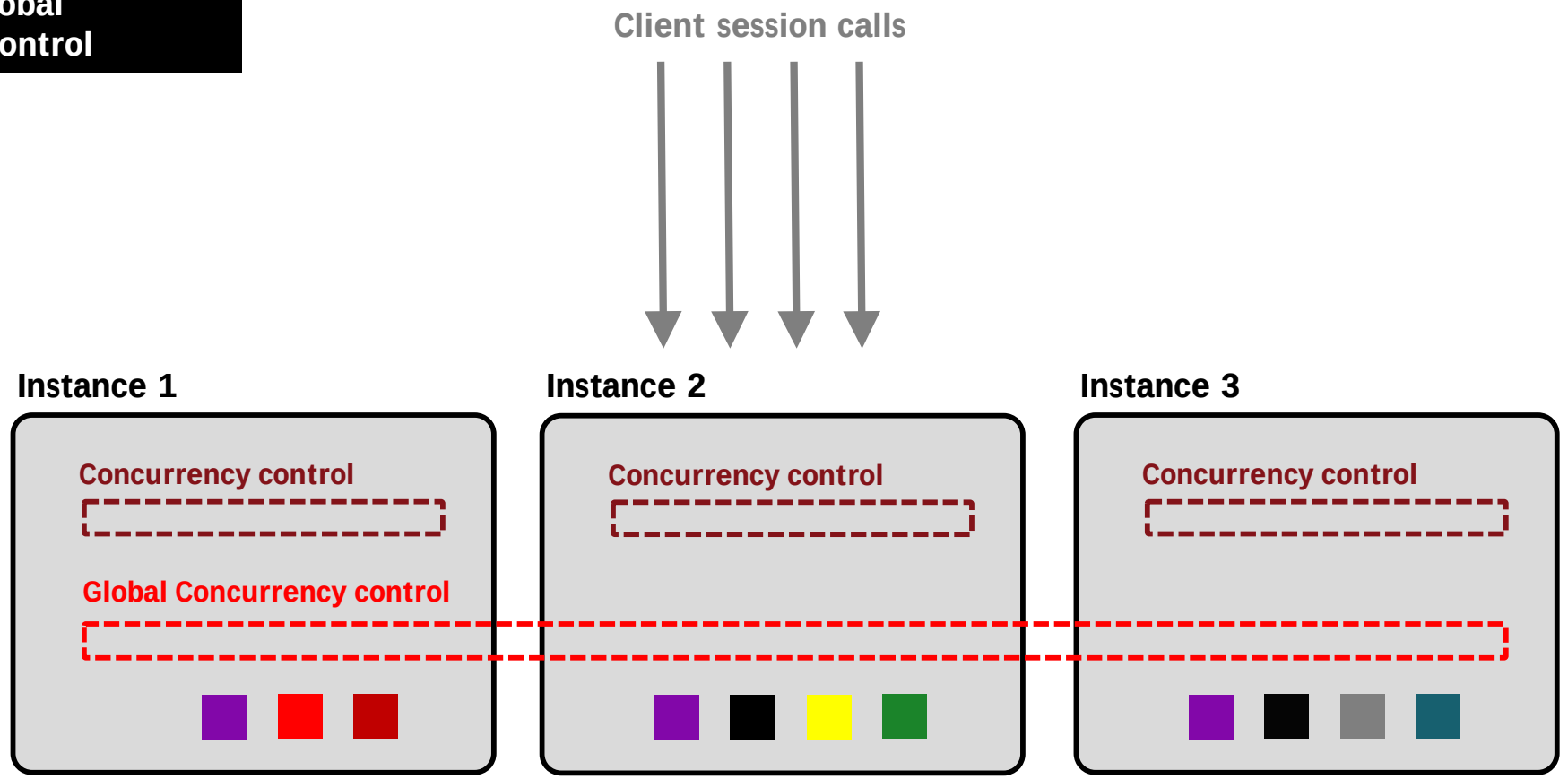
Single Instance Concurrency Control



Concurrency control:

- Memory structure: latches or mutexes
- Resource control: enqueue
- Cache management: buffer cache pins

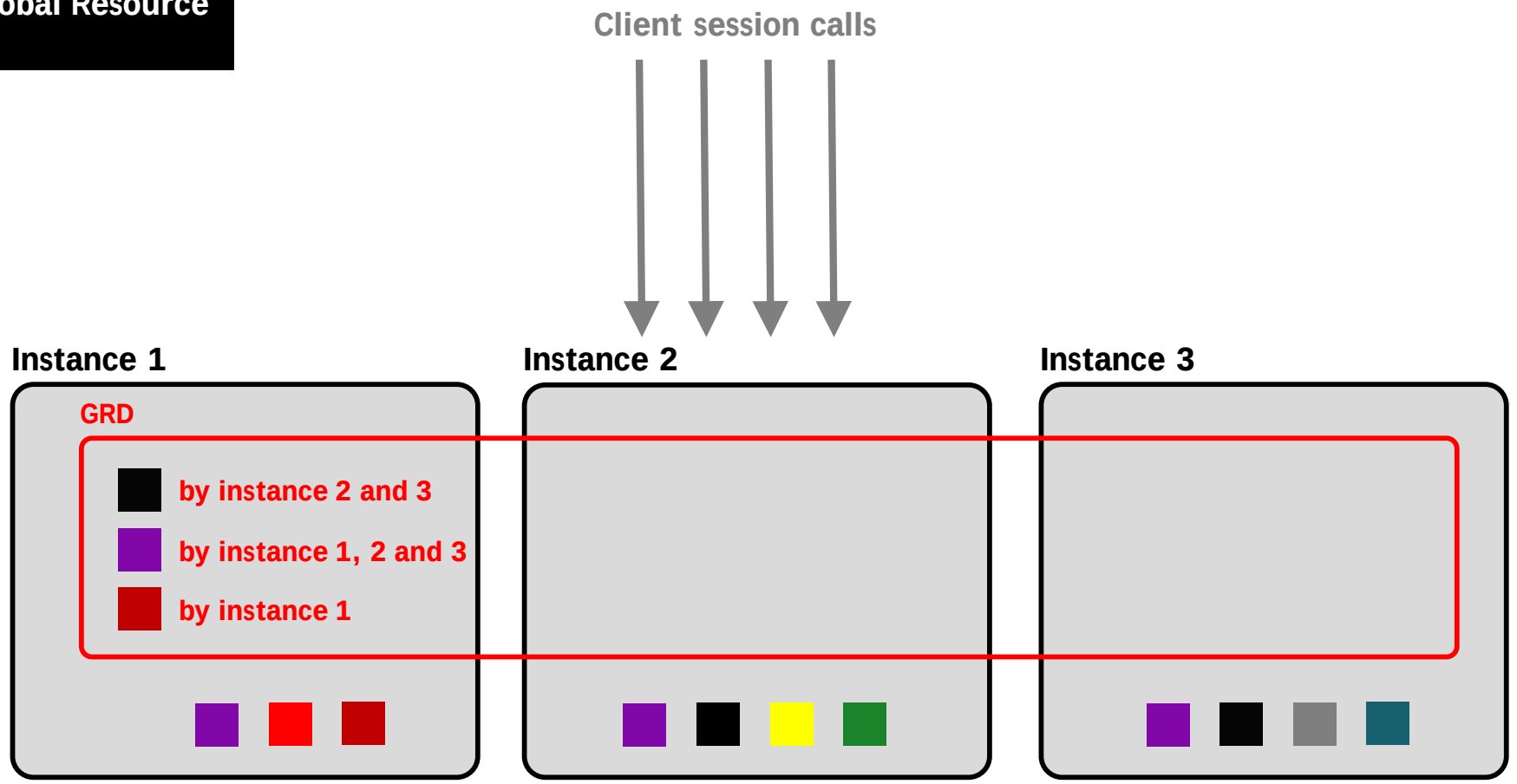
Oracle RAC: Global Concurrency Control



Global Concurrency control:

- Library and row cache access: global locks control
- Resource access: global enqueues
- Buffer cache access: cache fusion

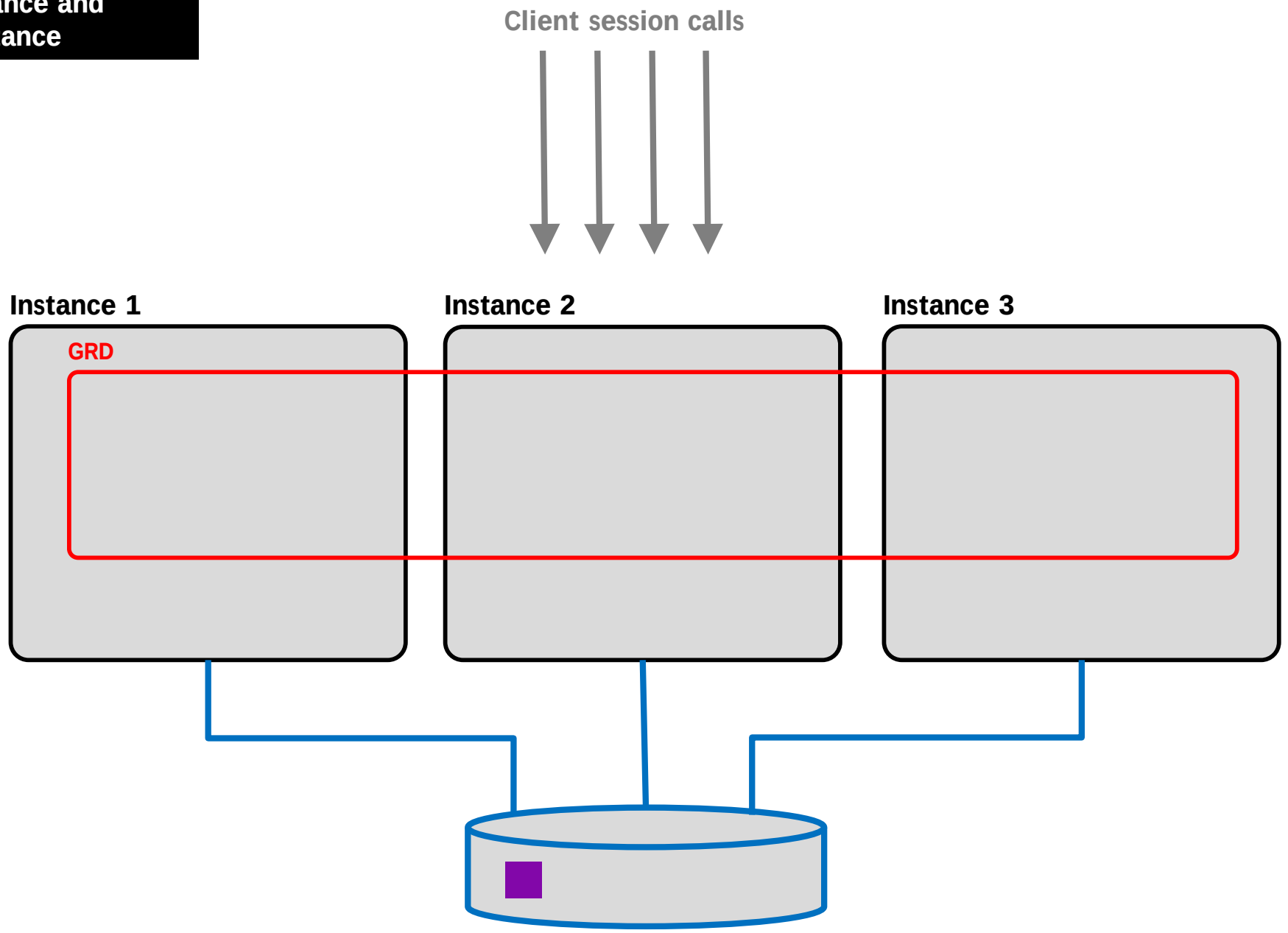
Oracle RAC: Global Resource Directory



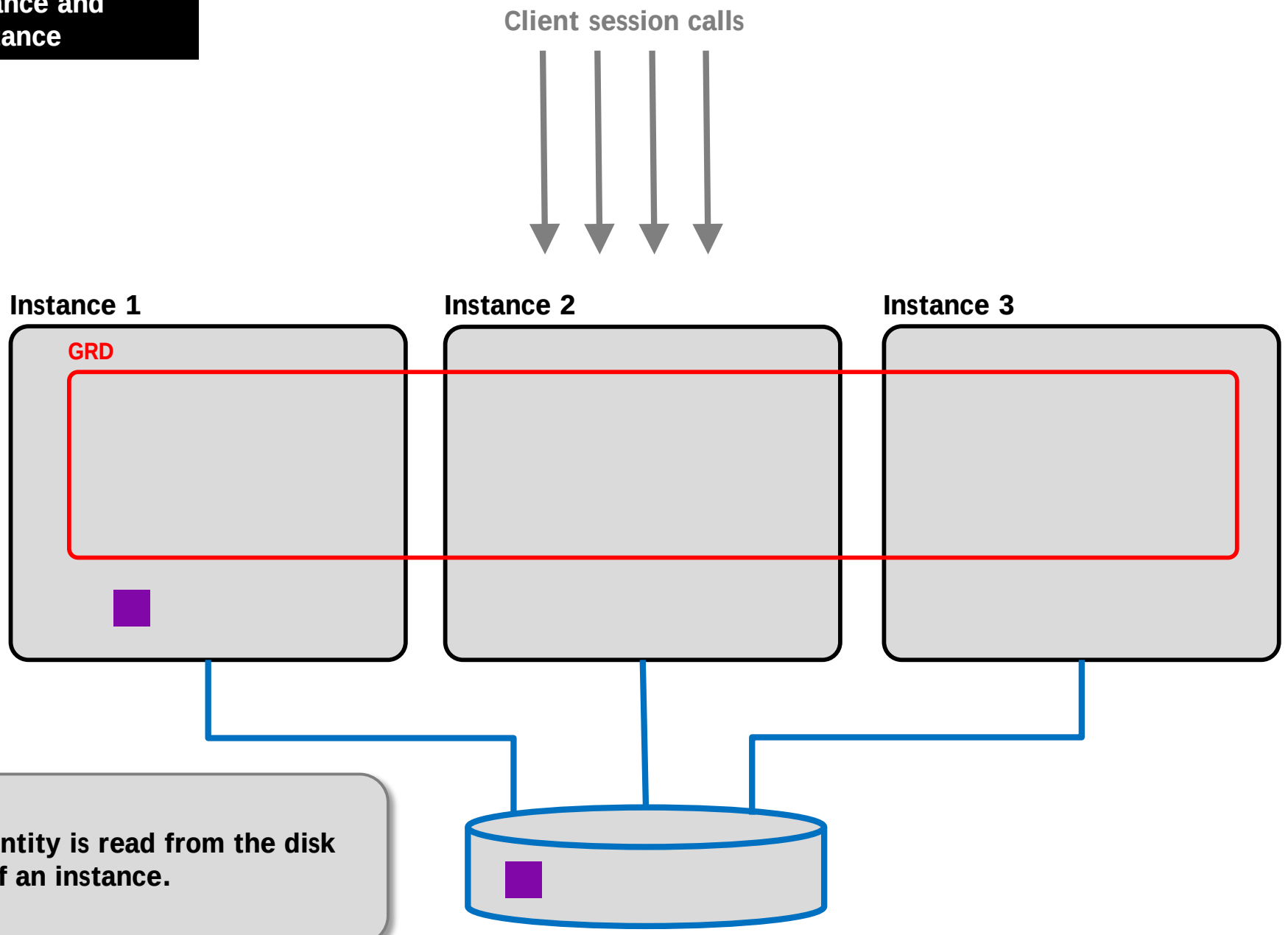
About Global Resource Directory (GRD)

- An object under global concurrency control is called a resource.
- Resource metadata is held in the Global Resource Directory (GRD).
- The GRD is distributed among all active instances of each database or ASM environment.
- The GRD uses memory from the shared pool
- It contains metadata of which instances are holding which resources

Mastering Instance and Shadowing Instance

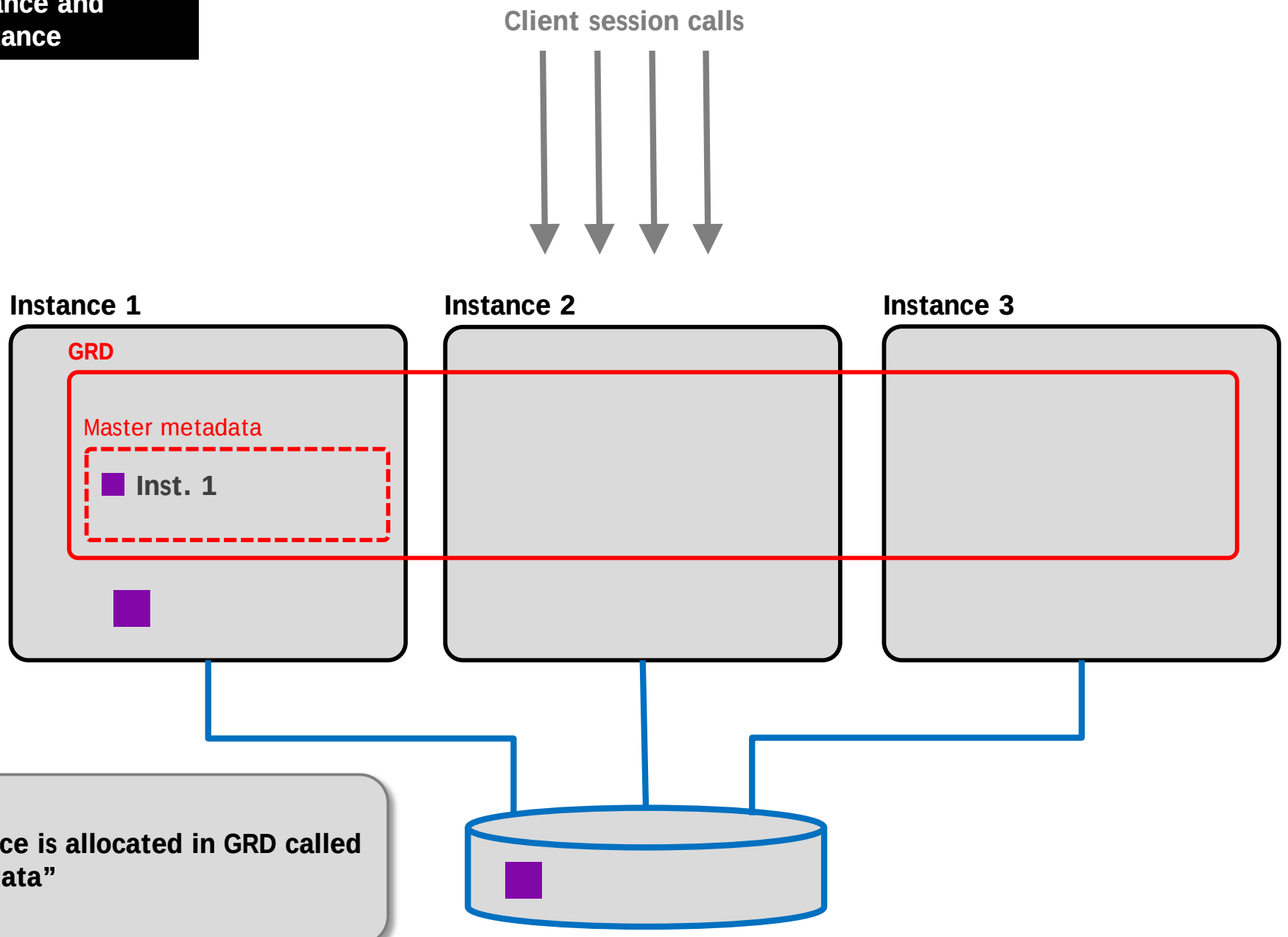


Mastering Instance and Shadowing Instance



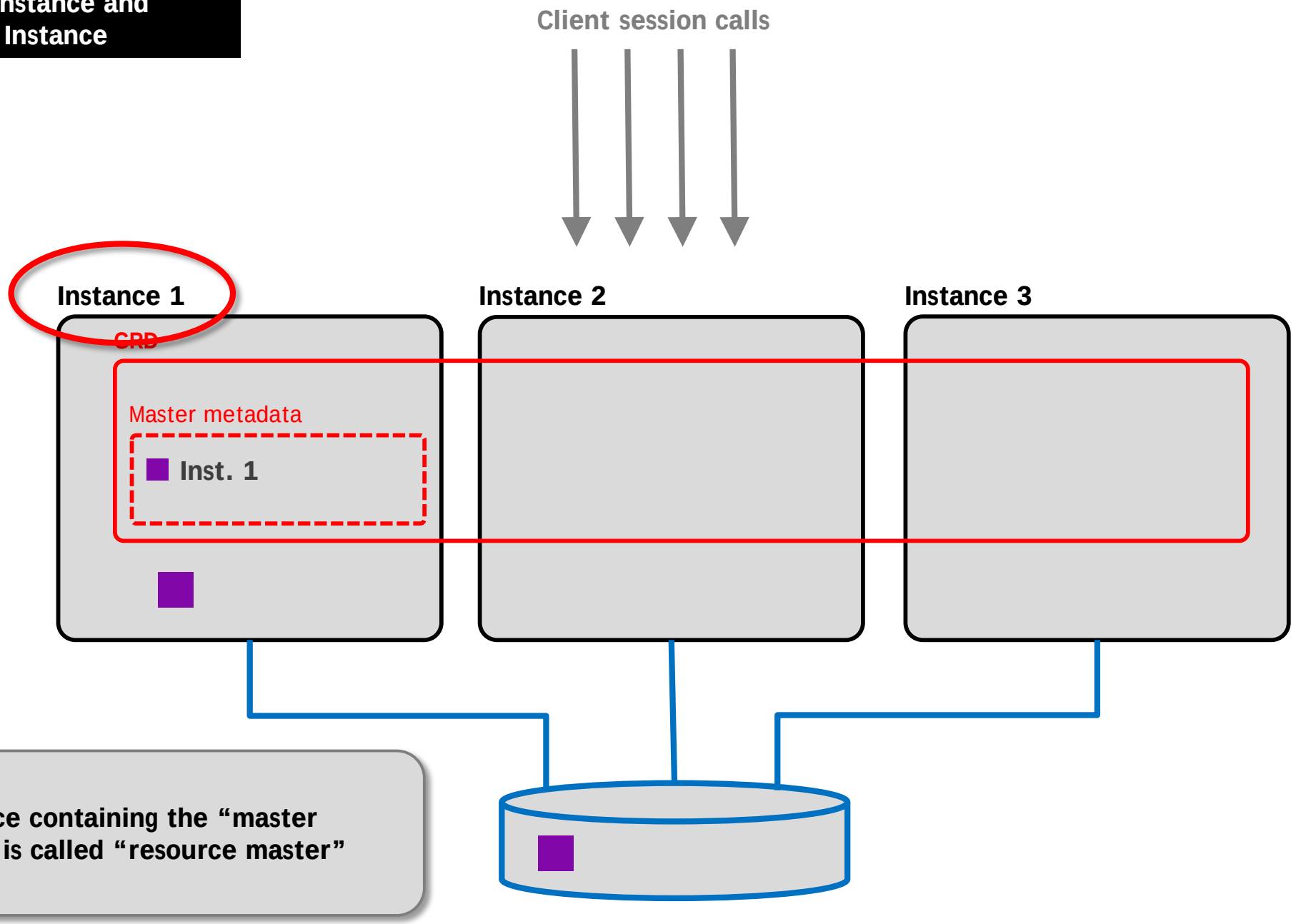
The required entity is read from the disk to the buffer of an instance.

Mastering Instance and Shadowing Instance



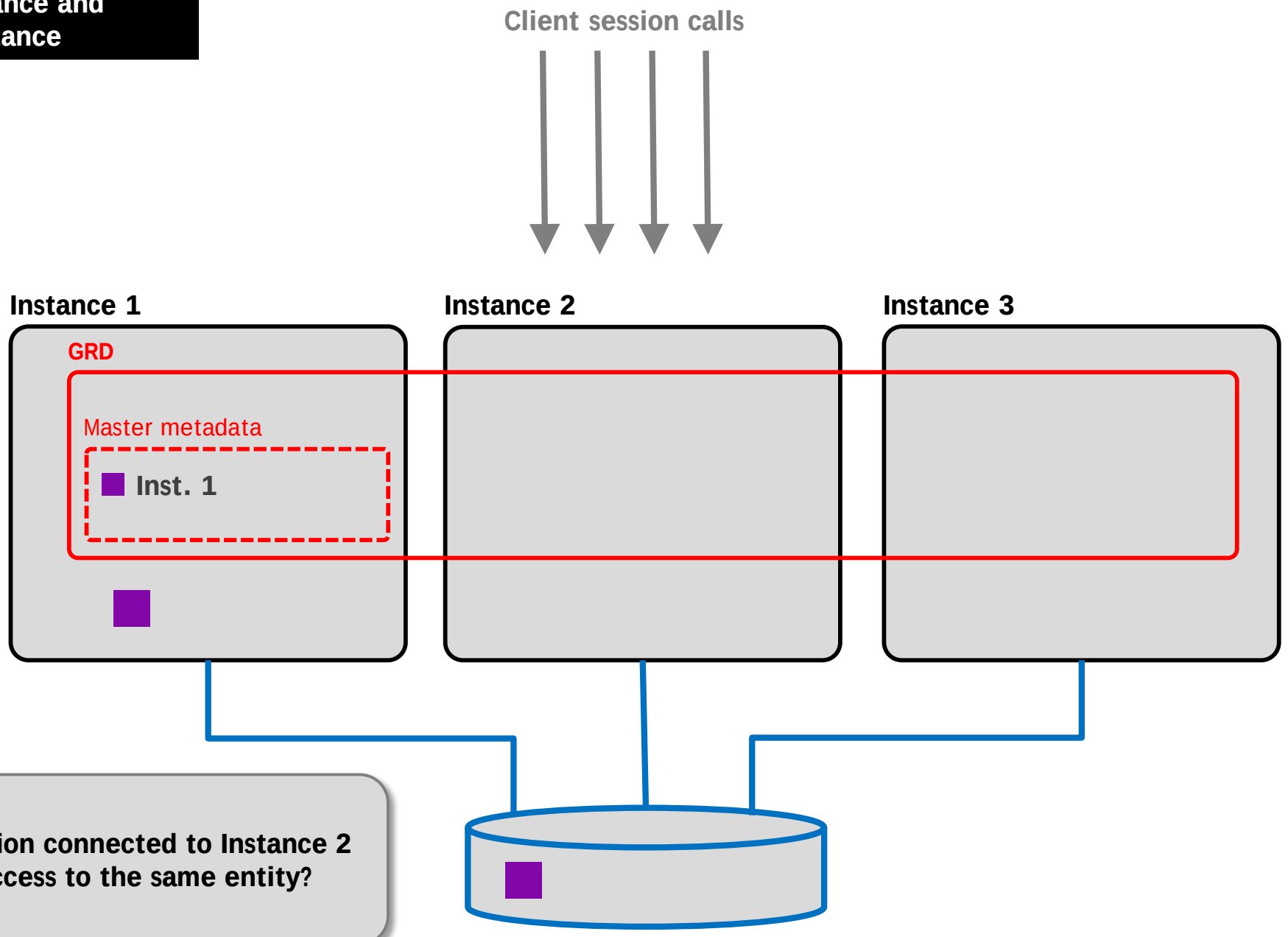
A global resource is allocated in GRD called "master metadata"

Mastering Instance and Shadowing Instance



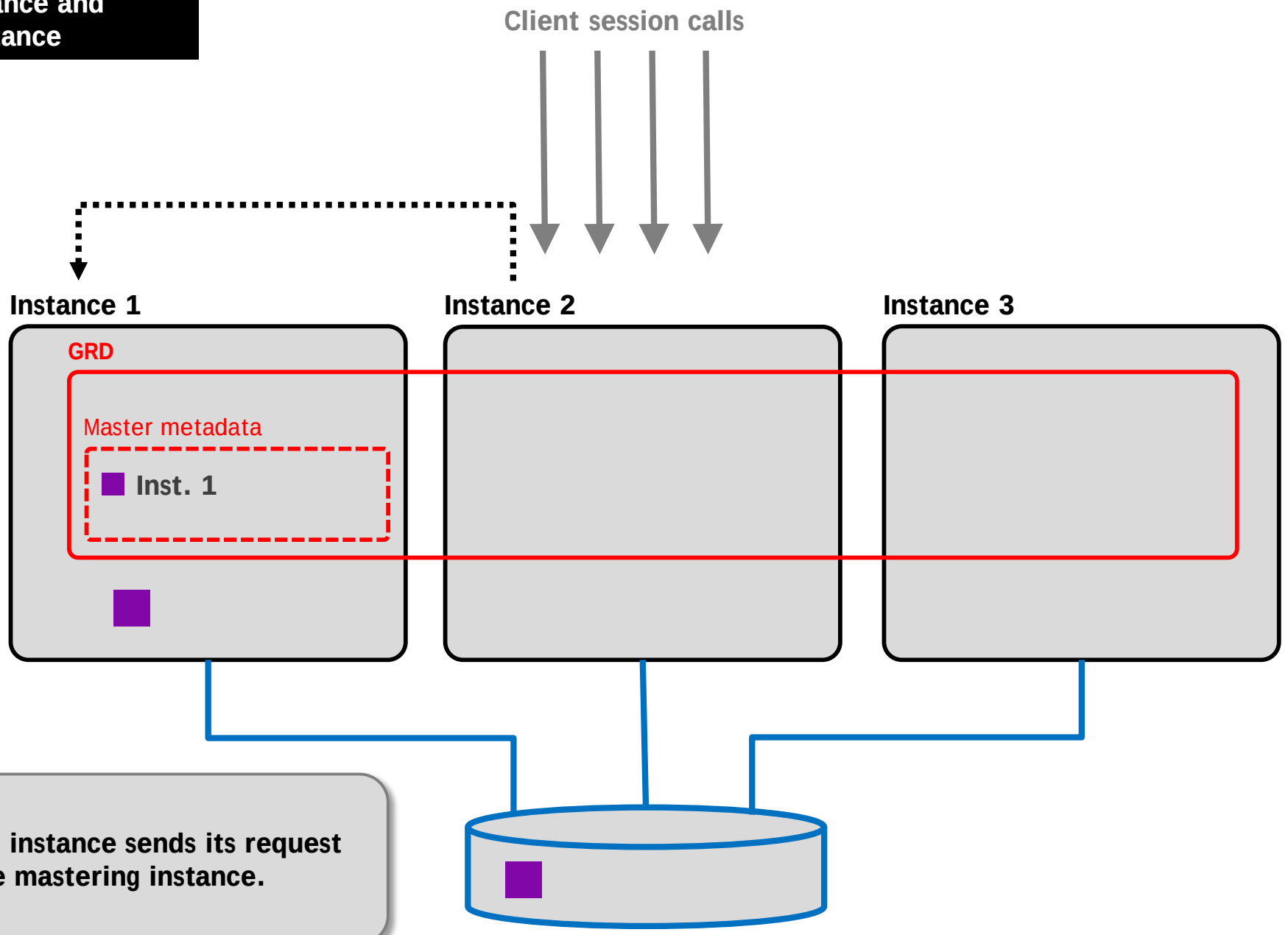
The instance containing the “master metadata” is called “resource master”

Mastering Instance and Shadowing Instance



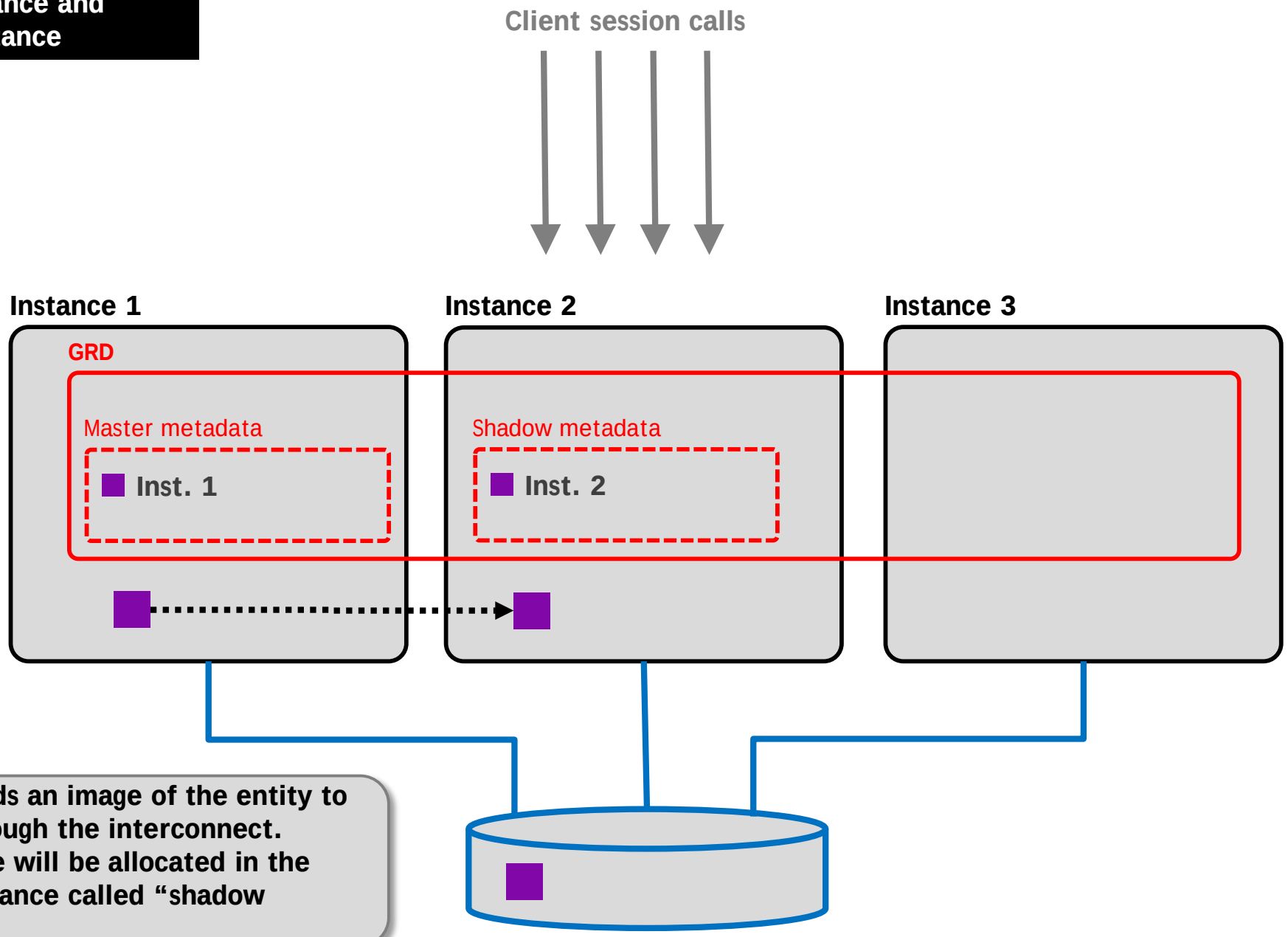
Suppose a session connected to Instance 2 wants to get access to the same entity?

Mastering Instance and Shadowing Instance



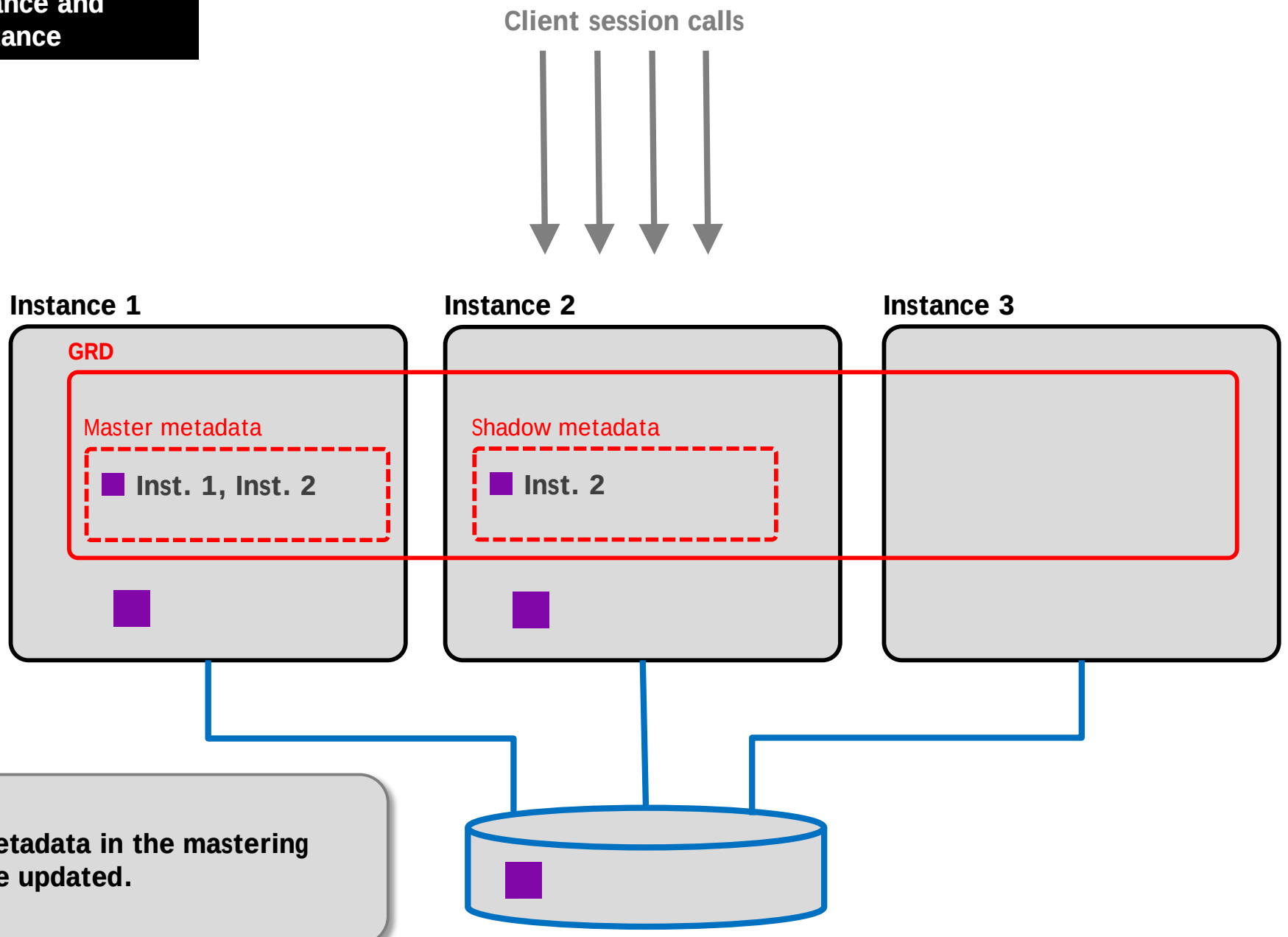
The requesting instance sends its request to the resource mastering instance.

Mastering Instance and Shadowing Instance



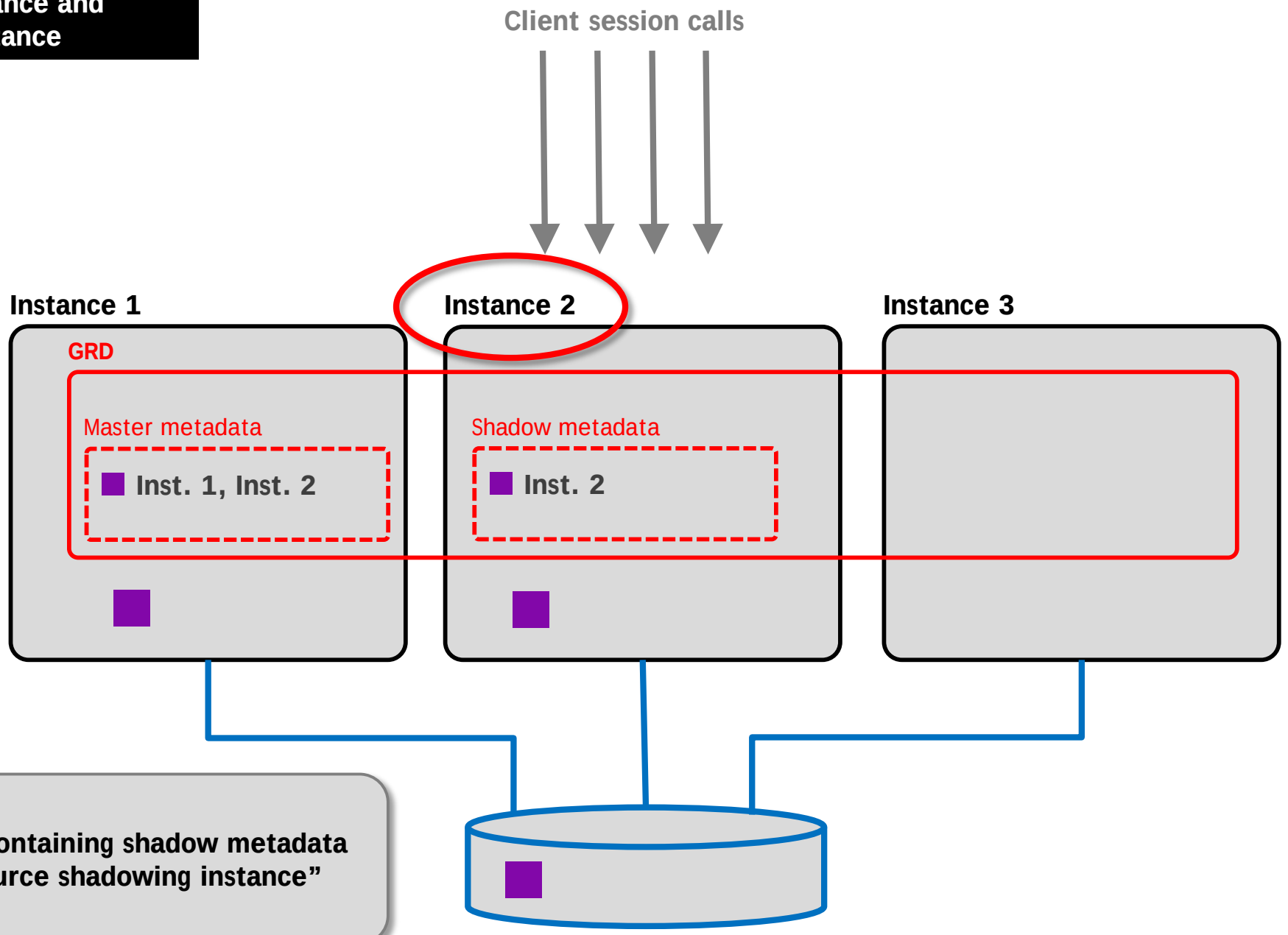
Instance 1 sends an image of the entity to instance 2 through the interconnect.
A GRD resource will be allocated in the requesting instance called “shadow metadata”.

Mastering Instance and Shadowing Instance



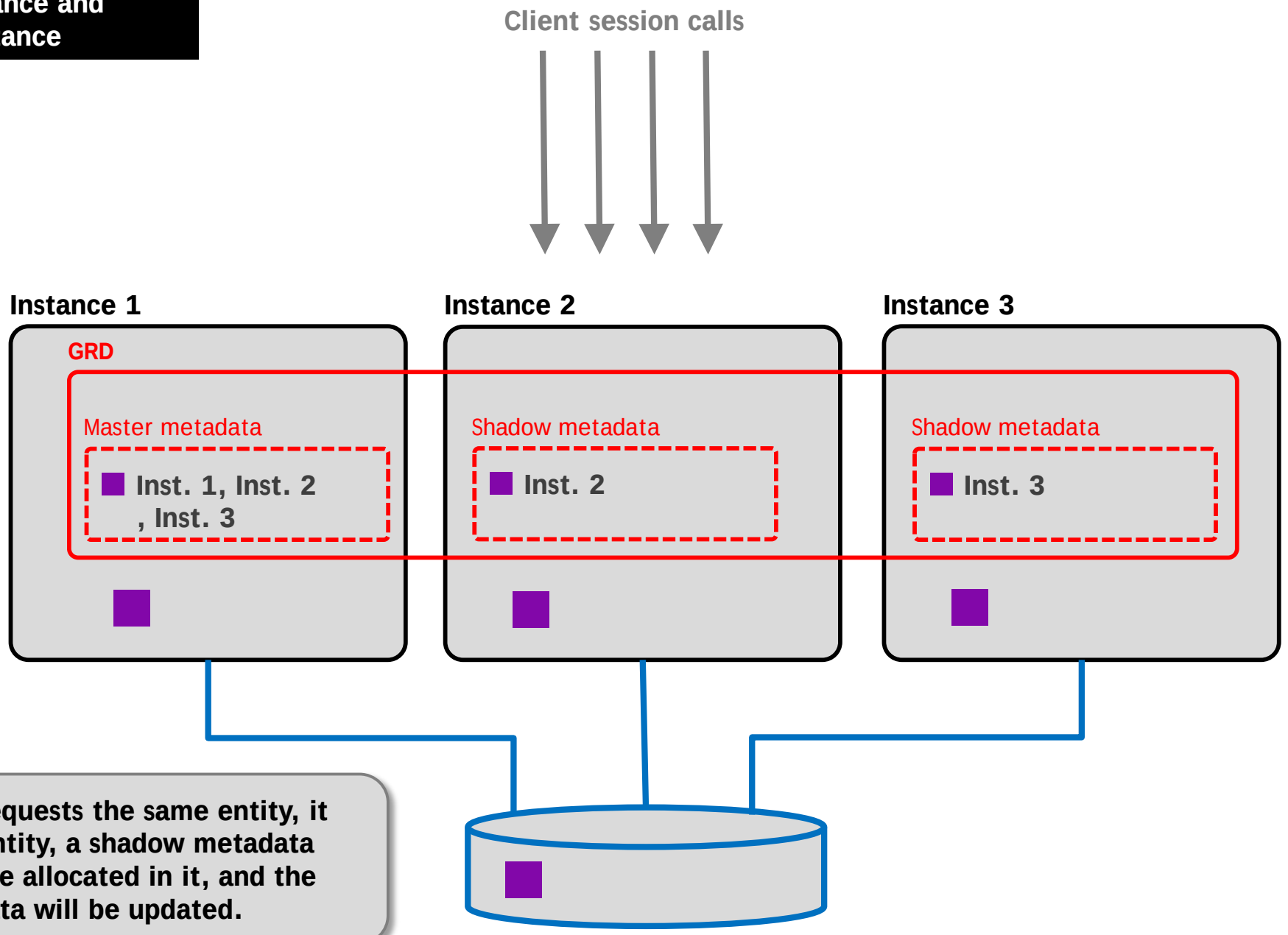
The master metadata in the mastering instance will be updated.

Mastering Instance and Shadowing Instance



The instance containing shadow metadata is called “resource shadowing instance”

Mastering Instance and Shadowing Instance



If instance 3 requests the same entity, it receives the entity, a shadow metadata resource will be allocated in it, and the master metadata will be updated.

Global Resource Management Terms

- A resource is allocated the first time an entity is accessed
- Mastering instance is the one that contains the master metadata
- A shadow metadata is allocated in any of the instances that requests the same entity.
- Master metadata contains locking information of the entity in **all** the instances, whereas shadow metadata contains locking information of the entity in the **current** instance
- Each instance could be the resource master for some of the database entities

Global Resource Remastering

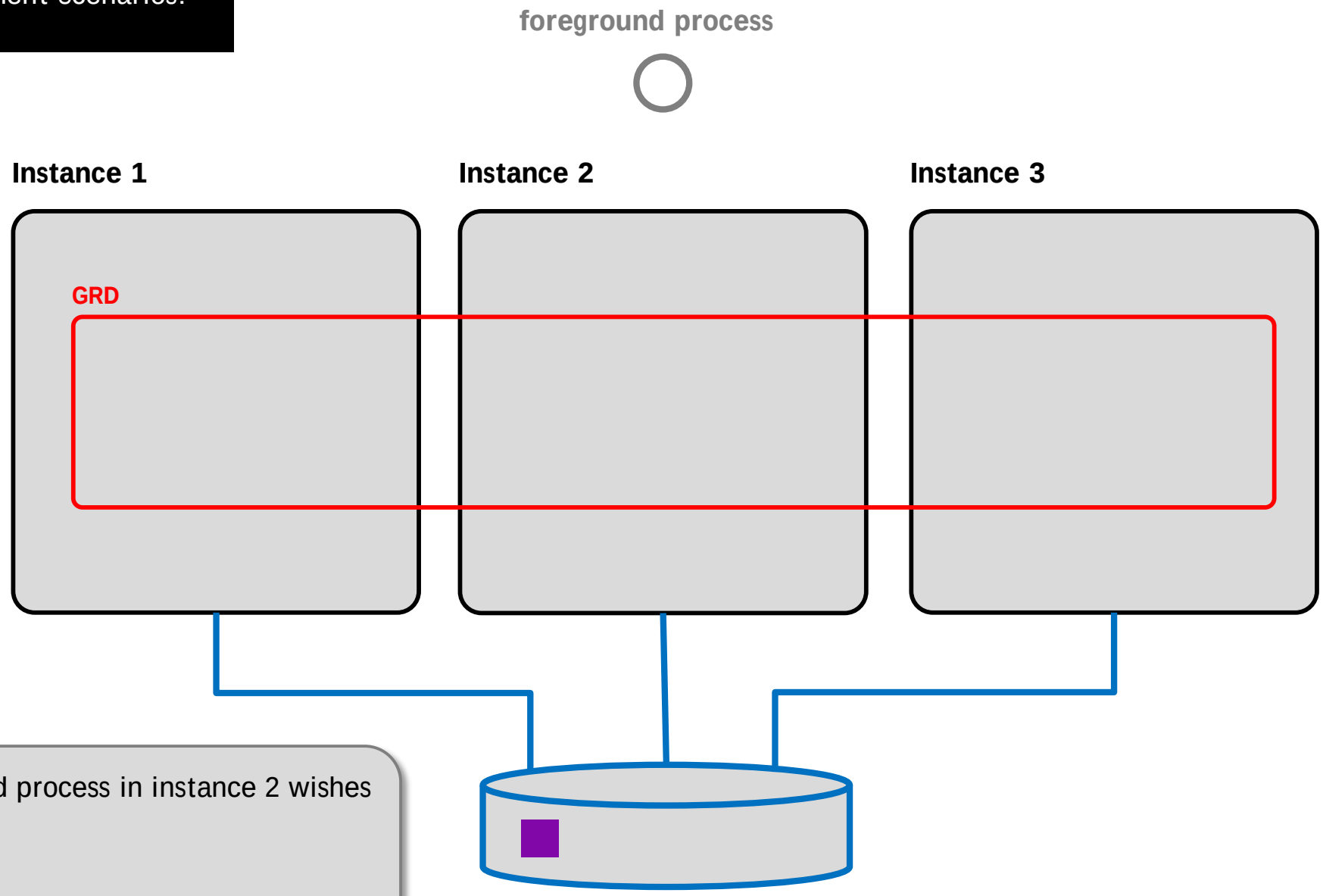
- Remastering: master metadata for a specific entity move to another instance
- It could happen in three levels:
  - **Instance-level** (also called lazy remastering) occurs when the mastering instance is gracefully shut down or a new instance starts up
  - **File affinity** remastering occurs when requests to access a specific **datafile** blocks come from an instance other than the mastering instance
  - **Object-affinity** remastering occurs when requests to access a specific **object** blocks come from an instance other than the mastering instance

Global Cache Management Scenarios for Single Block Access

There are several scenarios for single block reads:

- Read from Disk
- Read - Read
- Read - Write
- Write - Write
- Write - Read
- Write to Disk

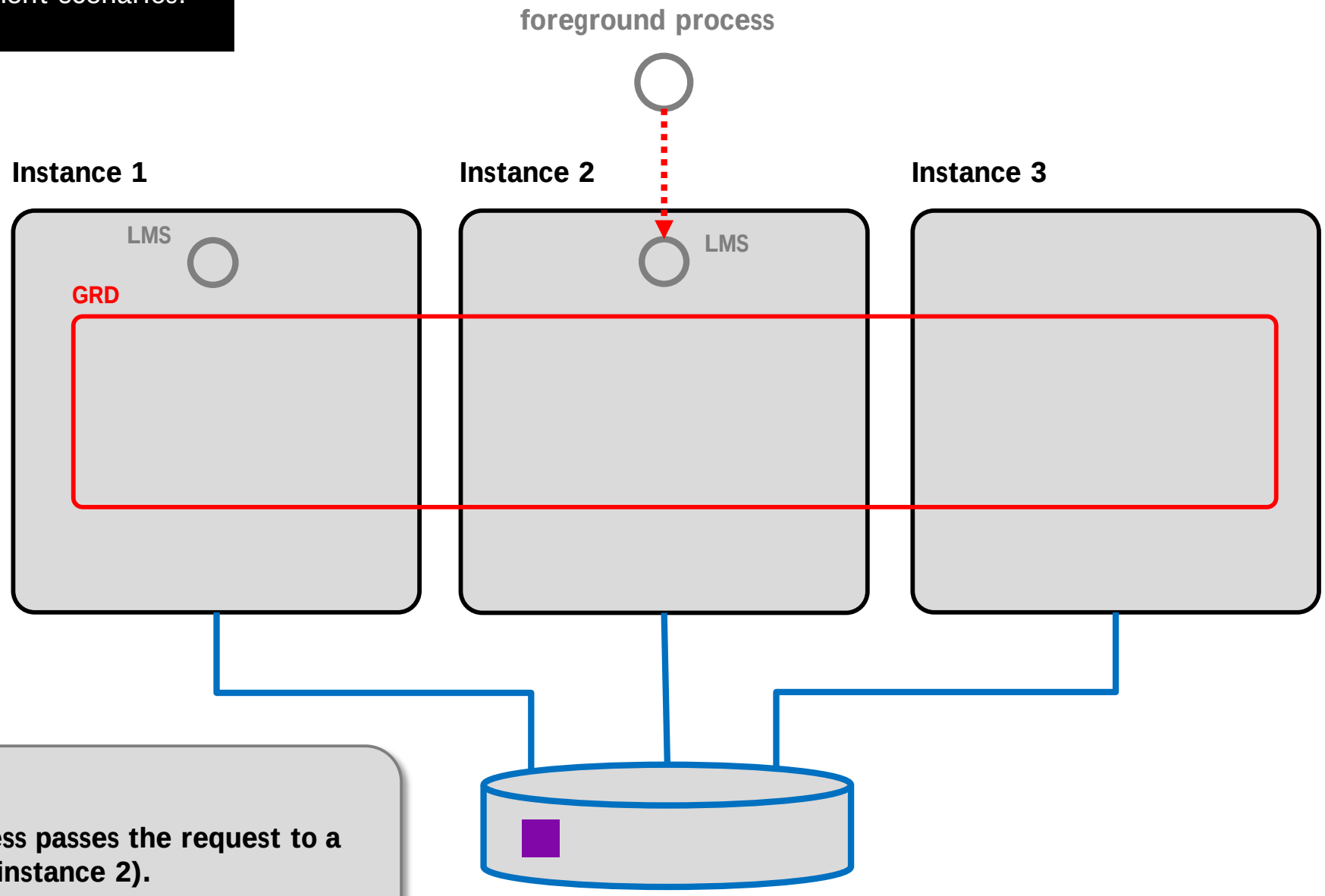
Global Cache Management Scenarios: **Read from Disk**



Scenario: A foreground process in instance 2 wishes to read a block.

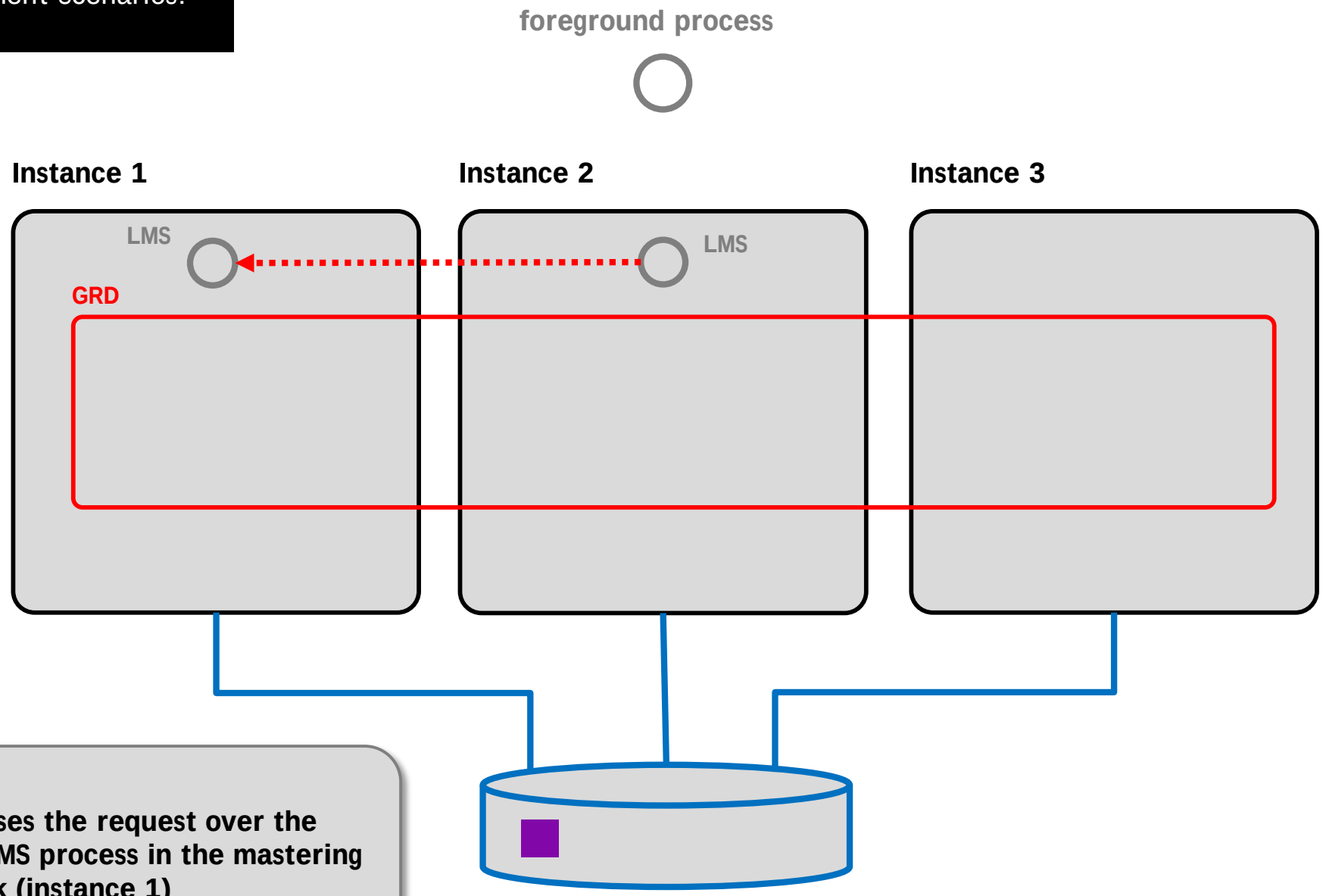
Assumptions:
Instance 1 is the mastering instance 1.

Global Cache Management Scenarios: **Read from Disk**



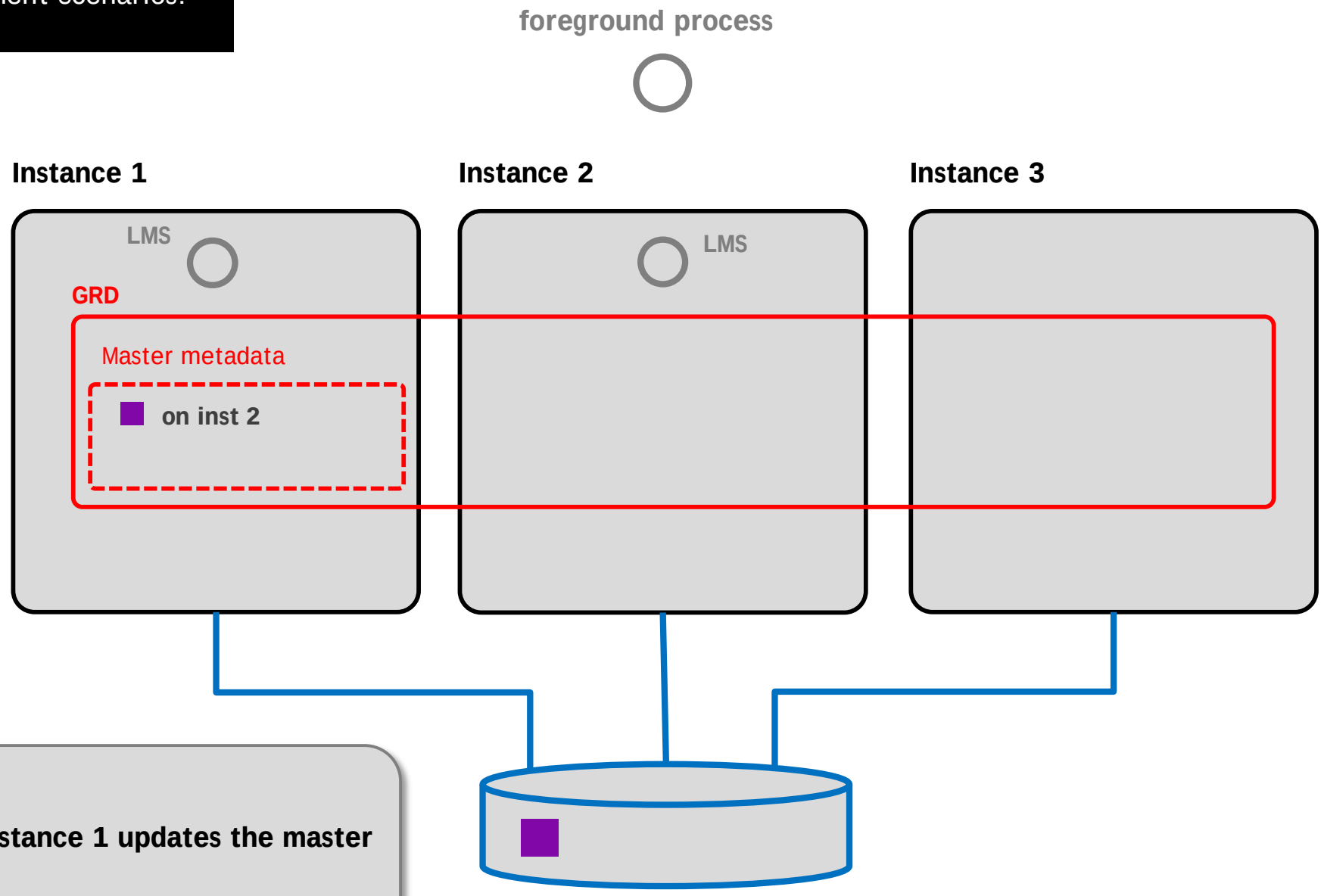
The foreground process passes the request to a local LMS process (in instance 2).

Global Cache Management Scenarios: **Read from Disk**



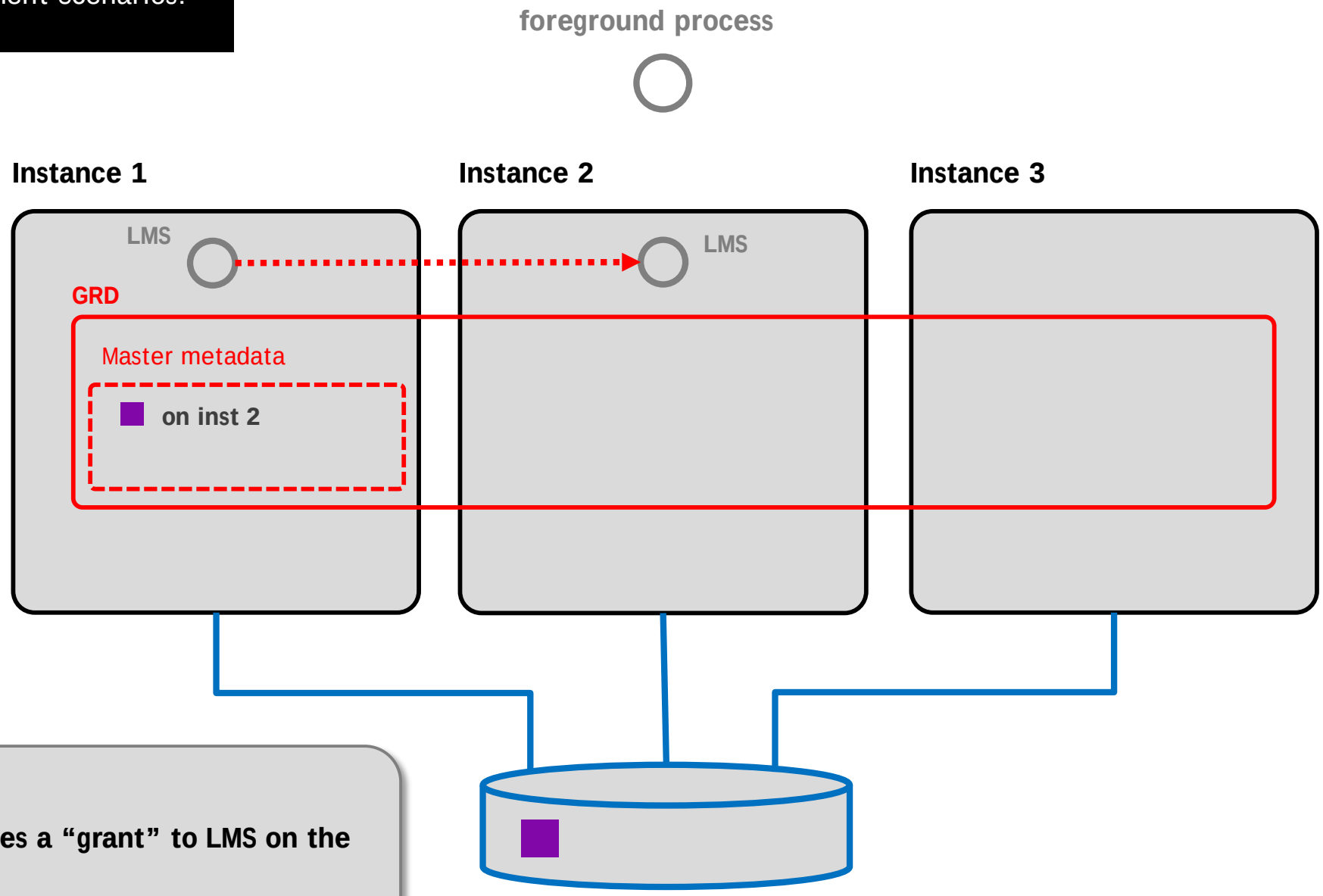
LMS in instance 2 passes the request over the interconnect to the LMS process in the mastering instance for this block (instance 1)

Global Cache Management Scenarios: **Read from Disk**



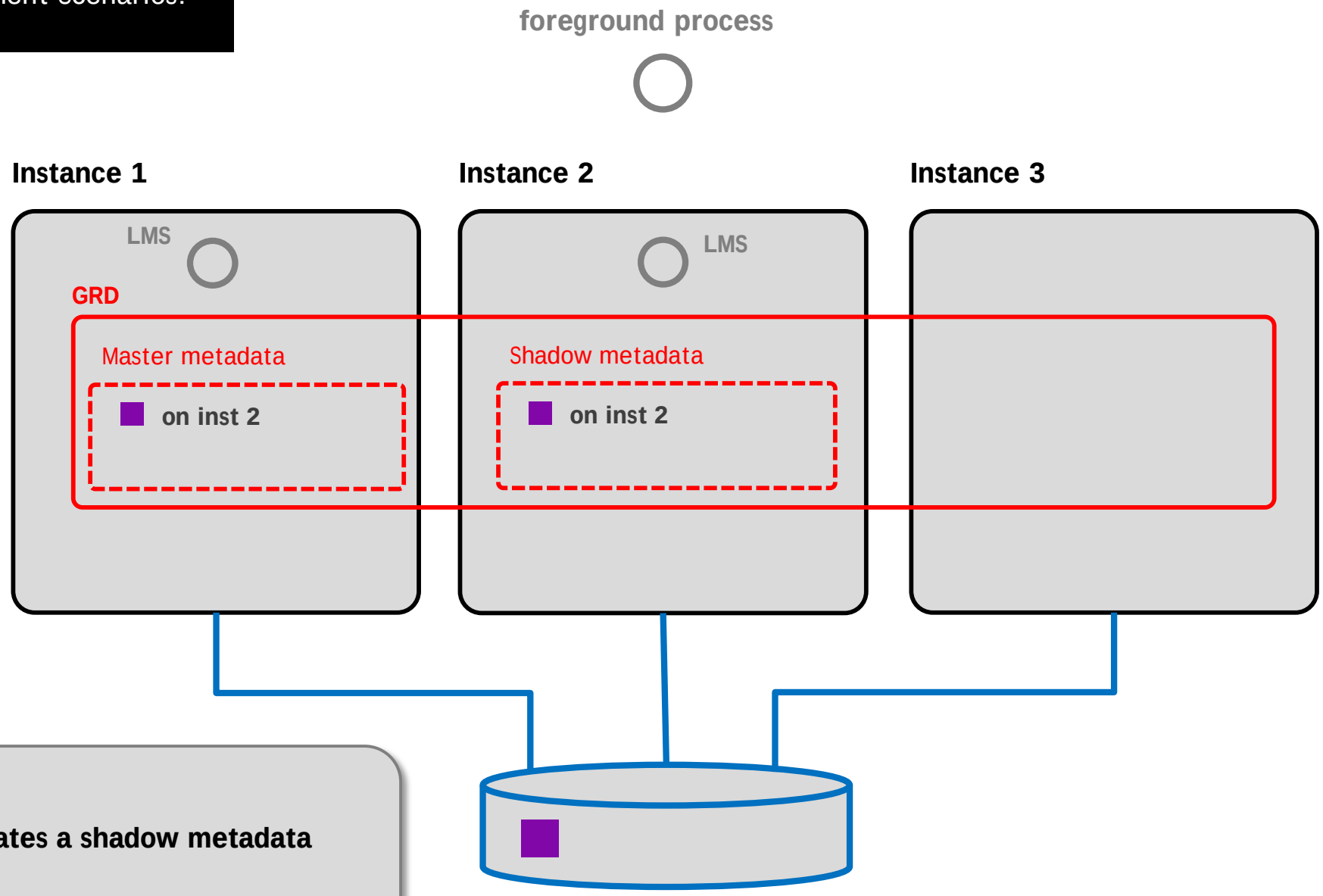
The LMS process in instance 1 updates the master metadata.

Global Cache Management Scenarios: **Read from Disk**



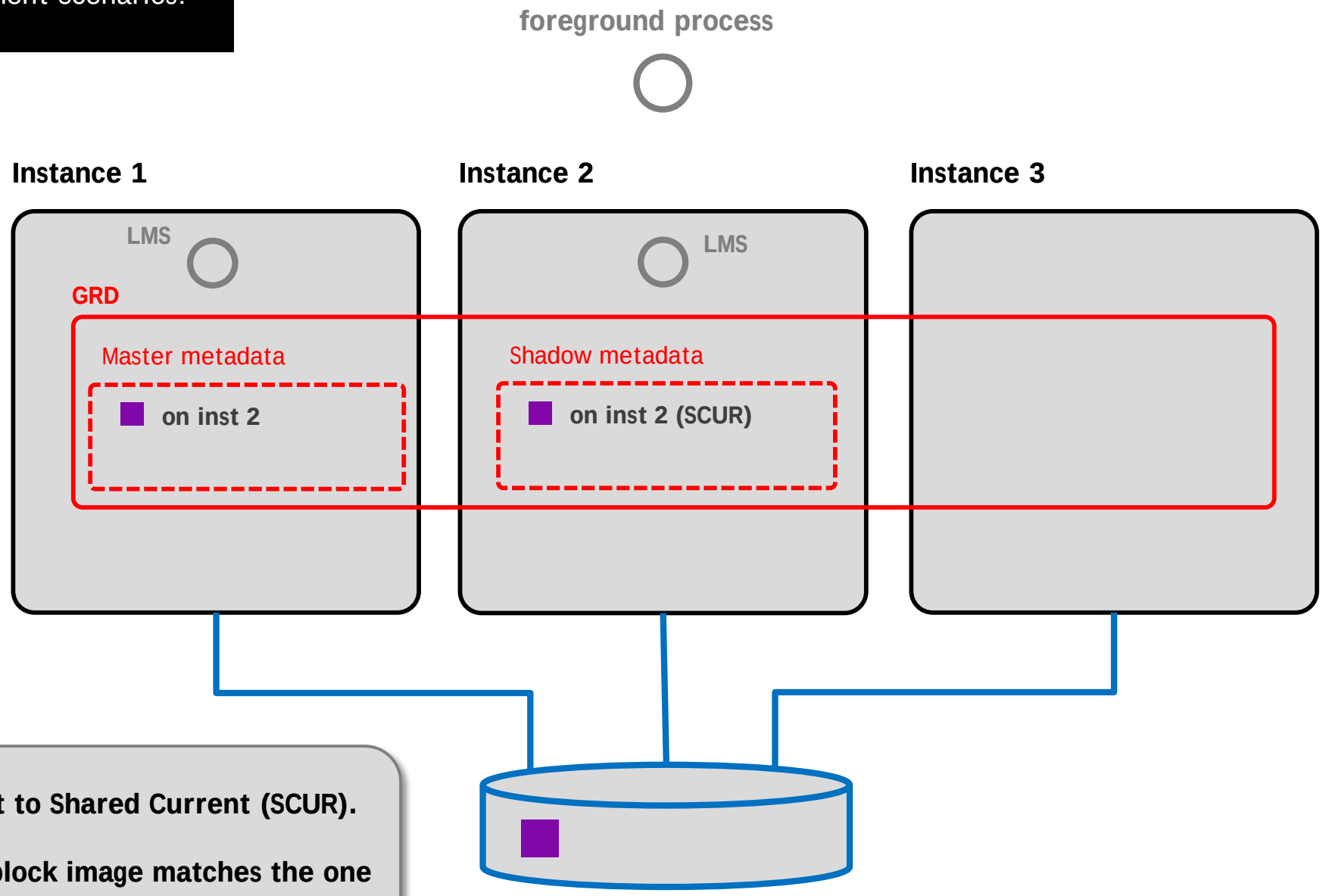
LMS in Instance 1 issues a "grant" to LMS on the requesting instance 2

Global Cache Management Scenarios: **Read from Disk**



LMS in instance 2 creates a shadow metadata resource.

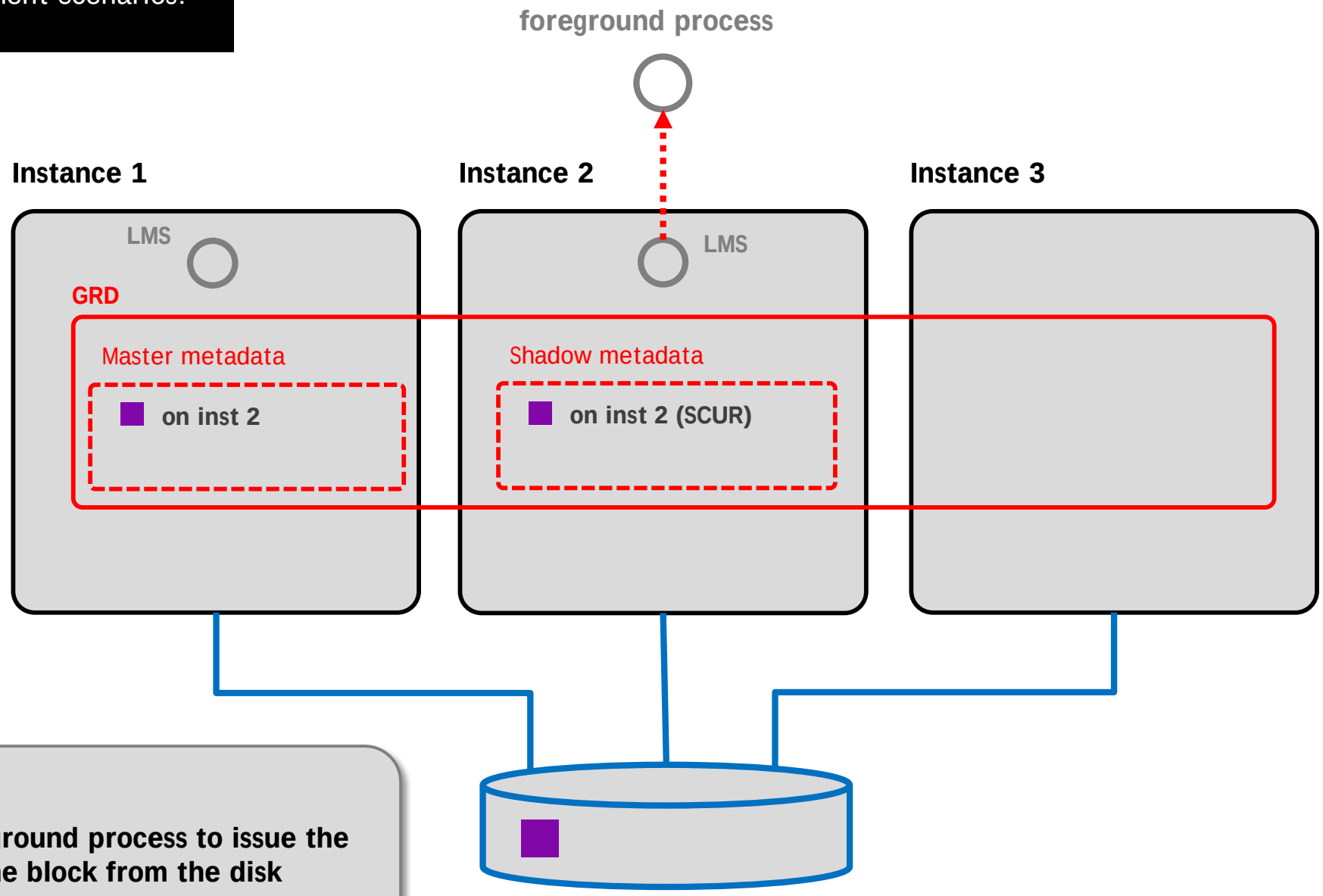
Global Cache Management Scenarios: **Read from Disk**



The buffer state is set to Shared Current (SCUR).

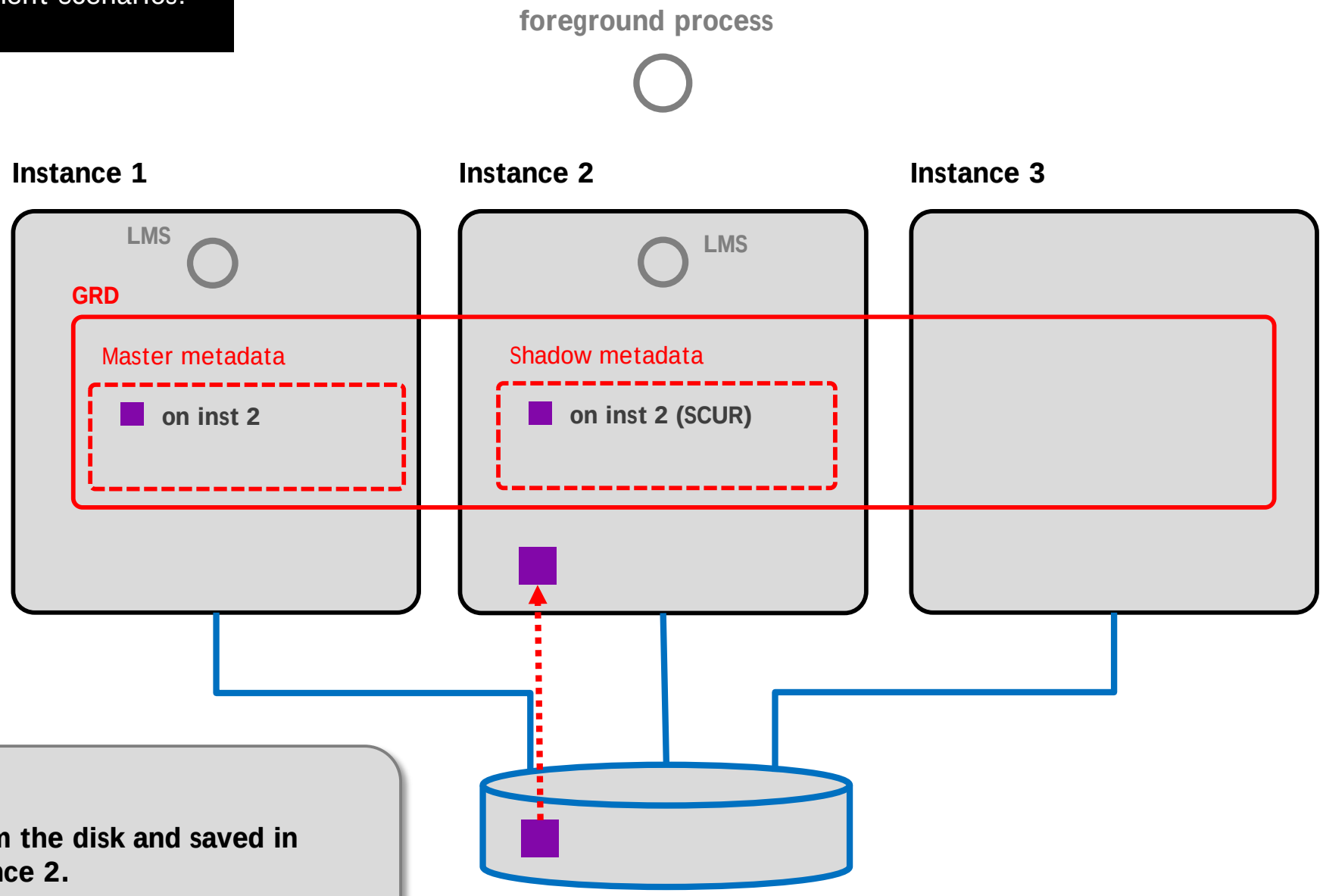
Shared Current: the block image matches the one on disk.

Global Cache Management Scenarios: **Read from Disk**



LMS notifies the foreground process to issue the I/O request to read the block from the disk

Global Cache Management Scenarios: **Read from Disk**

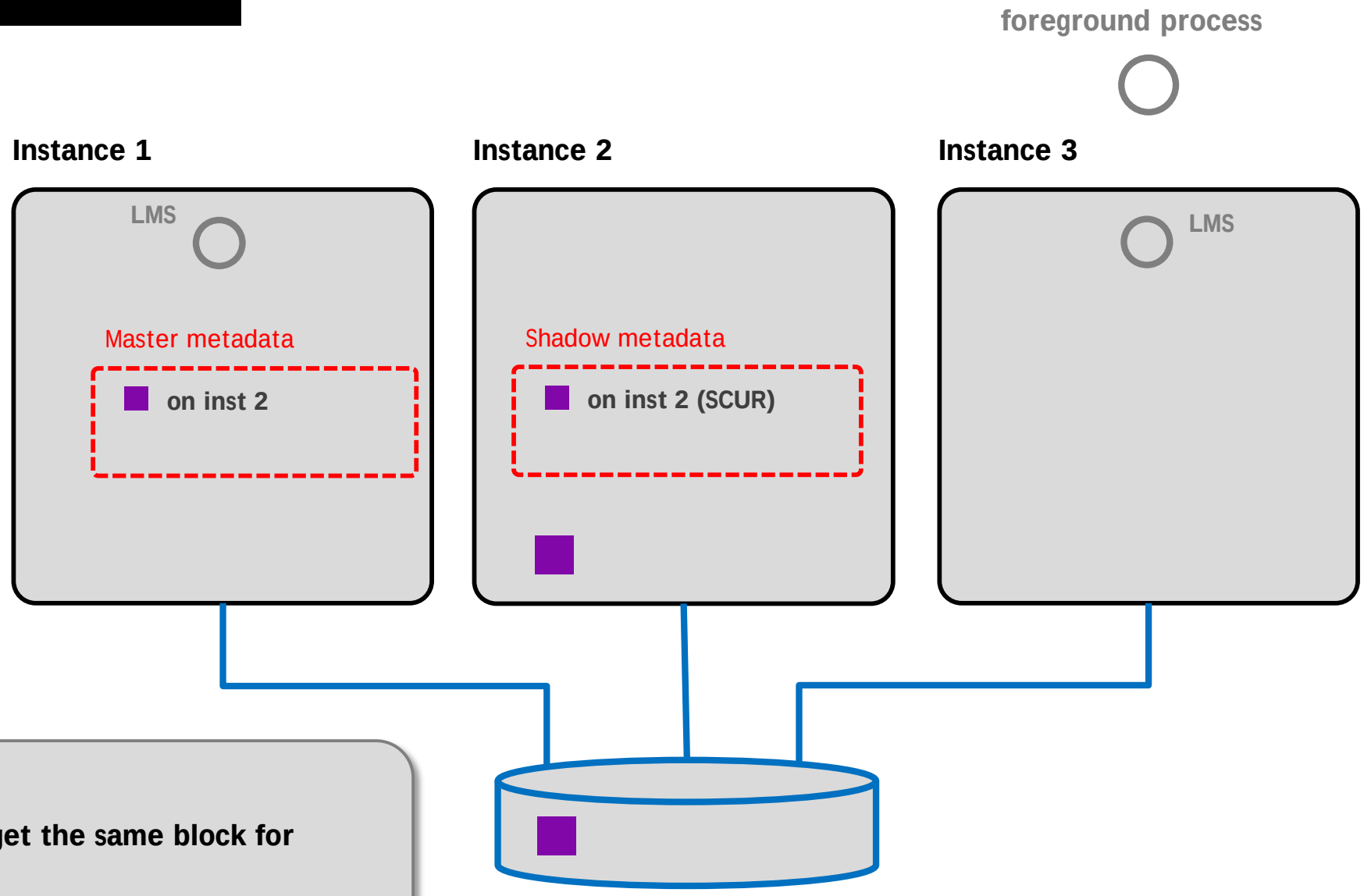


The block is read from the disk and saved in buffer cache of instance 2.

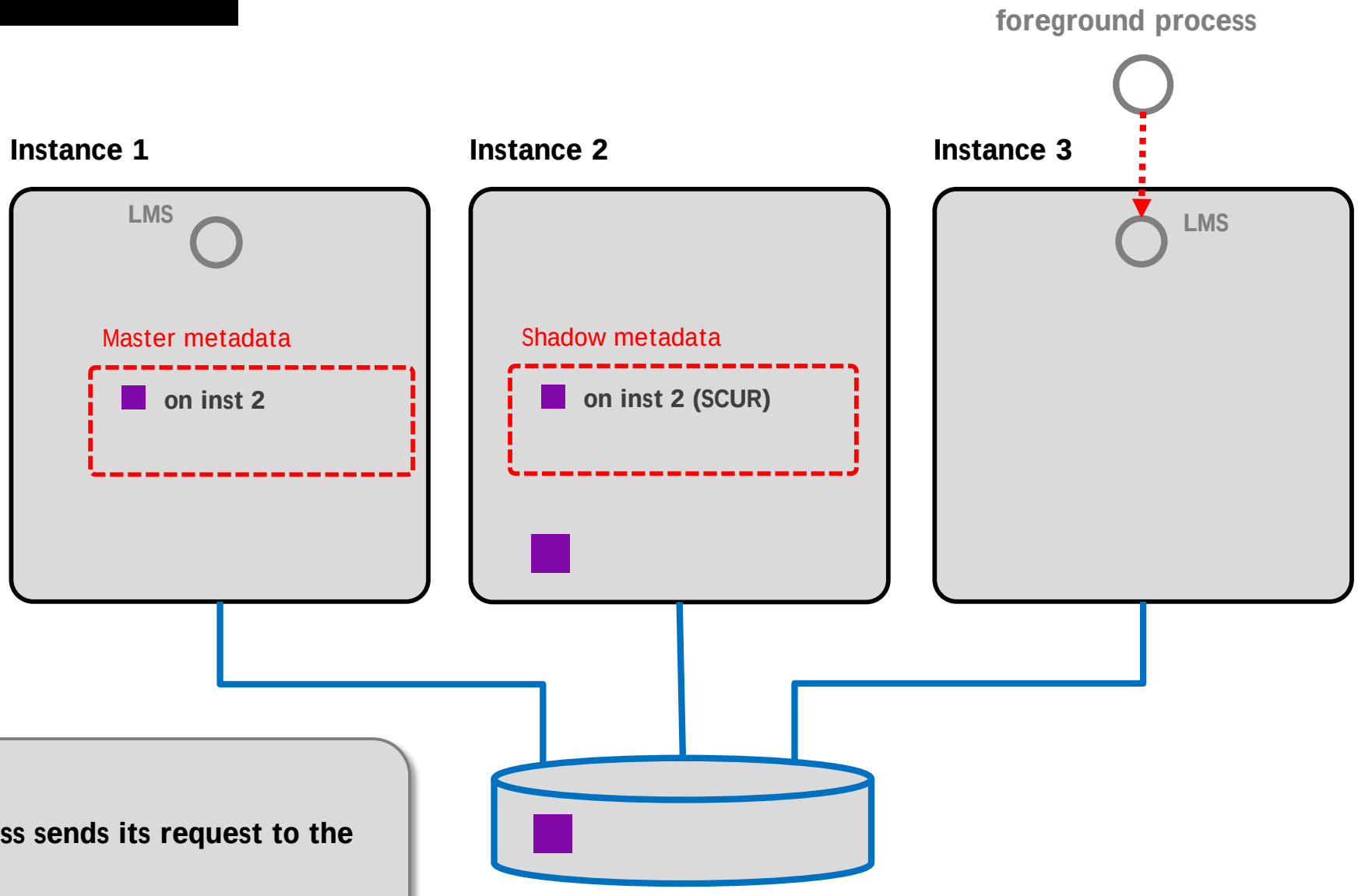
Global Cache Management Scenarios: Read from Disk Summary

- LMS of the requesting instance sends the request to the master instances
- As the block does not exist in any of the instances, it creates a master metadata in it and grant the requesting instance to read the block from the disk
- The requesting creates a shadow metadata and sets the buffer header status to “SCUR”
- The foreground process is notified to read the block from the disk

Global Cache Management Scenarios: **Read Write**

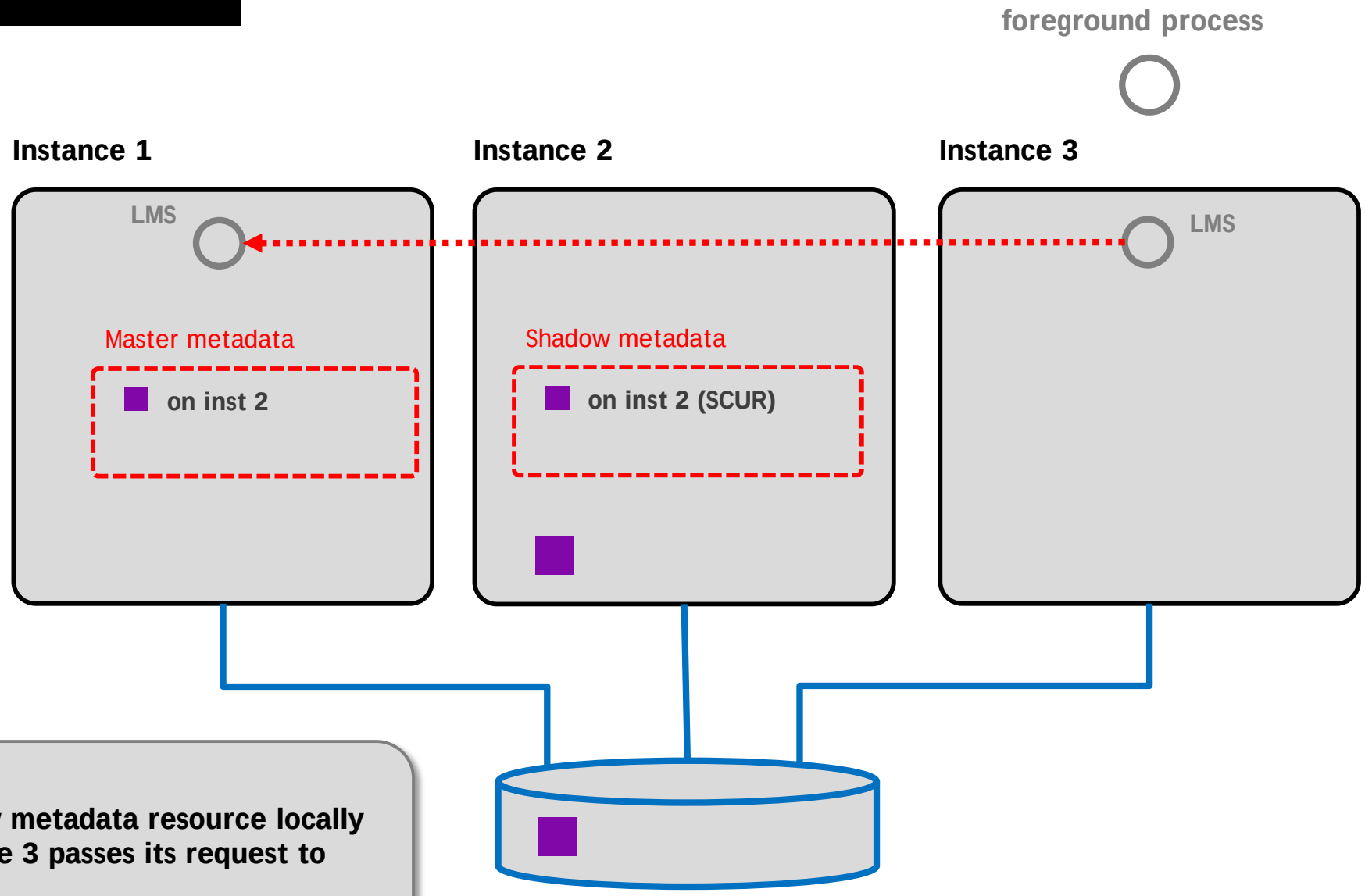


Global Cache Management Scenarios: **Read Write**



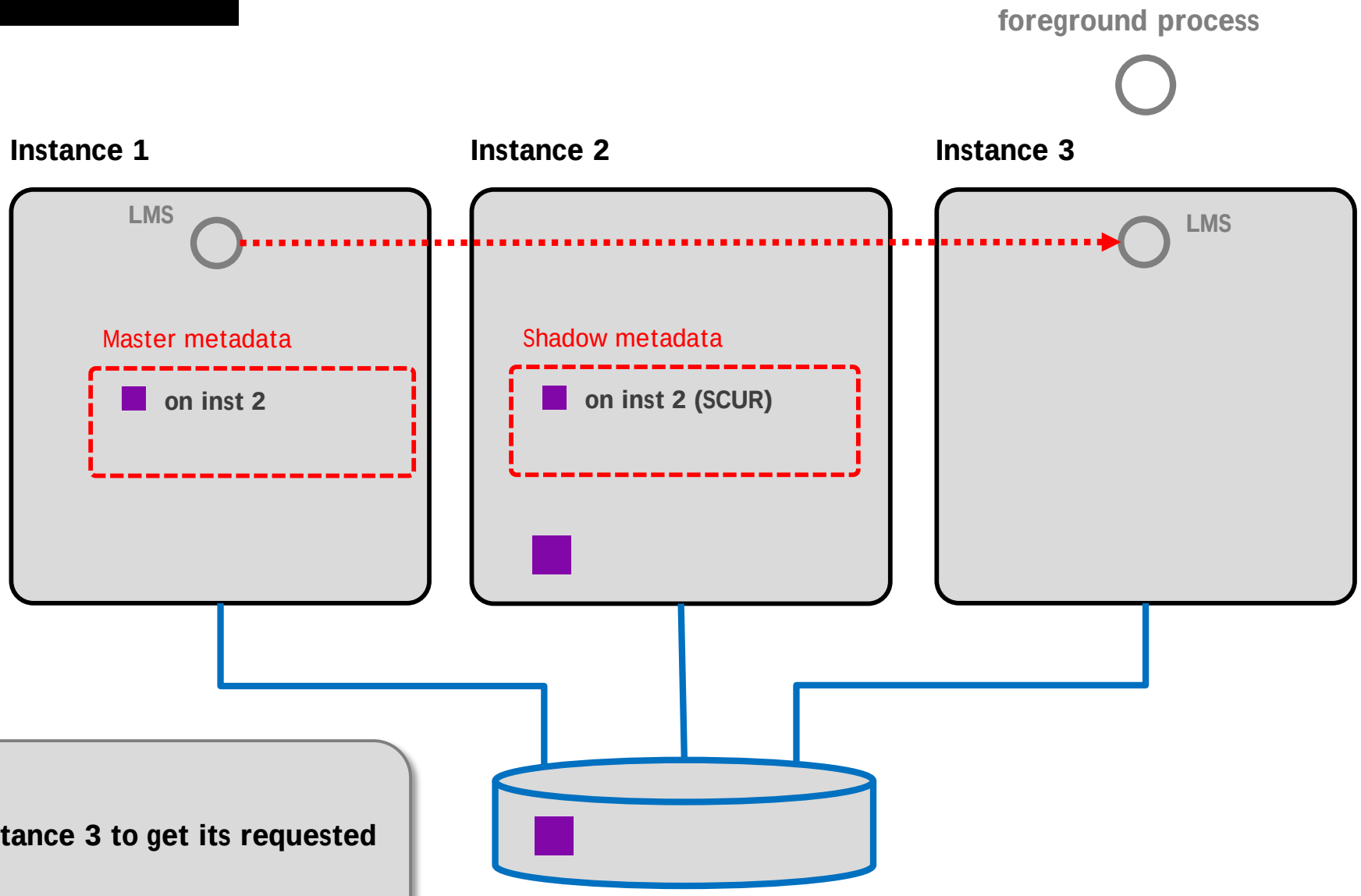
The foreground process sends its request to the local LMS process

Global Cache Management Scenarios: **Read Write**



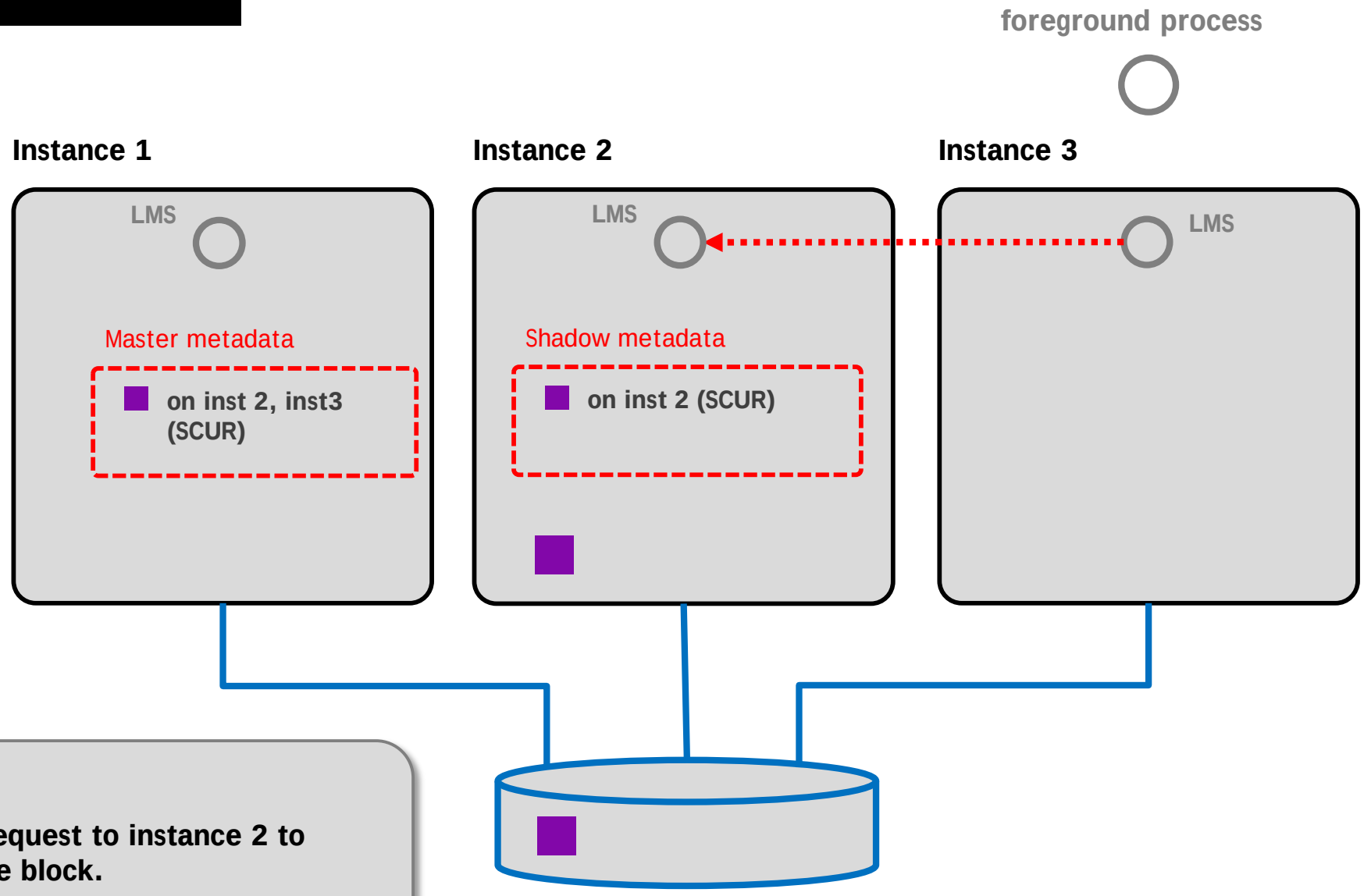
As there is no shadow metadata resource locally for the block, instance 3 passes its request to instance 1.

Global Cache Management Scenarios: **Read Write**



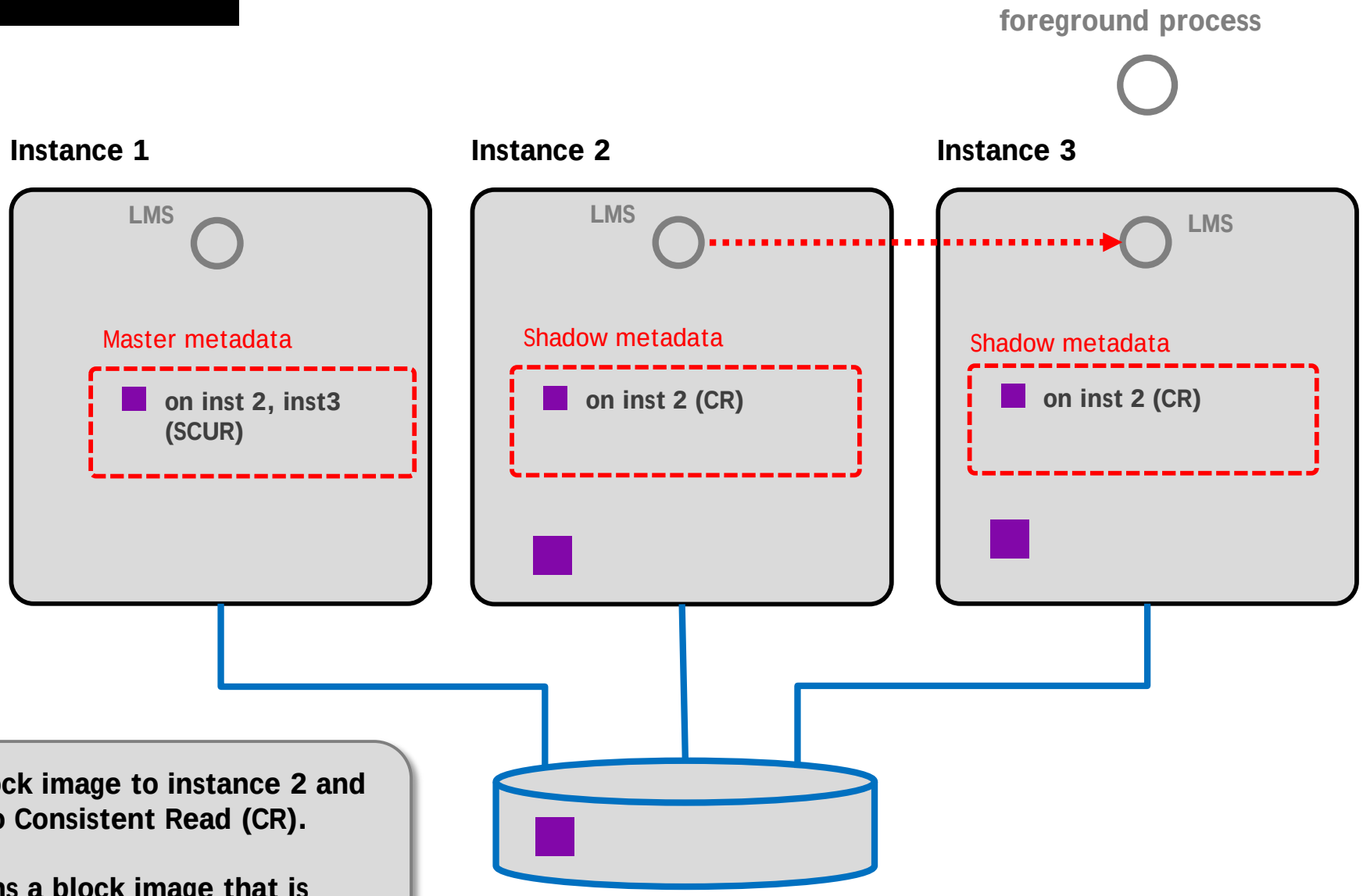
Instance 1 notifies instance 3 to get its requested block from instance 2

Global Cache Management Scenarios: **Read Write**



Instance 3 sends its request to instance 2 to obtain an image of the block.

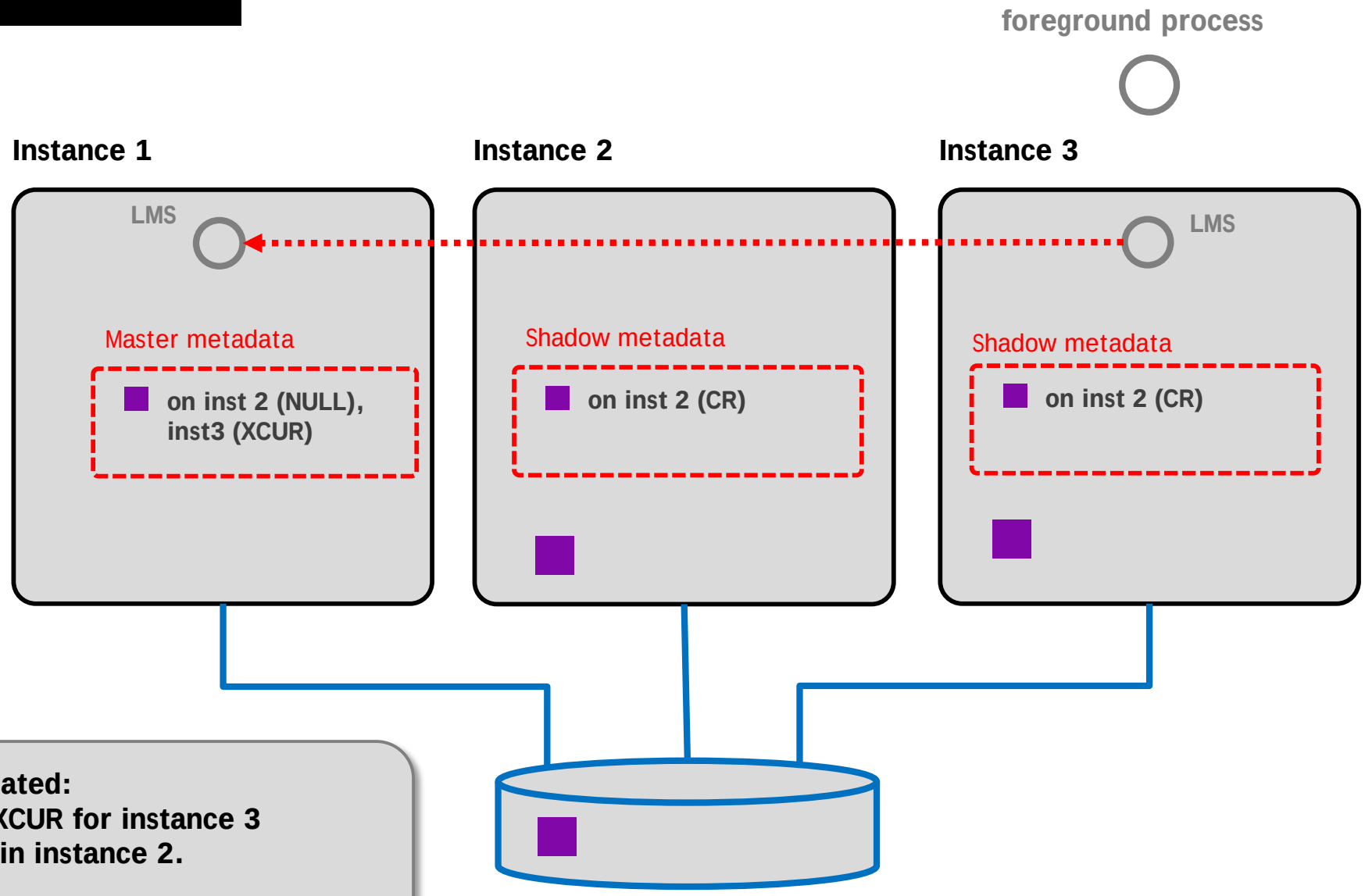
Global Cache Management Scenarios: **Read Write**



Instance 2 sends a block image to instance 2 and downgrade its state to Consistent Read (CR).

CR: the buffer contains a block image that is consistent with an earlier point in time.

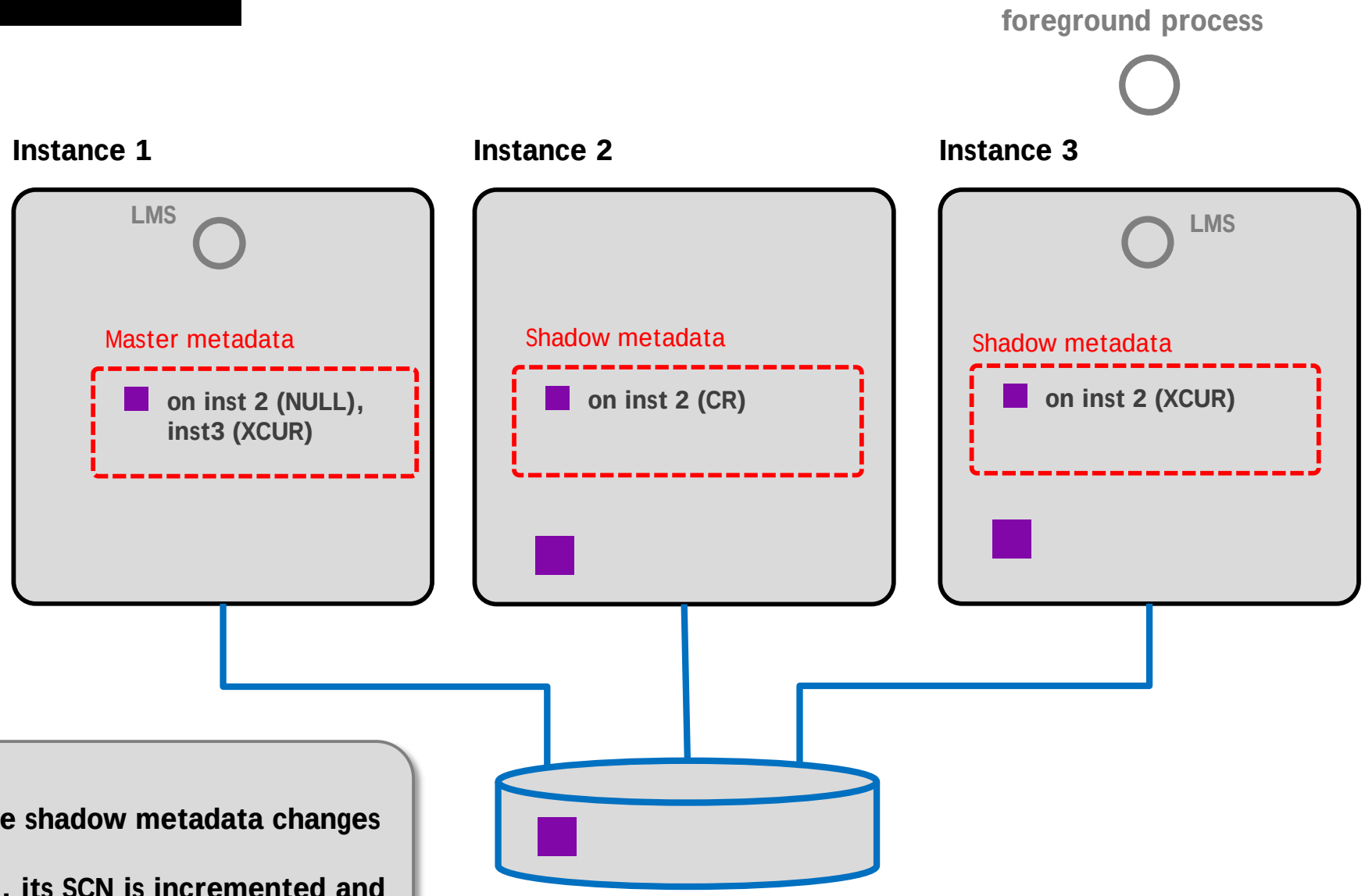
Global Cache Management Scenarios: **Read Write**



Master metadata updated:
Block state becomes XCUR for instance 3 and it becomes NULL in instance 2.

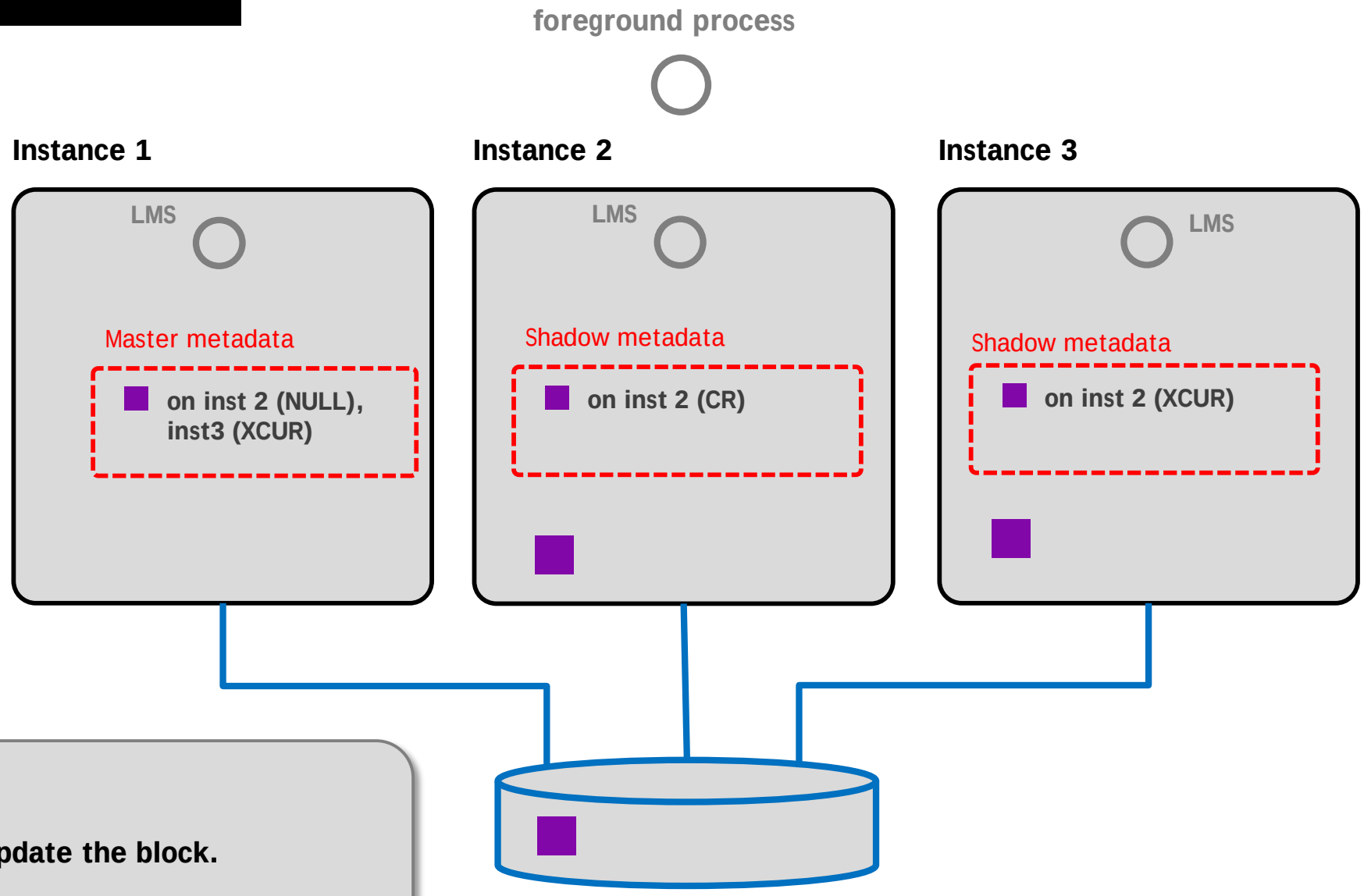
XCUR: the block image is to be updated

Global Cache Management Scenarios: **Read Write**



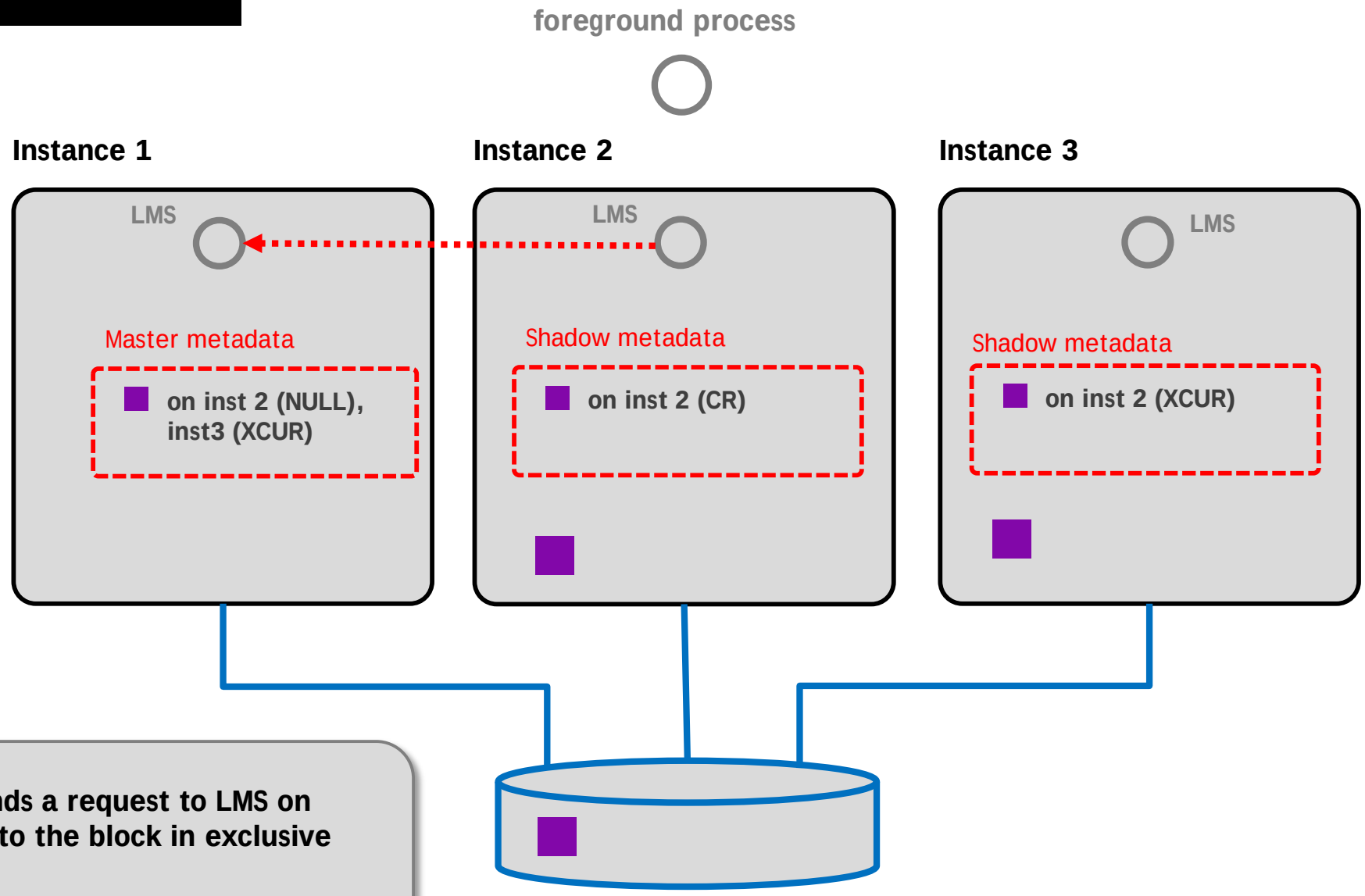
In instance 3:
- the block state in the shadow metadata changes to XCUR
- the block is updated, its SCN is incremented and it gets sent to the foreground process

Global Cache Management Scenarios: **Write Write**



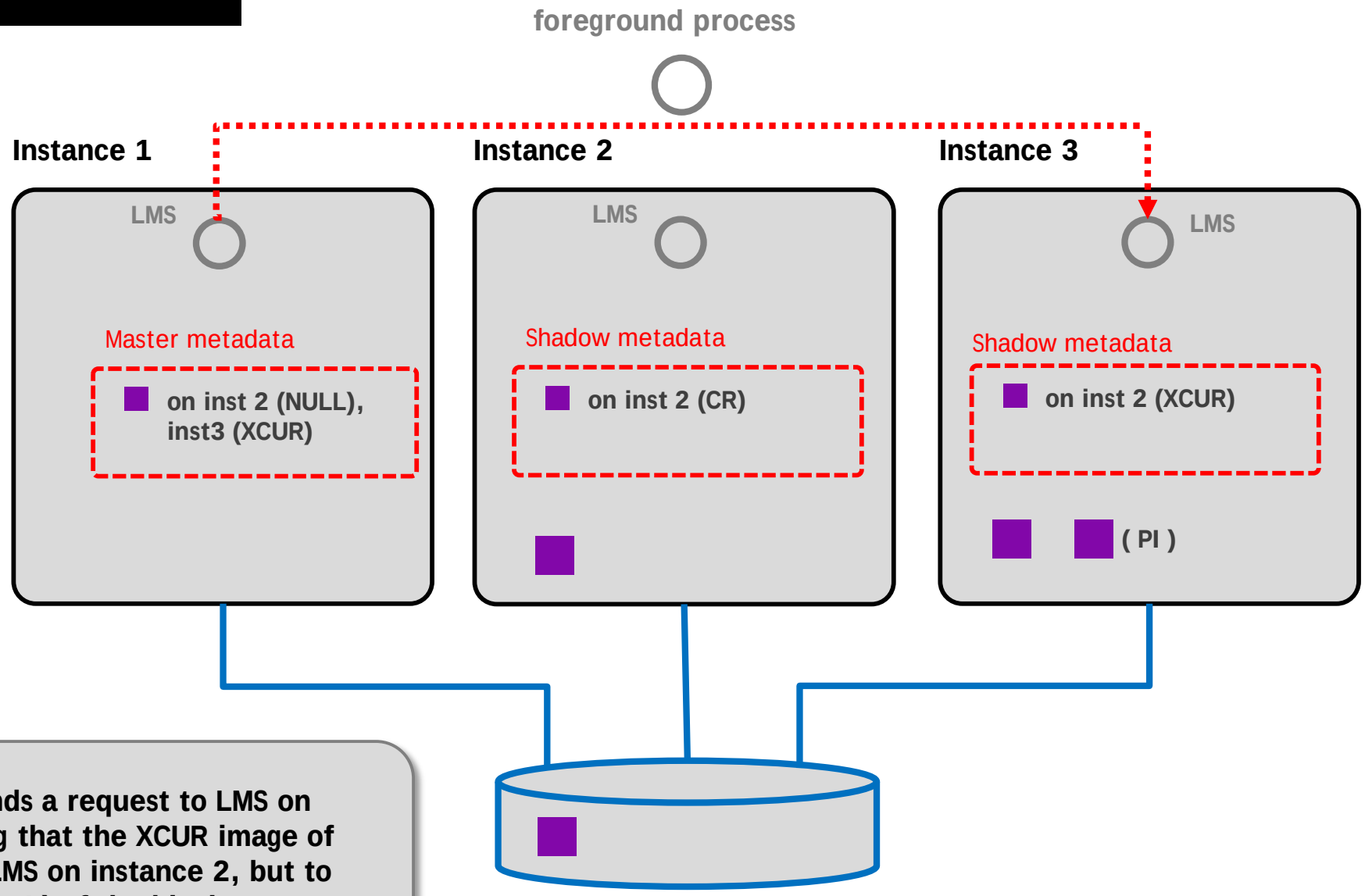
Instance 2 wants to update the block.

Global Cache Management Scenarios: **Write Write**



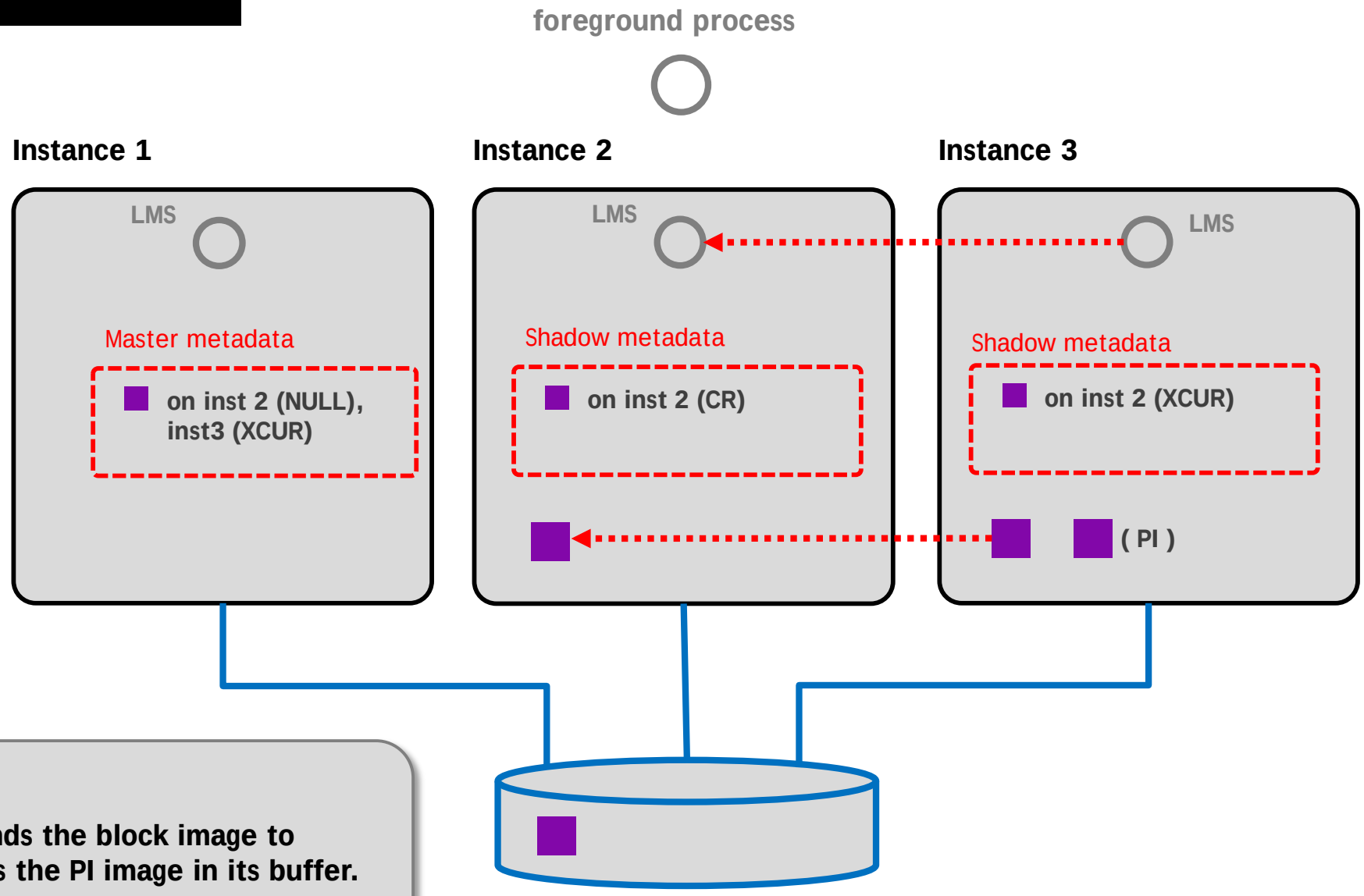
LMS on instance 2 sends a request to LMS on instance 1 for access to the block in exclusive mode.

Global Cache Management Scenarios: **Write Write**



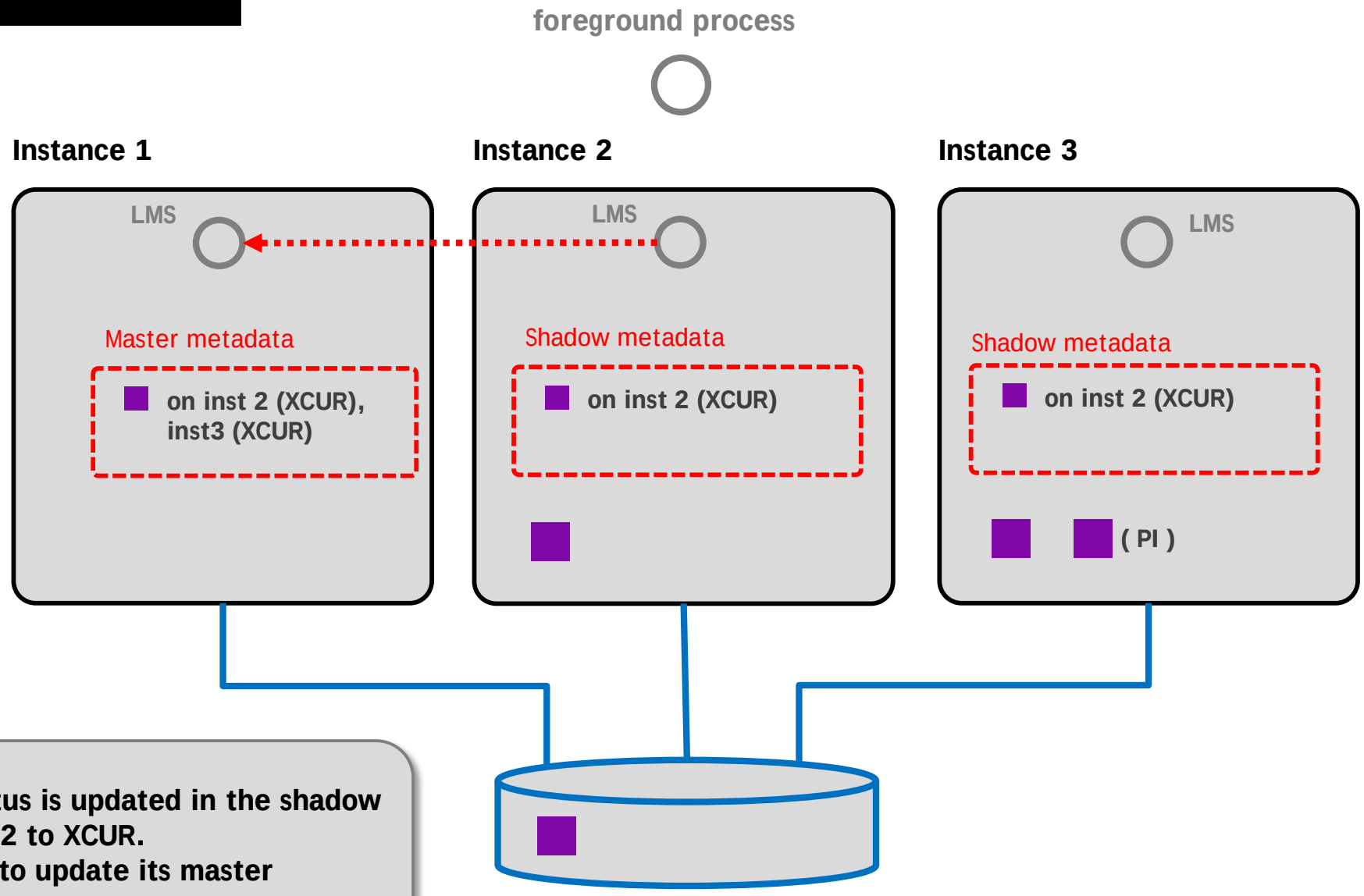
LMS on instance 1 sends a request to LMS on instance 3, requesting that the XCUR image of the block be sent to LMS on instance 2, but to keep the past image (PI) of the block.

Global Cache Management Scenarios: **Write Write**



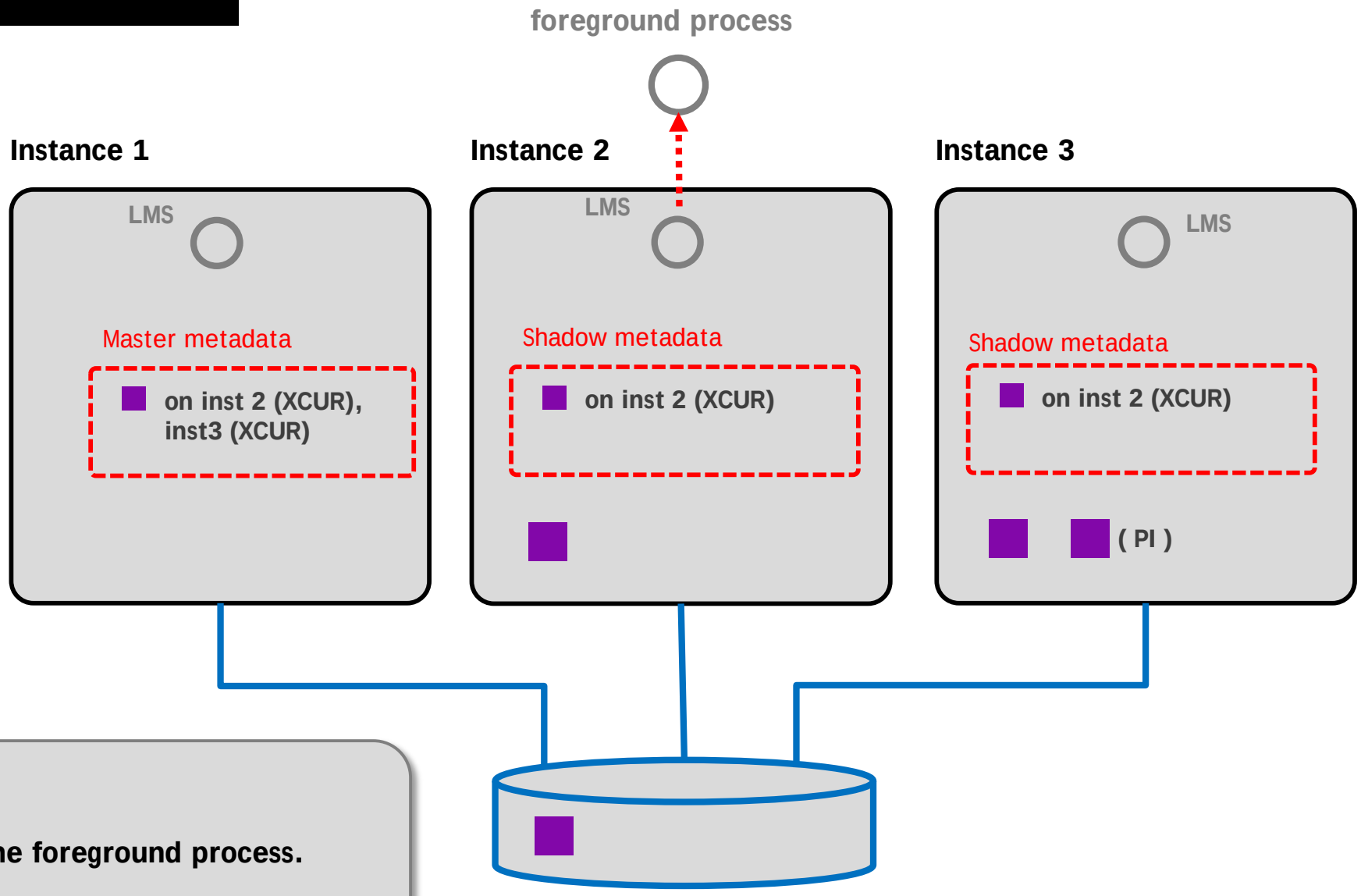
LMS on instance 3 sends the block image to instance 2 and retains the PI image in its buffer.

Global Cache Management Scenarios: **Write Write**



The block header status is updated in the shadow metadata of instance 2 to XCUR.
Instance 1 is notified to update its master metadata.

Global Cache Management Scenarios: **Write Write**



The block is sent to the foreground process.

Global Cache Buffer States

State	Description
Shared Current (**SC**)	The buffer block image matches the one on disk.
Exclusive Current (**XCUR**)	The block image is about to be updated, or has been updated.
Consistent Read (**CR**)	The block image is consistent with an earlier point in time.
Past Image (**PI**)	The block image is XCUR but then shipped to another instance using cache fusion.

Note: States can be retrieved from cache fusion in V$BH.STATUS

For Further Information

Summary

In this lecture, you should have learnt how to describe the following concepts:

- Global Concurrency Control
- Global Resource Directory (GRD)
- Mastering and shadowing instances
- Global Cache Management scenarios for single block access