Storage Grid using iSCSI

Felix Xavier
CloudByte Inc.

SNIA Legal Notice

- The material contained in this tutorial is copyrighted by the SNIA unless otherwise noted.
- Member companies and individual members may use this material in presentations and literature under the following conditions:
  - Any slide or slides used must be reproduced in their entirety without modification
  - The SNIA must be acknowledged as the source of any material used in the body of any document containing material from these presentations.
- This presentation is a project of the SNIA Education Committee.
- Neither the author nor the presenter is an attorney and nothing in this presentation is intended to be, or should be construed as legal advice or an opinion of counsel. If you need legal advice or a legal opinion please contact your attorney.
- The information presented herein represents the author's personal opinion and current understanding of the relevant issues involved. The author, the presenter, and the SNIA do not assume any responsibility or liability for damages arising out of any reliance on or use of this information.

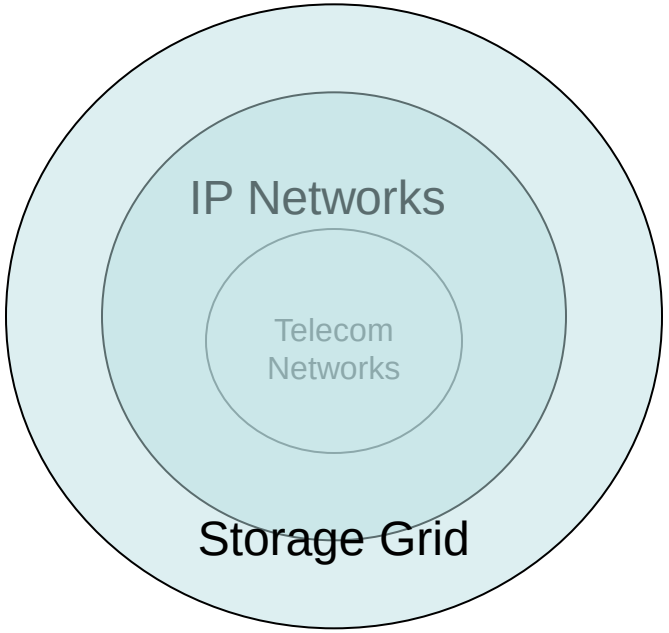
NO WARRANTIES, EXPRESS OR IMPLIED. USE AT YOUR OWN RISK.

Abstract

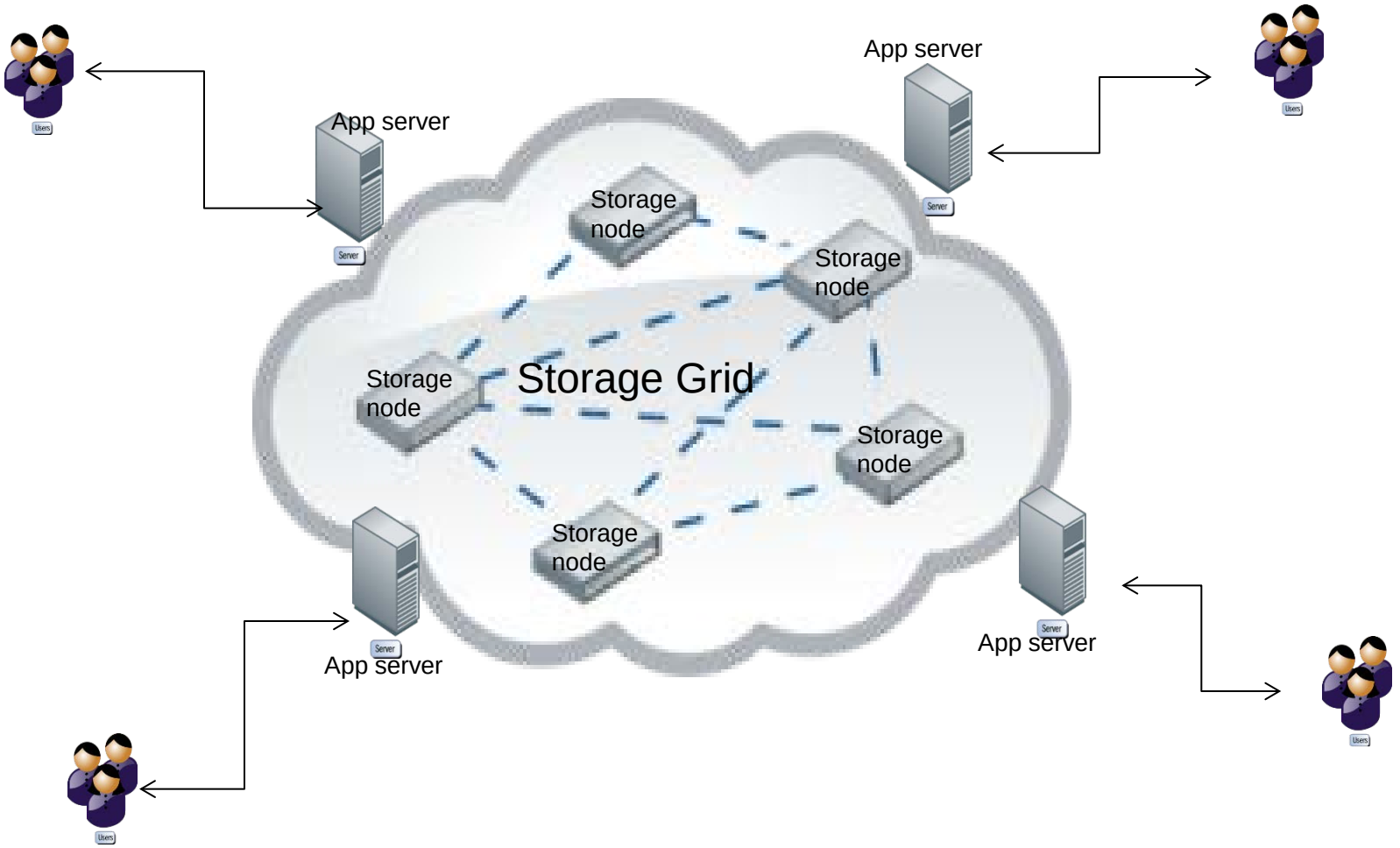
- Advent of cloud brought a new requirement on the storage: the storage nodes in the cloud have to communicate with each other and bring the hot data near the application across the data centres
- The communication must be standard-based
- This session proposes iSCSI protocol to achieve the inter-storage node communication.
- This is NOT intended to cover the underlying storage implementation to optimally support this communication

IT Evolution: New Storage Core

- Telecom Networks
  - The first phase of electronic communication
  - Telex and fax operated on this network
- IP Networks
  - When internet evolved, the IP networks became the core
  - www, FTP, and email ran on this network
- Storage Grid
  - Now cloud is emerging where storage forms the new core
  - Web apps already use this grid while enterprise apps are moving towards it

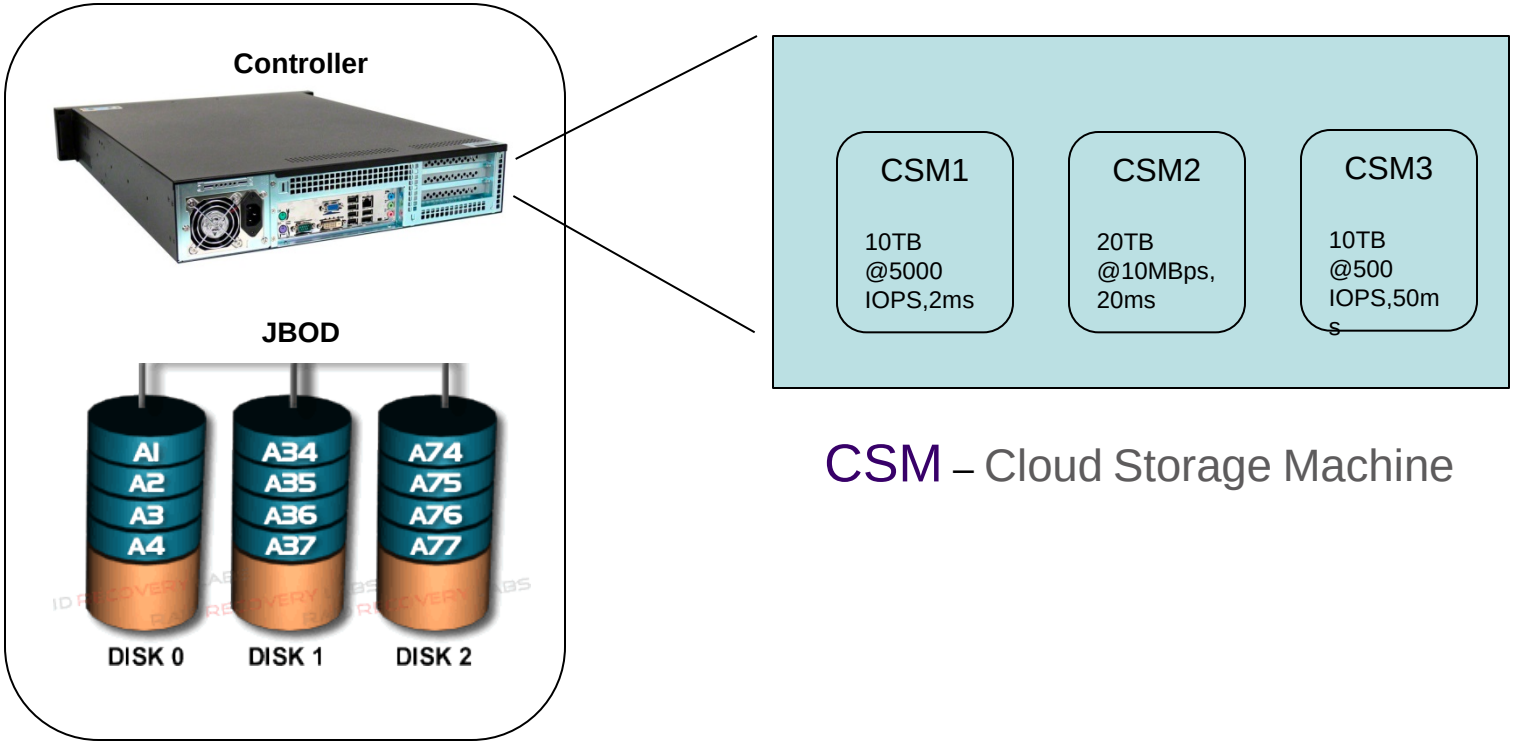


The New Core: A Broader View



The Intelligent Storage Node

Controller Software Architecture

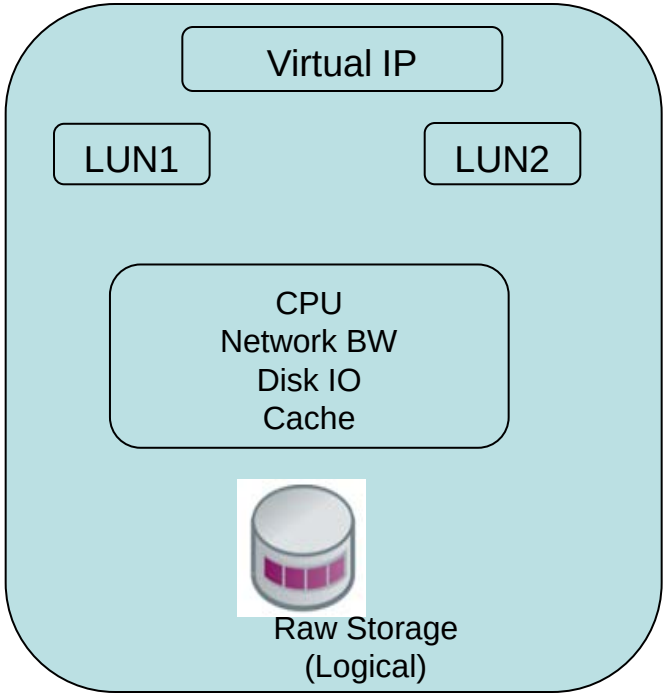


CSM – Cloud Storage Machine

Storage Node

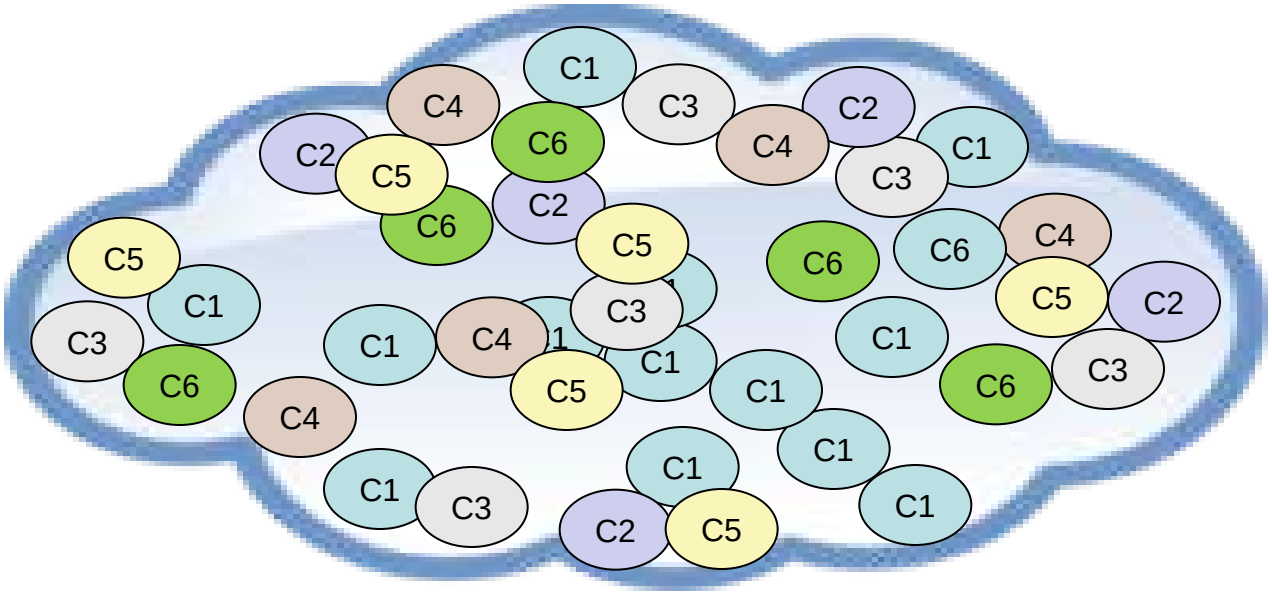
CSM Architecture

- CSM abstracts the hardware characteristics into the software
- Each CSM has dynamically allocated hardware resources in terms of
  - CPU
  - Network bandwidth
  - Disk I/O
  - Cache
- Each CSM can host one or more storage volumes
- CSMs freely move across the storage nodes



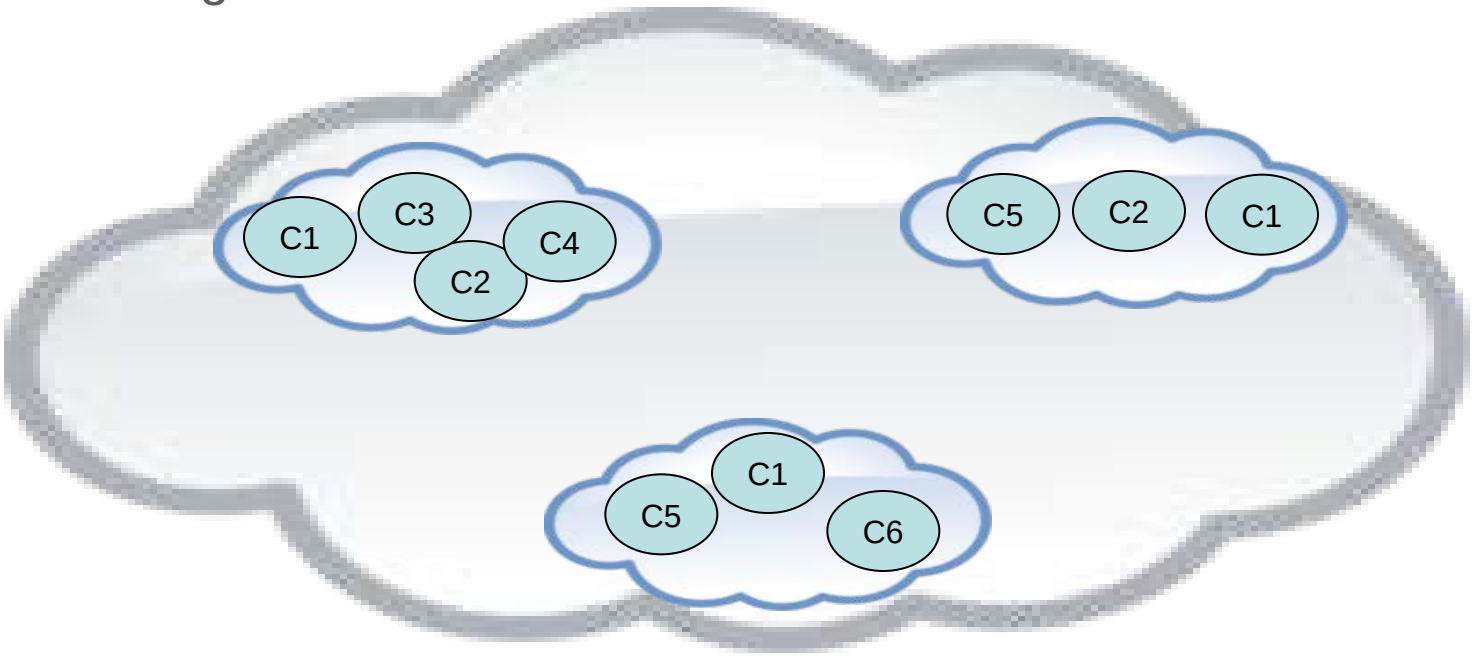
Global Namespace for CSMs

- CSM is the fundamental block in the storage grid
- One CSM can have multiple instances in the grid, depending on the access patterns from the app servers
- CSM name can resolve to the closest instance of the CSM in the storage grid



CSM – Migration

- CSM can completely migrate from one storage node to another without app disruption
  - Within the datacenter
  - Across data centers
  - Along with all characteristics

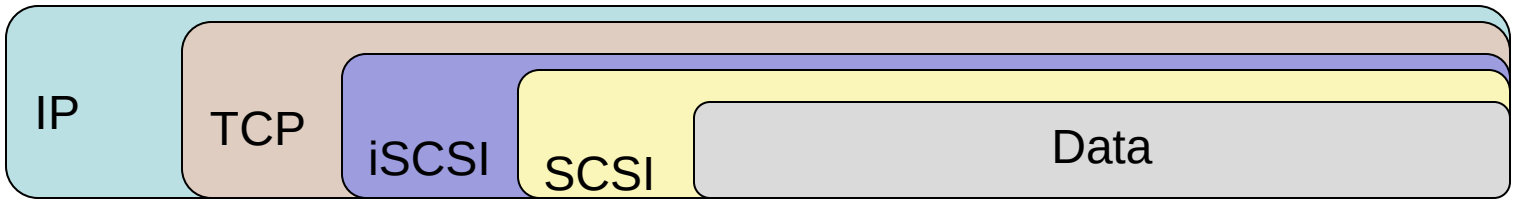


iSCSI Overview

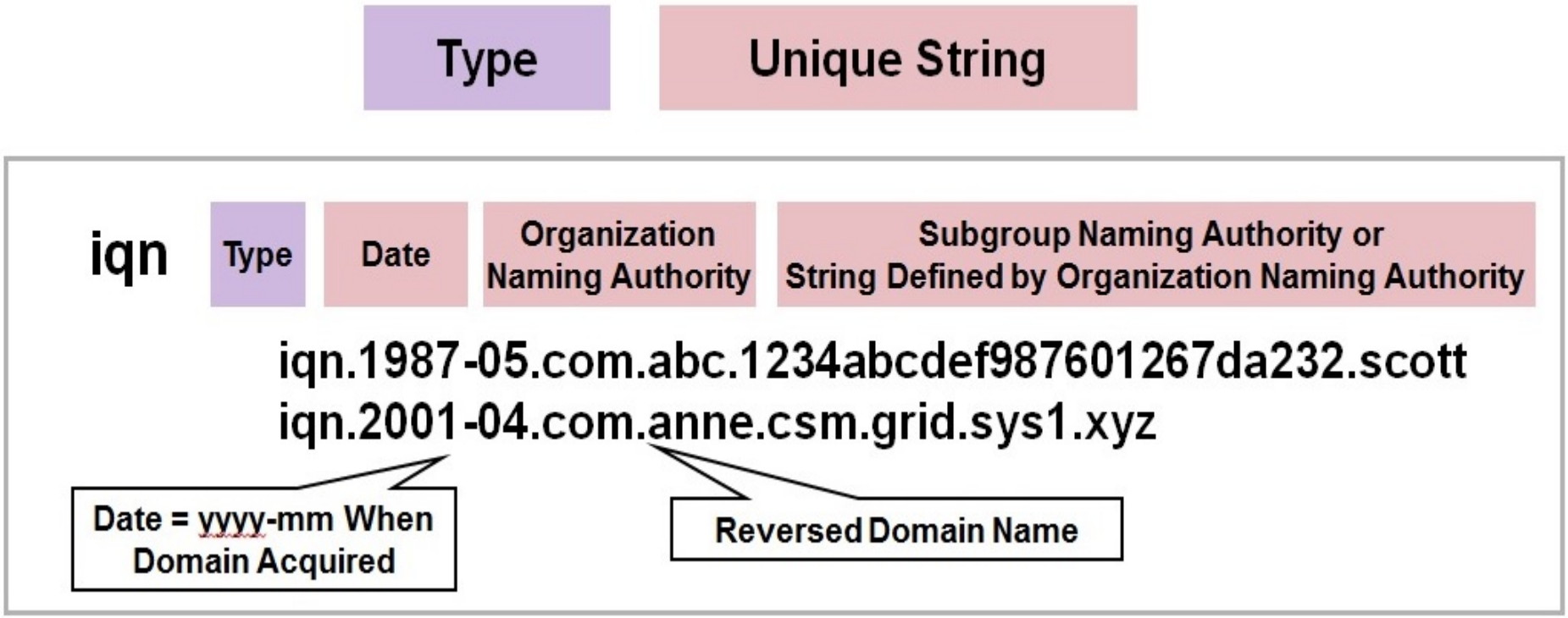
- A SCSI transport protocol that operates over TCP/IP
  - Encapsulates SCSI CDBs (operational commands, for example READ or WRITE) and data into TCP/IP byte streams
  - Allows IP hosts to access IP-based SCSI targets
- Standards status
  - RFC 3720 on iSCSI
  - Collection of RFCs describing iSCSI
    - RFC 3347—iSCSI Requirements
    - RFC 3721—iSCSI Naming and Discovery
    - RFC 3723—iSCSI Security
- Broad industry support
  - Initiator support from server vendors
  - Native iSCSI storage arrays

Storage Networking over iSCSI

- iSCSI provides solution to carry storage traffic within IP
- Uses TCP, a reliable transport for delivery
- Applicable to local data center and long-haul applications
- Typical format



iSCSI Name Structure



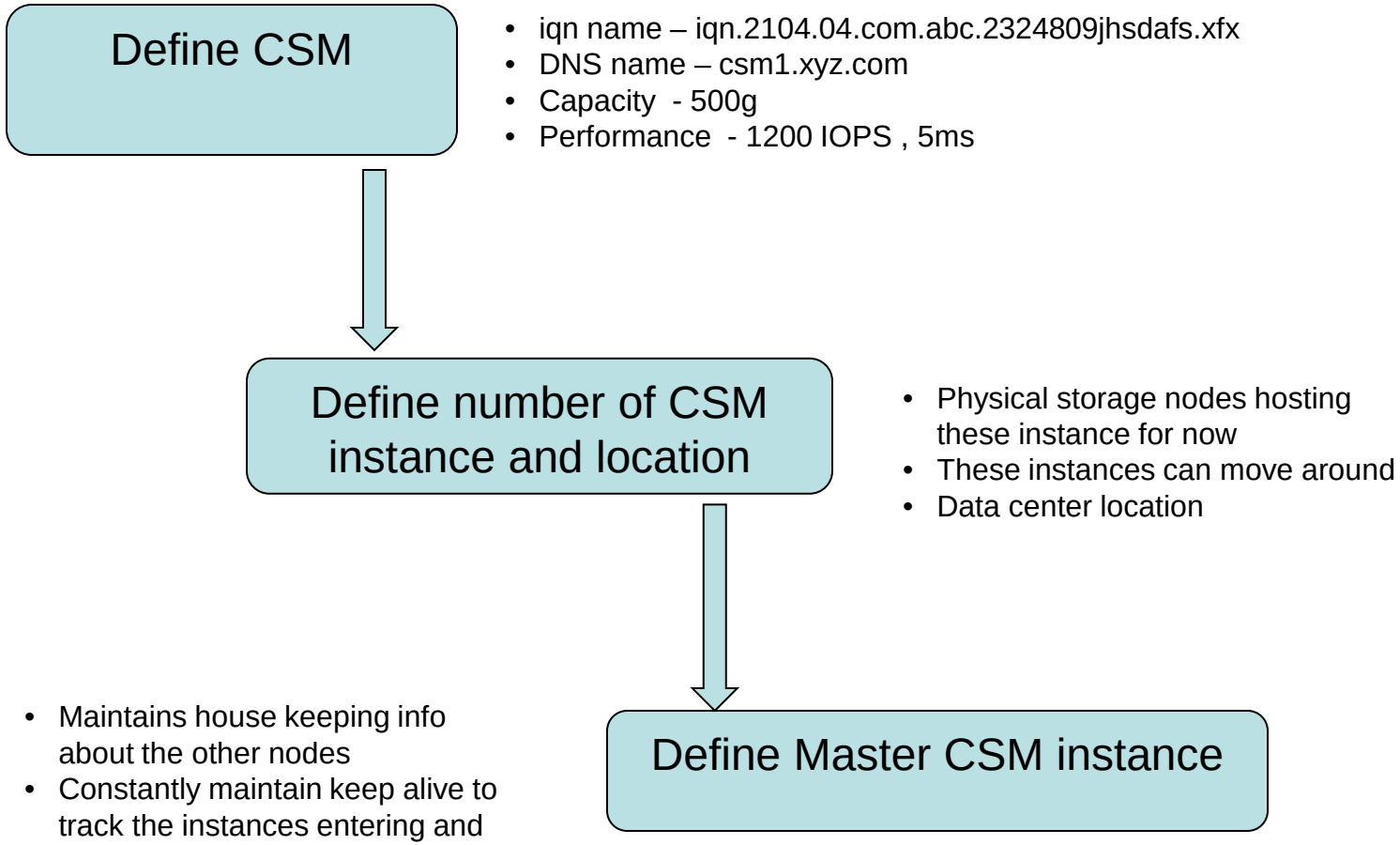
iSCSI Message Types

- Initiator to Target
  - NOP-out
  - SCSI Command
  - SCSI Task Management Command
  - Login Command
  - Text Command
  - SCSI Data-Out
  - Logout Command
- Target to Initiator
  - NOP-IN
  - SCSI Response
  - SCSI Task Management Response
  - Login Response
  - Text Response
  - SCSI Data-In
  - Logout Response
  - Ready to Transfer
  - Async Event

iSCSI Message Types - NEW

- CSM (Target) to same CSM (Target) instance
  - Read lock
    - Start block to end block
  - Write lock
    - Start block to end block
  - Cache validity check
  - Advertise new instance
  - Elect master
  - In-sync
  - Create instance/response
  - Init transfer
  - Update transfer

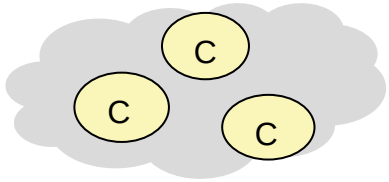
Management Workflow



Data Flow

Users

App servers



Grid

End user or middleware raises application request

- HTTP
- database query

Mount the storage in the app server

DNS Query to grid

DNS response

Mount request

Mount response

Closest CSM instance responds back

Pull the data from storage if it is not available in local cache

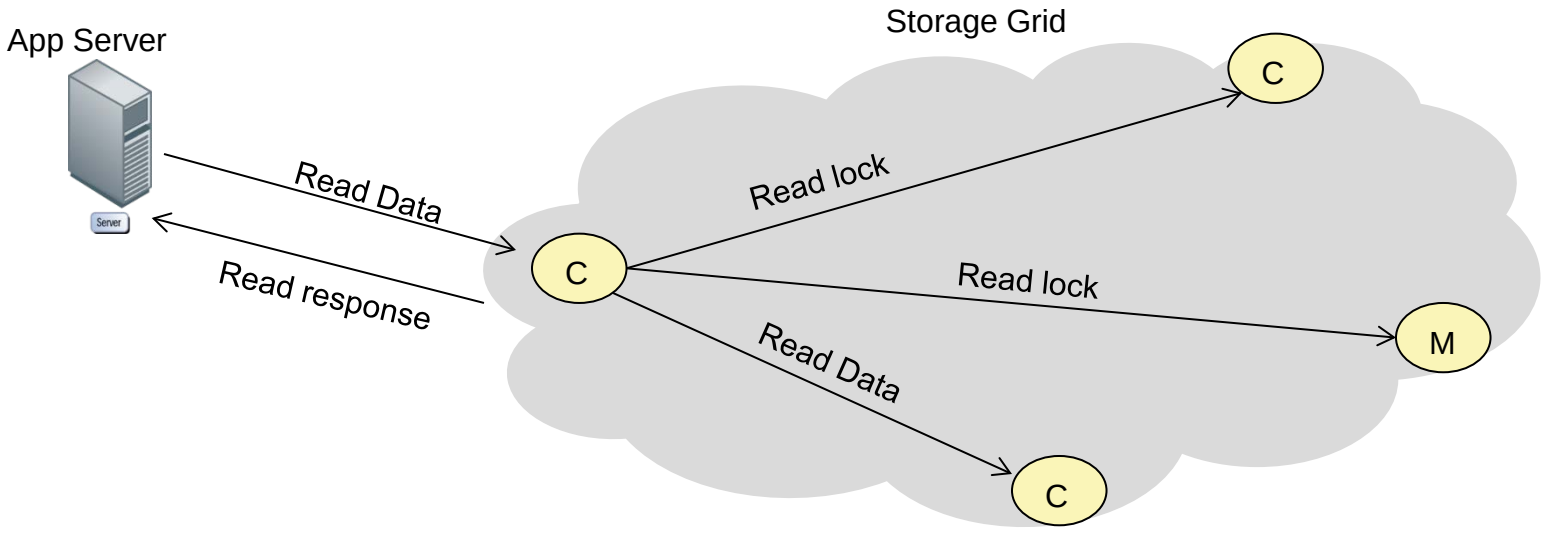
ask data using NFS/iSCSI/..

Respond with data

CSM co-ordinates with other instances if the requested data is not local

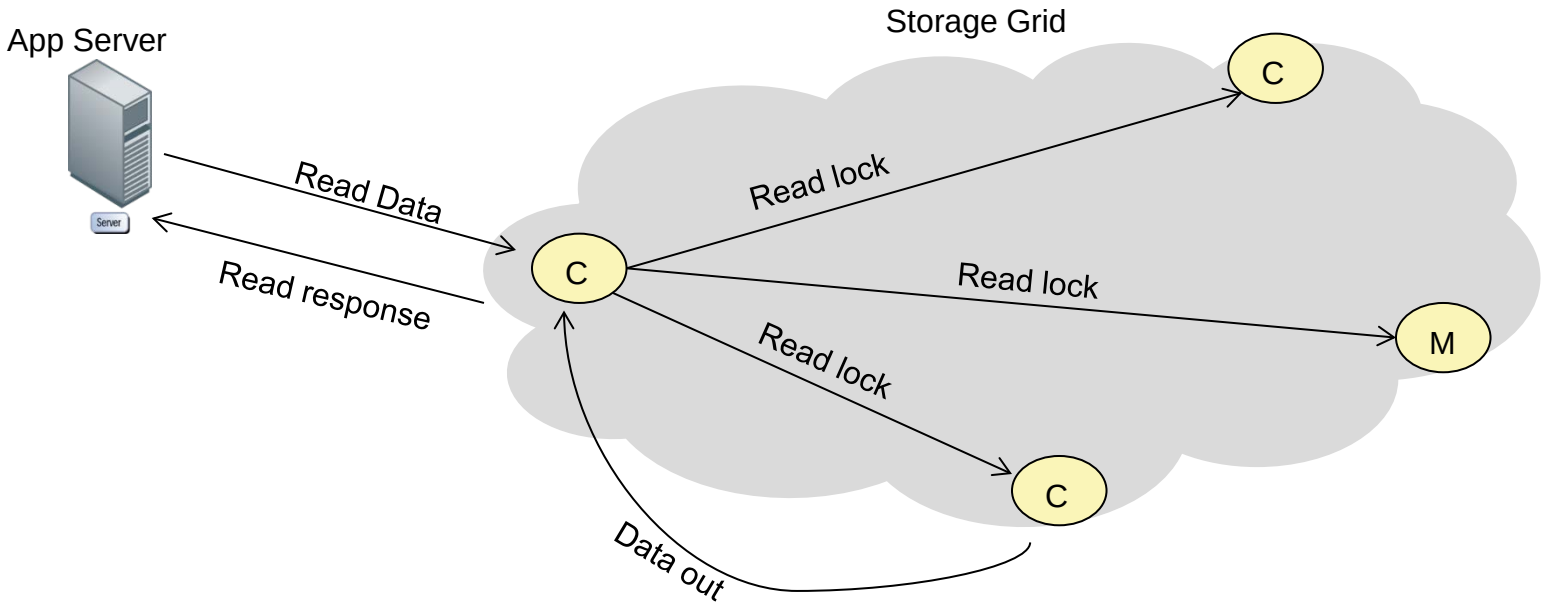
Response back to user

Read data at same instances



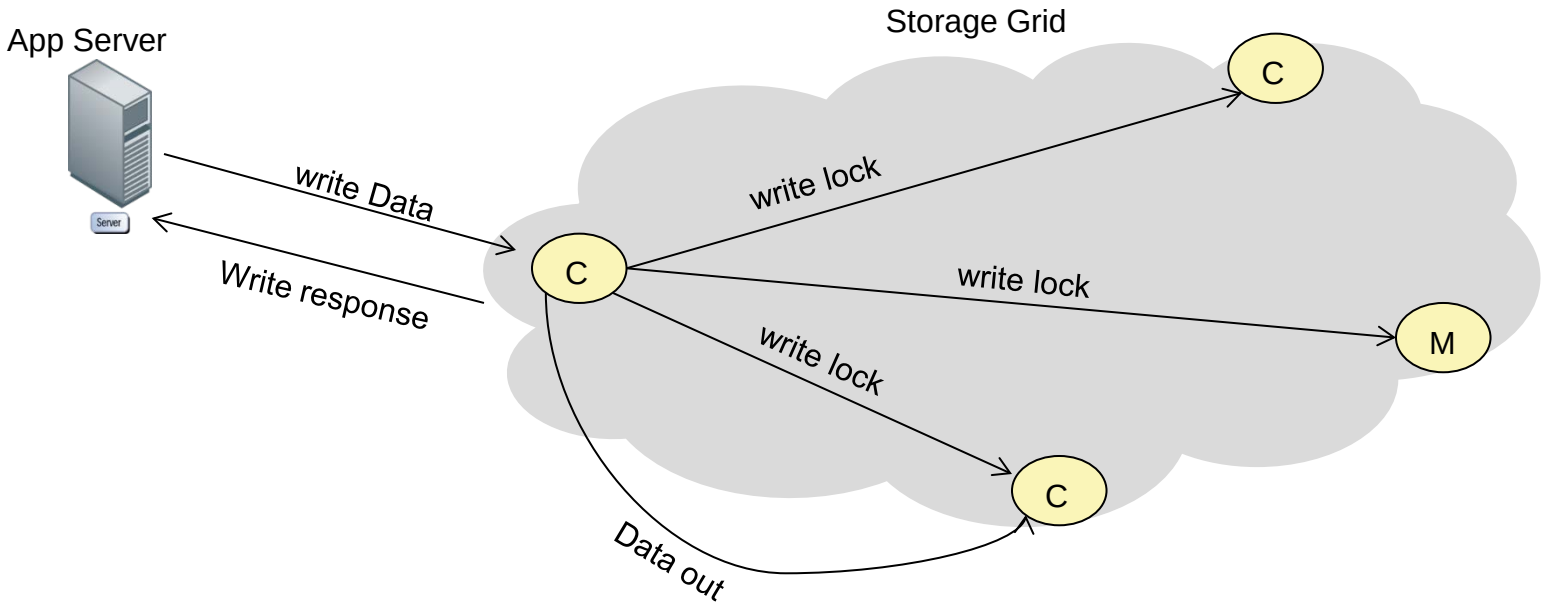
Step 1. App server issues the Data request to the connected CSM instance
Step 2 . CSM issues read lock to its peer instance for the group of blocks
Step 3. On success of read lock, it returns data to the application

Read data from other instances



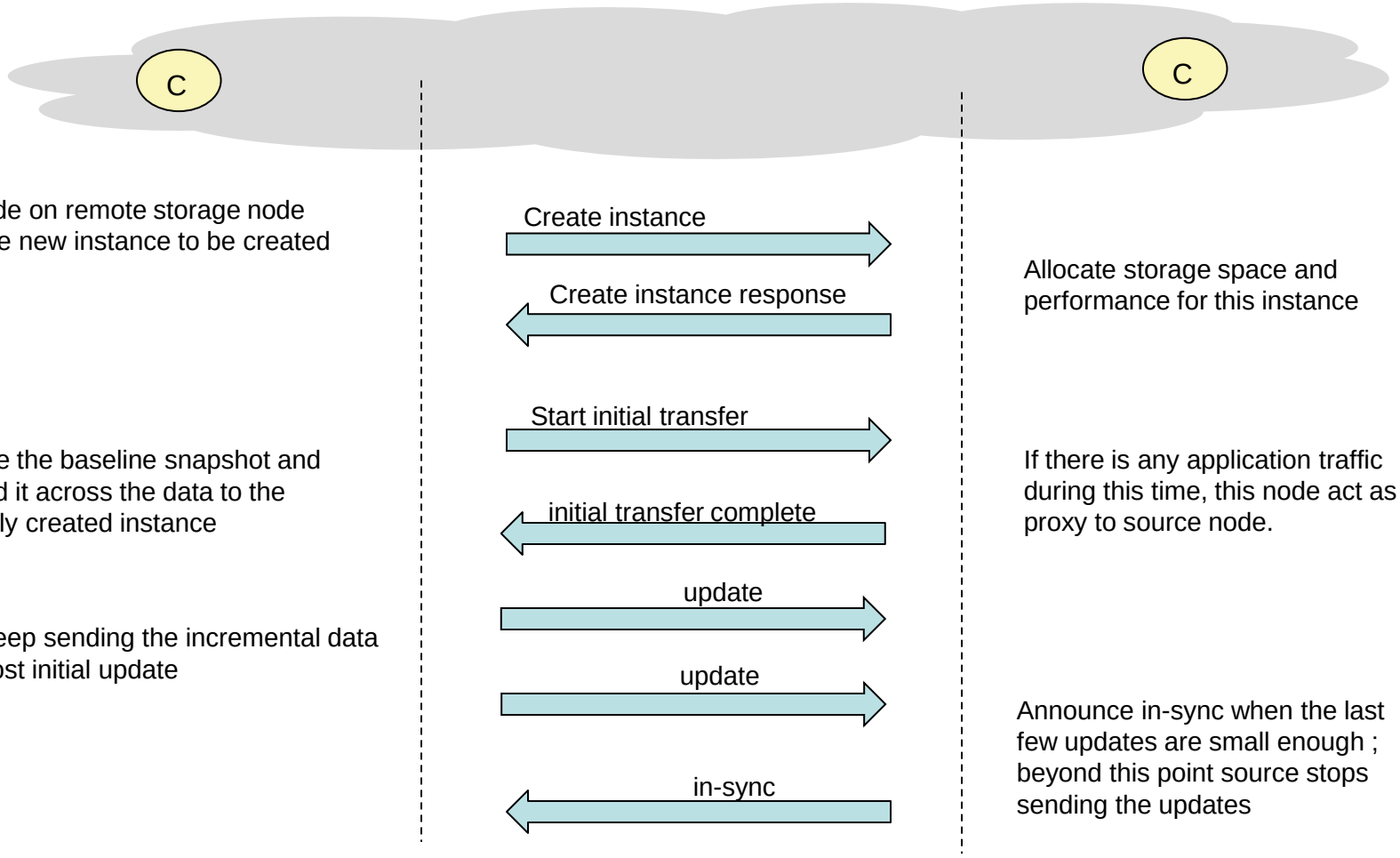
Step 1. App server issues the Data request to the connected CSM instance
Step 2.CSM issues read lock to its peer instance for the group of blocks
Step 3. When someone has latest data than this instance, it fetches the data from there
Step 4.It returns data to the application

Write data



Step 1. App server issues the write request to the connected CSM instance
Step 2.CSM issues write lock to its peer instance for the group of blocks
Step 3.Returns the acknowledgement to app server after writing locally.
Step 4.Writes back to one of the instance immediately and to others upon read request.

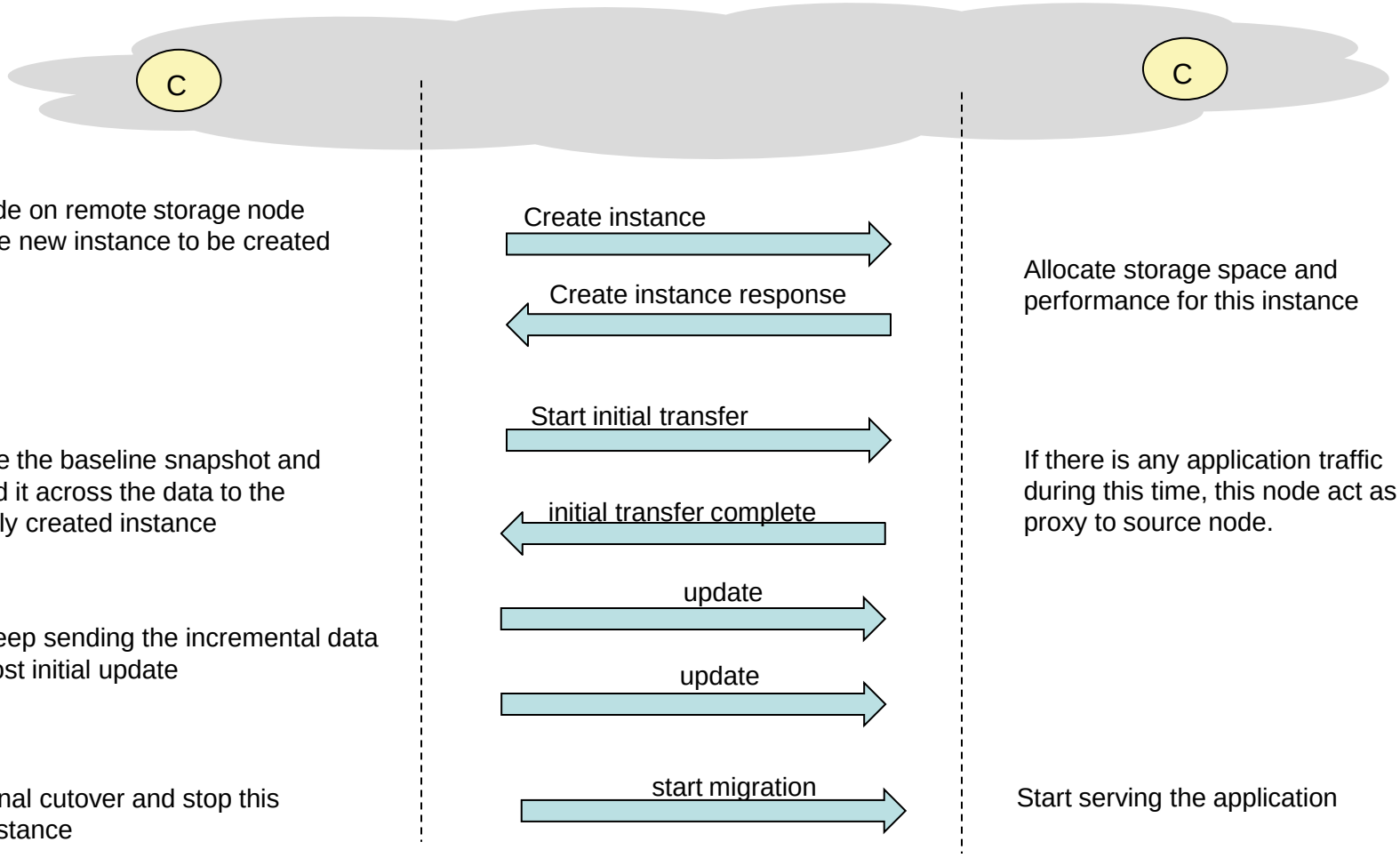
Create New instance



Failure handling

- Master CSM keep sending keepalives to all the CSM instance
  - maintain the state all the instance.
- If Master CSM fails, other instances elects the new master based on traffic density.
  - Highest traffic density instance within the surviving instance becomes master
- App server reconnects with the surviving instance whenever there is a instance failure
- CSM instance re-entering into the system should get into in-sync state first.

CSM Migration with one instance



Summary

- For Cloud to be real for enterprise apps, hardware characteristics of storage need to be abstracted to software
- Abstracted storage should be available across data center to get the same level of benefits of web apps today.
- iSCSI is the standard supported by most of the storage arrays, hence iSCSI can be used for inter storage node communication.

References

- https://www.ietf.org/rfc/rfc3720.txt
- http://cloudcomputing.sys-con.com/node/2831125

Attribution & Feedback

The SNIA Education Committee thanks the following individuals for their contributions to this Tutorial.

Authorship History

Name/Date of Original Author here:
Felix Xavier, March 2014

Updates:

- **Name/Date**
- **Name/Date**
- **Name/Date**

Additional Contributors

Umasankar Mukkara
Nadeem Kattangere

Please send any questions or comments regarding this SNIA Tutorial to ***tracktutorials@snia.org***