Open Unified Data Protection and Business Continuity Framework

Presenter:

Dr. Anupam Bhide

Calsoft, Inc.

Email: anupam@calsoftinc.com | Phone: +91 (20) 4079 2900

Authors:

Anupam Bhide (Calsoft)

Parag Kulkarni (Calsoft)

Sunu Engineer (Calsoft)

Abstract

A unified model for Data Protection and Business Continuity in complex enterprise systems

Today's data center administrators are faced with challenges of managing and protecting complex enterprise systems comprising of physical and virtual components composed of heterogeneous hardware and software with complex interconnects. A typical enterprise system consists of

- Applications – SAP, Exchange, Share Point more
- Middleware – Databases, Web Servers,
- Operating Systems – Windows, Linux, UNIX & Hypervisors
- Servers – Servers (Physical), Blades, Servers (Virtualized)
- Network – Switches Physical & Virtual, Routers, Firewalls
- Storage – SAN, NAS, DAS, Cloud

All components in the enterprise system use persistent storage in the form of DAS, NAS or SAN.

The unified model presented and implemented by Calsoft explores an open independent framework for data protection using storage level snapshots. The framework uses SMI-S to interact with various enterprise system components and ensures a consistent state to perform data protection and disaster recovery.

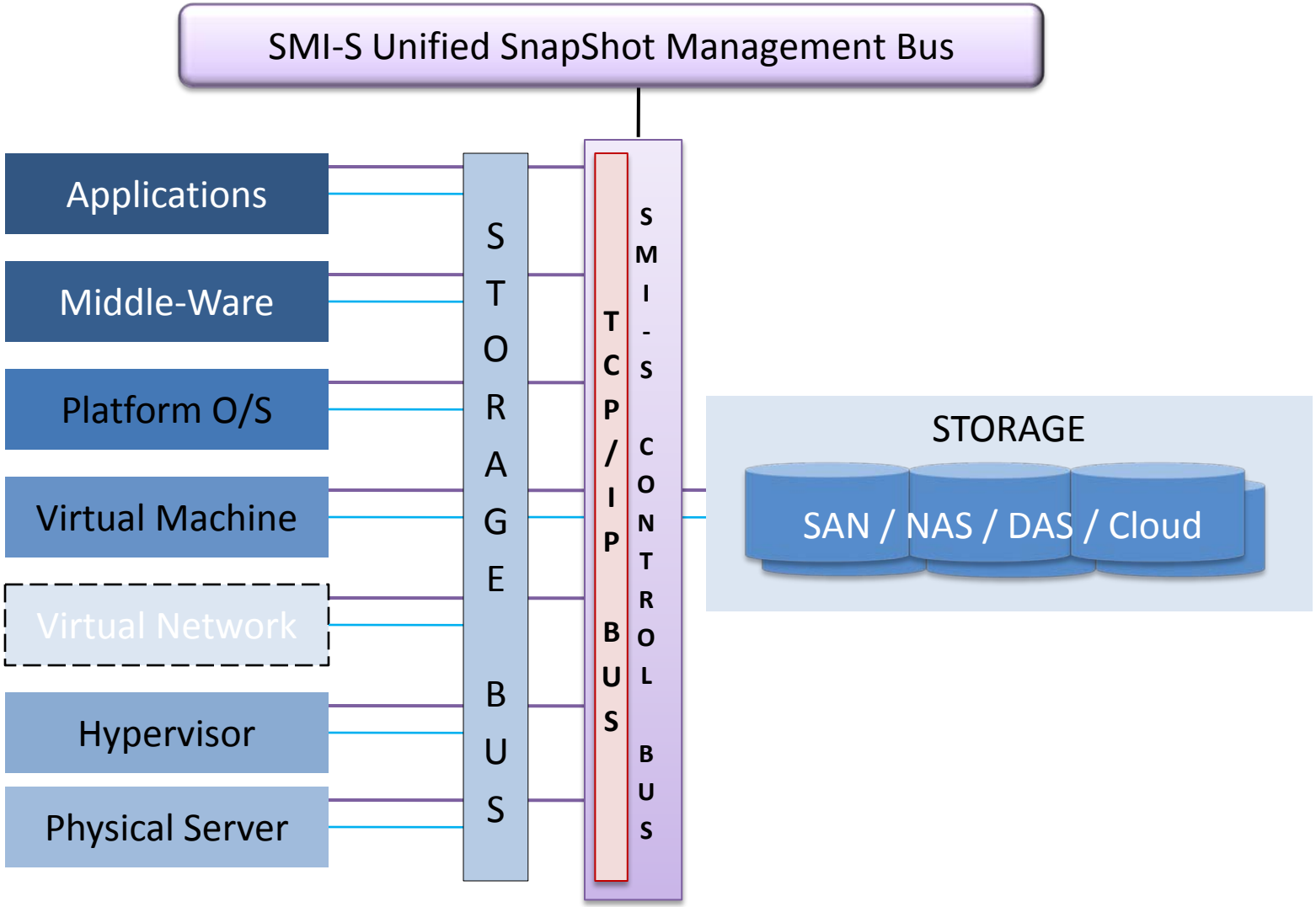
The Problem

- Data protection mechanisms such as backups, archiving, snapshots require a consistent and stable view/snapshot of the data
  - Without consistent view, recovery is generally not possible
- Application stacks consist of multiple entities such as applications themselves, middleware, hypervisors, operating Systems/file systems that are all modifying data on persistent storage
- To recover an application successfully requires that every entity recover its data successfully
  - A SAP application will require recovery of its own data in the file system, data in Oracle database, file system meta-data, hypervisor meta-data if virtualized
- How does one co-ordinate between these multiple entities that are writing concurrently to storage and ensure consistency of data during backups?

New challenges

- Application stacks are becoming more complex – consist of
  - Applications,
  - Middleware,
  - Databases,
  - Virtual Platforms (Hypervisors),
  - Operating Systems,
  - Storage
- Managing Dependencies and Integration between all above entities is required to ensure consistent data protection
- Applications run on different servers and especially virtual machines: hence applications are becoming more distributed
- Virtualization is adding yet another layer
  - This layer is even more complex than the others
- Storage is becoming more heterogeneous and more distributed
- Cloud computing and cloud storage adds to the complexity
- SNIA – SMI-S and a move to build an open standards for providing data protection at all levels.

Unified Snapshot Management Bus



Microsoft VSS – Platform specific solution

- Helps applications running on Windows platforms achieve application consistent backups
  - Platform (Windows) specific solution
- VSS offers to integrate the contributors of data protection process
  - Business application
  - Backup applications
  - Storage
- Provides interfaces to integrate the components:
  - VSS requestor – for the backup applications integration
  - VSS writer – for Business application (Exchange and SQL Server) integration
    - Application vendors must implement these interfaces to participate in VSS backups
    - Called by the VSS service to quiesce the application and get backup metadata
  - VSS provider - supplied each storage box manufacturer

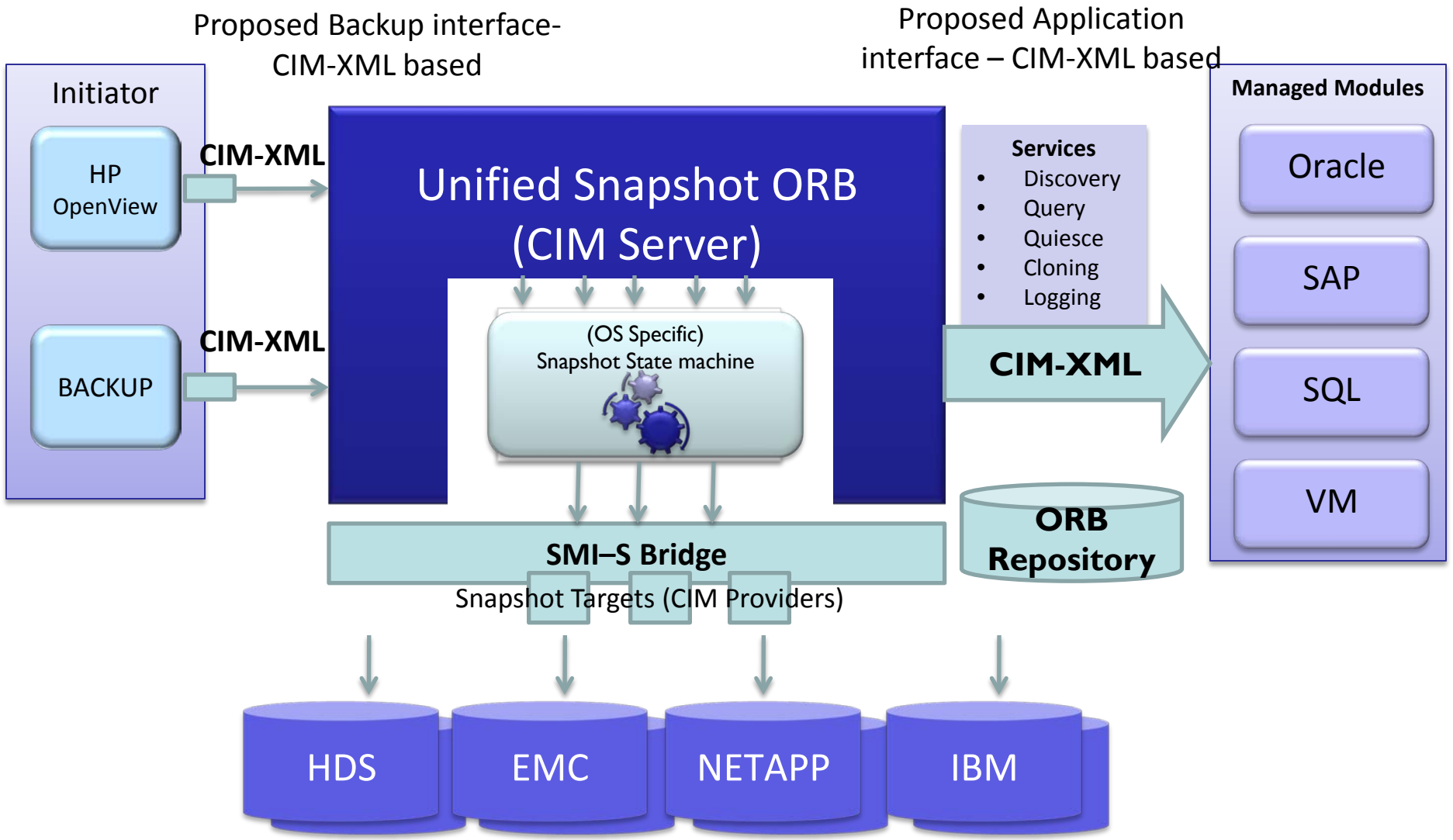
VSS Limitations

- VSS only works on Windows
  - Other operating systems are not covered
- Does not deal with distributed applications
  - E.g. SAP running on one server using Oracle DBMS on another server
- Generally only Windows applications have integration with VSS
  - Very few applications outside Microsoft ones
- Does not cover virtualized environments
  - A host based framework and runs inside VM
  - Cannot distinguish between virtual and physical storage in virtualized environment.
- There are no mechanisms to track application dependencies across servers
  - VSS concept is that every application running on that Windows server is quiesced for backups

Unified snapshots architectural elements

- A methodology to specify all the components of multi-tier applications i.e. parts that compose a logical application as viewed by users
  - Dependencies between applications and in what order they should be quiesced
    - In case of SAP and Oracle – first quiesce SAP and then quiesce Oracle
- A methodology to specify all the data stores used by a logical multi-tier application
  - And what data stores are used by each component
- Architecture of a Unified Snapshot ORB that can run on different operating systems and can work across multiple servers in a distributed system
- A Unified Snapshot Initiator API based on CIM consisting of a set of calls
- A Unified Snapshot Application API that any application can interface to for quiescing
- A Unified Snapshot Target API that storage boxes can support

Unified Snapshot ORB



Specifying structure of multi-tier apps

- A methodology to specify all the components of multi-tier applications i.e. parts that compose a logical application as viewed by users
  - A typical business application (e.g. SAP, MOSS) has multi-tier architecture including
    - Presentation Tier
    - Web/Application Server Tier – can consist of one or more instance of application server
    - Database Tier (e.g. Oracle, DB2) – can consist of one or more databases/distributed databases
    - File System data hosting temporary data or index data
    - OS data – installation , configuration information
  - Dependencies between applications and in what order they should be quiesced
    - In case of SAP and Oracle – first quiesce SAP and then quiesce Oracle
    - In case of MOSS, all index and content databases must be quiesced for consistent data state
  - A methodology is needed to define application hierarchy and these dependencies in snapshot process.
    - A CIM representation to define the logical and physical constituent of the application
    - Identify the dependencies between various components and their backup/restore order

Specifying structure of multi-tier apps


```
<CLASS NAME="CIM_ApplicationComponents" SUPERCLASS="CIM_Application">
   <QUALIFIER NAME="Description" TYPE="string">
      <VALUE>Defines all application constituents - physical or virtual </VALUE>
   </QUALIFIER>
   <Application>
      <Name>SAP-CRM1</Name>
      <HostPath>\\TSys-Corp\LNX2612-R1-H34\SAP-CRM1</HostPath>
      <ServiceLocation>172.17.10.52</ServiceLocation>
      <Instance>{b6cd39af-f25d-11df-a072-806e6f6e6963}</Instance>
      <DependentComponent>
         <Component ID=2 Type=CIM_Database>
            <Name>ORA-DB1</Name>
            <HostPath>\\TSys-Corp\LNX2612-R1-H35\ORA-DB1</HostPath>
            <Instance>{b6cd39af-f25d-11df-a072-806e6f6e6963}</Instance>
            <ServiceLocation>172.17.10.101</ServiceLocation>
            ...
         </Component>
         <Component ID=3 Type=CIM_FileSystem>
            <Name>/usr/sap/trans</name>
            <HostPath>\\TSys-Corp\LNX2612-R1-H34\trans</HostPath>
            <ServiceLocation>172.17.10.52</ServiceLocation>
            ...
         </Component>
      </DependentComponent>
      ....
   </Application>
</CLASS>
```

Specifying structure of multi-tier apps

- A methodology to specify all the data stores used by a logical multi-tier application
  - And what data stores are used by each component
- A typical datacenter, have heterogeneous storage from NAS & SAN environment. E.g.
  - SAP file system data (usr/sap/trans) may be hosted on an NFS share from a Filer
  - Oracle database hosted on LUNs exported from a storage array over FC network
- A CIM representation to specify these complex storage environments and target
  - Define the storage configuration and location
  - Define Physical and Virtual nature of storage
  - Dependencies between various storage elements
    - File -> Volume -> LUN

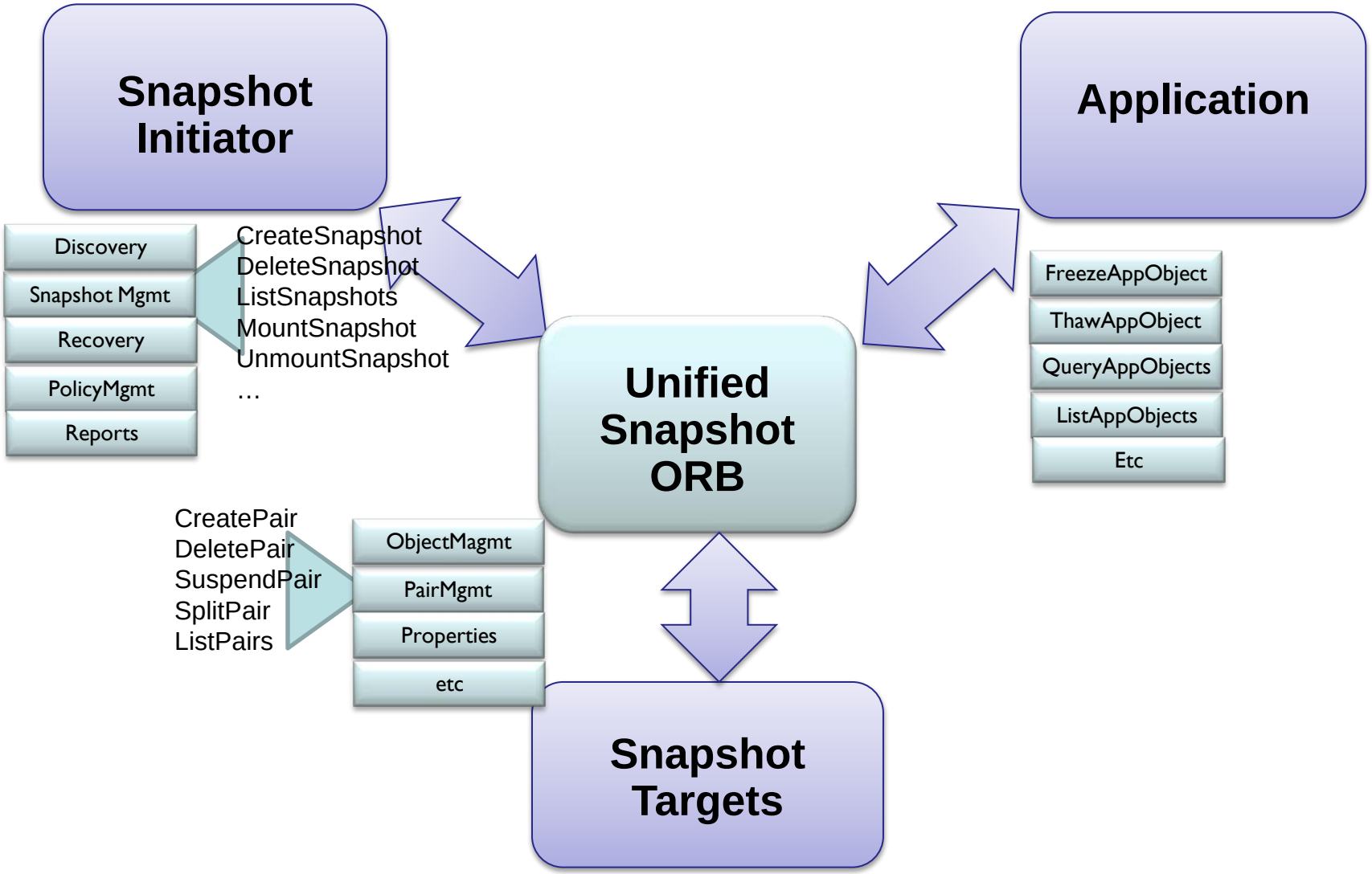
Distributed ORB architecture

- Unified Snapshot ORB will be platform/OS independent snapshot framework
  - Built using OS independent technologies – JAVA and Open source components
  - Designed to be portable and highly modular to run on any OS
  - OS specific components are wrapped under a pluggable architecture
- Designed to run on any datacenter host – regardless of platform and OS
  - Designed to run on any physical or virtualized environment
  - Can run inside a virtual machine on any hypervisor
  - Can run on an appliance under restricted (hardened) environment
- Communicate with application and storage targets
  - An agent based framework to manage any application and storage
  - Uses Open source information model (CIM and SMIS) for communication
- Will support high availability to ensure zero downtime
  - High availability configuration between ORBs itself
  - Stores snapshot metadata in SQL datastore for highly availability and security

Distributed ORB architecture

- Uses open source management framework – Open Pegasus
  - A CIM based communication framework between ORB & Apps
  - Uses Vendor neutral Interface (SNIA's SMI-S specifications) for all storage vendors
- Unified framework requires an open standard that is supported by all vendors
  - Use of CIM facilitates vendor neutral integration
  - Use CIM Core schema to integrate infrastructure components
  - Use CIM Common schema to integrate the applications elements
  - Defined Extension Schema to support Snapshot state and operation
- Unified Snapshot ORB will manage the overall state of snapshot process
  - A state machine that will orchestrate the snapshot operation
- It will execute the snapshot workflow to create consistent copy of application data
  - Identify application and dependent components
  - Prepare storage for snapshot
  - Freeze/Thaw the application and dependent objects
  - Commit snapshot

Unified Snapshot API



Unified Snapshot Initiator API

- An Unified Snapshot Initiator API based on CIM consisting of a set of calls
  - CIM Clients (Backup vendors) invokes the initiator APIs to initiate/request a backup
  - An xmlCIM representation to define the methods

```
<METHOD NAME="CreateSnapshot" TYPE="uint32">
        <METHODPARAMETER Name="APPID">
        <QUALIFIER NAME= ...>...</QUALIFIER>
        <PARAMETER TYPE="uint16"/>
       </METHODPARAMETER>
       ...
</METHOD>
```

  - Methods are invoked on CIM target that is defined by an Application Target identified by Application Structure described earlier.

Unified Snapshot Target & Application APIs

- Storage vendors implement agents using the target APIs (based on SMI-S)
  - Application and Platform vendors implement the Application APIs to participate in Snapshot backups.

- Application vendors implement Application APIs (xmlCIM representation)
  - Application and Platform vendors implement the Application APIs to participate in Snapshot backups.
  - Unified Snapshot ORB will invoke these APIs using CIM-XML protocol

Learning Objectives

We intend to address the challenges faced in data protection of complex enterprise systems

- Efficient Storage Provisioning and Storage Management
- Protect Data from Corruptions and Attacks
- Integration between Applications, Middle tier, OS /Hypervisors, Physical (Server) Layer and Storage
- SNIA – SMI-S and a move to build open standards for providing data protection at all levels.

Biography

Dr. Anupam Bhide – CEO, Co-Founder, Calsoft Inc.

- Storage industry veteran
- More than 21 years of industry experience
- Founder-member of the DB2/6000 Parallel Edition team at IBM Research Center
- Senior Architect in the RDBMS development group at Oracle Corp, designed some of the key features of Oracle8
- Visiting Faculty at University of California – Berkeley
- Ph.D. in Computer Science, University of California-Berkeley
- BS in Computer Science: Indian Institute of Technology, Bombay and MS: University of Wisconsin-Madison