



iSCSI Management API

John Forte
Sun Microsystems

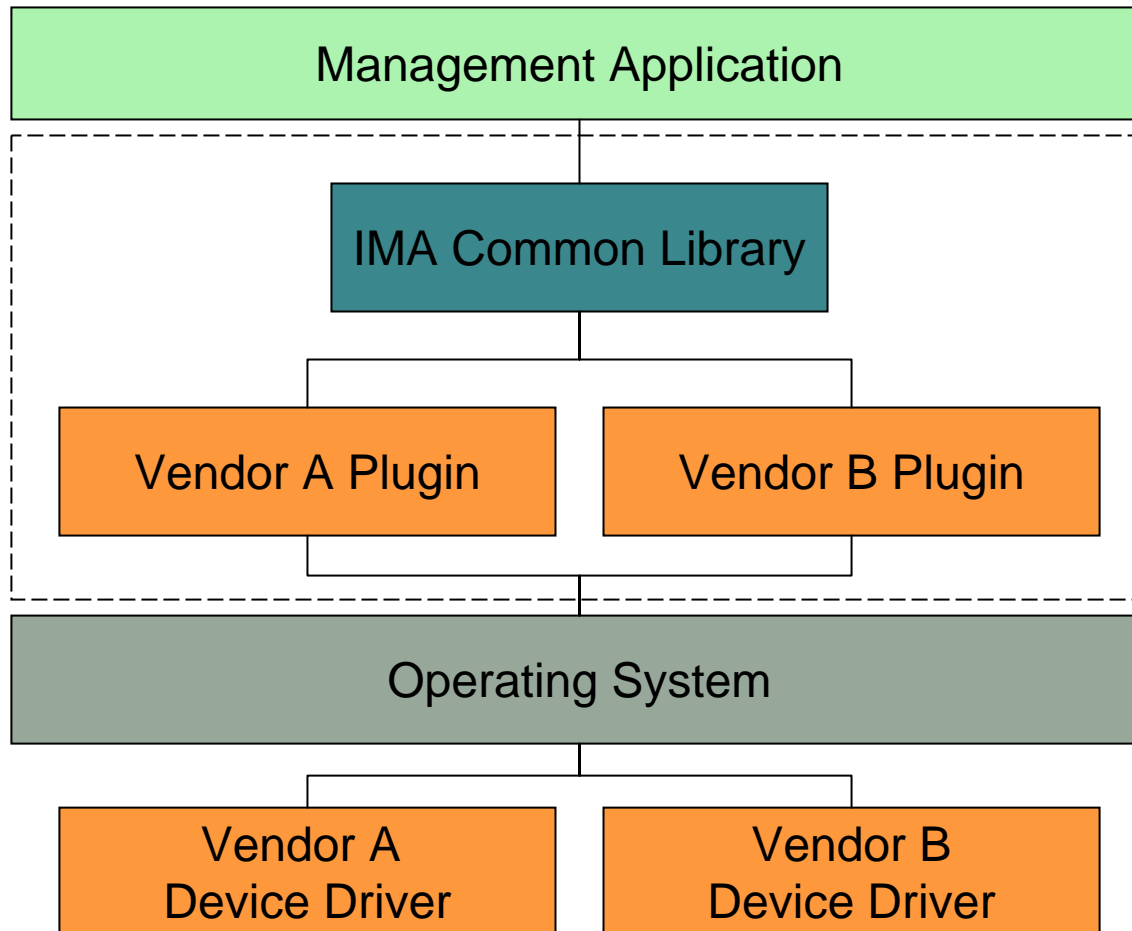
iSCSI Management API

- IMA Introduction
 - Overview
 - Relevance
 - Relation to FC-HBA API, MMA and SMI-S
- IMA Functionality
 - What it provides
- IMA Implementation
 - Objects
 - APIs
- IMA Version 2

IMA Introduction

- C Software Library
- Two Main Components
 - Common library (vendor independent)
 - Zero or more plugins (vendor specific)
- Facilitates initiator based management for iSCSI storage
- iSCSI analogue to FC-HBA for fibre channel
- Complements Multipath Management API
- SMI-S
 - Interfaces with iSCSI provider

IMA Introduction: IMA Components



IMA Introduction: Relevance

- Common API across vendors
- Enables smaller scale iSCSI management apps
- Enabling API for SMI-S
 - iSCSI Initiator Profile IMA mapping

IMA Functionality

- Provides
 - Initiator view of the IP SAN
 - Target and logical unit enumeration
 - Management of discovery methods
 - Configuration of login parameter preferences
 - Configuration of authentication parameters
 - Callbacks for status / topology changes
 - Statistics collection
 - Support for both software and hardware initiators

IMA Implementation

- Object based framework
 - Object ID (OID) uniquely identifies an object
 - Object properties are gettable and optionally settable
- OID details
 - Object type
 - Object owner

IMA Implementation: Objects

- iSCSI Objects
 - Nodes
 - iSCSI Targets
 - Network Portals

IMA Implementation: Objects

- Logical Objects
 - Logical HBAs
 - Representation of a SCSI HBA to the OS
 - Logical Network Ports
 - Collection of one or more physical ports
 - Plugins
 - Zero or more

IMA Implementation: Objects

- Physical Objects
 - Physical HBAs
 - Physical Network Ports

IMA Implementation: APIs

- General APIs
 - Get a list of OIDs of a given type
 - Get object properties for a given OID
 - Register callbacks for status and topology changes

IMA Implementation: APIs

- Discovery APIs
 - IMA_SetIsnsDiscovery
 - IMA_SetSendTargetsDiscovery
 - IMA_Add/RemoveStaticDiscoveryTarget
 - IMA_Add/RemoveDiscoveryAddress

IMA Implementation: APIs

- Logical Unit APIs
 - IMA_GetLuOidList
 - Gets all logical unit objects for a given LHBA or target
 - IMA_LuReportLuns
 - IMA_LuInquiry
 - IMA_GetLuProperties
 - IMA_Expose/UnexposeLu

IMA Implementation: APIs

- Logical Unit Properties Structure

```
typedef struct IMA_lu_properties
{
    IMA_OID          associatedTargetOid;
    IMA_BYTE         targetLun[8];

    IMA_BOOL         exposedToOs;
    IMA_DATETIME    timeExposedToOs;

    IMA_BOOL         osDeviceNameValid;
    IMA_WCHAR       osDeviceName[64];

    IMA_BOOL         osParallelIdsValid;
    IMA_UINT32      osBusNumber;
    IMA_UINT32      osTargetId;
    IMA_UINT32      osLun;

    IMA_BYTE        reserved[128];
} IMA_LU_PROPERTIES;
```

IMA Implementation: APIs

- Target APIs
 - IMA_GetTargetOidList
 - Gets all discovered target objects for a given LHBA or a logical network port
 - IMA_SetMaxConnections
 - IMA_GetTargetErrorStatistics
 - IMA_GetAddressKeys
 - IMA_GetTargetProperties

IMA Implementation: APIs

- Target Properties Structure

```
typedef struct IMA_target_properties
{
    IMA_OID          associatedNodeOid;
    IMA_OID          associatedLhbaOid;

    IMA_NODE_NAME    name;
    IMA_NODE_ALIAS   alias;
    IMA_UINT32       discoveryMethodFlags;

    IMA_BOOL         sendTargetsDiscoverySettable;
    IMA_BOOL         sendTargetsDiscoveryEnabled;
    IMA_BYTE         reserved[128];
} IMA_TARGET_PROPERTIES;
```

IMA Implementation: APIs

- Authentication APIs
 - IMA_SetInitiatorAuthMethods
 - IMA_SetInitiatorAuthParms
- Status / Topology Callback APIs
 - IMA_[De]RegisterForObjectPropertyChanges
 - IMA_[De]RegisterForObjectVisibilityChanges

IMA Version 2

- New Functionality
 - RADIUS Server Support
 - Header/Data Digest Support
 - Mutual Authentication
 - Session And Connection Objects

IMA Version 2:APIs

- IMA_GetSessionOidList
- IMA_GetConnectionOidList
- IMA_GetSessionProperties
- IMA_GetConnectionProperties
- IMA_SetRadiusAccess



IMA Version 2:APIs

- Session Properties Structure

```
typedef struct _IMA_session_properties
{
    IMA_OID          associatedLhbaOid;
    IMA_OID          associatedTargetOid;
    IMA_AUTHMETHOD   authMethod;
    IMA_BOOL         dataPduInOrder;
    IMA_BOOL         dataSequenceInOrder;
    IMA_UINT32       defaultTime2Retain;
    IMA_UINT32       defaultTime2Wait;
    IMA_UINT32       errorRecoveryLevel;
    IMA_UINT32       firstBurstLength;
    IMA_BOOL         immediateData;
    IMA_BOOL         initialR2T;
    IMA_BYTE         isid[6];
    IMA_UINT32       maxBurstLength;
    IMA_UINT32       maxConnections;
    IMA_UINT32       maxOutstandingR2T;
    IMA_UINT16       targetPortalGroupTag;
    IMA_UINT16       tsih;
} IMA_SESSION_PROPERTIES;
```



IMA Version 2:APIs

- Connection Properties Structure

```
typedef struct _IMA_connection_properties
{
    IMA_OID                associatedSessionOid;
    IMA_UINT16             connectionId;
    IMA_DIGEST_TYPE       dataDigest;
    IMA_DIGEST_TYPE       headerDigest;
    IMA_BOOL               ifMarker;
    IMA_UINT32             ifMarkInt;
    IMA_UINT32             maxRecvDataSegmentLength;
    IMA_UINT32             maxTransmitDataSegmentLength;
    IMA_BOOL               ofMarker;
    IMA_UINT32             ofMarkInt;
} IMA_CONNECTION_PROPERTIES;
```

IMA Status

- SNIA IPS TWG
- Version 1 Due to become an ANSI standard CY08
- Version 2 will follow soon after
- Common library sourceforge project
 - <http://sourceforge.net/projects/iscsi-mgmt-api/>
 - iscsi-mgmt-api-reflib@lists.sourceforge.net
- Common Library and Plugin Implementations
 - Solaris, OpenSolaris, Linux



IMA Coding Example

```
/*
 * This function processes a "system" by getting all of the LHBAs
 * in the system and processing each of those LHBAs.
 */
void ProcessSystem()
{
    IMA_OID_LIST *pList;
    /* Get the list of LHBA IDs. */
    IMA_STATUS status = IMA_GetLhbaOidList(&pList);
    if ( IMA_SUCCESS(status) )
    {
        /* Iterate through all of the LHBAs in the system processing */
        /* the targets of each LHBA. */
        for ( unsigned i = 0; i < pList->oidCount; i++ ) {
            void ProcessLhba(pList->oids[i]);
        }
        /* Always remember to free an object ID list when it's no longer */
        /* needed. */
        IMA_FreeMemory(pList);
    }
}
```

IMA Coding Example

```
/*
 * This function processes an LHBA by getting the list of targets
 * associated with the LHBA and processing each of those targets.
 */
void ProcessLhba(IMA_OID lhbaOid)
{
    IMA_OID_LIST *pList;
    /* Get the list of target OIDs for this LHBA. */
    IMA_STATUS status = IMA_GetTargetOidList(lhba, &pList);

    if ( IMA_SUCCESS(status) ) {
        for ( unsigned i = 0; i < pList->oidCount; i++ ) {
            void ProcessTarget(pList->oids[i]);
        }
        /*
         * Always remember to free an object ID list when it's no longer
         * needed. Failing to do so will cause memory leaks.
         */
        IMA_FreeMemory(pList);
    }
}
```



IMA Coding Example

```
/*
 * This function processes a target by getting all of the LUNs of the
 * target and then processing each of the LUs associated with a LUN.
 */
void ProcessTarget(IMA_OID targetOid)
{
    IMA_BYTE reportLunsBuf[8192];
    IMA_STATUS status = IMA_LuReportLuns(targetOid, IMA_FALSE, 0,
        reportLunsBuf, sizeof(reportLunsBuf), NULL, 0);
    if ( IMA_SUCCESS(status) ) {
        unsigned listLength = GetReportLunsListLength(reportLunsBuf);
        for ( unsigned i = 8; i < listLength; i += 8 ) {
            IMA_UINT64 lun = GetLun(reportLunsBuf, i);
            ProcessLu(targetOid, lun);
        }
    }
}
```

Contact Information

- SNIA IP Storage TWG
 - snia-ips@snia.org
- Sourceforge
 - iscsi-mgmt-api-reflib@lists.sourceforge.net
- OpenSolaris
 - storage-discuss@opensolaris.org