SyncML Device Management

An overview and toolkit implementation

Radu State

Ph.D.

MADYNES

The MADYNES Research Team
LORIA – INRIA Lorraine
615, rue du Jardin Botanique
54602 Villers-lès-Nancy
France
Radu.State@loria.fr

Tutorial Outline

- SyncML-DM : an Approach for Managing Dynamic Devices
  - Representation Protocol for Device Management
  - Device Management Protocol
  - Standardized Objects
  - Device Management Tree
  - MAD-MAX Toolkit
  - Benchmarkings

Towards open data synchronization

2000: creation of the SyncML consortium (www.syncml.org).

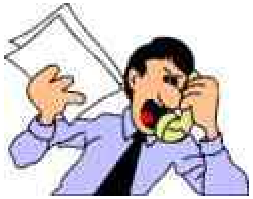
Members : More than 600 companies-IBM, Nokia, Motorola, Ericsson, Matsushita, Symbian, Openwave …..

Objective : Develop an Open Standard for data Synchronization

Additional Results : Device Management Framework for managing devices .

Synchronization Markup Language is an Open Specification for universal synchronization

Use cases for Device Management



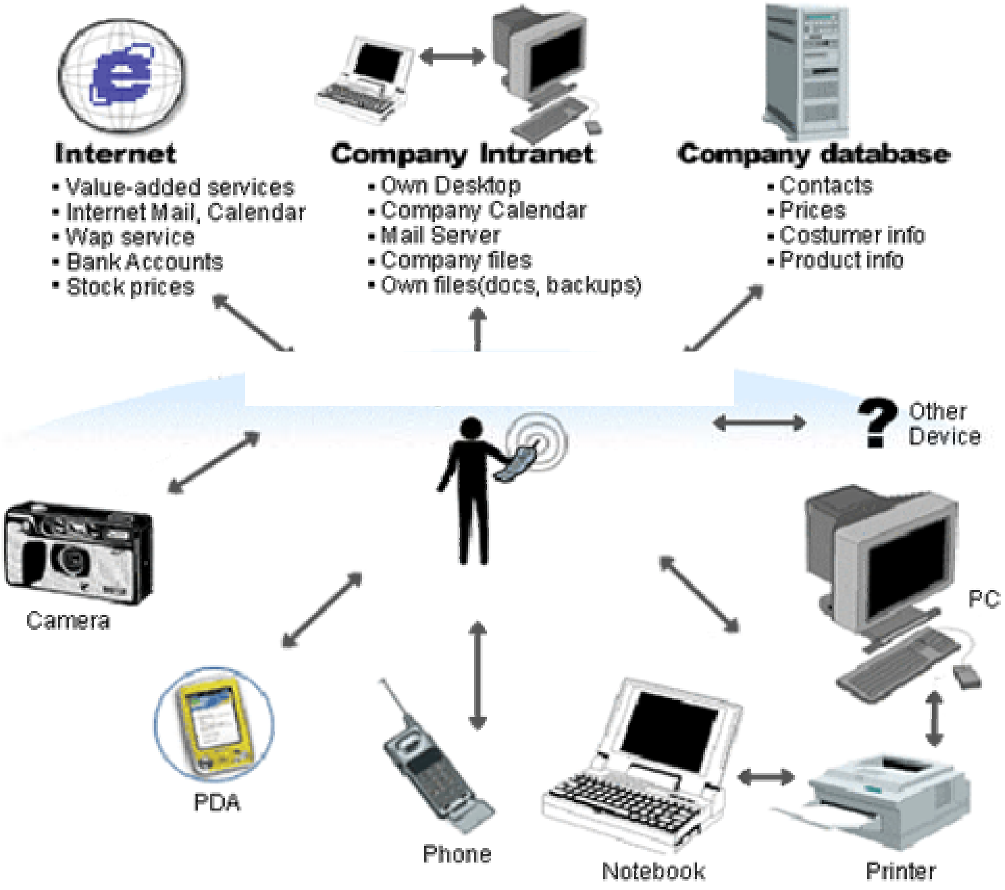
•Troubleshooting

•Over the air Mass Configuration

- Remote service management
- Personal Management
- Storage Management
- Monitoring
- Software download

Challenges in Device Management

- Heterogeneous devices
- Many applications
- Multiple network connectivity
- Limited resources



Internet

- Value-added services
- Internet Mail, Calendar
- Wap service
- Bank Accounts
- Stock prices

Company Intranet

- Own Desktop
- Company Calendar
- Mail Server
- Company files
- Own files(docs, backups)

Company database

- Contacts
- Prices
- Costumer info
- Product info

Requirements for Device Management

- Operate effectively over wireless and wired networks
- Support a variety of transport protocols
- Support arbitrary networked data
- Enable data access from a variety of applications
- Address the resource limitations of the mobile device
- Build upon existing Internet and Web technologies

Wireless/Wired Network Operations

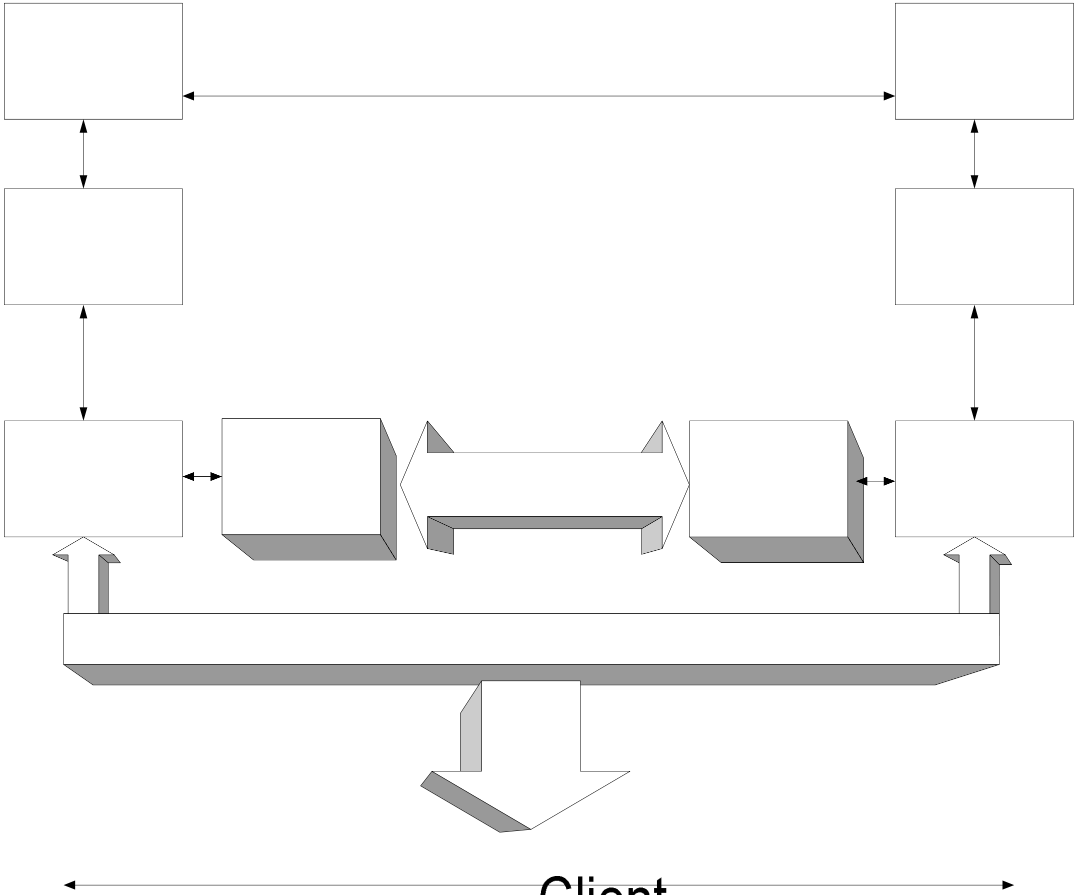
Ubiquitous wireless access

1. High network latency
2. Limited bandwidth
3. Low reliability of both data and connectivity
4. Dynamic Addresses and network connectivity
5. Firewall, Nats – several management domains
6. Out of coverage factors

Redefine Management

1. Management might be to late
2. Manage efficiently (whenever it's required without wasting resources)
3. Connection Oriented management
4. Application level naming and addressing
5. Device initiated management
6. Different Fault behavior management

Scope of SyncML



Client
Application

XML based framework for data synchronization

Message oriented data exchange protocol

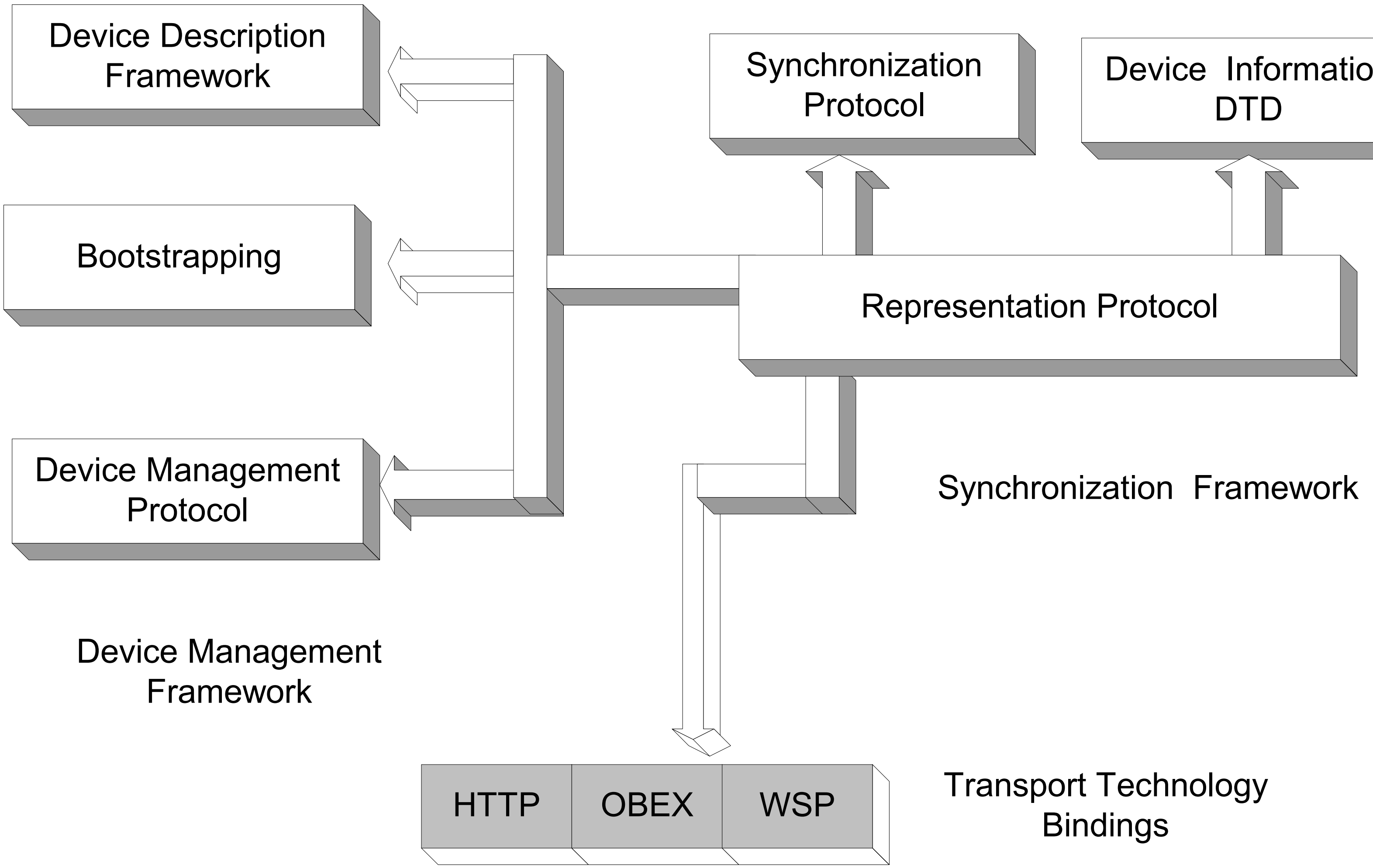
Transport agnostic

Universal deployment

Extension for device management

Application Leve

SyncML Specifications – More than just XML

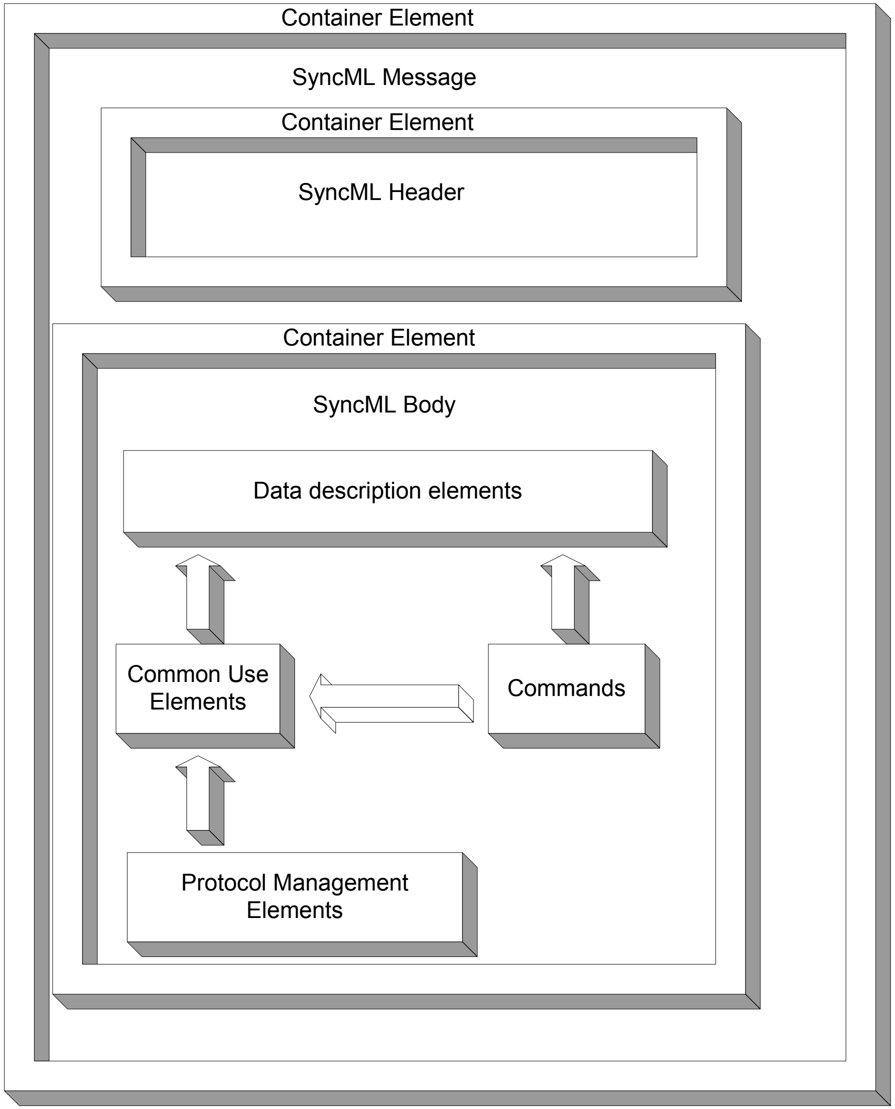

Device Description Framework
Bootstrapping
Device Management Protocol
Synchronization Protocol
Device Information DTD
Representation Protocol
Synchronization Framework
Device Management Framework
HTTP
OBEX
WSP
Transport Technology Bindings

Representation Protocol - Objective

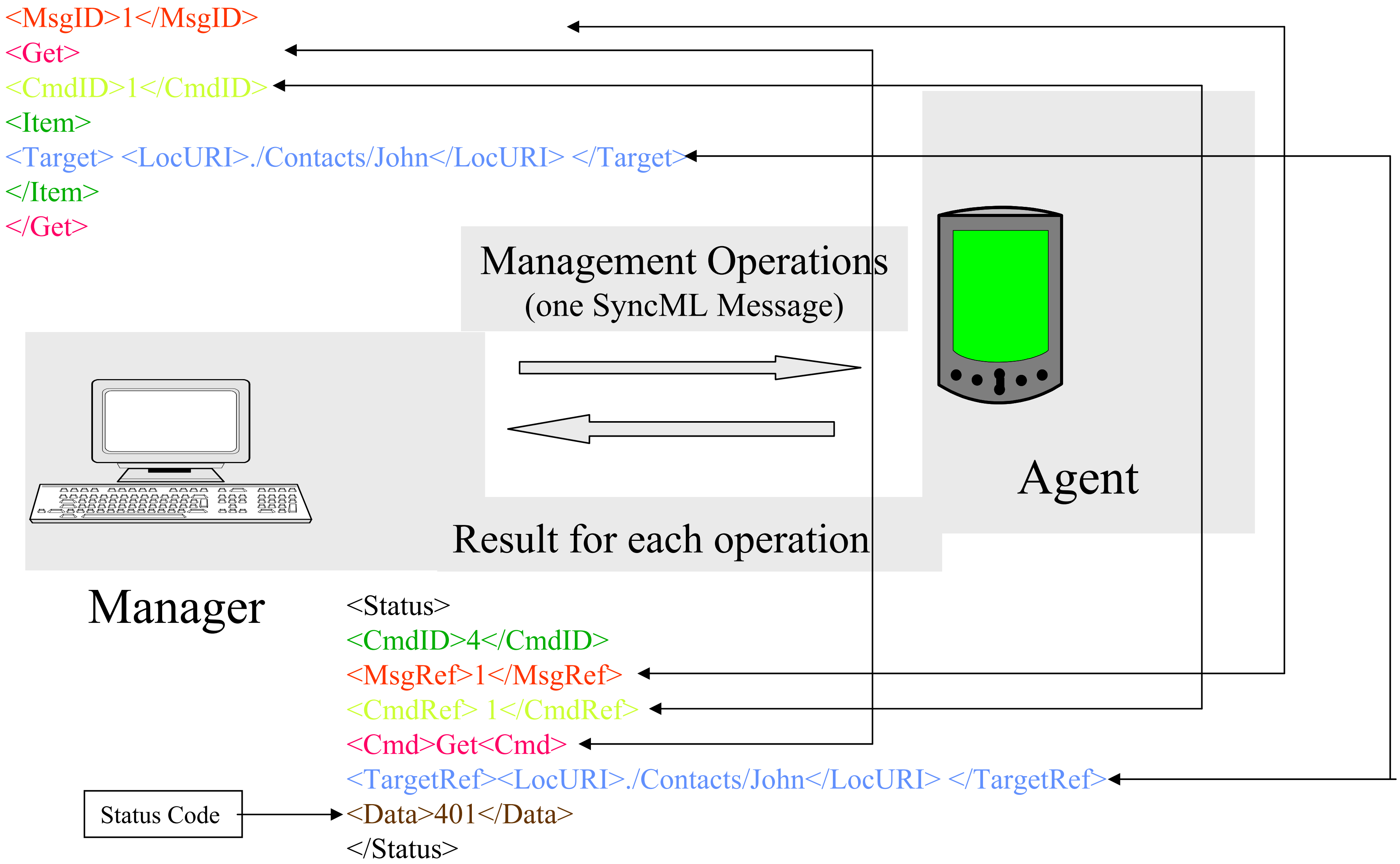
1. Specifies the structure of SyncML messages
2. Defines a vocabulary to describe management data and operations
3. Core component of the SyncML framework on which all others components are based

We will cover them in this order:

•Common Use Elements
•Protocol Management Elements
•Command Elements
•Data Description Elements
•Message Container Elements



Protocol Management Element(s)



```
<MsgID>1</MsgID>
<Get>
<CmdID>1</CmdID>
<Item>
<Target> <LocURI>./Contacts/John</LocURI> </Target>
</Item>
</Get>
```

```
<Status>
<CmdID>4</CmdID>
<MsgRef>1</MsgRef>
<CmdRef> 1</CmdRef>
<Cmd>Get<Cmd>
<TargetRef><LocURI>./Contacts/John</LocURI> </TargetRef>
<Data>401</Data>
</Status>
```

Command Elements –functional classification

1. Data Command Elements – used to change application data

   Add - creates a new interior node
   Copy –copies values from a node to another at the client side
   Delete – deletes a node (and all its subnodes)
   Exec – process execution on the target
   Replace – overwites value for an existing node
   Get – retrieves data from the target

1. DataStore Command Elements – Actions for an entire datastore

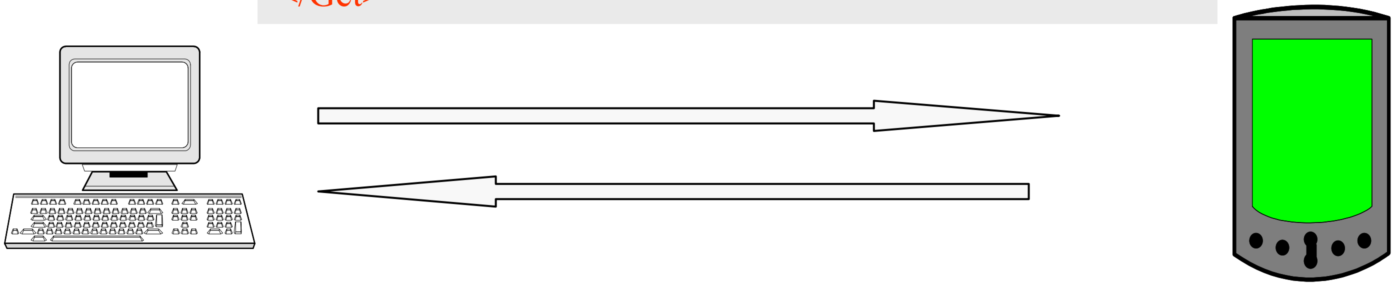
   Alert – used for notifications, text displays
   Results – contains results from a Get

2. Process Flow Commands – enhanced processing control

   Atomic – all subcommands must be executed
   Sequence – subcommands must be executed in order

The Duo: Get and Results

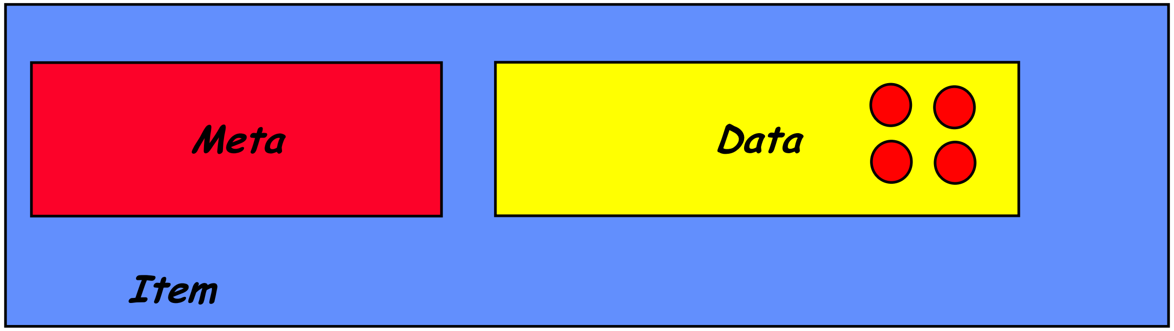
```
<MsgID>1</MsgID>
<Get>
<CmdID>2</CmdID>
<Item>
<Target> <LocURI>./Contacts/John</LocURI> </Target>
</Item>
</Get>
```



```
<Results>
<MsgRef>1</MsgRef>
<CmdRef>2</CmdRef>
<CmdID>2</CmdID>
<Item>
<Source> <LocURI>./Contacts/John</LocURI> </Source>
<Data>Tel:01564433</Data>
</Item>
</Results>
```

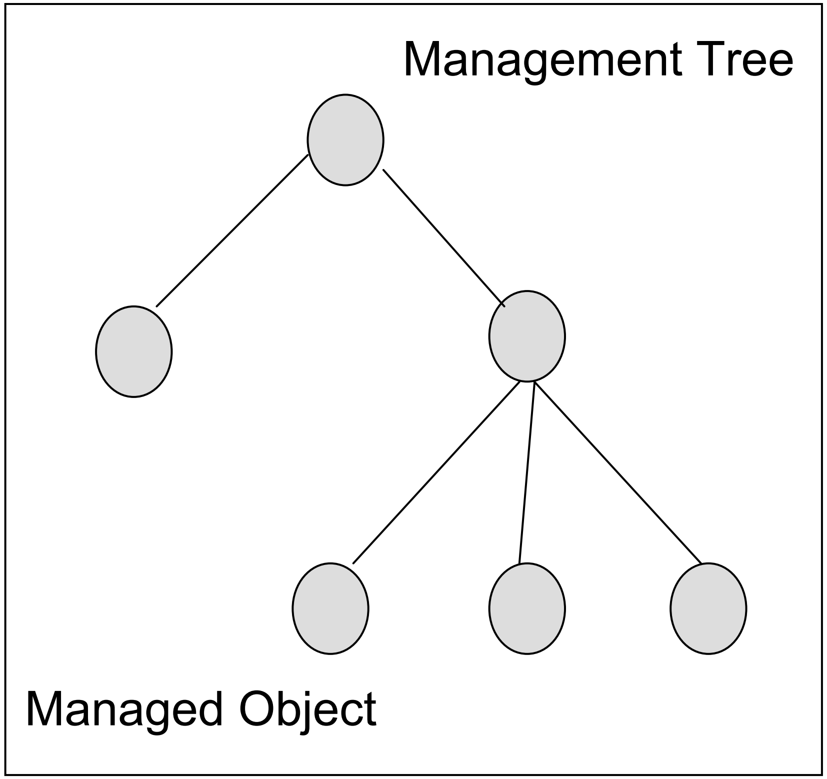
Data Description Elements

1. <Data>….</Data> encloses SyncML payload data
2. <Item>….</Item>
   - isolates a command from the underlying data
   - Contains Data, Identification and Metadata
3. <Meta>…..</Meta> provides meta-information about the data
   - Type of the data
   - Size of the data



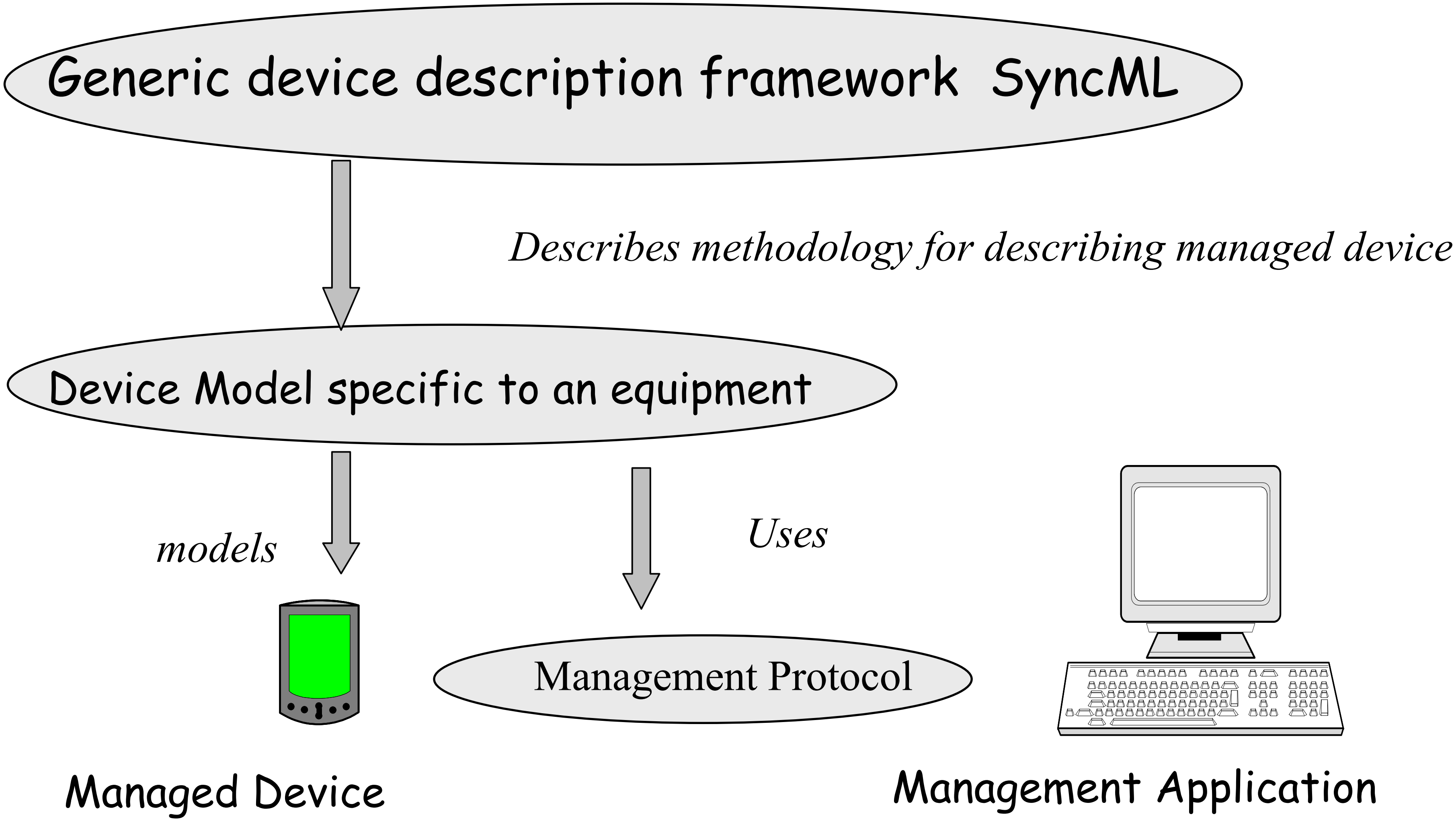
Device Management Framework

Describes the management information and how to access it



1. Framework for describing management information
2. Management Tree
3. Standardized Objects

Purpose of the Device Management Framework



Generic device description framework SyncML

Describes methodology for describing managed device

Device Model specific to an equipment

models

Uses

Management Protocol

Managed Device

Management Application

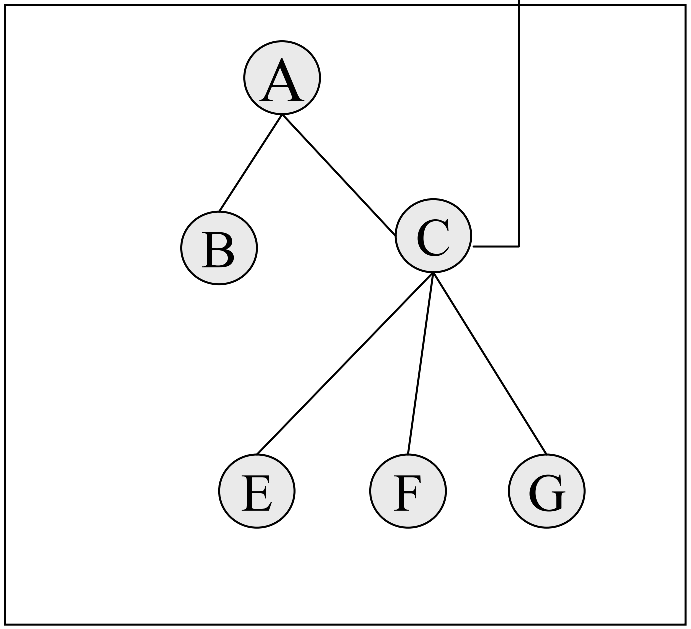
Modeling Management Information in SyncML

```
<!ELEMENT Node (NodeName, Path?, RTProperties?, DFProperties, (Node* | Value?))>
<!ELEMENT NodeName (#PCDATA)>
<!ELEMENT Path (#PCDATA)>
<!ELEMENT Value (#PCDATA)>
<!ELEMENT RTProperties (ACL, Format, Name, Size?, Title?, TStamp?, Type?, VerNo?)>
<!ELEMENT ACL (#PCDATA)>
<!ELEMENT Format (b64 | bool | chr | int | node | null | xml)>
```

Dynamic self-described management information

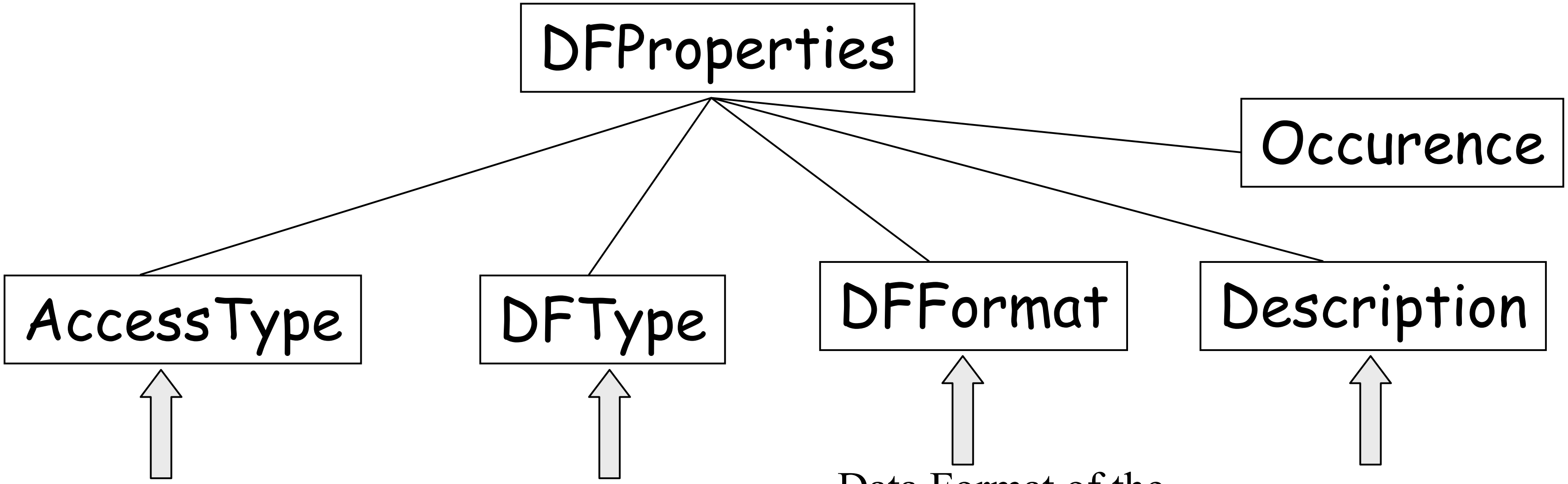
XML Tree Node captures information about :

- Name of the managed object
- ACL
- Subtrees
- Value
- Data Type of the Managed Object



Management Information

Framework properties of a Managed Object



DFProperties

Occurence

AccessType

DFType

DFFormat

Description

Which operations are possible ?
Add, Copy, Delete, Exec,
Get, Replace

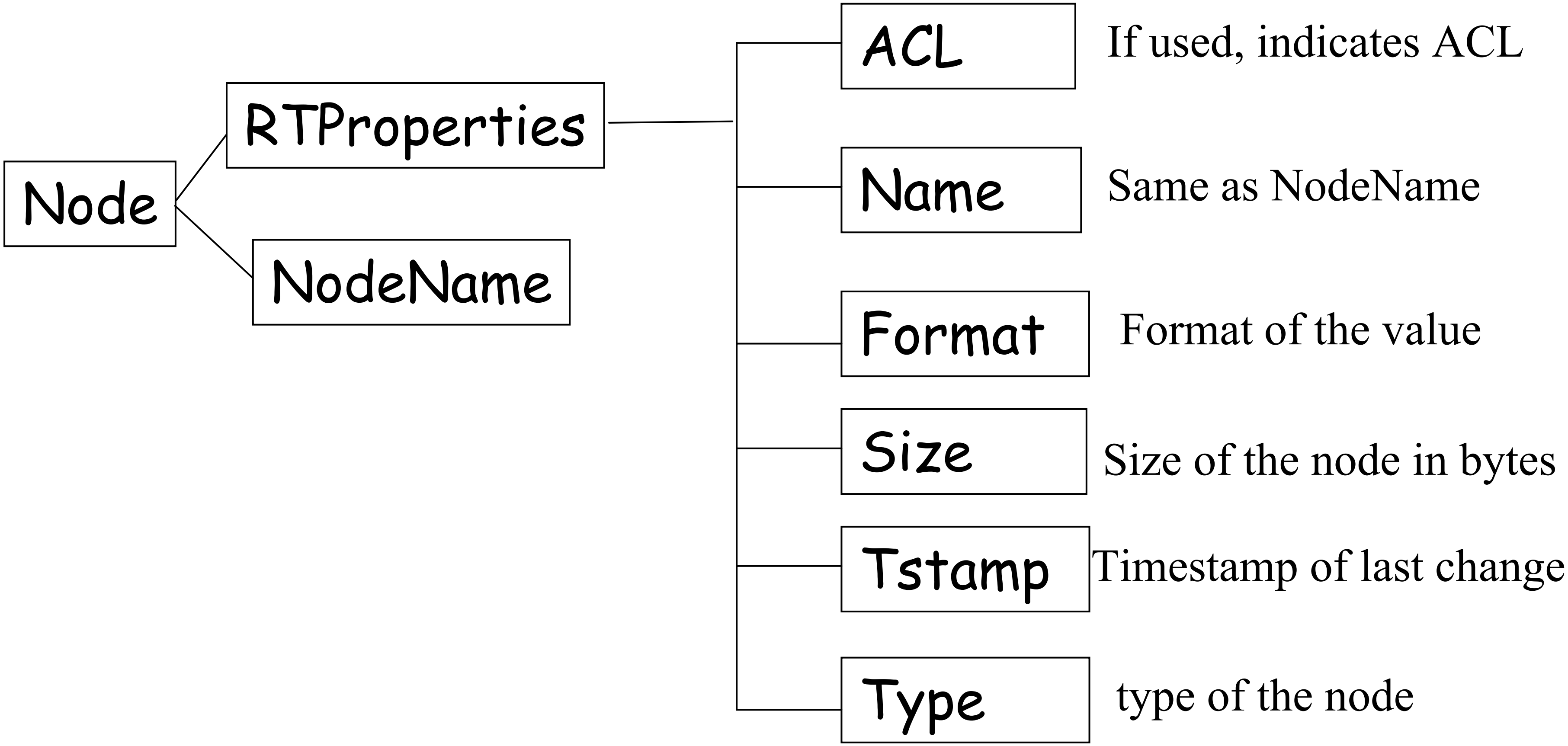
Mime type
of the node value

Data Format of the node
Eg. bool, chr, int,
node, null, xml

Description of the node

Framework properties do not change at run-time.
Analogous to a MIB2 definition

Runtime Properties of the Managed Object



Node

RTProperties

NodeName

ACL — If used, indicates ACL

Name — Same as NodeName

Format — Format of the value

Size — Size of the node in bytes

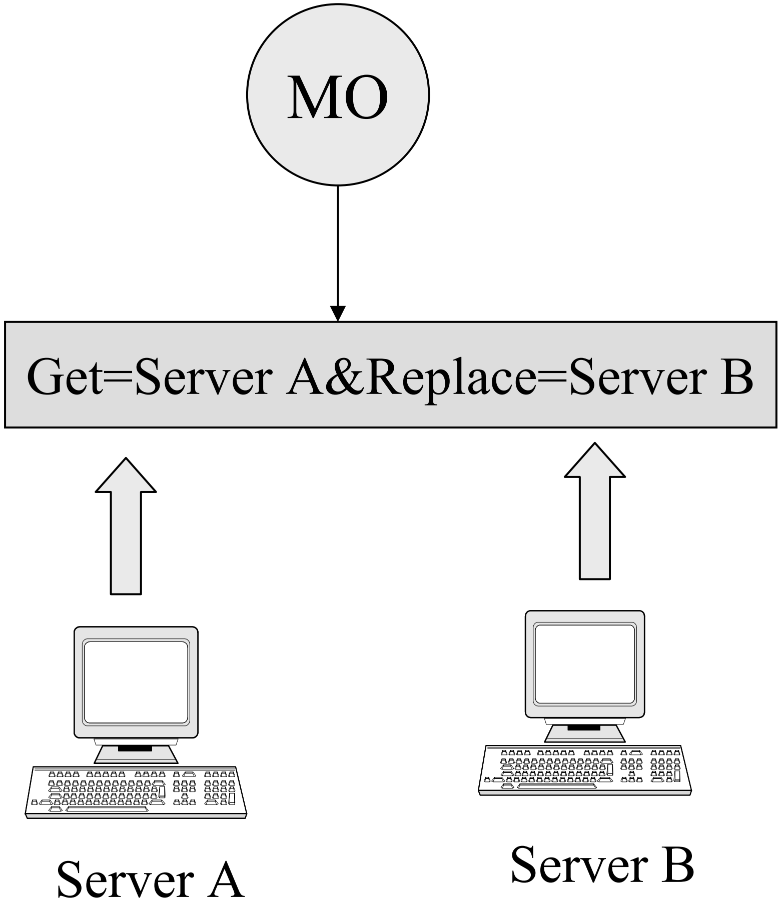
Tstamp — Timestamp of last change

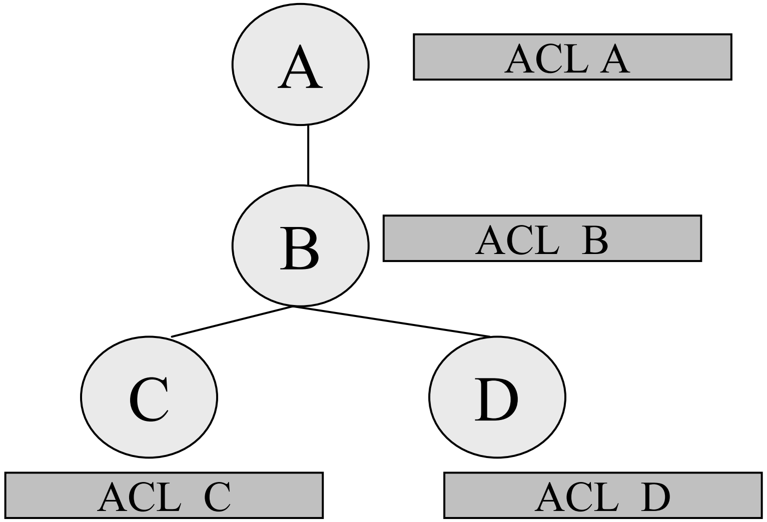
Type — type of the node

Objective : Describe instantiated Managed Object

Access Control Lists (ACL)

ACL Property regulate access to a MO



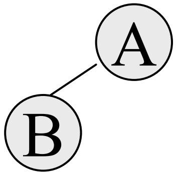


ACL operation

1. Internal Nodes – access to ACL is regulated by the ACL itself
2. Terminal Nodes – Parent Node ACL regulates ACL modifications
3. If ACL empty then closest ancestor ACL is used

Addressing Object Values and Properties

Addressing node values :
Object is identified by complete path to the root of the management tree
Example : *<LocURI>NodeA/NodeB</LocURI>*



Extended usage of the tags:
Meta, Format, Type

<Meta>
*<Format>*chr*</Format>*
*<Type>*text/plain*</Type>*
</Meta>

Meta to indicate metainformation
Format (string in this example) for the data format via *Runtime property Format*
Type =value for the Mime type

Addressing Property values : *node URI+?prop=<property_name*>
Example : addressing ACL property of Node B : */SyncML/NodeA/NodeB?prop=ACL*

Device Management Standardized Objects

Mandatory Device Management Objects for any SyncML device

Management Information Regarding :

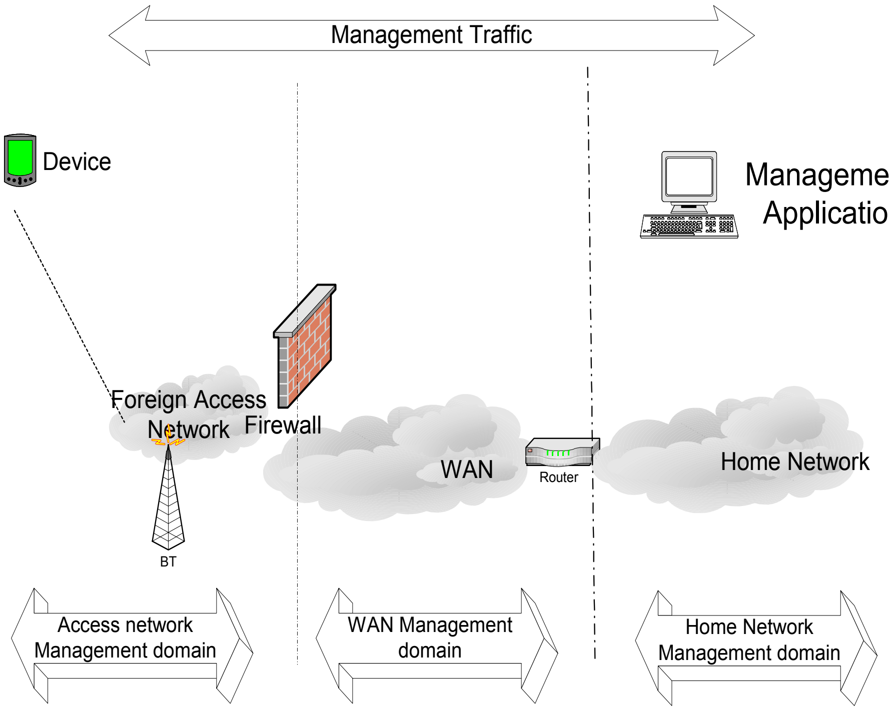
1. Connectivity information (protocol IPv4/IPV6, addresses, ports)
2. Accepted authentication methods
3. Bearer type (eg. Obex, GSM, CDMA)
4. Management server ID
5. Manufacturer ID,
6. Model and Device ID
7. Software version

Think MIB 2 for device management…..

Requirements for the Device Management Protocol

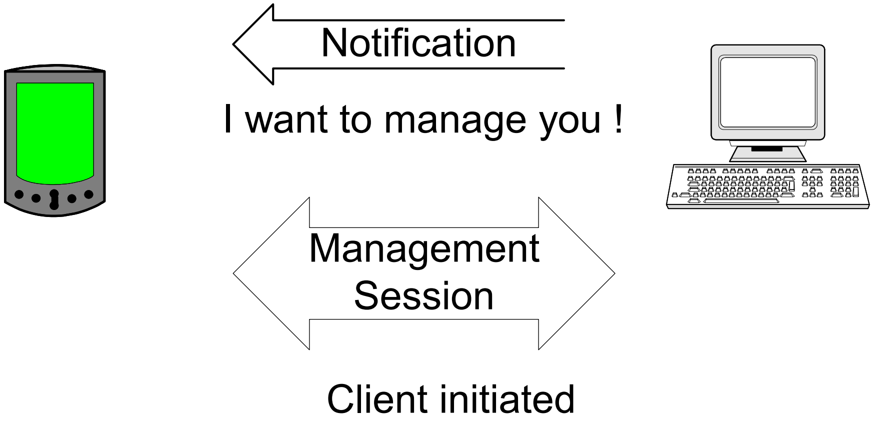
Requirement: Ability to deal with dynamic environments

- Dynamic network connectivity
- Unreliable communication medium
- Not always on-line devices due to out of coverage or limited mobility management
- Limited incoming connections for devices



Design choices for the Device Management Protocol

1. Session Oriented
2. Client initiated
3. Out of band notification support
4. XML encoded
5. User interaction enabled



Notification

I want to manage you !

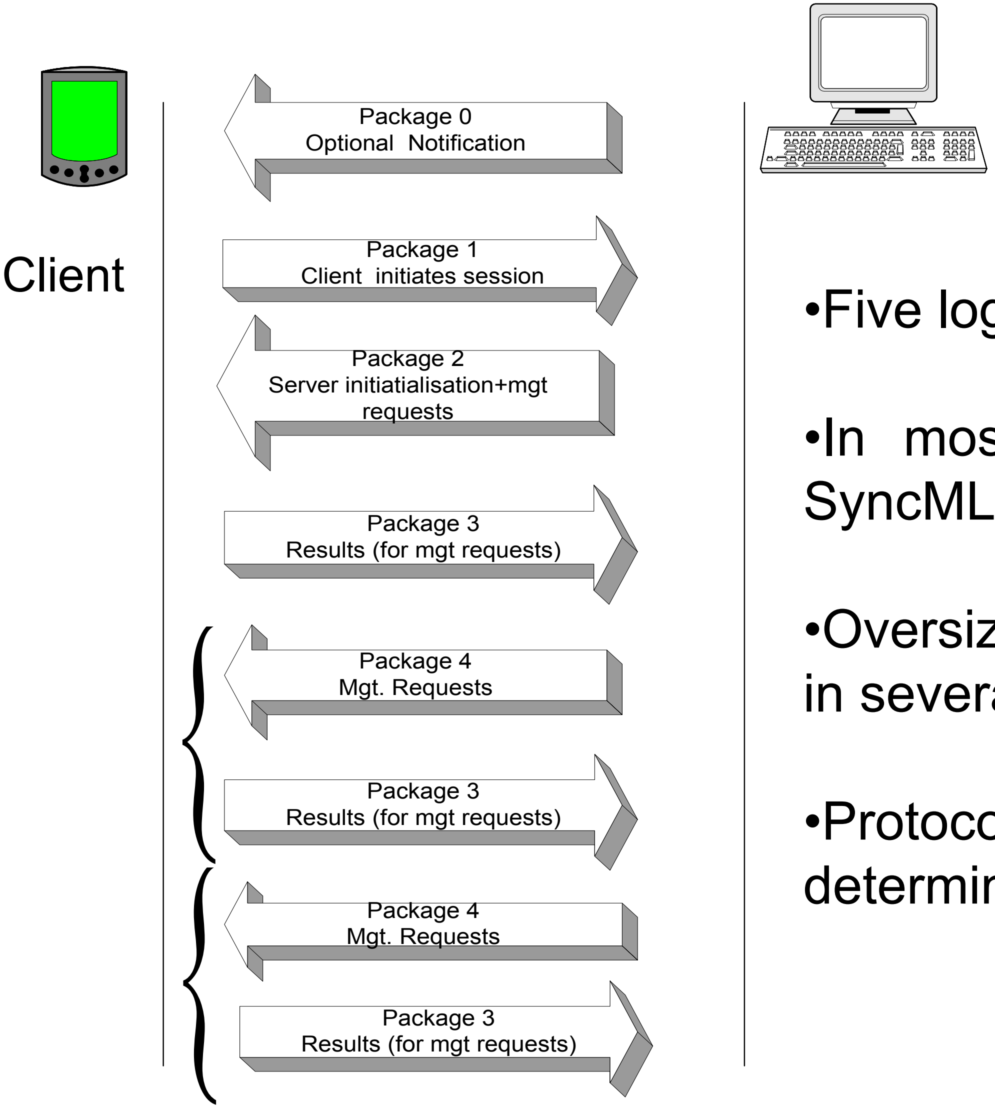
Management Session

Client initiated

Rationale

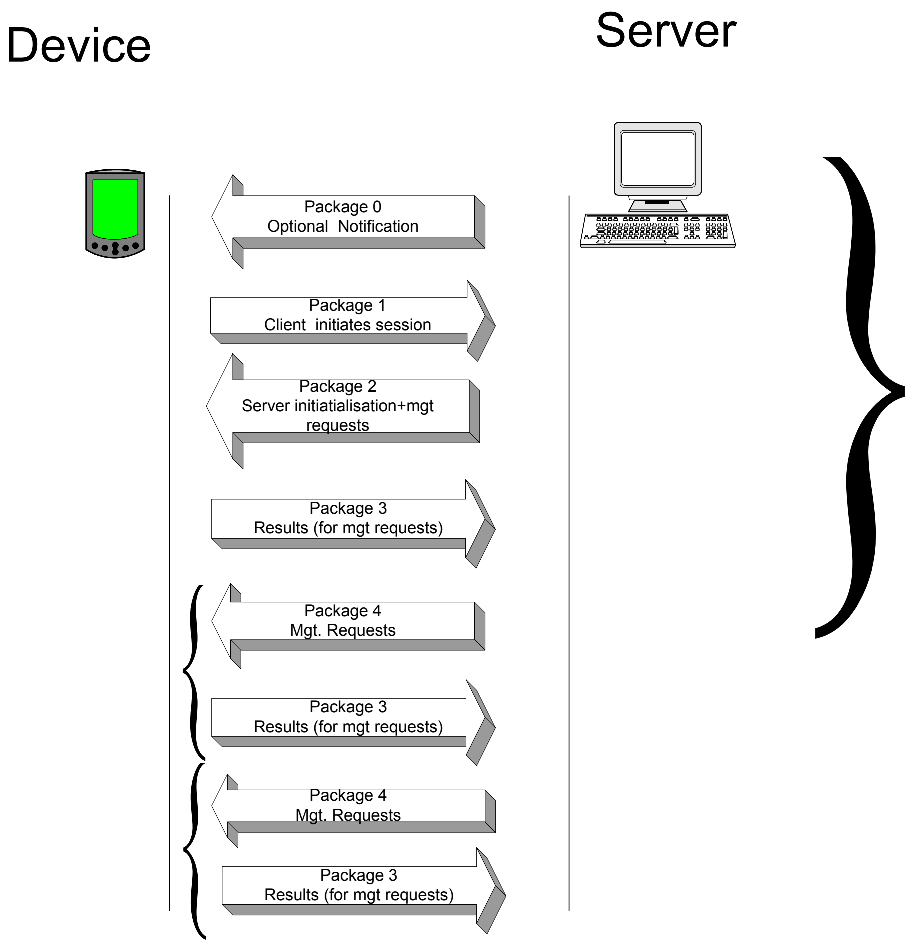
1. Devices might not be always online
2. Most firewalls will allow only device initiated connections
3. Notification support needed for “server initiated “ sessions
4. User might have other priorities

A Management Session



- Five logical packages
- In most cases one package is a SyncML message
- Oversized packages can be included in several message
- Protocol state (each site) is determined by current protocol

Security in SyncML Management



1. Authentication of Server
2. Integrity check of the message (MD5/basic authentication)

1. Two way authentication
2. Integrity check of the message
3. Access Control List mechanism
4. Confidentiality provided by transport level (HTTPS, SSL, OBEX)

Existing Implementations

Proprietary implementations :

Major actors and contributors to the standard.

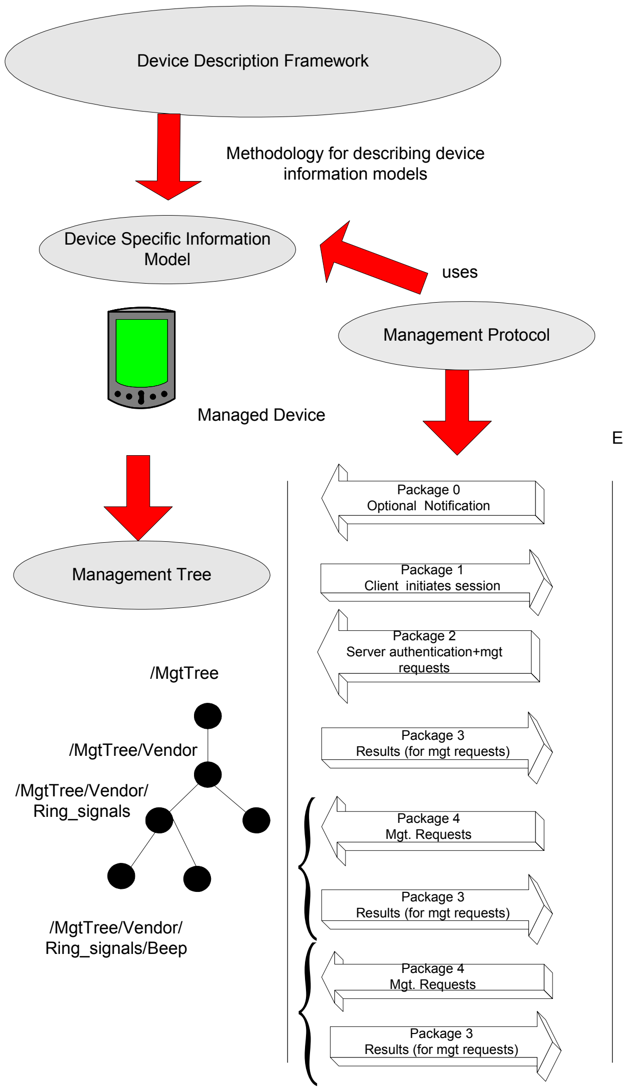
Open Source :

Sync4J (http://sync4j.sourceforge.net) for the data synchronization.
No device management functionality.

LORIA/MADYNES :

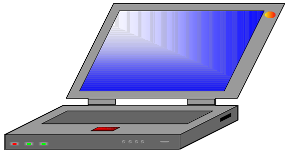
SyncML Agent toolkit (www.madynes.org/software)

MAD-MAX –Overall functionality



Device Description Framework
Methodology for describing device information models
Device Specific Information Model
uses
Management Protocol
Managed Device
Management Tree
/MgtTree
/MgtTree/Vendor
/MgtTree/Vendor/ Ring_signals
/MgtTree/Vendor/ Ring_signals/Beep
Package 0
Optional Notification
Package 1
Client initiates session
Package 2
Server authentication+mgt requests
Package 3
Results (for mgt requests)
Package 4
Mgt. Requests
Package 3
Results (for mgt requests)
Package 4
Mgt. Requests
Package 3
Results (for mgt requests)

Management Application

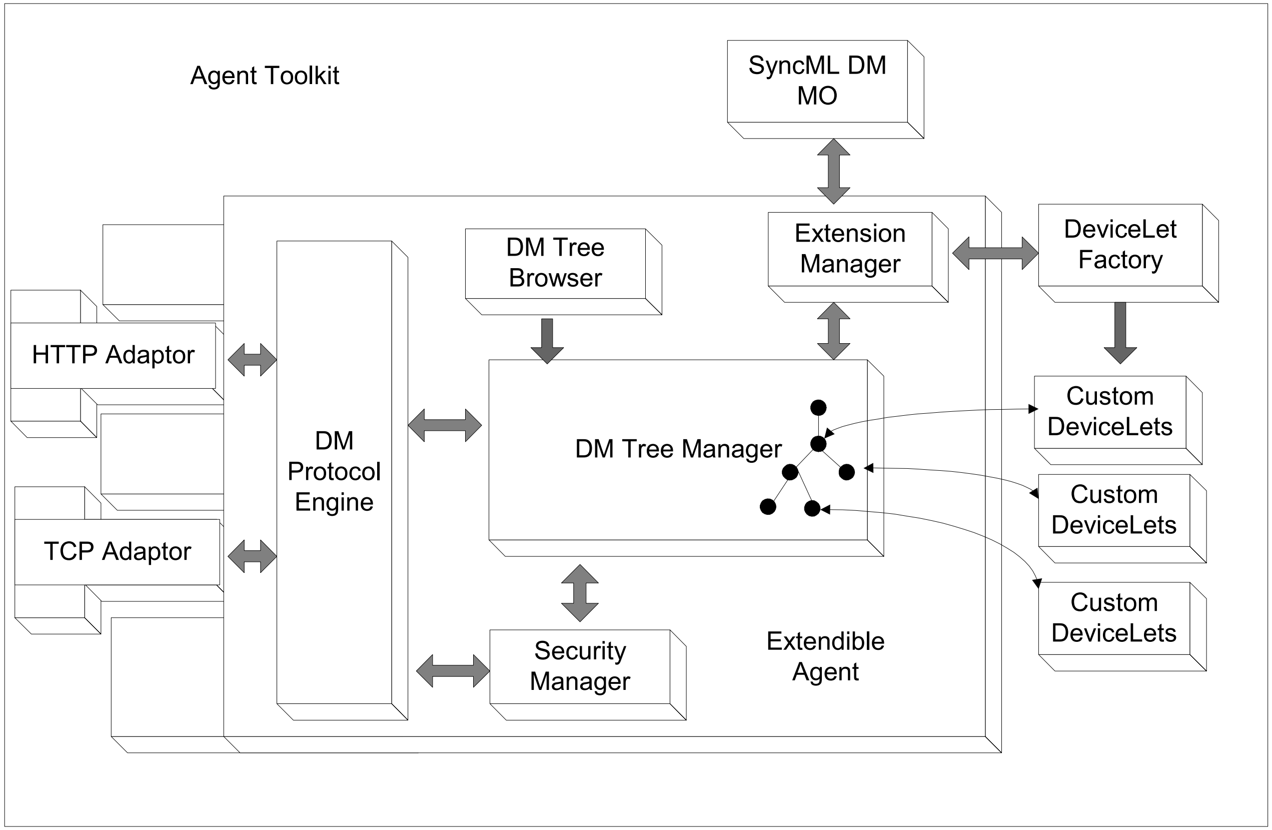


Example of a SyncML message

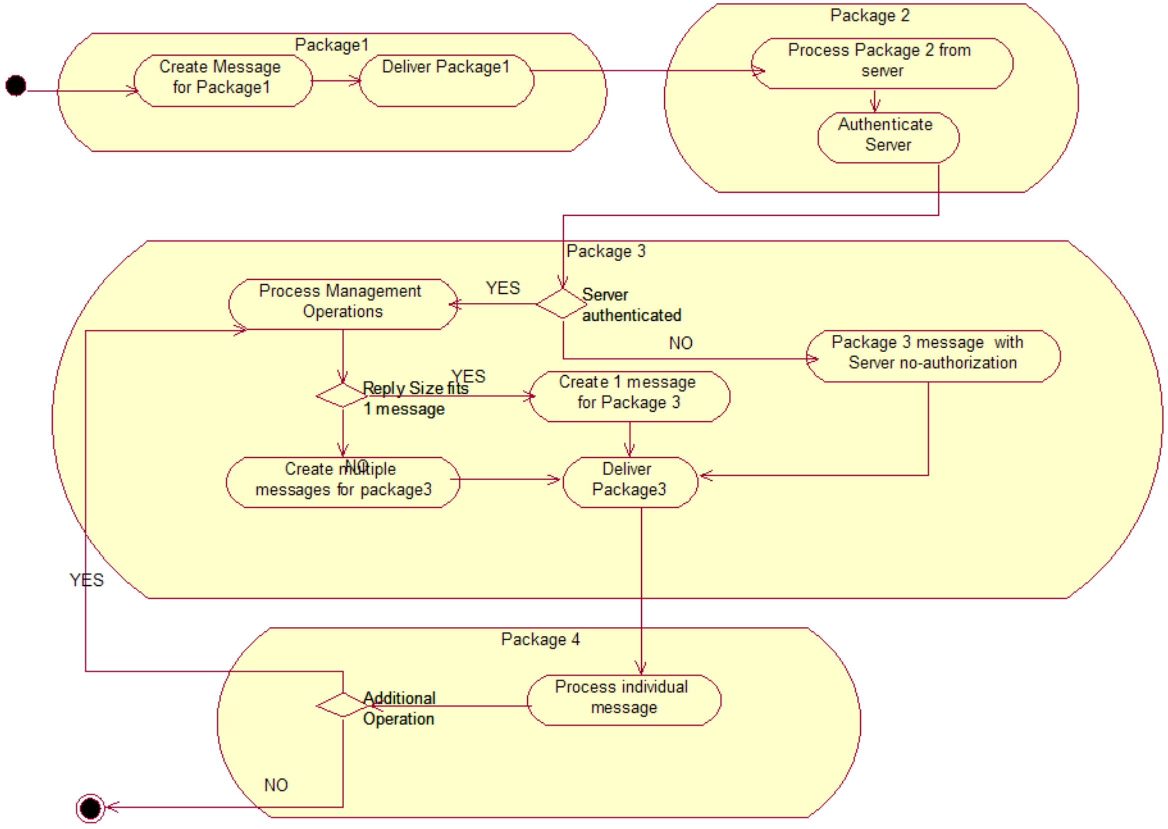
```
<SyncHdr>
……………….
</SyncHdr>
<SyncBody>
<Status>
<CmdID>5</CmdID>
<MsgRef>2</MsgRef>
<CmdRef>0</CmdRef>
<Cmd>SyncHdr</Cmd>
<Data>212</Data>
</Status>

<Results>
<CmdID>6</CmdID>
<CmdRef>1</CmdRef>
<MsgRef>2<MsgRef>
<Cmd>Get</Cmd>
<item>
<Source>..</Source>
<Data>Returned value</Data>
<Item>
</Results>
<Final/>
….
```

MAD-MAX Functional Architecture

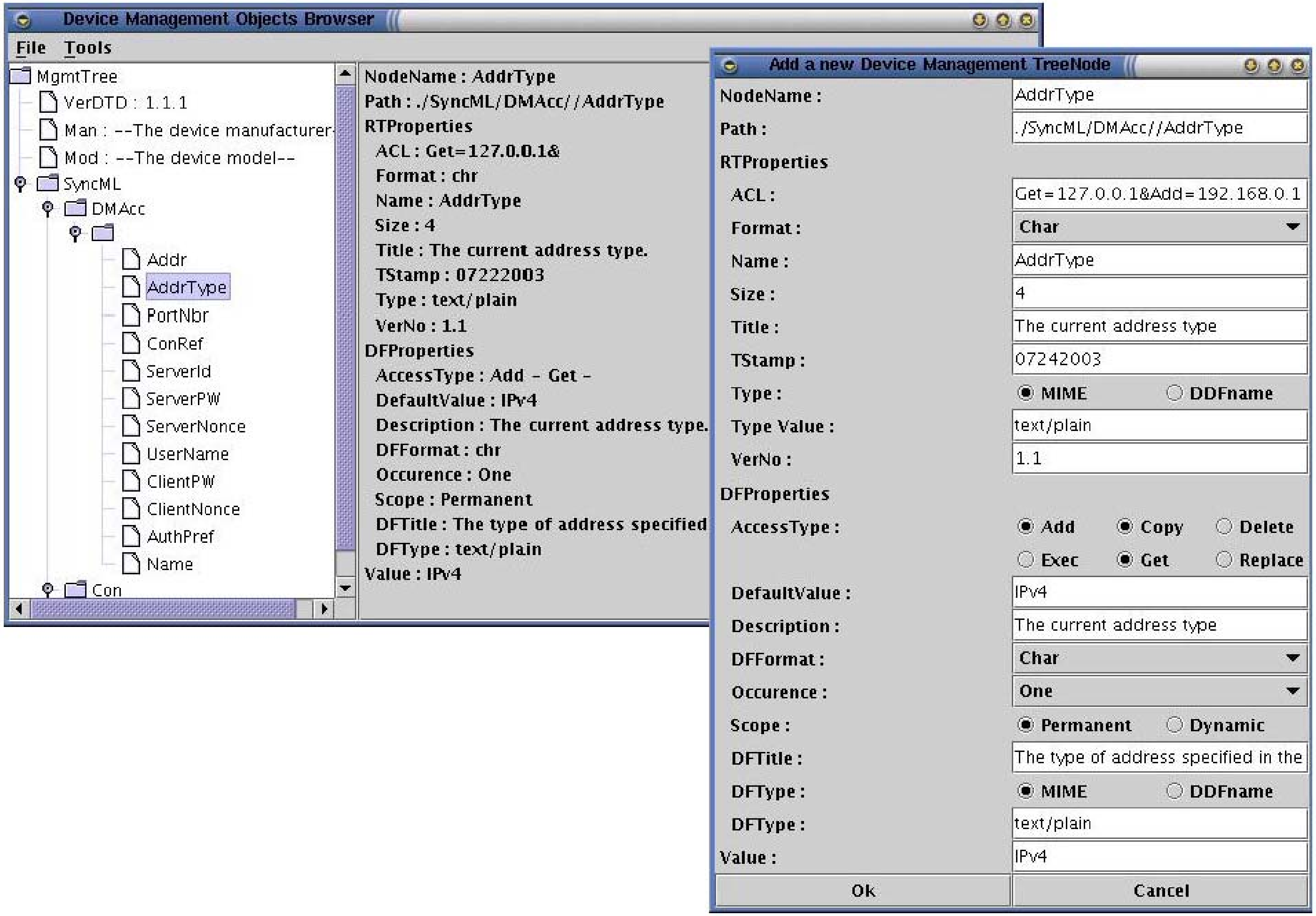


The Protocol Engine


Package1
Create Message for Package1
Deliver Package1
Package 2
Process Package 2 from server
Authenticate Server
Package 3
Process Management Operations
YES
Server authenticated
NO
Package 3 message with Server no-authorization
Reply Size fits 1 message
YES
Create 1 message for Package 3
Create multiple messages for package3
NO
Deliver Package3
YES
Package 4
Process individual message
Additional Operation
NO

Protocol state machine design and implementation

DM Tree Browser

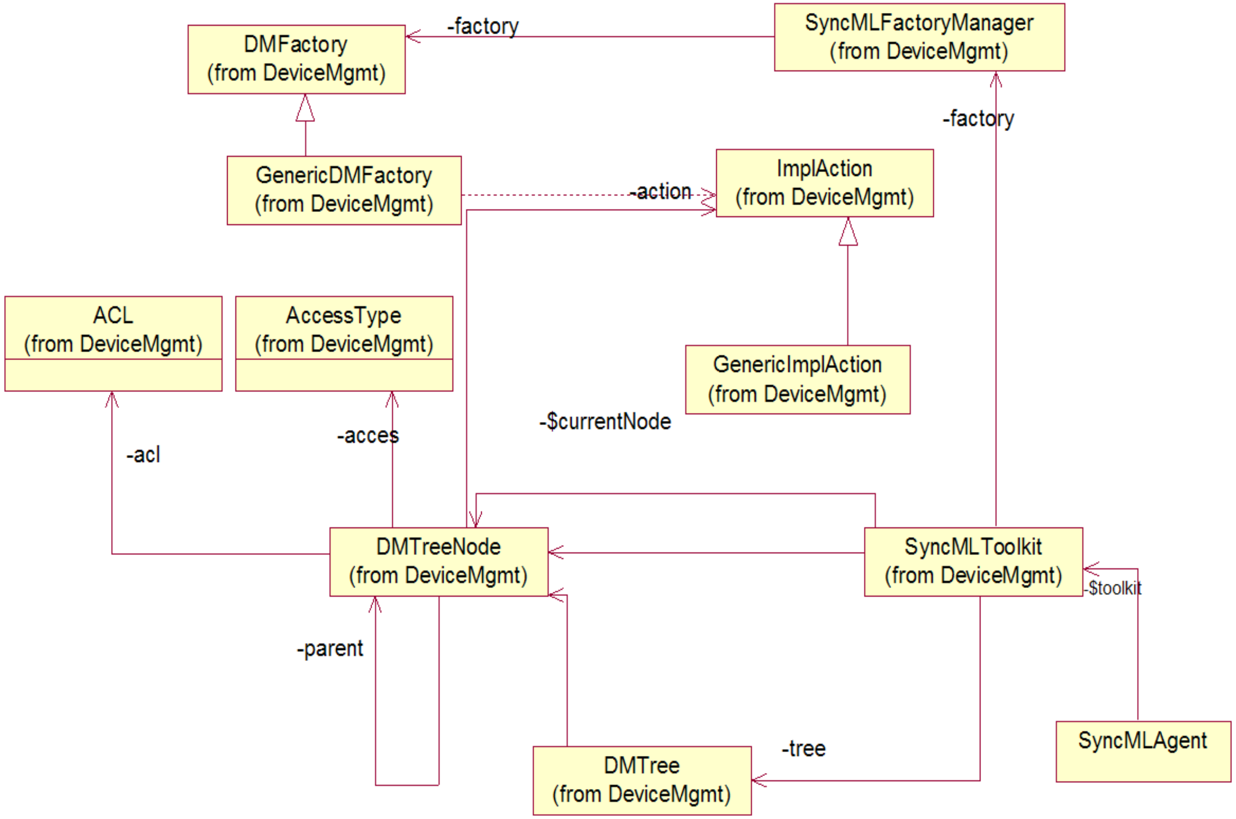

Device Management Objects Browser
File
Tools
Add a new Device Management TreeNode
Ok
Cancel

- MgmtTree
  - VerDTD : 1.1.1
  - Man : --The device manufacturer-
  - Mod : --The device model--
  - SyncML
    - DMAcc
      -
        - Addr
        - AddrType
        - PortNbr
        - ConRef
        - ServerId
        - ServerPW
        - ServerNonce
        - UserName
        - ClientPW
        - ClientNonce
        - AuthPref
        - Name
    - Con

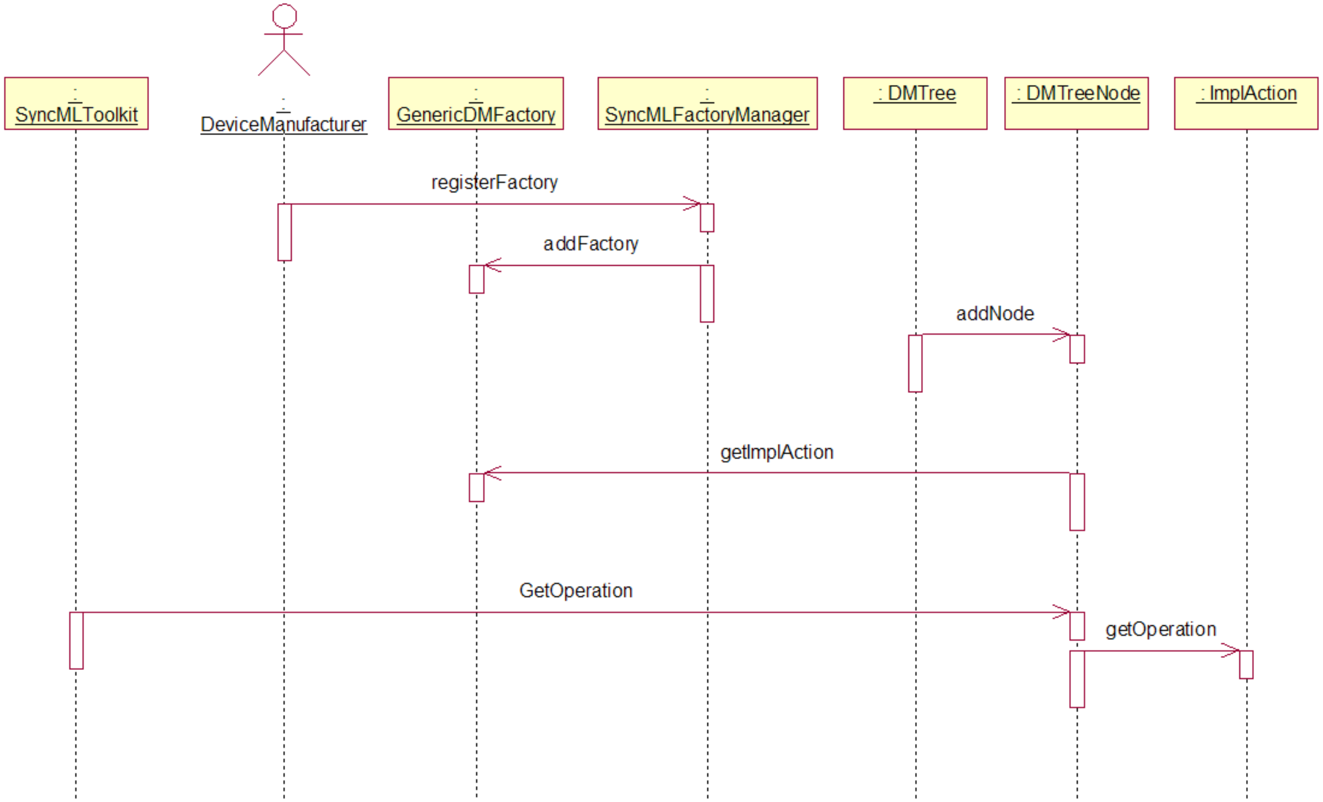
NodeName : AddrType
Path : ./SyncML/DMAcc//AddrType
RTProperties
ACL : Get=127.0.0.1&
Format : chr
Name : AddrType
Size : 4
Title : The current address type.
TStamp : 07222003
Type : text/plain
VerNo : 1.1
DFProperties
AccessType : Add - Get -
DefaultValue : IPv4
Description : The current address type.
DFFormat : chr
Occurence : One
Scope : Permanent
DFTitle : The type of address specified
DFType : text/plain
Value : IPv4

NodeName :	AddrType		
Path :	./SyncML/DMAcc//AddrType		
RTProperties			
ACL :	Get=127.0.0.1&Add=192.168.0.1		
Format :	Char		
Name :	AddrType		
Size :	4		
Title :	The current address type		
TStamp :	07242003		
Type :	● MIME	○ DDFname	
Type Value :	text/plain		
VerNo :	1.1		
DFProperties			
AccessType :	● Add	● Copy	○ Delete
	○ Exec	● Get	○ Replace
DefaultValue :	IPv4		
Description :	The current address type		
DFFormat :	Char		
Occurence :	One		
Scope :	● Permanent	○ Dynamic	
DFTitle :	The type of address specified in the		
DFType :	● MIME	○ DDFname	
DFType :	text/plain		
Value :	IPv4		

Extensibility

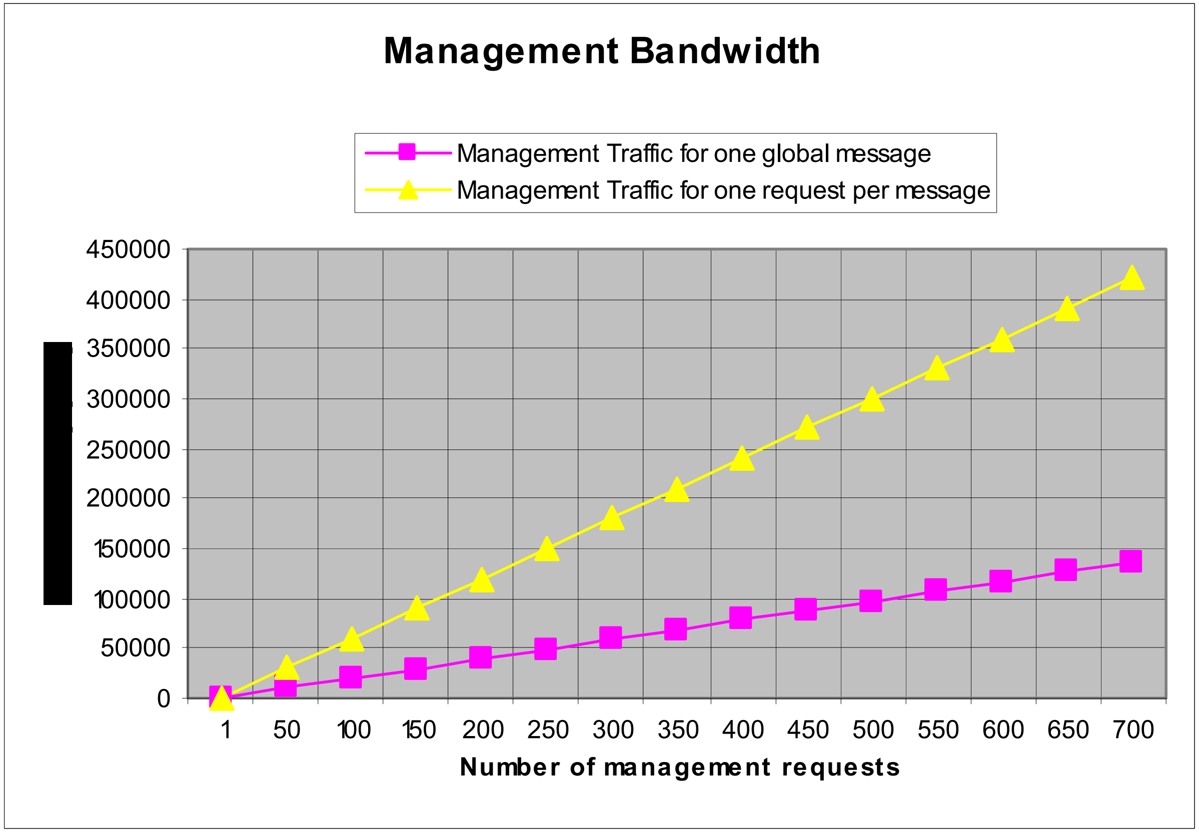


Extension Mechanisms



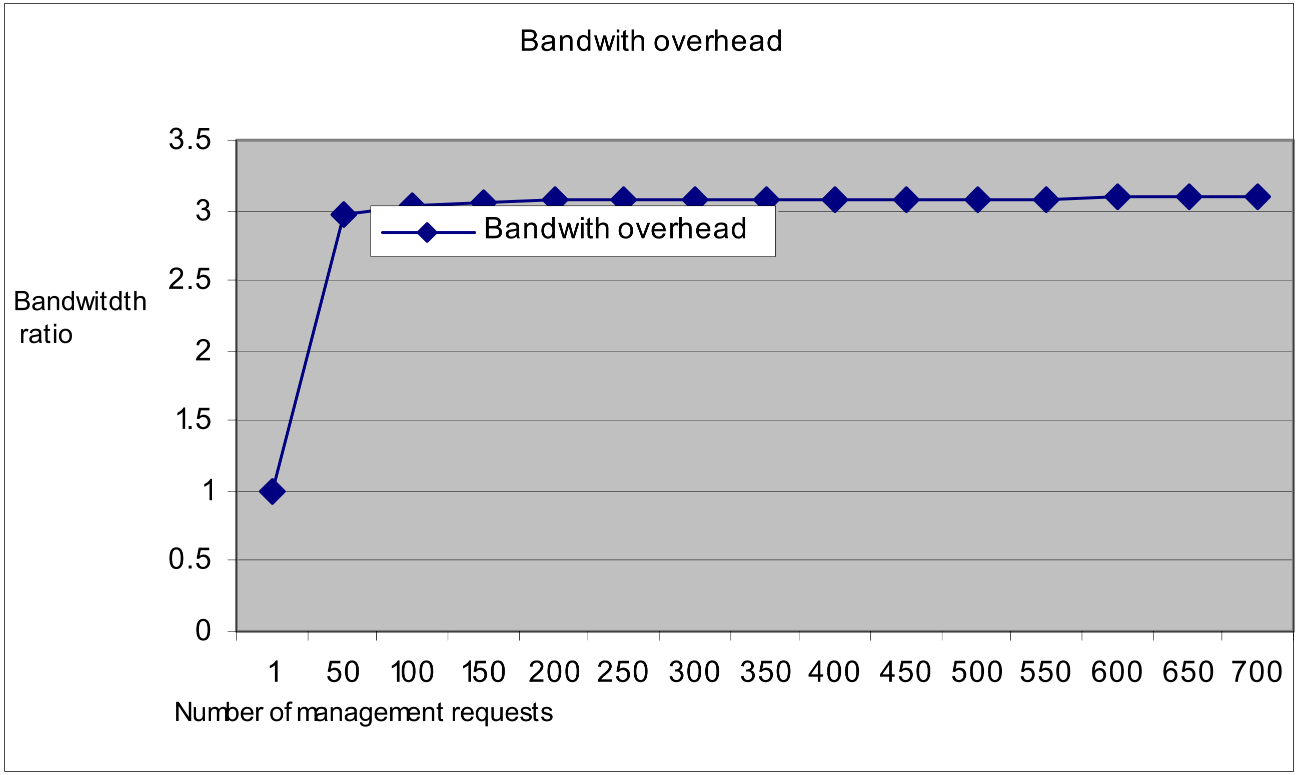
Benchmarks – Encoding strategies

One message with several requests versus several individual messages



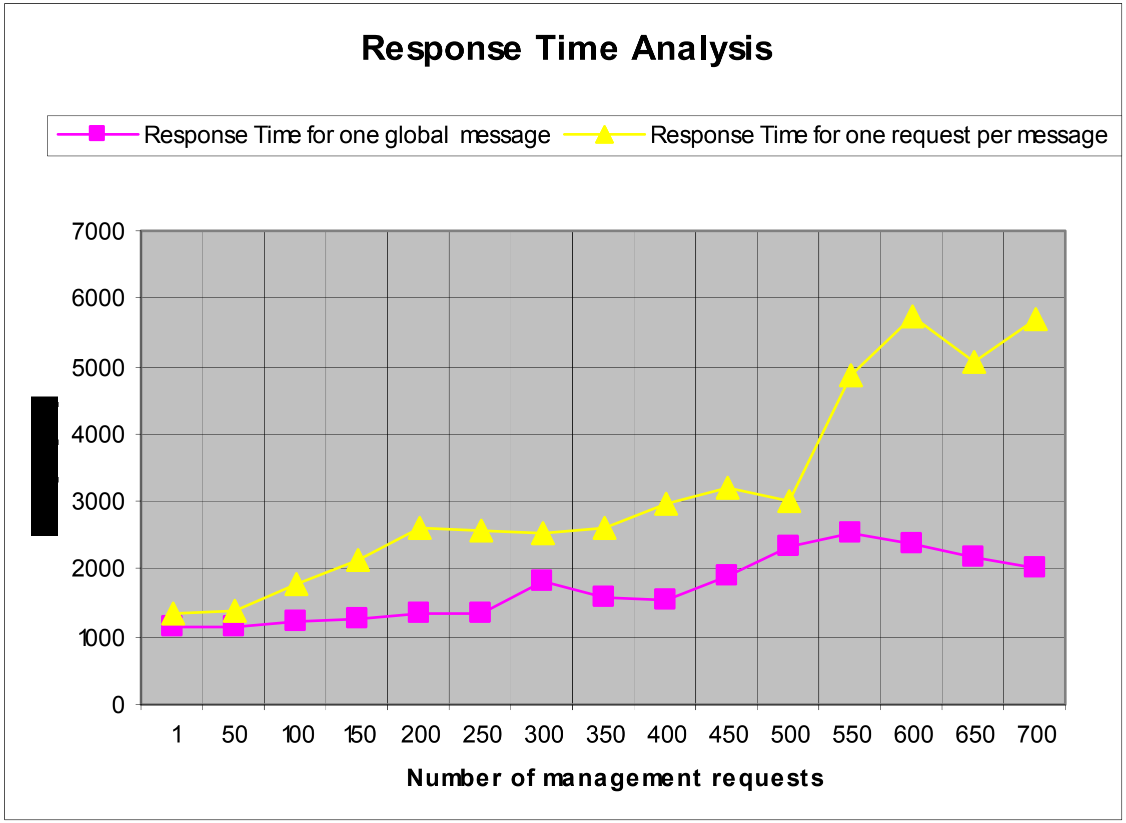
Bandwidth efficiency

Experimental efficiency assessment



Benchmarking the latency

One message with several requests versus several individual messages



Time is Money ☺

SyncML DM summary

- SyncML DM is more than just XML based configuration
  - Framework for describing management information
  - A set of standard Managed Objects
  - Transaction oriented Management Protocol
  - Network level transport agnostic
  - Flexible Access Control
- Adaptation of conceptual design approaches from the SNMP framework towards device mobility
  - Similar building blocks to the SMI, MIB2 and SNMP and Get/Set semantics
  - Reversed roles : a managed Device initiates a management session
  - Power for the users : allow/deny management actions
- What to expect
  - More OpenSource implementations
  - Interoperability studies
  - Performance analysis and large scale deployment tests