

SyncML

Varun Ranjit Singh
02/09/2004

Contents

- Introduction
- SyncML Basics
- SyncML: Device Synchronization
- SyncML: Device Management

□ SyncML – Synchronization Markup Language

- SyncML was formed as an industry initiative to develop and promote a single data synchronization protocol for all types of devices. Such as:-
 - PDAs,
 - Portable PCs,
 - Pagers,
 - Mobile phones.

- Founded in February 2000, SyncML quickly obtained over 500 supporting organizations with backing from major industry players. Such as:
 - Nokia, Ericcson, IBM, Lotus, Motorola, Palm, Psion, and Starfish software.
 - Early 2001, SyncML specification 1.0 was released.
 - SyncML 1.2 is the current specification.

□ SyncML Basics

- Works over Wired and Wireless Networks.
- Support for multiple transport protocols. (Obex, WSP, HTTP)
- Independent of actual data being synchronized. (vCard, vCalender, iCalender, email etc.)
- Optimized for small memory footprint devices.
- Independent of the OS.
- Standards based as such Interoperable.

□ SyncML: Device Synchronization (DS)

- Mobile phone (PDA) acts as the SyncML client, and a server acts as the SyncML server.
- Synchronization can be initiated by the Server or the Client.
- SyncML specifies seven different synchronization types,
 - Two-way sync
 - Slow sync
 - One-way sync from client only
 - Refresh sync from client only
 - One-way sync from server only
 - Refresh sync from server only
 - Server-alerted sync
- Data can be changed at either end so there might be conflicts, the conflicts are resolved by data-merge, duplication, client-win.
- SyncML, comprises of 2 parts, Representation Protocol and Synchronization Protocol.

□ SyncML DS: Representation Protocol

- The Syntax, called the *Representation Protocol* (RepPro)
- SyncML operations are conceptually bound to SyncML packages, which are a collection of SyncML messages
- A pkg. has a SyncHdr and a SyncBody
 - SyncHdr has the routing, identification etc.
 - SyncBody is a container for multiple Sync Commands.
- Defines 3 DTD's:
 - SyncML DTD, defines the elements of SyncML commands.
 - Meta DTD, defines the format of SyncML data.
 - Device DTD, defines how Client and Server should exchange their information.

□ SyncML DS: Synchronization Protocol

- The Semantics, called the *Synchronization Protocol* (SyncPro)
- Initialization Phase (kind of a Handshake protocol)
 - First, the client and server exchange device and service capabilities.
 - The client conveys its device capabilities in the first package and the server conveys it's in the second package (only conveying the subset of the clients capabilities).
 - The capability exchange is done to convey database information, clients memory details, the supported sync type, etc.
 - They authenticate using Basic / MD5 authentication (auth) schemes
 - If the credentials (cred) are wrong, a challenge is issued with the scheme and MIME type. If the cred are wrong again, session is terminated.
 - The client can challenge (chal) the server.
 -

contd...

□ SyncML DS: Synchronization Protocol

- Sync Anchors are also exchanged (last and current anchors as alerts).
- Pkg.#1 - The client sends this initialization package to the server with the credentials, device information and the alerts to the databases it wants to sync with. (Only Get, Put and Results commands)
- Pkg.#2 - The server sends a package containing the status for the all the commands sent by the client and its service capability, and the alerts for the databases the need to be synced.

SyncML DS: Synchronization Protocol

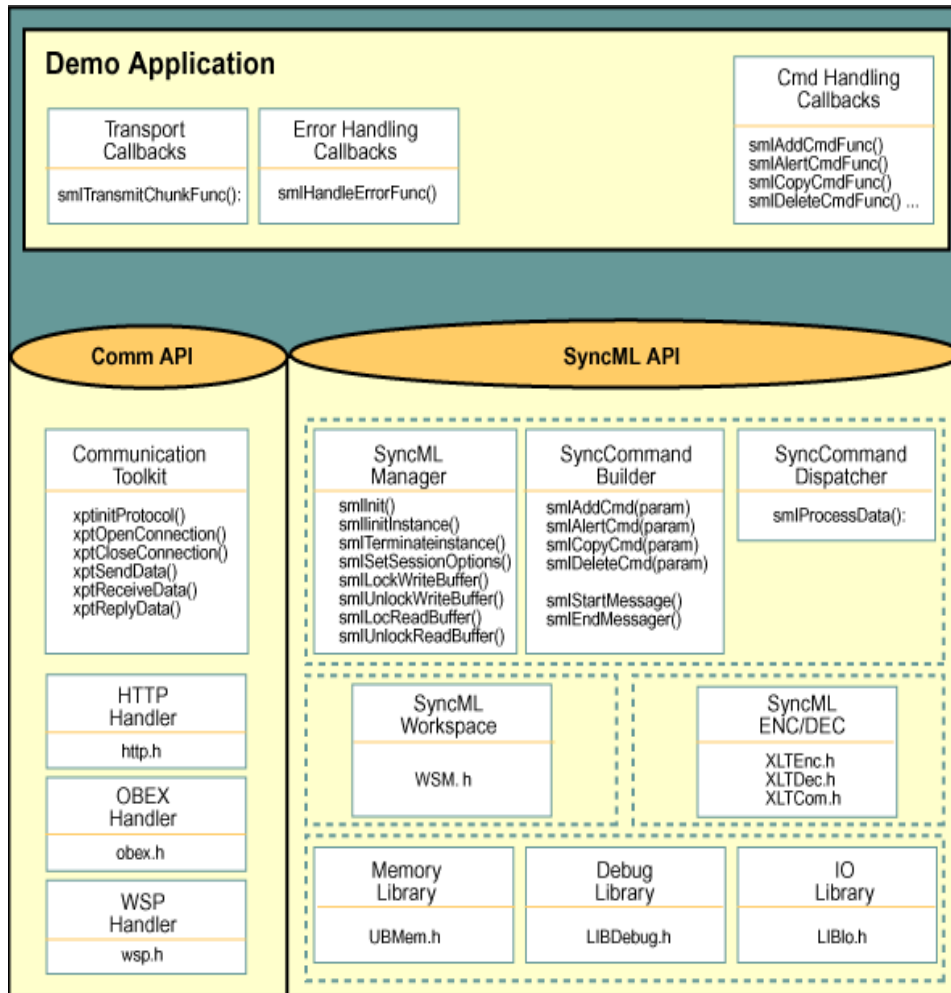
contd...

- Synchronization Phase (The DB sync commands are sent now)
- Depending on the type of the Sync session, all data or only the modified data is sent.
- Pkg. #3 - The client sends the modifications of each database enclosed in separate SYNC Command which contains information about the Database URI's [and the credentials required to access the database (Optional)]. Any modifications to the client's database after sending this Pkg. should not be included in the subsequent Packages but should be reported in the next session. The modifications in the databases is sent using commands like 'Add', 'Delete' and 'Replace'.
- Pkg. #4 - This package contains the status for commands the client sent in Pkg. #3. The server modifications are also sent in this package.

□ SyncML DS: SyncPro contd...

- Mapping Phase
 - If there are no mapping operations the sync session can be terminated after Pkg. #4.
 - The client and the server use their own, independent unique ids for each data item. These ID's maynot be the same (LUID – Client and GUID – Server)
 - The server must store the client's ids and maintain a mapping table indicating the map between its id
 - Pkg #5 - The client sends this package for responding with the status for the server's sync package and to send 'Map' commands to the server if any are required.
 - Pkg. #6 - This server sends this package to acknowledge that it has received the Maps from the clients and send the status about the execution of the Map commands. This Package should be sent even if the server did not receive the Maps in the package #5 to end the session.

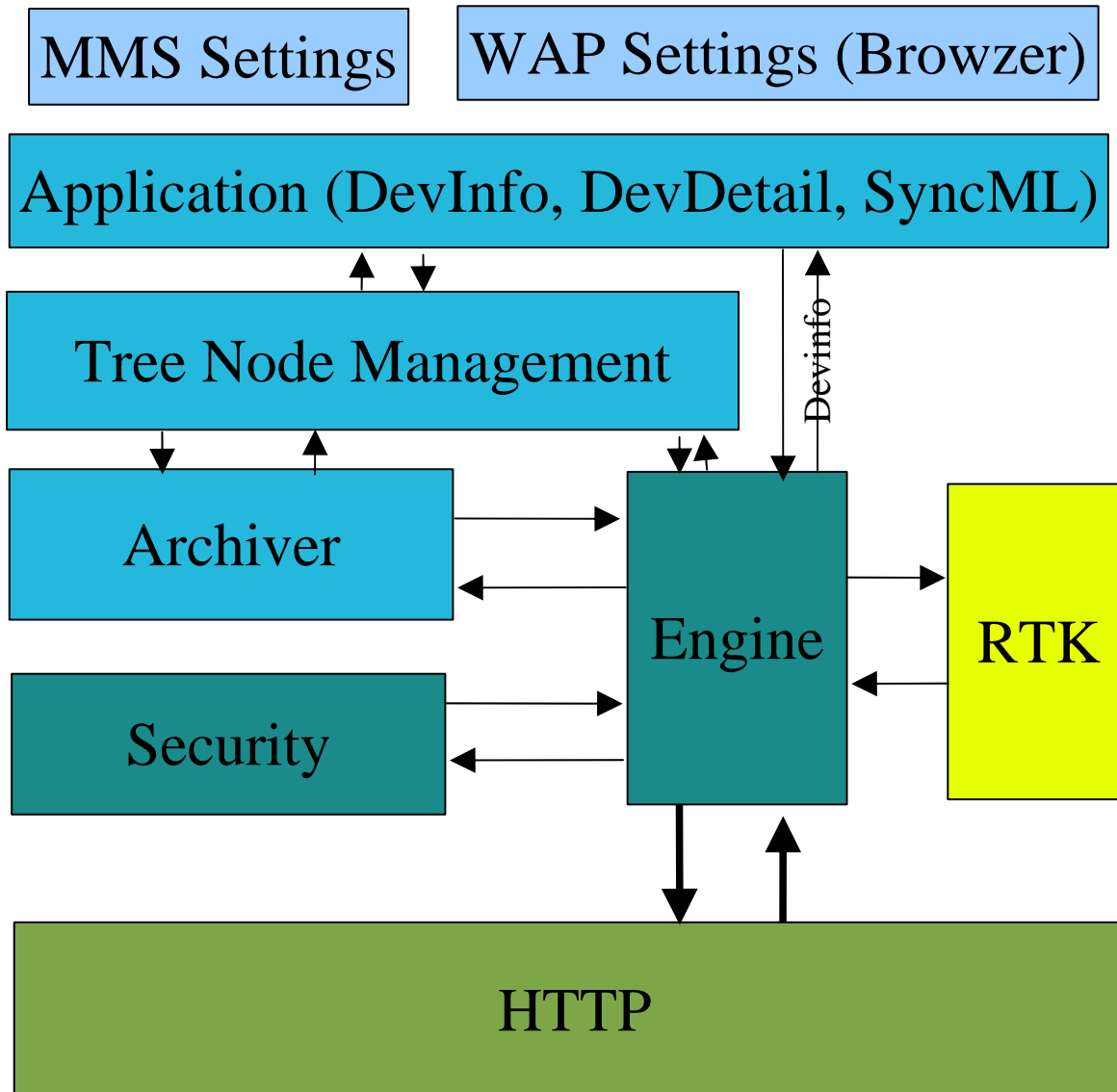
SyncML Reference ToolKit (RTK)



□ SyncML Device Management

- SyncML DM allows provisioning and remotely management of device.
- Representation Protocol is the same except for exclusion of some DS SyncML tags such as 'Put', 'Map' etc
- Synchronization Protocol is replaced by a DM protocol which is similar to the SyncPro except it has 2 instead of 3 phases.
- Security has HMAC as an added auth scheme.

DM Architecture



□ Contribution to DM

- SyncML DM Client
 - DM Tree Archiver
 - DM Session Logs/Recovery
 - DM Client Engine (Testing Only) – using RTK

- SyncML DM Server (Test Tool)
 - DM Server Engine (using RTK)

□ References

- SyncML SyncPro 1.1.1
- IBM Developerworks
 - www-106.ibm.com/developerworks/wireless/library/wi-syncml2/
 - www-106.ibm.com/developerworks/library/wi-syncml/

