FRASCATI, MAGGIO 2007
REV. 1

ARCHITETTURA E MECCANICHE PER L'ACCESSO DELLE PIATTAFORME WINDOWS AI SERVIZI DI DIRECTORY BASATI SUL PROTOCOLLO LDAP

VALUTAZIONI DI FATTIBILITA' PER L'ACCESSO LDAP SERVITO DA INFRASTRUTTURE NON WINDOWS.

Nunzio Amanzi – INFN Laboratori Nazionali di Frascati

L’architettura del Sistema Operativo Windows (con particolare riguardo alle versioni 2000 e succ.) può essere schematizzata secondo i seguenti quattro livelli di astrazione:

1) Kernel e moduli di basso livello, caratterizzati tra l’altro, dal sottosistema GDI per la gestione dell’output *gui oriented*, dai driver di periferica e dalle dll che esportano le API e i tipi canonici di sistema;
2) Oggetti COM, che costituiscono l’infrastruttura di interfaccia all’O.S. poiche’ offrono la connettività ai servizi, a basso ed alto livello, e riesportano le API. L’engine che li contraddistingue e’ duale:
   - oggetti *in-process*: eseguiti ed istanziati nell’ambito dello stack e/o contesto di processo della stessa applicazione client;
   - oggetti *out-process*: istanziati nell’ambito dell’applicazione client, i cui metodi sono eseguiti nell’ambito di un contesto di processo esterno che assume il ruolo di server.

   La coesistenza di *in-process* e *out-process* definisce lo scenario client-server peculiare di Windows in termini di comunicazione ed interoperabilità tra i processi. Il protocollo che sottende a tale comunicazione e’ l’OLE (‘Object Linking and Embedding’) che e’ utilizzato da moduli di interfaccia per implementare connessioni RPC, quando l’applicazione server e’ eseguita su un host distinto da quello che esegue l’applicazione client o, viceversa, connessioni LPC quando client e server sono eseguiti nell’ambito nel medesimo host. In sostanza gli oggetti COM svolgono il ruolo di interfaccia RPC/LPC di alto livello, essendo gli stessi definiti mediante specifiche dll che hanno funzione di provider di accesso;
3) Componenti e moduli di libreria, che costituiscono un framework di sviluppo per le applicazioni. Tale infrastruttura, nota con il nome *MS .Net Framework*, riesporta gli oggetti di livello 2 e definisce nuove gerarchie di classi nell’ambito di un globale spazio di nomi. Fornita inizialmente come componente di update per le versioni client di windows (2000/XP), e’ attualmente distribuita come parte integrante le versioni 2003 Srv R2 rappresentando un nuovo standard di riferimento, in fase di evoluzione e consolidamento, per le applicazioni GUI e Web based;
4) Moduli di alto livello, tra i quali vanno annoverati le *Management Console* della piattaforma, l’interfaccia utente e le applicazioni. I moduli di alto livello accedono alle risorse e ai servizi locali/di rete interagendo con il S.O. mediante le API, gli oggetti COM a layer 2, che attualmente costituisce lo scenario funzionale più diffuso/opportuno, e/o i componenti serviti dal *.Net Framework.*

Per l’accesso ai servizi di directory, in particolare per il servizio su protocollo LDAP, il S.O. Windows fornisce specifiche infrastrutture di interfaccia denominate *Directory Services Provider*.
In generale un provider di accesso, che e’ caratterizzato da uno spazio di nomi che definisce una gerarchia di oggetti di livello 2, e’ esportato da librerie a link dinamico (dll) e fornisce strumenti di connessione a servizi locali e/o di rete eseguendo il *relay* delle richieste su protocolli distinti tra il *front-end* e il *back-end*: per esempio il *Provider WMI*, che esporta gli oggetti per il management di una piattaforma windows, consente l’inoltro delle richieste e la connettivita’ su protocolli RPC/Netbios e SNMP.
Il S.O. Windows esporta i seguenti 4 provider di accesso ai servizi di directory: *ADSI WINNT, ADSI IIS, ODBC, LDAP*.

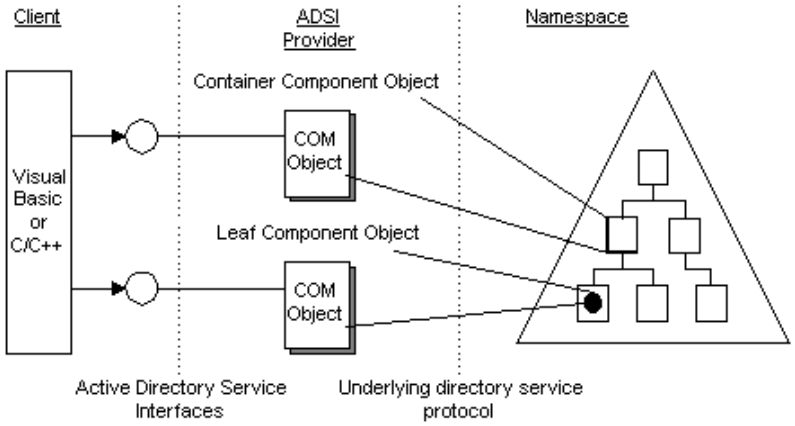

Fig. 1: Interfacciamento allo spazio dei nomi di AD tramite provider

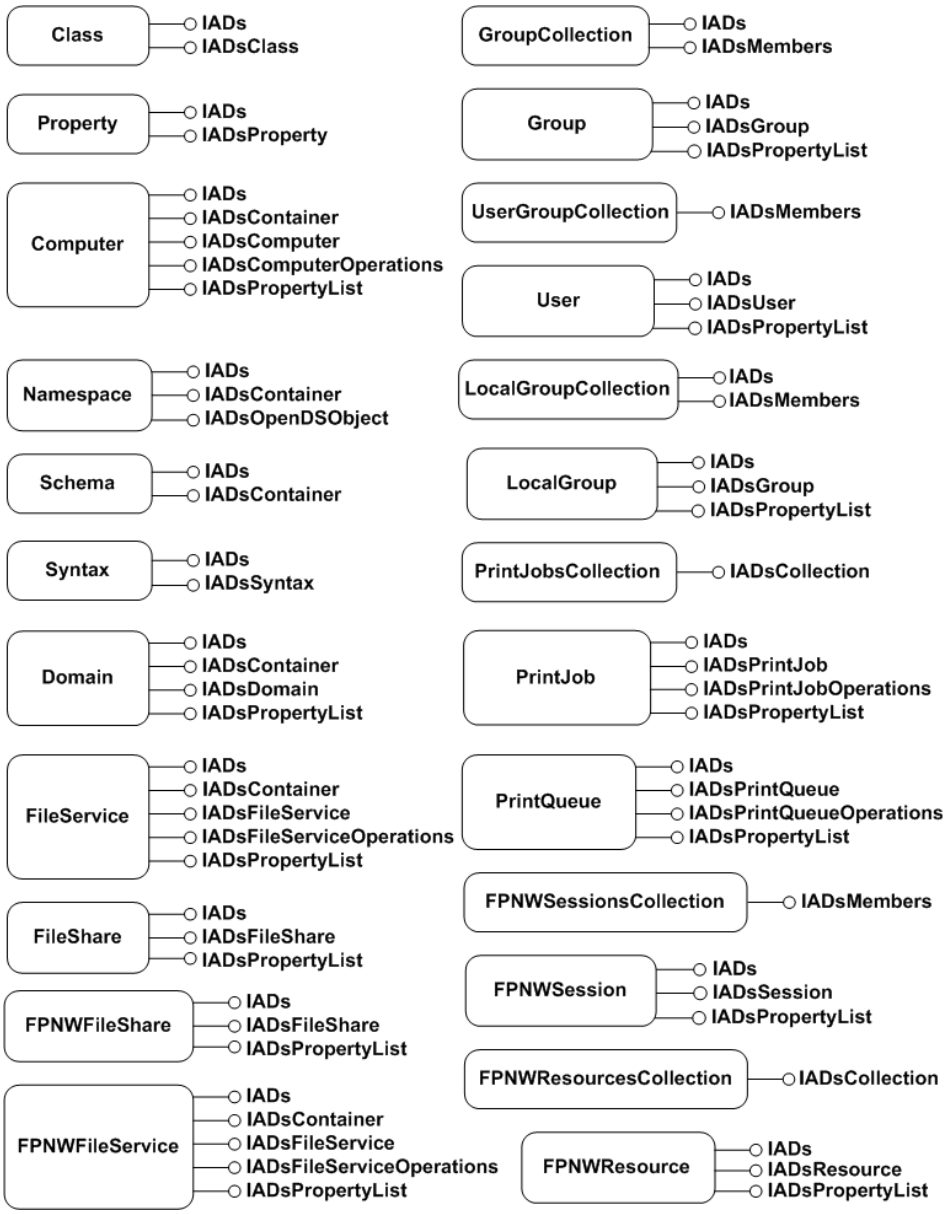

Fig. 2: Schema e interfacce relative al provider ADSI WINNT

Fondamentalmente i suddetti provider servono classi che ad alto livello riesportano tra l'altro gli oggetti di *Active Directory* implementando il polimorfismo mediante interfacce funzionali che possono essere condivise da differenti classi definite nello stesso provider o in provider distinti (figg. 1 e 2).
I provider *WINNT*, *IIS* e *ODBC* costituiscono strumenti specifici per l'accesso ai servizi di directory windows poiche' definiscono/esportano esclusivamente l'*AD Schema*: in tal senso il provider *LDAP* appare il piu' idoneo per le valutazioni sulle meccaniche e i presupposti di accesso da piattaforme windows ai servizi di directory su protocollo LDAP esportati da piattaforme non windows.

Per accedere al servizio LDAP di *AD* le applicazioni windows basate sul provider *LDAP*, possono istanziare/utilizzare direttamente le sue classi o referenziarle attraverso i *direcory services components* del framerwork *.NET*.

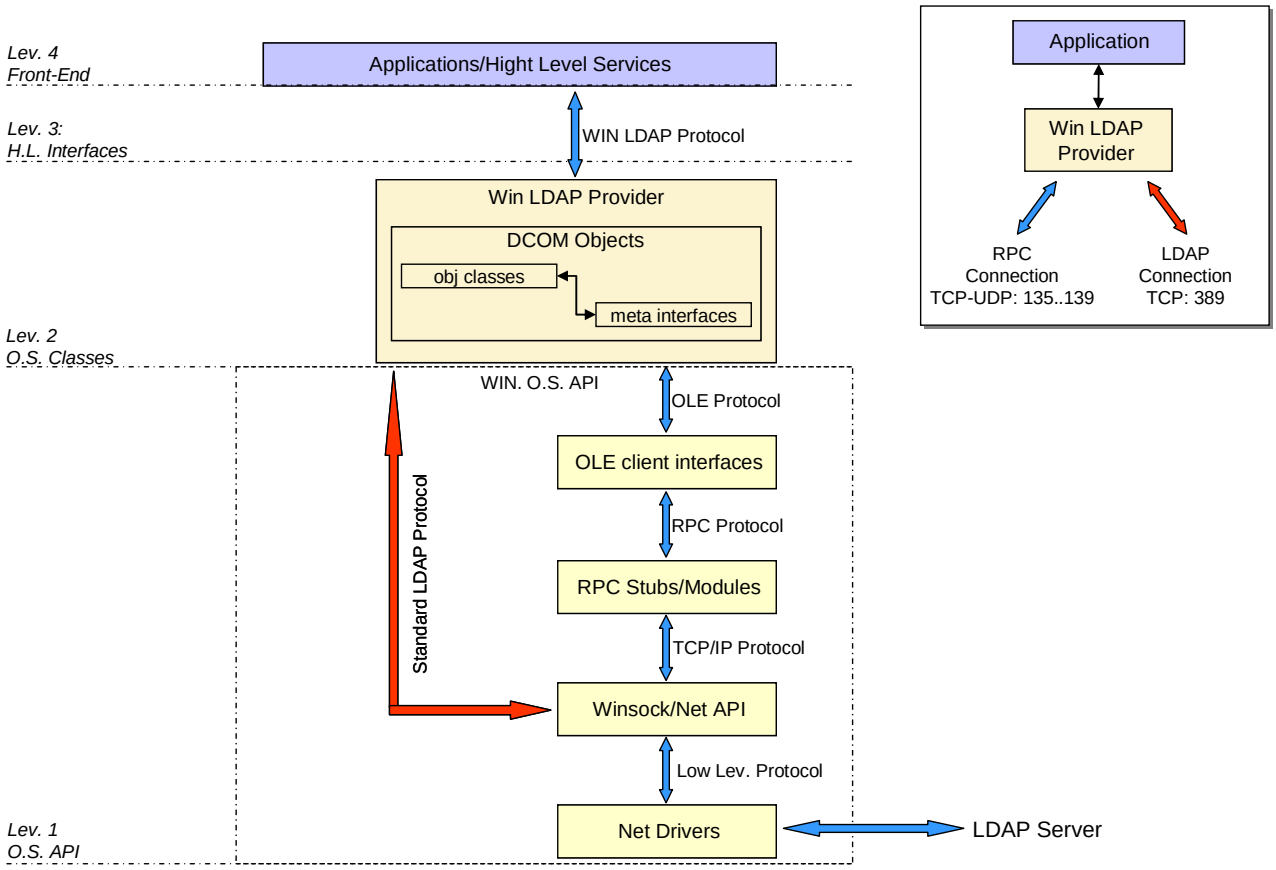

Fig. 3: Schema di accesso LDAP mediante provider

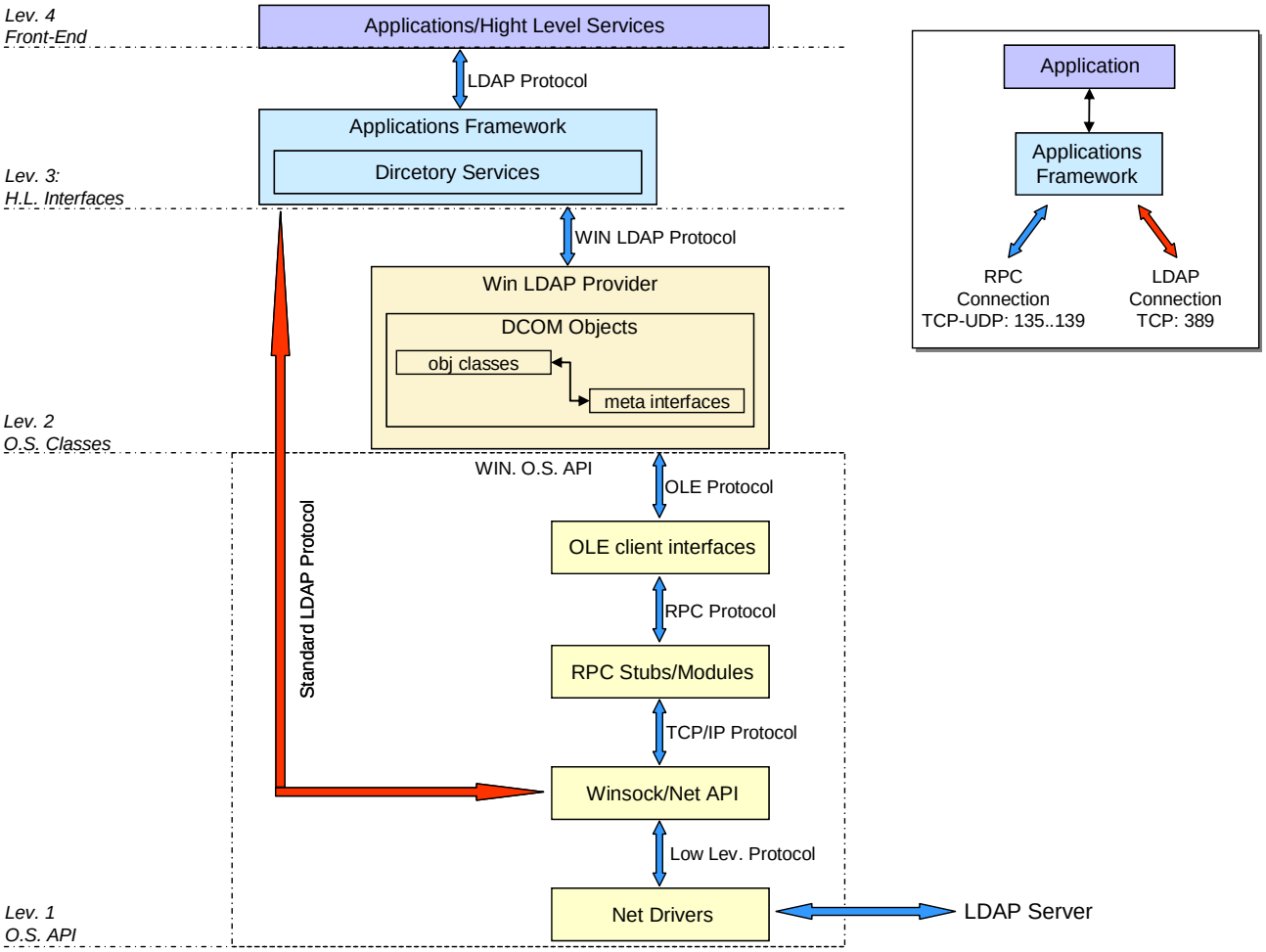

Fig. 4: Schema di accesso LDAP tramite framework di alto livello

Nelle figg. 3 e 4 e' illustrato il processo di inoltro delle richieste (mediante le frecce azzurre) e il relativo flusso dei protocolli utilizzati per stabilire la connessione con il server. In questo scenario l'applicazione, che generalmente istanzia oggetti dello schema di *AD*, interagisce su protocollo WIN LDAP con i moduli di interfaccia a livello 2 e 3 che a loro volta instaurano connessioni richiedendo servizi LPC, RPC, Netbios.

Affinche' l'infrastruttura client di cui alle figg. 3 e 4 possa essere compliant per l'accesso ai servizi LDAP non windows e' necessario il positivo riscontro ai test eseguiti sui seguenti aspetti salienti:

- individualizione ed utilizzo di applicazioni basate su moduli di interfaccia ai livelli 2 e 3, come il provider *LDAP*, integrati nell'O.S.;
- supporto del protocollo LDAP standard per il dialogo tra l'applicazione e i moduli di interfaccia e tra i detti moduli e il server, in particolare:
  - localizzazone del/dei server autoritativo(i) per un determinato albero LDAP non necessariamente mediante l'utilizzo del sistema DNS;
  - supporto dell'autenticazione LDAP;
  - visita del tree LDAP e accesso agli attributi degli oggetti referenziati mediante path/DN basato sui *domain components* e sul *sistema geografico*.

All'uopo i test dovrebbero essere condotti verificando il supporto del protocollo LDAP standard sia da parte del provider a livello 2 (frecce rosse fig. 3) sia a carico del framework di livello 3 (frecce rosse fig. 4). Nell'ottica delle valutazioni di compatibilita' risulterebbe inoltre prioritario/preferibile riscontrare l'idoneita' del provider di livello 2 rispetto al framework perche':

- e’ un componente nativo ed integrato nel S.O. Windows;
- garantisce attualmente maggiore compatibilita’ e portabilita’ anche perche’ le applicazioni basate su framework *.NET* non sono ancora molto diffuse;
- le relative classi di interfaccia possono essere istanziate anche nell’ambito di script Web/Gui/CmdLine poiche’ non richiedono a priori ne’ l’esecuzione di include e/o link statici dei moduli che le definiscono ne’ la compilazione del codice.

In tal senso le attivita’ di riscontro dovrebbe essere eseguite prima sul provider di livello 2 e poi sul framework di livello 3 secondo il seguente workflow:

- creazione di oppurtuni moduli stub, utilizzando le tecniche relative allo scripting e/o agli eseguibili, per il mounting delle classi di interfaccia;
- verifica della consistenza e della funzionalita’ nell’accesso al tree LDAP di *AD* mediante i *domain components*;
- verifica dell’accesso, su autenticazione LDAP, verso un tree LDAP, non servito da windows, basato sui *domain components*;
- verifica dell’accesso, su autenticazione LDAP, verso un tree LDAP, non servito da windows, basato sul *sistema geografico*.

Qualora i test sui moduli di interfaccia non forniscano l’adeguato riscontro sugli aspetti di compatibilita’ sopra elencati, l’indagine dovrebbe essere infine condotta a livello 4 (applicazione) valutando caso per caso la sussistenza dei suddetti presupposti in moduli e applicazioni di terze parti (fig. 4).

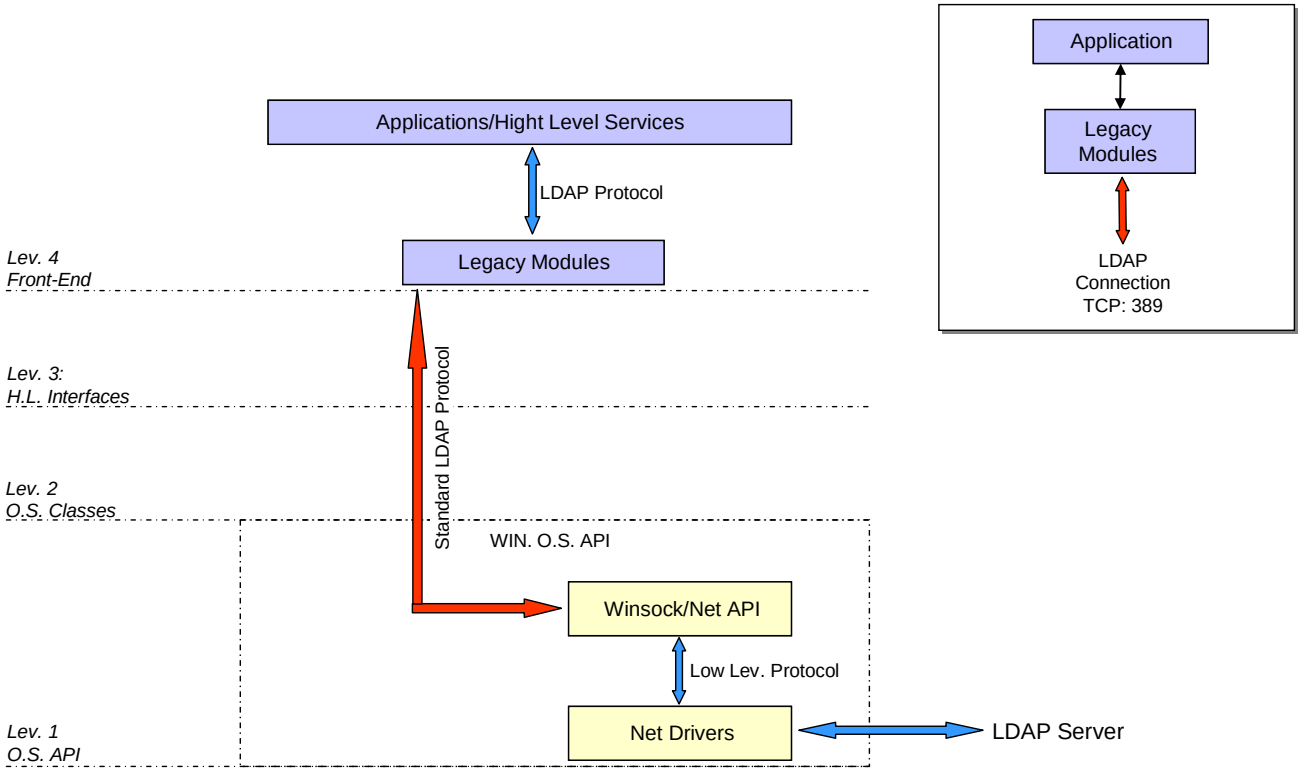

Fig. 4: Schema di accesso LDAP Standard a livello applicazione

Digitally signed by Nunzio Amanzi
DN: cn=Nunzio Amanzi, c=IT, l=LNF, o=INFN, ou=Personal Certificate, email=Nunzio.Amanzi@lnf.infn.it
Reason: I am the author of this document
Date: 2007.05.29 13:01:41 +02'00'