Università degli Studi di Bologna
Scuola di Ingegneria

Corso di
Reti di Calcolatori T

SISTEMI DI NOMI

Antonio Corradi

Anno accademico 2023/2024

SISTEMI DI NOMI

Spesso nei sistemi distribuiti siamo in presenza di molti sistemi di nomi anche molto diversi e che hanno proprietà variabili

Proprietà di base dei Sistemi di NOMI

Generalità

- varietà dei nomi disponibili e trattati

Definizioni multiple della stessa entità

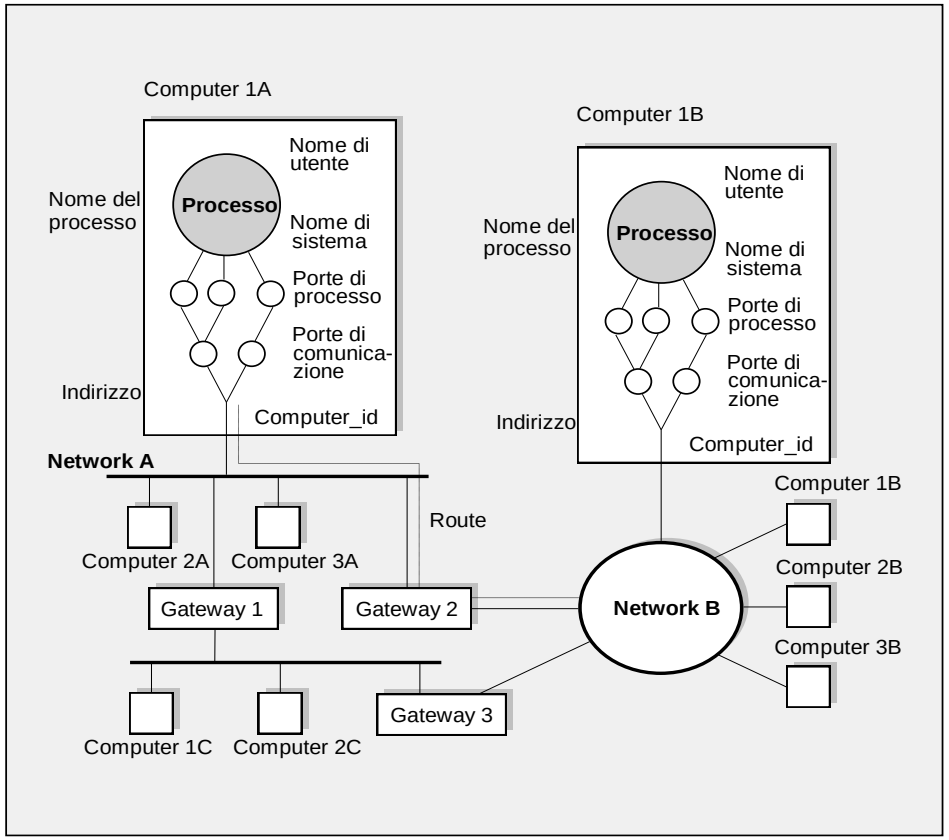
- varietà di nomi per lo stesso oggetto con mapping capace di traslare tra questi

Distribuibilità

- uso di direttori partizionati e/o replicati

User-friendliness

- nomi facili per l'utente



NOMI

Problema fondamentale nel distribuito la necessità di ritrovare (cioè identificare) le altre entità nel sistema

- Complessità del problema e difficoltà di soluzioni generali

Entità diverse eterogenee ⇨ livelli diversi di nomi

- Più sistemi di naming e più livelli di nomi nel sistema

con contesti di visibilità

- più funzioni di trasformazione da nome all'entità

I NOMI possono essere identificatori come

- **stringa di caratteri - nomi esterni (nomi di utente)**
- **numero binario - nomi interni (nomi di sistema)**

Le entità di un programma sono oggetti e spesso sono associati a diversi sistemi di nomi

sia nomi di utente (significativi per l'utilizzatore)

sia nomi di sistema (meno leggibili ma più efficienti)

LIVELLI DI NOMI

Spesso si possono considerare alcuni livelli di nomi per il distribuito NOME in tre possibili livelli (Shock)

- **Nome** **LOGICO esterno**
  specifica quale oggetto (entità) si riferisce e denota la entità
- **Indirizzo** **FISICO**
  indirizzo specifica dove l'oggetto risiede e lo riferisce dopo un binding
- **Route** **Organizzazione per la raggiungibilità**
  specifica come raggiungere l'oggetto

Funzioni di corrispondenza o MAPPING per passare da una forma ad un'altra e per aiutare l'utente finale

- mapping nomi ➔ indirizzi
- mapping indirizzi ➔ route

I nomi sono scelti dall'utente, gli indirizzi assegnati dal sistema

l'utente deve specificare nomi e ritrovare route

GLOBALITÀ via LOCALITÀ

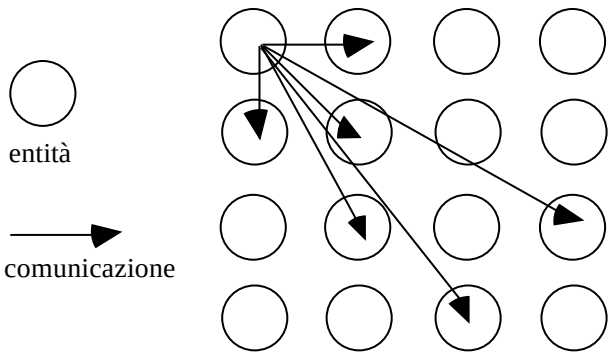
Per la **scalabilità** è importante avere coordinamento tra **diverse località contigue**

concetto di località (limiti alla comunicazione)
vs. globalità (nessun vincolo)

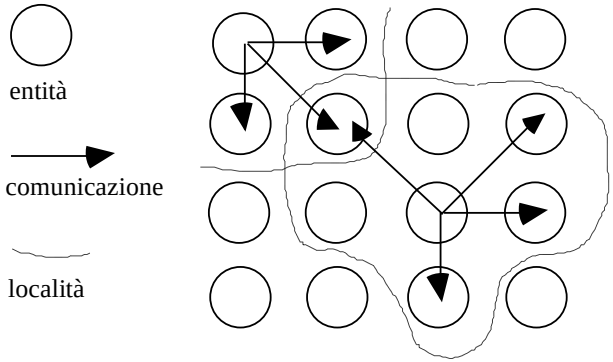
modelli **globali** non impongono restrizioni alle interazioni ⇨ **operazioni non scalabili dipendenti dal diametro del sistema**

modelli **locali** (o **ristretti**) prevedono limiti alla interazione ⇨ operazioni scalabili poco dipendenti dal diametro del sistema

Modelli globali



Modelli locali



Si va verso la località (con vincoli) per ottenere scalabilità routing

SISTEMI DI NOMI

SPAZI dei NOMI più usati

piatto (flat)

- con nessuna struttura, ma adatto per pochi utenti e poche entità

partizionato

- gerarchia e contesti (DNS), ad esempio, disi33.disi.unibo.it

descrittivo

- con riferimento ad una struttura di oggetto caratterizzato da attributi per identificare la entità corrispondente (OSI X.500)
  - username e password
  - attributi con liste di valori (rigidi o meno)

In questi scenari, spesso ci possono essere problemi nell'identificare nomi di gruppo

- un **nome di gruppo** identifica una lista di nomi di entità
- Si noti che il problema è simile a quello di un multicast vs. comunicazione punto a punto

SCELTE SUI NOMI

Piatto (flat) **antonio, acorradi**

- Chi distribuisce i nomi? Problemi nel caso di molti servitori di nome

Partizionato **disi33.cineca.it** **disi33.disi.unibo.it**

- Ogni responsabile di partizione mantiene e distribuisce i nomi

Descrittivo **nome=Antonio & organizzazione=UniBologna**

- Ogni nome può identificare anche una molteplicità di entità che si possono trovare solo con una ricerca esaustiva sul sistema di nomi

Nomi di gruppo **IP classe D**

- Ogni gruppo individua un insieme di entità denotate dal nome stesso anche molto lontane, poco correlate, e spesso non gestite dallo stesso servitore di nomi
- Necessità di una infrastruttura di supporto al gruppo

COMPONENTI DI UN SISTEMA DI NOMI

In un servizio di nomi, consultato implicitamente o esplicitamente, i clienti del name server sono

- i **clienti** che devono risolvere un nome per potere riferire una risorsa
- le **entità risorse per il sistema di nomi** (**server** rispetto ai clienti di prima, ossia che devono essere riferiti) che devono rendersi note al servizio e diventano clienti del name server

… a parte **questi clienti e le loro richieste**, il supporto deve considerare:

- **Comunicazione dei clienti con il name server**
- **Name Server (anche più di uno - agenti)**
- **Gestione dei nomi veri e propri (coordinamento)**
- **Gestione tabelle e coordinamento (spesso ottimizzato localmente)**

Le comunicazioni dei clienti con il name server possono essere ottimizzate per le operazioni più frequenti

- I clienti propongono la maggiore parte del traffico
- Le **risorse da registrare** fanno operazioni più rare e producono traffico più limitato

COMPONENTI: NAME SERVER

Name server devono **fornire operazioni** per consentire la migliore operatività sugli oggetti interni, ossia le **tabelle di corrispondenza** modellate come **tuple di attributi**

Le operazioni

Query	ricerca un oggetto
AddTuple	aggiungi una tupla dal server
ModifyTuple/DeleteTuple	modifica/togli una tupla
Enumerate	lista tutte le tuple, una alla volta

Ogni sistema di nomi decide:

- il formato delle tuple
- il formato specifico delle operazioni

Le realizzazioni prevedono sia Unico servitore sia Agenti Multipli
Il servizio può essere centralizzato,
o molto più spesso distribuito e anche replicato (vedi DNS)

COMPONENTI: COMUNICAZIONE

Nelle realizzazioni con **molteplici Name Server** il servizio prevede una comunicazione tra loro, usando

- messaggi singoli, o datagrammi
- connessioni
- invocazioni di alto livello come RPC

Il traffico tra i diversi Name Server deve essere supportato mentre si continua a fornire il servizio

Il coordinamento dei servitori deve essere **minimizzato in tempo e uso** delle risorse tenendo conto anche delle proprietà che si vogliono garantire

- In uno **spazio piatto**, necessità di fare una partizione dello spazio dei nomi per limitare la coordinazione
- In uno **spazio partizionato**, i nomi sono usati dalla autorità preposta senza coordinamento

Le gestione delle tabelle e il coordinamento creano problemi di consistenza e affidabilità, complicato dalla replicazione

COMPONENTI: GESTIONE NOMI

I **nomi in molte forme** e in base a queste la gestione dipendenti dalla locazione, dipendenti dalla autorità (uso di domini) organizzati in gerarchia (uso di un albero unico riconosciuto di domini)

liberi da struttura (uso di un insieme di attributi e del loro valore)

Nella gestione dei nomi sono fondamentali due decisioni

- **Distribuzione** dei nomi
- **Risoluzione** dei nomi

Distribuzione dei nomi

I nomi sono mantenuti in **oggetti** che ne hanno la responsabilità o autorità con un partizionamento tra i server responsabili

Come dividere la gestione e il mantenimento?

Con politiche di Clustering di vario genere

- **Algoritmico** (hash table / tabelle hash) es. funzione di mapping
- **Sintattico** (pattern matching) es. iniziale del nome
- Basato su **Attributi** (tuple) es. sulla base del valore di attributi

RISOLUZIONE NOMI

Per la Risoluzione dei nomi, le richieste dal cliente devono fluire fino al server che può dare risposta

Il processo di risoluzione dei nomi per dare una risposta prevede alcune fasi (non sempre presenti)

- **Trovare la autorità corretta**
- **Verificare le autorizzazioni alla operazione**
- **Eseguire la operazione**

Ogni nodo specifica i name server noti e tende a limitare se possibile le comunicazioni tra i server

Strategie usuali per limitare i costi sono:

- Politiche di **caching**
- Politiche di **route** tra server
- Creazione di **contesti** o **vicinati** tra i server
- Propagazione di conoscenza tra **vicini**

SISTEMA GLOBALE

Le strategie **di coordinamento tra i server sono cruciali per avere un buon comportamento ed efficienza**
devono essere a basso costo, se possibile, magari usando cache per favorire la località

Nel caso del DNS, i server sono inseriti in un albero gerarchico previso e le comunicazioni sono sempre tra vicini e rispettando la gerarchia

Nel caso del directory X.500, la organizzazione interna dei server è del tutto trasparente e non visibile all'esterno
i server interni sono staticamente presenti per garantire che le informazioni siano sempre presenti, ma dinamicamente si possono aggiungere copie e servitori per query più pressanti e con QoS che richiede vincoli di tempo

ESEMPIO DI SISTEMA GLOBALE

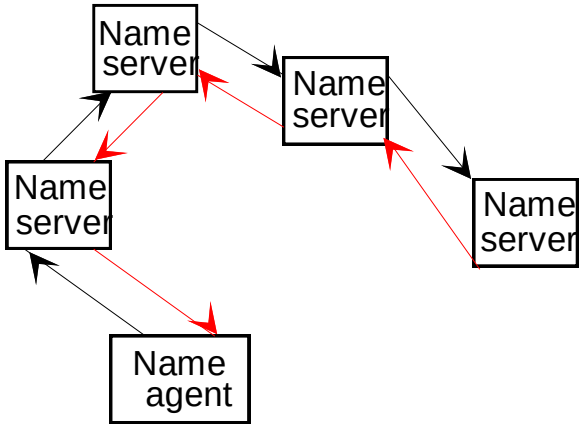
USO DI CONTESTI E LOCALITÀ nei sistemi globali tipo DNS

Si distribuisce e risolve nel contesto locale

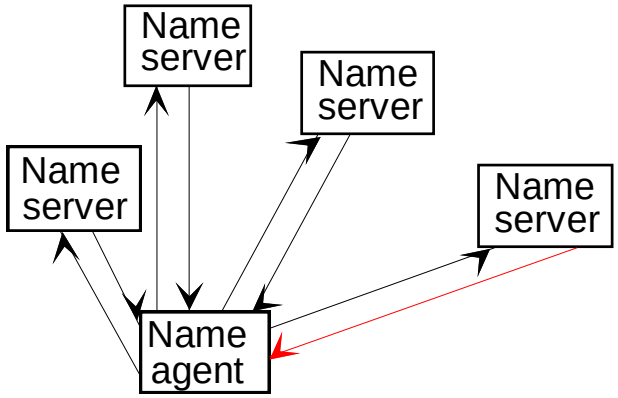
Si ricorre ad altri contesti solo in caso sia necessario

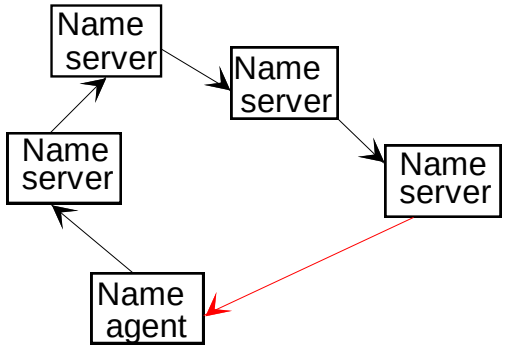
Le strategie di coordinamento tra i server devono essere a basso costo, se possibile

RISOLUZIONE RICORSIVA



RISOLUZIONE ITERATIVA





RISOLUZIONE TRANSITIVA (il DNS non ce l'ha)

ALTRI SISTEMI DI NOMI (OLTRE DNS)

Altri sistemi di nomi, oltre al DNS molto specifico, hanno organizzato i nomi attraverso **attributi** e **ricerca su questi**

OSI X.500 o Directory - Servizio standard di Direttorio e di Nomi con realizzazione partizionata, decentralizzata, disponibile 24/7

- **CCITT** definisce **X.500** come "una collezione di sistemi aperti che cooperano per mantenere un database logico di informazioni sugli oggetti del mondo reale. Gli utenti della Directory possono leggere o modificare l'informazione, o parte di essa, solo se hanno i privilegi necessari"

CCITT	ISO	TITLE
X.500	9594-1	Overview of Concepts, Models and Services
X.501	9594-2	Models
X.509	9594-8	Authentication Framework
X.511	9594-3	Abstract Service Definition
X.518	9594-4	Procedures for Distributed Operation
X.519	9594-5	Protocol Specifications
X.520	9594-6	Selected Attribute Types
X.521	9594-7	Selected Object Classes
X.525	9594-9	Replication

DIRECTORY X.500

X.500 è un insieme di nodi organizzati in un albero che li interconnette e che devono avere una certa QoS

- **Ogni nodo è costituito da attributi tipizzati che possono assumere valori: età = intero**
- **Ogni nodo prevede attributi in base alla gerarchia: aggiunge attributi rispetto al genitore**
- **Ogni gerarchia ha attributi in parte condivisi (padre) e poi differenziati da ogni nodo**
- **Le autorizzazioni alle operazioni sono molto granulari (anche su sottoalbero o nodo)**

Non visibile e QoS interne

- **I nodi sono replicati in modo trasparente e si garantisce sempre l'accesso alle informazioni**
- **Causa replicazione le operazioni raccomandate ed efficienti sono letture; le scritture devono essere limitate**

DIRECTORY X.500

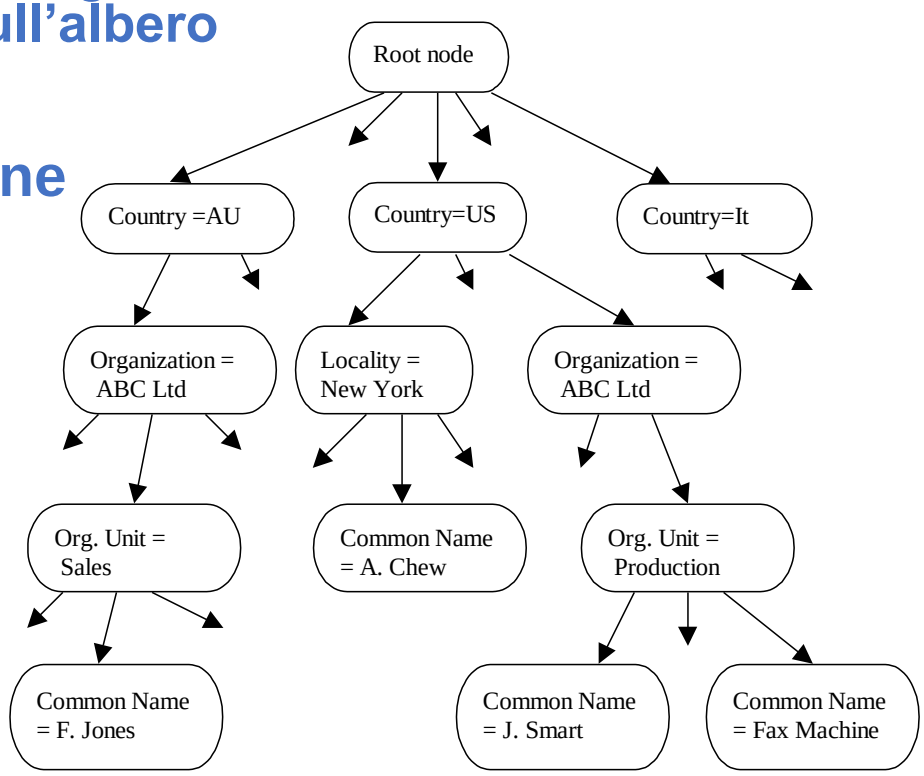
X.500 è un insieme di standard di nomi, articolato e completo

La base è l'insieme delle informazioni che caratterizzano la struttura di **directory**, organizzate in un **albero logico** detto Directory Information Tree (**DIT**) a formare il Directory Information Base (**DIB**)

L'albero logico è costruito da nodi organizzati in base al valore di attributi del tutto liberi e definiti sull'albero

La **novità** sta nella **organizzazione basata sui contenuti** e le **ricerche** (operazioni di lettura) che si possono fare in **modo molto flessibile** per **singole entità** e anche per **attributo, ritrovando gruppi di elementi (nodi) anche molto numerosi**

(interi sottoalberi e foreste)



DIRECTORY X.500 QUERY

Su un direttorio X.500 si fanno query in lettura che possono portare a identificare almeno un nodo

1 solo nodo

- **Si definisce il nome del nodo attraverso un percorso assoluto o relativo**

Molti nodi (in gerarchia o meno)

- **La query è determinata dai valori degli attributi e dalle relazioni specificate tra queste** (le risposte potrebbero anche arrivare, richiedendo di attraversare l'intero albero)

I nodi ottenuti come risultato sono disponibili in modo completo in ogni valore degli attributi

Come avviene la operazione di ricerca?
(sequenziale, parallela)

DIRECTORY X.500 QUERY

Su un direttorio X.500 si fanno query in lettura che possono portare ad identificare almeno un nodo

1 solo nodo

Molti nodi (in gerarchia o meno)

La operazione di ricerca può essere molto articolata e quindi viene ad essere molto **parallelizzabile e parallelizzata**

NOMI E QUERY IN DIRECTORY X.500

Ogni entry (o nodo) si ritrova attraverso diverse notazioni

- **Distinguished Name** (**DN**) che identifica univocamente l'oggetto all'interno del DIT
- **Relative Distinguished Name** (**RDN**) che definisce univocamente un oggetto all'interno di un contesto

DN o **RDN** può fungere da chiave per **identificare unicamente un nodo**

I nodi sono anche selezionabili **attraverso valori delle tuple**, insiemi di coppie **attributo = valore**

Ad esempio:	country	organization	organization unit	Common name
	US	ABCLtd	Production	J.Smart

Le ricerche portano a ottenere informazioni su uno o più nodi ciascuno dei quali contiene tutti i propri valori degli attributi

NOMI E QUERY IN DIRECTORY X.500

Le ricerche (lettura) possono essere fatte in modo **globale** o **contestuale** per uno **specifico DN** ma **anche attraverso query su valori contenuti in attributi, ossia** in base ad **attributi ricercati su tutti i nodi dell'albero**

Una query porta ad avere come risultato **uno o più nodi**

Di conseguenza **tutti i valori degli attributi di quei nodi**

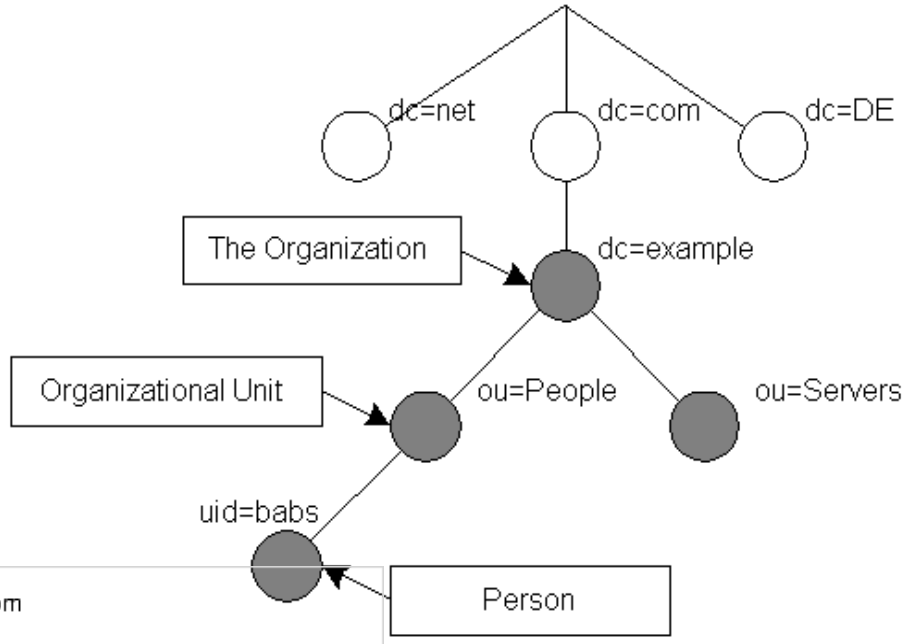
Se avessimo chiesto i nodi con organizzazione ABCLtd avremmo una serie di nodi

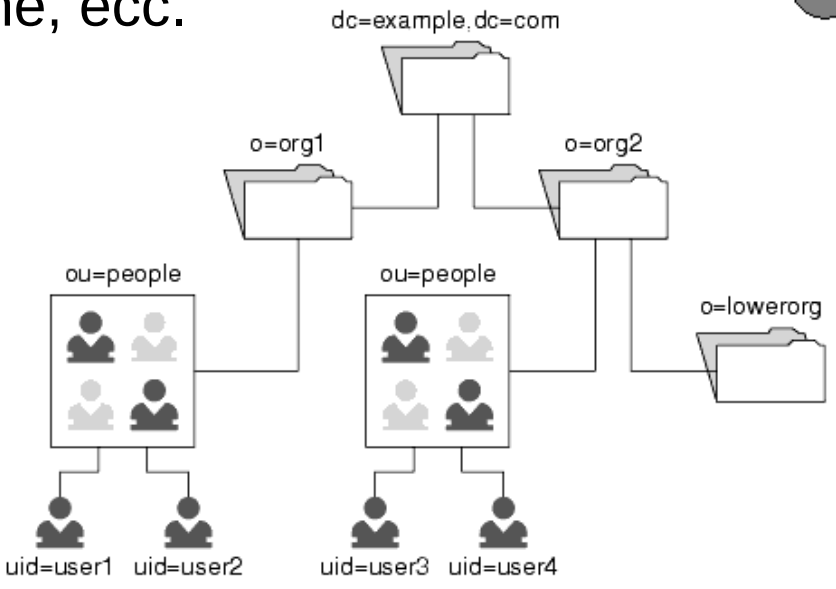
	country	organization	organization unit	Common name
N1	US	ABCLtd	Production	J.Smart
N2	DK	ABCLtd	Management	J.Soren
N3	FR	ABCLtd	Industry	A.Serin
…				

INFORMAZIONI IN DIRECTORY X.500

Le **informazioni** (attributi) dei **nodi** sono **molto eterogenee ma ordinate**

Si possono descrivere organizzazioni, persone, beni patrimoniali, risorse geografiche, ecc.





RICERCHE CON FILTRI

I **filtri** sono **molto potenti come capacità espressiva**

ad esempio, sono permesse **condizioni logiche sugli attributi**

```
CN=Corradi AND C=Italy
```

anche con espressioni regolari `email=*hotmail*`

anche con condizioni aritmetiche `(age >18) AND(cookies <10)`

Le ricerche si possono applicare anche a scope limitati (contesti o sottoalberi)

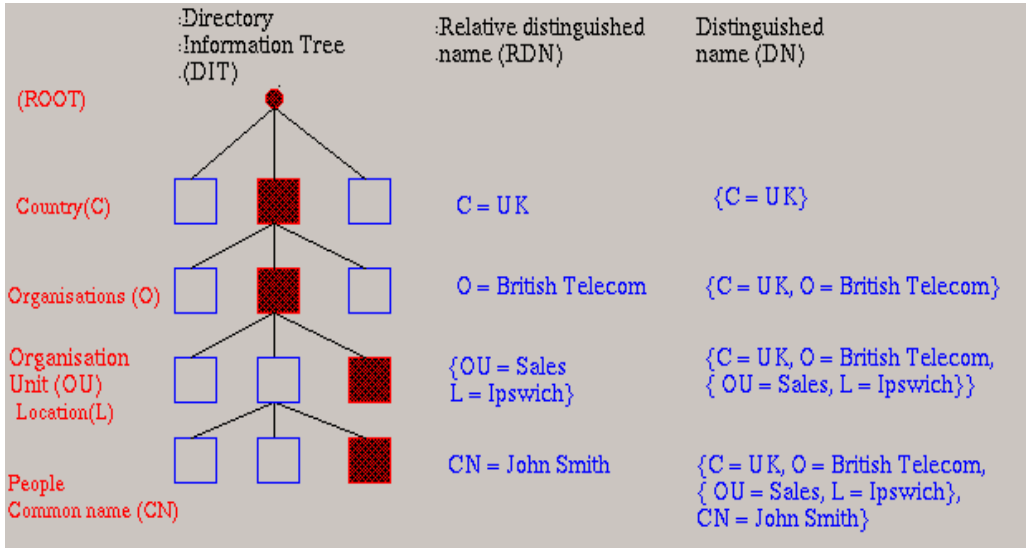
Le operazioni previste sono molte

La **prima** è il **bind** con il

directory poi **ricerche frequenti**

e **cambiamenti** rari

L'interfaccia con il direttorio anche molto complessa tenendo conto anche della durata delle operazioni



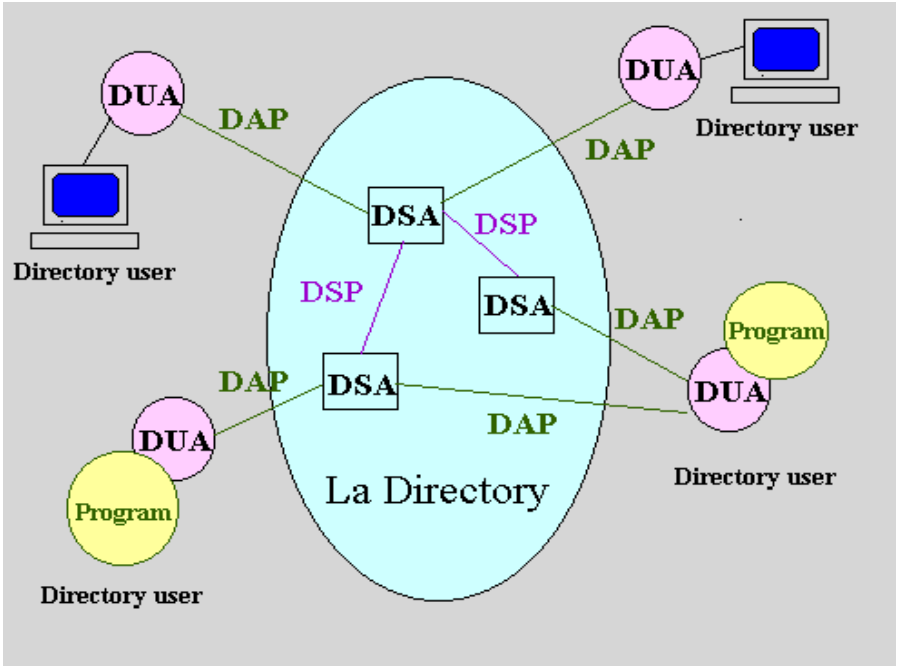
ORGANIZZAZIONE INTERNA X.500

Ricerca sul direttorio X.500 avviene attraverso agenti:

- **DUA**, Directory User Agent tramite per fare richieste
- **DSA**, Directory System Agent che mantiene informazioni di contesto
- **DSP**, Directory System Protocol per scambiare informazioni tra DSA
- **DAP**, Directory Access Protocol, protocollo di accesso al direttorio

Dopo la connessione, si fanno operazioni di lettura, confronto, ricerca, lista delle entità con tecniche di ricerca ricorsive ed iterative

LDAP,
Lightweight Directory Access Protocol
Protocollo limitato compatibile Internet
usato per infrastrutture di
verifica certificati



DIRECTORY E NON DB

Obiettivo di un directory è **mantenere informazioni su un insieme di risorse eterogenee** per un ambiente **evitando duplicazioni di informazioni e problemi di sincronizzazione,** consentendo inoltre:

- una capacità espressiva ampia ed estendibile
- una gestione anche molteplice con più autorità per parti
- una sicurezza anche partizionata e differenziata

Al contrario di un database (o più database), in un directory

- si **associano attributi anche diversi con i singoli oggetti**
- gli **oggetti sono indipendenti tra di loro** (e possono essere diversi)
- si considera la **relazione di contenimento** alla base del directory
- si possono avere proprietà di **accesso differenziate** per i singoli oggetti
- si considerano un **numero elevato di letture** e '**poche**' scritture

Necessità di un protocollo standard unificato per accedere ai dati, esprimere le specifiche di accesso ai dati, ed estrarre informazioni in modo efficace

USO DI DIRECTORY

Le Directory sono tipicamente usati (**vedi costo elevato**) per rappresentare entità eterogenee, come persone, risorse, servizi, server, ...

- In generale, per **informazioni concettuali** che rappresentano la gestione di oggetti comuni in un ambiente condiviso
- **i certificati** per autenticare e autorizzare accesso

MANAGEMENT DELLE RISORSE

- In un sistema di gestione in cui consideriamo una molteplicità di **gestori con autorità e responsabilità** anche non completamente note e forse variabili lentamente nel tempo

si usano DIRECTORY per

- Localizzare i **diversi gestori e le loro politiche**
- Trattare i problemi di **domini incrociati** (cross-domain)
- Ritrovare le proprietà delle **risorse**
- Ritrovare le proprietà dei **gestori delle risorse**

I costi sono dipendenti dal numero dei nodi (e attributi)

motivati dalle proprietà di QoS offerto (affidabilità e disponibilità)

SVILUPPO DEI SISTEMI DI NOMI

Due forme di evoluzione

Protocolli di Directory vs. Protocolli di Discovery

Considerando che una entità possa avere sia attributi con lente variazioni sia attributi con variazioni veloci

Directory soluzioni di nomi globali

servizi completi e complessi con costo elevato delle operazioni

Discovery soluzioni di nomi locali

servizi essenziali e funzioni limitate costo limitato adatto a variazioni rapide

Ad esempio: **Un utente generico** che si muova in un sistema globale

vuole avere accesso a informazioni globali, essenzialmente stabili, come

- descrizione dei dispositivi, delle preferenze proprie del suo profilo,
- delle sue firme digitali e PKI, delle sue sorgenti di informazioni, ecc.

informazioni locali, anche molto variabili, come

- descrizione delle risorse locali, dei gestori presenti, ecc.

PROTOCOLLI DI DIRECTORY

Un servizio di directory garantisce le proprietà di QoS, ossia di replicazione, sicurezza, gestione multiserver, ..., tutto il supporto per memorizzare le informazioni organizzate prevedendo molti accessi in lettura e poche variazioni

UPnP (Universal Plug-and-Play) Standard per architetture Microsoft

Servizi di Nomi basati su variazioni di DAP (o LDAP)

- Windows2000 propone Active Directory come un servizio di direttori integrato nel e per il sistema operativo

Salutation - Service Location Protocol (RFC 2165)

- si possono registrare servizi diversi
- i servizi vengono divisi in località distinte
- i servizi vengono protetti in diversi modi
- interfacce compatibili con i browser (Web) e uso di nomi URL
- Le operazioni definite e implementate permettono di fare ricerche evolute sulle informazioni (memorizzate in modo globale) e compatibili con la maggior parte degli strumenti e dei sistemi di nomi più diffusi
- implementazioni: Cisco, Apple, Novell

PROTOCOLLI DI DISCOVERY

Per computazione distribuita e cooperativa in ambito locale

- Una unità deve ritrovarne altre, in modo veloce e economico
- Si prevedono azioni come il broadcast e solleciti periodici

un servizio di discovery definisce e standardizza come si possano ritrovare altre entità correntemente visibili (località delle risorse)

JINI protocollo Java per il discovery di **appliance**

Si vuole rispondere alle esigenze di chi arriva in un contesto e vuole operare senza conoscenze predefinite

Protocolli di lookup

Start up con multicast in ambiente locale

Il discovery server verifica la presenza delle risorse ad intervalli opportuni