

COM & DCOM (Distributed) Component Object Mode

Valentina Del Frate
Corso Di Controllo Digitale
A.A. 1999/2000

COM : Prima Definizione (Glossario Informatico)

- Una Architettura software messa a punto da Microsoft e DEC per consentire le funzionalità OLE

Breve Storia (1)

- Microsoft crea Windows : sistema operativo per gestione contemporanea di applicazioni.
- Necessità di un sistema per scambio dati -comunicazioni fra applicazioni: *Clipboard* (Appunti).
 - Utile ma
 - Limitata
- **DDE (Dynamic Data Exchange)**.
DDE (Dynamic Data Exchange) metodo(Microsoft) che consente di :
 - Incorporare, in un documento, parte di altri documenti, anche prodotti con software diversi
 - Mantenere i documenti inseriti aggiornati automaticamente con le modifiche effettuate nei documenti originali.
 - Capace di lavorare su reti LAN (Network DDE).
 - Poco affidabile.

Breve Storia (2)

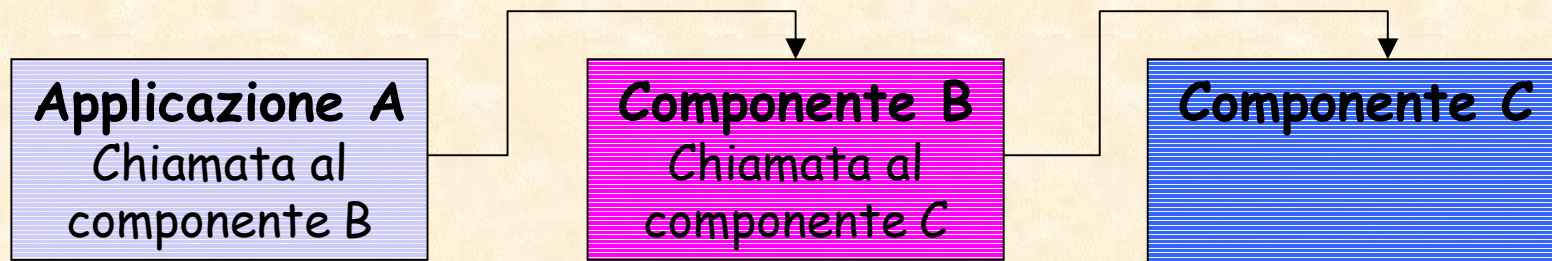
- **OLE (Object Linking and Embedding):** metodo(Microsoft-Windows '9x) che permette di:
 - Inserire in un documento, parte di un altro documento (un grafico, un'immagine, una tabella...) preparato con un altro programma.
 - La parte inserita rimane modificabile attraverso il programma originario e le modifiche sono riportate automaticamente nel documento che la contiene.
- OLE ha due livelli:
 - **OLE 1** quando si fa doppio click sul documento inserito viene caricata l'applicazione ed il documento originari
 - Utilizza DDE per comunicazioni tra *applicazioni*.
 - Permette di creare *Documenti Composti*

Breve Storia (3)

- **OLE 2**
 - **OLE 2** quando si fa doppio click sul documento inserito, si rimane all'interno della stessa applicazione, ma la barra dei menu e le barre degli strumenti vengono sostituite con quelle dell'applicazione originaria.
 - Abbandona DDE
 - Nuovo protocollo di comunicazione basato su componenti :
COM
- **COM**
 - Sviluppo su componenti
 - Possibilità di suddividere applicazioni in grandi in parti
 - Possibilità di avviare ed eseguire componenti su computer in rete usando le sue risorse (CPU, memoria)
 - Visual Basic 4
- Uso di controllori OLE
- **DCOM** (Distributed Component Object Model)
 - Uso di controllori ActiveX

Definizioni

- **Componente:** blocco di software che svolge un particolare servizio.
 - Utilizzabile da altri software.
 - Combinabile per formare nuove applicazioni.
 - Termine nato per non creare confusione per quei programmi che svolgono contemporaneamente funzioni di client OLE e server OLE.
- **COM** permette di:
 - Assemblare componenti di software
 - Riutilizzarli nelle applicazioni



- **Componenti COM:**
suddivisi in funzione del luogo dove vengono eseguiti
- **Server in process (DLL) :**
 - Comunicano con il loro client senza aiuto di Com
 - Utili quando si privilegia la velocità
 - Client non protetto da malfunzionamenti del server
 - i.e. Controlli ActiveX
- **Server out-of-process locali (EXE)**
 - Usato quando si vuole creare una applicazione che possa lavorare come programma autonomo e che renda disponibili all'esterno i suoi oggetti programmabili
 - Richiedono servizi Com per comunicare con l'esterno : comunicazione lenta rispetto ai precedenti
 - i.e.: applicazioni pacchetto Microsoft Office (applicazioni indipendenti-fornitori di componenti per altri programmi

Server out-of-process remoti (EXE)

- Programmi eseguiti su macchina diversa da quella su cui viene eseguita l'applicazione client
- Utilizzo del protocollo DCOM per comunicazione client-server
- Comunicazione ancora più lenta
- Possibilità di creare applicazioni distribuite

Applicazioni Distribuite

- Benefici all'utente in termini di ottimizzazione uso della rete e risorse computer
- Fanno eseguire componenti
 - Dal lato client quando possibile
 - Dal lato server se necessario (vedi Indipendenza da Locazione)
- Flessibilità
- Possibilità di aumentare il throughput senza lavorare su miglioramenti hardware

Attenzione !

Programmazione a oggetti

- Modo di pensare e progettare codice sulla base di oggetti
- Possibilità di sviluppare componenti di software a oggetti (i.e. componente per la correzione automatica)
- Riguarda la creazione di oggetti

Programmazione COM

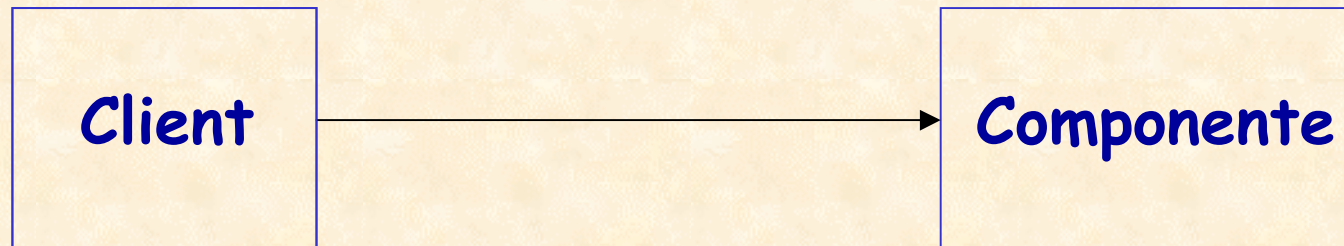
- Possibilità di combinare componenti esistenti sviluppati in qualsiasi linguaggio
- Riguarda il funzionamento congiunto di più oggetti creati

Da COM a DCOM

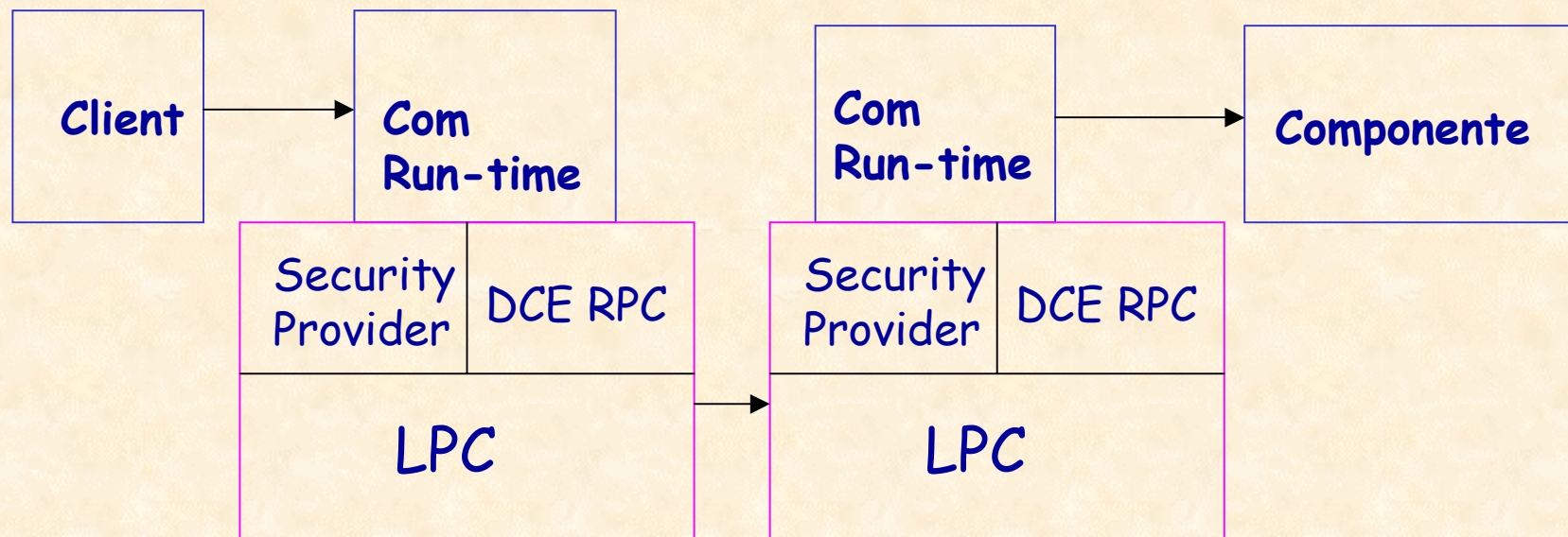
Architettura

- Nei normali S.O.: i processi che devono comunicare con altri processi non li possono chiamare direttamente ma devono usare alcune forme di comunicazione "inter-processo" del S.O. stesso
- **COM** :
 - Definisce l'interazione fra componenti e loro client fornendo una connessione senza componenti intermedi di sistema
 - Fornisce una comunicazione trasparente tra processi diversi, intercettando la chiamata del client ed inoltrandola al componente del processo richiesto

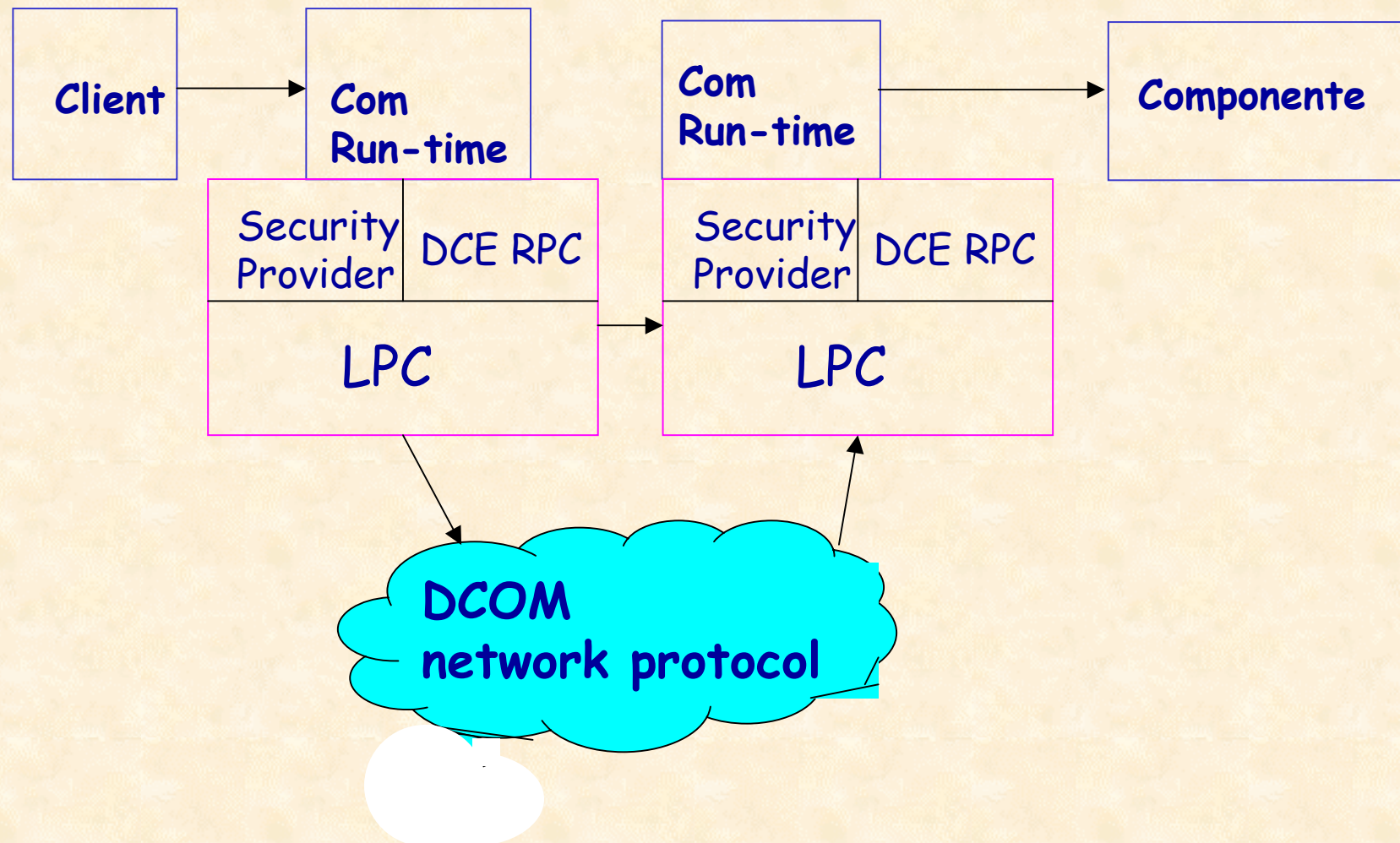
- Richiamo di componenti nello stesso processo (in-process)



- Richiamo di componenti in processi differenti con COM (out-of-process local)



- Quando i componenti risiedono su macchine differenti DCOM sostituisce alla comunicazione inter-processo un protocollo di rete(out-of-process remote)
- **RPC** è un servizio di chiamata a procedura remota in accordo allo standard DCE



L'esecuzione di COM (COM run-time) usa RPC e il security Provider per generare pacchetti di rete standard conformi al protocollo DCOM

Per chiamate a componenti risiedenti su macchine diverse DCOM fa in modo che né il client né il componente si rendano conto che il cammino di connessione fra loro si è allungato

DCOM

Indipendenza dalla locazione

- Generalmente implementazione di applicazioni distribuite su rete ➡ vincoli di progettazione
- **Soluzione DCOM:**
 - Non mostra la locazione di un componente ovunque esso sia (stesso processo, processi differenti macchine diverse)
 - Il modo in cui il client effettua la chiamata al componente non cambia (come se il client non se ne rendesse conto)
 - Non effettua modifiche al codice né ricompilazione, ma riconfigura solo il modo in cui i componenti sono connessi l'uno con l'altro
- I componenti possono risiedere nel posto più adatto alle funzioni che devono compiere

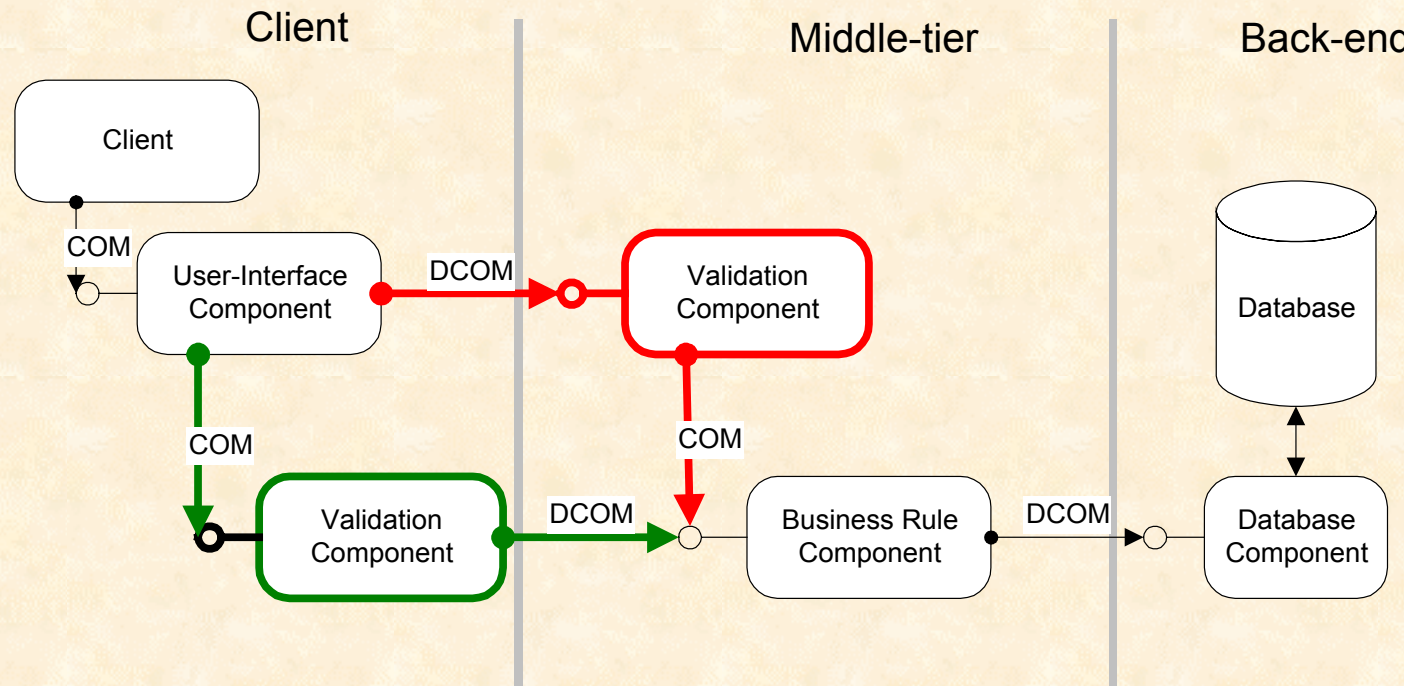
DCOM

Indipendenza dalla locazione

- Esempio:

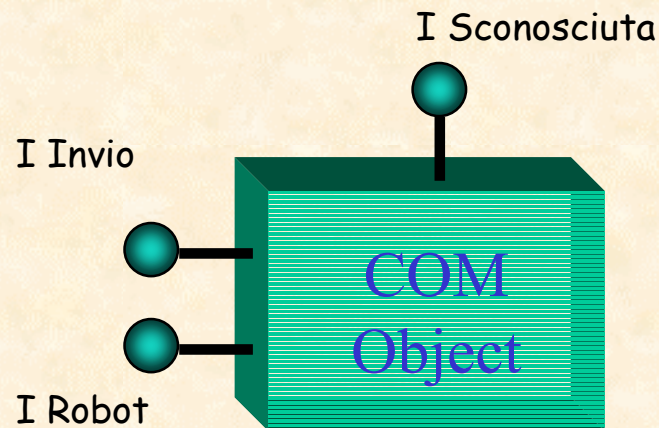
In un'applicazione commerciale abbiamo che:

- L'interfaccia utente e la validazione dei dati risiederanno sul client
- L'elaborazione dei dati sul server che accede al database



Neutralità di Linguaggio

- DCOM, come anche COM cui deriva, è completamente indipendente dal linguaggio usato
- Componenti DCOM possono essere scritti in quasi tutti i linguaggi (praticamente tutti quelli di produzione Microsoft)
- Possono essere usati in tutti i linguaggi per sistemi Windows e in applicazioni evolute
- Oggetto Com visto come una scatola nera:
 - Tutta la manipolazione dell'oggetto tramite interfacce



Gestione della Connessione

- Connessioni di rete più fragili di quelle presenti all'interno della macchina
- Componenti in applicazioni distribuite devono essere avvisati
 - Se un client è ancora attivo
 - Se rete o hardware falliscono
- DCOM gestisce connessioni a componenti dedicate ad un singolo client nello stesso modo in cui gestisce quelli condivisi da molteplici client
- Usa un protocollo di ping per verificare se il client è ancora attivo:
 - La macchina client invia un messaggio periodico
 - DCOM considera caduta la connessione se il componente non riceve nessun messaggio di ping per più di tre periodi
 - Quando la connessione è caduta DCOM decrementa il numero di client connessi a quel componente.
 - Se questo numero diventa zero, allora rilascia il componente

- Dal punto di vista del componente la disconnessione di un client o un crash nella rete e nella macchina client sono trattati nello stesso modo
- Ricordiamo che la comunicazione tra un componente e il suo client non è unidirezionale (ogni componente può essere sia un fornitore che un consumatore di una certa funzionalità), ma simmetrica

Scalabilità

- **Scalabilità:** fattore critico per applicazioni distribuite (aumento del numero di utenti-dati-funzionalità richieste)
- **SMP:** Symmetric Multiprocessing Protocol di Windows NT a cui le applicazioni DCOM si appoggiano
- **Soluzione DCOM :** l'indipendenza da locazione permette di distribuire componenti su altri computer al crescere del carico dell'applicazione
- **Componenti DCOM possono essere riposizionate.**
 - Il carico dell'utente può essere distribuito fra le macchine, con criteri che possono riguardare la capacità delle macchine o anche il carico attuale delle stesse.
 - Possibilità di cambiare il modo con cui i client si connettono alle componenti e le componenti si connettono fra loro.
 - Riposizionamento delle componenti non richiede ricompilazione e può essere fatto dinamicamente.
 - Unica cosa necessaria è aggiornamento di registro, file system, database dove è registrata la locazione di ogni componente.

- **ESEMPIO:**

- Una organizzazione con uffici in diverse località (i.e. capitali mondiali) può installare i componenti nei suoi server.

200 utenti contemporaneamente accadono a 50 componenti per gestire operazioni di borsa in un server aspettandosi certe prestazioni.

- Quando nuove applicazioni di borsa sono consegnate agli utenti, alcune di esse usano componenti di borsa già esistenti e altre dei nuovi. Il carico del server arriva a 600 utenti e il numero di componenti di borsa sale a 70.

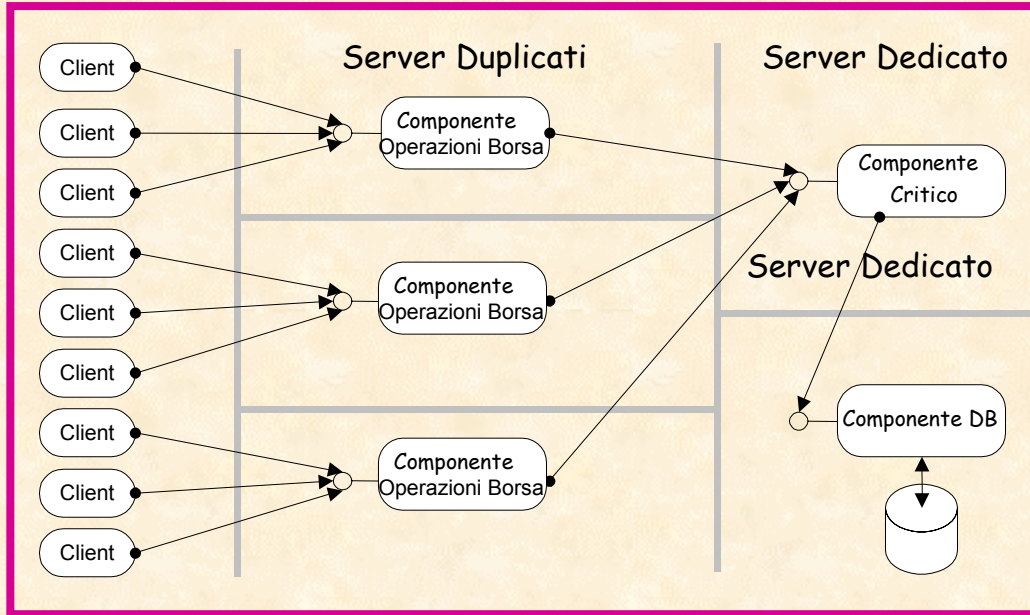
Il tempo di risposta diventa inaccettabile!

Soluzione: l'amministratore aggiunge un secondo server e riposiziona 30 delle componenti esclusivamente su questo.

20 componenti rimangono esclusivamente sul vecchio server e i restanti 20 sono fatti girare su entrambe le macchine (vengono duplicati).

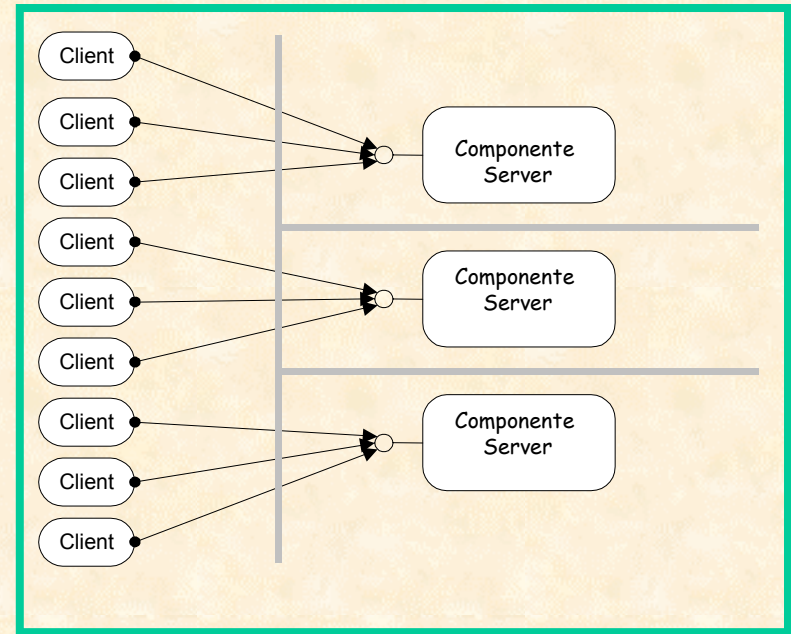
Nota Bene

- Esistono componenti "critici" che non possono essere duplicati (i.e. quelli che necessitano accesso seriale, e che quindi hanno lo scopo di sincronizzare gli utenti).
- Questi devono essere sistemati su macchine dedicate (server potenti).
- DCOM permette di muovere i componenti critici su macchine dedicate dopo averli prima isolati e poi aver sistemato inizialmente i componenti multipli su una singola macchina.
- **Soluzione DCOM** per diminuire il rallentamento introdotto da componenti critici è suddividere il componente critico in sottocomponenti e creare una sorta di catena di montaggio, in cui i sottocomponenti lavorino in parallelo processando (sempre sequenzialmente) le richieste.

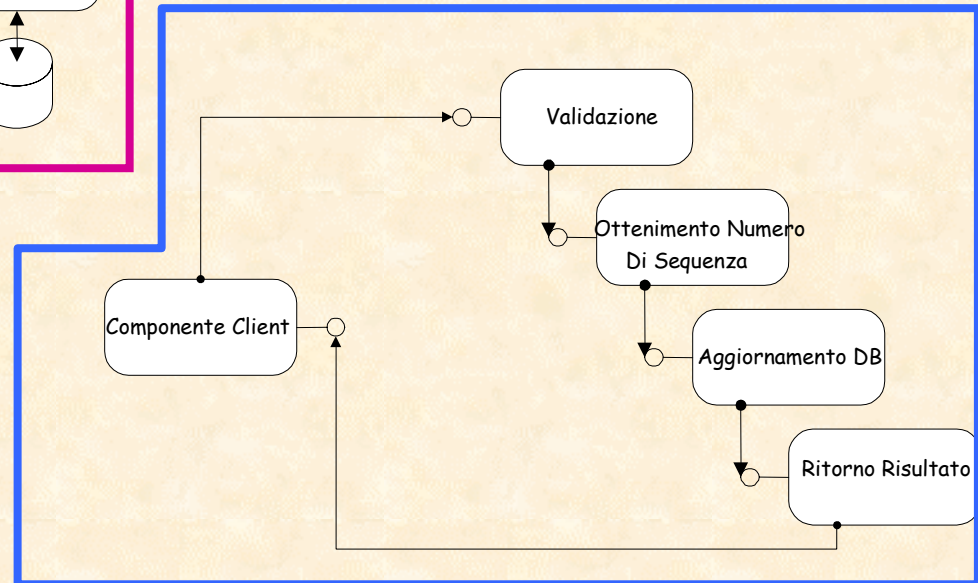


Isolamento dei Componenti Critici

Suddivisione Sequenziale Componente Critico



Riposizionamento in Parallelo (Esempio Operazioni Borsa)



Aggiornamenti

- Un componente COM può apparire diverso a client diversi
- Un client che usa una certa caratteristica ha bisogno di accedere solo a metodi e proprietà che usa.
- Se altre caratteristiche vengono aggiunte ai componenti, queste non incidono sul vecchio client che neanche se ne rende conto
- Quindi : il componente iniziale espone un set base di caratteristiche come interfacce COM, su cui ogni client può contare.
- Se il componente acquista nuove caratteristiche, la maggior parte delle interfacce già esistente sarà ancora necessaria, e nuove funzioni e proprietà appariranno in interfacce aggiuntive senza cambiare le interfacce originali.
- I vecchi client accederanno al set base come se nulla fosse cambiato.
- I nuovi client hanno la possibilità , se disponibili, di usare le nuove interfacce.

Performance

- Il modo con cui client comunica con componenti è attraverso le chiamate
- Esiste una tabella di indirizzi di metodo a cui il client accede per ottenere gli indirizzi dei metodi di cui ha bisogno
- Se il componente è un componente in-process che gira sulla stessa **linea** del client la chiamata del metodo arriva direttamente al componente.
- Se client e componente si trovano su processi differenti o su macchine diverse ai due capi del mondo, allora COM inserisce il codice infrastrutturale della sua procedura remota di chiamata (RPC) nella tabella di indirizzi e poi impacchetta ogni chiamata di metodo in una rappresentazione standard di buffer, che invia al lato componente, che spacchetta e ripristina l'originale chiamata al metodo
- La performance si riferisce alla velocità del meccanismo RPC

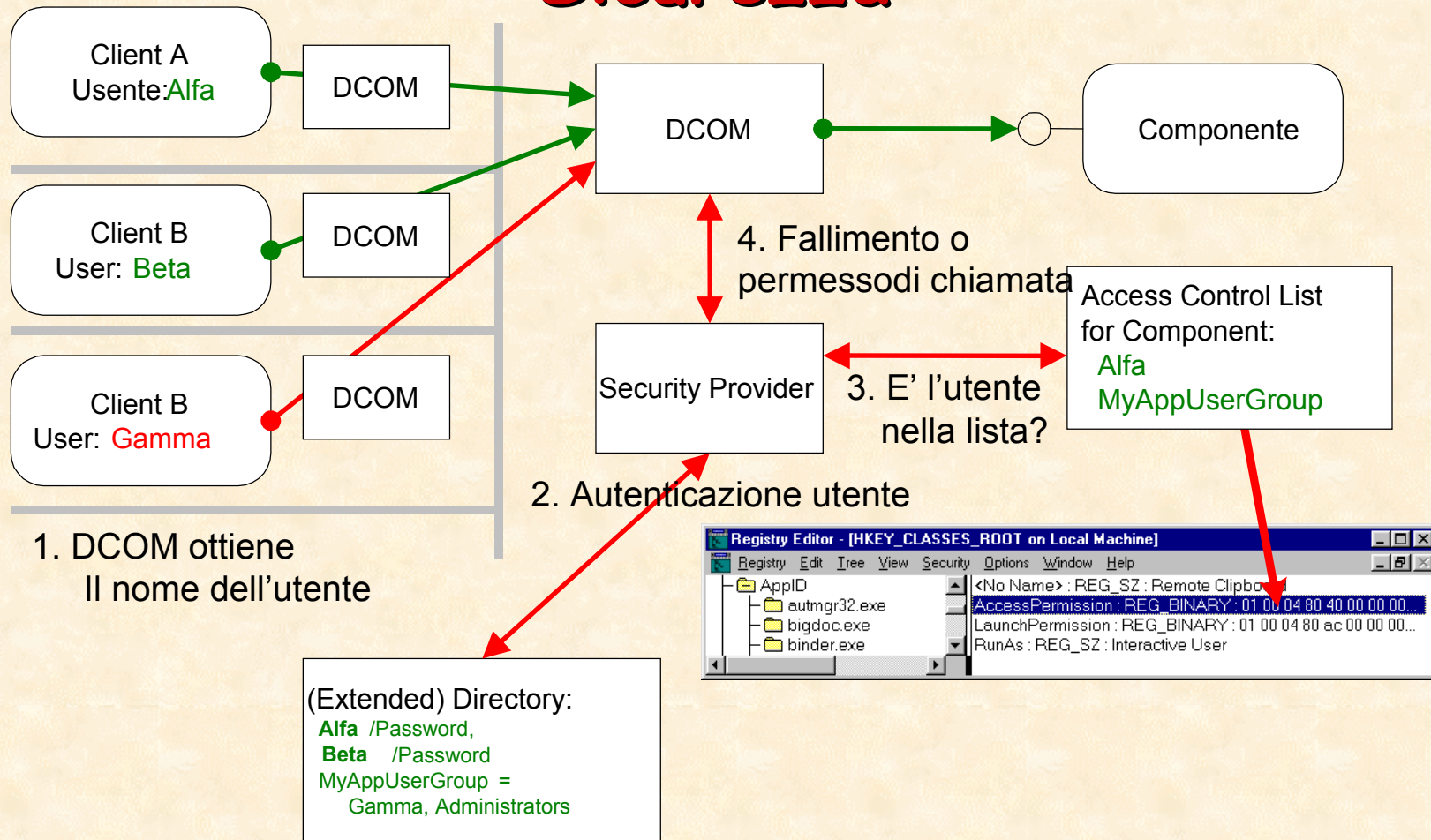
Larghezza di Banda e Latenza

- Larghezza di banda: la grandezza dei parametri passati in una chiamata di metodo influisce direttamente sul tempo impiegato a completare la chiamata
- Latenza: la distanza fisica e il numero di elementi di rete coinvolti ritarda anche pacchetti con piccolo numero di data
- Soluzione DCOM per minimizzare le due: uso del protocollo di trasporto UDP (senza connessione)

Sicurezza

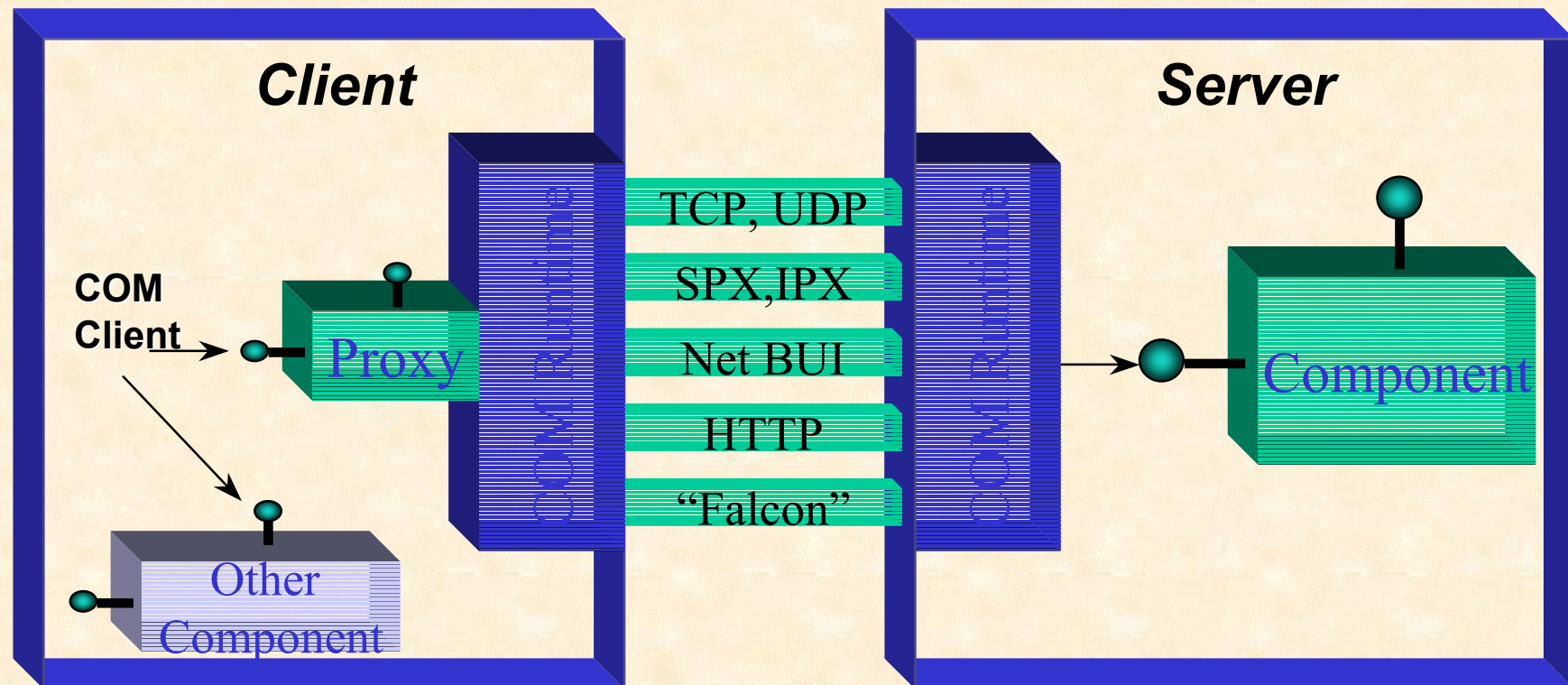
- Poiché ora molte operazioni sono accessibili fisicamente da chiunque con un accesso alla rete, c'è bisogno di una limitazione a un livello più alto (i.e. username e password)
- DCOM utilizza un solido insieme di build-in security providers che supporta identificazioni e meccanismi di identificazione multipli.

Sicurezza



Neutralità di Protocollo

- DCOM può usare qualsiasi protocollo di trasporto (TCP/IP, UDP, IPX/SPX e NetBios), fornendo una struttura sicura su tutti questi protocolli, includendo protocolli orientati alla connessione e non.
- Si possono quindi utilizzare le caratteristiche fornite da DCOM ed essere certi che le applicazioni così create sono neutrali per quanto riguarda il protocollo di trasporto



Piattaforme Disponibili

- DCOM su Windows: piattaforma Windows NT in Windows NT Workstation 4.0 e Windows NT server 4.0.
- DCOM su AppleMacintosh: Microsoft è al lavoro per ottenere una implementazione di DCOM per questa piattaforma.
- DCOM per UNIX/Mainframe: Microsoft è al lavoro per ottenere una implementazione di DCOM per questa piattaforma.
- DCOM e Java: Microsoft fornisce l'implementazione di riferimento per la Virtual Machine Java per piattaforma Windows.

DCOM E Babybot

- Uso di componenti permette di
 - Stratificare il codice
 - Creare una separazione fra livello basso che maneggia risorse hardware (ad esempio frame grabber, assi di controllo, etc) e livello alto di implementazione (ad esempio anelli di controllo, apprendimento,...).
- Una volta che un componente , ad esempio un algoritmo di rete neurale, è progettato, esso può essere riusato in tutto il sistema
- Possiamo realmente progettare un sistema distribuito con un ambiente di programmazione uniforme (al contrario del sistema operativo, DCOM è uno standard).
- Sistema parallelo dove la più piccola unità e' un singolo PC. Questi PC possono essere facilmente connessi tramite fast Ethernet .
- Flessibilita' ed espandibilita' garantite.

- Gli oggetti sono distribuiti attraverso le macchine → in ogni momento un qualsiasi oggetto può accedere ad ogni altro oggetto nel sistema senza difficoltà.
- Piattaforma di sviluppo con approccio opposto alla "corrente": mentre molti sistemi di controllo sono centralizzati - o su una singola macchina o su un singolo sistema di bus- l'architettura Babybot è invece distribuita e connessa attraverso una rete relativamente lenta (se paragonata a un bus).

Architettura Distribuita di Babybot

- Attualmente consiste di 4 pc connessi con una rete Ethernet a 100mbit/sec. Uno che monta frame grabber per processi visuali, un secondo è dedicato al controllo della testa e un terzo al controllo del braccio. Il quarto è impiegato per scopi di monitoraggio e non esegue nessun processo rilevante.

