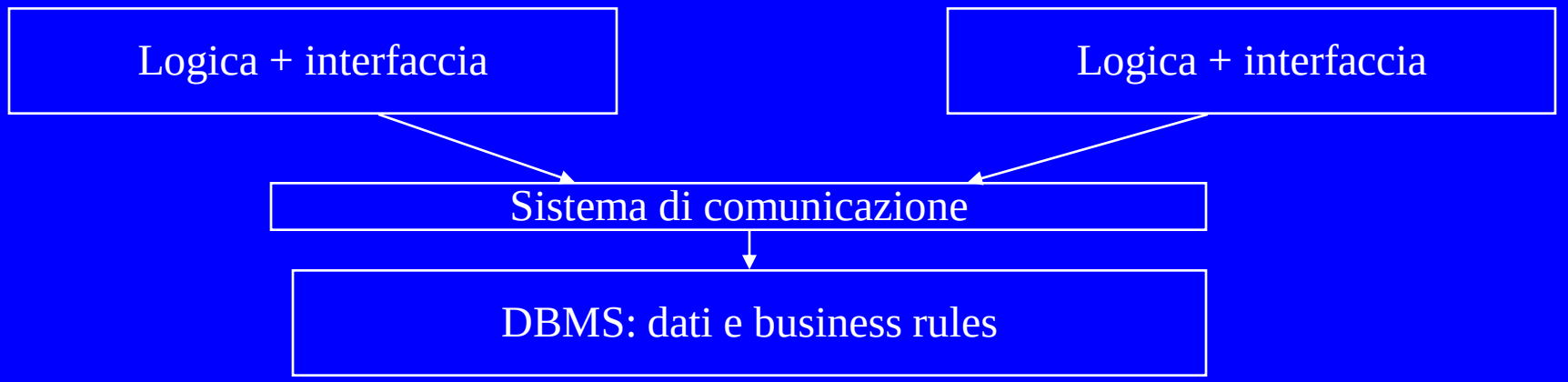
LA REALIZZAZIONE DI APPLICAZIONI

- Quattro parti:
  - Gestione dati
  - Business rules
  - Logica applicativa
  - Interfaccia utente
- Molte possibili architetture
- L’approccio tradizionale: uso di un linguaggio

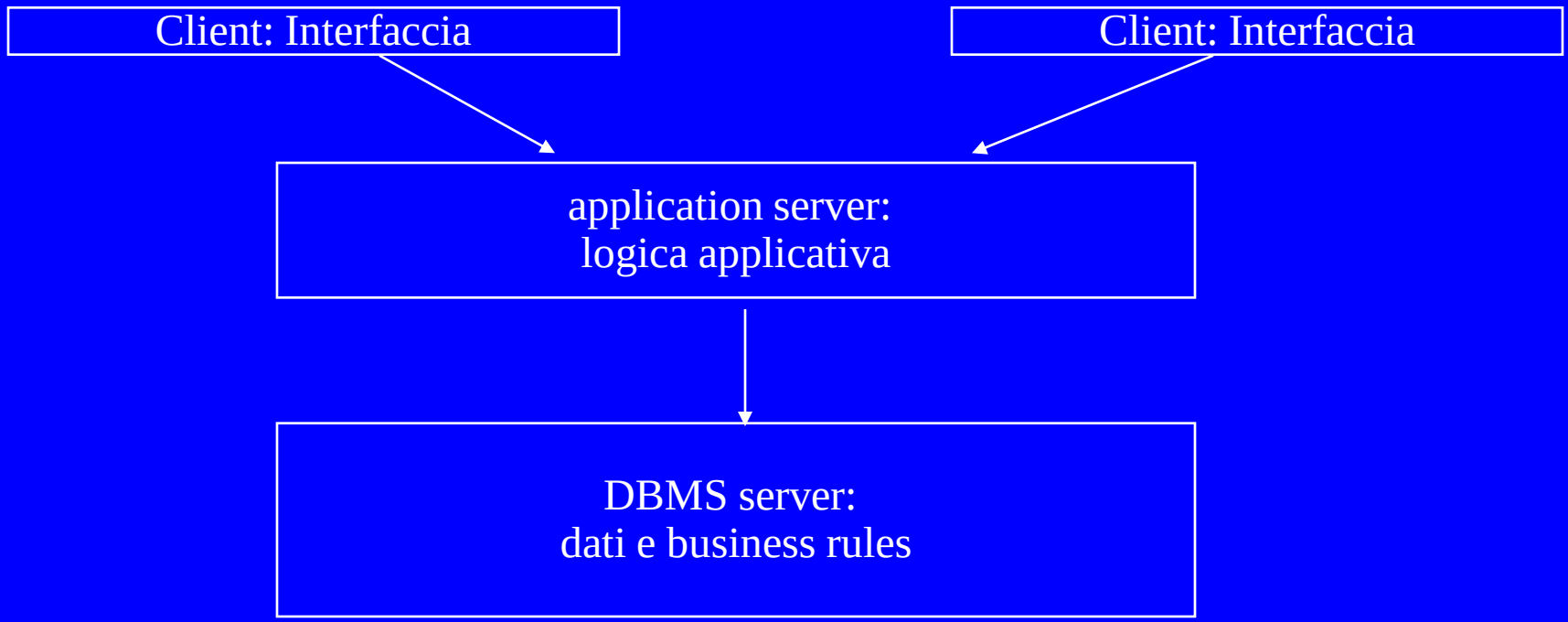
ALCUNE ARCHITETTURE

Client-Server

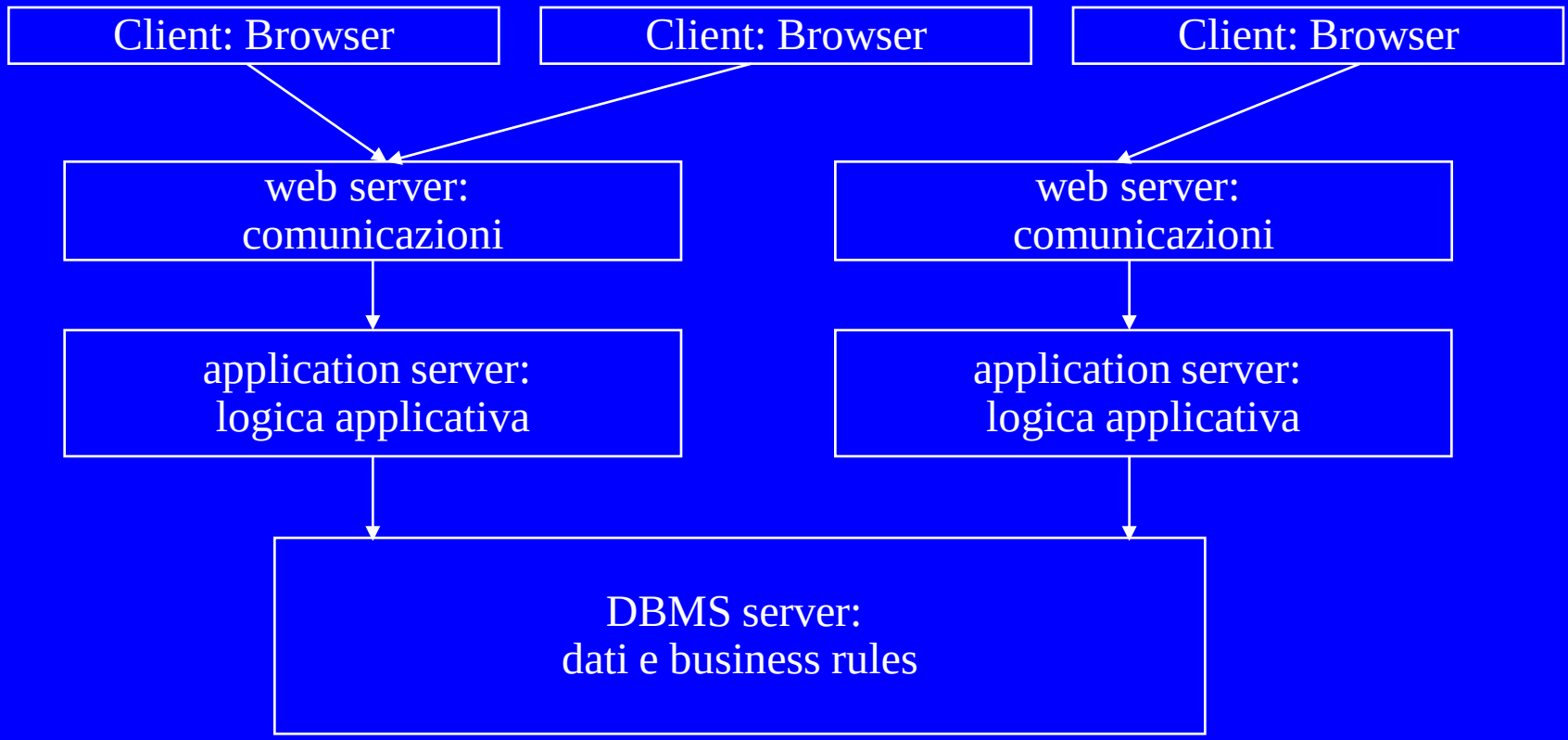


Thin Client

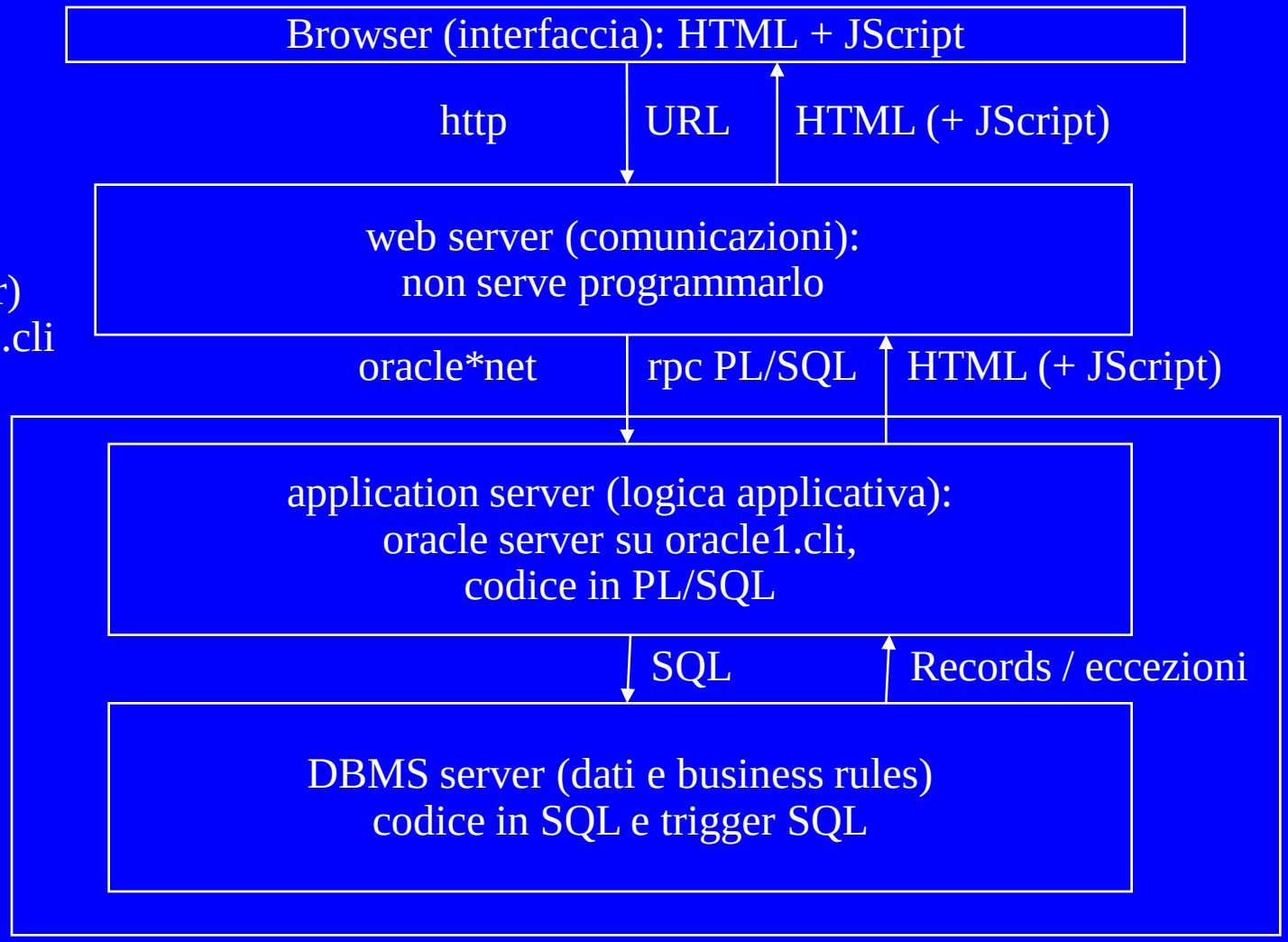
THREE TIERS


Client: Interfaccia
Client: Interfaccia
application server:
logica applicativa
DBMS server:
dati e business rules

WEB SERVER


Client: Browser
Client: Browser
Client: Browser
web server:
comunicazioni
web server:
comunicazioni
application server:
logica applicativa
application server:
logica applicativa
DBMS server:
dati e business rules

LA NOSTRA ARCHITETTURA



IL LINGUAGGIO

- Scrivere un’applicazione che visualizzi una schermata, raccolga dei dati, segnali eventuali inconsistenze con i dati nella BD oppure inserisca i nuovi dati
- Serve un linguaggio che possa:
  - Effettuare I/O
  - Effettuare interrogazioni
  - Controllare il flusso in un modo che dipende dal risultato dell’interrogazione
  - Effettuare aggiornamenti

IL LINGUAGGIO

- Tre soluzioni:
  - Linguaggio di programmazione + API
  - Linguaggio immerso
  - Linguaggio integrato

LINGUAGGIO INTEGRATO

- Integrazione dei tipi di dato
- Integrazione dello scoping
- Integrazione del DML

UNA SOLUZIONE

```
procedure prenotaIf(
    ilLogin          in prenota.login%TYPE,
    laData           in date,
    lOra             in prenota.ora%TYPE) is
unaPrenotazione prenota%ROWTYPE;
cursor c is select *
            from prenota
            where data = laData and ora = lOra;
begin
  open c;
  fetch c into unaPrenotazione;
  if c%NOTFOUND
  then insert into prenota
          values (codiceSeq.nextval, ilLogin, laData,
                      lOra, ilTerm, laDataP, ilTermP);
  else ...;
  end if;
end prenotaIf;
```

L'ASPETTO DEL CODICE

```
procedure creaElaborato(
    CodTemp          AllocazioniTemp.CodTemp%type,
    Resp             number,
) is
 i     binary_integer;
 cursor callocazioni is select CodEl, Matricola,
 from Allocazioni
 where Allocazioni.CodEl = CreaElaborato.CodTemp;
begin
 insert into Elaborati (CodEl, PassEl, CodCo)
 values (CodEl, PassEl, '1');
 open callocazioni;
 loop
  fetch callocazioni into IlCodTemp,LaMatricola;
  exit when callocazioni%notfound;
  insert into Allocazioni (Matricola,CodEl,Responsabile)
  values(LaMatricola,CodEl,'N');
 end loop;
 close callocazioni;
endcreaElaborato;
```

PL/SQL

- Un linguaggio per manipolare basi di dati che integra DML (SQL) con il linguaggio ospite
- Un linguaggio a blocchi con una struttura del controllo completa che contiene l'SQL come sottolinguaggio
- Permette:
  - Di definire variabili di tipo scalare, record (annidato), insieme di scalari, insiemi di record piatti, cursore
  - Di definire i tipi delle variabili a partire da quelli della base di dati
  - Di eseguire interrogazioni SQL ed esplorarne il risultato
  - Di modificare la base di dati
  - Di definire procedure e moduli
  - Di gestire il flusso del controllo, le transazioni, le eccezioni

STRUTTURA

- Il blocco:
  - DECLARE <dichiarazioni>
  - BEGIN <comandi>
  - EXCEPTION <gestori>
  - END;

- La procedura:
  - PROCEDURE <parametri> IS
  - <dichiarazioni>
  - BEGIN <comandi>
  - EXCEPTION <gestori>
  - END;

Il modulo: interfaccia e implementazione

- CREATE PACKAGE <nome> AS ... END <nome>;
- CREATE PACKAGE BODY <nome> AS ...
  END <nome>;

LE DICHIARAZIONI DI VARIABILI

- <nome> <tipo>;
- <nome> CONSTANT <tipo> := <expr>;
- CURSOR <nome> IS <query>;
  - Operazioni: OPEN, FETCH, CLOSE
- Attributi di variabili e cursori:
  - Var%TYPE, dove var è una variabile o una colonna
  - Cur%ROWTYPE dove cur è un cursore o una tabella

IL FLUSSO DEL CONTROLLO

- IF - THEN - (ELSIF THEN) - ELSE - END IF
- LOOP <comandi> END LOOP (EXIT WHEN <cond>)
- FOR var IN seq LOOP <comandi> END LOOP
- GOTO label ... <<label>> comandi

TABELLE E RECORD

- Una tabella è un’array associativo dinamico, di tipo
  - TABLE OF <tipo> INDEX BY <tipo>
- I tipi record sono annidabili:
  - RECORD (nome tipo,...,nome tipo)

ECCEZIONI

- DECLARE aaa EXCEPTION;
- RAISE aaa
- EXCEPTION WHEN aaa THEN ... END

ARCHITETTURA

- Tre possibilità:
  - Un programma PL\SQL può essere eseguito da uno strumento, che invia i comandi SQL al server;
  - Un programma PL\SQL può essere inviato da uno strumento al server, il quale lo esegue;
  - Un programma PL\SQL può risiedere nel server, che lo esegue.