




Databases

Client/Server architectures

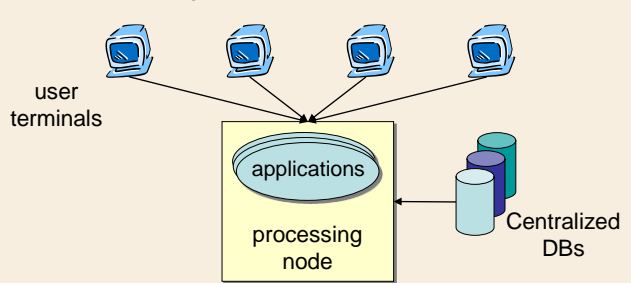
DBG



Centralized architecture

⊳ An architecture is centralized when data and applications (programs) reside in a single processing node

- The intelligence is in the central elaborator (host or mainframe)



The diagram illustrates a centralized architecture. On the left, four icons representing user terminals are connected by arrows to a central yellow box labeled 'processing node'. Inside this box, there is an oval labeled 'applications'. To the right of the processing node, there are three cylindrical icons representing 'Centralized DBs' (databases), with an arrow pointing from the databases to the processing node.

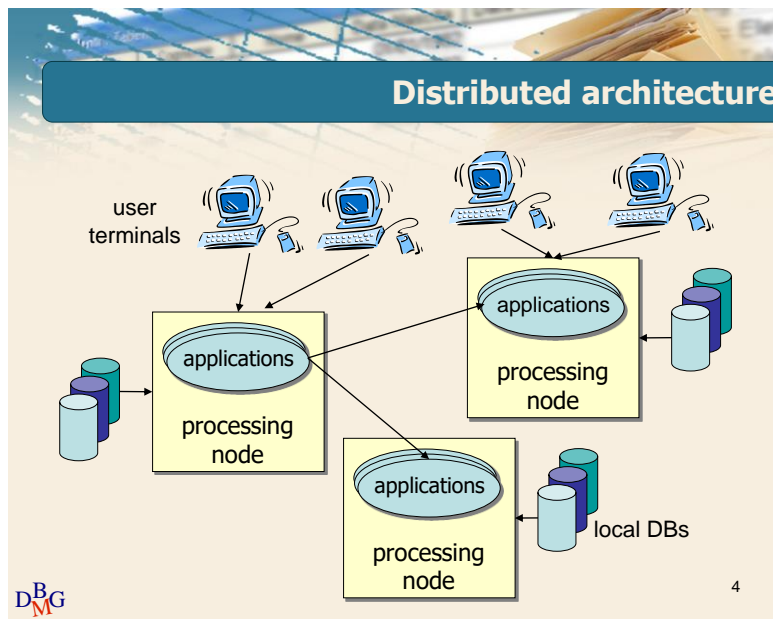
DBG

Distributed architecture

▷ An architecture is distributed when at least one of these two conditions is verified

- **Distributed processing:** applications, cooperating among themselves, reside on more than one processing node
- **Distributed database:** Data are hosted on more than one processing node

Distributed architecture



Distributed architecture

- ▷ It is a set of processing nodes linked in a determined way
 - LAN: Local Area Network
 - WAN: Wide Area Network
- ▷ Each node is able to communicate with some other nodes
- ▷ Programs in execution on each node are able to share their information and to ask for the execution of other programs by other nodes

Distributed architecture

- ▷ Different complexity levels
 - It depends on independence level of nodes
- ▷ Benefits
 - Improved performances
 - Greater availability
 - Improved reliability

Important properties

- ▷ Portability
 - Capacity to move an application from a system to another one
 - Guaranteed by SQL standard
- ▷ Interoperability
 - Different DBMSs are able to cooperate in the execution of a given work
 - Interaction protocols are needed
 - ODBC
 - X-Open-DTP

Distributed architecture

- ▷ Applications are characterized by the role they perform in the system
 - Client: when the application uses services that are made available by other applications
 - Server: when the application provides services used by other applications
 - Actor: when the application assume both roles of the client and of the server, in different contexts

The client-server paradigm

- ▷ Client: service user
- ▷ Server: service supplier
- ▷ Client and server have no meaning without a communication protocol that
 - Defines possible interactions between client and server
 - Specifies details of each interaction
 - Defines error conditions and related actions to perform

Protocols example

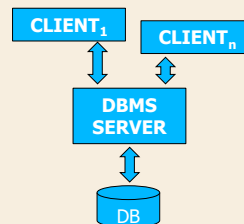
Service	Client	Protocol
World Wide Web	Mozilla Firefox, Internet Explorer, Google Chrome, Opera, ...	Hypertext Transfer Protocol (HTTP)
File transfer	UNIX FTP, MS FTP, Filezilla, browser Web	File Transfer Protocol (FTP)
Electronic mail	Eudora, Outlook, Thunderbird	Simple Mail Transfer Protocol (SMTP)

Client-server architecture

- ▷ Particular case of a distributed system
 - The easiest and most diffused architecture
- ▷ A client sends a request to a server for the execution of a task
 - A task may consist simply on requesting information, or on executing complex elaborations
- ▷ The server can be the client of another service at the same time

Client-server architecture

- ▷ 2-levels architecture (2-Tier)
 - *Thick* client
 - Contains application logic
 - DBMS server
 - Permits data access



Client-server architecture

▷ 3-levels architecture (3-Tier)

- *Thin* client
 - the browser
- Application server
 - It implements application logic
 - It is usually a web server
- DBMS Server
 - Permits data access

```
graph TD; C1[CLIENT_1] <--> AS[APPLICATION SERVER]; Cn[CLIENT_n] <--> AS; AS <--> DBMS[DBMS SERVER]; DBMS <--> DB[(DB)];
```

DBG

13

SQL execution

▷ Compile & Go

- Query is sent to the server
- Query is compiled
 - Generation of the execution plane
- Query is executed
- Result is returned

▷ Strong for the execution of not repeated queries

- Flexible, good for dynamic SQL execution

DBG

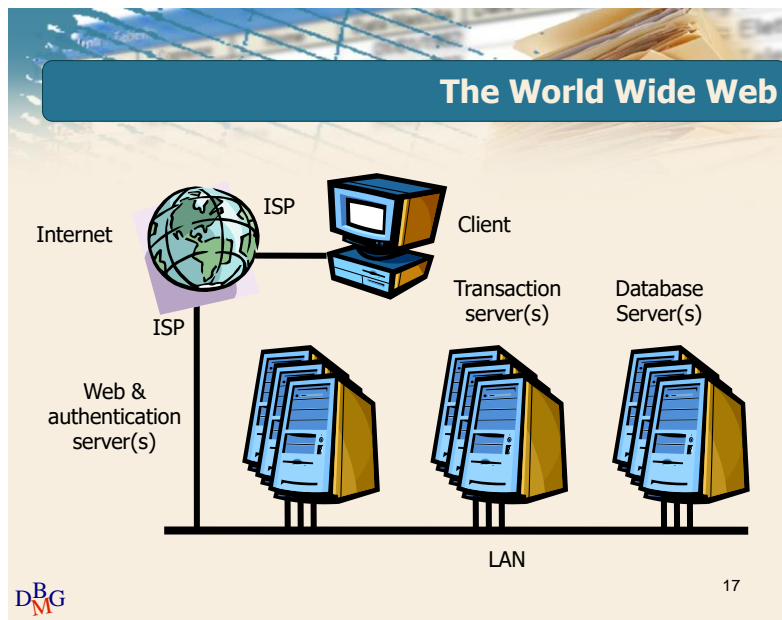
14

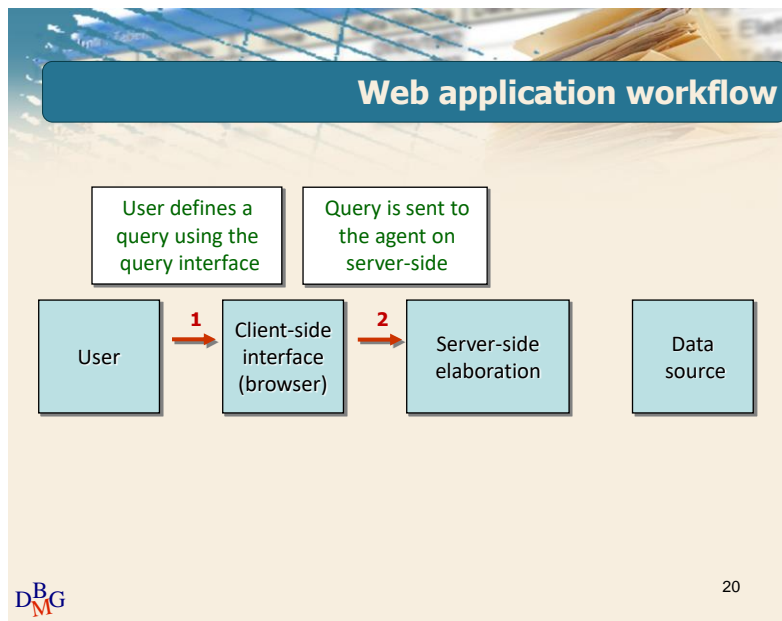
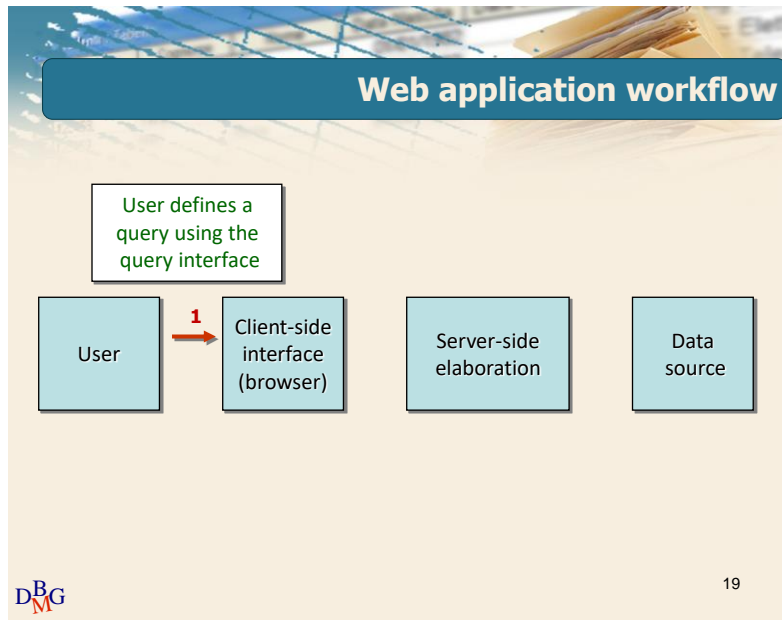
SQL execution

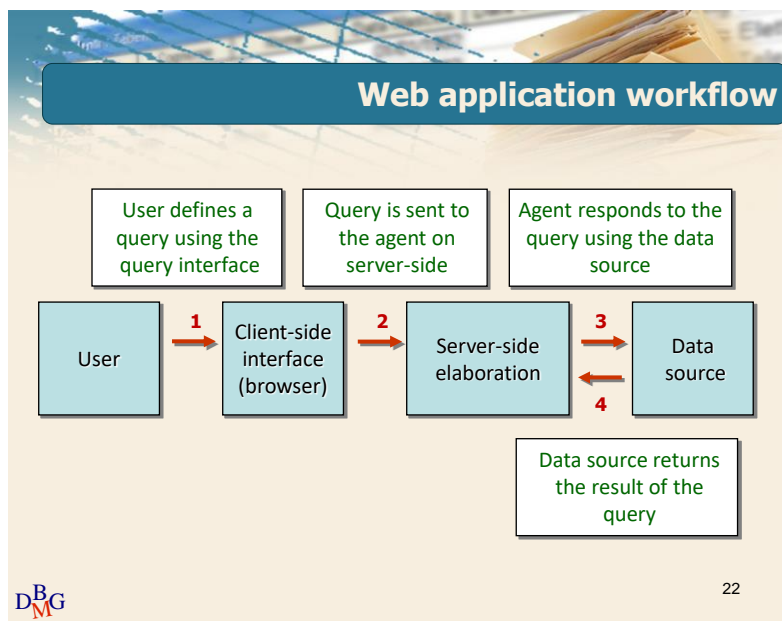
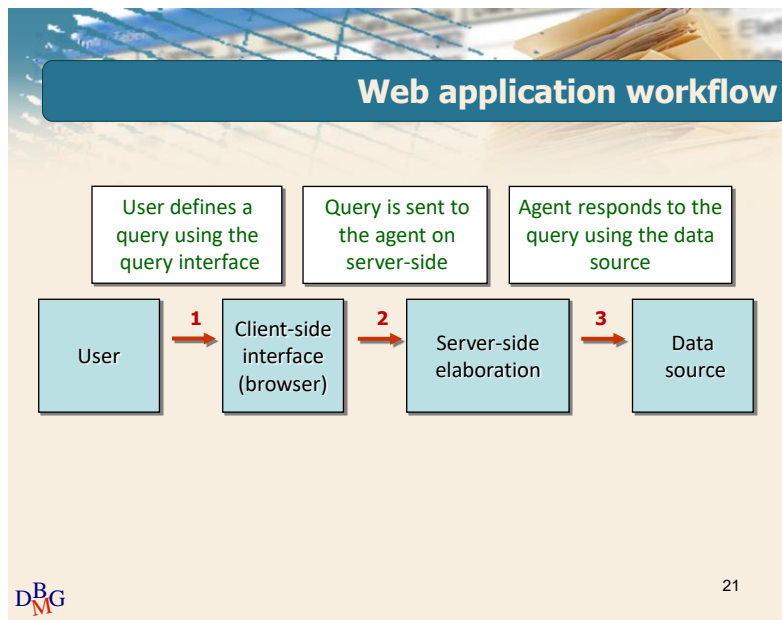
- ▷ Compile & Store
 - Query is sent to the server
 - Query is compiled
 - Generation of the execution plane
 - The execution plane is stored for a future use
 - Query is executed
 - Result is returned
- ▷ Efficient for repeated queries
 - Parametric executions of the same query

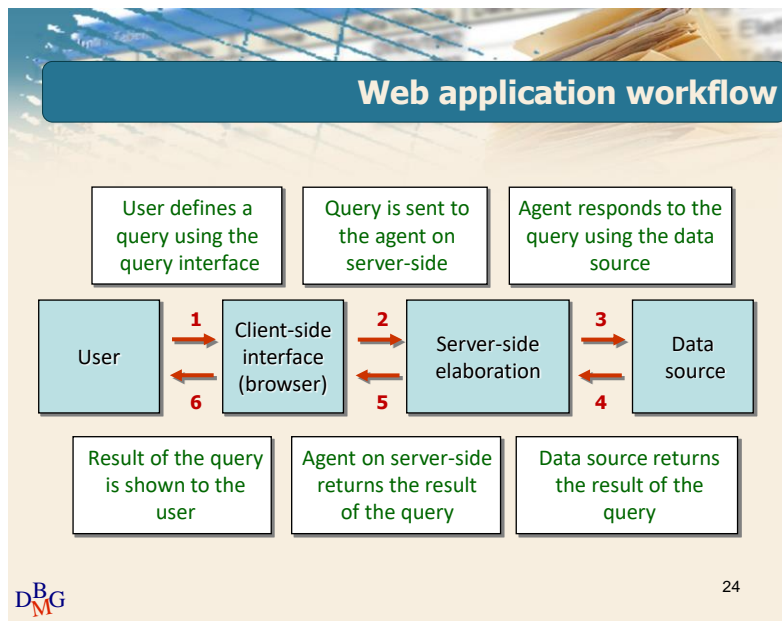
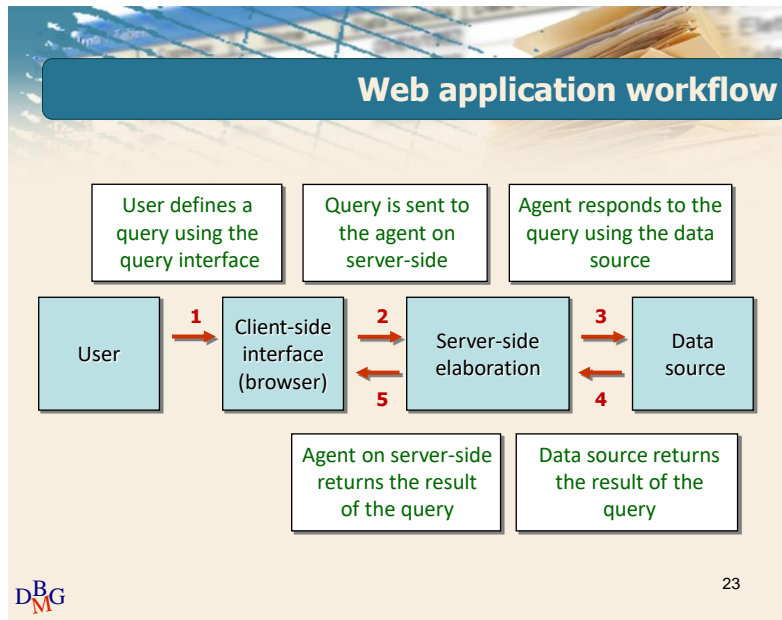
C/S architecture: the World Wide Web

- ▷ The World Wide Web is a graph of documents
- ▷ Server keeps a local set of documents and sends one of them to the client (browser Web) on demand
 - text, picture, video, ...
- ▷ Some types of script ask server for executing a program (e.g. PHP script)









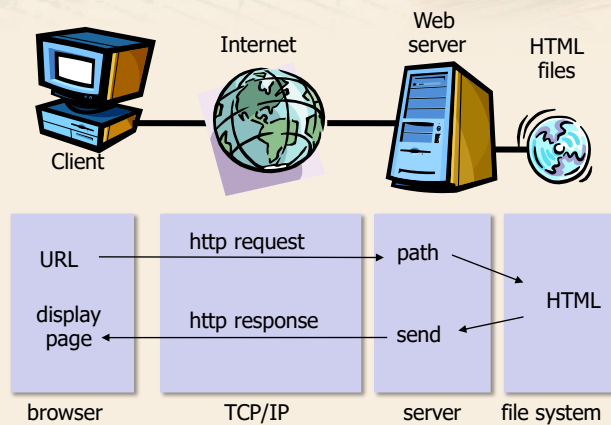
Web application workflow

- ▷ Web applications perform more or less the same tasks:
- Provide an interface for asking informations (query interface);
 - e.g.: search in a database, file request, purchase a book, booking a flight, ...
 - Transmit user data to the Web server
 - Process data on server-side, accessing a database if necessary
 - Transmit query results to the client



25

Static transaction



26

