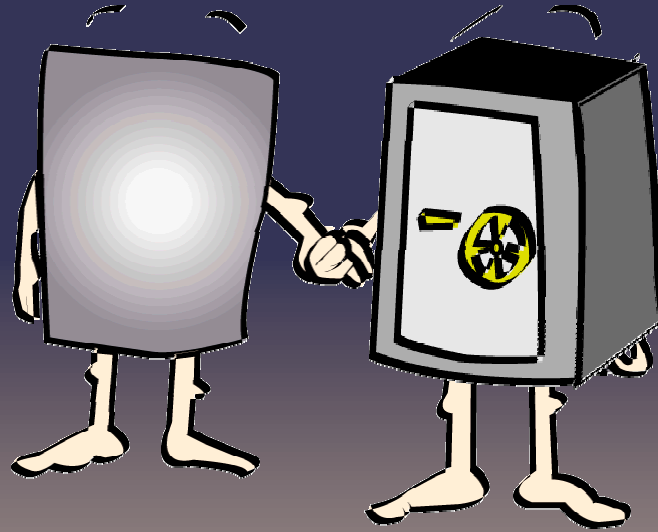


Web e basi di dati

Il Web come Interfaccia Utente di un Sistema Informativo

- Occorre un meccanismo di interazione con il DBMS (attraverso il server Web) per la specifica di query e/o modifiche, es. basate sull'utilizzo di FORM
- Occorre un meccanismo dinamico di generazione delle pagine, il cui contenuto corrisponda ai risultati di una query

Web & basi di dati



Obiettivi:

- ottenere la generazione dinamica di pagine Web a partire da dati contenuti in una base di dati
- sfruttare i pregi di Web e basi di dati, aggirandone i difetti

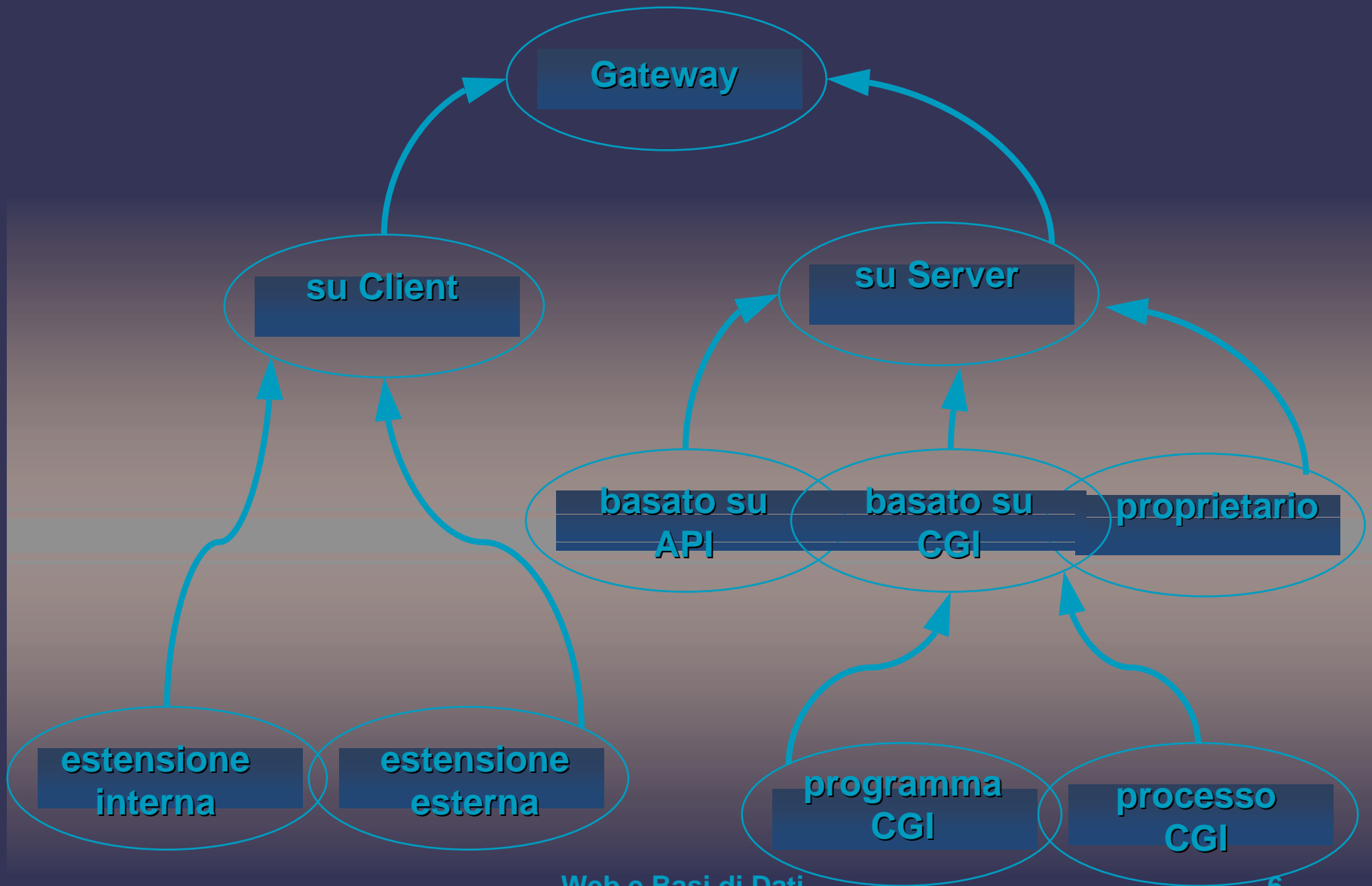
Pregi e difetti di basi di dati e Web

Web	pro	contro
	<ul style="list-style-type: none">• semplice• portabile• a basso costo• indipendente dalle interfacce• ipermediale	<ul style="list-style-type: none">• basato su file• statico
basi di dati	<ul style="list-style-type: none">• modelli dei dati• linguaggi di interrogazione• funzioni di amministrazione	<ul style="list-style-type: none">• complesse• proprietarie• navigazione e presentazione assenti

Gateway Web-base di dati

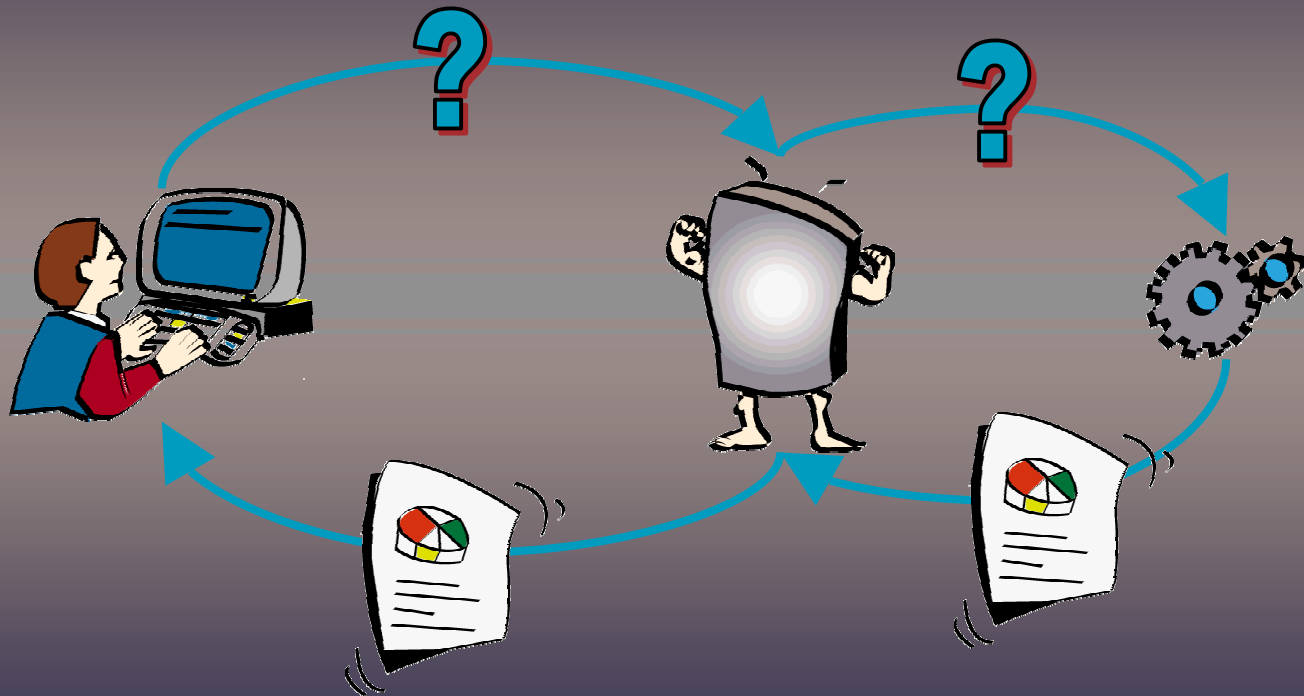


Una gerarchia di soluzioni



Common Gateway Interface

- Protocollo che consente al Web Server di eseguire applicazioni esterne in grado di creare pagine dinamicamente



Caratteristiche di CGI

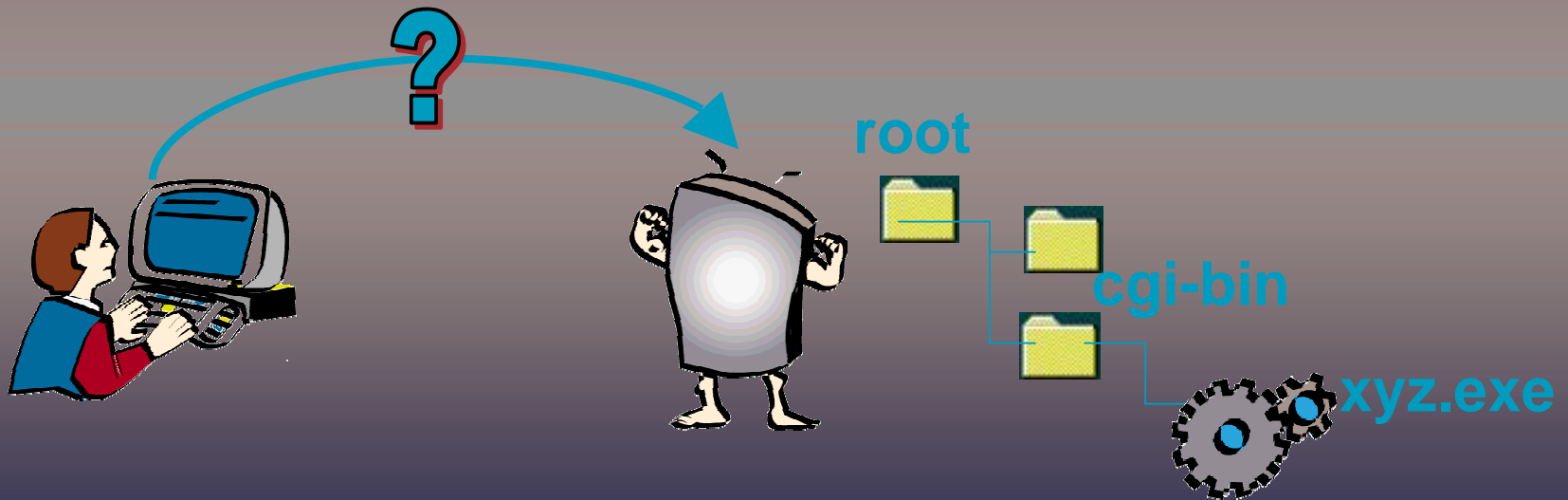
- **Non è:**
 - un linguaggio di programmazione
 - un protocollo di comunicazione
- **Definisce solo un insieme di variabili di ambiente utili alla applicazione (es. parametri inviati dal client)**



Invocazione

- Il cliente specifica nell'**URL** il nome del programma da eseguire
- Il programma deve stare in una posizione precisa (di solito il direttorio **cgi-bin**)

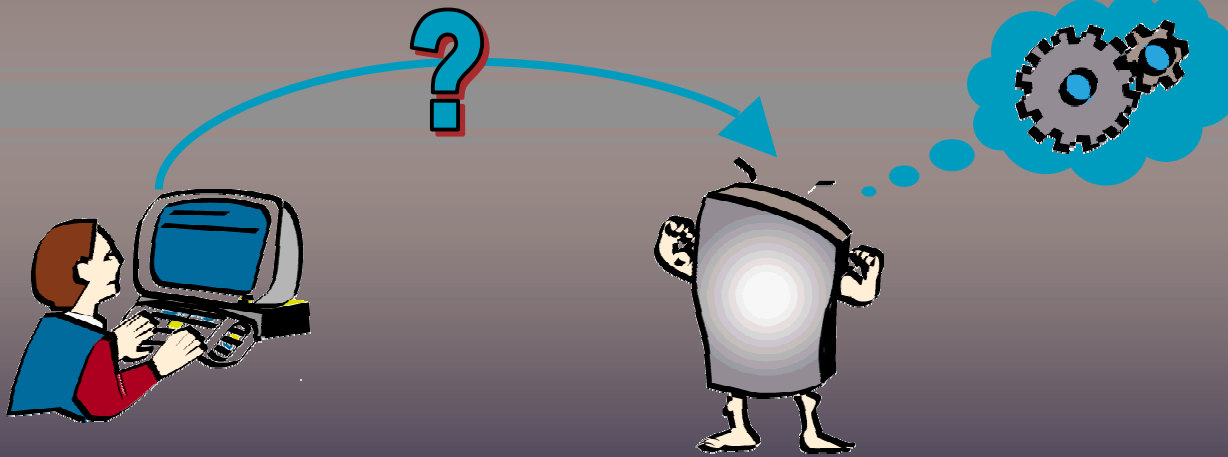
<http://mio.server.web/cgi-bin/xyz.exe>



Esecuzione

1. Il server **riconosce** dall'URI che la risorsa richiesta dal cliente e' un **eseguibile**

`http://mio.server.web/cgi-bin/xyz.exe`

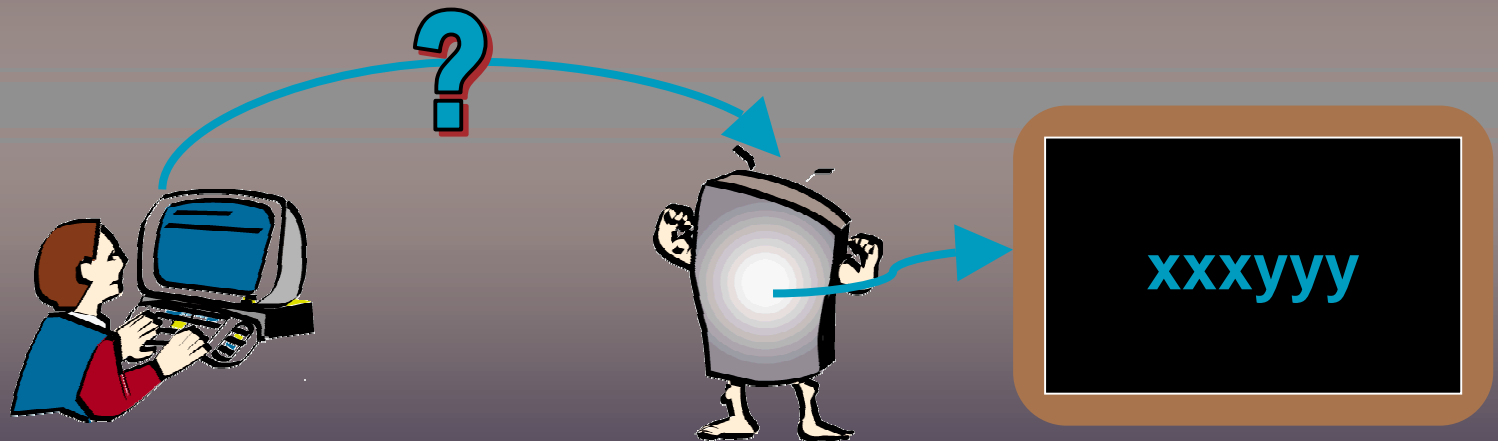


Esecuzione

2. Il server **decodifica** i parametri inviati dal cliente e riempie le variabili d'ambiente

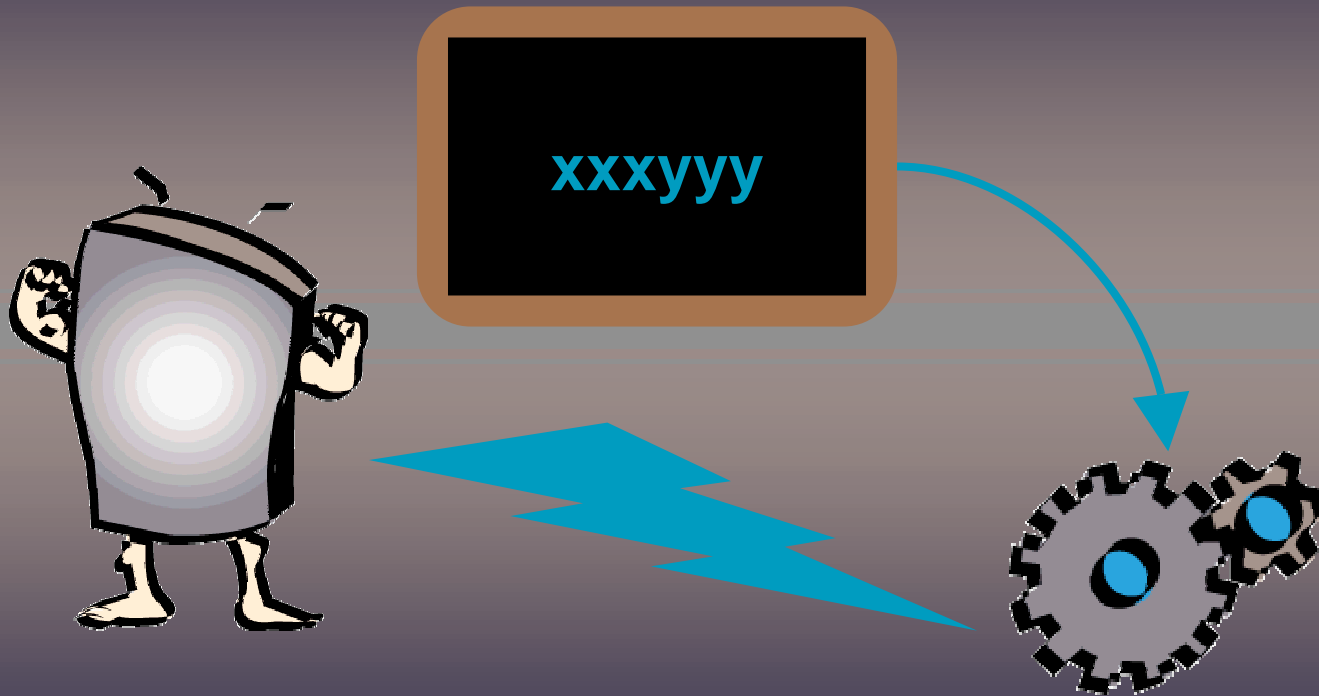
es: `request_method`, `query_string`,
`content_length`, `content_type`

`http://mio.server.web/cgi-bin/xyz.exe?xxxyyy`



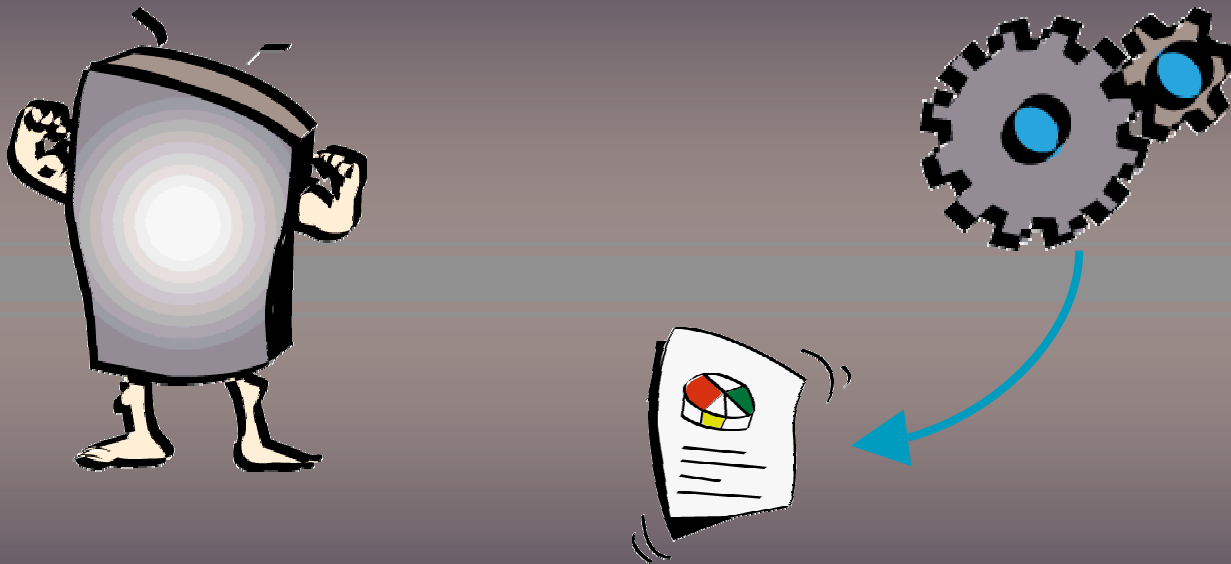
Esecuzione

3. Il server **lancia** in esecuzione l'applicazione richiesta



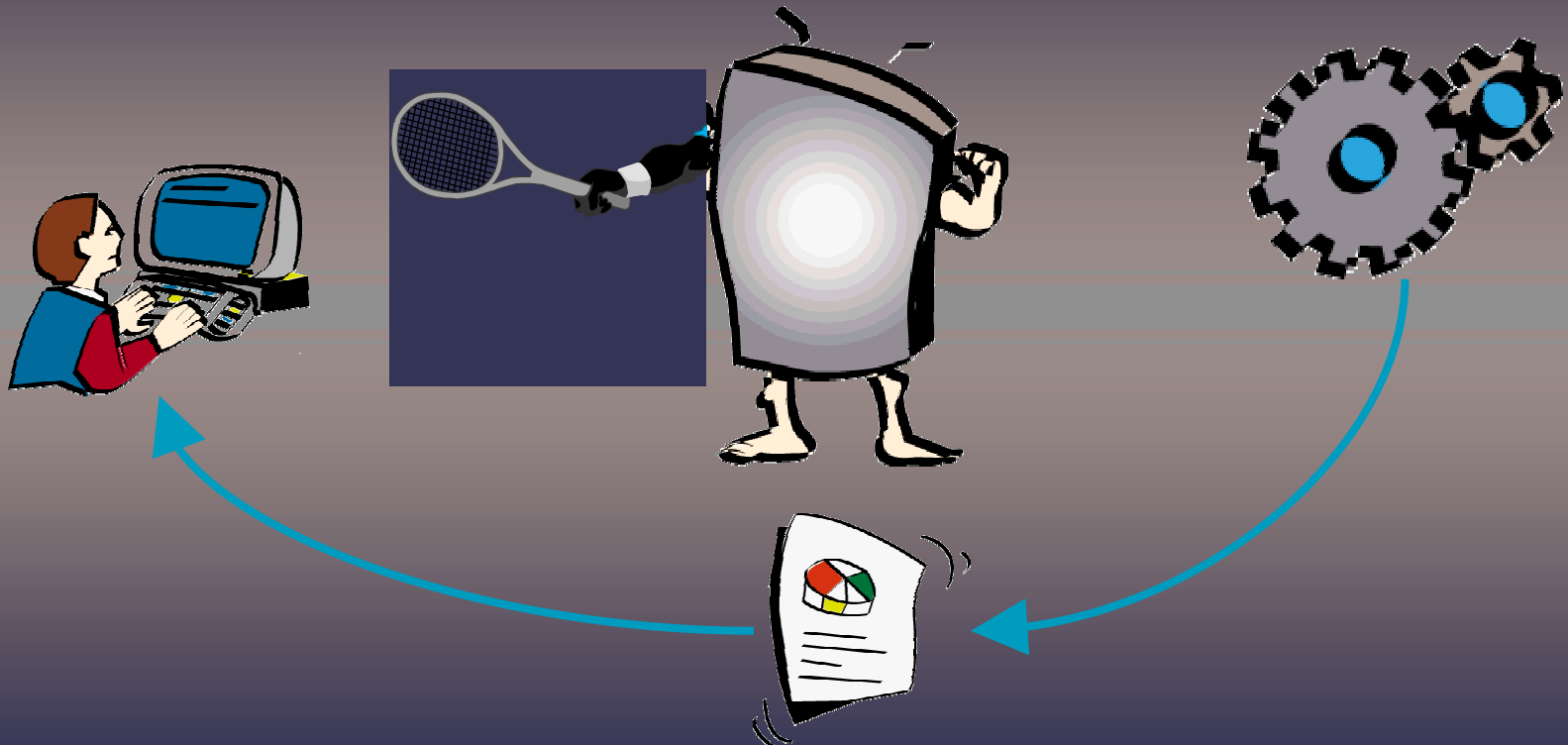
Esecuzione

4. L'applicazione **stampa** la sua risposta sullo standard output



Esecuzione

5. Il server **redireziona** lo standard output sulla rete e quindi verso il client



Invio di parametri a un programma CGI

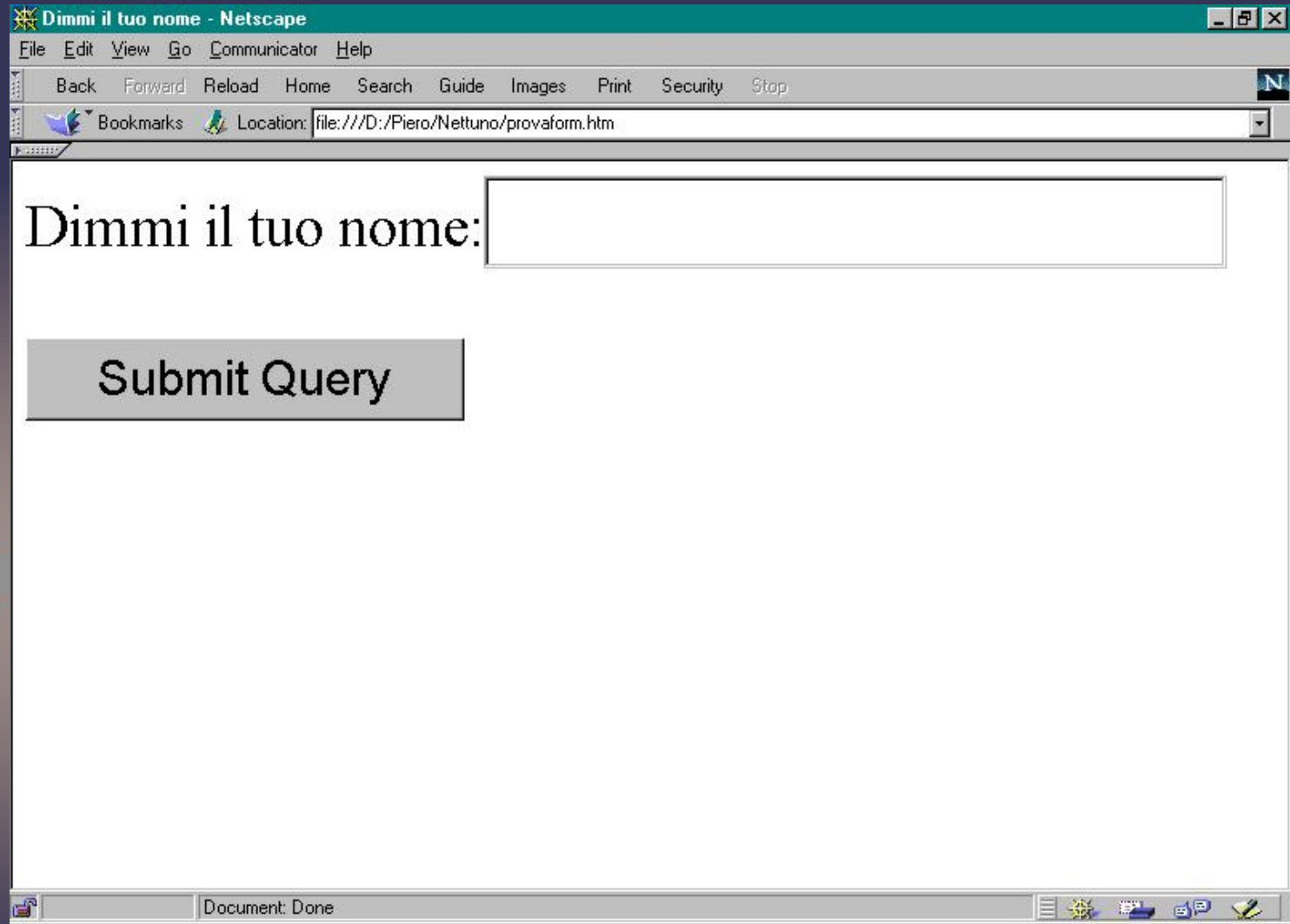
- Il client puo' usare due metodi:
 - GET
 - POST
 - GET: i parametri sono codificati nell'URL
`http://www.mioserver.it/cgi-bin/xyz?par=val`
 - POST: i parametri sono spediti al server separatamente, usando il body del messaggio di richiesta HTTP
- NB:** il metodo POST richiede l'uso di un costrutto HTML chiamato FORM

FORM HTML

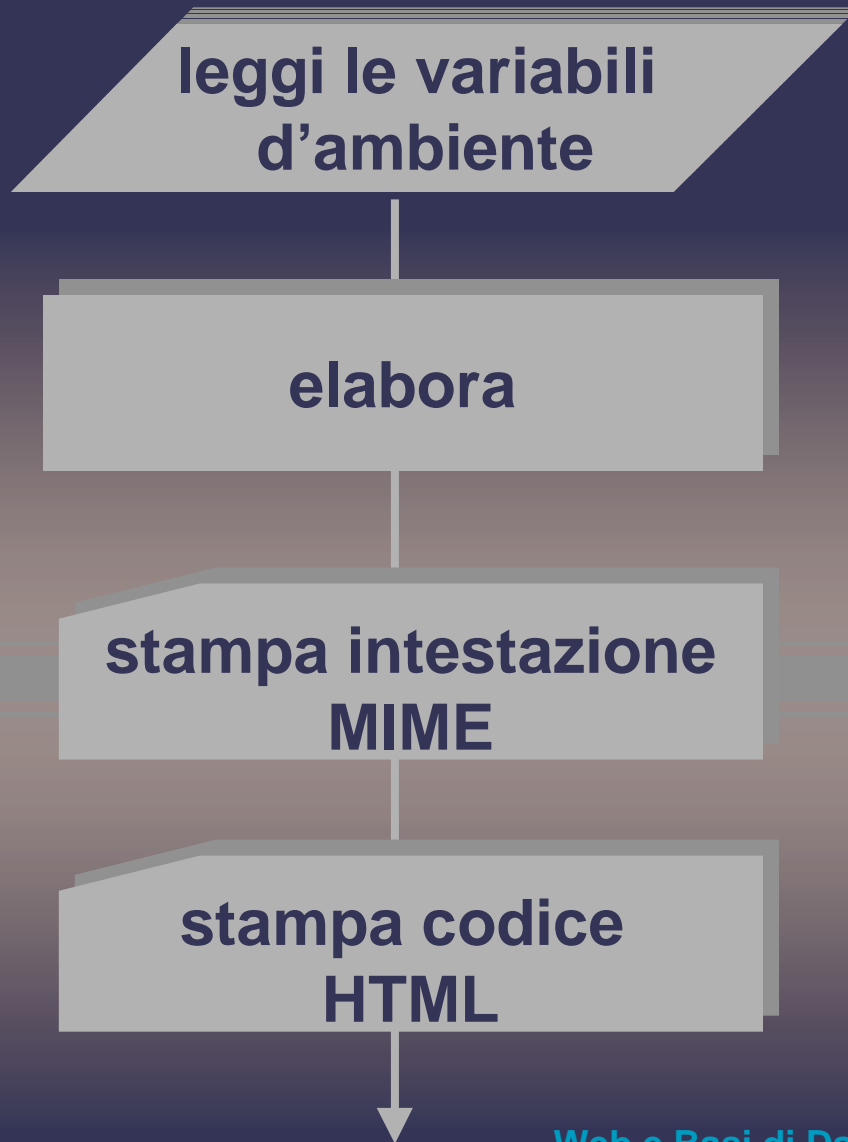
Esempio: invio al server il nome dell'utente

```
<form  
  action=http://www.mysrvr.it/cgi-bin/xyz.exe  
  method=post>  
<p>Dimmi il tuo nome:  
<input type=text name="chisei" ></p>  
<input type=submit >  
</form>
```


FORM HTML



Struttura di un programma CGI



```
cout <<  
"Content-type: text/html"  
<< endl << endl;
```

Decodifica dei parametri

leggi la variabile
Request_method

Valore?

GET

POST

leggi la variabile
Query_string

leggi la variabile
content_length

leggi content_length
caratteri da
standard input

Applicazioni eseguibili via CGI

- Programmi “tradizionali” compilati (es. scritti in C, C++, Java...)
- “Script” compilati e/o interpretati
 - PERL
(Practical Extraction and Report Language)