



Protocolo HTTP

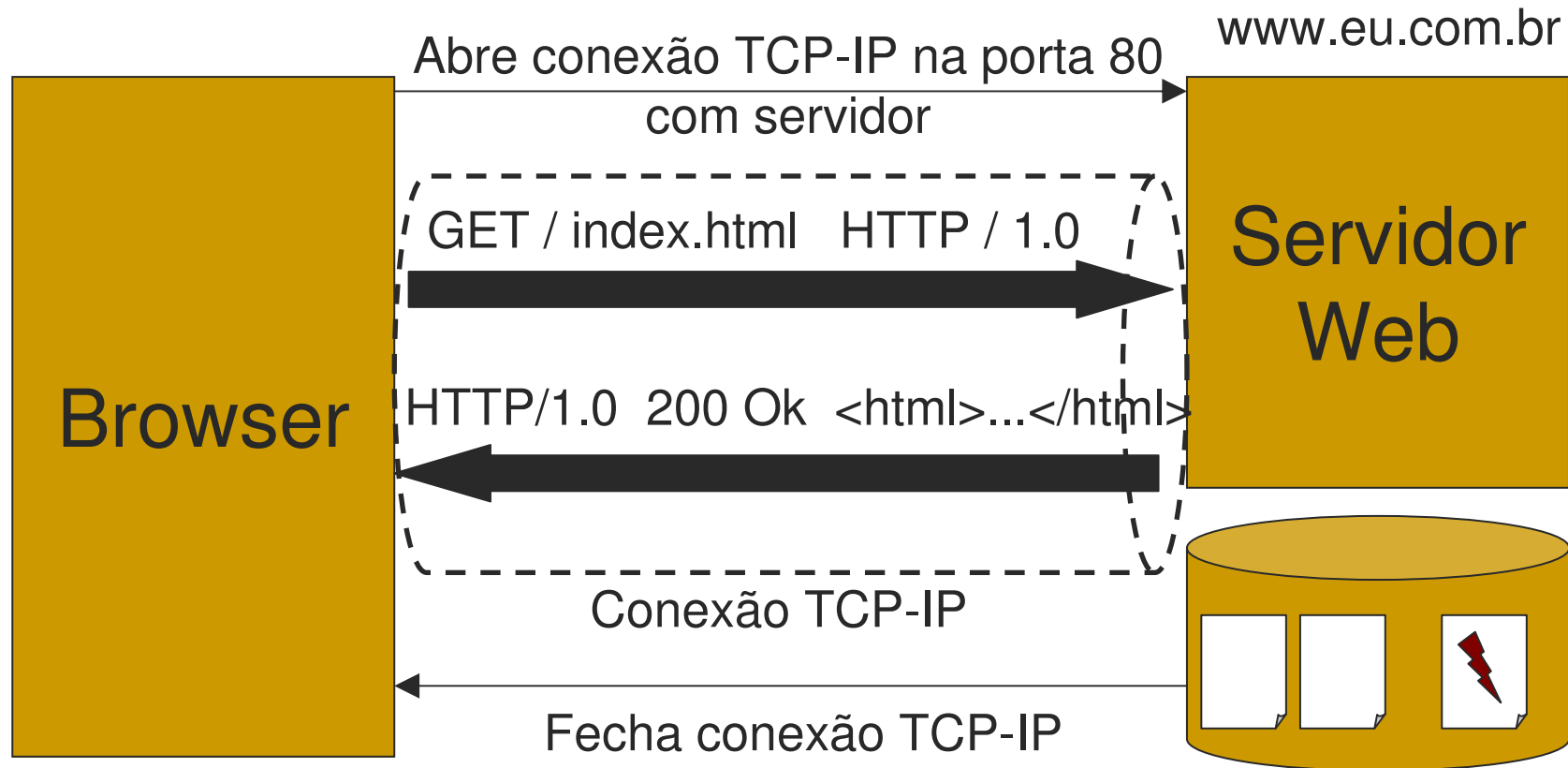
- Características
 - Modelo Requisição/Resposta
- Common Gateway Interface (CGI)

[Características]

- **Hypertext Transfer Protocol (HTTP)**
 - Protocolo utilizado para transferir documentos de hipertexto e seus recursos de máquinas remotas.
- **Características gerais**
 - Protocolo da camada de aplicação;
 - Funciona baseado na troca de requisição-resposta;
 - Cabeçalho das mensagens é texto puro (não binário);
 - Não orientado a conexões;
 - Não guarda estado entre conexões distintas, isto é, cada conexão é nova para o servidor.

[Características]

- Exemplo (<http://www.eu.com.br/index.html>)



[Requisição HTTP]

- Sintaxe geral de uma requisição HTTP (RFC-822):

```
comando /uri-recurso HTTP/versão  
{cabeçalho: valor}n  
<linha em branco>  
dados
```

- A primeira linha é chamada linha de comando
- Podem haver vários cabeçalhos de requisição
- Alguns comandos HTTP: **GET**, **POST**, HEAD, PUT, DELETE, outros

[Requisição HTTP]

■ Comando **GET**

- Solicita ao servidor um recurso (página HTML, figura GIF, documento do word, etc.)
- A URI do recurso pode conter dados separados por '?'
- Exemplo: /index.html?dado1=valor1&dado2=valor2

■ Comando **POST**

- Usado para enviar dados para o servidor (p.e., upload de arquivo, dados de formulário HTML)
- Ao contrário do GET, os dados seguem após a linha em branco que separa cabeçalhos dos dados

[Requisição HTTP]

- Exemplo (<http://www.eu.com.br/index.html>)



[URIs e URLs]

- URI = URL + URN

- URI = Identificador Uniforme de Recursos
- URL = Localizador Uniforme de Recursos
- URN = Nome Uniforme de Recurso

- Sintaxe geral de uma URL:

`<protocolo>://<servidor>:<porta>/<caminho>/<recurso>`

- A porta é opcional para serviços em portas *default*
- Caminho e recurso podem ser omitidos (URLs parciais)
- URLs podem conter dados depois do nome do recurso

[URIs e URLs]

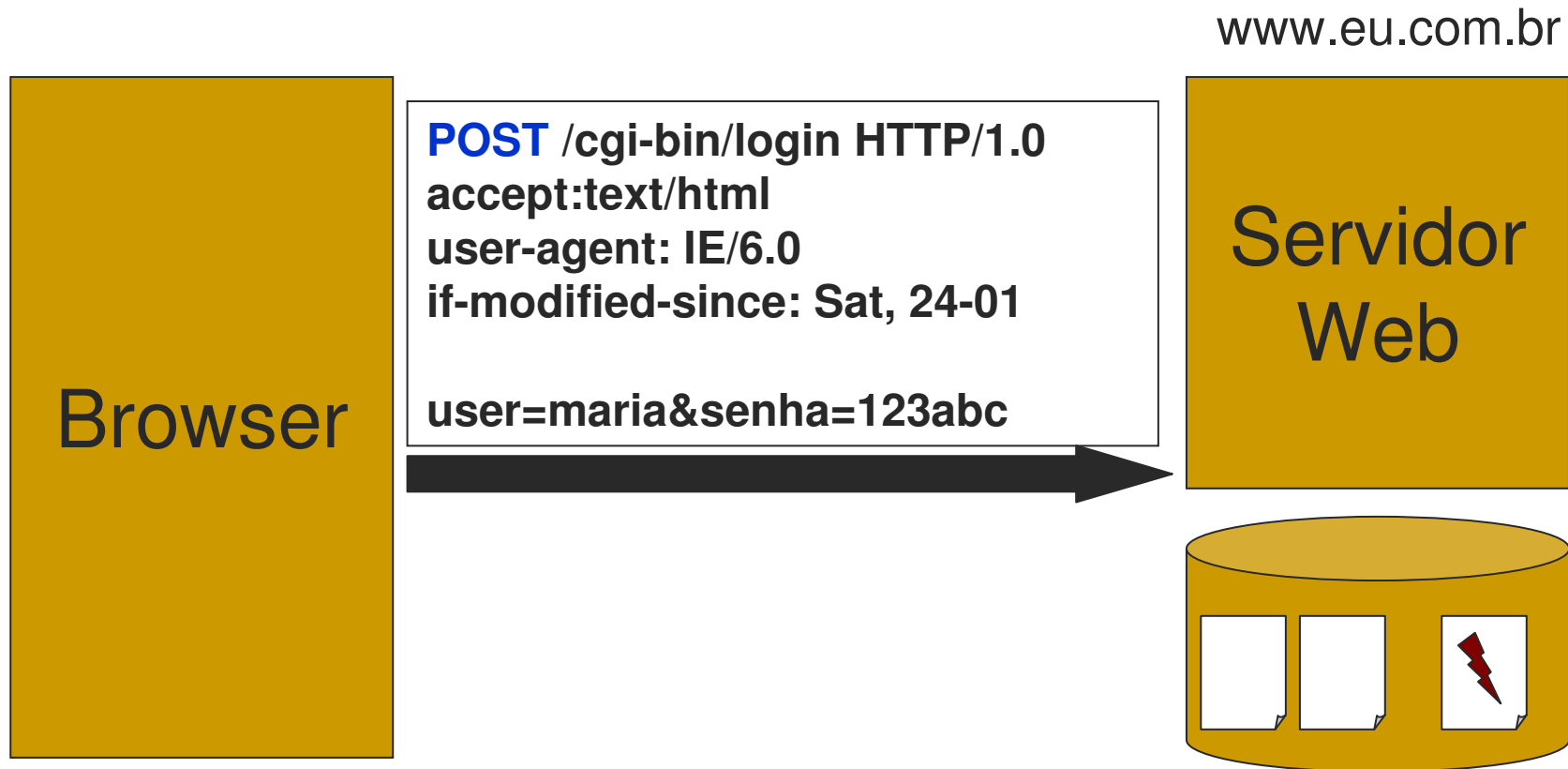
- Exemplos de URLs:
 - <http://java.sun.com/docs/servlets/servlets.html>
 - <http://java.sun.com/docs/servlets/>
 - <http://java.sun.com/cgi-bin/reverse?string=fred>
 - <http://localhost:8080/fred/servlets/ListaServlet?tipo=superior&curso=334>

[Formato x-www-form-urlencoded]

- Formato para transferência de dados no HTTP
- Usado para codificar os dados produzidos pelo cliente
- Sintaxe
 - Não pode haver nenhum espaço em branco nos dados
 - Dados são agrupados em pares **nome=valor**
 - Pares são separados por **&**
 - Espaços em branco são codificados com **+**
 - Caracteres de 8 bits são codificados com **%HH**, onde HH é o código hexadecimal do caractere

[Requisição HTTP]

- Exemplo (<http://www.eu.com.br/cgi-bin/login>)



[Resposta HTTP]

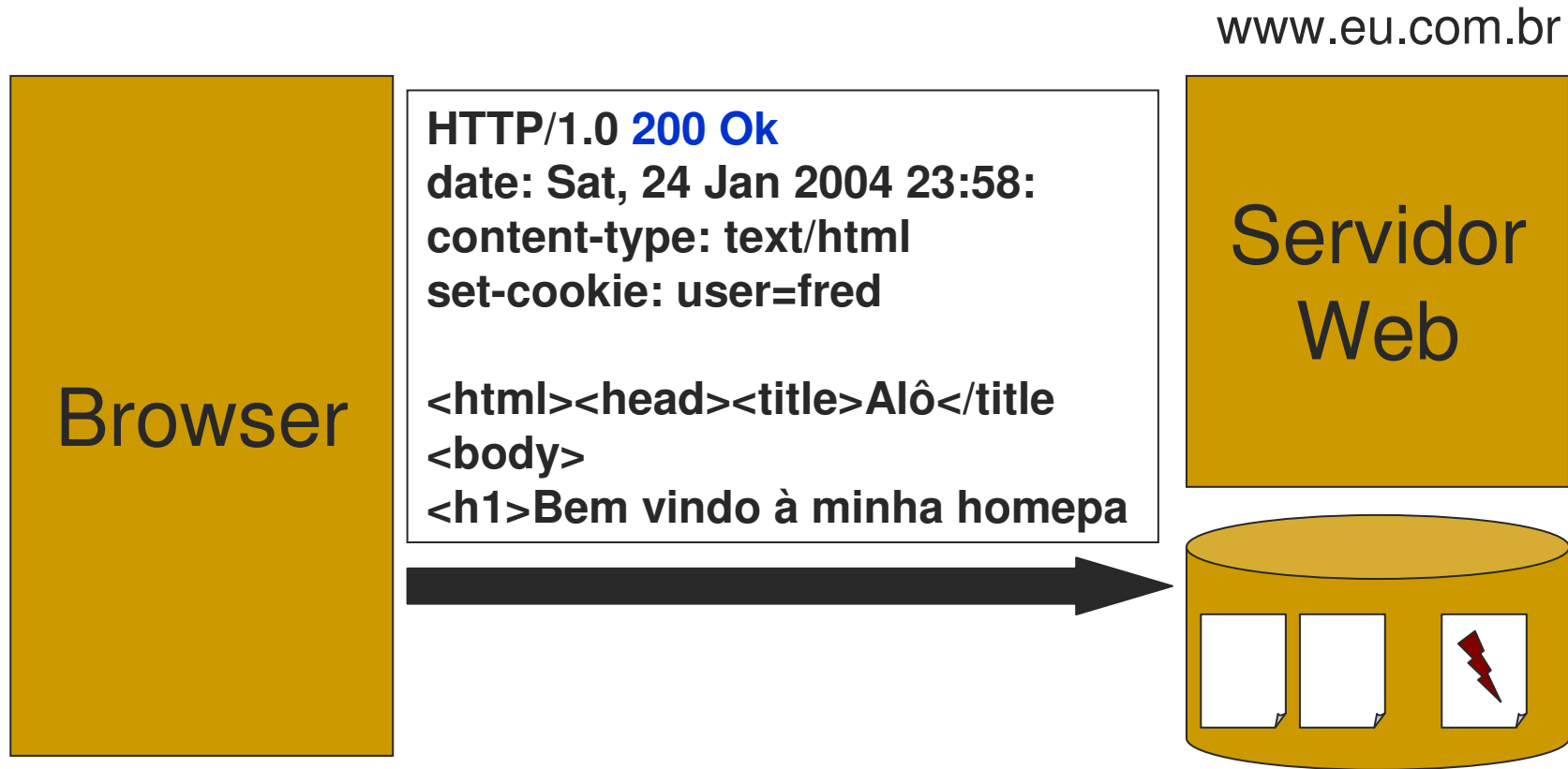
- Sintaxe geral de uma resposta HTTP (RFC-822):

```
HTTP/versão código mensagem  
{cabeçalho: valor}n  
<linha em branco>  
dados
```

- A primeira linha é chamada linha de status
- Podem haver vários cabeçalhos de resposta
- Os dados podem ser texto (página HTML) ou binário (figura GIF)

[Resposta HTTP]

- Exemplo (<http://www.eu.com.br/index.html>)



[Resposta HTTP]

- Toda resposta HTTP deve ter um cabeçalho **content-type**
 - Ele indica o tipo MIME do recurso enviado
 - Protocolo HTTP é multimídia
- Códigos de status da resposta:
 - 1xx: Mensagem de informação
 - 2xx: Sucesso no atendimento à requisição
 - 3xx: Redirecionamento para outro servidor/recurso
 - 4xx: Erro no atendimento ao cliente
 - 5xx: Erro do servidor Web

Resposta HTTP

A Sample FORM using POST - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Media Mail Print

Address: C:\Documents and Settings\redjmy\Documents\%o\Pictures\jao\%P\amece's\post1.cfm.htm

Formulário HTML

Item Number:

Quantity:

Price Each: \$

First Name:

Last Name:

Middle Initial:

Shipping Address:

Credit Card:

☐ Visa

☐ MasterCard

☐ American Express

☐ Discover

☐ Other Credit Card

Credit Card Number:

Repeat Credit Card Number:

Done My Computer

inum = 11

qty = 1

fname = Maria

lname = Joao

minit = C. G.

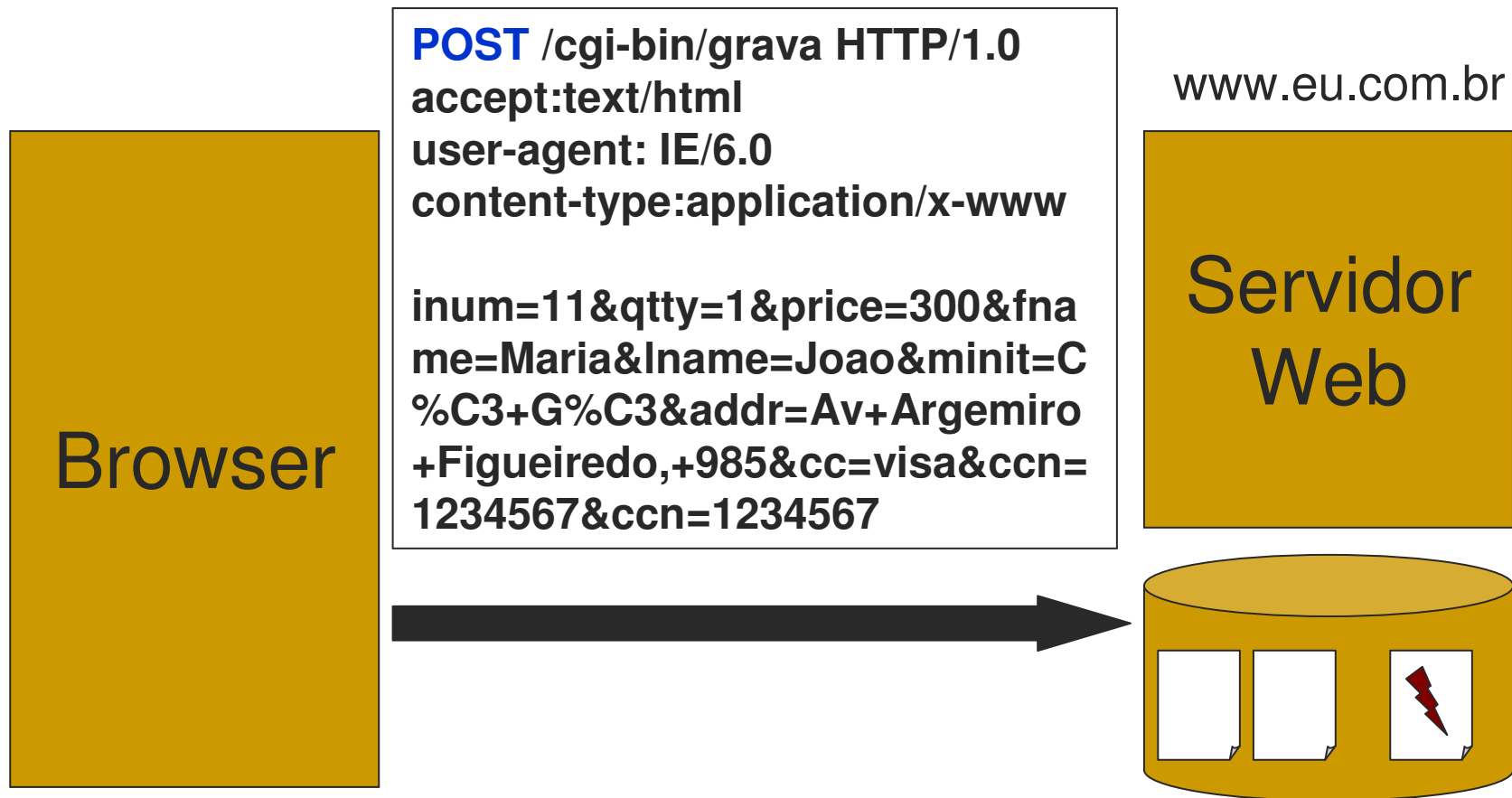
addr = Av Argemiro de Figueiredo, 985

cc = Visa

ccn = 1234567

ccn = 1234567

[Resposta HTTP (Exemplo)]



[Resposta HTTP]

```
<form method="POST"
action="http://www.abc.com/cgi-bin/grava">
Item Number: <input type="text"
name="inum"><br>
Quantity: <input type="text"
name="qty"><br>
... <!-- outros elementos -->
<input type="submit" value="Enviar" />
</form>
```

POST /cgi-bin/grava HTTP/1.0

accept:text/html

user-agent: IE/6.0

content-type:application/x-www

**inum=11&qty=1&price=300&fname=Maria&lname=Joao&
minit=C%C3+G%C3&addr=Av+Argemiro
+Figueiredo,+985&cc=visa&ccn=1234567&ccn=1234567**

[Common Gateway Interface]

- O servidor web não é apenas um servidor de arquivos
- É possível solicitar ao servidor web a **execução** de programas
- CGI é uma especificação que diz como estes programas devem ser implementados para poderem ser executados pelo servidor web
 - Como recebem os dados enviados pelo cliente?
 - Como devem gerar a resposta?
 - Como acessam informações passadas pelo servidor?

[Common Gateway Interface]

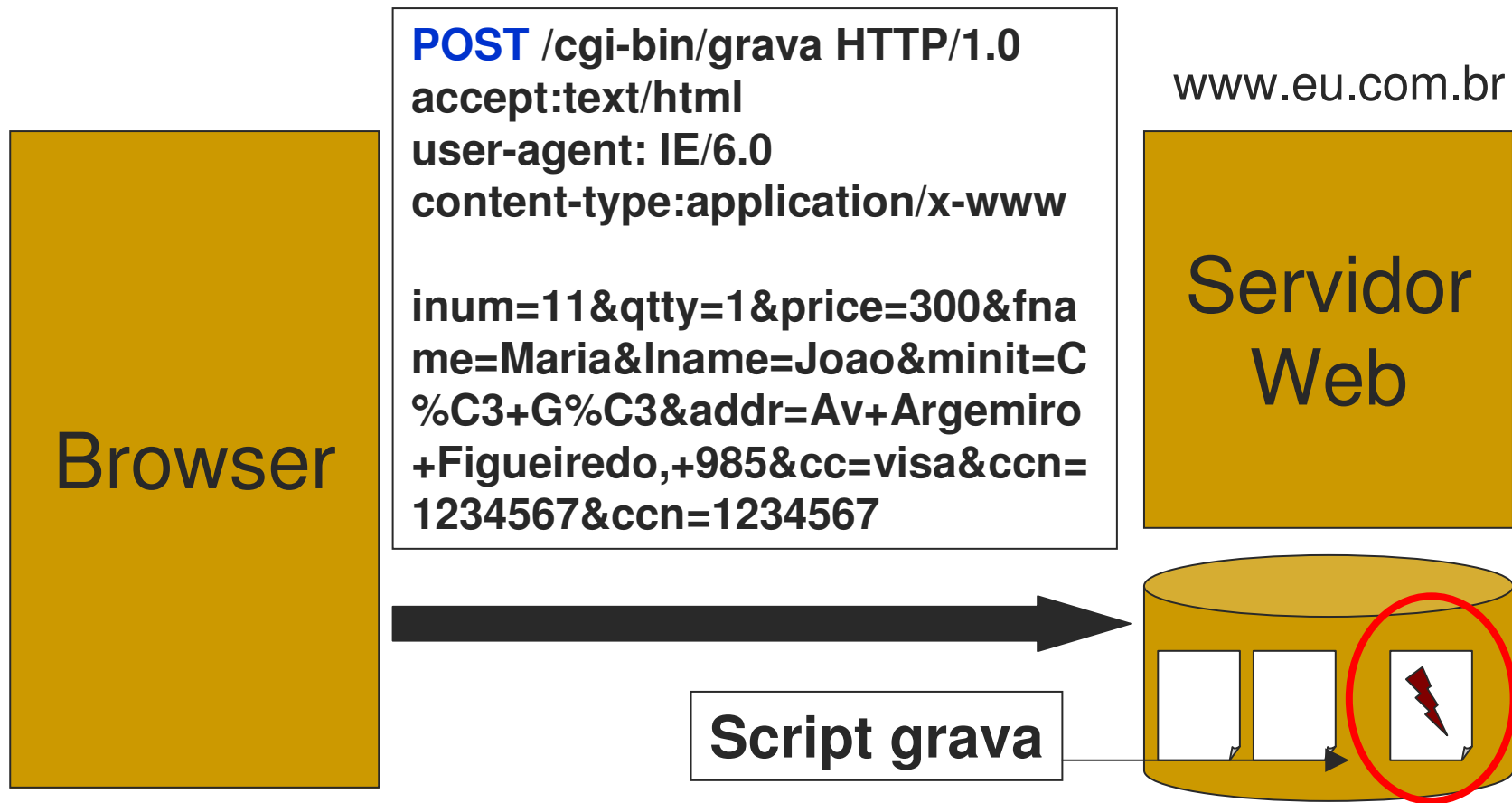
■ Aplicações

- Geração de documentos com conteúdo dinâmico
- Produzir resultados em *sites* de busca
- Gravar dados em arquivos ou SGBDs no servidor

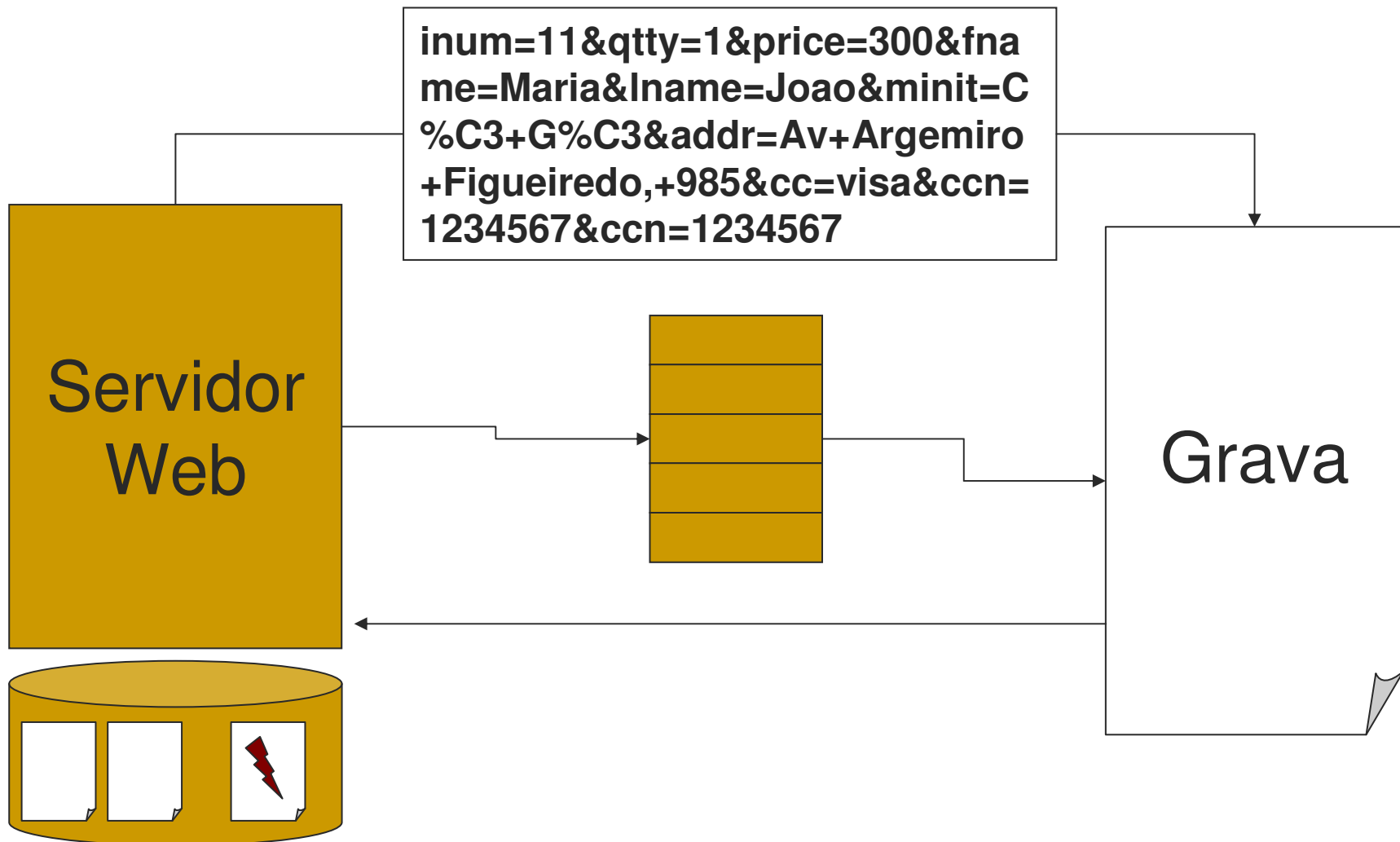
■ Forma

- O programa executa no lado do servidor
- Pode ser escrito em qualquer linguagem de programação
- Cada requisição pelo programa cria um novo processo idêntico na memória (ineficiente)

[Common Gateway Interface]



[Passando dados com POST]



[Passando dados com GET]

