

DESENVOLVIMENTO WEB 2

Jair C Leite



Ementa

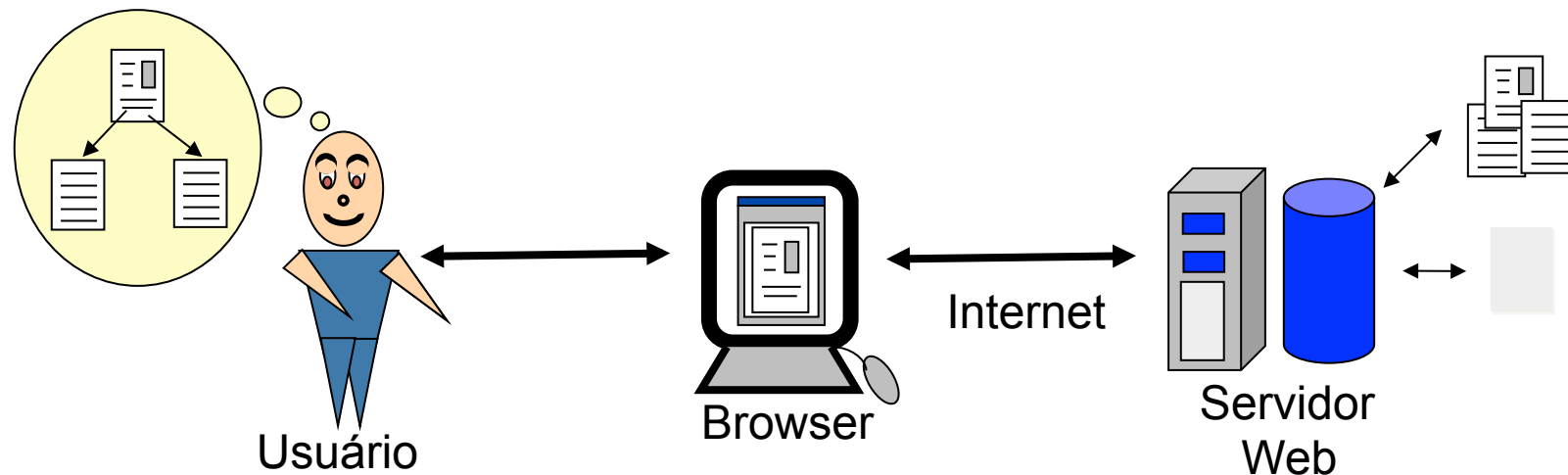
- Introdução ao Desenvolvimento Web no Cliente.
- Fundamentos de HTML5.
- Linguagem de Estilo (CSS).
- Introdução a linguagem Javascript e Ajax.
- Biblioteca JQuery.
- Padroes e Boas praticas para o desenvolvimento de aplicações de Internet Ricas (RIA).
- Arquitetura de Serviços Web.
- Programação avançada de serviços SOAP.
- Programação avançada de serviços Rest.
- Segurança de serviços Web.
- Analise de Desempenho de serviços Web.
- Programando Serviços Web para dispositivos moveis.
- Uso de serviços web nas empresas.

Programa

1. Introdução – Sistemas Web: conceitos e tecnologias – 24/8
2. Tecnologias fundamentais para o lado cliente – 5/9 e 12/9
3. Web Design – 12/9
 1. Conceitos, princípios e padrões
 2. Avaliação de UX
4. Aplicações de Internet Ricas (RIA) – 19/9 e 26/9
 1. Frameworks front-end
 2. SPA – Single Page Applications
 3. Responsive design
5. Serviços Web – 17/10 e 24/10
 1. Conceitos e Arquitetura
 2. Programação
 3. Desempenho e Segurança

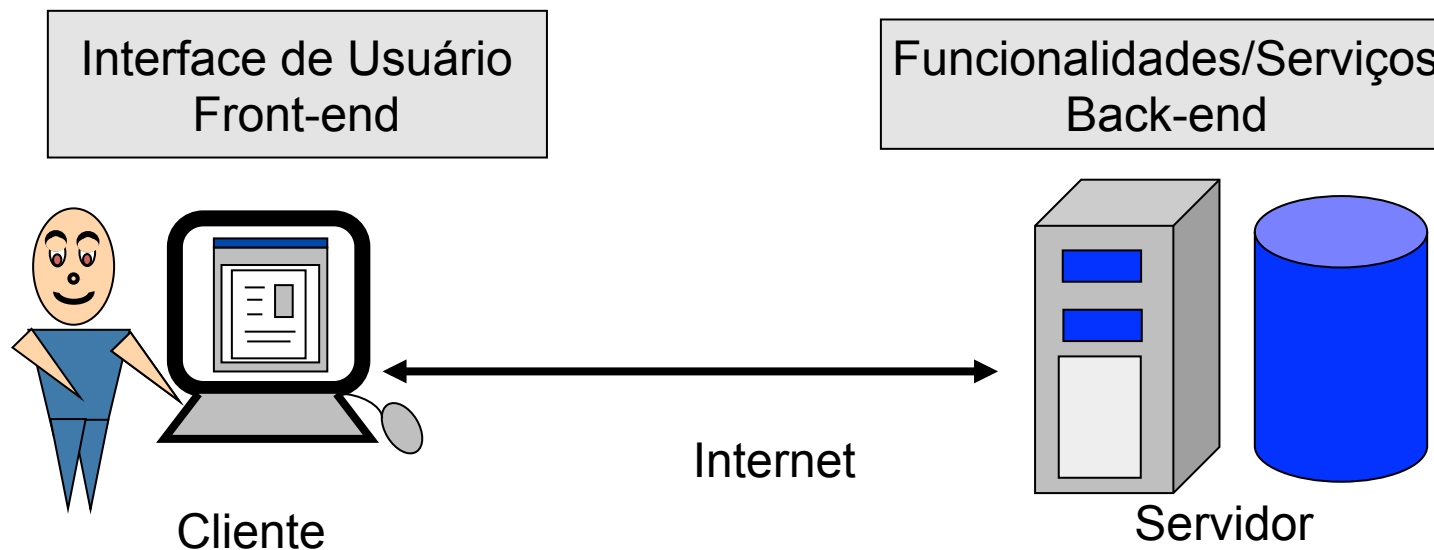
Sistema Web

- Sistema computacional implementado na plataforma Web
- Conjunto de software e hardware necessários para o seu funcionamento
- Do ponto de vista do usuário é o *Web Site* ou Aplicação ou Siste



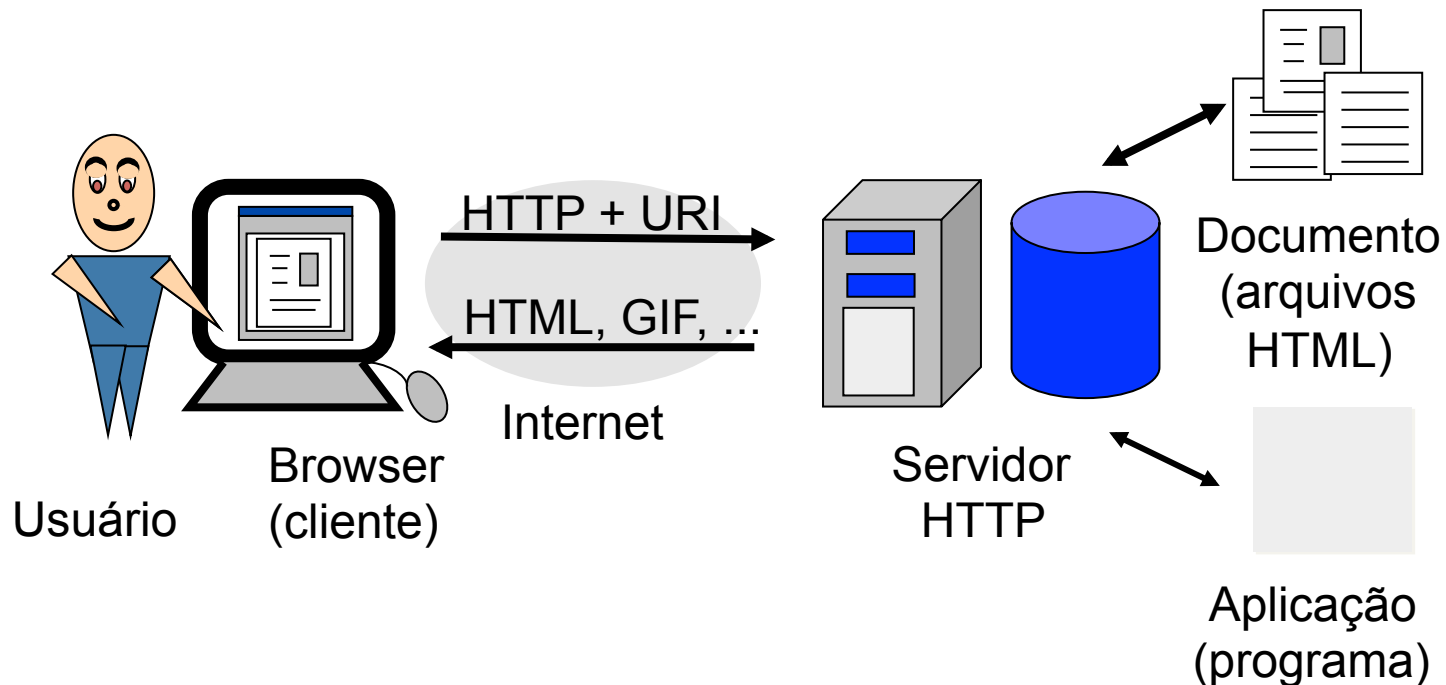
Sistemas Web – cliente e servidor

- A interface de usuário – front-end – responsável pela visualização e envio de informações no lado cliente
- Funcionalidades/Serviços – back-end – responsáveis pelo processamento destas informações, consulta a dados e pela geração dinâmica da interface no lado servidor
- Cliente e servidor se comunicam pela Internet



Modelos e tecnologias principais

- Arquitetura Cliente-Servidor
- Hipertexto
- Tecnologias HTTP, URI, HTML



Tim Berners-Lee

- Cientista da computação
- Inventor da WWW
 - proposta de um sistema de compartilhamento de informações para os cientistas do CERN
 - Implementou um sistema cliente-servidor usando o protocolo HTTP
- Diretor do W3C
- Idealizador da Web Semântica



Tecnologias básicas originais

- *HyperText Markup Language* (HTML)
 - Linguagem que permite definir a estrutura de um documento a ser exibido por um browser
- *Uniform Resource Identifiers* (URI)
 - Esquema pelo qual os recursos da internet são endereçados
- *HyperText Transfer Protocol* (HTTP)
 - Protocolo que define a interação entre um browser (cliente) e um servidor de documentos hipertextuais.

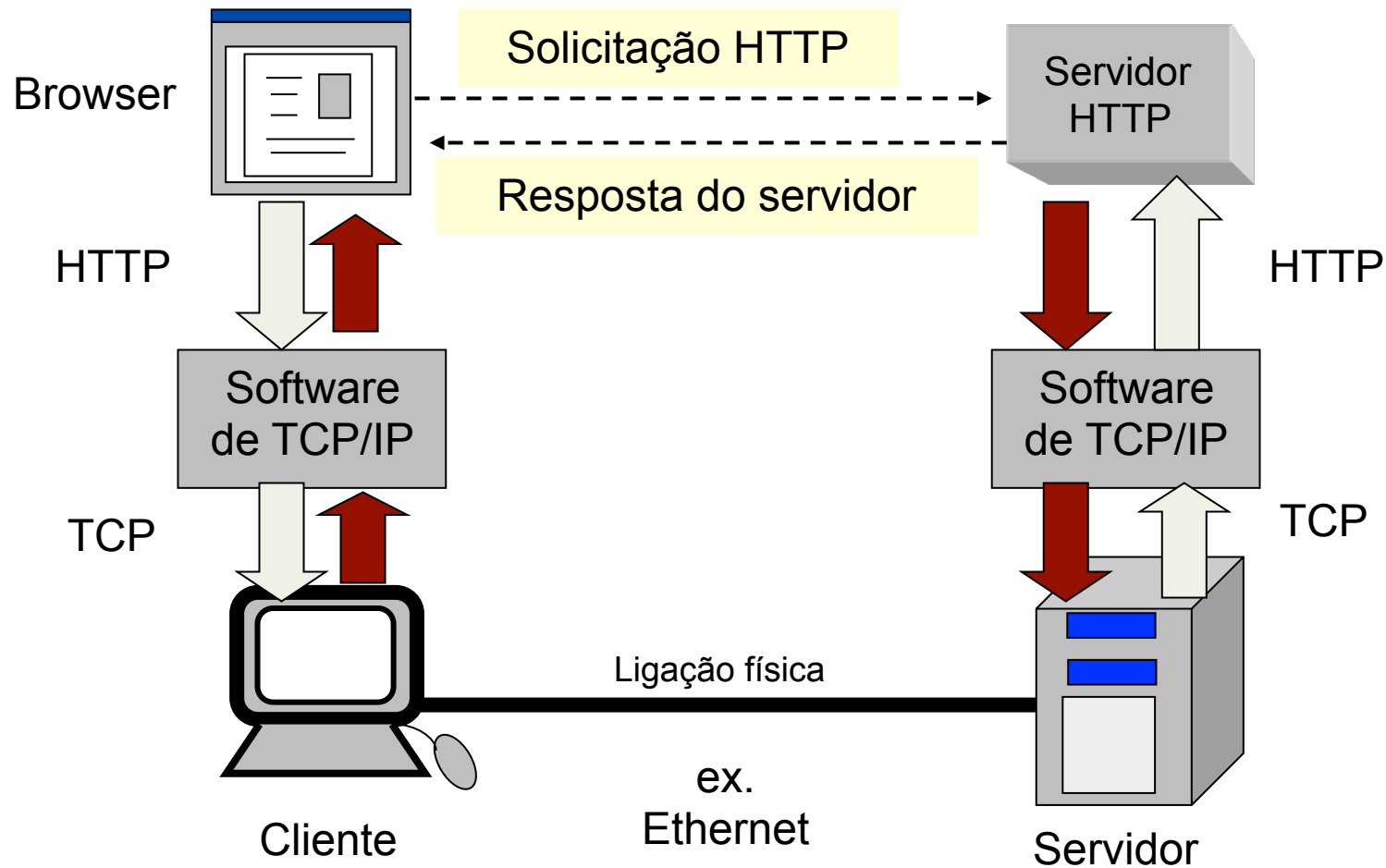
W3C – WWW Consortium



Consórcio World Wide Web (W3C) é uma comunidade internacional que desenvolve padrões com o objetivo de garantir o crescimento da Web

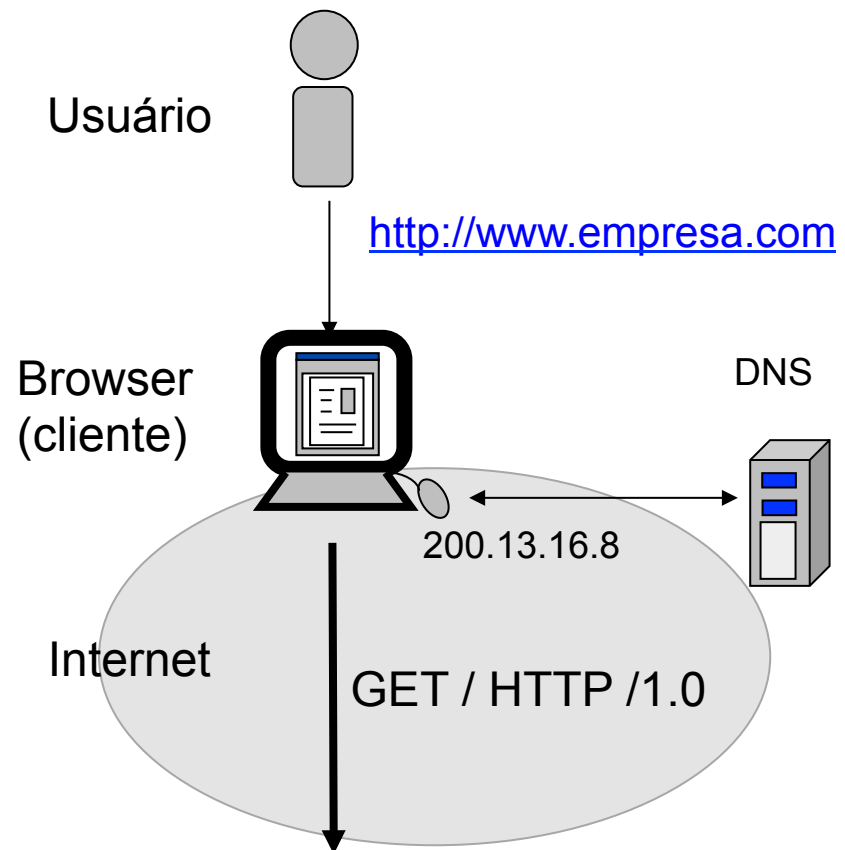
Missão do W3C: conduzir a Web ao seu potencial máximo.

Transação cliente-servidor na Web



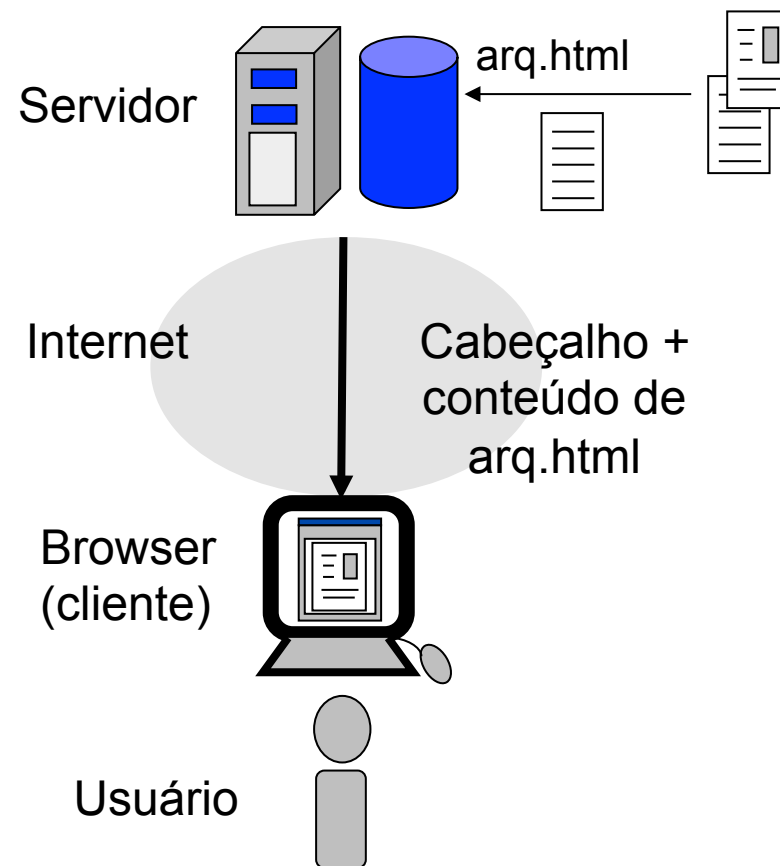
Exemplo de uma sessão Web –1/2

- 1 Usuário solicita
<http://www.empresa.com/arq.html>
- 2 DNS é consultado e fornece o endereço IP
1 200.13.16.8
- 3 O browser faz a conexão e envia a solicitação em HTTP
 - GET /[arq.html](#) HTTP / 1.0
 - ... (seguem outras informações)



Exemplo de uma sessão Web –2/2

- 5 Servidor recebe solicitação e procura pelo recurso (arq.html)
- 6 Servidor:
HTTP/1.1 200 OK
Date: Thu, 23 Oct 1997
21:45:56 GMT
... (após o cabeçalho segue o conteúdo de arq.html)
- 7 Browser apresenta o resultado na tela



HTML – HyperText Markup Language

<html>

<head>

<title> Meu documento </title>

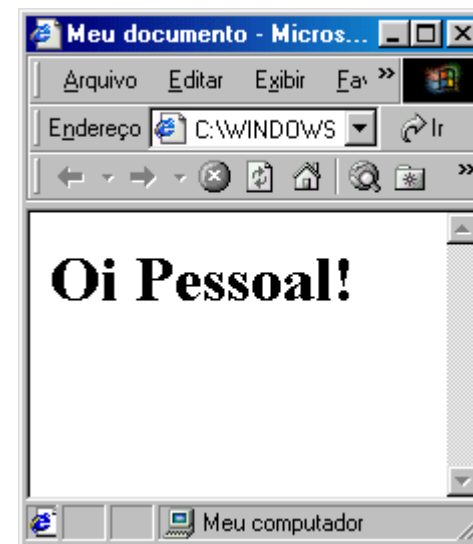
</head>

<body>

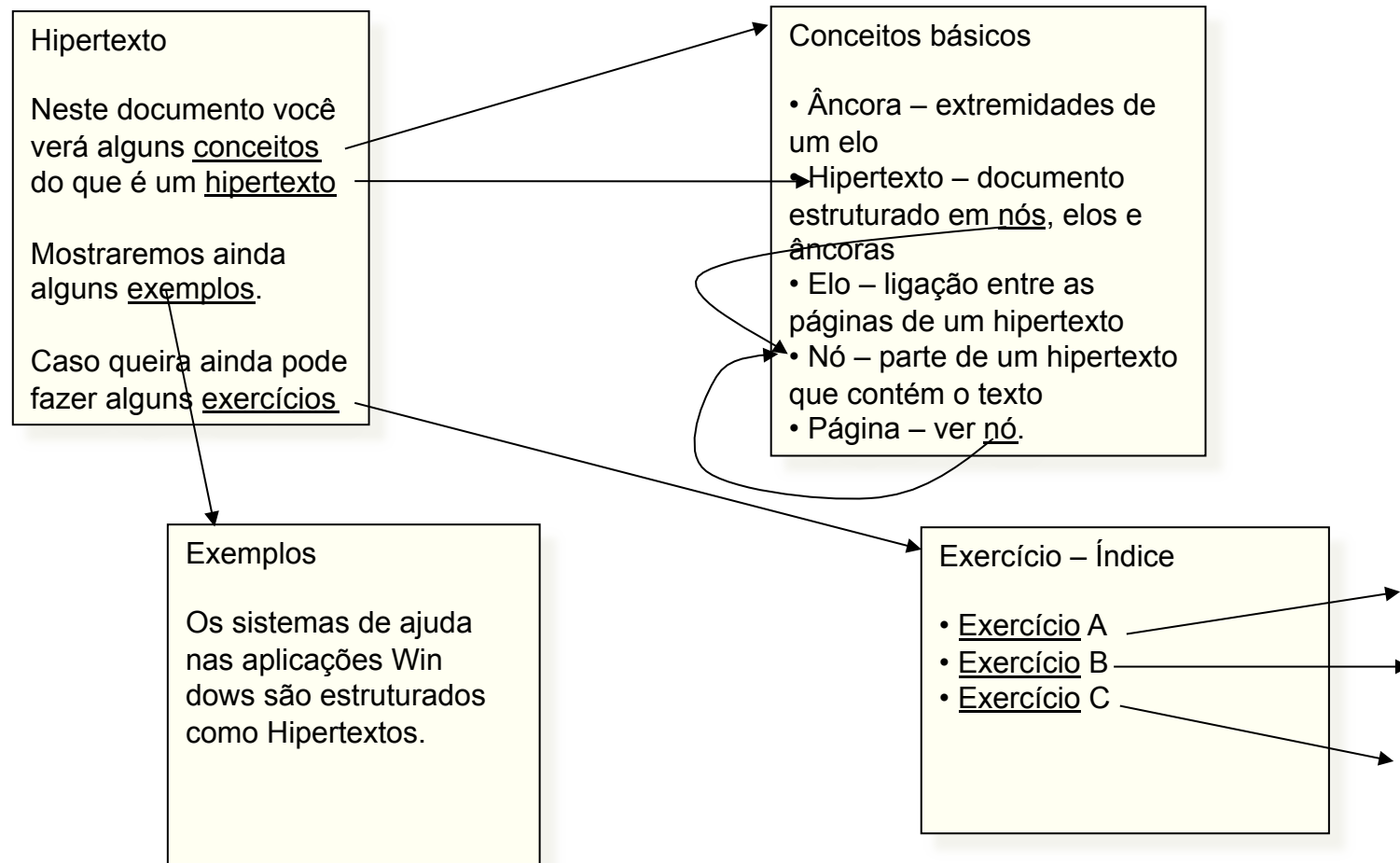
Oi Pessoal!

</body>

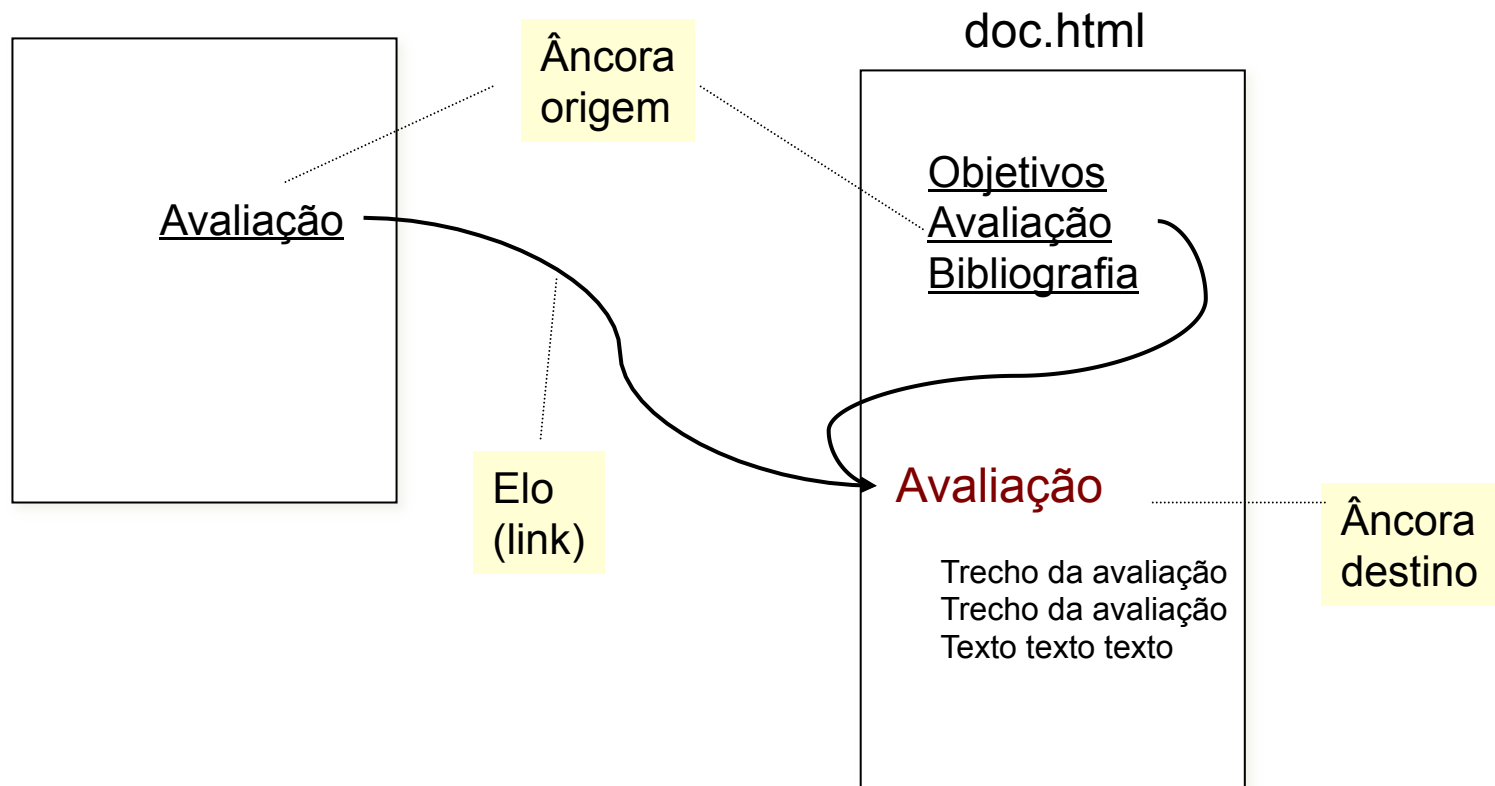
</html>



Hipertexto



Hipertexto - conceitos



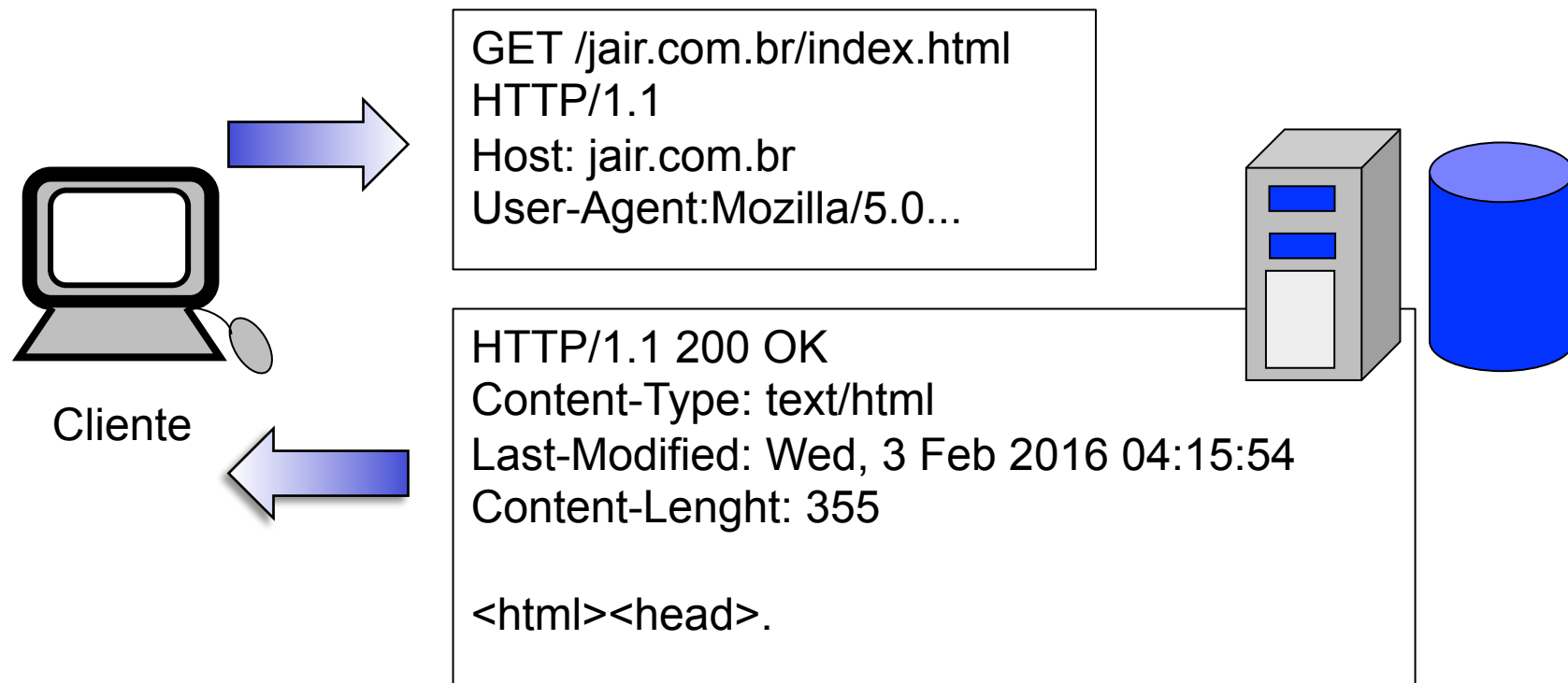
URI – Uniform Resource Identifier

Protocol		Port	
http:	/www.empresa.com:	8080/	path/subdir/file.ext
	Domain name		Diretórios e detalhes do recurso

- **Protocolos:** HTTP, FTP, Gopher, WAIS, mailto, news, file
- **Domain Name:** Nome fantasia de um servidor de recursos na internet, deve ser traduzido num endereço IP
- **Porta:** número da porta do servidor a ser utilizada (opcional)
- **Diretórios:** utiliza-se estilo Unix (/) qualquer que seja o sistema operacional.
- URL – Uniform Resource Locator

HTTP – HyperText Transfer Protocol

- O protocolo HTTP faz a comunicação entre o cliente e o servidor por meio de mensagens requisição-resposta



HTTP – HyperText Transfer Protocol

- O cliente envia uma mensagem de requisição de um recurso
 - A mensagem possui um cabeçalho com o método e informações adicionais
 - O corpo da mensagem é opcional
- O servidor envia uma mensagem de resposta ao cliente com a solicitação.
 - O cabeçalho envia um código resposta (200 OK ou outro) e informações adicionais
 - O corpo da mensagem é o recurso que foi requisitado pelo cliente, se houver

HTTP - métodos

- GET
- POST
- PUT
- DELETE
- HEAD
- TRACE
- OPTIONS
- CONNECT

HTTP – get e post

- As requisições do cliente podem ser através de dois métodos HTTP
- O método GET
 - Os dados são enviados juntos com o endereço
- O método POST
 - Os dados são enviados ao final da mensagem completa do protocolo

Método GET

GET /mydir/program1 HTTP/1.1

Accept : text/plain

Accept : application/x-html

Accept :

.

.

.

User-Agent: NSCA Mosaic for X Window System/2.4
(uma linha contendo apenas CRLF)

Método GET com dados enviados

GET /mydir/program1?primnome=Jair&sobrenome=Leite...

Accept : text/plain

Accept : apliation/x-html

Accept :

.
.
.

User-Agent: NSCA Mosaic for X Window System/2.4

(uma linha contendo apenas CRLF)



Query String

Método POST com dados

POST /mydir/program1 HTTP/1.0

Accept : text/plain

Accept : application/x-html

Accept :

.

.

.

User-Agent: NSCA Mosaic for X Window System/2.4

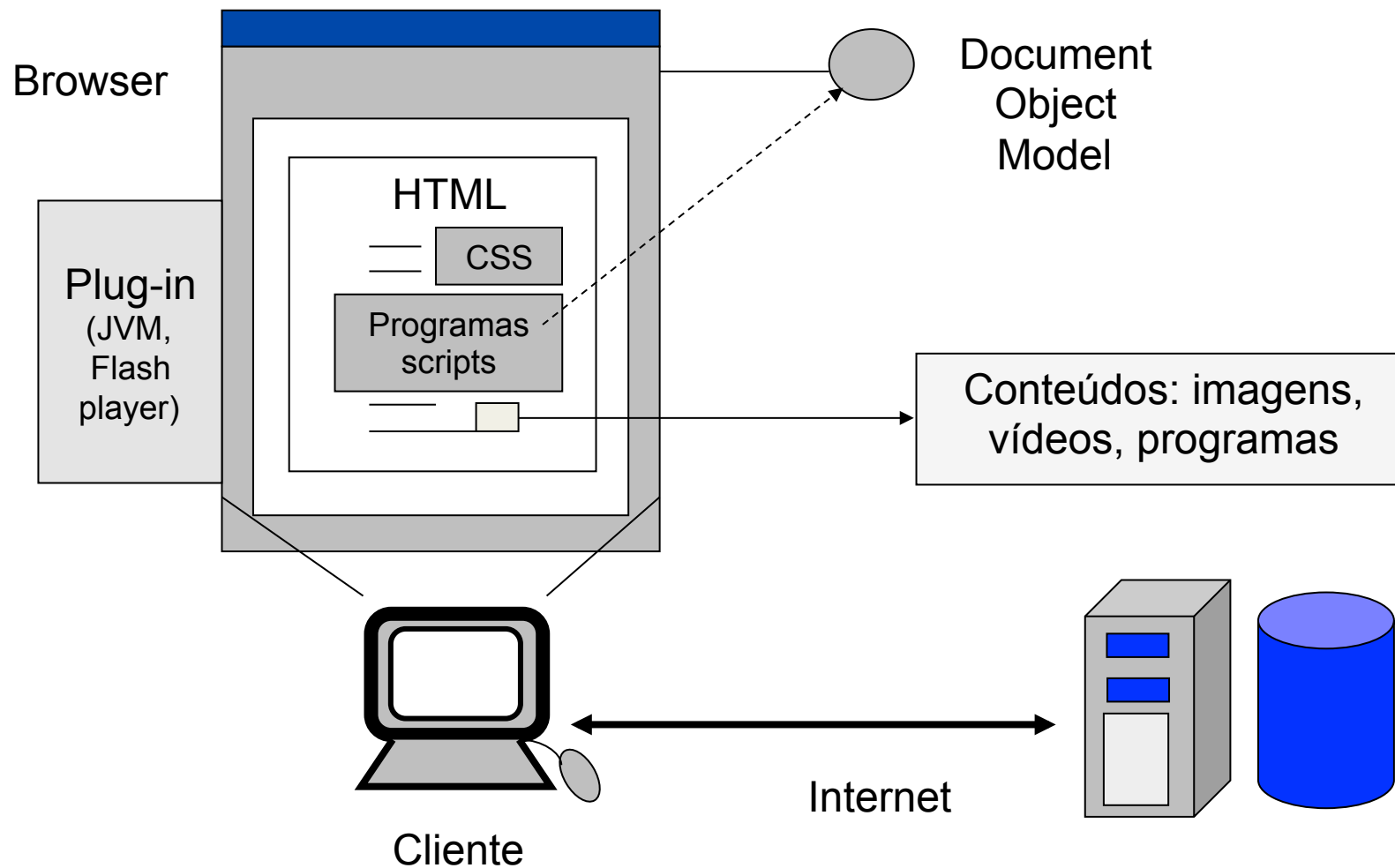
Content-type application/x-www-form-urlencoded

Content-length: 58

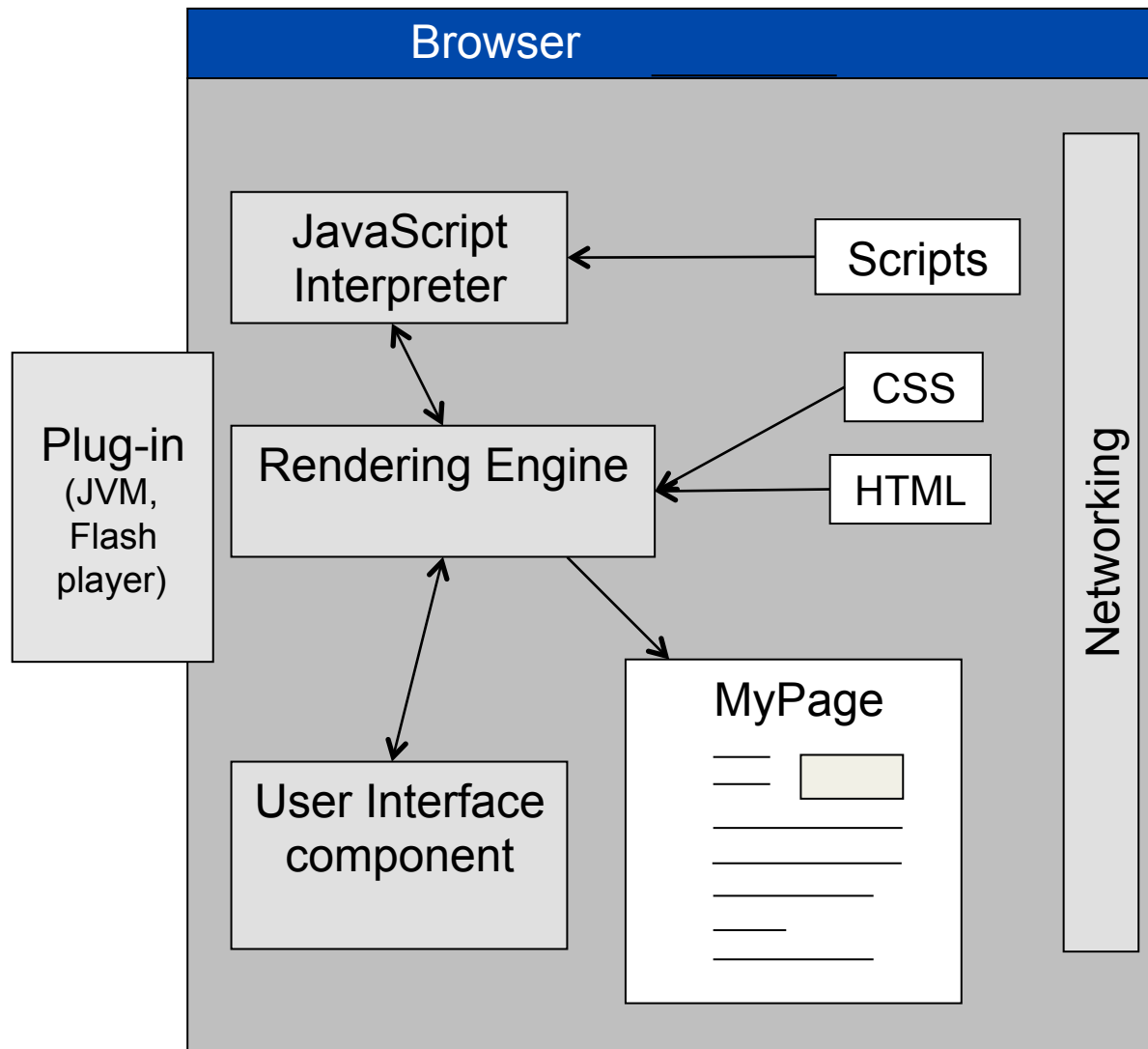
nome=Jair&sobrenome=Leite&sexo=M

TECNOLOGIAS

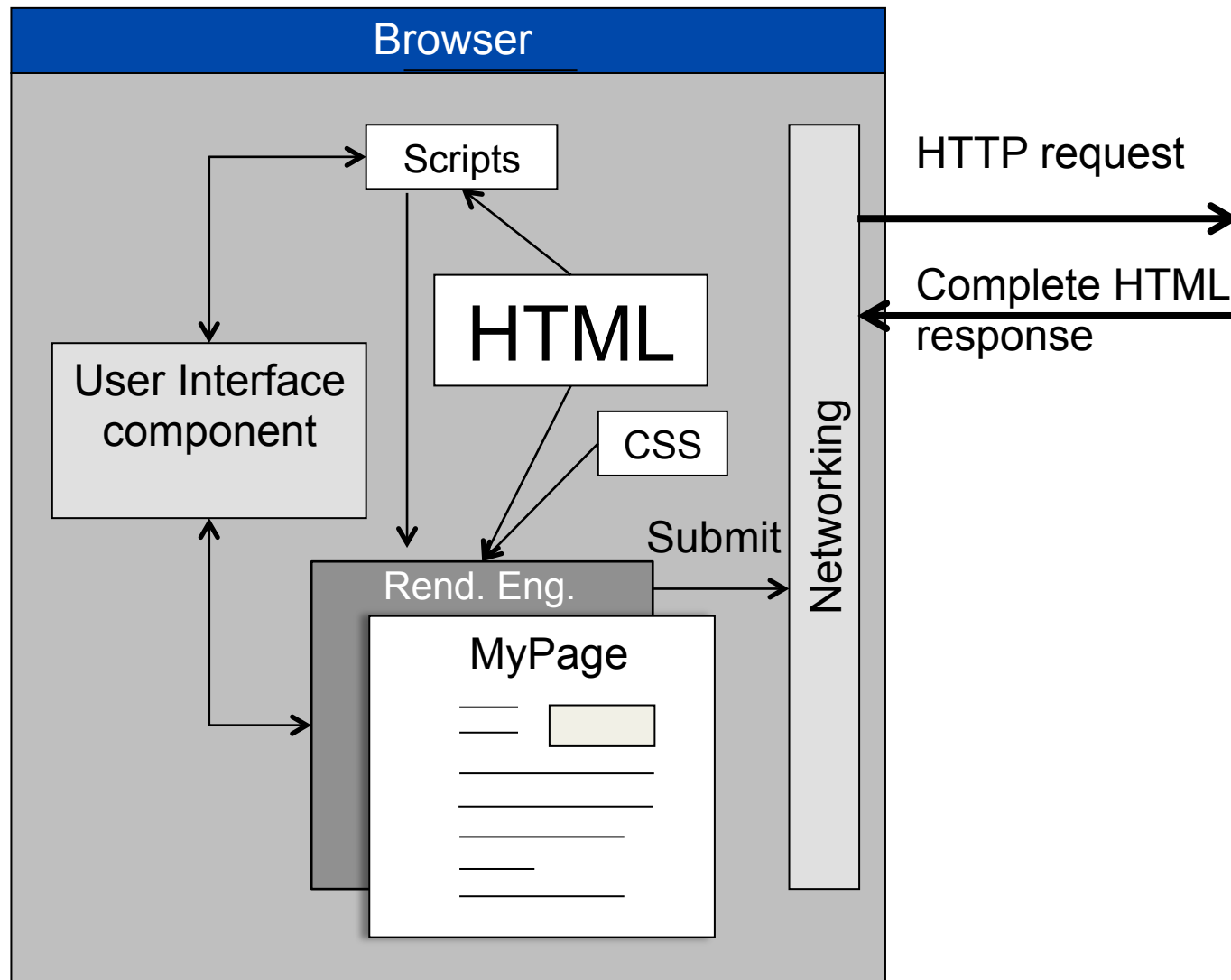
Tecnologias do lado cliente



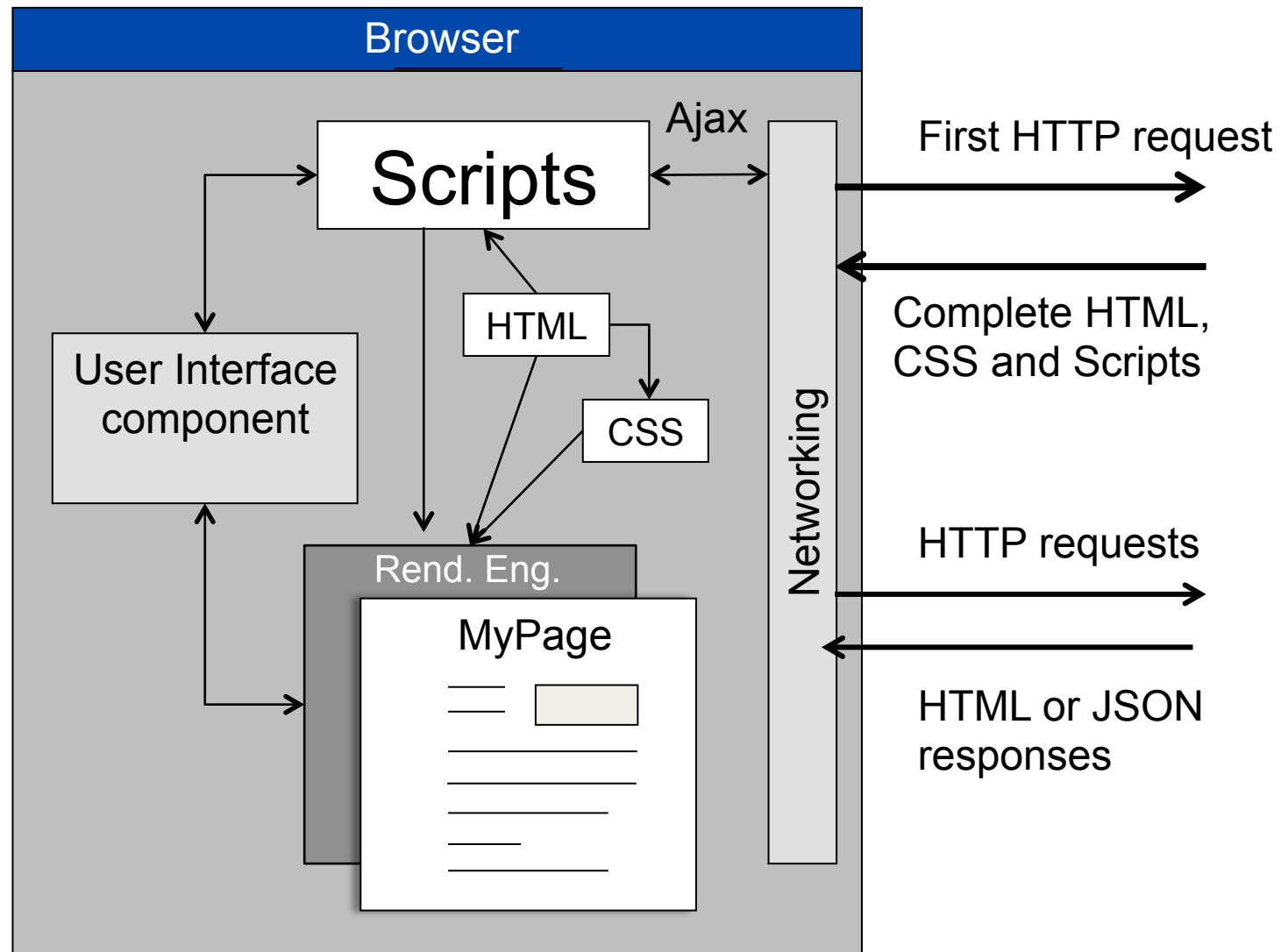
Arquitetura do Browser (visão geral)



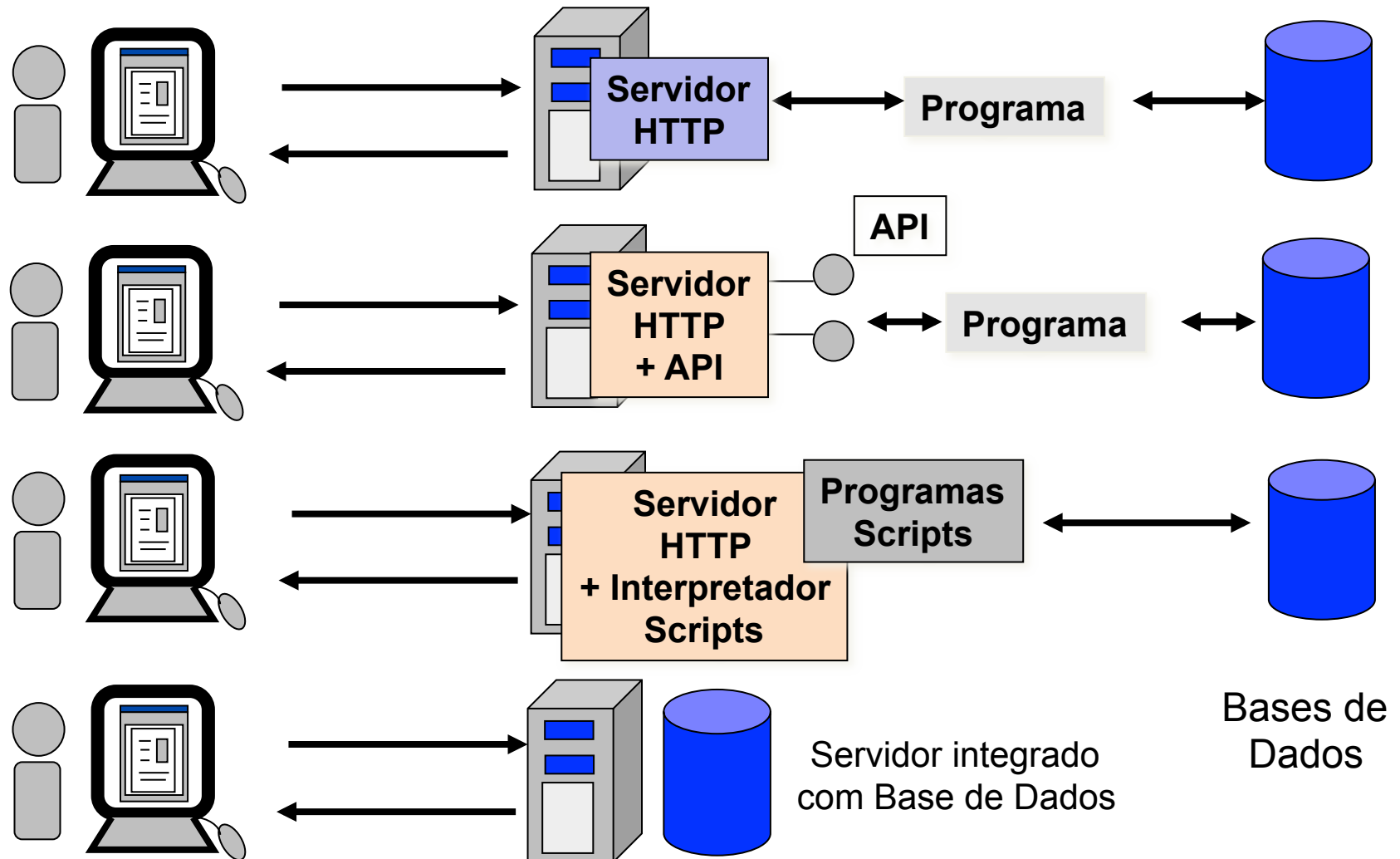
Cliente magro (pobre)



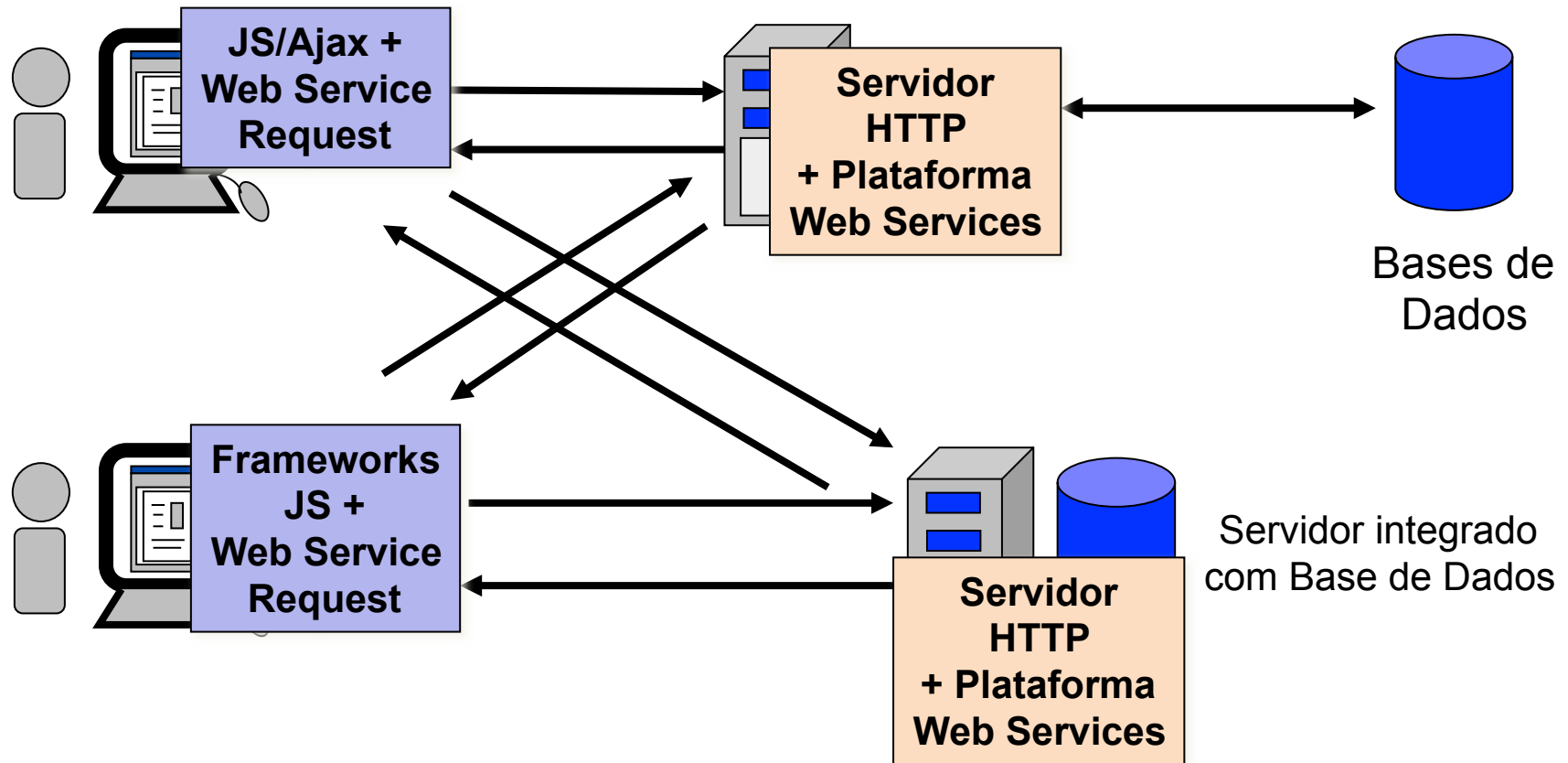
Cliente gordo (rico – RIA, SPA)



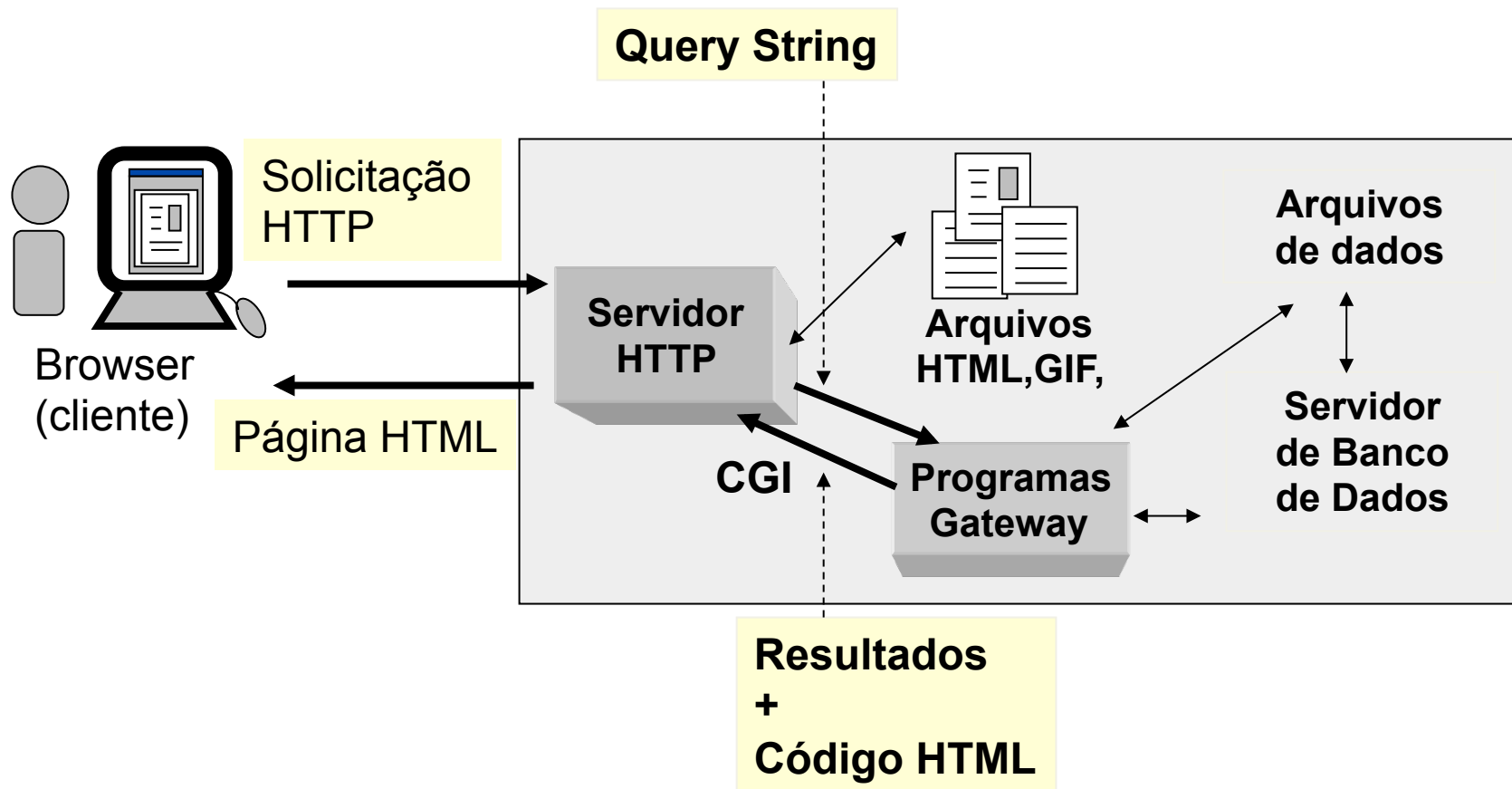
Variações de tecnologias no servidor – 1



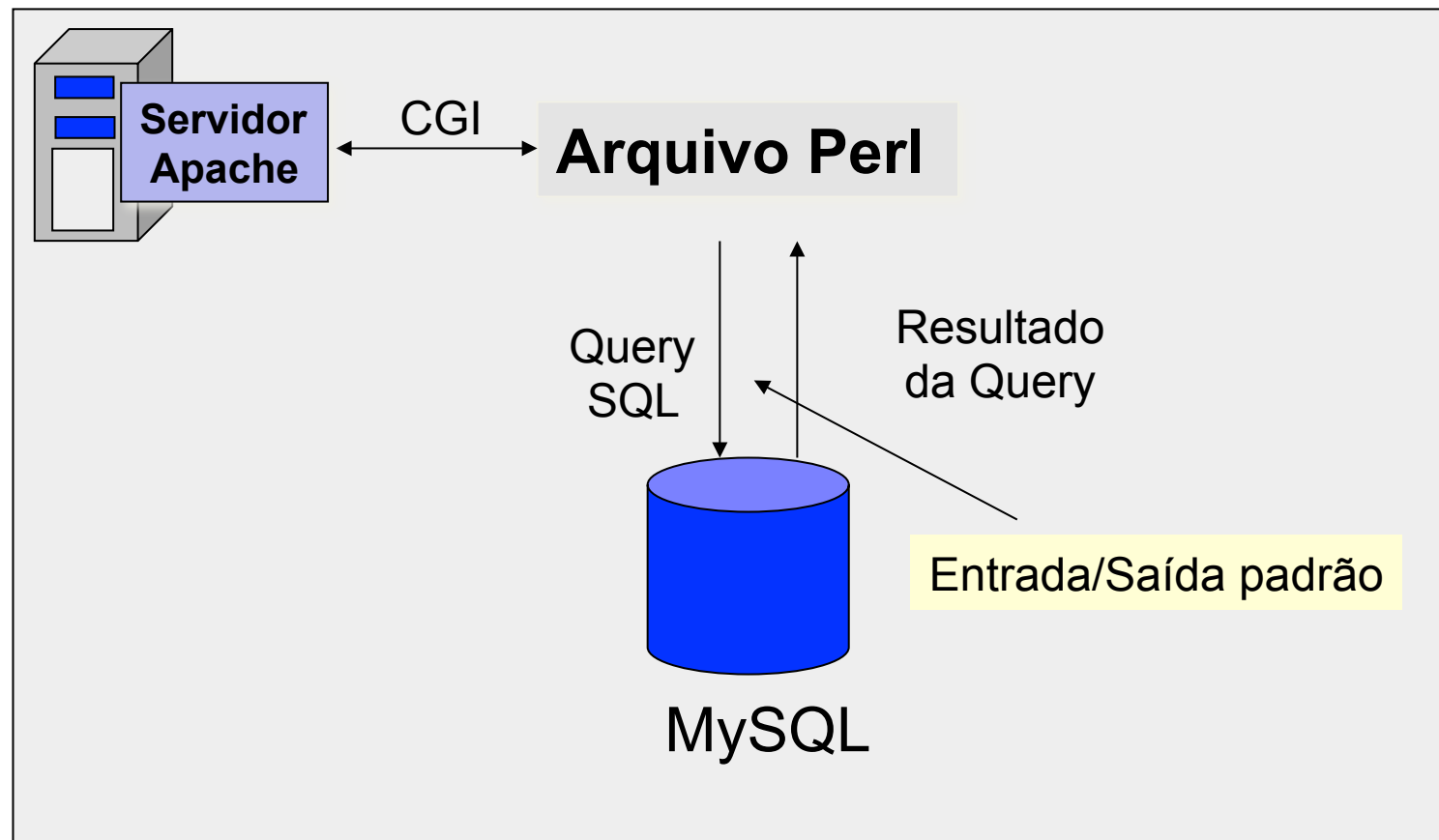
Variações de tecnologias no servidor – 2



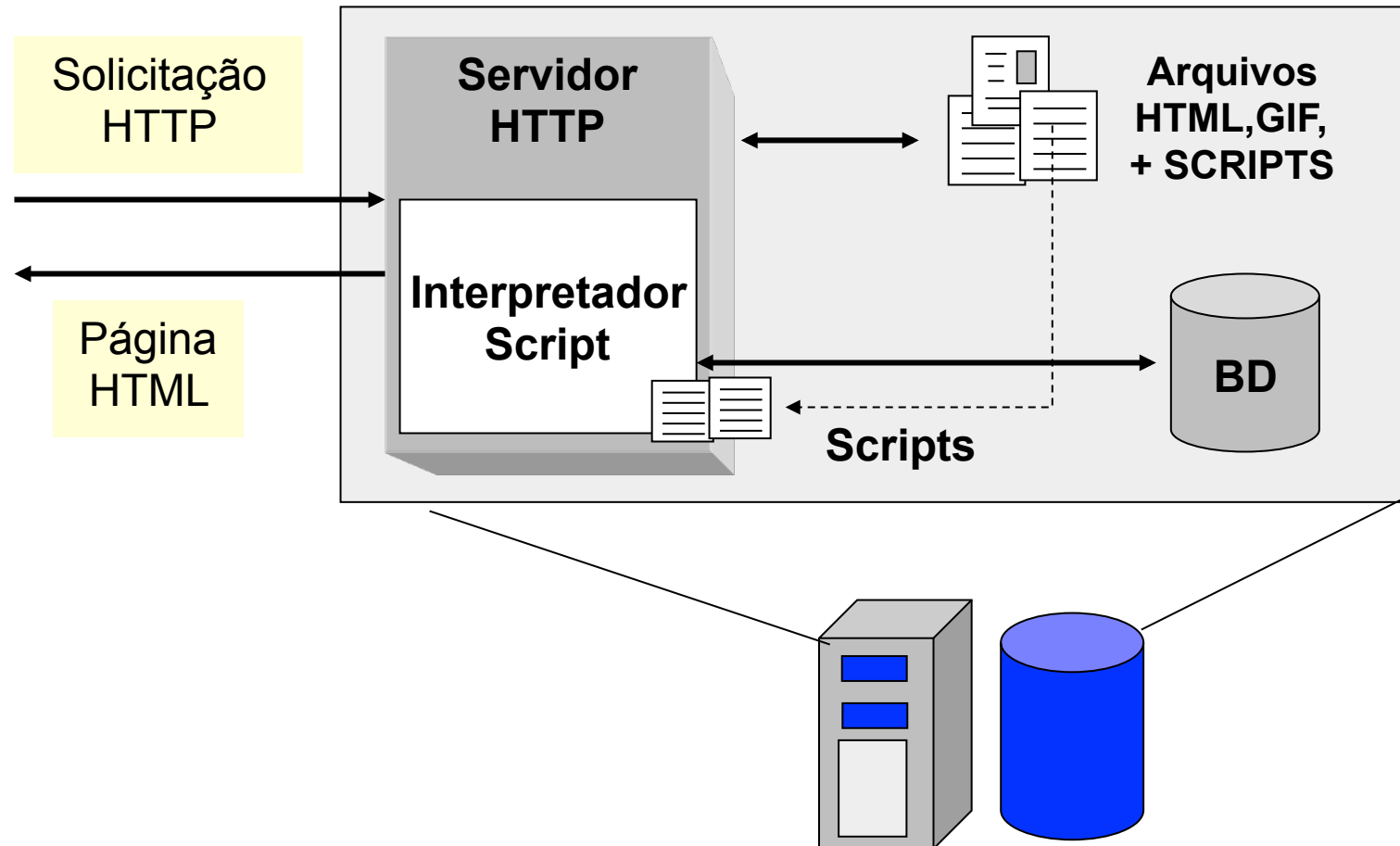
Tecnologias do lado servidor - *original*



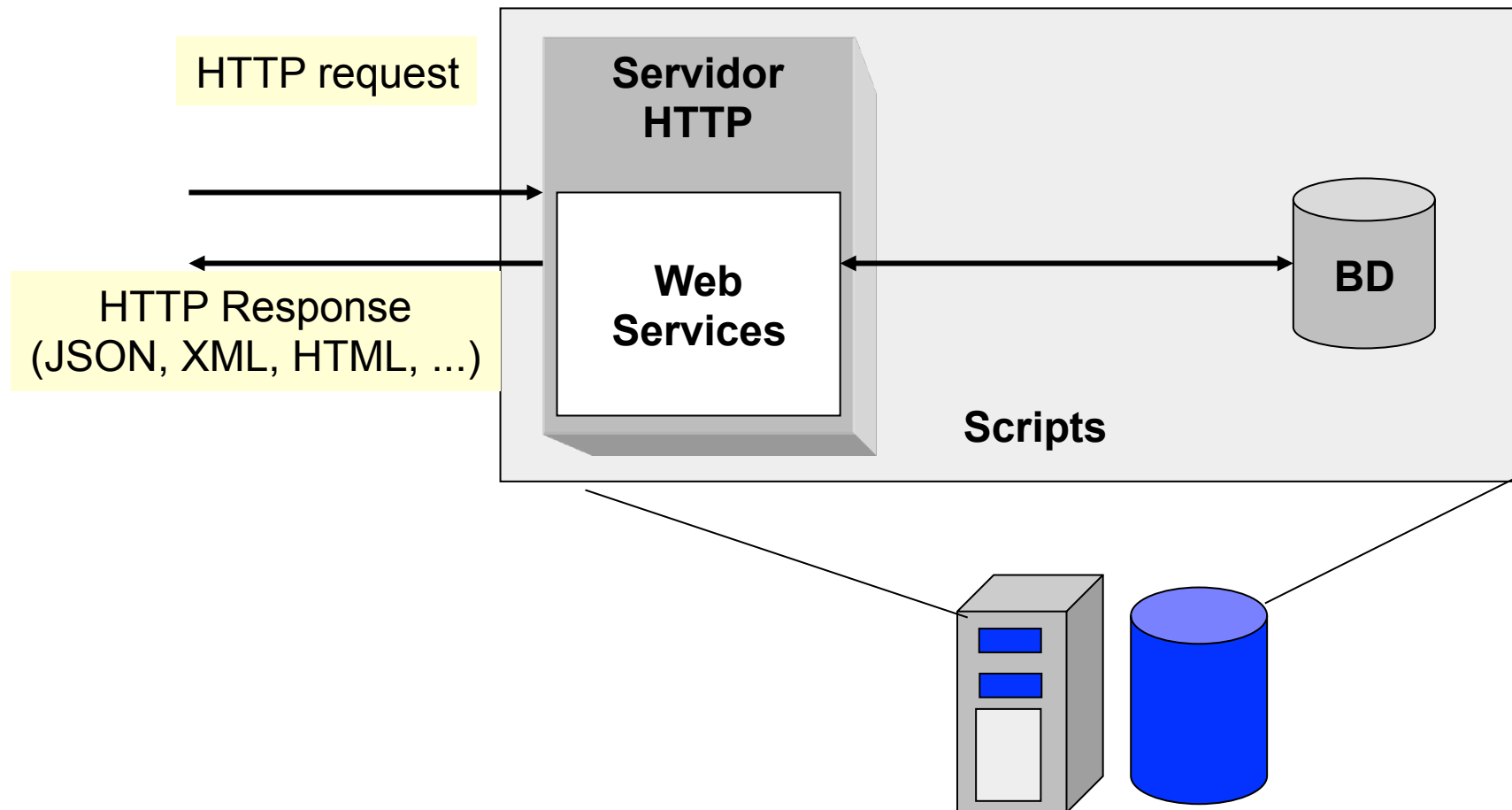
Exemplo: Usando Apache, Perl e MySQL



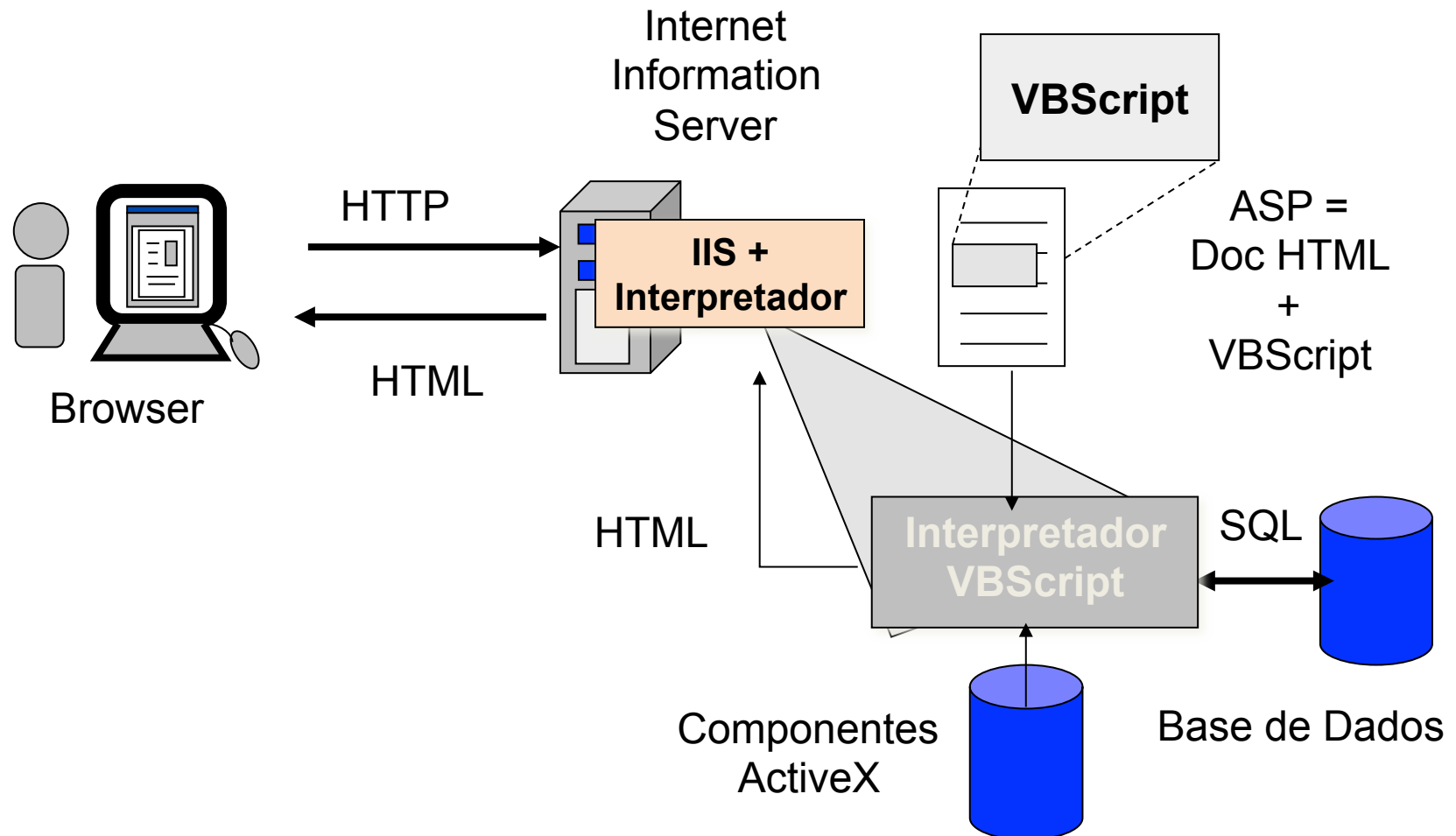
Scripts lado-servidor



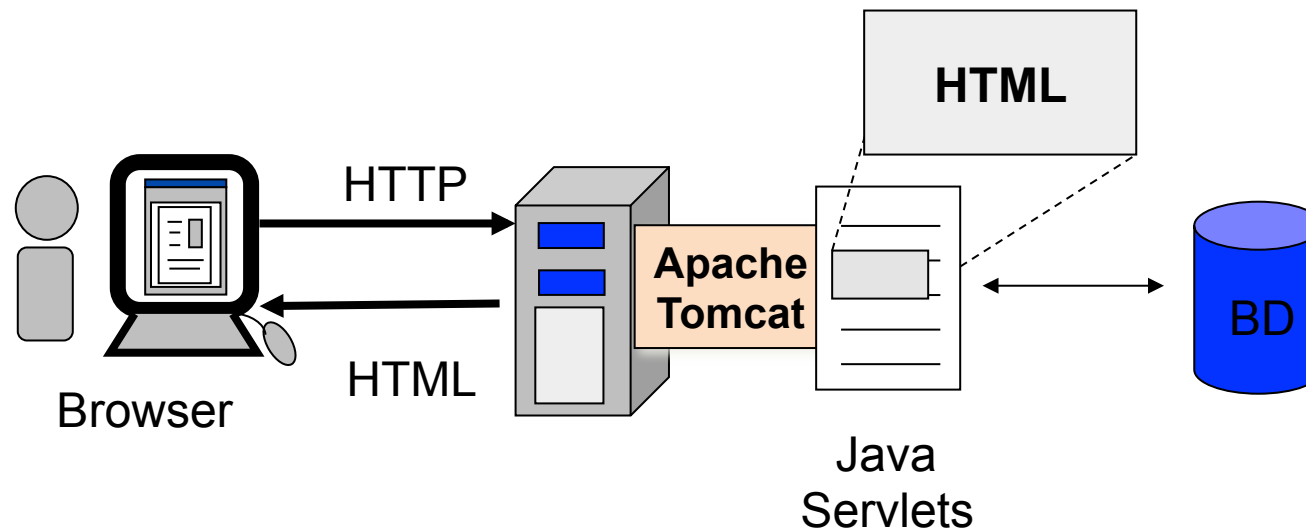
Web Services



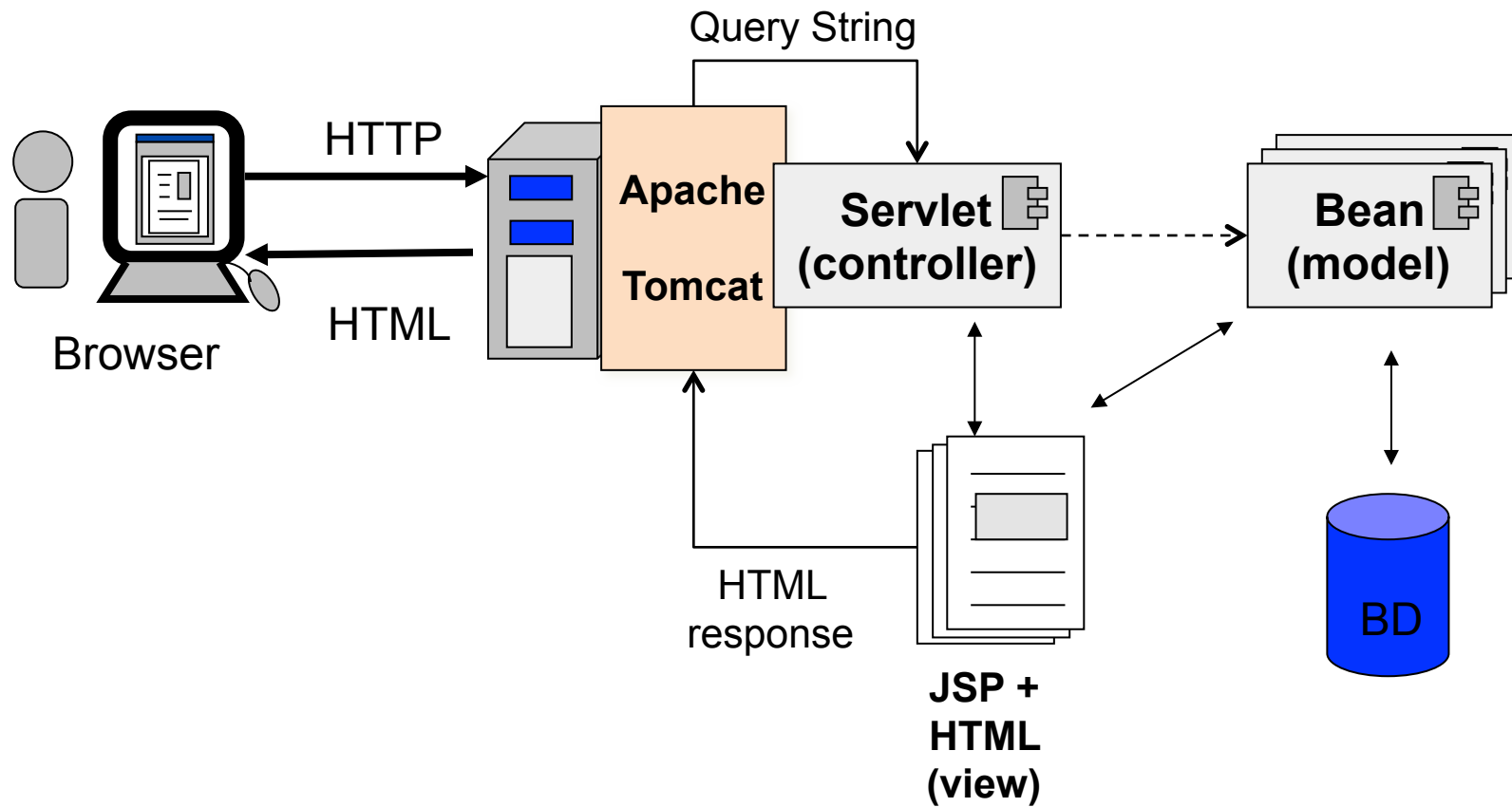
ASP - *Active Server Pages* – Clássico



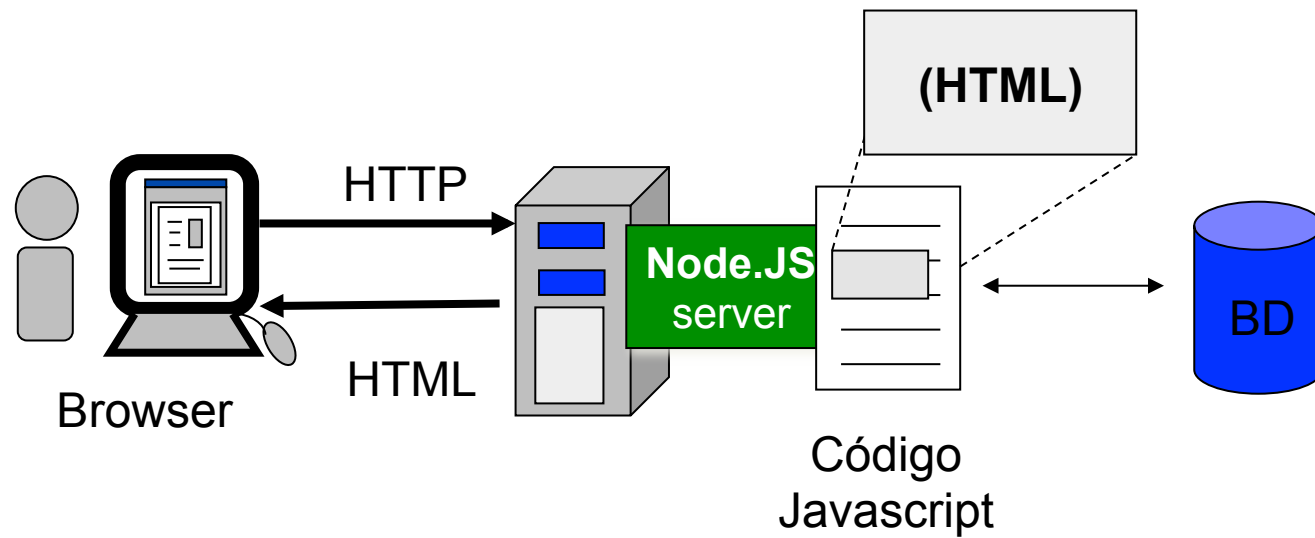
Arquitetura com Java Servlets



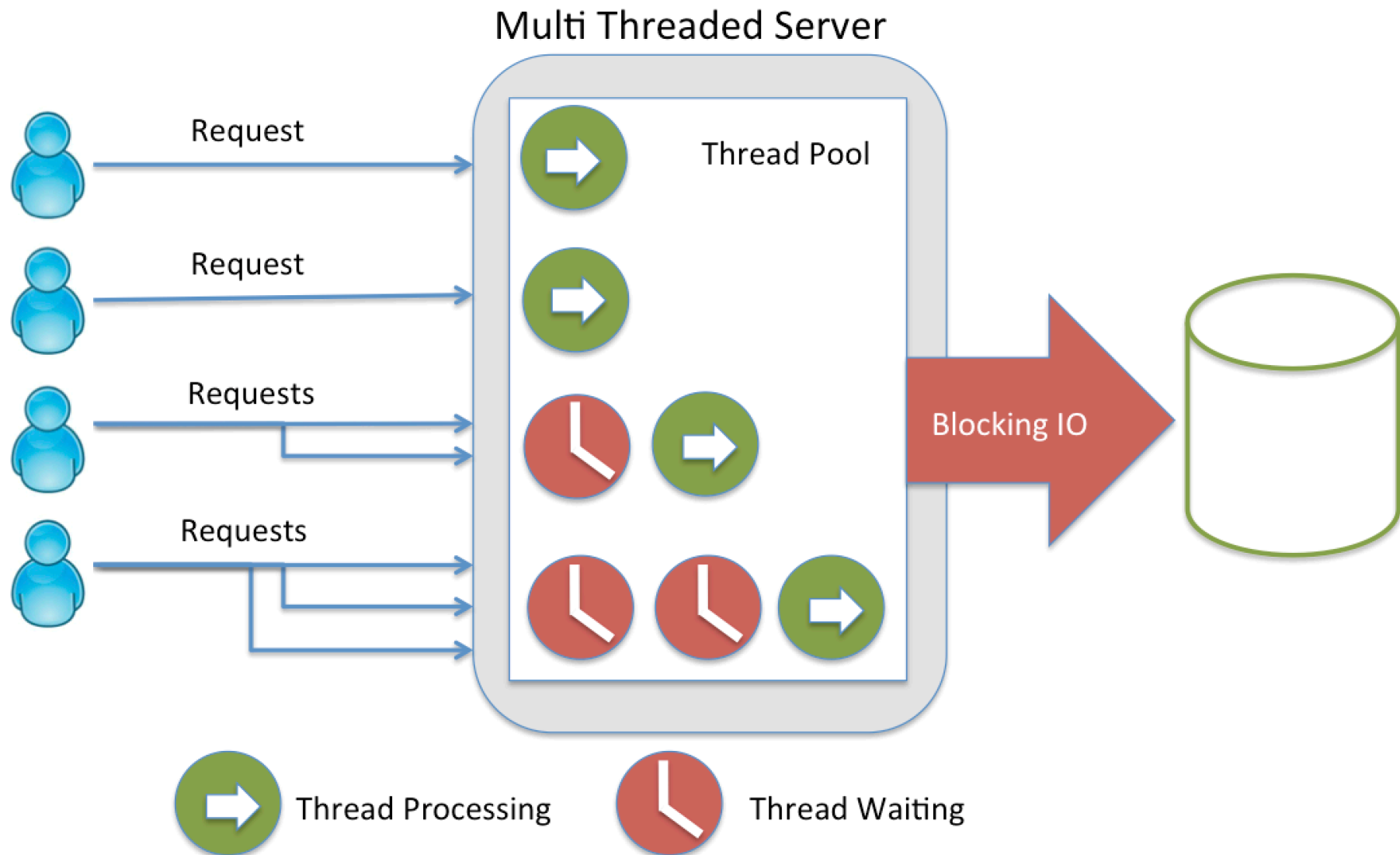
Frameworks Java MVC



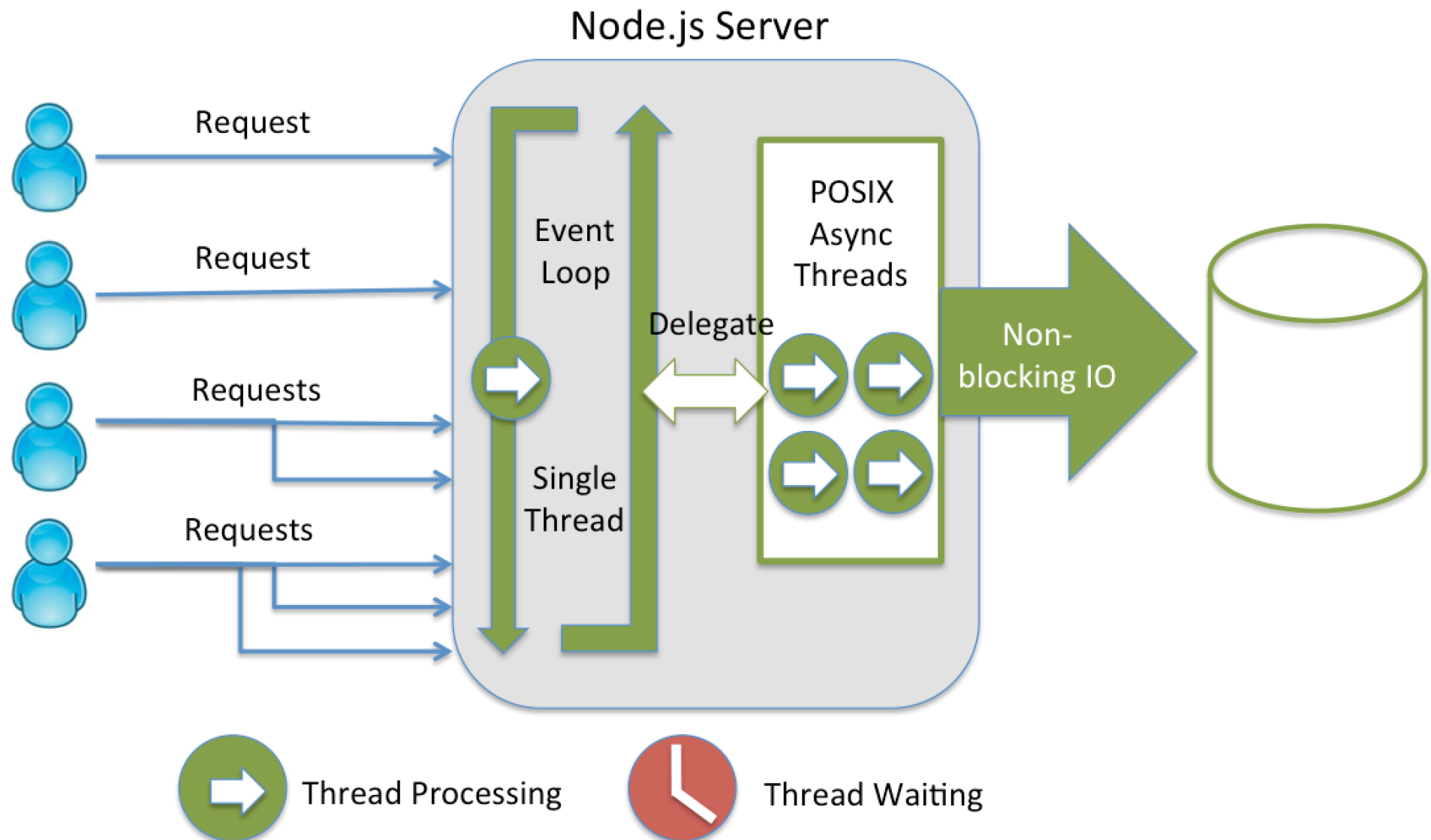
Node.JS



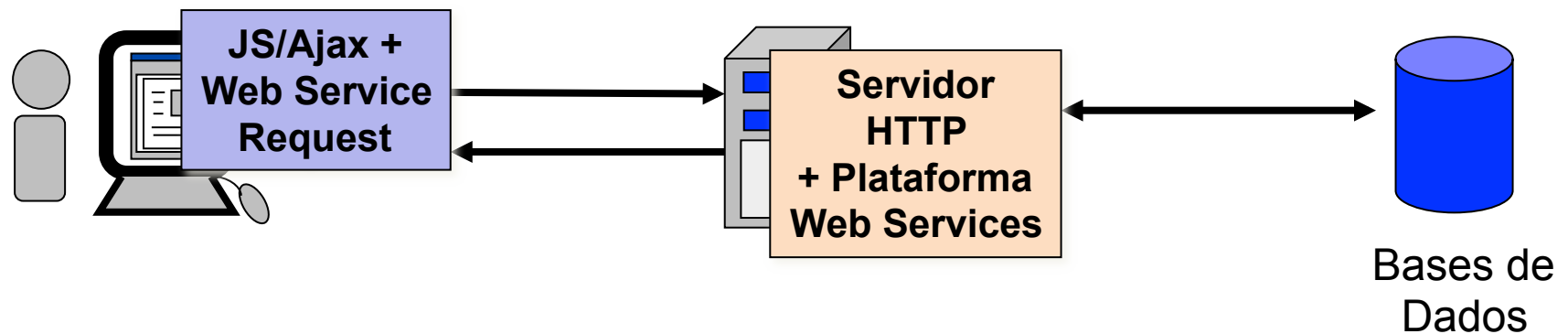
Servidor multi-threads



Node.js: Servidor single-thread



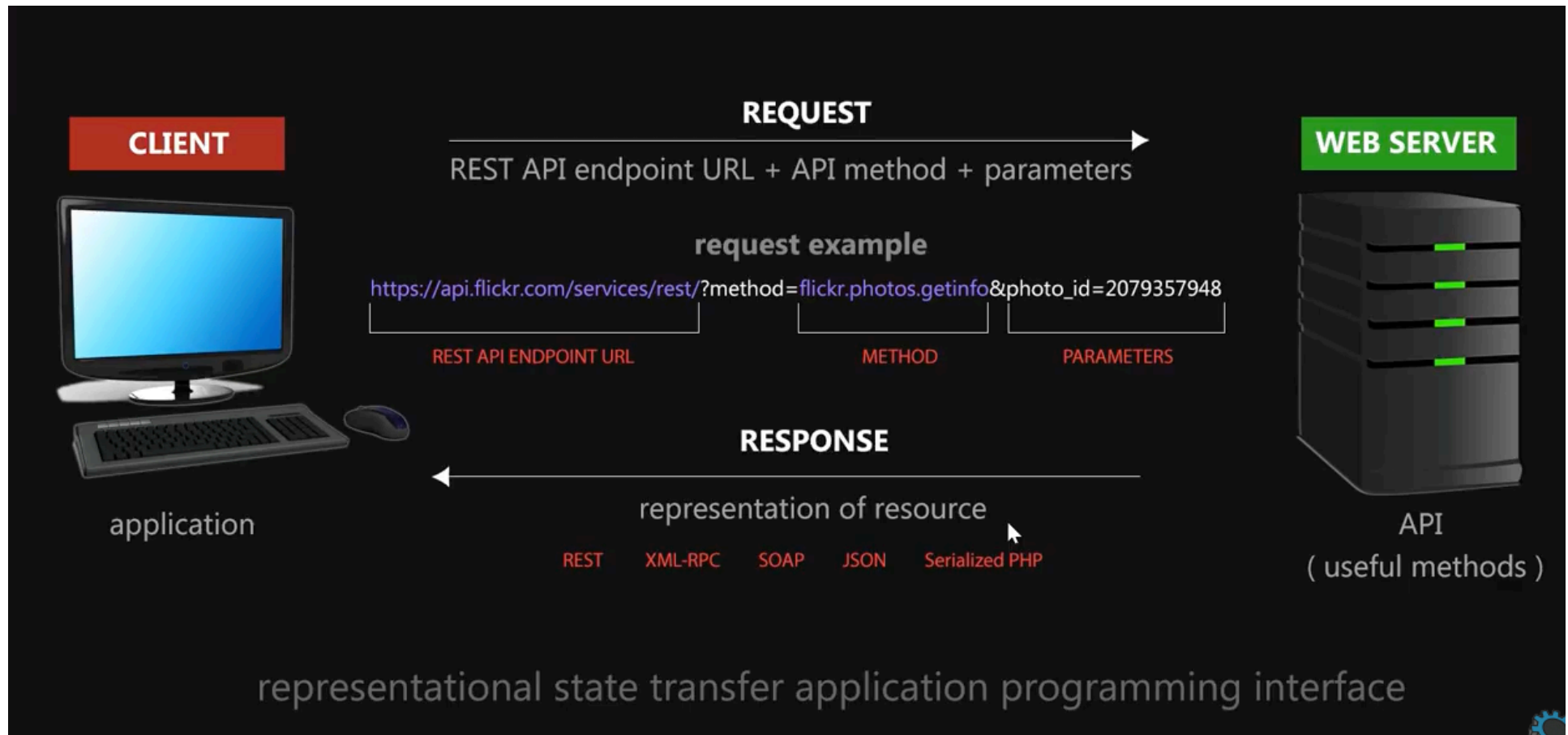
Web Services



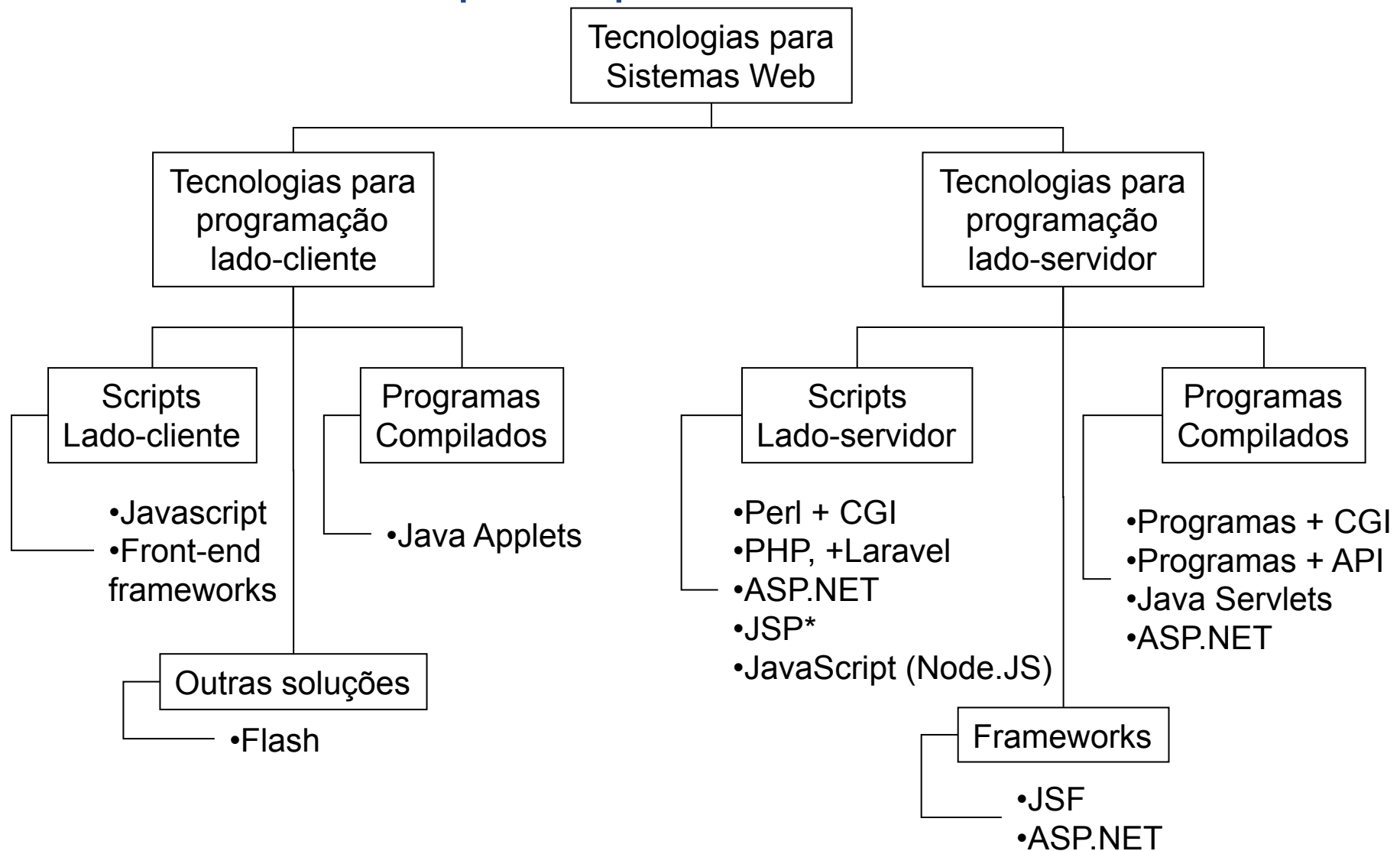
Restful – Request/Response



Restful – Request/Response ++



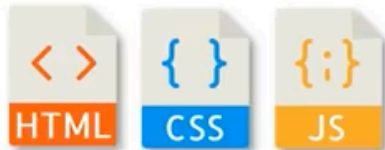
Tecnologias para programação na Web



Tendências 2018

Top Trends in Web Dev 2018

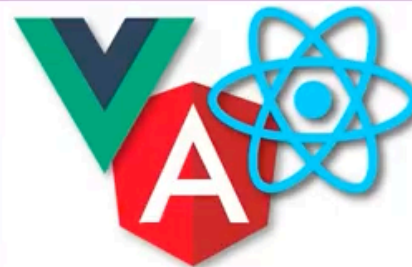
The Basics



Node.js



Frontend Frameworks



Web Components / Compilers



Static Web Pages



Progressive Web Apps



Serverless



AI / Bots

