PUC Minas

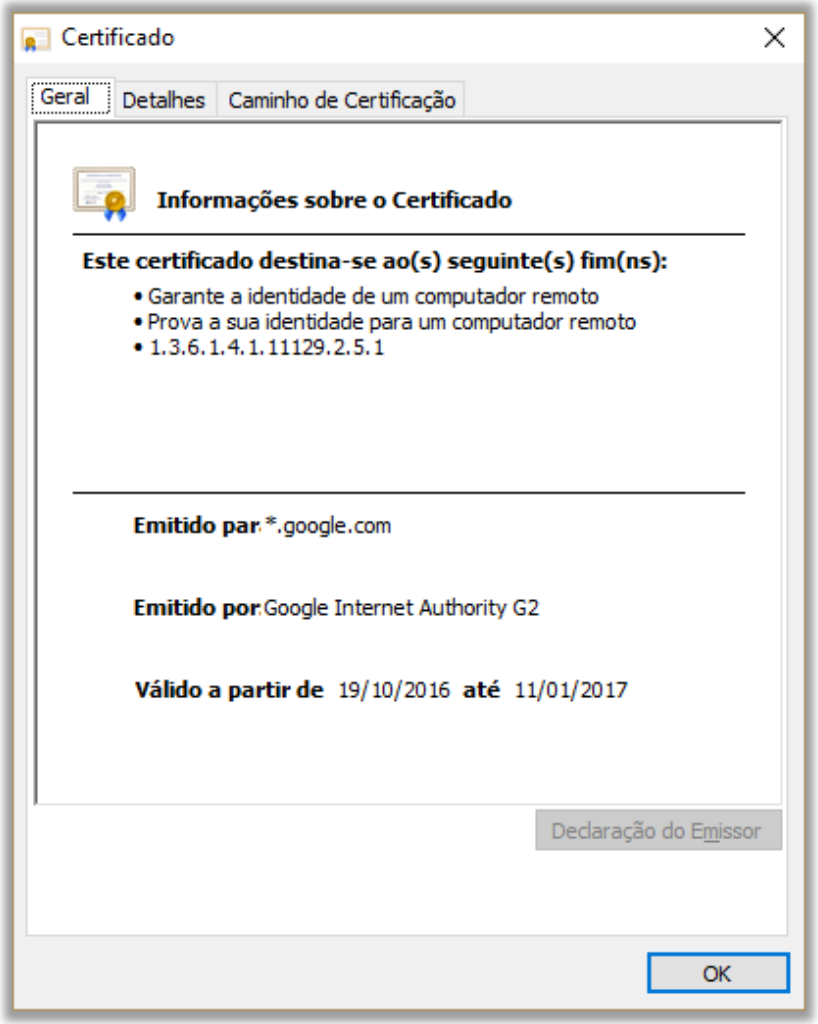
Plataforma Node.js

Segurança na Web

Prof. Rommel Vieira Carneiro

Comunicação HTTPS

HTTPS identifica a comunicação segura por meio do protocolo HTTP, na porta 443 (por padrão), utilizando os protocolos TLS ou SSL.

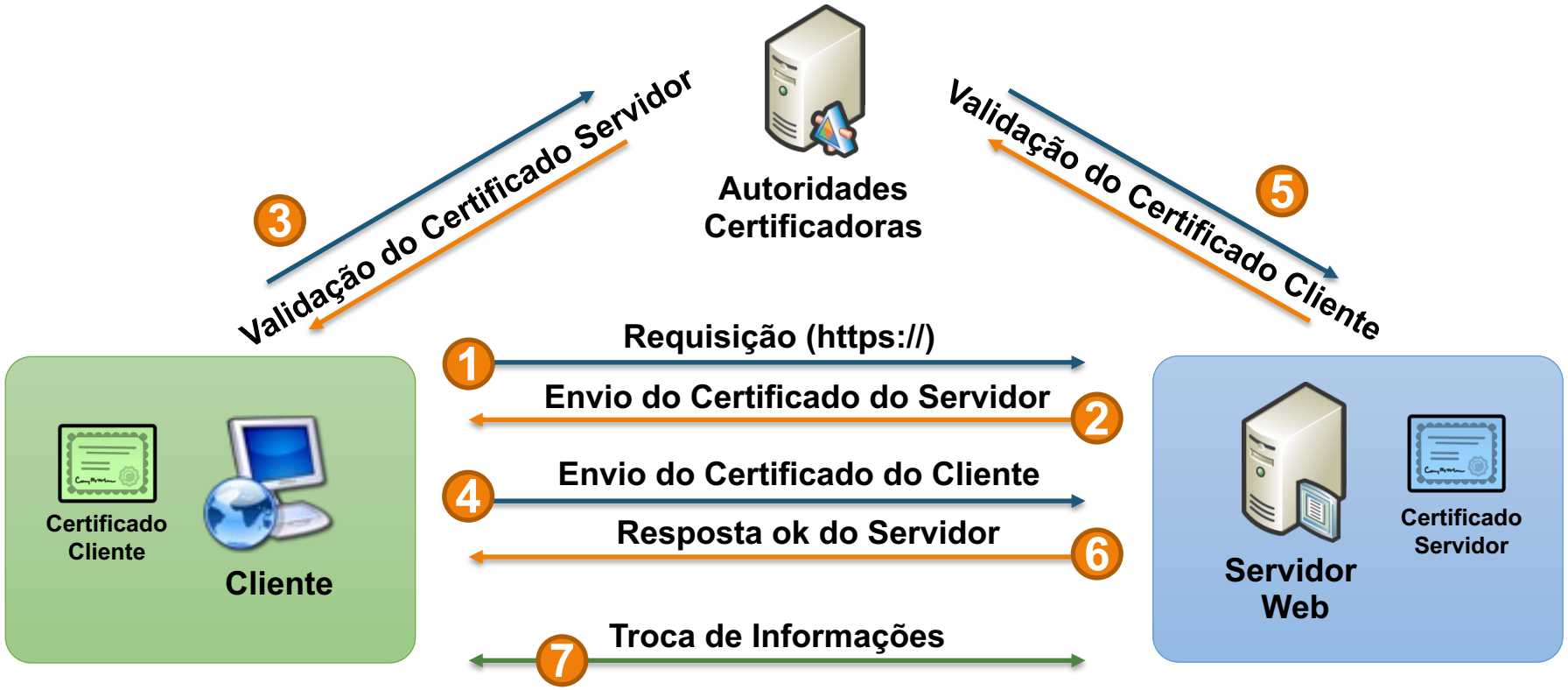


Características

- Fornece uma conexão criptografada com identificação de cliente e servidor
- Baseado em certificados digitais emitidos por autoridades certificadoras
- Requer que servidores Web sejam configurados com certificados digitais
- Requer que os navegadores reconheçam as autoridades certificadores emissoras dos certificados do servidor

Comunicação HTTPS

HTTPS – Fluxo de Comunicação



Comunicação HTTPS

HTTPS – Problemas com Certificados

Um certificado pode apresentar diversos problemas tais como:

- Certificado Expirado
- Certificado de um site diferente do acessado
- Certificado revogado pela autoridade certificadora
- Certificados de autoridades certificadoras desconhecidas
- Certificados assinados pelo próprio site

Testes de Certificados

Utilize o site de testes BADSSL que apresenta um conjunto de sites com certificados com problemas:
https://badssl.com/

Autenticação HTTP

Processo para verificar a identidade do usuário de uma aplicação Web.

Esquemas de Autenticação

- Usuário Anônimo + Forms da aplicação (Login)
- Autenticação Basic
- Autenticação Digest
- Autenticação Bearer (Token Authentication)

Provedores / Integrações

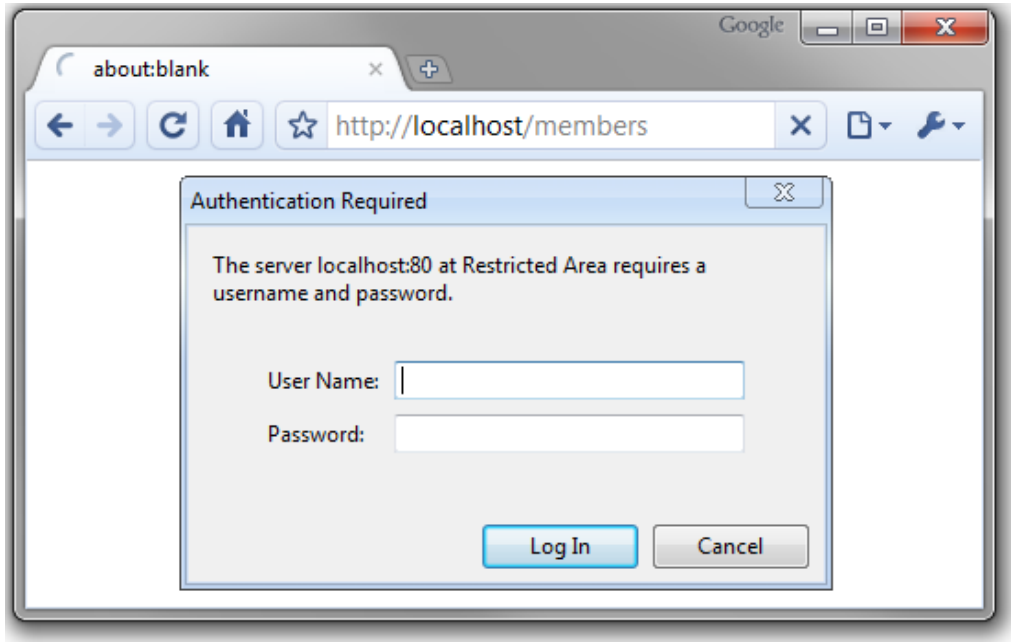
- LDAP (Active Directory, Novell NDS)
- Kerberos
- Formulários de login providos pelas aplicações Web

Fonte: RFC 2617 - HTTP Authentication: Basic and Digest Access Authentication (https://tools.ietf.org/html/rfc2617)

Autenticação HTTP – Basic

Tela de login exibida pelo próprio browser e envio de *string* codificada em Base64 com informação de usuário e senha.

IMPORTANTE: Recomenda-se utilizar apenas com conexões HTTPS.



Fonte: RFC 2617 - HTTP Authentication: Basic and Digest Access Authentication (https://tools.ietf.org/html/rfc2617)

Autenticação HTTP – Basic

Cliente

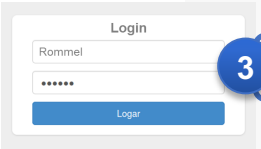
Servidor Web

1 Navegador solicita página: http://servidor/index.html

```
GET /index.html HTTP/1.1
Host: Servidor
```

Servidor exige autenticação 2

```
401 Unauthorized HTTP/1.1
WWW-Authenticate: Basic realm="pucminas.br"
```



3 Usuário informa login e senha

4 Navegador encaminha dados de login e senha (Base64)

```
GET /index.html HTTP/1.1
Authorization: Basic Um9tbWVsOlNlbmhhQDEyMw==
```

Servidor envia página index.html 4

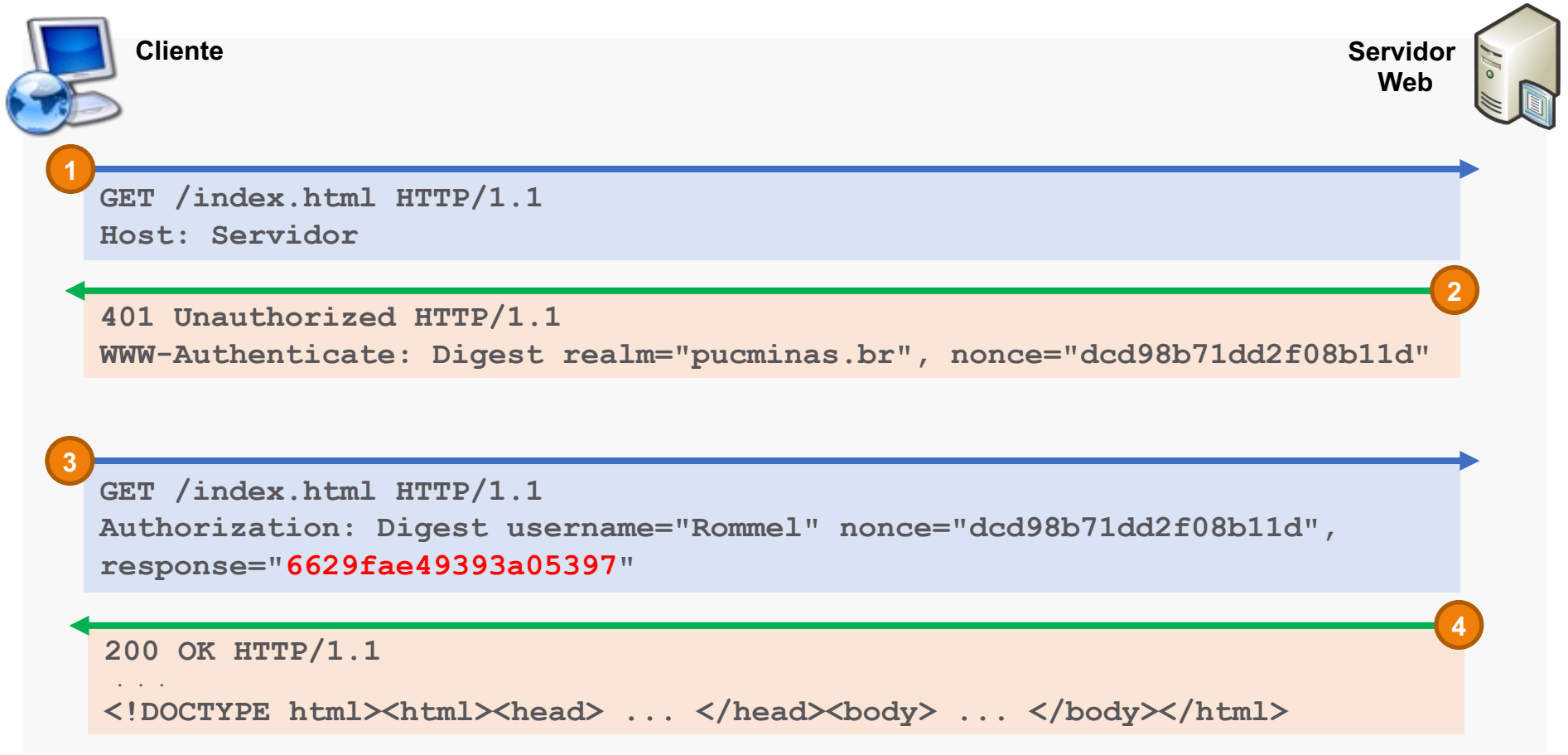
```
200 OK HTTP/1.1
. . .
<!DOCTYPE <html><html><head>...</head><body>...</body></html>
```

Informações

- **WWW-Authenticate**: Cabeçalho da resposta que exige o envio de dados de autenticação
- **Realm**: Definição do espaço protegido pela autenticação
- **Authorization**: cabeçalho da requisição que leva os dados de autenticação do cliente.
- **Base64**: algoritmo de codificação de dados para a Internet.

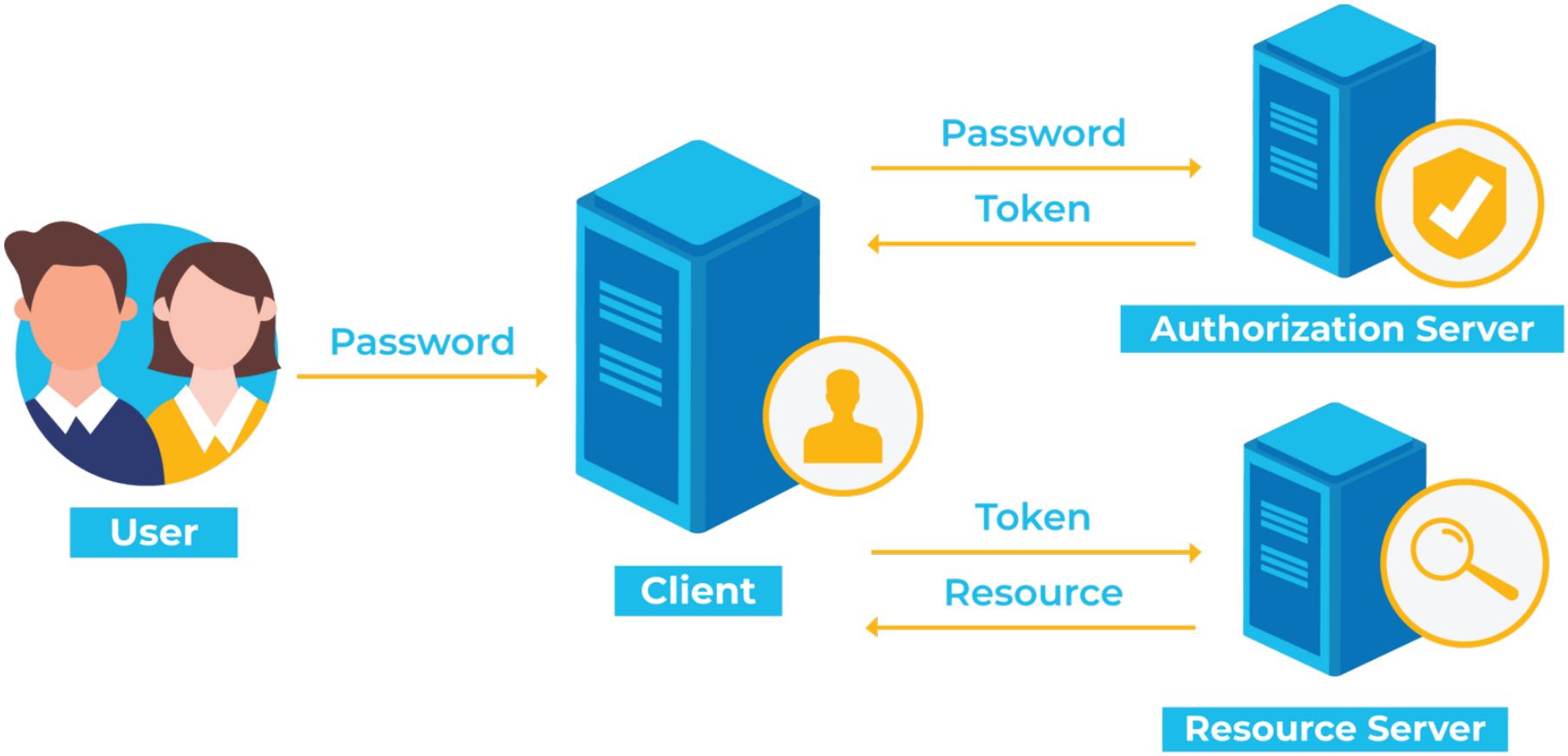
Autenticação HTTP – Digest

Cliente e servidor não trocam informações de senha, apenas o hash.



Autenticação HTTP – Bearer

Token Authentication



Fonte: What Is Token-Based Authentication? (Okta)

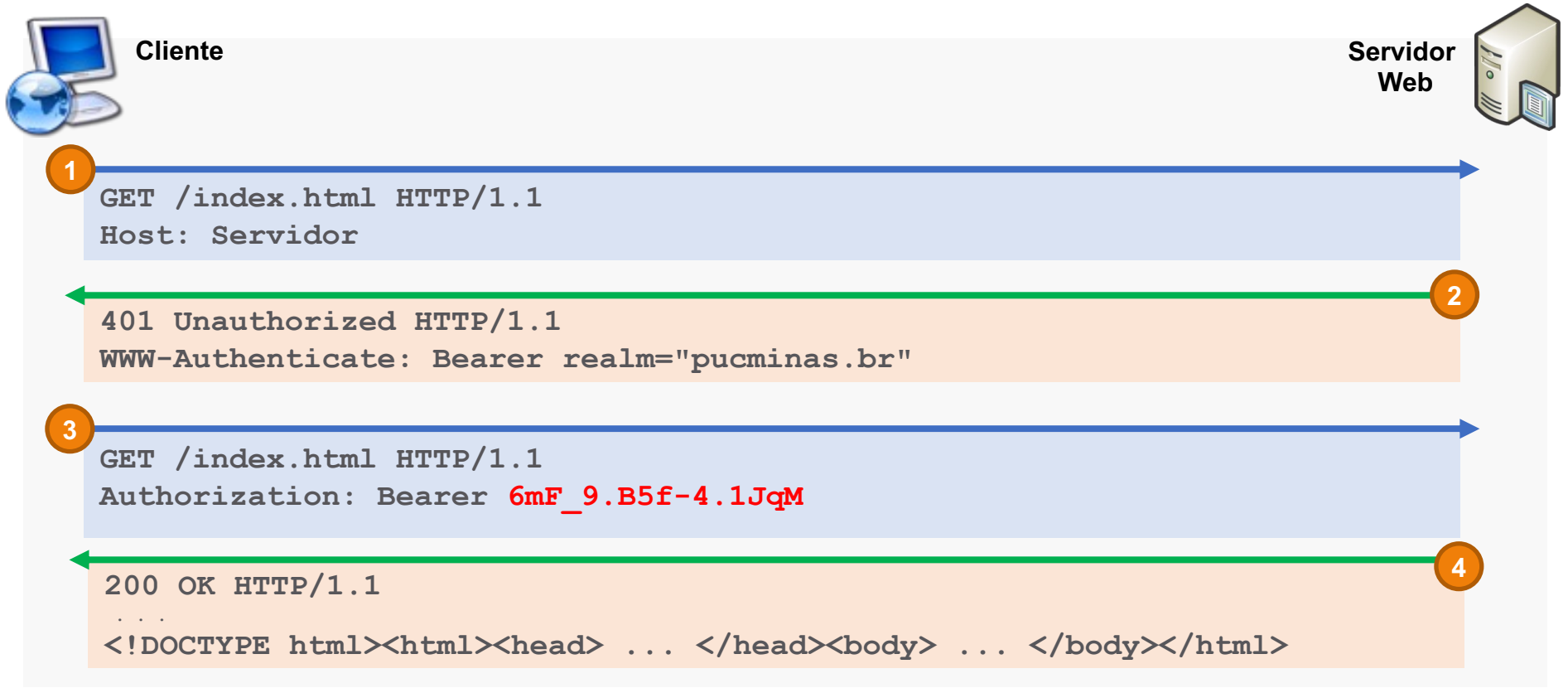
okta

Autenticação HTTP – Bearer

Token Authentication

Cliente e servidor trocam uma token previamente acordada.

IMPORTANTE: Recomenda-se utilizar apenas com conexões HTTPS.



Obrigado!

Prof. Rommel Vieira Carneiro