

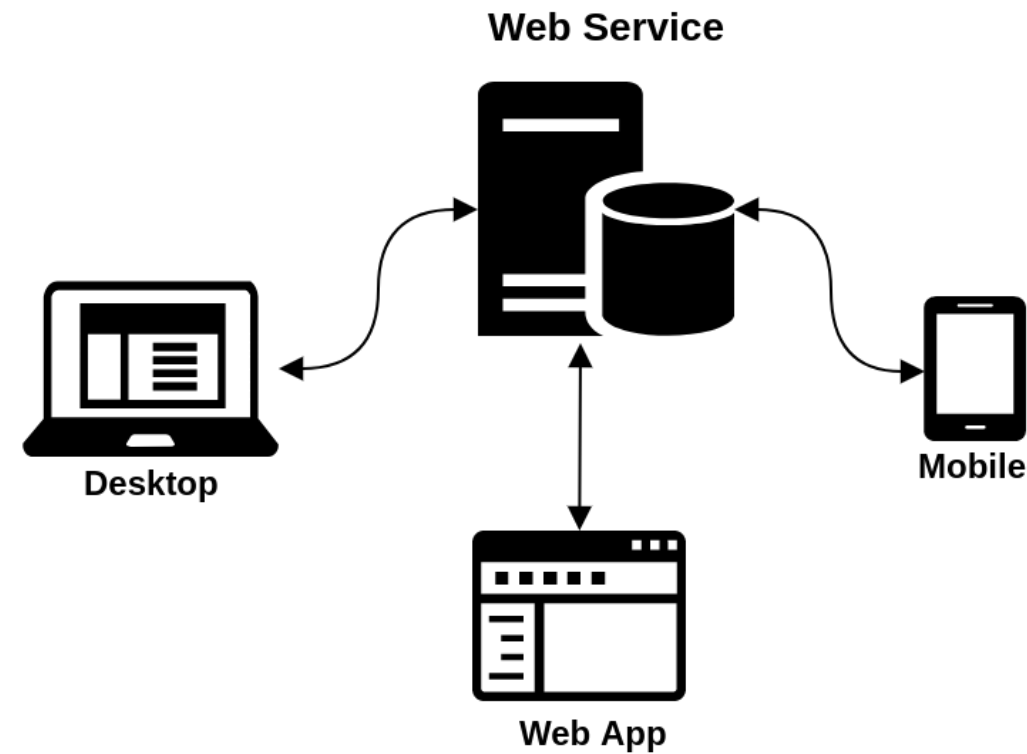
# AULA 09 - PROTOCOLO HTTP

Disciplina de Backend - Professor Ramon Venson - SATC 2024

# Serviços Web

O que é um web service?

- Troca de dados entre aplicações
- Geralmente intermediário
- Usa protocolos web (HTTP/HTTPS)
- Um tipo de Application Programming Interface (API)



## Exemplos de Web Services

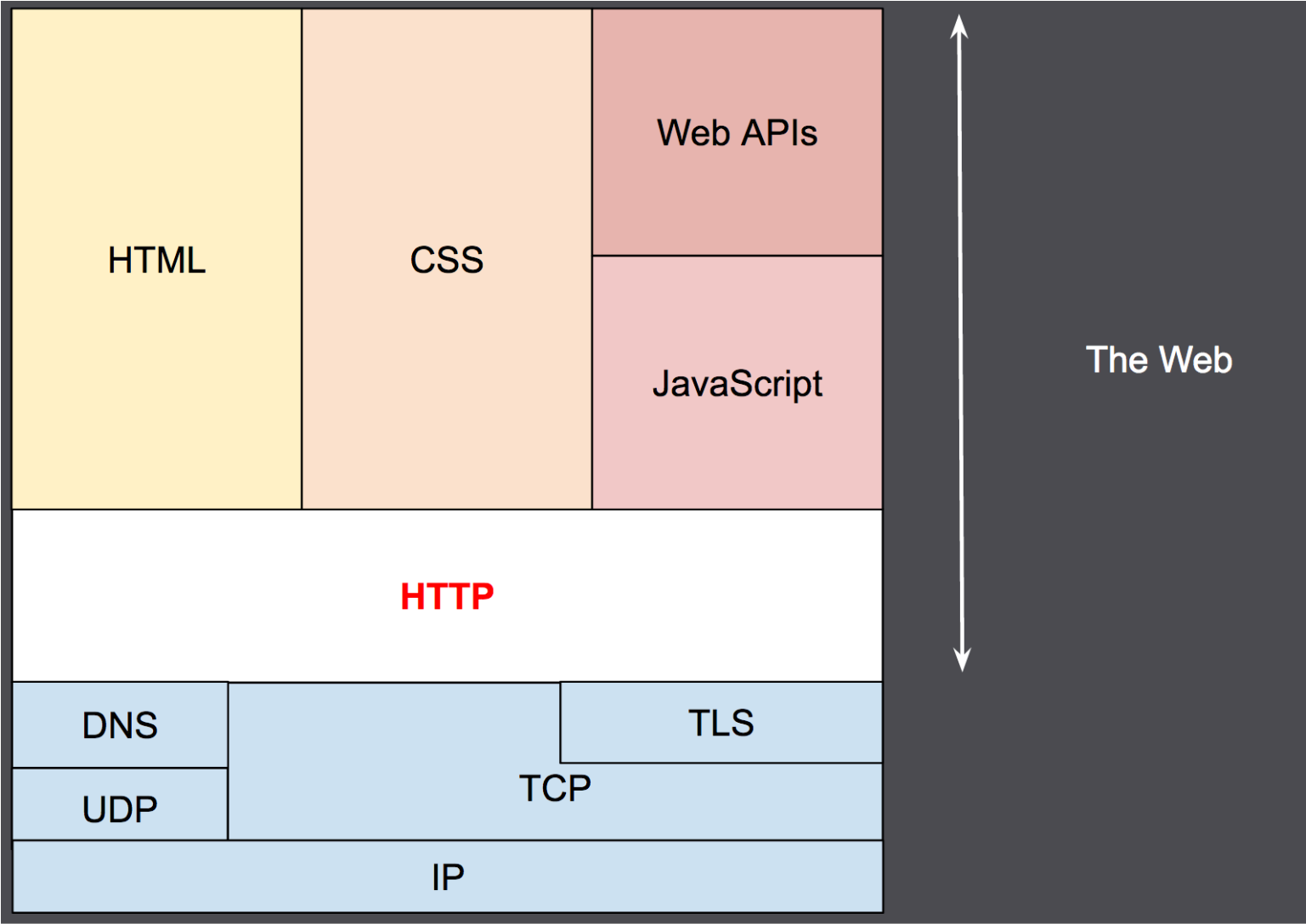
- Ler dados de sensores e fornecer dados para um aplicativo mobile
- Verificar e fornecer autenticação para outras aplicações
- Enviar mensagens através de um BOT para mensageiros instantâneos
- Receber e armazenar fotos de um aplicativo nativo para desktop
- Fornecer uma interface para acesso e gerenciamento de dados (REST)

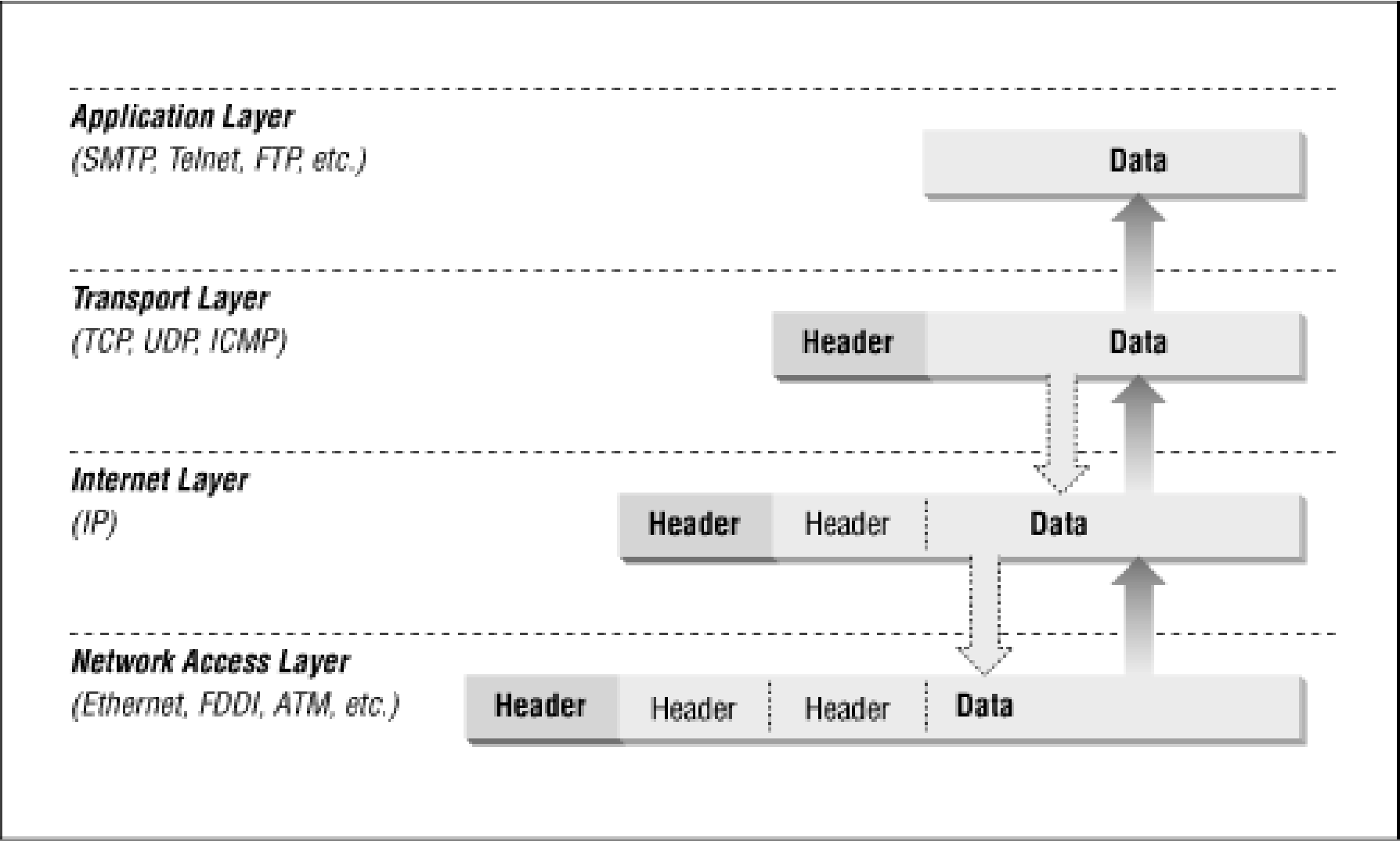
Atualmente podemos encontrar milhares de API disponíveis de maneira pública na Internet. Alguns exemplos:

- [Fotos de Cachorro](#)
- [Fotos de Gato](#)
- [Dados do Studio Gimbli](#)
- [Jogo de Cartas](#)
- [Tabela FIPE](#)
- Uma [outra lista](#) completa

# Protocolos

- TCP e IP , responsáveis pelo endereçamento e transmissão de pacotes;
- DNS , responsável pela tradução de endereços para nomes de domínio;
- HTTP e HTTPS , responsáveis pelo tráfego de mensagens entre aplicações;





## TCP e IP

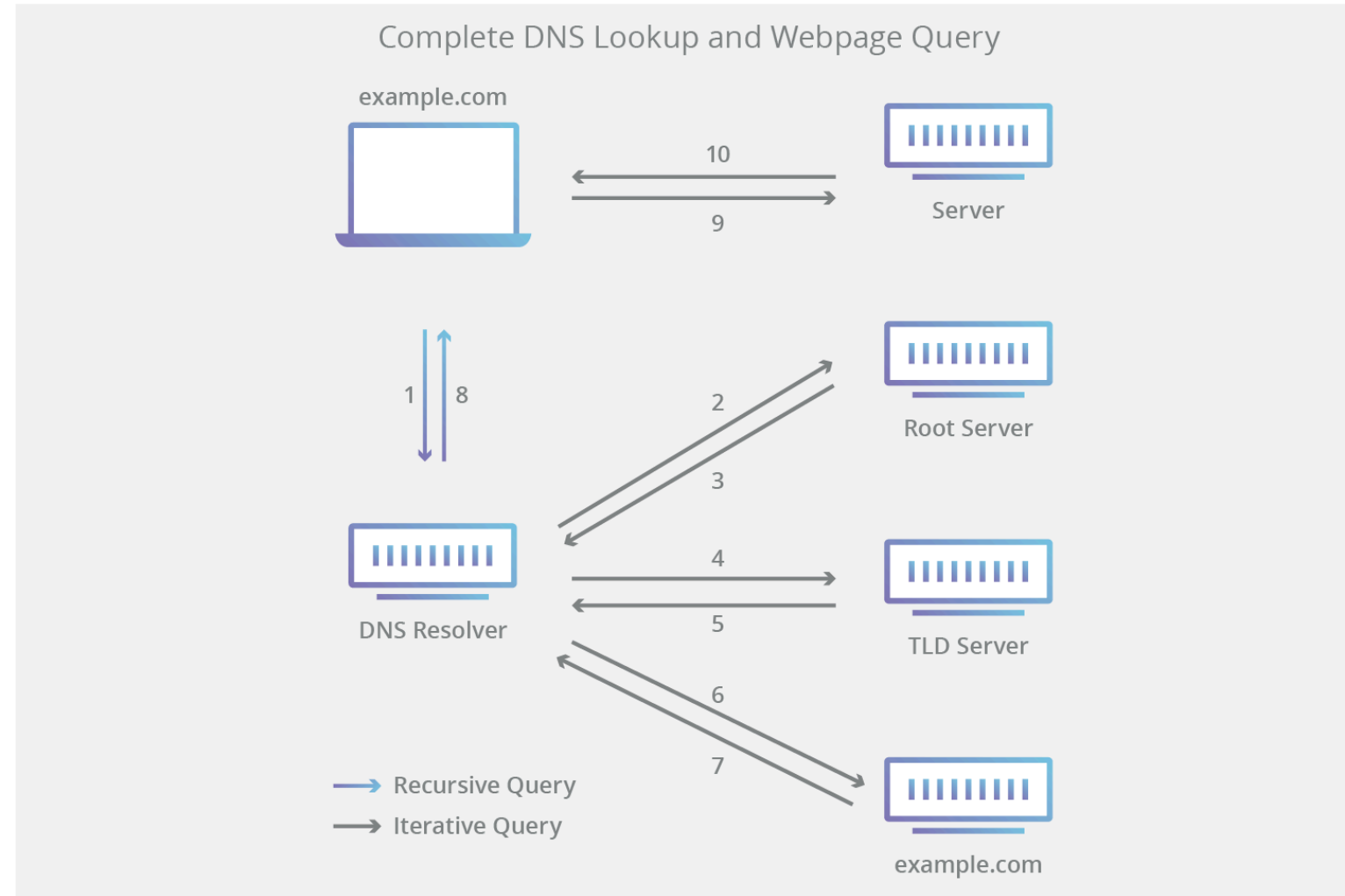
Os protocolos **TPC** e **IP** são geralmente utilizados como base para todas as trocas de dados em web services

- **TPC**: Garante integridade, ordem e entrega final dos pacotes
- **IP**: roteamento dos pacotes através do endereço IP



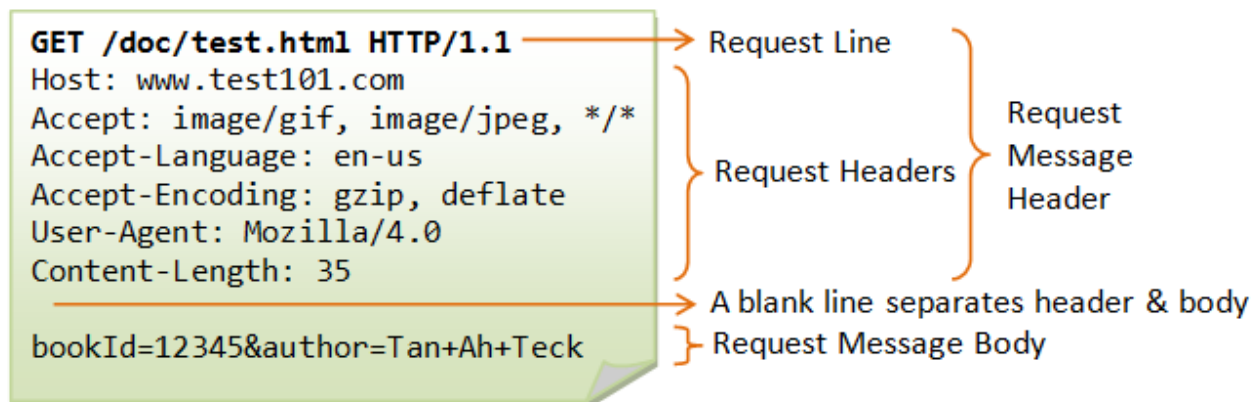
# DNS

Domain Name System,  
responsável pela  
tradução dos nomes de  
domínio (ex.:  
exemplo.com.br) para um  
endereço IP (ex.:  
20.30.2.1)



# Protocolo HTTP

- Camada de Aplicação
- Requisição/Resposta
- Protocolo sem estado (stateless)
- Porta 80



## Exemplo de Requisição HTTP

- Método + Caminho + Versão + Cabeçalhos + Corpo

```
POST /login HTTP/1.1  
Connection: Keep-Alive  
{user: "teste", password: "123456"}
```

Repare que não é necessário especificar o servidor (host) pois é a camada de rede/transporte o responsável por realizar a conexão

## Exemplo de Resposta HTTP

- **Versão + Status + Cabeçalhos + Corpo**

```
HTTP/1.1 200 OK
Host: example.com
Content-Length: 73
Date: Fri, 06 Aug 2030 01:31:51 GMT
<html><head><title>Introducao</head><body>Olá Mundo!</body></html>
```

## Tipos de Conteúdo

O formato do corpo da mensagem geralmente é identificado usando um cabeçalho chamado `Content-Type`. Exemplos:

- Texto plano genérico ( `text/plain` )
- Áudio ( `audio/mpeg` ), Vídeo ( `video/mp4` ) e Imagem ( `image/png` )
- Binário ( `application/octet-stream` )
- HTML ( `text/html` )
- Formulário ( `multipart/form-data` )
- JSON ( `application/json` )

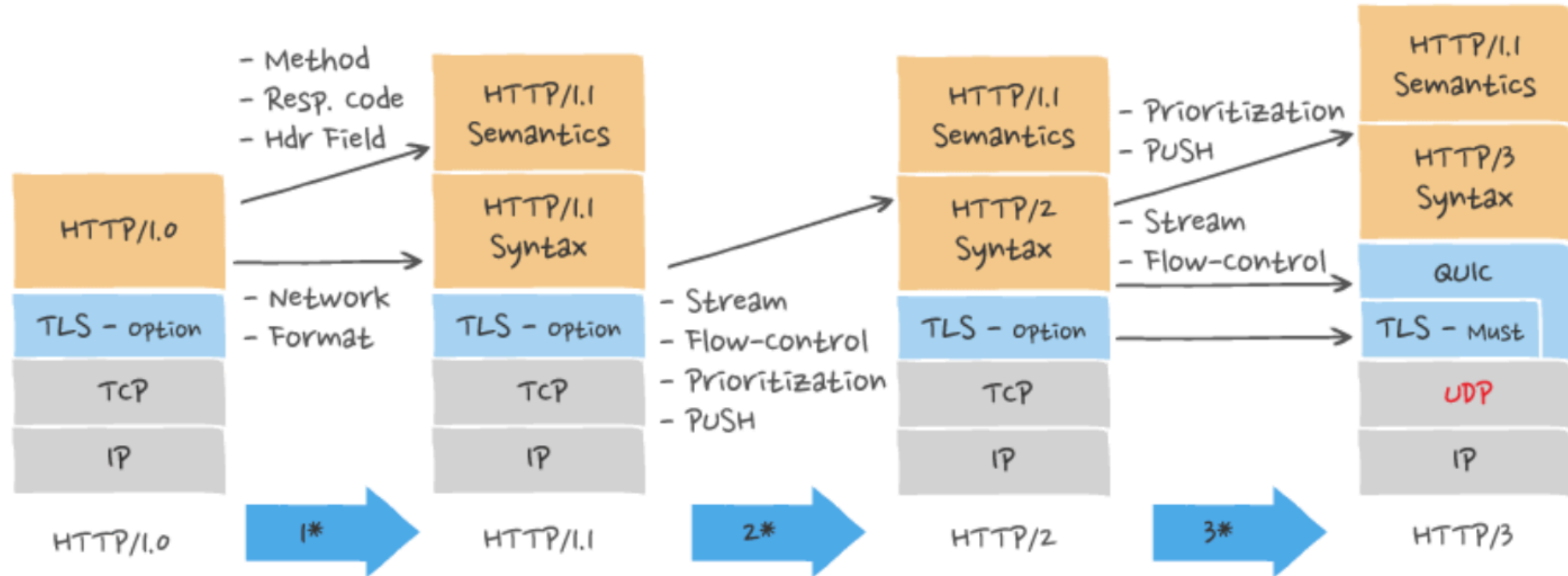
Estabelecendo uma conexão

Sempre que uma requisição HTTP é feita, é preciso que as camadas de rede/transporte realizem a conexão com o servidor antes de realizar qualquer requisição.

Conexões do tipo ***Keep-Alive*** podem ser utilizadas para múltiplas requisições.

# Comparação entre versões HTTP

HTTP protocol stack transition and comparison



- 1.0 : 1996
  - Pouco/Nenhum suporte a compressão
  - Apenas GET , HEAD e POST
- 1.1 (mais utilizado): 1997
  - Novos Métodos e Códigos de Resposta
  - Keep-Alive
  - Compressão aprimorada do corpo da mensagem



- 2.0 : 2015
  - Compressão do cabeçalho
  - Resposta e Requisições Multiplexada na camada de aplicação
    - Permite "enfileirar" requisições e respostas
    - Pacotes perdidos na camada de rede/transporte ainda bloqueiam as requisições
  - Server-Push (servidor envia respostas sem necessidade de requisições)

- 3.0 (draft): 2018
  - Troca do TCP+TLS para UDP+QUIC
  - Multiplexação não bloqueante na camada de transporte
    - Pacotes perdidos na camada de rede/transporte não bloqueiam outros pacotes

## HTTPS

- Extensão do protocolo HTTP
- Utiliza uma camada de segurança chamada *Transport Layer Security* (TLS)
- Negociação da criptografia usando chaves públicas
- Iniciada durante a fase de conexão
- Porta 443

O TLS é uma versão mais recente do *Secure Sockets Layer* (SSL), termo ainda usado para descrever a mesma camada.

## O que aprendemos nessa aula?

- O que são serviços web e exemplos
- Quais os protocolos mais utilizados por aplicações web
- Como são realizadas as requisições HTTP
- Quais ferramentas vamos utilizar para construir aplicações web