Modelos arquiteturais para comunicação entre sistemas

Sistemas embarcados

Prof. Allan Rodrigo Leite

Modelos de comunicação entre sistemas

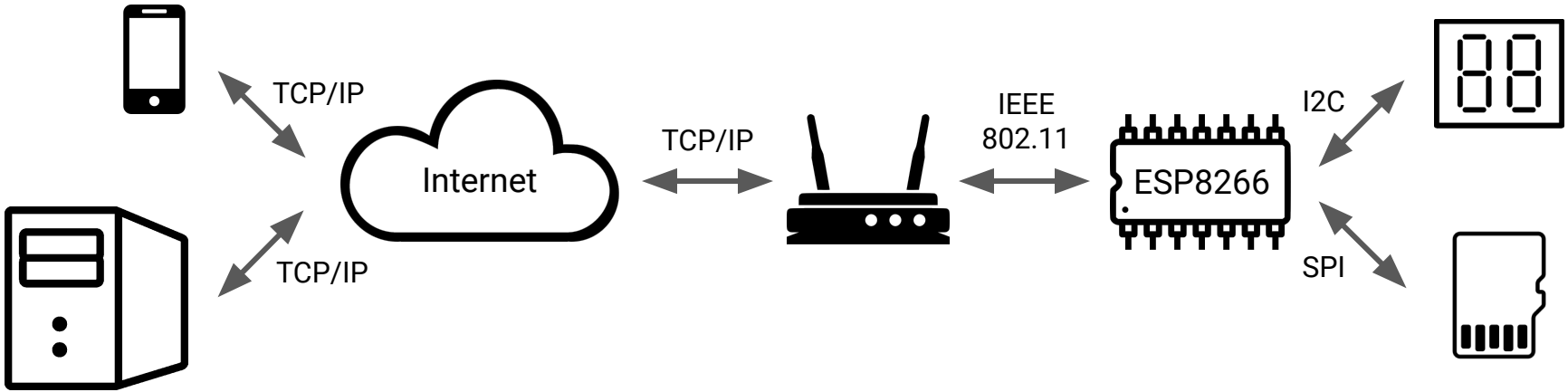
- Comunicação síncrona
  - Emissor e receptor estão em um estado de sincronia antes da comunicação iniciar e permanecem em sincronia durante a transmissão

- Comunicação assíncrona
  - Emissor e receptor realizam uma comunicação cujos dados podem ser transmitidos intermitentemente em um fluxo estável

Modelos de comunicação entre sistemas

- Modelos de comunicação entre dispositivos em sistemas embarcados
  - UART, I2C e SPI são exemplos de protocolos

- Comunicação entre sistemas distribuídos
  - Utilizando um modelo de comunicação de alto nível, por exemplo, WiFi



Arquitetura de software

- Definição dos componentes de software, propriedades externas e relacionamentos com outros softwares

- Exemplos de arquiteturas de software
  - Monolíticas
  - Cliente/Servidor
  - P2P
  - Arquitetura orientada a serviços
  - Arquitetura orientada a eventos
  - Microsserviços

Arquitetura orientada a serviços

- Estilo arquitetural para sistemas distribuídos
  - Conhecido por SOA (*Service Oriented Architecture*)
  - Funcionalidades providas são disponibilizadas na forma de serviços
  - Baseado no modelo request/response
    - Estabelece a comunicação entre clientes e provedores dos serviços

- Serviços podem ser conectados através de um barramento de serviços
  - Possuem contratos bem definidos e acessíveis através de Web Services
  - São interoperáveis que podem ser reutilizados e compartilhados

Arquitetura orientada a serviços

- Serviço é uma função de um sistema computacional disponibilizada para outro sistema
  - Deve funcionar de forma independente do estado de outros serviços

- Um serviço deve possuir uma interface bem definida
  - A comunicação entre o sistema cliente e quem disponibiliza o serviço é realizada através de Web Services
  - São serviços funcionais disponíveis a partir de protocolos da Internet
  - São independentes de plataformas ou linguagens de programação

Arquitetura orientada a serviços

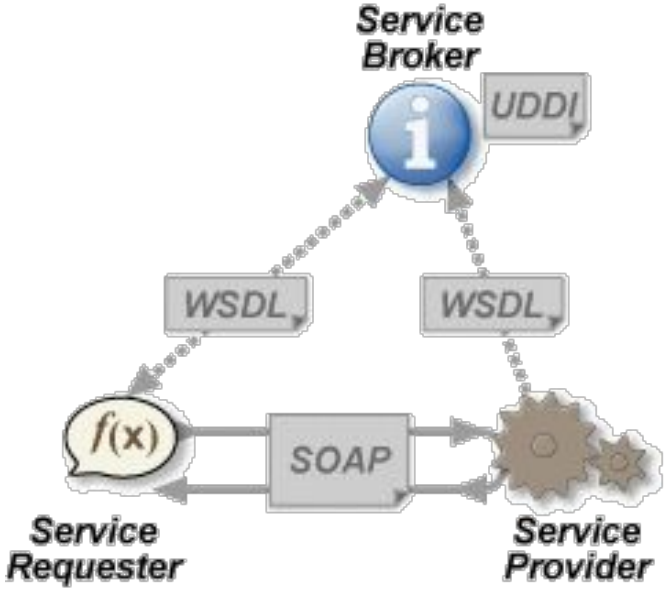
- Características de um Web Services
  - *Stateless*
    - Os serviços não dependem de nenhuma condição pré-existente
  - Coesão e acoplamento dos serviços
    - Uma das características do SOA é o baixo acoplamento
    - Um serviço é considerado coeso ao possuir uma finalidade bem definida
  - Tecnologia utilizada para prover o serviço
    - Não deve fazer parte da definição do serviço

Arquitetura orientada a serviços

- Componentes de um Web Service
  - Service Provider
    - Quem oferece os serviços (provedor) de um Web Service
    - Possui uma interface bem definida para acesso à informação e do serviço
  - WSDL
    - Descritor do serviço (Web Services Description Language)
  - SOAP ou REST
    - Padrões arquiteturais de comunicação baseado no protocolo HTTP
  - XML, YAML ou JSON
    - Meio de representação dos dados

Web Service SOAP

- SOAP (*Simple Object Access Protocol*)

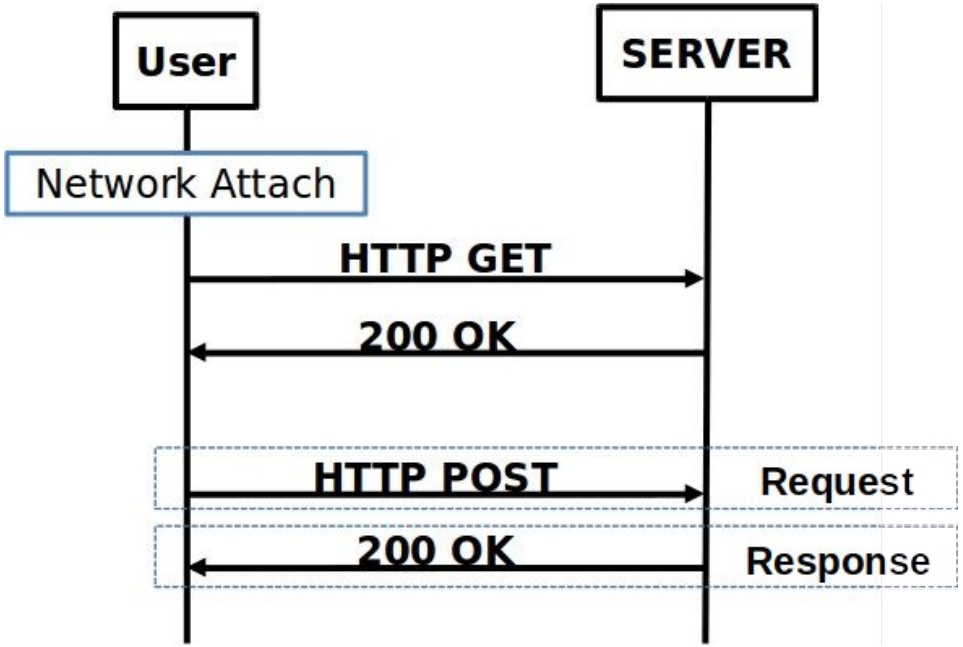


Web Service REST

- Anatomia do protocolo HTTP
  - Localização e endereço de acesso
    - URL, URN e URI
  - Mensagem
    - Cabeçalho, corpo e requisição (solicitação e resposta)
  - Métodos
    - GET, POST, PUT, DELETE, OPTIONS, PATCH
  - Códigos de estados
    - Sucesso, não autorizado, erro interno, recurso não encontrado

Web Service REST

- REST (*REpresentational State Transfer*)



Serviços IFTTT

- IFTTT (*If This Then That*)
  - Plataforma gratuita para automatizar tarefas baseadas em serviços web
  - Serviços web para criar cadeias de instruções de condições simples
    - Permite conectar diferentes serviços por meio de comandos e condições

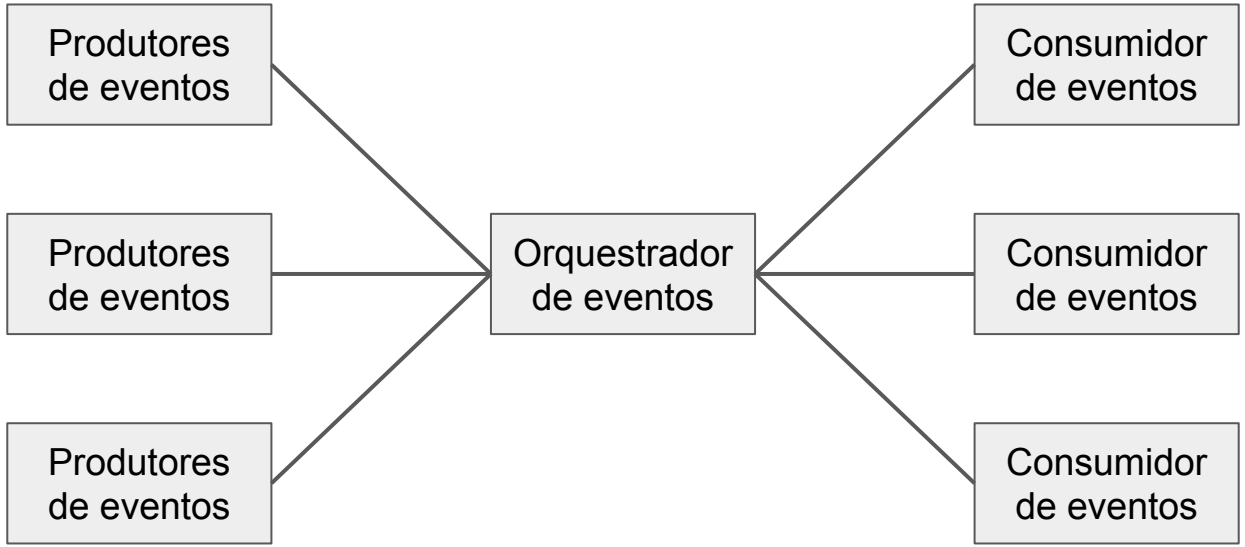
- Componentes do IFTTT
  - Receitas
    - São combinações entre comandos e condições (se isso acontecer)
    - Especifica o que é necessário para tomar alguma ação e o que fazer
  - Ações
    - São gatilhos disparados quando satisfeitas as condições (então faça aquilo)
    - Especifica o que o dispositivo deverá fazer em resposta à receitas

Arquitetura orientada a eventos

- Tem por objetivo segregar os módulos de uma aplicação em
  - Produtores: geradores de um fluxo de eventos
  - Consumidores: escutam e atuam sobre determinados eventos

- Evento representa um conjunto de alterações nos dados
  - Eventos são entregues quase em tempo real
  - Requer um orquestrador que organize os eventos em uma fila
  - A comunicação dos eventos são realizadas por meio de mensagens
    - As mensagens devem possuir um formato predefinido

Arquitetura orientada a eventos



Arquitetura orientada a eventos

- Modelo *publish/subscribe*
  - Troca de mensagens a partir de um controle de assinaturas

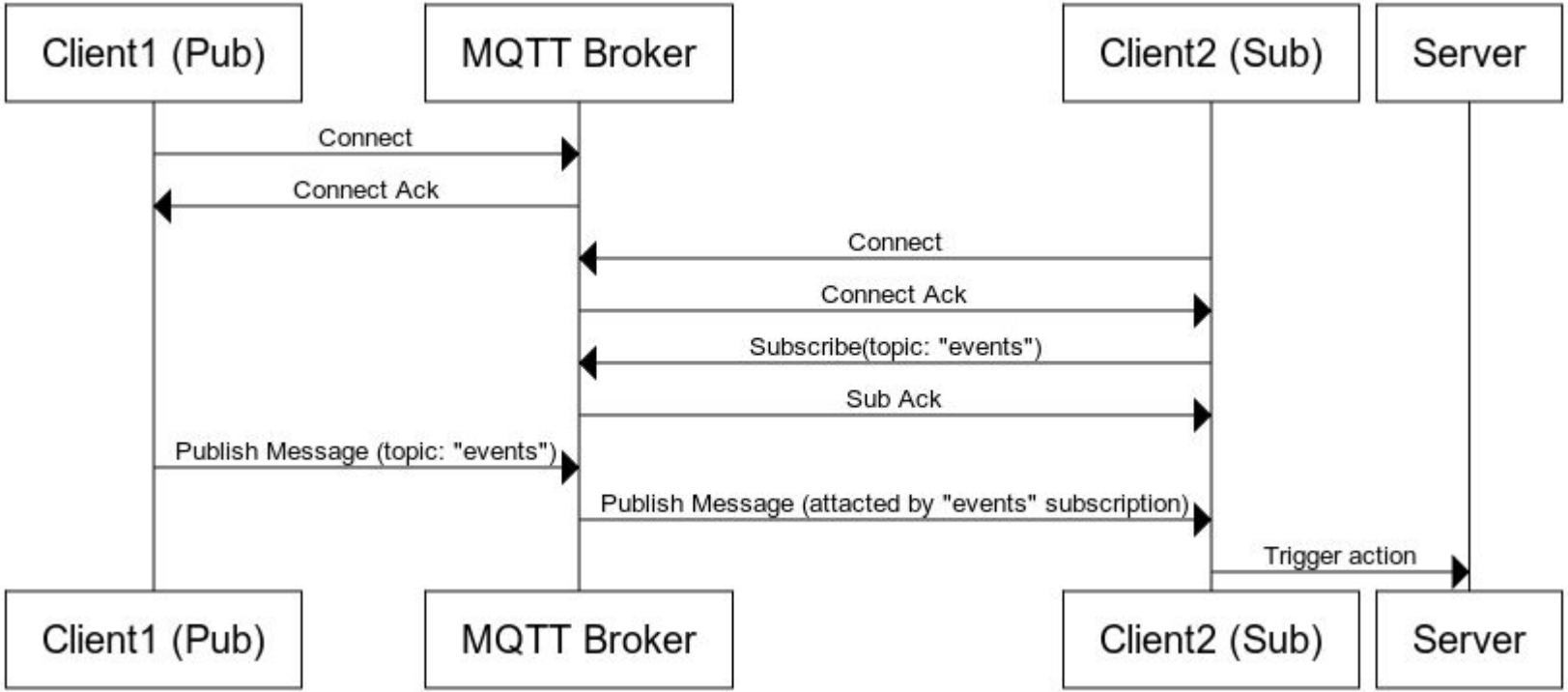
- Fluxo de mensagens
  - Cada evento refere-se a um determinado tópico
  - Quando um novo evento é publicado, o orquestrador notifica os assinantes deste tópico
  - Ao receber uma notificação, esta mensagem não pode ser mais exibida

Arquitetura orientada a eventos

- MQTT (Message Queuing Telemetry Transport)
  - Protocolo de mensagens leve para dispositivos IoT
  - Implementa o modelo *publish*/*subscribe*

- Principais métodos
  - CONNECT: solicita uma conexão ao servidor
  - PUBLISH: publicar mensagem de um dado tópico
  - SUBSCRIBE: inscrever-se em um tópico
  - UNSUBSCRIBE: retirar inscrição de um tópico

Arquitetura orientada a eventos



Modelos arquiteturais para comunicação entre sistemas

Sistemas embarcados

Prof. Allan Rodrigo Leite