Reunião de 25/06/2020
MISIoT

<brunocarneirodacunha@usp.br>
<bruno.rotondaro@unifesp.br>

30 de junho de 2020

1 Resumo

1.1 Arquitetura

Durante a reunião, o consenso entre Diego e Higor foi que não é necessário criar um micro-serviço separado para fazer a integração ao MQTT. É mais recomendável somente modificar o serviço *resource-adaptor* existente, mantendo a funcionalidade dos testes de integridade atuais.

Uma possibilidade levantado pelo Diego foi escrever uma *gem* do Ruby. Assim seria possível rodar o nosso código em diferentes partes da plataforma, simplesmente incluindo essa *gem*.

1.2 Kong Gateway

Diego mencionou que o uso do *Kong* na plataforma pode ser um problema, já que o mesmo captura todo o tráfego e redireciona de acordo com a URL do request. Esse serviço não suporta o protocolo MQTT, e qualquer tráfego que não utilize HTTP será ignorado.

Uma solução levantada seria implementar MQTT no *Kong*, a outra seria abrir uma outra instância do RabbitMQ que não esteja debaixo do gateway, e possa ser acessada livremente por dispositivos externos.

Também foi falado que a plataforma não é compatível com HTTPS, mas que provavelmente não é muito complicado fazer o *Kong* aceitar esse tipo de tráfego.

1.3 Autenticação e Autorização

Ficou claro na conversa que não há, no momento, nenhum tipo de mecanismo no INCT que preveja a autenticação dos usuários, e a autorização de cada um quanto à leitura e escrita dos recursos.

2 Recomendações

Quanto ao problema do tráfego de MQTT no *Kong*, é possível que tenha acontecido algum tipo de engano na argumentação do Diego. Não parece haver motivo pelo qual não seria possível o *resource-adaptor* abrir uma porta aleatória na instância do INCT que estaremos modificando. Atualmente, o *Kong* somente captura o tráfego que chega na porta 80, estando as outras livres para interagir com processos na Internet.

Somente se justificaria caso não fosse desejável abrir *nenhuma* outra porta no INCT, o que não é possível para a nossa aplicação. O caminho recomendado é simplesmente ignorar o Kong para o tráfego MQTT.

Em relação ao uso do HTTPS, será necessário gerar certificados, e configurar o *Kong* para usar esses certificados.

Também recomendo que implementemos a autenticação do usuário usando chaves de API, como muitos serviços de backend fazem. Usando essa chave, o usuário poderá ser identificado nos requests, e a plataforma poderá controlar o acesso do mesmo aos recursos.

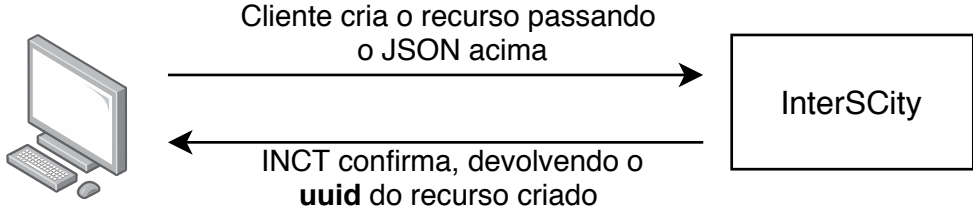
3 Esboço do adaptador para integração ao TTN

A partir das discussões da reunião, propomos o seguinte adaptador para integrar dispositivos registrados no *The Things Network* à plataforma.

3.1 Criação do Recurso

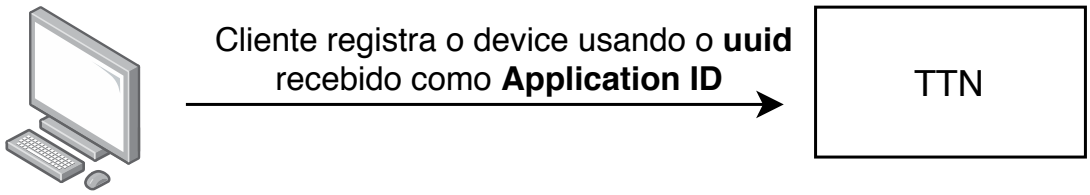
```
{
    "data": {
        "description": "A public bus",
        "capabilities": [
            "temperature",
            "humidity",
            "illuminate"
        ],
        "status": "active",
        "lat": -23.559616,
        "lon": -46.731386
    },
    "adapter": {
        "protocol": "TTN"
        "address": "brazil.thethings.network"
        "port" : "8883"
        "appid": "misiot"
        "appkey": "ttn-key-********"
    }
}
```

Ao criar um recurso, o usuário que deseja acessar dados de uma *Application* do TTN deve também passar o objeto **adapter**, com os parâmetros **protocol**, **address**, **port** (opcional), **appid** e **appkey**. Será necessário manter o certificado usado na conexão MQTT-TLS na plataforma.

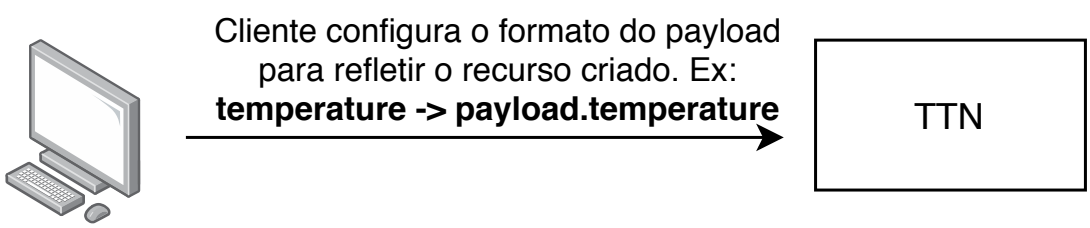


3.2 Configuração da *Application* no TTN

3.2.1 Registro de um novo device



3.2.2 Formatação do *payload*



3.3 Interação entre TTN e o INCT