

## Laboratório 3

### Objectivos:

- Implementação de uma solução cliente servidor, designada **Forum**, baseada no padrão *Publish/Subscribe* usando o *middleware* Google RPC (gRPC).

O padrão Publish/Subscribe é um paradigma usado em múltiplos sistemas onde utilizadores expressam os seus interesses (*subscriptions*) em determinados conteúdos ou tópicos (*publications*). Um intermediário (*mediator ou broker*) mantém estado sobre os interesses dos múltiplos *subscribers*, garantindo que os eventos emitidos pelos vários *publishers* são notificados aos interessados.

Usando o *middleware* gRPC, desenvolva um sistema **Forum** de troca de mensagens entre utilizadores como se apresenta na Figura 1 e com os requisitos indicados de seguida:

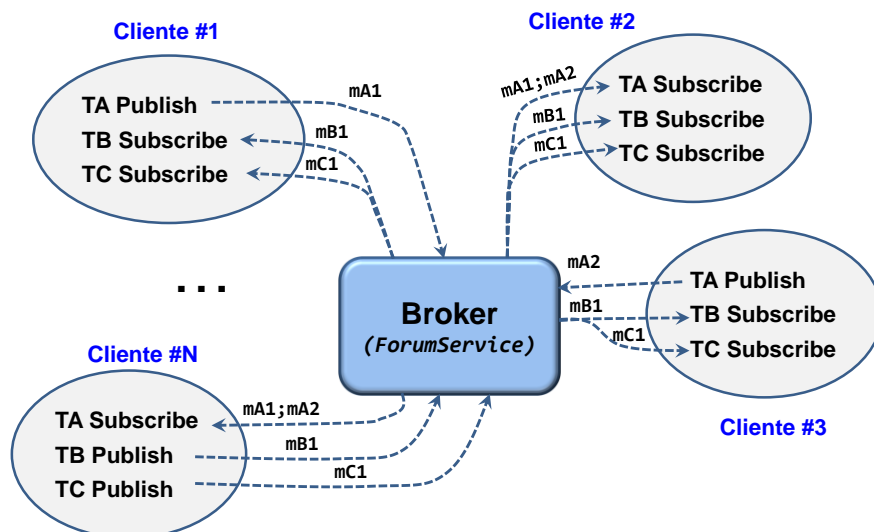


Figura 1 – Vista geral do sistema *Forum*

- Existe um servidor (*Broker*) que implementa um serviço gRPC, que gere as interações entre utilizadores, para que os utilizadores possam trocar mensagens associadas a um tópico (ex. desporto, isel, cloud, política, etc.):
- Uma mensagem (*mA1*, *mA2*, *mB1*, *mC1*, etc.) contém sempre o tópico, o nome do utilizador que a publicou e o conteúdo da mensagem que é um texto;
- Um utilizador, identificado por um nome único, pode publicar (*Publish*) ou ser subscritor (*Subscribe*) de mensagens em vários tópicos, podendo em qualquer altura subscrever ou retirar a subscrição sobre um ou mais tópicos.
- Assume-se a existência de um tópico no servidor assim que um primeiro cliente subscreva o tópico;
- Assume-se que não há armazenamento de mensagens no servidor, isto é, quando um cliente publica (*Publisher*) uma mensagem no servidor, este limita-se a encaminhar a mensagem para todos os subscritores do tópico associado à mensagem. Se um cliente publicar uma mensagem num tópico sem subscritores deve receber uma exceção;

- Assim o serviço (*ForumService*) deverá implementar as seguintes operações:
  - **TopicSubscribe**: inscrever um tópico
  - **TopicUnsubscribe**: retirar a subscrição de um tópico
  - **getAllTopics**: obter a lista de tópicos já existentes
  - **MessagePublish**: publicar uma mensagem

### Sugestões:

- Na definição das operações em linguagem *proto* sugere-se a utilização das seguintes mensagens;

<pre>message SubscribeUnSubscribe {     string usrName = 1;     string topicName = 2; }  message ExistingTopics {     // lista de topic names     repeated string topicName = 1; }</pre>	<pre>message ForumMessage {     string fromUser = 1;     string topicName = 2;     string txtMsg = 3; }</pre>
--	---

- Crie um projeto para criar a biblioteca com o contrato entre a aplicação cliente e o servidor (Broker com implementação do *ForumService*), discutindo com o professor a especificação proto das operações RPC;
- Desenvolva as aplicações Cliente e servidor e realize os testes necessários executando o servidor e vários clientes na sua máquina local;
- Após o bom funcionamento local, realize o *deployment* do servidor na VM Ovirt do grupo e convide os seus colegas a visitarem o seu servidor para trocarem mensagens;
- Crie uma VM na *Google Cloud Plataform* e convide os seus colegas, mesmo fora da rede do ISEL a visitarem o seu servidor para trocarem mensagens.

## Nota Final:

Este trabalho de Laboratório terá como ponto de entrega o seguinte:

- O formulário de avaliação do desenvolvimento do trabalho
- Um ficheiro zip com o código da solução desenvolvida