Cenario 2   
  
MFE -> BFF ->API Manager  
Para um iniciante em Java, entender o fluxo de uma API com auditoria pode parecer complexo, mas vamos quebrar isso em uma história lógica. Imagine que o seu sistema é como um **Banco Físico**.

Aqui está o passo a passo do que acontece quando o Front-end faz uma requisição:

**🏛️ 1. O Cenário: O Balcão do Banco**

1. **O Cliente (Front-end):** Chega no balcão e pede: *"Quero ver os contratos do CPF 123"*.
2. **O Atendente (Sua API REST/Controller):** Recebe o pedido. Ele não sabe as respostas de cabeça, então ele precisa consultar o arquivo central.
3. **O Registro (OpenTelemetry):** Assim que você começa a falar com o atendente, uma câmera de segurança (OTel) começa a gravar.

**🔄 2. O Fluxo Passo a Passo**

**Passo A: A Chegada (Request)**

O Front-end envia uma mensagem HTTP. O Java recebe isso no Controller.

* **Ação de Auditoria:** O sistema cria um "envelope" chamado **Span**. Dentro dele, ele anota: *"Início do atendimento para o CPF 123"*.

**Passo B: A Consulta (Service + Feign)**

O seu código (Service) chama um Microserviço externo usando uma ferramenta chamada **Feign**. É como se o atendente usasse um rádio para falar com a central.

* **Ação de Auditoria:** O "estagiário" (OpenTelemetry) anota no envelope: *"Pedindo dados para a Central de Contratos agora"*.

**Passo C: A Resposta da Central**

A central responde com os dados. O seu Java pega esses dados, organiza e prepara para entregar ao cliente.

* **Ação de Auditoria:** O sistema anota: *"Central respondeu com sucesso. Encontramos 2 contratos"*.

**Passo D: A Entrega e o Registro Final (Event Hub)**

O Java envia a resposta (JSON) de volta para o Front-end.

* **Ação de Auditoria:** O envelope (Span) é fechado. Agora, esse registro completo é enviado para o **Azure Event Hub**. É como se, ao fim do dia, a gravação da câmera fosse enviada para um cofre digital seguro na nuvem.

**📊 3. Diagrama do Fluxo de Informação**

**🕵️ 4. Em que momento a trilha é gravada?**

Essa é a dúvida mais comum! A trilha é **construída** durante todo o processo, mas ela é **enviada (gravada definitivamente)** em dois momentos principais:

1. **No Sucesso:** Quando o método termina e o return é executado. O sistema entende que a "viagem" acabou e despacha os dados para o Event Hub.
2. **No Erro (Try-Catch):** Se o microserviço cair ou o CPF for inválido, o Java entra no bloco catch. O OpenTelemetry captura o erro e envia a trilha imediatamente avisando: *"Atenção, a operação falhou no passo B pelo motivo X"*.