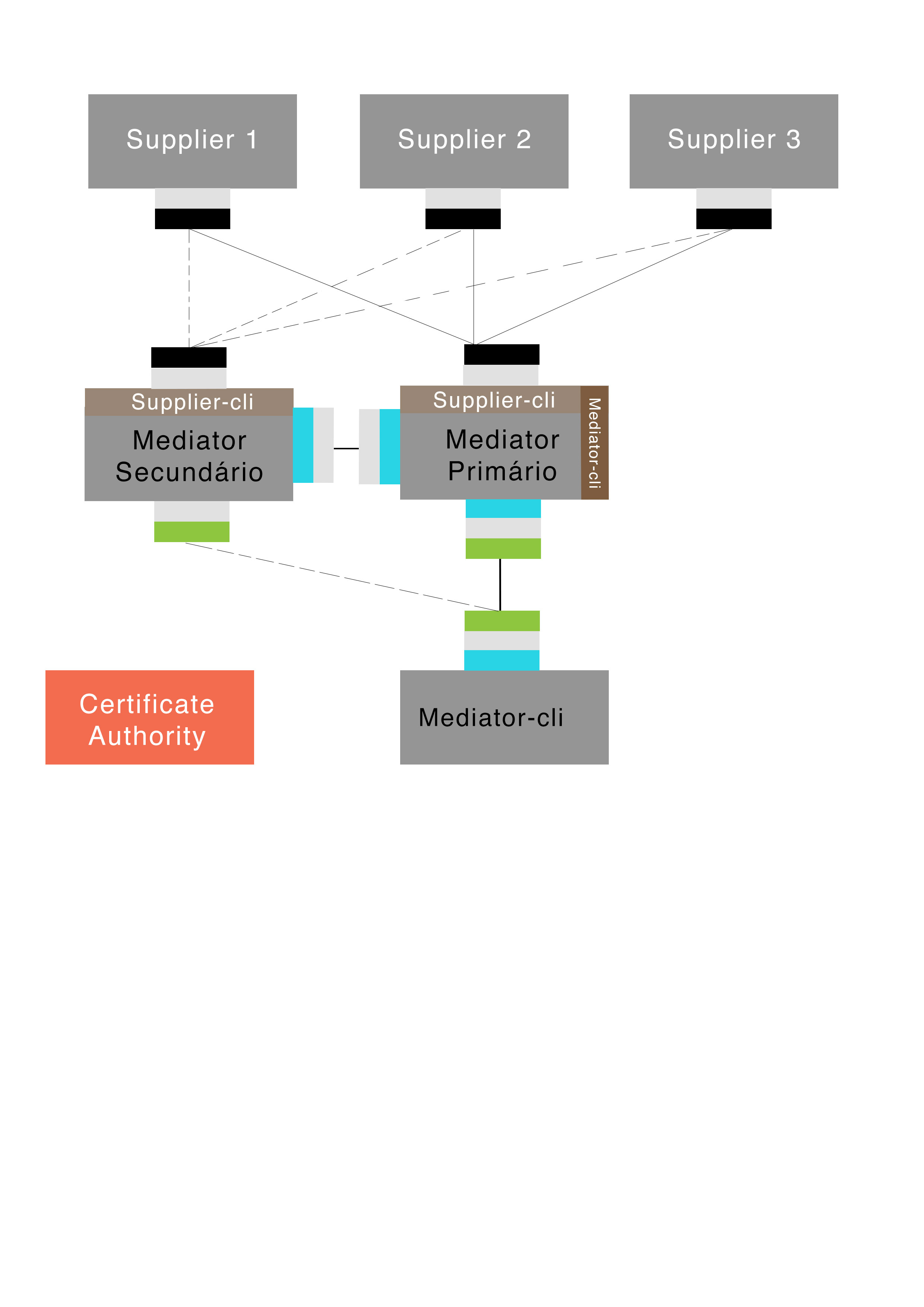


Sistemas Distribuídos

|  |  |  |
| --- | --- | --- |
| Grupo T04  Repositório GitHub: <https://github.com/tecnico-distsys/T04-Komparator> | | |
|  |  |  |
| 81900 – Nuno Anselmo | 81936 – Liliana Oliveira | 82047 – André Mendes |

**Pressupostos:**

Ligações que só são estabelecidas quando o Mediador Primário falhar

Ligações estabelecidas inicialmente

---

\_\_\_

AtMostOnceHandler

TimestampHandler

SignatureHandler

CipherHandler

Módulos

**Legenda:**

* Certificate Authority tem ligação a todos os módulos.
* Foi ignorada a representação do LoggingHandler visto não ter qualquer impacto no funcionamento da tolerância a faltas.
* Quando o Mediator-ws-cli faz um pedido ao Mediator Primário, o Secundário encontra-se ligado.
* A ligação entre Mediator Primário e Secundário é não necessita de ser protegida, e a localização do Mediator Secundário é conhecida *a priori*.

**Descrição:**

Mediator Primary - Responsável por atender todos os pedidos do Mediator-cli. Igualmente responsável por avisar um mediator secundário de todos os pedidos (addToCart e buyCart) e por notificar o mesmo que (ainda) está vivo (*imAlive*).

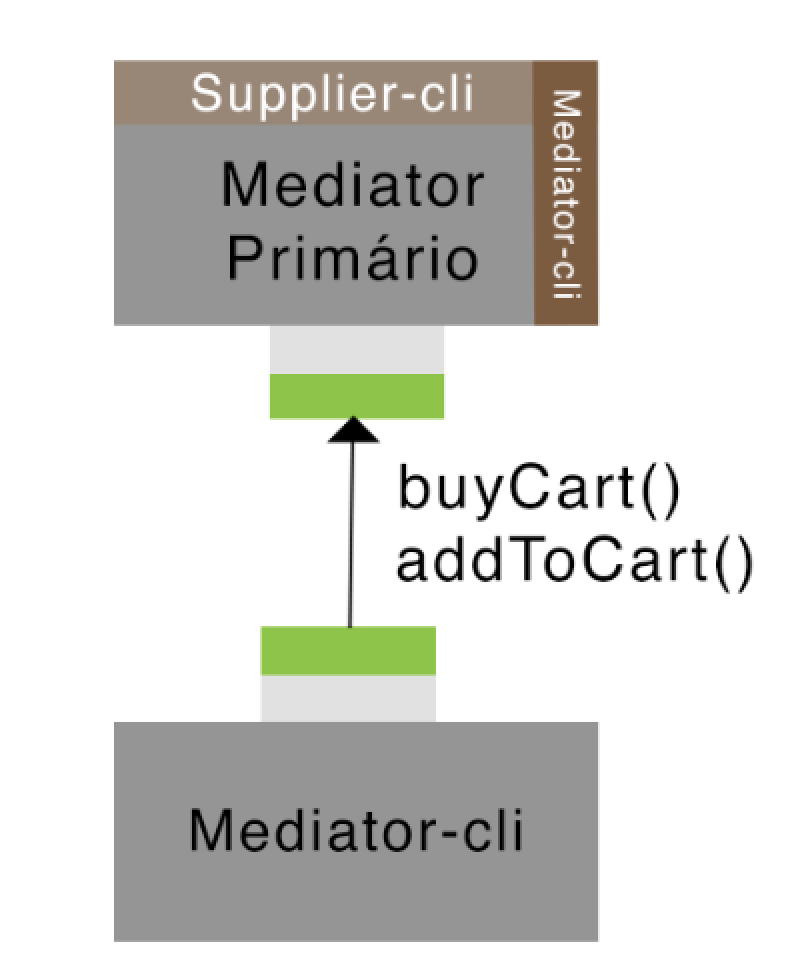
Mediator Secondary – Responsável por receber os pedidos de atualização pelo mediator primário e com os mesmos guardar uma cópia do estado do mediator primário. Responsável também por substituir o mediator primário quando este deixar de dar provas de vida (*imAlive*); caso isto aconteça, o mediator-cli restabelece ligação com o mediator que se publicar como primário.

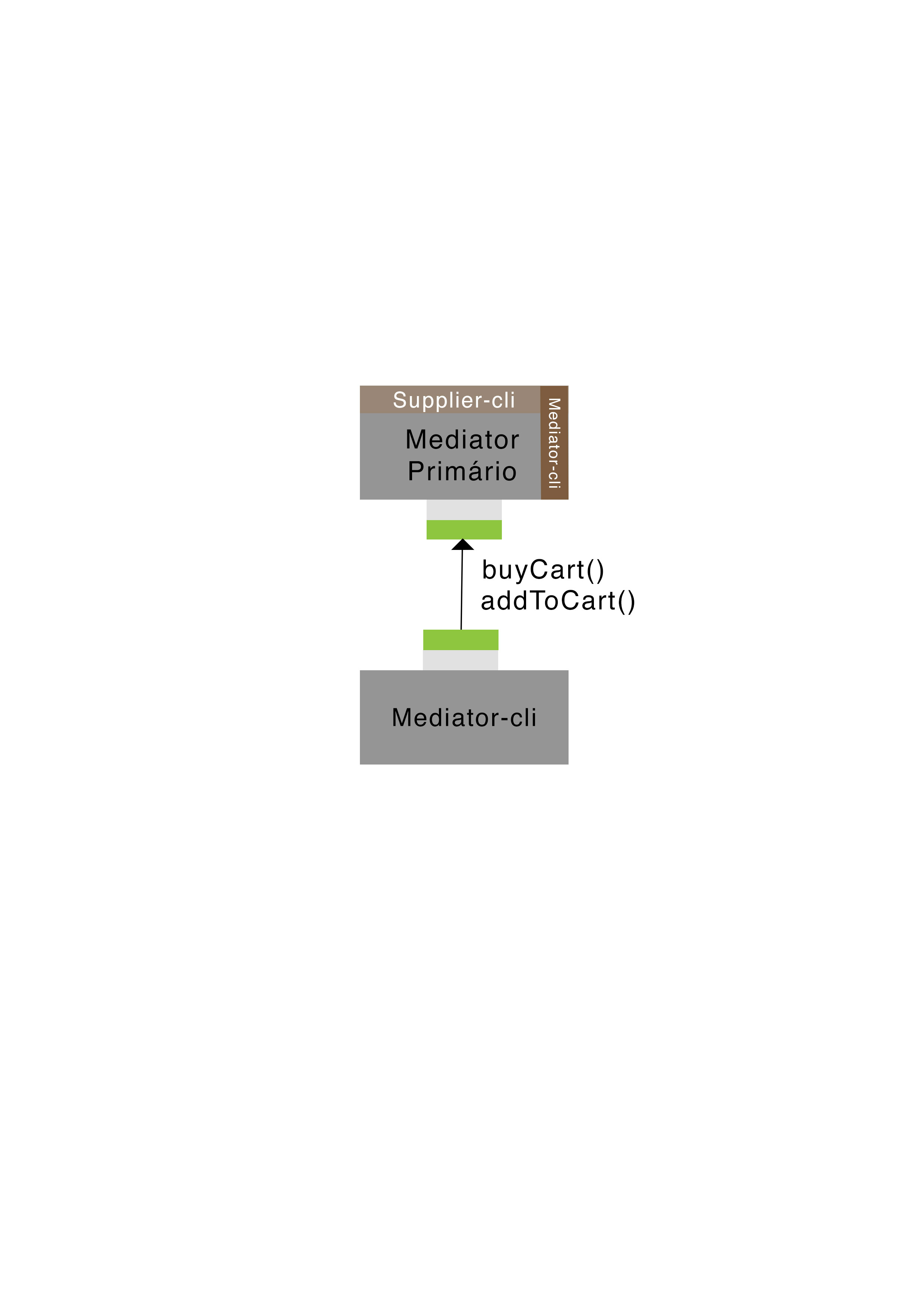
Semântica de “no máximo uma vez”: esta semântica é garantida pelo handler “AtMostOnceHandler”. A garantia é feita ao inserir um identificador em todos os pedidos, que é sincronizado entre mediators quando é executada uma operação que não seja idempotente. O identificador é reutilizado pelo mediator-cli caso não seja obtida nenhuma resposta. Caso seja recebido o mesmo identificador, a acção não é executada e o mediator-cli é informado. Nesse momento, o mediator-cli poderá repetir o pedido (caso seja idempotente), ou tratar a falha como for adequado.

**Troca de mensagens (em paralelo)**

**Execução de pedido:**

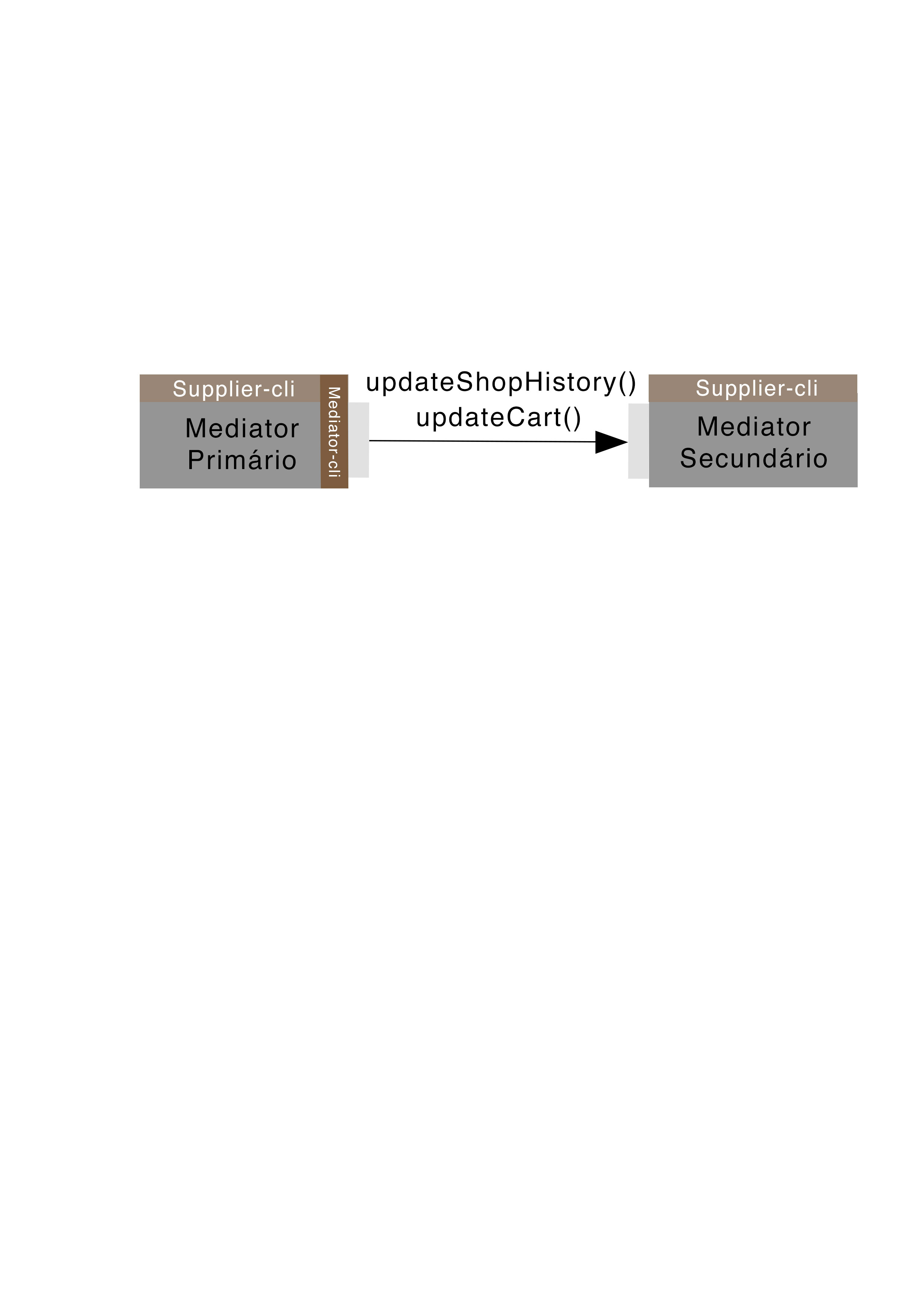
**Passo 1:**

Cliente envia pedido ao mediator

**Passo 2:**

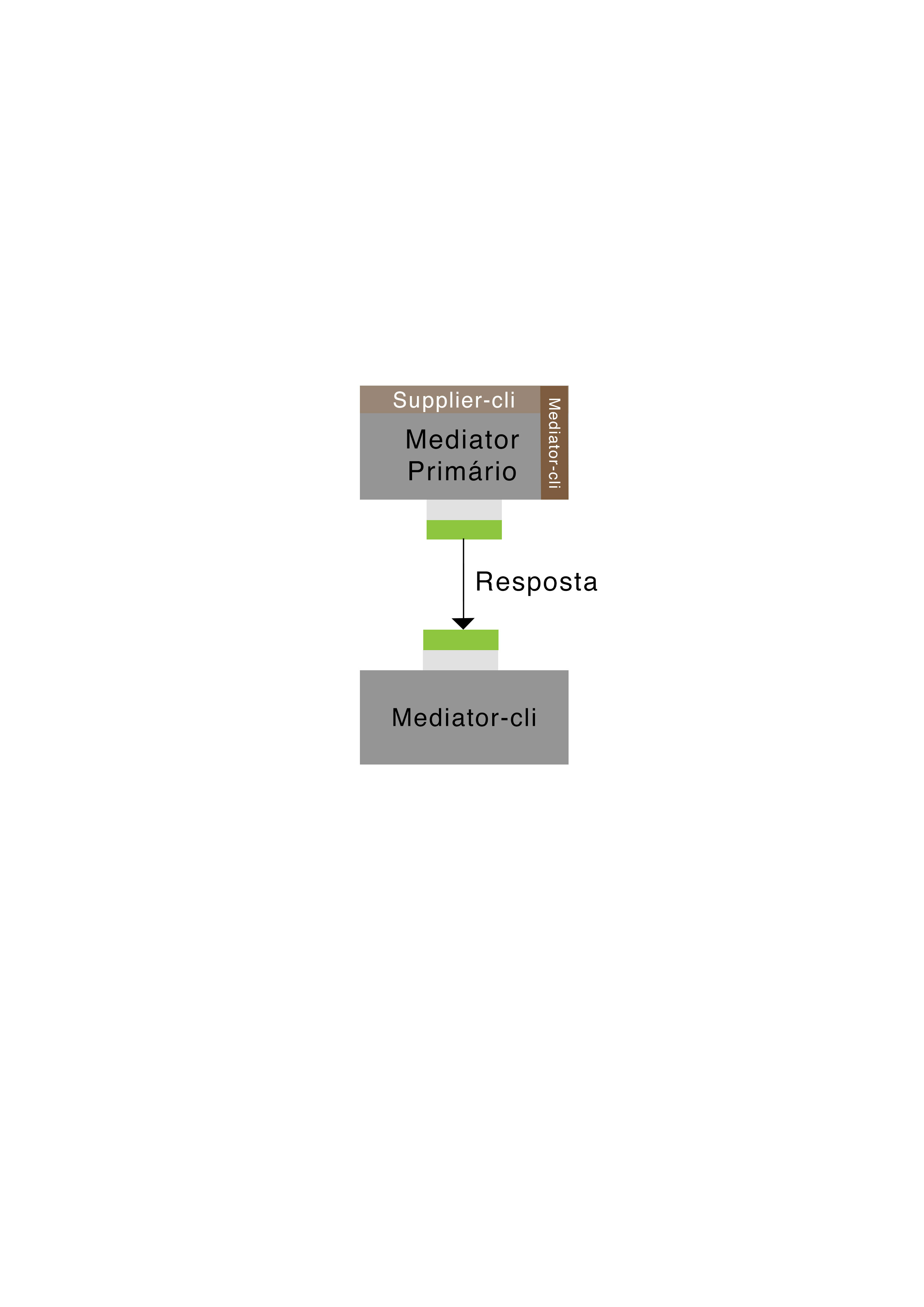
Mediator efectua o pedido, acedendo aos suppliers

**Passo 3:**



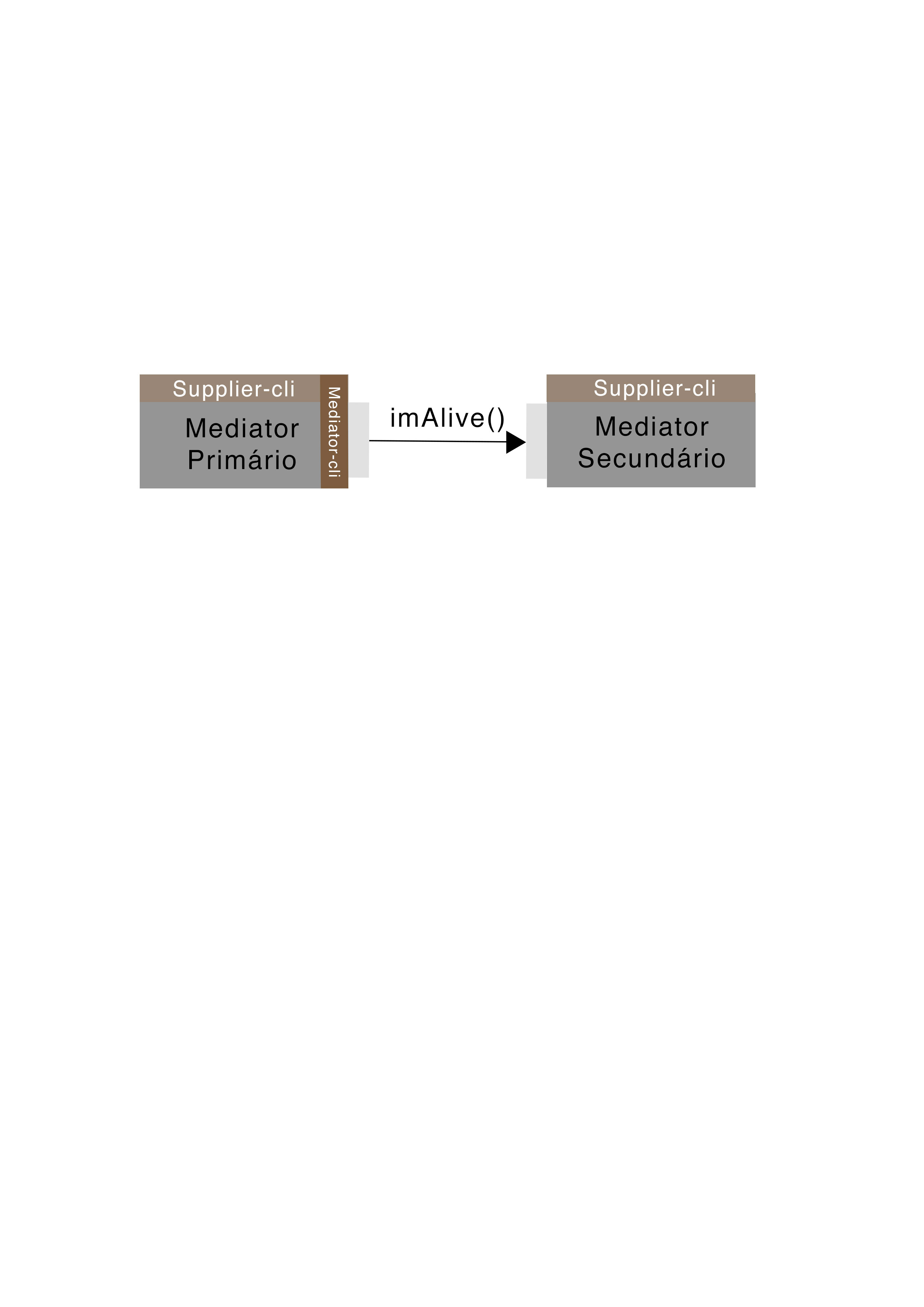
Mediator primário dá update ao Mediator secundário consoante o pedido efectuado pelo cliente. Neste update envia também o identificador do pedido (para garantir semântica de “no máximo uma vez”), através do handler “AtMostOnceHandler”.

**Passo 4:**



Mediator primário dá resposta ao cliente com o retorno do método invocado

**Execução contínua:**



Mediator primário, de x em x segundos, envia um *imAlive* para o Mediator secundário. Quando o secundário não recebe esta notificação, o mesmo regista-se no UDDI como primário, tomando o seu papel.