

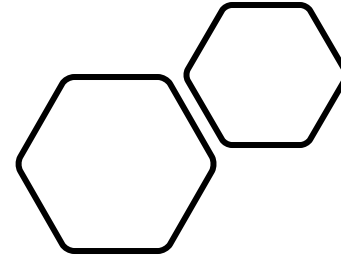
Arquitetura de Micros Serviços (P3)

Prof. Bernardo Copstein

Prof. Júlio Machado



Micros serviços: projeto de dados



Leituras recomendadas:

- Sommerville, Ian. Engineering Software Products – An Introduction to Modern Software Engineering. Pearson, 2019. Capítulo 6.
- Newman, Sam. Building Microservices. O'Reilly Media Inc, 2015.



Projeto de dados de micros serviços

- Você deve isolar os dados em cada serviço do sistema com o mínimo de compartilhamento de dados possível.
- Se o compartilhamento de dados for inevitável, você deve projetar micros serviços para que a maioria dos compartilhamentos seja 'somente leitura', com um número mínimo de serviços responsáveis pelas atualizações de dados.
- Se os serviços forem replicados em seu sistema, você deverá incluir um mecanismo que possa manter as cópias do banco de dados usadas pelos serviços de réplica consistentes.



Gerenciamento de inconsistências

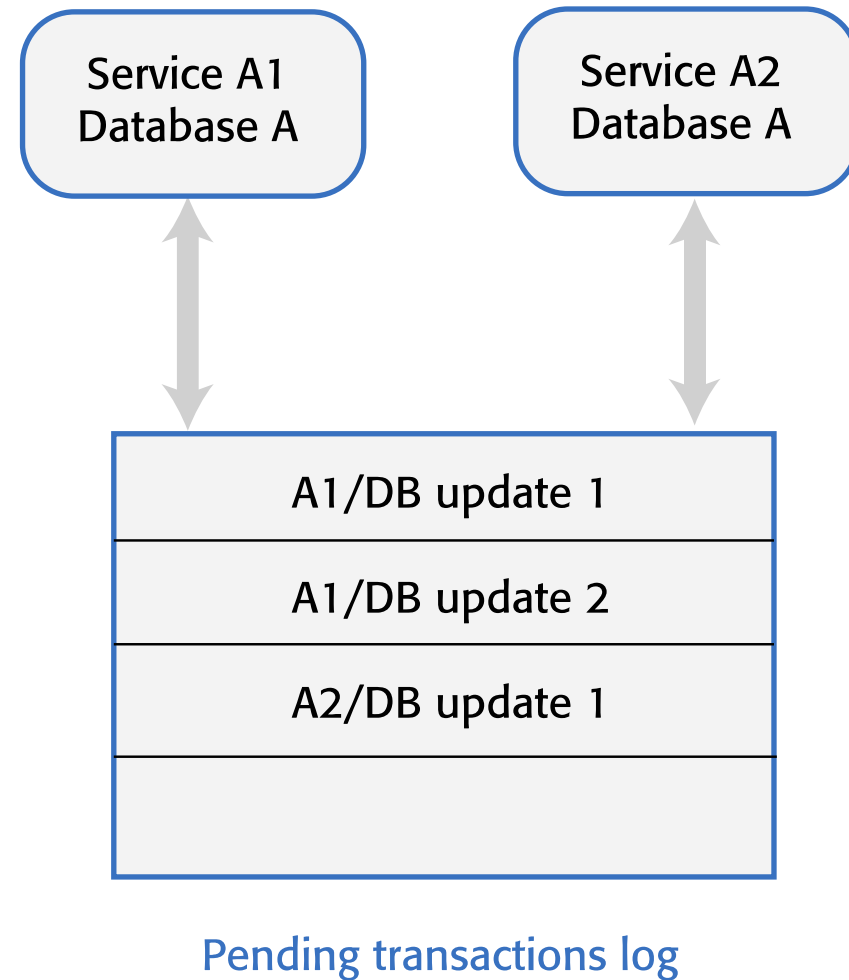
- Uma transação ACID agrupa um conjunto de atualizações de dados em uma única unidade para que todas as atualizações sejam concluídas ou nenhuma delas seja. As transações ACID são “limitantes” em uma arquitetura de micros serviços.
- Os bancos de dados usados por diferentes micros serviços ou réplicas de micros serviços não precisam ser completamente consistentes o tempo todo.
- Inconsistência de dados dependentes
 - As ações ou falhas de um serviço podem fazer com que os dados gerenciados por outro serviço se tornem inconsistentes.
- Inconsistência da réplica
 - Existem várias réplicas do mesmo serviço que estão sendo executadas simultaneamente. Todos eles têm sua própria cópia do banco de dados e cada um atualiza sua própria cópia dos dados do serviço. Você precisa de uma maneira de tornar esses bancos de dados 'eventualmente consistentes' para que todas as réplicas trabalhem nos mesmos dados.



Consistência eventual

- A consistência eventual é uma situação em que o sistema garante que os bancos de dados eventualmente se tornarão consistentes.
- Você pode implementar consistência eventual mantendo um log de transações.
- Quando uma alteração no banco de dados é feita, isso é registrado em um log de 'atualizações pendentes'.
- Outras instâncias de serviço examinam esse log, atualizam seu próprio banco de dados e indicam que fizeram a alteração.

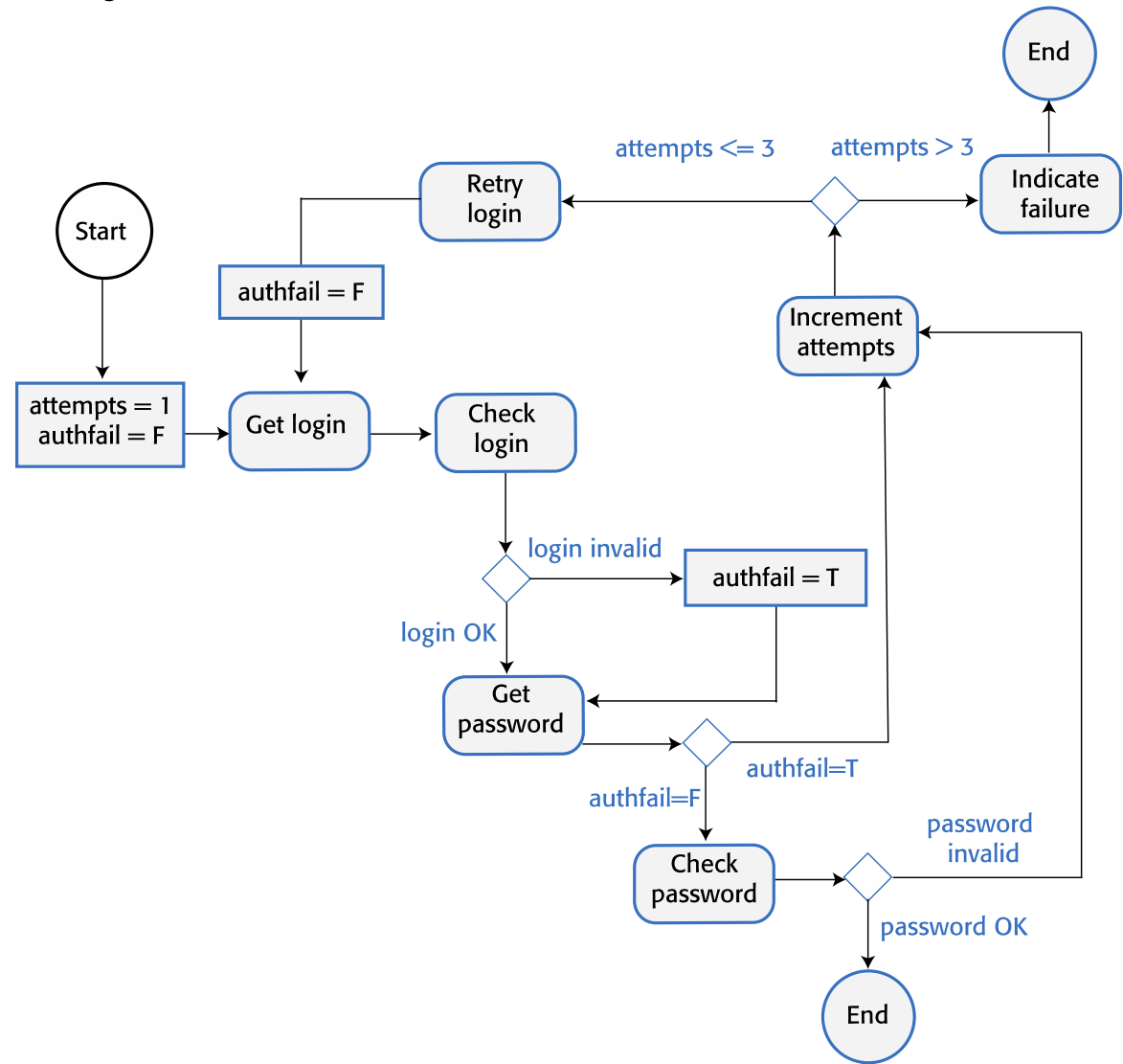
Usando um
log de
transações
pendentes



Coordenação de micros serviços

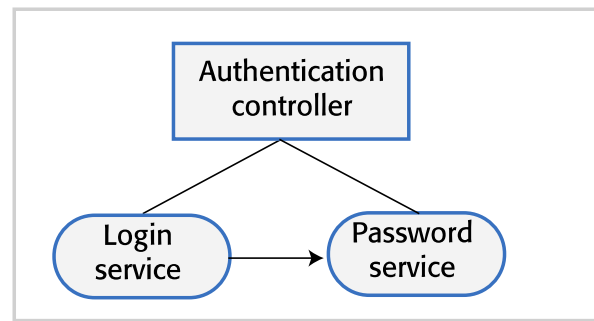
- A maioria das sessões de usuário envolve uma série de interações nas quais as operações devem ser realizadas em uma ordem específica.
- Isso é chamado de fluxo de trabalho.
 - Um fluxo de trabalho de autenticação para autenticação de UID/senha mostra as etapas envolvidas na autenticação de um usuário.
 - Neste exemplo, o usuário tem 3 tentativas de login antes que o sistema indique que o login falhou.

Fluxo de trabalho de autenticação

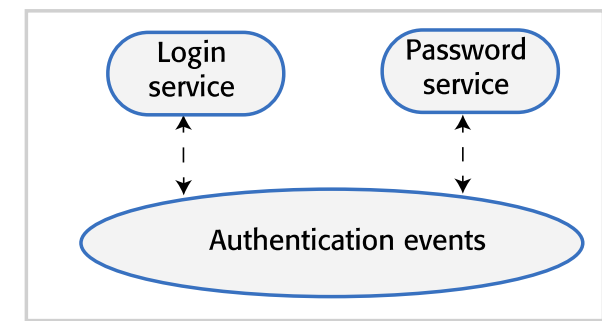


Orquestração e coreografia

Service orchestration



Service choreography



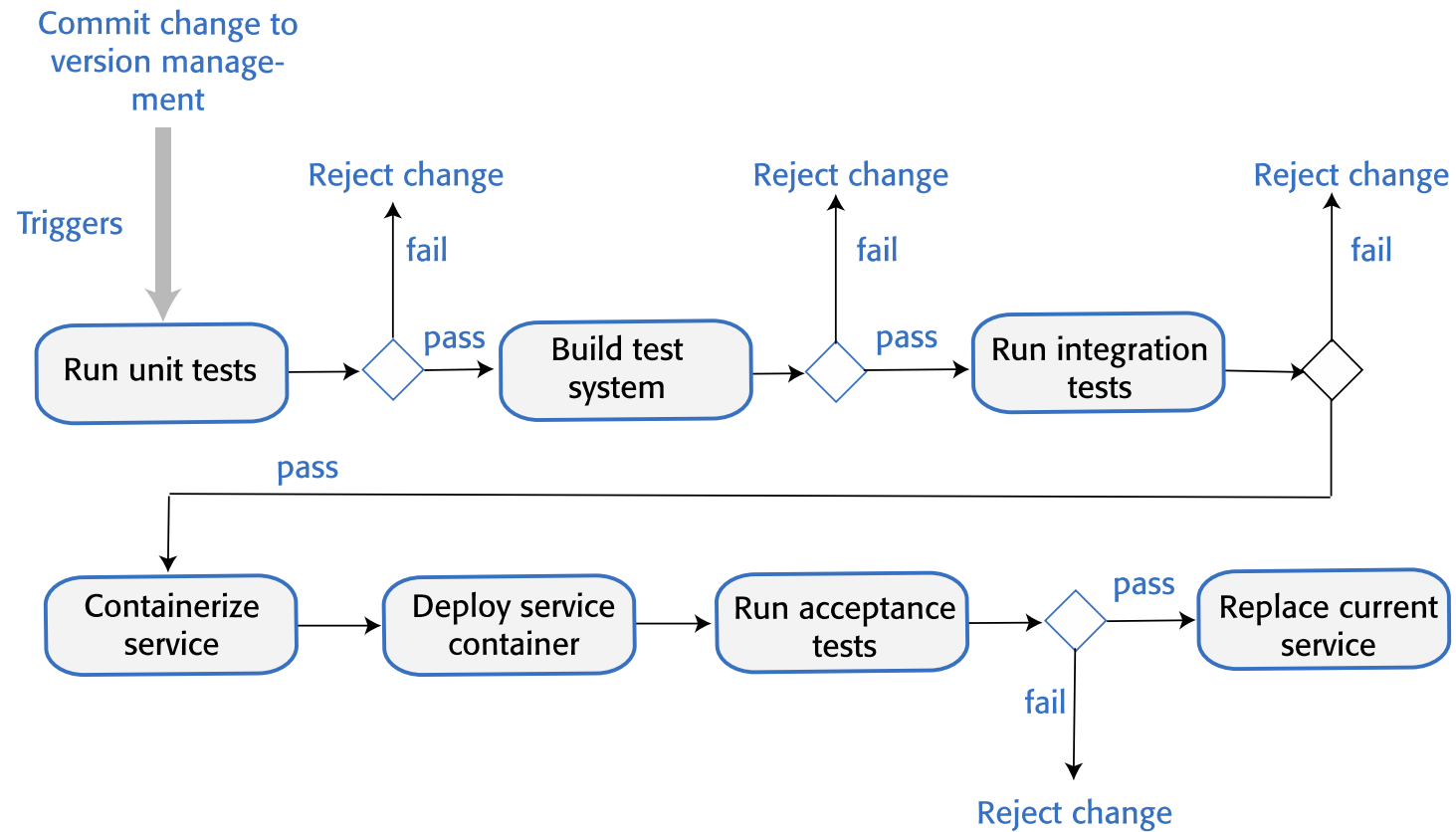
Tipos de falhas

- *Falha de serviço interno*
Essas são condições detectadas pelo serviço e podem ser relatadas ao cliente de serviço em uma mensagem de erro. Um exemplo desse tipo de falha é um serviço que usa uma URL como entrada e descobre que esse é um link inválido.
- *Falha de serviço externo*
Essas falhas têm uma causa externa, o que afeta a disponibilidade de um serviço. A falha pode fazer com que o serviço não responda e ações devem ser tomadas para reiniciar o serviço.
- *Falha de performance*
O desempenho do serviço degrada-se a um nível inaceitável. Isso pode ser devido a uma carga pesada ou um problema interno com o serviço. O monitoramento de serviços externos pode ser usado para detectar falhas de desempenho e serviços sem resposta.

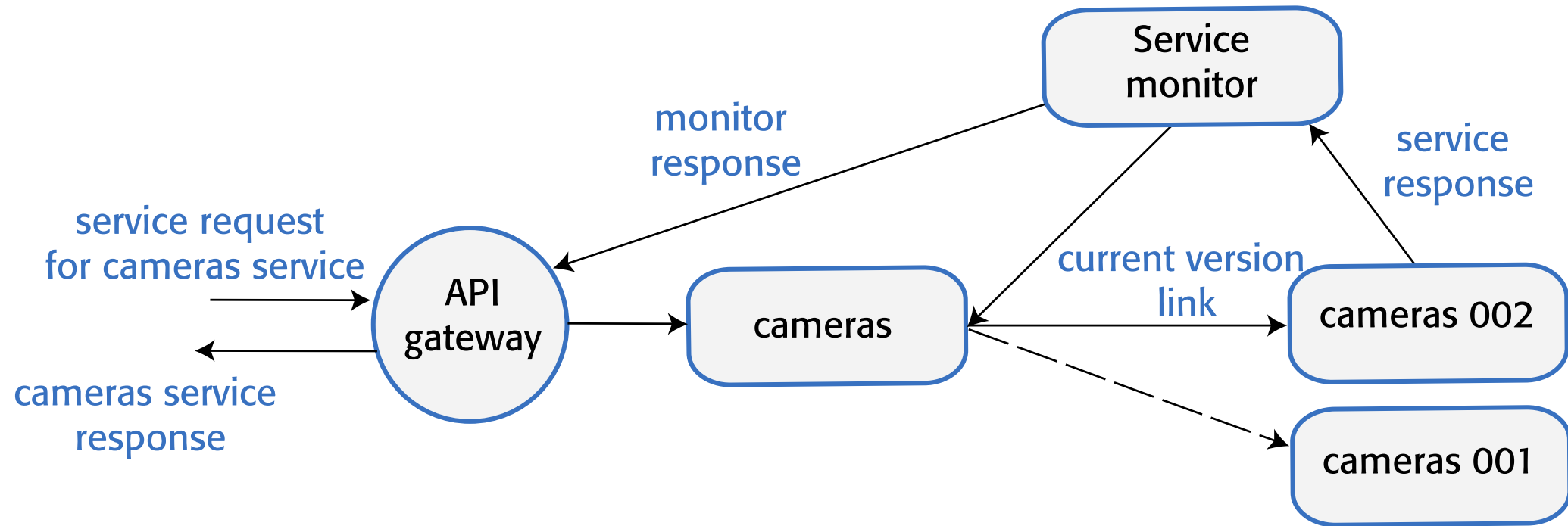
Automação de implantação

- A implantação contínua depende da automação para que, assim que uma mudança seja confirmada, uma série de atividades automatizadas seja acionada para testar o software.
- Se o software 'passar' nesses testes, ele entra em outro pipeline de automação que empacota e implanta o software.
- A implantação de uma nova versão de serviço começa com o programador confirmando as alterações de código em um sistema de gerenciamento de código como o Git.
- Isso aciona um conjunto de testes automatizados que são executados usando o serviço modificado. Se todos os testes de serviço forem executados com êxito, será criada uma nova versão do sistema que incorpora o serviço alterado.
- Outro conjunto de testes automatizados do sistema é então executado. Se eles forem executados com êxito, o serviço estará pronto para implantação.

Um pipeline de implantação contínua



Serviços com versão






Exercícios

1) Que tipo de inconsistência de dados pode ocorrer na seguinte situação: um sistema de compra e venda de dólares é composto por dois micros serviços. Um faz atualizações constantes no custo da moeda. O outro é responsável pela manutenção do estoque de moeda na casa e um terceiro pelas operações de compra e venda.

2) Seria possível replicar o micro serviço que mantém o estoque de moeda na casa? Em caso positivo, como tratar?



Pontos-chave

1

- Um micro serviço é um componente de software independente e autocontido que é executado em seu próprio processo e se comunica com outros microserviços usando protocolos leves.
- Os micro serviços em um sistema podem ser implementados usando diferentes linguagens de programação e tecnologias de banco de dados.
- Os micro serviços têm uma única responsabilidade e devem ser projetados para que possam ser facilmente alterados sem a necessidade de alterar outros microserviços no sistema.
- A arquitetura de micro serviços é um estilo de arquitetura no qual o sistema é construído a partir de micro serviços de comunicação. É adequado para sistemas baseados em nuvem, onde cada micro serviço pode ser executado em seu próprio contêiner.
- As duas responsabilidades mais importantes dos arquitetos de um sistema de micro serviços são decidir como estruturar o sistema em micro serviços e decidir como os micro serviços devem se comunicar e ser coordenados.

Pontos-chave

2

- As decisões de comunicação e coordenação incluem decidir sobre protocolos de comunicação de micros serviços, compartilhamento de dados, se os serviços devem ser coordenados centralmente e gerenciamento de falhas.
- O estilo arquitetural RESTful é amplamente utilizado em sistemas baseados em micros serviços. Os serviços são projetados para que os verbos HTTP GET, POST, PUT e DELETE sejam mapeados para as operações do serviço.
- O estilo RESTful é baseado em recursos digitais que, em uma arquitetura de micros serviços, podem ser representados usando XML ou, mais comumente, JSON.
- A implantação contínua é um processo em que novas versões de um serviço são colocadas em produção assim que uma alteração no serviço é feita. É um processo totalmente automatizado que se baseia em testes automatizados para verificar se a nova versão é de 'qualidade de produção'.
- Se a implantação contínua for usada, talvez seja necessário manter várias versões dos serviços implantados para poder alternar para uma versão mais antiga se forem descobertos problemas em um serviço recém-implantado.

