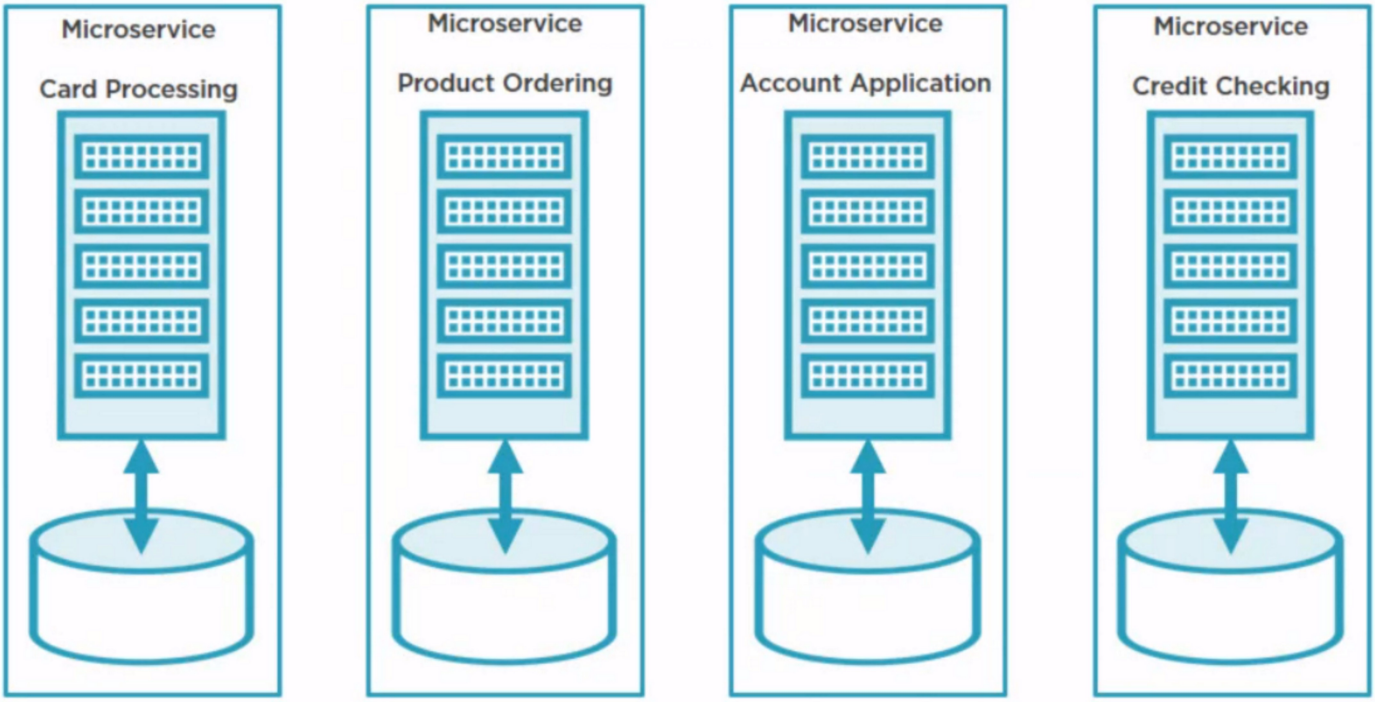
Arquitetura de microsserviço

sexta-feira, 17 de setembro de 2021 10:10

Microsserviços são serviços pequenos e autônomos que colaboram todos juntos.



Os microsserviços usam a mesma abordagem para serviços independentes. Você concentra seus limites de serviço nos limites de negócios, tornando óbvio onde o código reside para uma determinada funcionalidade. Ao manter este serviço focado em um limite explícito, evitamos a tentação de que ele se torne muito grande, evitamos as dificuldades associadas que isso pode apresentar. Por exemplo, acima temos quatro microsserviços diferentes:

- Processamento de cartão: para processamento e gerenciamento de pagamentos com cartão;
- Pedido de produto: em que um cliente pode solicitar os produtos da empresa;
- Conta: em que um cliente solicita uma conta no sistema e;
- Verificação de crédito: onde verificações de crédito podem ser feitas contra clientes que desejam celebrar um contrato de financiamento.

E não estamos falando apenas sobre o serviço real ser independente. Também estamos estendendo isso para o banco de dados. Com isso, quero dizer que, em um mundo de microsserviços, cada serviço tem seu próprio banco de dados, não apenas tabelas separadas, mas sua própria instância de banco de dados.

Então, queremos que esses serviços sejam pequenos e focados, mas quão pequeno é pequeno? Uma maneira de pensar sobre isso é seguir o princípio da responsabilidade única e que eles fazem apenas uma coisa e a fazem bem. Por exemplo, se estamos interessados em receber pagamentos e processar pedidos de compra, portanto, vamos compor nosso sistema em dois pequenos serviços, um serviço de pagamento com cartão de débito que possui um único método para processar um pagamento e um serviço de pedido de compra para enviar novos pedidos de compra ao departamento de contabilidade. Mas e quanto aos reembolsos de cartão? Bem, se fôssemos implementar um sistema de reembolso, o

implementaríamos como outro serviço separado do serviço que recebe pagamentos. Esse é o tipo de separação de funcionalidade de que estamos falando. No passado, ao lidar com serviços, especialmente serviços da web baseados em XML e SOAP, talvez tenhamos que colocar todos os nossos métodos de pagamento com cartão em um serviço. Isso geralmente é bom, mas correu o risco de criar grandes serviços monolíticos e se tornou difícil de manter e, como todos sabemos no mundo real, quando a pressão aumenta com os prazos, um refator bom e limpo é a última coisa que acontecerá . Indo direto para serviços muito pequenos e focados, reduzimos o risco de criar grandes monólitos. À medida que seus serviços ficam menores, os benefícios em torno da interdependência aumentam, mas também aumenta parte da complexidade que surge por ter cada vez mais peças móveis. À medida que você melhora em lidar com essa complexidade, pode se esforçar para serviços cada vez menores.