



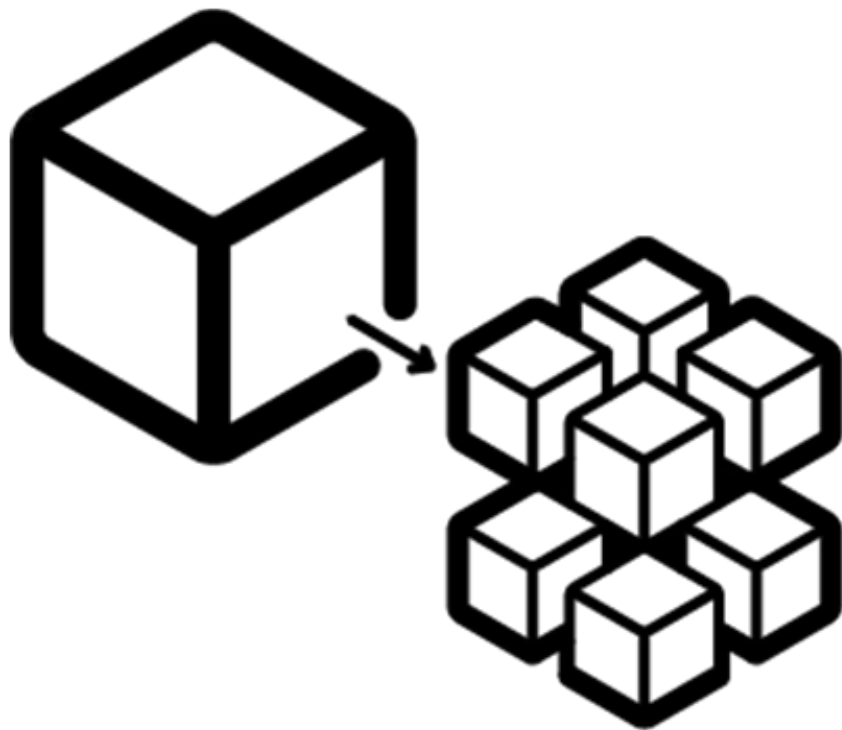
As vantagens de usar DDD para modelar seu Microservice



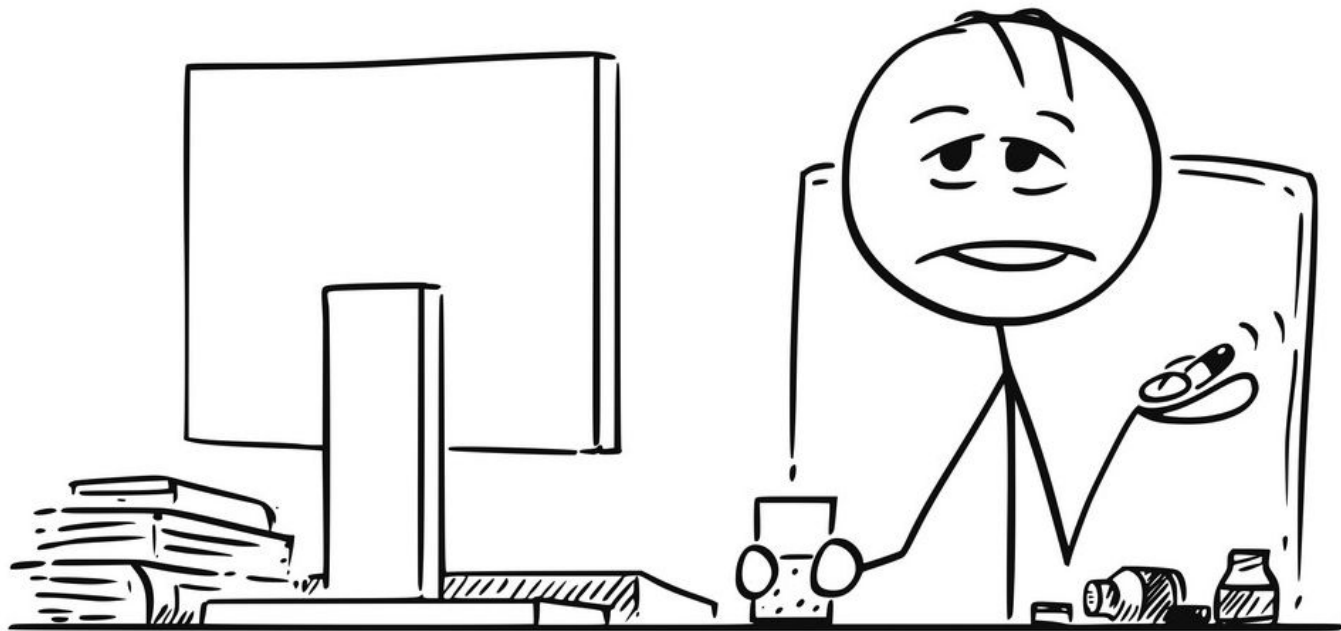
Andrews dos Reis
Software Developer



<https://www.linkedin.com/in/andrewsdosreis/>
<https://twitter.com/andrewsreis>
<https://github.com/andrewsdosreis>





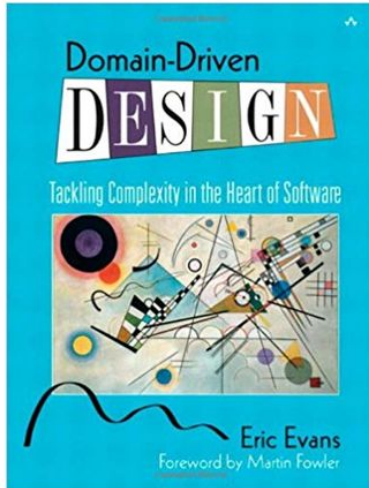




Qual o tamanho certo?

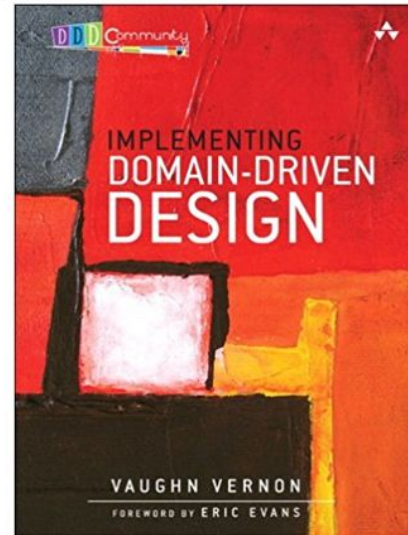


Domain Driven Design



“Domain-Driven Design: Tackling Complexity in the Heart of Software” by Eric Evans

“Implementing Domain-Driven Design” by Vaughn Vernon





Domain

- Core
- Support
- Generic



Domain Model





“DDD é primariamente sobre modelar uma linguagem ubíqua em contexto delimitado” – Vaughn Vernon



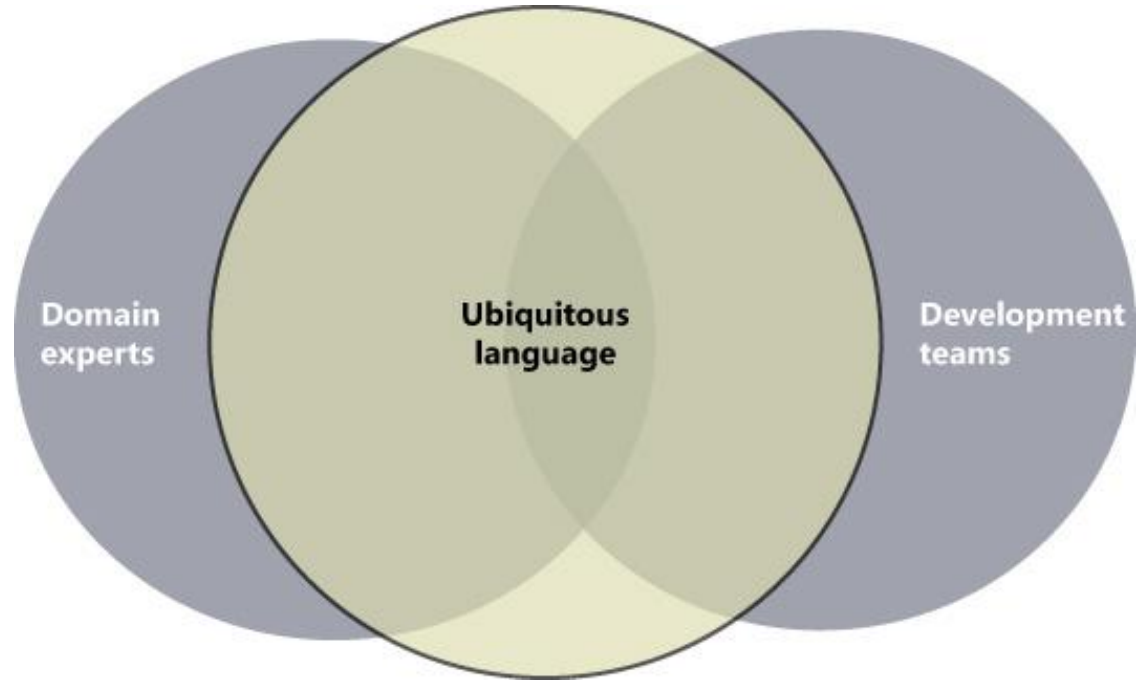
Context Matters

Ubiquitous Language

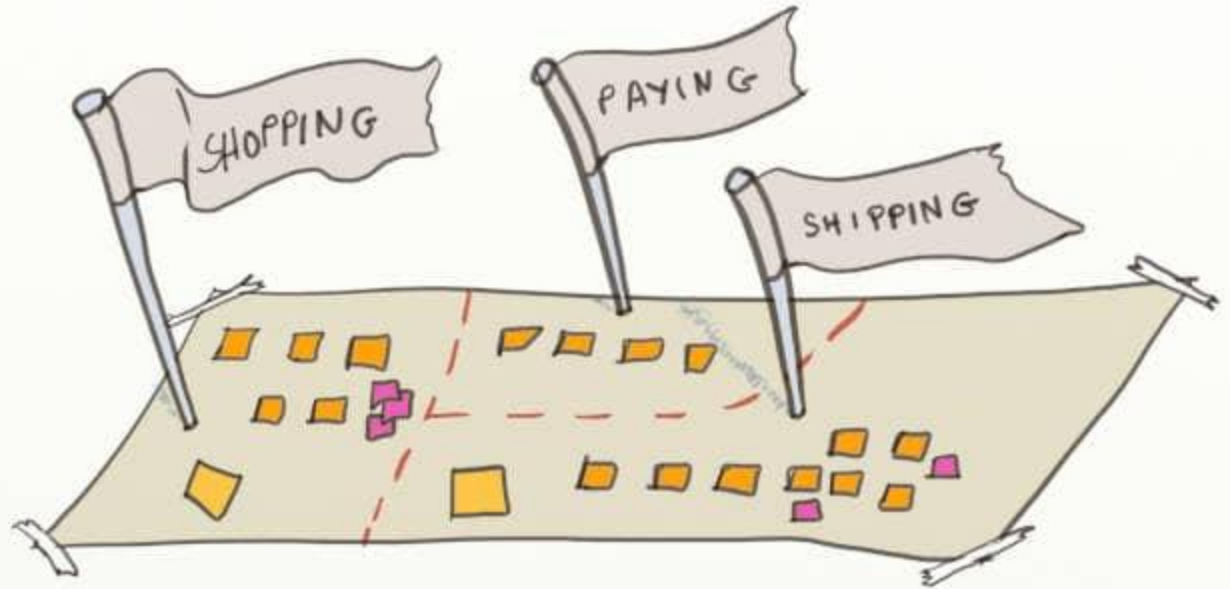




Ubiquitous Language



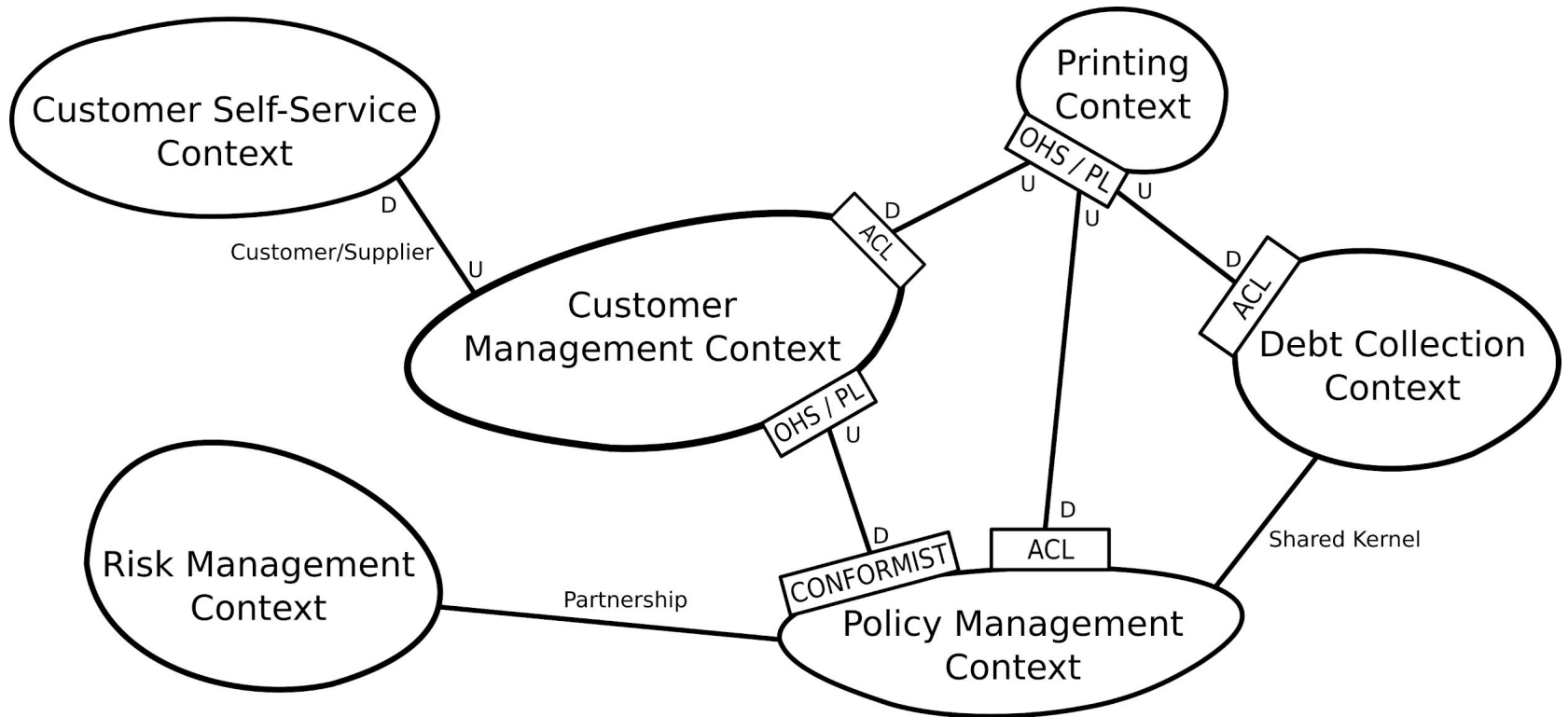
Bounded Context

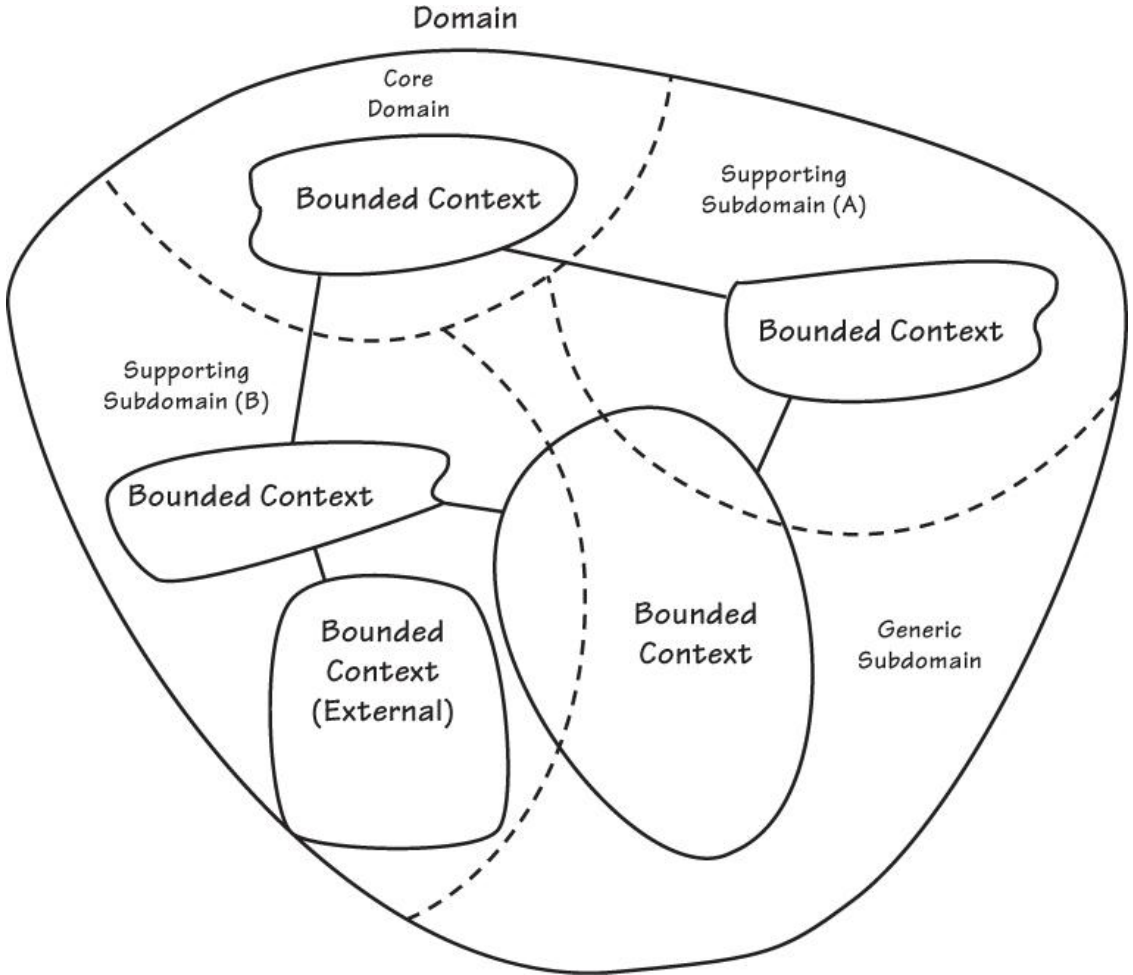


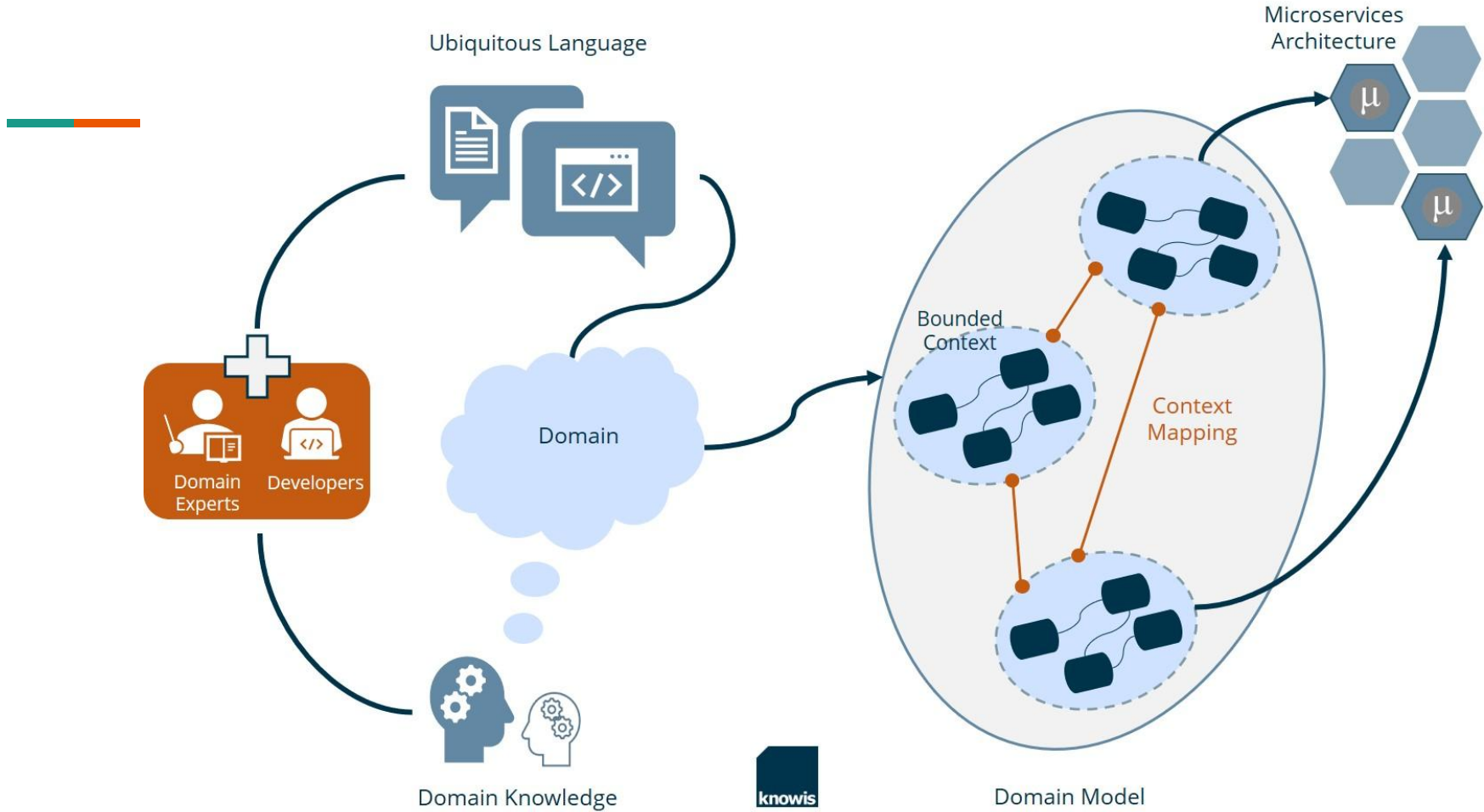


Context Map

- Shared Kernel
- Partnership
- Conformist
- Customer/Supplier
- Anti-Corruption Layer
- Open Host Service









“Se os limites dos nossos serviços alinham-se aos contextos delimitados do nosso domínio, nós começamos bem para garantir que nossos microserviços estão com baixo acoplamento e alta coesão” — Sam Newman



References

[Domain-Driven Design: Tackling Complexity in the Heart of Software](#)

[Implementing Domain-Driven Design](#)

[Domain Driven Design Quickly - InfoQ](#)

[Eric Evans: Domain Driven Design and Microservices](#)

[Martin Fowlers blog: DDD](#)

[Domain-Driven Design: uma visão geral - Eriksen Costa](#)

[Lambda 3 - Graziella Bonizi](#)

