

# Microservices mit .Net Framework und Core ohne Docker

Momentan liest man überall von Microservices und welche Vorteile diese Architektur mit sich bringt. Zwar wird immer wieder aufgeführt, dass Microservices Polyglot (Eine Architektur mit unterschiedlichen Programmiersprachen) unterstützen können. Tatsächlich findet man aber nur Anleitungen (In der Microsoft Welt) Zu .Net Core, in Verbindung mit Docker und Azure.

Mein Ziel ist es eine Architektur ohne Docker (Da auf VMs nicht unterstützt wird) und eine Verbindung aus Core und .Net Framework herzustellen. Aus einer Reihe von nachhaltigen Frameworks soll die Entwicklung in CD (Continues Delivery) gestaltet werden. TFS und GitLab scheiden wegen ihren Lizenzmodell aus.

- CodeRepository: Git -> <https://git-scm.com/download/win>
- Build Server: Jenkins -> <https://jenkins.io/download/>
  - <https://www.guru99.com/jenkin-continuous-integration.html>
- Api Gateway (Kommunikation ClientApi zu Microservices) Ocelot : <https://github.com/ThreeMammals/Ocelot>
- Kommunikationsprotokoll zwischen Microservices: gRPC (Da schneller als Http)
- Sicherheit unter Microservices: JWT Token
- Authentifizierung Microservice mit OAuth 2.0 und OWIN Middleware: <https://oauth.net/code/dotnet/>
- Repository: Implementierung mit Dapper: <https://github.com/StackExchange/Dapper>
- Repository Cache: <https://github.com/MichaCo/CacheManager>
- Datenbank Code Migration SSDT: SQL Server Data Tools

- [https://www.youtube.com/watch?v=6ass\\_PYECmM&t](https://www.youtube.com/watch?v=6ass_PYECmM&t)
- <https://arapaima.uk/post/2017-04-04-jenkins-windows-git-ssdt-profit/>
- (Optional) Search Engine:  
<https://www.elastic.co/guide/en/elasticsearch/client/net-api/current/introduction.html>
  - <https://www.red-gate.com/simple-talk/dotnet/net-development/how-to-build-a-search-page-with-elasticsearch-and-net/>