

# Entwickler und Architekten

## Strategien 2022

Miro <https://osturm.me/3B5WQ1G>

Passwort nicht mehr relevant

Oliver Sturm - @olivers[@fosstodon.org] - oliver@oliversturm.com


# Für Programmierer und Software-Architekten: Strategien 2022


Oliver Sturm  
@olivers[@fosstodon.org]  
DevExpress


Willkommen!


Am Morgen des Workshops werden wir gemeinsam eine Themenliste für den Tag aufstellen und priorisieren. Unten findet ihr einige Vorschläge, die auch als Anregungen verstanden werden dürfen. Wenn ihr mögt, fügt einfach eure eigenen Ideen hinzu (unten rechts - damit wir sie auch finden!) - je mehr, desto besser! Falls es Klärungsbedarf gibt, reden wir im Workshop darüber, und dann stimmen wir zu allen Themen ab.


Eure Vorschläge - bitte bei Bedarf Platz machen - gern so ausführlich wie ihr wollt!


COVID-19 - was ändert sich?  



Microservices, ja oder nein? Und wie?  



Cloud - muss das sein? Welche? Worauf kommt's an?  



CQRS? Event Sourcing? Eventual Consistency? GraphQL? Moderne Datenzugriffspatterns.  



Dapr oder Eigenbau?  



Serverless - wird da alles einfacher? Oder schwieriger?  



Container und VMs - Docker, Kubernetes, Vagrant, Terraform, ...  



Blazor? Ist das die Zukunft? Und darf's lieber SignalR oder WASM sein?  



Welcher Weg führt am schnellsten zur mobilen Plattform? Und wo steht "Mobile" im Vergleich zu "Desktop" oder "Web"?  



gRPC statt JSON-Diensten?  


Best Practices nach Oli Sturm  


TypeScript oder JavaScript? Oder?  


SPA, CSR, SSR oder was?  


bisschen lachine oder so?  


React, Vue, Angular, Svelte, usw...  


Architektur Analysen und Strategie bei "User-Centered-Design" Prozessen - Wie kann das untergebracht werden im agilen Kontext  


Architektur-Strategie für UI-Technologie wechsel bei langlebigen Softwareprojekten  


Architekturdokumentation  
Wie viel braucht es, und wo sollte man auf Code verweisen?  
Was lässt sich generieren, und was sollte man lieber schreiben?  


Architektur + Dateisystem / Code Organisation  
-> Wie lege ich, im Bezug auf Wartbarkeit, Code ab (Ordnerstruktur)?  
-> Wie gehe ich mit geteiltem Code um?  
- "Shared Library" oder direkte Abhängigkeiten unter spezifischen Modulen?  
- Vorteile / Nachteile  


Wir bauen eine neue Software welche eine alte Desktop Anwendung ablösen soll.  
Geplant: Angular(devextreme) + ASP.net WebApi  
- Wie erkenne ich Anforderungen welche Auswirkungen auf Architektur haben frühzeitig?  
Welche Strategien, Ansätze gibt es hierzu?  


Was ist die Aufgabe eines Software Architekten?  


Offlinefähige Webanwendung  


Wie finde ich ein geeignetes Architekturmuster für eine neue Angular/Vue/React-Anwendung (z.B. MVC/MVVM/MVP/?).

Redundanter Code lässt sich bei Microservices oft nicht vermeiden. Wie stellt man am besten sicher, dass bei Änderungen alle Services gleichermaßen angepasst werden?  


Wie können große Architekturänderungen (z.B. Migration auf Microservices) in verteilten Teams geplant und gesteuert werden?  


Wegwerf-Software vs. Nachhaltige Technologien:  
Unsere Software muss länger als 3 (eher 30) Jahre halten (Stichwort: .Net Core). Was tun und was lassen?  


Wir setzen aktuell Prism-Library 6.3 im WPF-Projekt ein. Ich suche Ähnliches für künftige ASP.NET-Projekte (Blazor Server). Ist es sinnvoll sich an ein solches Application-Framework zu binden? Gibt es bessere Alternativen?  


Design Pattern und SOLID  


Web oder Desktop  


gRPC statt JSON-Diensten?  


Allgemeine Tipps zur Entscheidungsfindung  
- wie kann man diese strukturiert und objektiv treffen?  
- Welchen Aufwand sollte man in die Entscheidungsfindung stecken?  
- wie vermeidet man bei der Suche nach Optimum nie fertig zu werden?  


Webapplikationen/Cloud/Datebanken in kritischer Netzwerkinfrastruktur (Klinik)  
- worauf muss man sich einstellen?  
- welche Grenzen gibt es?  
- was sollte man lieber lassen?  


Darf ich als Architekt auf "alt-bewährte" Technologien setzen (wie Web-Standards ohne Frameworks), weil sie funktionieren und Erfahrung vorhanden ist - oder muss ich mich fortlaufend mit den neusten Techniken beschäftigen und diese bewerten?  


Web-API:  
- Ist eine Repository-Schicht mit EF-Core noch notwendig/sinnvoll?  
- Ersetzt EF-Core die Repo-Schicht?  
- Wie baut man Integrationstests sinnvoll ohne Repo-Schicht auf?  


Ist Angular ein totes Pferd? Wann macht es Sinn Angular/ React/ Vue zu nehmen? Unser alte Anwendung wurde in Angular JS entwickelt. Jetzt wollen wir upgraden/neu bauen.  


Google Protocol Buffers + Alternativen  


gRPC statt JSON-Diensten?  


Wie realisiert man Suchen in Microservice-Landschaften, die über die Grenzen eines Microservices hinaus gehen?  
- Wie realisiert man die Authentifizierung/Autorisierung bei Event-Basierten Microservice-Systeme (No\_Trust Umgebung). Werden in den Events auch Tokens (z. B. JWT's) mitgegeben?  


Neue Design Web UI  
=> Wie wählt man am besten die Technologie. Client-Side- (Angular/React/Vue/Blazor) vs Server-Side-Rendering (Razor Pages)  


Im Internet findet man immer gleiche Beispiele für Microservices. Kannst du einmal ein grösseres komplexeres Beispiel vorstellen.

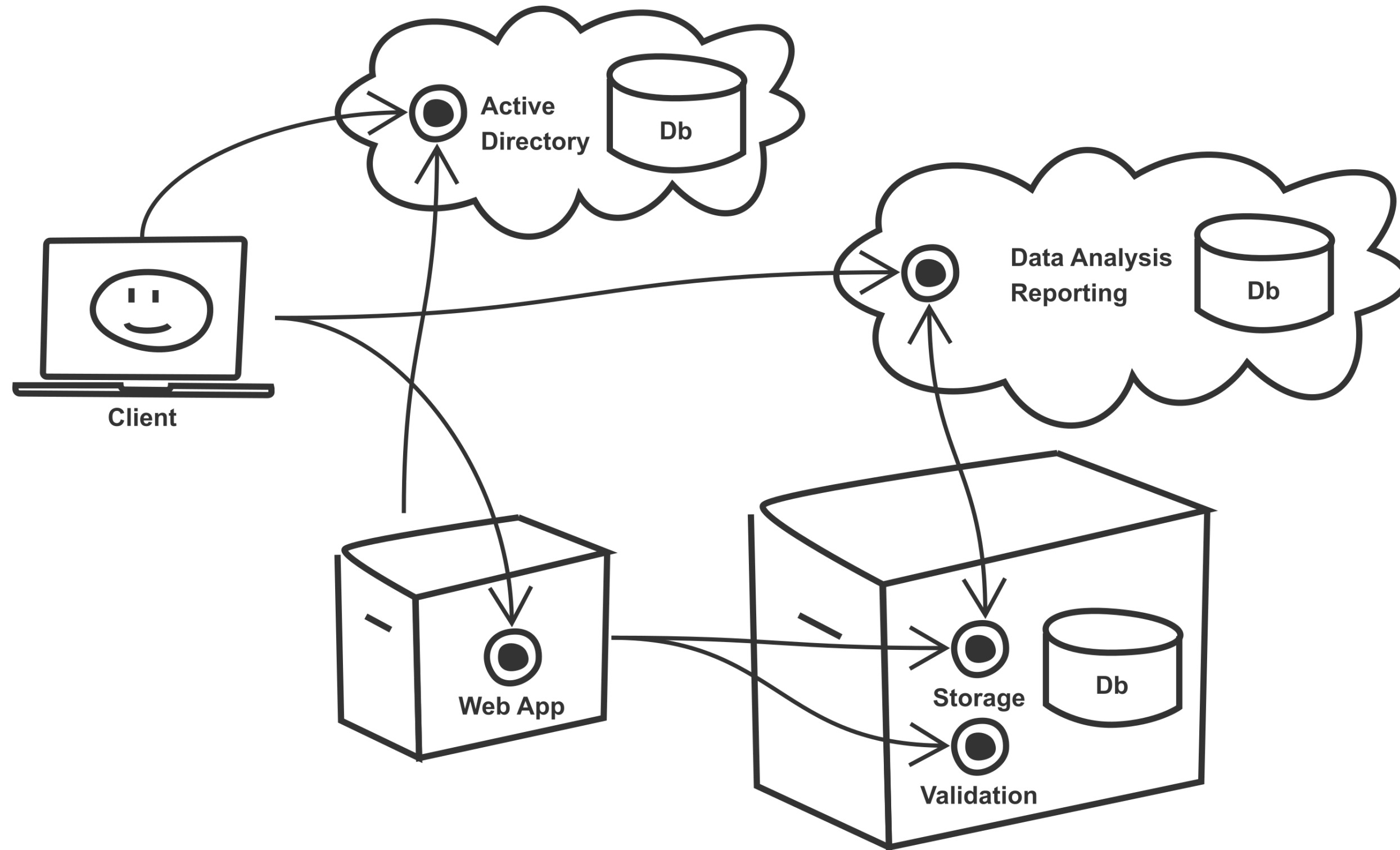
WebAPI (Weniger ist Mehr?):  
Muss es immer objektorientiert sein um Daten aus einer DB zu lesen, reicht nicht auch mal Python, Pandas, AdoDbApi FastAPI. In einem schönen Docker-Container versteckt. Dann ist ML nicht mehr weit weg.  


GitHub oder GitLab oder.... welche Plattform ist die richtige für unser Unternehmen?

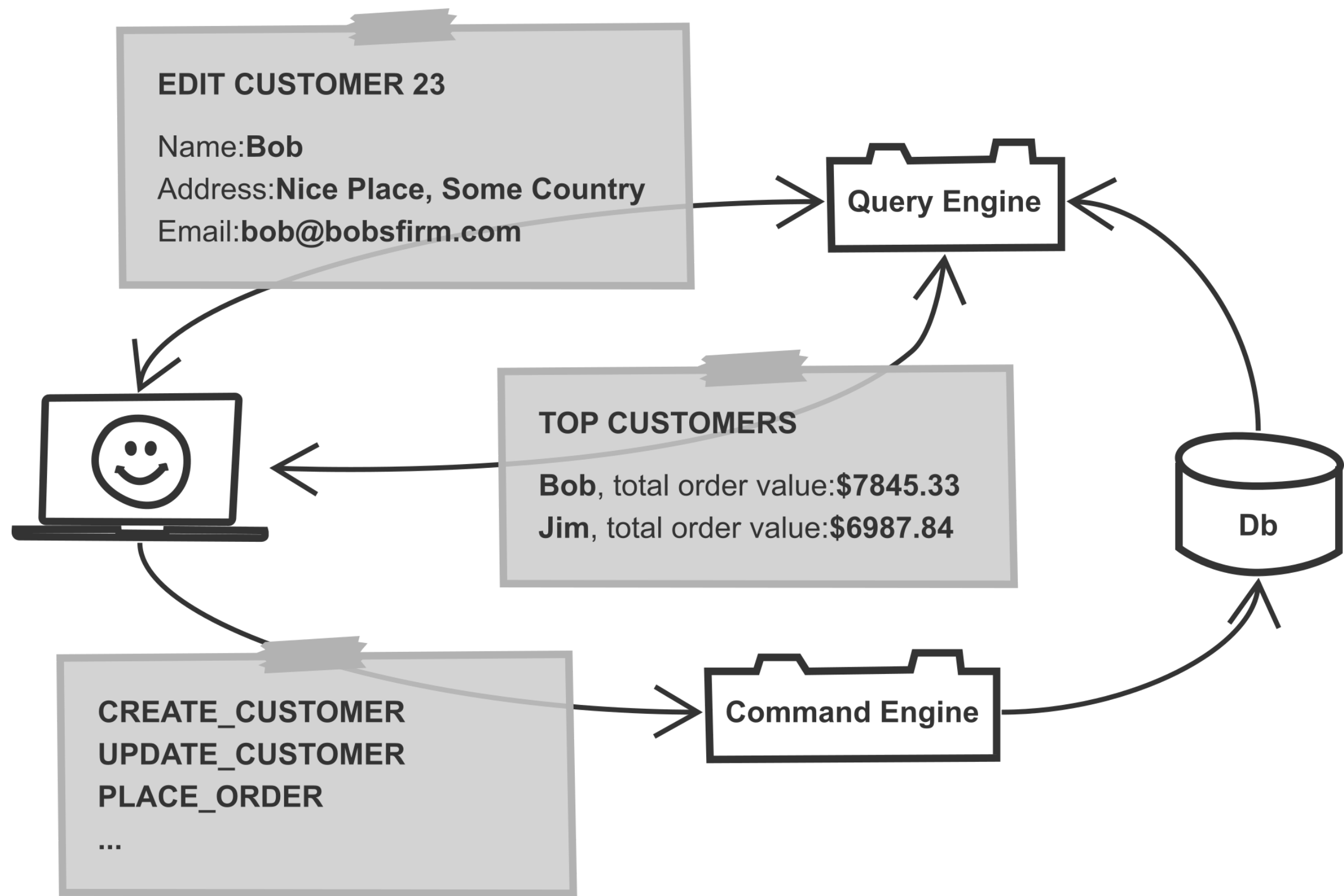
Wir migrieren große Teile unserer Produktionssoftware von C nach .NET. Welche Anfängerfehler sollten wir vermeiden?

Wie kann man mit microservices mit CQRS pattern transactional-DB (MongoDB) arbeiten?

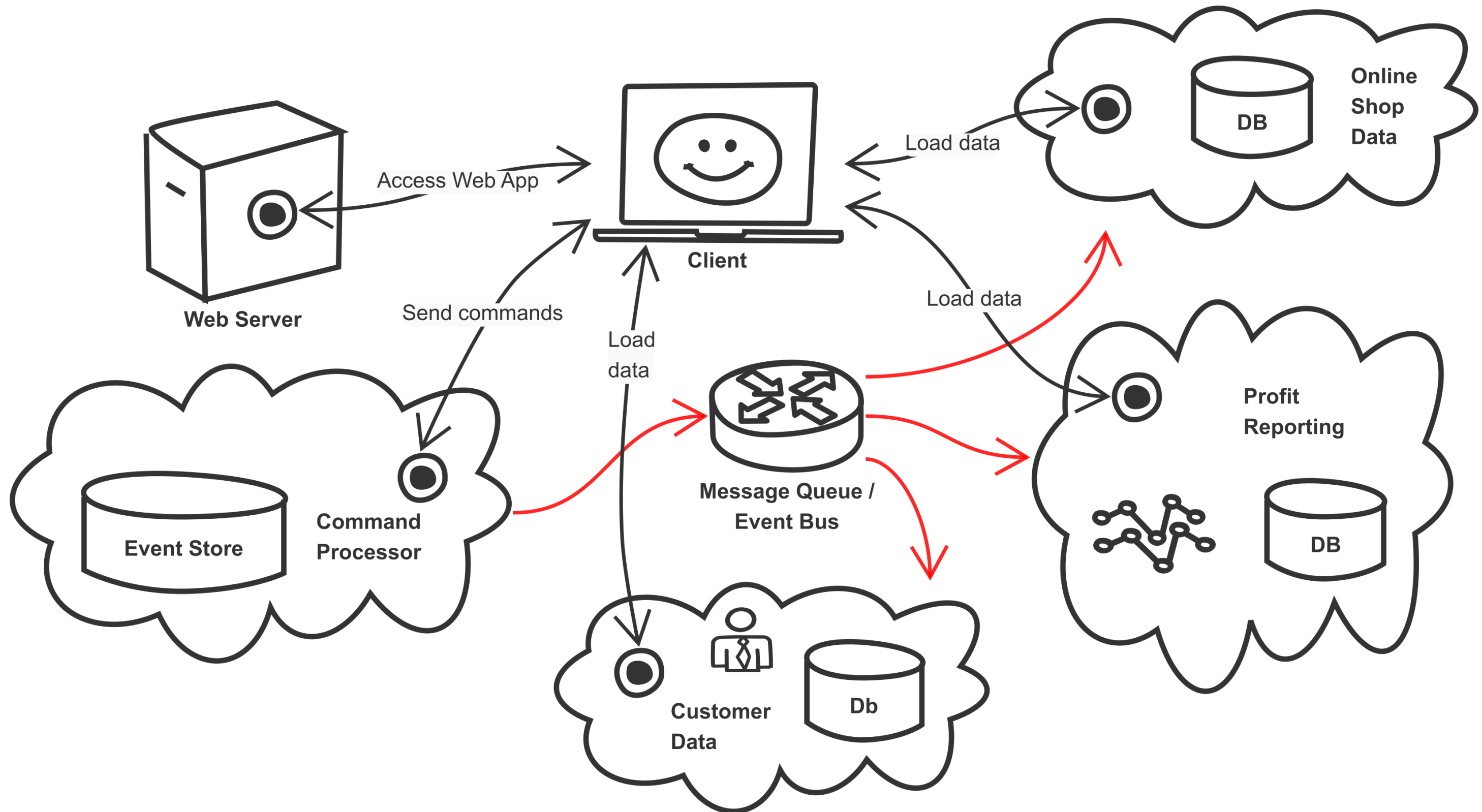
# Dienststruktur - Micro oder Größer



# CQRS



# CQRS mit Event Sourcing



# gRPC vs JSON/REST Services

## Use gRPC

You benefit significantly from HTTP/2, perhaps from full-duplex streaming

## Prefer gRPC

You work in a statically typed programming language

Your messages contain mainly typable data

You exchange a large number of small messages

## Prefer JSON/REST

You work in a dynamic programming language

You transfer mostly dynamic data

## Use JSON/REST

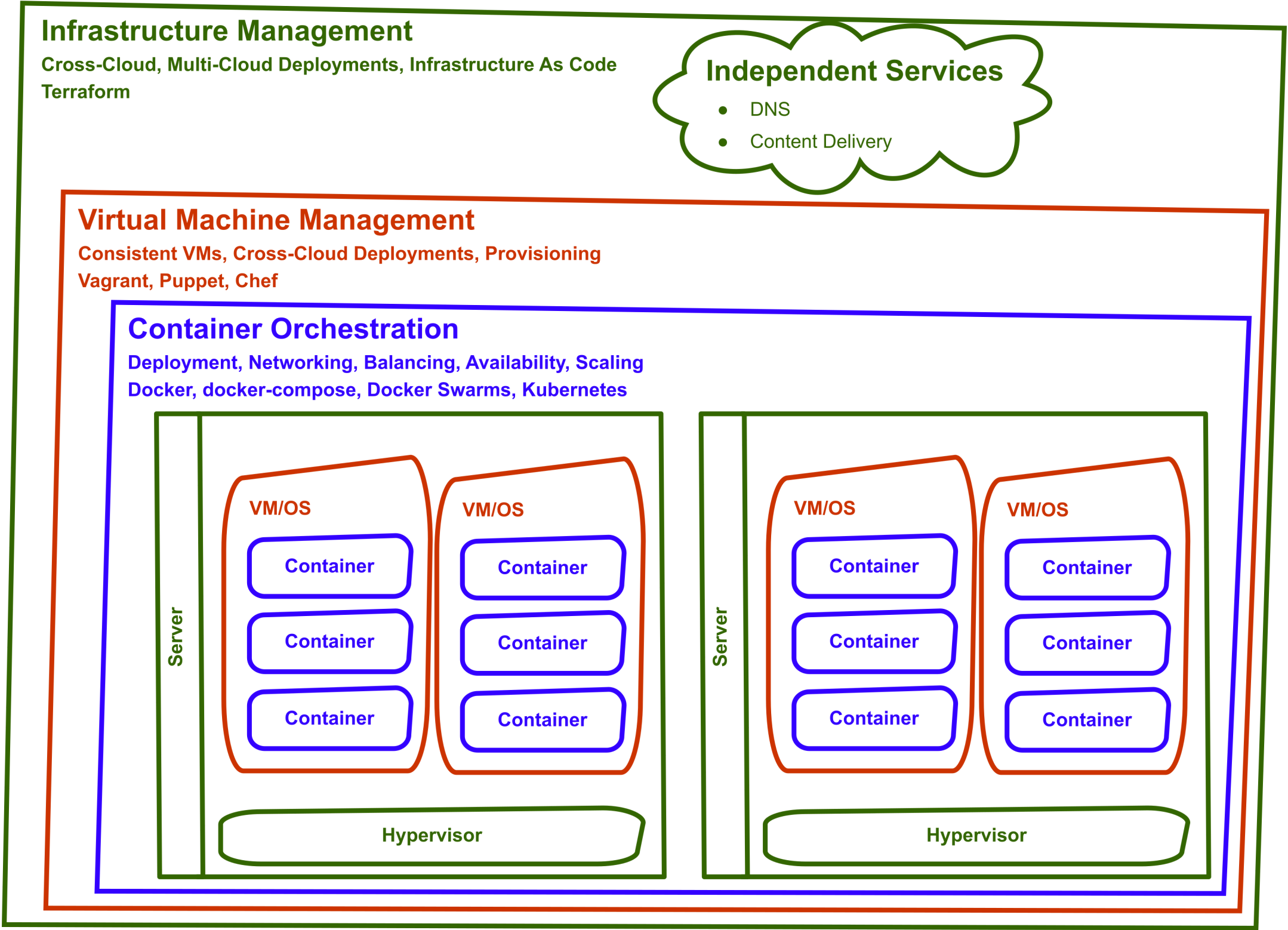
You need a really open call interface, e.g. callable from third parties, command line tools etc.

You strongly prefer human-readable message content

Pro gRPC

Pro JSON/REST

# Container und VMs



# Thank you!

Download:

<https://osturm.me/devarch-basta2022>

Please feel free to contact me about the content anytime.

**Oliver Sturm - @olivers[@fosstodon.org] - oliver@oliversturm.com**