Approach 1

Do authN/Z globally

API Gateway	AuthN	AuthZ
Microservice		

pros

- makes developer happy :)
- less implementation errors
- less risk of forgetting to handle at all
- centrally defined and handled
- smaller micro services
- less repetition in the code in the micro services

cons

- service can not have fine grained object permissions
- all or nothing authorization
- global auth bottleneck

Approach 2

Do authentication globally, and authorization in every microservice

API Gateway	AuthN	
Microservice		AuthZ

pros

- global authentication is easier to manage/control
- fine grained object permissions are possible

cons

- slightly more code in the micro services
- needs some effort to have an overview what you can do with which permission

Approach 3

Do authentication and authorization in every microservice

API Gateway		
Microservice	AuthN	AuthZ

pros

- fine grained object permissions are possible
- different user authentication mechanisms are possible for different microservices

cons

- error prone
- many repetitions
- bigger micro services
- needs some effort to have an overview what you can do with which permission
- no happy developer :-(

Approach 4

Do authN at microservice, auth Z globally

API Gateway		AuthZ
Microservice	AuthN	

pros

- is listed only for completeness.

cons

- it does not make sense
- worst of both worlds
- no fine grained object permissions
- error prone
- tedious repetitive authentication for consumer

links

- https://blog.andyet.com/2015/05/12/micro-services-user-info-and-auth
- http://nordicapis.com/how-to-control-user-identity-within-microservices/
- https://www.appsflyer.com/blog/how-we-solved-authentication-and-authorization-in-our-microservices-architecture/
- http://microservices.io/patterns/apigateway.html