*API Gateway*

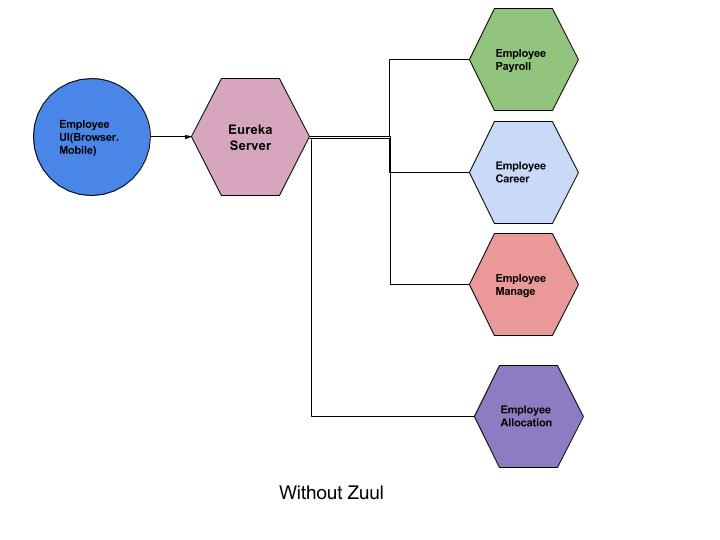
*Zuul Proxy*

O principal objetivo da implementação de microserviços é criar serviços que possam ser dimensionados e implementados de forma independente. Portanto, em um domínio de negócios complexo, a existência de 50 a 100 microserviços são muito comuns. Imaginando um sistema desse tipo, temos que implementar uma interface do usuário que chamará vários serviços para buscar e mostrar as informações nessa interface.

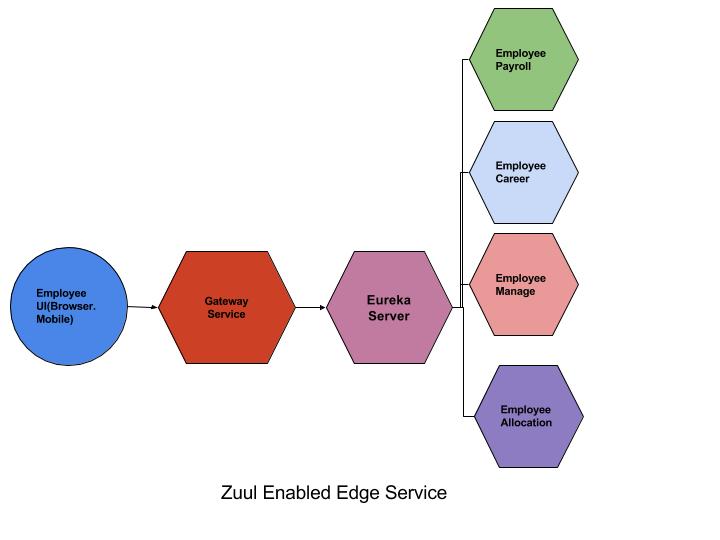
Para acessar as informações, seria necessário chamar todas as API’s REST para realizar a comunicação. Assim, a interface do usuário precisaria conhecer todos os detalhes do servidor e porta de microserviços para consultar os serviços.

Além das implementações de CORS, autenticação, segurança e monitoramento, comuns entre os microserviços, logo seriam replicadas em todos eles.

Alterações nos requisitos de autenticação ou na política de CORS repercutirão em todos os serviços. Para otimizar e tornar o projeto menos propenso a erros, é preciso ter um ponto de entrada onde todos os aspectos comuns sejam escritos, aqui surge o conceito de Zuul.



O Zuul atua como um API gateway. Ele recebe as solicitações e, em seguida, as delega para microserviços internos. É criado um microserviço que seja habilitado pelo Zuul. Sua API de serviço é exposta ao cliente / interface do usuário. O cliente chama esse serviço como um proxy para um microserviço interno e, em seguida, esse serviço delega a solicitação ao serviço apropriado.



*Fonte: https://dzone.com/articles/microservices-communication-zuul-api-gateway-1*