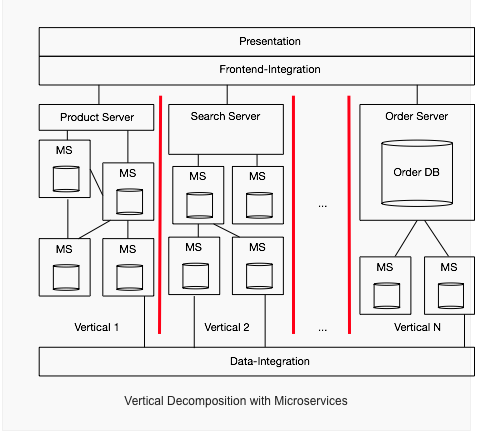
url: <http://dev.otto.de/2014/07/29/scaling-with-microservices-and-vertical-decomposition/>

keyword: microservices, architecture, decomposition

The monolith is first decomposed along domain and then into autonomous bounded context. The microservices along each domain is unaware of microservices in another domain. There can be various layer of integration within each domain microservices.



url: <http://www.slideshare.net/aahoogendoorn/modeling-microservices>; <file:///Users/d063398/Downloads/Shaping_Service_Orientation_Using_Smart_Use_Cases_.pdf>; https://www.youtube.com/watch?v=NrfuKTGvsFE

keywords: microservices, architecture, decomposition

The smart use case technique can be used to map microservices. It will help to decompose business process into various user goals, and each use goal into various smart use cases. Each smart use cases can be a candidate for microservices. And the use cases at higher level can be orchestration services.

url: <https://www.youtube.com/watch?v=yk_VlKUDKMA>

keyword: microservices, architecture

microservices architecture is solving not only software architecture problem but also infrastructure, operations. Overall it is also targeting devops. Miroservices should only do one function (either business or platform) and encapsulates the domain model completely (business rules and data model).

There can be various level of microservices depending upon their scope.

url: <http://practical-ddd.blogspot.de/2012/07/designing-aggregates.html>

keyword: microservices, modeling, aggregate, ddd

An aggregate has to be chosen as small as possible so that the single task performed on it is done within consistency. Two users don’ t have to be blocked when performing different request. This resembles SRP and can give hint towards the size of the microservice.

url: <https://www.microsoftpressstore.com/articles/article.aspx?p=2248811&seqNum=3>

keywords: microservices , bounded context

It talks about how to define boundary using relational patterns such as anti-curruption layer

url: <https://www.safaribooksonline.com/library/view/the-principles-of/9781491935811/part07.html>

keywords: microservices, principles, bounded context, guidelines, event storming

There are specific principles to implement microservices:

1. Modeled around business domain
2. Culture of automation
3. Hide implementation details
4. Decentralize all things
5. Consumer First
6. Isolate Failure
7. Highly Observable

url: <http://www.slideshare.net/jeppec/event-storming-48594742>; <http://ziobrando.blogspot.de/2013/11/introducing-event-storming.html>; <https://plus.google.com/s/%23EventStorming>;

<http://verraes.net/2014/07/event-storming-fake-domains-happy-paths/>;

<http://sch3lp.github.io/2014/07/12/event-storming-exercise/>;

http://verraes.net/2015/03/event-storming-storytelling-visualisations/

video: <https://skillsmatter.com/skillscasts/5193-alberto-brandolini#showModal?modal-signup-complete>; <http://www.slideshare.net/ziobrando/event-storming-recipes>

keyword: microservices, bounded context, domain, event storming

It describes about the technique of event storming.