Microservices

The important links first

- https://microservices.io/ - a lot of good advice. Must have bookmark.
- Self Contained Systems - very good things to consider when trying to cut services into smaller sizes
- The twelve factor App - even more good things to know!
- Domain Driven Design - some very good thoughts (not only related to microservices) that will help you a lot.
  also have a look at the articles from Martin Fowler related to this in his bliki.

Dealing with transactions in distributed systems

First of all: try not to. If you are sure you really have to: try harder, try some more. The consequences of skipping this step are complexity and a lot of work.

If you are willing to try hard, have a look at the links above, especially the SCS and DDD links.

Still reading? Well some links:

- The very famous SAGA pattern: https://microservices.io/patterns/data/saga.html
- Slideshare: Saturn 2018: Managing data consistency in a microservice architecture using Sagas from Chris Richardson, also listen to the audio track:

- Bern Rücker about this SAGA topic using the Camunda Workflow Engine.
  - The sample sources from the example.
  - and the all new https://zeebe.io/ engine.
- Similiar approach, but using the Flowable engine: Article at FreeCodeCamp.
  - based on Apache Camel, SEDA
- The eventuate tram framework.
- Another blog post at couchbase with some implementation hints.
- yet another implementation example at itnext.io
- implementation example using ActiveMQ.
- and there is also a series on this using the Axon Framework.