Material:

1. Article: Event Driven Communication
   1. Link: <https://docs.microsoft.com/en-us/dotnet/architecture/microservices/multi-container-microservice-net-applications/integration-event-based-microservice-communications>
2. Blog Post: Saga
   1. Link: <http://chrisrichardson.net/post/microservices/2019/07/09/developing-sagas-part-1.html>
3. Article: CQRS Pattern
   1. Link: <https://docs.microsoft.com/en-us/azure/architecture/patterns/cqrs>
4. Article: Event Sourcing
   1. Link: <https://docs.microsoft.com/en-us/azure/architecture/patterns/event-sourcing>
5. Event Sourcing Disadvantages
   1. Stack Overflow: <https://stackoverflow.com/questions/33279680/what-are-the-disadvantages-of-using-event-sourcing-and-cqrs>

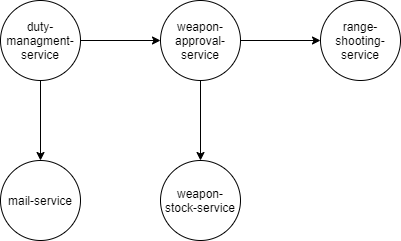
Questions:

1. How saving the state of our application in a fragmented way as this can help us investigate past errors?
2. Contrary to CRUD, rolling back an operation is no longer viable in a distributed system. How would you roll-back the state? How the asynchronous nature of the system can affect this?

Exercise:  
You have the following use case: A soldier needs to sign up for guard-duty, so he uses a web application.  
The user clicks on a button to approve his mission, the web browser sends a request to the duty-management-service, the duty-management-service then requests the weapon-approval-service for a weapon which in turn requests the range-shooting-service to verify he practiced shooting recently. Additionally, the weapon-approval-service requests the weapon-stock-service to check if there are enough weapons.

Also the duty-management-service makes a request to the mail-service to send an email for his commanding officer (so he knows that his soldier is on duty).

Diagram:



Convert the following communications to be event based, take into account failures and add necessary components as needed.