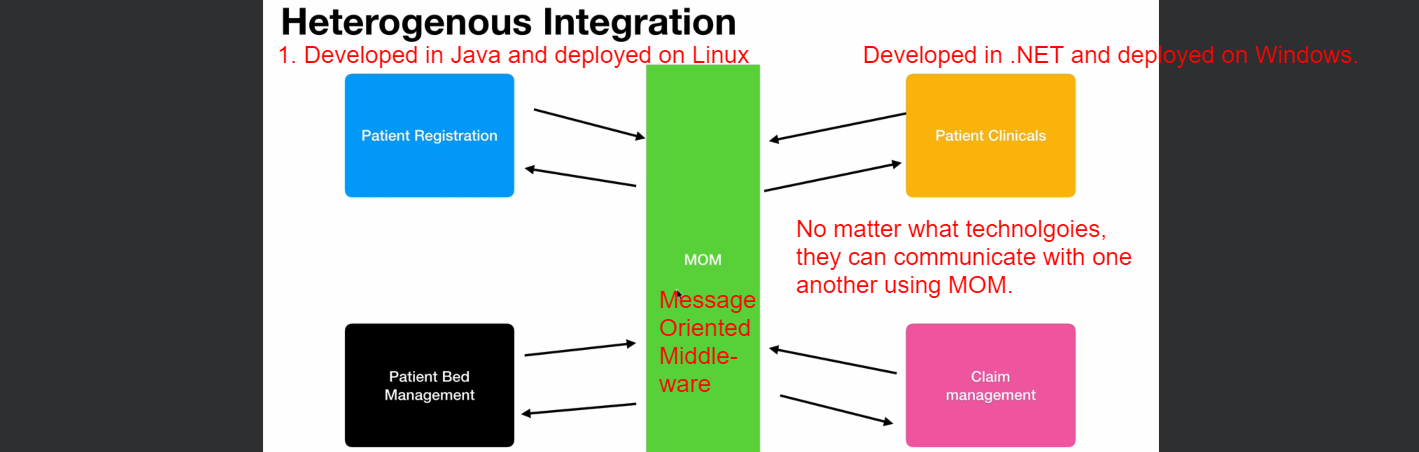
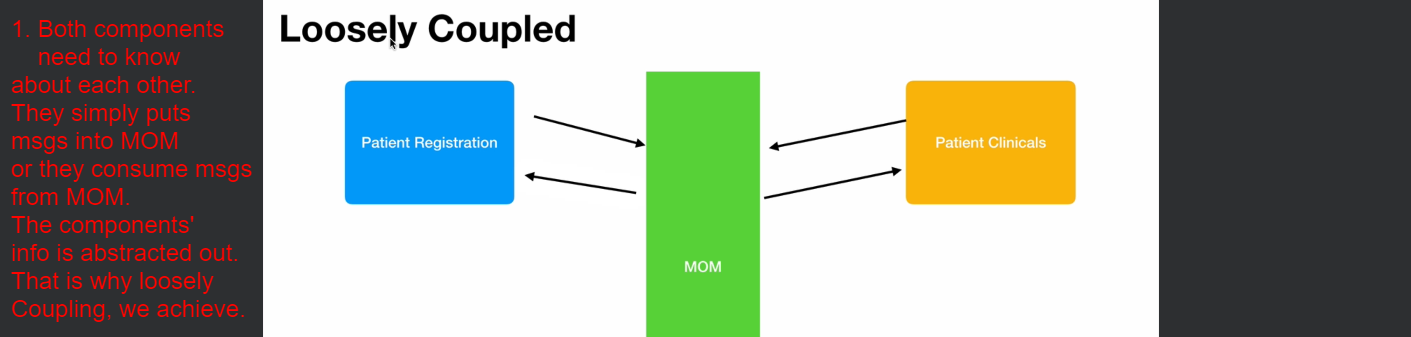
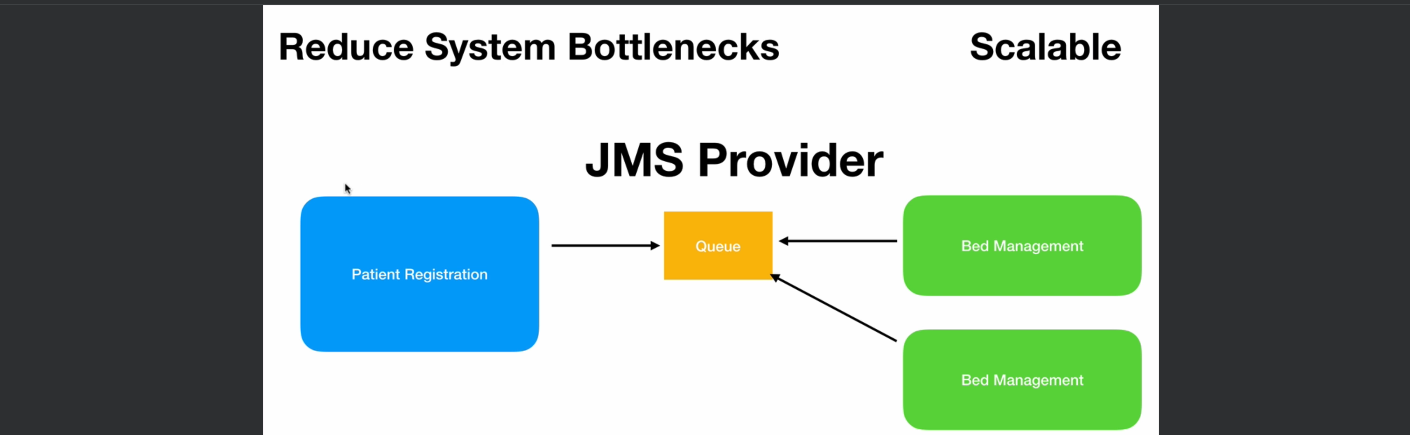
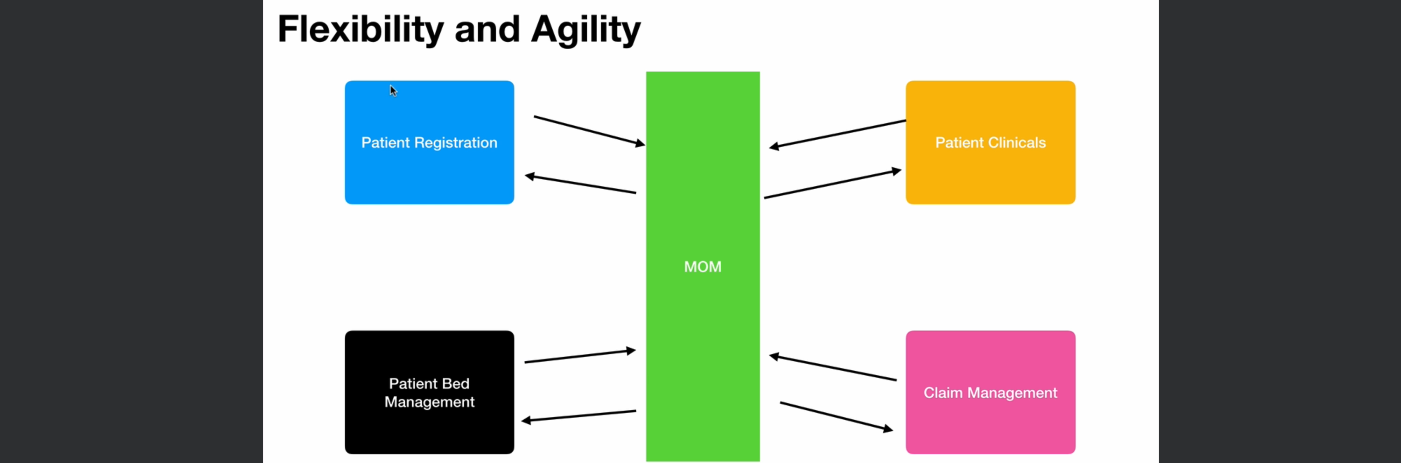
1. **Agenda**:
   1. Why to use Messaging?
2. Why means advantages of messaging.
3. **Advantages**:
4. **Heterogenous Integration**:
   1. The apps built using different technologies deployed on different Oss and they communicate with one another seamlessly.  
      
5. **Loosely Coupled**:
   1. Before Messaging came in, for apps to communicate with each other, they have to either shared a DB or use Remote Procedural Calls which are tightly coupled.  
        
      We can also use Web Service as MOM (Message Oriented Middle-ware) but messaging is more reliable as msgs are persisted in it and so not lost.  
      But in case of Web Service, a msg can be lost.
6. **Reduce System Bottlenecks**:  
   
   1. For example. In this patient management system, **Patient Registration** keeps on sending a lot of msgs to **Bed Management** to process as patients come into the hospital at some point.  
      At some time, there might be too many msgs on the queue and the Bed Management application will be clogged with too many msgs and the **Patient Management** has to wait for the responses to come back which will delay the processing of the incoming requests and as a result, the Patient Registration will experience delays.   
      Jatin: Basically above discussion is about when patient application sends a msg and waits for the response and meanwhile it doesn’t do anything and the following Solution discussion is when it sends msg and meanwhile it does something else instead of waiting.  
      **Solution**:  
      But messaging uses an option of asynchronous processing as well as we can use multiple consumers on the same queue that is as the load increases, we can spawn of multiple applications and they can work in parallel and also these can be in parallel (3:28) JMS allows us to create a asynchronous listeners and when a msg arrives the JMS provider (Like ActiveMQ) will call those asynchronous listeners and our calling app (patient application) needs not to wait for the response whenever the asynchronous response comes back, patient application can process the request and in the mean time, patient application can do some other work.  
      In that way, JMS Messaging reduces system bottleneck which will bring us to scalability so we can easily scale our app by creating multiple consumers or asynchronous listeners as load on our app increases also increasing end user productivity (in this case Patient App) is major advantage of messaging.  
      So, our end user will not face any delay due to scalability and asynchronous processing.
7. **Flexibility and Agility**:  
   
   1. It gives flexibility and agility.
   2. **Flexibility**: That is we can **replace** any of our architectural component or services with some other new application.  
      Like in the diagram, we can replace patient clinical component with a 3rd party component without affecting the current BAU (as long as contract is not broken).
   3. **Agility**: In the continuously changing world/environment, we can **modify** each of these apps without affecting other app/service.