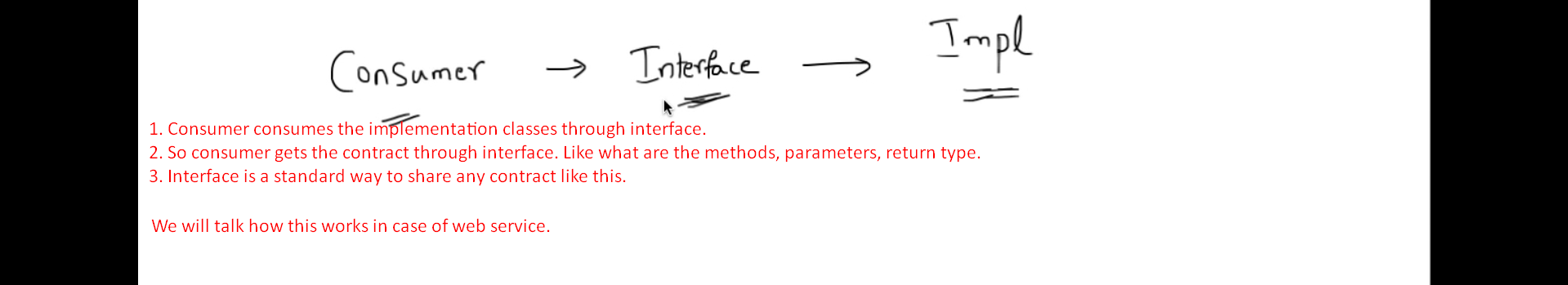
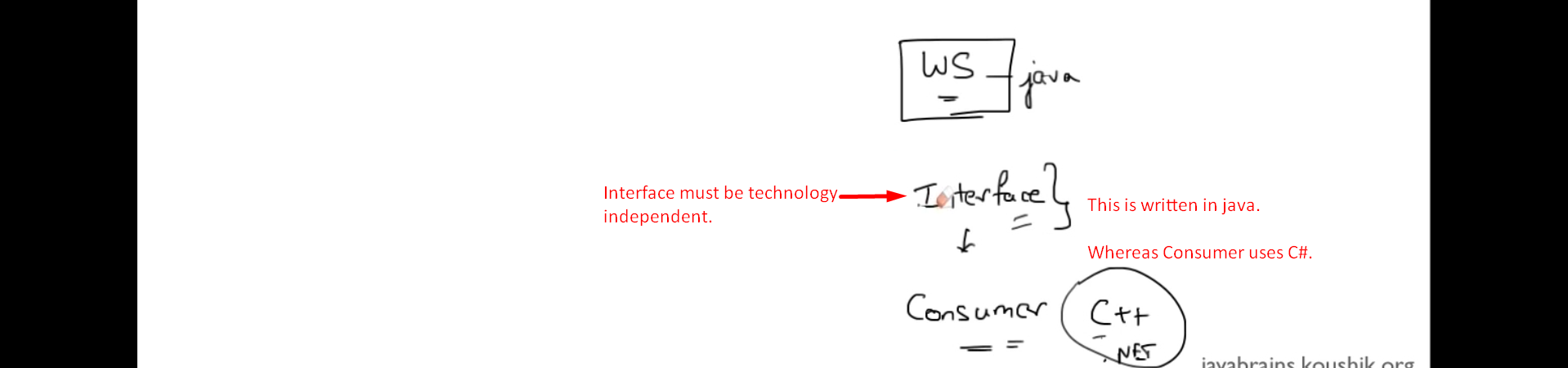
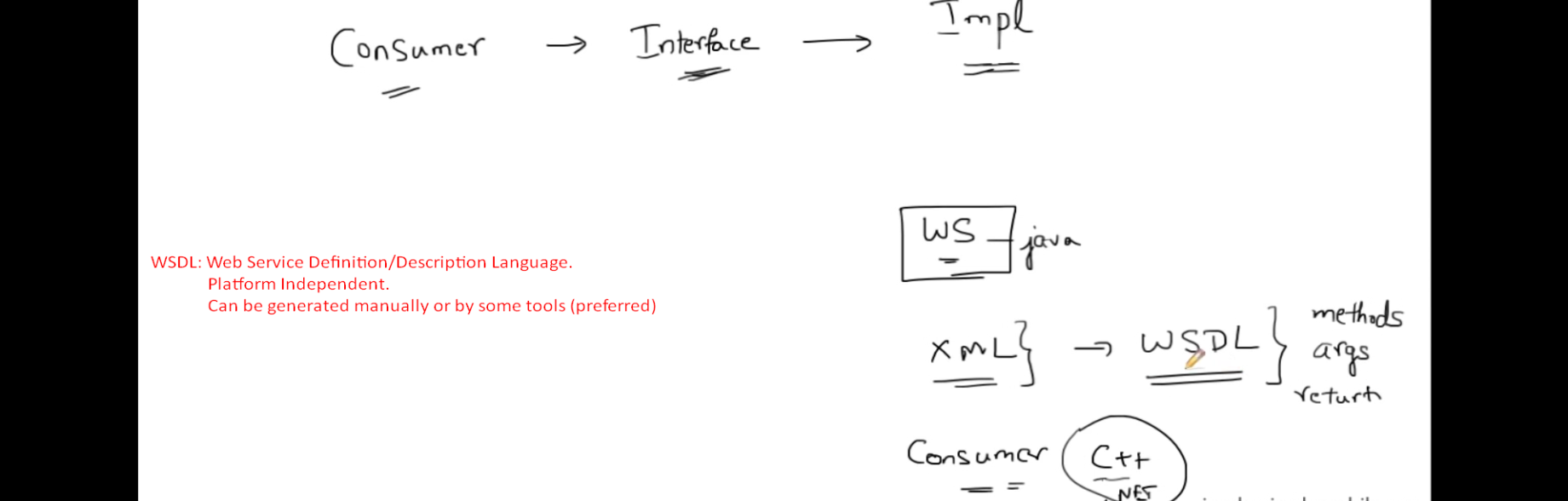
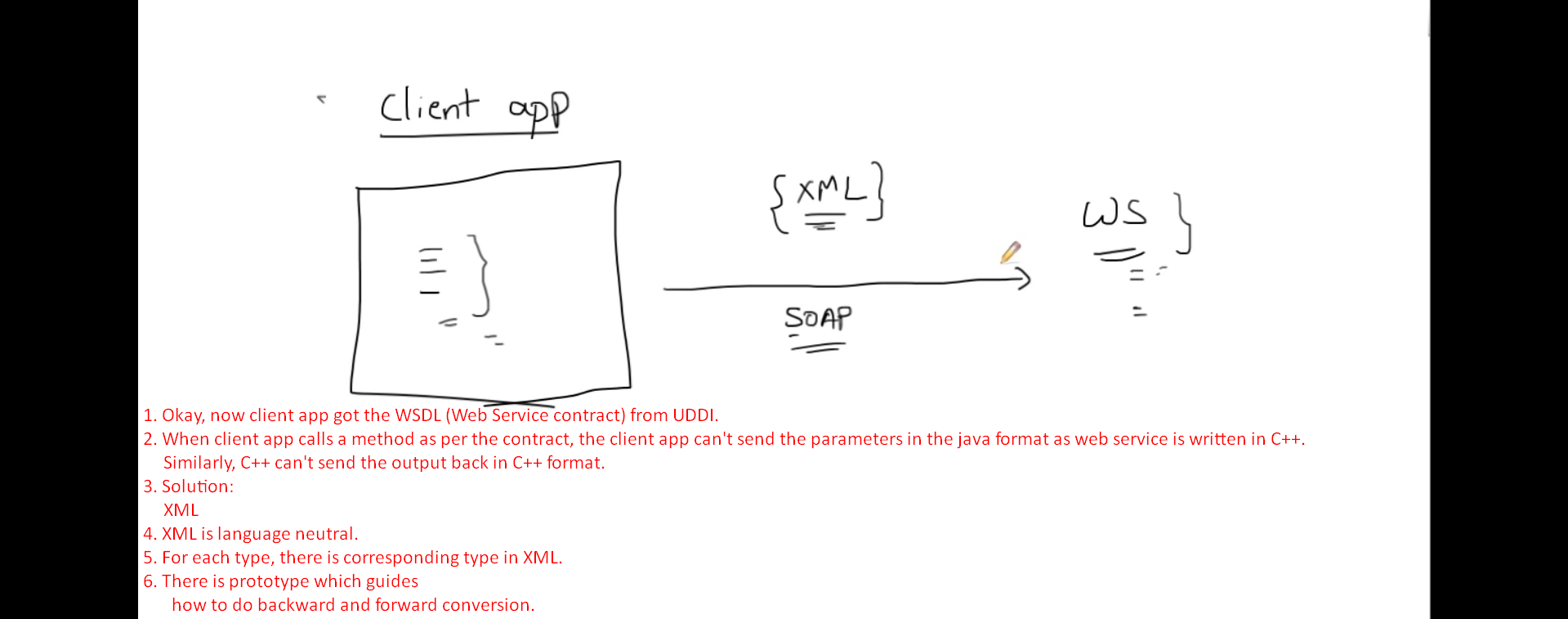
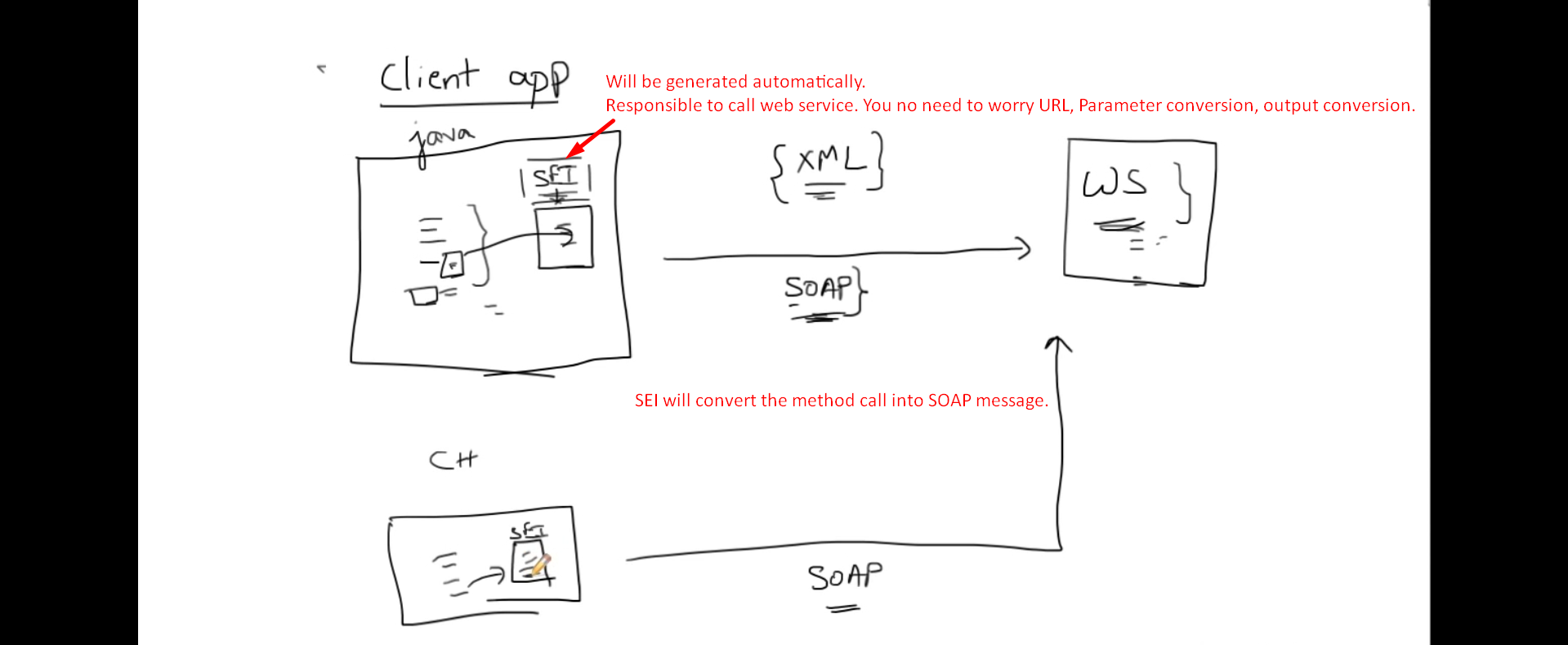
1. Let’s say we want to share the implementation classes with consumer.  
   In other words, the contract of implementation classes.
   1. **Answer**: Through an interface.
   2. 
2. **Web Service**:
   1. We have web service implementation.
   2. Now we want to share this info with consumer.
   3. We create contract by interfaces.  
      Will sharing interfaces with consumer work?  
      Probably, not. As the consumer may be using some other technologies like C++, C#.
   4. **Solution**: The interface must be technology independent.   
      
3. The creator of the web service specification when thought about this problem, they came up with a format that can be understood by all the technologies (consumers).  
   That format was XML.
4. So now when anyone creates a web service, they share the contract in the format of XML.
5. This XML documentation is called 🡺 WSDL (Web Service Description/Definition Language).
6. We don’t need to create this WSDL manually even though we can create it manually but we have tools which can generate WSDL.
7. 
8. Now suppose you’re writing some client app to consume web service.  
   But the question is how to hold on WSDL which is contract to web service.
   1. **UDDI**:
      1. Universal Description Discovery & Integration.
      2. A directory/registry.
      3. Where web service is registered.
   2. 
   3. Let’s say you queried UDDI and got the WSDL for the web service which you want to consume.
   4. 
   5. This type conversion from java to XML type and from xml type into java is done intermediate classes.  
      Actually, the whole method call is converted into SOAP message.  
      These intermediate classes are called SEI (Service Endpoint Interface).
   6. 
   7. 