Security Mechanisms  
  
1. Input validations  
Validate all input parameters - Length, Type, Sanitize user input, Size etc, content types.  
Use secure parsers to avoid json/xml parsing vulnerabilities  
Log all validation failures  
  
2. Further Authorization and Authentication can be done through various mechanisms like OAuth,OpenID, Api Keys, JWT , JWT with encrypted payload etc.   
Since it's mentioned in the assessment about  a week's time up to delivery, I would go with JWT with encrypted payload.

Once user authenticates by passing id/password or any other sign-in details, the Authentication Service/ Server can perform the authentication, create a JWT token for a short duration and return it to user. For subsequent API calls, for a fixed duration, user then would pass the JWT token to Application Server, Application Server verifies and processes API calls. Since the data inside JWT is encoded and not encrypted, to achieve additional security, the payload can be further encrypted. Further, JWT should be sent over HTTPS. Once the token expires the same process can be repeated to generate new tokens. Refresh tokens can be used for this purpose, i.e to generate new tokens up on expiration of old ones. For Spring boot, spring-security-jwt, that Spring provides can be integrated with Spring Security OAuth.

3. To secure data, data in transit, storage needs to encrypted and should happen on secure channels. Checksums and Hashes can verify the integrity of data.

4. Additional security measures like rate limiting needs to be considered.