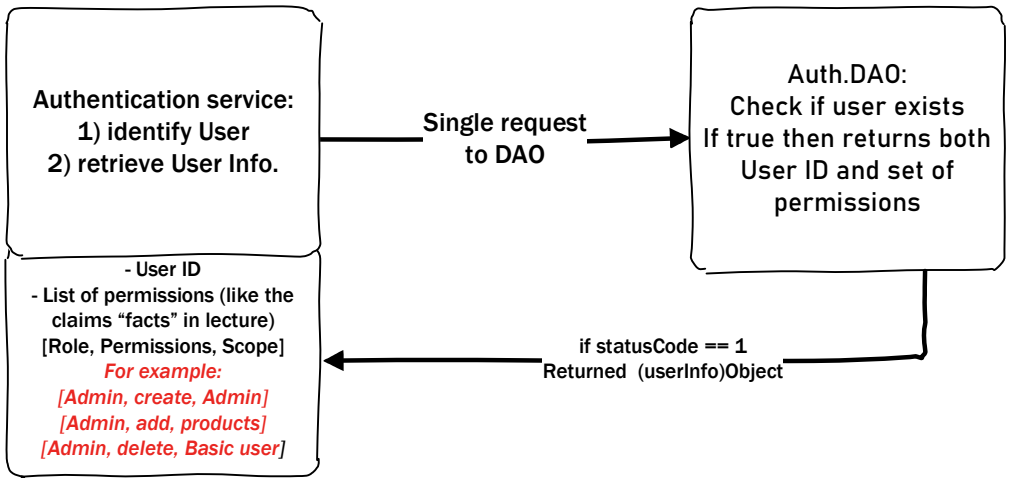
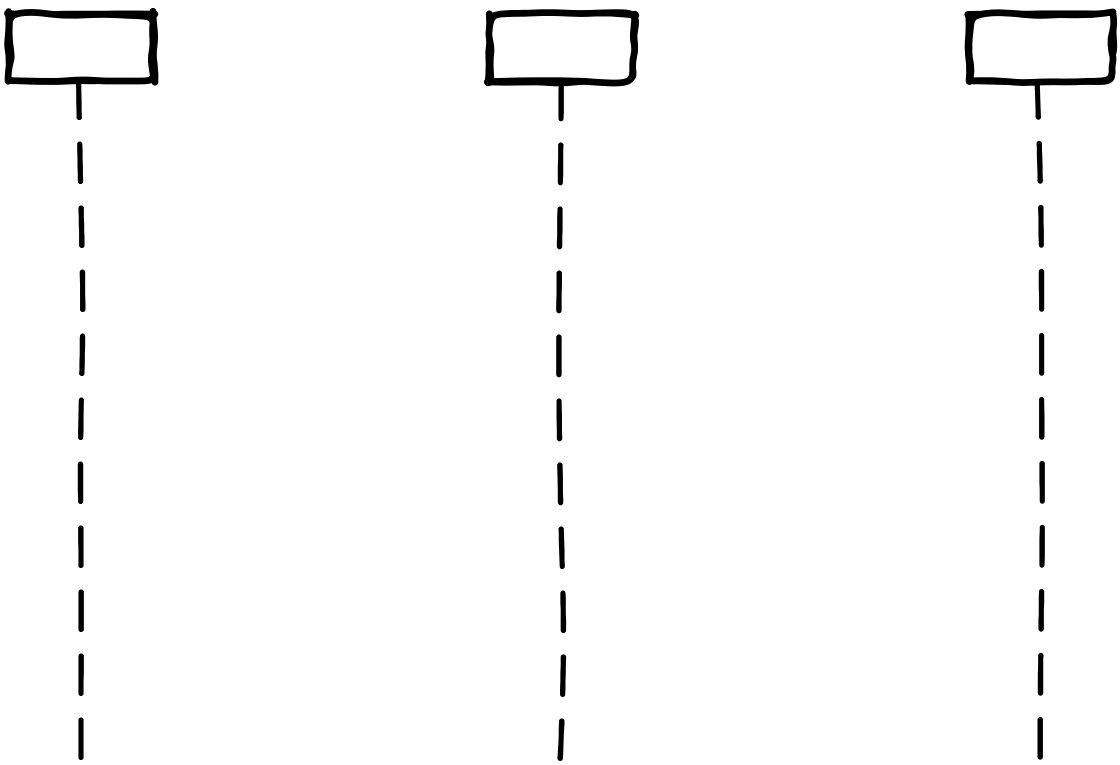
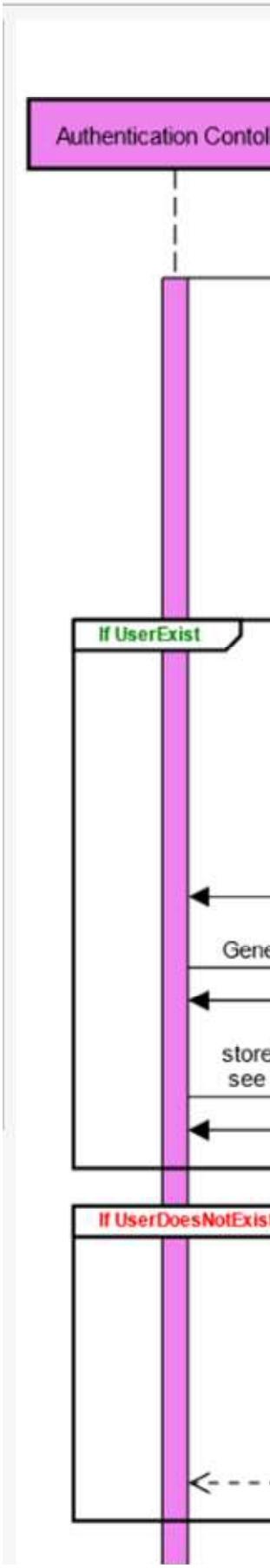


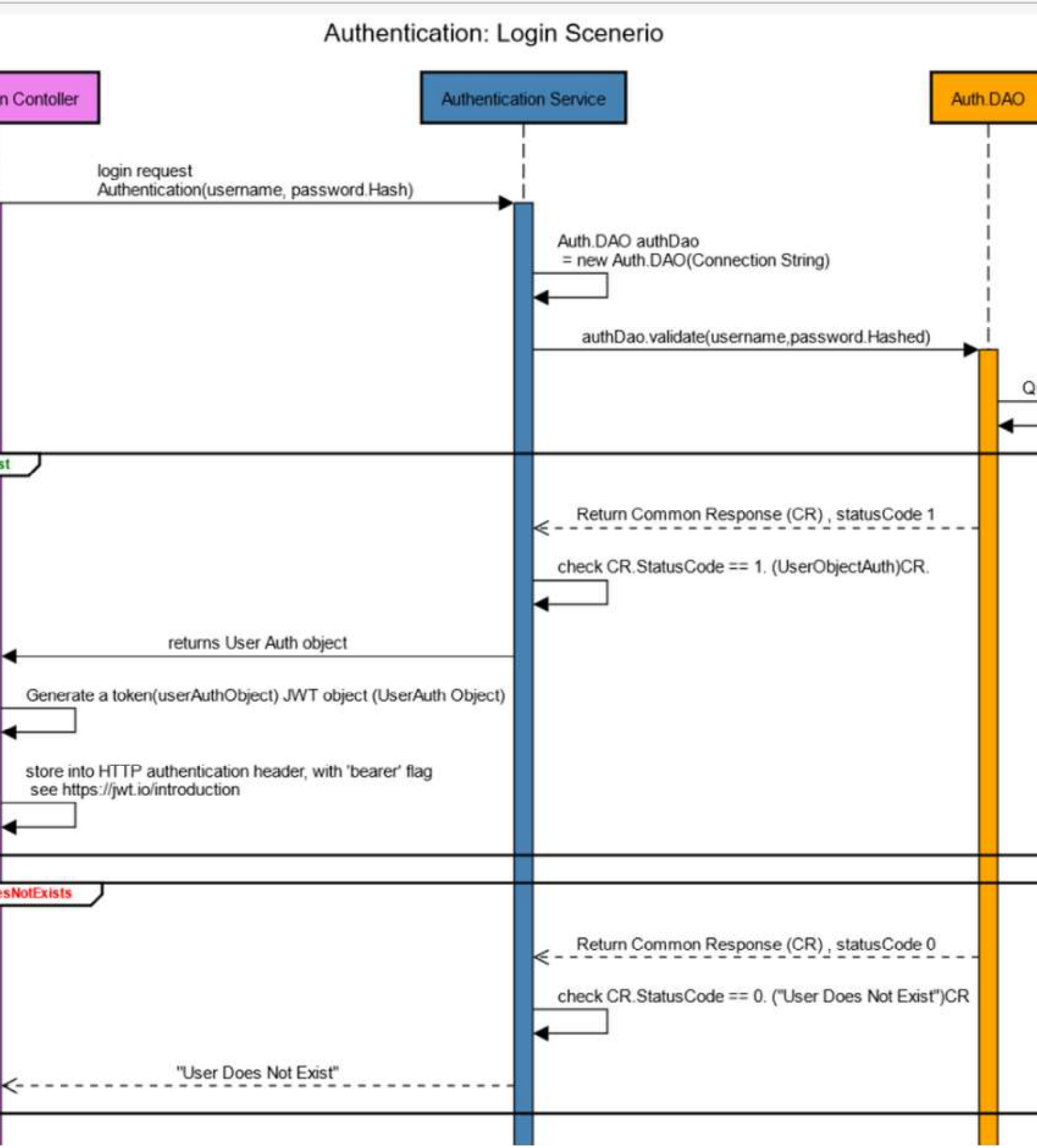
Success, timeout,failure.
Status code -> not enough info.
Errorcode = cateory
Error message = specific mssg

Not maintainable. List the possible categories and let the statuc message to be the msg

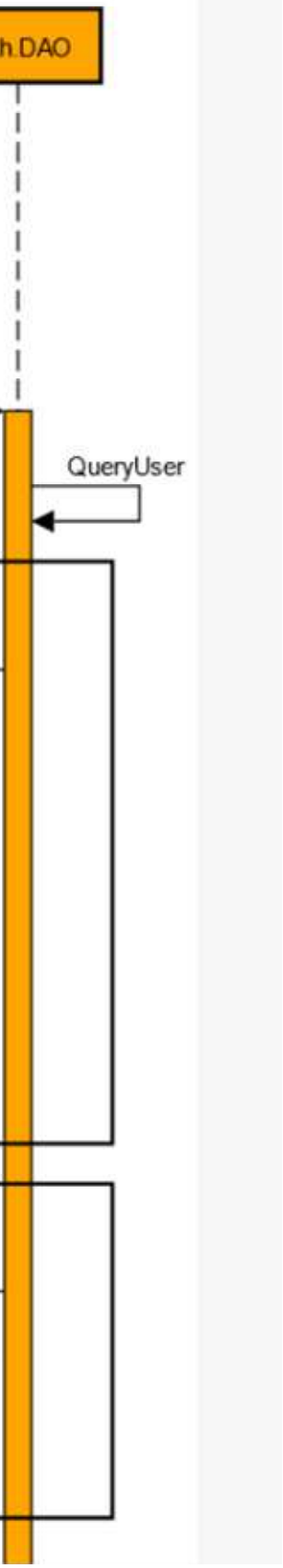
Make a clear distinction:
This an object rest of system knows about
This is an object that only x layer knows about
Or use a DTO




Authentication Contol
If UserExist
Gene
store
see
If UserDoesNotExis

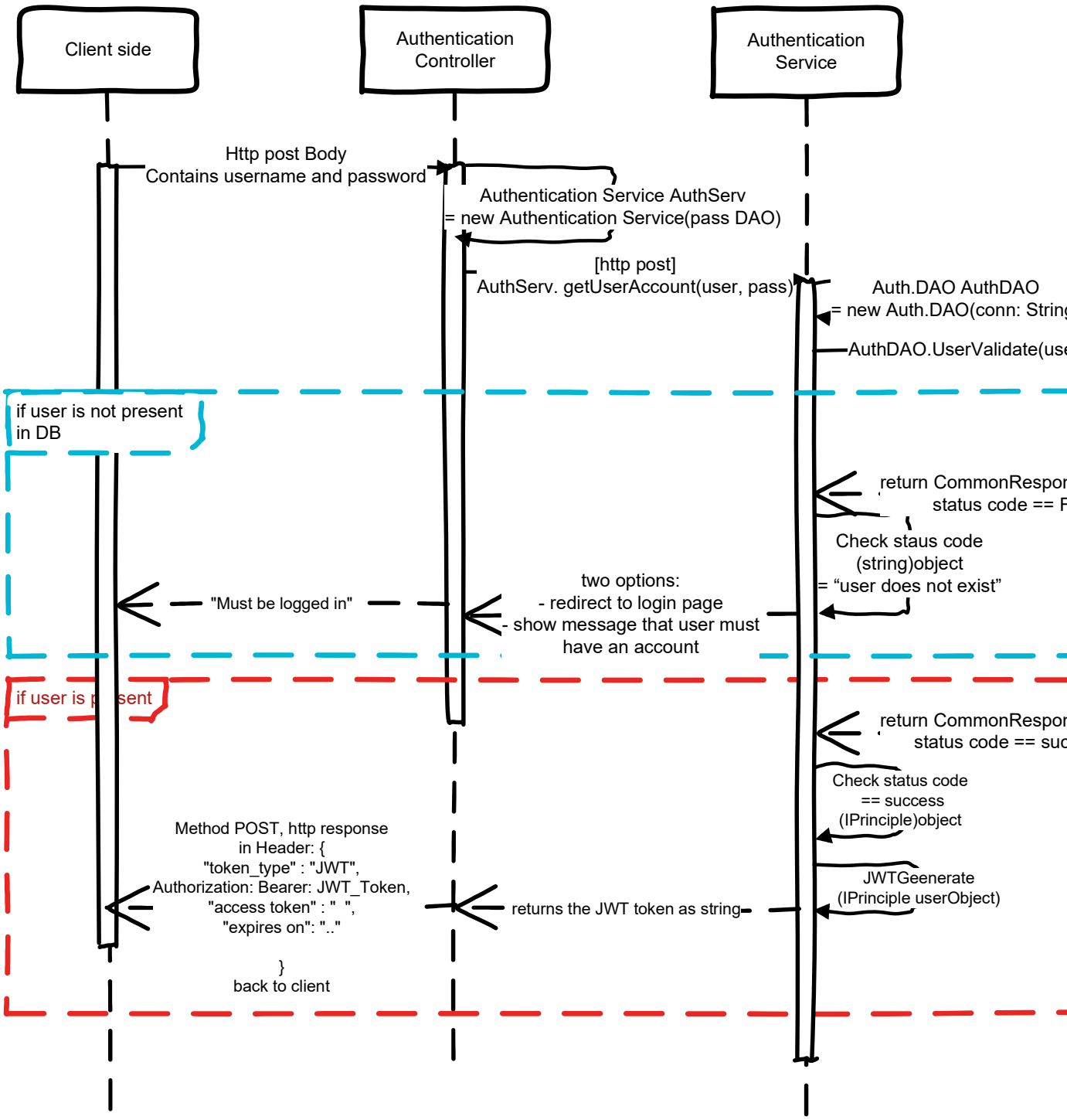

Authentication: Login Scenerio
n Contoller
Authentication Service
Auth.DAO
login request
Authentication(username, password.Hash)
Auth.DAO authDao
= new Auth.DAO(Connection String)
authDao.validate(username,password.Hashed)
Q
st
Return Common Response (CR) , statusCode 1
check CR.StatusCode == 1. (UserObjectAuth)CR.
returns User Auth object
Generate a token(userAuthObject) JWT object (UserAuth Object)
store into HTTP authentication header, with 'bearer' flag
see https://jwt.io/introduction
esNotExists
Return Common Response (CR) , statusCode 0
check CR.StatusCode == 0. ("User Does Not Exist")CR
"User Does Not Exist"

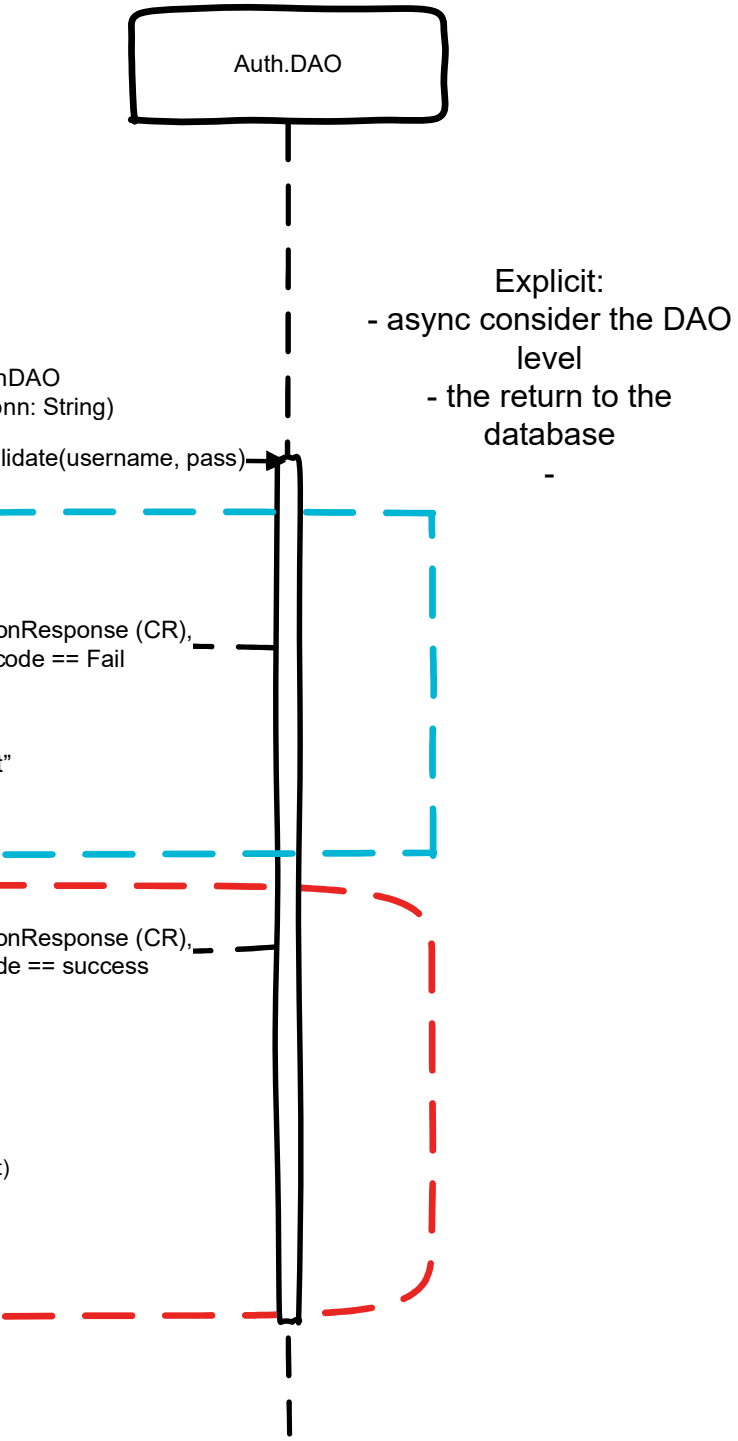
dddd


h DAO
QueryUser

Scenario: request to

equest to login





Explicit:
- async consider the DAO level
- the return to the database
-

3 Answers

111

JWT is an encoding
JSON data payload

JWT can be used f
tokens, i.e. a piece
some service that b
"bearer") grants you

Bearer tokens can
different ways, one
being the Authoriza
into a request para
That is mostly betw
access.

Using in our API client to pass it as a header to every API call
Check if a user is logged in by seeing if the JWT variable is set.
Optionally, we can even decode the JWT on the client to access data in the payload. Let's say we need the user-id or the username on the client, which we can extract from the JWT.

Active | Oldest | Votes

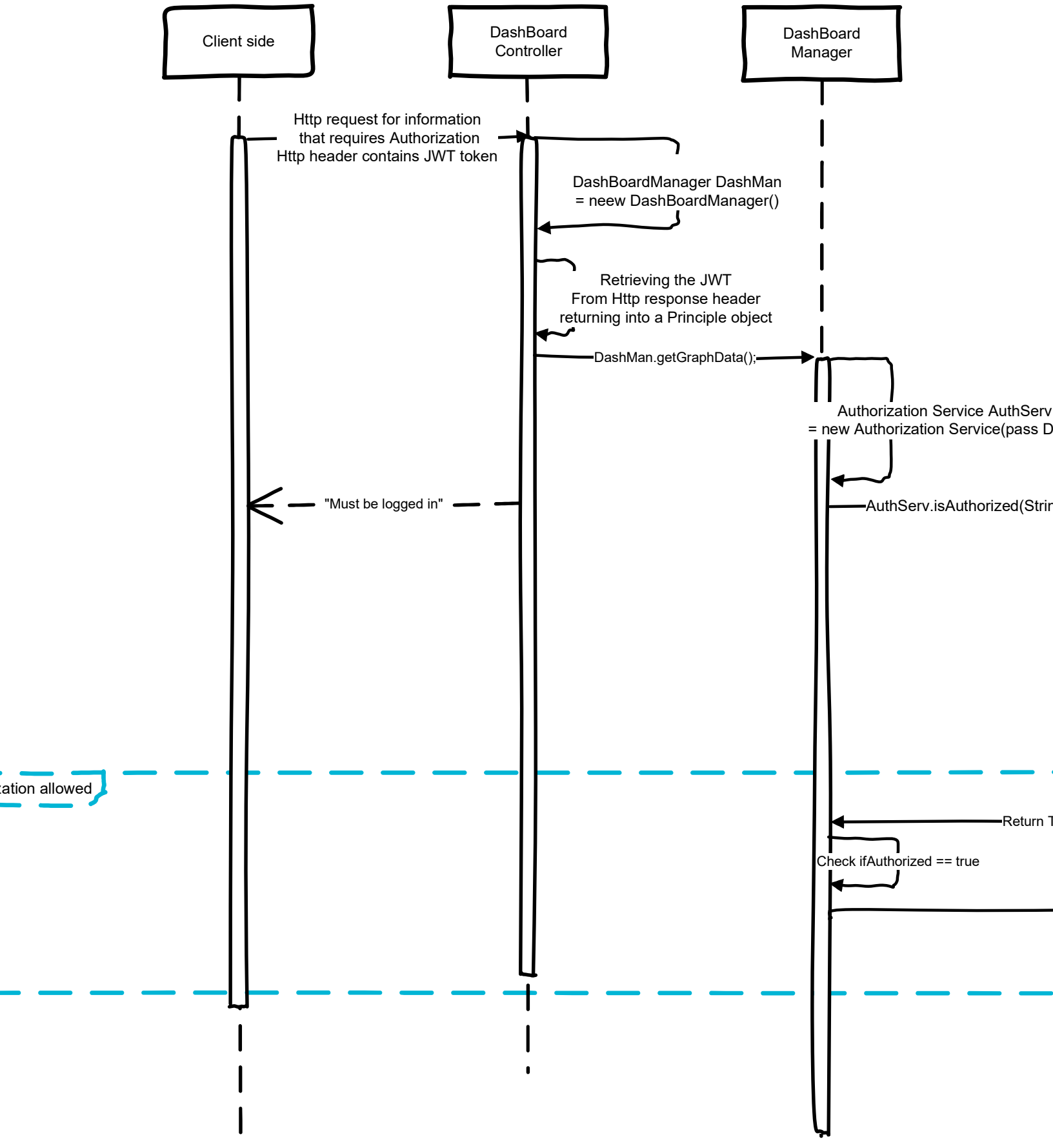
oding standard for tokens that contains a
yload that can be signed and encrypted.

sed for many things, among those are bearer
iece of information that you can present to
that by virtue of you having it (you being the
ts you access to something.

can be included in an HTTP request in
one of them (probably the preferred one)
orization header. But you could also put it
parameter, a cookie or the request body.
between you and the server you are trying to

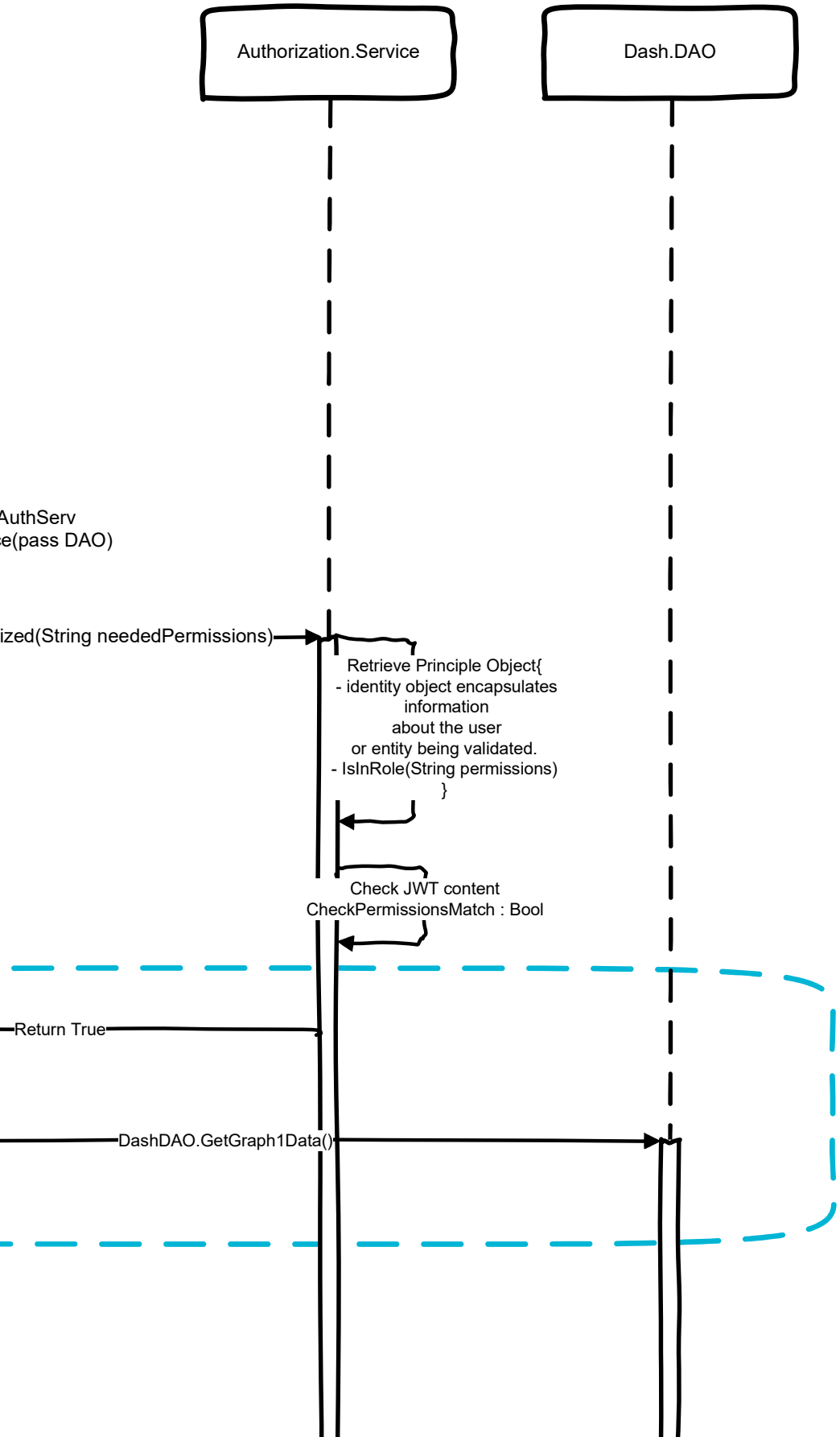

Authorization allo

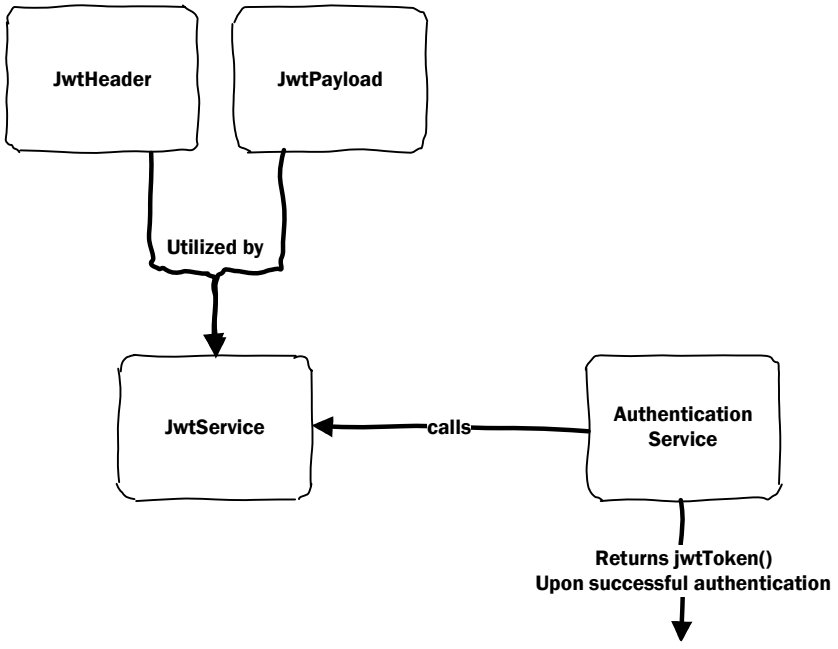
Scenario: request that needs authorization



access data in the payload. Let's say we need the user-id or the username on the client, which we can extract from the JWT.

https://hasura.io/blog/best-practices-of-using-jwt-with-graphql/

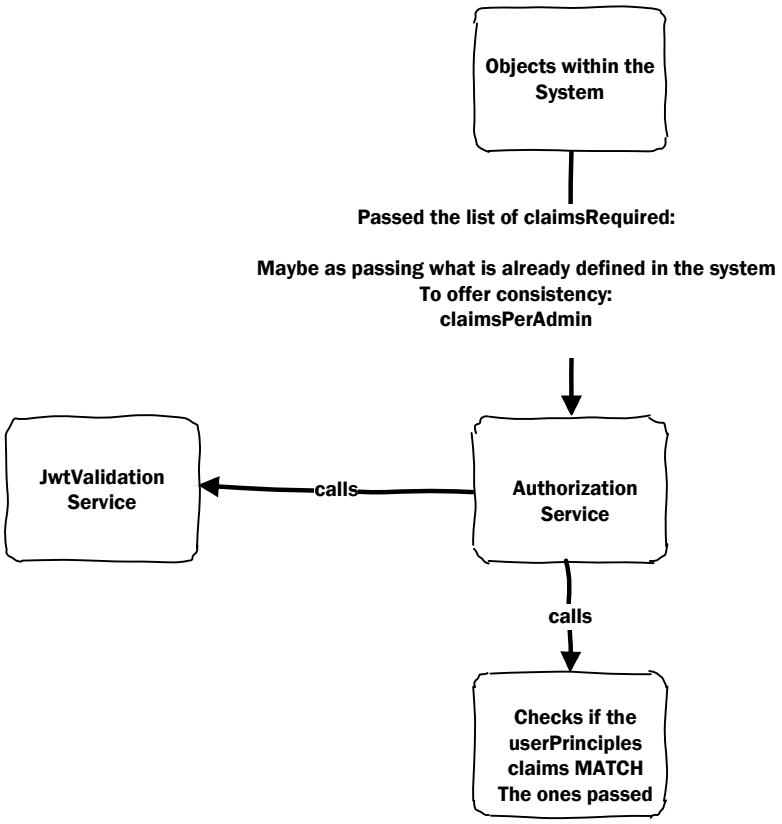




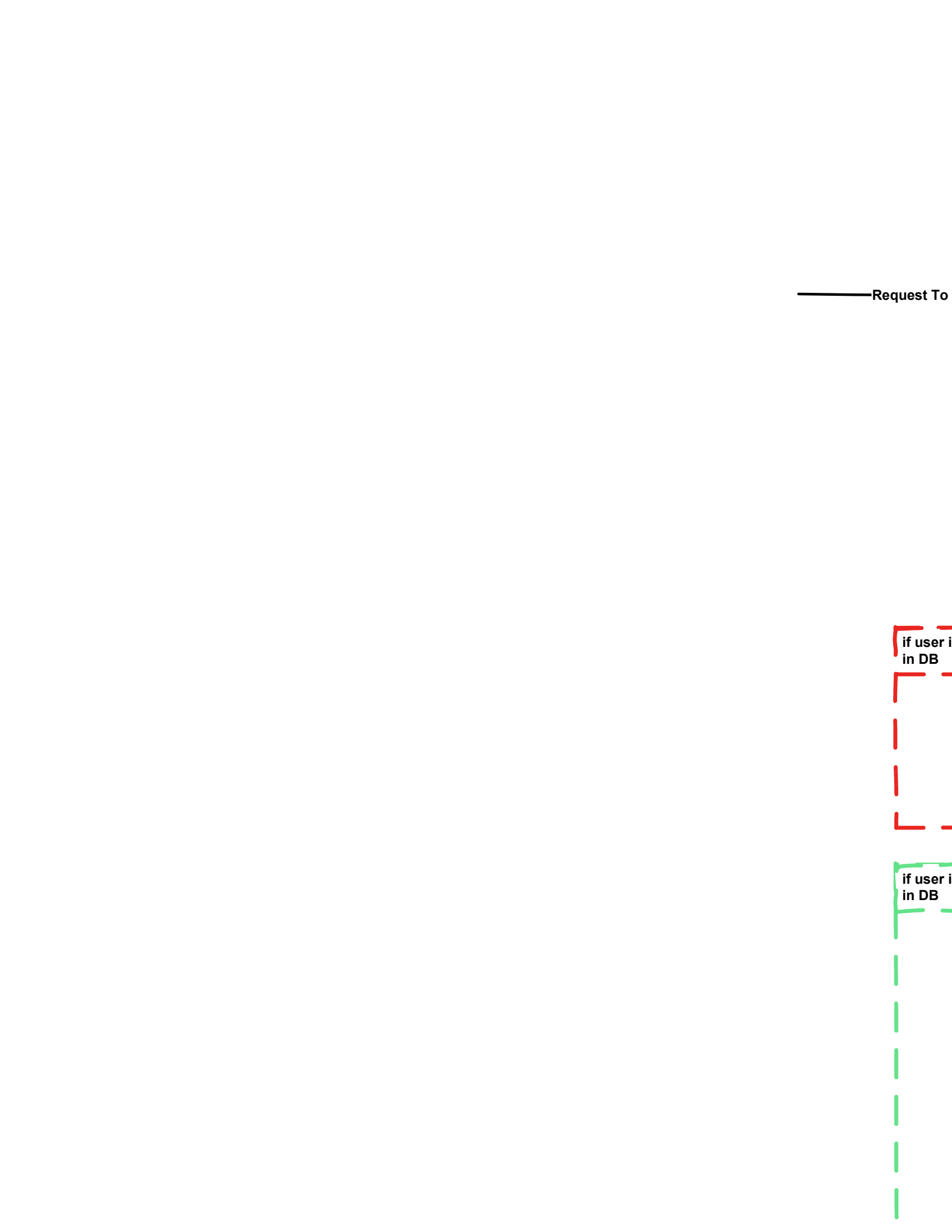
1. Notes go here!

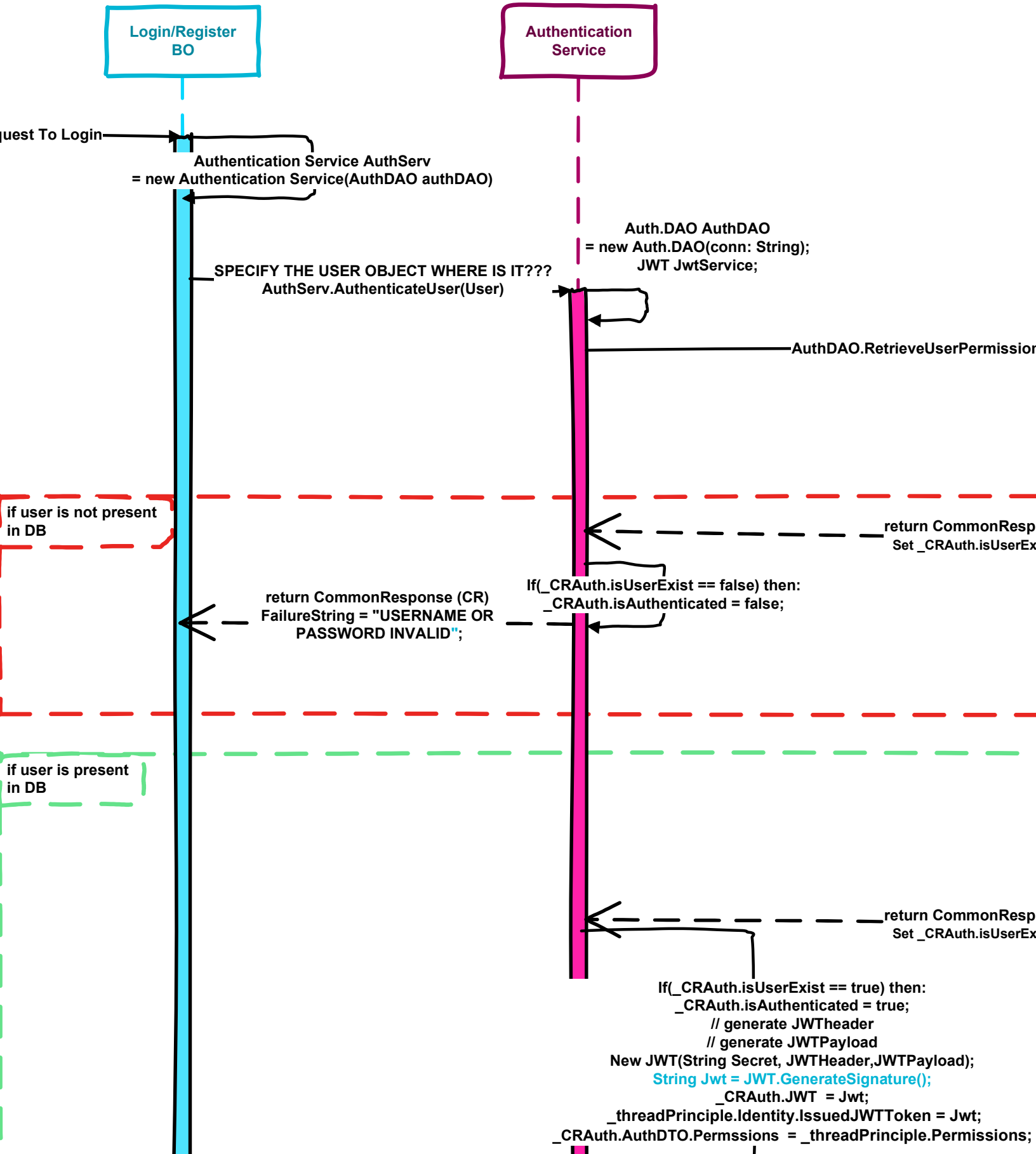
upon successful Authentication of a user then the Authentication service will generate a JWT string to return to Be stored into the http header {“authorization” : “Bearer: JWT”}

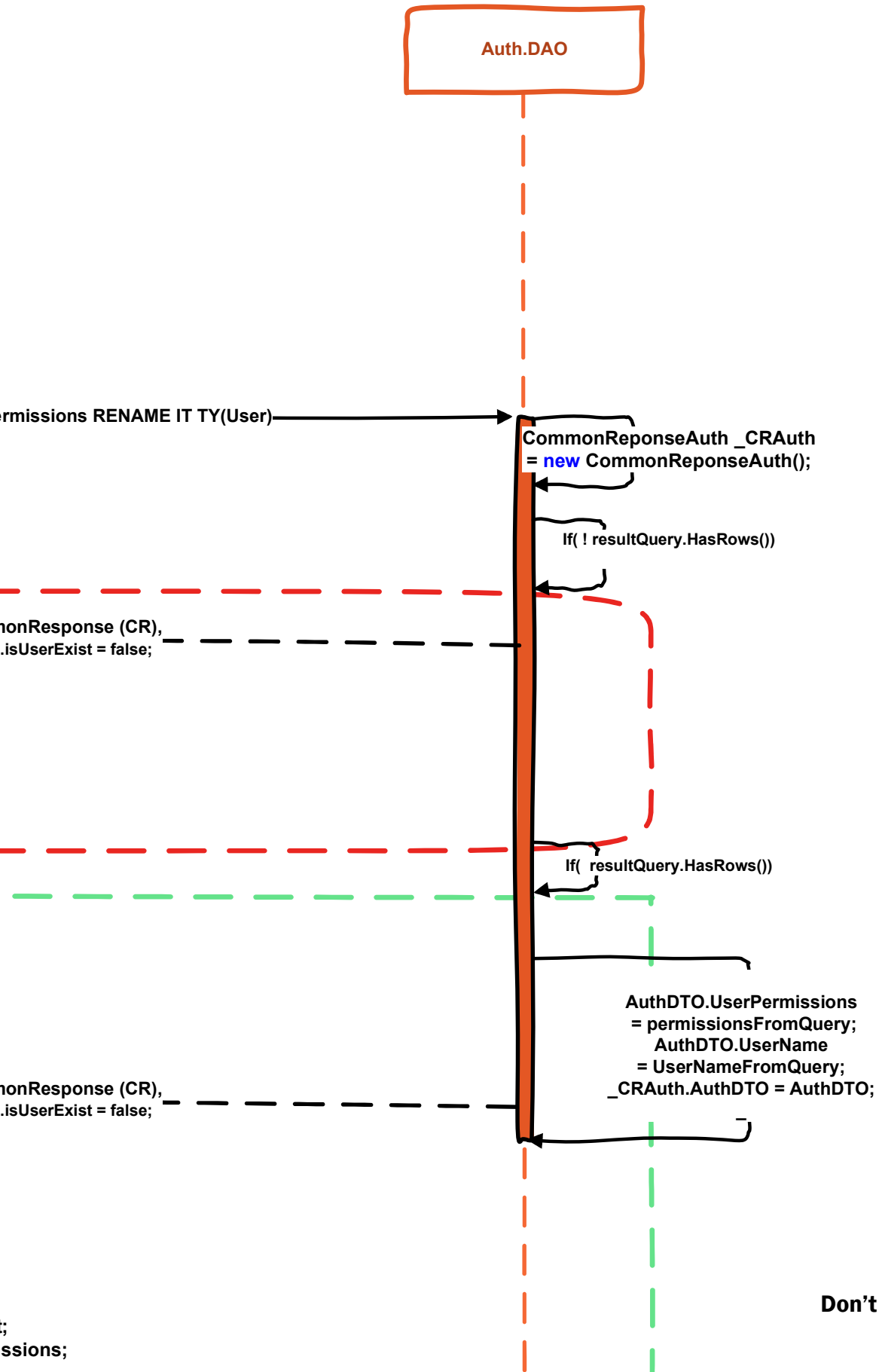
ALSO if the user authenticated then the userPrinciple will be updated to hold the JWT token, $ isAuthenticated = true.

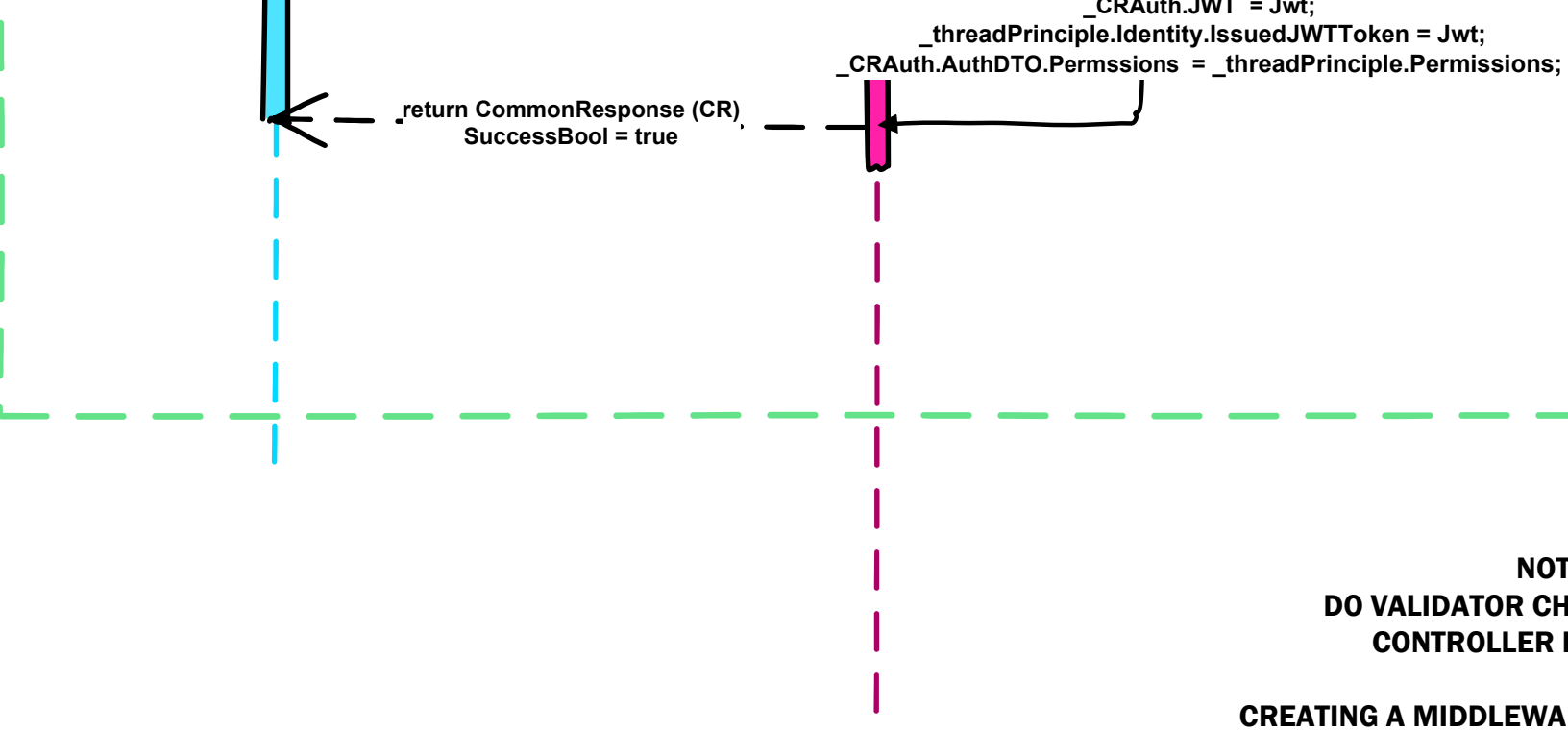



```
1.example of claims:
IList<Claims>
claimsPerAdmin = new List<Claims>()
        {  new Claims("all","all"),  };


IList<Claims>
 claimsPerVendor = new List<Claims>()
        {
new Claims("ReadOnly","AutoBuild"),
new Claims("Delete","self"),
new Claims("Update","self"),
new Claims("Edit","self"),
new Claims("Create" ,"reviews"),
new Claims("Delete" ,"selfReview"),
 new Claims("Update" ,"selfReview"),
new Claims("Create","Products"),
new Claims("Update","VendorProducts"),
new
Claims("Delete","VendorProductsOnly"),
new Claims("Update","VendorProducts")

        };
```


Request To
if user i
in DB
if user i
in DB

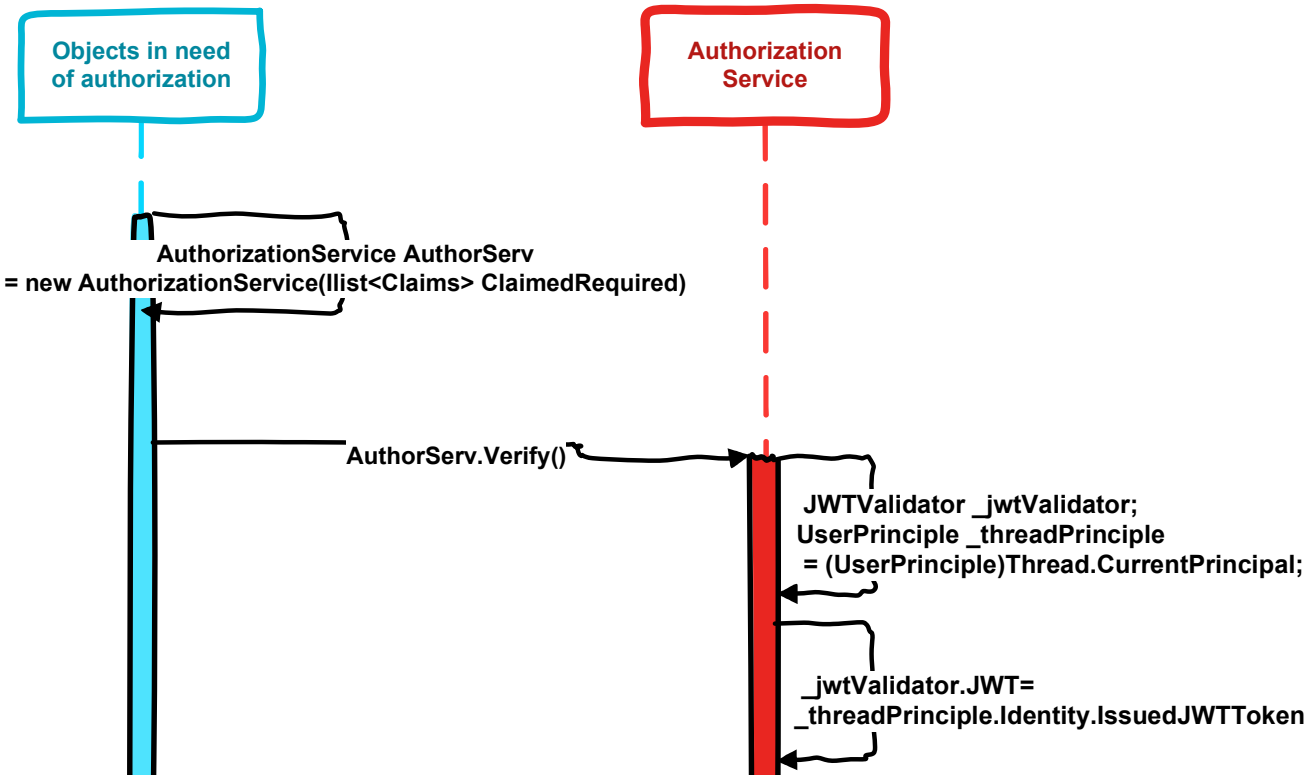

Login/Register BO
Authentication Service
quest To Login
Authentication Service AuthServ = new Authentication Service(AuthDAO authDAO)
Auth.DAO AuthDAO = new Auth.DAO(conn: String); JWT JwtService;
SPECIFY THE USER OBJECT WHERE IS IT??? AuthServ.AuthenticateUser(User)
AuthDAO.RetrieveUserPermissior
if user is not present in DB
return CommonResp
Set _CRAuth.isUserEx
If(_CRAuth.isUserExist == false) then: _CRAuth.isAuthenticated = false;
return CommonResponse (CR) FailureString = "USERNAME OR PASSWORD INVALID";
if user is present in DB
return CommonResp
Set _CRAuth.isUserEx
If(_CRAuth.isUserExist == true) then:
_CRAuth.isAuthenticated = true;
// generate JWTheader
// generate JWTPayload
New JWT(String Secret, JWTHeader,JWTPayload);
String Jwt = JWT.GenerateSignature();
_CRAuth.JWT = Jwt;
_threadPrinciple.Identity.IssuedJWTToken = Jwt;
_CRAuth.AuthDTO.Permssions = _threadPrinciple.Permissions;


Auth.DAO
ermissions RENAME IT TY(User)
CommonReponseAuth _CRAuth
= new CommonReponseAuth();
If(! resultQuery.HasRows())
monResponse (CR),
.isUserExist = false;
If(resultQuery.HasRows())
AuthDTO.UserPermissions
= permissionsFromQuery;
AuthDTO.UserName
= UserNameFromQuery;
_CRAuth.AuthDTO = AuthDTO;
monResponse (CR),
.isUserExist = false;
Don’t stor
;
ssions;



NOT
DO VALIDATOR CH
CONTROLLER I

CREATING A MIDDLEWA
HTTP REQUESTS HIT THE
THE CODE. CREATE A C
PROS: EARLY EXIT IN CA
FA
EVERY REQUEST-RESPO
B4 REACHING YOU CODE
EARLY BEFORE REACH
WHY IN OUR CASE: SIN
ALL THE TIME F

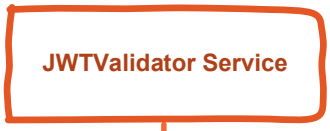


;
ssions;

Don't stor

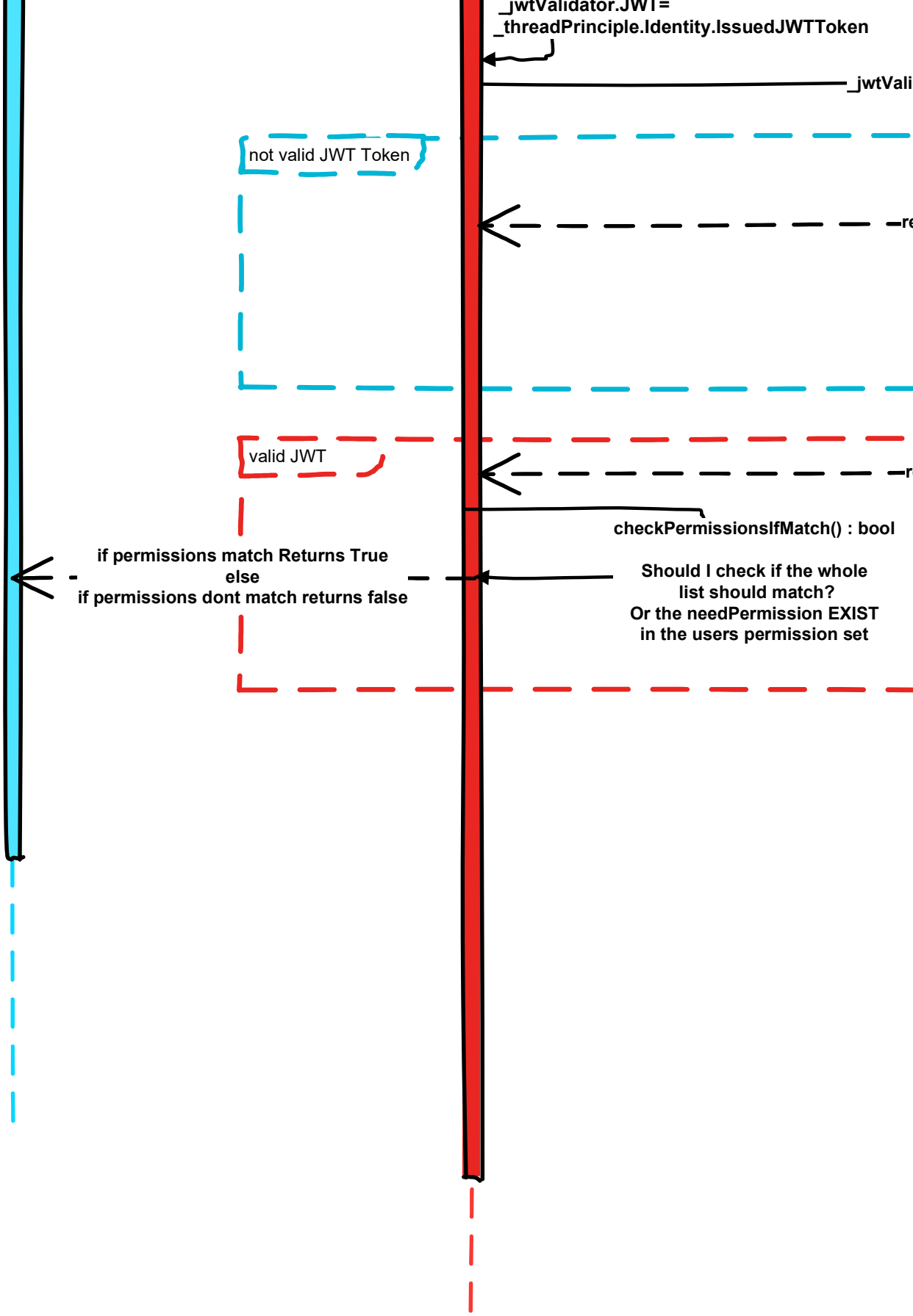
NOTES:
TOR CHECK BEFORE THE
OLLER IS CREATED ().

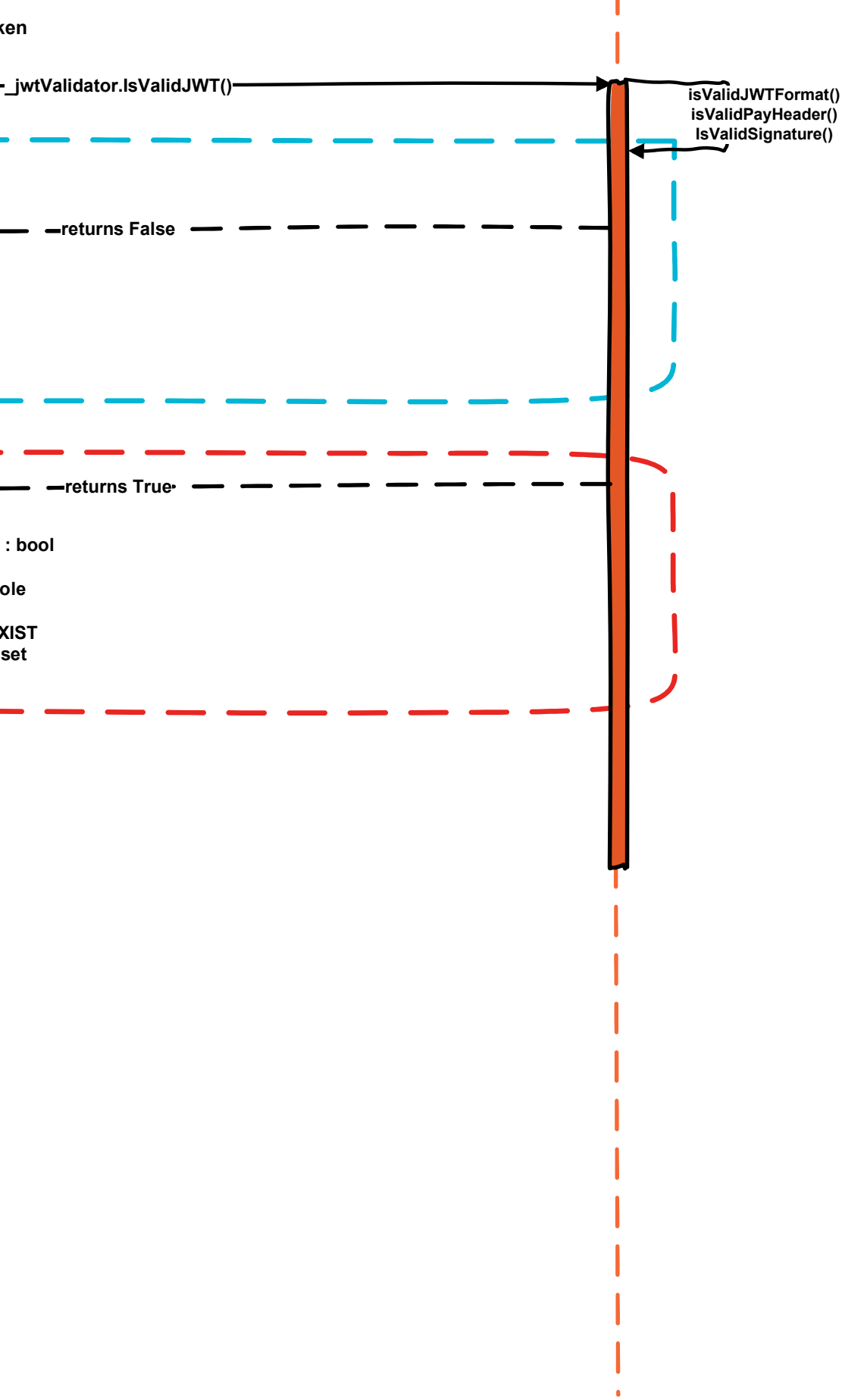
OLEWARE (HTTP PIPELINE) THE
HIT THE PIPLINE B4 IT GOES TO
ATE A CUSTOM MIDDLEWARE.
IT IN CASE OF JWTVALIDATION
FAIL.
RESPONSE HAS COME PIPLINE
U CODE. GOAL: KILL A REQUEST
REACHING ENDING STAGES.
SE: SINCE WE ARE CHECKING
TIME FOR JWT VALID.

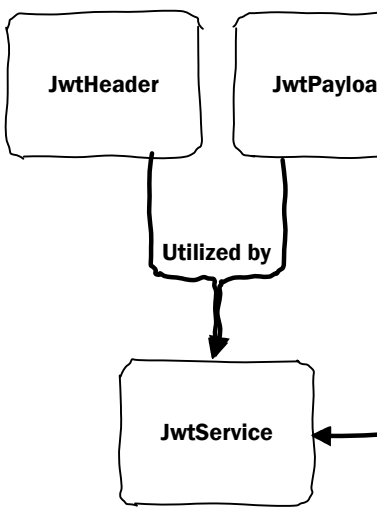


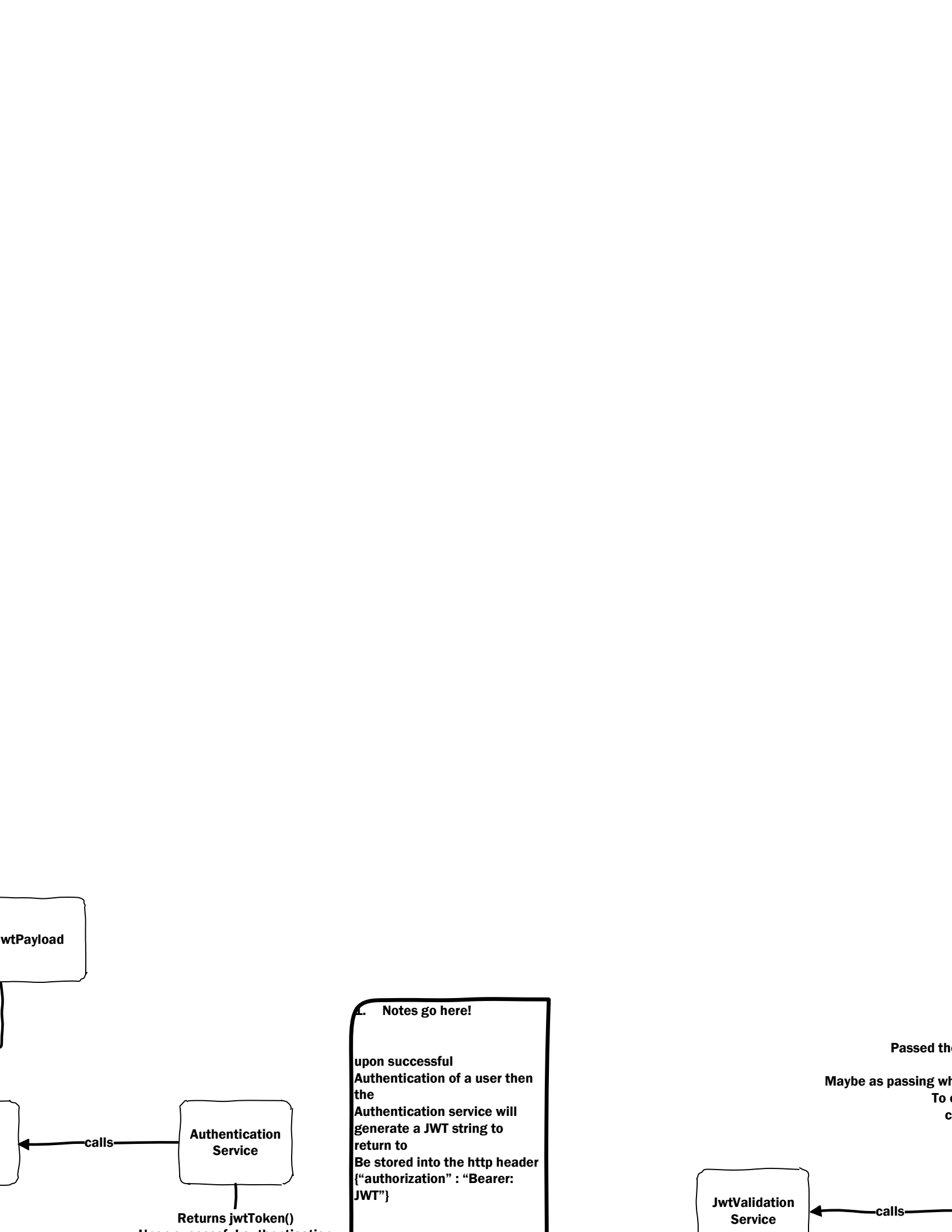
al;

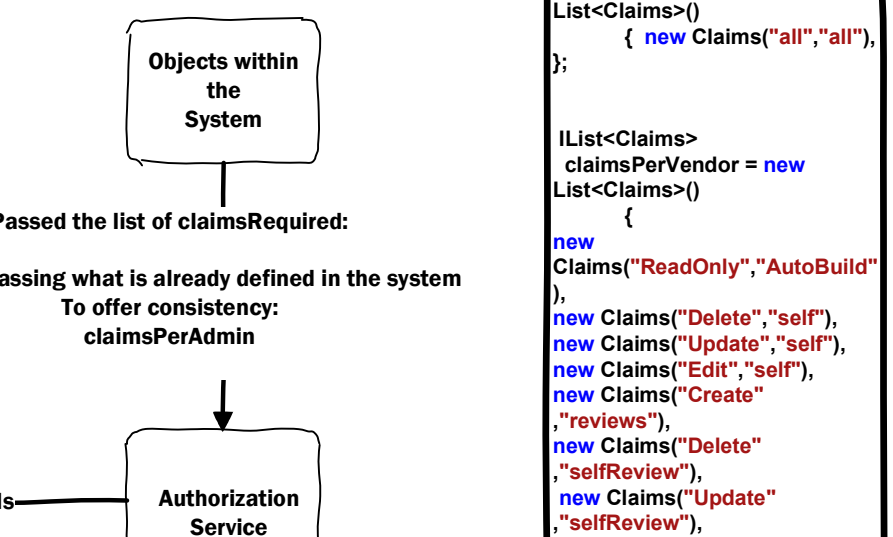
ken


_jwtValidator.JWT=
_threadPrinciple.Identity.IssuedJWTToken
_jwtVali
not valid JWT Token
valid JWT
checkPermissionsIfMatch() : bool
if permissions match Returns True
else
if permissions dont match returns false
Should I check if the whole
list should match?
Or the needPermission EXIST
in the users permission set

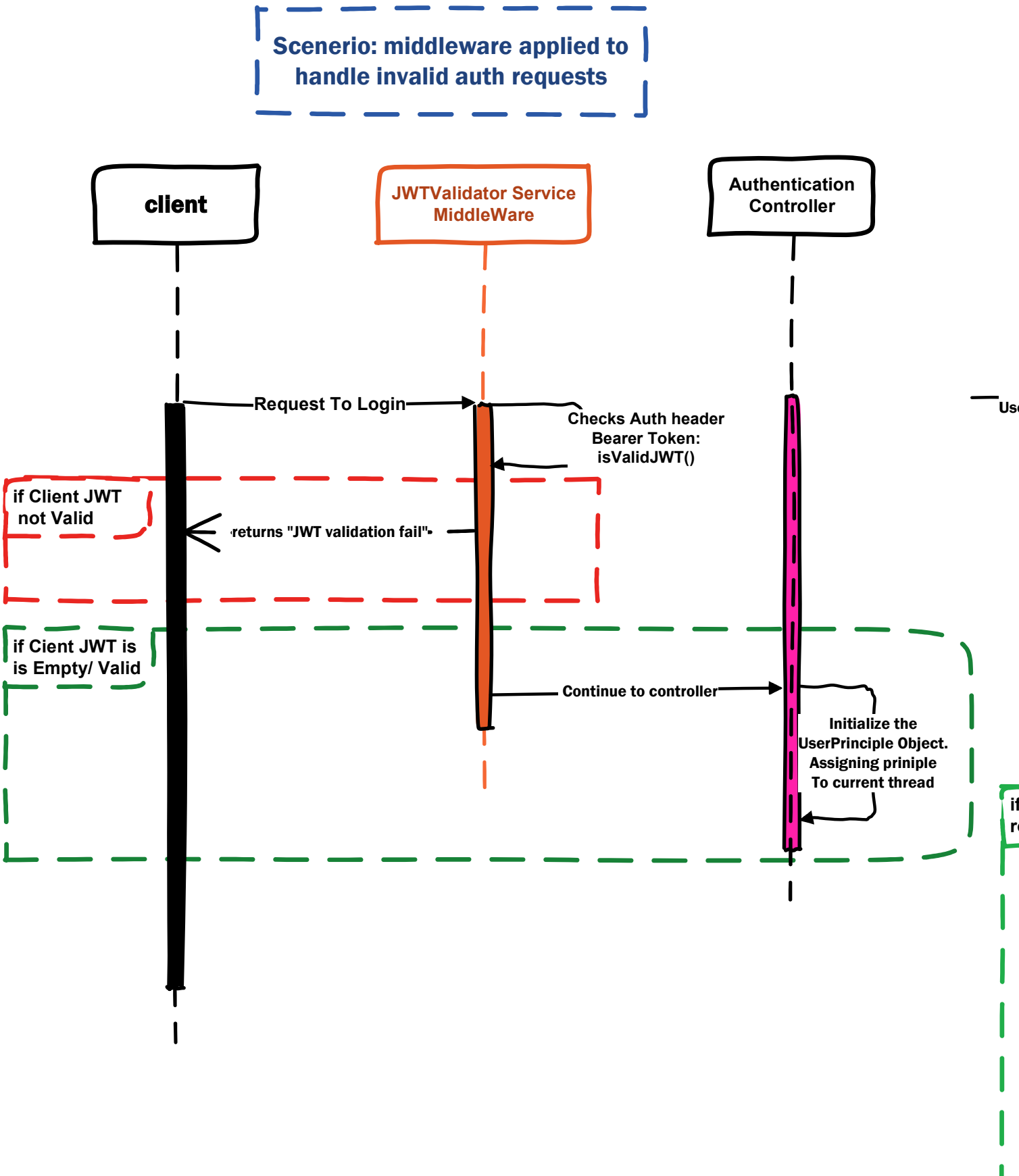

ken
_jwtValidator.IsValidJWT()
isValidJWTFormat()
isValidPayHeader()
IsValidSignature()
returns False
returns True
: bool
ole
XIST
set

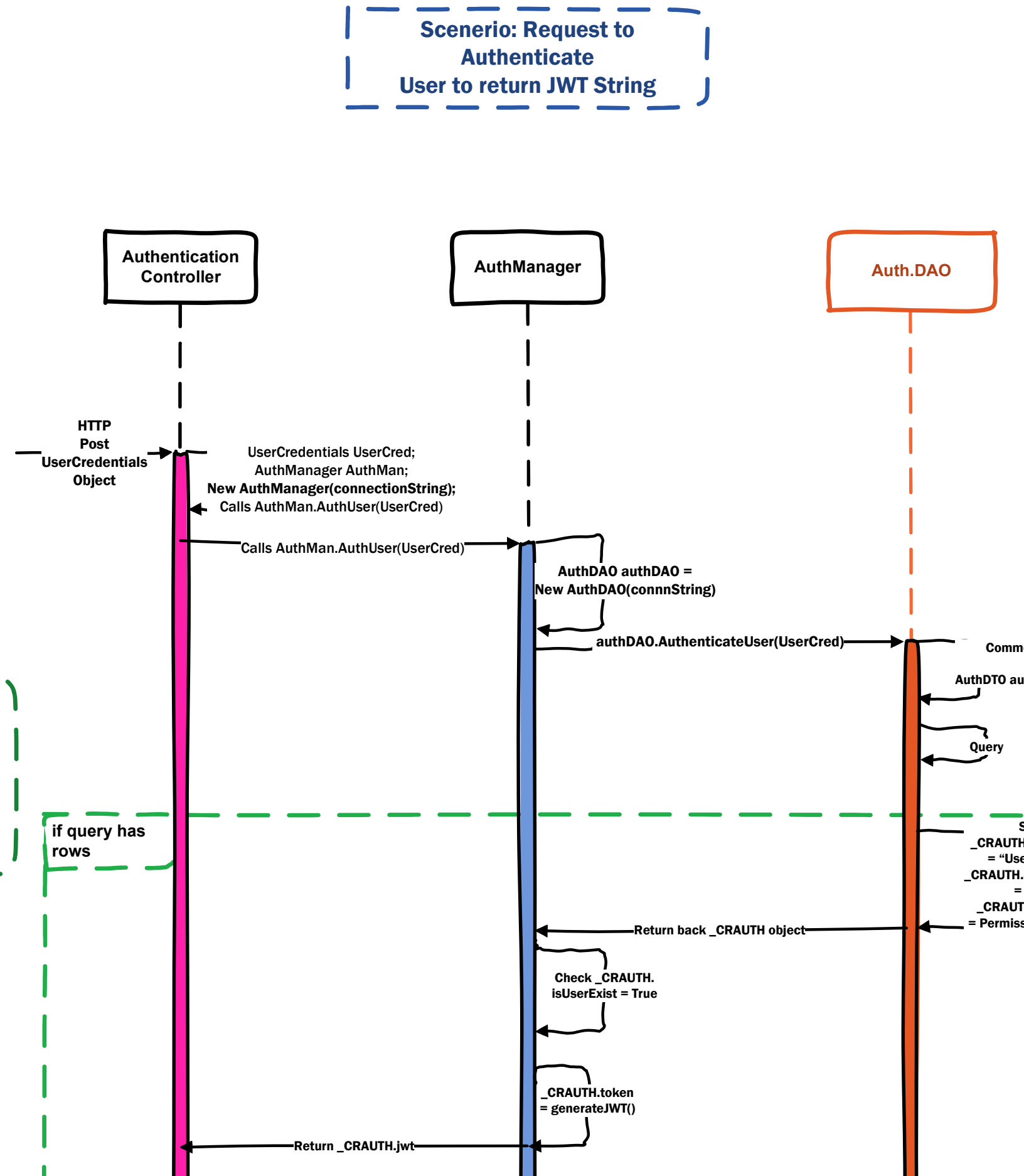

JwtHeader
JwtPayloa
Utilized by
JwtService

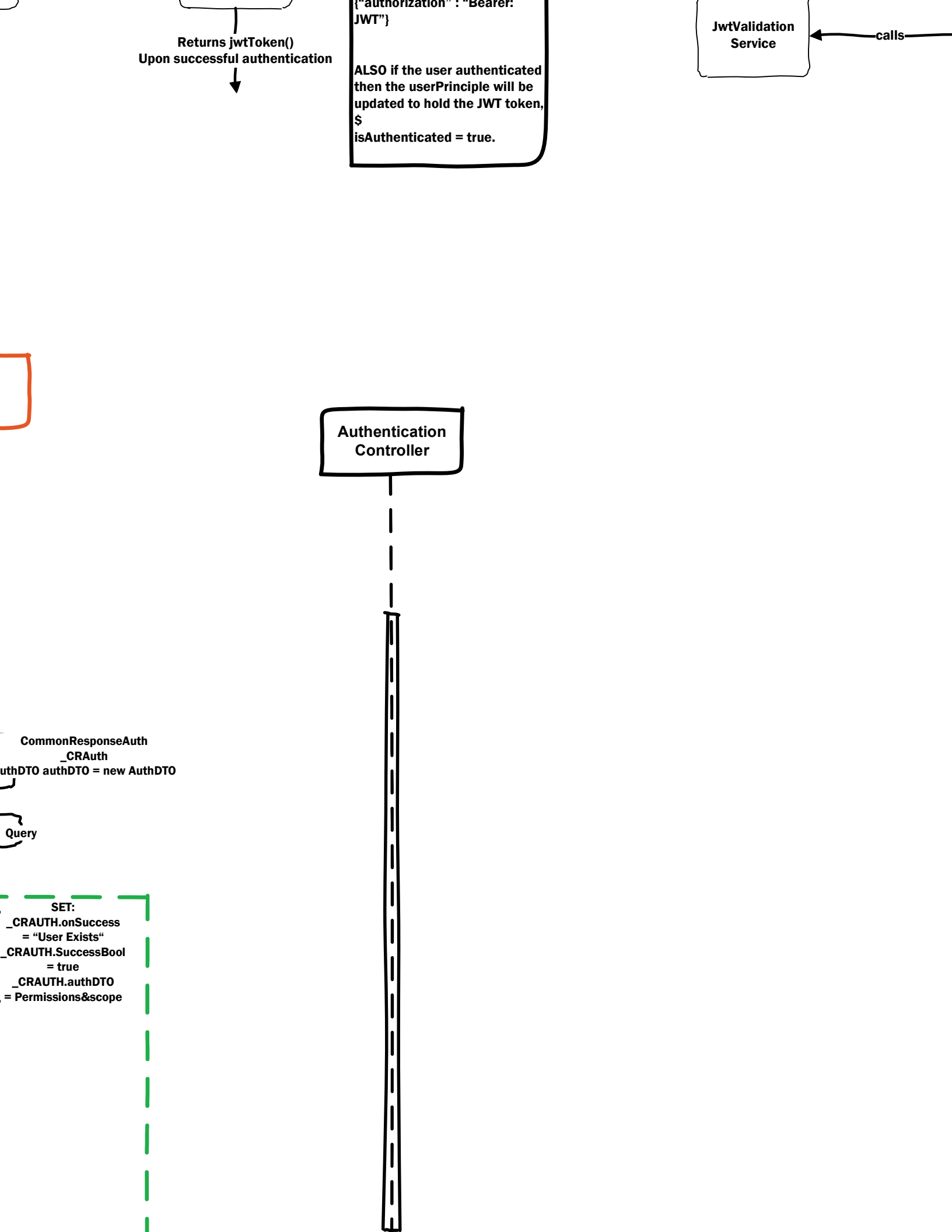

wtPayload
calls
Authentication Service
Returns jwtToken()
1. Notes go here!
upon successful Authentication of a user then the Authentication service will generate a JWT string to return to
Be stored into the http header {“authorization” : “Bearer: JWT”}
Passed th
Maybe as passing wh
To
c
JwtValidation Service
calls

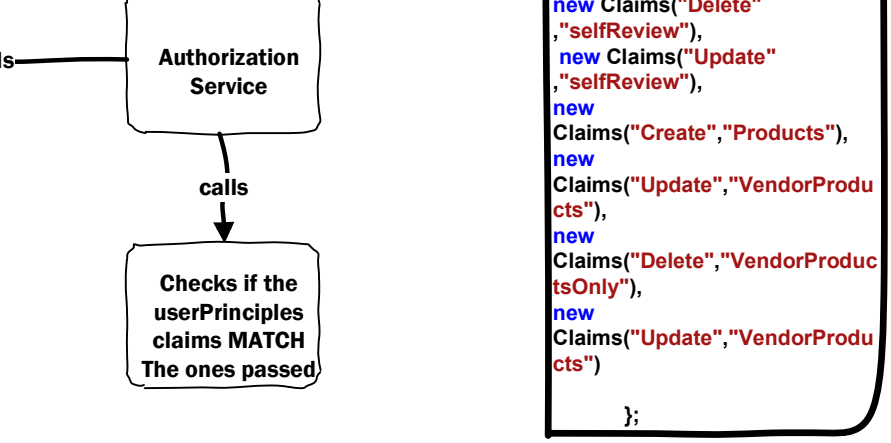


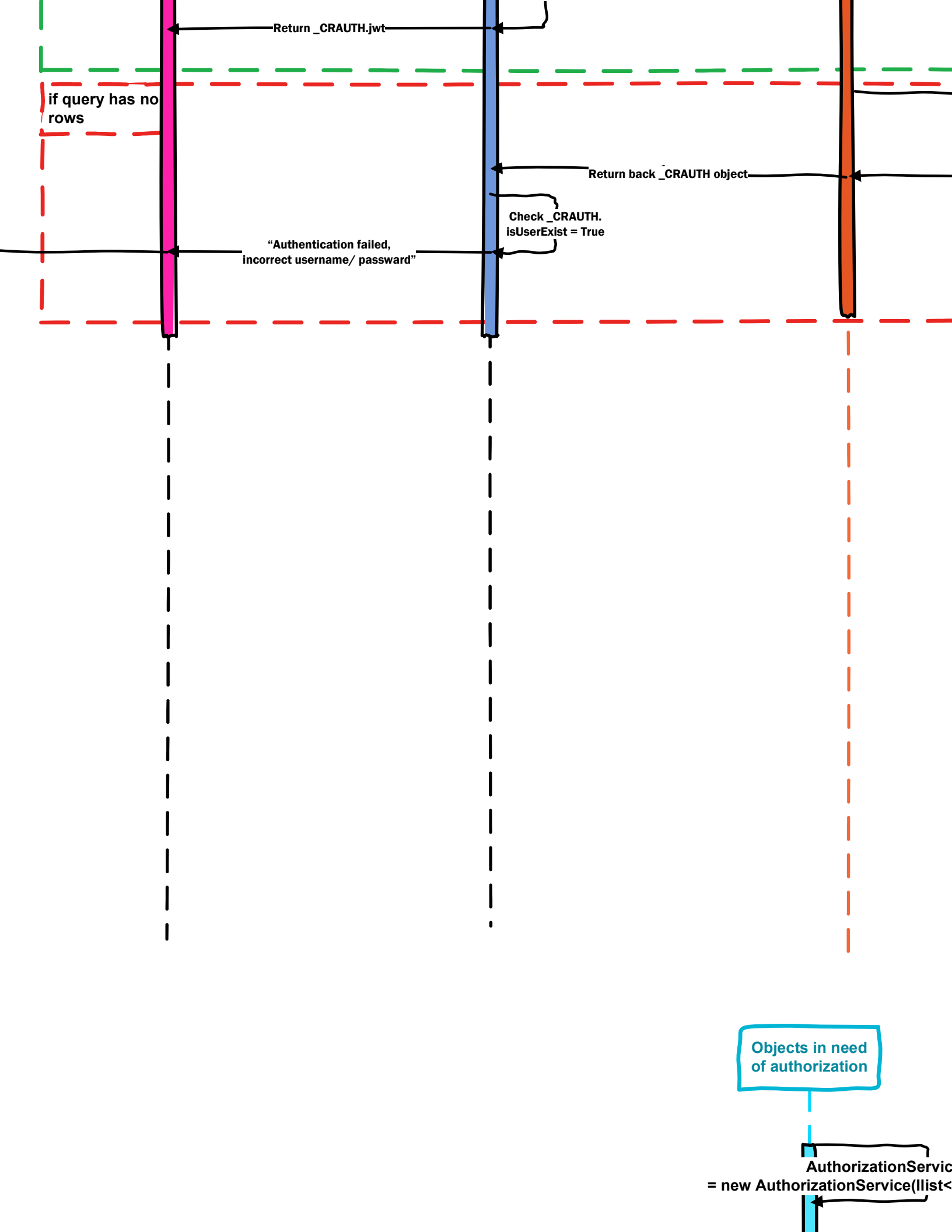

```
1.example of claims:
 IList<Claims>
 claimsPerAdmin = new
List<Claims>()
          {  new Claims("all","all"),
};


 IList<Claims>
  claimsPerVendor = new
List<Claims>()
          {
new
Claims("ReadOnly","AutoBuild"
),
new Claims("Delete","self"),
new Claims("Update","self"),
new Claims("Edit","self"),
new Claims("Create"
,"reviews"),
new Claims("Delete"
,"selfReview"),
 new Claims("Update"
,"selfReview"),
```


Scenerio: middleware applied to handle invalid auth requests
client
JWTValidator Service MiddleWare
Authentication Controller
Request To Login
Checks Auth header Bearer Token: isValidJWT()
if Client JWT not Valid
returns "JWT validation fail"
if Cient JWT is is Empty/ Valid
Continue to controller
Initialize the UserPrinciple Object. Assigning priniple To current thread


Scenerio: Request to Authenticate User to return JWT String
Authentication Controller
AuthManager
Auth.DAO
HTTP Post UserCredentials Object
UserCredentials UserCred;
AuthManager AuthMan;
New AuthManager(connectionString);
Calls AuthMan.AuthUser(UserCred)
Calls AuthMan.AuthUser(UserCred)
AuthDAO authDAO = New AuthDAO(connnString)
authDAO.AuthenticateUser(UserCred)
Comm
AuthDTO au
Query
if query has rows
Return back _CRAUTH object
Check _CRAUTH. isUserExist = True
_CRAUTH.token = generateJWT()
Return _CRAUTH.jwt


Returns jwtToken()
Upon successful authentication
{"authorization" : "Bearer: JWT"}
ALSO if the user authenticated then the userPrinciple will be updated to hold the JWT token, $
isAuthenticated = true.
JwtValidation Service
calls
Authentication Controller
CommonResponseAuth
_CRAuth
uthDTO authDTO = new AuthDTO
Query
SET:
_CRAUTH.onSuccess
= "User Exists"
_CRAUTH.SuccessBool
= true
_CRAUTH.authDTO
= Permissions&scope


ls
Authorization Service
calls
Checks if the userPrinciples claims MATCH The ones passed
new Claims("Delete" ,"selfReview"), new Claims("Update" ,"selfReview"), new Claims("Create","Products"), new Claims("Update","VendorProducts"), new Claims("Delete","VendorProductsOnly"), new Claims("Update","VendorProducts") };


Return _CRAUTH.jwt
if query has no rows
Return back _CRAUTH object
Check _CRAUTH.
isUserExist = True
“Authentication failed,
incorrect username/ passward”
Objects in need of authorization
AuthorizationServic
= new AuthorizationService(Ilist<

SET:
_CRAUTH.onSuccess
= “User Not Valid“
_CRAUTH.SuccessBool
= false
AuthDTO.UserPermissions
= permissionsFromQuery;
AuthDTO.UserName
= UserNameFromQuery;
_CRAuth.AuthDTO = AuthDTO

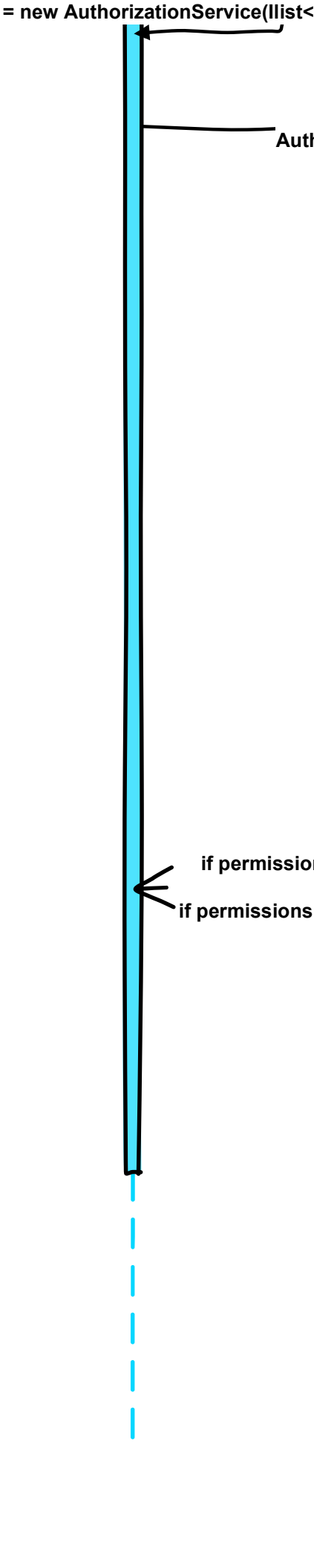
NOTES:
DO VALIDATOR CHECK BEFORE THE CONTROLLER IS CREATED ().

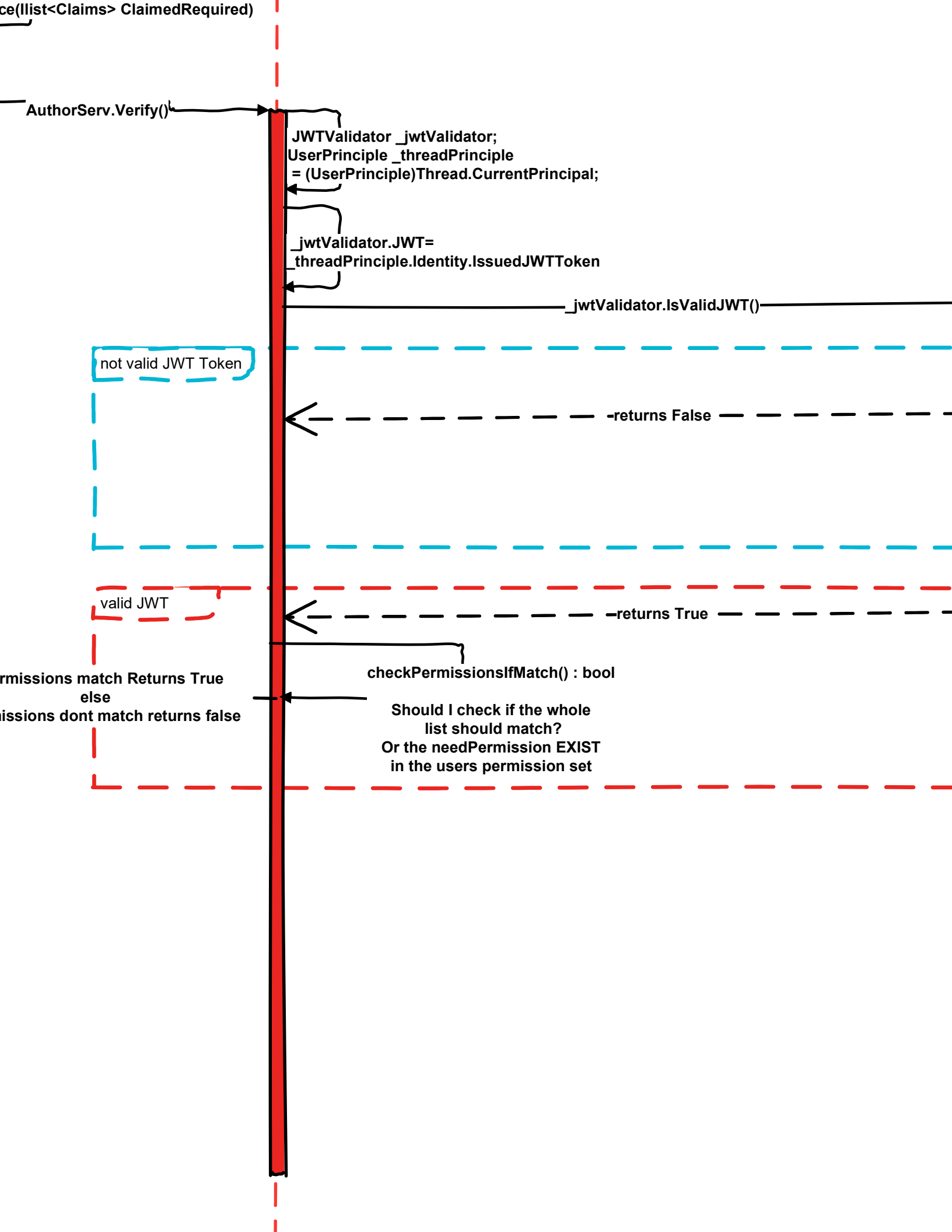
CREATING A MIDDLEWARE (HTTP PIPELINE) THE HTTP REQUESTS HIT THE PIPLINE B4 IT GOES TO THE CODE. CREATE A CUSTOM MIDDLEWARE. PROS: EARLY EXIT IN CASE OF JWTVALIDATION FAIL.
EVERY REQUEST-RESPONSE HAS COME PIPLINE B4 REACHING YOU CODE. GOAL: KILL A REQUEST EARLY BEFORE REACHING ENDING STAGES.
WHY IN OUR CASE: SINCE WE ARE CHECKING ALL THE TIME FOR JWT VALID.

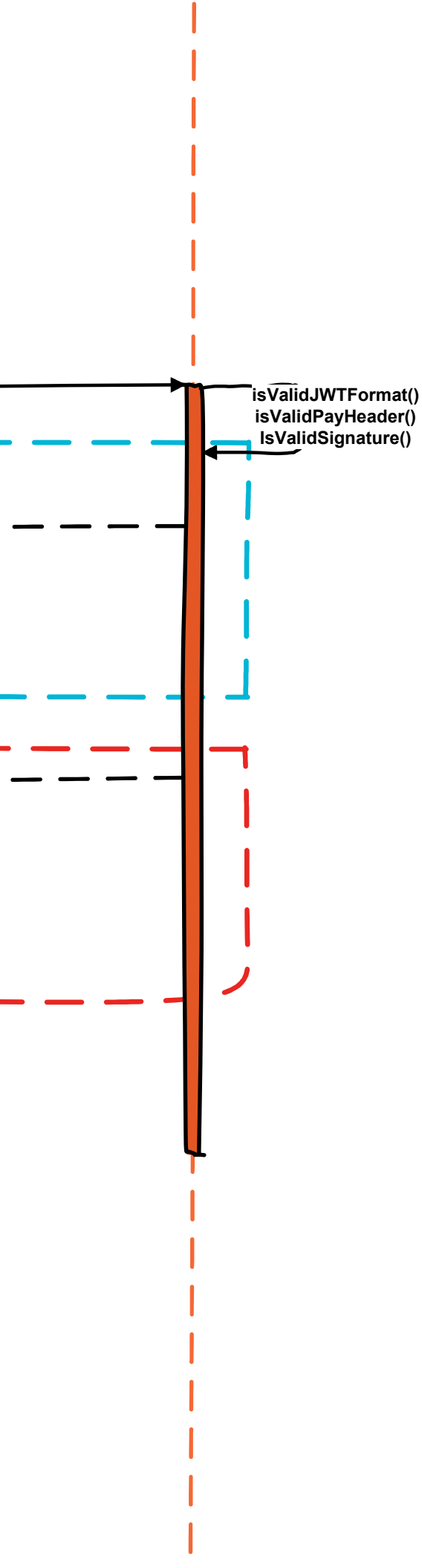
Authorization Service

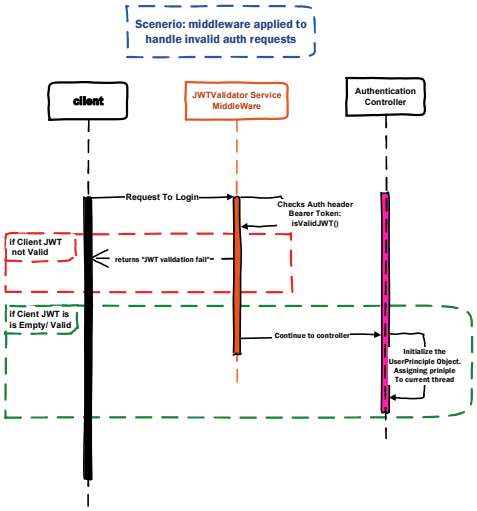
nService AuthorServ
ce(Ilist<Claims> ClaimedRequired)

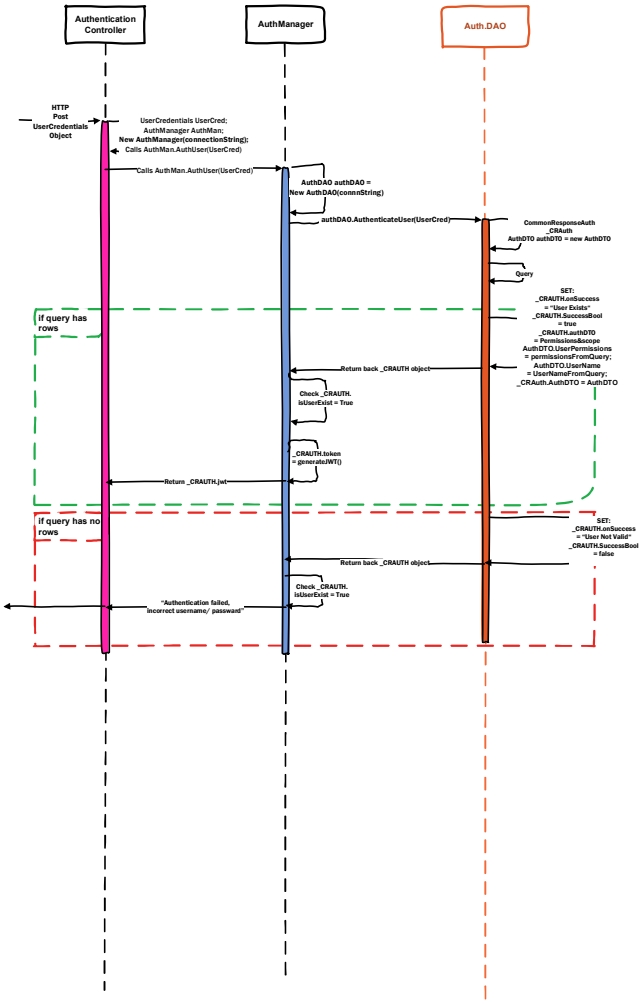

JWTValidator Service


= new AuthorizationService(Ilist<
Auth
if permissio
if permissions


ce(Ilist<Claims> ClaimedRequired)
AuthorServ.Verify()
JWTValidator _jwtValidator;
UserPrinciple _threadPrinciple
= (UserPrinciple)Thread.CurrentPrincipal;
_jwtValidator.JWT=
_threadPrinciple.Identity.IssuedJWTToken
_jwtValidator.IsValidJWT()
not valid JWT Token
returns False
valid JWT
returns True
checkPermissionsIfMatch() : bool
rmissions match Returns True
else
issions dont match returns false
Should I check if the whole
list should match?
Or the needPermission EXIST
in the users permission set


isValidJWTFormat()
isValidPayHeader()
IsValidSignature()


Scenerio: middleware applied to handle invalid auth requests
client
JWTValidator Service MiddleWare
Authentication Controller
Request To Login
Checks Auth header Bearer Token: isValidJWT()
if Client JWT not Valid
returns "JWT validation fail"
if Client JWT is is Empty/ Valid
Continue to controller
Initialize the UserPrinciple Object. Assigning priniple To current thread


Scenerio: Request to Authenticate User to return JWT String
Authentication Controller
AuthManager
Auth.DAO
HTTP Post UserCredentials Object
UserCredentials UserCred;
AuthManager AuthMan;
New AuthManager(connectionString);
Calls AuthMan.AuthUser(UserCred)
Calls AuthMan.AuthUser(UserCred)
AuthDAO authDAO = New AuthDAO(connnString)
authDAO.AuthenticateUser(UserCred)
CommonResponseAuth _CRAuth
AuthDTO authDTO = new AuthDTO
Query
SET: _CRAUTH.onSuccess = "User Exists" _CRAUTH.SuccessBool = true
_CRAUTH.AuthDTO = Permissions&scope AuthDTO.UserPermissions = permissionsFromQuery; AuthDTO.UserName = UserNameFromQuery; _CRAuth.AuthDTO = AuthDTO
if query has rows
Return back _CRAUTH object
Check _CRAUTH. isUserExist = True
_CRAUTH.token = generateJWT()
Return _CRAUTH.jwt
if query has no rows
SET: _CRAUTH.onSuccess = "User Not Valid" _CRAUTH.SuccessBool = false
Return back _CRAUTH object
Check _CRAUTH. isUserExist = True
"Authentication failed, incorrect username/ passwand"