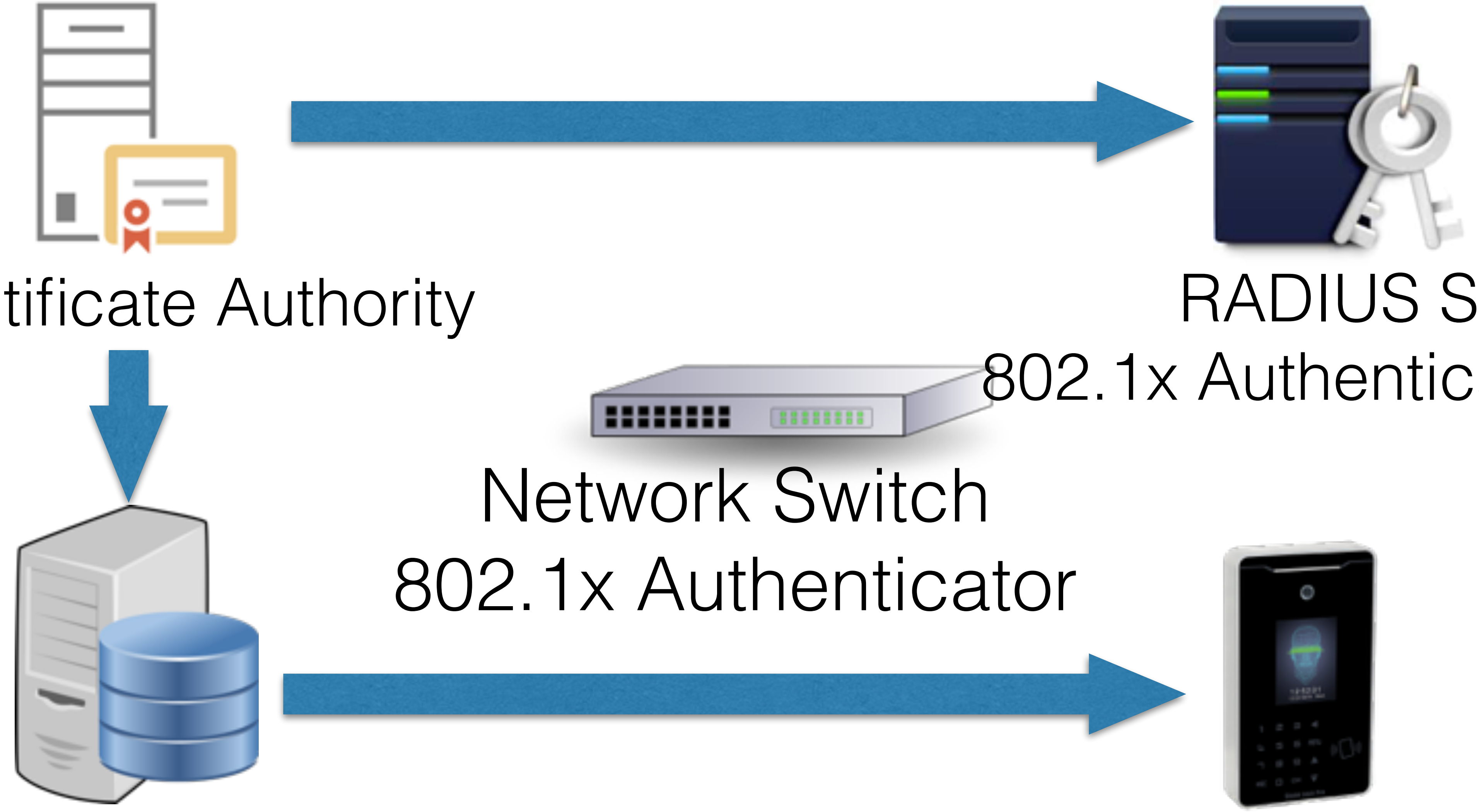
TLS/802.1x Credential Workflow

Key Components



Certificate Authority

RADIUS Server
802.1x Authentication Server

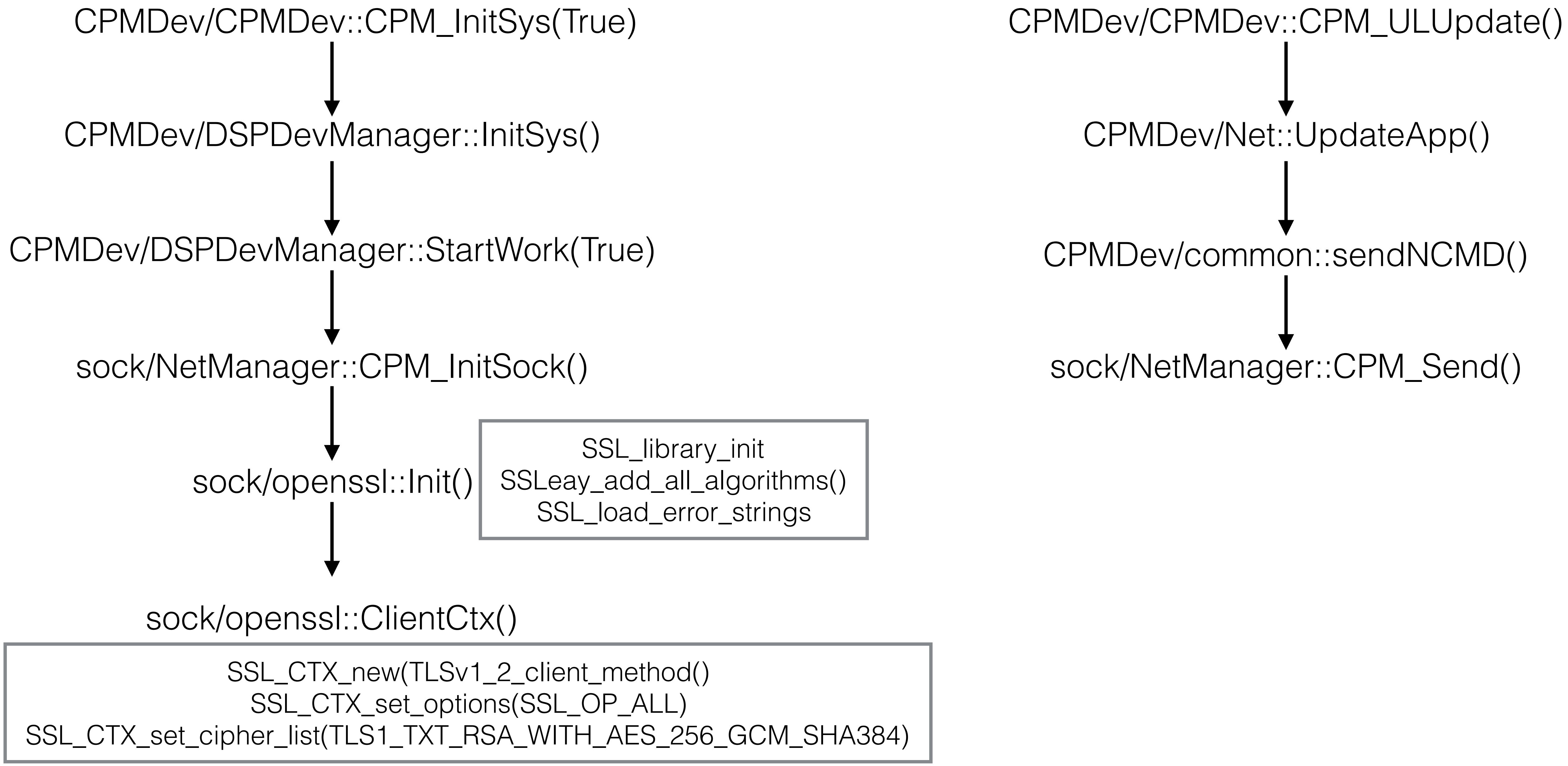
Network Switch
802.1x Authenticator

Access Control Server
TLS Client
802.1x Supplicant

Reader
TLS Server
802.1x Supplicant

1. The Access Control Server (ACS) generates a key pair (public/private) using openssl
2. The ACS requests an X.509 certificate from the Certificate Authority (CA) for the reader (device)
3. The CA generates a certificate based on the Fully Qualified Domain Name of the reader
4. The name of the certificate will be the FQDN of the reader
5. The ACS stores this certificate in the directory ‘c:\\rootca'
6. The ACS packages the X.509 certificate and the key pair into a PKCS 12 file
7. The ACS reads the PKCS 12 file as binary data
8. The ACS calls the CPMDev library method CPM_ULUpdate to transfer the certificate binary contents to the reader
9. The CPM_ULUpdate method includes a password for authentication by the server.
10. The CPMDev library completes a network transfer of the byte array and password through port 13333
11. The reader authenticates the CPM_ULUpdate message with the transferred password
12. Note: verify password network format and the ability to change the password if compromised
13. if the password is authenticated, the reader saves the PKCS 12 byte array as a file accessible by openssl
14. The reader openssl library will verify the certificate and read the PKCS 12 keys
15. The reader will initialize the TLS/TCP server with the read keys as the public key
16. The ACS calls the CPMDev library method CPM_CNDev to initialize the connection
17. The CPMDev will callback the status of the connection to a function defined in the CPM_RegDevConnectionStatusCB method
18. Before proceeding with the client TLS request the CPMDev openssl library will verify the reader’s certificate in ‘c:\\rootca'
19. If the certificate is expired, the CPMDev will send a callback certificate failure message and close the connection
20. If the certificate is expired, the ACS will request a new certificate, generate new keys and restart the process
21. Note: openssl will need to periodically check the certificate while the connection is open
22. The CPMDev library TLS client will make a “client hello” call to the reader
23. The reader’s TLS server will reply with a “server hello” including the certificate and public key received in the PKCS 12 file
24. The TLS client uses the server’s public key to encrypt the random byte string to derive the message keys
25. The TLS client sends the secret key information to the server
26. The server and client exchange ‘finished’ messages
27. Bi-directional message transmissions start with the new shared secret key

Client Library Call Stack



CPMDev/CPMDev::CPM_InitSys(True)

CPMDev/DSPDevManager::InitSys()

CPMDev/DSPDevManager::StartWork(True)

sock/NetManager::CPM_InitSock()

sock/openssl::Init()

SSL_library_init
SSLeay_add_all_algorithms()
SSL_load_error_strings

sock/openssl::ClientCtx()

SSL_CTX_new(TLSv1_2_client_method()
SSL_CTX_set_options(SSL_OP_ALL)
SSL_CTX_set_cipher_list(TLS1_TXT_RSA_WITH_AES_256_GCM_SHA384)

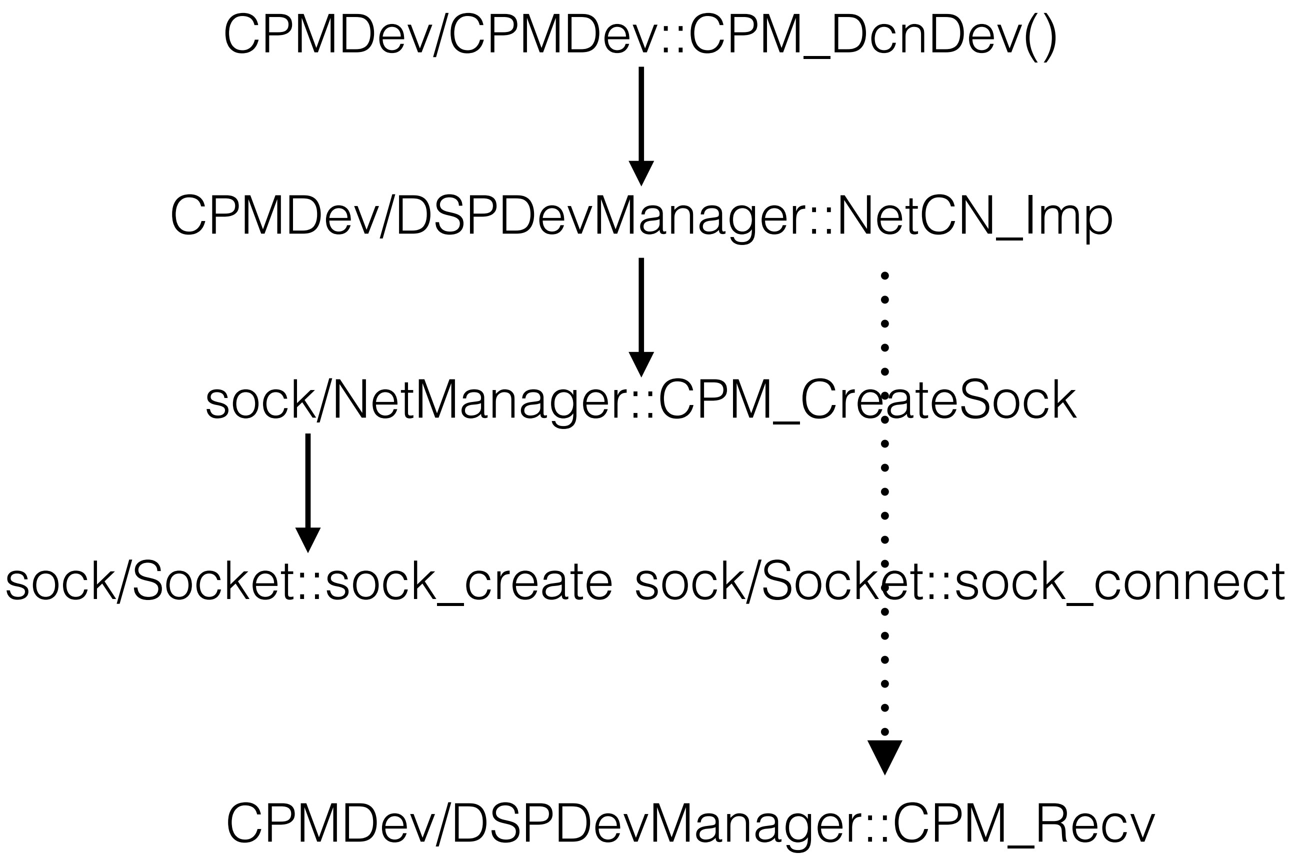
CPMDev/CPMDev::CPM_ULUpdate()

CPMDev/Net::UpdateApp()

CPMDev/common::sendNCMD()

sock/NetManager::CPM_Send()

Client Call Stack Cont’d



CPMDev/CPMDev::CPM_DcnDev()

CPMDev/DSPDevManager::NetCN_Imp

sock/NetManager::CPM_CreateSock

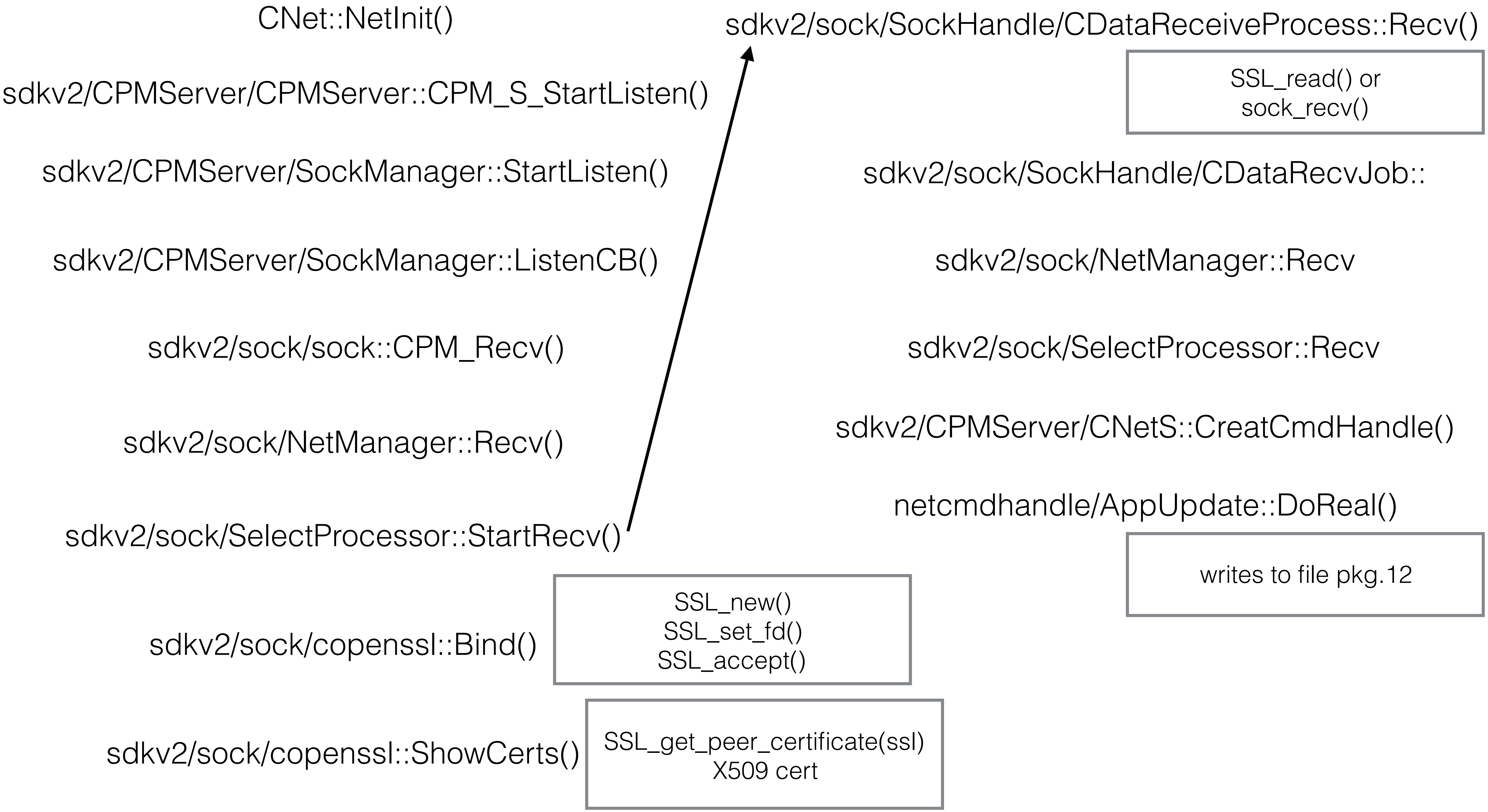
sock/Socket::sock_create sock/Socket::sock_connect

CPMDev/DSPDevManager::CPM_Recv

SSL_new(ctx)
SSL_set_fd(ssl,server)
SSL_connect(ssl)

I cannot find these openssl calls!

Server Call Stack



CNet::NetInit()

sdkv2/CPMServer/CPMServer::CPM_S_StartListen()

sdkv2/CPMServer/SockManager::StartListen()

sdkv2/CPMServer/SockManager::ListenCB()

sdkv2/sock/sock::CPM_Recv()

sdkv2/sock/NetManager::Recv()

sdkv2/sock/SelectProcessor::StartRecv()

sdkv2/sock/copenssl::Bind()

SSL_new()
SSL_set_fd()
SSL_accept()

sdkv2/sock/copenssl::ShowCerts()

SSL_get_peer_certificate(ssl)
X509 cert

sdkv2/sock/SockHandle/CDataReceiveProcess::Recv()

SSL_read() or
sock_recv()

sdkv2/sock/SockHandle/CDataRecvJob::

sdkv2/sock/NetManager::Recv

sdkv2/sock/SelectProcessor::Recv

sdkv2/CPMServer/CNetS::CreatCmdHandle()

netcmdhandle/AppUpdate::DoReal()

writes to file pkg.12

Credentialing Steps 802.1x