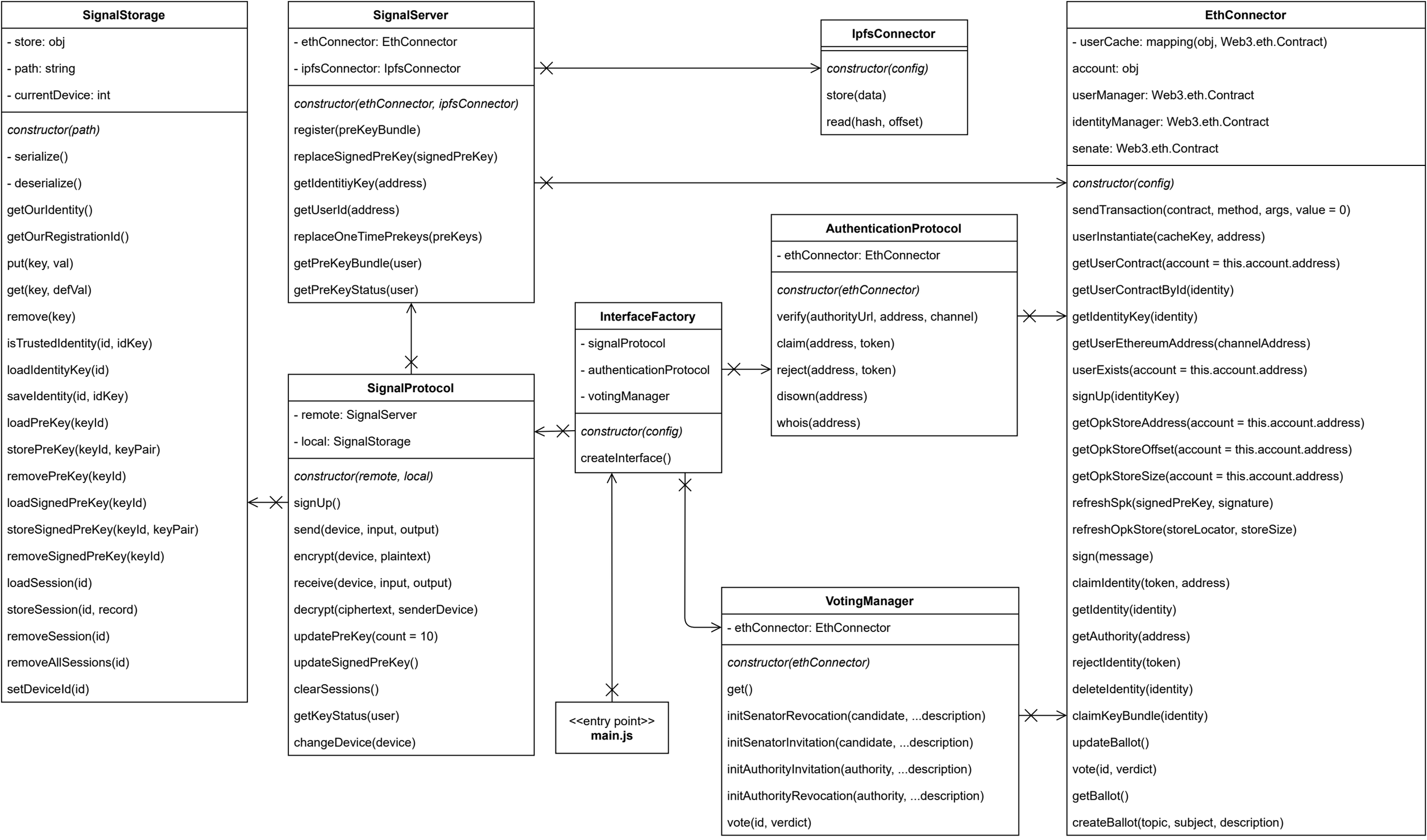
SignalStorage
- store: obj
- path: string
- currentDevice: int
constructor(path)
- serialize()
- deserialize()
getOurIdentity()
getOurRegistrationId()
put(key, val)
get(key, defVal)
remove(key)
isTrustedIdentity(id, idKey)
loadIdentityKey(id)
saveIdentity(id, idKey)
loadPreKey(keyId)
storePreKey(keyId, keyPair)
removePreKey(keyId)
loadSignedPreKey(keyId)
storeSignedPreKey(keyId, keyPair)
removeSignedPreKey(keyId)
loadSession(id)
storeSession(id, record)
removeSession(id)
removeAllSessions(id)
setDeviceId(id)
SignalServer
- ethConnector: EthConnector
- ipfsConnector: IpfsConnector
constructor(ethConnector, ipfsConnector)
register(preKeyBundle)
replaceSignedPreKey(signedPreKey)
getIdentitiyKey(address)
getUserId(address)
replaceOneTimePrekeys(preKeys)
getPreKeyBundle(user)
getPreKeyStatus(user)
IpfsConnector
constructor(config)
store(data)
read(hash, offset)
EthConnector
- userCache: mapping(obj, Web3.eth.Contract)
account: obj
userManager: Web3.eth.Contract
identityManager: Web3.eth.Contract
senate: Web3.eth.Contract
constructor(config)
sendTransaction(contract, method, args, value = 0)
userInstantiate(cacheKey, address)
getUserContract(account = this.account.address)
getUserContractById(identity)
getIdentityKey(identity)
getUserEthereumAddress(channelAddress)
userExists(account = this.account.address)
signUp(identityKey)
getOpkStoreAddress(account = this.account.address)
getOpkStoreOffset(account = this.account.address)
getOpkStoreSize(account = this.account.address)
refreshSpk(signedPreKey, signature)
refreshOpkStore(storeLocator, storeSize)
sign(message)
claimIdentity(token, address)
getIdentity(identity)
getAuthority(address)
rejectIdentity(token)
deleteIdentity(identity)
claimKeyBundle(identity)
updateBallot()
vote(id, verdict)
getBallot()
createBallot(topic, subject, description)
AuthenticationProtocol
- ethConnector: EthConnector
constructor(ethConnector)
verify(authorityUrl, address, channel)
claim(address, token)
reject(address, token)
disown(address)
whois(address)
InterfaceFactory
- signalProtocol
- authenticationProtocol
- votingManager
constructor(config)
createInterface()
SignalProtocol
- remote: SignalServer
- local: SignalStorage
constructor(remote, local)
signUp()
send(device, input, output)
encrypt(device, plaintext)
receive(device, input, output)
decrypt(ciphertext, senderDevice)
updatePreKey(count = 10)
updateSignedPreKey()
clearSessions()
getKeyStatus(user)
changeDevice(device)
<<entry point>>
main.js
VotingManager
- ethConnector: EthConnector
constructor(ethConnector)
get()
initSenatorRevocation(candidate, ...description)
initSenatorInvitation(candidate, ...description)
initAuthorityInvitation(authority, ...description)
initAuthorityRevocation(authority, ...description)
vote(id, verdict)