IMPORTANT NOTICE:

NCC API document is moved to the Open Source Project of NCC.
https://github.com/NewEconomyMovement/NemCommunityClient/blob/master/api.md

Following specification is OUTDATED as of 2014-08-24.

Team

	Time Zone	Dev Skills
Thies1965	UTC+01:00	Java
LoiTran4		Web Design, UX Design, Graphic Designer
Krysto		Web Design, HTML, JavaScript
BloodyRookie	UTC+01:00	Java

NCC Core Team

High Level Specification

Overview and Scope
The NEM community client (NCC) will be developed as a servlet based application which provides a REST Api using JSON.
NCC connects to a NIS (NEM infrastructure server) in order to get access to the NEM cloud (please NEM development plan for further details on the NEM API).
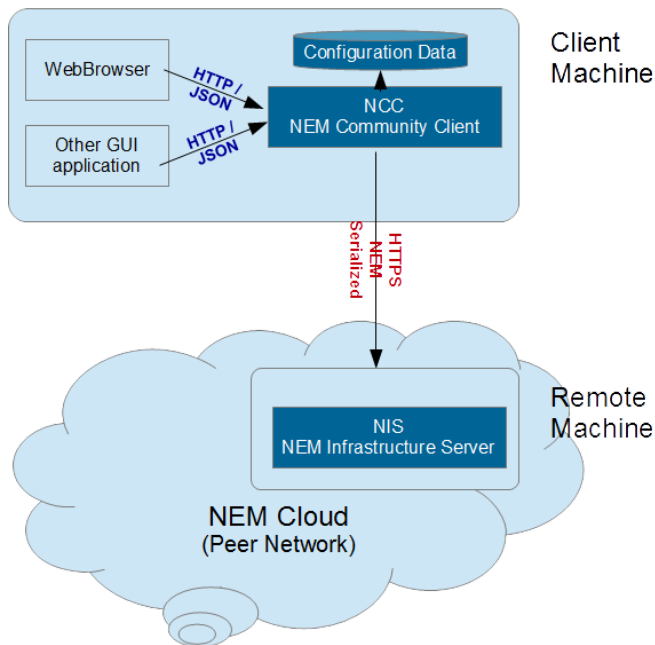The NIS may reside locally or on a remote machine.

Operational Model I
Client and Node on same machine
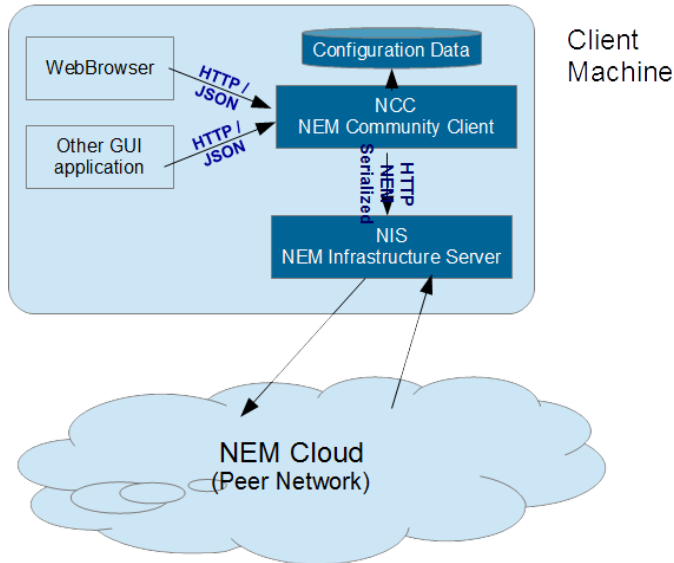
Operational Model: Local NIS

Operational Model II
Client and Node on different machines

Operational Model: NCC access remote NIS

NCC will be split into three components, one provides the REST Api, the other component includes the Web GUI as HTML, JavaScript, and corresponding JavaScript libraries, and the last very small component bootstraps the Jetty Servlet container.

The NCC will be deployed via Java WebStart functionality (including also a NIS via lazy loading). NCC will leverage the JavaWebStart API to pop-up the WebBrowser.

Component overview

The NCC will be comprised out of three components: WebStarter, API, and Web.

NCC component model

WebStarter

The WebStarter component has the responsibility to enable JavaWebStart to start the Java application. Starting the application means in this context to configure and start Jetty as an embedded Servlet Container.
This WebStarter also has to handle the situation that only one NCC client can be started. In case of Port conflicts, WebStarter checks whether the other server response as being an NCC client. In case this is not true, WebStart tries with a different port number.
When finally the server is started, the WebBrowser of the platform is started (via WebStart API). The link provided to the Browser points to the NCC server (and port).

Web

This component is responsible to provide the HTML, JavaScript and other static content to the WebBrowser. It will be packaged as a Jar file in order to be deployable via Java WebStart. It contains also a simple file serving servlet that is Java WebStart enabled.

The component provides the look and feel of the client. The access to the wallet and the functionality will be realized utilizing JavaScript and appropriate JavaScript frameworks for GUI widgets or for Model-View-ViewModels.

A first UI sketch is drafted here.

The newest design is available here.
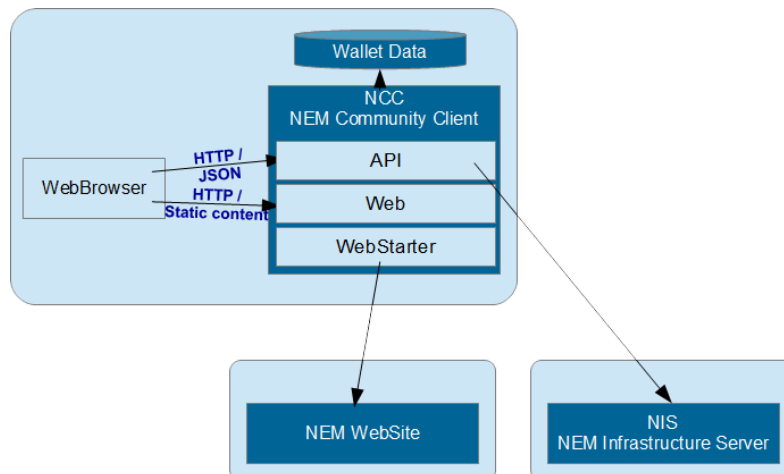
API

The component API will process the requests from the WebBrowser and hold the connection to the NEM cloud. To fullfill this task, NCC utilizes the NIS API.

API will be realized as a JAR file which will be provided to the embedded Jetty server. The JAR file is required to be startable via Java WebStart.
The physical deployment unit covers both logical components: WebStarter and API.

The following API list is not yet complete and will likely be changed during implementation. The web-context is defined as "/ncc/api" (full URI would be, for instance, "/ncc/api/client/info")

Amount values are all in the smallest possible NEM fraction, that means that 1000000 means 1.000000 NEM

- /info
    - GET
    - Input: -
    - Output: **NCCInfo**
    - NIS Dependent: **yes**
      in case NIS is not available, then the section of NIS in NCCInfo is empty.
- /info/ncc
    - GET
    - Input: -
    - Output: **NCCInfo**
    - NIS Dependent: **no**
- /info/nis
    - GET
    - Input: -
    - Output: **NCCInfo**
    - NIS Dependent: **yes**
- /info/nis/check
    - GET
    - Input: -
    - Output: "1" if NIS is running, "0" otherwise
    - NIS Dependent: **no**
- /shutdown
    - GET
    - Input: -
    - Output: -
    - NIS Dependent: **no**
- /configuration/get
    - GET

- ○ Input: -
- ○ Output: **ConfigurationExternal**
- ○ NIS Dependent: **no**
- **/configuration/update**
  - ○ POST
  - ○ Input: new configuration like ConfiguartionExternal
  - ○ Output: success / failure
  - ○ NIS Dependent: **no**
- **/wallet/create**
  - ○ POST
  - ○ Input: {wallet} name of wallet, {password}
    - ■ Example:
      {"wallet" : "PrivateWallet", "password" : "The very long password with some ÄÖÜäöüß"}
  - ○ Output: **WalletExternal** - Wallet Information
  - ○ NIS Dependent: **no**
- **/wallet/export** (formerly /wallet/backup)
  - ○ POST
  - ○ Input: {wallet}, {password}, {export_path}
    - ■ Example:
      {"wallet" : "PrivateWallet", "export_path" : "c:\\temp", "password" : "The very long password with some ÄÖÜäöüß"}
  - ○ Output: Success / failure
    - ■ Example:
      {"ok":42}
  - ○ NIS Dependent: **no**
- **/wallet/file/export**
  - ○ POST
  - ○ Input: {wallet}, {password}, {export_path}
    - ■ Example:
      {"wallet" : "PrivateWallet"}
  - ○ Output: Binary stream from wallet file.
  - ○ NIS Dependent: **no**
- **/wallet/import** (formerly /wallet/restore)
  - ○ POST
  - ○ Input: {import_path}, {password}
    - ■ Example:
      {"import_path" : "c:\\temp\\PrivateWallet.wlt", "password" : "The very long password with some ÄÖÜäöüß"}
  - ○ Output: **WalletExternal** - Wallet Information
  - ○ NIS Dependent: **no**
- **/wallet/file/import**
  - ○ POST

- ○ Input: Multipart: {password} and {file-data}
  - ■ Example:
    Content-Type: multipart/mixed

    --edt7Tfrdusa7r3lNQc79vXuhIIMlatb7PQg7Vp
    Content-Disposition: form-data; name="password"
    Content-Type: application/json; charset=UTF-8
    Content-Transfer-Encoding: 8bit

    {
        "password": "The very long password with some ÄÖÜäöüß"
    }
    --edt7Tfrdusa7r3lNQc79vXuhIIMlatb7PQg7Vp
    Content-Disposition: form-data; name="file-data"; filename="mywallet.wlt"
    ... File Data ...
- ○ Output: **WalletExternal** - Wallet Information
- ○ NIS Dependent: **no**
- ● /wallet/mobilize *(phase 2)*
  - ○ Input: {passphrase}
  - ○ Output: QR-Code
- ● /wallet/list
  - ○ GET
  - ○ Input: -
  - ○ Output: List of wallet files found
    - ■ Example:
      {
          "wallets": [{
                  "wallet": "öäü@",
                  "location":
      "C:\Users\Thies1965\git\nem-main\ncc\nem-client-api\%C3%B6%C3%A4%C3%BC%40.wlt"
          },
          {
                  "wallet": "Hello world",
                  "location":
      "C:\Users\Thies1965\git\nem-main\ncc\nem-client-api\Hello+world.wlt"
          }
      }
  - ○ NIS Dependent: **no**
- ● /wallet/open
  - ○ POST
  - ○ Input: {wallet} of the wallet, {password}
    - ■ Example:

{"wallet" : "PrivateWallet", "password" : "The very long password with some ÄÖÜäöüß"}

- ○ Output: **WalletExternal** - Wallet Information
- ○ NIS Dependent: **no**
- ● /wallet/close
    - ○ POST
    - ○ Input: {wallet} name of the wallet to be closed
        - ■ Example:
          {"wallet" : "PrivateWallet"}
    - ○ Output: success / failure
        - ■ Example:
          {"ok":42}
    - ○ NIS Dependent: **no**
- ● /wallet/merge
    - ○ POST
    - ○ Input: {wallet},{password} of the wallet to be merged into, {import_path},{merge_password} wallet to be merged
        - ■ Example:
          {"import_path" : "c:\\temp\\PrivateWallet.wlt", "wallet" : "PrivateWallet", "password" : "The very long password with some ÄÖÜäöüß", "merge_password" , "Password of the to-be-merged-wallet"}
    - ○ Output: **WalletExternal** - Wallet Information
    - ○ NIS Dependent: **no**
- ● /wallet/file/merge
    - ○ POST
    - ○ Input: Multipart: Part: parameter {wallet} {password} {merge_password} and Part: file-data
        - ■ Example:
          Content-Type: multipart/mixed

          --edt7Tfrdusa7r3lNQc79vXuhIIMlatb7PQg7Vp
          Content-Disposition: form-data; name="parameter"
          Content-Type: application/json; charset=UTF-8
          Content-Transfer-Encoding: 8bit

          {"wallet" : "PrivateWallet", "password" : "The very long password with some ÄÖÜäöüß", "merge_password" , "Password of the to-be-merged-wallet"}
          --edt7Tfrdusa7r3lNQc79vXuhIIMlatb7PQg7Vp
          Content-Disposition: form-data; name="file-data"; filename="mywallet.wlt"
          ... File Data ...
    - ○ Output: **WalletExternal** - Wallet Information
    - ○ NIS Dependent: **no**

- **/wallet/info**
  - POST
  - Input: {wallet} name of wallet
    - Example:
      {"wallet" : "PrivateWallet"}
  - Output: **WalletExternal** - Wallet Information
  - NIS Dependent: **no**
- **/wallet/password/change**
  - POST
  - Input: {wallet}, {password}, {new_password}
    - Example:
      {"wallet" : "PrivateWallet", "password" : "Test", "new_password" : "Test2"}
  - Output: **success/failure**
    - Example:
      {"ok":42}
  - NIS Dependent: **no**

- **/wallet/account/info**
  - POST
  - Input: {account}, {wallet}
  - Output: **AccountExternalDetailed -** Data of the account like balance, transaction list (only the newest one, see /wallet/account/transactions)
  - NIS Dependent: **yes**
- **/wallet/account/add**
  - POST
  - Input: {WalletName}, {password}, {Private Key}
    - Example:
      {"wallet" : "Test", "password" : "Test" , "account_key" : "9423482400235950272048987635396736428157179442700870184629 4565448981507901894"}
  - Output: **WalletExternal** - Wallet Information
  - NIS Dependent: **no**
- **/wallet/account/new**
  - Input: {WalletName}, {password}, {label}
    - Example:
      {"wallet" : "Test", "password" : "Test" , "label" : "my public account label "}
  - Output: **WalletExternal** - Wallet Information
  - NIS Dependent: **no**
- **/wallet/account/primary**
  - Input: {WalletName}, {password}, {account}
    - Example:
      {"wallet" : "Test", "password" : "Test" , "account" : "TCE7WBBHFUAOJRPTMQM3RYQCD6CJN4DLXN6UPQAU"}

- ○ Output: **WalletExternal** - Wallet Information
- ● /wallet/account/label
  - ○ Input: {WalletName}, {password}, {AccountID}, {label}
    - ■ Example:
      {"wallet" : "Test", "password" : "Test" , "account" : "TDTUF53QEEQT4HLZW7T4ECDPKCDBW6IVPHYHYCZT", "label" : "my public account label "}
  - ○ Output: **AccountExternal** - Account Information
  - ○ NIS Dependent: **no**
- ● /wallet/account/remove
  - ○ Input: {WalletName}, {password}, {AccountID}
    - ■ Example:
      {"wallet" : "Test", "password" : "Test" , "account" : "TDTUF53QEEQT4HLZW7T4ECDPKCDBW6IVPHYHYCZT"}
  - ○ Output: **WalletExternal** - Wallet Information
  - ○ NIS Dependent: **no**

- ● /wallet/name/change
  - ○ Input: {WalletName}, {password}, {AccountID}
    - ■ Example:
      {"wallet" : "Test", "password" : "Test" , "new_name" : "My new wallet"}
  - ○ Output: **WalletExternal** - Wallet Information
  - ○ NIS Dependent: **no**

- ● /wallet/account/forage (removed, see /wallet/account/unlock)
  - ○ Input: {AccountID}
  - ○ Output: success / failure
    - ■ Example:
      {"ok":42}
  - ○ NIS Dependent: **yes**
- ● /wallet/account/message/list (removed, part of account info)
  - ○ Input: {AccountID}
  - ○ Output: Messages
- ● /wallet/account/message/send (removed, part of transaction)
  - ○ Input: {AccountID}, {passphrase}, {message}, {recipient}
  - ○ Output:
- ● /wallet/account/asset/list
  - ○ Input: {AccountID}
  - ○ Output: Assets
- ● /wallet/account/asset/create
  - ○ Input: {AccountID}, {passphrase}, {message}
  - ○ Output:
- ● /wallet/account/transaction/send

- ○ POST
- ○ Input: {WalletName}, {Account}, {RecipientAccount}, {message}, {password}, {fee}, {amount} {encrypt}*
    - ■ Example:
      {"wallet" : "Test", "password" : "Test" , "account" : "TCE7WBBHFUAOJRPTMQM3RYQCD6CJN4DLXN6UPQAU", "recipient" : "TDTUF53QEEQT4HLZW7T4ECDPKCDBW6IVPHYHYCZT", "message" : "small msg", "amount" : "10000", "hours_due" : "12", "fee" : 11, "encrypt" : 1}
- ○ Output: Success / Failure
    - ■ Example:
      {"ok":42}
- ○ NIS Dependent: **yes**
- ● /wallet/account/transaction/fee
    - ○ POST
    - ○ Input: {WalletName}, {Account}, {RecipientAccount}, {message}, {amount} {encrypt}*
        - ■ Example:
          {"wallet" : "Test", "account" : "TCE7WBBHFUAOJRPTMQM3RYQCD6CJN4DLXN6UPQAU", "recipient" : "TDTUF53QEEQT4HLZW7T4ECDPKCDBW6IVPHYHYCZT", "message" : "small msg", "amount" : "10000", "hours_due" : "12", "encrypt" : 0}
    - ○ Output: Fee
        - ■ Example:
          {"fee":11}
    - ○ NIS Dependent: **yes**
- ● /wallet/addressbook/list
    - ○ Input: {AccountID}
    - ○ Output: List of Addresses
- ● /wallet/addressbook/add
    - ○ Input: {AccountID}, {Address}
    - ○ Output: Success / Failure.
- ● /account/find
    - ○ POST
    - ○ Input: {Account}
        - ■ Example:
          {"account" : "TDTUF53QEEQT4HLZW7T4ECDPKCDBW6IVPHYHYCZT"}
    - ○ Output: {AccountExternal}
        - ■ Example:
          {"publicKey":"02c0ed044ddfe8a2eaed1931f45584f3b9c5d3d1c5f6e8a3a6 a8b0391e7ab8b412","balance":500000000000000,"address":"TAUULYJ4

MSHXON3GDQVUN4WFRTAQNADYL7FQPZDR","label":"My Wallet"}
- ○ NIS Dependent: **yes**
- ● /account/transactions
  - ○ Input: {AccountID} {timestamp}
  - ○ Output: **List<TransactionExternal>** - Transactions before provided timestamp (UTC Java timestamp)
  - ○ NIS Dependent: **yes**
- ● /account/harvests
  - ○ POST
  - ○ Input: {AccountID} {timestamp}
    - ■ Example:
      {"account" : "TDTUF53QEEQT4HLZW7T4ECDPKCDBW6IVPHYHYCZT", "timestamp" : 123456}
  - ○ Output: **SerializableList<HarvestInfo>** - List of HarvestInfos (at most 25) before provided timestamp (UTC Java timestamp)
  - ○ NIS Dependent: **yes**

- ● /wallet/account/lock
  - ○ POST
  - ○ Input: {WalletName} {Account}
    - ■ Example:
      {"wallet":"Test", "account" : "TDTUF53QEEQT4HLZW7T4ECDPKCDBW6IVPHYHYCZT"}
  - ○ Output: success / failure
    - ■ Example:
      {"ok":42}
  - ○ NIS Dependent: **yes**

- ● /wallet/account/unlock
  - ○ POST
  - ○ Input: {WalletName} {Account}
    - ■ Example:
      {"wallet":"Test", "account" : "TDTUF53QEEQT4HLZW7T4ECDPKCDBW6IVPHYHYCZT"}
  - ○ Output: success / failure
    - ■ Example:
      {"ok":42}
  - ○ NIS Dependent: **yes**

- ● /node/boot
  - ○ POST
  - ○ Input: {WalletName} {Account}{NodeName}

- - - Example:
      - {"wallet" : "Test", "account" : "TDTUF53QEEQT4HLZW7T4ECDPKCDBW6IVPHYHYCZT"}, "node_name" : "evil bob"}
  - Output: success / failure
    - - Example:
        - {"ok":42}
  - NIS Dependent: **yes**
- **/node/status**
  - **GET**
  - **Input:**
  - **Output: status of the node**
    - - Example:
        - {"ok":42}
      - {"fault":{"faultId":600,"parameter":"Local node not booted."}}
  - NIS Dependent: **yes**
- **/network**
  - GET
  - Input: url*
    - - Example:
        - /network?url=127.0.0.1:7890
        - /network
  - Output: NetworkGraph
    - - <**NetworkGraph**>
  - NIS Dependent: **yes**
- **/debug/logfile**
  - GET
  - Input: -
  - Output: Absolute path to the logfile
  - NIS Dependent: **no**
- **/debug/logLevel**
  - GET
  - Input: -
  - Output: Name of the current active log level
  - NIS Dependent: **no**
- **/debug/logLevel**
  - POST
  - Input: {log_level}
    - - Example:
        - {"log_level":"WARNING"}
  - Output: status
    - - Example:
        - {"ok":42}

- ○ NIS Dependent: **no**

- ● TBD

Result Object Definition

In case an error occured during executing the API, a fault object is returned and the HTTP Status is 200

- ● **Fault**
  ```
  {
     "fault": {
           "fault-id": 301,
           "parameter": ""
     }
  }
  ```

- ● **WalletExternal**
  ```
  {
  "additionalAddresses": [{AccountExternal}, ...],
     "lastRefresh": 1398593774662,
     "name": "Test",
     "primaryAccount": { <AccountExternal>  }
  }
  ```

- ● **AccountExternal**
  ```
  {
     "address": "TCE7WBBHFUAOJRPTMQM3RYQCD6CJN4DLXN6UPQAU",
     "balance": 0,
     "importance": 0,
     "label": {
           "privateLabel": "n",
           "label": ""
     },
     "publicKey": "A8Vb0lDlbCku1MiYsIg2djEygyUdIbapCZu5iduZ1zbS",
  }
  ```

- ● **AccountExternalDetailed**
  ```
  {
     "address": "TCE7WBBHFUAOJRPTMQM3RYQCD6CJN4DLXN6UPQAU",
     "balance": 0,
     "importance": 0,
     "label": {
           "privateLabel": "n",
           "label": ""
  ```

```
        },
        "publicKey": "A8Vb0lDlbCku1MiYsIg2djEygyUdIbapCZu5iduZ1zbS",
        "transactions": [<TransactionExternal>, ...]
    }
```

● **TransactionExternal**

```
    {
        "timeStamp": 1402273437436,
        "amount": 2100000,
        "sender": "TATHIESMY6QO6XKPCBZFEVVVFVL2UT3ESDHAVGL7",
        "fee": 2000000,
        "recipient": "TDHDSTFY757SELOAE3FU7U7KRYSTOP6FFB7XXSYH",
        "id": 6228954685176834000,
        "message": "Howdy",
         "encrypted": "1",
        "confirmations": 28
        }
```

● **NCCInfo**

```
    {
      "metadata": {
        "currentTime": 1061604,
        "application": "NEM",
        "startTime": 1061260,
        "version": "DEVELOPER BUILD",
        "signer": null
      },
      "remoteServer": "",
      "language": "en",
      "nisNode": {
        "activePeers": 1,
        "metaData": {
                "currentTime": 1061604,
                "application": "NEM",
                "startTime": 1061186,
                "version": "DEVELOPER BUILD",
                "signer": "CN=NEM Community,OU=Development
Team,O=NEM,L=Internet,ST=web,C=WD"
        },
        "application": "NIS",
        "maxHeight": 527,
        "ping": 30,
        "version": "DEVELOPER BUILD",
```

```
            "url": "http:\/\/localhost:7890\/",
            "platform": "Oracle Corporation (1.8.0_05) on Windows 7",
            "height": 526
    }
}
```

- **ConfigurationExternal**
```
{
        "language" : "en",
        "nisBootInfo" : {
                "bootNis" : 1,
                "remoteServer" : "",
                "accounId" : "TDHDSTFY757SELOAE3FU7U7KRYSTOP6FFB7XXSYH",
                "nodeName" : "Alice"
        }
}
```

- **HarvestInfo**
```
{
        "blockHash" : {"data" :
"JMKm5lmvVDElw6fjh65WOSbEqOKFrVUzJ8Wp4YOsVDU=" },
        "height" : 1023,
        "timestamp" : 12345,
        "totalFee" : 12000000
}
```

- **SerializableList<Item>**
```
{
        "data" : [list of Item as json]
}
```
  Example: SerializableList<HarvestInfo>
  { "data" : [{ "blockHash":{"data":"KzYwl269q4gYh5kFaHyjJq3tj0jDHU5NFZpndOcZj44="},
"totalFee":0 ,"height":20967 ,"timestamp":2994381 } , {"blockHash":
{"data":"MEV9F08iG7k64fzvtPvLbiQFP8ejb\/Yk+DMmOjvoWng="}, "totalFee":0, "height":20966,
"timestamp":2994327}] }

- **NetworkGraph**

Fault Ids
- FAULT_FILE_NOT_FOUND = 101
    - 'File not found.'

- FAULT_WALLET_NOT_CREATED = 102;
  - 'Wallet has not been created.'

- FAULT_WALLET_CORRUPT = 103
  - 'Wallet file is corrupted.
    Please recover your wallet from a back-up you should have taken when you created the wallet or added an account to it.'

- FAULT_WALLET_WRONG_PASSWORD = 104
  - 'The provided password is not correct.
    Hopefully you can remember the correct password. The password cannot be recoved if lost!'

- FAULT_WALLET_STORAGE_LOCATION_NOT_SET = 105;
  - 'Wallet storage location has not been set'

- FAULT_WALLET_NOT_OPENED_BEFORE = 106;
  - 'Before you can work with a wallet, it has to be opened. To ensure that you are eligible for accessing the wallet, you have to provide the password for that wallet.'

- FAULT_WALLET_DOES_NOT_CONTAIN_ACCOUNT = 107;
  - 'Wallet does not contain this account'

- FAULT_WALLET_ACCOUNT_CANNOT_BE_REMOVED = 108;
  - 'The account cannot be removed. Most likely because the account still has a balance greater than 0 NEMs.'

- FAULT_WALLET_NAME_EXISTS_ALREADY = 109
  - 'Another wallet with the same name exists already. Please choose an other wallet name.'

- FAULT_MINIMUM_FEE = 200;
  - 'Minimum fee is not met'

- FAULT_MAXIMUM_DEADLINE = 201;
  - 'Due hours for a transaction must less than one day'

- FAULT_NO_PUBLIC_KEY = 202;
  - 'No public key'

- FAULT_HTTP_CLIENT_NOT_STARTED = 300;
  - 'HTTP client has not been started'

- FAULT_HTTP_IO_EXCEPTION = 301;
  - 'HTTP I/O exception'

- FAULT_HTTP_STATUS_400 = 302;
  - 'HTTP status 400'

- FAULT_HTTP_STATUS_UNKNOWN = 303;
  - 'HTTP status unknown'

- FAULT_URL_MALFORMED = 304;
  - 'URL malformed'

- FAULT_NIS_NOT_AVAILABLE = 305;
  - 'NEM Infrastructure Server is not available'

- FAULT_UNEXPECTED_ERROR = 306;
  - 'An error occured that the development team did not have foreseen.
  - Please apologize for this, maybe a retry might help. Otherwise, please open up an issue within the NEM NIS/NCC community'

- FAULT_NIS_VALIDATION_ERROR = 307;
  - 'NIS validation error'

- FAULT_PARAMETER_MISSING = 400
  - 'Some parameter is missing'

- FAULT_NO_LONG_VALUE = 401;
  - 'No long value'

- FAULT_NO_INT_VALUE = 402;
  - 'No integer value'

- FAULT_NO_BIGINT_VALUE = 403;
  - 'No big integer value'

- FAULT_INVALID_BOOT_STRATEGY_VALUE = 404;
  - 'Invalid Boot strategy value'

- FAULT_CONFIGURATION_SAVE_FAILED = 500;
  - 'Failed to save configuration file'

- FAULT_CONFIGURATION_DIR_UNUSABLE = 501;

- ○ 'Configuration directory is unusable'

- ● FAULT_NODE_NOT_BOOTED = 600;
    - ○ 'NCC requires NIS server to be booted for sending and receiving transactions from the NEM cloud. Please use the NCC menu entry for booting the local node.'

- ● FAULT_NODE_ALREADY_BOOTED = 601
    - ○ 'The nis node is already booted. A second attempt to boot nis is not possible.'

- ● FAULT_NIS_API_FAILURE_FORAGING_INELIGIBLE = 700;
    - ○ 'The provided account does not satisfy the basic criteria for harvesting. Mainly it is related to the amount of NEMs within the account. Harvesting starts with at least 10.000 NEMs.'

- ● FAULT_NIS_API_FAILURE_PAST_DEADLINE = 701;
    - ○ 'The provided deadline is in the past. The deadline must be provided within a 1 day period.'

- ● FAULT_NIS_API_FAILURE_FUTURE_DEADLINE = 702;
    - ○ 'The provided deadline is too far in the future. The deadline must be within one day time period.'

- ● FAULT_NIS_API_FAILURE_INSUFFICIENT_BALANCE = 703;
    - ○ 'Your account does not have the right balance to send the provided amount of NEMs.'

- ● FAULT_NIS_API_FAILURE_MESSAGE_TOO_LARGE = 704;
    - ○ 'The provided message text is too large to be send via NEM. Please try to reduce the length of the message you want to send.'

- ● FAULT_NIS_API_FAILURE_HASH_EXISTS = 705;
    - ○ 'The transaction hash already exists in the database or the list of unconfirmed transactions.'

- ● FAULT_NIS_API_FAILURE_SIGNATURE_NOT_VERIFIABLE = 706;
    - ○ 'The signature of the transaction could not be verified.'

- ● FAULT_NIS_API_FAILURE_TIMESTAMP_TOO_FAR_IN_PAST = 707;
    - ○ 'The time stamp of the transaction id too far in the past.

,

- ● FAULT_NIS_API_FAILURE_TIMESTAMP_TOO_FAR_IN_FUTURE = 708;
    - ○ 'The time stamp of the transaction is too far in the future.'

- FAULT_NIS_API_FAILURE_UNKNOWN_ACCOUNT = 709;
  - 'The account is unknown. An account needs to be part of at least one transaction (sender or recipient) to be known to the network.'

Account statuses
- "UNKNOWN"   The status of the account is unknown.
- "LOCKED"   The account is not foraging.
- "UNLOCKED"   The account is foraging.

Wallet file definition
The client will store the information on wallets locally. The file is a simple JSON structure:

{ "wallet":{

  "lastRefresh":1395500866755, not sure whether is required within the wallet or just within the cache, but since we have NEM snapshots a cache is not required, since the wallet catches up with the NEM blockchain very fast

  "primaryAccount":{

  "key":729254191078562471138616581832657356531156824525583576662757 26896940038915476, NEM's private key required for all transactions

  "label":""}, Label of the account which is also known to the NEM cloud.

  "additionalAccounts":[] In case the wallet holds more than just one NEM account. This is an array of accounts. The same structure as for primaryAccount.

  }

}

The wallet file has the extension "wlt" and is encrypted with a password :
- Given password is hashed with "SHA3-256"
- CBCBlockCipher(new AESEngine()) (according bouncy castle)

Command Line Parameter
The NCC application accepts the following command line parameter:
- **-nis_jnlp**
  The URL to the JNLP file of the nis-application. This URL is used to start NIS.
  In addition, NCC has to be configured to work with a local NIS (default).
- **-nem_folder**
  The path to the wallet store. The default is **<home of the logged-in user>/nem/ncc**.
  In case NCC should be run from a USB stick, the current directory could be passed to avoid any left-overs on the host machine.
- **-nis_ws**
  If set to 1, NIS will be started via Java WebStart. A zero (0) indicates that no NIS should be started.

In case Java WebStart is not used to start NCC and to keep NCC and NIS at the most current version, the zip-file of NCC contains start scripts.

Version	Comment	Author
0.1 - 2014-03-18	Initial version	Thies1965
0.2 - 2014-03-22	Wallet file structure definition added.	Thies1965
0.3 - 2014-04-08	Completed the list of developers	Thies1965
0.4 - 2014-04-12	Wallet API finalized	Thies1965
0.5 - 2014-04-27	Highlighted API are implemented	Thies1965
0.6 - 2014-05-10	info response enhanced account response enhanced /wallet/account/info implemented	Thies1965
0.7 - 2014-05-18	info object extended to show signer new API added. /wallet/account/new /wallet/account/add	Thies1965
0.8 - 2014-05-20	NccInfo extended to reflect NIS Node info API fee: unnecessary parameter password removed	Thies1965
0.9 - 2014-05-23	/account/transactions /wallet/account/forage added Transaction List detailed	Thies1965
0.16 - 2014-05-31	/wallet/account/primary	Thies1965
0.33 - 2014-06-06	NCCInfo enhanced	Thies1965

0.37 - 2014-06-10	Wallet import, export and merge are enabled to receive file content instead of path information	Thies1965
0.0.43 alpha - 2014-06-16	new api calls: lock/unlock accounts and node/boot, added account status. Use AccountMetaDataPair in account/get request result.	BloodyRookie
0.0.44 alpha - 2014-06-19	Miscellaneous bug fixes.	BloodyRookie
0.0.50 alpha - 2014-06-22	NisBootInfo class	BloodyRookie
0.0.52 alpha - 2014-06-24	/wallet/account/label /wallet/account/remove	BloodyRookie
0.0.53 alpha - 2014-06-25	/wallet/account/harvests	BloodyRookie
0.0.56 alpha - 2014-06-??	/node/status	BloodyRookie
0.0.56 alpha - 2014-06-??	TransactionExternal extended to reflect whether message was encrypted or not	Thies1965
0.0.56 alpha - 2014-06-??	/wallet/password/change	BloodyRookie
0.0.68 alpha	/wallet/name/change	Thies1965
0.0.68 alpha	Command line options added Debug REST Api added	Thies1965
0.0.72 alpha	/info/nis/check added for quick check for NIS running	Thies1965