Obsah

Spojení NPA ↔ NPA přes VPN-ID (dd_npaid)

Platí pro verzi NPA R04: 2.15.beta1 · Vytvořeno: 2026-07-16

Tento dokument popisuje mechanismus, kterým se jednotka NPA připojuje k jiné jednotce NPA přes VPN pomocí **ID modulu** místo pevné IP adresy, a obsahuje **zadání pro serverovou stranu** (endpoint `moduleslist`), bez které nelze výběr ID na webu naplnit a spojení zprovoznit.

Starý web (Flask aplikace v `~/Nextcloud/src/NPA_R04_MODBUS/ea/ea.py`) endpoint `/moduleslist` poskytoval; nový monitoring (EnergyCore, monitoring.dev.energyadmin.cz) ho zatím nemá — proto NPA nedokáže seznam modulů nabídnout a `dd_npaid` není jak vyplnit.

---

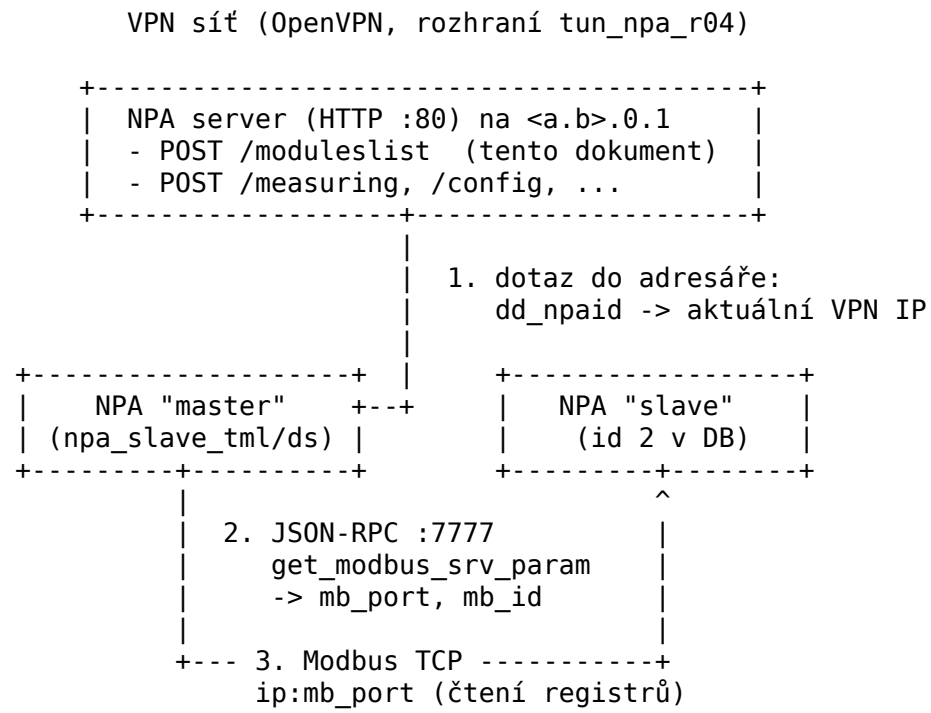
1. K čemu to je

Zařízení typu **„Obchodní měření na podřízeném NPA"** (`npa_slave_tml`) a **„Rozpadové místo na podřízeném NPA"** (`npa_slave_ds`) čtou data z jiné jednotky NPA po Modbus TCP. Nadřízená jednotka se k podřízené může připojit dvěma způsoby (web: *Zařízení → nastavit → Datová komunikace*, šablona `device_edit_commTcpId.html`):

- **Zadat IP adresu** (`dd_ip_npaid = "dd_ip"`) — pevně zadaná `dd_ip`, `dd_port` a `dd_idNet`;
- **Vybrat ID** (`dd_ip_npaid = "dd_id_npaid"`) — uloží se `dd_npaid` (ID modulu), IP se dohledává za běhu.

Výhoda ID: adresa přidělená VPN serverem se může změnit (reinstalace, přečíslování sítě), ID modulu v databázi serveru je stabilní. IP se dohledává **při každém navazování spojení**, takže změna adresy nevyžaduje zásah do konfigurace nadřízené jednotky.

2. Architektura



Datový tok jde přímo mezi jednotkami po VPN (Modbus TCP). Server slouží pouze jako adresář „ID → aktuální VPN IP".

Jak jednotka najde server

`GetVpnNpaServerIp()` (`util/nutils.cpp`) vezme IPv4 adresu rozhraní `tun_npa_r04` a nahradí poslední dva oktety `0.1` — z `10.81.0.14` tedy vyjde server `10.81.0.1`. Server proto **musí poslouchat na HTTP portu 80 na VPN adrese `<a.b>.0.1`**. Bez navázané VPN vrací funkce prázdný řetězec a všechny cloudové funkce se přeskočí.

3. Průběh připojení (strana NPA, stávající kód)

1. **Výběr na webu** — `NHtml::GetNpaToSelect()` (`nhtml.cpp`) zavolá `NModules::GetModules()` a naplní `<select id="dd_npaid">` položkami `id → "name - sn - online"`. Online modul je zvýrazněn.
2. **Uložení** — `dd_npaid` (číselné `id` modulu) se uloží do configu zařízení.
3. **Připojení** — `DeviceModbus::NpaIdConnect()` (`device/devicemodbus.cpp`):
   1. `NModules::GetModuleIp(dd_npaid)` → `POST http://<a.b>.0.1/moduleslist`, v odpovědi najde modul se shodným `id` a vezme jeho `ip`;
   2. `NJCli::GetModbusSrvParam(ip)` → JSON-RPC na `http://<ip>:7777/` s `{"req": "get_modbus_srv_param"}`; podřízená jednotka vrátí `mb_port` (= její `ns_netPort`) a `mb_id` (= `ns_netId`);
   3. otevře Modbus TCP spojení `ip:mb_port` a čte registry (`npa_slave_tml`: blok 2000–2056; `npa_slave_ds`: blok 3000+offset, 121 registrů).
4. Při výpadku spojení se celý postup opakuje (znovu se dohledá i IP).

Požadavek na klienta `/moduleslist` (přesný tvar)

`NModules::GetModules()` posílá:

```
POST /moduleslist HTTP/1.1
Host: <a.b>.0.1
```

```
Content-Type: application/json

{ "sn": "NPA20240418-1801", "time": "2026-07-16T07:30:00" }
```

- `sn` — sériové číslo žádající jednotky (`NConfig::mSN`),
- `time` — UTC čas jednotky (ISO 8601).

Očekávaná odpověď — objekt s polem `modules`:

```
{
  "modules": [
    {
      "id": 1,
      "sn": "NPA20240418-1801",
      "name": "ABEGU - Testovací",
      "ip": "10.81.0.14",
      "online": 1
    },
    { "id": 2, "sn": "NPA20240905-2116", "name": "ABEGU - Testovací 2",
      "ip": "10.81.0.10", "online": 0 }
  ]
}
```

Neregistrované `sn`: `{ "error": "modul neexistuje" }` (klient pak zobrazí prázdný výběr).

Klient čte pouze klíče `id`, `ip`, `name`, `sn`, `online`; ostatní klíče ignoruje (starý server posílal celé řádky tabulky `modules` včetně `setting` — není potřeba).

4. Zadání pro serverovou stranu

4.1 Cíl

Implementovat na monitorovacím serveru (EnergyCore) endpoint **`POST /moduleslist`** dostupný **na VPN adrese `<a.b>.0.1`, port 80** tak, aby jednotky NPA mohly:

1. na webu jednotky nabídnout seznam modulů pro výběr `dd_npaid`,
2. za běhu překládat `dd_npaid → aktuální VPN IP` pro spojení NPA ↔ NPA.

4.2 Funkční požadavky

1. **Kontrakt** přesně dle kapitoly 3 (request `{sn, time}`, response `{modules: [...]}`). Klient je nasazený v terénu — tvar odpovědi nelze měnit.
2. **Povinné klíče** každého modulu: `id` (int, stabilní identifikátor — plní se do `dd_npaid`), `sn` (string), `name` (string), `ip` (string — aktuální VPN adresa), `online` (**číslo** 0/1). Klient čte `id` a `online` tolerantně (číslo i řetězec), přesto posílejte čísla — starý server měl `online` jako String.
3. **Filtrování**: vrátit pouze moduly, které patří ke stejné instalaci/zákazníkovi jako žádající `sn` (starý server filtroval přes `practitioner`; v nové DB odpovídá vazbě instalace/zákazník). Jednotka nesmí vidět cizí moduly.
4. **`ip` musí být aktuální VPN adresa** jednotky. Doporučené řešení: pevné přidělení adres v OpenVPN (`ccd/ifconfig-push`) + evidence v DB (monitoring už sloupec „IP VPN" u PLC má). Alternativa: číst OpenVPN status log / management interface.
5. **`online`**: stav připojení k VPN (OpenVPN status), případně odvozený z posledního příjmu na `/measuring` (např. < 2 min = online) — stejný údaj, který monitoring zobrazuje jako „Je online".
6. **Neznámé `sn`** → `{"error": "modul neexistuje"}`, HTTP 200 (kompatibilita).

4.3 Nefunkční požadavky

- **Dostupnost pouze z VPN sítě** — endpoint nesmí být vystaven do internetu; bind na VPN adresu serveru nebo firewall na `tun` rozhraní. Jiná autentizace není (klient posílá jen `sn`), proto

je izolace na VPN nutná podmínka.
- **Latence**: klient čeká na odpověď nejvýše 3 s, pak požadavek přeruší — pomalejší odpověď znamená, že se seznam na webu nenaplní a spojení přes ID se nenaváže. Odpověď proto musí být rychlá (< 1 s). **Nekešovat** — smysl VPN-ID je právě v tom, že se IP může měnit; odpověď musí vždy odrážet aktuální stav (IP, online).
- **Logování**: příchozí `sn` + čas se hodí evidovat (server tím zadarmo získává heartbeat jednotek, stejně jako to dělal starý web).

4.4 Návazné endpointy (informativně)

Na téže adrese očekává jednotka i endpointy, které dnes obsluhoval starý web (`NRemoteLog`): `POST /measuring` (periodické stavy, v odpovědi `srv_setting`), `POST /config` (záloha konfigurace), `POST /getlastmeas` (poslední uložený stav po startu). Nejsou předmětem tohoto zadání, ale při návrhu serveru je dobré s nimi počítat — viz `doc/interfaces.md`, sekce cloudoví klienti.

4.5 Akceptační test

```
# na stroji ve VPN (nebo přímo na serveru):
curl -s -X POST http://10.81.0.1/moduleslist \
     -H 'Content-Type: application/json' \
     -d '{"sn": "NPA20240418-1801", "time": "2026-07-16T07:30:00"}'
# očekávám: {"modules": [ ... id/ip/name/sn/online ... ]}, jen moduly stejné instalace
```

Poté na NPA:

1. *Zařízení → přidat/nastavit zařízení typu `npa_slave_tml` (MODBUS TCP)* — v sekci Datová komunikace zvolit **Vybrat ID**; v selectu se objeví seznam modulů (= `/moduleslist` funguje).
2. Vybrat podřízenou jednotku, uložit (`dd_npaid`).
3. Do ~10 s se zařízení připojí (stav na stránce *Přehled* / detail zařízení); v logu jednotky `remote connect - Device UUID ... mIp: <vpn-ip>, mPort: <port>`.
4. Ověřit tok dat: hodnoty TML (P, Q, U…) z podřízené jednotky na stránce Přehled.

5. Postup zprovoznění spojení (checklist)

1. Obě jednotky mají zapnutou vzdálenou správu (*Nastavení sítě → Vzdálená správa*) a navázanou VPN (stránka *Přehled → IP Adresa VPN*).
2. Server implementuje `/moduleslist` (kap. 4) a zná VPN IP obou jednotek.
3. Na podřízené jednotce zkontrolovat `ns_netPort`/`ns_netId` (Modbus TCP server) — tyto hodnoty si nadřízená jednotka zjistí sama přes JSON-RPC.
4. Na nadřízené jednotce nakonfigurovat zařízení `npa_slave_tml`/`npa_slave_ds` s volbou **Vybrat ID** (kap. 4.5, body 1–4).