2STON™ SPN v3.0

IoTSec : 2ip IoT Security Solution

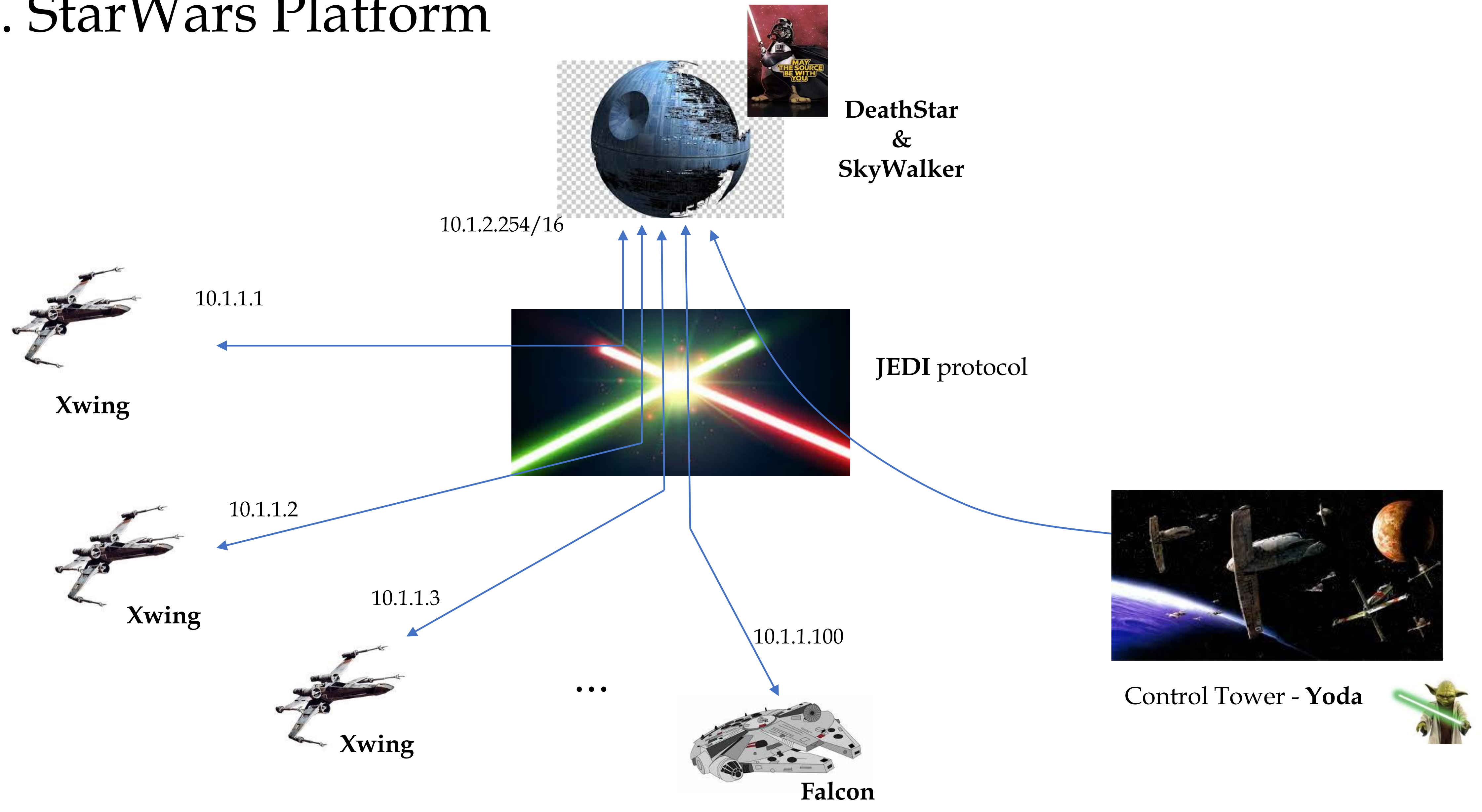
SPNBox Star Console

Michael Chunghan.Yi(michael@2ipco.com)
2ip Inc,
Doc. Revision: 1.3

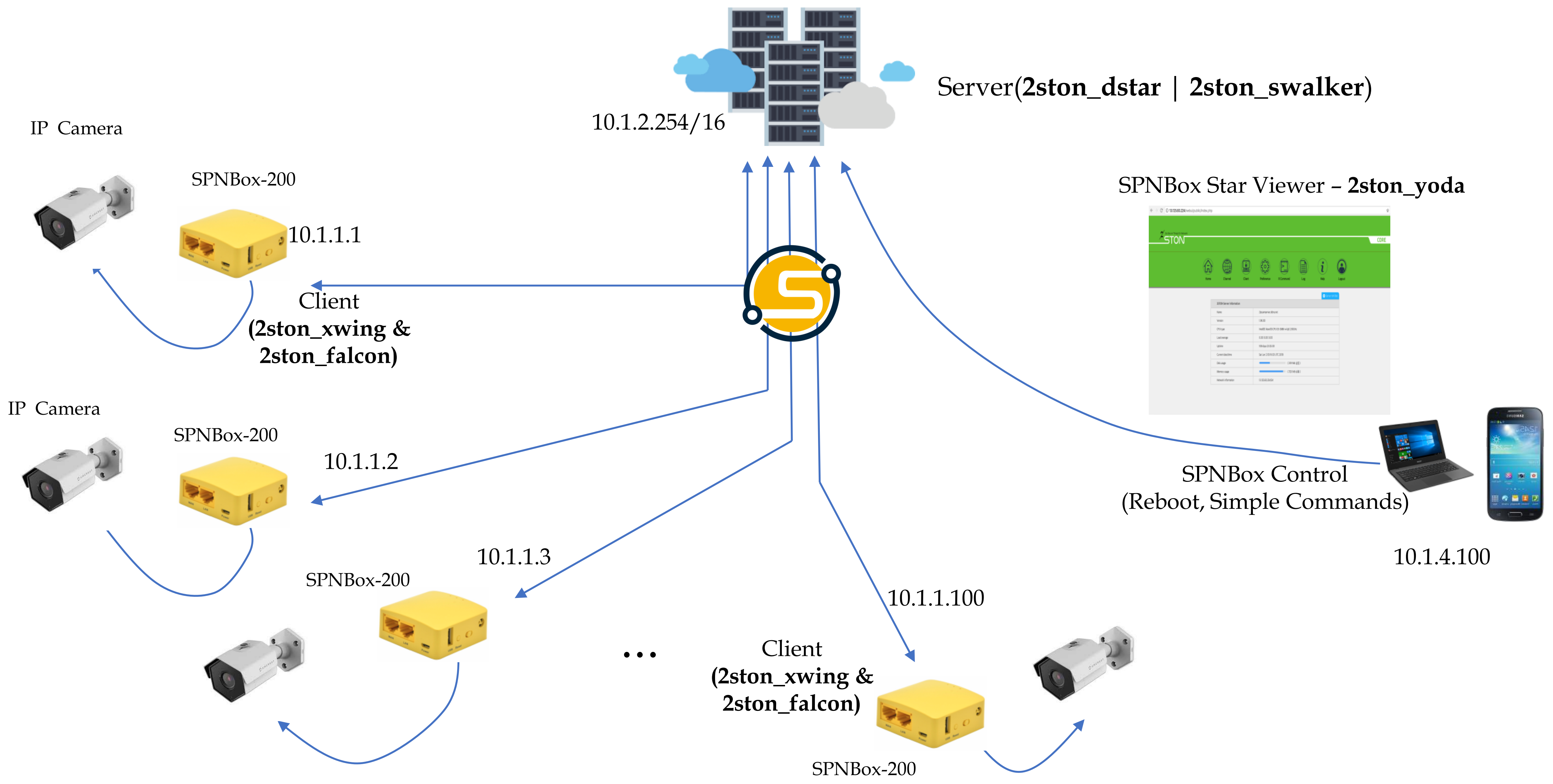
SPN v3.0

Table of Contents

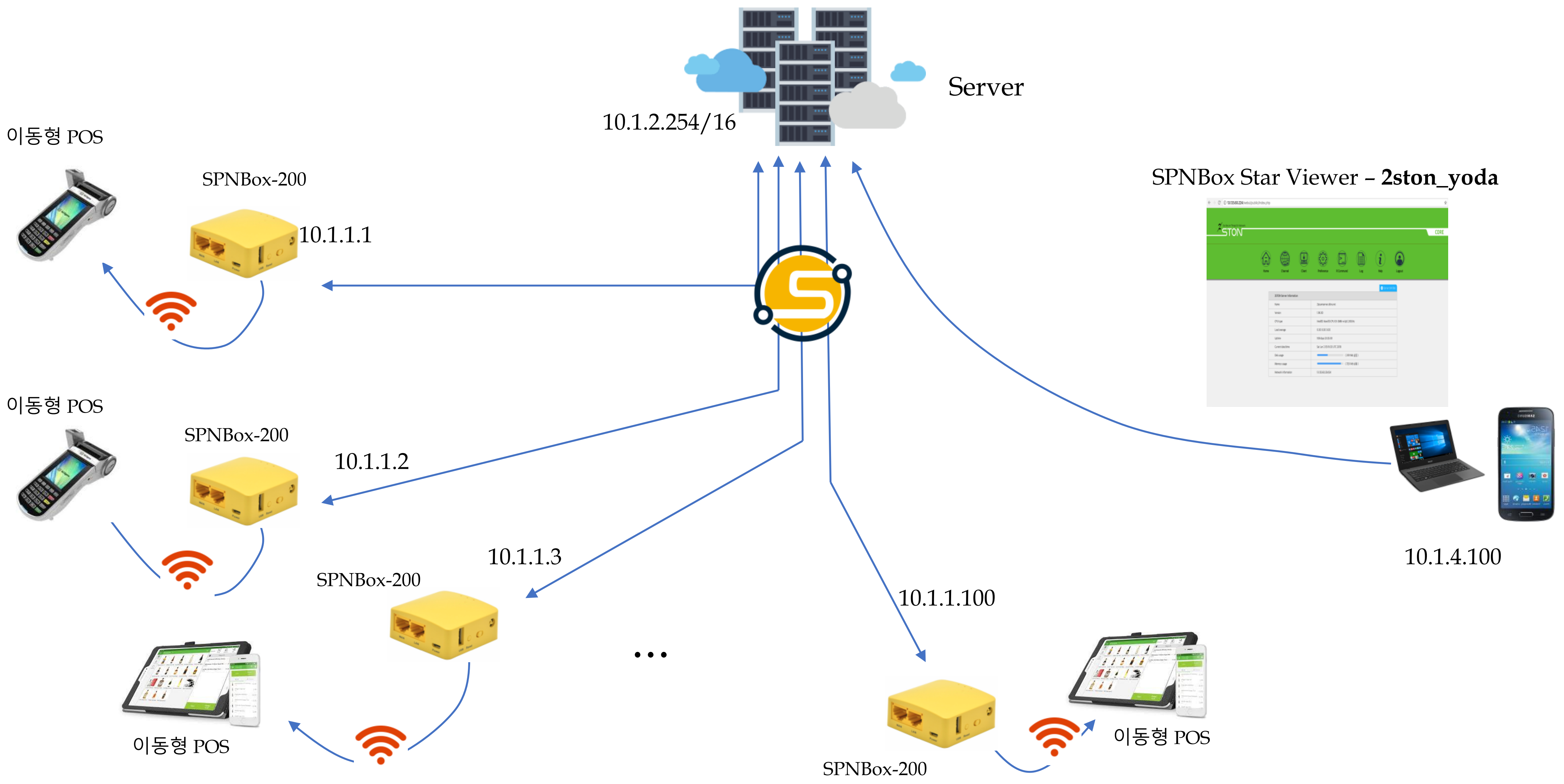
1. StarWars Platform



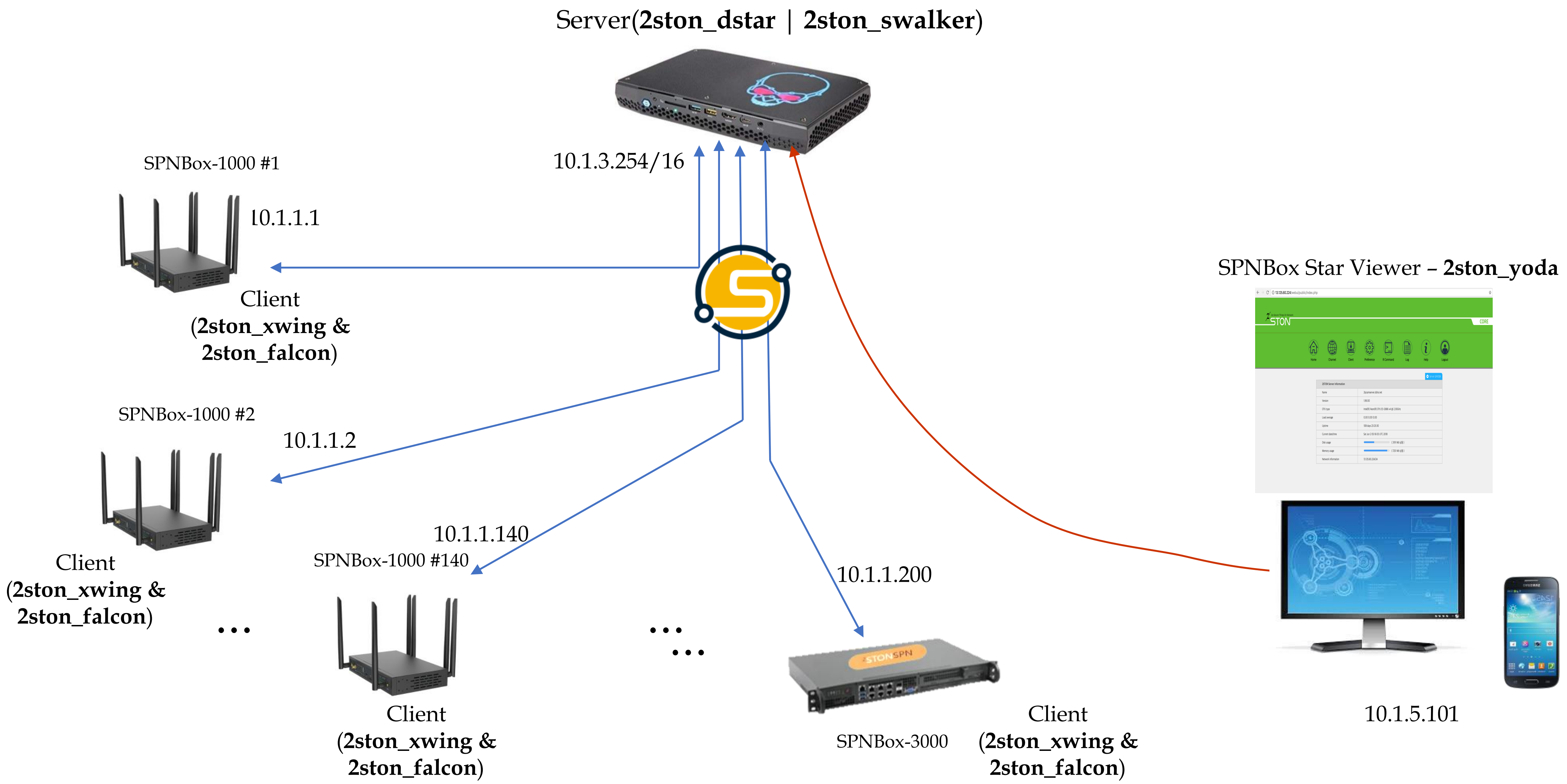
2. SPNBox Star Console Architecture(1)



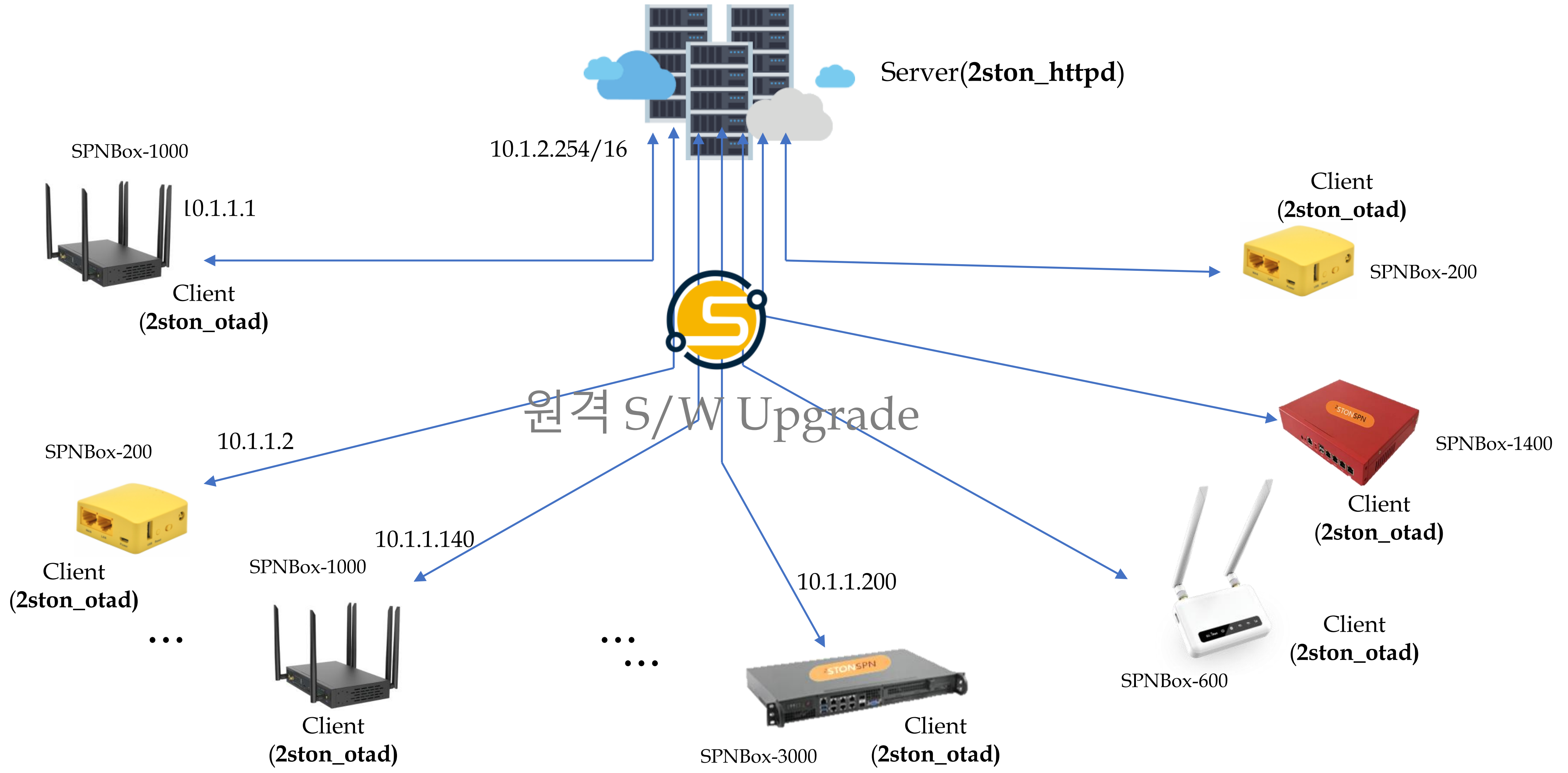
2. SPNBox Star Console Architecture(2)



2. SPNBox Star Console Architecture(3)



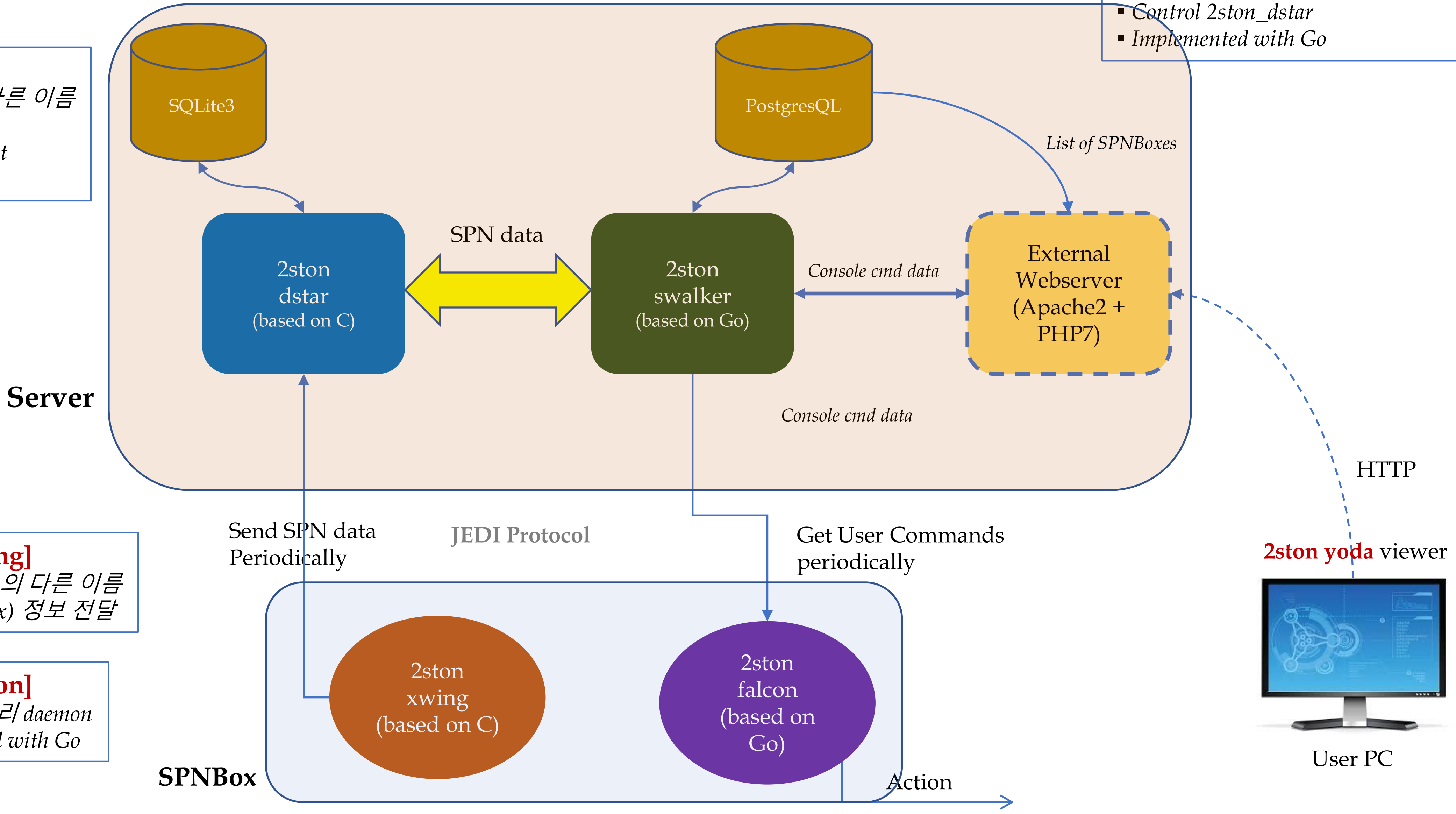
2. SPNBox Star Console Architecture(4)



2. SPNBox Star Console Architecture(5-1)

[2ston_swalker]
- *DB Manager*
- *User Command Queue Management*
- *Communication w/ 2ston_dstar*
- *Control 2ston_dstar*
- *Implemented with Go*

[2ston_dstar]
- *2ston_spnd2의 다른 이름*
- *ip 동적 할당*
- *client management*
- *sqlite3 db table*



[2ston_xwing]
- *2ston_spnc2의 다른 이름*
- *client(spnbox) 정보 전달*

[2ston_falcon]
- *user cmd 처리 daemon*
- *Implemented with Go*

2. SPNBox Star Console Architecture(5-2)

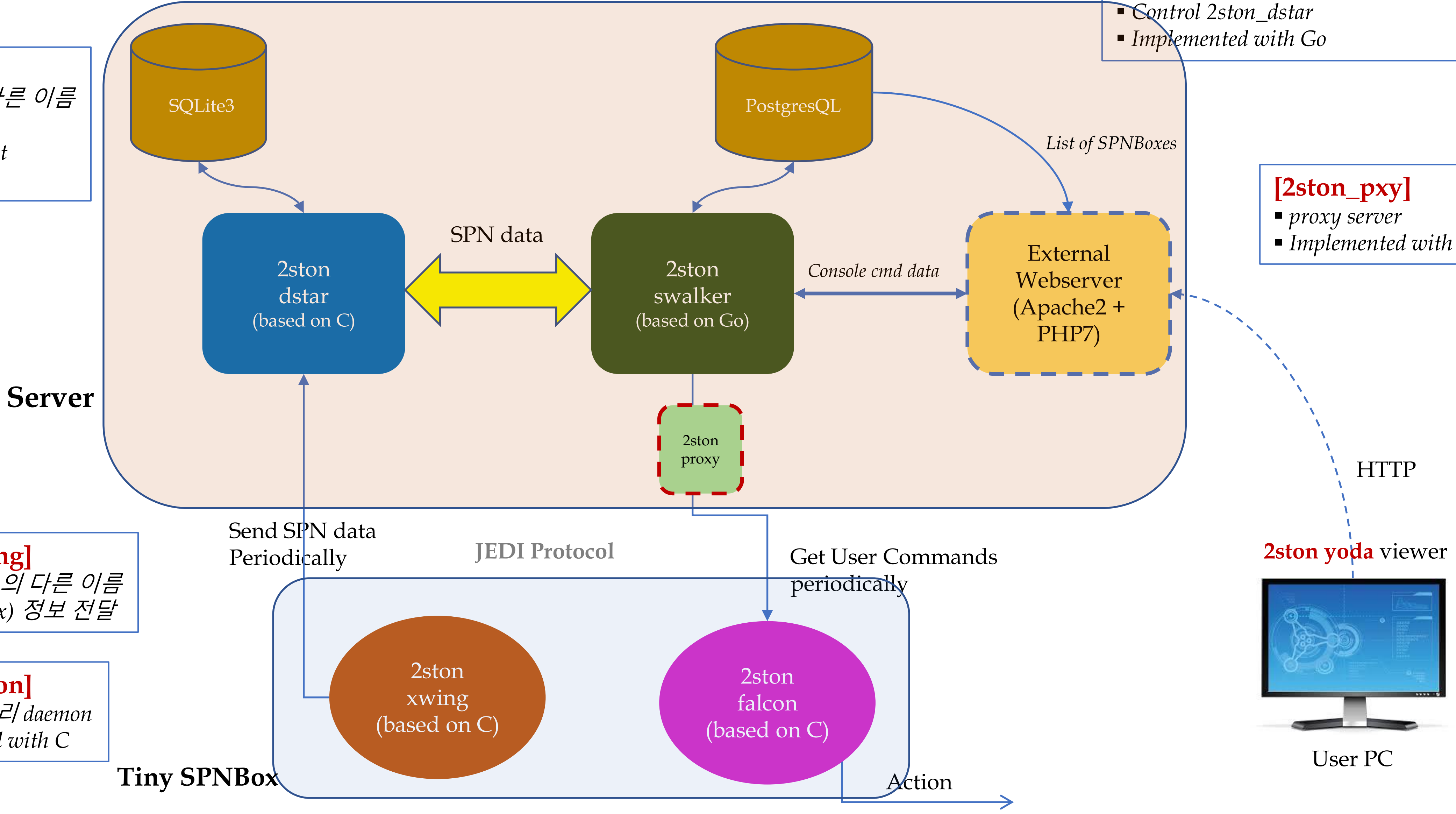
[2ston_swalker]
- *DB Manager*
- *User Command Queue Management*
- *Communication w/ 2ston_dstar*
- *Control 2ston_dstar*
- *Implemented with Go*

[2ston_dstar]
- *2ston_spnd2의 다른 이름*
- *ip 동적 할당*
- *client management*
- *sqlite3 db table*

[2ston_pxy]
- *proxy server*
- *Implemented with C*

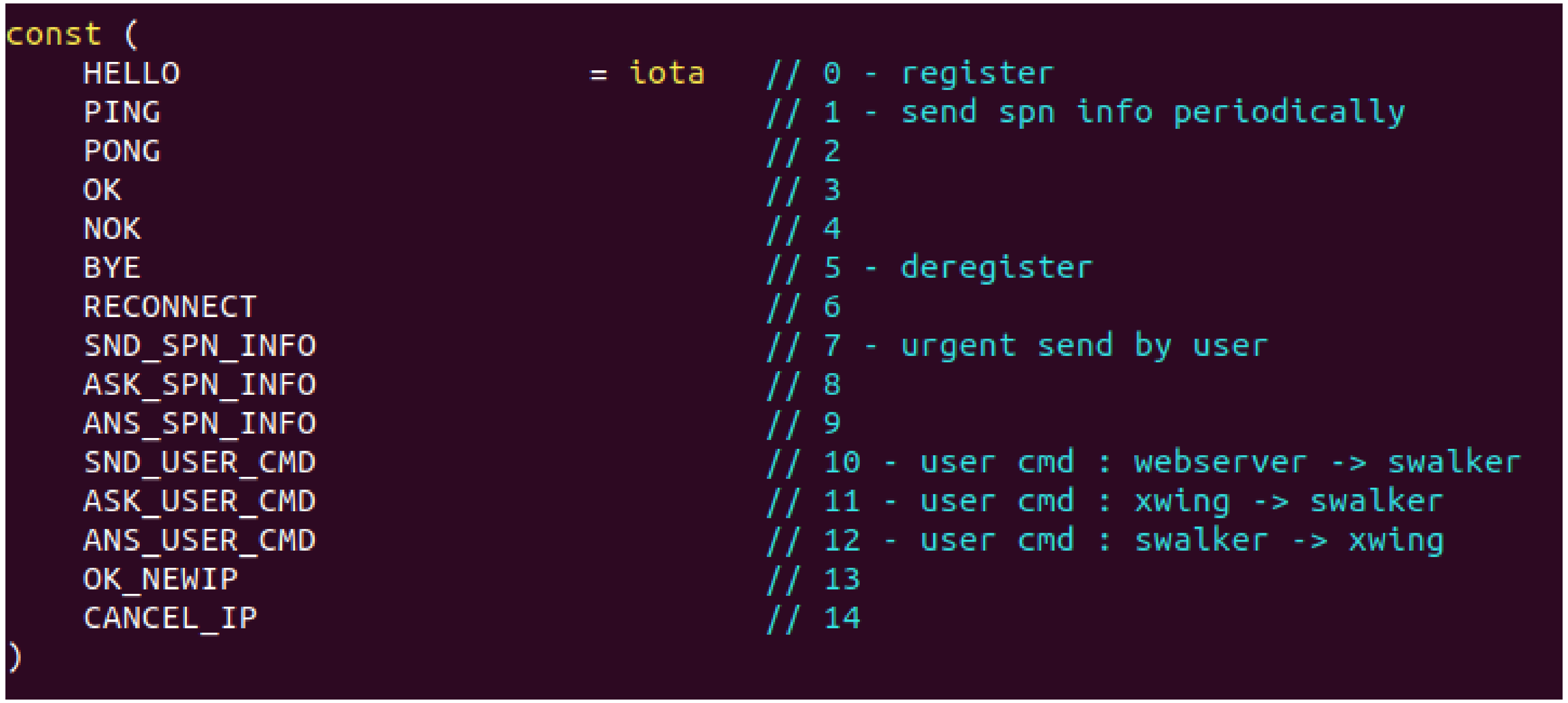
[2ston_xwing]
- *2ston_spnc2의 다른 이름*
- *client(spnbox) 정보 전달*

[2ston_falcon]
- *user cmd 처리 daemon*
- *Implemented with C*



Star Console Protocol

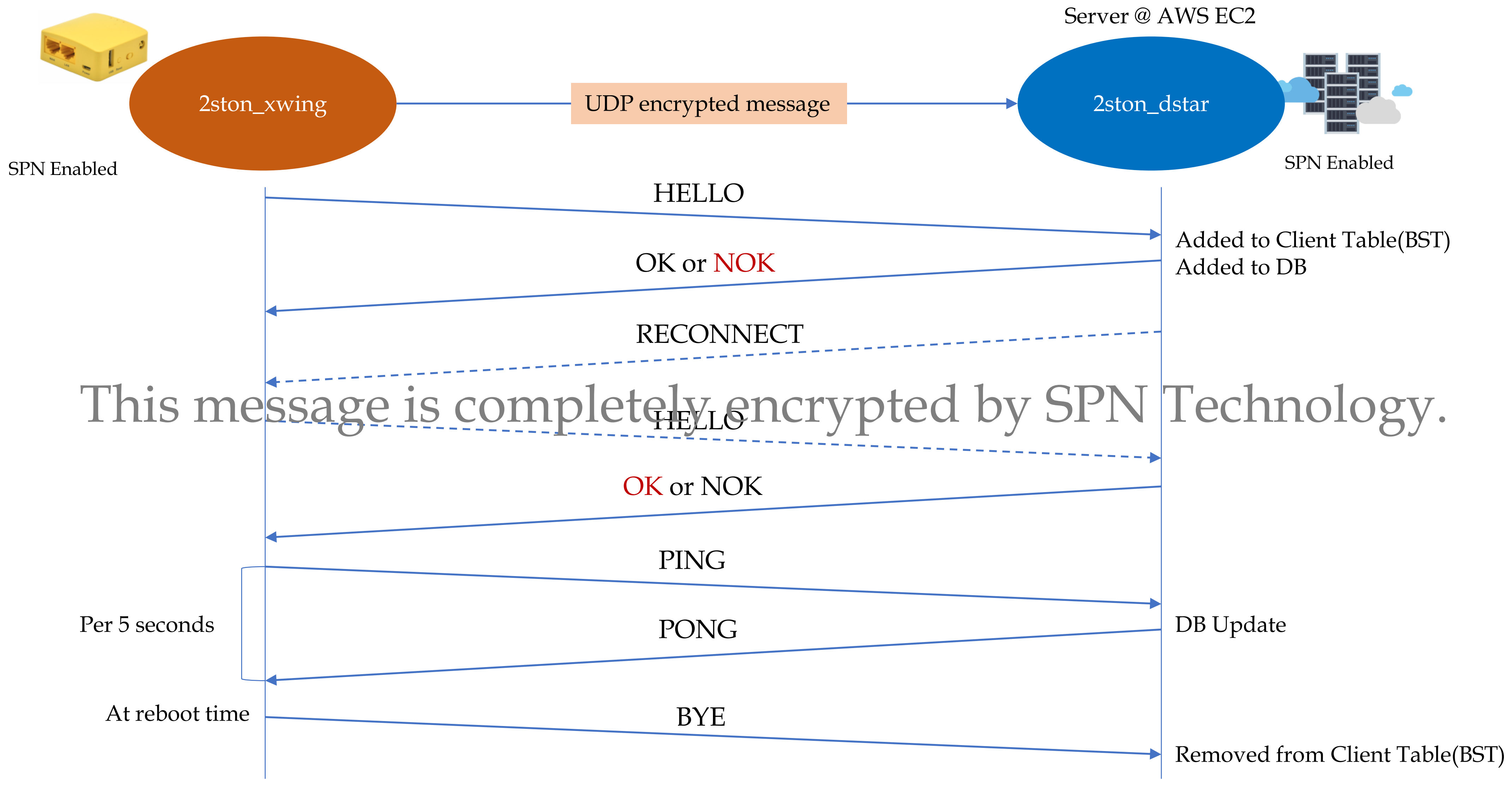
3. SPNBox Star Console Protocol(1)



```
const (
    HELLO                     = iota   // 0 - register
    PING                               // 1 - send spn info periodically
    PONG                               // 2
    OK                                 // 3
    NOK                                // 4
    BYE                                // 5 - deregister
    RECONNECT                          // 6
    SND_SPN_INFO                       // 7 - urgent send by user
    ASK_SPN_INFO                       // 8
    ANS_SPN_INFO                       // 9
    SND_USER_CMD                       // 10 - user cmd : webserver -> swalker
    ASK_USER_CMD                       // 11 - user cmd : xwing -> swalker
    ANS_USER_CMD                       // 12 - user cmd : swalker -> xwing
    OK_NEWIP                           // 13
    CANCEL_IP                          // 14
)
```

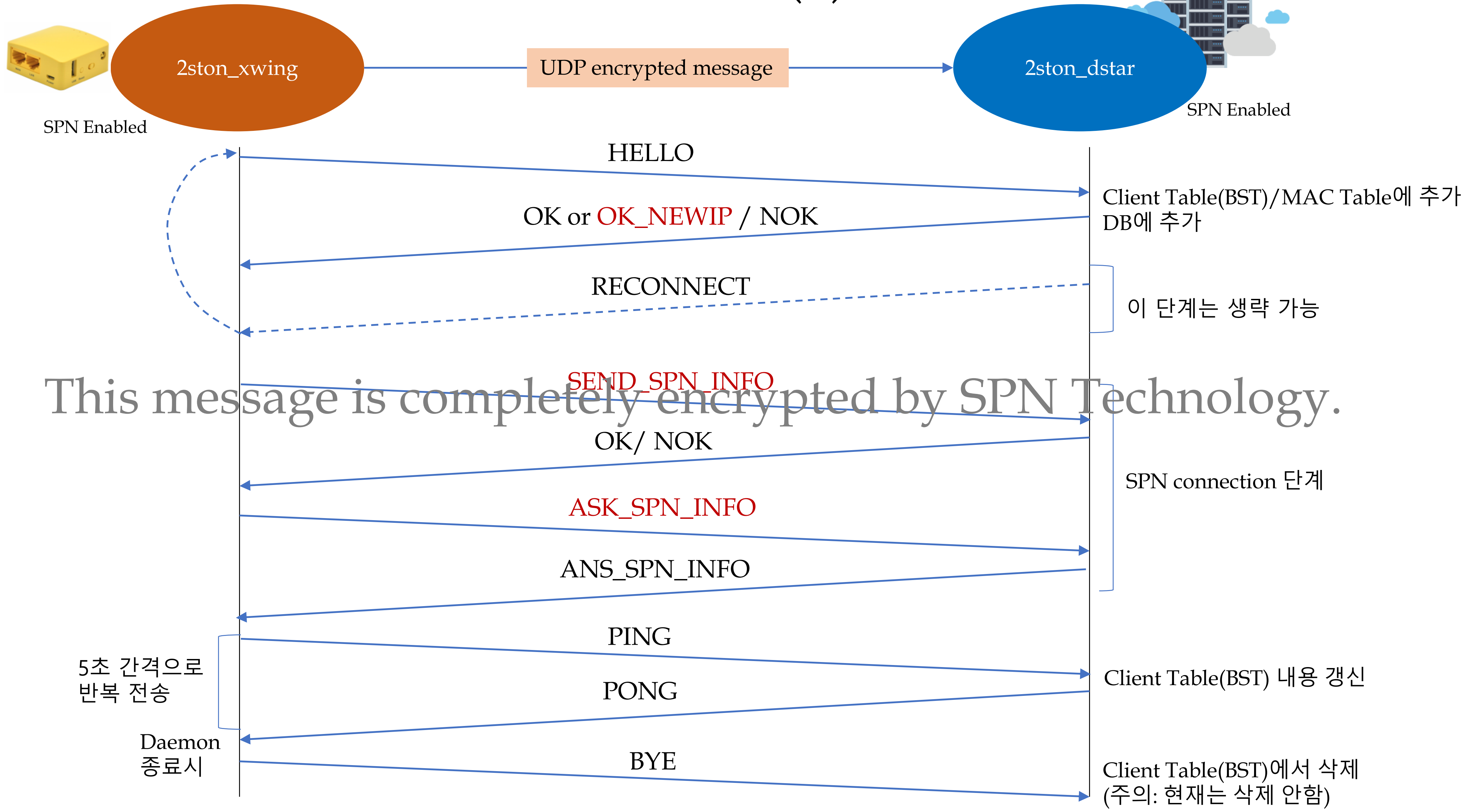
JEDI(Just Embedded Device Interface) Protocol Commands

3. SPNBox Star Console Protocol(2)



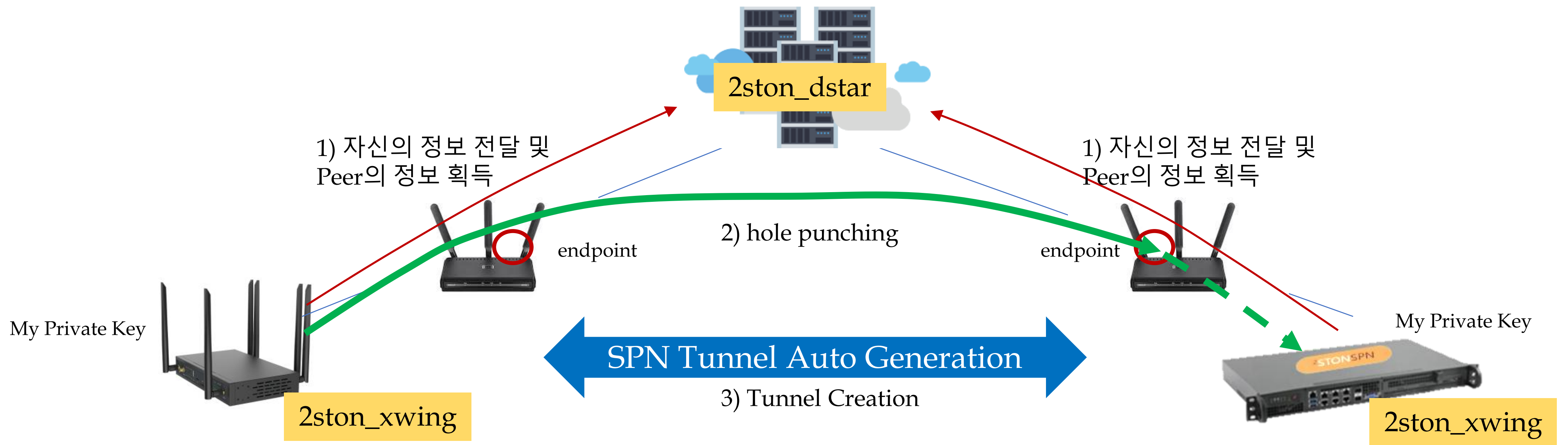
JEDI Protocol

3. SPNBox Star Console Protocol(3)



DeathStar Daemon

4. DeathStar Daemon(1) – Auto Connection(1)



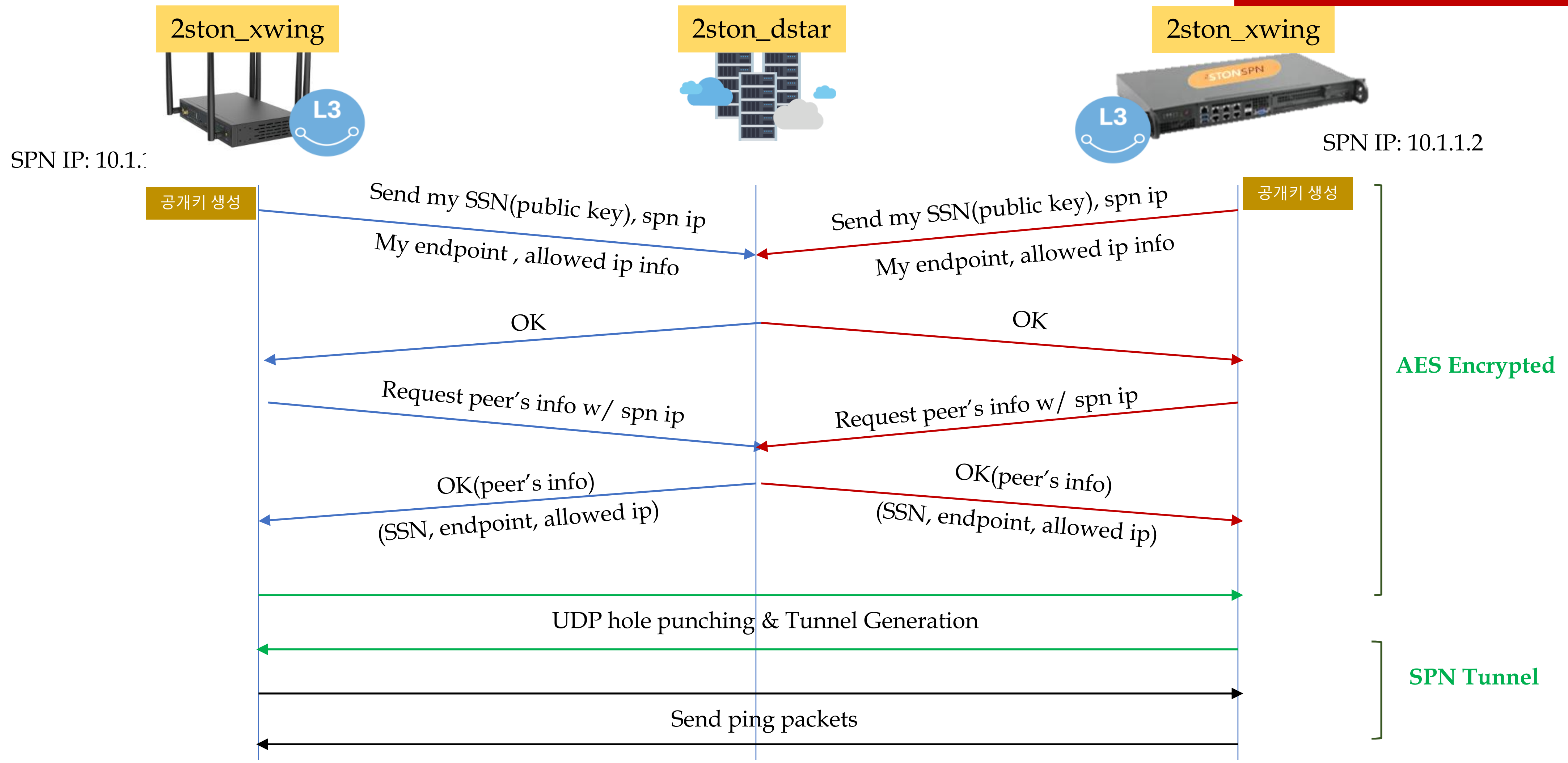
Peer's Public key(= SSN)	Peer's Public key(= SSN)
Peer's SPN IP Address	Peer's SPN IP Address
Peer's Endpoint(RealIP:Port)	Peer's Endpoint(RealIP:Port)
Peer's Allowed Networks	Peer's Allowed Networks

1) Peer's information 획득
2) UDP Hole Punching
3) Automatic Tunnel Creation

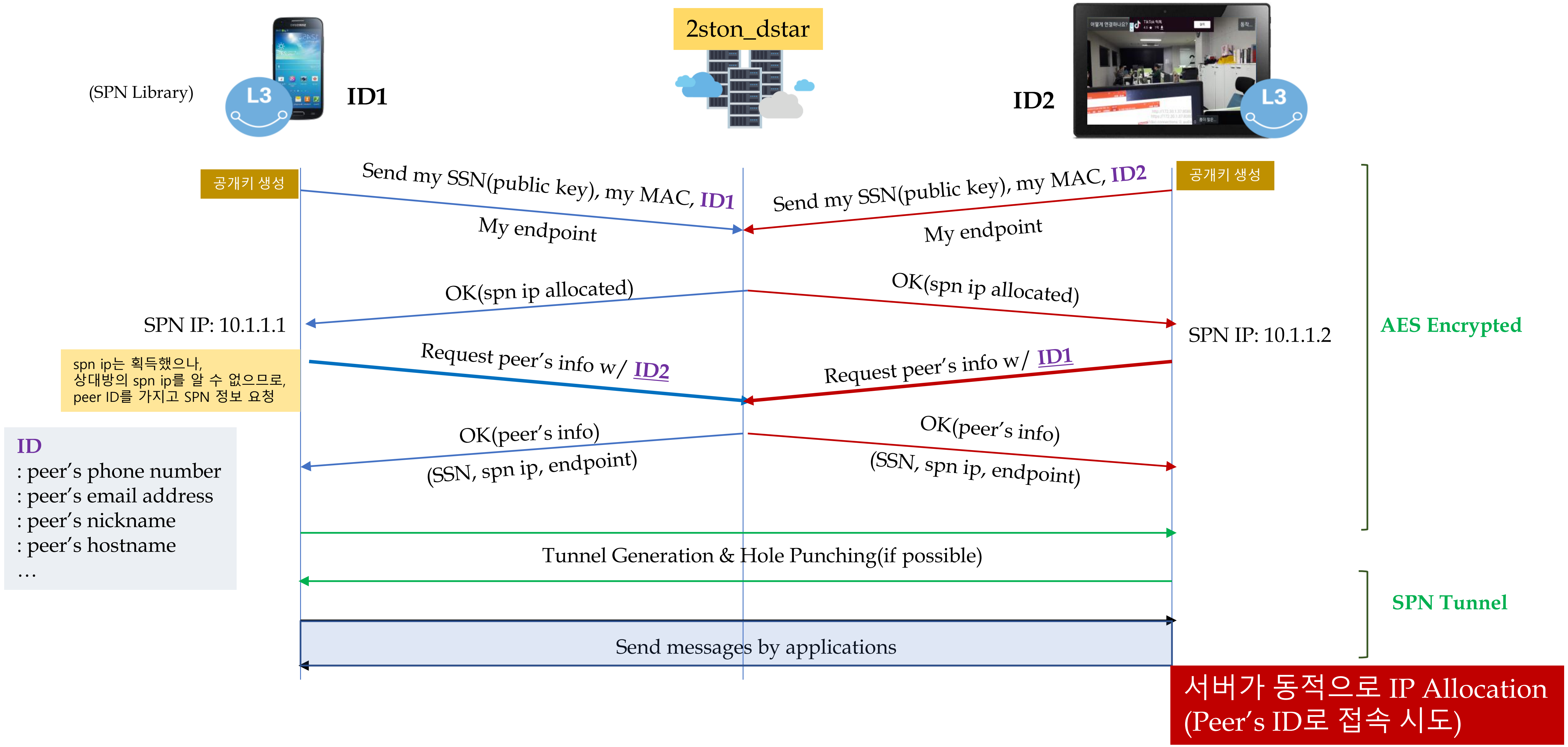
Full ConeNAT Only

4. DeathStar Daemon(1) – Auto Connection(2)

Client가 공개키를 생성하는 방식


2ston_xwing
2ston_dstar
2ston_xwing
L3
L3
SPN IP: 10.1.1
SPN IP: 10.1.1.2
공개키 생성
공개키 생성
Send my SSN(public key), spn ip
My endpoint , allowed ip info
Send my SSN(public key), spn ip
My endpoint, allowed ip info
OK
OK
Request peer's info w/ spn ip
Request peer's info w/ spn ip
OK(peer's info)
(SSN, endpoint, allowed ip)
OK(peer's info)
(SSN, endpoint, allowed ip)
UDP hole punching & Tunnel Generation
Send ping packets
AES Encrypted
SPN Tunnel

4. DeathStar Daemon(1) – Auto Connection(3)

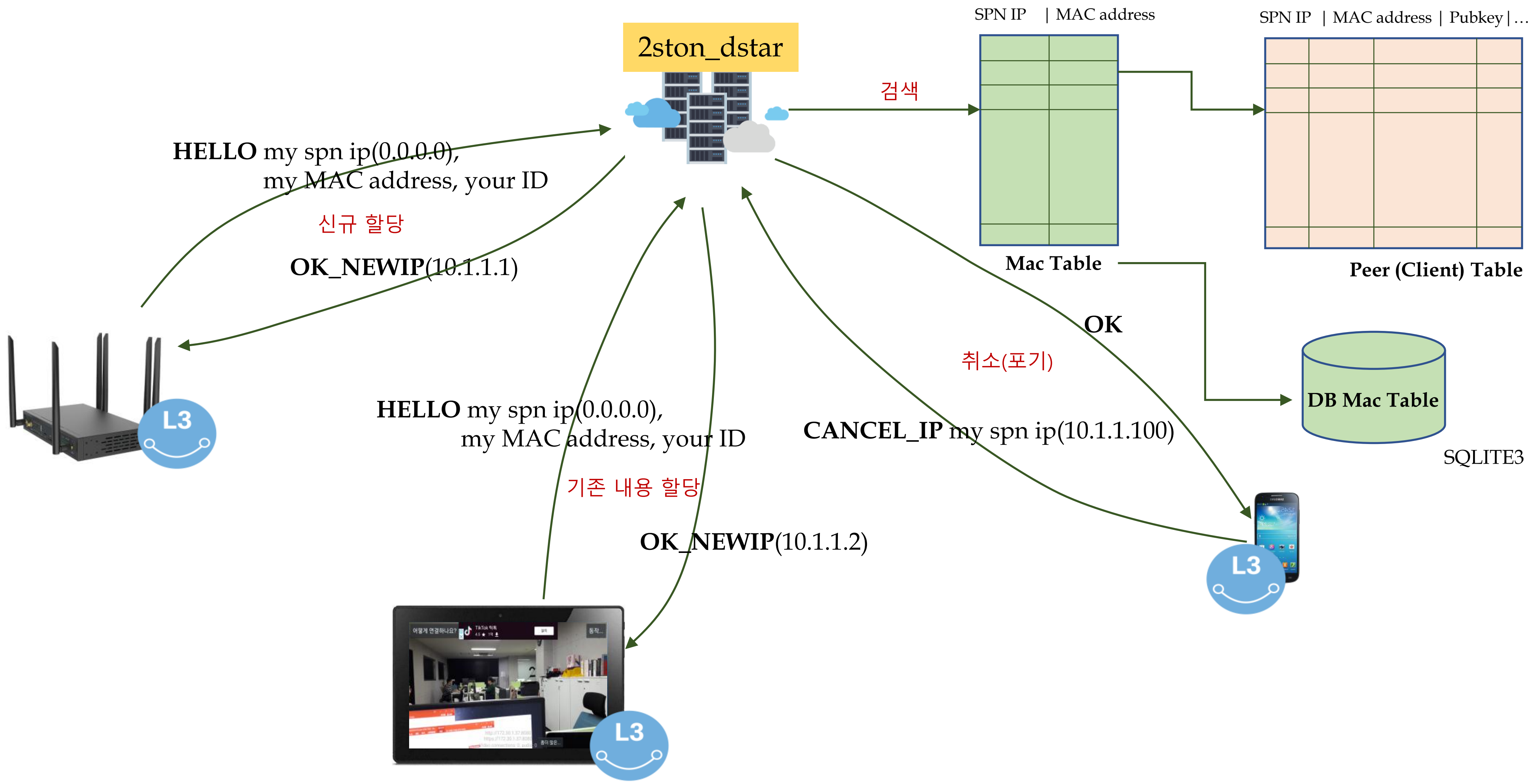


spn ip는 획득했으나,
상대방의 spn ip를 알 수 없으므로,
peer ID를 가지고 SPN 정보 요청

ID
: peer's phone number
: peer's email address
: peer's nickname
: peer's hostname
…

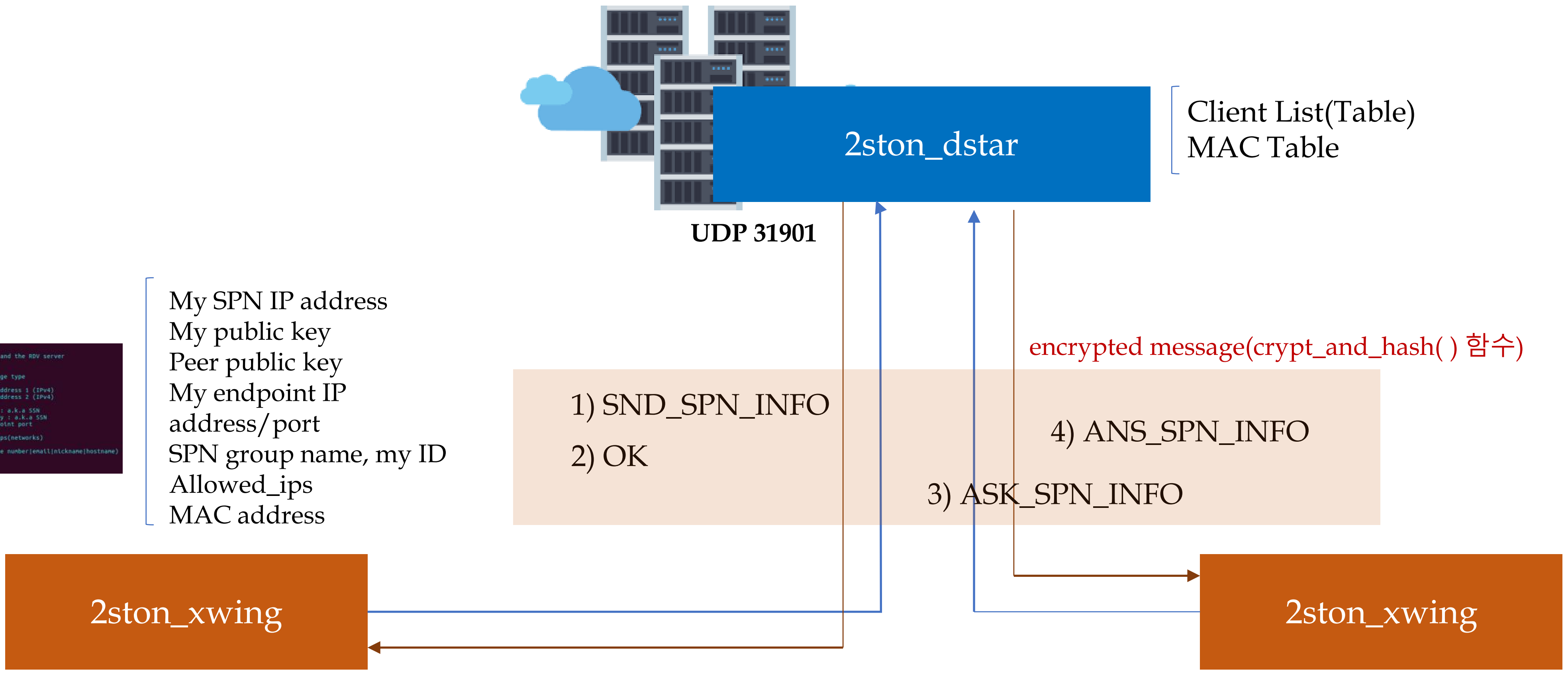
서버가 동적으로 IP Allocation
(Peer's ID로 접속 시도)

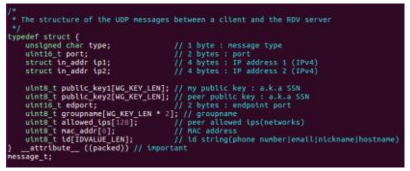
4. DeathStar Daemon(1) – Auto Connection(4)


2ston_dstar
검색
SPN IP | MAC address
SPN IP | MAC address | Pubkey | …
Mac Table
Peer (Client) Table
DB Mac Table
SQLITE3
HELLO my spn ip(0.0.0.0), my MAC address, your ID
신규 할당
OK_NEWIP(10.1.1.1)
HELLO my spn ip(0.0.0.0), my MAC address, your ID
기존 내용 할당
OK_NEWIP(10.1.1.2)
OK
취소(포기)
CANCEL_IP my spn ip(10.1.1.100)
L3
L3
L3

DHCP 처럼, 2ston_server가 SPN IP 주소를 자동으로 할당 & 관리해 줍니다.

4. DeathStar Daemon(2) – Communications(1)




```
/*
 * The structure of the UDP messages between a client and the RDV server
 */
typedef struct {
    unsigned char type;                 // 1 byte : message type
    uint16_t port;                      // 2 bytes : port
    struct in_addr ip1;                 // 4 bytes : IP address 1 (IPv4)
    struct in_addr ip2;                 // 4 bytes : IP address 2 (IPv4)

    uint8_t public_key1[WG_KEY_LEN]; // my public key : a.k.a SSN
    uint8_t public_key2[WG_KEY_LEN]; // peer public key : a.k.a SSN
    uint16_t edport;                    // 2 bytes : endpoint port
    uint8_t groupname[WG_KEY_LEN * 2]; // groupname
    uint8_t allowed_ips[128];           // peer allowed ips(networks)
    uint8_t mac_addr[6];                // MAC address
    uint8_t id[IDVALUE_LEN];            // id string(phone number|email|nickname|hostname)
}  __attribute__ ((packed)) // important
message_t;
```

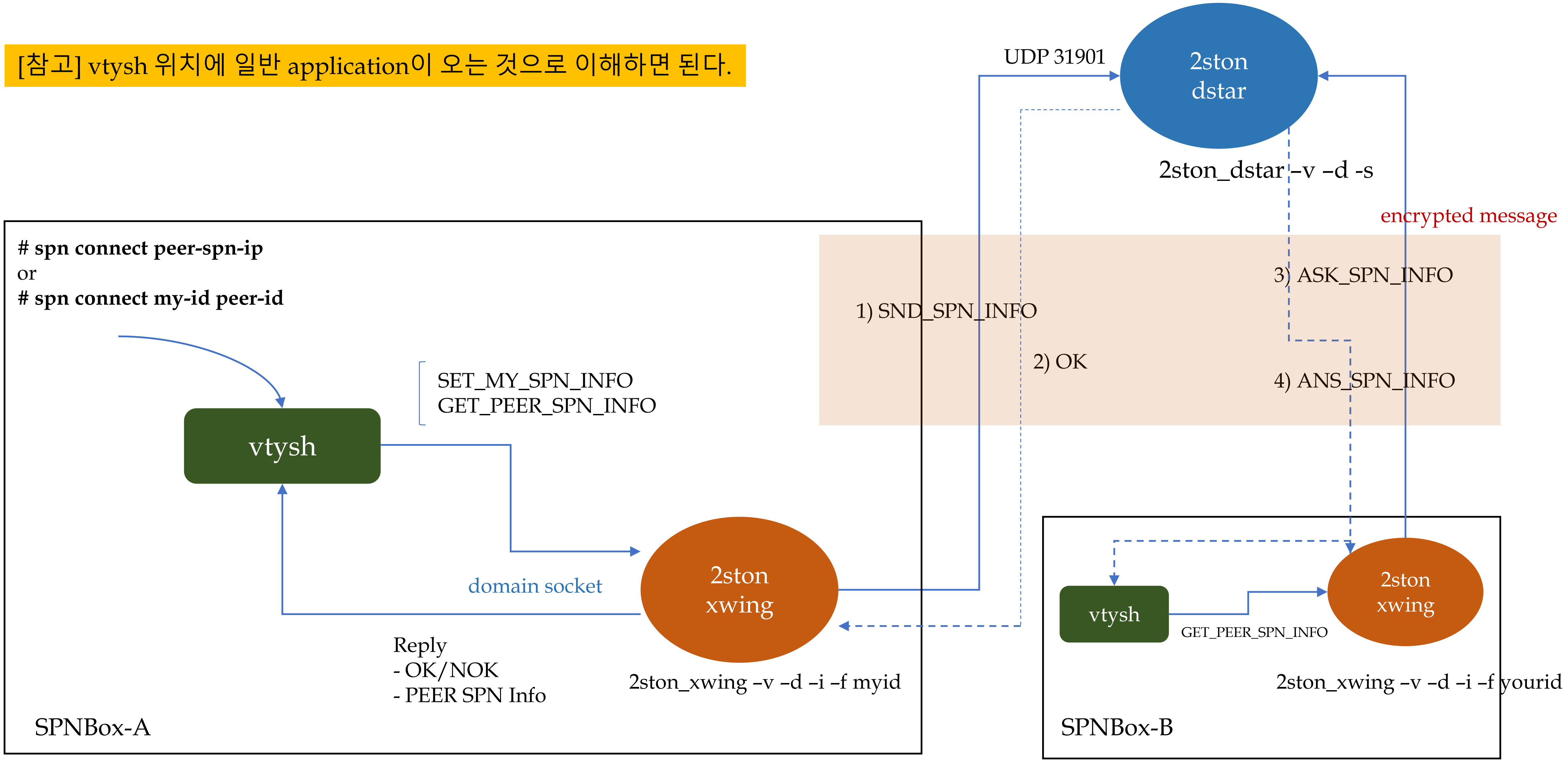
My SPN IP address
My public key
Peer public key
My endpoint IP address/port
SPN group name, my ID
Allowed_ips
MAC address

<u>명령 type 값</u>

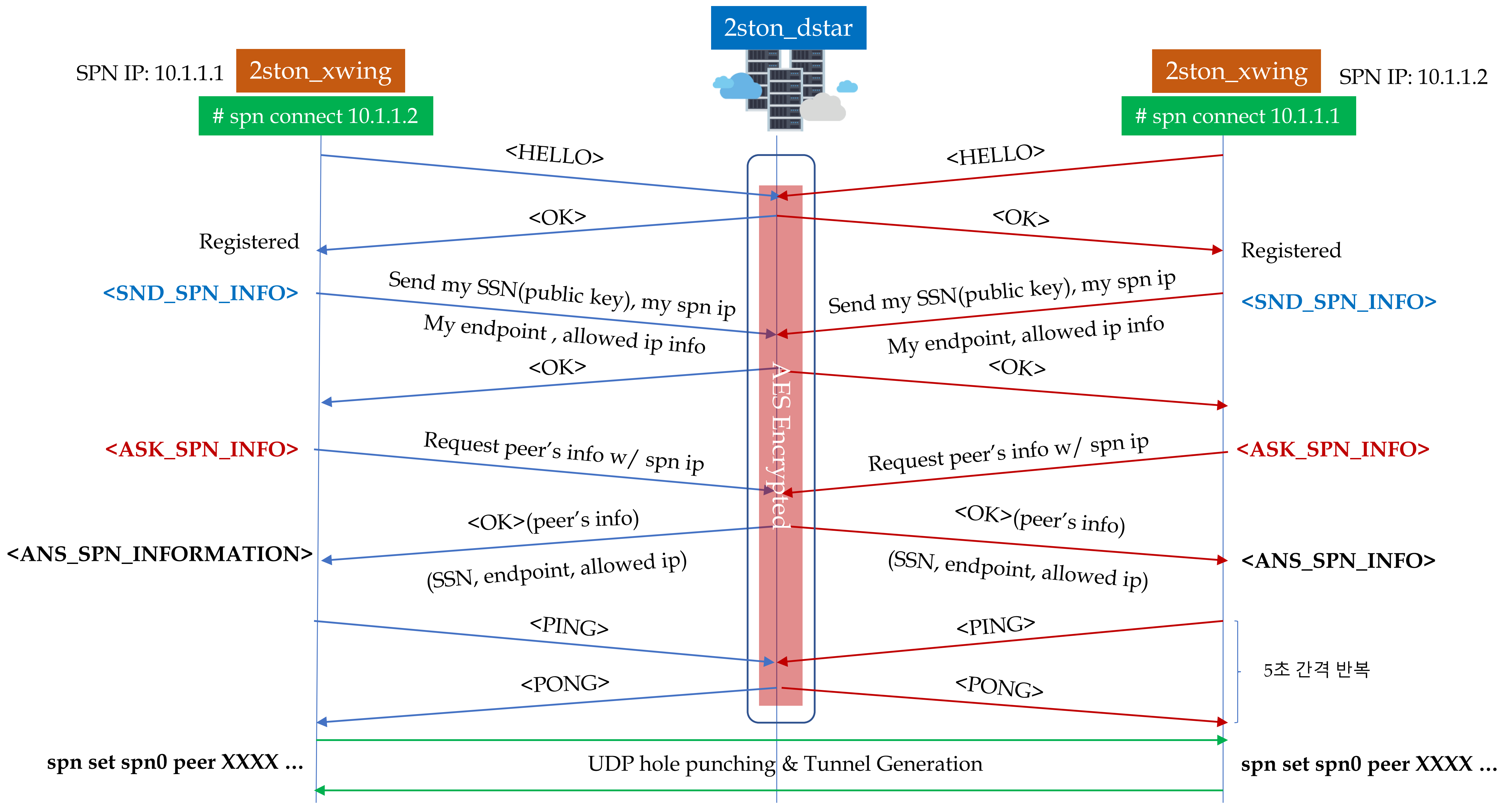
- ✓ HELLO ⇔ OK/NOK or OK_NEWIP/NOK : registration 시 or IP 할당 후 registration 시
- ✓ PING ⇔ PONG : 상태 정보 유지(갱신) 시
- ✓ SND_SPN_INFO ⇔ OK/NOK : my SPN 정보 등록 시
- ✓ ASK_SPN_INFO ⇔ ANS_SPN_INFORMATION : peer SPN 정보 요청 시
- ✓ BYE ⇔ [OK/NOK] : registration 해제 시(종료시)

4. DeathStar Daemon(2) – Communications(2)

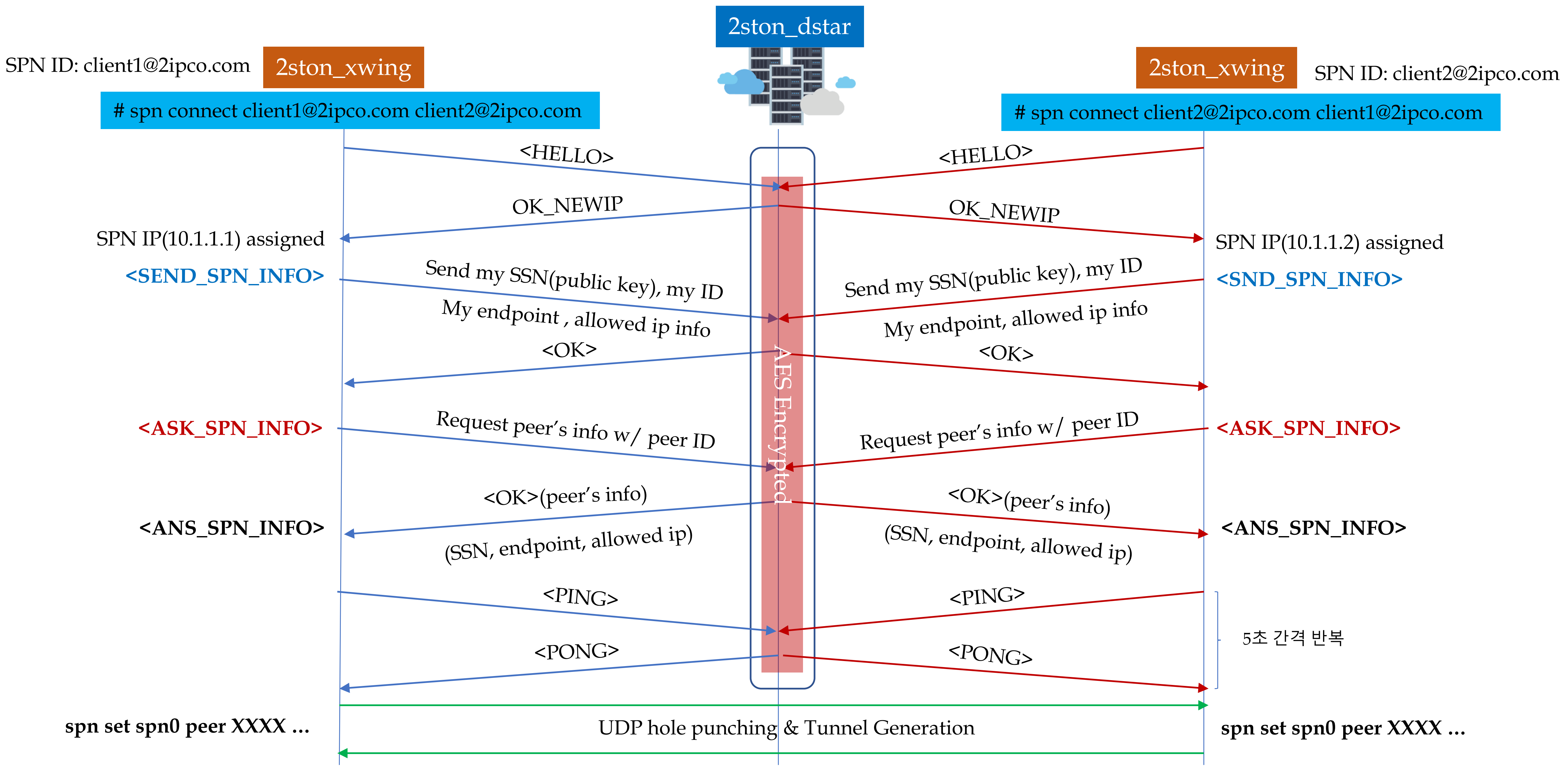
[참고] vtysh 위치에 일반 application이 오는 것으로 이해하면 된다.



4. DeathStar Daemon(3) – Protocol(1)

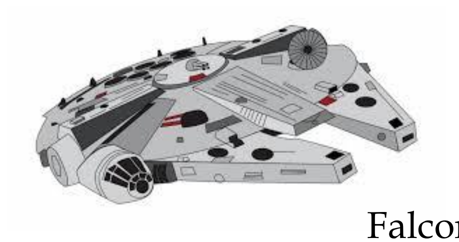


4. DeathStar Daemon(3) – Protocol(2)


2ston_dstar
SPN ID: client1@2ipco.com
2ston_xwing
# spn connect client1@2ipco.com client2@2ipco.com
2ston_xwing
SPN ID: client2@2ipco.com
# spn connect client2@2ipco.com client1@2ipco.com
<HELLO>
<HELLO>
OK_NEWIP
OK_NEWIP
SPN IP(10.1.1.1) assigned
SPN IP(10.1.1.2) assigned
<SEND_SPN_INFO>
Send my SSN(public key), my ID
My endpoint , allowed ip info
<SND_SPN_INFO>
Send my SSN(public key), my ID
My endpoint, allowed ip info
<OK>
<OK>
AES Encrypted
<ASK_SPN_INFO>
Request peer's info w/ peer ID
<ASK_SPN_INFO>
Request peer's info w/ peer ID
<ANS_SPN_INFO>
<OK>(peer's info)
(SSN, endpoint, allowed ip)
<ANS_SPN_INFO>
<OK>(peer's info)
(SSN, endpoint, allowed ip)
<PING>
<PING>
<PONG>
<PONG>
5초 간격 반복
spn set spn0 peer XXXX …
UDP hole punching & Tunnel Generation
spn set spn0 peer XXXX …

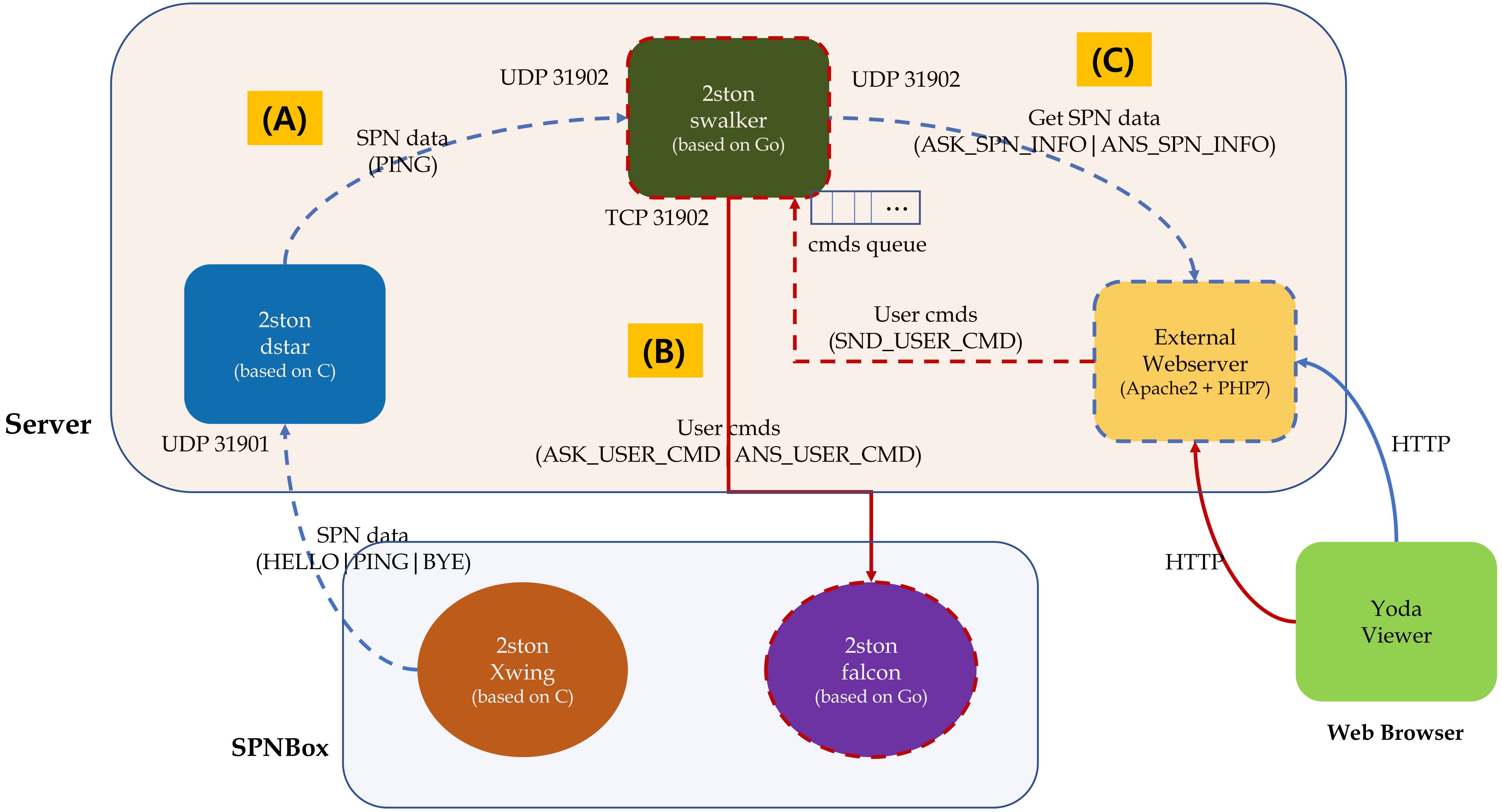
SkyWalker Daemon

Go

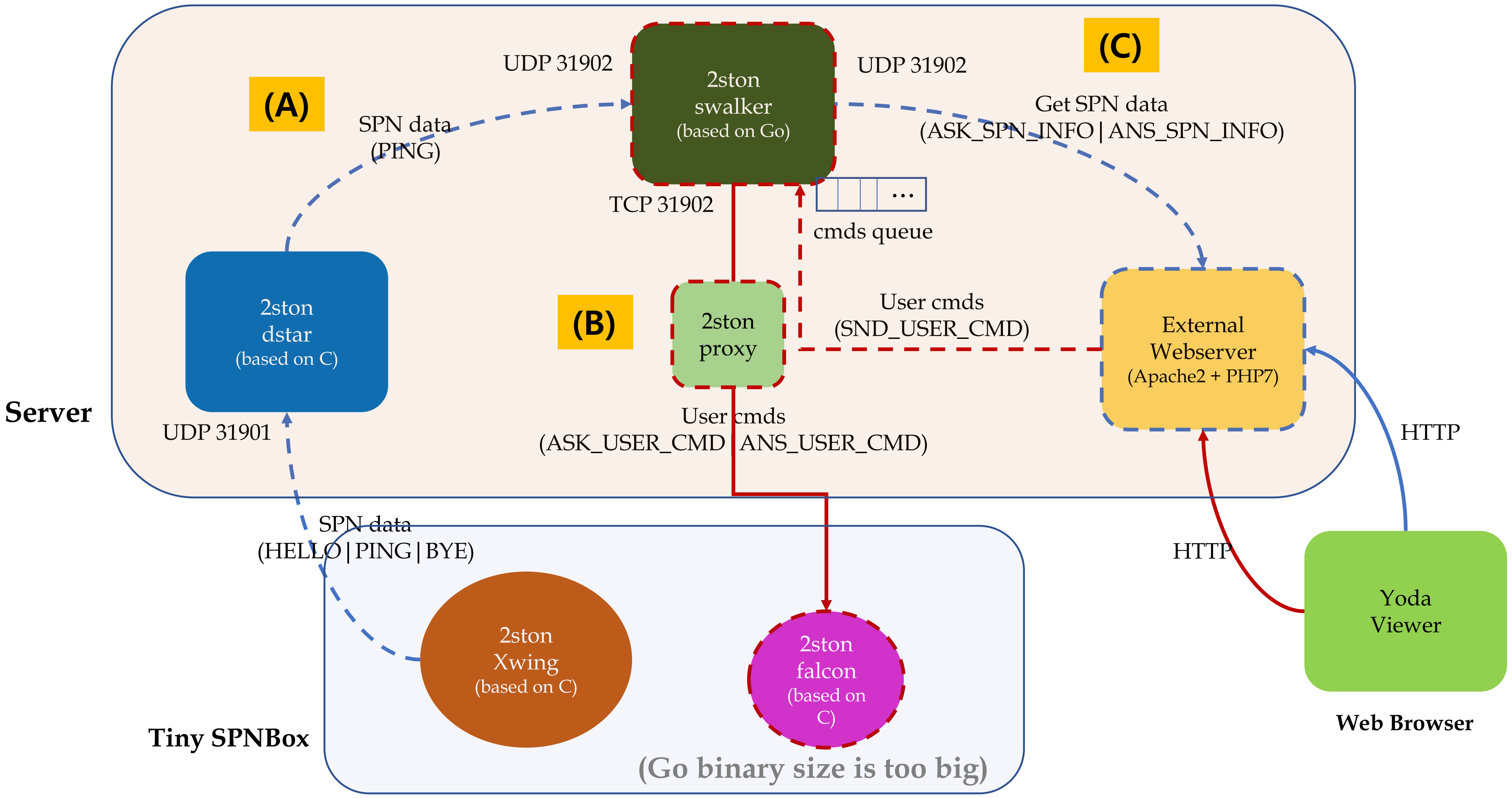

Falcon

SkyWalker

5. SkyWalker Daemon(1) – Architecture(1)



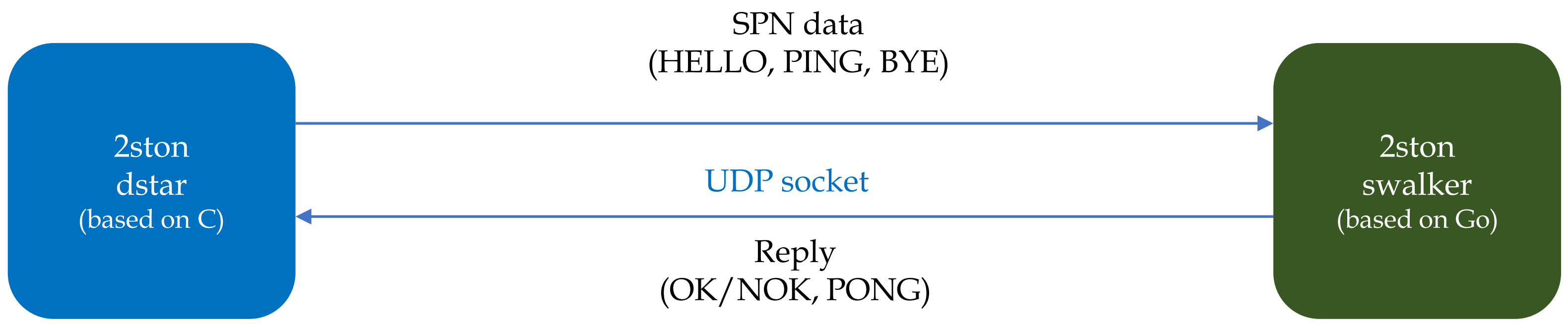
5. SkyWalker Daemon(1) – Architecture(2)



(A)
(B)
(C)
2ston swalker (based on Go)
UDP 31902
UDP 31902
SPN data (PING)
Get SPN data (ASK_SPN_INFO | ANS_SPN_INFO)
TCP 31902
cmds queue
…
2ston dstar (based on C)
UDP 31901
2ston proxy
User cmds (SND_USER_CMD)
External Webserver (Apache2 + PHP7)
User cmds (ASK_USER_CMD | ANS_USER_CMD)
Server
HTTP
HTTP
SPN data (HELLO | PING | BYE)
2ston Xwing (based on C)
2ston falcon (based on C)
Yoda Viewer
Web Browser
Tiny SPNBox
(Go binary size is too big)

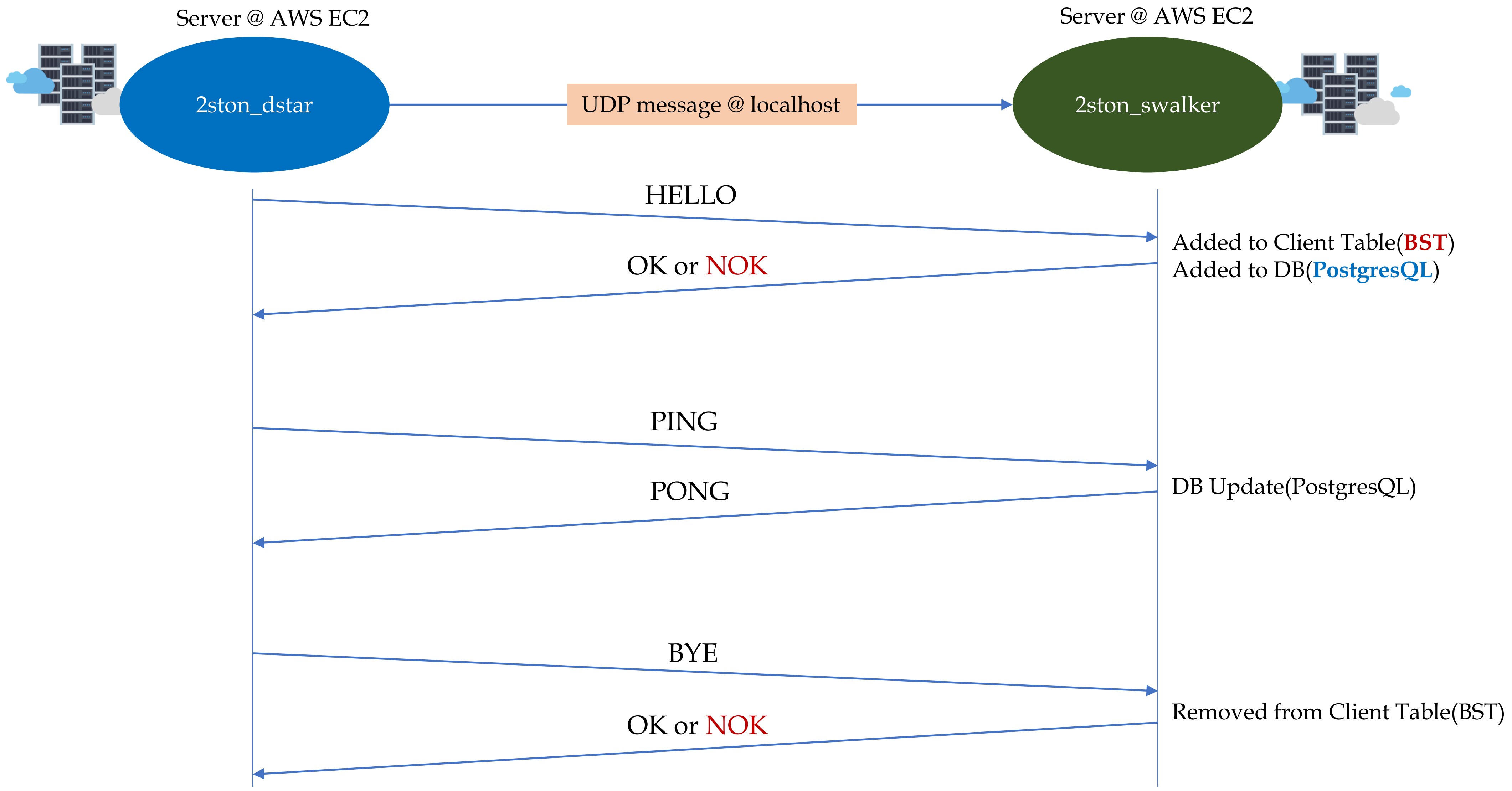
5. SkyWalker Daemon(2) – DSTAR interface(1)

(A)



Client(SPNBox) 2ston_xwing으로 부터 전달 받은 SPNBox 상태 정보를 2ston_swalker에게 그대로 넘긴다.

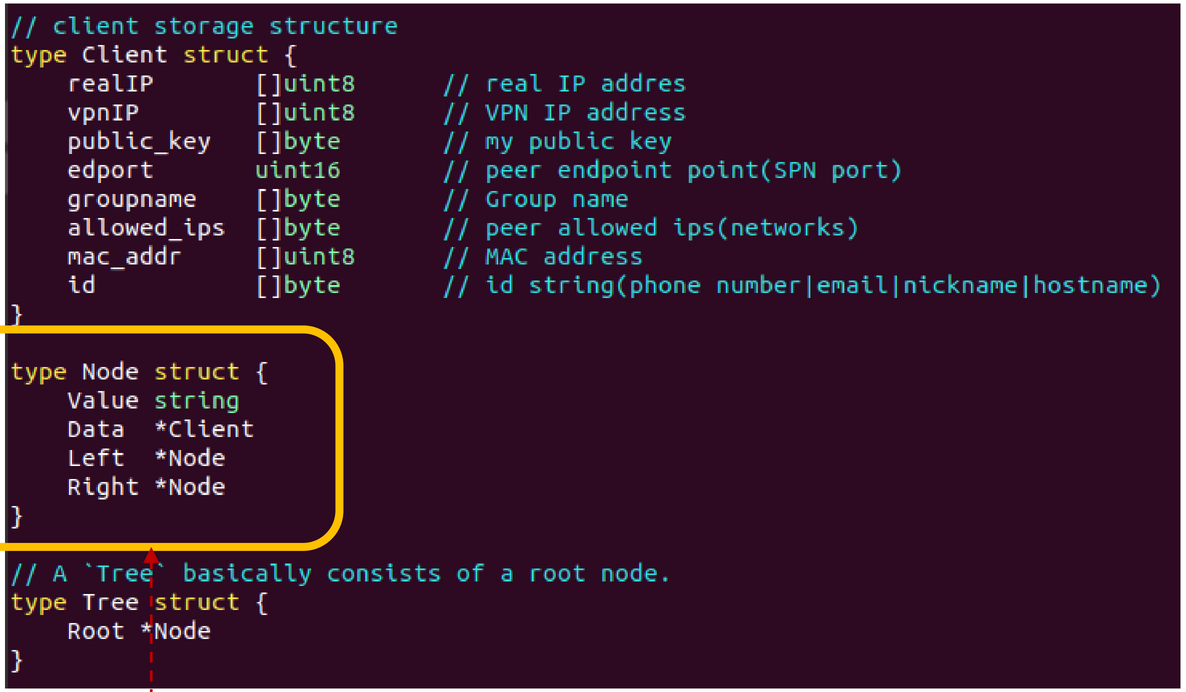
5. SkyWalker Daemon(2) – DSTAR interface(2)



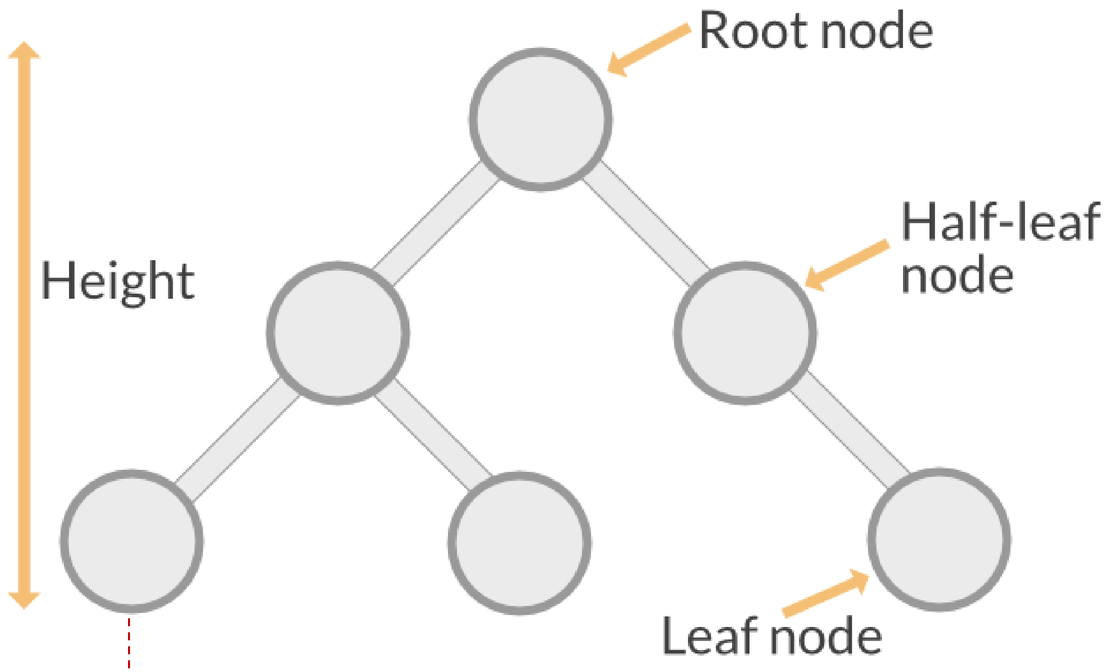
5. SkyWalker Daemon(2) – DSTAR interface(3)

Binary Search Tree

- Clients(SPNBoxes) 상태 정보 저장



```
// client storage structure
type Client struct {
    realIP        []uint8      // real IP addres
    vpnIP         []uint8      // VPN IP address
    public_key    []byte       // my public key
    edport        uint16       // peer endpoint point(SPN port)
    groupname     []byte       // Group name
    allowed_ips   []byte       // peer allowed ips(networks)
    mac_addr      []uint8      // MAC address
    id            []byte       // id string(phone number|email|nickname|hostname)
}

type Node struct {
    Value string
    Data  *Client
    Left  *Node
    Right *Node
}

// A `Tree` basically consists of a root node.
type Tree struct {
    Root *Node
}
```



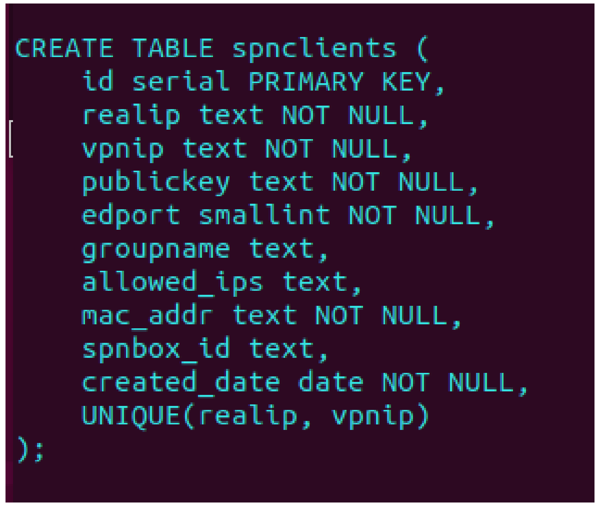
Binary Search Tree

- Value(검색 key 값): vpnIP string
- Data(실제 data) : Client struct

5. SkyWalker Daemon(2) – DSTAR interface(4)

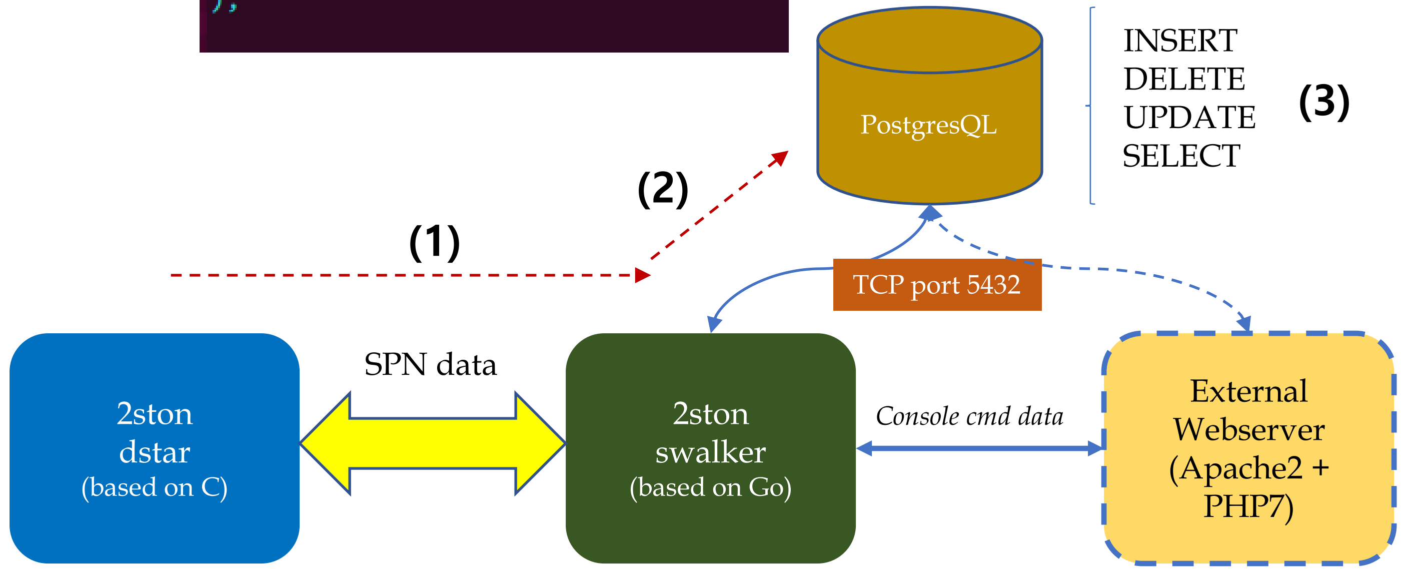
PostgresQL DBMS

- Clients(SPNBoxes) 상태 정보 저장

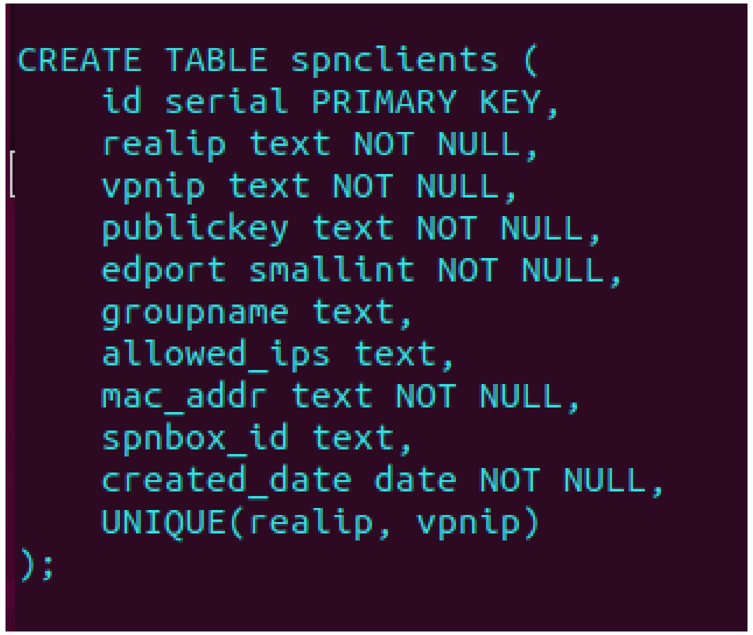


```
CREATE TABLE spnclients (
    id serial PRIMARY KEY,
    realip text NOT NULL,
    vpnip text NOT NULL,
    publickey text NOT NULL,
    edport smallint NOT NULL,
    groupname text,
    allowed_ips text,
    mac_addr text NOT NULL,
    spnbox_id text,
    created_date date NOT NULL,
    UNIQUE(realip, vpnip)
);
```

spnclients PostgresQL DB table

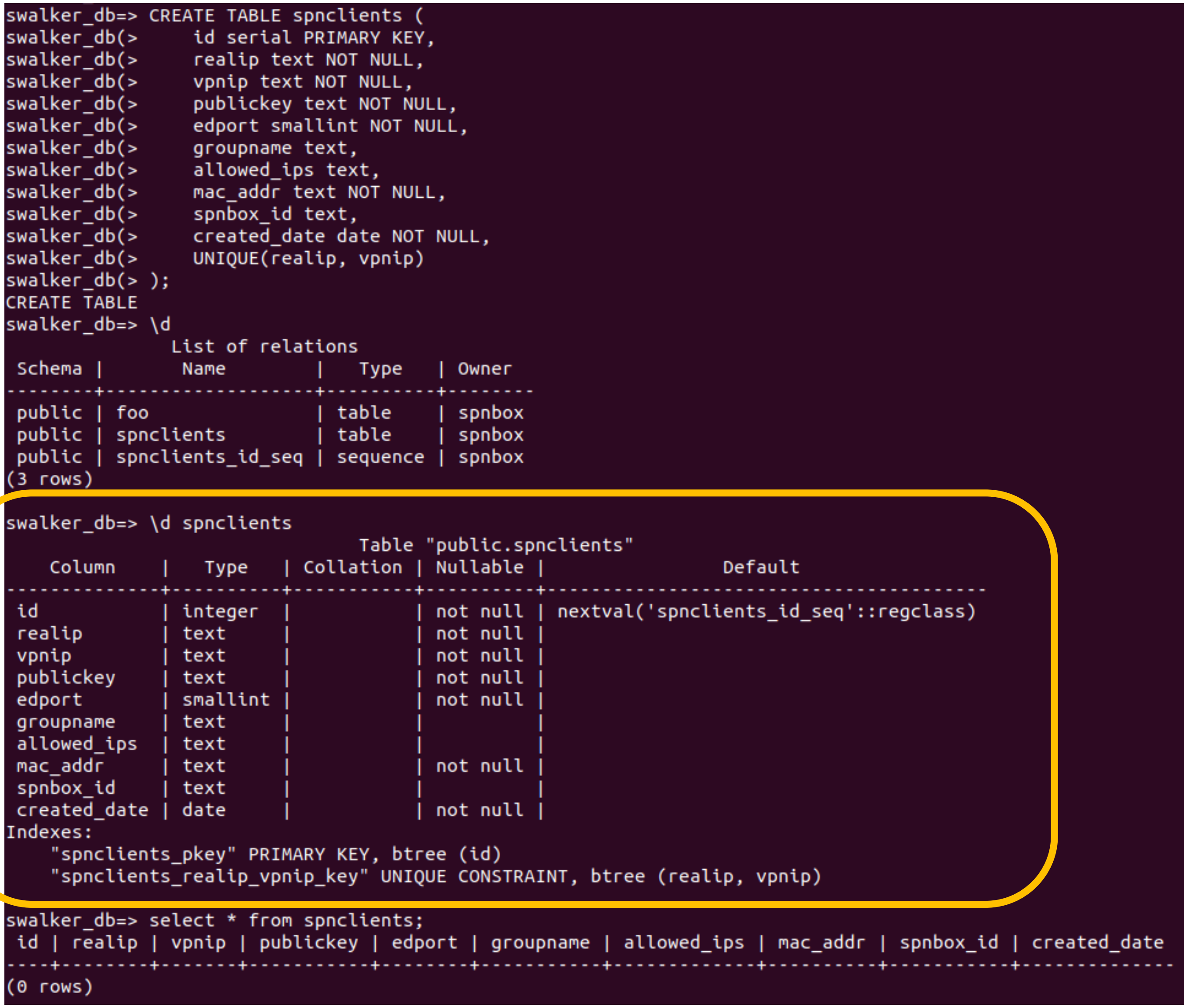


5. SkyWalker Daemon(3) – PostgresQL DBMS(1)



```
CREATE TABLE spnclients (
    id serial PRIMARY KEY,
    realip text NOT NULL,
    vpnip text NOT NULL,
    publickey text NOT NULL,
    edport smallint NOT NULL,
    groupname text,
    allowed_ips text,
    mac_addr text NOT NULL,
    spnbox_id text,
    created_date date NOT NULL,
    UNIQUE(realip, vpnip)
);
```

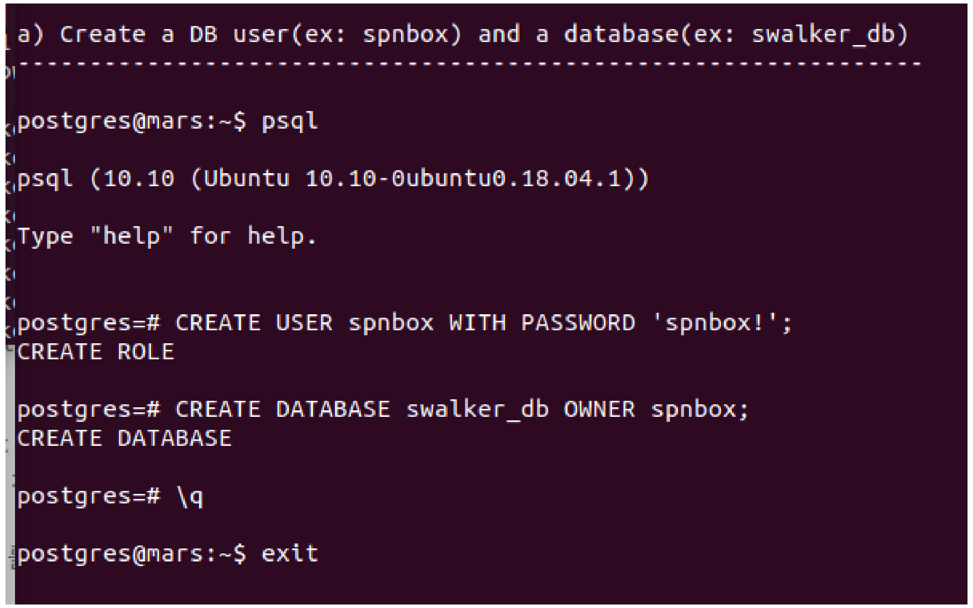
spnclients PostgresQL DB table



```
swalker_db=> CREATE TABLE spnclients (
swalker_db(>     id serial PRIMARY KEY,
swalker_db(>     realip text NOT NULL,
swalker_db(>     vpnip text NOT NULL,
swalker_db(>     publickey text NOT NULL,
swalker_db(>     edport smallint NOT NULL,
swalker_db(>     groupname text,
swalker_db(>     allowed_ips text,
swalker_db(>     mac_addr text NOT NULL,
swalker_db(>     spnbox_id text,
swalker_db(>     created_date date NOT NULL,
swalker_db(>     UNIQUE(realip, vpnip)
swalker_db(> );
CREATE TABLE
swalker_db=> \d
               List of relations
 Schema |        Name        |   Type   | Owner
--------+--------------------+----------+--------
 public | foo                | table    | spnbox
 public | spnclients         | table    | spnbox
 public | spnclients_id_seq  | sequence | spnbox
(3 rows)

swalker_db=> \d spnclients
                                   Table "public.spnclients"
    Column    |   Type   | Collation | Nullable |                Default
--------------+----------+-----------+----------+----------------------------------------
 id           | integer  |           | not null | nextval('spnclients_id_seq'::regclass)
 realip       | text     |           | not null |
 vpnip        | text     |           | not null |
 publickey    | text     |           | not null |
 edport       | smallint |           | not null |
 groupname    | text     |           |          |
 allowed_ips  | text     |           |          |
 mac_addr     | text     |           | not null |
 spnbox_id    | text     |           |          |
 created_date | date     |           | not null |
Indexes:
    "spnclients_pkey" PRIMARY KEY, btree (id)
    "spnclients_realip_vpnip_key" UNIQUE CONSTRAINT, btree (realip, vpnip)

swalker_db=> select * from spnclients;
 id | realip | vpnip | publickey | edport | groupname | allowed_ips | mac_addr | spnbox_id | created_date
----+--------+-------+-----------+--------+-----------+-------------+----------+-----------+--------------
(0 rows)
```

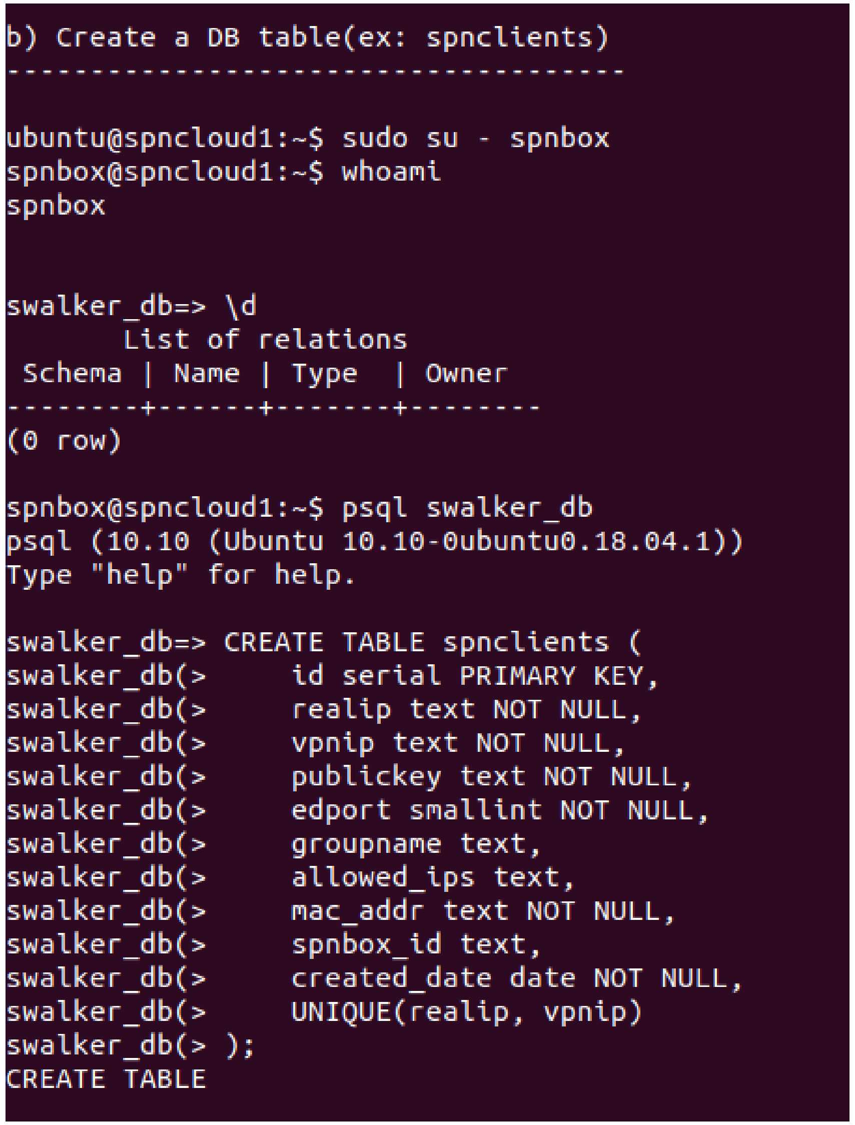
$ psql swalker_db

5. SkyWalker Daemon(3) – PostgresQL DBMS(2)



```
a) Create a DB user(ex: spnbox) and a database(ex: swalker_db)
---------------------------------------------------------------

postgres@mars:~$ psql

psql (10.10 (Ubuntu 10.10-0ubuntu0.18.04.1))

Type "help" for help.


postgres=# CREATE USER spnbox WITH PASSWORD 'spnbox!';
CREATE ROLE

postgres=# CREATE DATABASE swalker_db OWNER spnbox;
CREATE DATABASE

postgres=# \q

postgres@mars:~$ exit
```

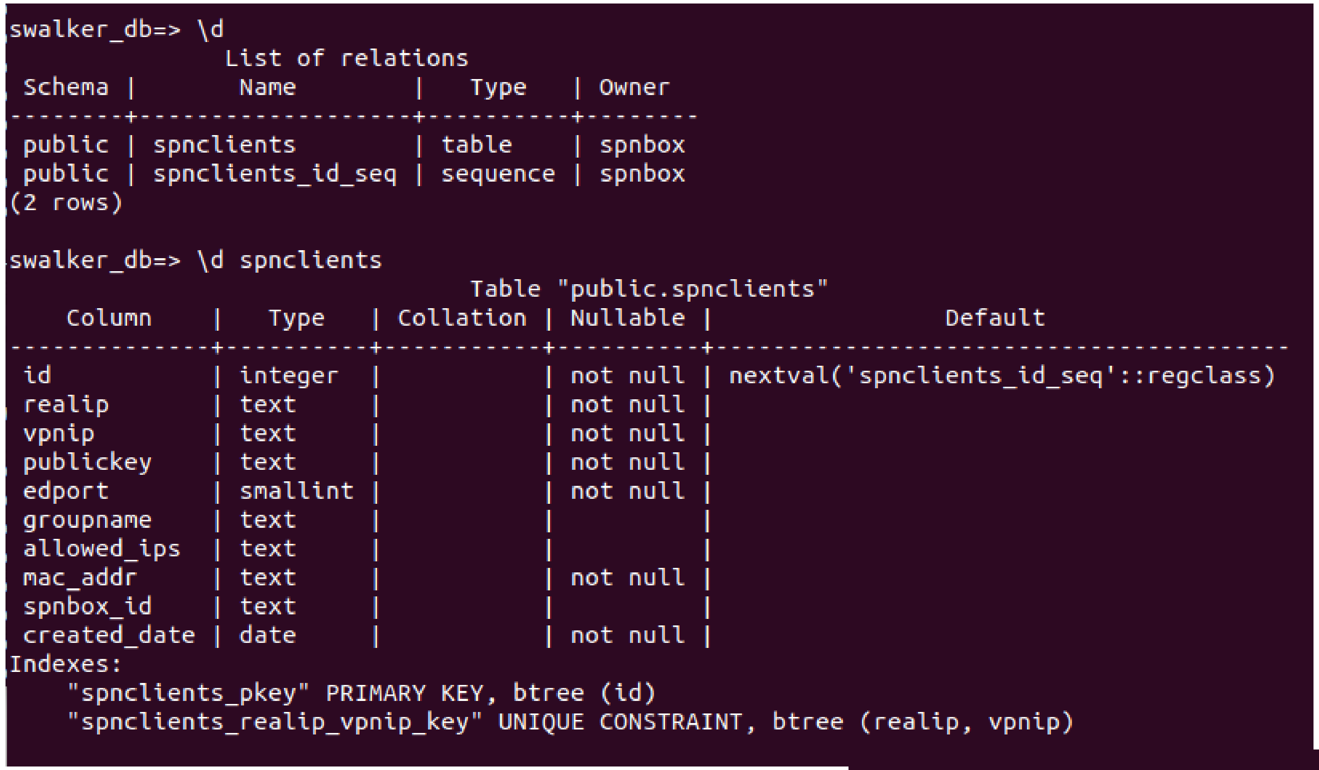
[1] PostgresQL DB user(spnbox) 추가 및 DB(swalker_db) 추가



```
b) Create a DB table(ex: spnclients)
-------------------------------------

ubuntu@spncloud1:~$ sudo su - spnbox
spnbox@spncloud1:~$ whoami
spnbox


swalker_db=> \d
       List of relations
 Schema | Name | Type  | Owner
--------+------+-------+--------
(0 row)

spnbox@spncloud1:~$ psql swalker_db
psql (10.10 (Ubuntu 10.10-0ubuntu0.18.04.1))
Type "help" for help.

swalker_db=> CREATE TABLE spnclients (
swalker_db(>     id serial PRIMARY KEY,
swalker_db(>     realip text NOT NULL,
swalker_db(>     vpnip text NOT NULL,
swalker_db(>     publickey text NOT NULL,
swalker_db(>     edport smallint NOT NULL,
swalker_db(>     groupname text,
swalker_db(>     allowed_ips text,
swalker_db(>     mac_addr text NOT NULL,
swalker_db(>     spnbox_id text,
swalker_db(>     created_date date NOT NULL,
swalker_db(>     UNIQUE(realip, vpnip)
swalker_db(> );
CREATE TABLE
```

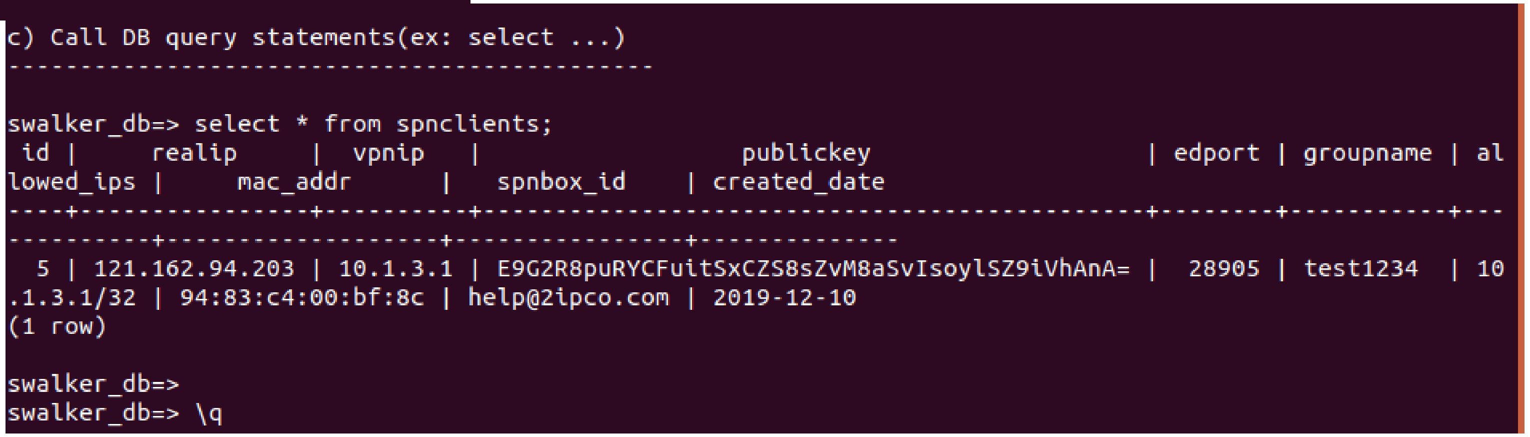
[2] PostgresQL DB Table(spnclients) 추가

5. SkyWalker Daemon(3) – PostgresQL DBMS(3)



```
swalker_db=> \d
               List of relations
 Schema |        Name         |   Type   | Owner
--------+---------------------+----------+--------
 public | spnclients          | table    | spnbox
 public | spnclients_id_seq   | sequence | spnbox
(2 rows)

swalker_db=> \d spnclients
                                 Table "public.spnclients"
   Column     |   Type   | Collation | Nullable |                Default
--------------+----------+-----------+----------+----------------------------------------
 id           | integer  |           | not null | nextval('spnclients_id_seq'::regclass)
 realip       | text     |           | not null |
 vpnip        | text     |           | not null |
 publickey    | text     |           | not null |
 edport       | smallint |           | not null |
 groupname    | text     |           |          |
 allowed_ips  | text     |           |          |
 mac_addr     | text     |           | not null |
 spnbox_id    | text     |           |          |
 created_date | date     |           | not null |
Indexes:
    "spnclients_pkey" PRIMARY KEY, btree (id)
    "spnclients_realip_vpnip_key" UNIQUE CONSTRAINT, btree (realip, vpnip)
```

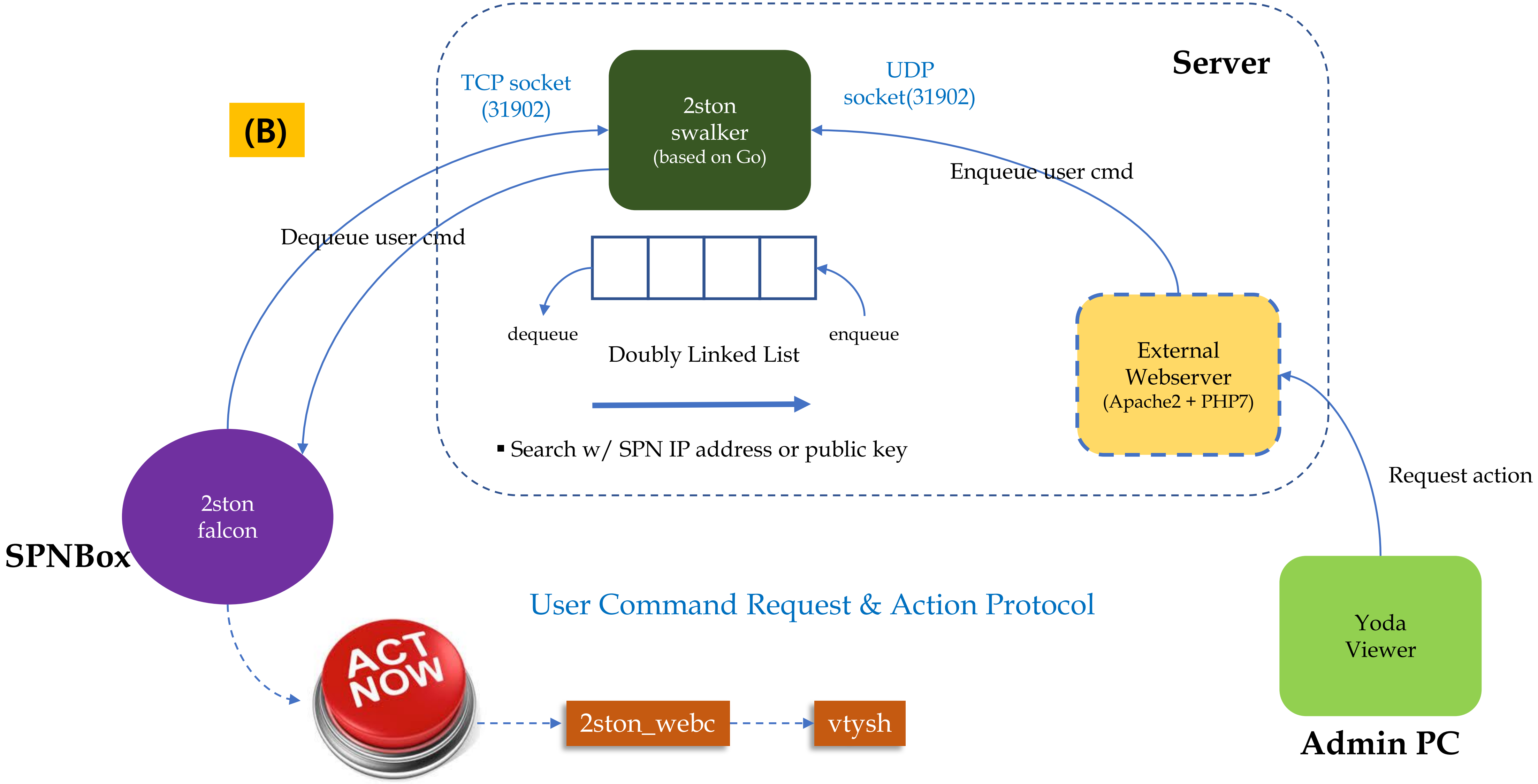
[3] PostgresQL DB Table(spnclients) 내용 확인



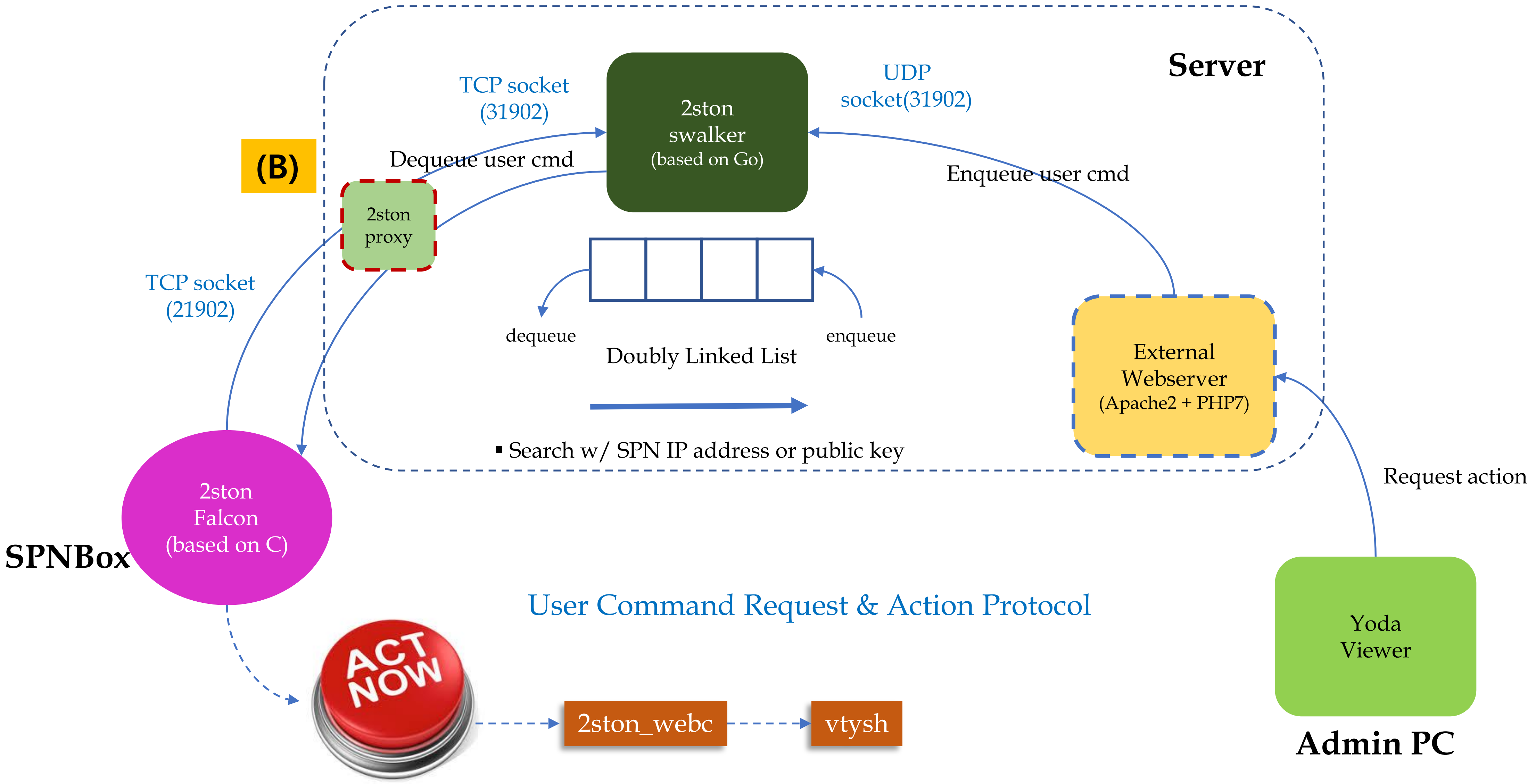
```
c) Call DB query statements(ex: select ...)
---------------------------------------------

swalker_db=> select * from spnclients;
 id |     realip     |  vpnip   |                  publickey                   | edport | groupname | al
lowed_ips |     mac_addr      |   spnbox_id    | created_date
----+----------------+----------+----------------------------------------------+--------+-----------+---
----------+-------------------+----------------+--------------
  5 | 121.162.94.203 | 10.1.3.1 | E9G2R8puRYCFuitSxCZS8sZvM8aSvIsoylSZ9iVhAnA= |  28905 | test1234  | 10
.1.3.1/32 | 94:83:c4:00:bf:8c | help@2ipco.com | 2019-12-10
(1 row)

swalker_db=>
swalker_db=> \q
```

[4] PostgresQL DB Table 검색(select 명령 실행)

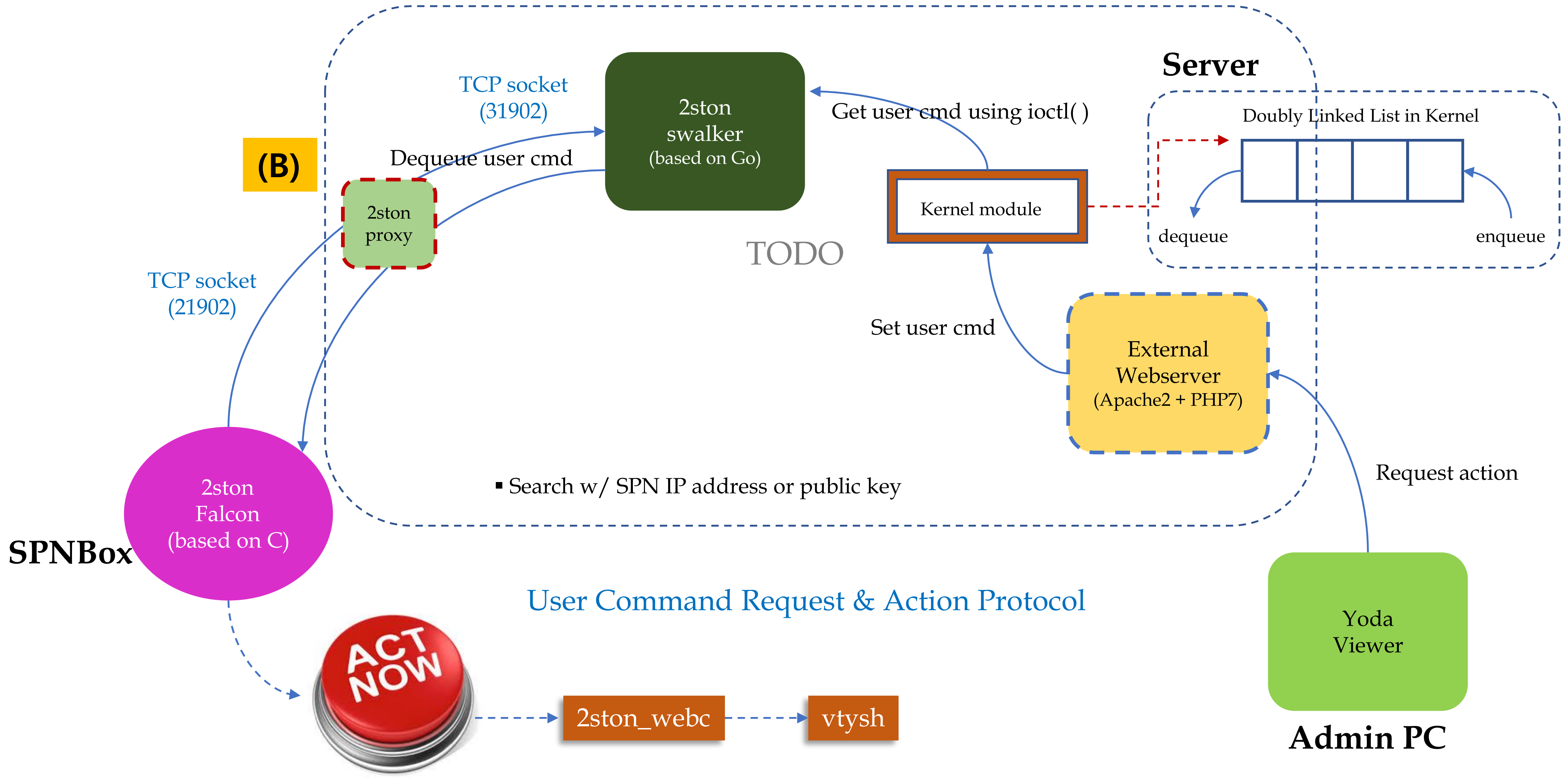
5. SkyWalker Daemon(4) – FALCON Interface(1-1)



5. SkyWalker Daemon(4) – FALCON Interface(1-2)



5. SkyWalker Daemon(4) – FALCON Interface(1-3)



5. SkyWalker Daemon(4) – FALCON Interface(2)

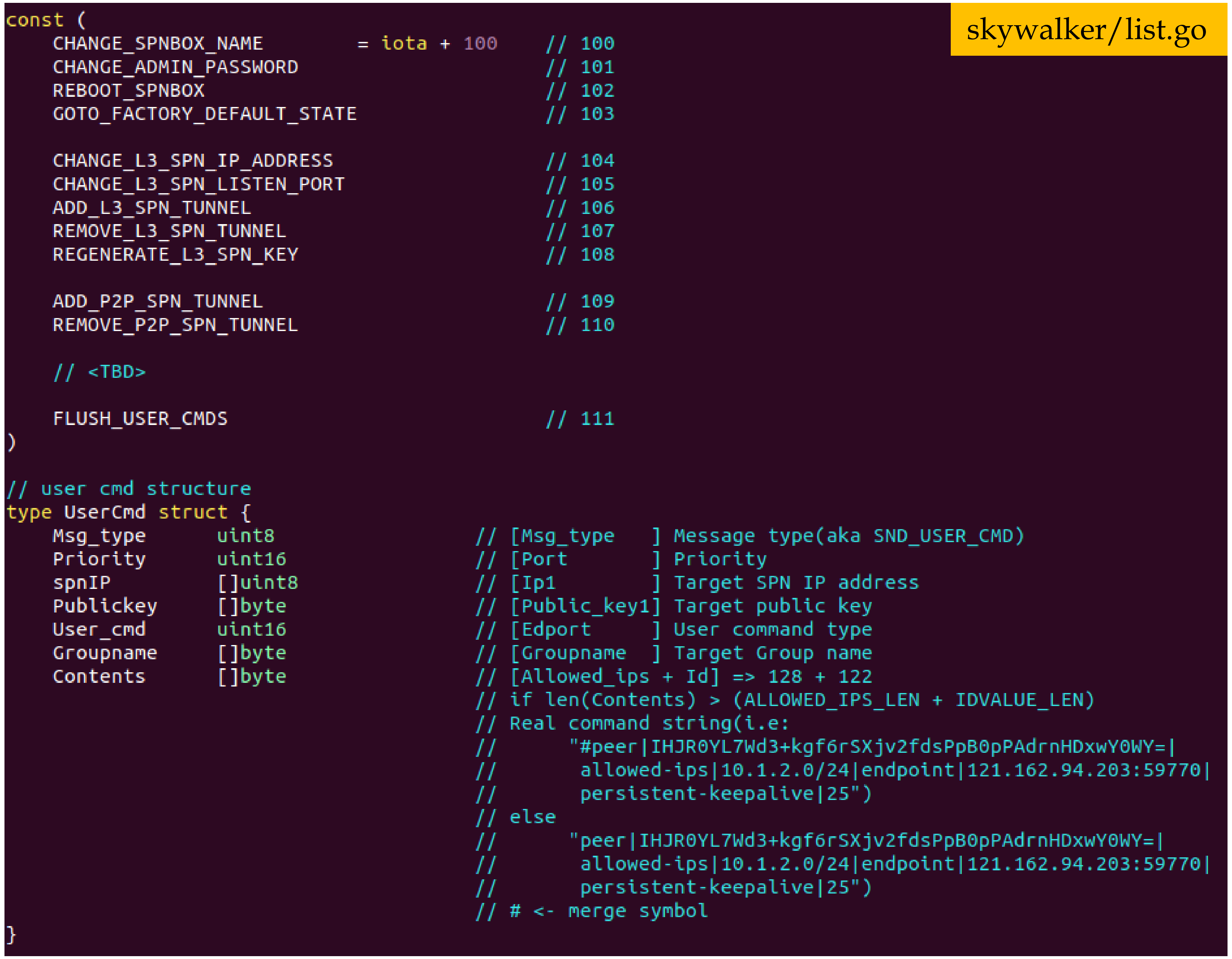
<시나리오>

- [1] swalker <=> dstar | falcon | webserver 간의 통신 message format을 일치시킨다(정확히 말하면 비슷하게 가져간다).
  - ✓ swalker는 UDP 31902 port를 열고, dstar로 부터는 client message를 수신한다. 또한 webserver로 부터는 같은 포트를 통해 user command message를 수신한다.
  - ✓ 한편, falcon과는 보다 안전한 user cmd 전달을 위해 UDP 대신 TCP(31902 port)를 활용한다.
- [2] swalker는 webserver로 부터 UserCmd message를 받아서 list(queue) buffer에 추가 후, list(doubly linked list)에 넣어 준다.
  - ✓ list에 넣을 때는 중복 정보가 있더라도, 신경쓰지 말고 무조건 넣는다(마지막에 추가한다).
  - ✓ 우선 순위가 높은 User Cmd는 list의 맨 앞에 위치시킨다.
- [3] falcon으로 부터의 user cmd 요청 시, list를 검색하여 SPN ip가 일치하는 녀석을 찾아낸 후, falcon에 응답한다.
  - ✓ [TBD] 이때 spnIP가 0.0.0.0(ALL 명령)인 명령도 자신의 것으로 간주한다.
  - ✓ [TBD] 0.0.0.0 ip 명령을 제거하기 위해서는 falcon(SPNBox)의 개수를 관리할 수 있어야 한다.

5. SkyWalker Daemon(4) – FALCON Interface(3)

User Cmd

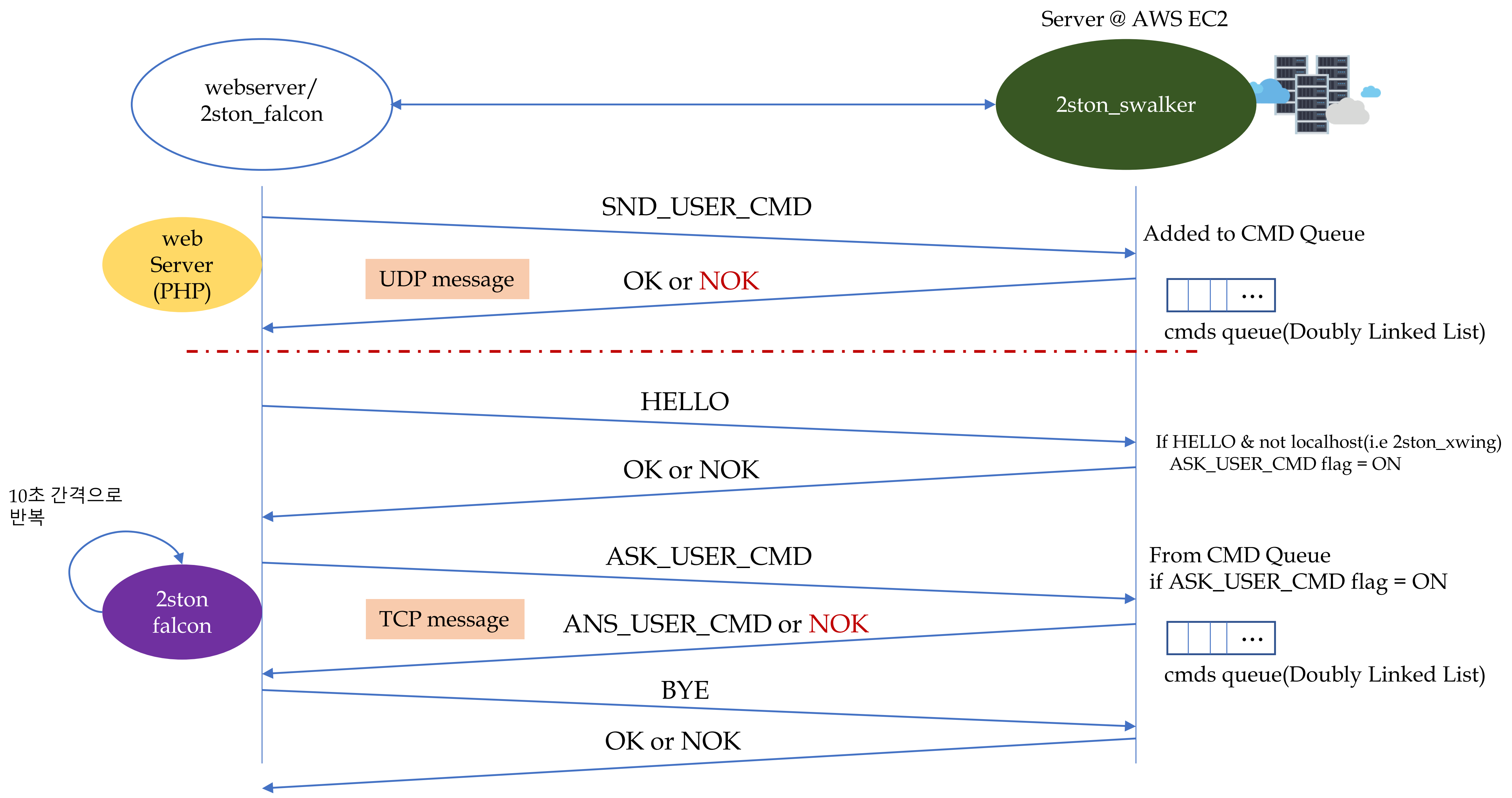
Structure for User Cmd



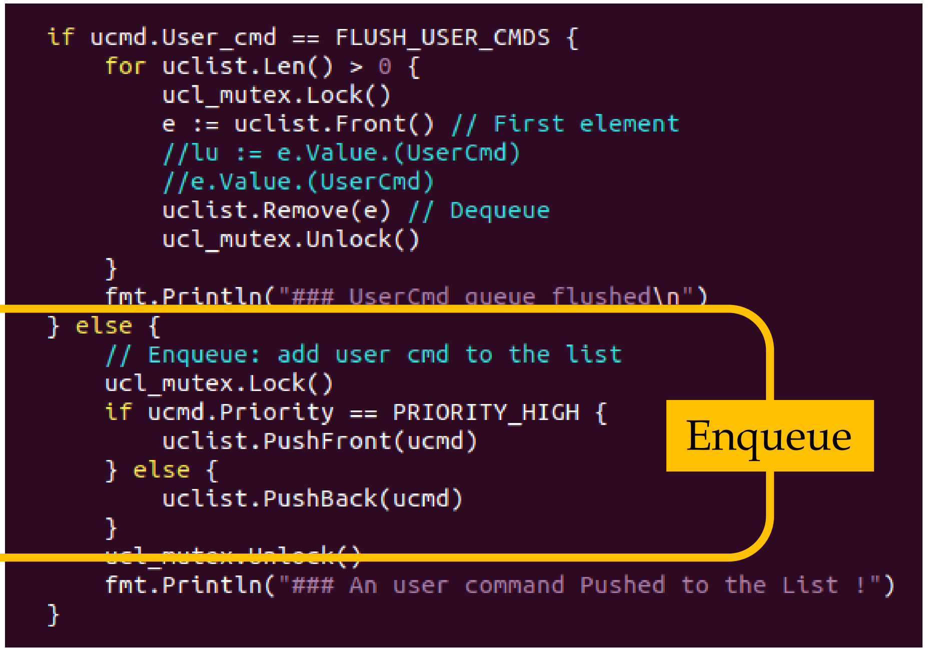
skywalker/list.go

```
const (
    CHANGE_SPNBOX_NAME        = iota + 100    // 100
    CHANGE_ADMIN_PASSWORD                     // 101
    REBOOT_SPNBOX                             // 102
    GOTO_FACTORY_DEFAULT_STATE                // 103

    CHANGE_L3_SPN_IP_ADDRESS                  // 104
    CHANGE_L3_SPN_LISTEN_PORT                 // 105
    ADD_L3_SPN_TUNNEL                         // 106
    REMOVE_L3_SPN_TUNNEL                      // 107
    REGENERATE_L3_SPN_KEY                     // 108

    ADD_P2P_SPN_TUNNEL                        // 109
    REMOVE_P2P_SPN_TUNNEL                     // 110

    // <TBD>

    FLUSH_USER_CMDS                           // 111
)

// user cmd structure
type UserCmd struct {
    Msg_type      uint8                 // [Msg_type    ] Message type(aka SND_USER_CMD)
    Priority      uint16                // [Port        ] Priority
    spnIP         []uint8               // [Ip1         ] Target SPN IP address
    Publickey     []byte                // [Public_key1] Target public key
    User_cmd      uint16                // [Edport      ] User command type
    Groupname     []byte                // [Groupname   ] Target Group name
    Contents      []byte                // [Allowed_ips + Id] => 128 + 122
                                        // if len(Contents) > (ALLOWED_IPS_LEN + IDVALUE_LEN)
                                        // Real command string(i.e:
                                        //      "#peer|IHJR0YL7Wd3+kgf6rSXjv2fdsPpB0pPAdrnHDxwY0WY=|
                                        //       allowed-ips|10.1.2.0/24|endpoint|121.162.94.203:59770|
                                        //       persistent-keepalive|25")
                                        // else
                                        //      "peer|IHJR0YL7Wd3+kgf6rSXjv2fdsPpB0pPAdrnHDxwY0WY=|
                                        //       allowed-ips|10.1.2.0/24|endpoint|121.162.94.203:59770|
                                        //       persistent-keepalive|25")
                                        // # <- merge symbol
}
```

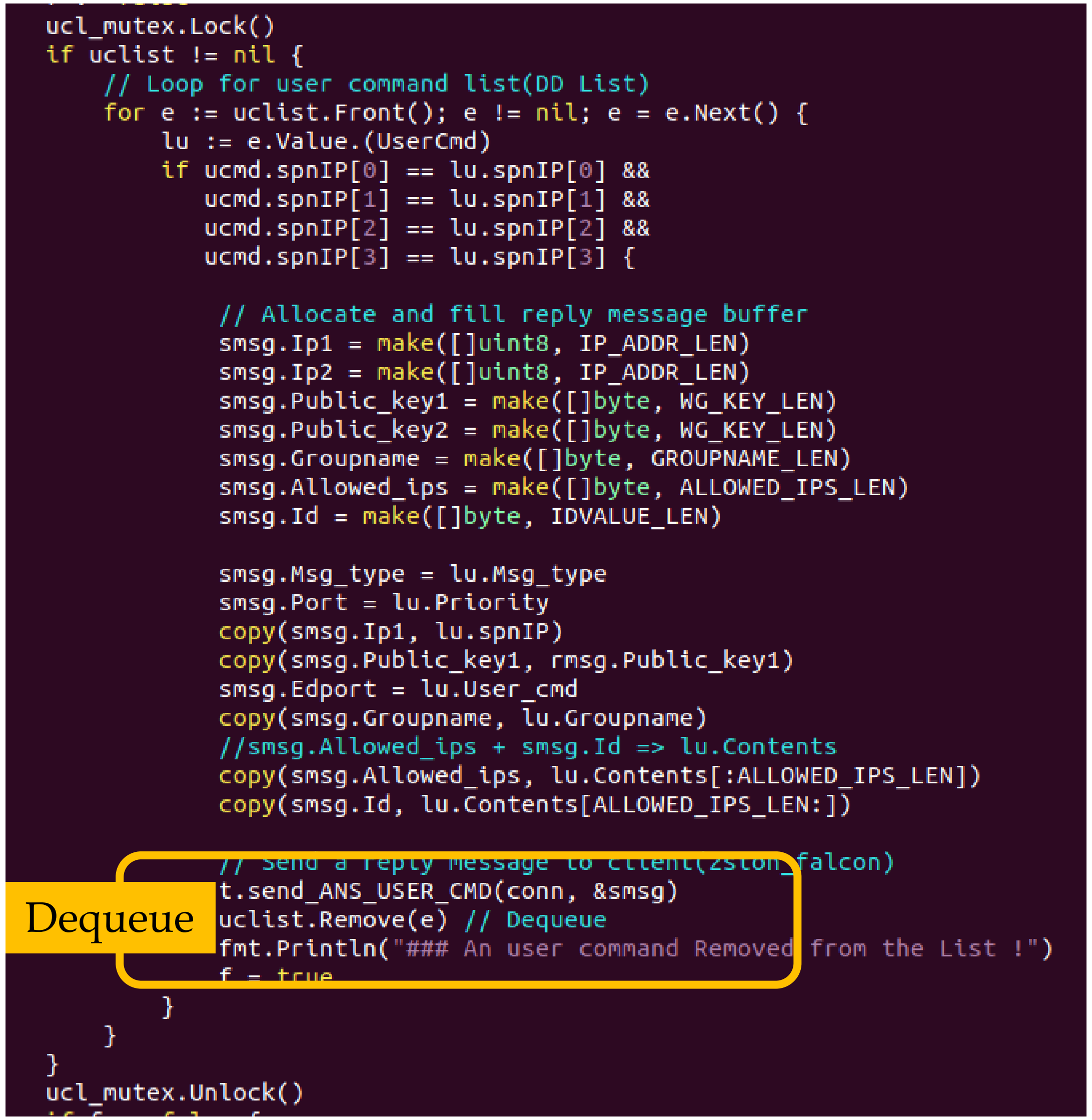
5. SkyWalker Daemon(4) – FALCON Interface(4)



5. SkyWalker Daemon(4) – FALCON Interface(5)

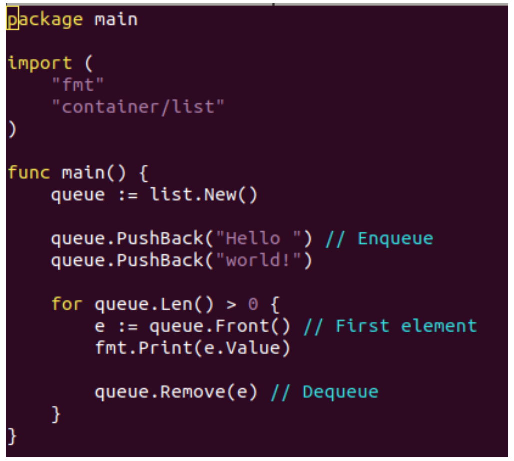


```
if ucmd.User_cmd == FLUSH_USER_CMDS {
    for uclist.Len() > 0 {
        ucl_mutex.Lock()
        e := uclist.Front() // First element
        //lu := e.Value.(UserCmd)
        //e.Value.(UserCmd)
        uclist.Remove(e) // Dequeue
        ucl_mutex.Unlock()
    }
    fmt.Println("### UserCmd queue flushed\n")
} else {
    // Enqueue: add user cmd to the list
    ucl_mutex.Lock()
    if ucmd.Priority == PRIORITY_HIGH {
        uclist.PushFront(ucmd)
    } else {
        uclist.PushBack(ucmd)
    }
    ucl_mutex.Unlock()
    fmt.Println("### An user command Pushed to the List !")
}
```



```
ucl_mutex.Lock()
if uclist != nil {
    // Loop for user command list(DD List)
    for e := uclist.Front(); e != nil; e = e.Next() {
        lu := e.Value.(UserCmd)
        if ucmd.spnIP[0] == lu.spnIP[0] &&
           ucmd.spnIP[1] == lu.spnIP[1] &&
           ucmd.spnIP[2] == lu.spnIP[2] &&
           ucmd.spnIP[3] == lu.spnIP[3] {

            // Allocate and fill reply message buffer
            smsg.Ip1 = make([]uint8, IP_ADDR_LEN)
            smsg.Ip2 = make([]uint8, IP_ADDR_LEN)
            smsg.Public_key1 = make([]byte, WG_KEY_LEN)
            smsg.Public_key2 = make([]byte, WG_KEY_LEN)
            smsg.Groupname = make([]byte, GROUPNAME_LEN)
            smsg.Allowed_ips = make([]byte, ALLOWED_IPS_LEN)
            smsg.Id = make([]byte, IDVALUE_LEN)

            smsg.Msg_type = lu.Msg_type
            smsg.Port = lu.Priority
            copy(smsg.Ip1, lu.spnIP)
            copy(smsg.Public_key1, rmsg.Public_key1)
            smsg.Edport = lu.User_cmd
            copy(smsg.Groupname, lu.Groupname)
            //smsg.Allowed_ips + smsg.Id => lu.Contents
            copy(smsg.Allowed_ips, lu.Contents[:ALLOWED_IPS_LEN])
            copy(smsg.Id, lu.Contents[ALLOWED_IPS_LEN:])

            // Send a reply message to client(2ston_falcon)
            t.send_ANS_USER_CMD(conn, &smsg)
            uclist.Remove(e) // Dequeue
            fmt.Println("### An user command Removed from the List !")
            f = true
        }
    }
}
ucl_mutex.Unlock()
```

Doubly Linked List example =>



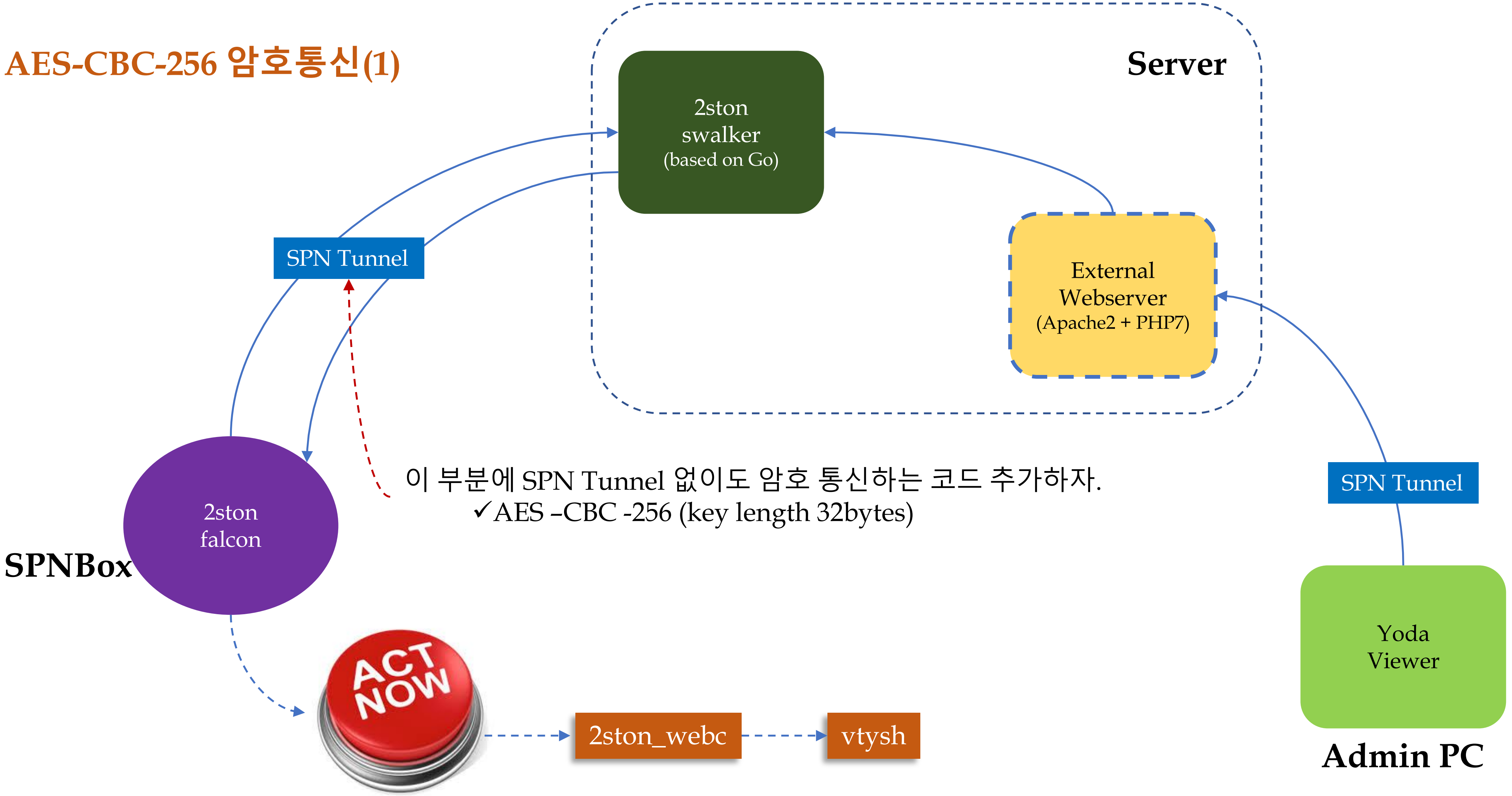
```
package main

import (
    "fmt"
    "container/list"
)

func main() {
    queue := list.New()

    queue.PushBack("Hello ") // Enqueue
    queue.PushBack("world!")

    for queue.Len() > 0 {
        e := queue.Front() // First element
        fmt.Print(e.Value)

        queue.Remove(e) // Dequeue
    }
}
```

User Cmd Queue

- 사용자 명령 저장

5. SkyWalker Daemon(4) – FALCON Interface(6-1)

AES-CBC-256 암호통신(1)

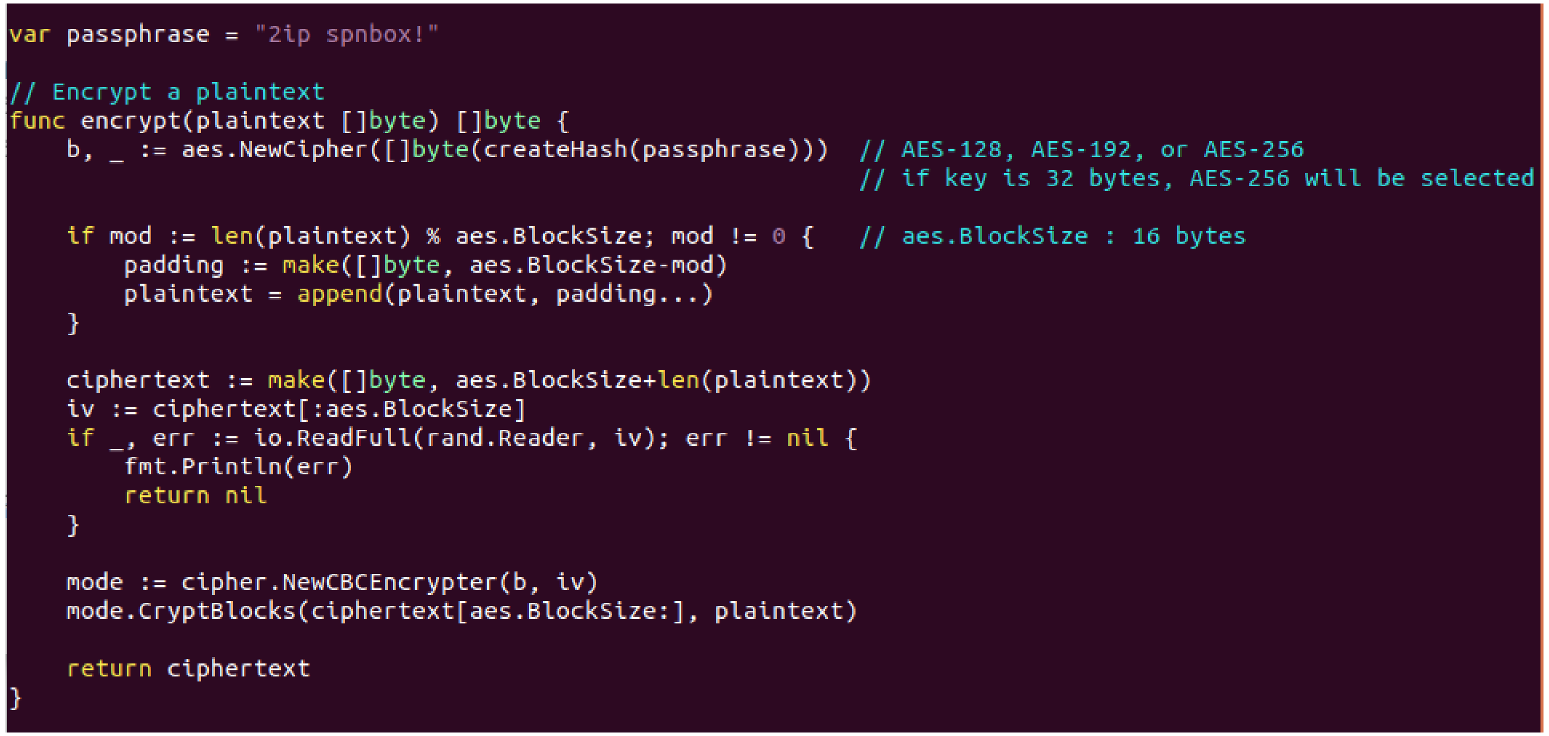


이 부분에 SPN Tunnel 없이도 암호 통신하는 코드 추가하자.

✓AES –CBC -256 (key length 32bytes)

5. SkyWalker Daemon(4) – FALCON Interface(6-2)

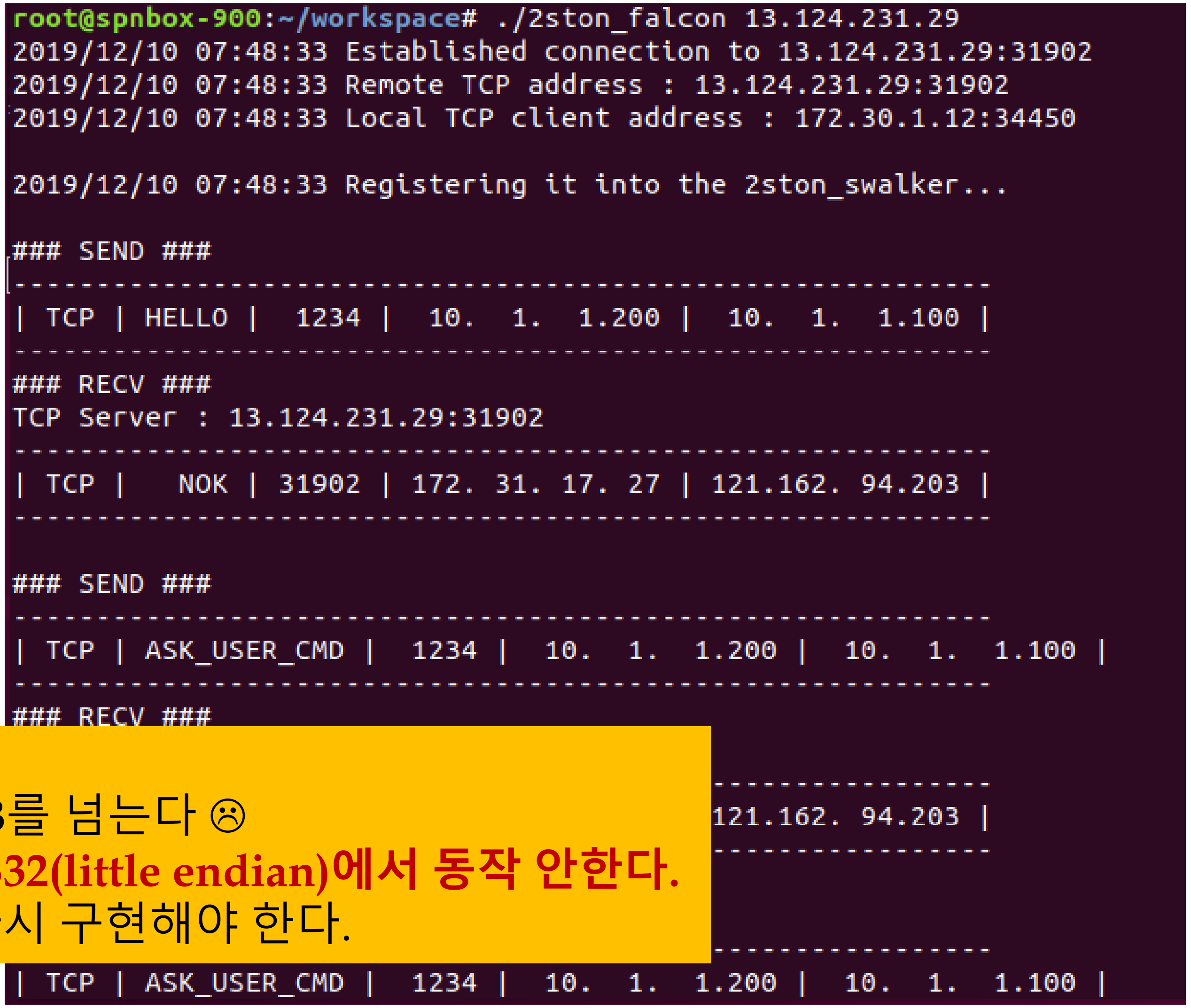
AES-CBC-256 암호통신(2)



```
var passphrase = "2ip spnbox!"

// Encrypt a plaintext
func encrypt(plaintext []byte) []byte {
    b, _ := aes.NewCipher([]byte(createHash(passphrase)))  // AES-128, AES-192, or AES-256
                                                           // if key is 32 bytes, AES-256 will be selected

    if mod := len(plaintext) % aes.BlockSize; mod != 0 {    // aes.BlockSize : 16 bytes
        padding := make([]byte, aes.BlockSize-mod)
        plaintext = append(plaintext, padding...)
    }

    ciphertext := make([]byte, aes.BlockSize+len(plaintext))
    iv := ciphertext[:aes.BlockSize]
    if _, err := io.ReadFull(rand.Reader, iv); err != nil {
        fmt.Println(err)
        return nil
    }

    mode := cipher.NewCBCEncrypter(b, iv)
    mode.CryptBlocks(ciphertext[aes.BlockSize:], plaintext)

    return ciphertext
}
```

5. SkyWalker Daemon(4) – FALCON Interface(7)

실제 동작 모습



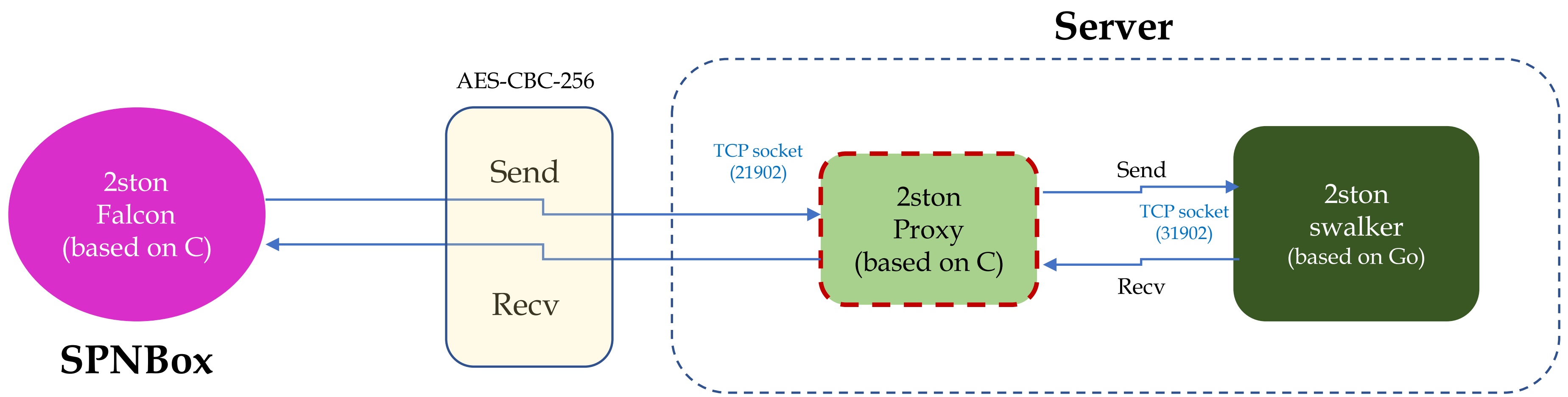
```
root@spnbox-900:~/workspace# ./2ston_falcon 13.124.231.29
2019/12/10 07:48:33 Established connection to 13.124.231.29:31902
2019/12/10 07:48:33 Remote TCP address : 13.124.231.29:31902
2019/12/10 07:48:33 Local TCP client address : 172.30.1.12:34450

2019/12/10 07:48:33 Registering it into the 2ston_swalker...

### SEND ###
-----------------------------------------------------------
| TCP | HELLO |  1234 |  10.  1.  1.200 |  10.  1.  1.100 |
-----------------------------------------------------------
### RECV ###
TCP Server : 13.124.231.29:31902
-----------------------------------------------------------
| TCP |   NOK | 31902 | 172. 31. 17. 27 | 121.162. 94.203 |
-----------------------------------------------------------

### SEND ###
-----------------------------------------------------------
| TCP | ASK_USER_CMD |  1234 |  10.  1.  1.200 |  10.  1.  1.100 |
-----------------------------------------------------------
### RECV ###
                                     ----------------
                                      121.162. 94.203 |
                                     ----------------

                                     ----------------
| TCP | ASK_USER_CMD |  1234 |  10.  1.  1.200 |  10.  1.  1.100 |
```

<TBD>

- ✓ Go binary size가 1MB를 넘는다 ☹
- ✓ **2ston_falcon이 MIPS32(little endian)에서 동작 안한다.**
- ✓ 2ston_falcon - C로 다시 구현해야 한다.

5. SkyWalker Daemon(4) – FALCON Interface(8-1)

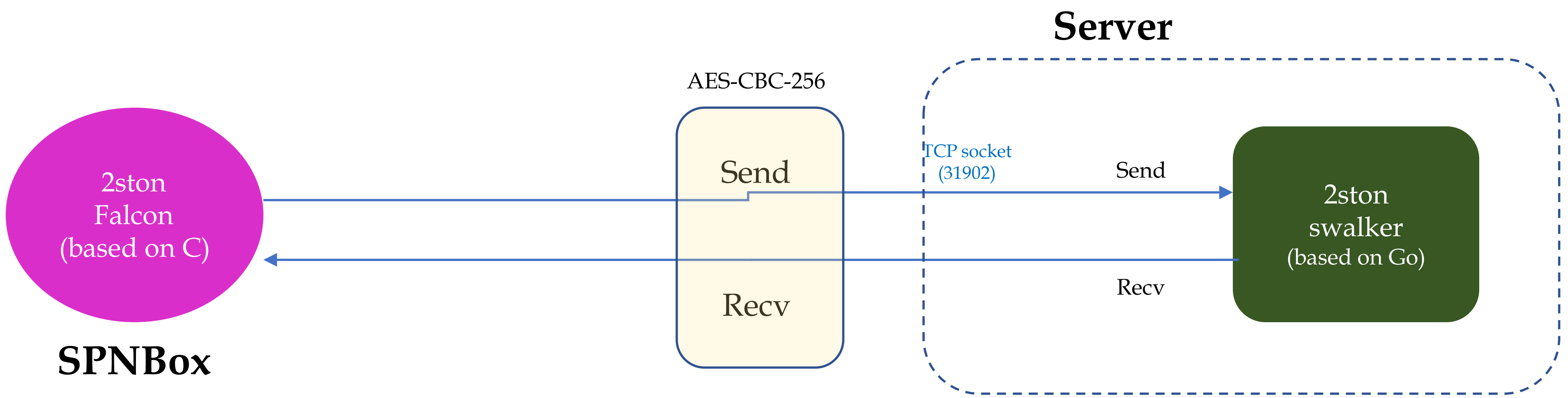
Proxy Interface



- 2ston_falcon - C로 다시 구현하자.
- 중간에 proxy를 하나 두자.
- 왜 ?
  - ✓ *C ⇔ Go routine 간 AES-CBC-256을 맞추는게 쉽지 않아 보여…*
  - ✓ *C로 proxy code를 구현하는 것이 상대적으로 빠른 작업이어서 …*

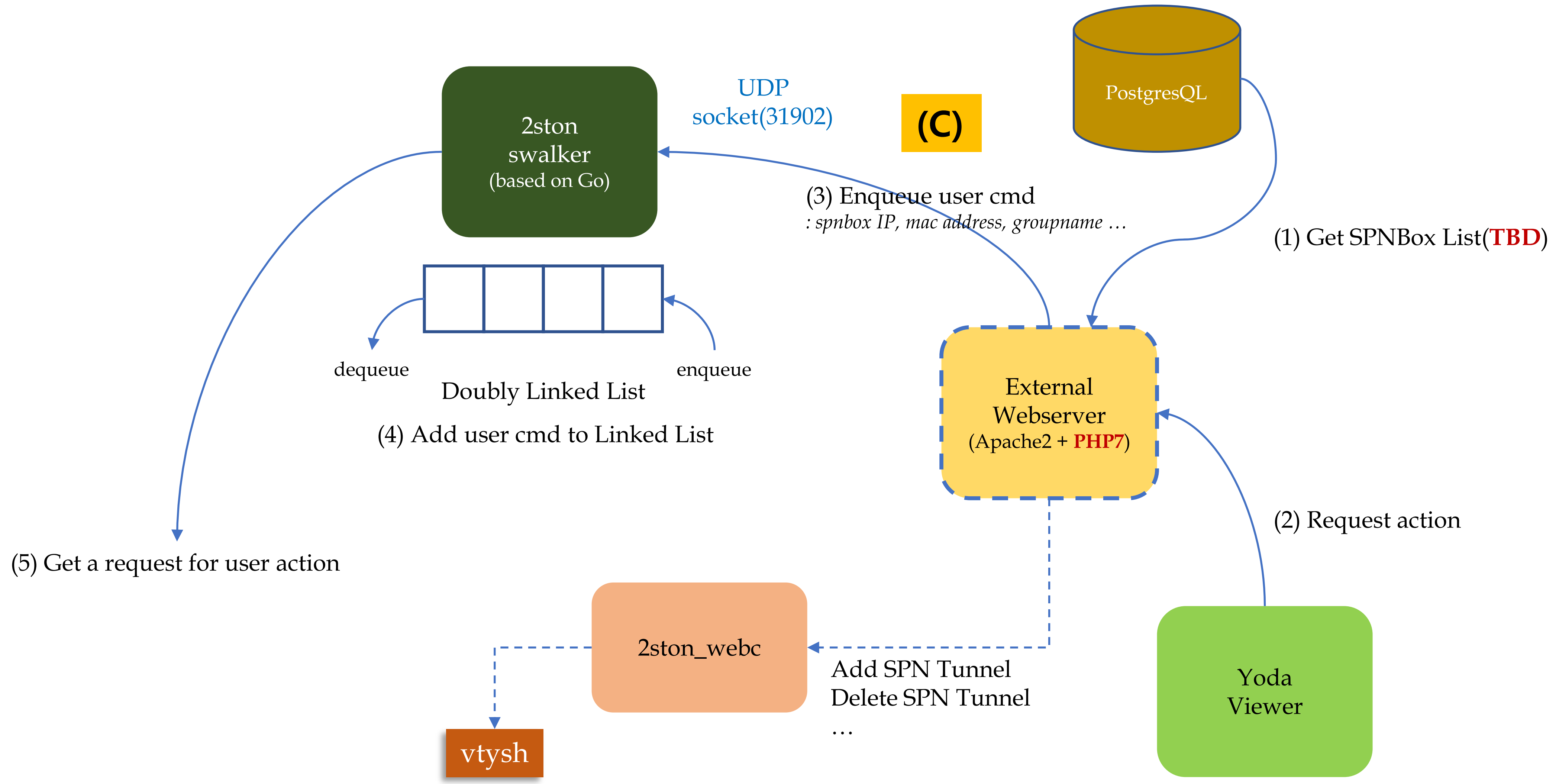
5. SkyWalker Daemon(4) – FALCON Interface(8-2)

Non-Proxy Interface



- 2ston_falcon (C version), no proxy
  - ✓ 중간에 proxy를 두는 방식이, 문제를 너무 복잡하게 만드는 듯 하니 direct로 연결하도록 하자.

5. SkyWalker Daemon(5) – WebServer Interface(1)

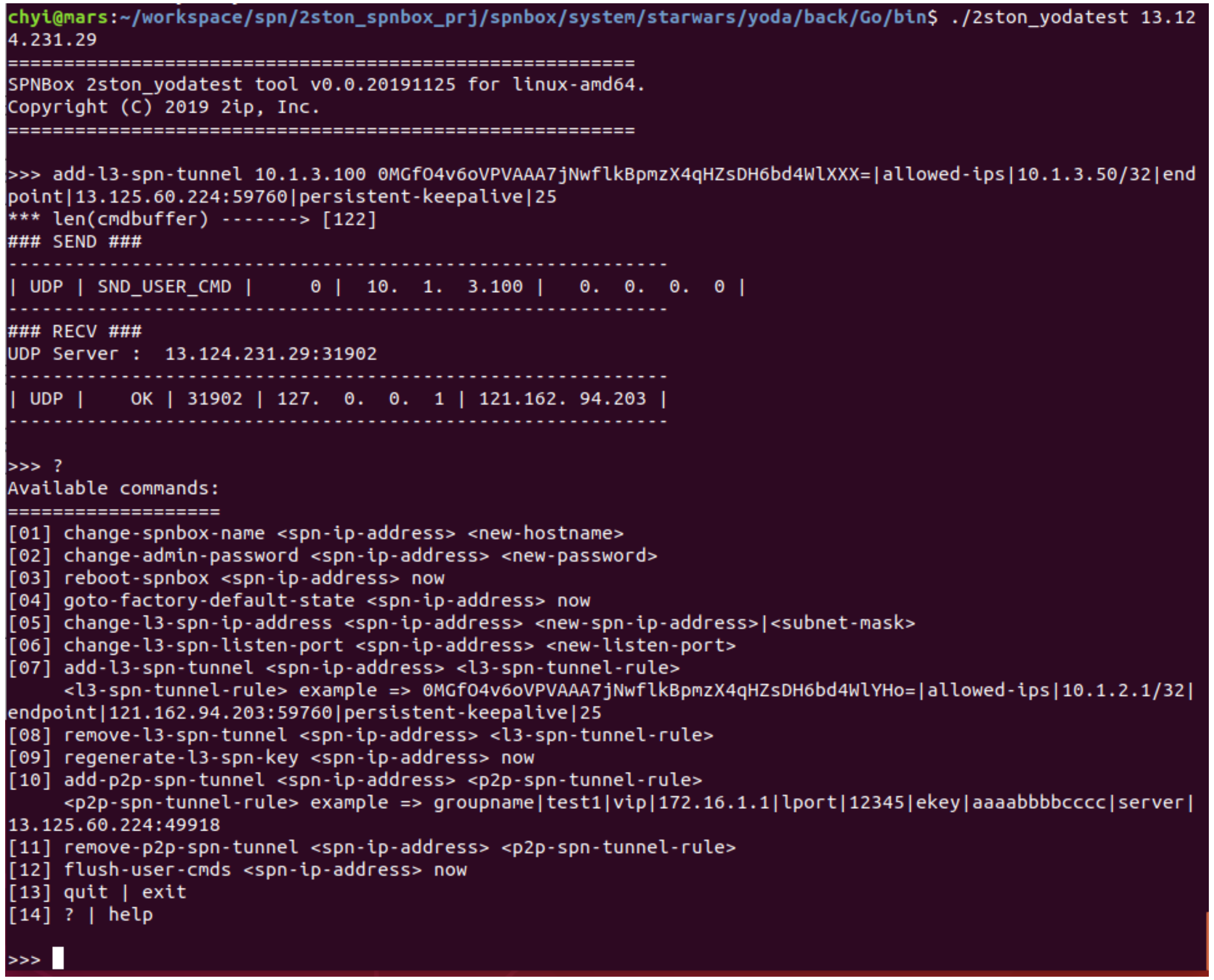


5. SkyWalker Daemon(5) – WebServer Interface(2)

<시나리오>

- [1] [TBD] WebServer backend(이하 PHP)는 주기적으로 PostgresQL DBMS에 연결하여 SPNBox List를 가져온다.
- [1]' 혹은 WebServer backend(이하 PHP)는 주기적으로 2ston_swalker에게 ASK_SPN_INFO 명령을 전달하여 ANS_SPN_INFO를 수신한다. 이 과정을 통해 전체 SPNBox List를 확보한다.
- [2] 사용자는 SPNBox List 중 하나를 선택한 후, 이를 토대로 사용자 명령을 내린다.
  - ✓ 사용자 명령으로는 change_spnbox_name, change_admin_password, reboot_spnbox, goto_factory_default_state, change_l3_spn_ip_address, … 등이 존재한다.
- [3] 사용자 명령을 전달 받은 WebServer PHP는 2ston_swalker에게 관련 명령을 전달한다.
  - Test code: 2ston_yodatest(based on Go)
- [4] 2ston_swalker는 사용자 명령을 Linked List에 추가한다.
- [5] 2ston_falcon은 주기적으로 2ston_swalker로 부터 사용자 명령을 가져와 관련 action을 수행한다.

5. SkyWalker Daemon(5) – WebServer Interface(3)



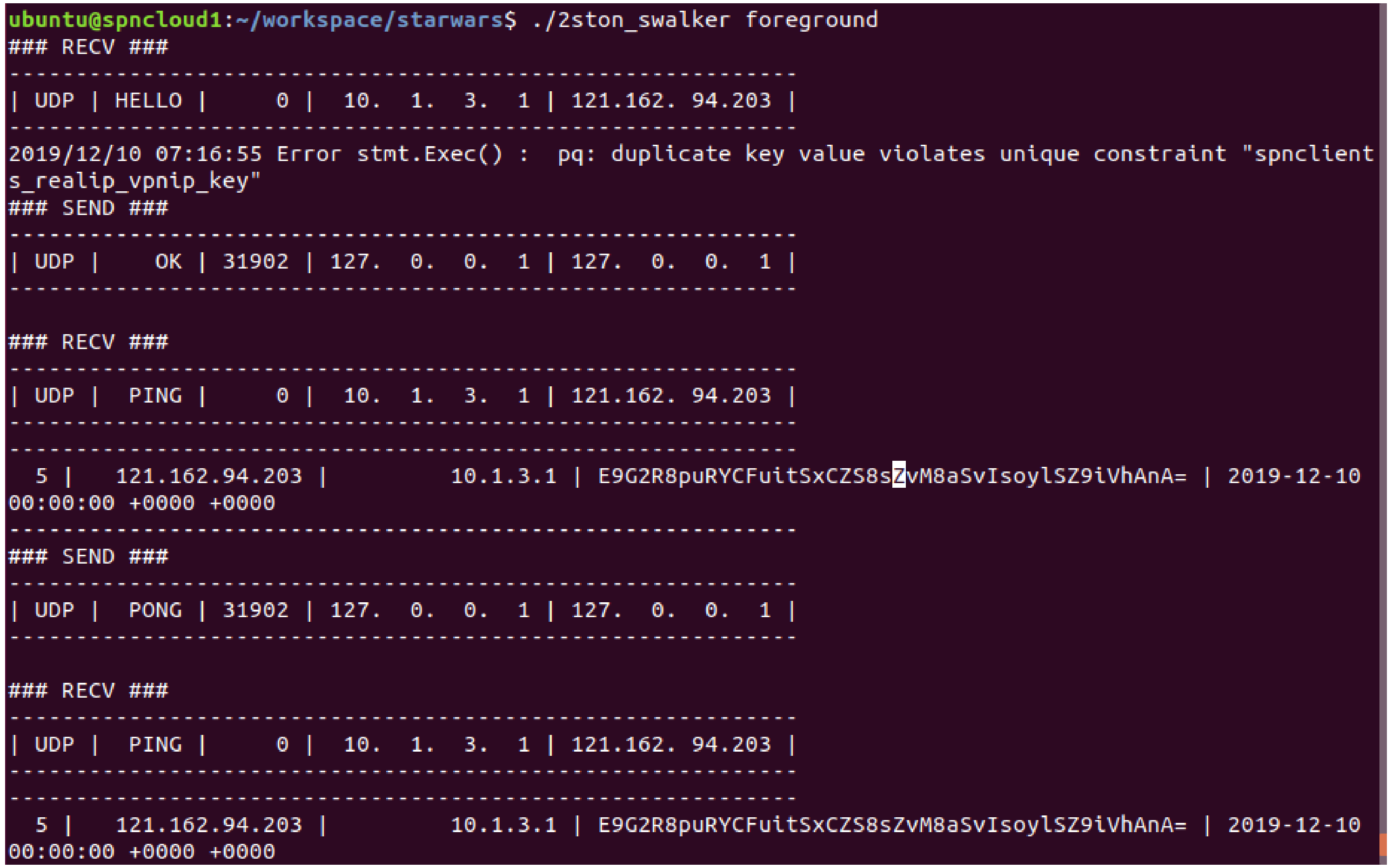
```
chyi@mars:~/workspace/spn/2ston_spnbox_prj/spnbox/system/starwars/yoda/back/Go/bin$ ./2ston_yodatest 13.12
4.231.29
========================================================
SPNBox 2ston_yodatest tool v0.0.20191125 for linux-amd64.
Copyright (C) 2019 2ip, Inc.
========================================================

>>> add-l3-spn-tunnel 10.1.3.100 0MGfO4v6oVPVAAA7jNwflkBpmzX4qHZsDH6bd4WlXXX=|allowed-ips|10.1.3.50/32|end
point|13.125.60.224:59760|persistent-keepalive|25
*** len(cmdbuffer) -------> [122]
### SEND ###
-----------------------------------------------------------
| UDP | SND_USER_CMD |     0 |  10.  1.  3.100 |   0.  0.  0.  0 |
-----------------------------------------------------------
### RECV ###
UDP Server :  13.124.231.29:31902
-----------------------------------------------------------
| UDP |    OK | 31902 | 127.  0.  0.  1 | 121.162. 94.203 |
-----------------------------------------------------------

>>> ?
Available commands:
===================
[01] change-spnbox-name <spn-ip-address> <new-hostname>
[02] change-admin-password <spn-ip-address> <new-password>
[03] reboot-spnbox <spn-ip-address> now
[04] goto-factory-default-state <spn-ip-address> now
[05] change-l3-spn-ip-address <spn-ip-address> <new-spn-ip-address>|<subnet-mask>
[06] change-l3-spn-listen-port <spn-ip-address> <new-listen-port>
[07] add-l3-spn-tunnel <spn-ip-address> <l3-spn-tunnel-rule>
     <l3-spn-tunnel-rule> example => 0MGfO4v6oVPVAAA7jNwflkBpmzX4qHZsDH6bd4WlYHo=|allowed-ips|10.1.2.1/32|
endpoint|121.162.94.203:59760|persistent-keepalive|25
[08] remove-l3-spn-tunnel <spn-ip-address> <l3-spn-tunnel-rule>
[09] regenerate-l3-spn-key <spn-ip-address> now
[10] add-p2p-spn-tunnel <spn-ip-address> <p2p-spn-tunnel-rule>
     <p2p-spn-tunnel-rule> example => groupname|test1|vip|172.16.1.1|lport|12345|ekey|aaaabbbbcccc|server|
13.125.60.224:49918
[11] remove-p2p-spn-tunnel <spn-ip-address> <p2p-spn-tunnel-rule>
[12] flush-user-cmds <spn-ip-address> now
[13] quit | exit
[14] ? | help

>>> 
```

이 코드는 Go 로작성한 것이며,
실제로는 PHP로 다시 작성해야 함.

5. SkyWalker Daemon(5) – WebServer Interface(4)

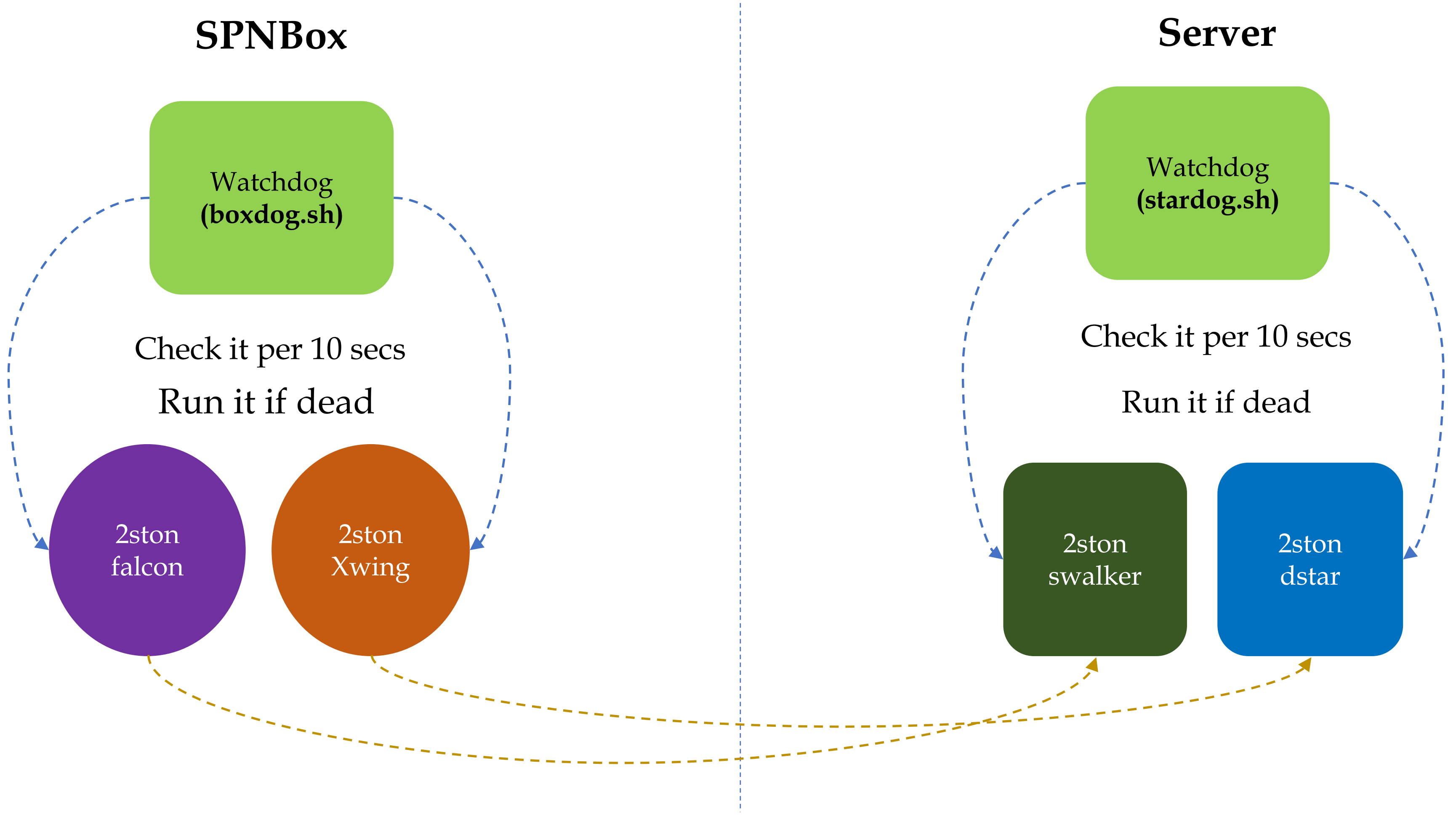
- <TBD> PHP codes(client)

5. SkyWalker Daemon(6) – 실제 동작 모습



```
ubuntu@spncloud1:~/workspace/starwars$ ./2ston_swalker foreground
### RECV ###
-----------------------------------------------------------
| UDP | HELLO |     0 |  10.  1.  3.  1 | 121.162. 94.203 |
-----------------------------------------------------------
2019/12/10 07:16:55 Error stmt.Exec() :  pq: duplicate key value violates unique constraint "spnclient
s_realip_vpnip_key"
### SEND ###
-----------------------------------------------------------
| UDP |    OK | 31902 | 127.  0.  0.  1 | 127.  0.  0.  1 |
-----------------------------------------------------------

### RECV ###
-----------------------------------------------------------
| UDP |  PING |     0 |  10.  1.  3.  1 | 121.162. 94.203 |
-----------------------------------------------------------
-----------------------------------------------------------
  5 |   121.162.94.203 |         10.1.3.1 | E9G2R8puRYCFuitSxCZS8sZvM8aSvIsoylSZ9iVhAnA= | 2019-12-10
00:00:00 +0000 +0000
-----------------------------------------------------------
### SEND ###
-----------------------------------------------------------
| UDP |  PONG | 31902 | 127.  0.  0.  1 | 127.  0.  0.  1 |
-----------------------------------------------------------

### RECV ###
-----------------------------------------------------------
| UDP |  PING |     0 |  10.  1.  3.  1 | 121.162. 94.203 |
-----------------------------------------------------------
-----------------------------------------------------------
  5 |   121.162.94.203 |         10.1.3.1 | E9G2R8puRYCFuitSxCZS8sZvM8aSvIsoylSZ9iVhAnA= | 2019-12-10
00:00:00 +0000 +0000
```

5. SkyWalker Daemon(7) – Watchdog Script



5. SkyWalker Daemon(8) – TODO

- 1) 2ston_falcon(C version)
  - 2ston_falcon(C version) ⇔ 2ston_swalker간 암호 통신
  - User command(원격 명령) 실제 처리 코드(예: reboot)
  - 2ston_falcon code를 2ston_xwing에 통합 할 수도 있음.

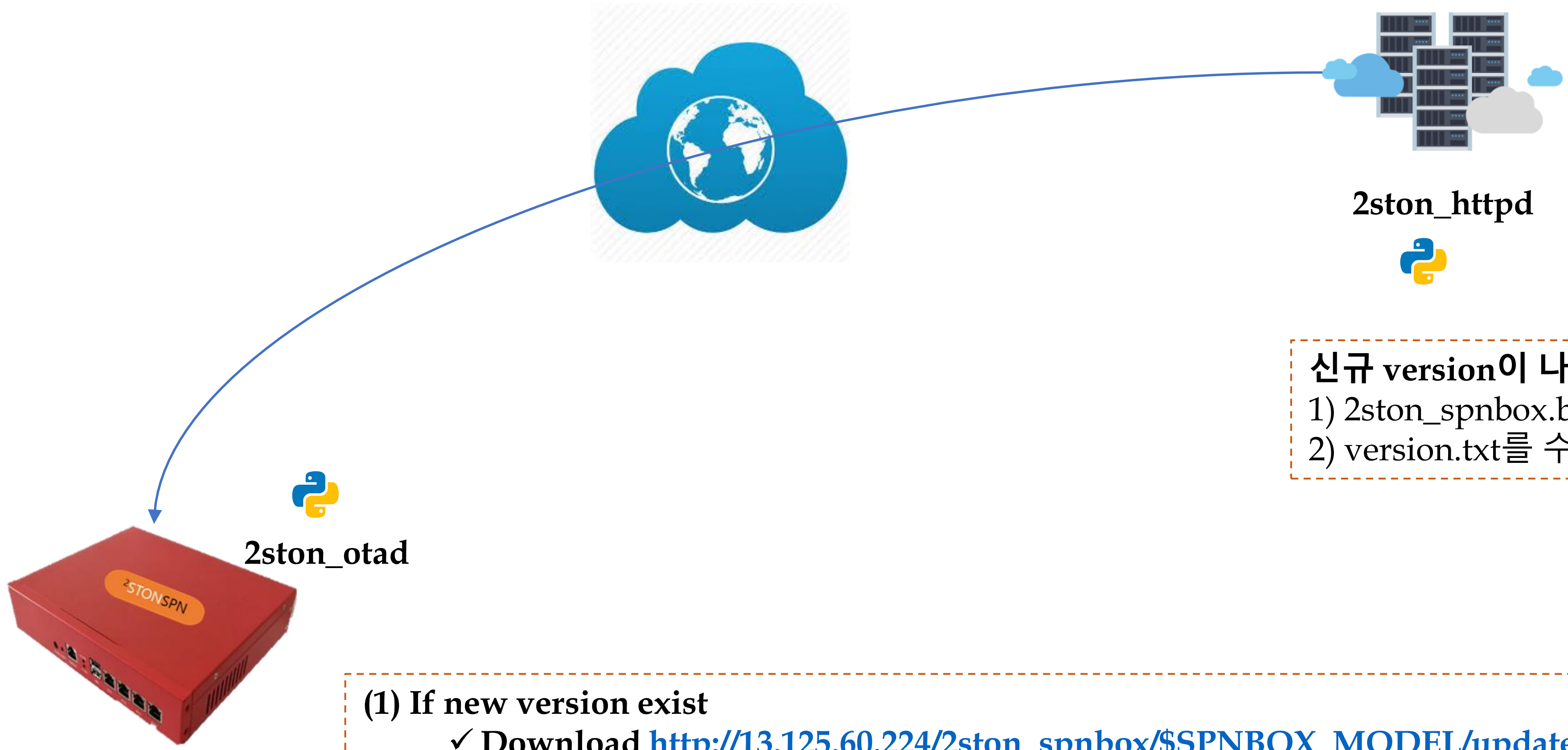
- 2) 2ston_swalker(Go version)
  - Swalker 내에서 DB table 생성하는 코드 추가
  - PostgresQL DB operation(추가 작업이 있을 듯)
  - Doubly Linked List(User command Queue) 관련 추가 작업 있음.
  - …

- 3) 2ston_dstar, 2ston_xwing
  - 추가 debugging(안정화 작업)

- 4) Startup script
  - Start/stop 관련 shell script 보강 작업

OTA Daemon

6. OTA Daemon(1) – S/W Upgrade



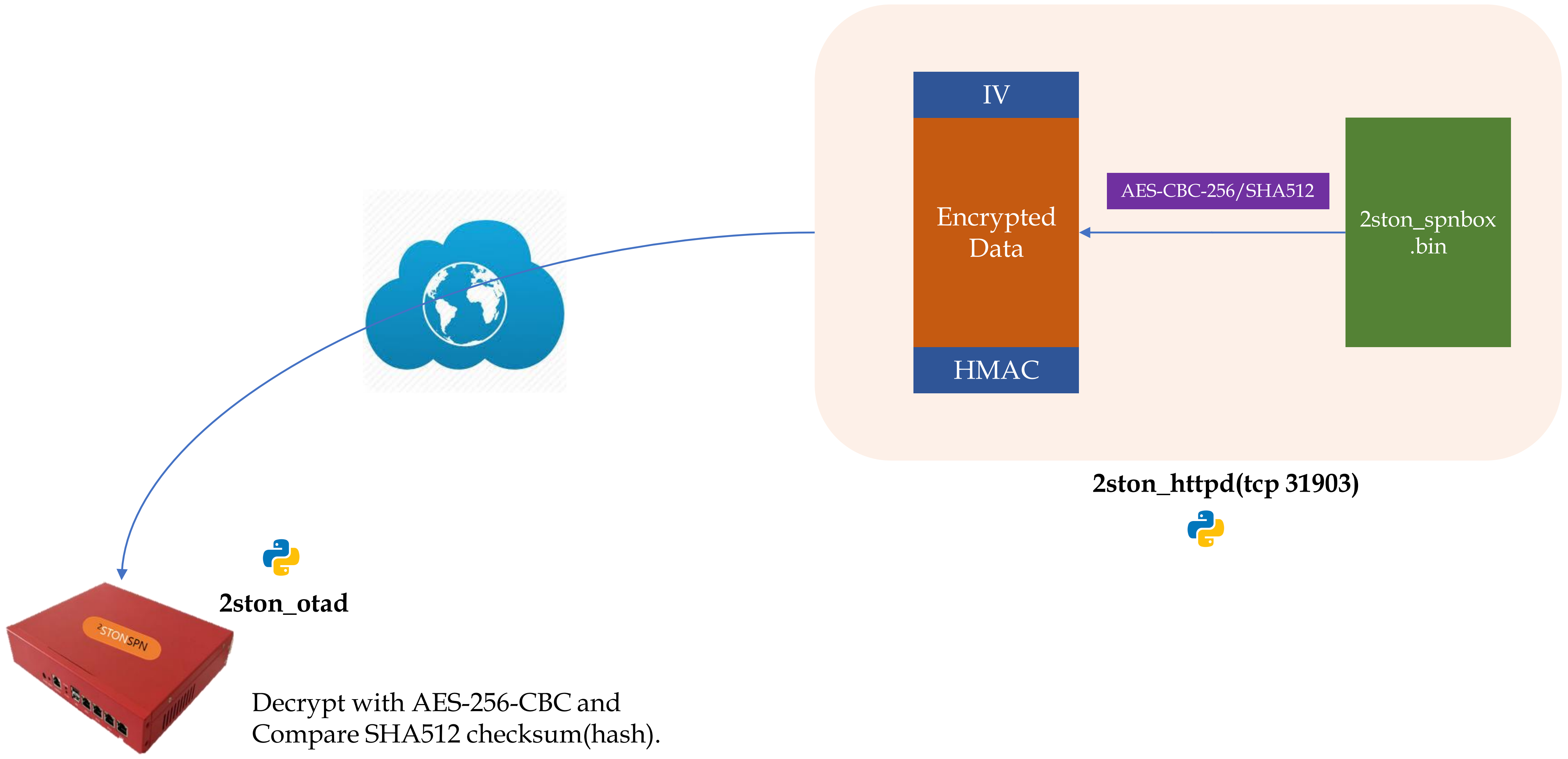
version.txt
update.bin
2ston_spnbox.bin
(*) AES 암호화/Hash

신규 version이 나올 경우
1) 2ston_spnbox.bin 갱신
2) version.txt를 수정(version up)

(1) If new version exist
- ✓ **Download http://13.125.60.224/2ston_spnbox/$SPNBOX_MODEL/update.bin**

(2) Decrypt update.bin and then run ./update.sh
- ✓ **Download 2ston_spnbox.bin and decrypt it.**
- ✓ **Uncompress and untar 2ston_spnbox.tar.gz**
- ✓ **Run AutoInstall.sh**
- ✓ **Reboot**

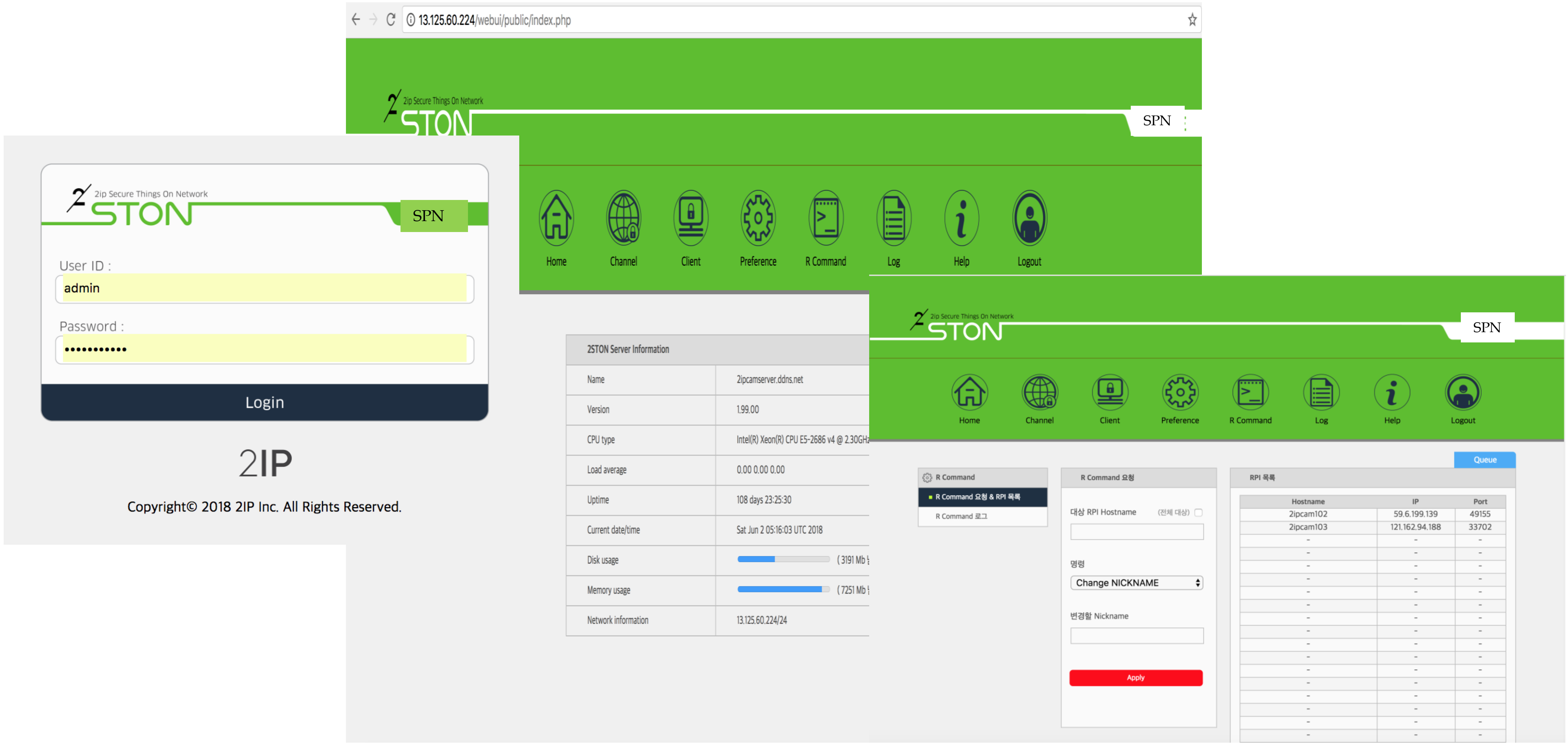
6. OTA Daemon(2) – S/W Upgrade



Decrypt with AES-256-CBC and
Compare SHA512 checksum(hash).

SPNBox Star Console Viewer

7. SPNBox Star Console Viewer – Yoda(1)



User ID : admin

Password : •••••••••••

Copyright© 2018 2IP Inc. All Rights Reserved.

2STON Server Information

Name	2ipcamserver.ddns.net
Version	1.99.00
CPU type	Intel(R) Xeon(R) CPU E5-2686 v4 @ 2.30GH
Load average	0.00 0.00 0.00
Uptime	108 days 23:25:30
Current date/time	Sat Jun 2 05:16:03 UTC 2018
Disk usage	(3191 Mb
Memory usage	(7251 Mb
Network information	13.125.60.224/24

R Command
- R Command 요청 & RPI 목록
- R Command 로그

R Command 요청

대상 RPI Hostname (전체 대상) ☐

명령: Change NICKNAME

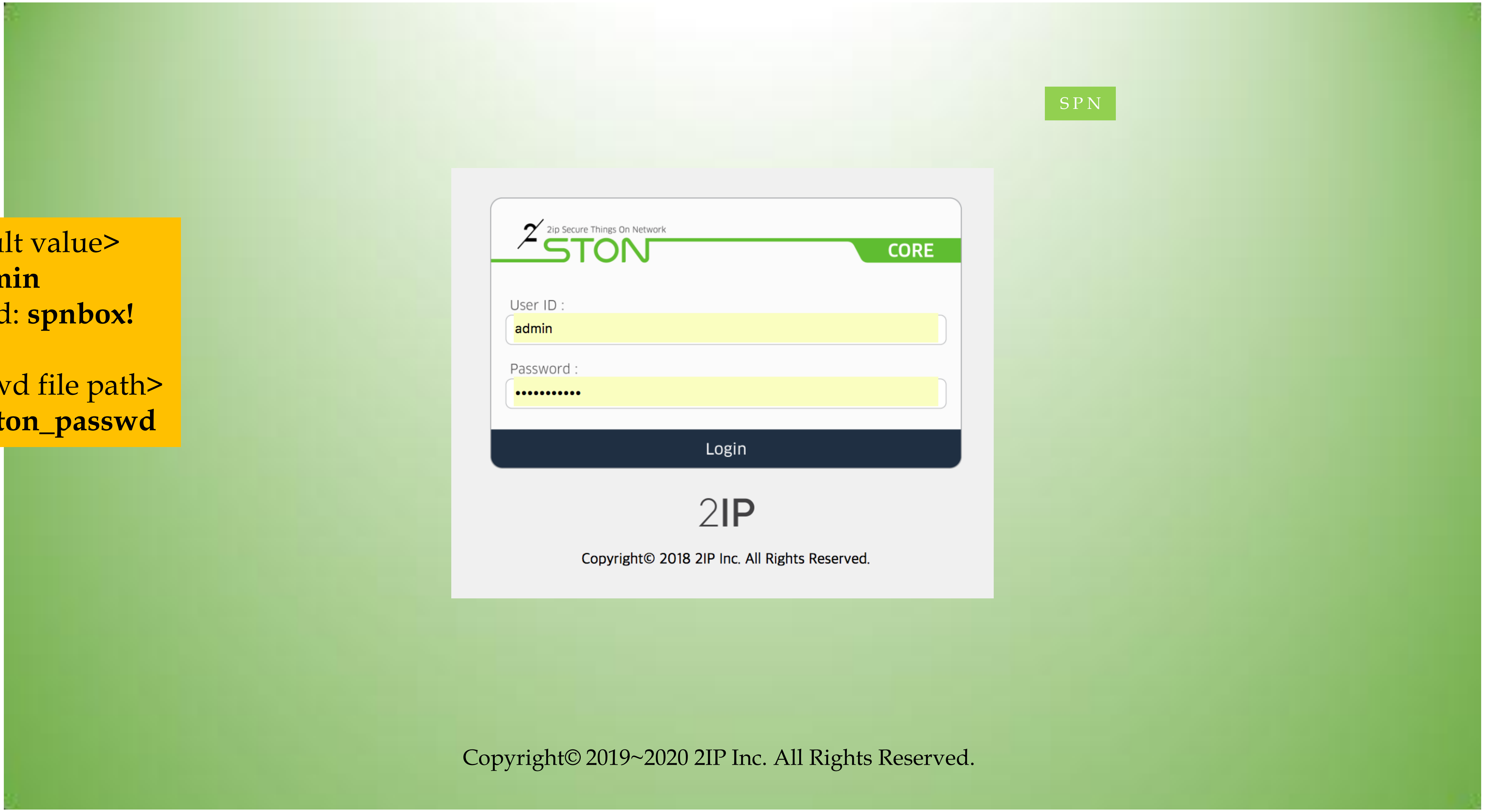
변경할 Nickname

RPI 목록

Hostname	IP	Port
2ipcam102	59.6.199.139	49155
2ipcam103	121.162.94.188	33702
-	-	-
-	-	-
-	-	-
-	-	-
-	-	-
-	-	-
-	-	-
-	-	-
-	-	-
-	-	-
-	-	-
-	-	-
-	-	-
-	-	-
-	-	-
-	-	-

7. SPNBox Star Console Viewer – Yoda(2)

Login Page



User ID :
admin
Password :
•••••••••••

Copyright© 2018 2IP Inc. All Rights Reserved.

Copyright© 2019~2020 2IP Inc. All Rights Reserved.

<default value>
id: **admin**
passwd: **spnbox!**

<passwd file path>
/etc/2ston_passwd

Thank You

2STONSPN

We Secure the Internet of Things with 2STON™