Auto Provisioning
Overview

Authors: Alexander Feldt

Challenges

- You're the IT admin!
- You have managed to configure 3 snom phones manually?!
  - Congratulation!
- Well, there are 1000 more waiting for you…
- Don't panic, we'll show you how!

Definition: Auto Provisioning

- **Auto Provisioning (aka „Mass deployment“) is**
  - a feature implemented proprietarily in the standard firmware of all snom IP phones (m9/m3 restrictions apply)
  - a remote administration tool allowing configuration and maintenance of unlimited number of different snom IP phone types
  - the first choice for out-of-box scenarios in large phone installations
  - the only way to configure ALL configuration parameters ("Settings") AND perform automatic firmware updates

Objectives

(1) Defining and setting up the appropriate AP scenario, i.e. how to “tell” a phone where to request its configuration parameters and firmware updates on first start-up

(2) Defining and setting up the AP server (APS), i.e. server protocol, client access and security rules, availability etc.

(3) Defining and setting up the APS data structure, i.e. what is the required format of the requested configuration parameters and where to store these data

Objectives (1)

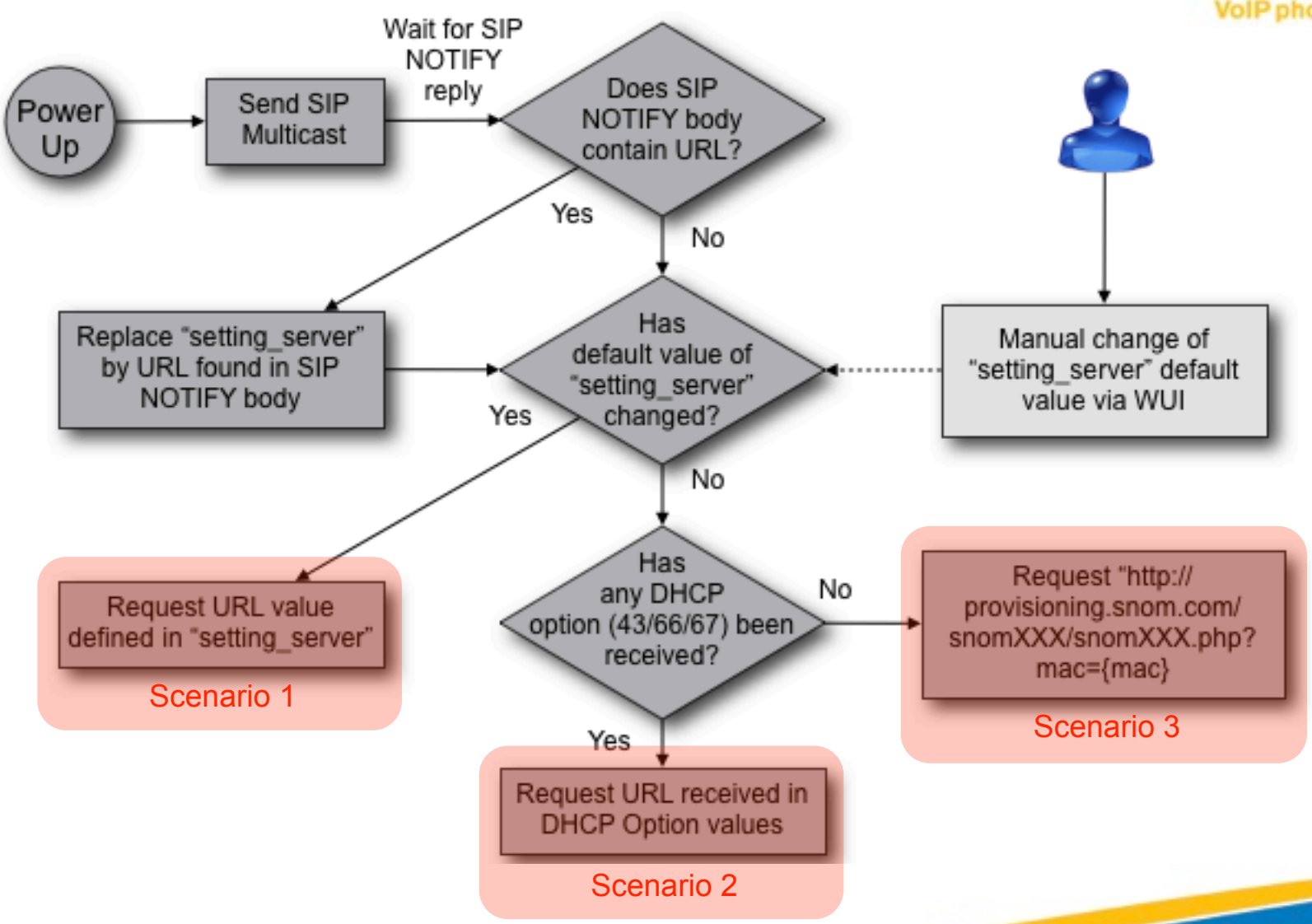
(1) **Defining and setting up the appropriate AP scenario, i.e. how to "tell" a phone where to request its configuration parameters and firmware updates on first start-up**

(2) Defining and setting up the AP server (APS), i.e. server protocol, client access and security rules, availability etc.

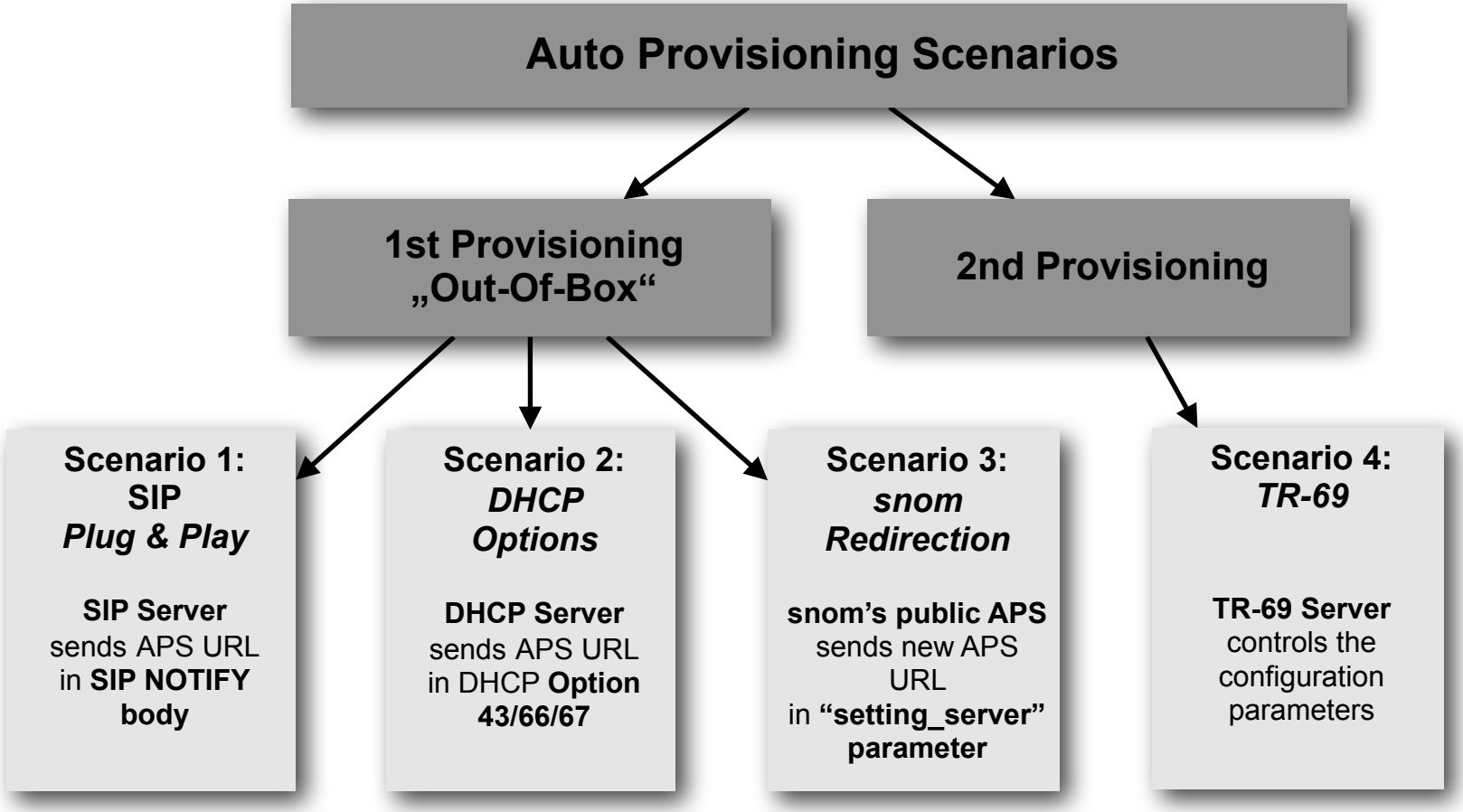
(3) Defining and setting up the APS data structure, i.e. what is the required format of the requested configuration parameters and where to store these data

Authors: Alexander Feldt

Scenarios

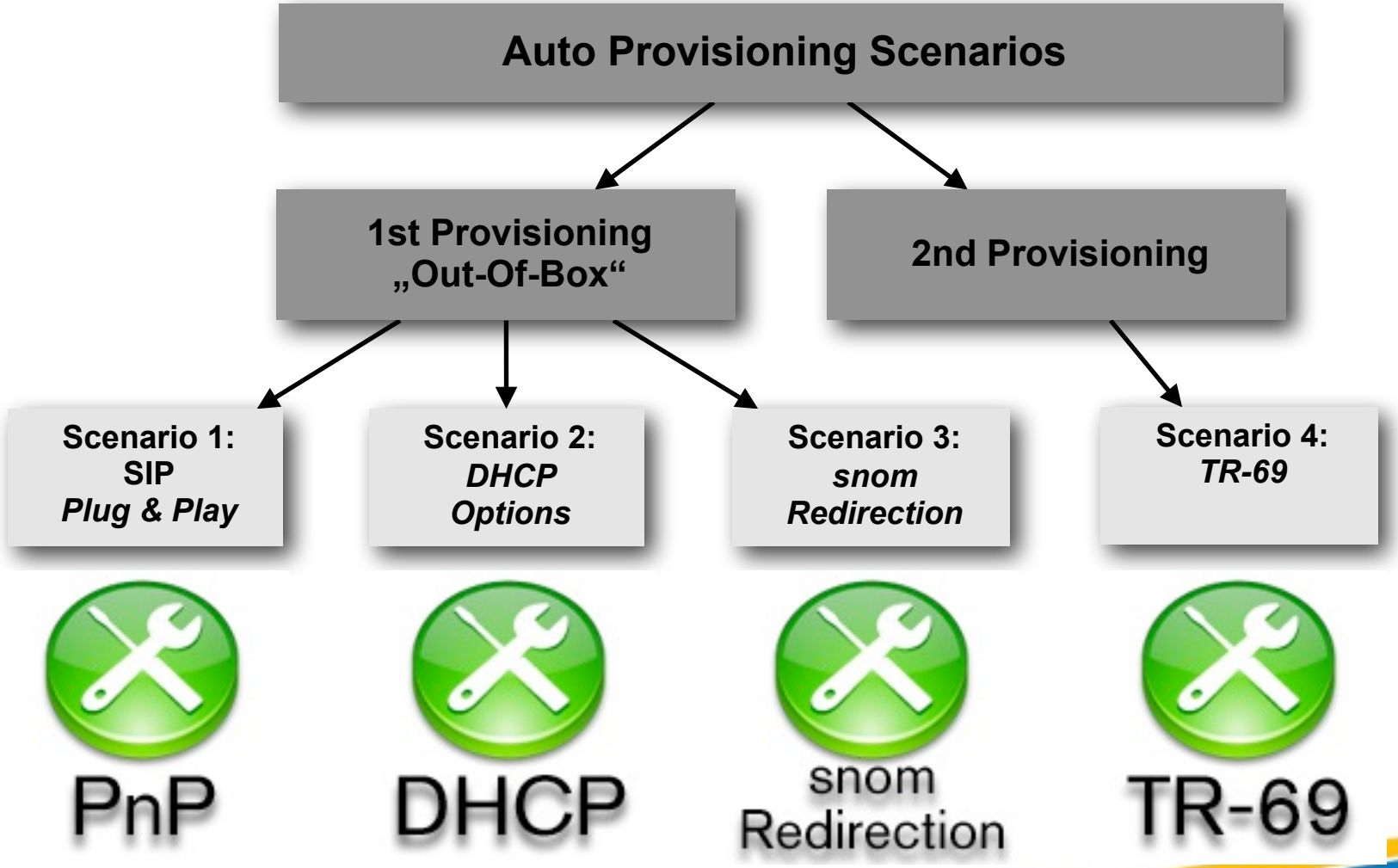


Scenarios (1)



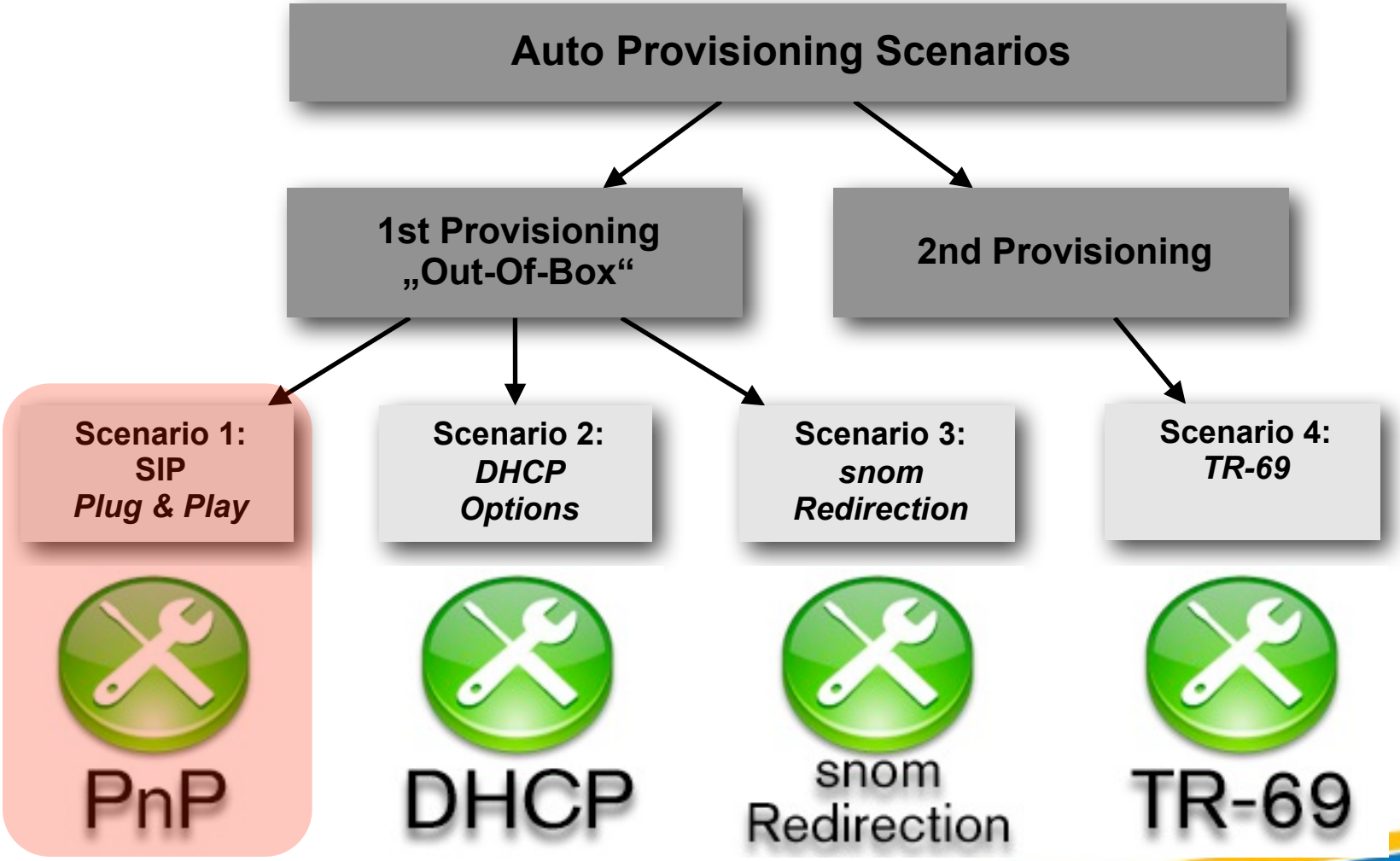
Authors: Alexander Feldt

Scenarios (2)



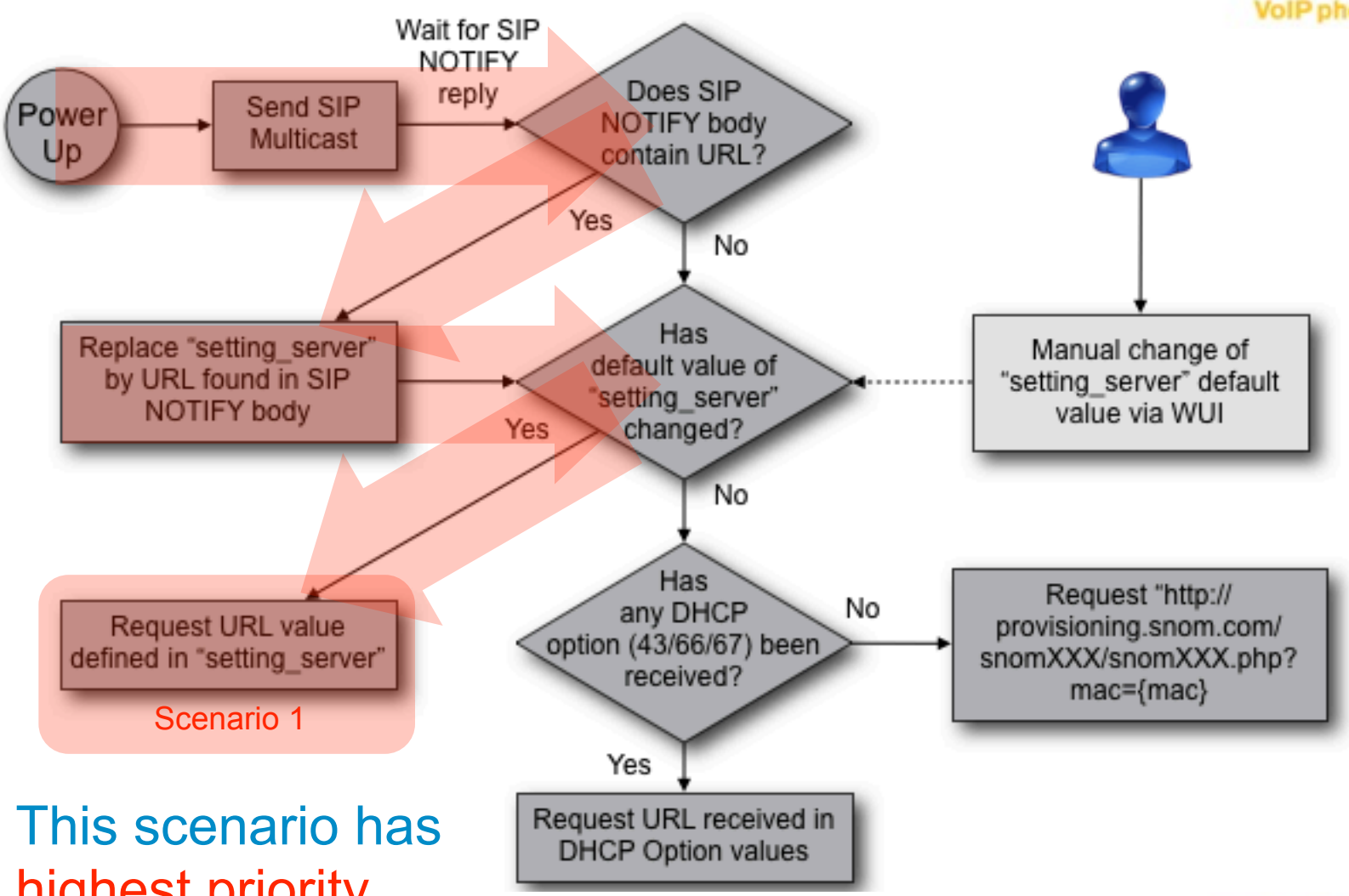
Authors: Alexander Feldt

Scenario 1: SIP Plug&Play (PnP)



Authors: Alexander Feldt

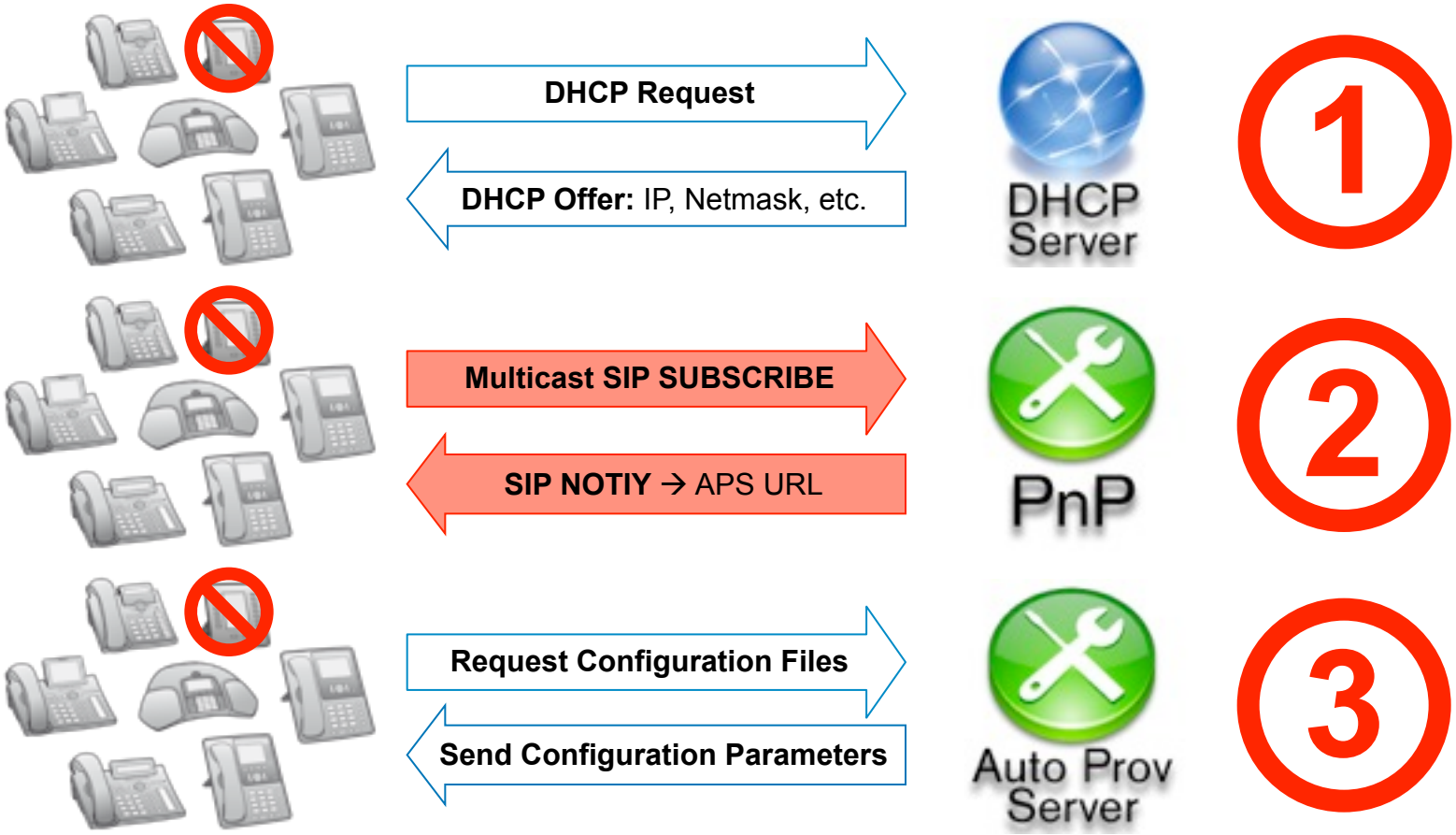
Boot-Up Priority: SIP PnP



This scenario has highest priority.

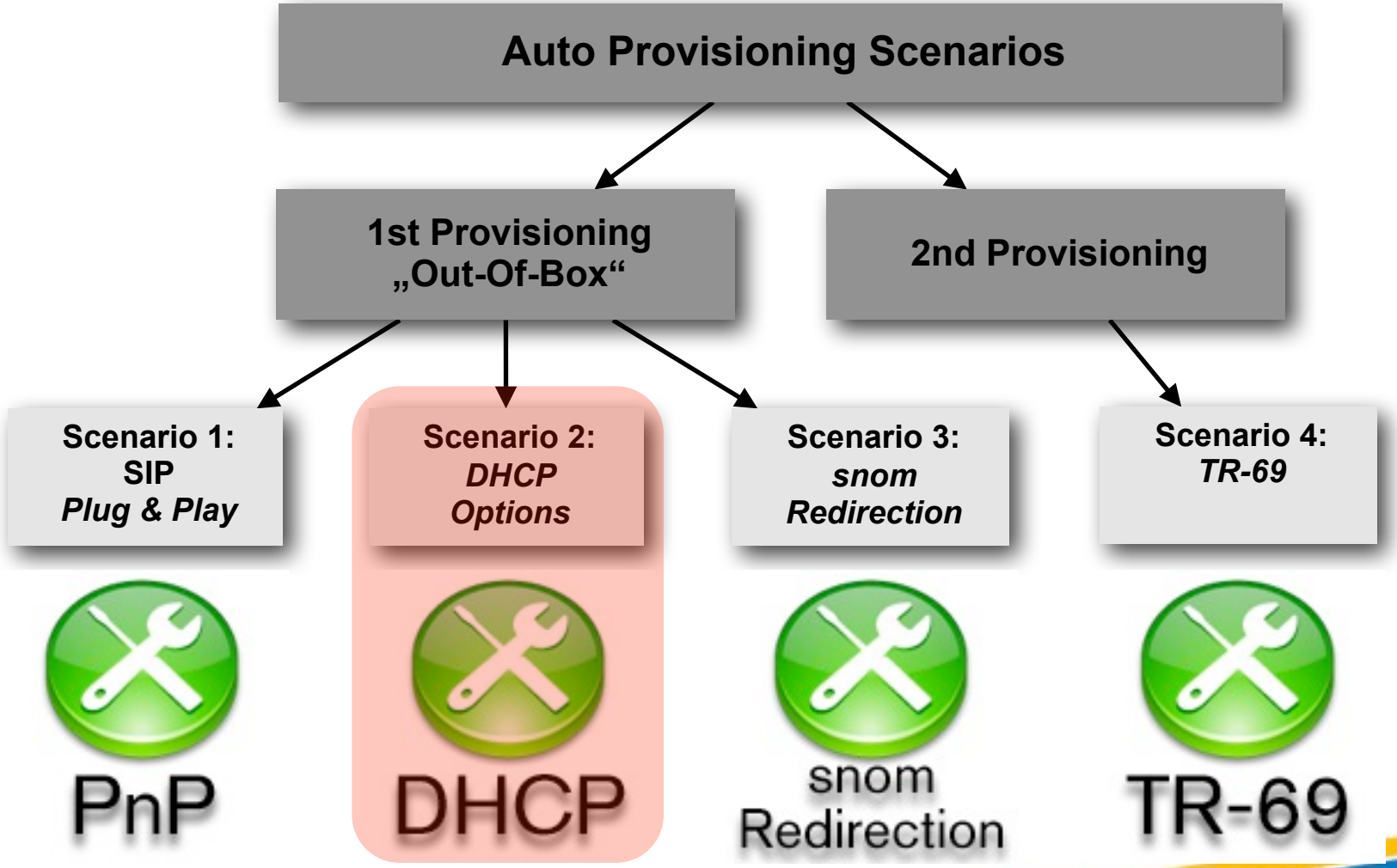
Authors: Alexander Feldt

Process Flow: SIP PnP

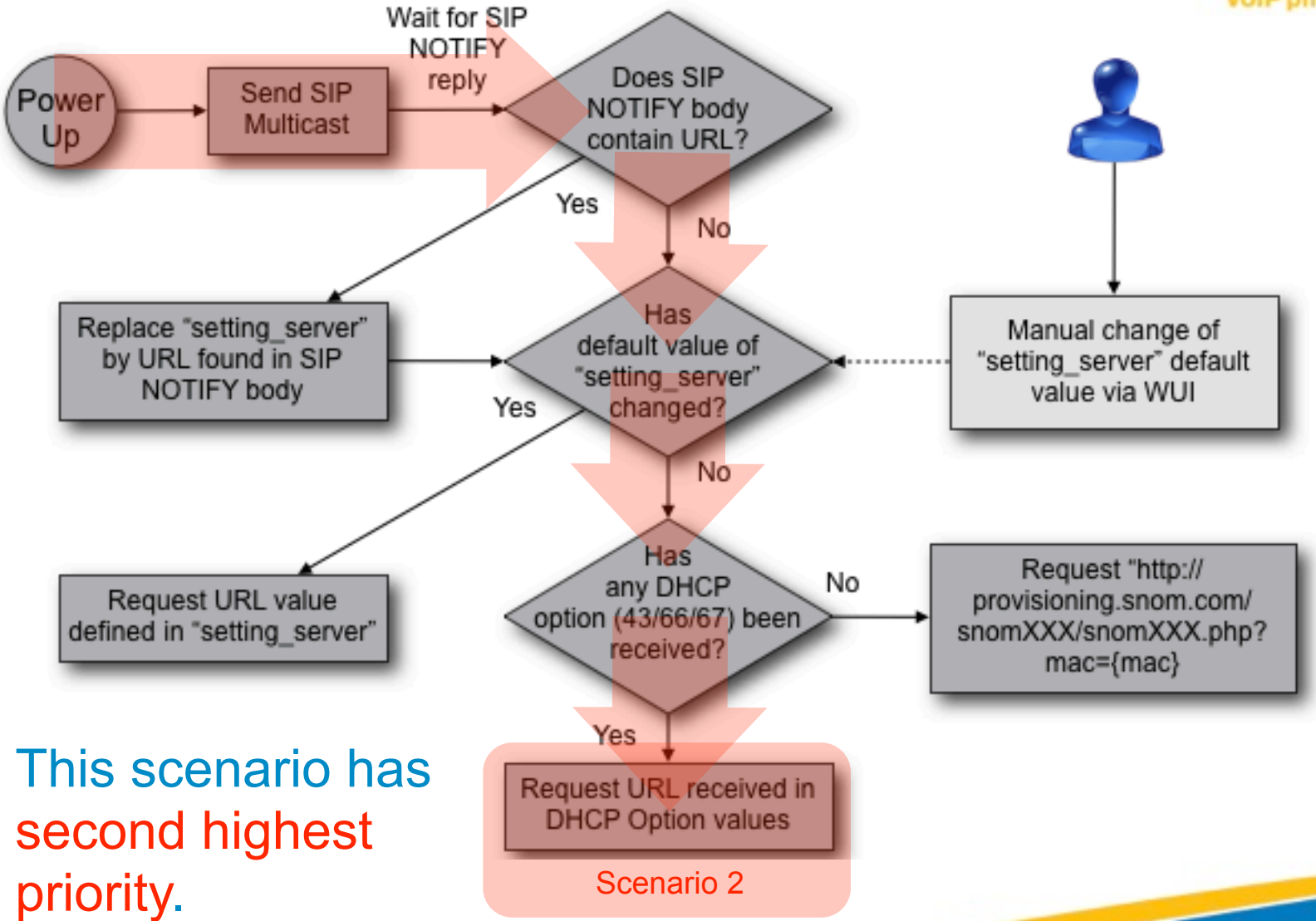


Authors: Alexander Feldt

Scenario 2: DHCP Options



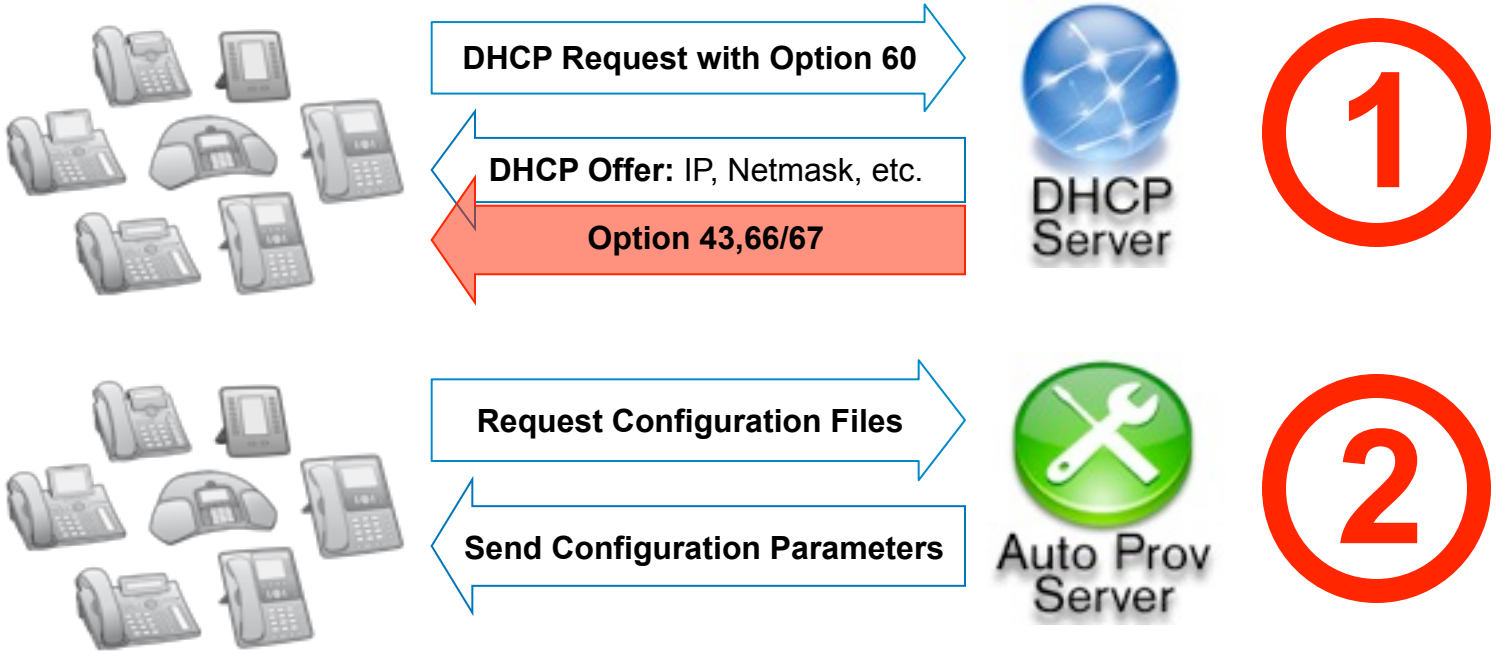
Boot-up Priority: DHCP Options



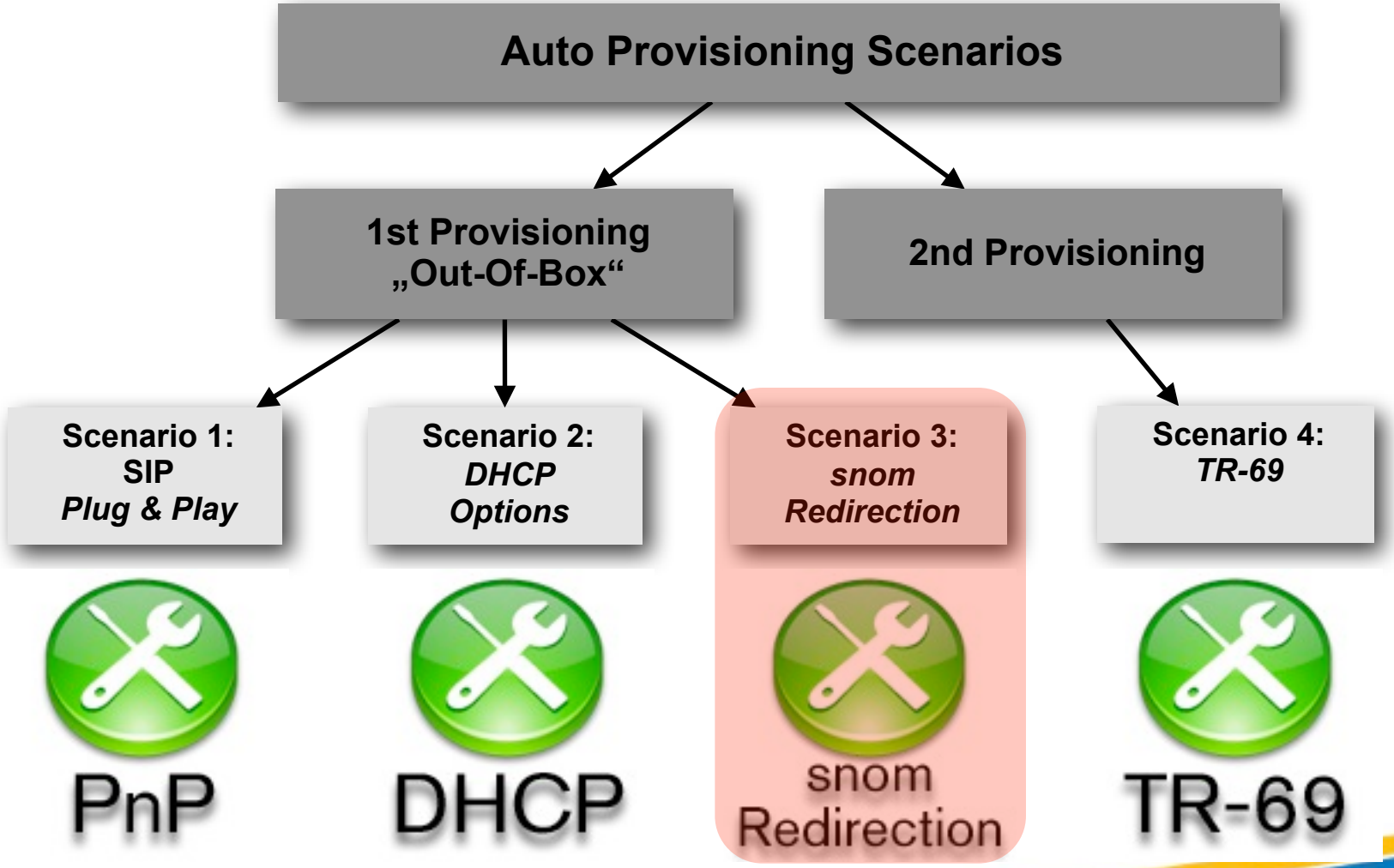
This scenario has second highest priority.

Authors: Alexander Feldt

Process Flow: DHCP Options

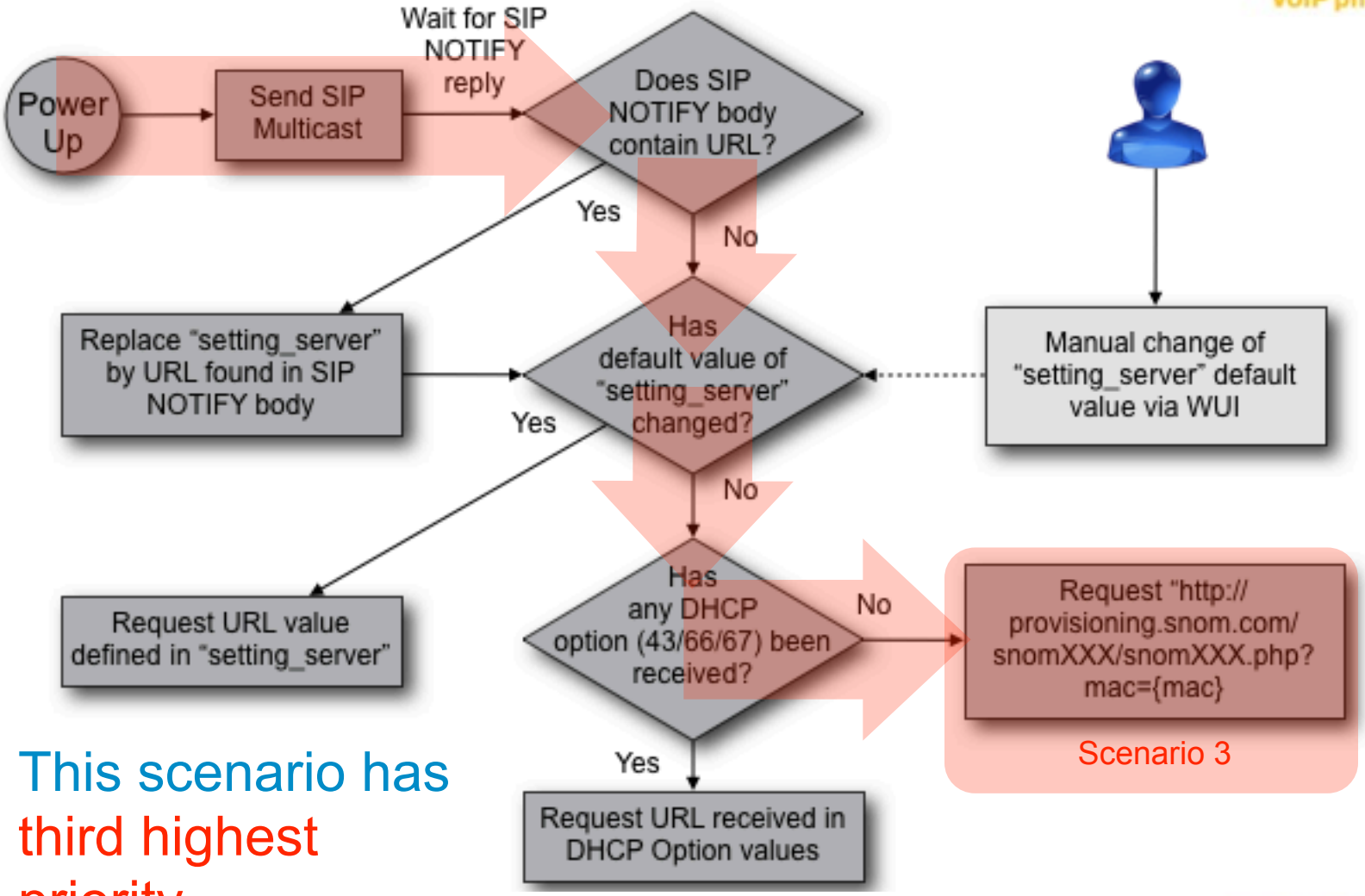


Scenario 3: snom Redirection



Authors: Alexander Feldt

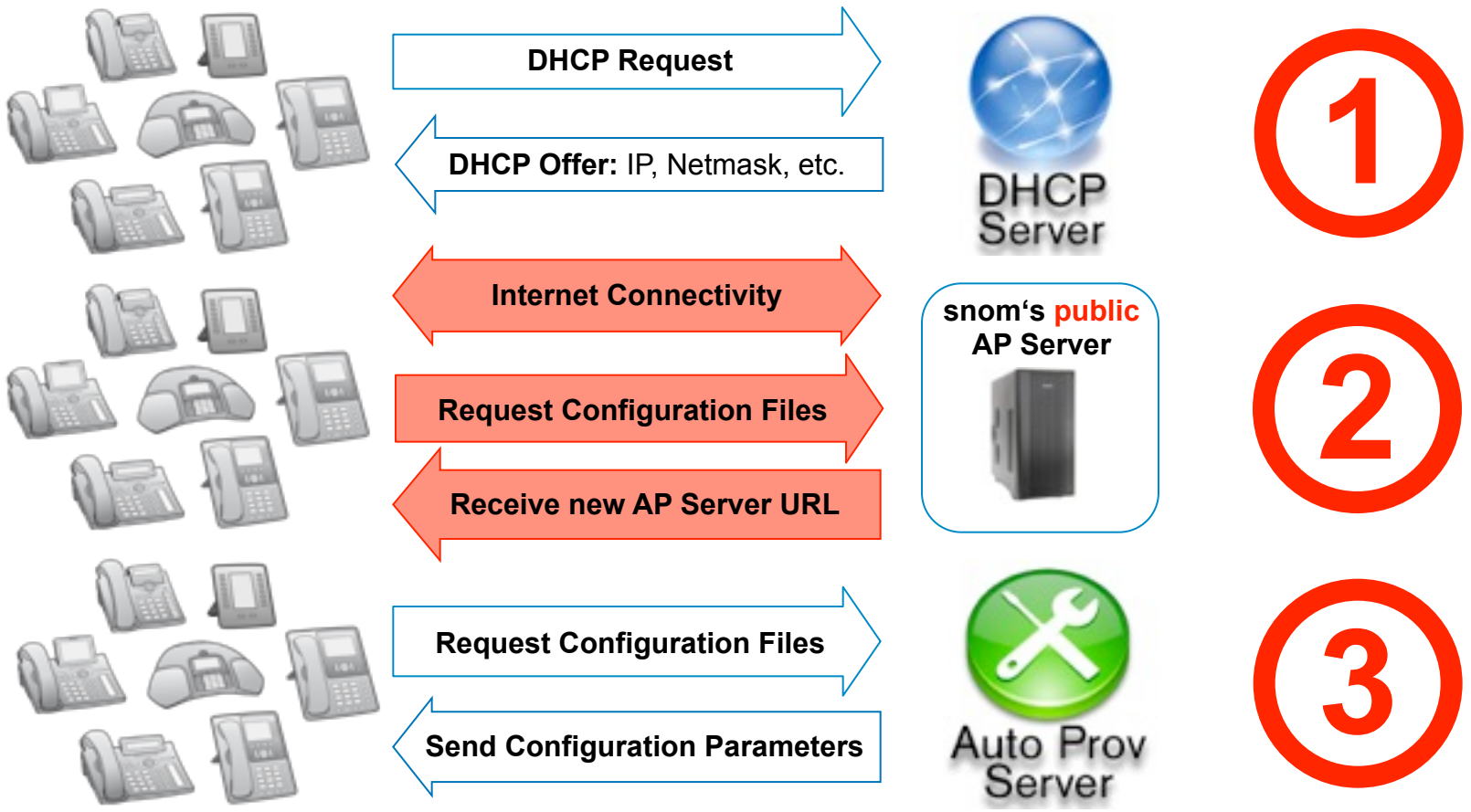
Boot-Up Priority: snom Redirection



This scenario has third highest priority.

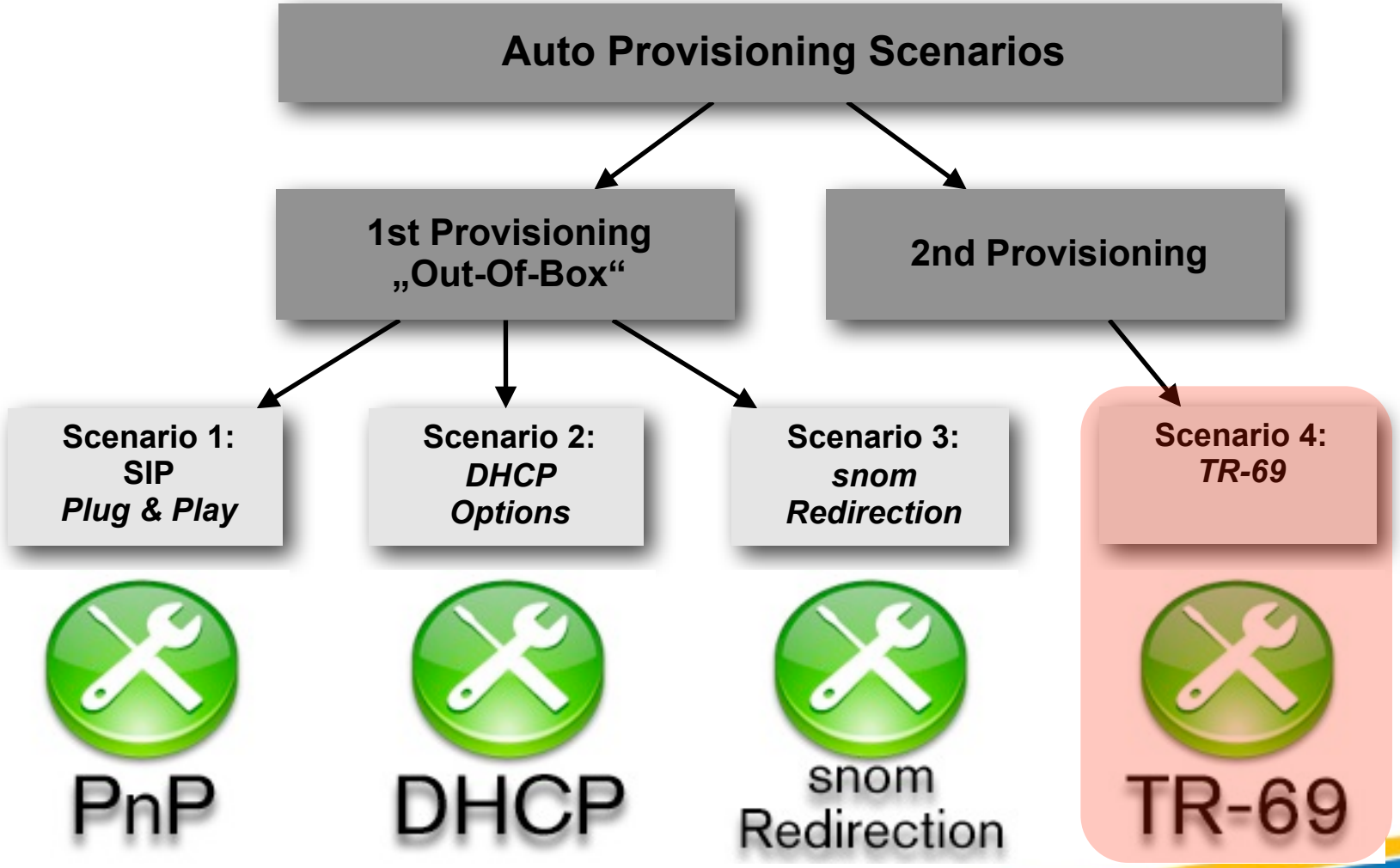
Authors: Alexander Feldt

Process Flow: snom Redirection



Authors: Alexander Feldt

Scenario 4: TR-69

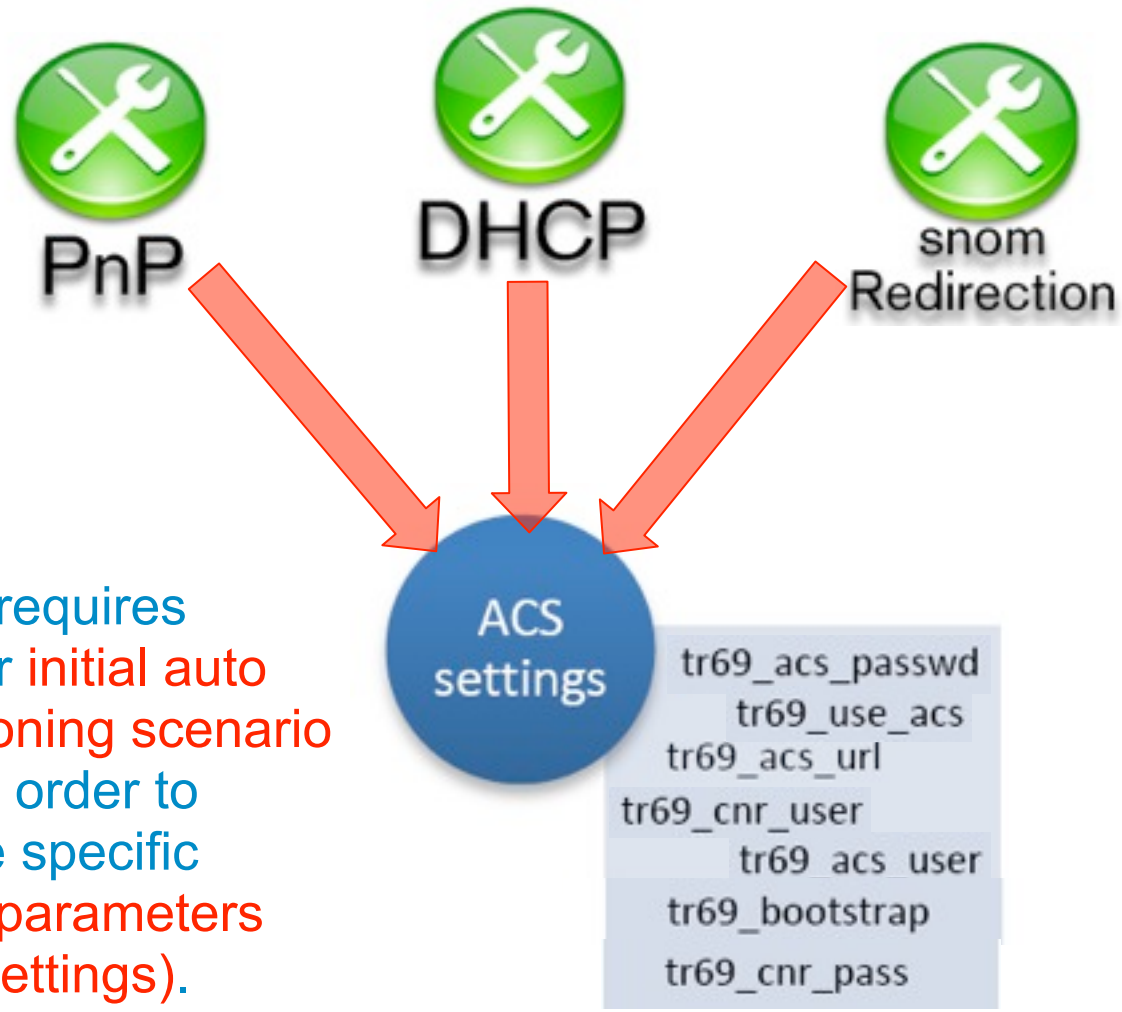


Authors: Alexander Feldt

Overview: TR-69

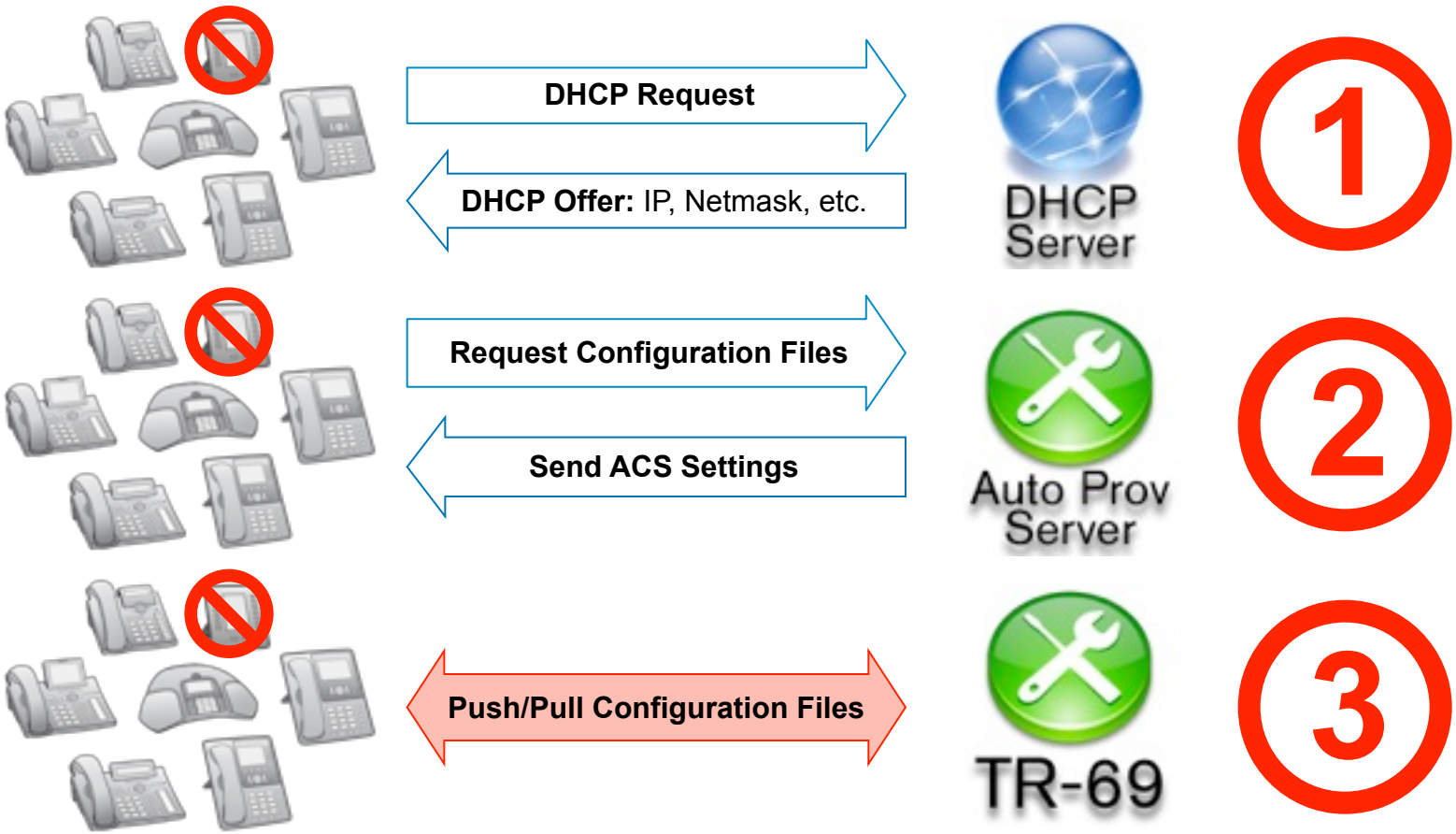
- TR-69 is a standard for remote management of CPE (Customer Premises Equipment) defined by the DSL Broadband Forum.
- TR-69 uses common transport mechanisms, i.e. HTTP and HTTPS, for communication with CPE's. The HTTP(S) messages contain XML-RPC methods defined in the standard for configuration and management of the device.
- TR-69 support has been introduced in firmware version 8.2.16 for the following phones: snom3xx, snom820, snom Meeting Point. Firmware 8.4.x supports snom870 and snom821. snom m9 will be added shortly.

Prerequisites: TR-69



TR-69 requires another initial auto provisioning scenario (1-3) in order to provide specific TR-69 parameters (ACS settings).

Process Flow: TR-69



Authors: Alexander Feldt

Conclusions (1)

- There are three proprietary out-of-box AP scenarios implemented into the standard firmware of snom IP phones:
  - SIP PnP
  - DHCP Options
  - snom Redirection
- Additionally there is one standardized method (TR-69) that can be used.
- All scenarios require a DHCP Server which provides general network connectivity to the phones, i.e. IP address, netmask, router, DNS server, etc.

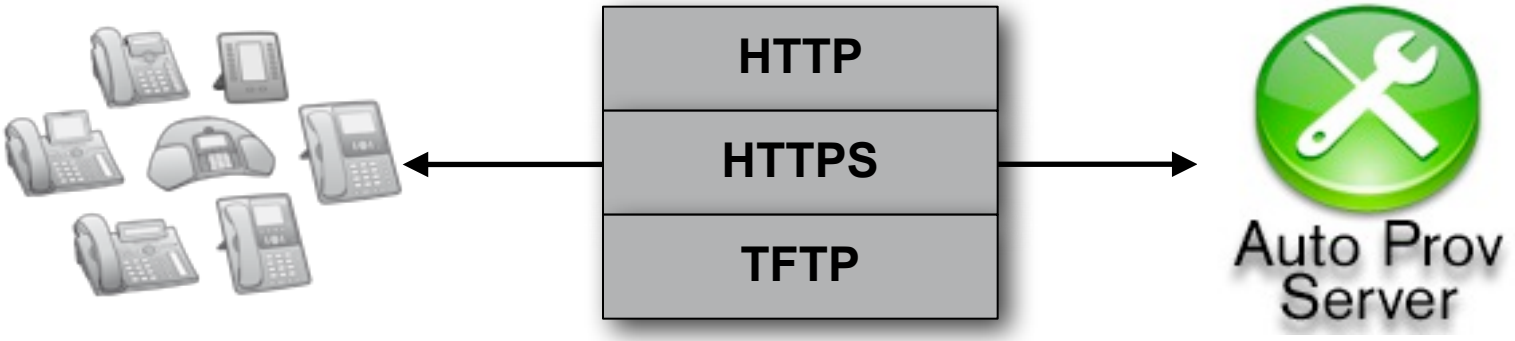
Objectives (2)

(1) Defining and setting up the appropriate AP scenario, i.e. how to “tell” a phone where to request its configuration parameters and firmware updates from on first start-up

(2) Defining and setting up the AP server (APS), i.e. server protocol, client access and security rules, availability etc.

(3) Defining and setting up the APS data structure, i.e. what is the required format of the requested configuration parameters and where to store these data

Supported APS Protocols

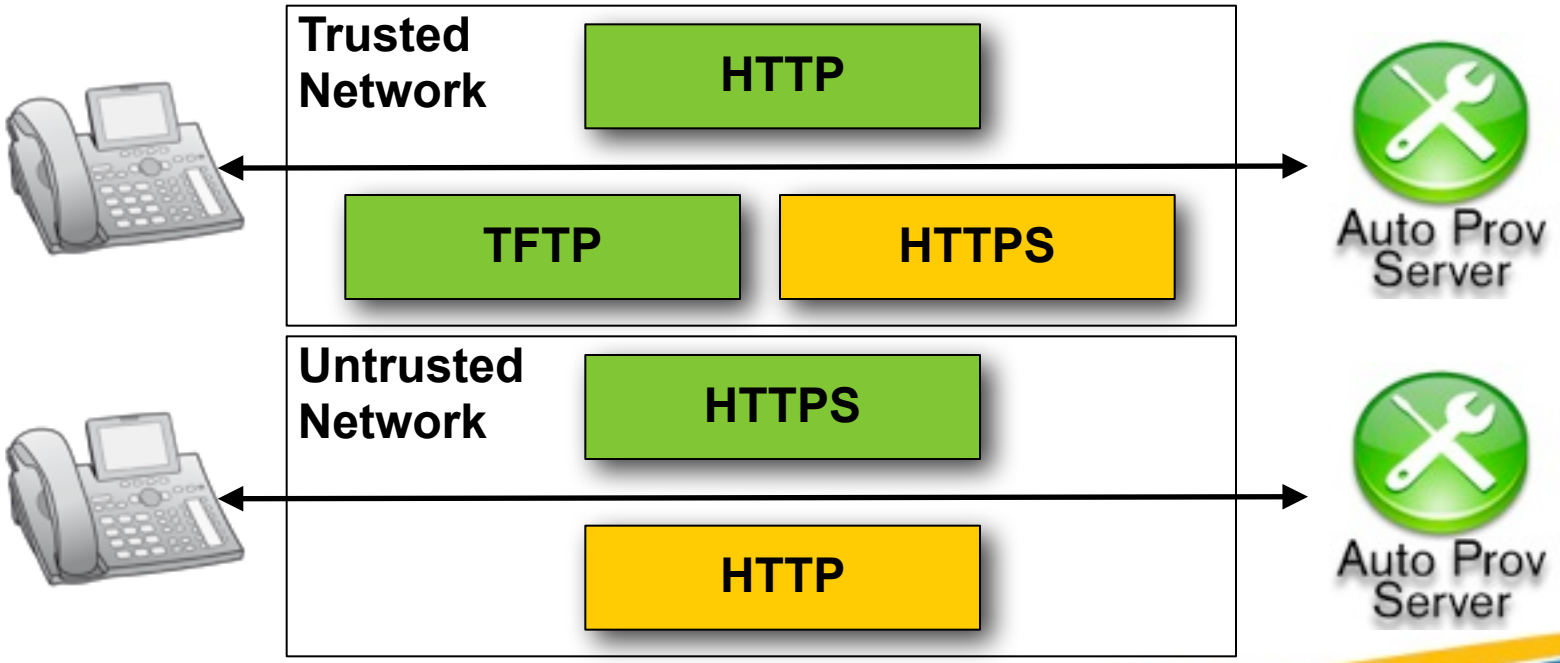


- snom IP products can request configuration files from the Auto Provisioning Server (APS) via these protocols: HTTP, HTTPS or TFTP.
- The protocol type depends on the AP scenario and the phone's firmware.
- The protocol type determines also the security of the AP process.

Authors: Alexander Feldt

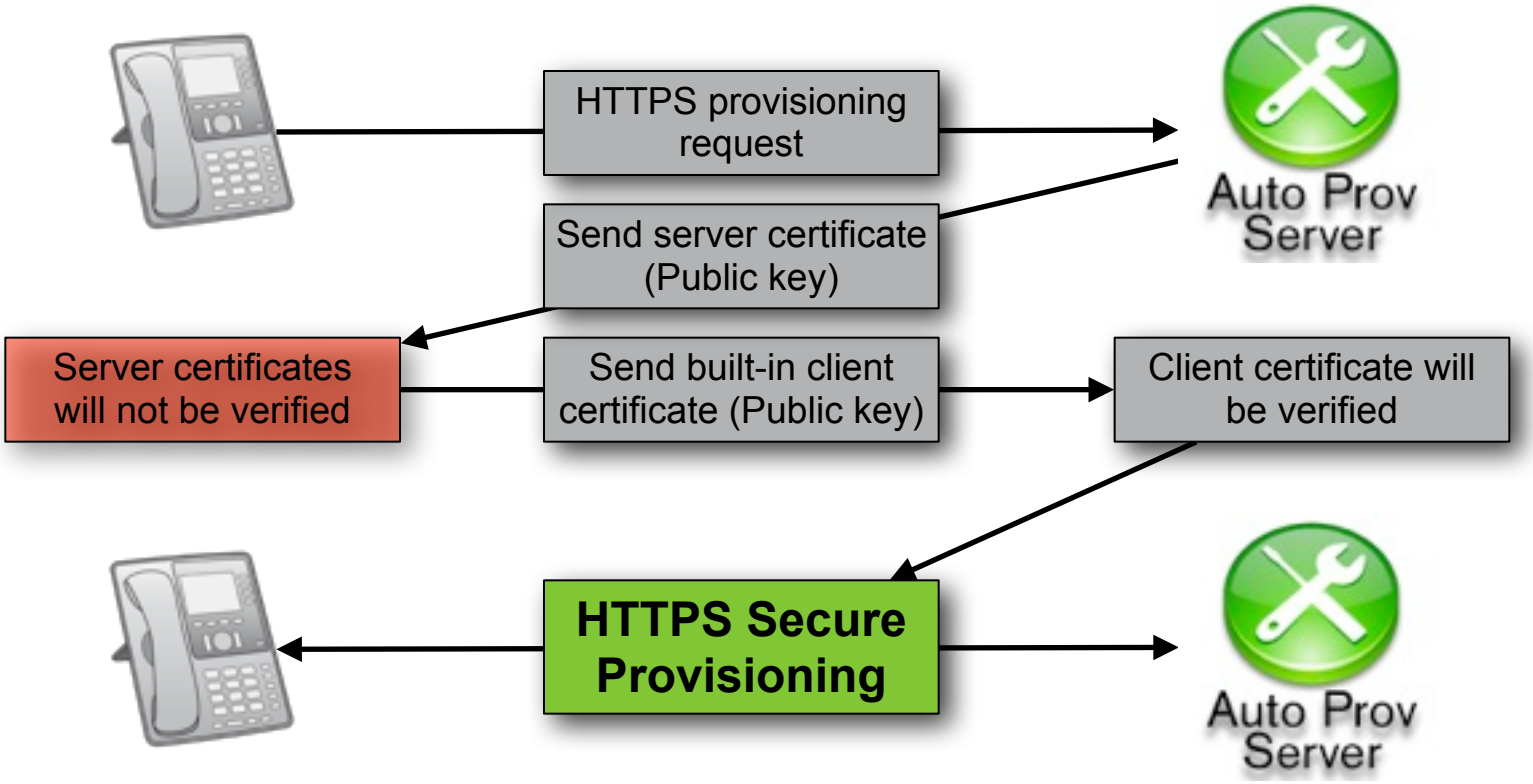
AP in Trusted/Untrusted Networks

- If phones and APS are within a trusted network, e.g. LAN, HTTP and TFTP are most suitable.
- If the network is untrusted, e.g. Internet, HTTPS is preferred.



Authors: Alexander Feldt

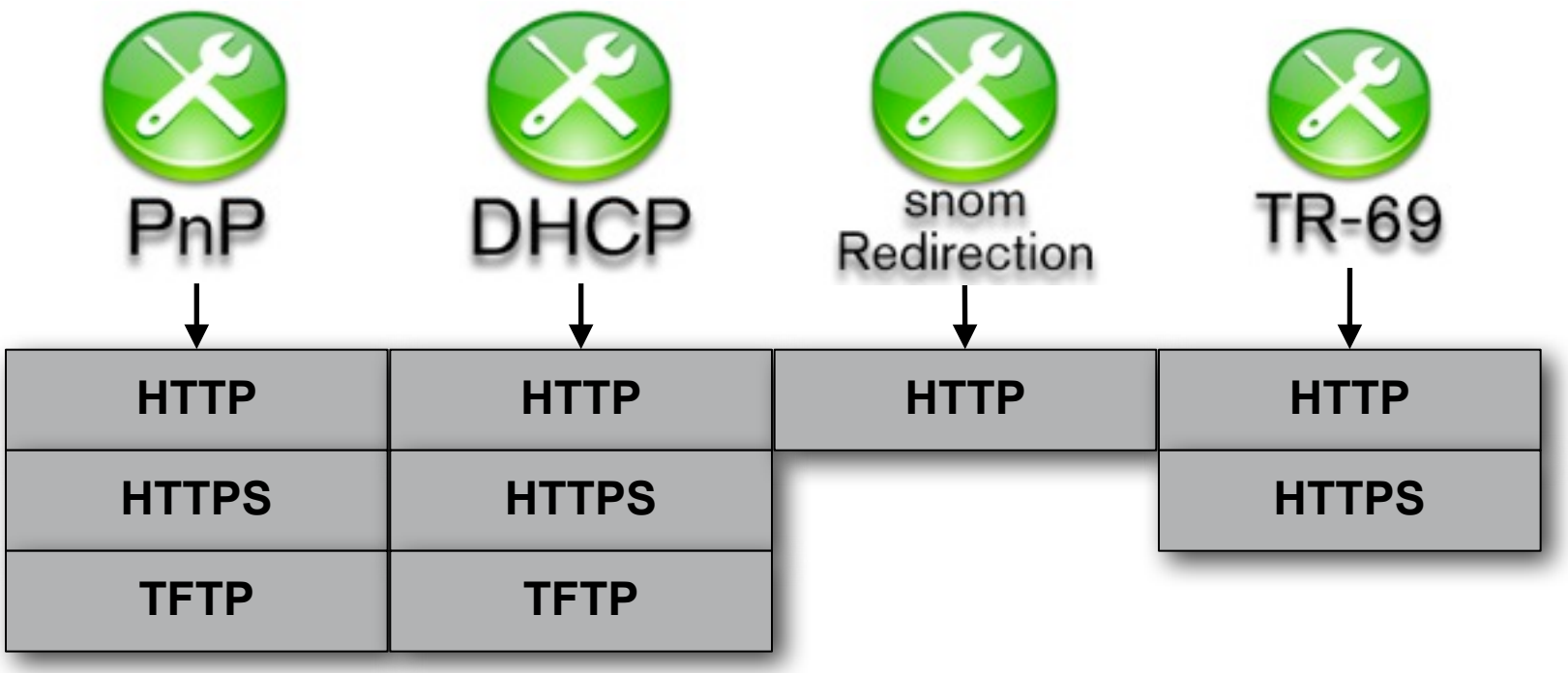
Secure Auto Provisioning



- **Full end-to-end security can only be achieved using HTTPS by means of MAC based client certificates on snom8xx.**

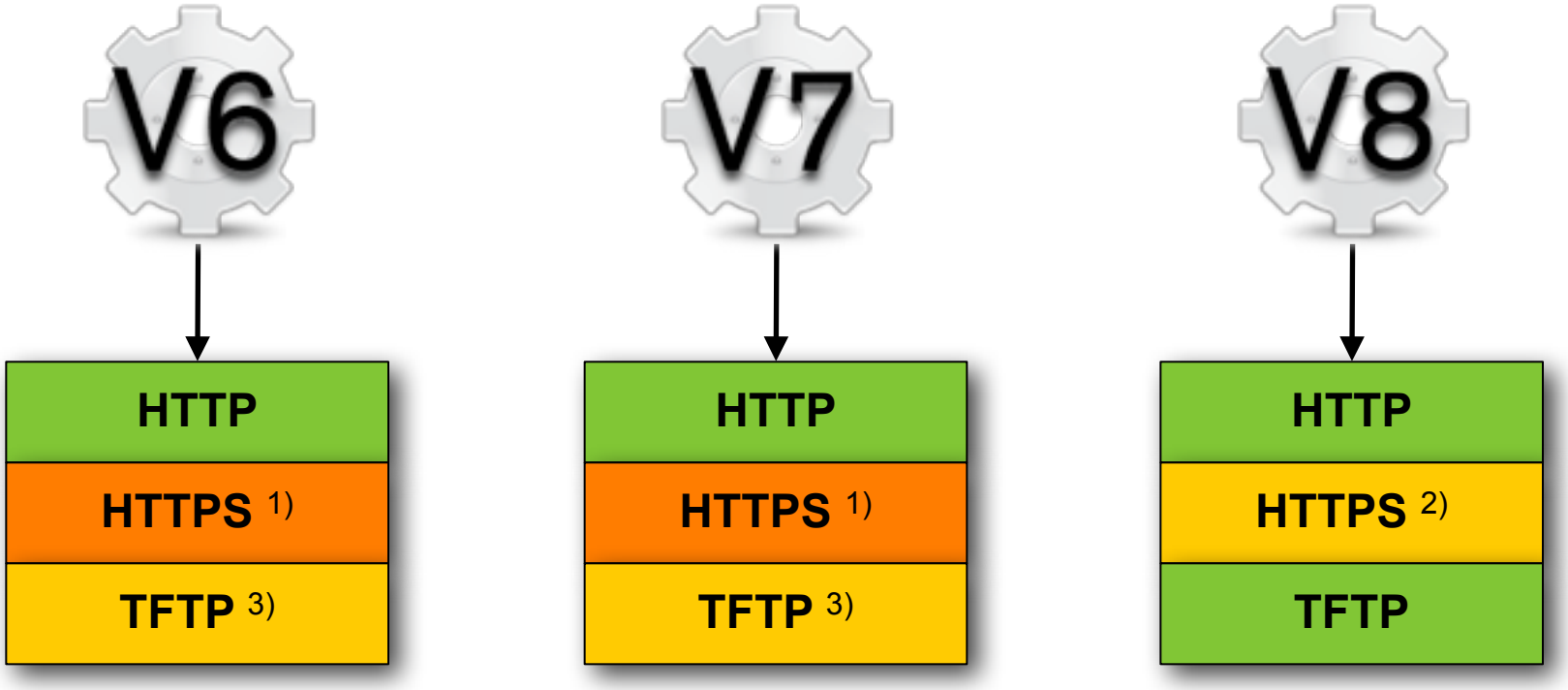
Authors: Alexander Feldt

Scenarios vs. APS Protocols



- PnP and DHCP support all protocol types.
- The default URL for snom redirection is:
  - http://provisioning.snom.com/snomXXX/snomXXX.php?mac={mac}
- TR-69 uses HTTP or HTTPS.

Firmware vs. APS Protocols



1) There is no individual client certificate on snom3x0 phones.
2) See 1). snom8xx phones do have a snom self signed individual client certificate. Also snom3x0 phones cannot perform firmware updates using HTTPS.
3) Files cannot be stored in subdirectories.

Conclusions (2)

- snom IP phones can request configuration files from the Auto Provisioning Server (APS) via these protocols: HTTP, HTTPS or TFTP.
- Depending on the AP scenario and the installed firmware certain restrictions apply.
- Only HTTPS (HTTP over SSL) guarantees secure provisioning. TFTP and HTTP should only be used for provisioning within a trusted network environment.

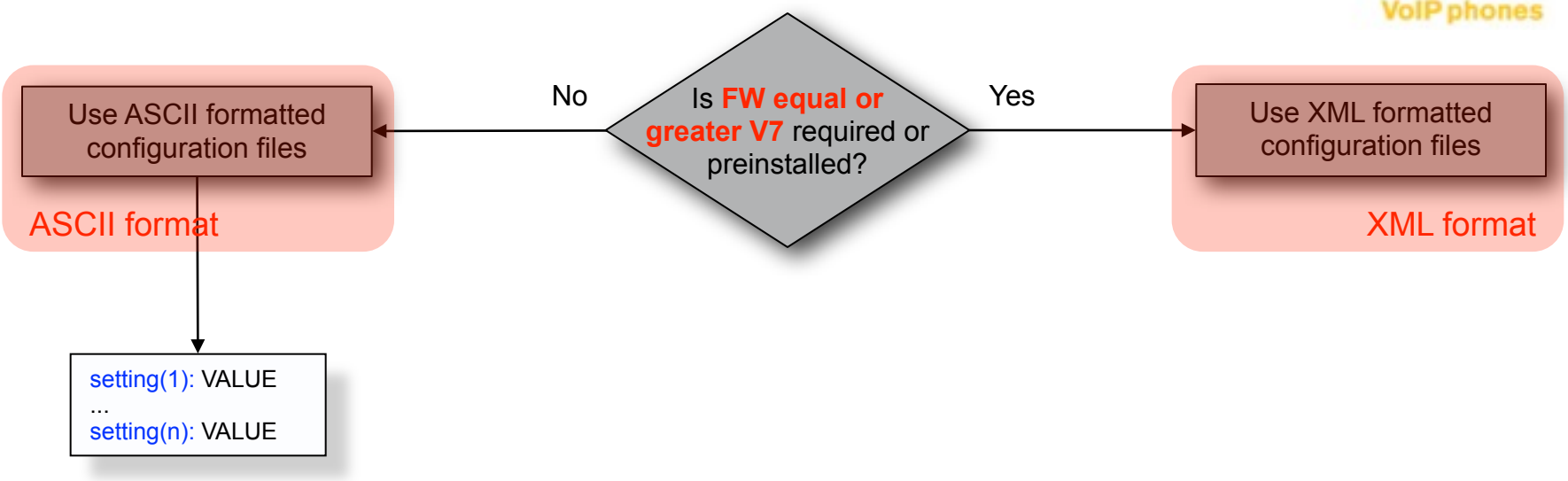
Objectives (3)

(1) Defining and setting up the appropriate AP scenario, i.e. how to “tell” a phone where to request its configuration parameters and firmware updates on first start-up

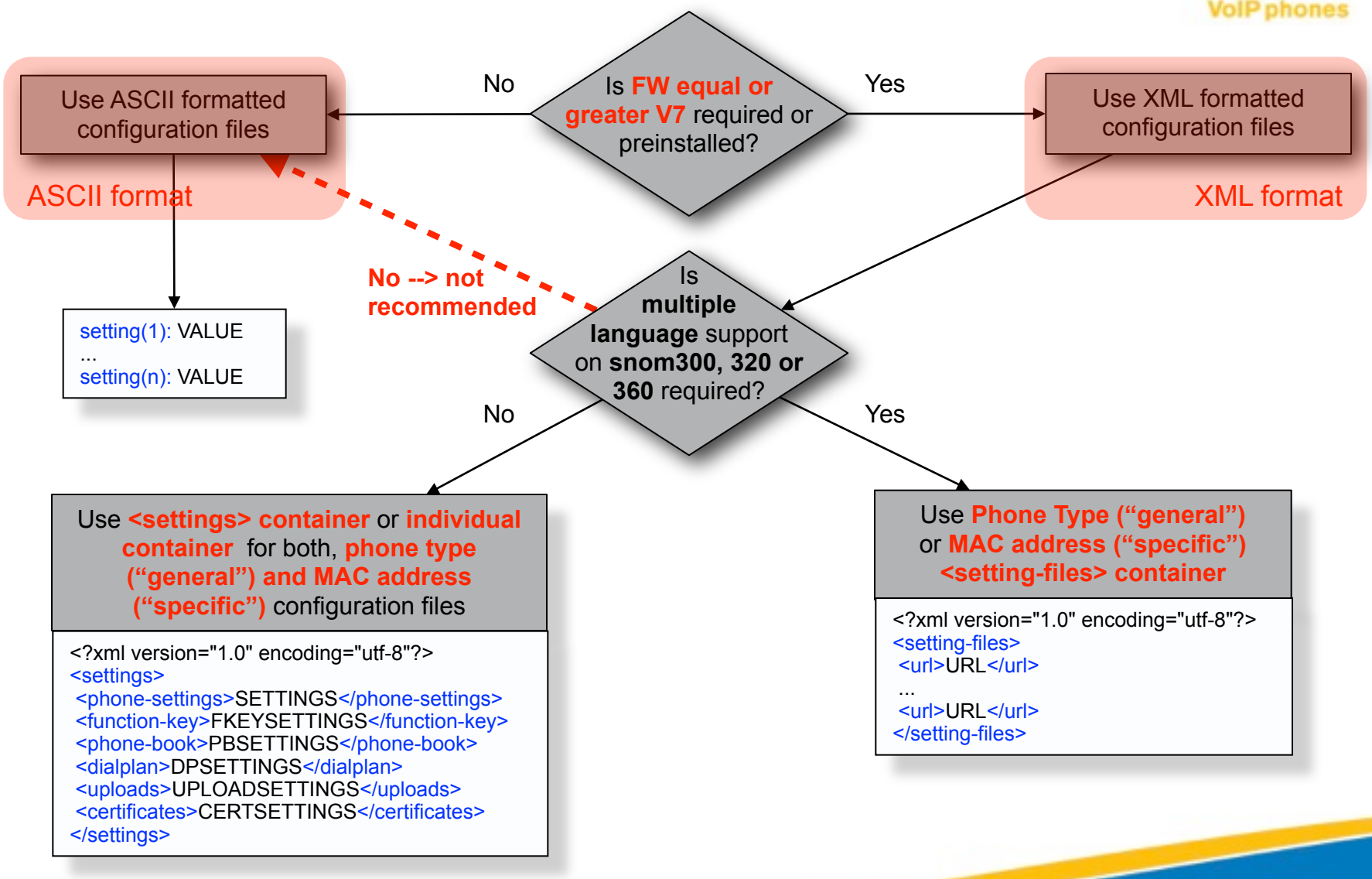
(2) Defining and setting up the AP server (APS), i.e. server protocol, client access and security rules, availability etc.

(3) Defining and setting up the APS data structure, i.e. what is the required format of the requested configuration parameters and where to store these data

APS Data Format: ASCII vs. XML?



XML Structure: Container Format?



XML Structure: <setting-files> container

- <setting-files> container can be used to upload individual setting files (**mandatory for multiple language support**)
- Useful when organizing settings in **individual files**
- The only permitted tag is <file>

```
<?xml version="1.0" encoding="utf-8" ?>
 <setting-files>
  <file url="PHONE-SETTINGS-FILE-URL"/>
  <file url="FUNCTIONKEY-SETTINGS-FILE-URL"/>
  <file url="DIRECTORY-SETTINGS-FILE-URL"/>
  <file url="XMLDIALPLAN-SETTINGS-FILE-URL"/>
  <file url="UPLOAD-SETTINGS-FILE-URL"/>
  <file url="CERTIFICATE-SETTINGS-FILE-URL"/>
  <file url="PUI-LANGUAGE-DEFINITION-FILE-URL"/>
  <file url="WUI-LANGUAGE-DEFINITION-FILE-URL"/>
 </setting-files>
```

Example XML Structure: <setting-files> container

```
<?xml version="1.0" encoding="utf-8" ?>
 <setting-files>
  <!--XML File containing selected phone settings-->
  <file url="http://aps.company.com/snom/snom300/phonesettings.xml"/>
  <!--XML File containing selected function key settings-->
  <file url="http://aps.company.com/snom/snom300/functionkeys.xml"/>
  <!--XML File containing selected directory settings-->
  <file url="http://aps.company.com/snom/snom300/directory.xml"/>
  <!--XML File containing selected XML dialplan settings-->
  <file url="http://aps.company.com/snom/snom300/dialplan.xml"/>
  <!--XML File containing selected uploads settings-->
  <file url="http://aps.company.com/snom/snom300/uploads.xml"/>
  <!--XML File containing selected certificat settings-->
  <file url="http://aps.company.com/snom/snom300/certificates.xml"/>
  <!--Multiple language provisioning V7 for snom300-360-->
  <file url="http://provisioning.snom.com/config/web_lang2.xml" />
  <file url="http://provisioning.snom.com/config/gui_lang2.xml" />
 </setting-files>
```

XML Structure: <settings> container

- **<settings> container can be used to organize settings in a single XML file**
- **The following tags are supported:**
  - <phone-settings>, <functionkeys>, <tbook>, <dialplan>, <uploads>, <certificates>

```
<?xml version="1.0" encoding="utf-8" ?>
 <settings>
  <phone-settings>PHONE-SETTINGS</phone-settings>
  <functionKeys>FUNCTIONKEY-SETTINGS</functionKeys>
  <tbook>DIRECTORY-SETTINGS</tbook>
  <dialplan>XMLDIALPLAN-SETTINGS</dialplan>
  <uploads>UPLOAD-SETTINGS</uploads>
  <certificates>CERTIFICATE-SETTINGS</certificates>
 </settings>
```

Example XML Structure: <settings> container

```
<?xml version="1.0" encoding="utf-8"?>
<settings>
 <phone-settings>
  <web_language perm="RW">English</web_language>
  <timezone perm="RW">USA-5</timezone>
  <date_us_format perm="R">on</date_us_format>
  <time_24_format perm="R">off</time_24_format>
 </phone-settings>
 <functionKeys>
  <fkey idx="0" context="active" perm="">line</fkey>
  <fkey idx="1" context="active" perm="">line</fkey>
 </functionKeys>
 <tbook>
  <item context="line1" type="none" index="0">
   <name>Adrian</name>
   <number>42965</number>
  </item>
 <dialplan>
  <template match="0" timeout="4" scheme="sip" user="phone" rewrite=""/>
  <template match="00" timeout="0" scheme="sip" user="phone" rewrite=""/>
  <template match="011....." timeout="4" scheme="sip" user="phone" rewrite=""/>
 </dialplan>
</settings>
```

XML Structure: Language File Container

- **Language file container consist of a list of language file URLs each one representing a different language.**
- **The following language file container are supported:**
  - Phone User Interface (PUI) language file container: <gui-languages> tag
  - Web User Interface (WUI) language file container: <web-languages> tag

```
<?xml version="1.0" encoding="utf-8" ?>
 <gui-languages>
  <language url="PUI-LANGUAGE-FILE-URL(1)" name="LANGUAGE-NAME(1)"/>
  ...
  <language url="PUI-LANGUAGE-FILE-URL(N)" name="LANGUAGE-NAME(N)"/>
 </gui-languages>
```

Example XML Structure: PUI Language File Container

```
<?xml version="1.0" encoding="utf-8" ?>
<gui-languages>
 <language url="http://aps.company.com/snom/lang/V8/pui_DE.xml" name="Deutsch"/>
 <language url="http://aps.company.com/snom/lang/V8/pui_EN.xml" name="English"/>
 <language url="http://aps.company.com/snom/lang/V8/pui_SP.xml" name="Espanol"/>
 <language url="http://aps.company.com/snom/lang/V8/pui_FR.xml" name="Francais"/>
 <language url="http://aps.company.com/snom/lang/V8/pui_IT.xml" name="Italiano"/>
</gui-languages>
```

```
<?xml version="1.0" encoding="utf-8" ?>
<web-languages>
 <language url="http://aps.company.com/snom/lang/V8/wui_DK.xml" name="Dansk"/>
 <language url="http://aps.company.com/snom/lang/V8/wui_NO.xml" name="Norsk"/>
 <language url="http://aps.company.com/snom/lang/V8/wui_FI.xml" name="Suomi"/>
 <language url="http://aps.company.com/snom/lang/V8/wui_SW.xml" name="Svenska"/>
</web-languages>
```

XML Structure: Language Files

- **Language files contain the language phrases.**
- **The following language file tags are supported:**
  - Phone User Interface (PUI) language files: <phrases> tag
  - Web User Interface (WUI) language files: <w_phrases> tag
- **Language files depend on the firmware version, i.e. each file is unique per firmware version. However the language files of the latest release are always backwards compatible.**

Syntax & Example XML Structure: PUI Language Files

```
<?xml version="1.0" encoding="utf-8"?>
 <phrases>
  <phrase i="INDEX(1)" n="NAME" t="TRANSLATION"/>
  ...
  <phrase i="INDEX(N)" n="NAME" t="TRANSLATION"/>
  <language i="INDEX" t="LANGUAGE_NAME"/>
 </phrases>
```

```
<?xml version="1.0" encoding="utf-8"?>
 <phrases>
  <phrase i="0" n="lang_none" t=""/>
  <phrase i="1" n="lang_language_name" t="Deutsch"/>
  <phrase i="2" n="lang_cntry_CHN" t="China"/>
  ...
  <phrase i="1180" n="lang_tls_unknown_certificate_alert_title" t="Unsichere Verbindung"/>
  <language i="0" t="Deutsch"/>
 </phrases>
```

Syntax & Example XML Structure: WUI Language Files

```
<?xml version="1.0" encoding="utf-8"?>
 <w_phrases>
  <w_phrase i="INDEX(1)" n="NAME" t="TRANSLATION"/>
  ...
  <w_phrase i="INDEX(N)" n="NAME" t="TRANSLATION"/>
  <language i="INDEX" t="LANGUAGE_NAME"/>
 </w_phrases>
```

```
<?xml version="1.0" encoding="utf-8"?>
 <w_phrases>
  <w_phrase i="0" n="wlang_iso_code" t="DE"/>
  <w_phrase i="1" n="wlang_web_language_name" t="Deutsch"/>
  ...
  <w_phrase i="1181" n="wlang_advanced_snom_clock_style_digital" t="Digital"/>
  <language i="0" t="Deutsch"/>
 </w_phrases>
```

XML Structure: Firmware Configuration File

- The Firmware Configuration File (<firmware-settings> tag) contains the firmware image URL.
- The Firmware Configuration File will only be requested if its URL had been specified by the configuration parameter *firmware_status* before. *firmware_status* should only be defined in the phone settings file (<phone-settings> tag)
- IMPORTANT: The firmware configuration file URL MUST not be specified in any container file.

Authors: Alexander Feldt

Syntax & Example XML Structure: Firmware Configuration File

```
<?xml version="1.0" encoding="utf-8" ?>
 <firmware-settings>
  <firmware perm="PERMISSION-FLAG">FIRMWARE-IMAGE-URL</firmware>
 </firmware-settings>
```

```
<?xml version="1.0" encoding="utf-8" ?>
 <firmware-settings>
  <firmware perm="">http://aps.company.com/snom/fw/snom300-7.3.30-SIP-f.bin </firmware>
 </firmware-settings>
```

APS URL: Structure

- **The APS URL value received via SIP PnP or DHCP options may have the following structure:**
  - **URL = authority**, e.g. “snom.com”, “192.168.2.1”, etc.
  - **URL = scheme://authority**, e.g. “http://snom.com”, “tftp://192.168.2.1”, “https://test.server:8081”, etc.
  - **URL = scheme://authority/path**, e.g. “http://snom.com/phones/snom320.php?mac=mac”
- **The phone will complete these URLs accordingly and request them in this order:**
  - **General** Configuration File (according to **phone model**)
  - **Specific** Configuration File (according to **MAC address**)
  - **Firmware** Configuration File (unless defined before)

APS URL: Request Order

Request phone type depending ("general") configuration file

Send phone type depending configuration file or error message

1

Request MAC address based ("specific") configuration file

Send MAC address based configuration file or error message

Auto Prov Server

2

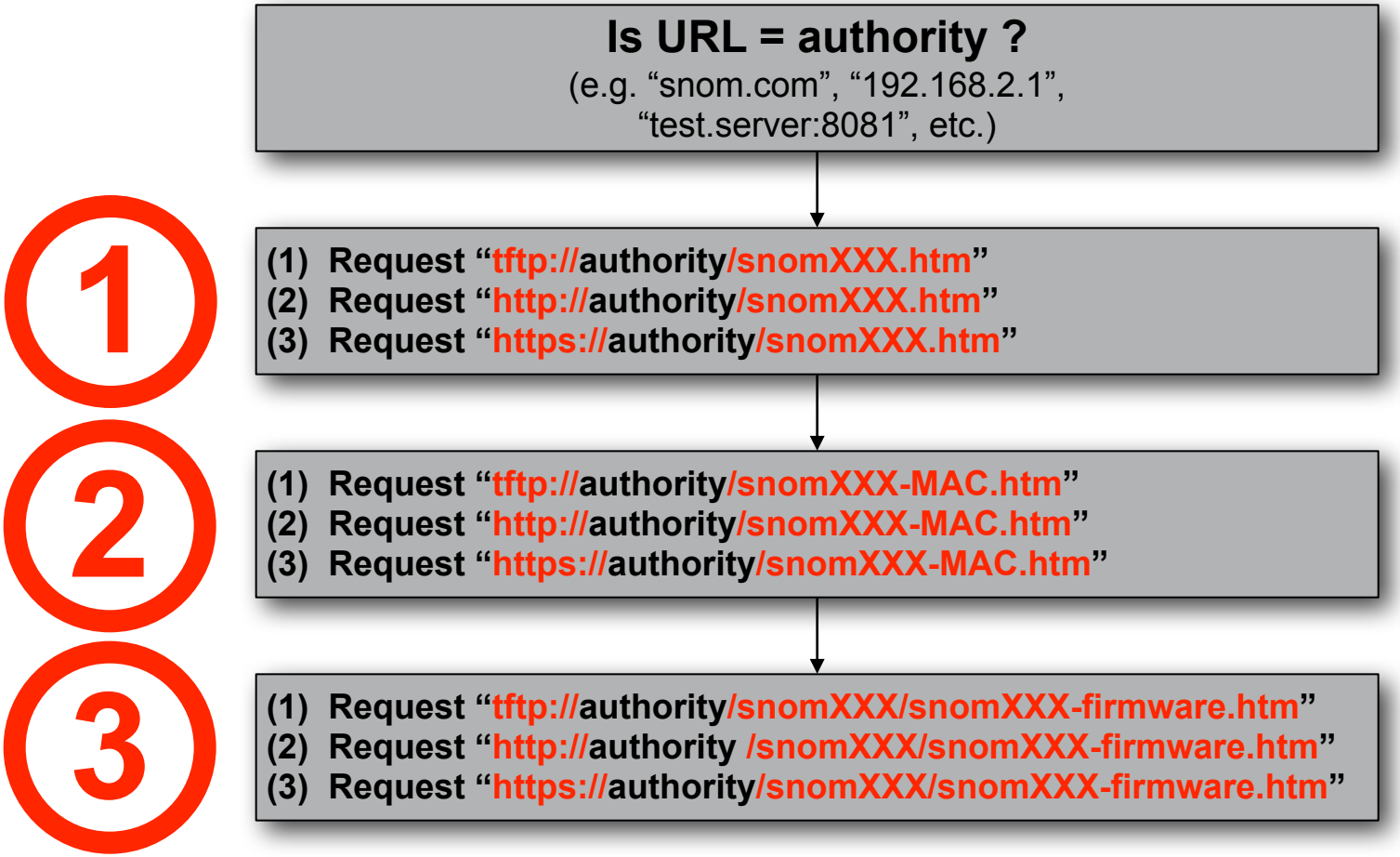
Request firmware configuration or/ and firmware image file

Send firmware configuration file or/ and firmware image file or error message

3

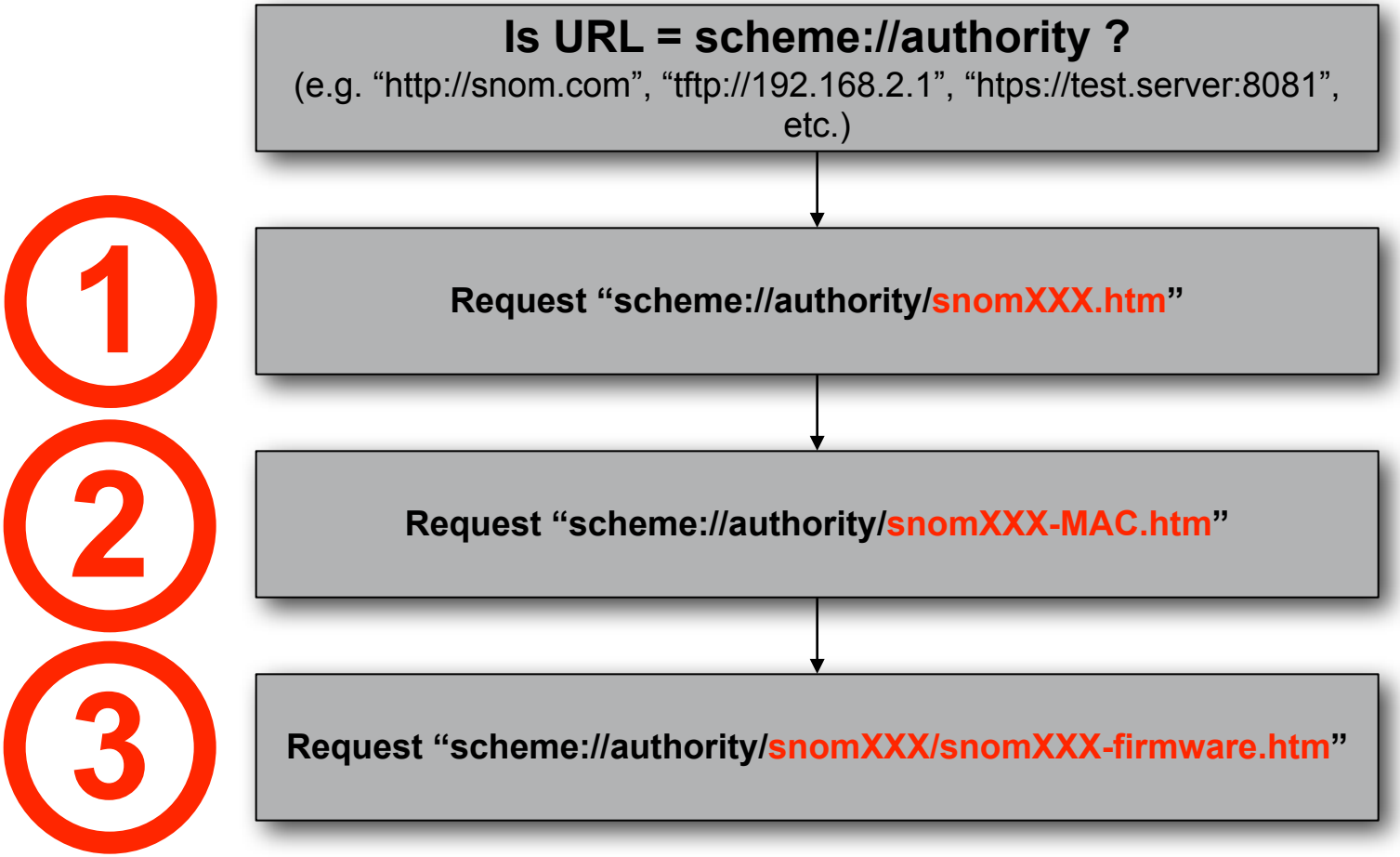
Authors: Alexander Feldt

URL = authority (RFC3968)



*snomXXX=phone type; **MAC=MAC address

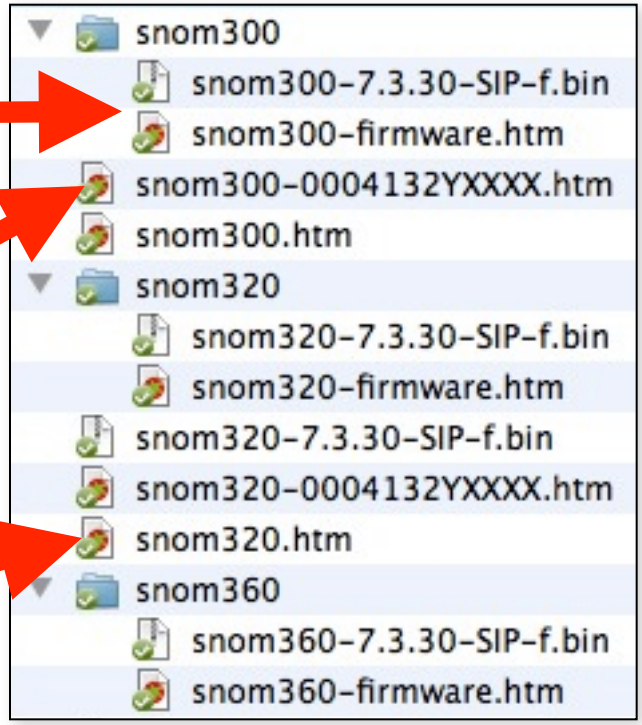
URL = scheme://authority (RFC3968)



*snomXXX=phone type; **MAC=MAC address

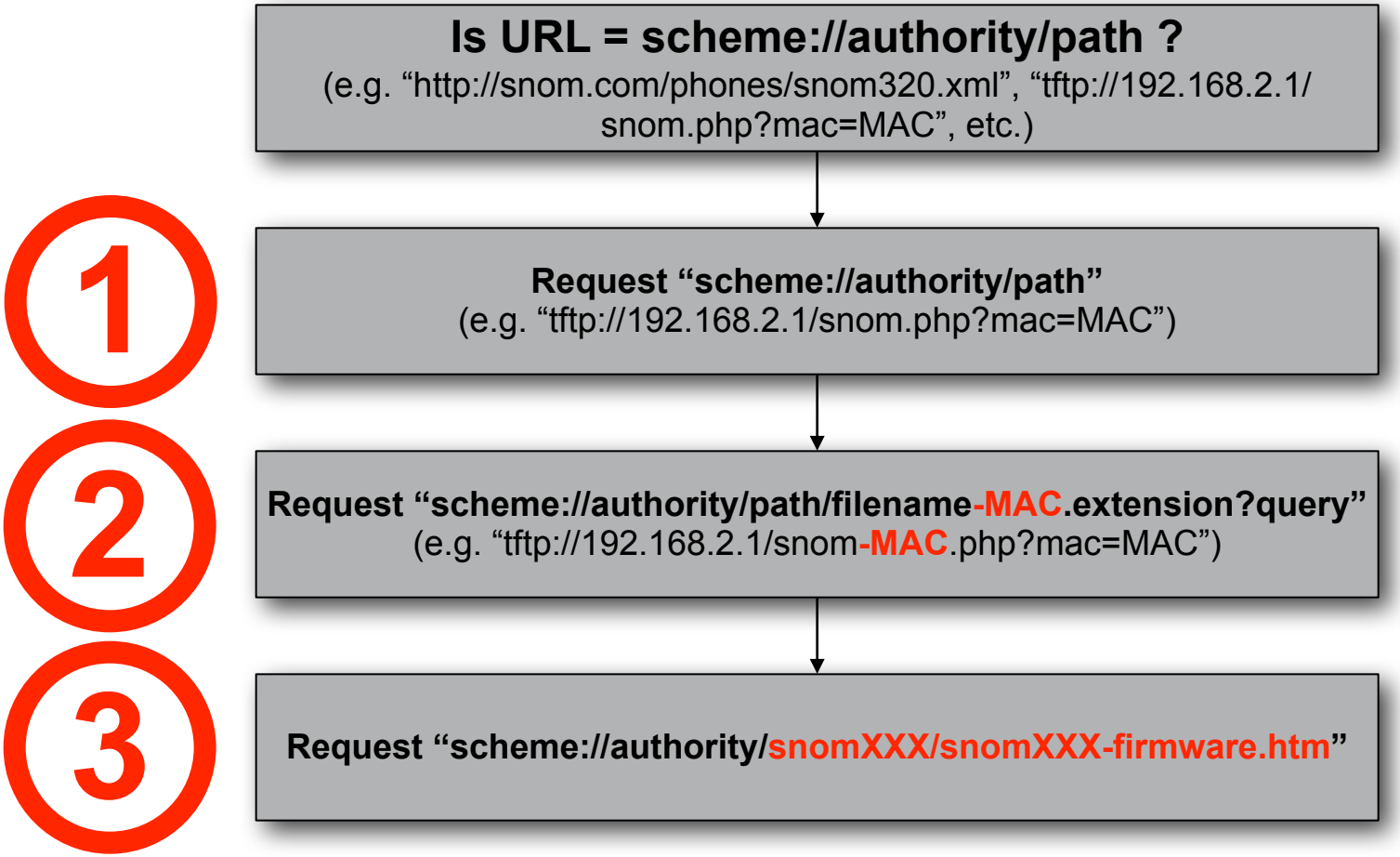
Example APS File Structure

- The structure of the APS directory may look like this:
- Firmware configuration file snomXXX-firmware.htm and firmware image file
- Specific configuration file snomXXX-MAC.htm in the root directory
- General configuration file snomXXX.htm in the root directory



Authors: Alexander Feldt

URL = scheme://authority/path (RFC3968)



*snomXXX=phone type; **MAC=MAC address

Conclusions (3)

- From firmware version 7 onwards it is highly recommended to use the XML formatted configuration file format for auto provisioning
- Only XML formatted files will allow multiple language support on snom300, 320, and 360.
- The storage location of configuration files and its structure depends on the APS URL received via SIP PnP or DHCP options
- In general the phone requests its configuration files in this order: first the phone type depending, secondly the MAC address specific and thirdly the firmware related ones.

Authors: Alexander Feldt

snomEssentials - ST131

- Thank you for attending this introduction course “Auto Provisioning Overview”. We hope you enjoyed it.
- For a deeper understanding on auto provisioning and examples we recommend seeing the complete training modules (available via classroom training):
  - ST312: Auto Provisioning via SIP PnP
  - ST313: Auto Provisioning via DHCP Options
  - ST314: Auto Provisioning via snom’s Redirection Service
  - ST315: Auto Provisioning Server & XML Structure
- Your snom Training Team

Authors: Alexander Feldt