Managing and monitoring equipment ACS connectivity with Zabbix

Presentation – Zabbix Summit 2025

Joni Taipale
Telia Company AB
Product Manager

10th October 2025

TR069/181 ACS connectivity

1. Device acquires IP-address (DHCP)
2. ACS URL included in device software
3. Device ‘calls home’ to ACS
4. HTTPS handshake with ACS
5. Management up and running (XML content exchange)

However…

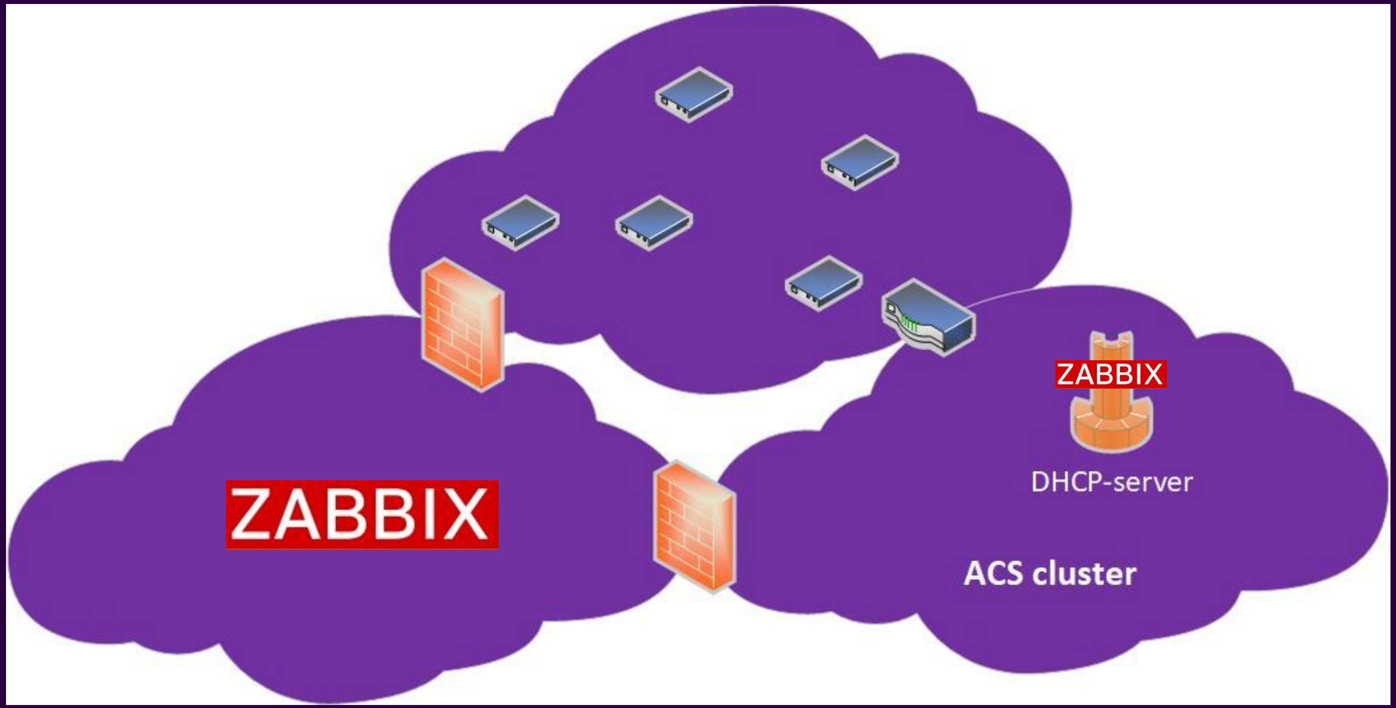
There’s always a lifecycle for everything…

Device ordering and stock management

Network topology

- Existing Zabbix environment
- Existing ACS environment
- IP connectivity & DHCP

DHCP-server knows all IP-addresses: Could we use this?



Low-level discovery based on DHCP-leases

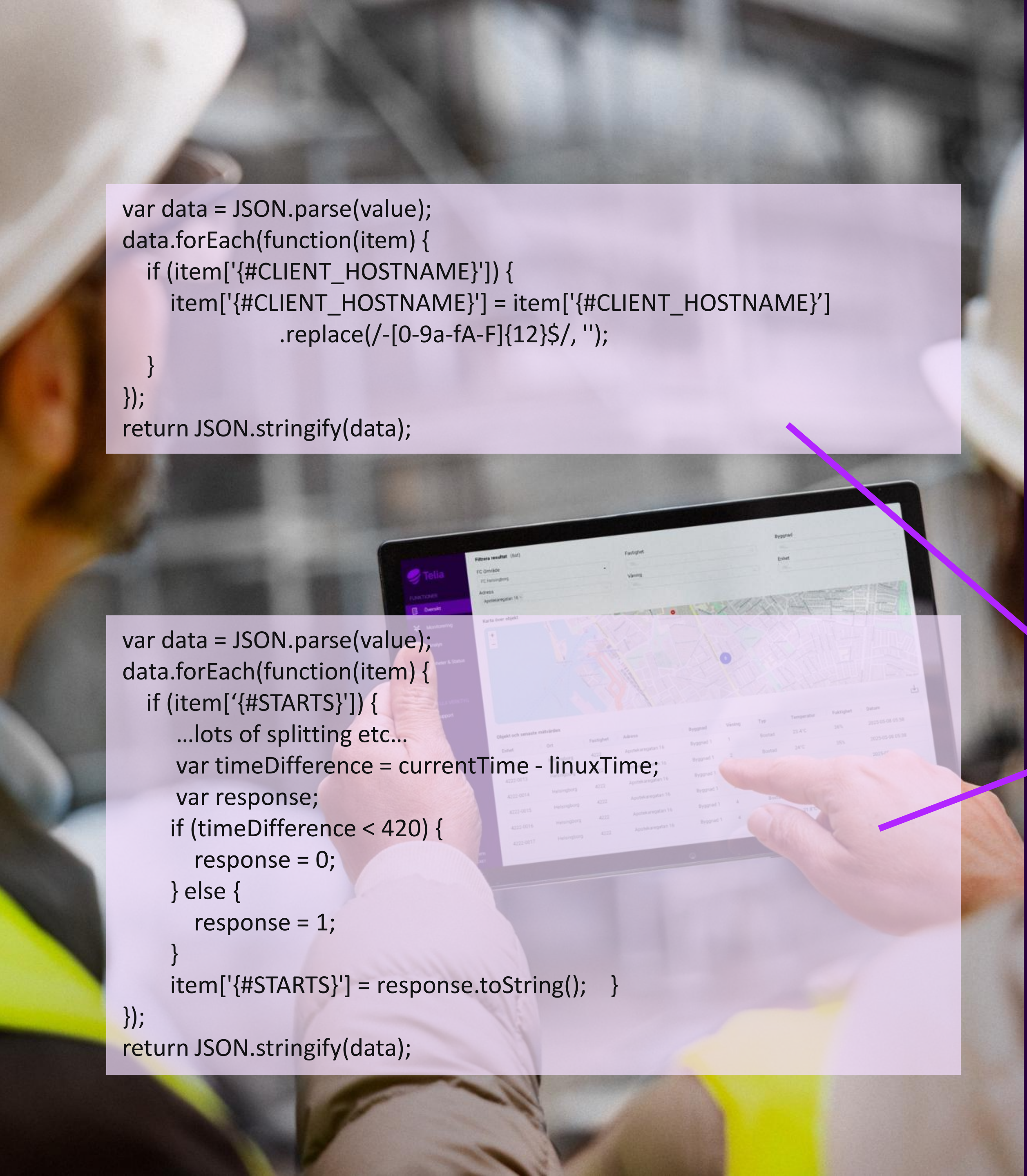
dhcpd.leases file in Linux as a basis

```
…
lease 10.2.3.4 {
  starts 4 2025/09/11 18:50:07;
  ends 4 2025/09/11 20:50:07;
  cltt 4 2025/09/11 18:50:07;
  binding state active;
  next binding state free;
  rewind binding state free;
  hardware ethernet 44:d4:37:12:34:56;
  set vendor-class-identifier = "udhcp 1.30.1";
  option agent.circuit-id "GigabitEthernet0/0/18:1666.0 ftth-xxxxxx-xxx/0/0/0/0/0";
  option agent.remote-id "AKU5000123456";
  client-hostname "GNX-580032a1b2c3";
} …
```

```
[{
"{#NR}":“1",
"{#MAC}":"44:d4:37:12:34:56",
"{#IP}" :"10.1.2.3",
"{#BINDING_STATE}":"active",
"{#REMOTE_ID}":"AKU5000123456",
"{#CLIENT_HOSTNAME}":“GNX-580032a1b2c3",
"{#STARTS}":"2 2025/09/11 18:50:07"
}]
```

ZABBIX

LLD Preprocessing



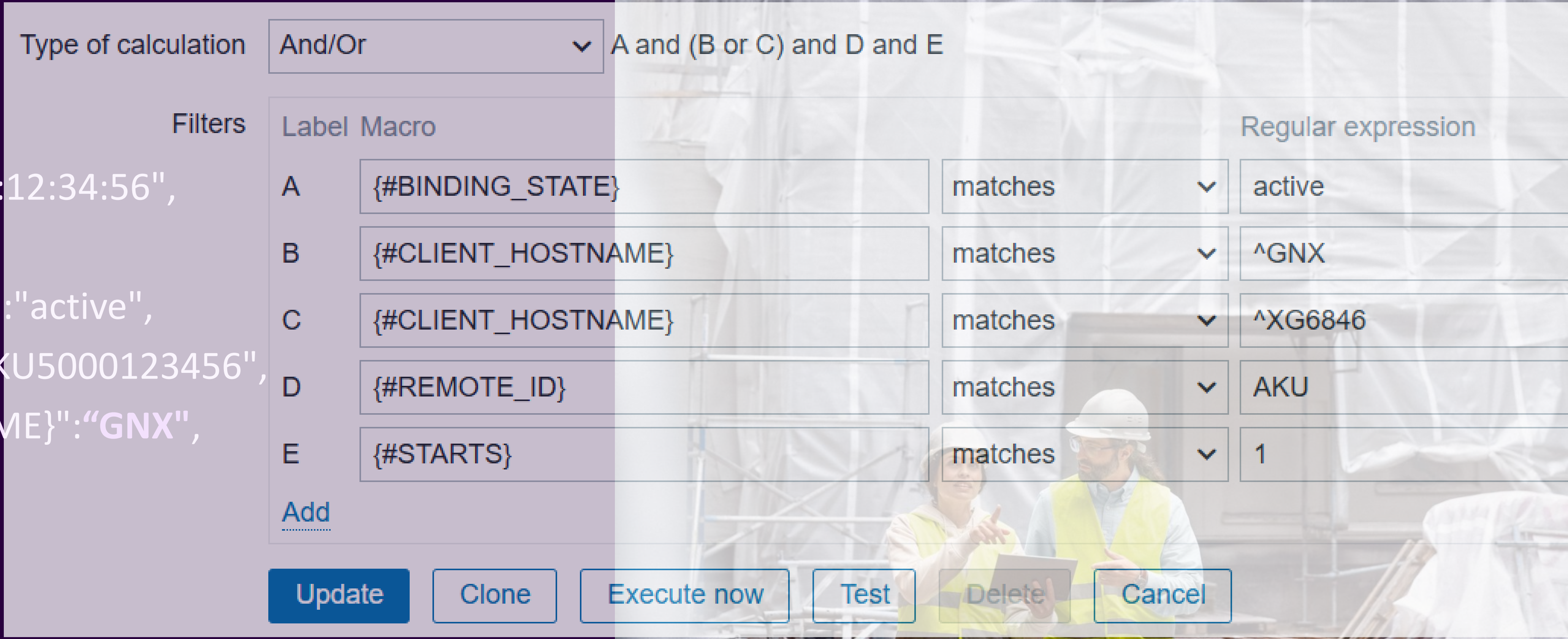
```
var data = JSON.parse(value);
data.forEach(function(item) {
   if (item['{#CLIENT_HOSTNAME}']) {
      item['{#CLIENT_HOSTNAME}'] = item['{#CLIENT_HOSTNAME}’]
                   .replace(/-[0-9a-fA-F]{12}$/, '');
   }
});
return JSON.stringify(data);
```

```
var data = JSON.parse(value);
data.forEach(function(item) {
   if (item[‘{#STARTS}']) {
       ...lots of splitting etc...
       var timeDifference = currentTime - linuxTime;
       var response;
      if (timeDifference < 420) {
         response = 0;
      } else {
         response = 1;
      }
      item['{#STARTS}'] = response.toString();    }
});
return JSON.stringify(data);
```

```
[{
"{#NR}":“1",
"{#MAC}":"44:d4:37:12:34:56",
"{#IP}" :"10.1.2.3",
"{#BINDING_STATE}":"active",
"{#REMOTE_ID}":"AKU5000123456",
"{#CLIENT_HOSTNAME}":“GNX",
"{#STARTS}":“1"
}]
```

LLD Filtering

```
[{
"{#NR}":“1",
"{#MAC}":"44:d4:37:12:34:56",
"{#IP}" :"10.1.2.3",
"{#BINDING_STATE}":"active",
"{#REMOTE_ID}":"AKU5000123456",
"{#CLIENT_HOSTNAME}":“GNX",
"{#STARTS}":“1"
}]
```



Type of calculation: And/Or — A and (B or C) and D and E

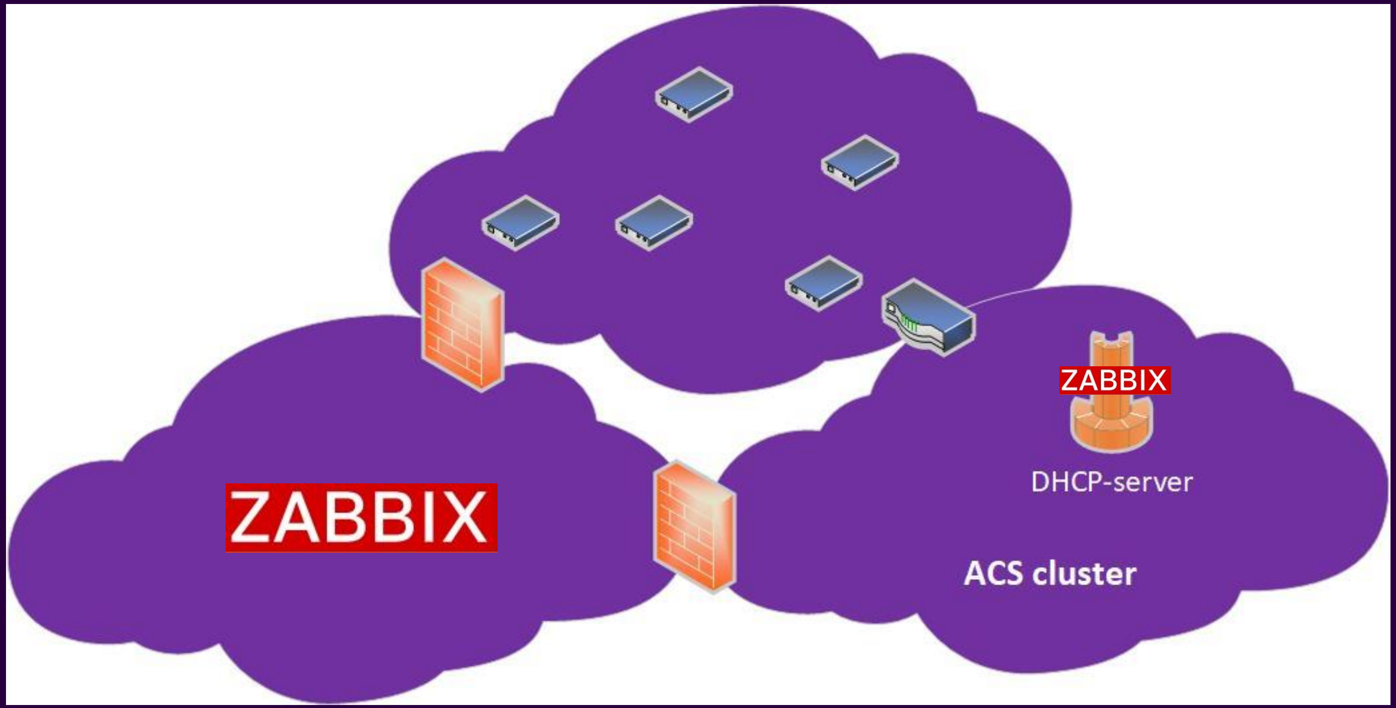
Filters

Label	Macro		Regular expression
A	{#BINDING_STATE}	matches	active
B	{#CLIENT_HOSTNAME}	matches	^GNX
C	{#CLIENT_HOSTNAME}	matches	^XG6846
D	{#REMOTE_ID}	matches	AKU
E	{#STARTS}	matches	1

Everything works as expected…?

There’s 8.6MB of DHCP-leases!!!

- Zabbix Agent is unable to process the LLD
- File mirroring between Zabbix Server and DHCP-server
- Currently more than 13k unique devices discovered
  (meaning one host with +50k items)



The rest is up to Zabbix Agent

For each unique device from LLD – Item prototypes to a single Zabbix host:

1. Device status in ACS
   - Zabbix Agent item using script to check last ‘call home’ event timestamp from ACS REST API – Error if not found
2. Last visit timestamp
   - Dependent item parsing the previous to unsigned integer in Linux time, 0 if error
3. Software upgrade result trapper
   - Zabbix trapper item storing result info if device software upgrade has been triggered
4. Software upgrade result
   - Dependent item to parse the previous to Boolean values

One Trigger

Last visit time exceeding the limit
(Trigger prototype)

↓

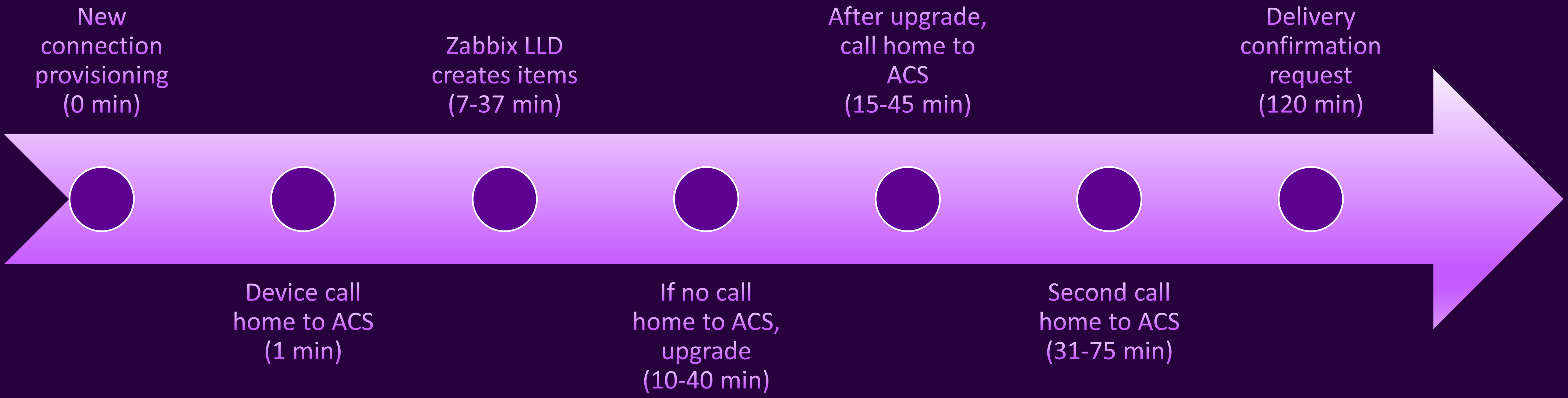
Run SW-upgrade script on Zabbix Agent

Process flow

1. Device discovered in DHCP-leases each 30min
2. Last visit time checked from ACS each 6h*
3. If threshold 90min passed in last visit, trigger software upgrade to device
4. Latest software has all what’s needed to restore ACS connectivity
5. Only if ACS connectivity is still failing, manual actions are required

* Accidentally found the new feature in Zabbix that new item is executed immediately when discovered

Complete automation flow for delivery



New connection provisioning (0 min)

Device call home to ACS (1 min)

Zabbix LLD creates items (7-37 min)

If no call home to ACS, upgrade (10-40 min)

After upgrade, call home to ACS (15-45 min)

Second call home to ACS (31-75 min)

Delivery confirmation request (120 min)

Success