Wi-Fi Network Monitoring with GÉANT WiFiMon

18-OCT-2023

Nikos Kostopoulos (NTUA/GRNET, Greece)
Elisantila Gaci (RASH, Albania)

Introduction

WiFiMon GÉANT Service

- Monitoring Wi-Fi performance as experienced by end users
- Combination of crowdsourced & hardware probe measurements
- IEEE 802.1X networks (**eduroam**): Data from RADIUS & DHCP logs for richer analysis, e.g. per Access Point (AP)

Contribution:

- Detection of Wi-Fi throughput degradation
- Determination of underperforming areas within a Wi-Fi network

→ Admins may enhance performance, e.g. by installing more *AP*s

WiFiMon vs Related Monitoring Tools

- Monitoring from the end-user perspective (***end-user experience***)
- No requirements for app installation or end-user intervention
- Centralized view of Wi-Fi performance available to the administrator

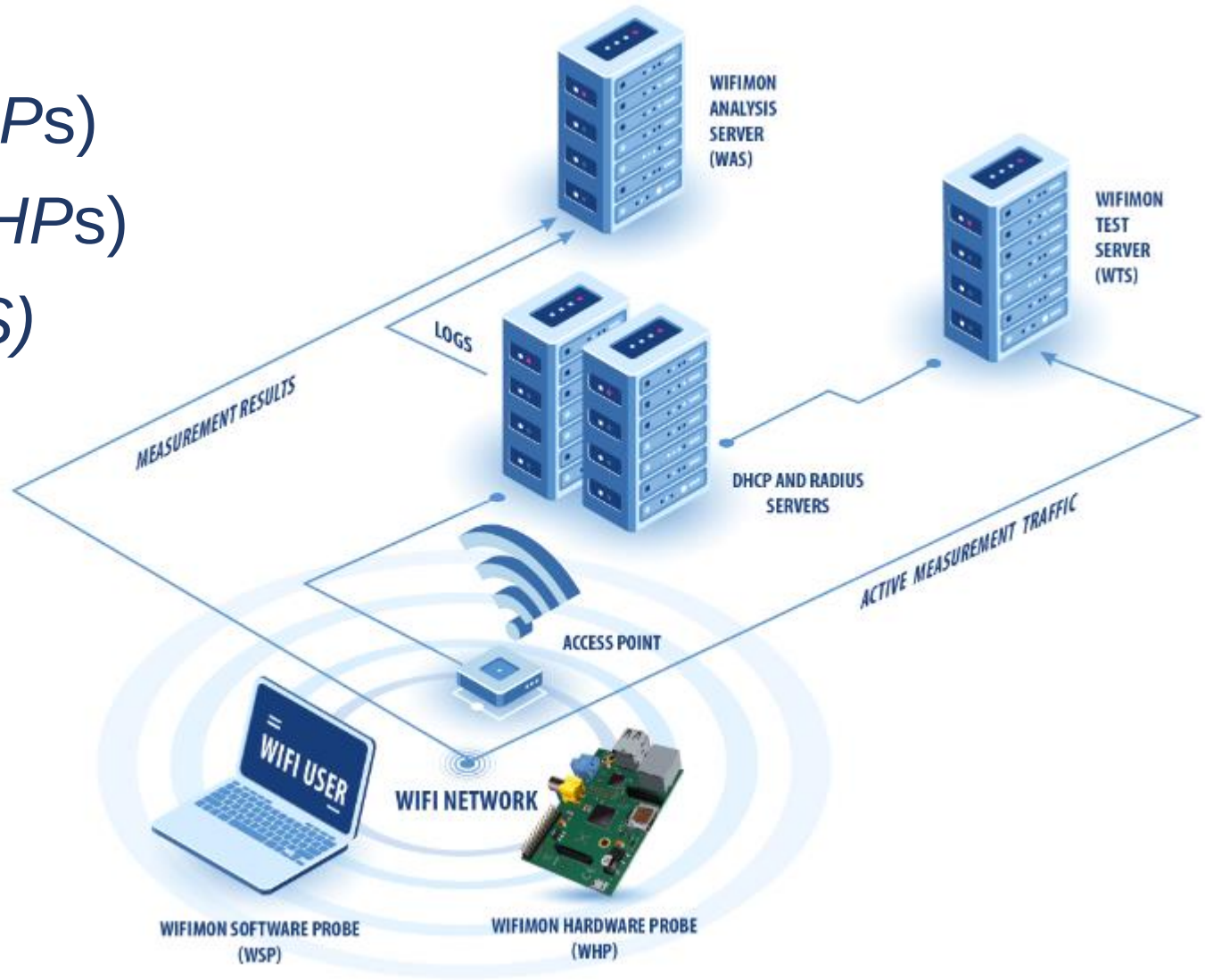
Example: WiFiMon vs Ookla Speedtest

	WiFiMon	*Ookla Speedtest*
Measurements are triggered:	**Automatically by visiting a site**	**By pressing “GO”**
Results collected by:	**Wi-Fi administrator**	**End users**

WiFiMon Operation

WiFiMon Components:

- *WiFiMon Software Probes (WSPs)*
- *WiFiMon Hardware Probes (WHPs)*
- *WiFiMon Analysis Server (WAS)*
- *WiFiMon Test Server (WTS)*



Components

WiFiMon Test Server (WTS)

Purpose: Holds code and test data for performance measurements

- Based on ***JavaScript* (*JS*)** technology
- *HTML* script tags pointing to test tools added to frequently visited sites

2 available test tools:

Akamai Boomerang	**LibreSpeed Speedtest**

***WTS* Placement:** Close to the monitored networks
(*RTT* between end devices and *WTS* included in results)

→ ***If impossible***: *WiFiMon* captures **relative** performance changes

WiFiMon Software Probes (WSPs)

End-user devices

- Crowdsourced measurements triggered against the *WTS* when users visit a *WiFiMon*-enabled site
- No requirement for additional software within user devices
- Repetitive measurements regulated via a cookie value

WiFiMon Hardware Probes (WHPs)

- Wi-Fi performance measurements from **fixed points** within the network
- Baseline throughput that complements crowdsourced measurements
- Performance measurements similar to *WSP* ones
- Additional data about monitored and nearby *ESSIDs*
- *TWAMP* Measurements, System data (CPU, memory, etc)

Triggering measurements based on *crontabs*

Tested for Raspberry Pi v3 and v4

WiFiMon User Interface (1)

Results per *WHP*

Aggregated Results

WiFiMon User Interface (2)

Dashboards available for:

- Average values
- Median values
- Maximum values
- Minimum values
- 95th Percentile values

Depicting estimations of:

- Download throughput
- Upload throughput
- HTTP ping Round Trip Time (RTT)

That may be:

- Uncorrelated
- Correlated with the available *AP*s

Sources:

- **Crowdsourced measurements**
- **Hardware Probe measurements**

Correlation with RADIUS/DHCP Logs

Logs are:

- Extracted from *RADIUS/DHCP* servers using ***Filebeat***
- Processed and transformed by ***Logstash*** in *WAS*
- Stored in ***Elasticsearch*** of *WAS*

Correlation options:

- With end-user IP address (only *RADIUS* logs)
- With end-user MAC address (both *RADIUS & DHCP* logs)

Personally Identifiable Information: IP/MAC addresses secured in transit using TLS-encrypted channels and stored hashed in *WAS* (*X-Pack*)

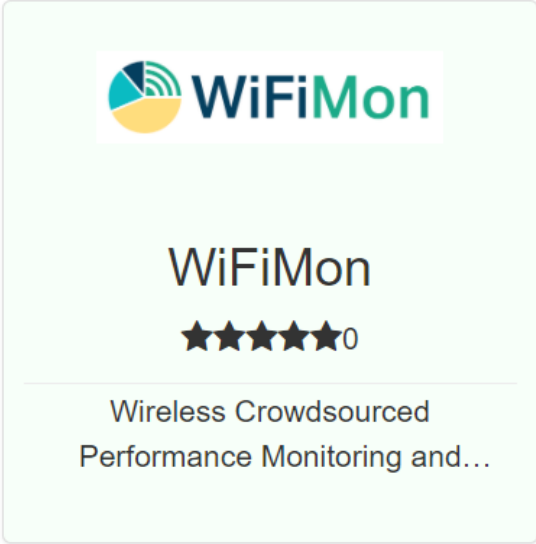
Installation

Installation Options

- Institutions install all components **on their premises**
  - **Ansible playbook** for ***WAS/WTS*** automated installation
  - All data stay within the institution premises

NMaaS Portfolio

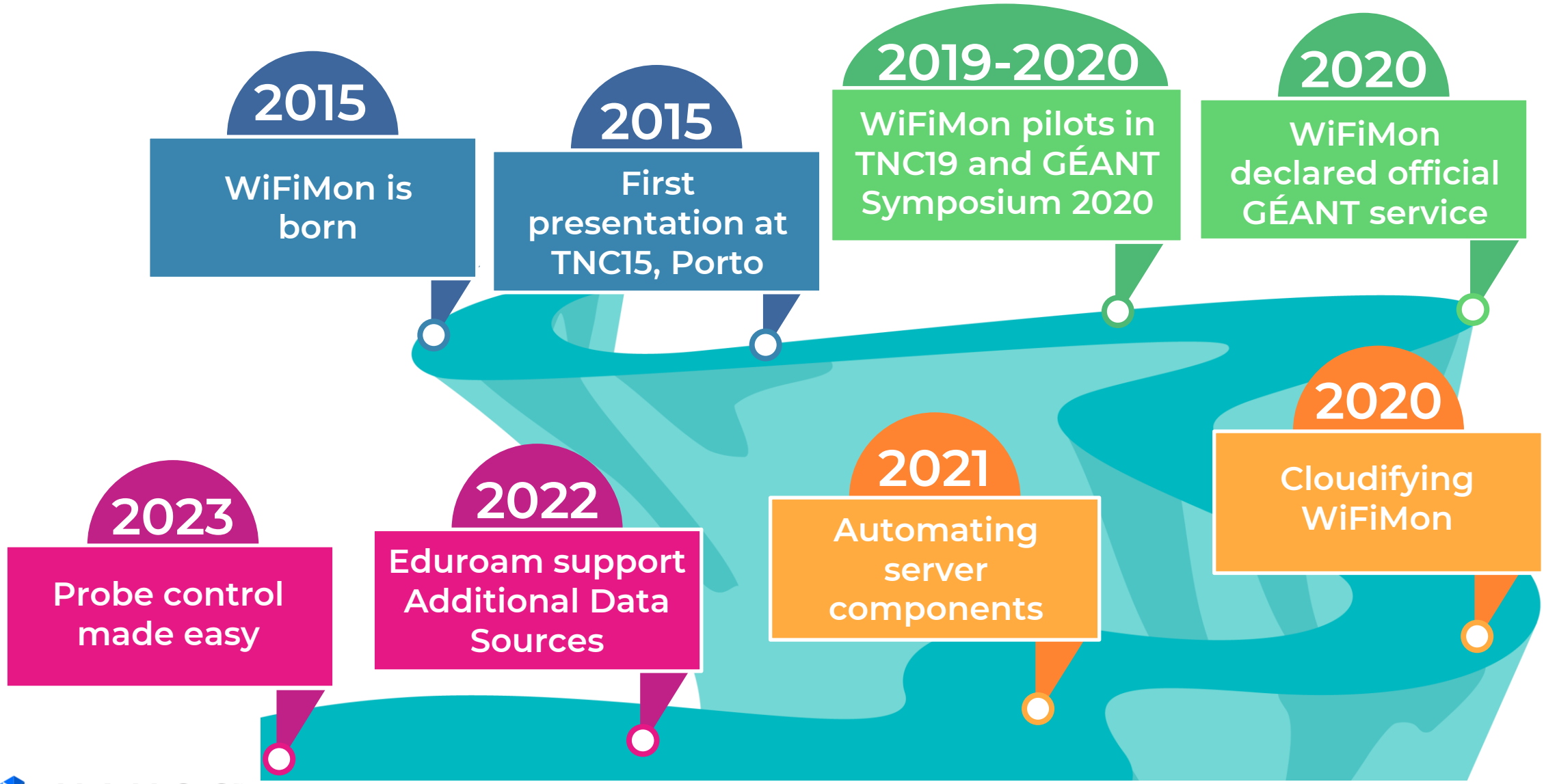
- ***NMaaS*** (simpler option for testing/trying *WiFiMon*)
  - Another *GÉANT* Service
  - *WiFiMon WAS* instance deployed on *NMaaS*
  - *WTS* installation still required by institutions
    (*should be close to the monitored network*)



Manual WAS installation: Abandoned by *WiFiMon*

WiFiMon Evolution

WiFiMon Evolution

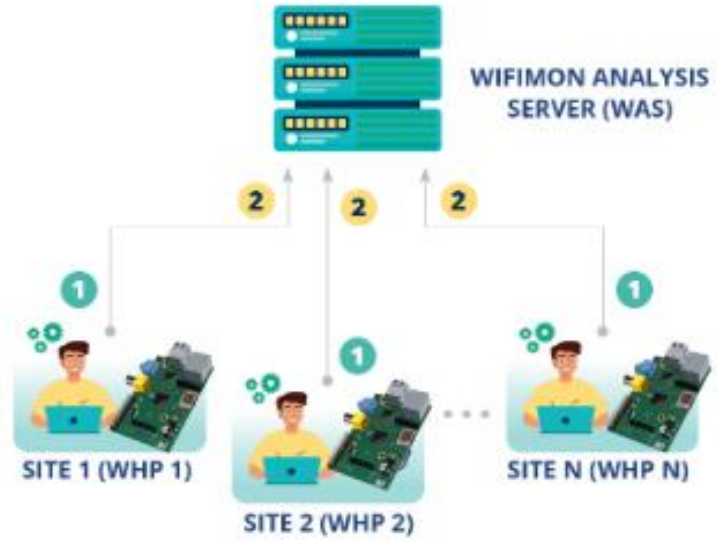


WHP Configuration & Control

Old approach

Administrator feedback demonstrated **limitations**:

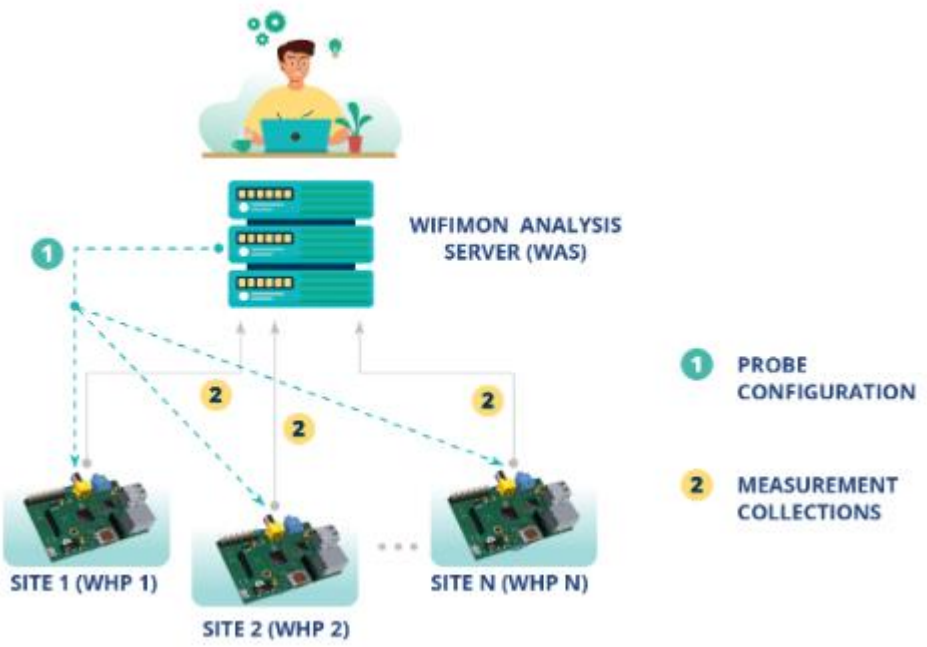
- In **NAT networks**
- In **public networks**
- Administrators edit config directly



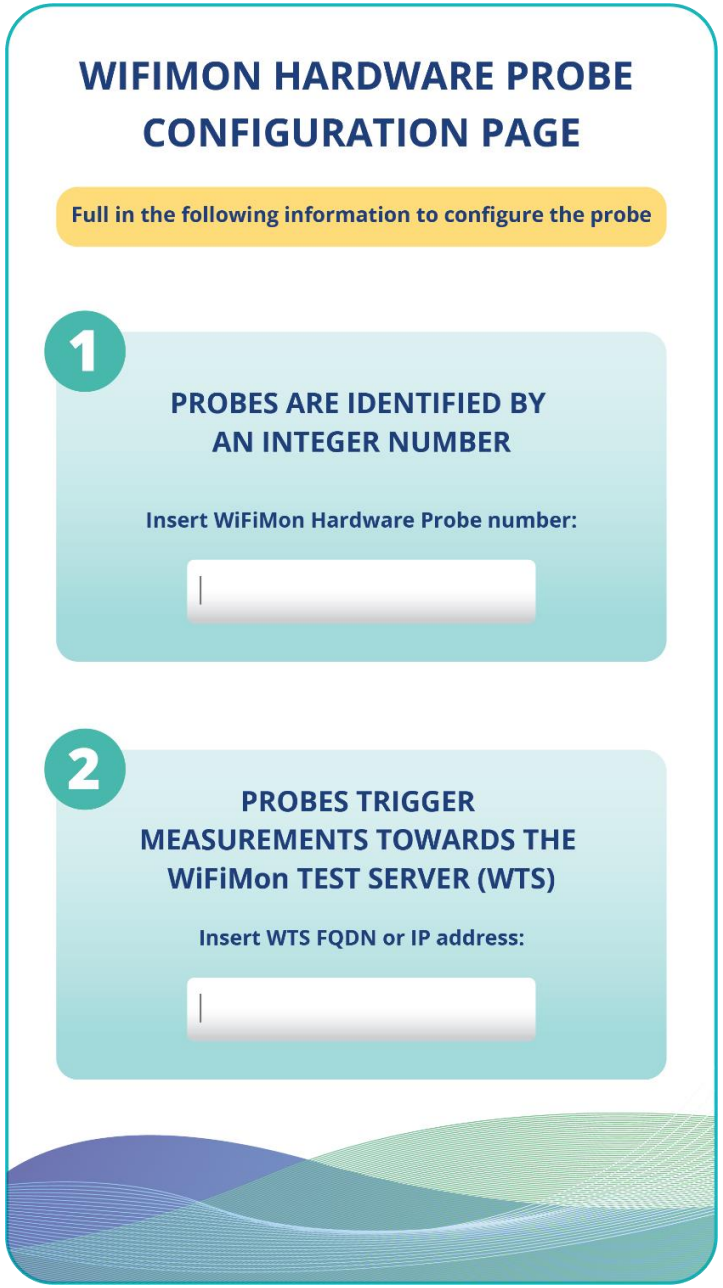
Novel approach required!!!

→ Remote & user-friendly configuration of *WHP*s from a central point (*WAS*)

→ Flexibility to control *WHP*s behind NAT networks



Configuration Made easy



Administrators (re)configure *WHPs* from the WiFiMon UI

Provided data:

- Device ID
- FQDNs/IP addresses of WiFiMon components
- Location information

Configuration files are generated based on *Jinja2* templates

Remote Configuration Made Possible

1. **_Salt_ establishes application layer communication:**
   - _WHP_s remotely configured from the WAS
   - Reconfiguration easier for _WHP_s behind NAT
   - Public IP addresses not required
     → IP space is conserved

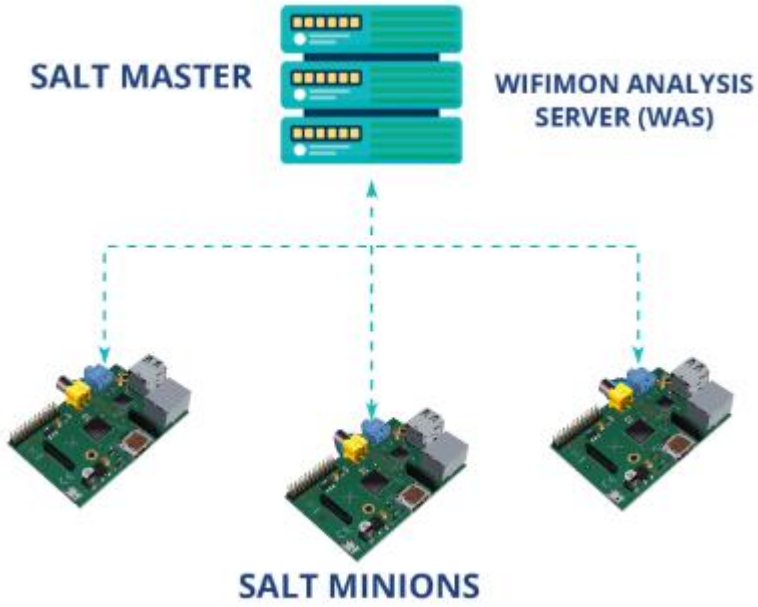
Based on _Salt_

WAS → _Salt_ Master

_WHP_s → _Salt_ Minions

2. **_Salt_ includes a _ZeroMQ_ message broker:**
   Parallel configuration regardless of the _WHP_ number

3. **Configuration files** generated from **templates** transferred from the _WAS_ to _WHP_s



Homepage: https://wiki.geant.org/display/WIF

WiFiMon mailing list: wifimon-ops@lists.geant.org

Thank you

18-OCT-2023

NANOG™