

Requirements for an Active Monitoring System

Branimir Rajtar, 5x9 Networks

Disclaimer

This is a commercial product.

However, this is not a marketing presentation.

The goal is to present technical considerations we had when building this product and what requirements we wanted to fulfill.

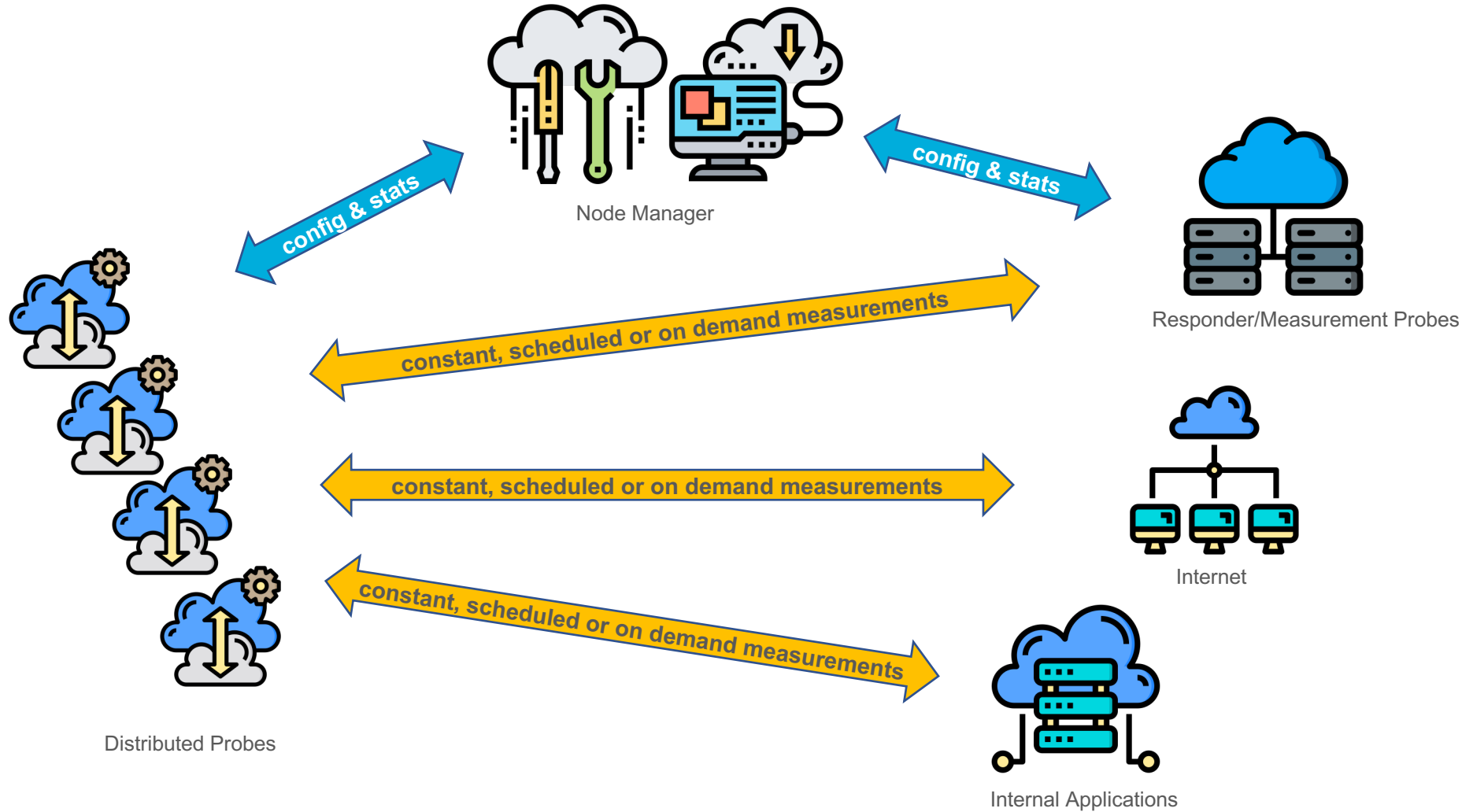
Why Active Performance Monitoring

- Active Performance Monitoring = acts as a client of the network/service/application and measures its performance
- Why would we build it?
 - Currently zero or low performance visibility
 - Existing tools are specialized, not flexible and expensive
 - Long time to pinpoint cause of issues

System Architecture

- Node Manager
 - Central component for automated Probe configuration and management
 - Measurement data collection, performance visualization and alerting
- Framework
 - Enabler for remote Probe configuration and management, SW modules distribution, data collection and processing
- Probe image or package for Linux OS
 - SW with zero-day configuration required to contact Node Manager
 - Runs Measurement SW modules and reports measurement raw data to Node Manager
 - Can run on any HW and Linux distribution
- Measurement modules
 - Standard and custom developed Probe modules responsible for specific measurement or set of measurements

System Architecture



System Configuration and Visualization

Home Services admin

Nodes Templates Alarms SW modules Locations Configuration Object service Monitoring

Probes / Managed probes

>> ID: 2 >> HW ID: dc:a6:32:1f:70:43 >> Name: Telekom-mobile-customer-01

Update Reboot Blink Shutdown Delete

Probe network management

Probe details

Probe data

Probe permissions

Assigned modules

mobileframework (v1.0.0)	default	running config	module config	graph	***
dns (v1.0.0)	mobile				
httpprobe (v1.0.0)	mobile				
icmplatency (v1.0.0)	mobile				
twamp (v1.0.0)	mobile				
twampreflector (v1.0.0)	default	running config	module config	graph	***
httpprobe (v1.0.0)	int-services	running config	module config	graph	***
dns (v1.0.0)	mobile	running config	module config	graph	***



Issues we faced

- Do we provide hardware?
- Scaling – what if somebody wants to deploy thousands of probes, how to configure them?
- Do we focus on one access technology or multiple? How to manage it?
- How to handle security?
- Would the user need training to configure the system?
- How to develop new modules as fast as possible?

Do we provide hardware?

- No, we're a software company!
- Use existing off-the-shelf hardware and virtualize as much as possible
 - All measurement modules are based on Linux
- Optimize module footprint – write in C, Golang
 - Support for single-board computer (e.g. Raspberry Pi) and CPEs
- Only hardware limitation – HATs for measurements of mobile networks

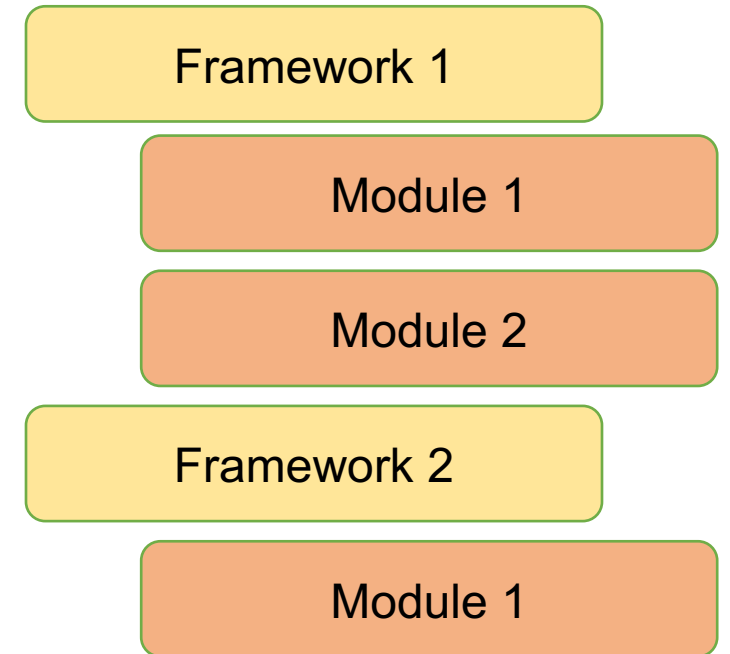


Scaling

- How to deploy and configure thousands of probes?
- Call home
 - Once turned on, the probe contacts centralized Node Manager
- Zero touch provisioning
 - Use templates to define probe configuration
- Pull model
 - Probes „pull” configuration changes, rather than changes being „pushed”
- Machine Learning
 - Do not define alarming tresholds manually
- Proven technologies and open-source products
 - Grafana, ActiveMQ, Jenkins, etc.

Measure over different access technologies

- Concept of Frameworks
 - Mobile network
 - Ethernet
 - PPPoE
 - VPN
 - WiFi
- All modules are the same, running over a specific framework
 - Speedtest, TWAMP, DNS, HTTP, voice, DHCP, etc.
 - Framework-specific modules – e.g. Circuit-switched voice, SMS

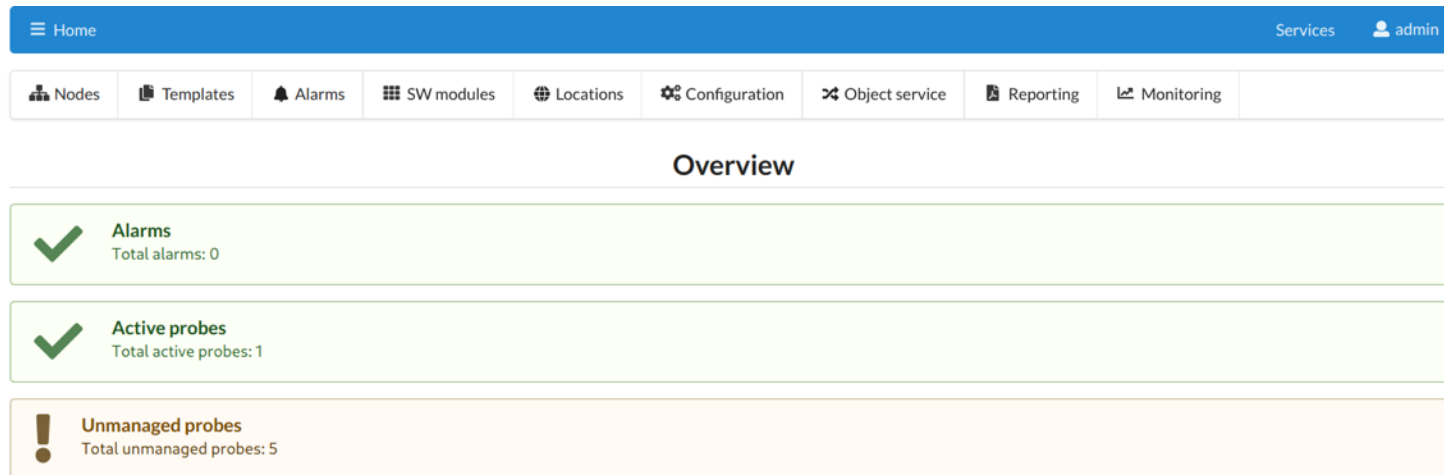


Security

- Node manager can be deployed in public or private cloud, communication to the probes can be via Internet
- All measurement modules are signed
- Communication is encrypted using SSL
- Encryption keys are unique per-customer

Improve Customer Experience

- How to enable customers to use the system on their own
- Only GUI is available
 - GUI has tooltips to help with configuration
 - No new CLI to learn
 - Simple dashboard
- Template concept
 - Easily apply existing configuration to new probes
- Grafana
 - Powerful and flexible visualization tool



The screenshot displays a web-based monitoring dashboard. At the top, there is a blue navigation bar with a 'Home' menu icon on the left and 'Services' and 'admin' (with a user icon) on the right. Below this is a horizontal menu with icons and labels for 'Nodes', 'Templates', 'Alarms', 'SW modules', 'Locations', 'Configuration', 'Object service', 'Reporting', and 'Monitoring'. The main content area is titled 'Overview' and contains three summary cards. The first card, 'Alarms', has a green checkmark icon and shows 'Total alarms: 0'. The second card, 'Active probes', also has a green checkmark icon and shows 'Total active probes: 1'. The third card, 'Unmanaged probes', has an orange exclamation mark icon and shows 'Total unmanaged probes: 5'.

Module development

- Standardize and automate!
- Standardize module structure
 - Define inputs and outputs of new module in well-known format (JSON)
 - Frontend automatically generates configuration input according to module JSON
 - Output defines Grafana panel
 - Enables easier onboarding of new developers
- Automate testing and deployments – Jenkins
 - Create and update deployment script
 - Build pipelines for testing
 - Automatically deploy in local lab and test new stuff

Key Takeaways

- It is important to think about possible issues before writing the first line of code
 - Writing code is easy, the idea matters
- Automate as much as you can
 - The tools are there, just use them
- Talk to people
 - Experience matters

Thank you!