Managing 1500 routers with a single click

A network automation tale

www.unimus.net

Presenter information

- Tomas Kirnak

  System Architect

  MikroTik Certified Trainer

  MikroTik Certified Consultant

Unimus

Network backup and management solution

Come visit us at our stand!

Why are we talking about any of this?

- Network automation is the way to go forward, and there is sadly little material around on how to do it properly.
- I recently completed a project that deployed 1500 MikroTiks in a nation-wide network with full automation.
- So lets talk about how automation can make a project like this better.

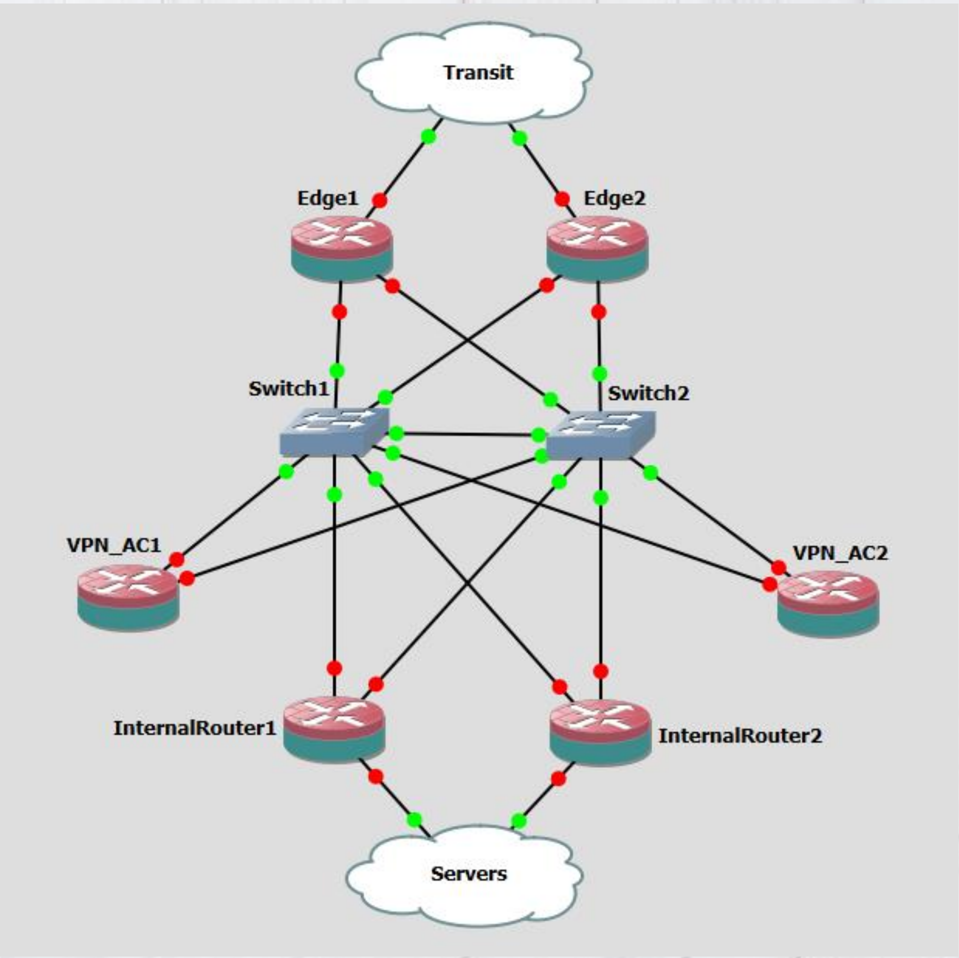
Disclaimer

- The implemented solution is under NDA, so I can not show you any actual screenshots / demo of the system nor a demo of the actual automated provisioning of a device.
- I did however receive approval to talk about how it was implemented, what we used and how we used it, and therefore to explain what is needed to implement full automation like we did in this project.
- Massive thanks to the people that can not be named (due to the NDA) for allowing me to present this information.

Lets talk about network architecture

How are we going to do this?

Network architecture talk - backend



- Your datacenter
  - You may or may not want the VPN ACs
  - Internally, you may have multiple subnets (dmz, mng, etc.)
  - Your servers (and SAN) should also be redundant and redundantly connected
  - You should have a failover site

Network architecture talk - transit

- So what about transit – how do the routers connect users to your services?

- You have 3 options:
  - Build your own transit network
  - Buy “WAN” services from a carrier/transit provider
  - Use the internet

- If we are talking about a nation-wide network, option 2 will provide the best stability, SLAs, consistency and services.
- Option 3 will be the cheapest one, albeit the most problematic one.

Network architecture talk – users

- What about the user-terminating subnets?

- This depends on:
  - the choice of transit method
  - are you building a new network, or inter-connecting existing sites
  - Do you have full control over “customer” sites

Network architecture talk – our case

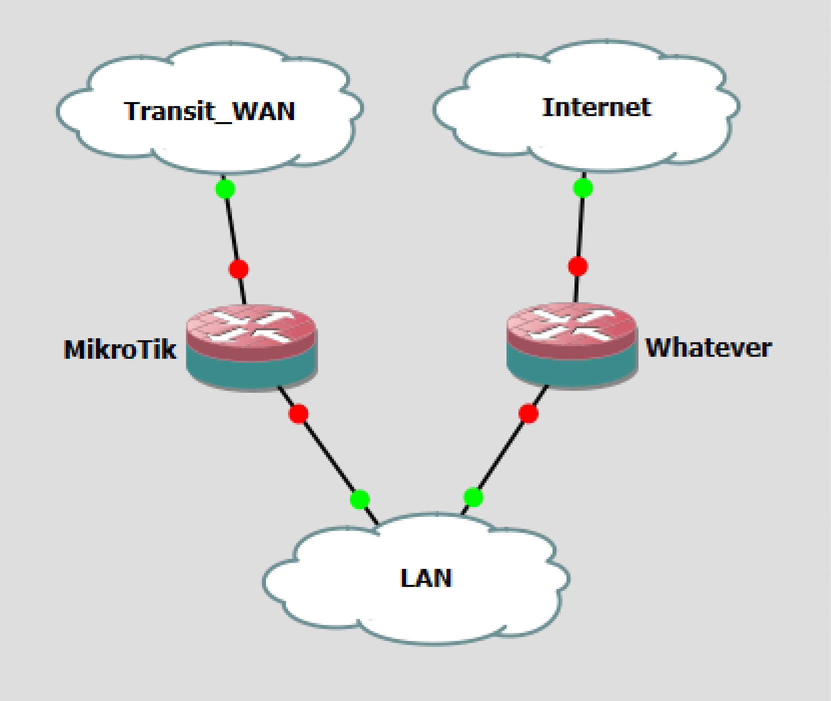
- In our case, we were inter-connecting existing sites
  (meaning, they were already setup “somehow” – 192.168.1.0/24)

- All sites already had internet
  (from various providers, with various cheapest-brand routers)

- We could have forced unified addressing
- We could have forced router change on every site

- But lets be realistic…

Network architecture talk – our solution



- In our case, the simplest, cheapest and fastest solution was to add a MikroTik to every site.
- This meant we didn't have to touch existing routers that provided internet access.
- Minimal change to existing networking on the sites.

Network architecture talk – our problems 1

- What about routing?

  (how do we get clients to send the traffic we want over our new router)

- Luckily all our client PCs are in AD, meaning a single GPO rule takes care of this all.

Network architecture talk – our problems 2

- What about subnet duplicity?

- Netmap is the answer!

  netmap - creates a static 1:1 mapping of one set of IP addresses to another one.

- We created a unique “virtual” subnet for each site

  Then we used that subnet with NAT netmap to assure addressing uniqueness

- Not the cleanest nor best solution, but a valid one for us

  (we don’t like it, but sometimes even us Network Architects have to make compromises)

A tangent – where is IPv6 when you need it?

- IPv6 would have made this project MUCH easier

- Sadly, our old-ish, existing networks are far from IPv6 ready
- The various random internet providers at the sites are far from IPv6 ready
- There wasn't enough money in the project to allow full IPv6 deployment
- Excuses and more excuses…

- IPv6 – there is no avoiding it, we all need to just DO IT!

MikroTik to the rescue

Because it’s the right choice

In MirkoTik we trust

- MikroTik is IDEAL for a project like this

- Awesome feature-set

  (all we needed for this project and MUCH more)

- Best price/performance ratio you will find

  (imagine the price difference between deploying 1500 [insert big vendor name] routers and 1500 MikroTiks)

- Great for automation

Lets talk about what we want to achieve here

Setting our goals to automate

The traditional approach – the deploy

- Network admins configure the router (somehow, manually)

  (5 min. x 1500 routers = 125 hours = ~16 work days)
- Techs go out and install the hardware

- There is problem x at y, because z (wait time)

  (something site-specific, or pre-deployed config on router wrong)

- Can this new box also do x (wait time)

  (for us, lot of existing sites didn't have WiFi, ideal addition when deploying new MikroTiks at each site anyway)

The traditional approach – the maintenance

- Upgrade RouterOS and RouterBOOT on 1500 routers

  (2 min. x 1500 routers = 50 hours = ~6,5 work days per upgrade)

- Deploy a configuration change to 1500 routers

  (1 min. x 1500 routers = 25 hours = ~3 work days per config change)

- Something specific at site x (we want WiFi)

  Support center calls, tickets, manual admin work, potentially different config at each site… nightmare

The automated approach – the deploy

- Automation system is put into place
- Data is fed to CMDB (run a whateverSQL script)

- Techs go out and install the hardware + click a button to provision the router

- Self-service portal for changing various things

  (we want WiFi SSID x and PSK y)

The automated approach – the maintenance

- Upgrade RouterOS and RouterBOOT on 1500 routers
  (one click, 2 minutes)

- Deploy a configuration change to 1500 routers
  (one click, 30 seconds)

- Something specific at site x (we want WiFi)
  (self service portal for users, they can do it themselves and its immediately automatically provisioned to their routers)

Welcome to the dream-land

- Human error is eliminated
- Client/customer satisfaction improved

- MUCH cheaper in the long run
- Much healthier network
- Mass changes/upgrades are pain-less

- Data driven approach
  - Analytics can be run against CMDB

I am a (W)ISP, how does this apply to me?

- Even as a (W)ISP, you will inevitably have cookie-cutter configs
  (configs that are the same, just slightly different)

- Customer CPEs and/or customer routers
- BRASes / PPPoE ACs
- APs and Wireless bridges
- Switches, tons of switches…
- Etc.

- You should consider automating configuration provisioning and management of all of these

When should you automate?

And when should you not?

When should I automate?

- Consider these 3 metrics:
- Metric 1
  - time to implement automation vs. time to do the task manually
- Metric 2
  - cost to implement automation vs. cost to do the tasks manually
- Metric 3
  - benefits of automation

Metric 1

- Time to implement automation vs. time to do the task manually

- Do not forget to account for
  - Initial deployment of things
  - Maintenance of things

- With any sizable network (300+ devices), automation will become favorable really fast in this metric

Metric 2

- Cost to implement automation vs. cost to do the tasks manually

- This can be hard to calculate, it depends on many factors
  - Cost of developers vs. cost of network admins
  - Availability/price of the right developers
  - Complexity of the automation system
  - Etc.

- With large-scale uniform networks, this metric is also very favorable
- In smaller scale, or complex/diverse networks, it becomes less favorable

Metric 3

- Other benefits of automation

- Elimination of human error
- Unification of configuration across your network
- One-click re-provisionings, config changes, software updates on mass-scale
- Improved network and service quality
- You can offer better SLAs
- Etc., etc.

- Here, network automation wins BIG TIME

What’s the conclusion?

- I am personally a big pro-automation enthusiast
- I honestly think (based on facts and many previous projects and experience in the field) automation is the right call
- All other fields in the IT industry are moving to automation
  (DevOps, virtualization, containerization, cloud platforms, etc.)
- But consider the previous metrics, and decide for yourself if it’s the right call for your network

Automation basics

Starting from the beginning

The anatomy of a network automation system

- Component 1
  - CMDB - configuration management database
  - Self service portal that allows users to change selected things in CMDB

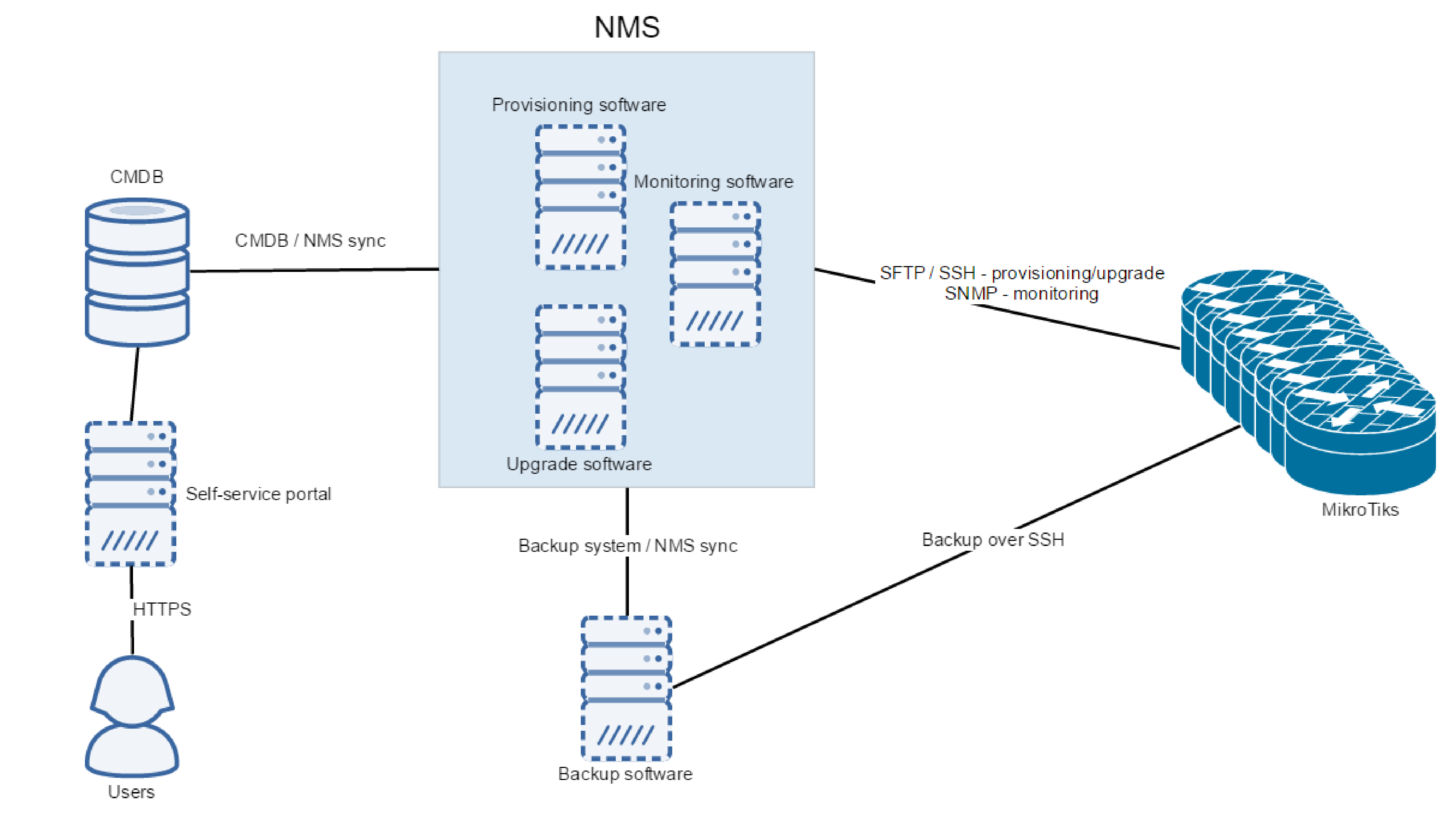
- Component 2
  - Provisioning system
  - Monitoring system
  - Upgrade system
  - Sync between CMDB and NMS

NMS – Network monitoring/management solution

- Component 3
  - Backup system
  - Sync backup system and CMDB/NMS

We are not talking about a backup system for your servers (which you should also have).
We are talking about a backup system for the configuration of your routers.

Anatomy diagram



Anatomy - component 1

- CMDB

  Keeps data about all our routers

  Unique things for each router (IP, gateway IP, identity, WiFi SSID/PSK, etc.)

- Self-service portal

  Allows users to change certain things (WiFi SSID/PSK)

  Shows health of their system

What did we use?

- We used the existing inventory system as the CMDB

  It allowed for custom fields that we used for router-unique data

- We wrote a simple web GUI as the self-service portal

  Takes < 2 days for an experienced coder

  Just remember, security, security, security…

Anatomy - component 2

- NMS – this is the heart of our system

- Makes sure all routers are on our validated RouterOS version
  - handles upgrade (if needed) during provisioning
  - handles mass-upgrades later on
- Takes care of provisioning config to new routers
  - generates router-unique config from a template and per-router data in CMDB
  - provisions the configuration to the router
- Full monitoring of our entire topology
  - next slide

Monitoring, monitoring, monitoring...

- Proper monitoring is absolutely essential

  Its so important that I have a separate MUM presentation just on this topic

  MUM 2015 CZ - https://youtu.be/McUCYuy9Cv0

- A difference between "I think everything is OK" and "I know everything is OK for a fact" is huge

- A proper monitoring system needs to be independent

  You should not have to look at it to see if there is a problem - the system should **tell you** when there is a problem

What did we use?

- We used NetXMS as the NMS
  - Open source NMS system, very flexible and extendable
  - See MUM presentation linked on previous slide for more

- We wrote our own provisioning and backup scripts
  - You can write TCL + Expect scripts, they are really easy
  - This was also used in our PoC system

- Final solution:
  - Java with JSch (SSH/SFTP) and Expect

On the MikroTik side

- As mentioned previously, RouterOS is really comfortable to automate around

- For upgrade, simply upload the appropriate package over SFTP connect over SSH and reboot

- For config provisioning, simply upload the per-router generated .rsc over SFTP, connect over SSH and execute:

  ```
  /system reset-configuration no-defaults=yes run-after-reset=config.rsc
  ```

CMDB / NMS sync

- When a new router is added to CMDB, it needs to be automatically added to the NMS

- We need to perform periodic sync between NMS and CMDB to merge changes (added router, decommissioned routers, etc.)
- We are talking about automation after all… nothing should be manual

What did we use?

- We wrote a little piece of software that connects to the CMDB and NetXMS over their APIs.

- Is simply adds/removes things from NetXMS as per CMDB

- This runs in cron once an hour, or can be manually run from within NetXMS

Anatomy - component 3

- Configuration backup system

- You should always have a configuration backup system, whether in an automation-based network, or a traditional network

- Even with full automation, config backups are necessary
  - Last line of defense, if you provisioning system breaks, you can always go back to valid previous configs
  - If someone or something corrupts data in CMDB or changes router-unique data in CMDB to invalid ones
  - Run analytics on actual deployed config history
  - Disaster recovery
  - Etc., etc.

What did we use?

- We used Unimus

  (self-promotion warning!)

- Your other options:
  - Rancid
  - Oxidized

- Write / script your own

Backup system / NMS sync

- When a new router is added to CMDB / NMS, again it needs to be automatically added to the Backup system

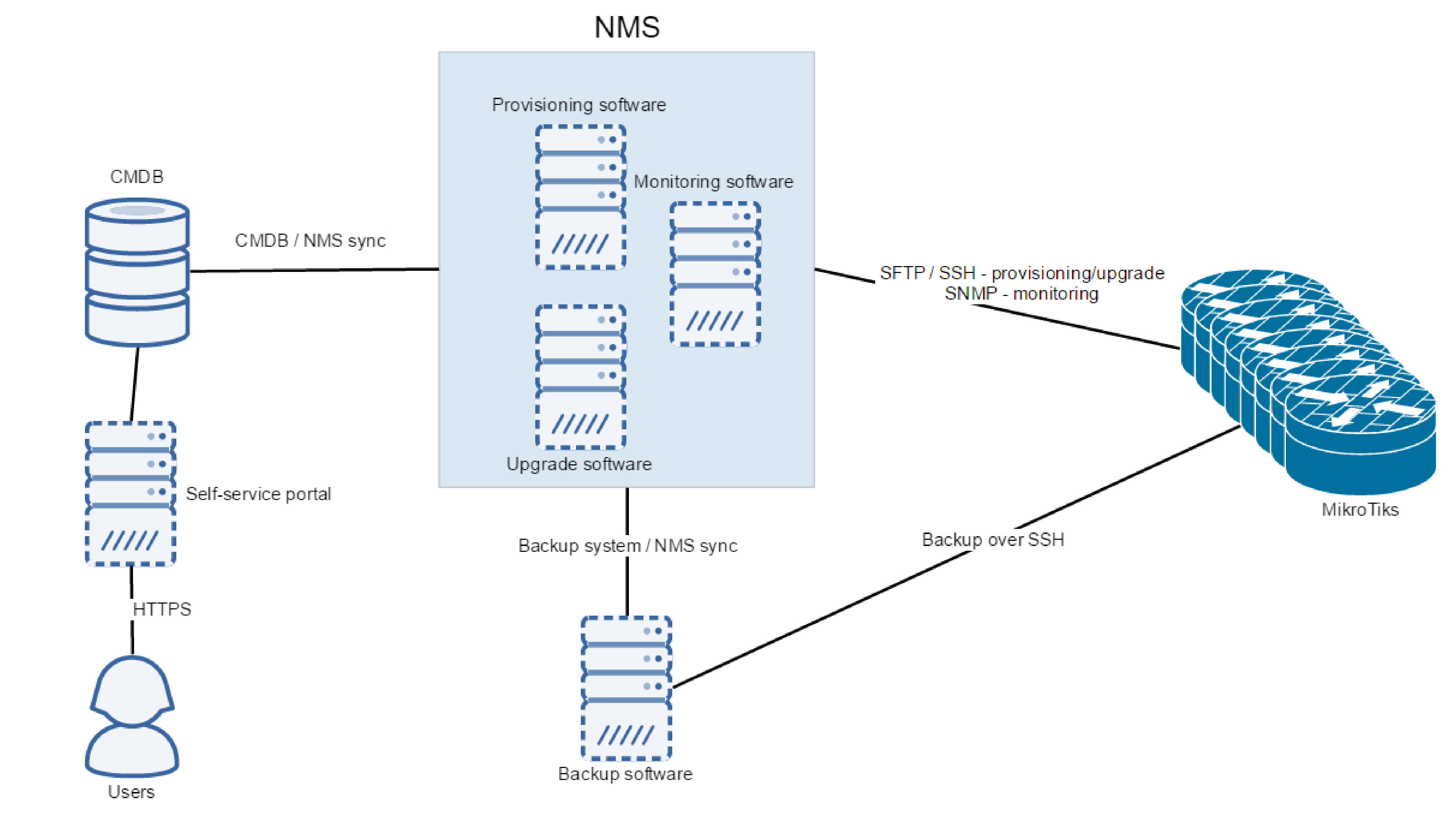
- Same point as previously with CMDB to NMS sync, everything should be automated

What did we use?

- Unimus has a native NetXMS sync connector, which can sync devices from NetXMS on a schedule and automatically start backing them up
  (more self-promotion warning!)

- If you use something different, you will have to implement this yourself

Anatomy diagram



The demo…

- This is the part of the presentation where I would usually do a demo to show you how it all works…
- While I would love to… it is not allowed by the an NDA (see the disclaimer at the start).
- I can however tell you how it works…

When adding a new router

- A new router is added to the CMDB with its appropriate router-unique information filled-out

  This is done by the data-entry team, not by the networking team

- An installer is dispatched to the field that installs a fresh-from-the-box MikroTik
- The installer logs into NetXMS, finds the router he just installed, and clicks “Provision router”

- Done

The automation effect

- This is much faster and simpler than a traditional, non-automated method

- This is much less error-prone than a traditional method
- This ensured network consistency
- All the other benefits we already discussed…

Customers/clients want to change something

- When the customers/clients was to change something (WiFi SSID or PSK for example)

- They log into the self-service portal
- They change the settings as they wish
- They click "apply changes"

- Data are updated in CMDB
- Their router is automatically provisioned with the newest config

The automation effect

- You can imagine how this is a much better experience for the customer/client than having to call a support line, create tickets, wait, etc.

- Customer satisfaction is increased, but network security / configuration integrity is not compromised

  Since we chose which things the users can influence, but they cant touch the config of the router in any real way

When updating RouterOS in entire topology

- When software updates are needed

- Network admin logs into NetXMS
- Selects a group of routers (or the entire network)
- Clicks “Upgrade routers”

- 2 minutes later, the entire topology of 1500 routers is running the new RouterOS

- All this of course assumes we tested the new version in our lab, it passed validation etc.

The automation effect

- Imagine how difficult this was without automation (actually, we calculated that about 20 slides ago)

- Total network consistency is achieved
- Attack surfaces are decreased, and much more defined (since we know all routers have same attack surfaces due to consistency)

When changing config in entire topology

- We need a configuration change

  We want to add new services to this network
  Or auditing shows our currently deployed configuration has a security/compliance issue
  Or our compliance requirements change, etc.

- We update our configuration template
- Log into NetXMS
- Select a group of routers (or the entire network)
- Clicks “Provision routers”

- 1 minutes later, the entire topology of 1500 routers is running the new config

- All this of course assumes we tested the new config in our lab, it passed validation, change management processes, etc.

The automation effect

- This would be a nightmare without automation.

- Configuration consistency is achieved
- Attack surfaces are decreased, and much more defined (since we know all routers have same attack surfaces due to consistency)

- Reaction time to change requests is massively decreased
- Etc., etc.

Additional resources

Things to watch/listen to

My other presentations and talks

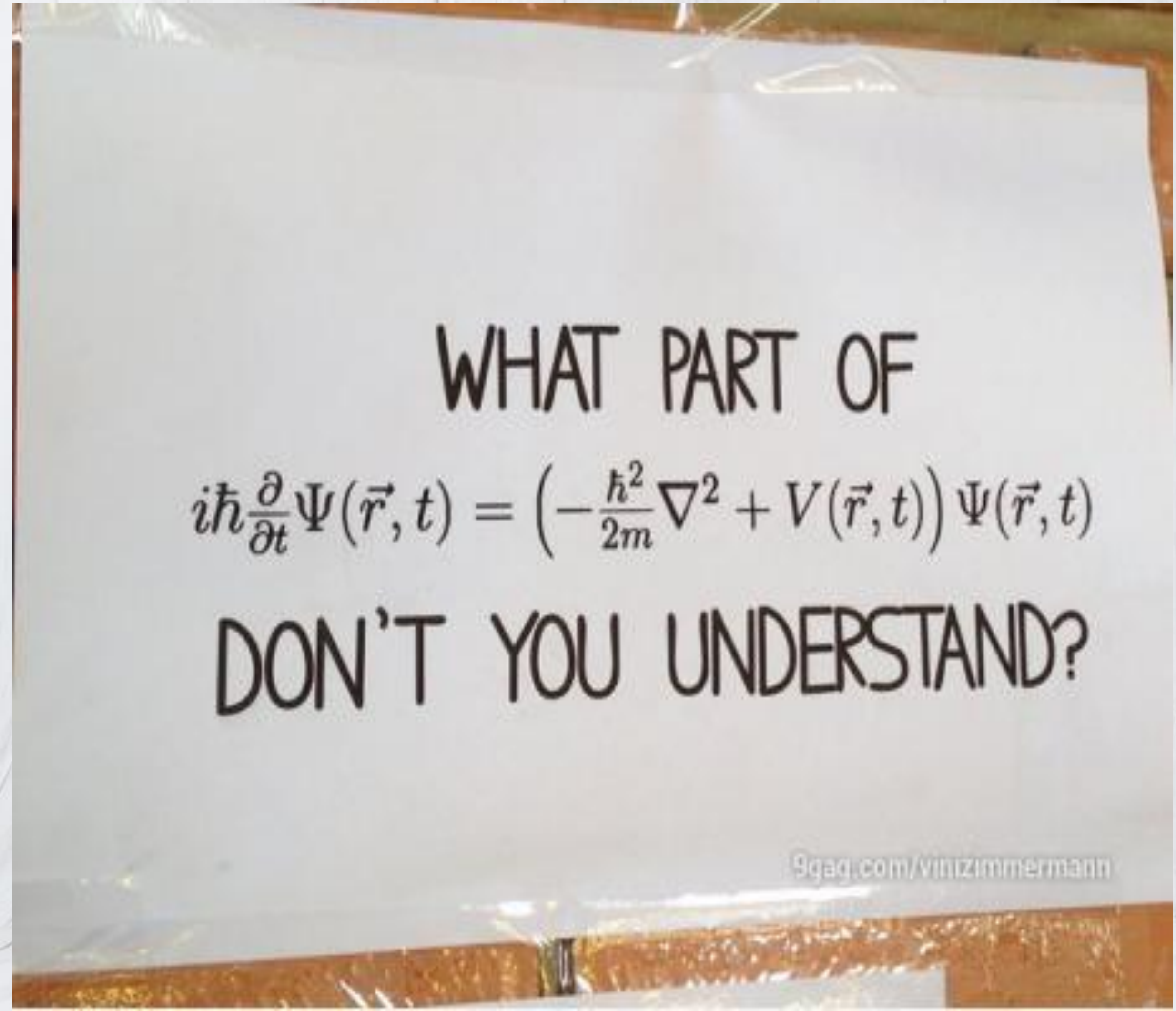
- Find all my other MUM presentations and more on YouTube:
  https://www.youtube.com/c/TomasKirnak/videos

  Load Balancing / Mangle deep dive
  L2TP / IPSec deep dive
  MLPS / VPLS / MTU deep dive
  Monitoring / SNMP deep dive

TheBrothersWISP

- I am a part of The Brothers WISP
- We do a bi-weekly networking podcast
  http://thebrotherswisp.com/
- Give us a listen if you feel like it!

Thank you very much for your attention!



WHAT PART OF

$$i\hbar\frac{\partial}{\partial t}\Psi(\vec{r},t) = \Big(-\frac{\hbar^2}{2m}\nabla^2 + V(\vec{r},t)\Big)\Psi(\vec{r},t)$$

DON'T YOU UNDERSTAND?

Tomas Kirnak
tomas@unimus.net