

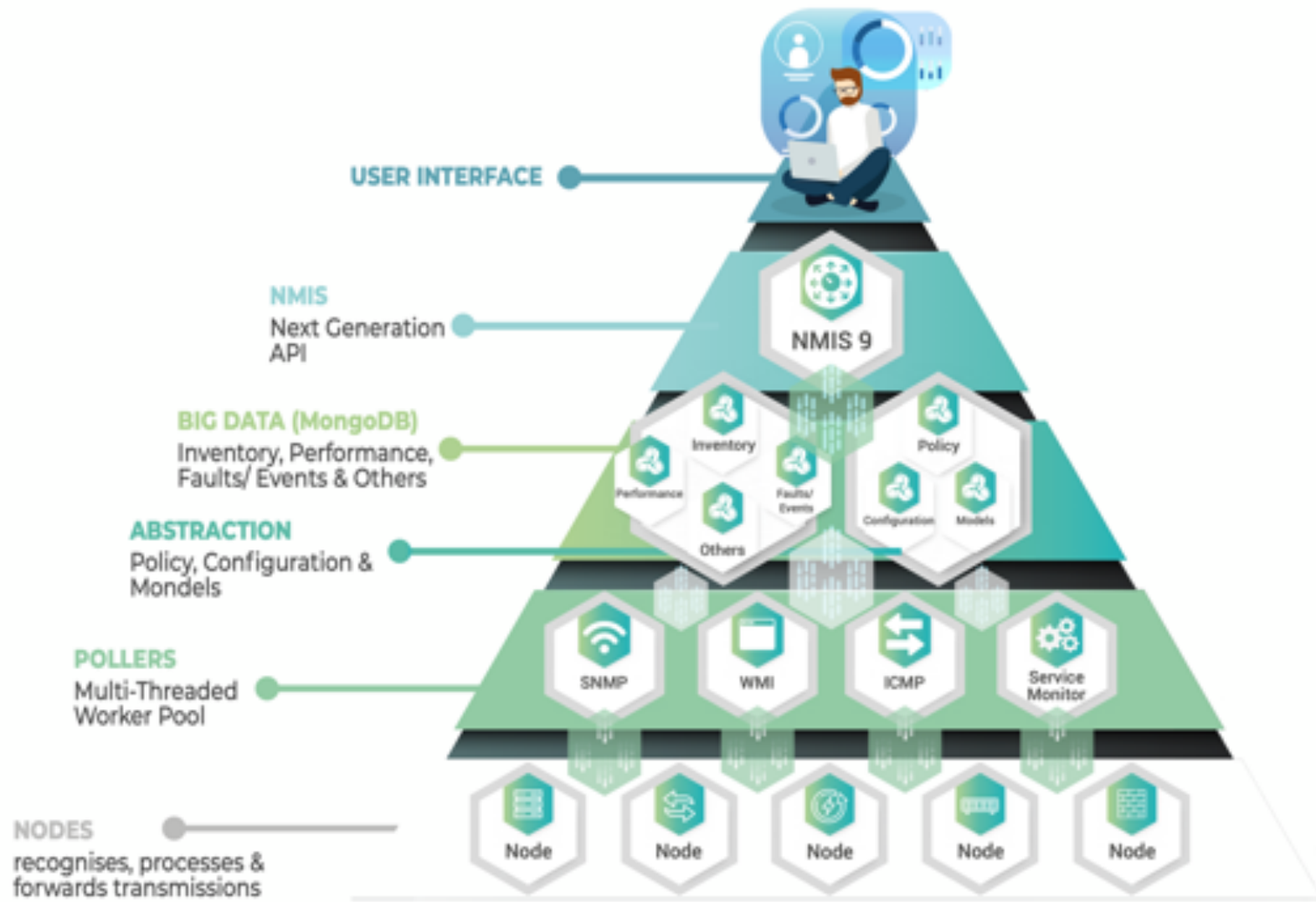


Automated Network Management Software



# New Release 24<sup>th</sup> June 2020

Taking your team to the next level of monitoring & automation



# Introducing NMIS9

Keep the best bits, make it better

## Sophisticated business rules engine

- The NMIS business rules engine classifies events on their business impact, not their technical nature.

## Pre-configured out-of-the-box solution.

- NMIS supports over 1000's vendors out of the box. The modelling capability allows the Network Engineer to immediately model new devices.

## Visible operational impact

- Network Operations can see how device performance is impacting the health of a single device, a group of devices or the whole network.

## Automated health live baselining

- NMIS measures a baseline of availability, response time and performance, and automatically shows the changes when compared to the previous period baseline.

## Massively scalable

- From the largest distributed global environments down to a single office implementation, NMIS handles the data, rules, and presentation.

## Customizable alert escalation procedures

- NMIS allows for customised alert escalation to suit your business. Escalate events based on your organisational structure, operational hours or chain of command.

# Why NMIS9?

# NMIS Timeline

**1999**

Keith's customers are asking how to monitor the new networks he has built. Commercial tools expensive and complicated. Existing open-source works but multiple pieces needed, NMIS 0 is born and released.

**2000**

NMIS 0.9 released. People finding and downloading NMIS, improvements made by community. NMIS added to RRDTOOL page and Yahoo Group launched.

**2001**

NMIS 1.0 and 2.0 released. SourceForge page created. NMIS downloads increase, community grows.

**2002**

NMIS 3 released. Eric Greenwood takes over as project lead. 20+ active developers / contributors.

**2005**

NMIS 4.0 is released. NMIS widely used, additional vendor support added.

**2007**

NMIS 4.x is stable and feature complete. NMIS 5 planning starts.

**2008-2010**

NMIS 5 in development.

**2011**

Opmantek formed with investment to formally commercialize NMIS.

NMIS 8.0 is released, based on NMIS 5 developments. Later in the year NMIS 8.1, 8.2 and 8.3 are released.

**2012**

Development starts on Opmantek Commercial Modules and the first are born. opReports, opMaps, opHA, opSLA and opFlow launched.

**2020**

NMIS 8.7 released

**2020**

**NMIS 9**

**Released**

**24<sup>th</sup> June 2020**

**2013-2019**

2013 - NMIS 8.4, opCharts and opConfig released  
2014 - NMIS 8.5, opEvents released  
2016 - NMIS 8.6, opTrend released  
2018 - NMIS 8.6.8G released



Enhancement of the NMIS engine, keep the best bits, improve performance



MongoDB replaces the NMIS8 file system database



Master server has more data available (**requires opHA**)

What's  
Changed in  
NMIS9 and  
Why it  
Matters

Enhancement  
of the NMIS  
engine, keep  
the best bits,  
improve  
performance



System remains configuration and model based, keeping the flexibility and agility of the engine.



New scheduling system (polling daemon), ensuring that resources are better utilised.



Improved out of the box functionality and easier to customise



Better long-term maintainability (no more configuration file or default model copying necessary)



More precise logging better reporting of statuses

MongoDB  
replaces the  
NMIS8 file  
system  
database



Big data approach for all node, event and status data.



Holistic view of data, data schema now reflects all the data, easier to look for data across many nodes. **(requires opCharts)**



Improved search, able to quickly search across all inventory data. **(requires opCharts)**



Reduced disk activity, more data in memory.

# Why mongoDB® ?

- MongoDB is a document database built on a scale-out architecture that has become popular with developers of all kinds who are building scalable applications using agile methodologies.
- MongoDB was built for people who are building internet and business applications who need to evolve quickly and scale elegantly. If you are doing that, you should consider MongoDB.
- Companies and development teams of all sizes use MongoDB because:
  - The document data model is a powerful way to store and retrieve data that allows developers to move fast.
  - MongoDB's horizontal, scale-out architecture can support huge volumes of both data and traffic.
  - MongoDB has a great user experience for developers who can install MongoDB and start writing code immediately.
  - MongoDB can be used everywhere by anyone:
    - *For free through the open source community edition*
    - *In the largest data centers through the enterprise edition*
    - *In any of the major public clouds through MongoDB Atlas*
  - MongoDB has developed a large and mature platform ecosystem, which means:
    - *MongoDB has a worldwide community of developers and consultants, so it is easy to get help.*
    - *MongoDB works on all types of computing platforms, both on-premise and in the cloud (both private, and public clouds such as [AWS](#), [Azure](#), and [Google Cloud](#))*
    - *MongoDB can be used from all major languages.*
    - *MongoDB can be accessed from all major ETL and [data management systems](#).*
    - *MongoDB has enterprise-grade support.*



Master server  
has more data  
available  
(requires  
opHA)



Better support for centralised data  
storage



Poller features now available on  
master like, TopN, Topology Maps,  
Interface and Inventory search



Most recent performance data  
available on the master



Seamless access to detailed  
performance data from the poller  
on demand

# What is the same, what has changed?

## What's the same?

- The GUI
- The features you love: thresholding, escalations, plugins, etc.
- Modelling is NMIS8 compatible
- Configuration is the same (Services, Escalations, Polling Policy, etc)
- RRD's are the same

## What has changed?

- Nodes.nmis is gone (node\_admin.pl still there)
- *nodename*-node.json and *nodename*-view.json are gone

# Important Concepts for NMIS9



NMIS9 JOB  
PROCESSING



CONFIGURATION AND  
MODELS

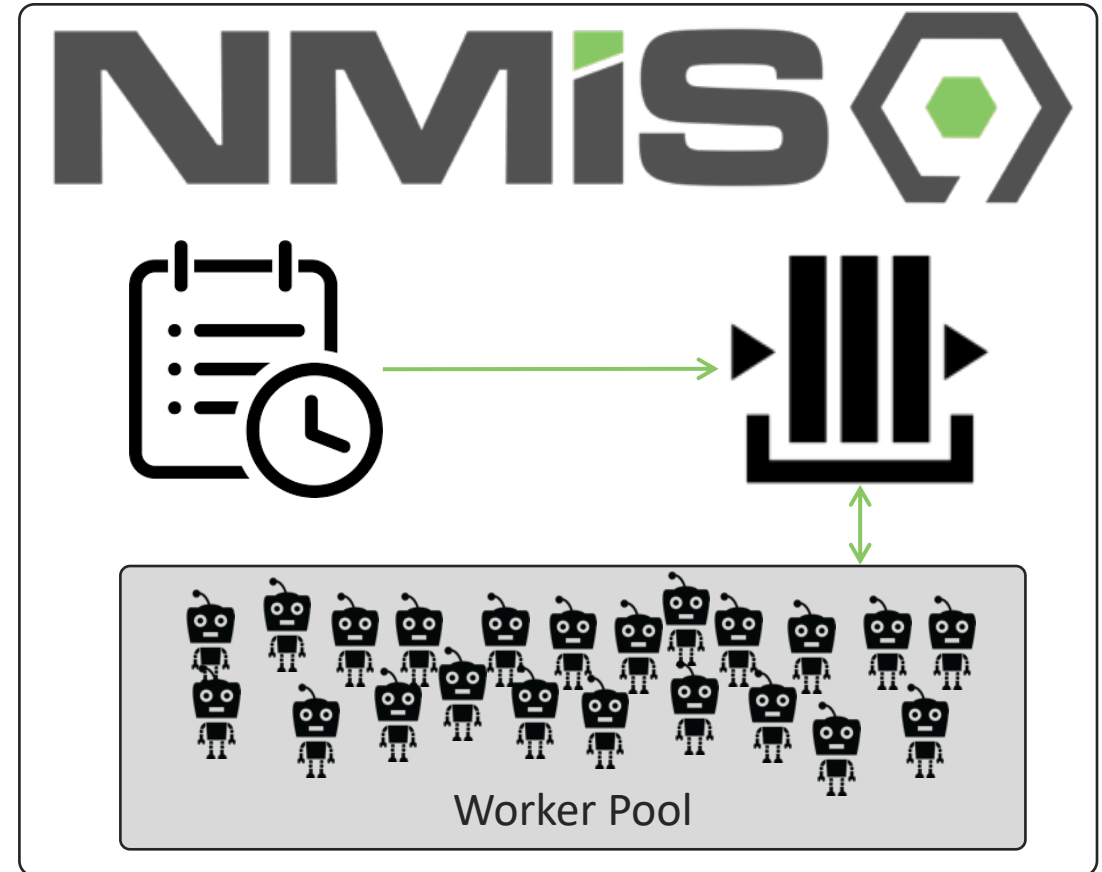


NMIS9 DATA  
STRUCTURES

# NMIS9 Job Processing

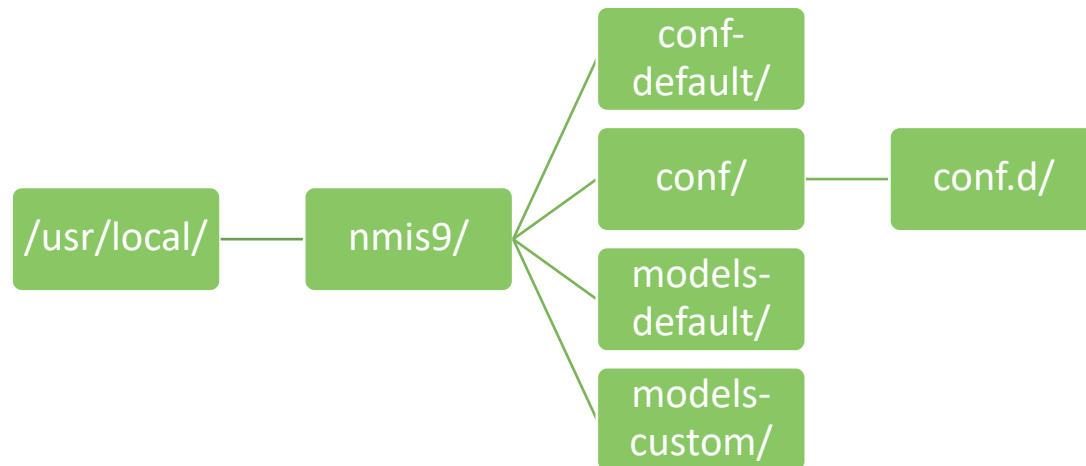
- The **NMIS9 Daemon** starts and spawns the minions which live in the **Worker Pool**.
  - How many workers NMIS9 has determines how much work can be done in parallel.
- The **NMIS9 Scheduler** runs and decides what jobs need to be done automatically, these are added to the **NMIS9 Queue**.
  - Items in the queue have a priority, you can change the default priorities.
- Using the **NMIS CLI**, you can query the queue, and add or delete things from the queue.
- **NMIS9 Workers** monitor the queue looking for work to do and then make it happen.
  - Collect, update, fping, summary, threshold, escalate etc all the regular jobs NMIS needs to do.

NMIS9 includes a non-blocking, scalable solution for making things happen on time, while maximizing resource utilization.



# NMIS9 Configuration and Models

**Simplifying configuration and model maintenance.**

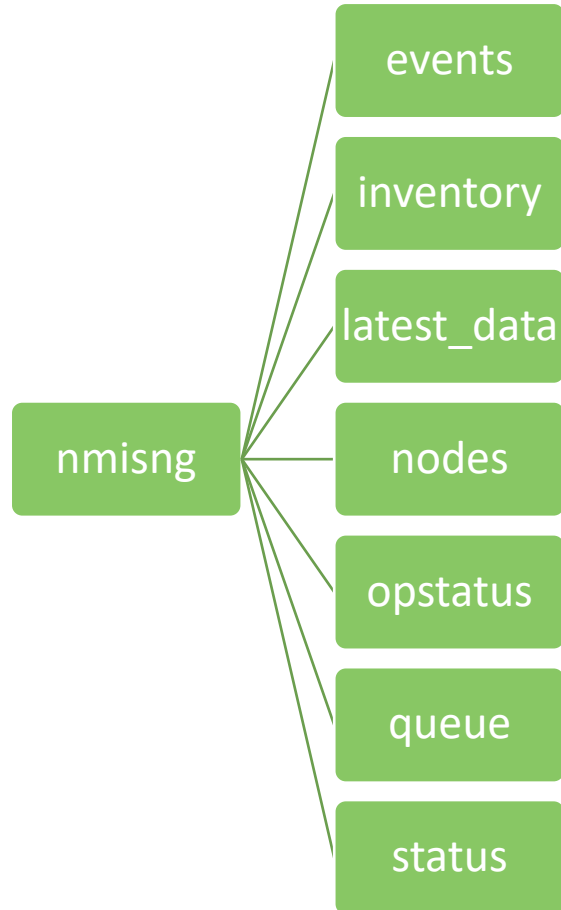


Directory	Description
conf-default	The default unchanged configuration files.
conf	Any changed configuration
conf/conf.d	Any configuration fragments
models-default	The default models maintained by NMIS9 community
models-custom	Any changed models, or new models

**Configuration Hierarchy** - Configuration is loaded from `conf-default`, then overridden by configuration files found in `conf` and `conf.d`.

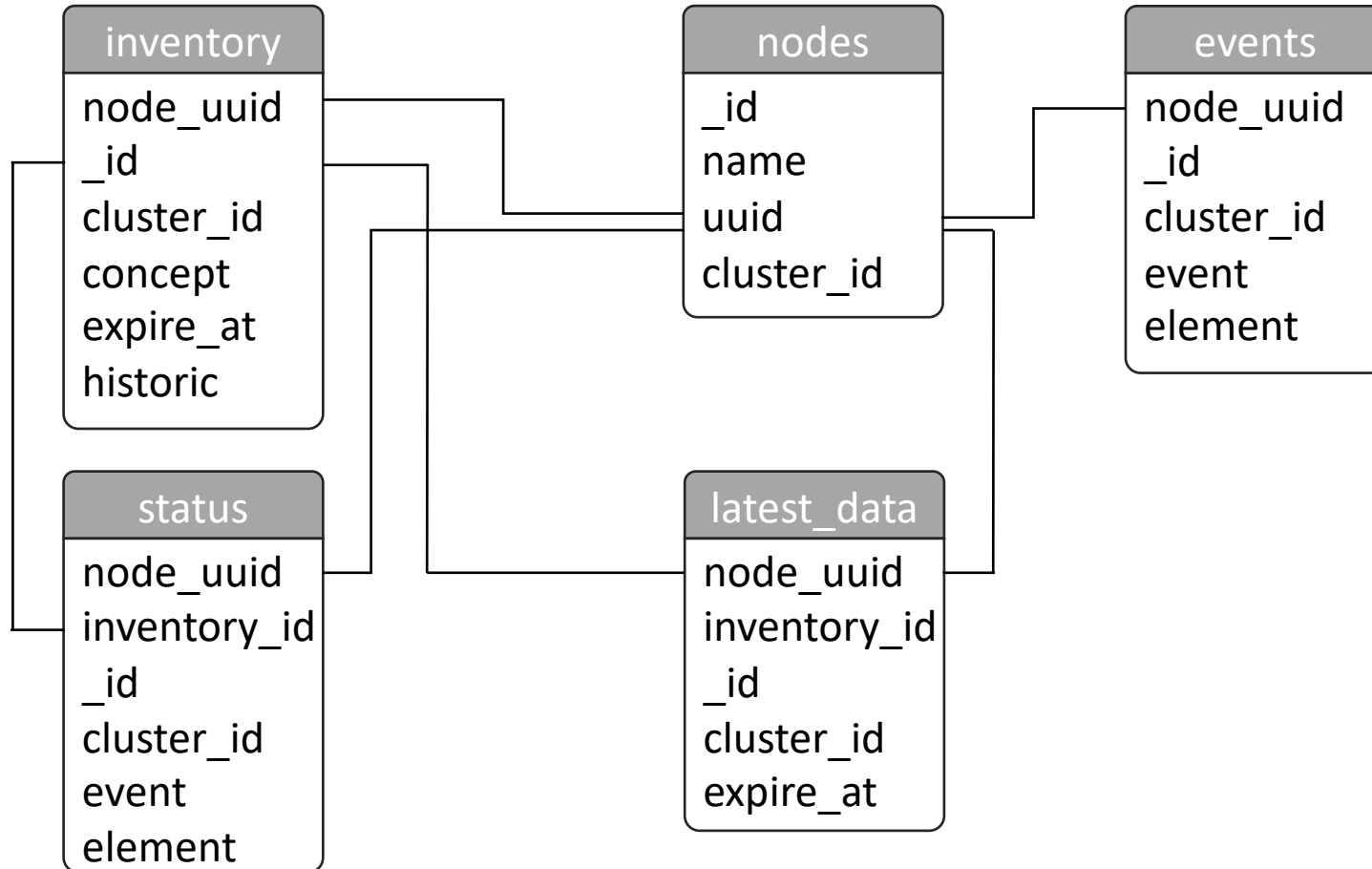
**Model Hierarchy** - Models are loaded from `models-default` first and if a file exists in `models-custom` it is loaded.

# NMIS9 Data Structures



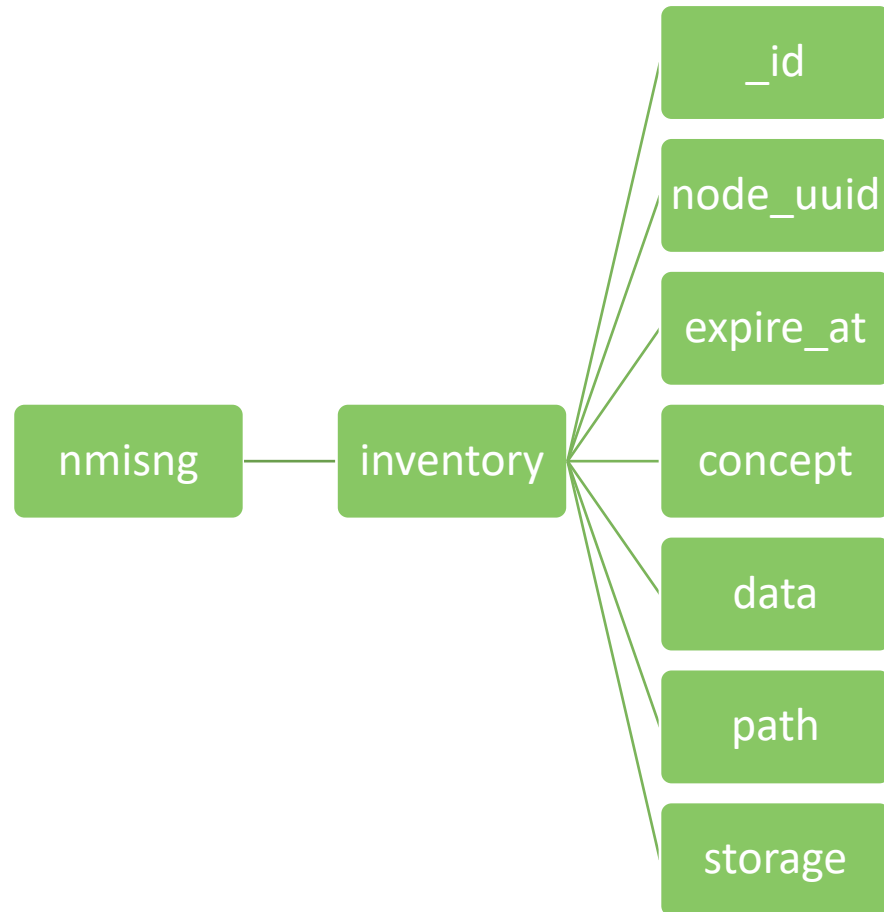
Collection	Description
events	Event database, current state of all things being managed.
inventory	'static' data NMIS collects from devices.
latest_data	'dynamic' data NMIS collects from devices.
nodes	Node configuration, Nodes.nmis and nodeConfig information.
opstatus	Operational status logs
queue	Work which is queued to be run
status	Proactive and alert conditions for each node

# NMIS9 Summary Entity Relationship Diagram



- Everything has a cluster\_id which is the id of the originating poller.
- Everything is related to a node using node\_uuid

# NMIS9 inventory



- The inventory collection in MongoDB holds all the inventory data, which is things which are static and change infrequently.
- Each inventory document has a concept, which is what type of data it is, e.g. interface, cdp, bgpPeer, etc
- There is a special inventory concept called catchall, which catches all the device specific inventory.





**OPMANTEK**  
NETWORK MANAGEMENT & IT AUDIT SOFTWARE

# Questions?