Hewlett Packard
Enterprise

Making Monasca Monitor More:

Extending Monasca's Data Gathering & Reporting Capabilities

Stefano Canepa & Domhnall Walsh

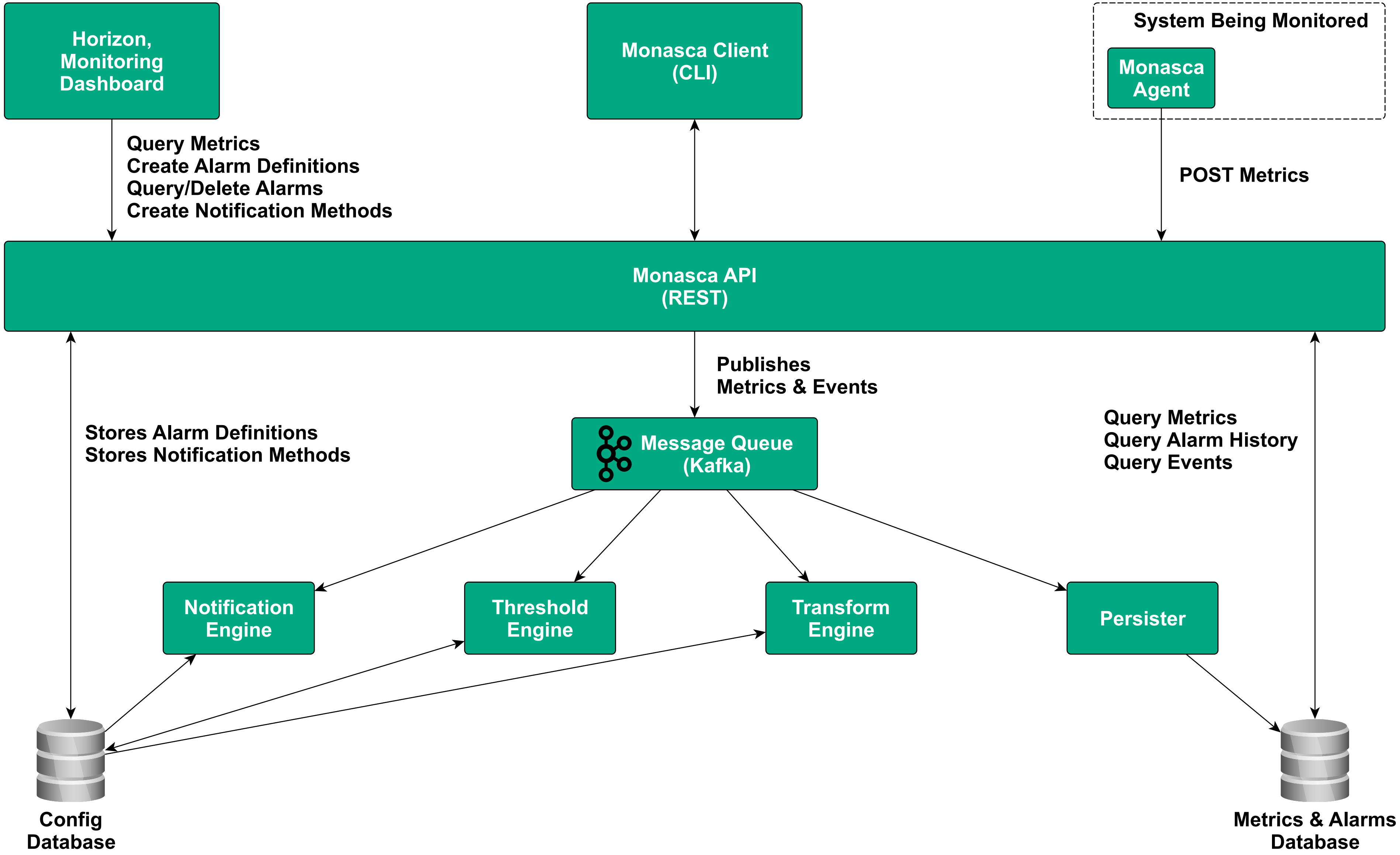
Who we are

- Stefano Canepa <stefano.canepa@hpe.com> aka <sc@linux.it> and sc on IRC
- Domhnall Walsh <domhnall.walsh@hpe.com>

What is Monasca?

- Monasca is a “…high-performance, scalable, fault-tolerant and extensible monitoring system based on a micro-services bus architecture”
- Read all about it at http://monasca.io/
- If you have time, please complete the Monasca team’s survey at https://goo.gl/1smB6i - it’d help a lot!

What is Monasca?



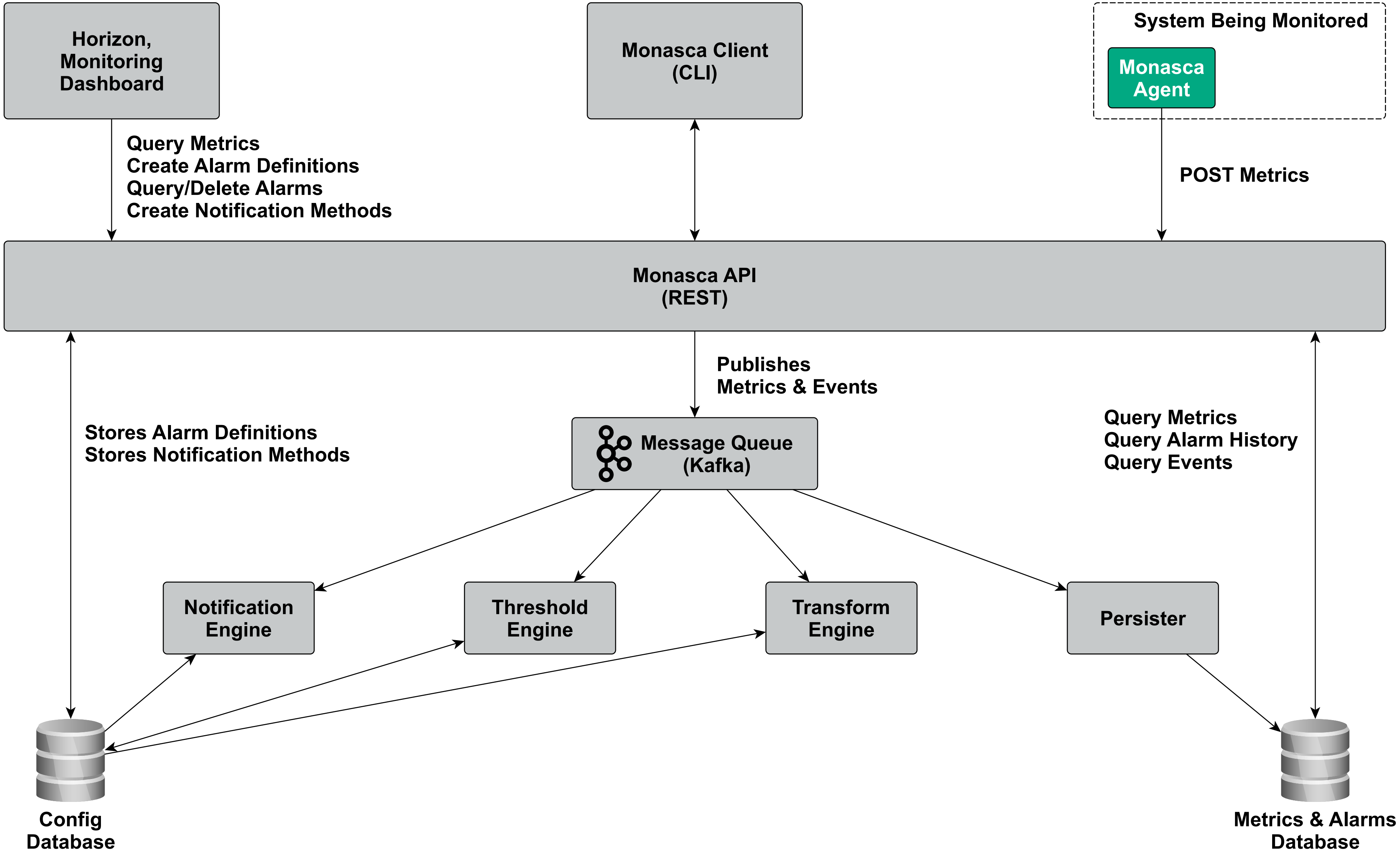
Monasca Terminology

- Metric – An attribute or property that we want to monitor
- Dimension – A property of a metric that helps define what it applies to, e.g. hostname, role, etc.
- Event – either:
  - An OpenStack event that Monasca consumes, or
  - What is raised when an alarm is triggered
- Measurement - an individual value for a metric – i.e. the state of that metric at a specific time
- Alarm definition
  - The rules that define when an alarm should be triggered
  - What should happen when it is triggered, i.e. what gets notified
- Alarm state
  - Several possible states – ALARM, OK, UNDETERMINED
- Notification Method - A mechanism that can inform something outside Monasca that an alarm triggered.

Customer Requirements

- Monitor storage clusters that were acting as Cinder back-ends
- Integrate Monasca alarms with their existing alerting system
  - Alerting system accepted input in the form of SNMP traps
- Generate reports about the status of their OpenStack cloud
- Integrate Monasca into their existing Nagios monitoring system

Monitoring Storage with Monasca



Hewlett Packard Enterprise

Monitoring Storage with Monasca

- **Tasks at hand:**
  - Determine what data needed to be collected
  - Find out how to access that data
  - Store that data in Monasca in the form of Metrics
- **Monasca uses an agent for monitoring**
  - We can extend that agent with **plugins**

Monasca Agent

- **Installed on every node that needs to monitor (or be monitored)**
- **Collects data from:**
  - `statsd` interface to various applications
  - Checks
- **Checks are just plugins**
  - Many included out of the box
  - Custom checks can be added

Monasca Agent Plugins

- **Types:**
  - **Detection:** detects, configures and activates check plugins
  - **Check**: perform checks on other applications, servers, or services
- **How they are run:**
  - **Detection**
    - When Monasca Agent starts up
    - When Monasca Agent is reconfigured (using `monasca-setup)`, or…
    - When explicitly invoked by `monasca-setup` (using `-d <plugin name>`)
  - **Check:** On a regular schedule, configurable for each plugin

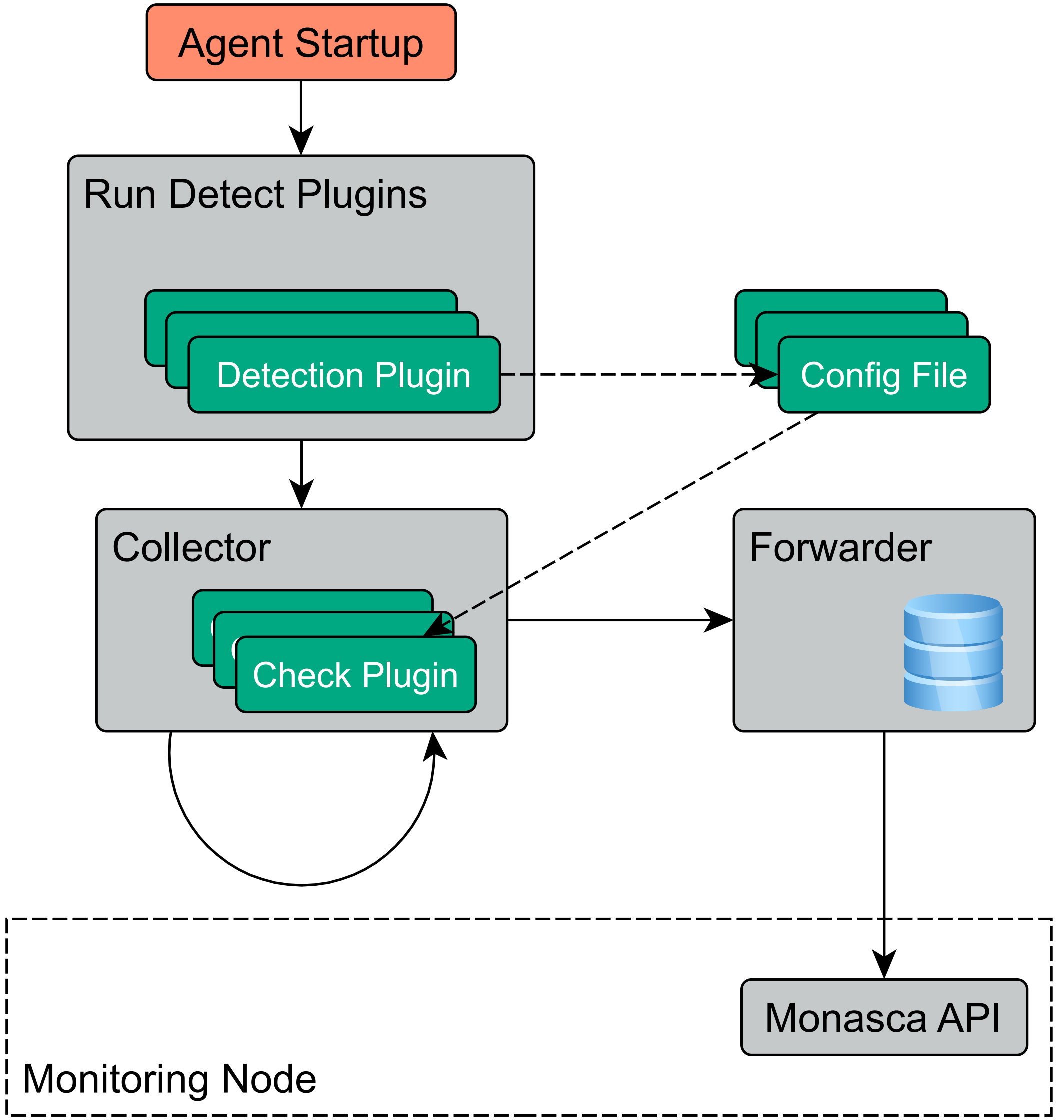
Our Solution

- **Our storage cluster (HPE VSA StoreVirtual) featured two APIs**
  - HTTP REST
  - Command-line via SSH with XML output
- **Each API provides only a subset of the required data**
  - REST API fast but lacking performance data
  - Command line interface (SAN/iQ) can produce performance data, but task is resource intensive
- **Solution: Use both = two check plugins, one per API**
  - Plugins can be run at different intervals for flexible configuration
- **One “master” source of config data to set up both checks:**
  - A list of clusters to monitor (“instances”)
  - Credentials for each one
- **Single detection plugin creates configuration files for both check plugins**

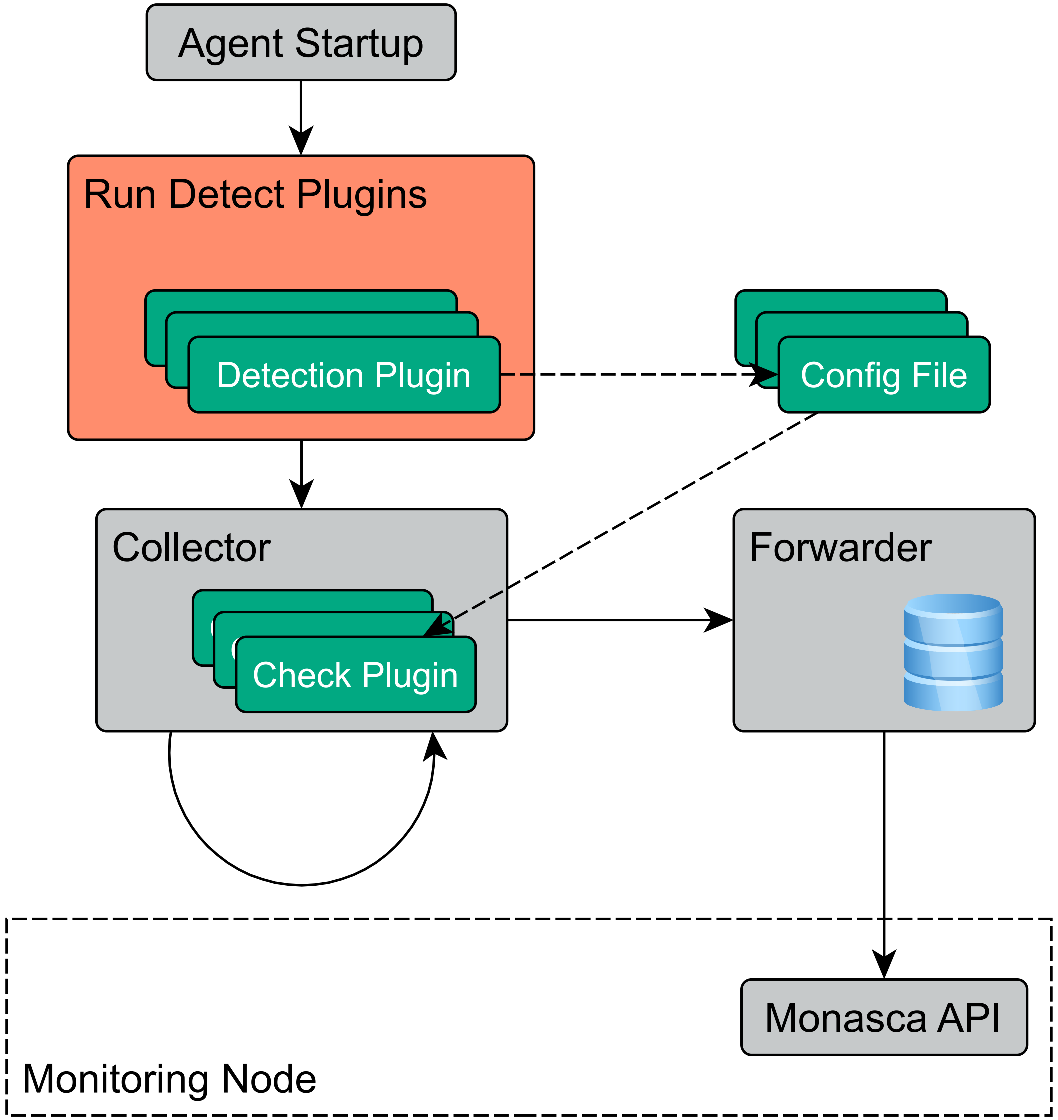
Our Solution

- **Normally, Monasca Agent monitors locally – the node it lives on**
  - In this case, we are checking a remote system, so have two points of failure
- **Run multiple instances for redundancy**
  - Checks are resource intensive, so elect one node to run checks at any given time
  - (Slight cheat...) Only run if ZooKeeper on same node is “leader” (not “follower”)
  - To Do – replace ZooKeeper dependency with own election process
- **Tune configuration to balance appliance load from monitoring against frequency of measurement**

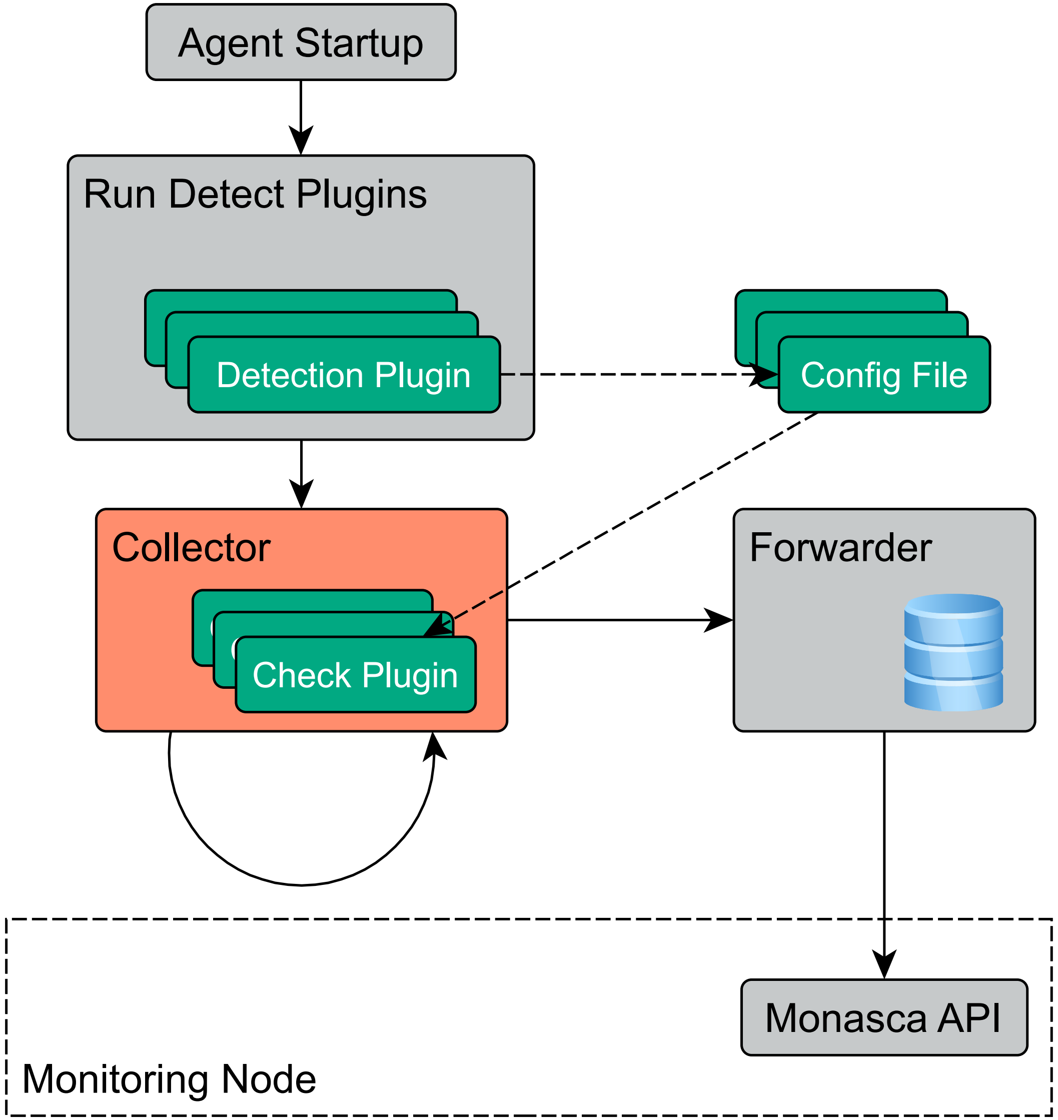
How Monasca Agent Plugins Run



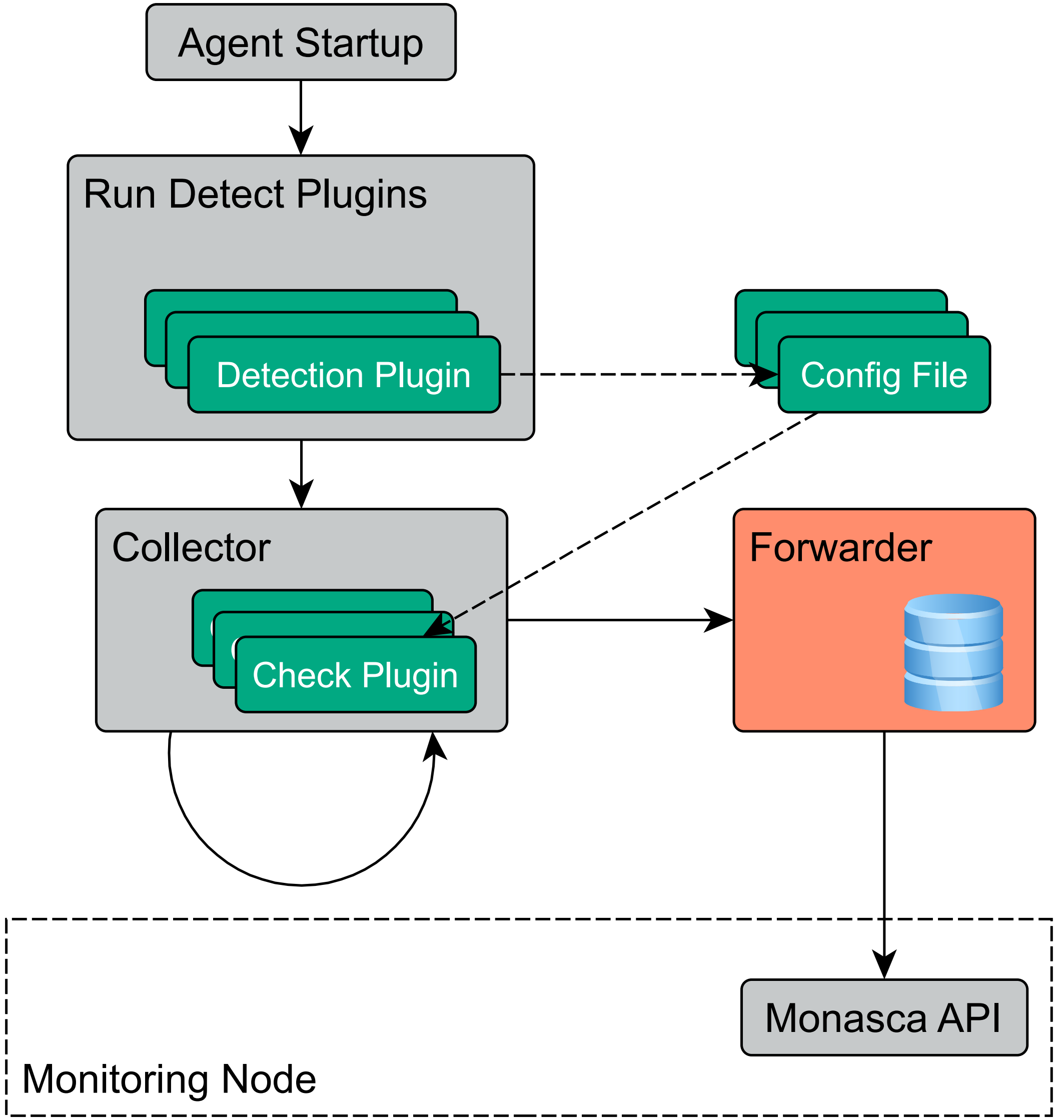
How Monasca Agent Plugins Run



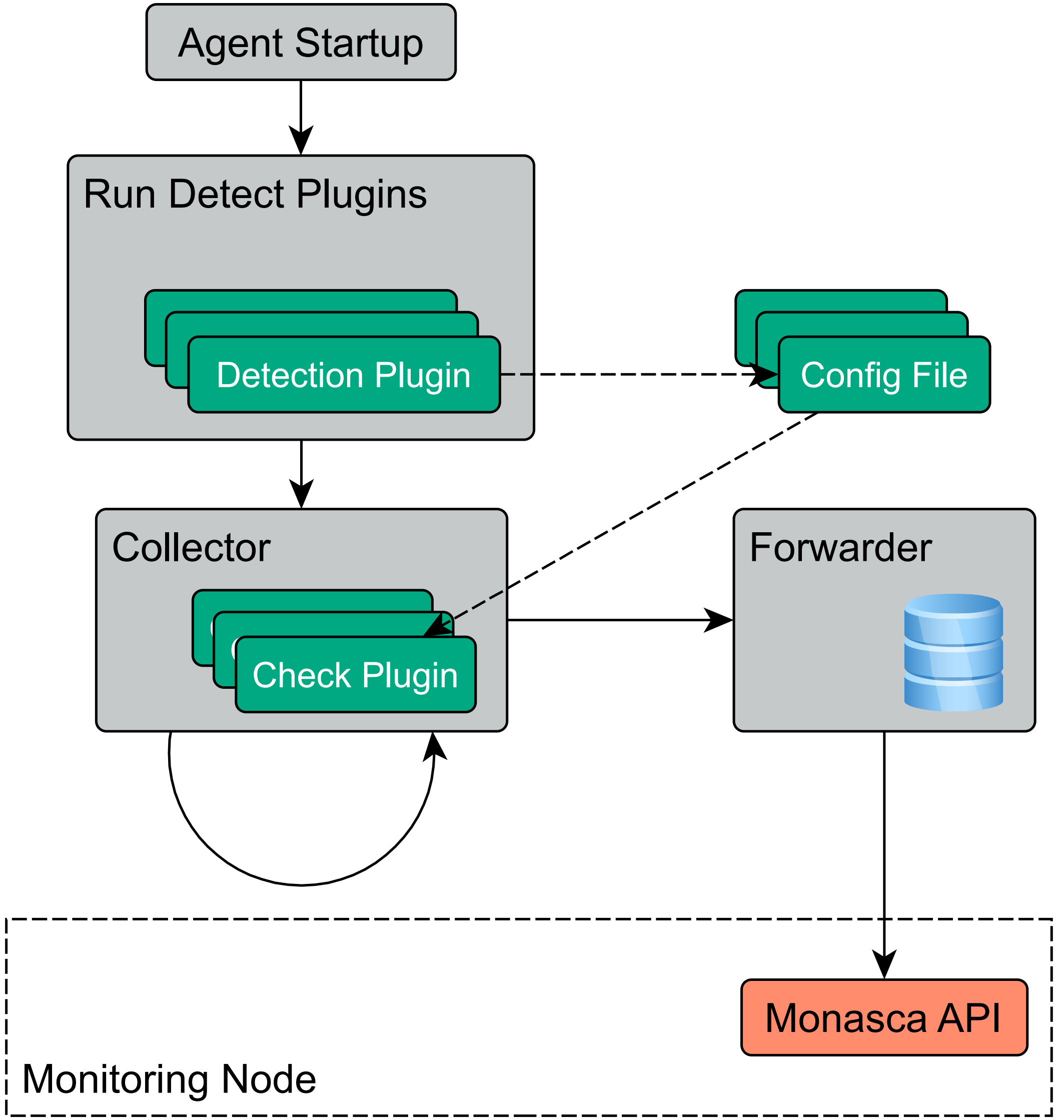
How Monasca Agent Plugins Run



How Monasca Agent Plugins Run



How Monasca Agent Plugins Run



Hewlett Packard Enterprise

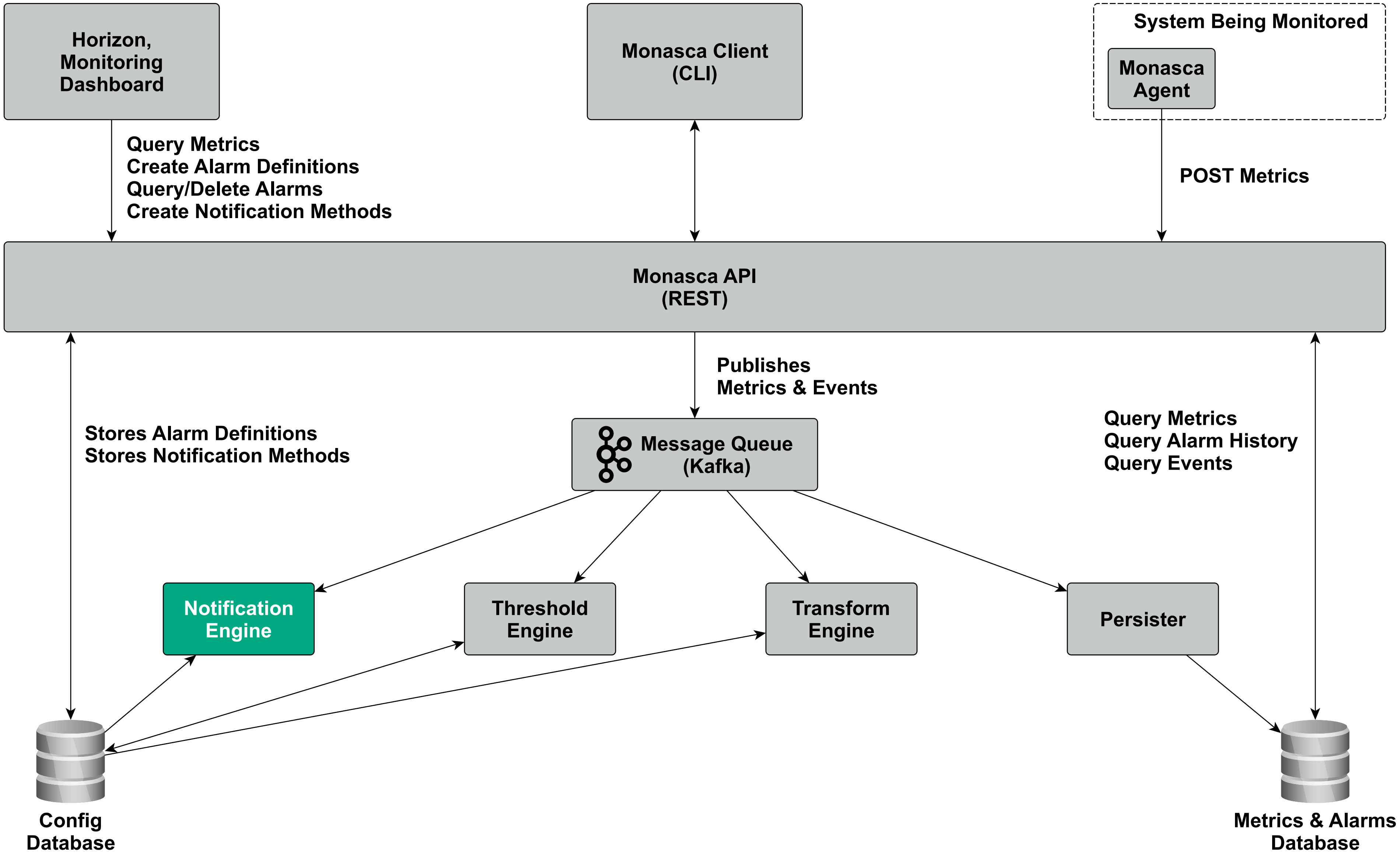
Monasca Agent File Locations

- **`/etc/monasca/agent` – Agent files**
  - `(…)/agent.conf` – Master configuration file for the agent.
  - `(…)/conf.d` – Config files for check plugins
    Each file is matched to a check plugin of the same name
- **`/usr/lib/monasca/agent/` - Plugin files**
  - `(…)/custom_checks.d` – Custom check plugins
  - `(…)/custom_detect.d` – Custom detection plugins

Plugin Config File Structure

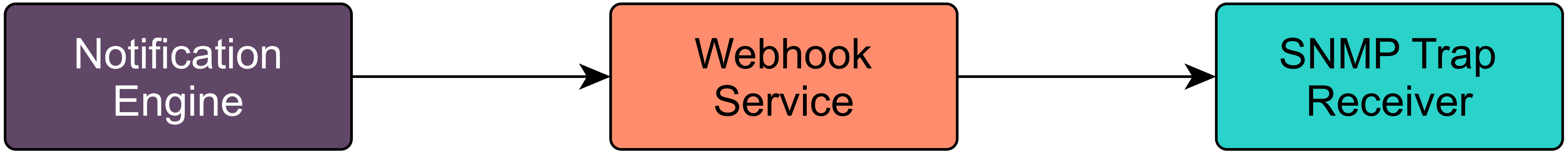
- **YAML, two main sections:**
  - `init-config` – used to inform Monasca Agent how to run the plugin. Important settings:
    - `check_frequency` – how often to run the check (sec)
    - `collect_period` – how often to send (buffered) data back to Monasca (sec)
  - `instances` – the items to check
    - Each must include all required data as key/value pairs, e.g. paths, login credentials, etc.

Monasca Notification Forwarding

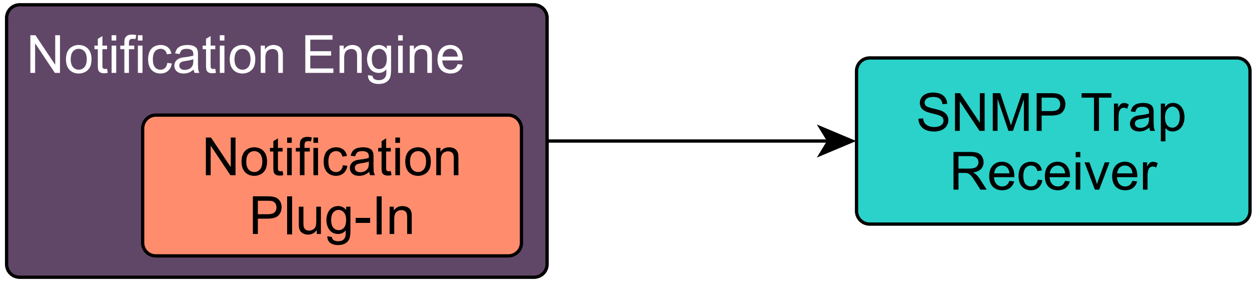


Monasca Notification Forwarding

– **The old way:**

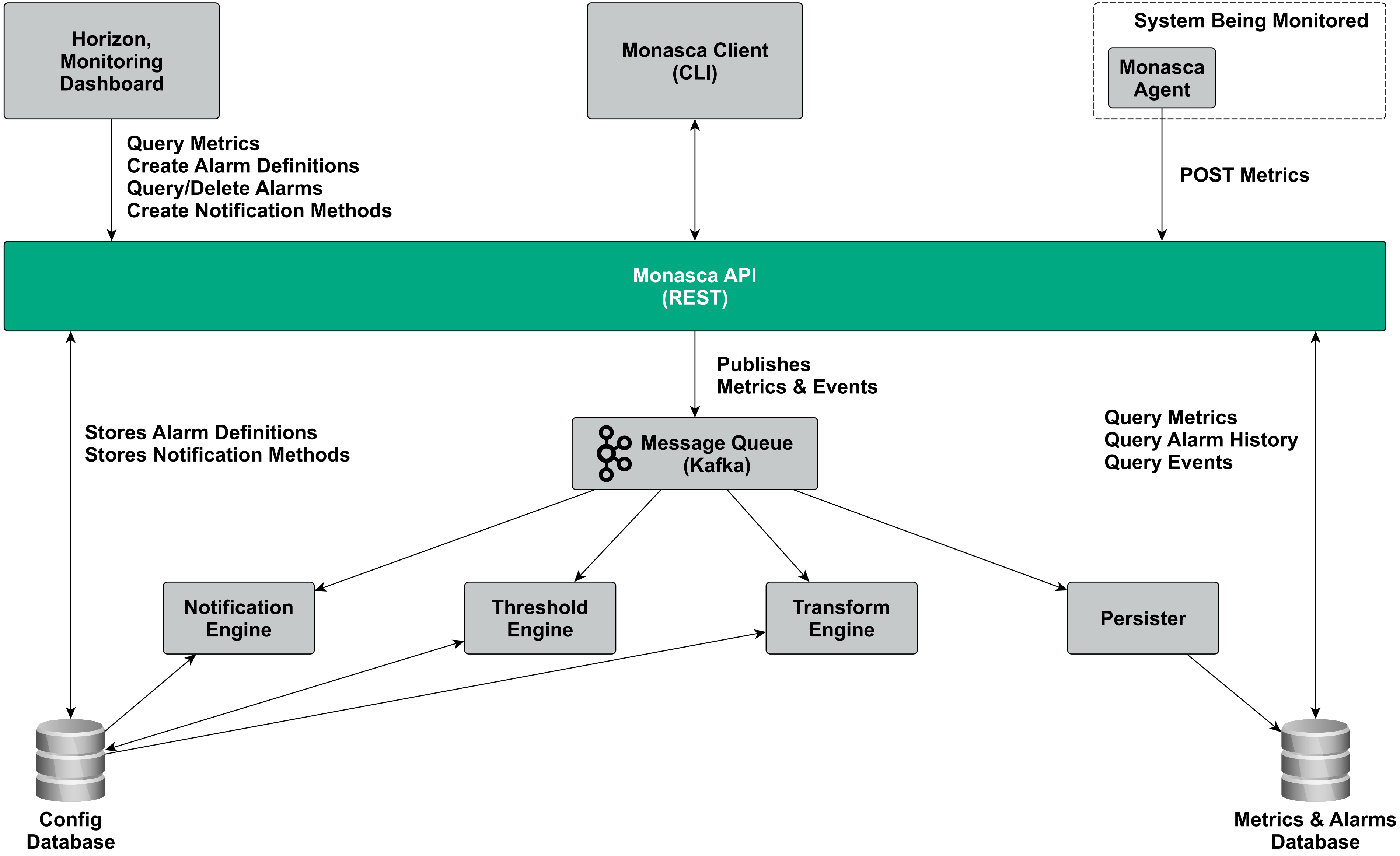


– **The new way:**



Hewlett Packard Enterprise

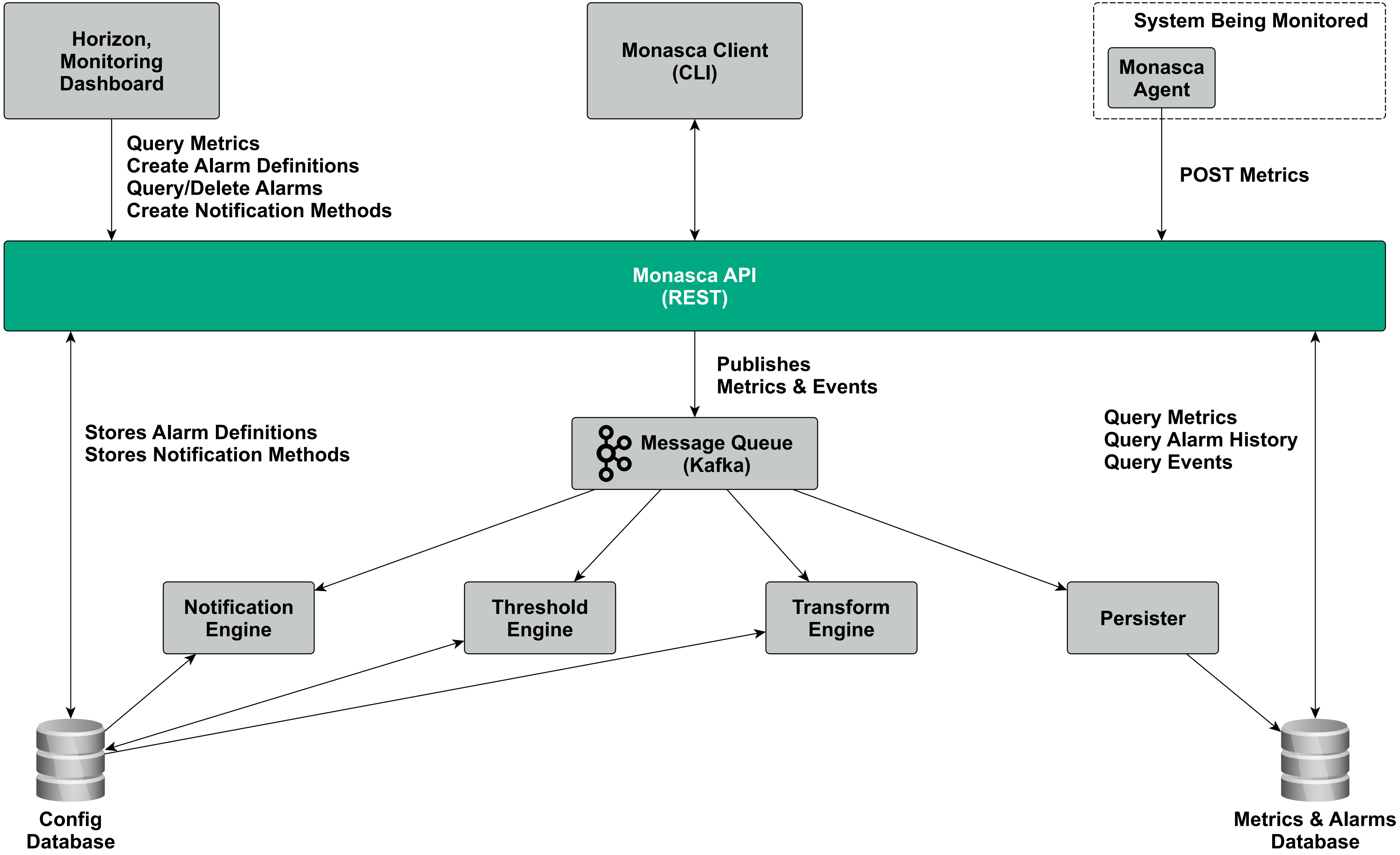
Reporting from Monasca



Reporting from Monasca

- **Get input from user:**
  - Start date
  - End date
  - Data to collect
- **Access measurements via API**
- **Collect data into temporary storage**
- **Render data as a PDF**

Integrating with Nagios



Integrating with Nagios

- **Nagios (and its clones) get data by running scripts that output data**
- **We have two ways to interact with Monasca API:**
  - Using the Python `monasca-client`
  - Accessing the REST API endpoints and interpreting the JSON results
- Monasca client package does all of the hard work for you
- Using the OpenStack REST APIs directly means you have to do all the work:
  - Managing Keystone tokens
  - Building API requests and parsing the results

Summary – Why is this Useful?

- **Agent Plugins**
  - Enable monitoring of items that Monasca can’t (yet)
- **Forwarder / Notification System**
  - If you need to hook Monasca (or just specific alarms) up to an external monitoring service
- **Reporting**
  - To generate reports <u>directly</u> from your monitoring system
- **Nagios Integration**
  - Export Monasca data to existing customer monitoring solution(s)

Hewlett Packard
Enterprise

Q & A