MONASCA

an OpenStack Community Project

Monasca: Project Onboarding

Witold Bedyk (Fujitsu)
witold.bedyk@est.fujitsu.com
IRC: witek

Johannes Graßler (SUSE)
johannes.grassler@suse.com
IRC: jgrassler

Preliminaries

Slides and Transcript

- Compiled slides (recommended)
  - http://btw23.de/johannes/talks/monasca-onboarding.tar.bz2
- odpdown sources
  - https://github.com/jgrassler/talks/tree/master/monasca-onboarding

This Session

- What it is:
  - Primer on Monasca
  - Overview of Monasca repositories and architecture
  - Introduction to the specifics of Monasca development
  - How can you contribute?
- What it is not
  - General introduction to OpenStack development
  - Refer to Code & Documentation Contributor Guide for that.

What is Monasca?

- Monitoring and Logging as a Service
  - Highly scalable
  - Fault tolerant
  - Highly performant
  - Multi-tenant

What is Monasca? (cont.)

- Features:
  - Metrics with dimensions (key/value pairs) as metadata
  - Real-time alerting
  - Pluggable notification engine
  - Flexible aggregation engine

Sources of Documentation

- https://docs.openstack.org/monasca-api
- https://wiki.openstack.org/wiki/Monasca
- http://monasca.io/

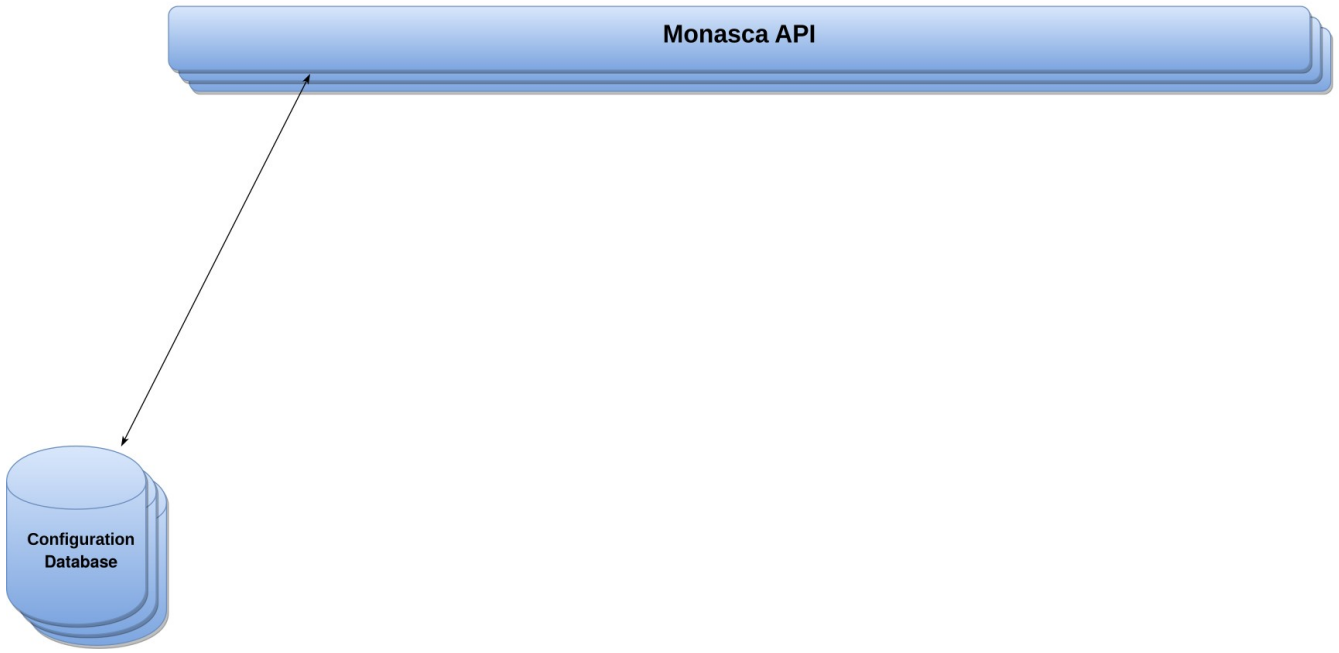
Main Contributors

- Fujitsu
- SUSE
- StackHPC
- Universidade Federal de Campina Grande

Architecture and Development

Metrics API (`monasca-api`)

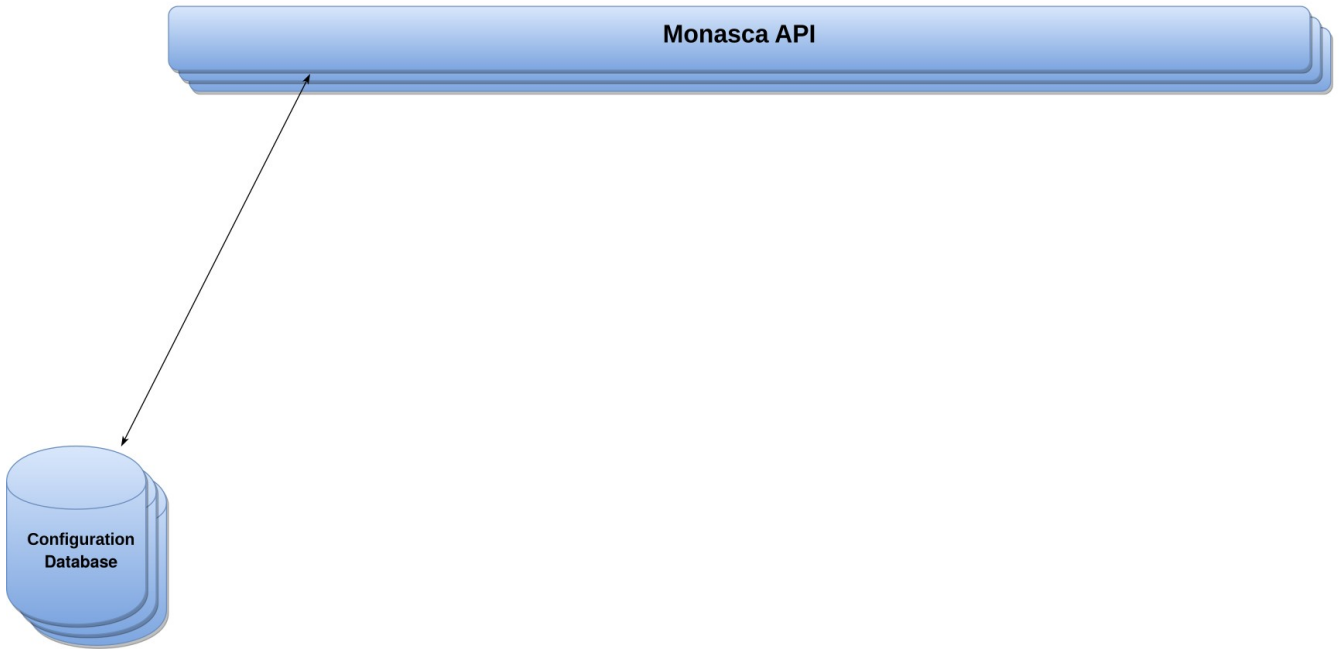
Metrics API (`monasca-api`)



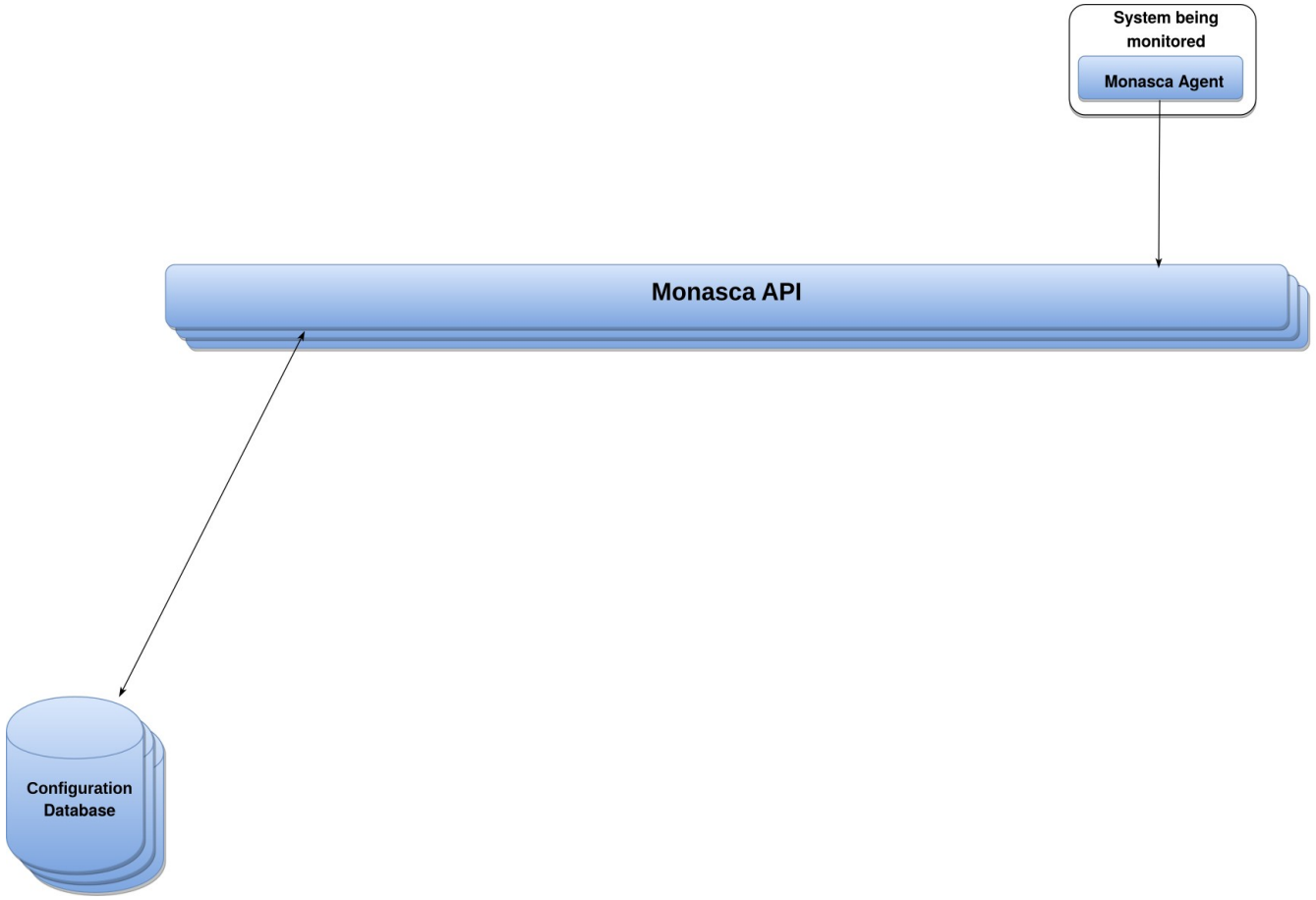
Metrics API (`monasca-api`)

- Repository
  - https://github.com/openstack/monasca-api
- Purpose
  - Receives metrics from agents
  - Makes metrics available for visualization/processing
  - Interface for modifying configuration database (alarms, notifications, ...)
- Development Information
  - Most important documentation repository for Monasca: source for https://docs.openstack.org/monasca-api
  - API reference: https://github.com/openstack/monasca-api/blob/master/docs/monasca-api-spec.md
  - Contains data model for configuration database (`monasca_api/common/repositories`)
  - Contains database migrations for configuration database (`monasca_api/db`)
  - Deprecated Java implementation: ignore when contributing

Metrics API (`monasca-api`)



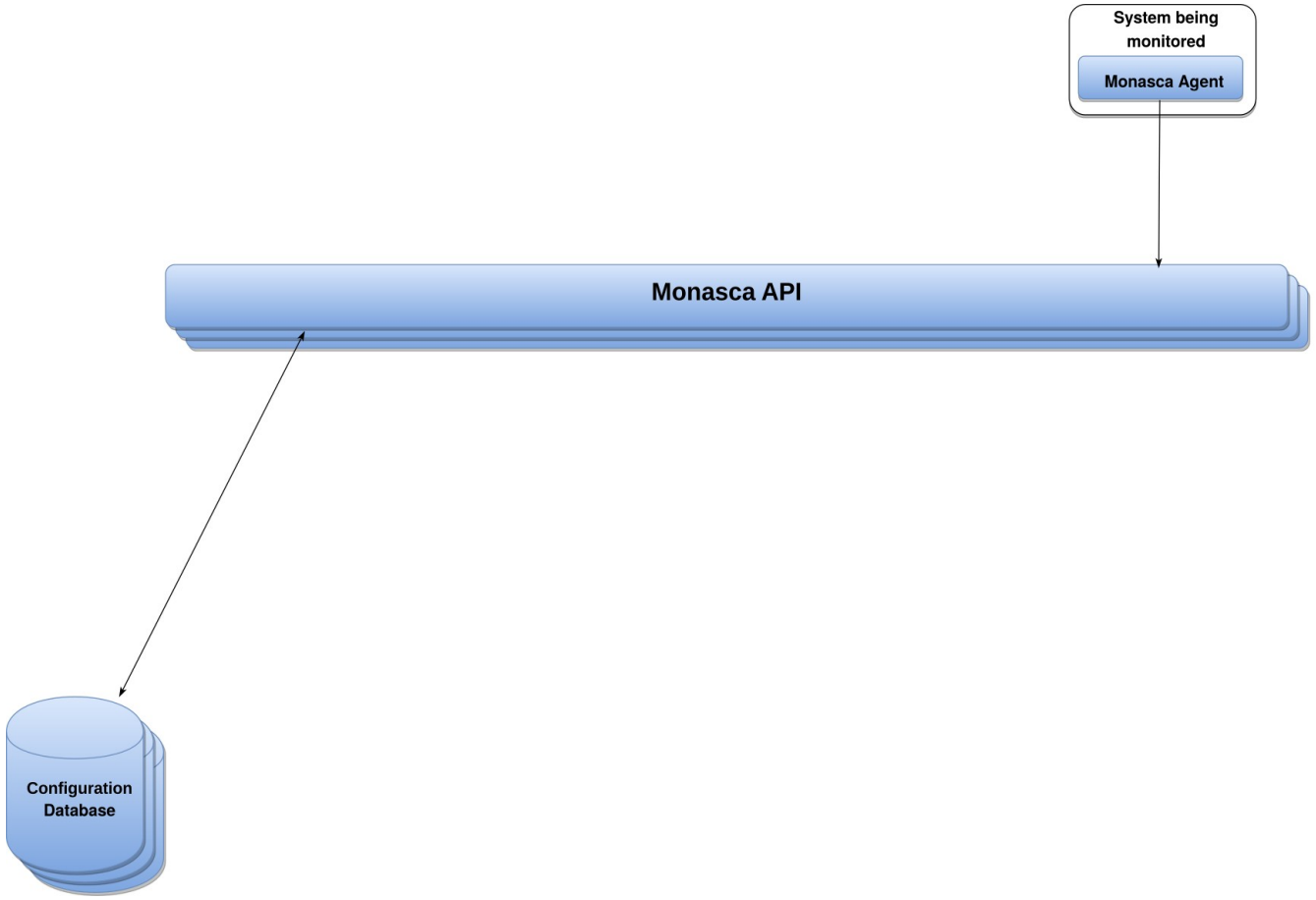
Monasca Agent (`monasca-agent`)



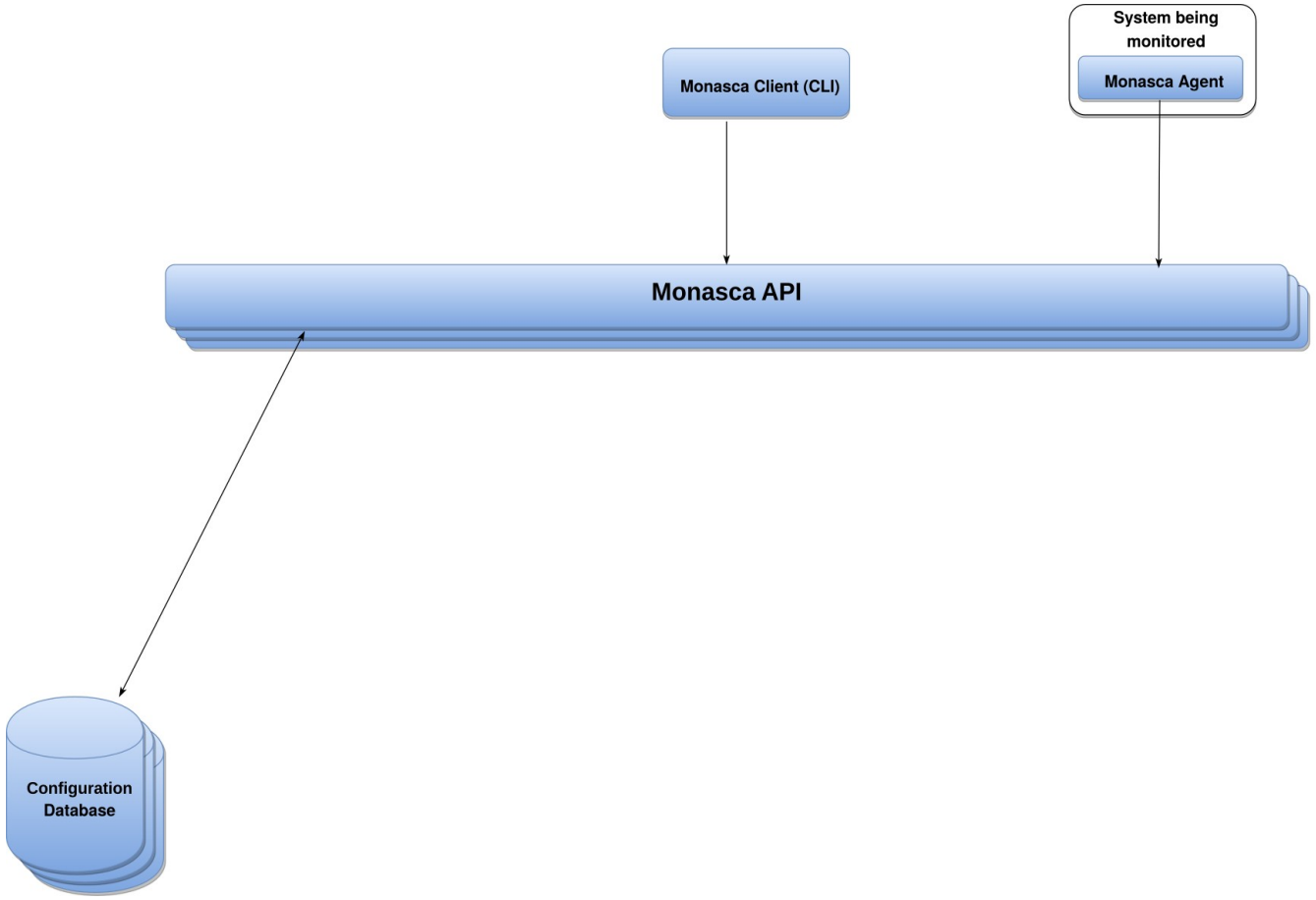
Monasca Agent (`monasca-agent`)

- Repository
  - https://github.com/openstack/monasca-agent
- Documentation
  - https://github.com/openstack/monasca-agent/blob/master/README.rst
- Purpose
  - Collect metrics on monitored systems and forward them to `monasca-api`
  - Easily extensible by adding custom plugins
- Development Information
  - Check plugins (for collecting metrics) in `monasca_agent/collector/checks_d`
  - Detection plugins (for detecting/configuring checks with `monasca-setup`) in `monasca_setup/detection/plugins`
  - Please create both if you add a new check.

Monasca Agent (`monasca-agent`)



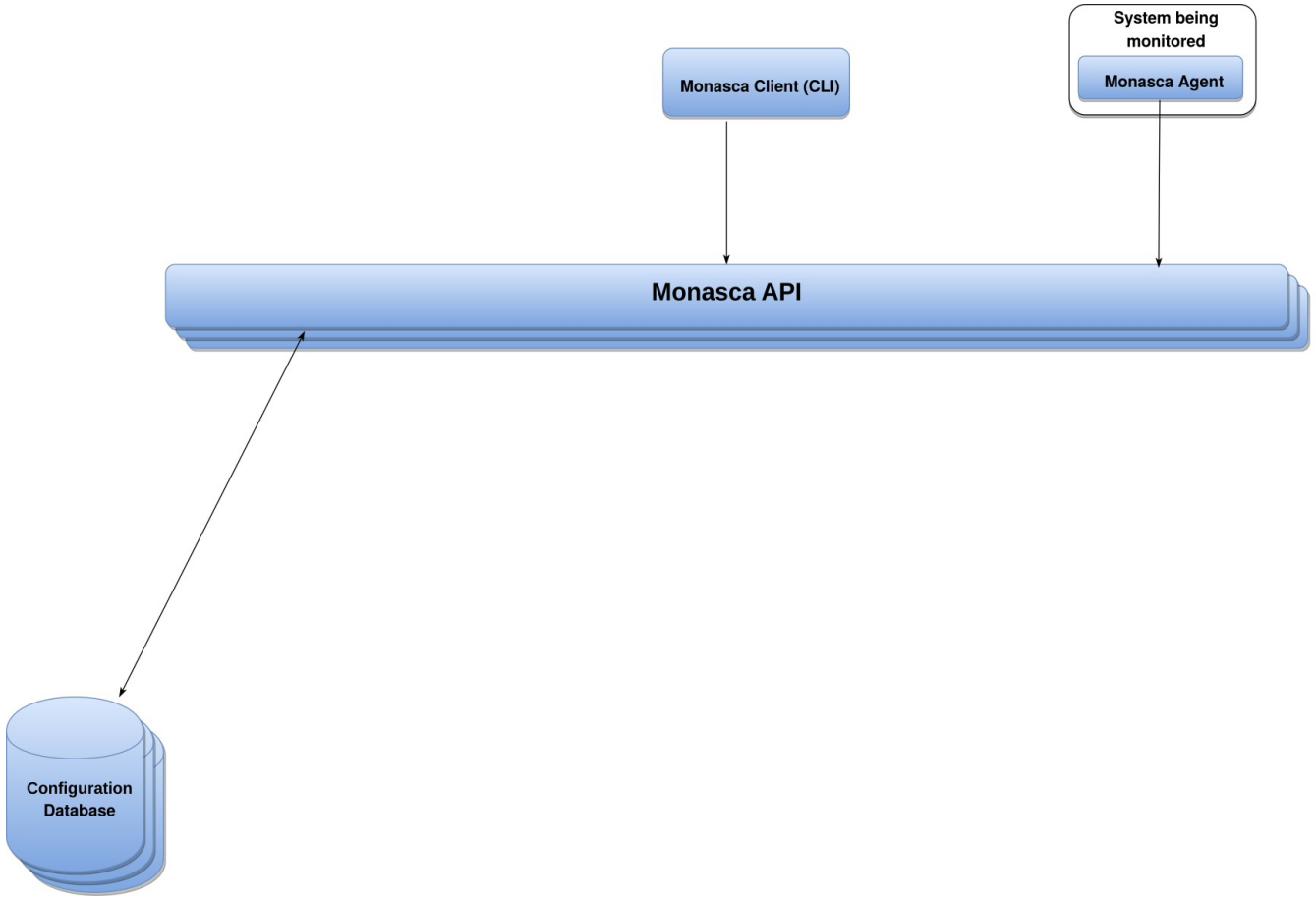
Monasca Client (`python-monascaclient`)



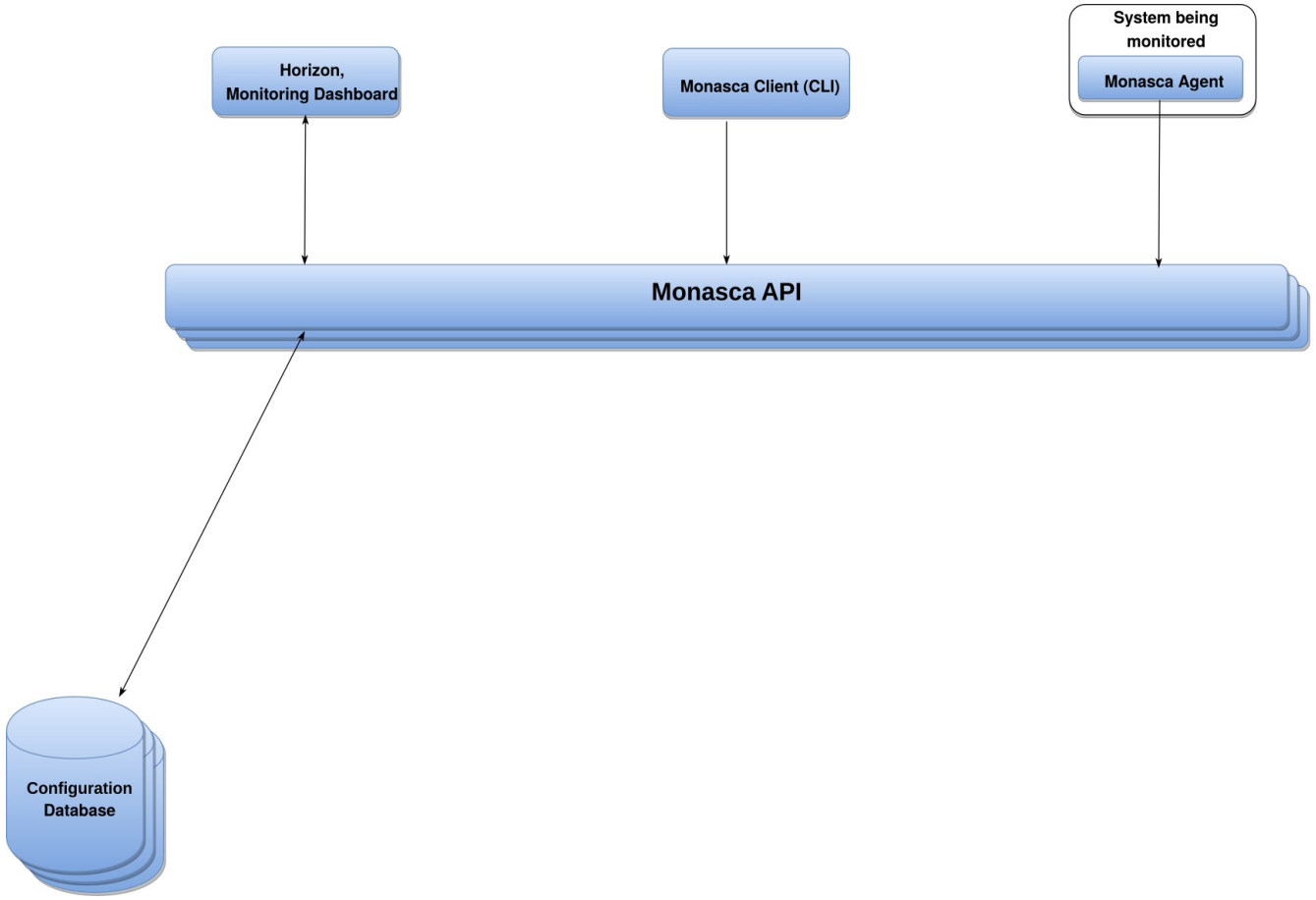
Monasca Client (`python-monascaclient`)

- Repository
  - https://github.com/openstack/python-monascaclient
- Documentation
  - https://docs.openstack.org/python-monascaclient
- Purpose
  - Python client library and CLI client for the Monasca Metrics API
  - Used by users to retrieve metrics/manipulate alarms and by all components that communicate with the Metrics API
- Development Information
  - If you extend the Metrics API, you will have to implement the client side of that extension in `python-monascaclient`.

Monasca Client (`python-monascaclient`)



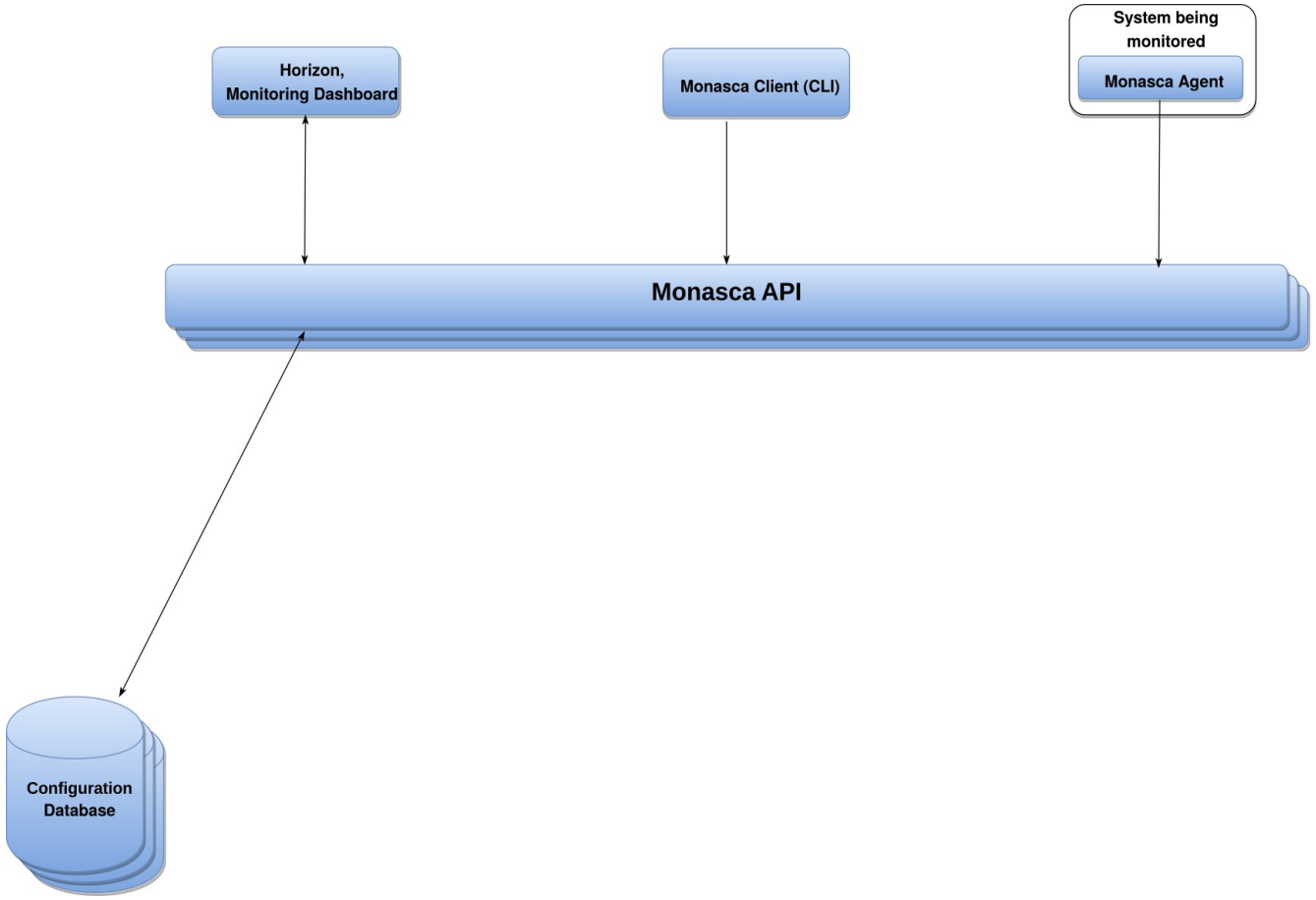
Horizon Plugin (`monasca-ui`)



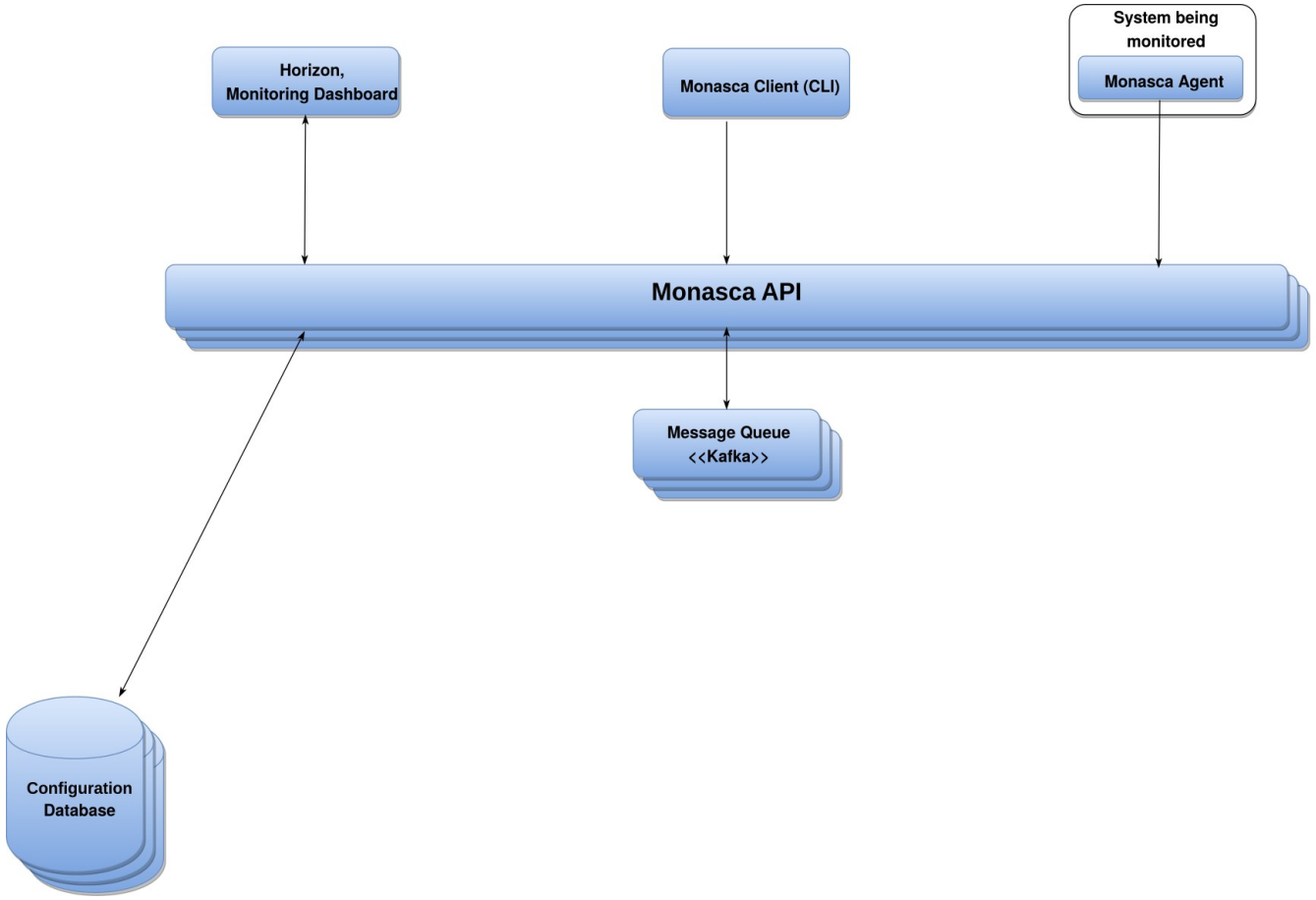
Horizon Plugin (`monasca-ui`)

- Repository
  - https://github.com/openstack/monasca-ui
- Purpose
  - Configuration of alarms/thresholds
  - Visualizing alarms
  - Provide links to metrics and log dashboards (integration with *Grafana* and *Kibana*)

Horizon Plugin (`monasca-ui`)



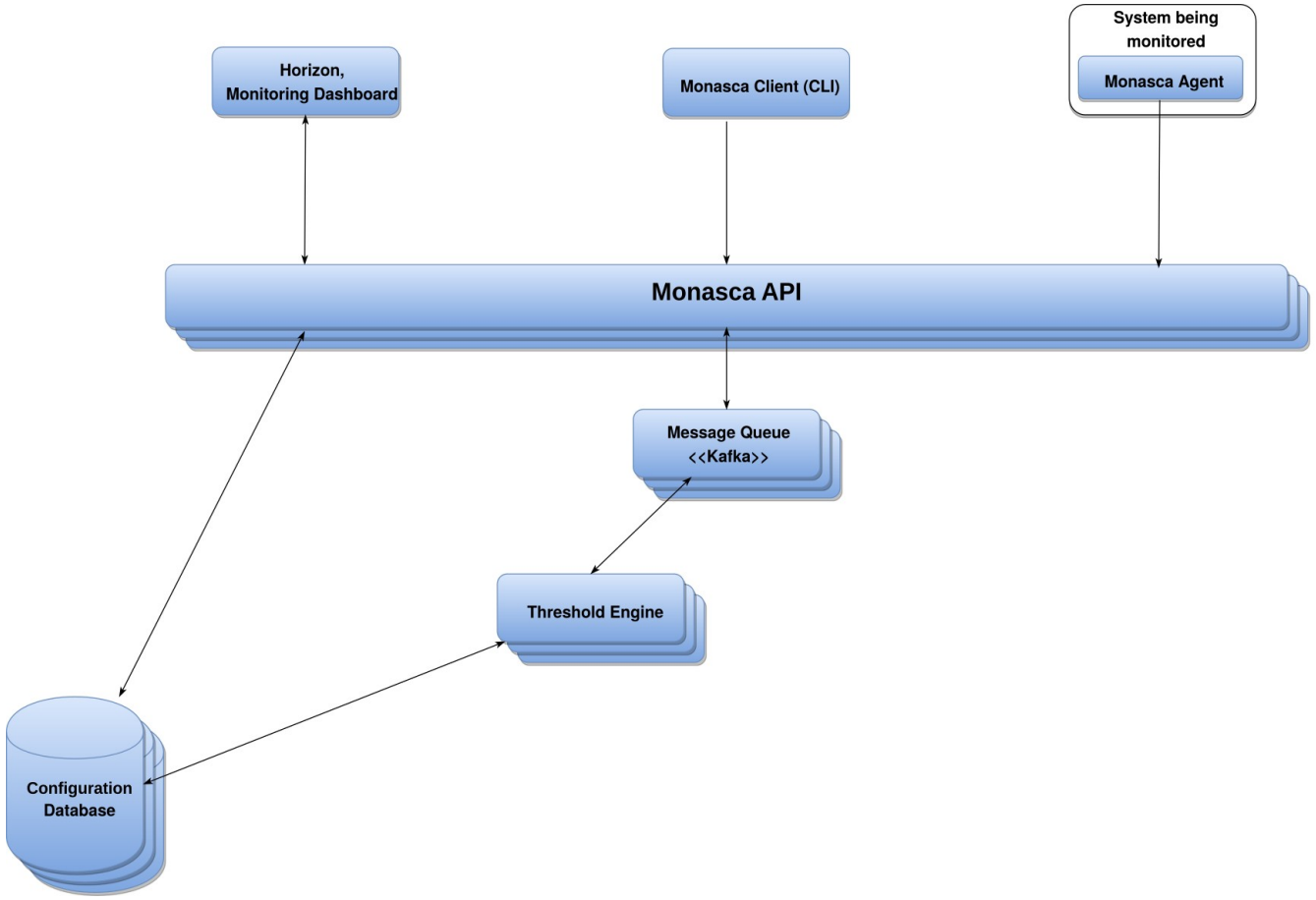
Message Queue



Message Queue: Interconnects Monasca Components

- Repository
  - N/A (third party component; *Apache Kafka*)
- Purpose
  - Shuttle metrics, notifications and log entries back and forth between components

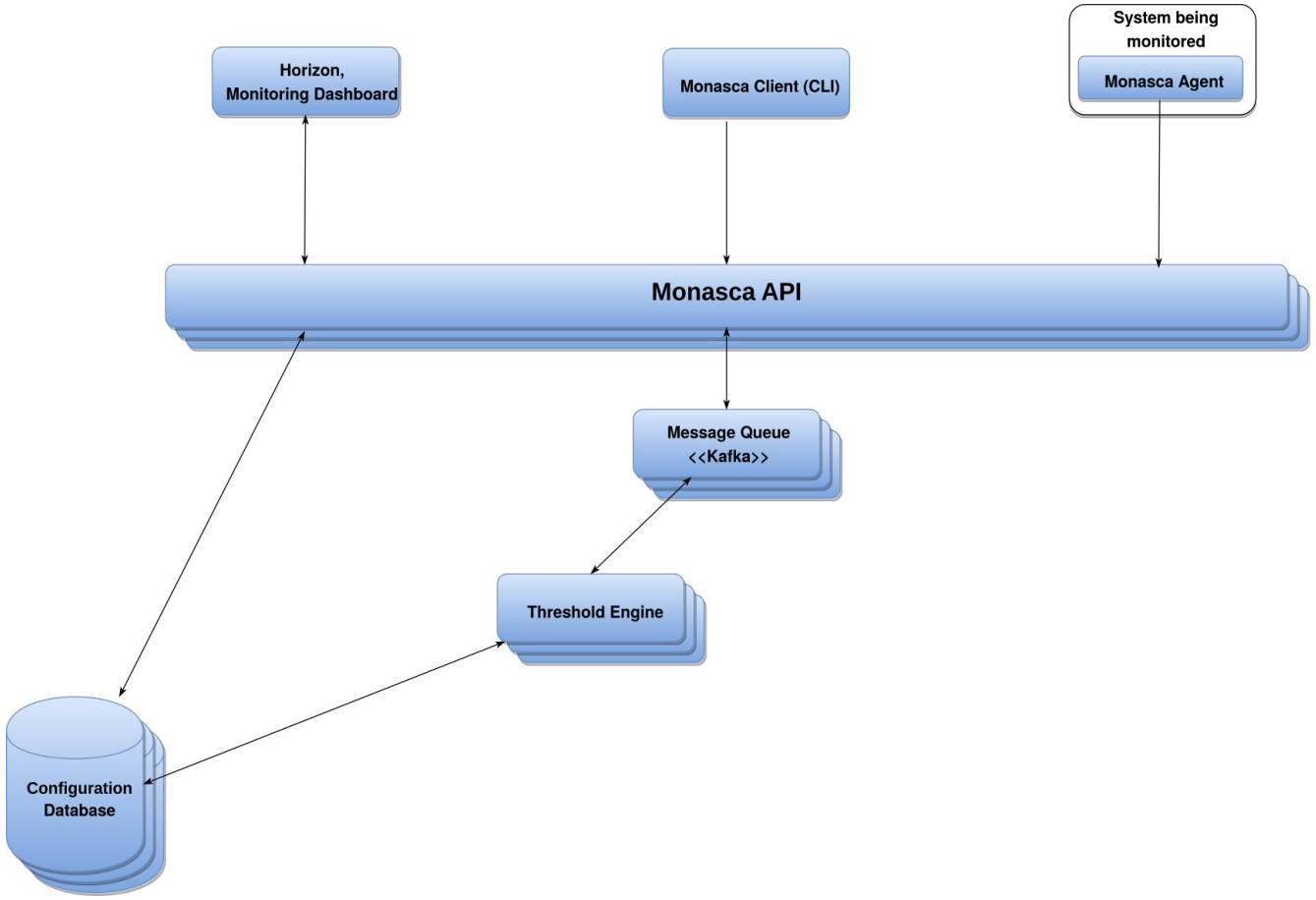
Threshold Engine (`monasca-thresh`)



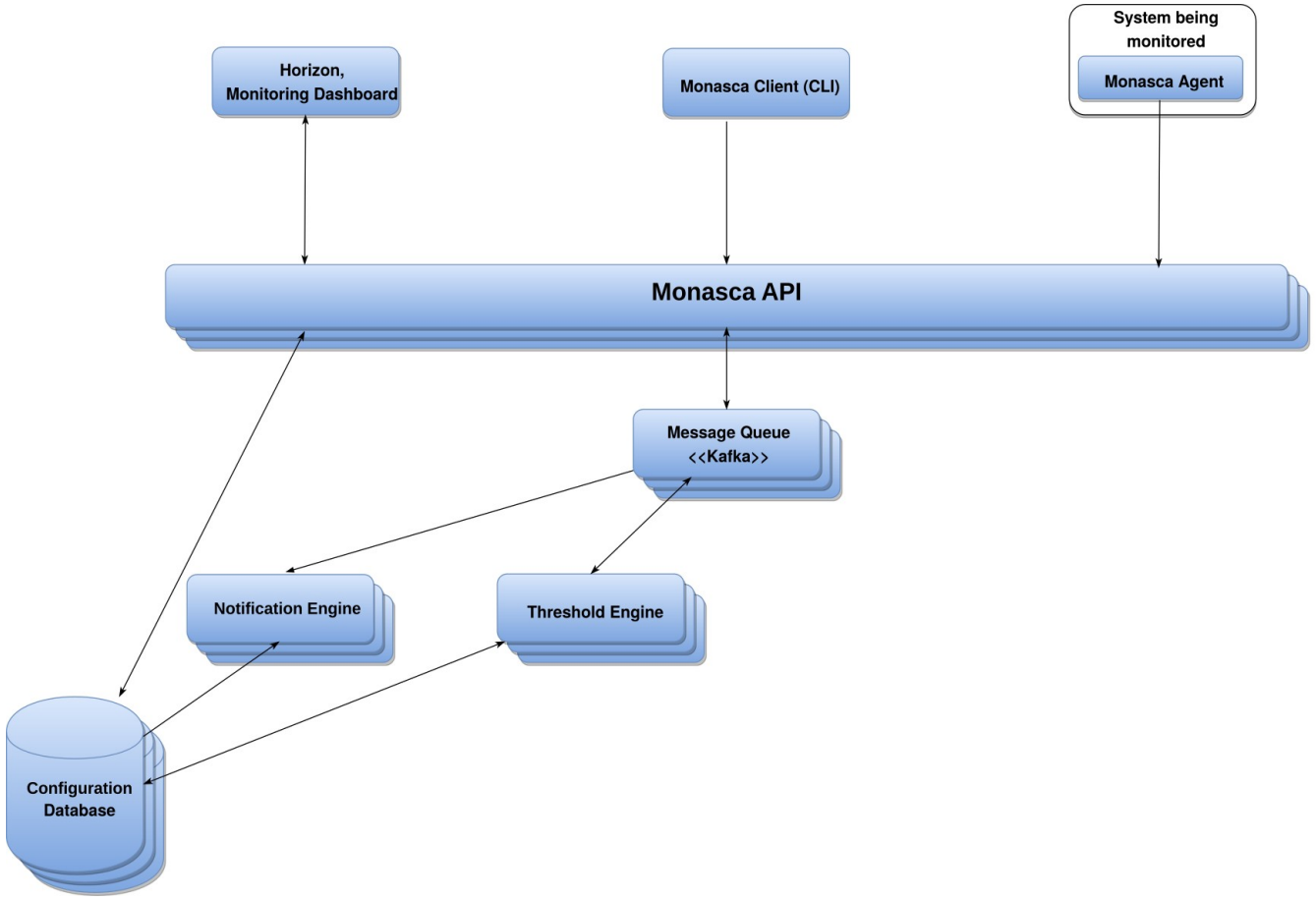
Threshold Engine (`monasca-thresh`)

- Repository
  - https://github.com/openstack/monasca-thresh
- Purpose
  - Listen in on metrics and check them against alarm thresholds
  - Produces messages for `monasca-notification` if thresholds exceeded
- Development Information
  - Implemented in Java
  - Contributions may entail changes to `monasca-common`
  - Uses *Apache Storm* for processing metrics

Threshold Engine (`monasca-thresh`)



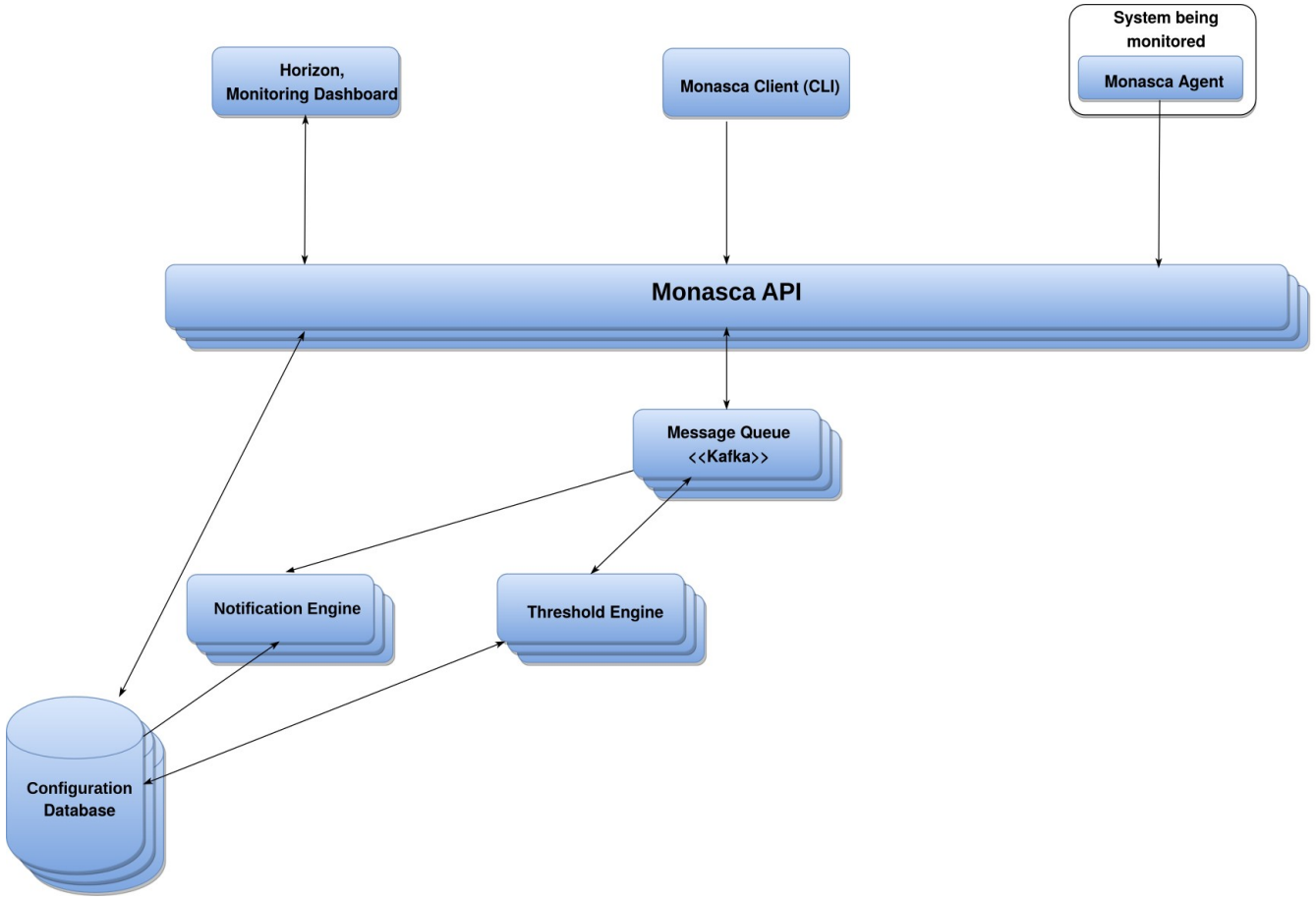
Notification Engine (`monasca-notification`)



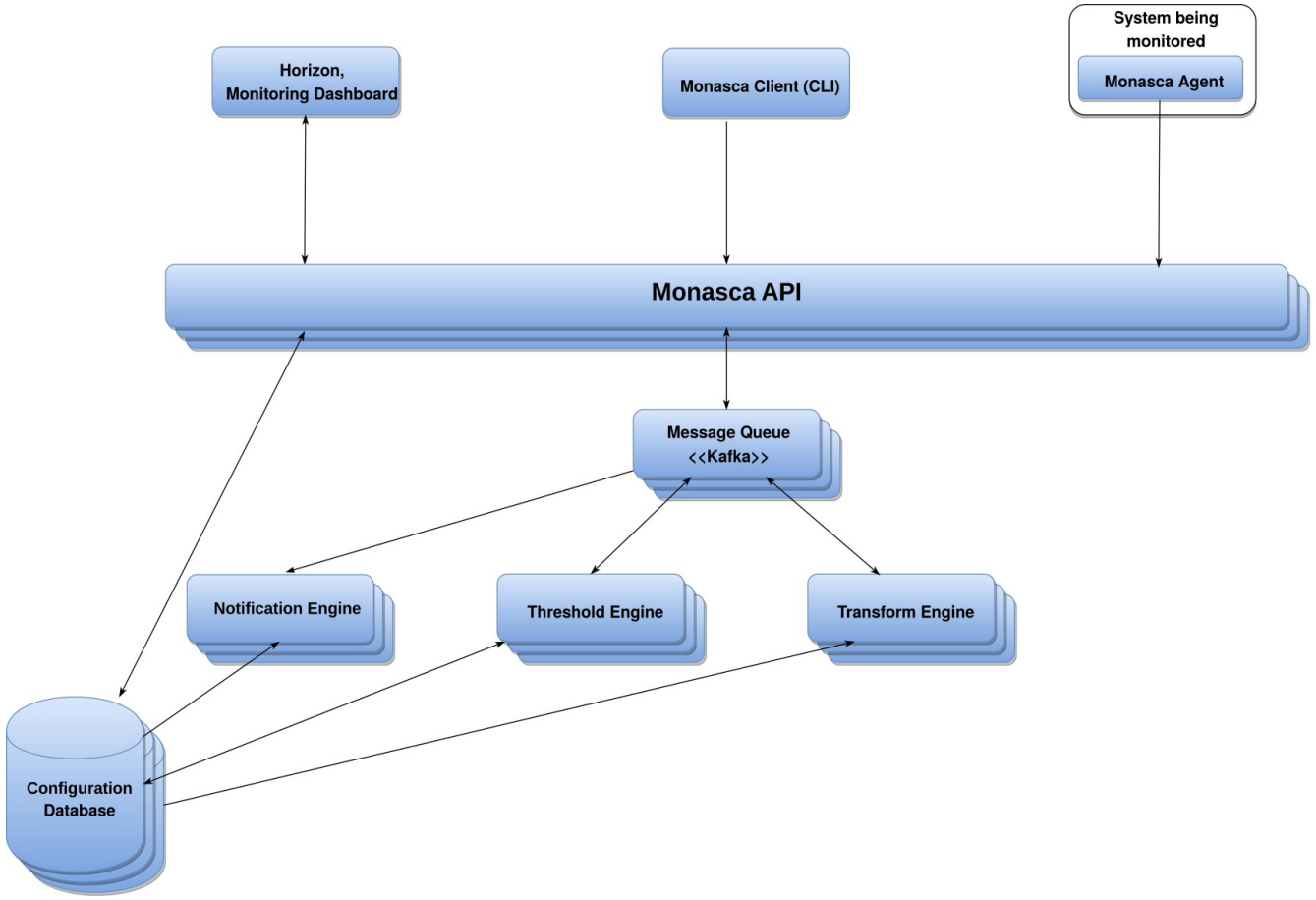
Notification Engine (`monasca-notification`)

- Repository
  - https://github.com/openstack/monasca-notification
- Purpose
  - Sends notifications if triggered by alarm
  - Supports E-Mail, Webhooks and various chat protocols
- Development Information
  - Plugin based
  - Plugins in `monasca_notification/plugins/`
  - Plugins must inherit from `monasca_notification.plugins.abstract_notifier.AbstractNotifier`
  - Plugins must be registered in configuration file

Notification Engine (`monasca-notification`)



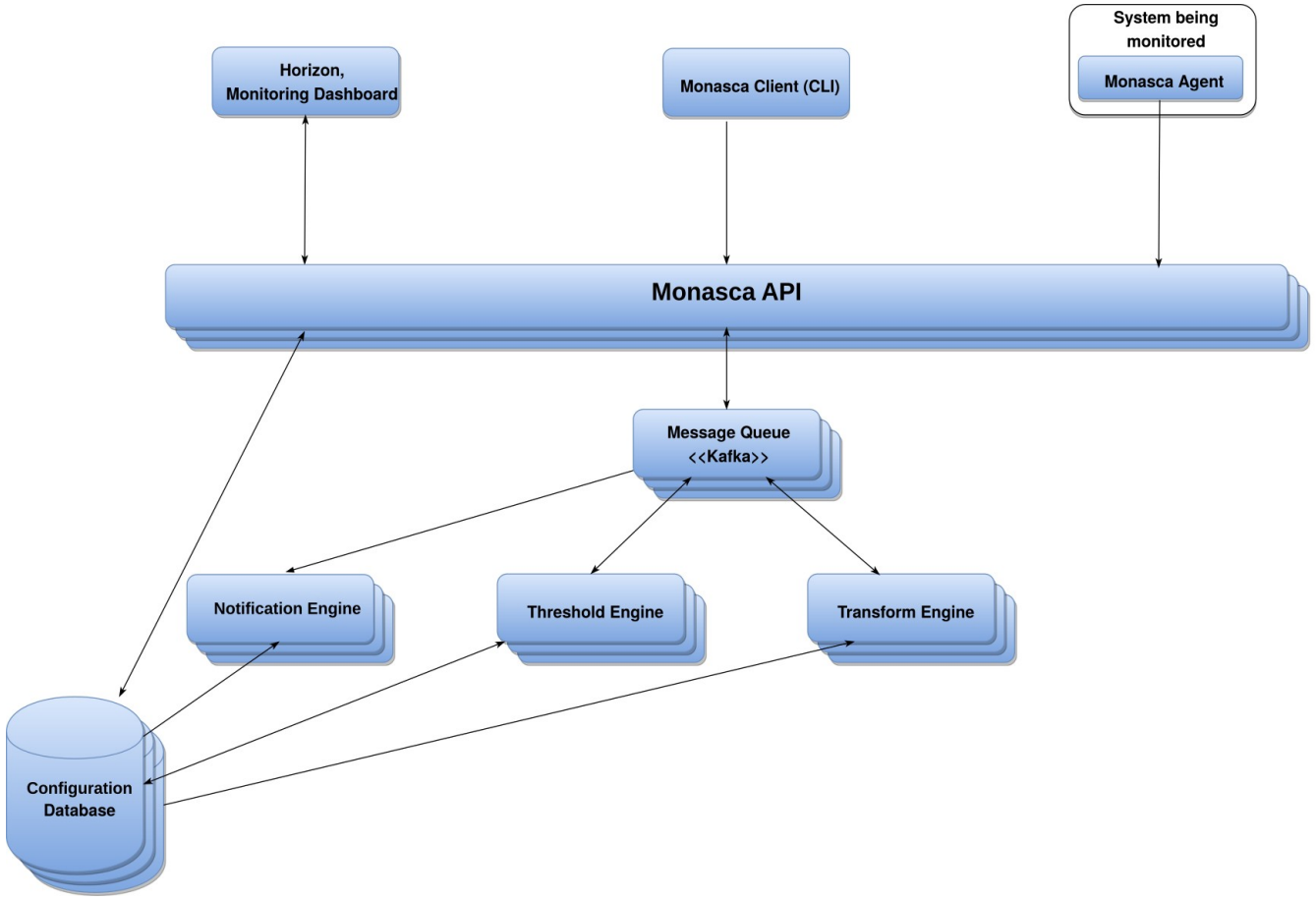
Transform Engine (`monasca-transform`)



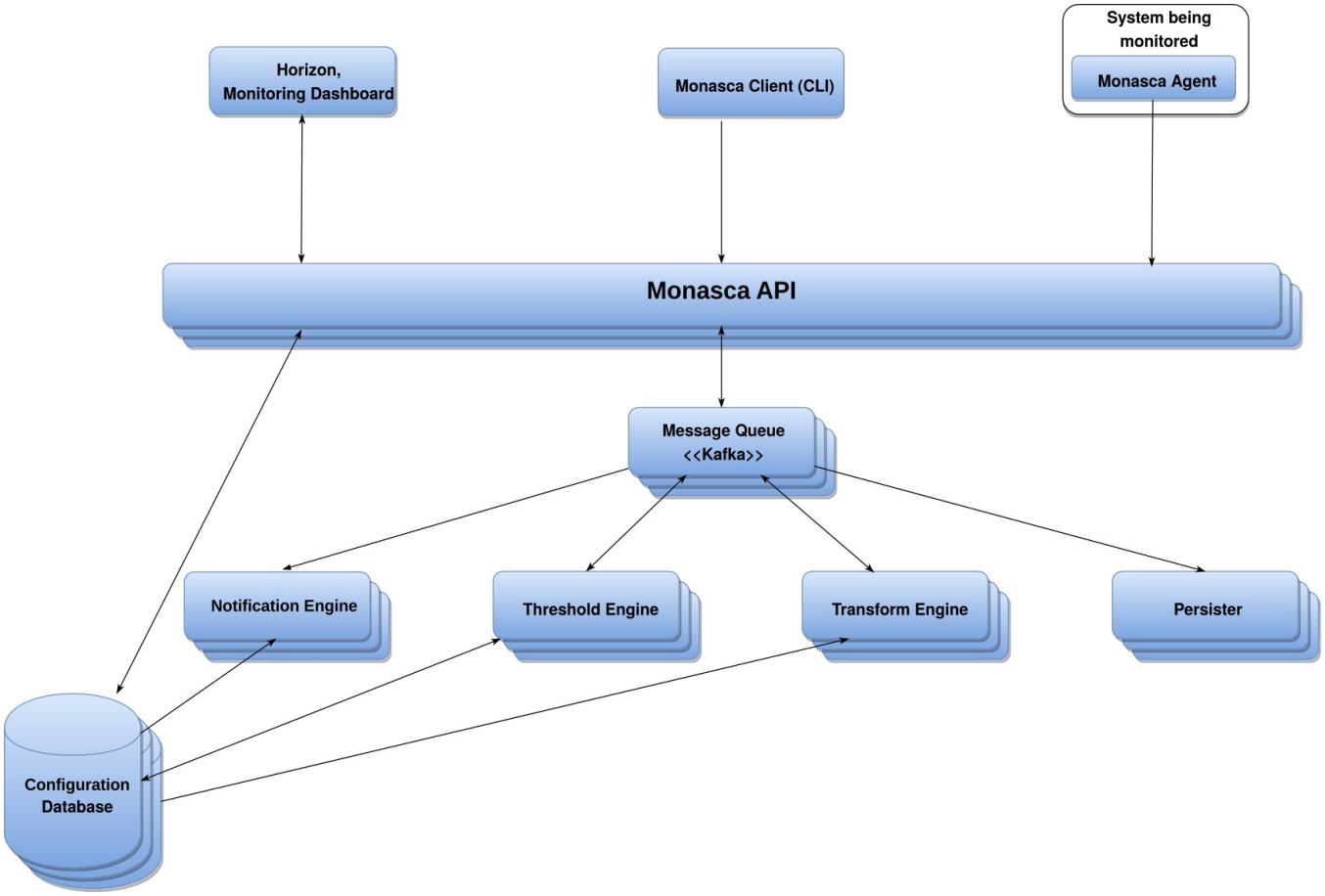
Transform Engine (`monasca-transform`)

- Repository
  - https://github.com/openstack/monasca-transform
- Purpose
  - Publish transformed (usually aggregated) metrics as synthetic new metrics

Transform Engine (`monasca-transform`)



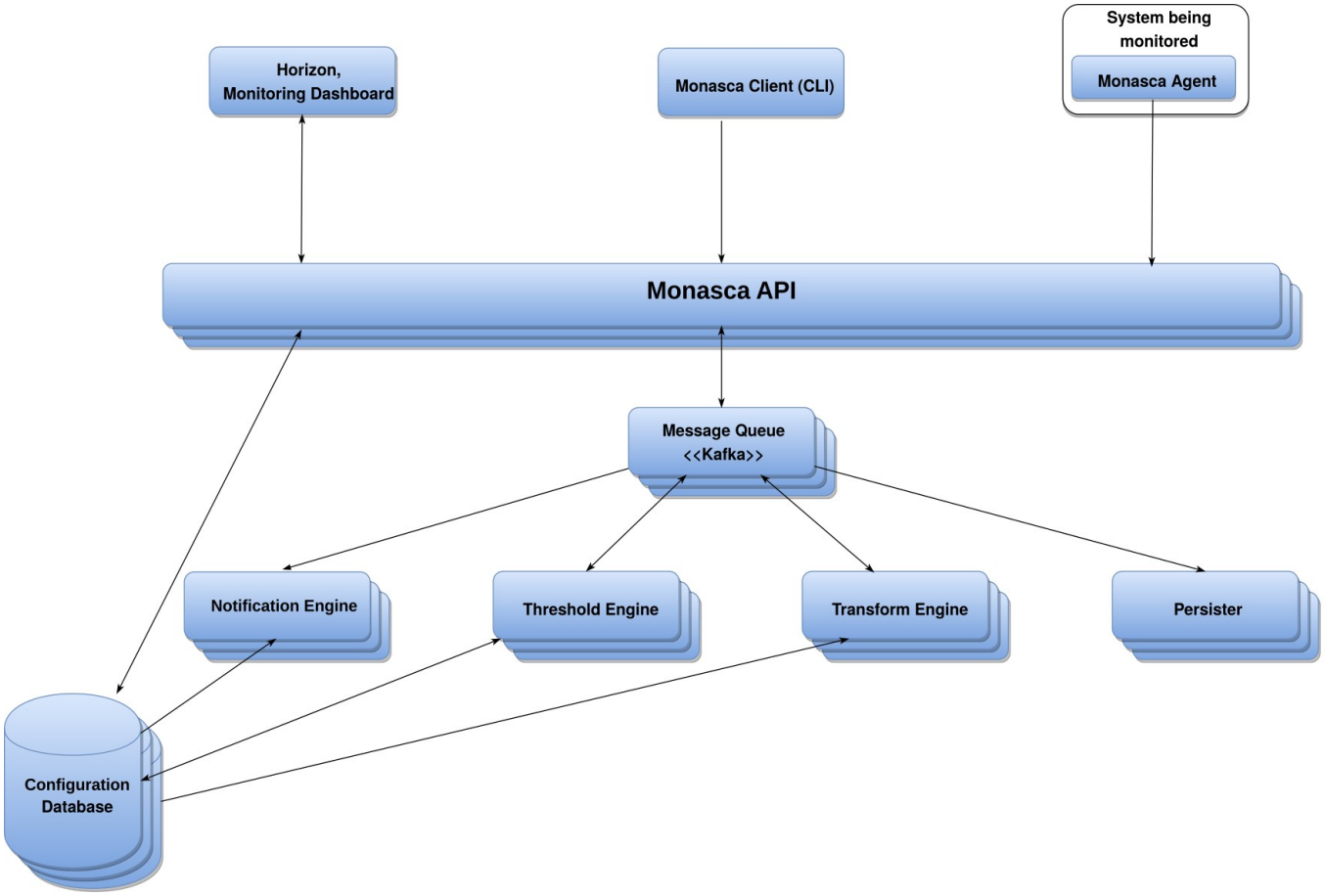
Persister (`monasca-persister`)



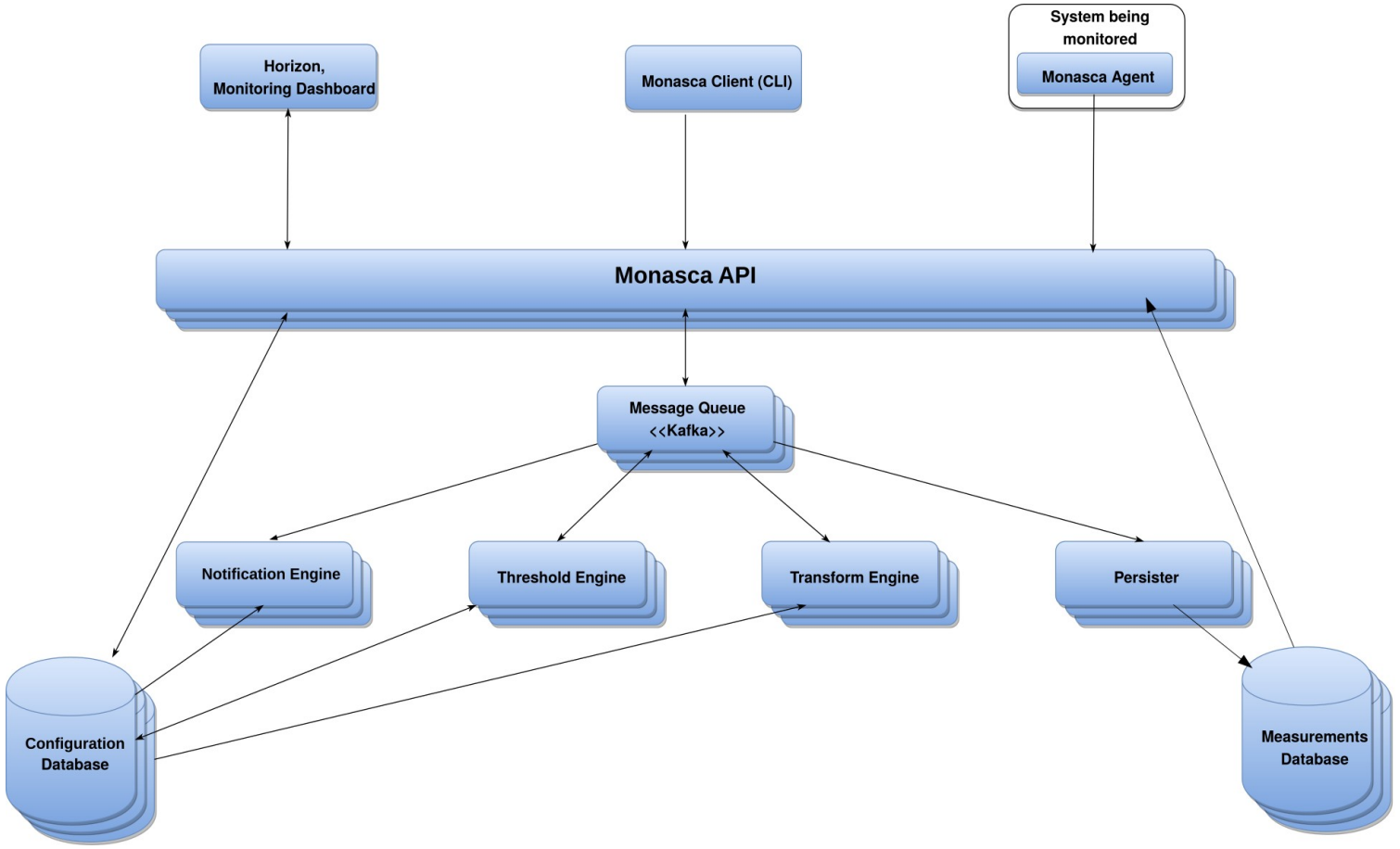
Persister (`monasca-persister`)

- Repository
  - https://github.com/openstack/monasca-persister
- Purpose
  - Consumes metrics from message queue
  - Stores metrics in time series database
- Development Information
  - Two implementations: Java and Python
  - Contributions may entail changes to `monasca-common`

Persister (`monasca-persister`)



Time Series Database for Measurements



Time Series Database for Measurements

- Repository
  - N/A (third party component; can be *InfluxDB*, *Apache Cassandra* or *Vertica*)
- Purpose
  - Store metrics
- Development Information
  - To support a new type of time series database, you will need to add code to `monasca-common`, `monasca-api` and `monasca-persister`.

Monasca Logging Architecture

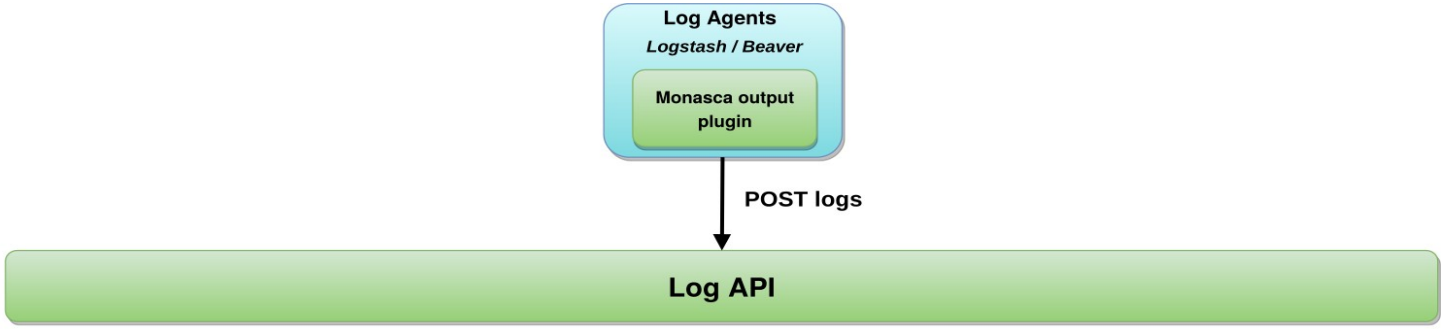
Monasca Log API (`monasca-log-api`)

Monasca Log API (`monasca-log-api`)

- Repository
  - https://github.com/openstack/monasca-log-api
- Purpose
  - Receives log messages from agents
- Development Information
  - Repository contains logging specific parts of documentation
  - Contributions may entail changes to `monasca-common`

Monasca Log API (`monasca-log-api`)

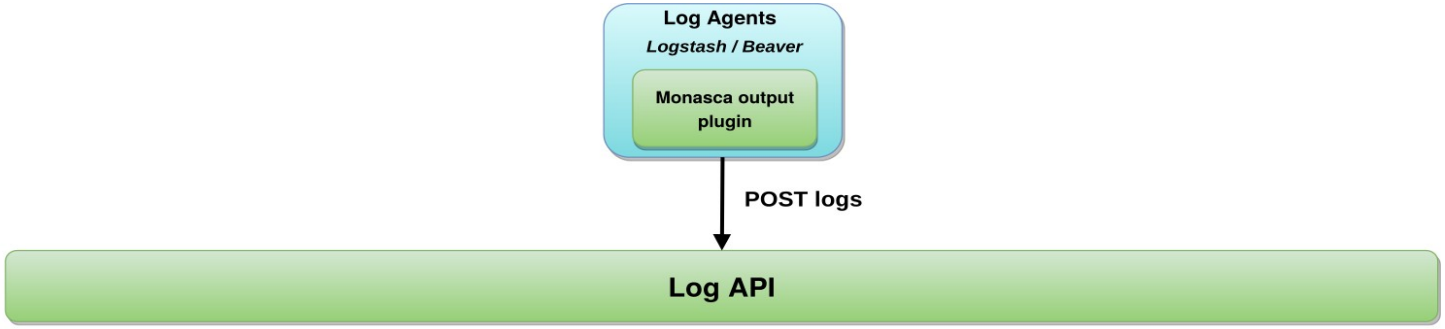
Log Agents


Log Agents
Logstash / Beaver
Monasca output plugin
POST logs
Log API

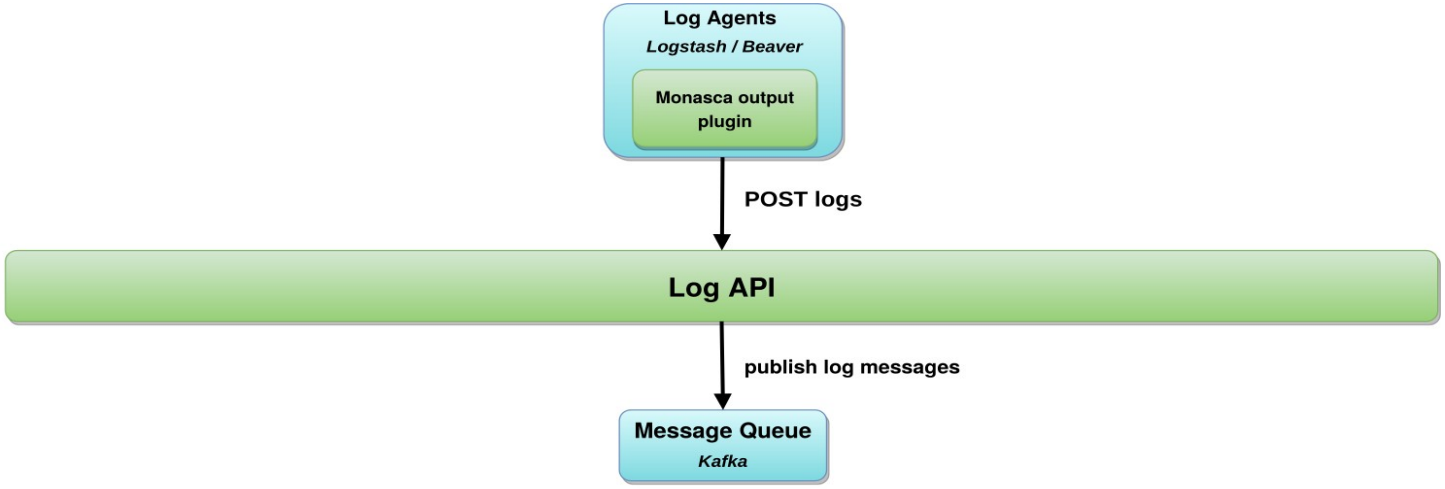
Log Agents

- Repository: N/A
- Purpose: Send logs
- Not part of Monasca: logstash, beaver or fluentd with Monasca output plugin.

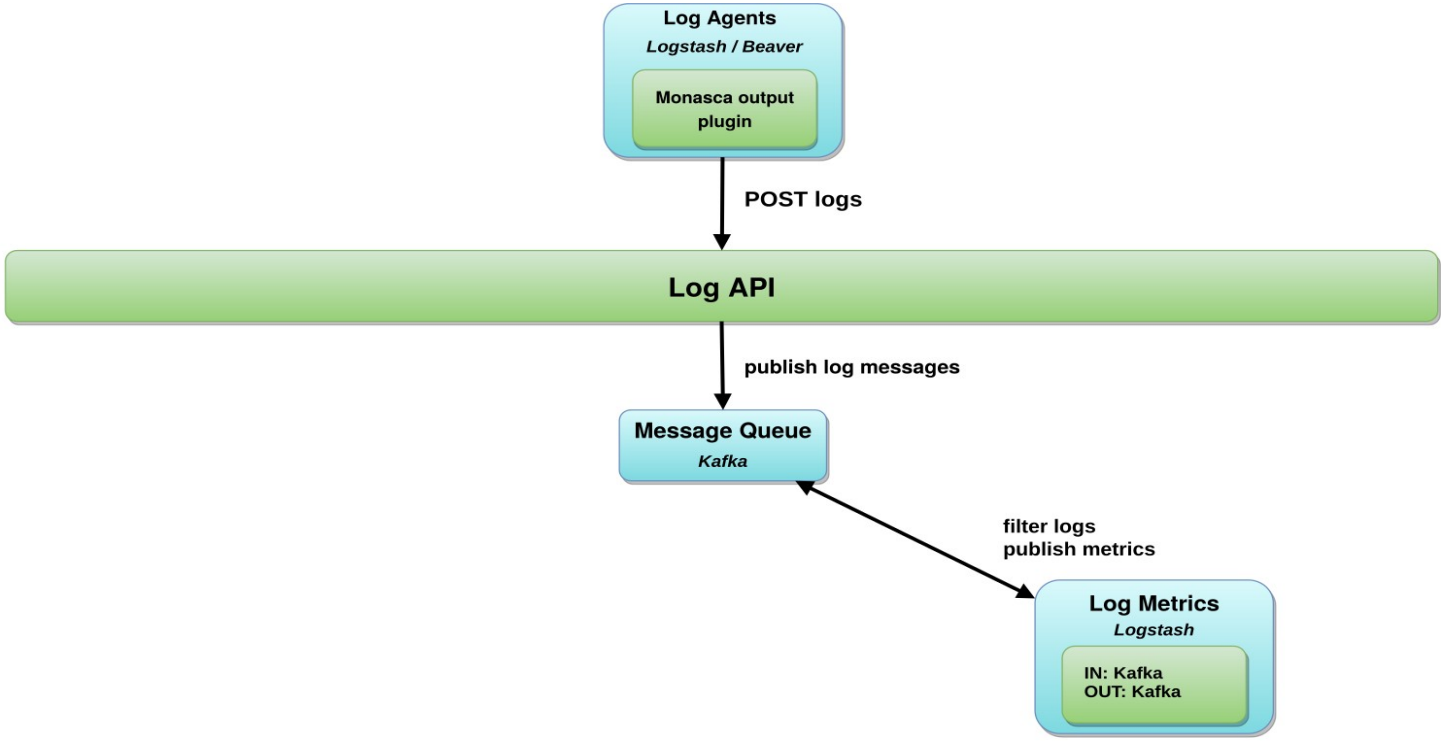
Log Agents


Log Agents
Logstash / Beaver
Monasca output plugin
POST logs
Log API

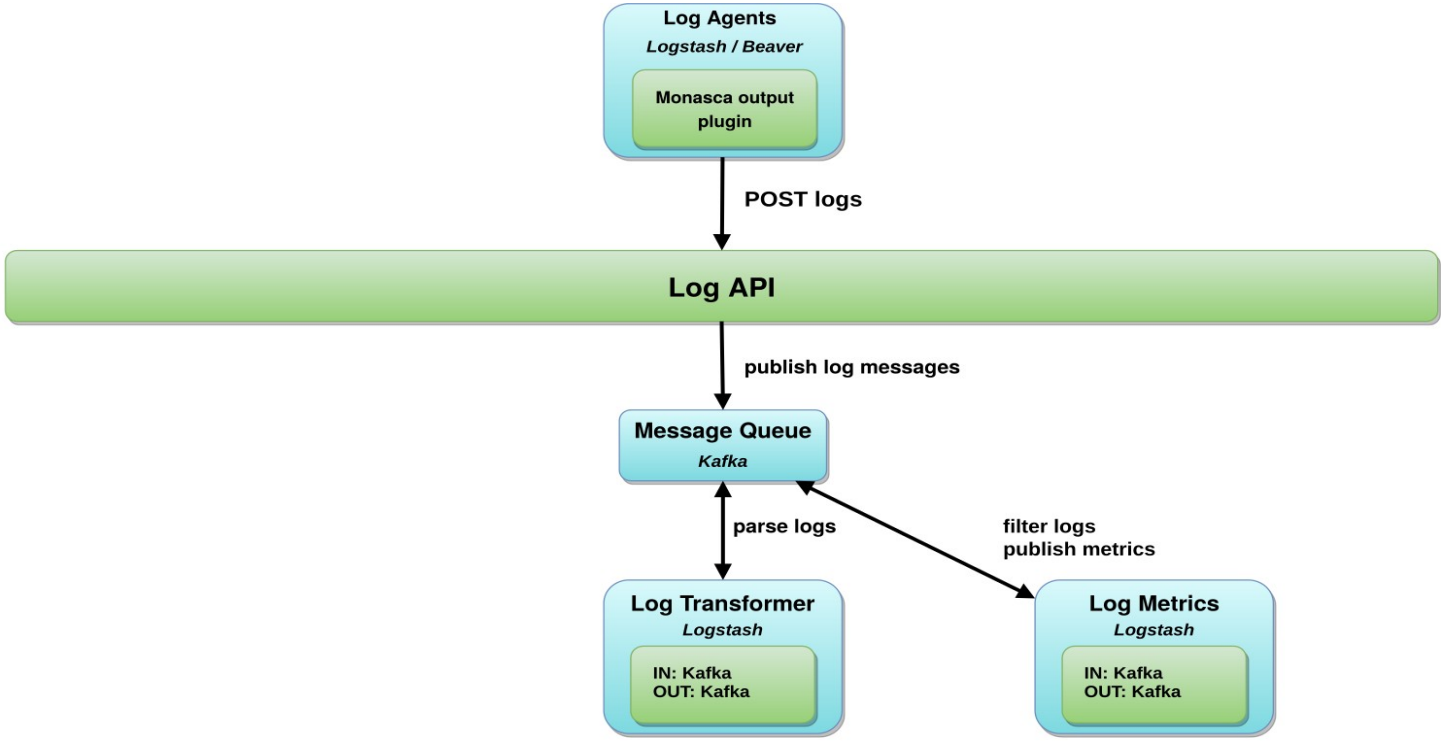
Monasca Logging



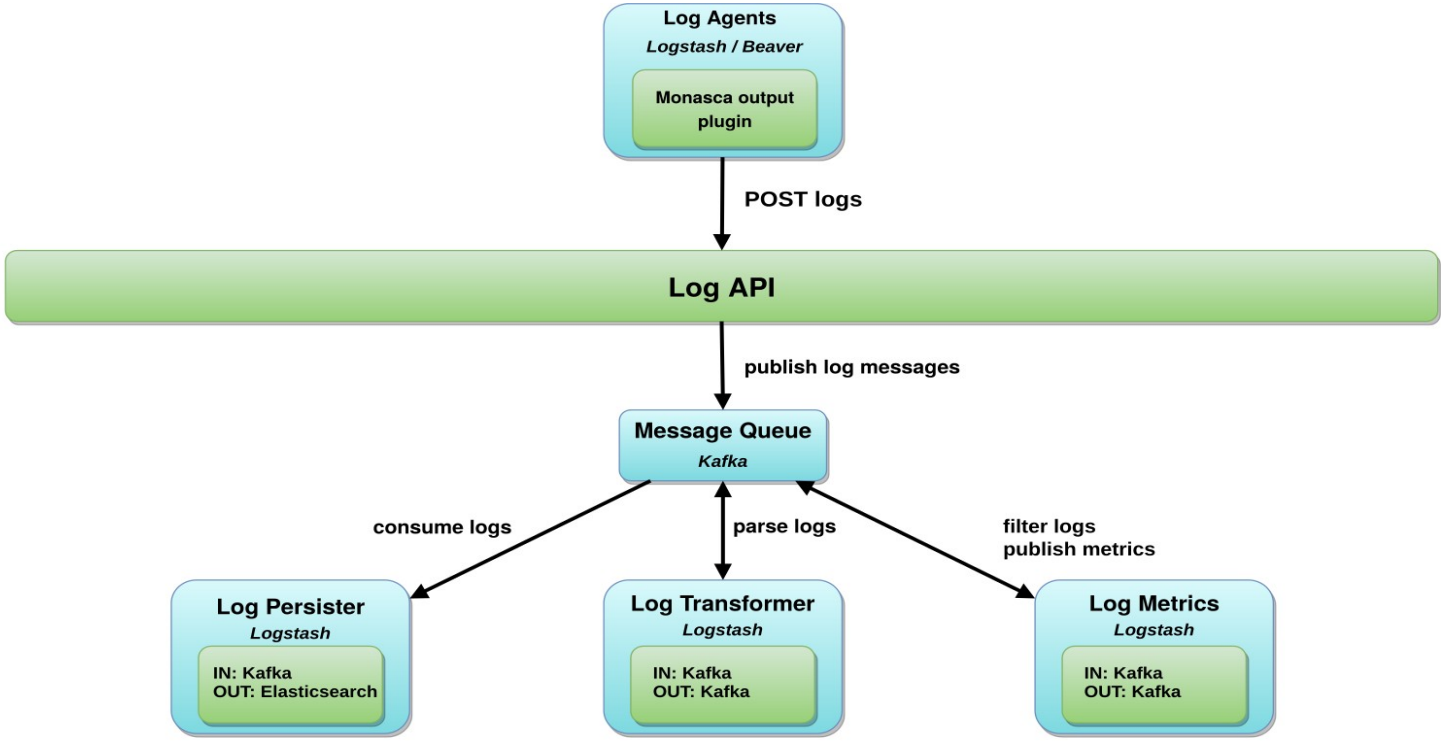
Log Metrics


Log Agents
Logstash / Beaver
Monasca output plugin
POST logs
Log API
publish log messages
Message Queue
Kafka
filter logs
publish metrics
Log Metrics
Logstash
IN: Kafka
OUT: Kafka

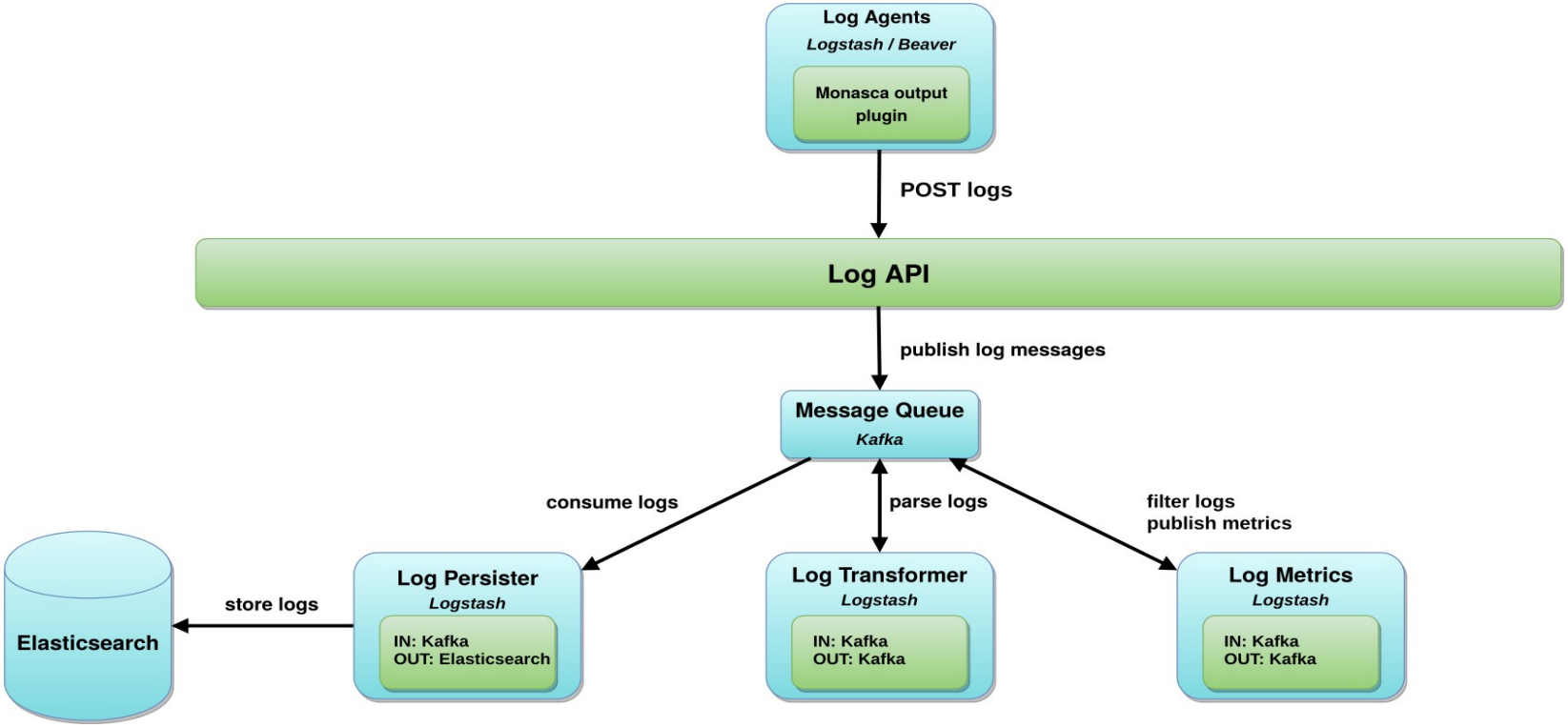
Log Transformer



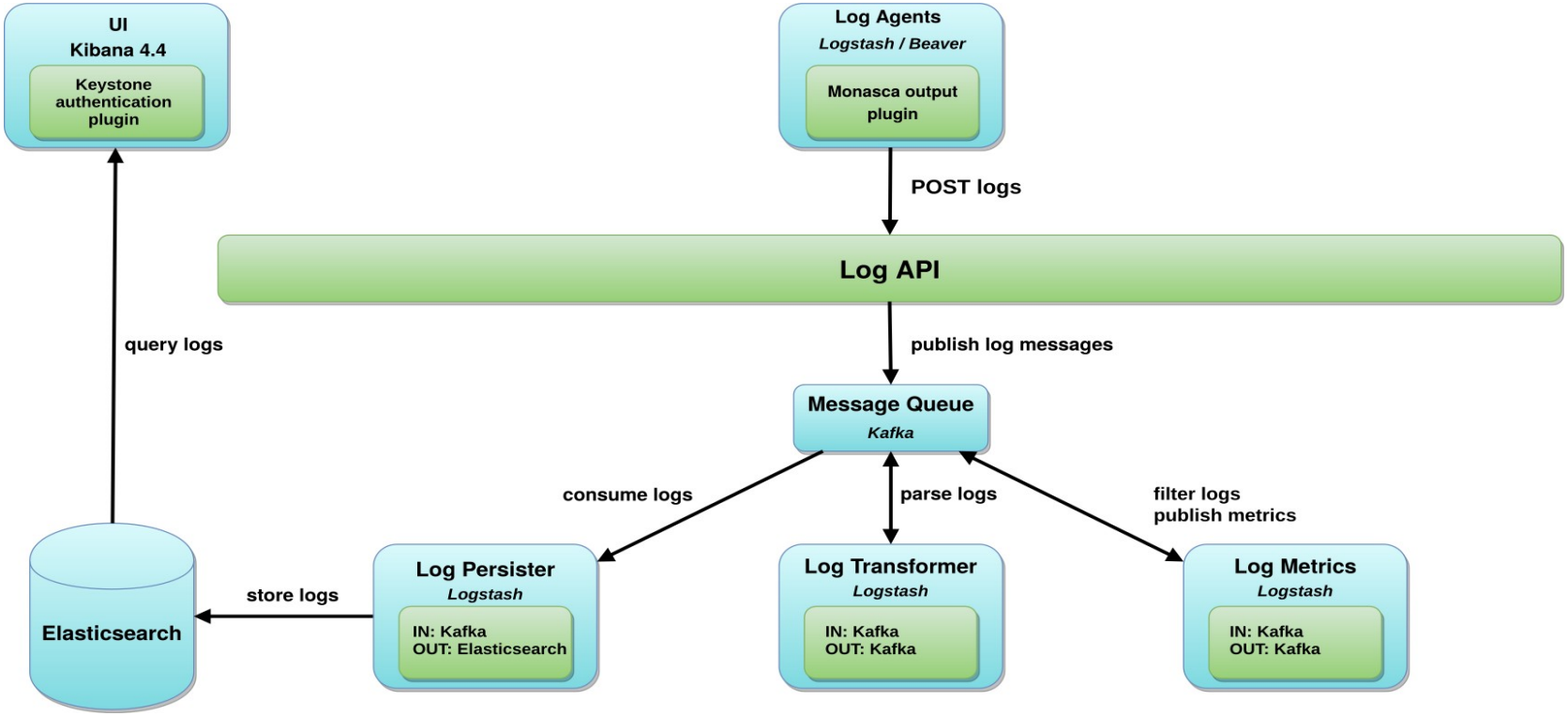
Log Persister



Elasticsearch



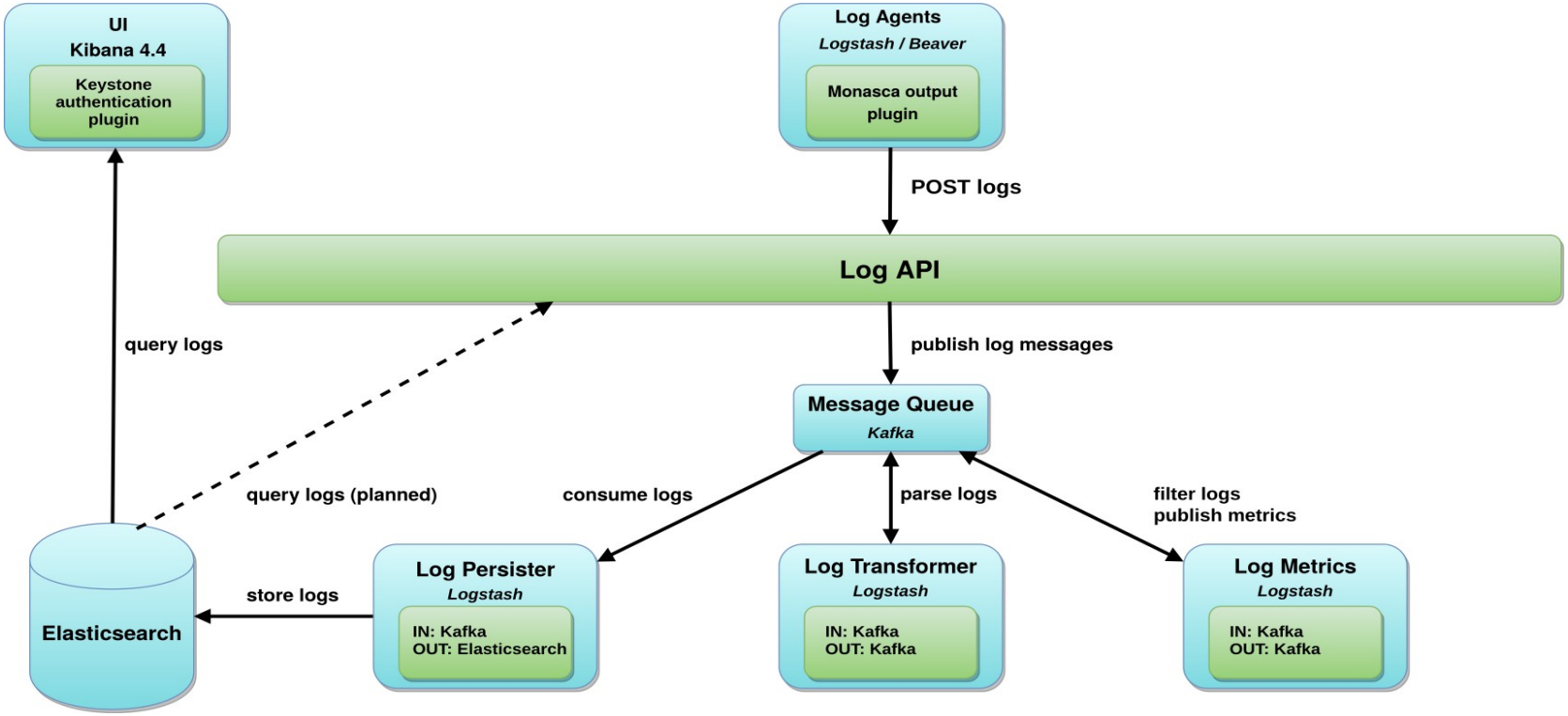
Kibana


UI
Kibana 4.4
Keystone authentication plugin
Log Agents
Logstash / Beaver
Monasca output plugin
POST logs
Log API
query logs
publish log messages
Message Queue
Kafka
consume logs
parse logs
filter logs
publish metrics
Elasticsearch
store logs
Log Persister
Logstash
IN: Kafka
OUT: Elasticsearch
Log Transformer
Logstash
IN: Kafka
OUT: Kafka
Log Metrics
Logstash
IN: Kafka
OUT: Kafka

monasca-kibana-plugin

- Repository
  - https://github.com/openstack/monasca-kibana-plugin

Monasca Logging



Hands-On Workshop

- Interactive Jupyter notebook
- Demonstrates main Monasca functionalities
- https://github.com/martinchacon/monasca-bootcamp/
- Today, 4:20pm - 5:50pm
  Hall 7 - Level 1 - 7.1a / NY1

Development Environment

Devstack Setup for Monasca

- `local.conf` for default (Python based) Monasca stack

```
enable_plugin monasca-api \
      git://git.openstack.org/openstack/
monasca-api
```

- `local.conf` setting for Java based persister

```
MONASCA_PERSISTER_IMPLEMENTATION_LANG=ja
va
```

Devstack Setup with Vagrant

```
# cd monasca-api/devstack
# vagrant up
```

monasca-docker

- Containerized Monasca deployed with Docker Compose

```
# git clone
https://github.com/monasca/monasca-
docker
# cd monasca-docker
# docker-compose up
```

Running unit tests

```
# cd $REPO
# tox
# tox -e py27,py35
# tox -e pep8
```

Running Integration (Tempest) Tests in Devstack

- Add *monasca-tempest-plugin* to `local.conf`

```
enable_plugin monasca-tempest-plugin \
  https://git.openstack.org/openstack/monasca-tempest-plugin
```

- Run tests

```
# cd /opt/stack/tempest
# tempest run -r monasca_tempest_tests.tests.api
# tempest run -r monasca_tempest_tests.tests.log_api
```

Running Tempest Tests with `monasca-docker`

- Add section to `docker-compose.yaml`:

```
tempest-tests:
  image: monasca/tempest-tests:latest
  environment:
    KEYSTONE_SERVER: "keystone"
    STAY_ALIVE_ON_FAILURE: "true"
    MONASCA_WAIT_FOR_API: "true"
```

- Run tests

```
# docker-compose up -d tempest-tests
```

Become part of our community

Why Contribute?

- Pluggable
- Modular
- Customizable
- Small and friendly community

How to contribute?

- Contributor Guide
- We use StoryBoard!
  - Priortized features
  - Bugs
  - Feature requests
- Specifications repository
  - openstack/monasca-specs

Work to do

- Project priorities
  - http://specs.openstack.org/openstack/monasca-specs/priorities/stein-priorities.html
- Kanban Board
  - https://storyboard.openstack.org/#!/board/111

Where can you help?

- Reviews
- Backlog
  https://storyboard.openstack.org/#!/board/111
- Bugfixes and Bug Reports
  https://storyboard.openstack.org/#!/board/114
- Community wide goals
- Installers
- Documentation

Monitoring as a Service in HPC Cloud

Wednesday, November 14, 3:30pm-3:40pm

Level 2 - Marketplace Demo Theater

Questions?

Thank You!

Slides and Transcript:

https://bit.ly/2IfT7c7

Thank you!