How we replaced a Gartner APM Leader with an open-source monitoring solution

Henning Schulz, Tobias Angerstein
...that’s how to operate software without monitoring!
Commercial APM tools have great features:

- Metric collection
- Alerting
- Baselining
- "Management-ready" reporting
- Dynamic instrumentation
- Visualization & dashboarding
- Distributed tracing
- User management
- UI-based configuration
- Dependency discovery
...and painpoints
There are lots of APM tools...
There are lots of APM tools...
There are lots of APM tools...
...to build a tailored APM solution
Dynamic Instrumentation of Java Services

inspectIT

Ocelot

Customer Service
Instrumentation of Java Services

Monitored Application

Customer Service

inspectIT

Ocelot

Customer Service

inspectIT

Ocelot
Runtime Attachment

```
java -javaagent:"/path/to/inspectit-ocelot-agent-0.3.jar" -jar my-java-program.jar
```

```
# Find the process ID of the JVM to which the agent should be attached to
# ps aux | grep java
# jps -lV # Requires an installed JDK
# Attach the latest Ocelot agent to your JVM
bash <(curl -s https://inspectit.github.io/inspectit-ocelot/attach.sh) <JVM_PID>

# See awesome OpenCensus metrics
wget -qO- localhost:8888
```
Interoperability
Interoperability

Monitored Application

Customer Service

inspectIT Ocelot

Open Census

Customer Service

inspectIT Ocelot
Metric Collection & Visualization

CPU Usage

Memory - Heap

Memory - Non-Heap

Loaded Classes

23633

Unloaded Classes

11

# Live Threads

# Threads by States

Current CPU Usage

0.4581%
Dependency Discovery
Dependency Discovery

- **discovery-server**
  - Requests: 80
  - Avg. Resp. Time: 1 ms

- **customers-service**
  - Requests: 3.2 k
  - Avg. Resp. Time: 0 ms

- **Java**
  - Requests: 20

---

NOVATEC
Distributed Tracing
## Extend Jaeger

![Jaeger UI](image)

**Trace Detail**

<table>
<thead>
<tr>
<th>Name</th>
<th>Count</th>
<th>Total</th>
<th>Avg</th>
<th>Min</th>
<th>Max</th>
<th>Total Exec</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET /api/gateway/owners/3</td>
<td>4</td>
<td>44.26ms</td>
<td>11.06ms</td>
<td>2.52ms</td>
<td>20.82ms</td>
<td>16.51ms</td>
</tr>
<tr>
<td>api-gateway</td>
<td>3</td>
<td>41.33ms</td>
<td>13.78ms</td>
<td>4.17ms</td>
<td>20.82ms</td>
<td>13.70ms</td>
</tr>
<tr>
<td>customers-service</td>
<td>1</td>
<td>2.02ms</td>
<td>2.92ms</td>
<td>2.82ms</td>
<td>2.92ms</td>
<td>2.81ms</td>
</tr>
<tr>
<td>HikariProxyPreparedStatement.executeQuery</td>
<td>2</td>
<td>0.33ms</td>
<td>0.16ms</td>
<td>0.11ms</td>
<td>0.22ms</td>
<td>0.33ms</td>
</tr>
<tr>
<td>customers-service</td>
<td>1</td>
<td>0.11ms</td>
<td>0.11ms</td>
<td>0.11ms</td>
<td>0.11ms</td>
<td>0.11ms</td>
</tr>
<tr>
<td>visits-service</td>
<td>1</td>
<td>0.22ms</td>
<td>0.22ms</td>
<td>0.22ms</td>
<td>0.22ms</td>
<td>0.22ms</td>
</tr>
<tr>
<td>GET /api/gateway/owners/3</td>
<td>2</td>
<td>7.02ms</td>
<td>3.91ms</td>
<td>3.63ms</td>
<td>4.19ms</td>
<td>3.97ms</td>
</tr>
<tr>
<td>GET /api/gateway/owners/3</td>
<td>1</td>
<td>22.61ms</td>
<td>22.61ms</td>
<td>22.61ms</td>
<td>22.61ms</td>
<td>1.79ms</td>
</tr>
</tbody>
</table>
Distributed Tracing

Monitored Application

Customer Service

inspectIT Ocelot

Open Census

InfluxDB

Grafana

Jaeger Collector

Elasticsearch

Jaeger UI
Configuration Server & UI

```yaml
inspectIT:
  # the name of the service which is being instrumented
  service-name: "InspectIT Agent"

  # defines common tags to be be set on the metrics
  tags:
    # different tag providers that can be configured
    providers:
      # environment provider adds 'service-name', 'host' and 'host-address' tags
      environment:
        # if environment provider is enabled
        enabled: true
        # should the host name be resolved using InetAddress.getLocalHost(), if false
        resolve-host-name: true
        # should the host address be resolved using InetAddress.getLocalHost(), if false
        resolve-host-address: true

    # specifies user defined tag keys and values as a map
    # these tag values would overwrite any value added by the providers, thus you can
    extra: {}

  # general settings regarding trace capturing
  tracing:
    # master switch for trace capturing. When set to false the following happens:
    # - all trace exporters are disabled
    # - tracing is disabled for all instrumentation rules
    enabled: true

    # global sample probability used to decide if a trace shall be sampled or not
    # this value can be overridden by the tracing settings of individual instrumentation
    sample-probability: 1.0

  # general settings regarding metrics capturing
  metrics:
```
Config Server
Config Server

Monitored Application

- Customer Service
- inspectIT Ocelot
- inspectIT Ocelot

Config Server

Open Census

- InfluxDB
- Jaeger Collector
- Elasticsearch
- Jaeger UI

Grafana
LDAP Integration

Monitored Application

Customer Service

inspectIT Ocelot

Config Server

Open Census

InfluxDB

Grafana

LDAP Server

Jaeger Collector

Elasticsearch

Jaeger UI

Customer Service
Enduser Monitoring

Enduser Response Time Trend

Request Rate

Top Requests

<table>
<thead>
<tr>
<th>Page</th>
<th>Number of Requests</th>
<th>Load Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>/app/splash.jsp</td>
<td>63</td>
<td>ms</td>
</tr>
<tr>
<td>/app/screen62.jsp</td>
<td>46</td>
<td>ms</td>
</tr>
<tr>
<td>/app/screen65.jsp</td>
<td>13</td>
<td>ms</td>
</tr>
<tr>
<td>/app/screen67.jsp</td>
<td>3</td>
<td>ms</td>
</tr>
<tr>
<td>/app/screen69.jsp</td>
<td>3</td>
<td>ms</td>
</tr>
<tr>
<td>/app/screen80.jsp</td>
<td>3</td>
<td>ms</td>
</tr>
<tr>
<td>/app/screen90.jsp</td>
<td>2</td>
<td>ms</td>
</tr>
<tr>
<td>/app/screen91.jsp</td>
<td>1</td>
<td>ms</td>
</tr>
</tbody>
</table>
Enduser Monitoring

- **Monitored Application**
  - Web Frontend
  - Customer Service
  - Boomerang
  - inspectIT Ocelot
  - inspectIT Ocelot EUM

- **Config Server**
  - Open Census
  - LDAP Server

- **Monitoring Tools**
  - InfluxDB
  - Elasticsearch
  - Jaeger Collector
  - Jaeger UI
  - Grafana

---

**Related Technologies**

- Boomerang
- LDAP Server
- Open Census
- InfluxDB
- Elasticsearch
- Jaeger Collector
- Jaeger UI
- Grafana
Alerting & Reporting

<table>
<thead>
<tr>
<th>Rule Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status: enabled</td>
</tr>
<tr>
<td>Creation Date: 2020-09-03 15:09:40</td>
</tr>
<tr>
<td>Last Modification: 2020-09-03 15:09:09</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>no description</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>Notification Channel: selection</td>
</tr>
<tr>
<td>agent_connect_interval: duration</td>
</tr>
<tr>
<td>crit_count: int</td>
</tr>
<tr>
<td>custom_message: string</td>
</tr>
<tr>
<td>notification_time: string</td>
</tr>
<tr>
<td>service_regex: string</td>
</tr>
<tr>
<td>time_window: duration</td>
</tr>
</tbody>
</table>

- selfmonitoring
- 30s
- 10
- Task Name: {{TaskName}} services: {{Group}} crontext: {{Level}} dnr:BD: {{ID}}
- always
- `<!--% config_server.* -->`
- 2am

This page displays the alerting and reporting settings for a specific rule, including details about the status, creation, and last modification dates. The description field is empty. The rule includes variables such as notification channel, agent connect interval, critical count, custom message, notification time, service regex, and time window. These variables are crucial for configuring how alerts are sent and monitored.
When will you go open-source?

- Successful replacement of APM Leader
When will you go open-source?

- Successful replacement of APM Leader
When will you go open-source?

- Successful replacement of APM Leader
- Further customization is necessary
When will you go open-source?

- Successful replacement of APM Leader
- Further customization is necessary
- Enterprise-ready OpenAPM solution
When will you go open-source?

- Successful replacement of APM Leader
- Further customization is necessary
- Enterprise-ready OpenAPM solution
- High flexibility through high-degree of interoperability
When will you go open-source?

- Successful replacement of APM Leader
- Further customization is necessary
- Enterprise-ready OpenAPM solution
- High flexibility through high-degree of interoperability

We are happy to support academics; please reach out to us
Tobias Angerstein
tobias.angerstein@novatec-gmbh.de
@tobiangerstein

Henning Schulz
henning.schulz@novatec-gmbh.de

OpenAPM
https://openapm.io
@openapmio
info@openapm.io
https://github.com/openapm

inspectIT Ocelot
https://inspectit.rocks
@inspectIT_APM
info.inspectit@novatec-gmbh.de
https://github.com/inspectit