MoSKito

for users.

by Leon Rosenberg
BEDCON 2013
@dvayanu

anotheria solutions

Why MoSKito

anotheria solutions

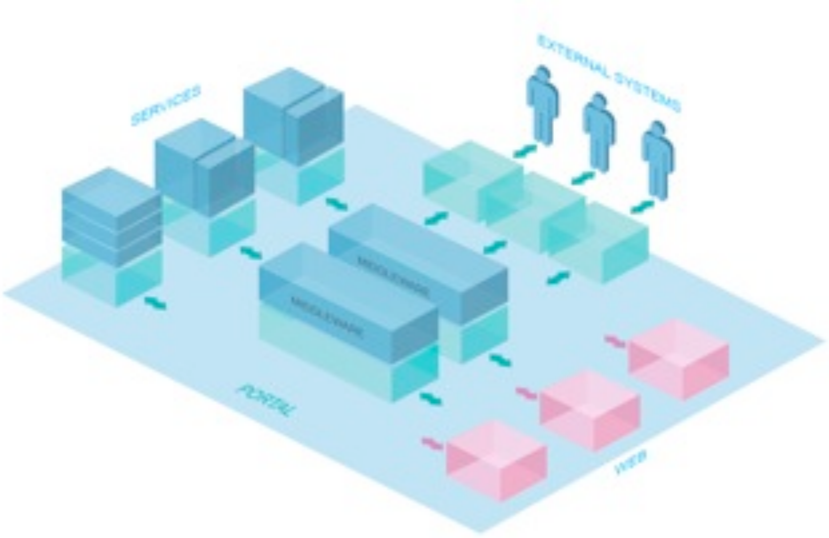
new features

BUGs

architectural changes

new markets

product pivots



Seasonal traffic

Service Level

Availability

Problem finding and fixing

Marketing campaigns

Disaster recovery

RAM/CPU Requirements

Hosting needs

Cost per user

Provider change

anotheria solutions

new features

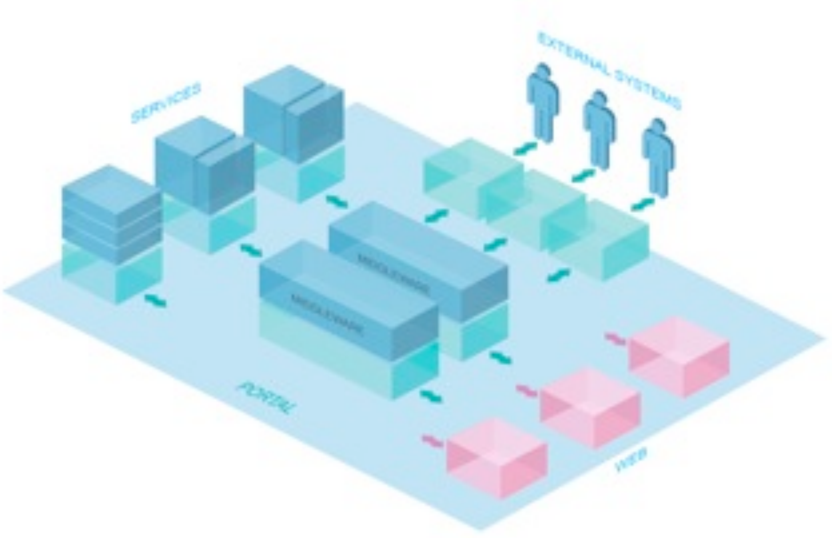
BUGs

architectural changes

Change Monitoring

new markets

product pivots



Seasonal traffic

Service Level

SLA Monitoring

Availability

Problem finding and fixing

Marketing campaigns

Disaster recovery

RAM/CPU Requirements

Capacity Monitoring

Hosting needs

Cost per user

Provider change

new features

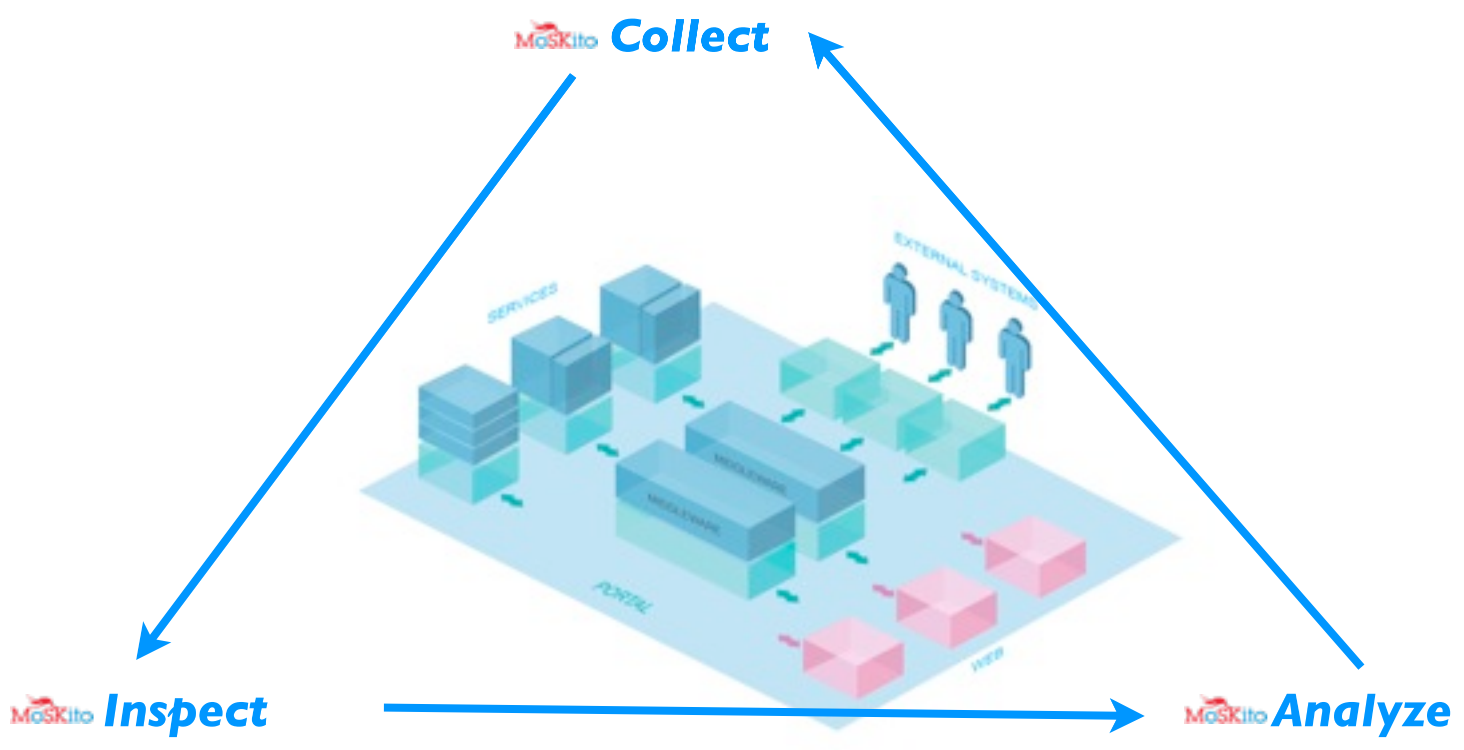
BUGs

architectural changes

Change Monitoring

new markets

product pivots



Seasonal traffic

Disaster recovery

Service Level

SLA Monitoring

Availability

Problem finding and fixing

Marketing campaigns

RAM/CPU Requirements

Capacity Monitoring

Hosting needs

Cost per user

Provider change

What is MoSKito

- MoSKito is a **multi-purpose**, **non-invasive**, **interval based monitoring system kit** for **collection**, **storage** and **instant analysis** of application’s performance and behavior data.

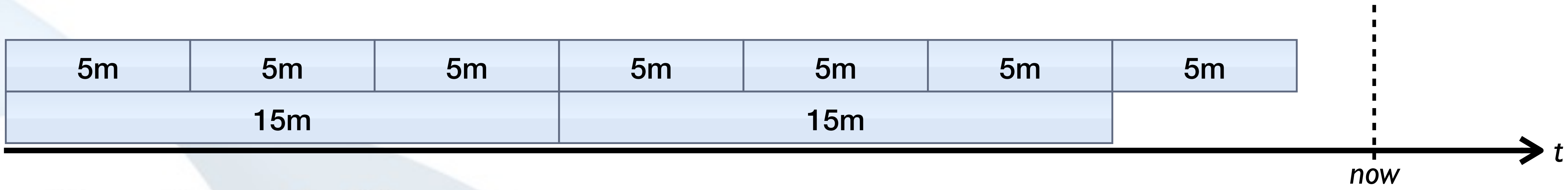
anotheria solutions

Key Features

- Collect and Store.
- Inspect and Monitor.
- Analyze and Alert.
- Continuos production profiling without performance impacts with Journeys.

Interval based.

- The behavior of a system depends on hour, weekday, weather, holidays and good karma.
- Large amounts of collected data make monitoring nonsensitive to anomalies.
- Inspection of short intervals offers more understanding about system’s behavior.



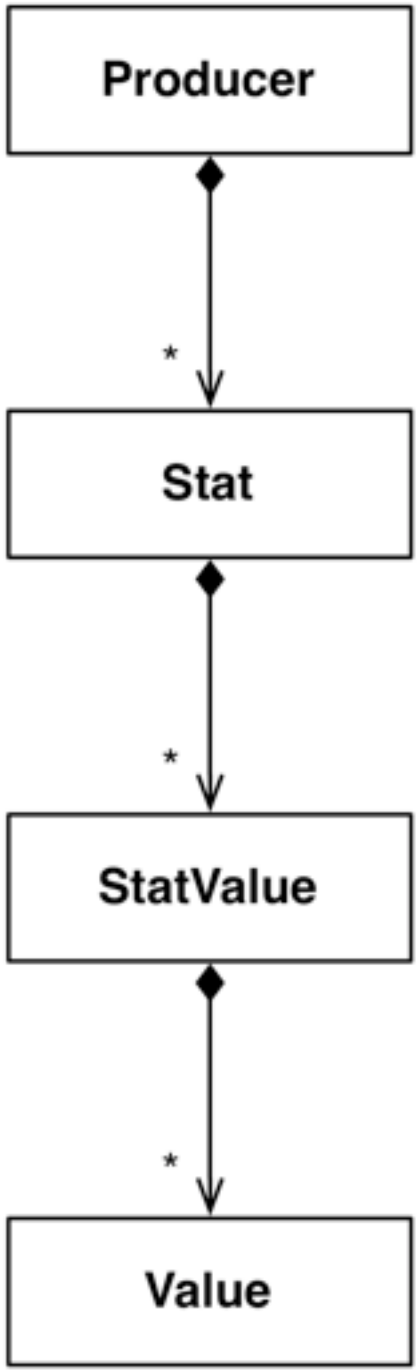
5m 5m 5m 5m 5m 5m 5m

15m 15m

now

t

Core Concepts



Do something measureable, produce stats.
Service, Filter, Action, Resource, Gateway, Payment Provider.

Statistic of a use case, i.e. method name, url,
cumulated producer statistics

Value type, i.e. request count, avg duration, error count,
cache hits, payments etc.

Container for different values for intervals

anotheria solutions

Core Sections

- Producers and Stats - gather monitoring data.
- Thresholds - monitor changes in critical sections of the application.
- Accumulators - builds trends and allow visual analysis.
- Journeys - make inner life of the application visible.

- A picture is worth 1000 words...

  ... and live presentation is worth 1000 pictures.

Integration

- AOP / CDI
- Proxies
- WEB

- Guide: https://confluence.opensource.anotheria.net/display/MSK/Integration+Guide

AOP

```
@Monitor
public class YourClass {
```

```
public class YourClass {
  @Monitor public void firstMonitoredMethod(){...
  @Monitor public void secondMonitoredMethod(){...
  public void notMonitoredMethod(){...
```

```
@Monitor
public class YourClass {
  public void thisMethodWillBeMonitored(){...

  @DontMonitor public void thisMethodWillBeExcludedFromMonitoring(){
```

```
@Count
public class PaymentCounter {
```

```
@Count
public class PaymentCounter {
    /**
     * Electronic card payment (lastchrifteinzug in germany).
     */
    public void ec(){}
    /**
     * Credit card payment.
     */
    public void cc(){}
    /**
     * Payment via paypal.
     */
    public void paypal(){}
}
```

AOP + MAVEN

```
<dependencies>
    <dependency>
        <groupId>net.anotheria</groupId>
        <artifactId>moskito-core</artifactId>
        <version>2.2.2</version>
    </dependency>
    <dependency>
        <groupId>net.anotheria</groupId>
        <artifactId>moskito-aop</artifactId>
        <version>2.2.2</version>
 </dependency>
```

```
<build>
    <plugins>
        <plugin>
            <groupId>org.codehaus.mojo</groupId>
            <artifactId>aspectj-maven-plugin</artifactId>
            <version>1.4</version>
            <configuration>
                <aspectLibraries>
                    <aspectLibrary>
                        <groupId>net.anotheria</groupId>
                        <artifactId>moskito-aop</artifactId>
                    </aspectLibrary>
                </aspectLibraries>
                <source>1.6</source>
                <target>1.6</target>
            </configuration>
            <executions>
                <execution>
                    <goals>
                        <goal>compile</goal>
                    </goals>
                </execution>
            </executions>
        </plugin>
    </plugins>
</build>
```

Proxies

```java
public interface SimpleService{
    void doSomethingMethod();
}
```

```java
public class SimpleServiceImpl implements SimpleService{
    public void doSomethingMethod(){
    }
}
```

```java
SimpleService service = ProxyUtils.createServiceInstance(new SimpleServiceImpl(), "default", SimpleService.class);
```

```java
SimpleService unmonitoredInstance = new SimpleServiceImpl();
MoskitoInvokationProxy proxy = new MoskitoInvokationProxy(
        unmonitoredInstance,
        new ServiceStatsCallHandler(),
        new ServiceStatsFactory(),
        "SimpleService",
        "service",
        "test-sub-system",
        SimpleService.class
        );
SimpleService monitoredInstance = (SimpleService)proxy.createProxy();
```

WEB

```
<filter>
  <filter-name>RequestURIFilter</filter-name>
  <filter-class>net.anotheria.moskito.web.filters.RequestURIFilter</filter-class>
  <init-param>
    <param-name>limit</param-name>
    <param-value>1000</param-value>
  </init-param>
</filter>

<filter-mapping>
  <filter-name>RequestURIFilter</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>
```

```
<filter>
  <filter-name>DomainFilter</filter-name>
  <filter-class>net.anotheria.moskito.web.filters.DomainFilter</filter-class>
  <init-param>
    <param-name>limit</param-name>
    <param-value>50</param-value>
  </init-param>
</filter>

<filter-mapping>
  <filter-name>DomainFilter</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>
```

WebUI

```
web.xml
   <!--  MOSKITO UI-->
    <!--  Adding filter to moskito ui which redirects requests to /mui/* to moskito user interface -->
    <filter>
        <filter-name>MoskitoUIFilter</filter-name>
        <filter-class>net.anotheria.moskito.webui.MoskitoUIFilter</filter-class>
        <init-param>
            <param-name>path</param-name>
            <param-value>/mui/</param-value>
        </init-param>
    </filter>
    <filter-mapping>
        <filter-name>MoskitoUIFilter</filter-name>
        <url-pattern>/mui/*</url-pattern>
    </filter-mapping>
    <!--  / MOSKITO UI END -->

<!-- somewhere else --->

    <listener>
        <listener-class>
            net.anotheria.moskito.webui.util.StartStopListener
        </listener-class>
    </listener>
    <listener>
        <listener-class>
            net.anotheria.moskito.web.session.SessionCountProducer
        </listener-class>
    </listener>
    <listener>
        <listener-class>
            org.anotheria.moskitodemo.threshold.presentation.listener.SetupThresholds
        </listener-class>
    </listener>
```

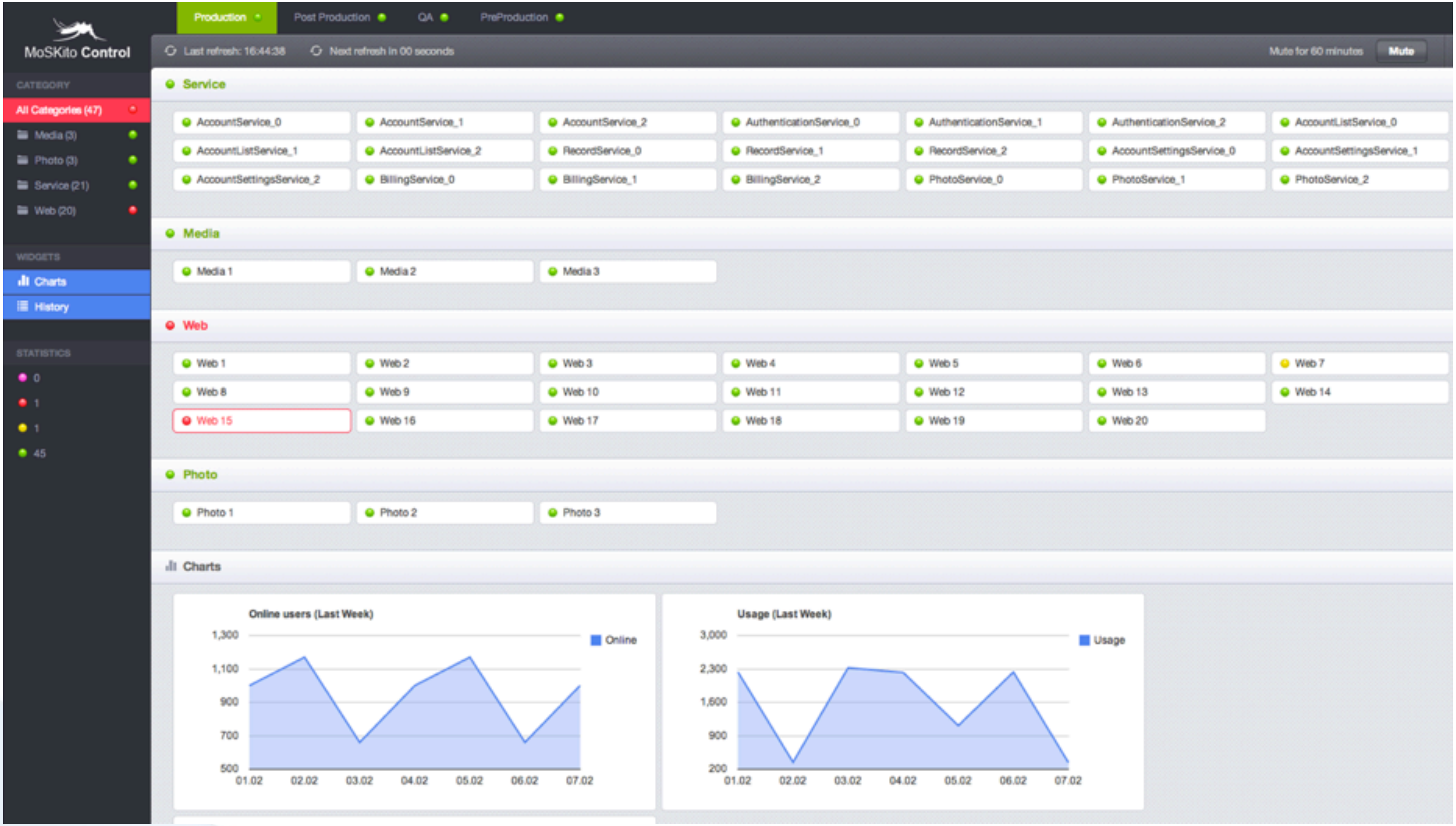
What else

- MoSKito Central
- MoSKito Control
- MoSKito @ Barbecue

MoSKito Central

- Central server for snapshot storage.
- Attachable storages can store data into sql- or nosql- based databases, xml/json files etc.
- Runs in remote or embedded mode.

MoSKito Control



● Service

● AccountService_0	● AccountService_1	● AccountService_2	● AuthenticationService_0	● AuthenticationService_1	● AuthenticationService_2	● AccountListService_0
● AccountListService_1	● AccountListService_2	● RecordService_0	● RecordService_1	● RecordService_2	● AccountSettingsService_0	● AccountSettingsService_1
● AccountSettingsService_2	● BillingService_0	● BillingService_1	● BillingService_2	● PhotoService_0	● PhotoService_1	● PhotoService_2

● Media

● Media 1	● Media 2	● Media 3

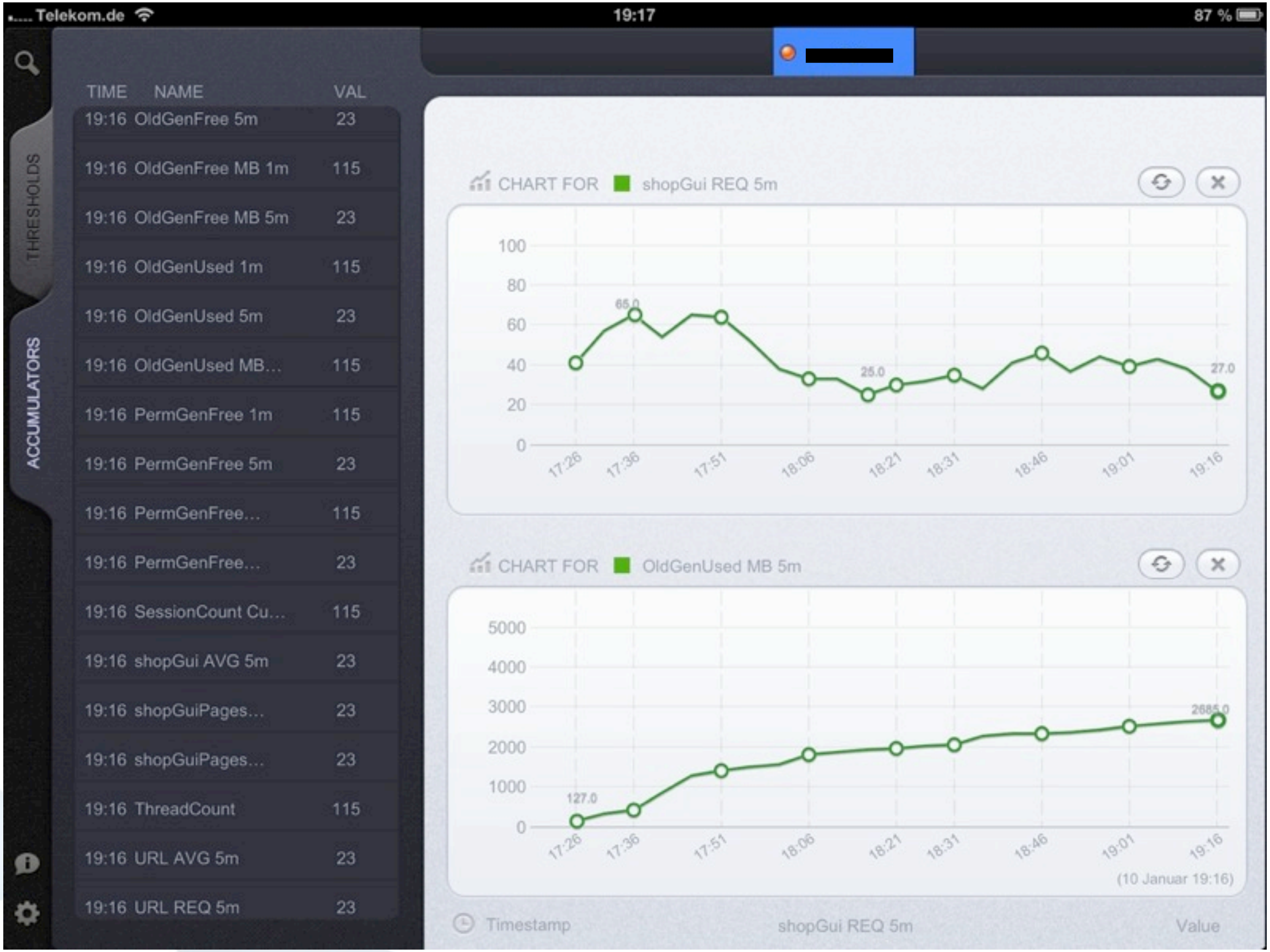
● Web

● Web 1	● Web 2	● Web 3	● Web 4	● Web 5	● Web 6	● Web 7
● Web 8	● Web 9	● Web 10	● Web 11	● Web 12	● Web 13	● Web 14
● Web 15	● Web 16	● Web 17	● Web 18	● Web 19	● Web 20	

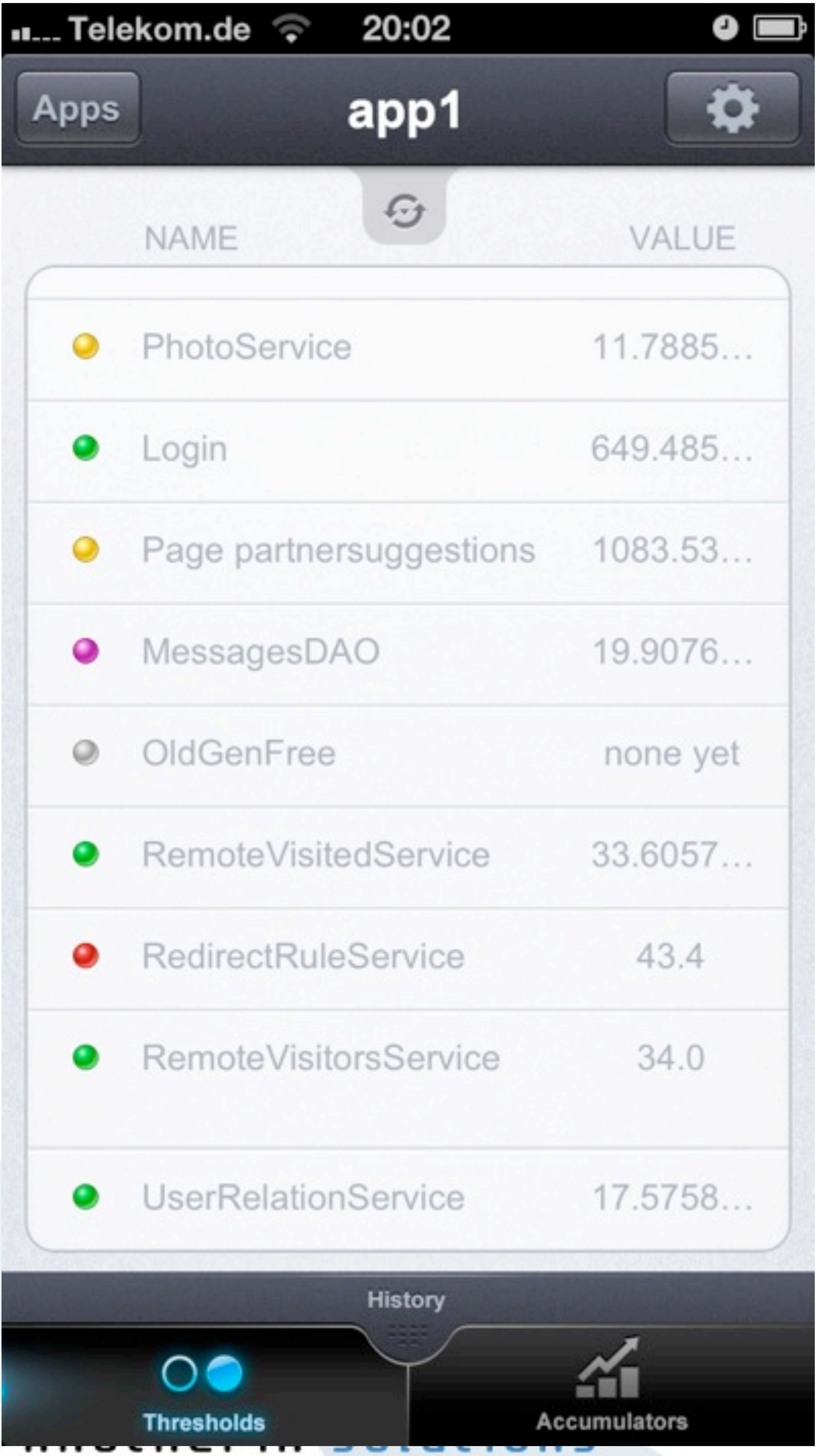
● Photo

● Photo 1	● Photo 2	● Photo 3

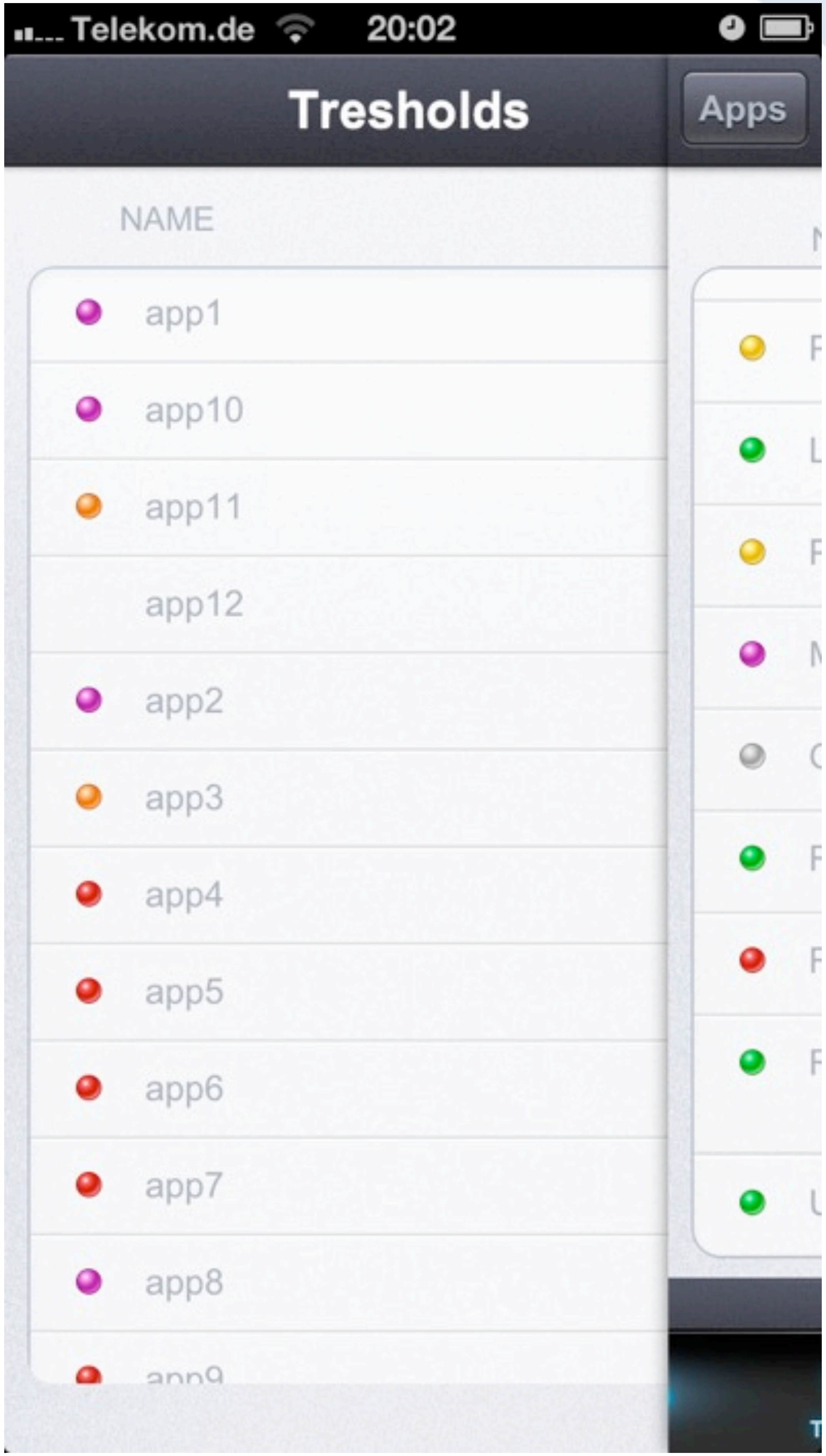
MoSKito To Go



TIME	NAME	VAL
19:16	OldGenFree 5m	23
19:16	OldGenFree MB 1m	115
19:16	OldGenFree MB 5m	23
19:16	OldGenUsed 1m	115
19:16	OldGenUsed 5m	23
19:16	OldGenUsed MB...	115
19:16	PermGenFree 1m	115
19:16	PermGenFree 5m	23
19:16	PermGenFree...	115
19:16	PermGenFree...	23
19:16	SessionCount Cu...	115
19:16	shopGui AVG 5m	23
19:16	shopGuiPages...	23
19:16	shopGuiPages...	23
19:16	ThreadCount	115
19:16	URL AVG 5m	23
19:16	URL REQ 5m	23



	NAME	VALUE
🟡	PhotoService	11.7885...
🟢	Login	649.485...
🟡	Page partnersuggestions	1083.53...
🟣	MessagesDAO	19.9076...
⚪	OldGenFree	none yet
🟢	RemoteVisitedService	33.6057...
🔴	RedirectRuleService	43.4
🟢	RemoteVisitorsService	34.0
🟢	UserRelationService	17.5758...



	NAME
🟣	app1
🟣	app10
🟠	app11
	app12
🟣	app2
🟠	app3
🔴	app4
🔴	app5
🔴	app6
🔴	app7
🟣	app8
🔴	app9

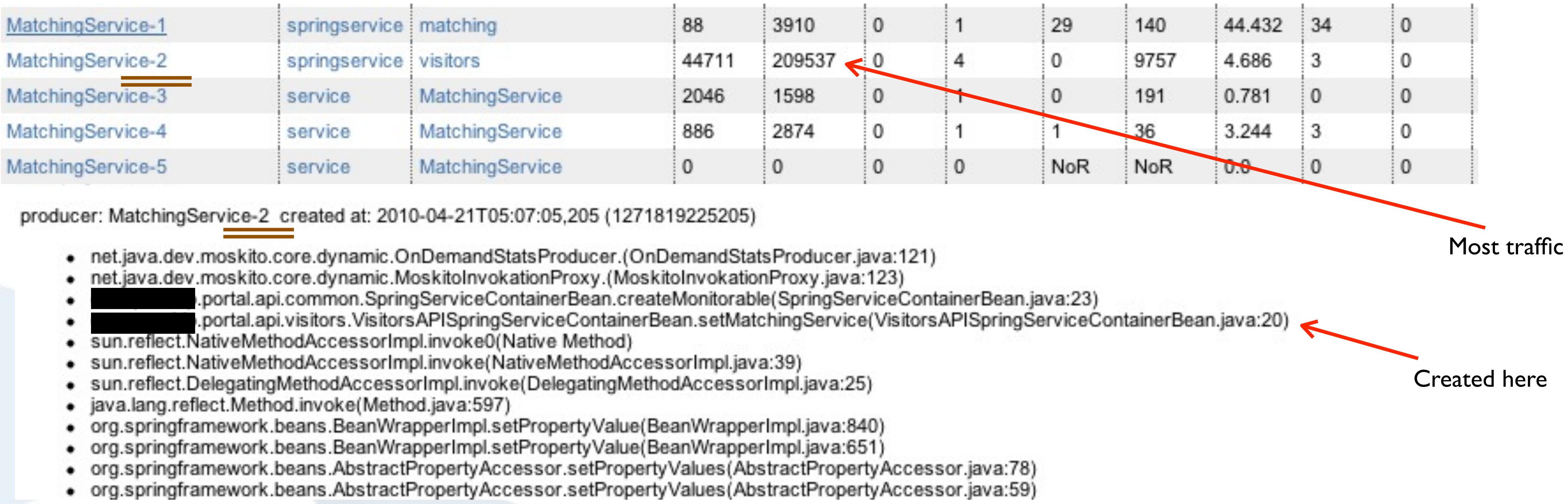
Success stories

- Far too many, but here are some :-)

I

- After a release of a new version huge traffic increase on one of the databases was detected.
- The database in question was used by a service. There were 20 clients (code components) using this service.
- MoSKito showed that 55% of the traffic to the service came from one client. With MoSKito inspection we were able to detect which client was producing most traffic.

- Closer inspection (code review) of the client revealed a bug which led to double calls to the service.
- Incident solved in 30 minutes.



MatchingService-1	springservice	matching	88	3910	0	1	29	140	44.432	34	0
MatchingService-2	springservice	visitors	44711	209537	0	4	0	9757	4.686	3	0
MatchingService-3	service	MatchingService	2046	1598	0	1	0	191	0.781	0	0
MatchingService-4	service	MatchingService	886	2874	0	1	1	36	3.244	3	0
MatchingService-5	service	MatchingService	0	0	0	0	NoR	NoR	0.0	0	0

producer: MatchingService-2 created at: 2010-04-21T05:07:05,205 (1271819225205)

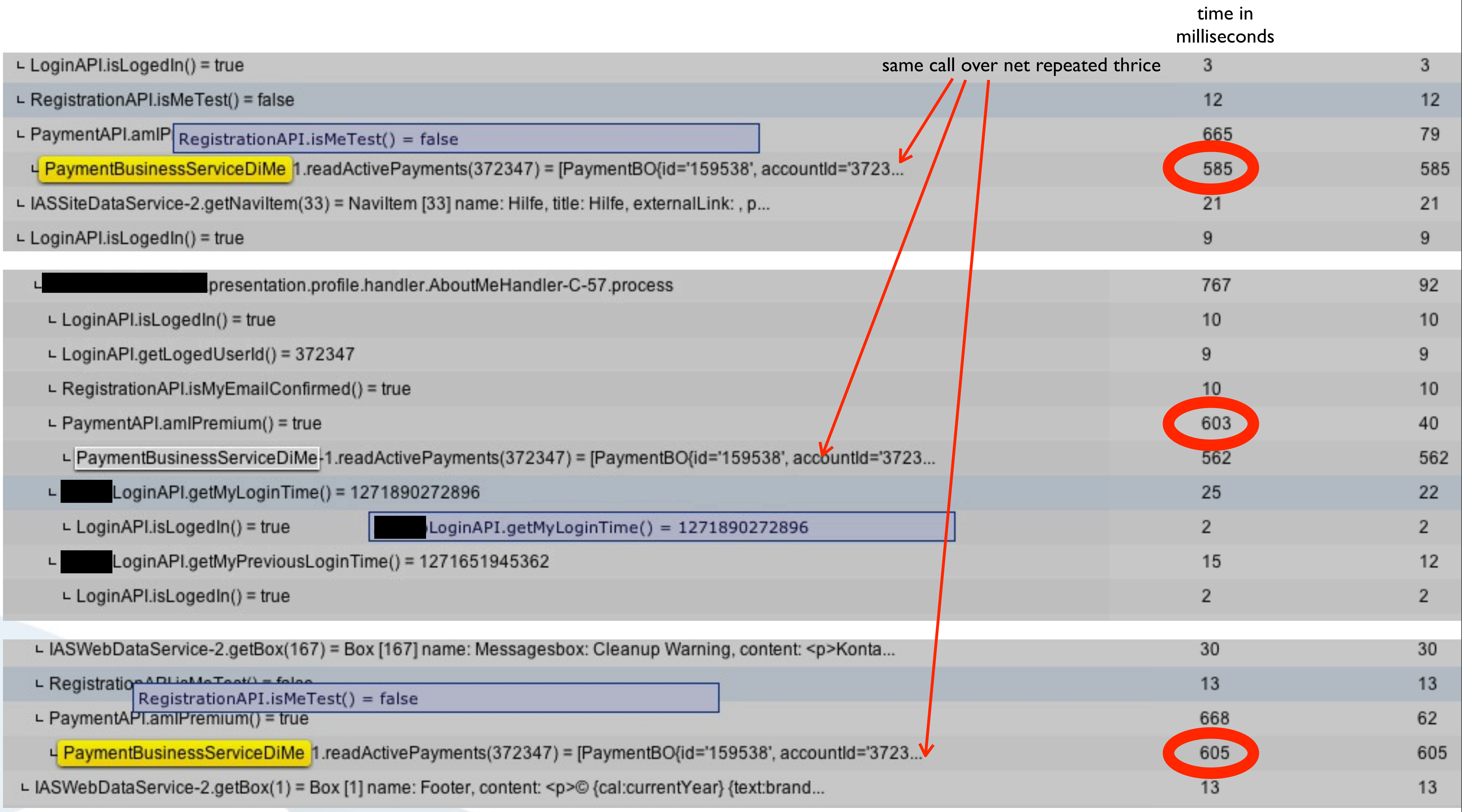
```
• net.java.dev.moskito.core.dynamic.OnDemandStatsProducer.(OnDemandStatsProducer.java:121)
• net.java.dev.moskito.core.dynamic.MoskitoInvokationProxy.(MoskitoInvokationProxy.java:123)
• [redacted].portal.api.common.SpringServiceContainerBean.createMonitorable(SpringServiceContainerBean.java:23)
• [redacted].portal.api.visitors.VisitorsAPISpringServiceContainerBean.setMatchingService(VisitorsAPISpringServiceContainerBean.java:20)
• sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
• sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39)
• sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:25)
• java.lang.reflect.Method.invoke(Method.java:597)
• org.springframework.beans.BeanWrapperImpl.setPropertyValue(BeanWrapperImpl.java:840)
• org.springframework.beans.BeanWrapperImpl.setPropertyValue(BeanWrapperImpl.java:651)
• org.springframework.beans.AbstractPropertyAccessor.setPropertyValues(AbstractPropertyAccessor.java:78)
• org.springframework.beans.AbstractPropertyAccessor.setPropertyValues(AbstractPropertyAccessor.java:59)
```

Most traffic

Created here

II

- Application overall performance was insufficient.
- With moskito journeys and call tree analysis we were able to find redundant calls to the backend and remove them.
- Request duration reduced to 50% with 4 hours analysis and 4 hours coding effort.



time in milliseconds

same call over net repeated thrice

```
└ LoginAPI.isLogedIn() = true                                                                   3      3
└ RegistrationAPI.isMeTest() = false                                                           12     12
└ PaymentAPI.amIP  RegistrationAPI.isMeTest() = false                                         665     79
  └ PaymentBusinessServiceDiMe1.readActivePayments(372347) = [PaymentBO{id='159538', accountId='3723...   585    585
└ IASSiteDataService-2.getNaviItem(33) = NaviItem [33] name: Hilfe, title: Hilfe, externalLink: , p...   21     21
└ LoginAPI.isLogedIn() = true                                                                   9      9

  └ [redacted]presentation.profile.handler.AboutMeHandler-C-57.process                        767     92
    └ LoginAPI.isLogedIn() = true                                                              10     10
    └ LoginAPI.getLogedUserId() = 372347                                                        9      9
    └ RegistrationAPI.isMyEmailConfirmed() = true                                              10     10
    └ PaymentAPI.amIPremium() = true                                                          603     40
      └ PaymentBusinessServiceDiMe-1.readActivePayments(372347) = [PaymentBO{id='159538', accountId='3723...   562    562
    └ [redacted]LoginAPI.getMyLoginTime() = 1271890272896                                      25     22
      └ LoginAPI.isLogedIn() = true          [redacted]LoginAPI.getMyLoginTime() = 1271890272896   2      2
    └ [redacted]LoginAPI.getMyPreviousLoginTime() = 1271651945362                              15     12
      └ LoginAPI.isLogedIn() = true                                                             2      2

  └ IASWebDataService-2.getBox(167) = Box [167] name: Messagesbox: Cleanup Warning, content: <p>Konta...   30     30
  └ Registratio  RegistrationAPI.isMeTest() = false                                            13     13
  └ PaymentAPI.amIPremium() = true                                                            668     62
    └ PaymentBusinessServiceDiMe1.readActivePayments(372347) = [PaymentBO{id='159538', accountId='3723...   605    605
└ IASWebDataService-2.getBox(1) = Box [1] name: Footer, content: <p>© {cal:currentYear} {text:brand...   13     13
```

End of Data.

Thank you.

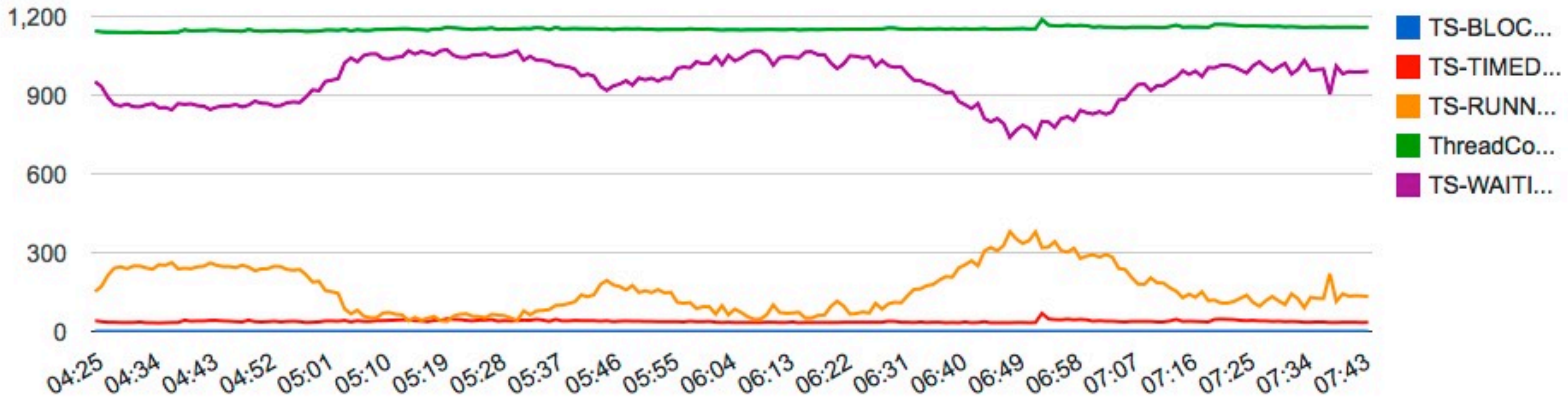
MoSKito http://moskito.anotheria.net/

Available on the App Store https://itunes.apple.com/de/app/moskito-ui/id531387262?l=en&mt=8

https://github.com/anotheria/moskito-examples

https://github.com/anotheria/moskito-jboss

Central http://search.maven.org/#search%7Cga%7C1%7Cmoskito


1,200
900
600
300
0
04:25
04:34
04:43
04:52
05:01
05:10
05:19
05:28
05:37
05:46
05:55
06:04
06:13
06:22
06:31
06:40
06:49
06:58
07:07
07:16
07:25
07:34
07:43
🟦 TS-BLOC...
🟥 TS-TIMED...
🟧 TS-RUNN...
🟩 ThreadCo...
🟪 TS-WAITI...

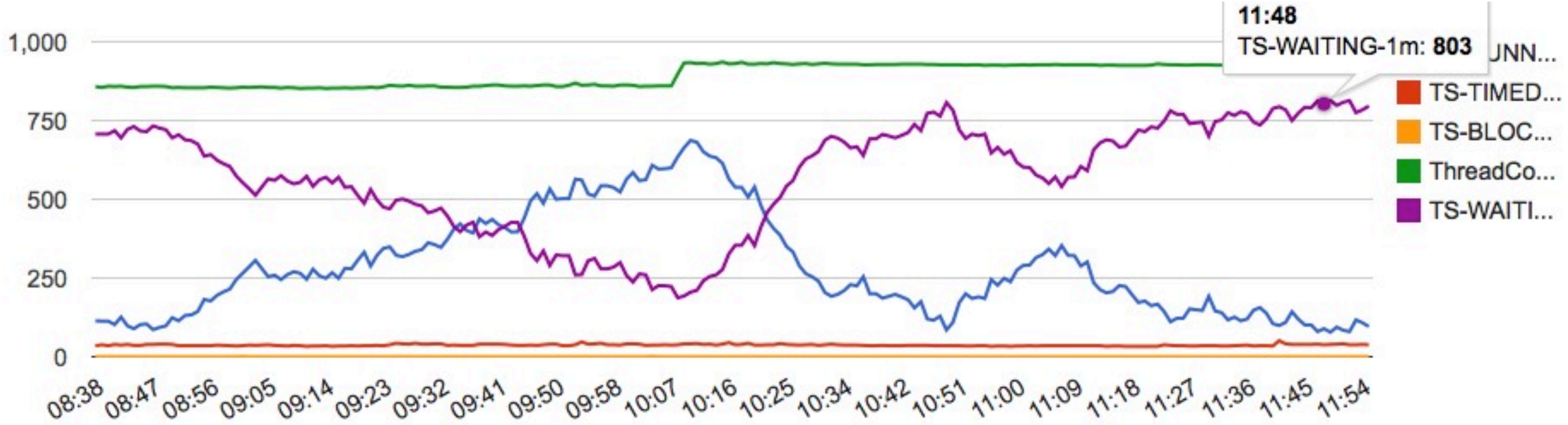

1,000
750
500
250
0
08:38
08:47
08:56
09:05
09:14
09:23
09:32
09:41
09:50
09:58
10:07
10:16
10:25
10:34
10:42
10:51
11:00
11:09
11:18
11:27
11:36
11:45
11:54
11:48
TS-WAITING-1m: 803
UNN...
🟥 TS-TIMED...
🟧 TS-BLOC...
🟩 ThreadCo...
🟪 TS-WAITI...

Chart for PhotoService REQ PhotoService AVG

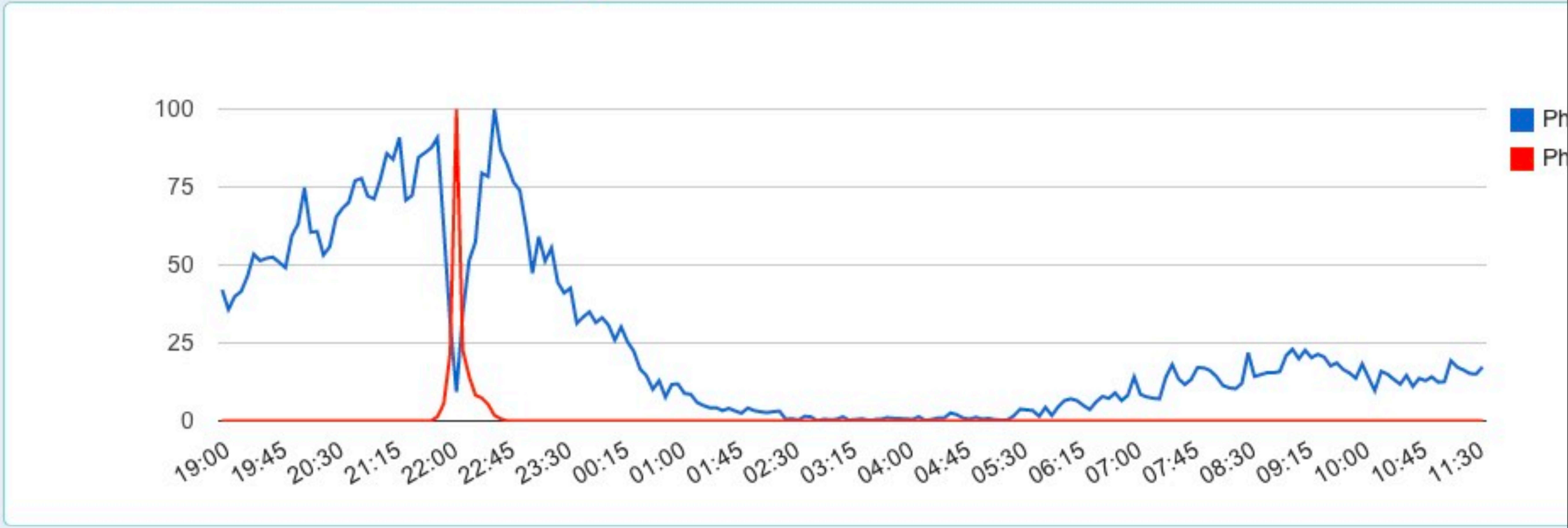
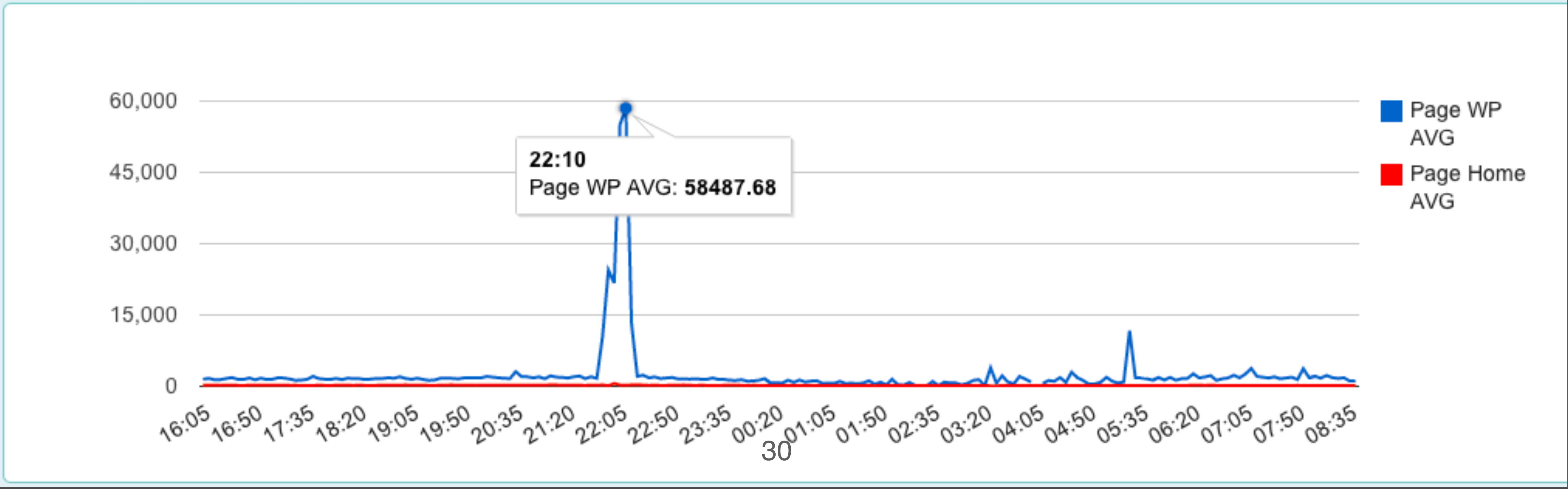


Chart for Page WP AVG Page Home AVG



22:10
Page WP AVG: **58487.68**

End of Data.

Thank you.

MoSKito http://moskito.anotheria.net/

Available on the App Store https://itunes.apple.com/de/app/moskito-ui/id531387262?l=en&mt=8

https://github.com/anotheria/moskito-examples

https://github.com/anotheria/moskito-jboss

Central http://search.maven.org/#search%7Cga%7C1%7Cmoskito

anotheria solutions