



Monitoring tomcat web application in production

Leon Rosenberg (@dvayanu)



# Today's Contents

- Basic Concepts of MoSKito
- Burgershop Demo
- Recap
- Integration (if time allows)
- Q & A

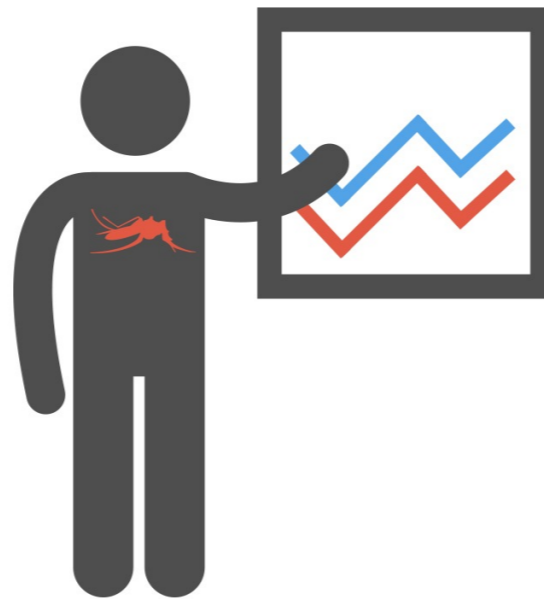
# Why MoSKito?



# MoSKito Core Team



Producer



Accumulator

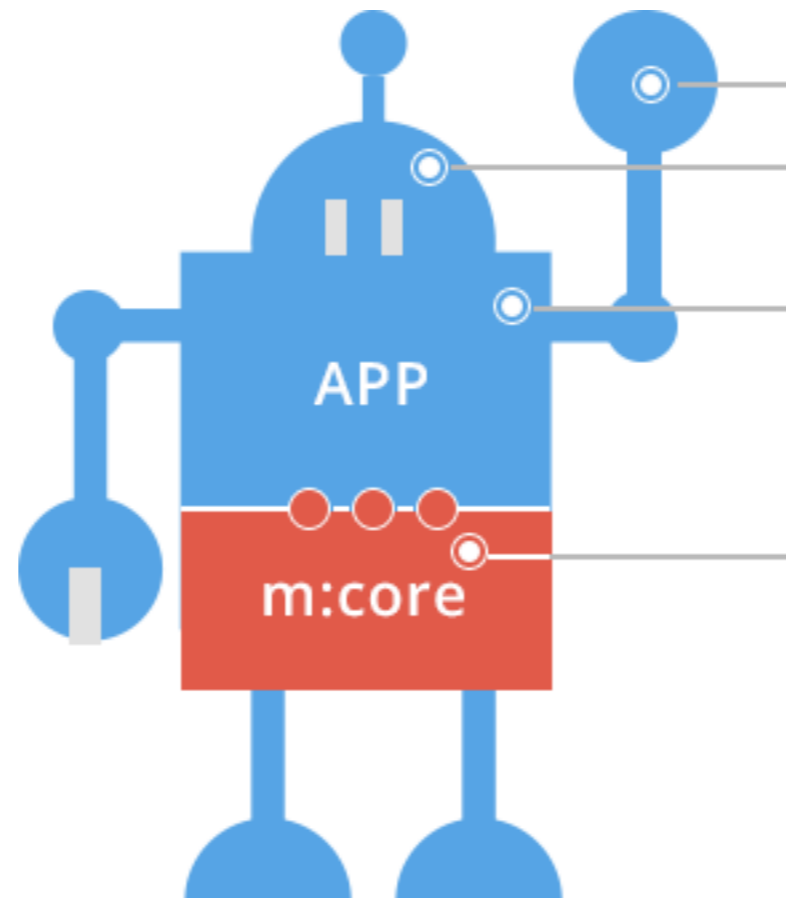


Threshold

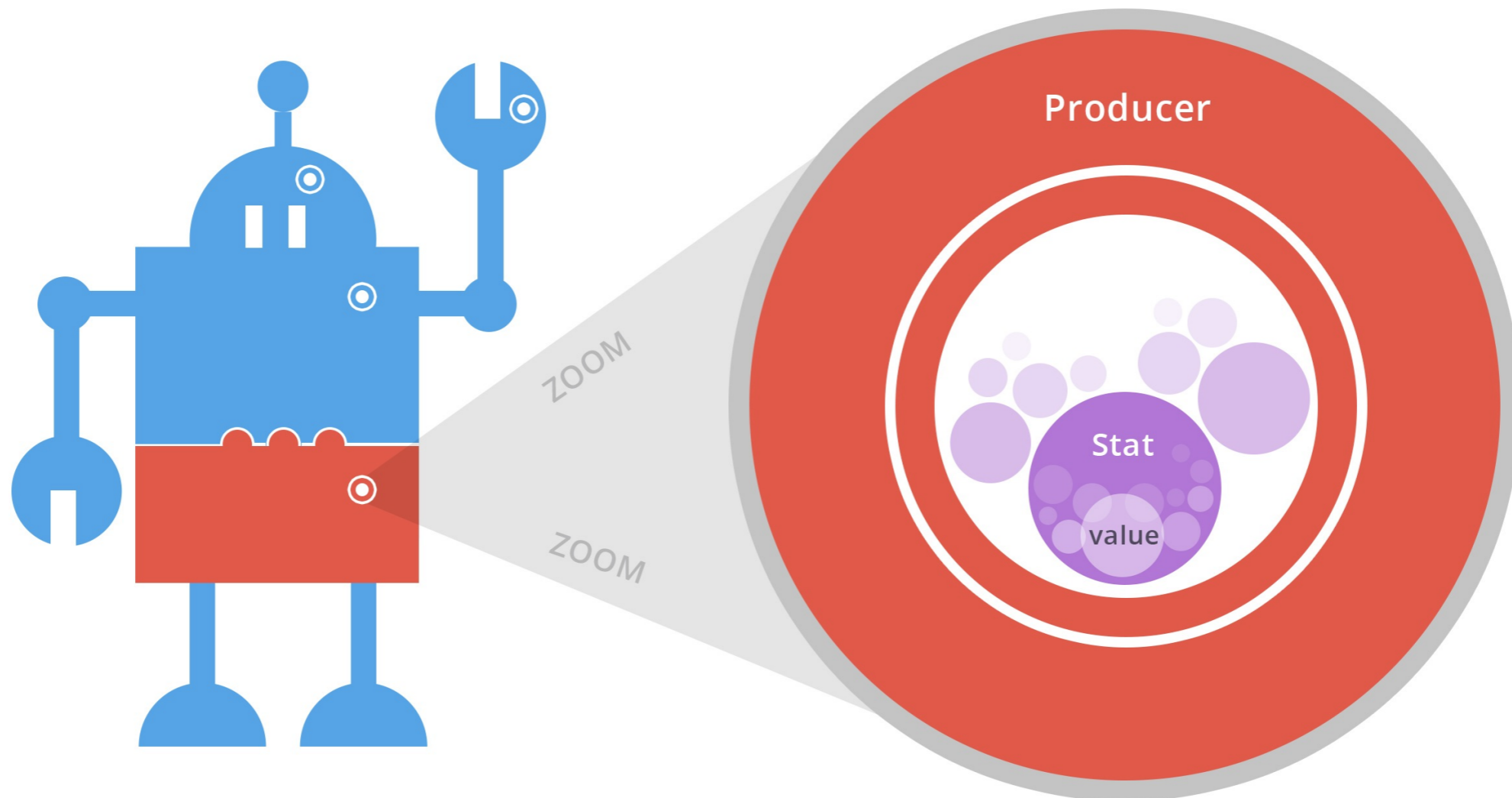


Journey

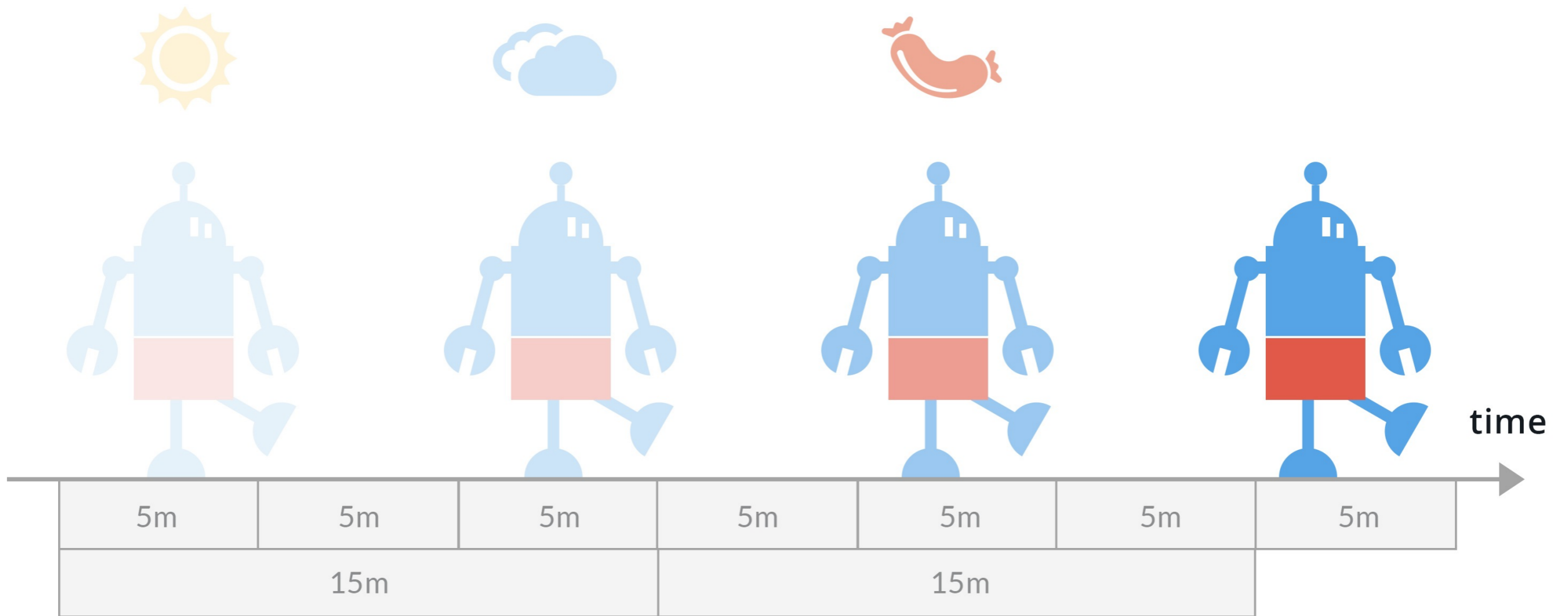
# Collect data



# Organise data

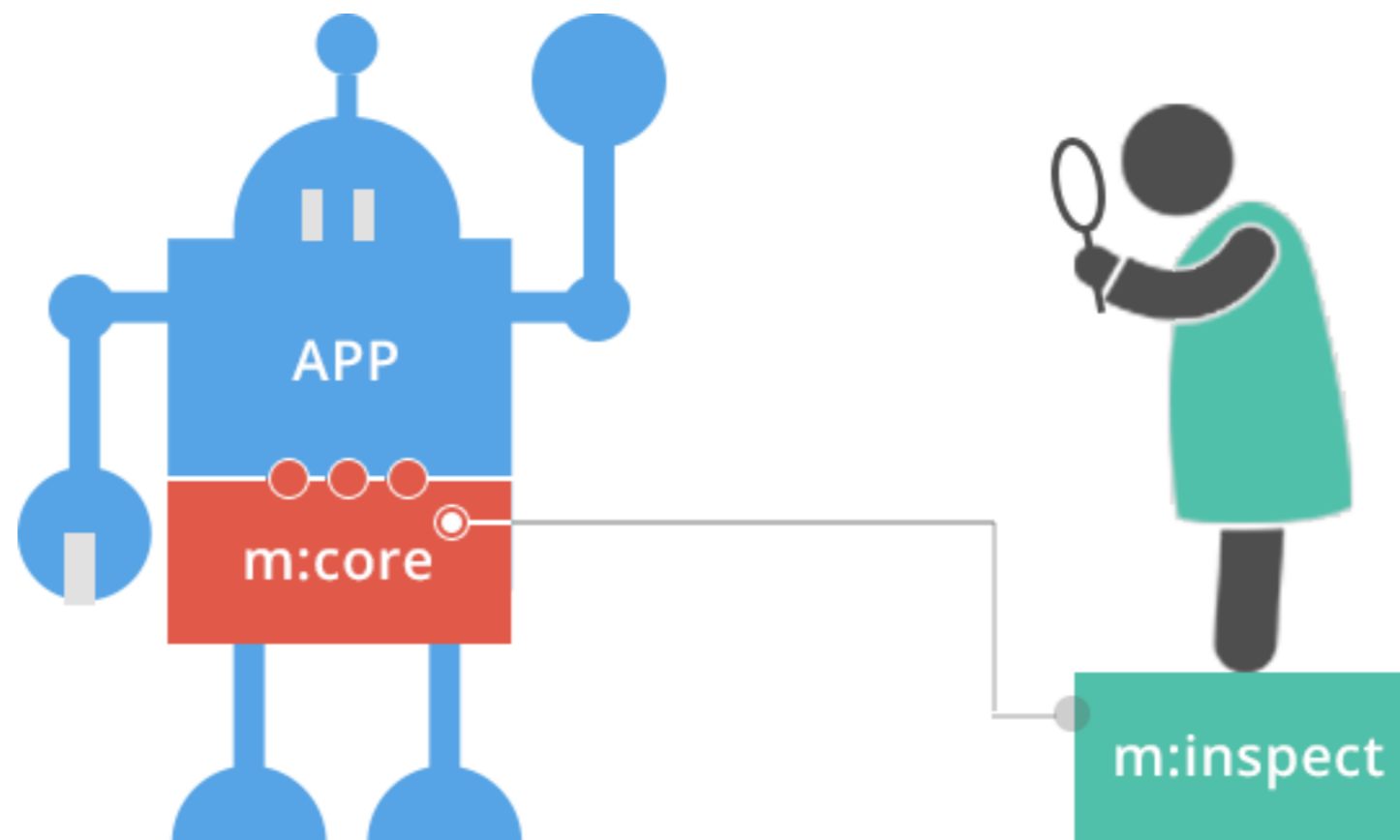


# Inspect Intervals

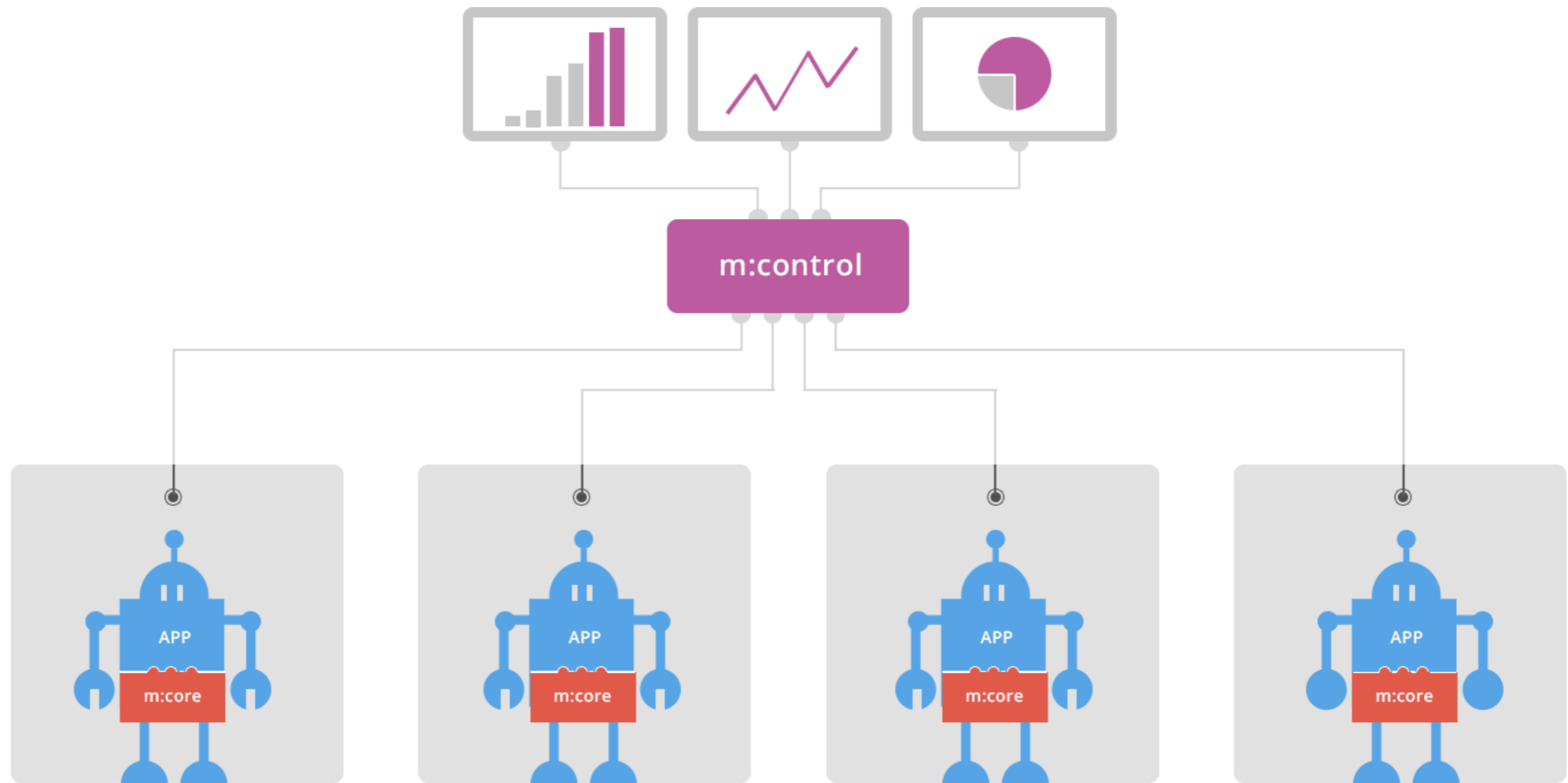




# Discover and analyse



# Scale and get in control



# DEMO TIME

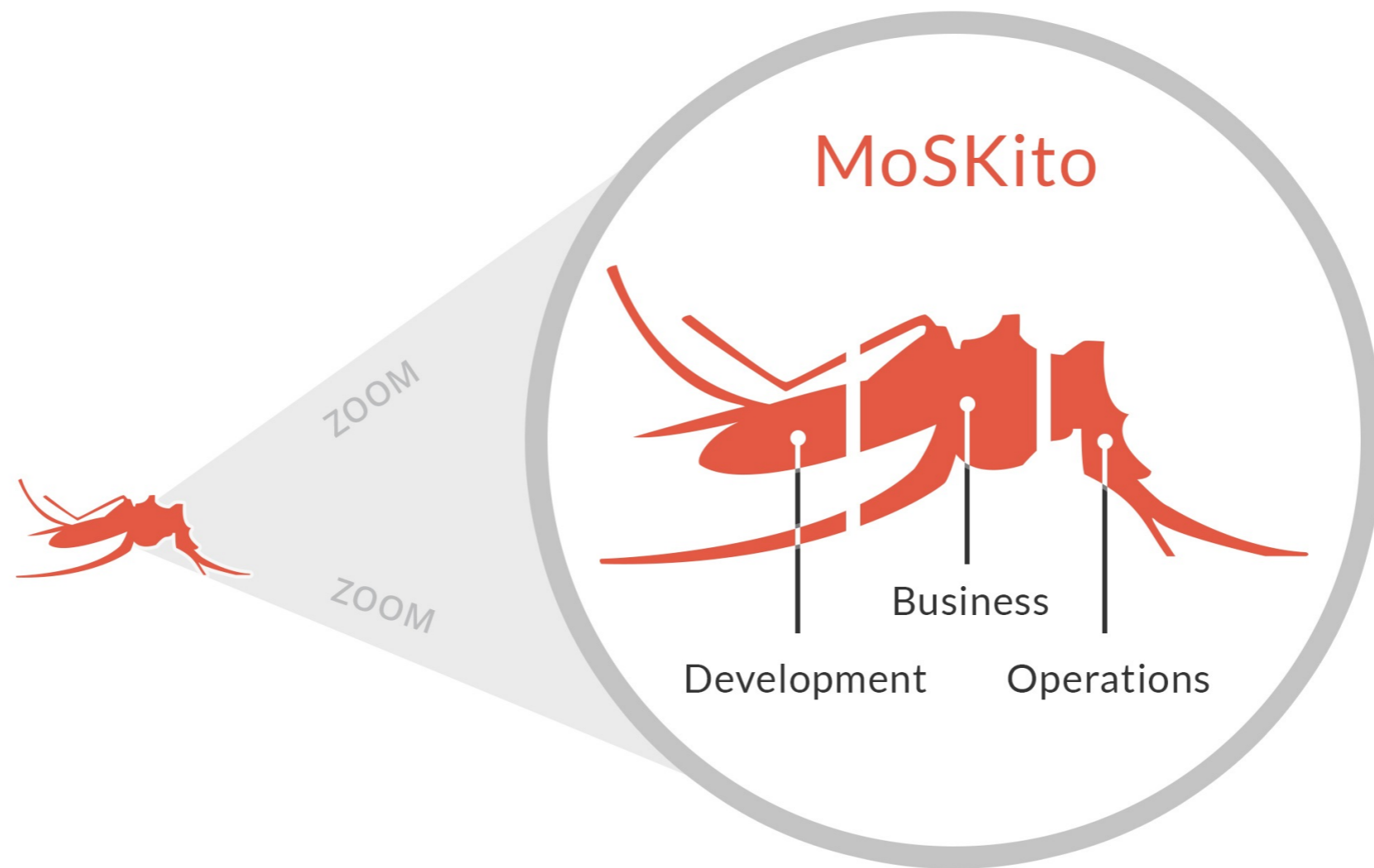
<http://burgershop-hamburg.demo.moskito.org/burgershop/>

<http://burgershop-hamburg.demo.moskito.org/burgershop/moskito-inspect/mskDashboard?dashboard=Burger+Board>

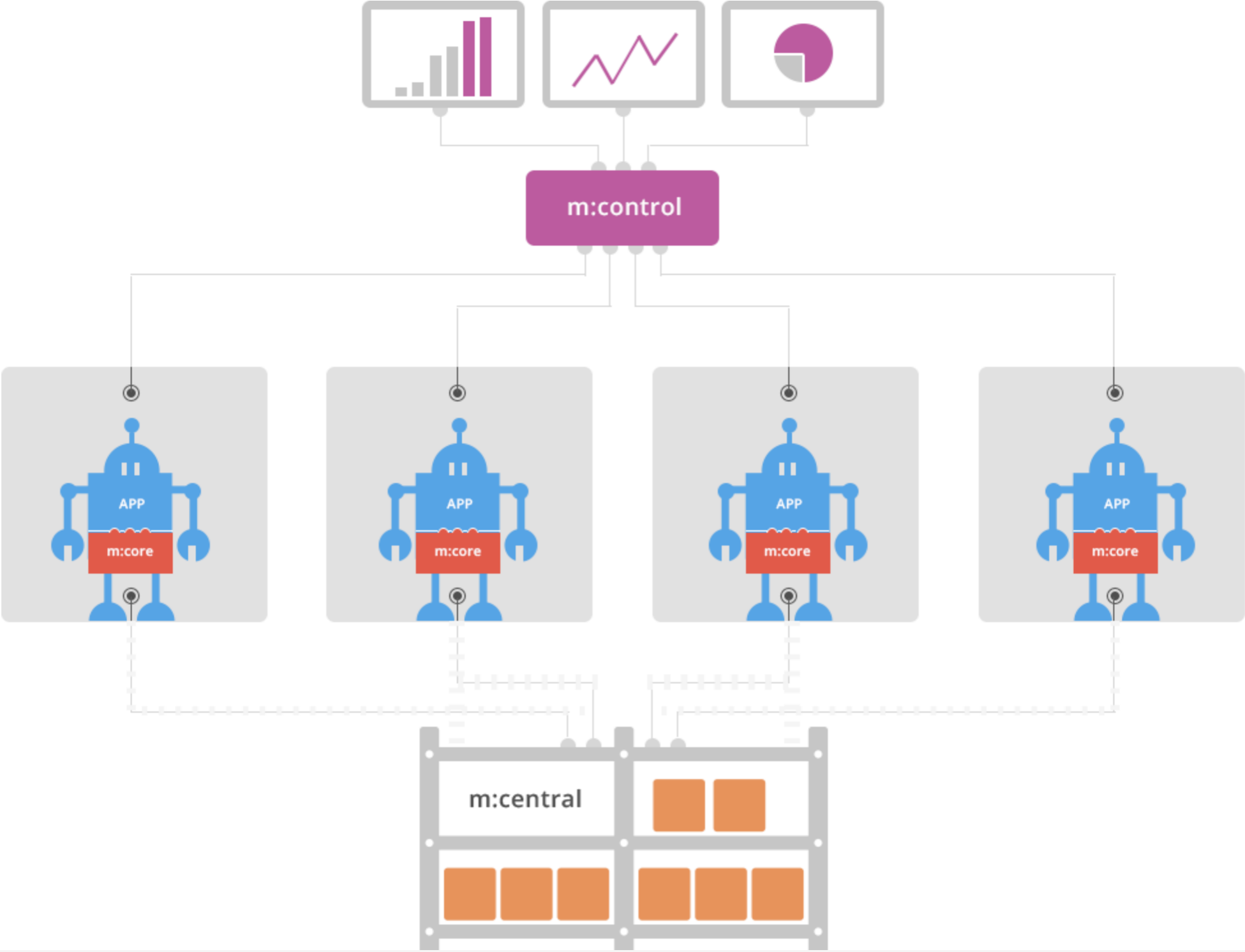
# Why MoSKito ?

- Performance data collection
- Application Insights with MoSKito Inspect
- Code and Dataflow Tracing with Journeys
- Error Management
- System overview with MoSKito Control

# Why MoSKito ?



# The big picture



# Q & A Time?



<http://www.moskito.org>



<https://itunes.apple.com/de/app/moskito-ui/id531387262?l=en&mt=8>

<https://itunes.apple.com/de/app/msk-control/id688838411?l=en&mt=8>



<https://github.com/anotheria/moskito>

<https://github.com/anotheria/moskito-control>

<https://github.com/anotheria/moskito-demo>



<http://search.maven.org/#search%7Cga%7CI%7Cmoskito>

# Integration

- ◉ AOP / CDI / Spring
- ◉ Proxies
- ◉ WEB
- ◉ Java Agent
- ◉ Guide: <https://confluence.opensource.anotheria.net/display/MSK/Integration+Guide>



# AOP

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

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

```
    @DontMonitor public void thisMethodWillBeExcludedFromMonitoring(){
```

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

```
@CountByParameter  
void ingredientUsed(String ingredient) {
```

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

```
@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>2.8.7</version>
  </dependency>
  <dependency>
    <groupId>net.anotheria</groupId>
    <artifactId>moskito-aop</artifactId>
    <version>2.8.7</version>
  </dependency>
</dependencies>
```

```
<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>1.7</source>
        <target>1.7</target>
      </configuration>
      <executions>
        <execution>
          <goals>
            <goal>compile</goal>
          </goals>
        </execution>
      </executions>
    </plugin>
  </plugins>
</build>
```

# CDI

```
@Monitor
@ProducerRuntime(producerId="Foo", category="my")
public class YourClass {
```

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

```
@Monitor(MonitoringCategorySelector.WEB)
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(){}
}
```

```
@CountByParameter
void ingredientUsed(String ingredient) {
```

# CDI

```
<beans xmlns="http://java.sun.com/xml/ns/javaee" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="
http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/beans\_1\_0.xsd">
  <interceptors>
    <class>net.anotheria.moskito.integration.cdi.CountInterceptor</class>
    <class>net.anotheria.moskito.integration.cdi.CallInterceptor</class>

    <class>net.anotheria.moskito.integration.cdi.WebCallInterceptor</class>
    <class>net.anotheria.moskito.integration.cdi.ServiceCallInterceptor</class>
    <class>net.anotheria.moskito.integration.cdi.DaoCallInterceptor</class>

  </interceptors>
</beans>
```

# Proxies

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

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

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

# 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>
```

## Automagically with servlet 3.0

```
<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>
```

# Inspect

```
<dependency>  
  <groupId>net.anotheria</groupId>  
  <artifactId>moskito-inspect-remote</artifactId>  
  <version>2.8.7</version>  
</dependency>
```

# Success stories

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





- 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. MoSKito Inspect showed 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.

<u>MatchingService-1</u>	springservice	matching	88	3910	0	1	29	140	44.432	34	0
<u>MatchingService-2</u>	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

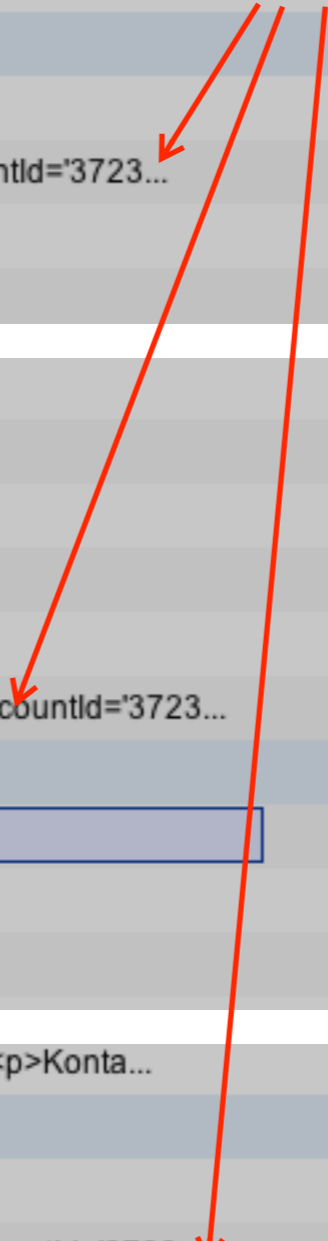


- Application overall performance was insufficient.
- With moskito journeys and call tree analysis we were able to find redundant calls to the backend and remove them.
- Request duration reduced to 50% with 4 hours analysis and 4 hours coding effort.



	time in milliseconds	
└ LoginAPI.isLoggedIn() = true	3	3
└ RegistrationAPI.isMeTest() = false	12	12
└ PaymentAPI.amIP [RegistrationAPI.isMeTest() = false]	665	79
└ <b>PaymentBusinessServiceDiMe</b> 1.readActivePayments(372347) = [PaymentBO{id='159538', accountId='3723...}	<b>585</b>	585
└ IASDataService-2.getNavitem(33) = Navitem [33] name: Hilfe, title: Hilfe, externalLink: , p...	21	21
└ LoginAPI.isLoggedIn() = true	9	9
└ [redacted]presentation.profile.handler.AboutMeHandler-C-57.process	767	92
└ LoginAPI.isLoggedIn() = true	10	10
└ LoginAPI.getLogedUserId() = 372347	9	9
└ RegistrationAPI.isMyEmailConfirmed() = true	10	10
└ PaymentAPI.amIPremium() = true	<b>603</b>	40
└ <b>PaymentBusinessServiceDiMe</b> -1.readActivePayments(372347) = [PaymentBO{id='159538', accountId='3723...}	562	562
└ [redacted]LoginAPI.getMyLoginTime() = 1271890272896	25	22
└ LoginAPI.isLoggedIn() = true [redacted]LoginAPI.getMyLoginTime() = 1271890272896	2	2
└ [redacted]LoginAPI.getMyPreviousLoginTime() = 1271651945362	15	12
└ LoginAPI.isLoggedIn() = true	2	2
└ IASWebDataService-2.getBox(167) = Box [167] name: Messagesbox: Cleanup Warning, content: <p>Konta...	30	30
└ RegistrationAPI.isMeTest() = false	13	13
└ PaymentAPI.amIPremium() = true	668	62
└ <b>PaymentBusinessServiceDiMe</b> 1.readActivePayments(372347) = [PaymentBO{id='159538', accountId='3723...}	<b>605</b>	605
└ IASWebDataService-2.getBox(1) = Box [1] name: Footer, content: <p>© {cal:currentYear} {text:brand...	13	13

same call over net repeated thrice





- ◉ A bug only reproduceable by one user.
- ◉ Second page of received messages won't show. No exception logged.

Die Aktion konnte nicht ausgeführt werden

Es ist leider ein unerwarteter Fehler aufgetreten. Unsere Technik ist informiert und wird alles tun, um den Fehler zu beheben und künftig zu vermeiden.

Bitte wiederholen Sie einfach Ihre gerade durchgeführte Aktion. Sollte erneut ein Fehler auftreten, versuchen Sie es bitte zu einem späteren Zeitpunkt noch einmal. Vielen Dank für Ihr Verständnis.

# Broken Call

```
Call
ROOT
└ ActivityAPI.notifyMyActivity(http://w...de...ssages/receivedpage=2) = null
└ MessageAPI.getMyMailBox(INBOX, MESSAGE_DATE, 2, false) = null
  └ ProfileService-2.getUserById(ATKB1PNA) = com.f...model.impl.UserBean@1f8808d...
  └ SettingsConfigurationService-1.getOverridableProperties() = ...module.Setti...
  └ PersonalityDAO-1.getPersonalityTraitMap(ATKB1PNA) = ATKB1PNA [1112451721, -1541634238, 268443665,...
  └ PersonalityDAO-1.getPersonalityTraitMap([GX5N48BW, IER1X2P3, AWZCKC01, ATYE78KY]) = {GX5N48BW=GX5...
  └ ProfileService-1.getAddressesByUserIds([GX5N48BW, IER1X2P3, ATYE78KY, AWZCKC01]) = {IER1X2P3=logi...
  └ SearchDAO-1.getLatitudeLongitude(AT, 1020) = com.f...search.model.impl.GeoDataBean@7...
  └ DistanceService-1.getDistanceToShow(GX5N48BW, 4) = 20
  └ SearchDAO-1.getLatitudeLongitude(US, 60076) = null
  └ DistanceService-1.getDistanceToShow(IER1X2P3, null) = null
  └ SearchDAO-1.getLatitudeLongitude(AT, 1230) = com.f...search.model.impl.GeoDataBean@1...
  └ DistanceService-1.getDistanceToShow(ATYE78KY, 15) = 20
  └ SearchDAO-1.getLatitudeLongitude(AT, 3012) = com.f...search.model.impl.GeoDataBean@6...
  └ DistanceService-1.getDistanceToShow(AWZCKC01, 23) = 50
  └ ProfileService-2.getUserById(ATKB1PNA) = com.f...profile.model.impl.UserBean@1f8808d...
  ,_minage=33 years,_maxage=55
  years,_smoking=REGULARSMOKER,_height=189,_version=2,_hidemyvisits=,_matchstatus=1,_searchstatus=1,_profilestatus=0,_minheight=165,_maxheight=17
  in der Personalverwaltung davor Sportlehrer,_relationshipType=[7, 2],_languages=[com.freiheit.parship.profile.model.LanguageMap@345b01e8],_instrumentFree
  WIDTH, 400)" onmouseout="UnTip()"> └ ProfileService-2.getUserById(GX5N48BW) = com.freiheit.parship.profile.model.impl.UserBean@2cefe40...
  └ PartnerAPI.getMyPartnerAlias(GX5N48BW) = Peter
    └ ContactsService-1.getAliasForContact(ATKB1PNA, GX5N48BW) = Peter
  ,_minage=33 years,_maxage=55
  years,_smoking=REGULARSMOKER,_height=189,_version=2,_hidemyvisits=,_matchstatus=1,_searchstatus=1,_profilestatus=0,_minheight=165,_maxheight=17
  in der Personalverwaltung davor Sportlehrer,_relationshipType=[7, 2],_languages=[com.freiheit.parship.profile.model.LanguageMap@345b01e8],_instrumentFree
  WIDTH, 400)" onmouseout="UnTip()"> └ ProfileService-2.getUserById(GX5N48BW) = com.freiheit.parship.profile.model.impl.UserBean@2cefe40...
  └ PartnerAPI.getMyPartnerAlias(GX5N48BW) = Peter
    └ ContactsService-1.getAliasForContact(ATKB1PNA, GX5N48BW) = Peter
  └ ProfileService-2.getUserById(IER1X2P3) = null
```

red color indicates that the method was terminated abnormally (exception)

possible reason - user didn't exist in the db and the code didn't know how to handle it.

# End of Data

Thank you.



<http://www.moskito.org>



<https://itunes.apple.com/de/app/moskito-ui/id531387262?l=en&mt=8>

<https://itunes.apple.com/de/app/msk-control/id688838411?l=en&mt=8>



<https://github.com/anotheria/moskito>

<https://github.com/anotheria/moskito-control>

<https://github.com/anotheria/moskito-demo>



<http://search.maven.org/#search%7Cga%7CI%7Cmoskito>