Java Instrumentation API
a working example

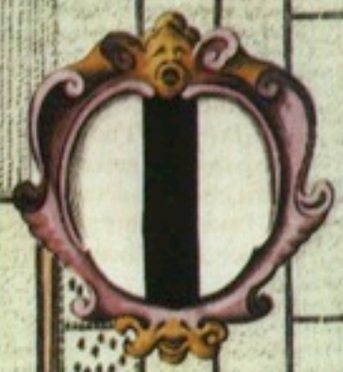
monitoring
made
simple

Kees Jan Koster
kjkoster@java-monitor.com


Java™


CASTELO
S. JORGE
Lisboa
Portugal
Lisboa

DELFI BATAVORVM
vernacule
DELFT

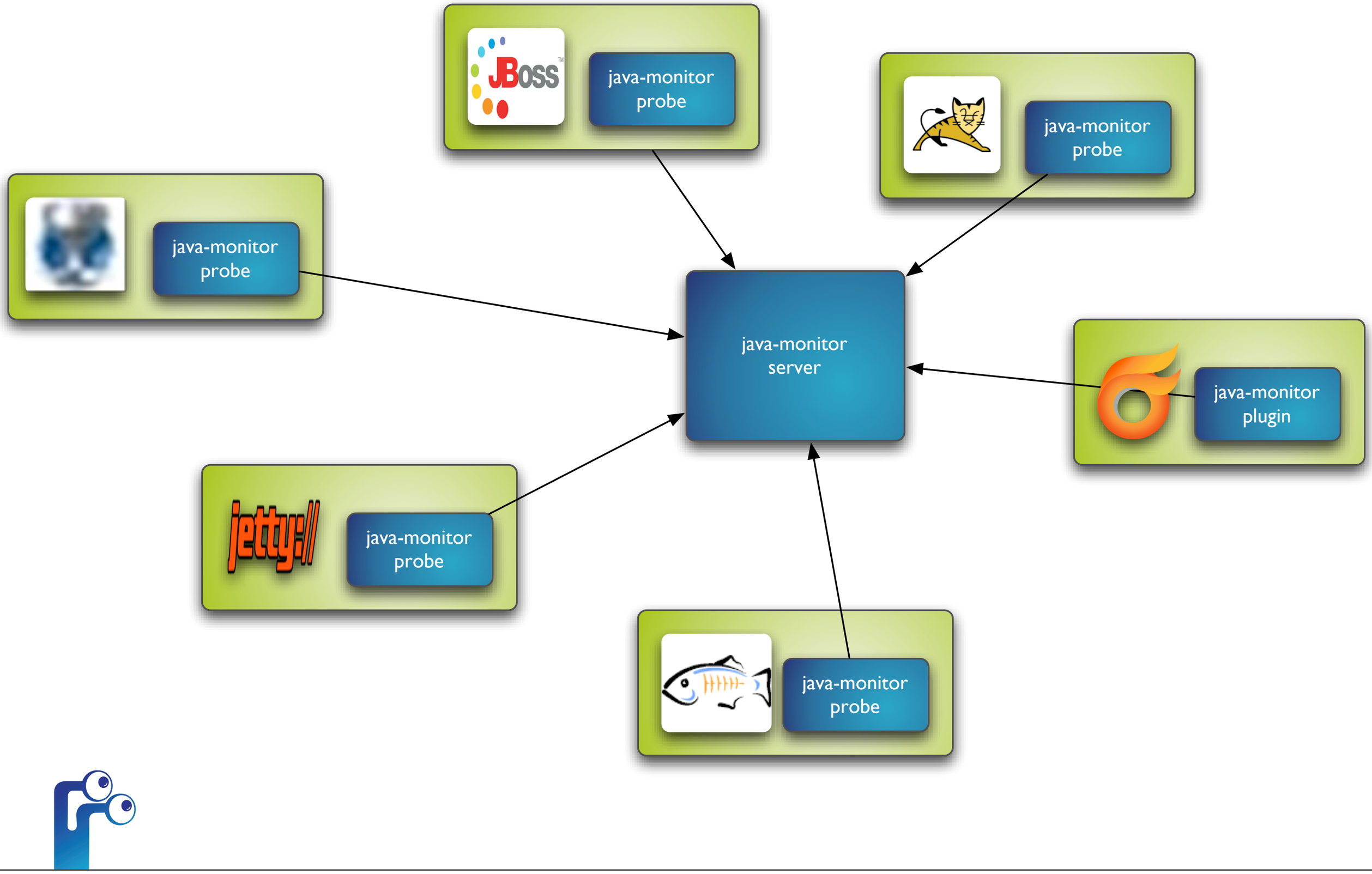


Tuesday, October 2, 12

monitoring made simple

- **http://java-monitor.com/livedemo.html**

- **free, on-line service**
- **up and running in 5 minutes**
- **access server stats from your mobile**
- **get help in the integrated forum**


java-monitor probe
java-monitor probe
java-monitor probe
java-monitor probe
java-monitor server
java-monitor plugin
java-monitor probe
java-monitor probe

my servers | my friends | download probe

	cpu	load	heap	non-heap	fd's	thread and database pools	
Apache Tomcat @ 188.201.178.225 P production view server »	0%	23%	44%	48%	1%	2%	6%
Apache Tomcat @ kjkoster.org P production view server »	0%						
lemongrass @ dill.beta-monitor.com P production view server »	2%	0%	14%	20%	1%	0%	2% 0% 0% 0%

my servers | my friends | download probe

admin	no servers
dkumar2	no servers
god	no servers
7scenes-admin	no servers
Kees de Kooter	no servers
lancetx	5000 servers
JohnDoe	1 server
guus	no servers

Apache
188.201
P production

heap 44% | non-heap 48% | fd's 1% | thread and database pools 2% 6%

Apache Tomcat @ kjkoster.org
P production
view server »

0%

lemongrass @ dill.beta-monitor.com
P production
view server »

2% 0% | 14% 20% 1% | 0% 2% 0% 0% 0%

my servers | my friends

download probe

cpu	load	heap	non-heap	fd's	thread and database pools				
2%	0%	14%	20%	1%	0%	2%	0%	0%	0%

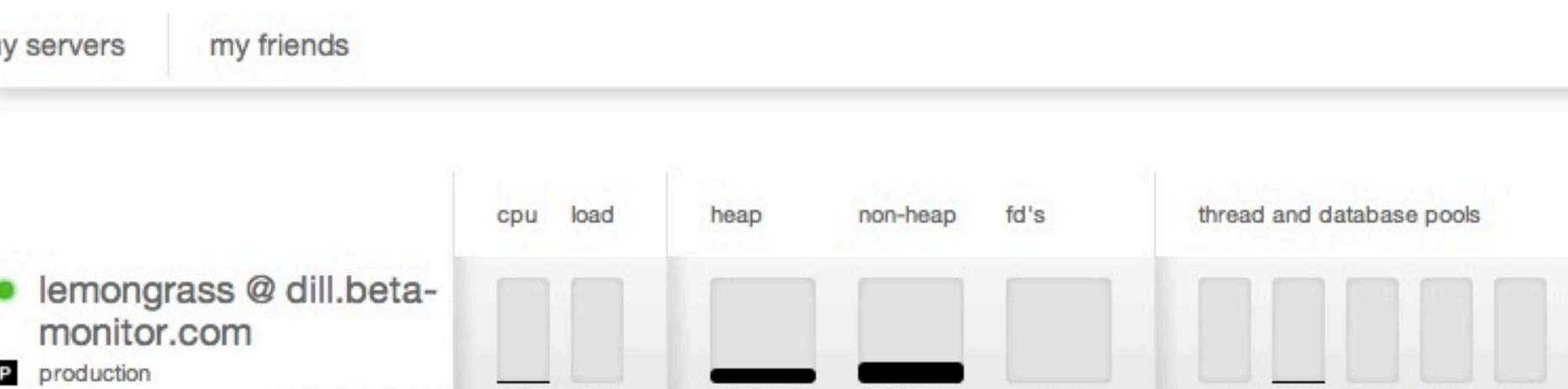
lemongrass @ dill.beta-monitor.com

P production

configuration »

details | processing | **memory** | thread pools | database pools | hibernate | file descriptors

Heap Memory



Post this graph on the Java Administration*5 forum Go


7,749
18 / 30
Text from javamonitor
OFF-LINE www @ kjkoster.org.
Last data Oct 01 20:38:47 GMT, 2
mins and 15 sec ago. Uptime 1 day.
Heap 51% non-heap 31% SMS left
87
Close
Reply

Java Instrumentation API
a working example

monitoring
made
simple

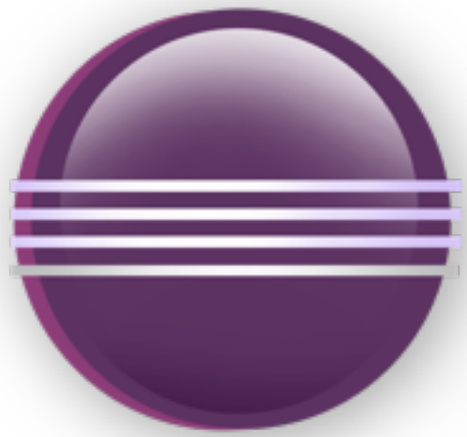
Kees Jan Koster
kjkoster@java-monitor.com

- **instrumentation API**
- **byte code manipulation**
- **hello world example**
- **web app example**
- **recap**

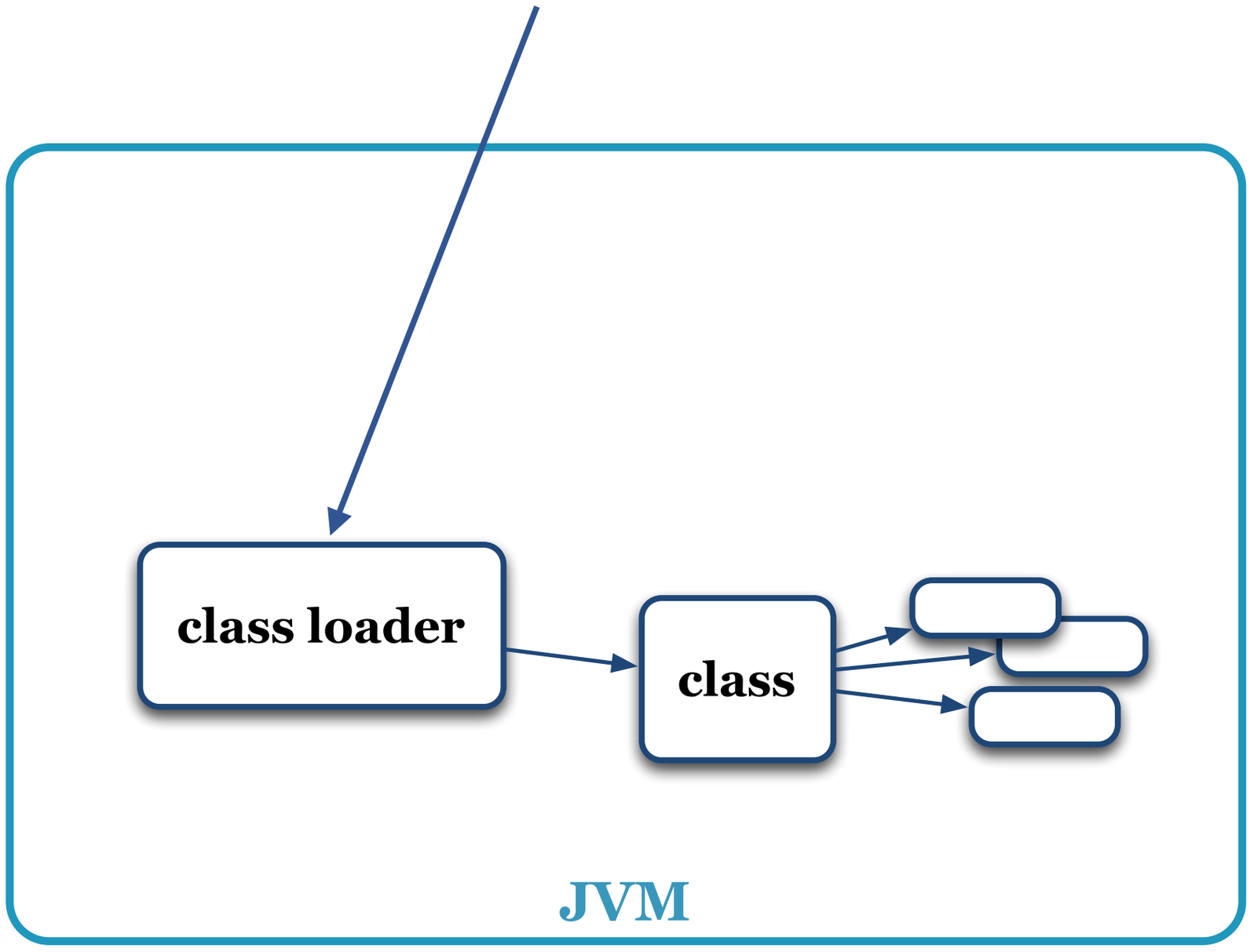
- 100% Java api for profilers
- portable between JVM implementations
- byte code transformation
- typical uses
  - AOP
  - logging
  - profiling
  - contract checking, e.g. non-null arguments
  - hot code (re)deploy

- **load system classes**
- **load main class**
- **call main() on main class**

- load system classes
- load agent class
- call premain() on agent
- load main class
- call main() on main class




```
CA FE BA BE 00 04 AD 00 DA 00 .. .. ..
```




```
CA FE BA BE 00 04 AD 00 DA 00 .. .. ..
```

Transformer

```
CA FE BA BE 00 04 AD 00 BE 45 76 .. .. ..
```

class loader

class

JVM


```
CA FE BA BE 00 04 AD 00 DA 00 .. .. ..
```

0xCAFEBABE	version #
constant pool e.g. strings	
access flags	
class info	
fields	
methods	
attributes	


```
CA FE BA BE 00 04 AD 00 DA 00 .. .. ..
```

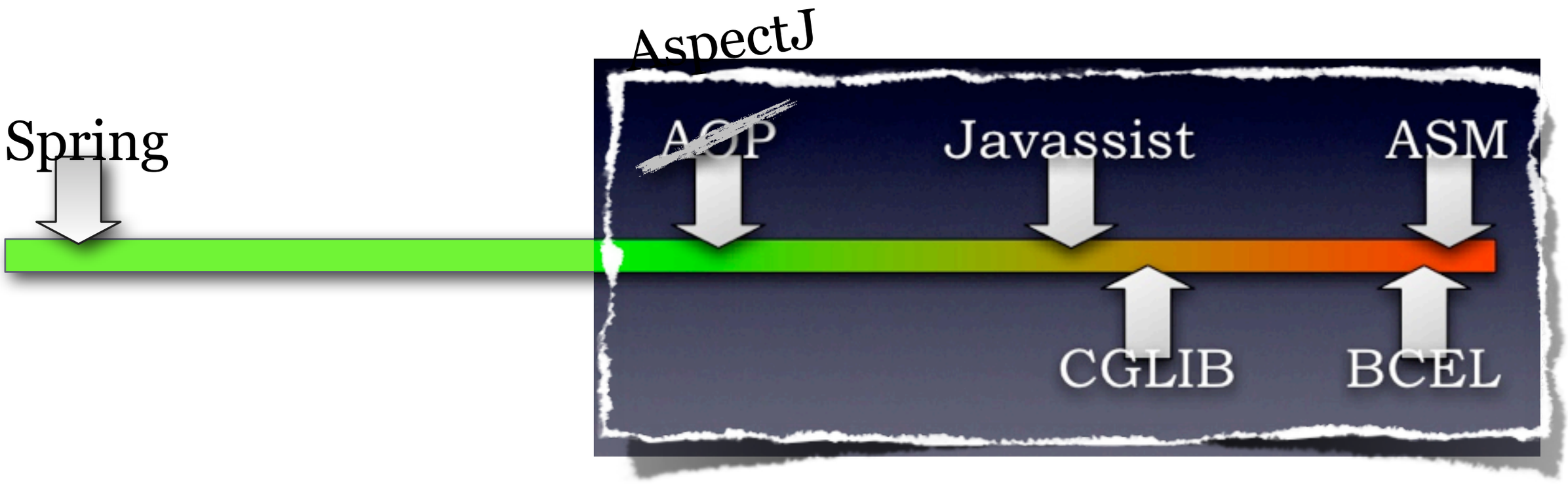
0xCAFEBABE	version #
constant pool e.g. strings	
access flags	
class info	
fields	
methods	
attributes	

Transformer

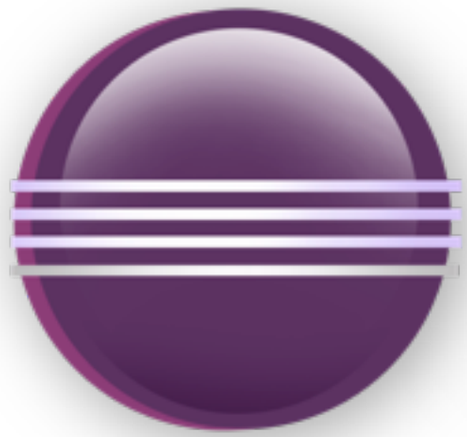
```
static final Logger _log = new ...
```

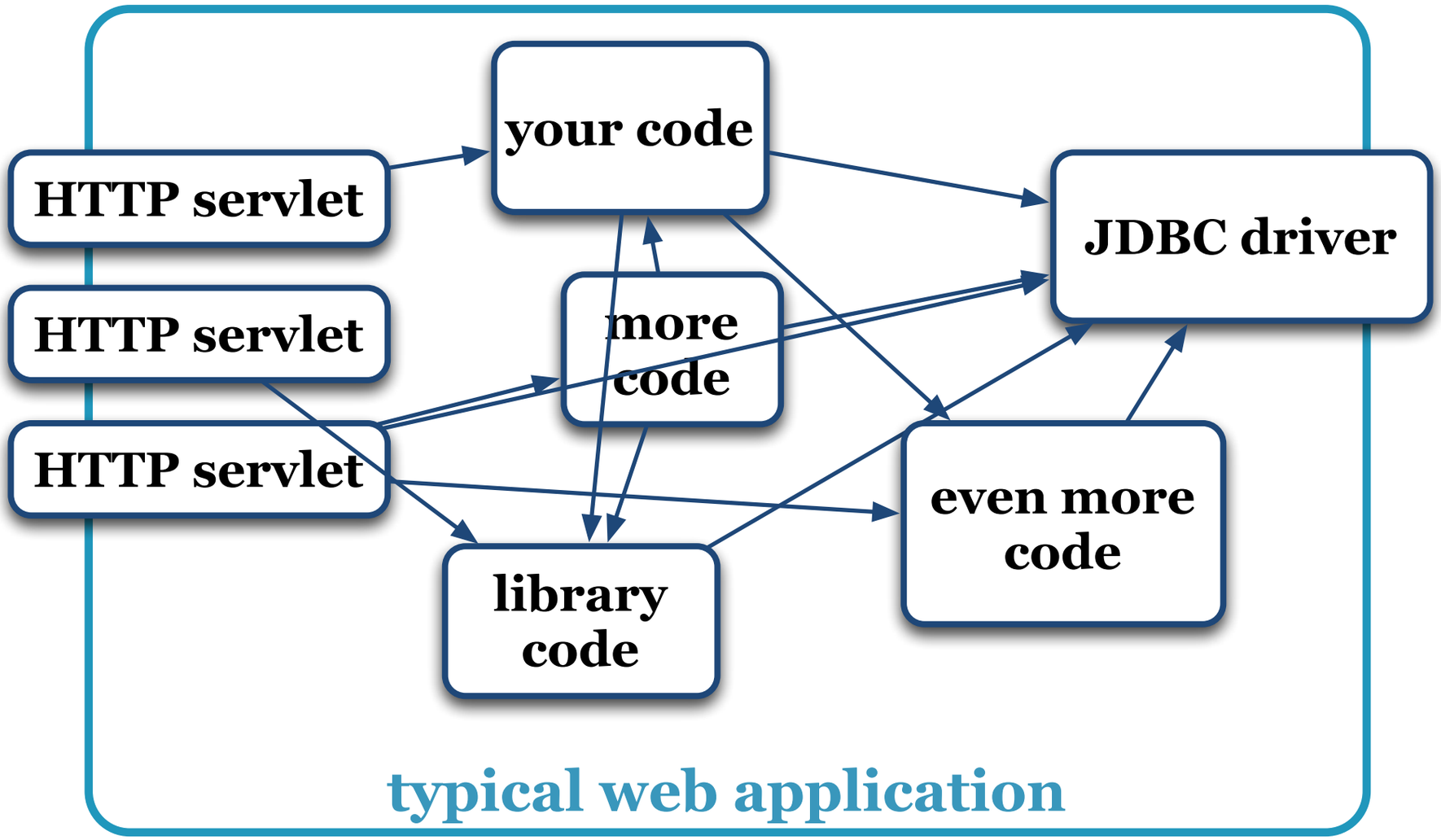
```
_log.info(…);
```

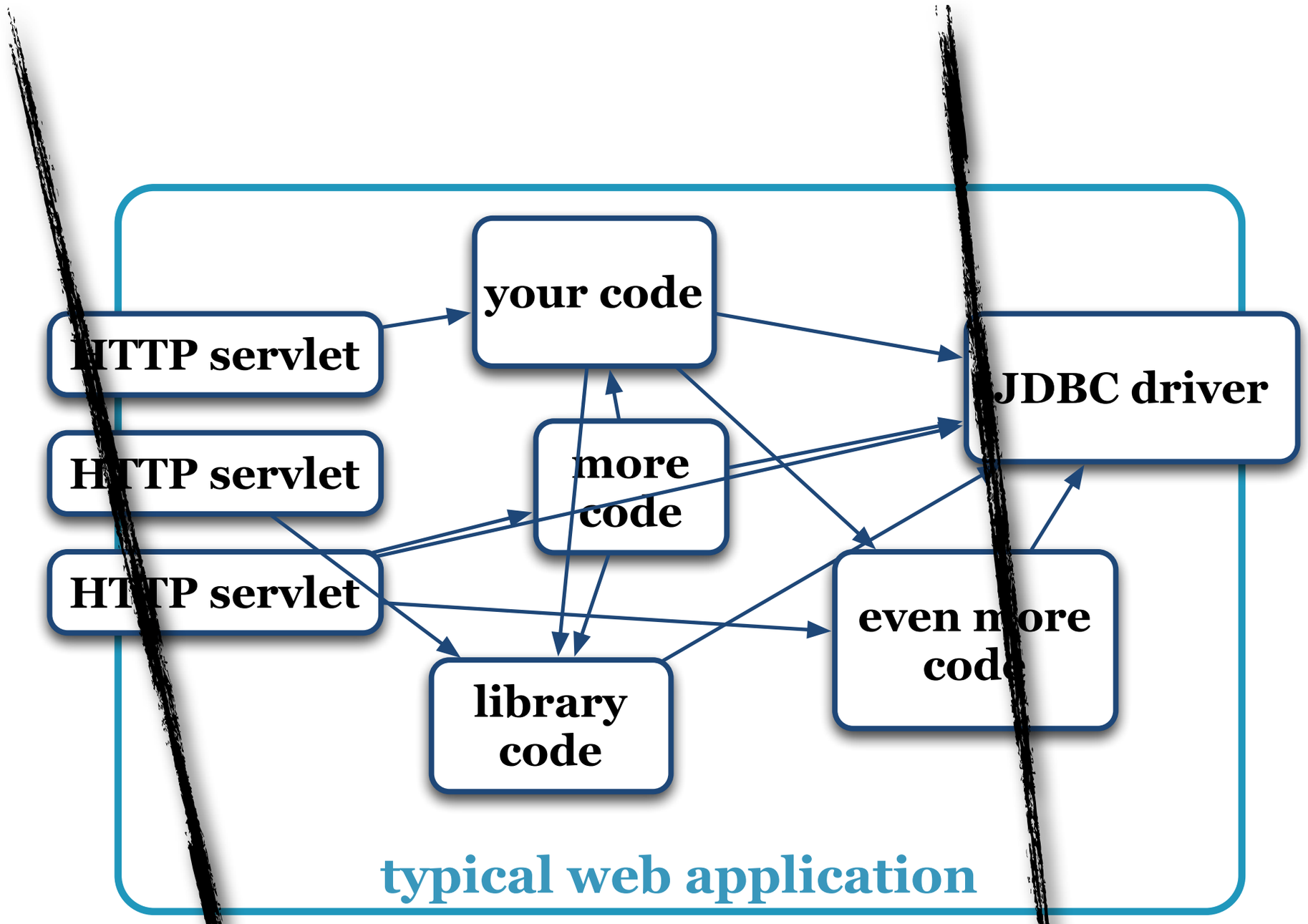
```
_log.info(…);
```



http://www.slideshare.net/tcurdt/no-dark-magic-byte-code-engineering-in-the-real-world

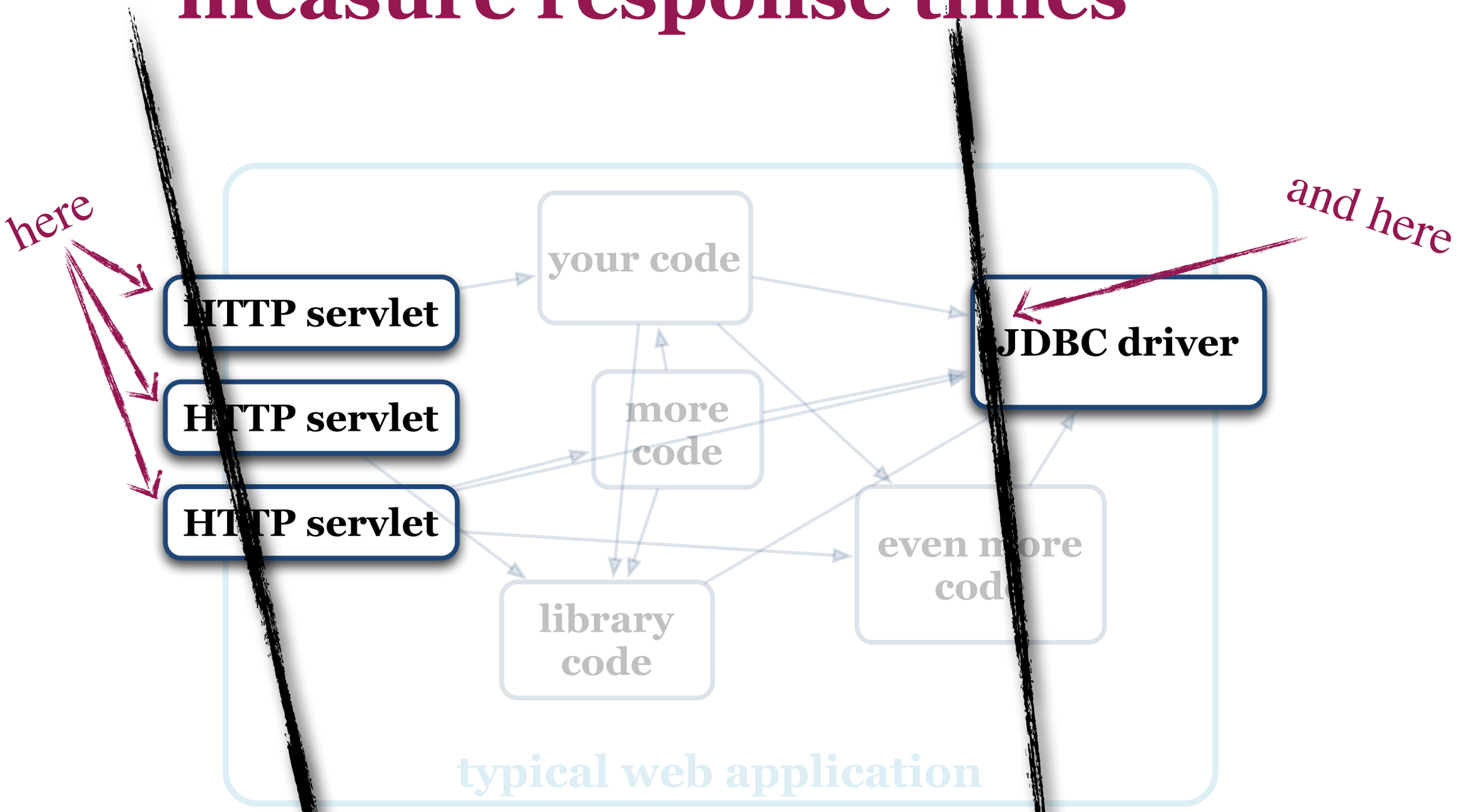



your code
HTTP servlet
HTTP servlet
HTTP servlet
more code
JDBC driver
even more code
library code
typical web application



slow network **slow code** **slow database**

measure response times



slow network **slow code** **slow database**

Transformer

```
long start = currentTimeMillis();
```

HTTP servlet

HTTP servlet

HTTP servlet

```
_log.info(currentTimeMillis()-start);
```

your code

more code

library code

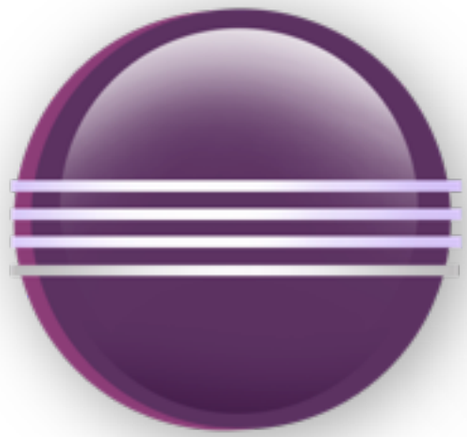
even more code

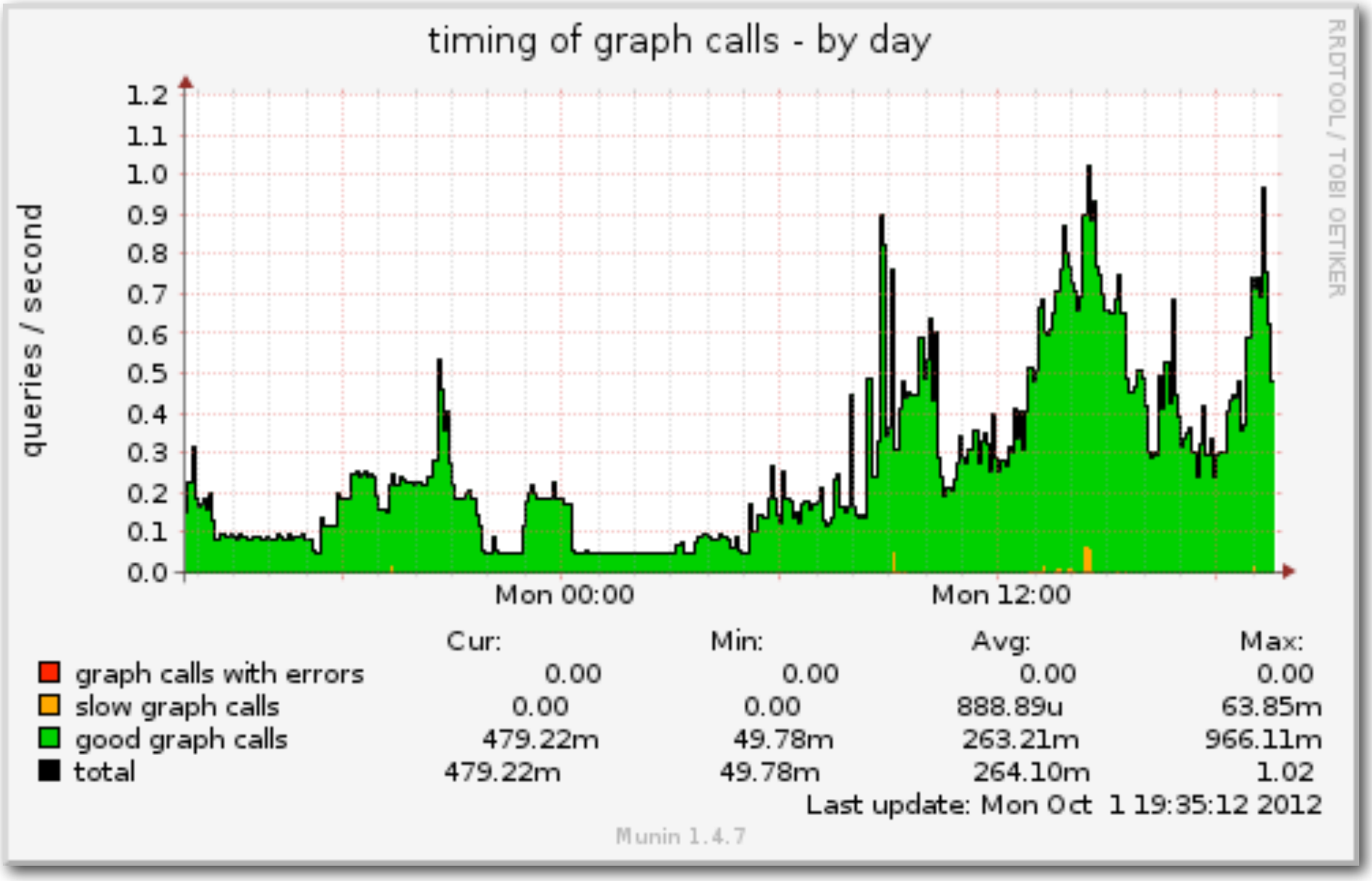
typical web application

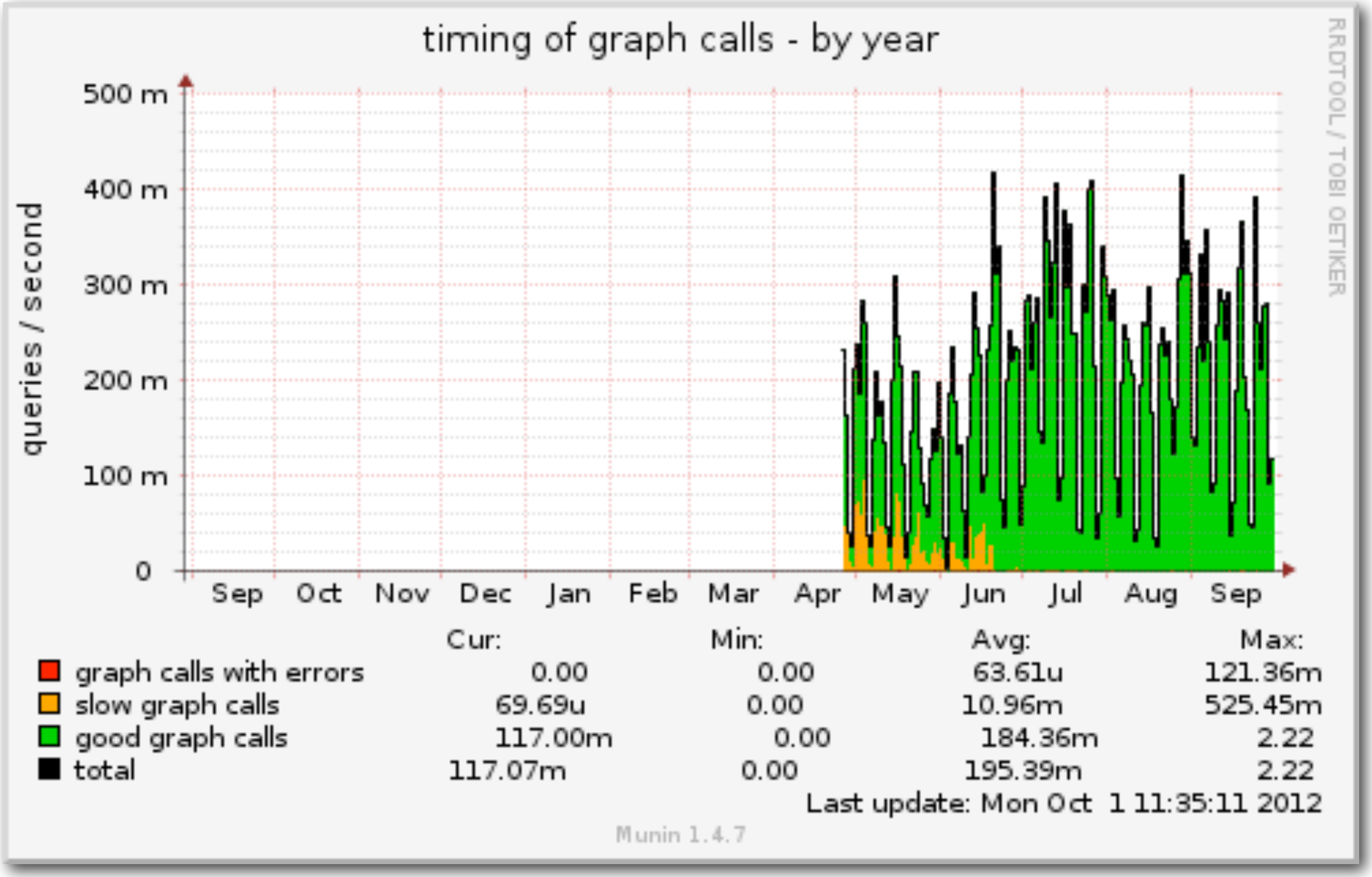
```
long start = currentTimeMillis();
```

JDBC driver

```
_log.info(currentTimeMillis()-start);
```




timing of graph calls - by day
RRDTOOL / TOBI OETIKER
queries / second
1.2
1.1
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
Mon 00:00
Mon 12:00
Cur:
Min:
Avg:
Max:
graph calls with errors 0.00 0.00 0.00 0.00
slow graph calls 0.00 0.00 888.89u 63.85m
good graph calls 479.22m 49.78m 263.21m 966.11m
total 479.22m 49.78m 264.10m 1.02
Last update: Mon Oct 1 19:35:12 2012
Munin 1.4.7


timing of graph calls - by year
RRDTOOL / TOBI OETIKER
queries / second
500 m
400 m
300 m
200 m
100 m
0
Sep
Oct
Nov
Dec
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Cur:
Min:
Avg:
Max:
graph calls with errors 0.00 0.00 63.61u 121.36m
slow graph calls 69.69u 0.00 10.96m 525.45m
good graph calls 117.00m 0.00 184.36m 2.22
total 117.07m 0.00 195.39m 2.22
Last update: Mon Oct 1 11:35:11 2012
Munin 1.4.7

webapp

Tomcat

JVM

hardware
& OS

munin-node

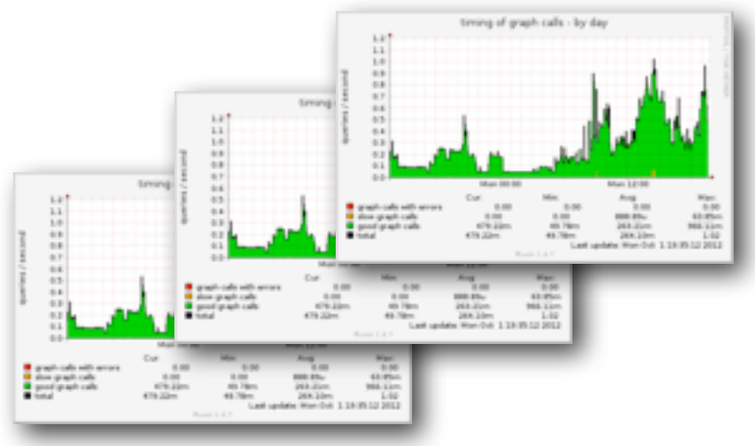
script

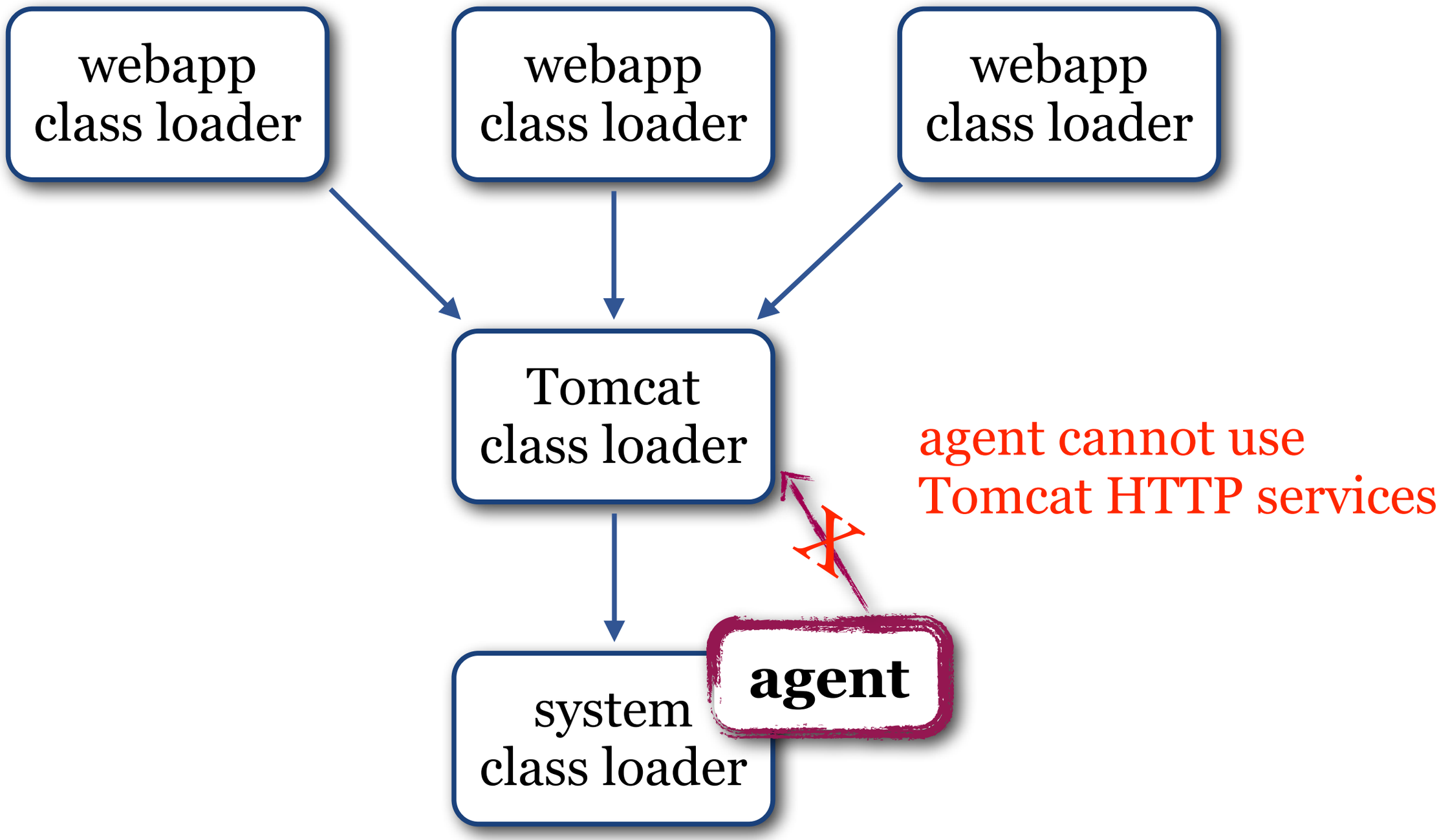
script

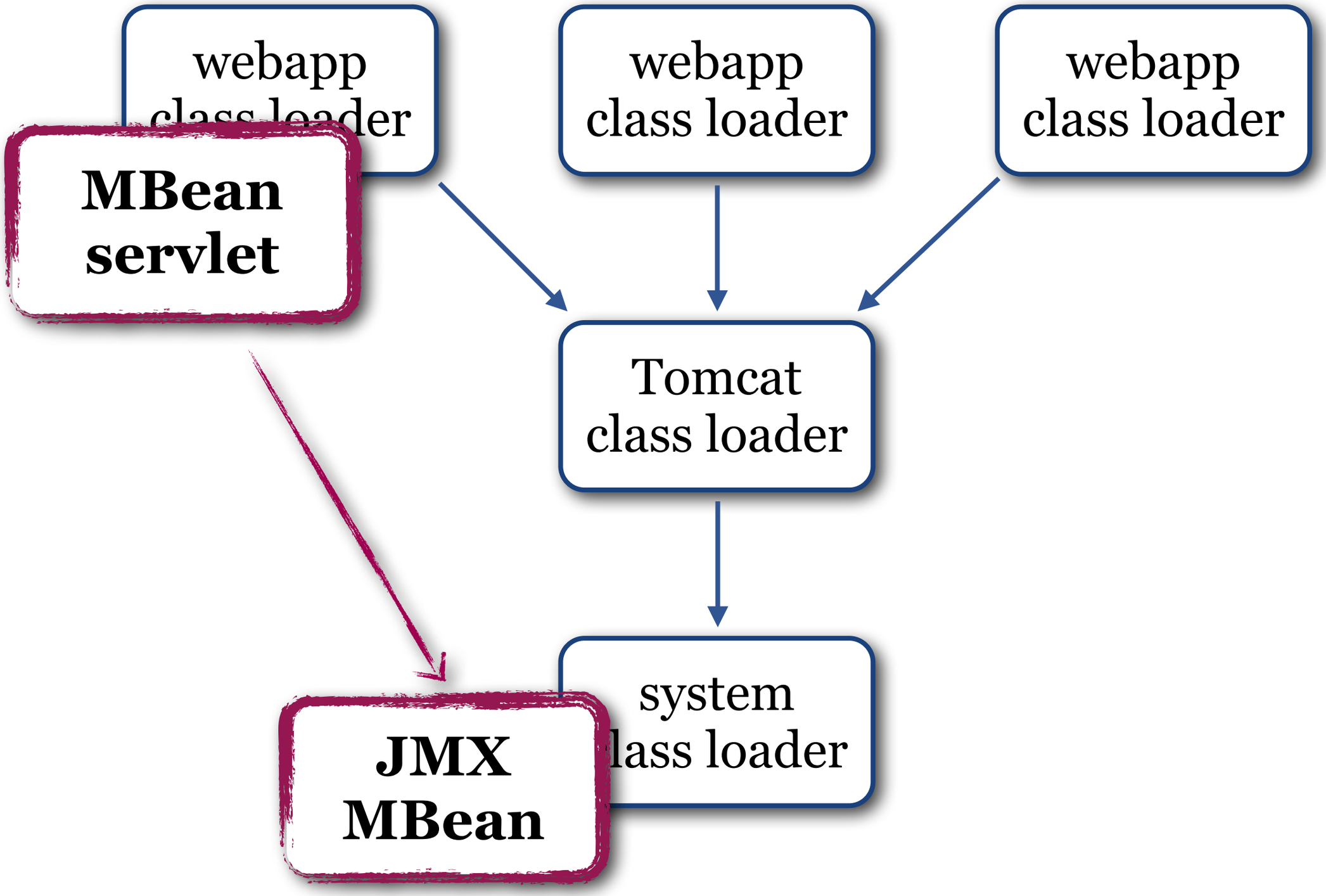
script

script

munin
master




webapp
class loader
webapp
class loader
webapp
class loader
Tomcat
class loader
system
class loader
agent
agent cannot use
Tomcat HTTP services


webapp
class loader
webapp
class loader
webapp
class loader
MBean
servlet
Tomcat
class loader
system
class loader
JMX
MBean