# DevOps: Теперь Java не тормозит

ПОТЕРЯННЫЙ МАНУАЛ О ТОМ, КОМУ РЕШАТЬ НЕУДОБНЫЕ ЗАДАЧИ

# Disclaimer

The following is intended to outline our general devops process direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release and timing of any features or functionality described for Sberbank-Technology's products or services remains at the sole discretion of Sberbank-Technology.

Информация предназначается чтобы обозначить наше общее направление формирования процесса DevOps. Она предоставляется только в целях ознакомления, и не может быть использована в контрактах или договорах любого вида. Эта информация не является попыткой предоставления какого-то материала, код, или функциональности, и не должна быть использована в принятии коммерческих решений. Разработка, выпуск, календарные сроки любых объектов или функциональности, описанных в контексте продуктов или услуг компании Сбербанк-Технологии, остается на усмотрение компании Сбербанк-Технологии.

# Кто здесь?

- Госуслуги (с обеих сторон баррикады)
- ЕГИСЗ - (Единая государственная информационная система в сфере здравоохранения),
- ИЭМК - (Интегрированная электронная медицинская карта). Интеграция с ГИБДД, МВД, итп
- IUPAT - (The International Union of Painters and Allied Trades) - главная информационная система профсоюзов
- Учствовал в разработке двух языков программирования, SDK для них, и плагинов для IDE
- Администрировал веб-приложения и видеостриминговые сервера

**Сейчас:**

- Сбербанк-Технологии, система для выполнения BPMN бизнес-процессов

СБЕРБАНК ТЕХНОЛОГИИ

# Определение по Википедии

DevOps (акроним от англ. development и operations) — набор практик, нацеленных на активное взаимодействие и интеграцию специалистов по разработке и специалистов по информационно-технологическому обслуживанию.



https://ru.wikipedia.org/wiki/DevOps

# Определение по Википедии

DevOps (акроним от англ. development и operations) — набор практик, нацеленных на активное взаимодействие и интеграцию специалистов по разработке и специалистов по информационно-технологическому обслуживанию.

▸▸▸ Значит всё, что угодно



https://ru.wikipedia.org/wiki/DevOps

# Определение перечислением

- **Code** — Code development and review, version control tools, code merging;
- **Build** — Continuous integration tools, build status;
- **Test** — Test and results determine performance;
- **Package** — Artifact repository, application pre-deployment staging;
- **Release** — Change management, release approvals, release automation;
- **Configure** — Infrastructure configuration and management, Infrastructure as Code tools;
- **Monitor** — Applications performance monitoring, end–user experience.



https://en.wikipedia.org/wiki/DevOps

# Определение перечислением

- **Code** — Code development and review, version control tools, code merging;
- **Build** — Continuous integration tools, build status;
- **Test** — Test and results determine performance;
- **Package** — Artifact repository, application pre-deployment staging;
- **Release** — Change management, release approvals, release automation;
- **Configure** — Infrastructure configuration and management, Infrastructure as Code tools;
- **Monitor** — Applications performance monitoring, end–user experience.

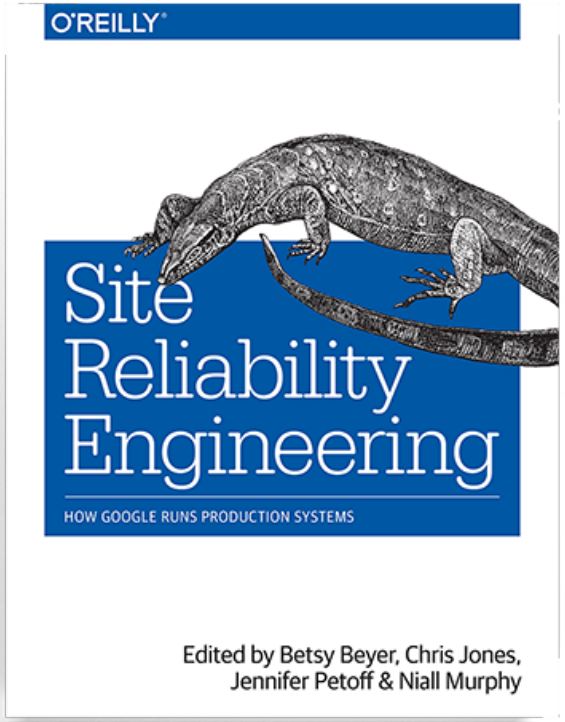
▶▶▶ Перечислена вся разработка



https://en.wikipedia.org/wiki/DevOps

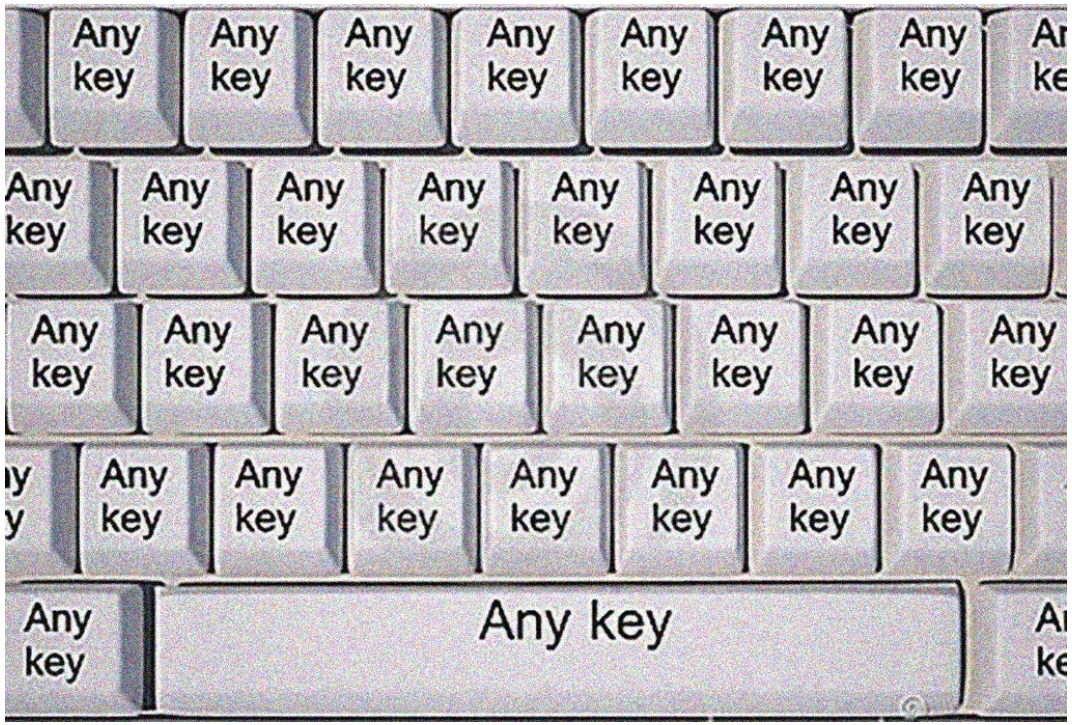
# DevOps or SRE?

**One could equivalently view SRE as a specific implementation of DevOps with some idiosyncratic extensions.**




https://landing.google.com/sre/book/chapters/introduction.html

# Первое впечатление: эникейщик




СБЕРБАНК ТЕХНОЛОГИИ

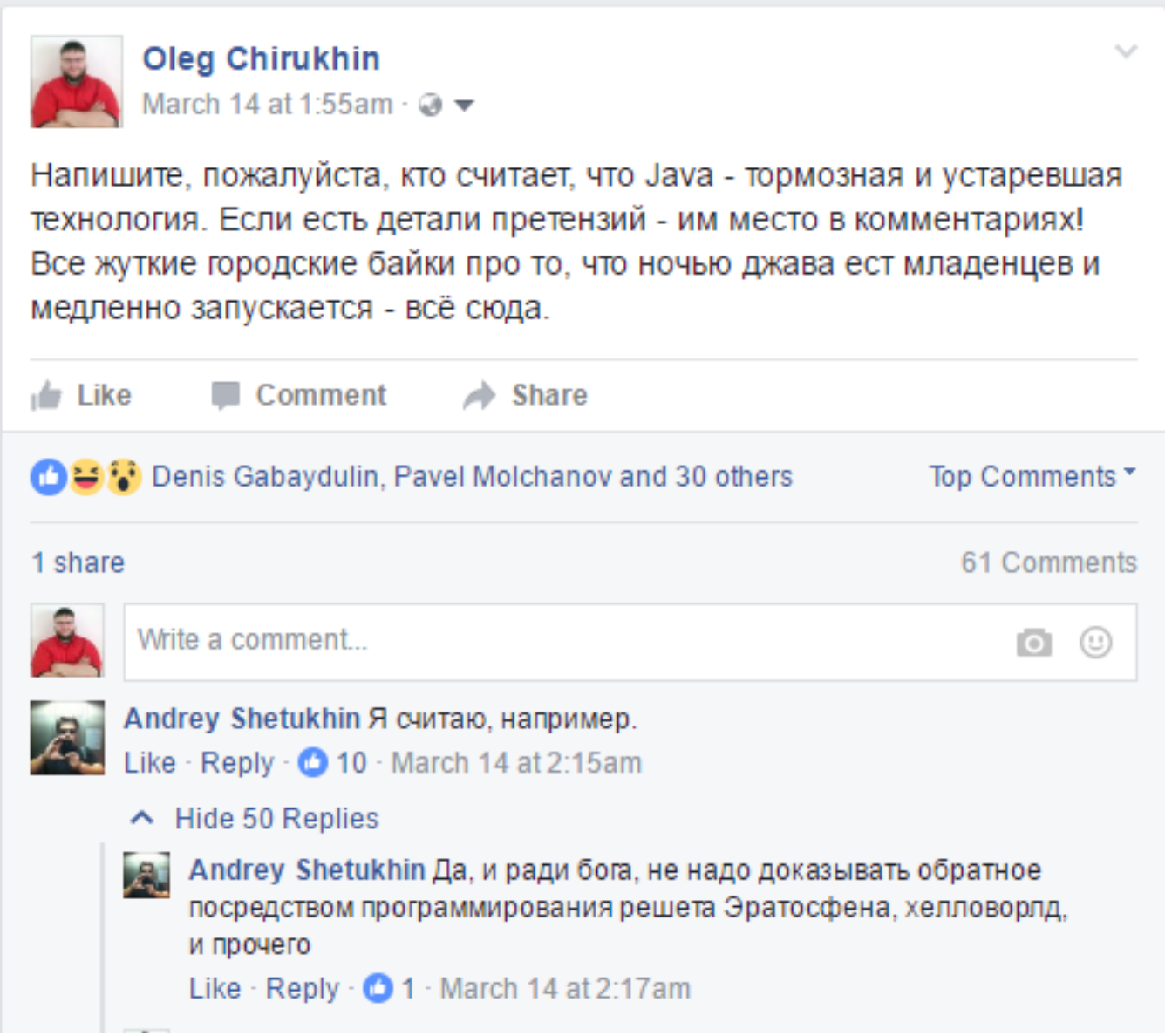
# Второе впечатление

- **MTTR** - Mean Time To Repair
- **MTBF** - Mean Time Between Failures
- **MTTF** - Mean Time To Failure
- **TTM** - Time To Market (что это здесь делает?)

http://disney.wikia.com/wiki/Spider-Man

# Логика повествования

- Психологическая часть вопроса
- Покажу что девопсы нам реально нужны
- Техническая часть вопроса и типичные задачи
- Собрать всё вместе и сделать какое-то заключение




https://www.facebook.com/olegchir/posts/1382735871749416

# Андрей Шетухин

- Долгое время руководил разработкой Рамблер-Почты
- 100 миллионов пользователей
- 1,5 петабайта данных
- 10 000 запросов в секунду
- По поводу Java – это его официальная позиция

## ▸▸▸ Реальные люди, не анонимные тролли



https://2017.codefest.ru/speaker/8745/

# Типичные претензии

- Тормозит и ест RAM
- Не развивается
- Неудобная
- Устаревшая
- Долго выкатывать
- Сложно обслуживать

# Типичные контр-аргументы

- Сравнили наш компилятор и GC с Golang и поняли, что мы круче
  - особенно если GC это Shenandoah от Алексея Шипилёва
  - Mid-stack inlining в Go реализовали только сейчас (https://goo.gl/LVfVKg)

# Mid-stack inlining

- Инлайнинг заменяет вызов функции на тело этой функции
- Mid-stack inlining — замена промежуточных функций
- 10% на типичном бенчмарке
- Go1.9Maybe (https://github.com/golang/go/issues/19348)

```
func Hypot(x, y float64) {
        return Sqrt(x*x + y*y)
}

for x := range xs {
        s += Hypot(x, C)
}
```

```
for x := range xs {
        s += Sqrt(x*x + C*C)
}
```

# Типичные контр-аргументы

- Сравнили наш компилятор и GC с Golang и поняли, что мы круче
  - особенно если GC это Shenandoah от Алексея Шипилёва
  - Mid-stack inlining в Go реализовали только сейчас (https://goo.gl/LVfVKg)
- В Джаве одна из лучших инфраструктур
  - даже C++ собирают через Maven! (https://goo.gl/dDxOXW)

# Типичные контр-аргументы

- Сравнили наш компилятор и GC с Golang и поняли, что мы круче
  - особенно если GC это Shenandoah от Алексея Шипилёва
  - Mid-stack inlining в Go реализовали только сейчас (https://goo.gl/LVfVKg)
- В Джаве одна из лучших инфраструктур
  - даже C++ собирают через Maven! (https://goo.gl/dDxOXW)
- Огромное количество инноваций
  - Большинство докладов на Jbreak/Jpoint
  - Graal & Truffle
  - Project Jigsaw
  - Project Panama

# Типичные контр-аргументы

- Сравнили наш компилятор и GC с Golang и поняли, что мы круче
  - особенно если GC это Shenandoah от Алексея Шипилёва
  - Mid-stack inlining в Go реализовали только сейчас (https://goo.gl/fvKg
- В Джаве одна из лучших инфраструктур
  - даже C++ собирают через Maven (https://goo.gl/JDxbXW)
- Огромное количество инноваций
  - Большинство докладов на Jbreak/Jpoint
  - Graal & Truffle
  - Project Jigsaw
  - Project Panama

MYTH BUSTED!

СБЕРБАНК ТЕХНОЛОГИИ

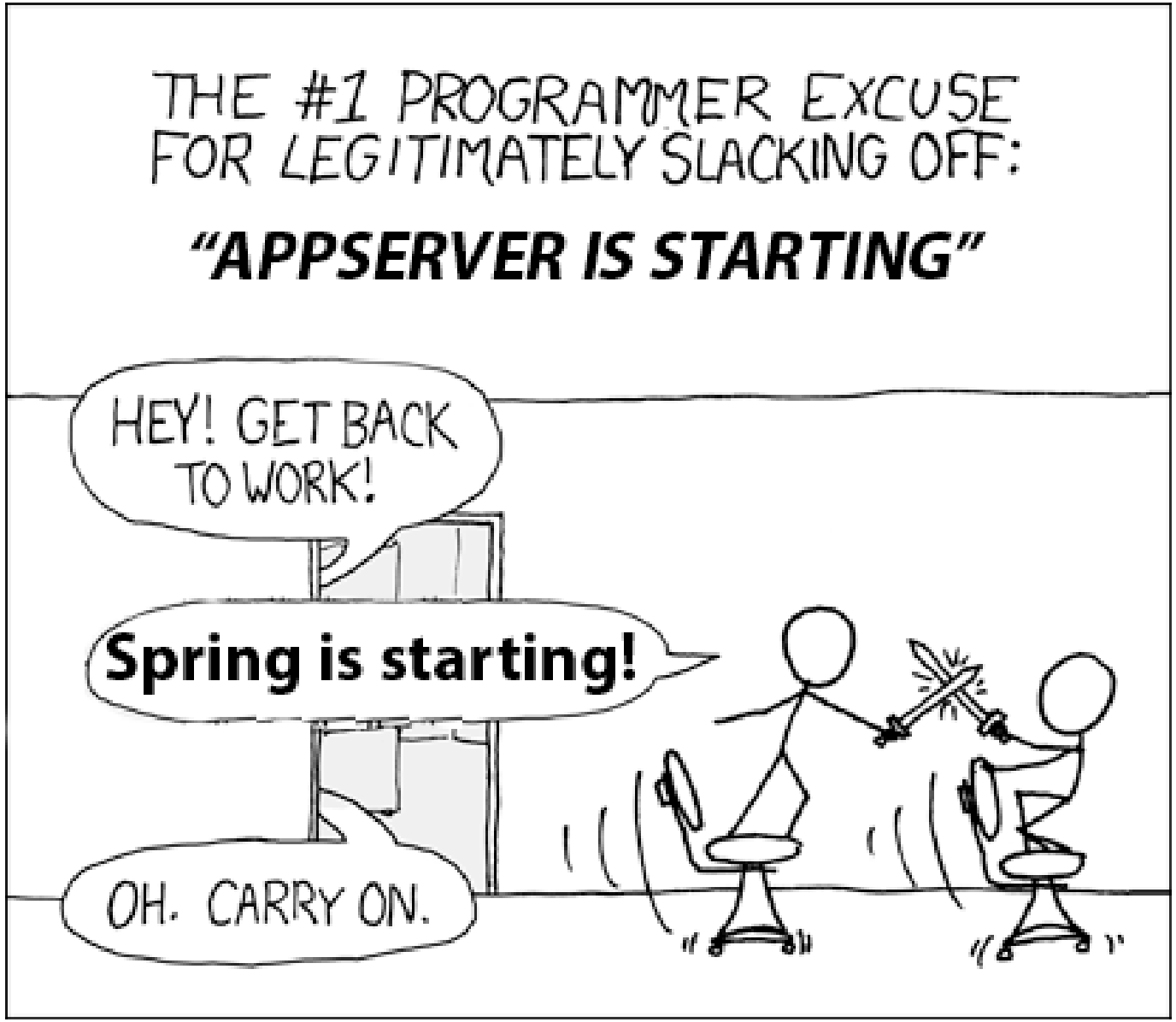
# Доклады: улучшение качества кода

- **Антон Кекс** — Как нам спасти Java?
- **Николай Алименков** — Парадигмы ООП
- **Егор Бугаенко** — Объектно-ориентированное вранье
- Другие доклады на **http://jug.ru**

# Доклады: улучшение качества кода

- **Антон Кекс** — Как нам спасти Java?
- **Николай Алименков** — Парадигмы ООП
- **Егор Бугаенко** — Объектно-ориентированное вранье
- Другие доклады на **http://jug.ru**

▸▸▸ Не только прикладного кода, но и базовой платформы




https://xkcd.com/303/

# Качество платформы. Spring Boot

- Spring DevTools
  - Настоящие рестарты

# Качество платформы. Spring Boot

- Spring DevTools
  - Настоящие рестарты
- HotSwapAgent (DCEVM – патч к OpenJDK/HotSpot)
  - Java 8 и выше?

# Качество платформы. Spring Boot

- Spring DevTools
  - Настоящие рестарты
- HotSwapAgent (DCEVM – патч к OpenJDK/HotSpot)
  - Java 8 и выше?
- Jrebel
  - Закрытый исходный код
  - Коммерческий продукт
  - Стоит приличных денег
  - Иногда тормозит сам по себе
  - В остальном супер

‘The planet is fine.
The people are f[redacted].’
- George Carlin

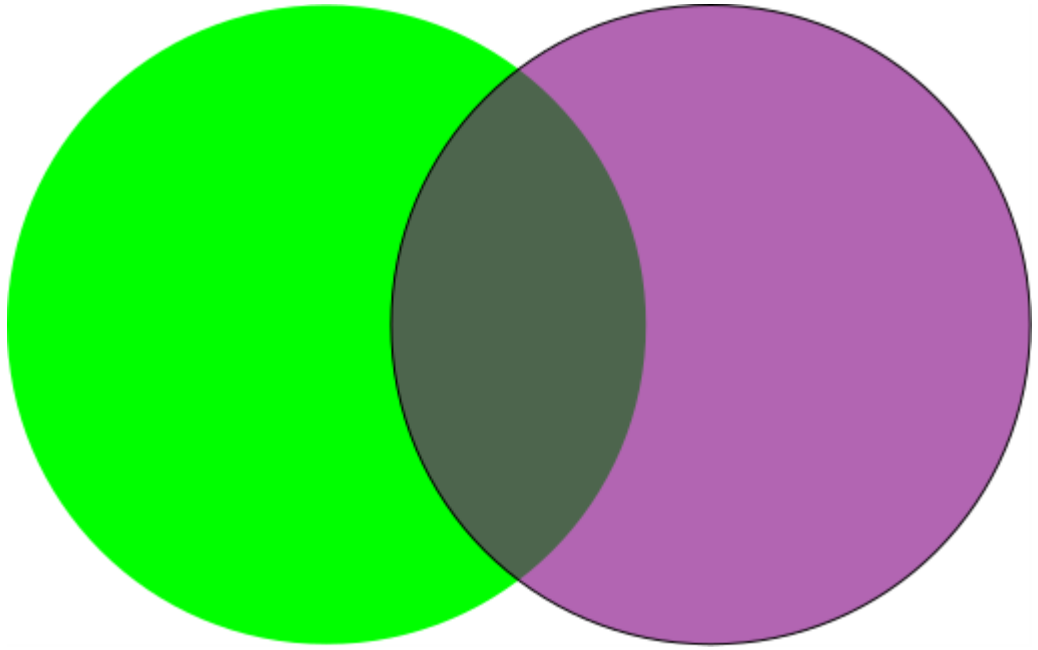
# Java тормозит на тех проблемах, которые никто не решает.

# Java тормозит на тех проблемах, которые никто не решает.

а что, такие проблемы есть?

# Функциональная специализация

**Админы**
**Ops**



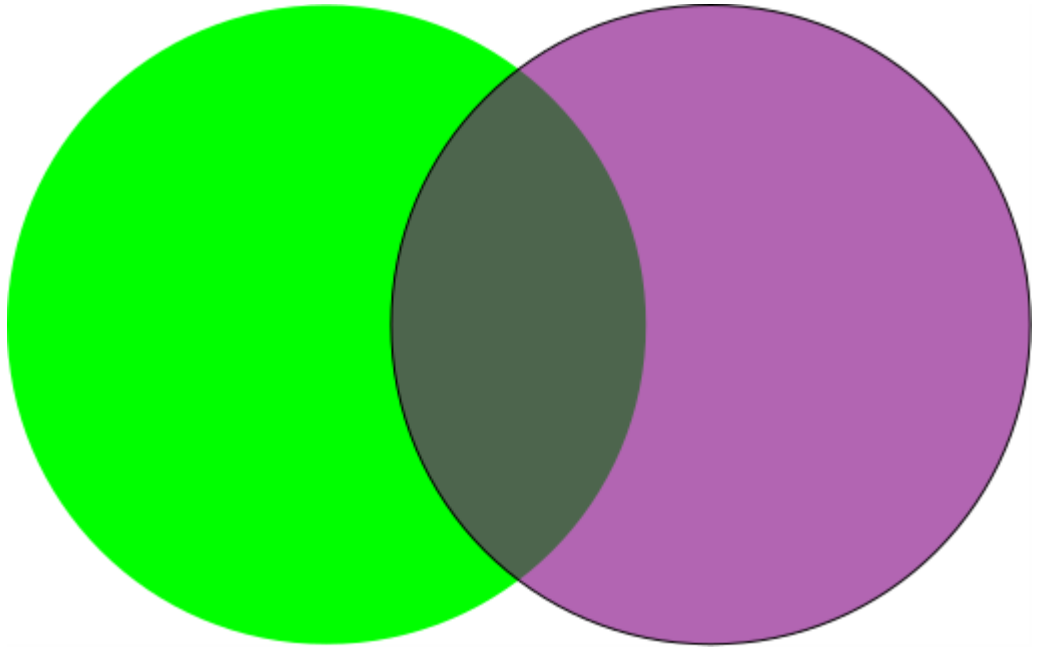
**Программисты**
**Dev**

СБЕРБАНК ТЕХНОЛОГИИ

# Конфликт интересов

**Админы**
**Ops**

Стабильность



**Программисты**
**Dev**

Скорость

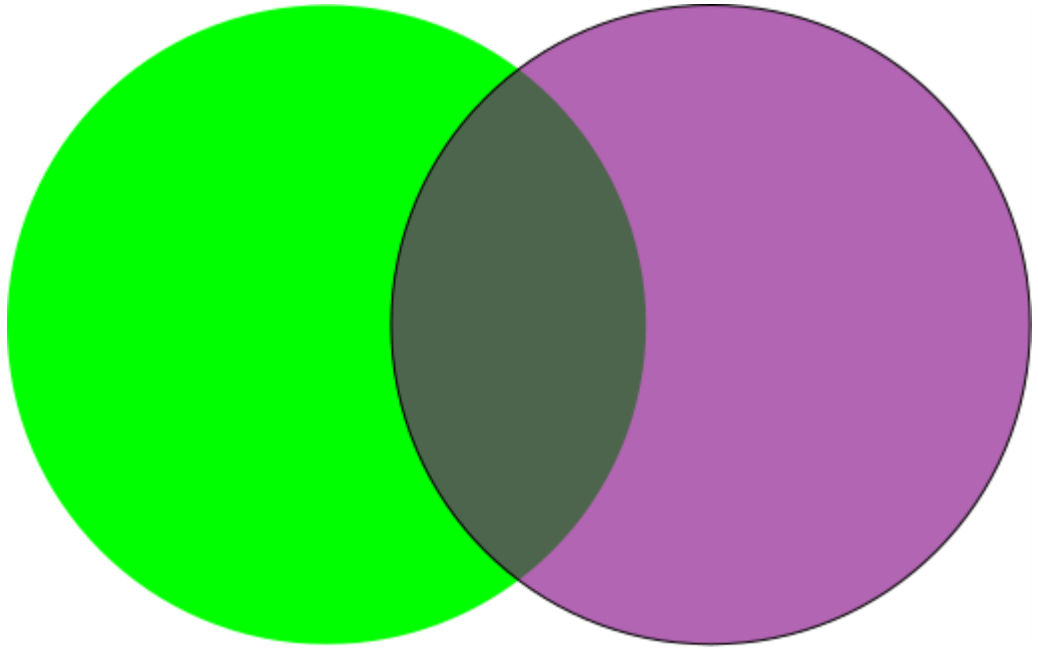
СБЕРБАНК ТЕХНОЛОГИИ

# Конфликт интересов

**Админы**
**Ops**

Стабильность

Регламенты



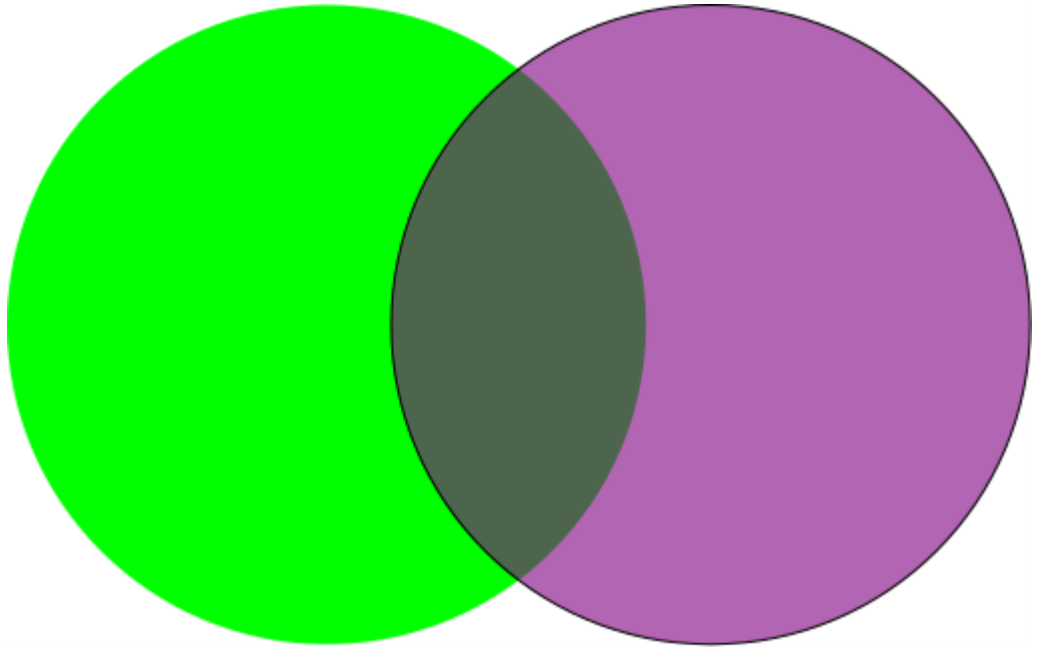
**Программисты**
**Dev**

Скорость

Хотфиксы

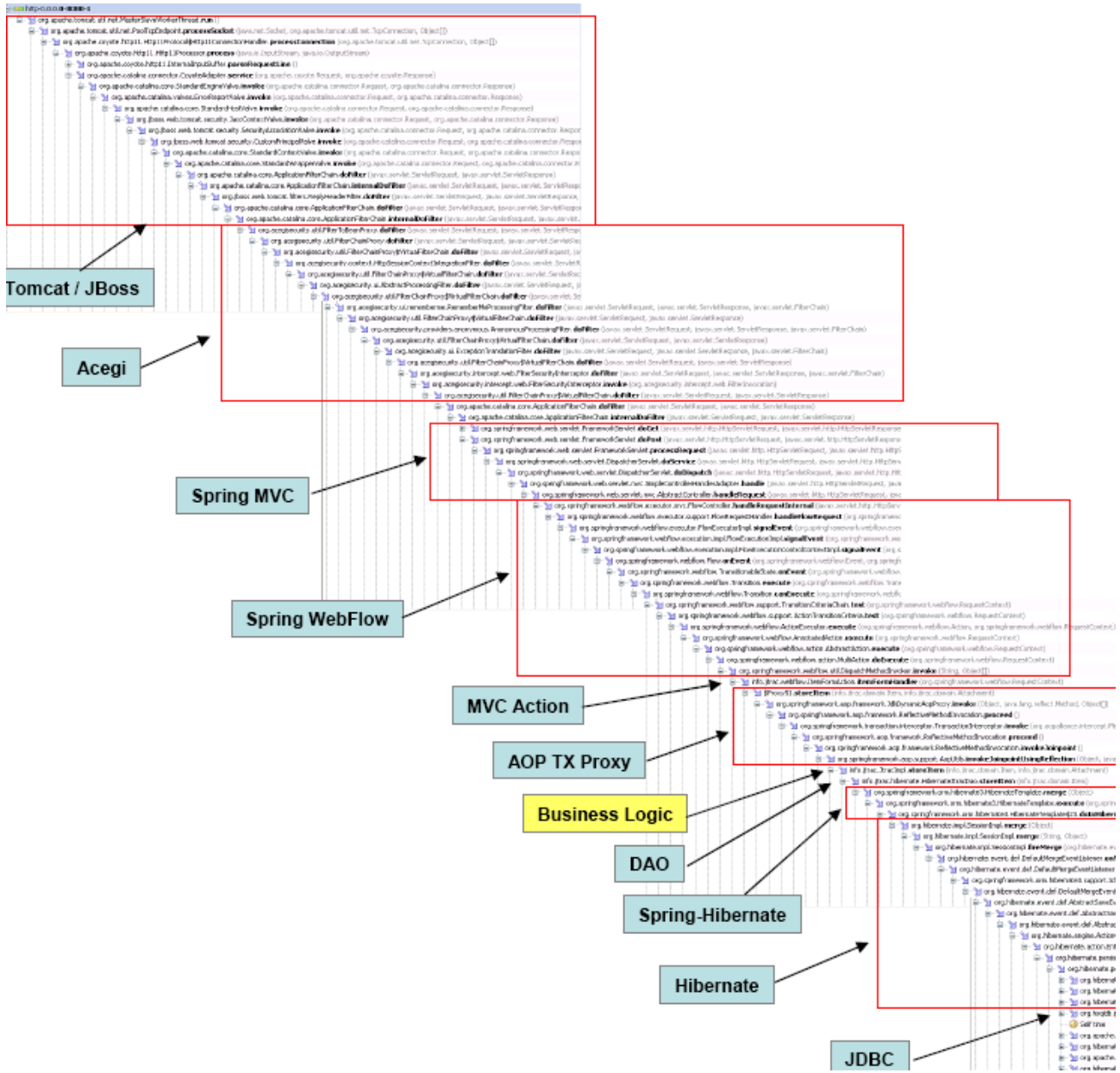
# Агрессия и паника

?



?

# Как выбесить админа?




СБЕРБАНК ТЕХНОЛОГИИ

https://ptrthomas.wordpress.com/2006/06/06/java-call-stack-from-http-upto-jdbc-as-a-picture/

# Clojure

```
shoutout_test/fn (shoutout_test.clj:6)
clojure.test$test_var$fn__7145.invoke (test.clj:701)
clojure.test$test_var.invoke (test.clj:701)
clojure.test$test_all_vars$fn__7149$fn__7156.invoke (test.clj:717)
clojure.test$default_fixture.invoke (test.clj:671)
clojure.test$test_all_vars$fn__7149.invoke (test.clj:717)
clojure.test$default_fixture.invoke (test.clj:671)
clojure.test$test_all_vars.invoke (test.clj:713)
clojure.test$test_ns.invoke (test.clj:736)
clojure.core$map$fn__4207.invoke (core.clj:2487)
clojure.lang.LazySeq.sval (LazySeq.java:42)
clojure.lang.LazySeq.seq (LazySeq.java:60)
clojure.lang.Cons.next (Cons.java:39)
clojure.lang.RT.boundedLength (RT.java:1654)
clojure.lang.RestFn.applyTo (RestFn.java:130)
clojure.core$apply.invoke (core.clj:619)
clojure.test$run_tests.doInvoke (test.clj:751)
clojure.lang.RestFn.applyTo (RestFn.java:137)
clojure.core$apply.invoke (core.clj:617)
user$eval132$fn__187$fn__218.invoke (form-init5113868333329748009.clj:1)
user$eval132$fn__187$fn__188.invoke (form-init5113868333329748009.clj:1)
```



http://yellerapp.com/posts/2015-01-22-upside-down-stacktraces.html

# Golang

```
_/Users/tcrayford/proj/yeller/upsidedown_test.foo3()
        /Users/tcrayford/proj/yeller/upsidedown/example_test.go:16 +0x1b
_/Users/tcrayford/proj/yeller/upsidedown_test.foo2()
        /Users/tcrayford/proj/yeller/upsidedown/example_test.go:12 +0x1b
_/Users/tcrayford/proj/yeller/upsidedown_test.foo1()
        /Users/tcrayford/proj/yeller/upsidedown/example_test.go:8 +0x1b
_/Users/tcrayford/proj/yeller/upsidedown_test.TestPanic(0x2082ae000)
        /Users/tcrayford/proj/yeller/upsidedown/example_test.go:168 +0x1b
testing.tRunner(0x2082ae000, 0x202f10)
        /usr/local/Cellar/go/1.4/libexec/src/testing/testing.go:447 +0xbf
created by testing.RunTests
        /usr/local/Cellar/go/1.4/libexec/src/testing/testing.go:555 +0xa8b

goroutine 1 [chan receive]:
testing.RunTests(0x190260, 0x202f10, 0x1, 0x1, 0x524469b40b5e0501)
        /usr/local/Cellar/go/1.4/libexec/src/testing/testing.go:556 +0xad6
testing.(*M).Run(0x2082820a0, 0x20f380)
        /usr/local/Cellar/go/1.4/libexec/src/testing/testing.go:485 +0x6c
main.main()
        _/Users/tcrayford/proj/yeller/upsidedown/_test/_testmain.go:52 +0x1d5
exit status 2
FAIL    _/Users/tcrayford/proj/yeller/upsidedown        0.008s
```



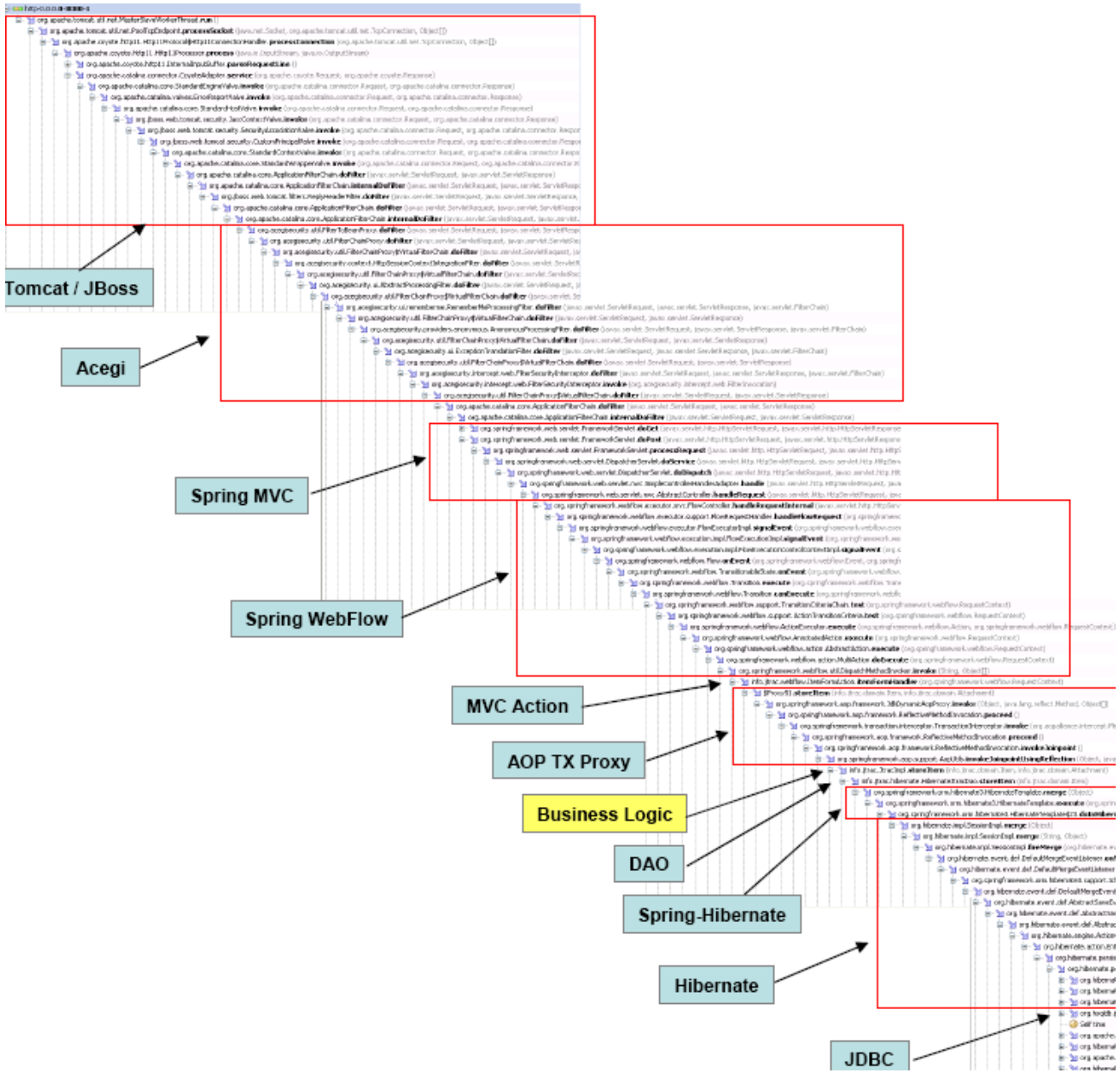
http://yellerapp.com/posts/2015-01-22-upside-down-stacktraces.html

# Ruby

```
from foo.rb:158:in `foo40'
from foo.rb:154:in `foo39'
from foo.rb:150:in `foo38'
from foo.rb:146:in `foo37'
from foo.rb:142:in `foo36'
from foo.rb:138:in `foo35'
from foo.rb:134:in `foo34'
from foo.rb:130:in `foo33'
from foo.rb:126:in `foo32'
from foo.rb:122:in `foo31'
from foo.rb:118:in `foo30'
from foo.rb:114:in `foo29'
from foo.rb:110:in `foo28'
from foo.rb:106:in `foo27'
from foo.rb:102:in `foo26'
from foo.rb:98:in `foo25'
from foo.rb:94:in `foo24'
from foo.rb:90:in `foo23'
```



http://yellerapp.com/posts/2015-01-22-upside-down-stacktraces.html



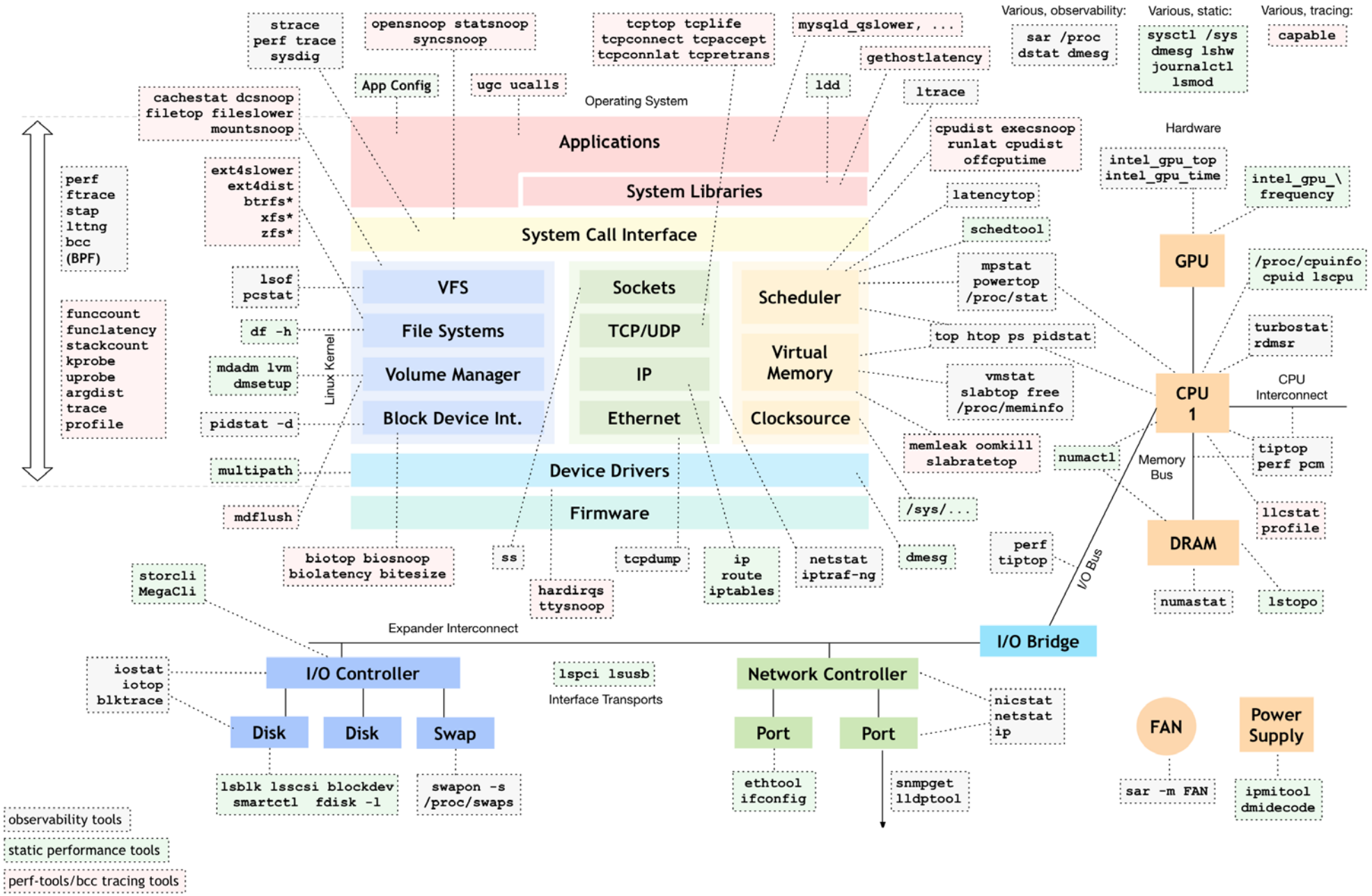
https://ptrthomas.wordpress.com/2006/06/06/java-call-stack-from-http-upto-jdbc-as-a-picture/

# Как выбесить программиста?

# Как выбесить программиста

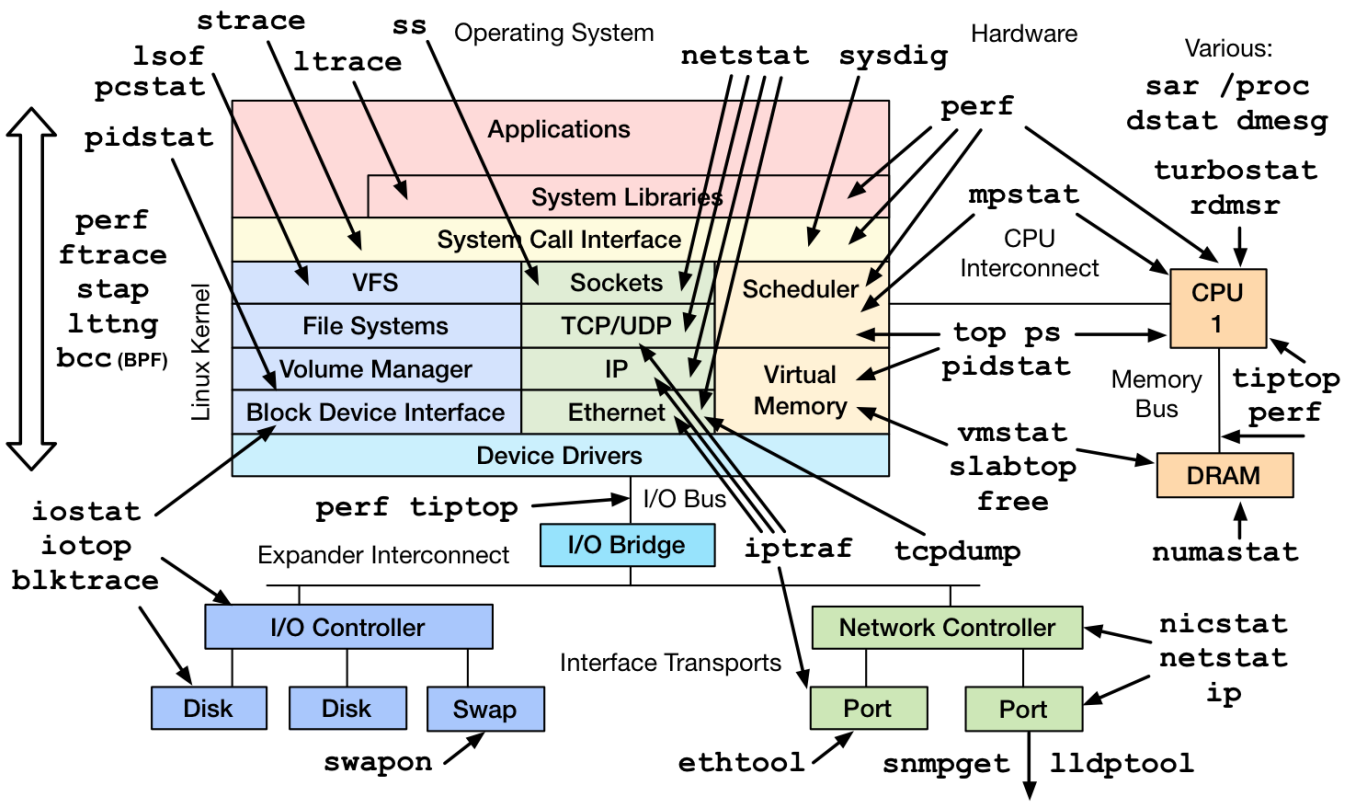
Не понимаю, как вы вообще
живете без депрессии.
Чем занимаетесь.
Хохочете целыми днями?

*Франц Кафка*




СБЕРБАНК ТЕХНОЛОГИИ

http://www.brendangregg.com/linuxperf.html



СБЕРБАНК ТЕХНОЛОГИИ

http://www.brendangregg.com/linuxperf.html

# Только «хорошие» программисты?

Разработчики OpenJDK достаточно хороши?

СБЕРБАНК ТЕХНОЛОГИИ

https://www.youtube.com/watch?v=3pvS2Cctx3o

web code dev write music λ χ φ #irc []= andy - File Manager

WiFi: [ TP-LINK_965F18 ] | Lang: [en] | Power: [57] | Vol: 100% 100% | 5 Jun 2016 Sun | 15:42

```
andy@dwmX-jessie ~/clion-2016.1.2/bin
 % ./clion.sh
Jun 05, 2016 3:42:09 PM java.util.prefs.FileSystemPreferences$6 run
WARNING: Prefs file removed in background /home/andy/.java/.userPrefs/prefs.xml
```

File Edit View Go Help

Open Parent Home /home/andy/

DEVICES

File System

PLACES

andy

.9menu .adobe .aptitude

o15 .cache .CLion2016.1

ig .dbus .emacs.d

s .gconf .gnome

og .grc_gnuradio .gstreamer-0.10

.icons .java

.jssc .local .macromedia

.mozilla .mpd .mplayer

146 items (86.2 MB), Free space: 202.4 GB

https://www.linux.org.ru/forum/development/13203322

# Пример: Re-parenting window manager

- **X Window System** – система, которая рисует картинки на дисплей, обрабатывает мышку и клавиатуру
- **Window Manager** – это средство управления окнами, которое НЕ взаимодействует с железом (железо – дело display server)
- **Re-parenting window manager** – это Window Manager, который является родителем/хозяином для всех окон в системе
- Все современные Window Managers – re-parenting.
- **За исключением**: **dwm**, cwm, PLWM, ratpoison, stumpwm, xmonad

Последовательно показывать

```java
109 @    public String toString() {
110         switch  (WMID) {
111           case NO_WM:
112               return "NO WM";
113           case OTHER_WM:
114               return "Other WM";
115           case OPENLOOK_WM:
116               return "OPENLOOK";
117           case MOTIF_WM:
118               return "MWM";
119           case CDE_WM:
120               return "DTWM";
121           case ENLIGHTEN_WM:
122               return "Enlightenment";
123           case KDE2_WM:
124               return "KWM2";
125           case SAWFISH_WM:
126               return "Sawfish";
127           case ICE_WM:
128               return "IceWM";
129           case METACITY_WM:
130               return "Metacity";
131           case COMPIZ_WM:
132               return "Compiz";
133             case UNITY_COMPIZ_WM:
134               return "Unity Compiz";
135           case LG3D_WM:
136               return "LookingGlass";
137           case CWM_WM:
138               return "CWM";
139           case MUTTER_WM:
```

СБЕРБАНК ТЕХНОЛОГИИ

короче

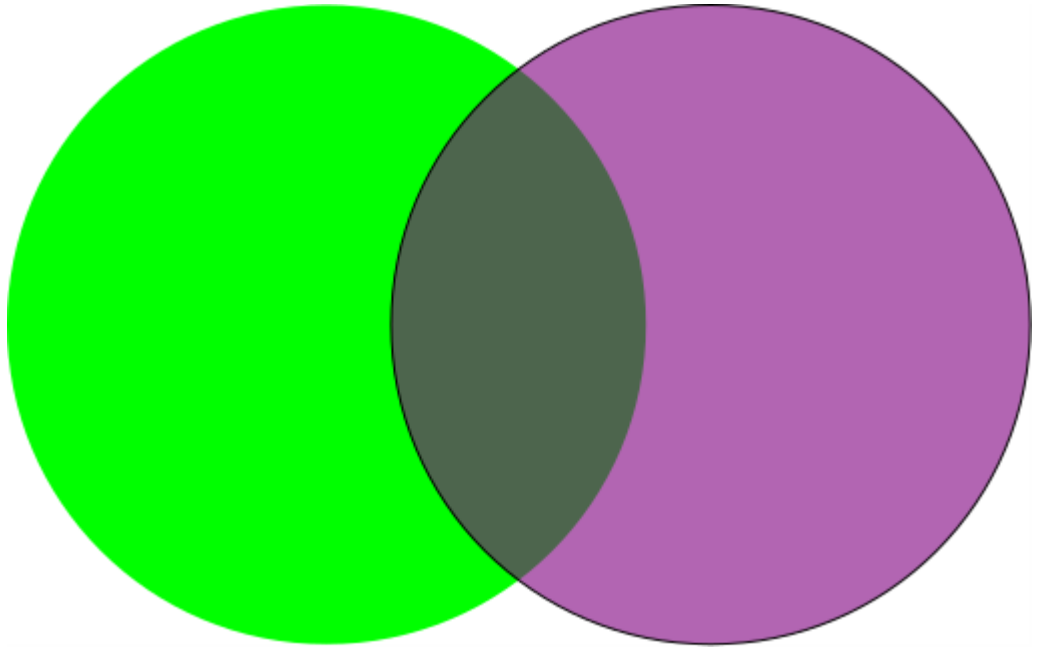
```
163
164         static Insets normalize(Insets insets) {
165             if (insets.top > 64 || insets.top < 0) {
166                 insets.top = 28;
167             }
168             if (insets.left > 32 || insets.left < 0) {
169                 insets.left = 6;
170             }
171             if (insets.right > 32 || insets.right < 0) {
172                 insets.right = 6;
173             }
174             if (insets.bottom > 32 || insets.bottom < 0) {
175                 insets.bottom = 6;
176             }
177             return insets;
178         }
179
```

```
782         if (isEnlightenment()) {
783             awt_wmgr = XWM.ENLIGHTEN_WM;
784         } else if (isMetacity()) {
785             awt_wmgr = XWM.METACITY_WM;
786         } else if (isMutter()) {
787             awt_wmgr = XWM.MUTTER_WM;
788         } else if (isSawfish()) {
789             awt_wmgr = XWM.SAWFISH_WM;
790         } else if (isKDE2()) {
791             awt_wmgr =XWM.KDE2_WM;
792         } else if (isCompiz()) {
793             awt_wmgr = XWM.COMPIZ_WM;
794         } else if (isLookingGlass()) {
795             awt_wmgr = LG3D_WM;
796         } else if (isCWM()) {
797             awt_wmgr = CWM_WM;
798         } else if (doIsIceWM && isIceWM()) {
799             awt_wmgr = XWM.ICE_WM;
800         } else if (isUnityCompiz()) {
801             awt_wmgr = XWM.UNITY_COMPIZ_WM;
802         }
```

# sun.awt.X11.XWM

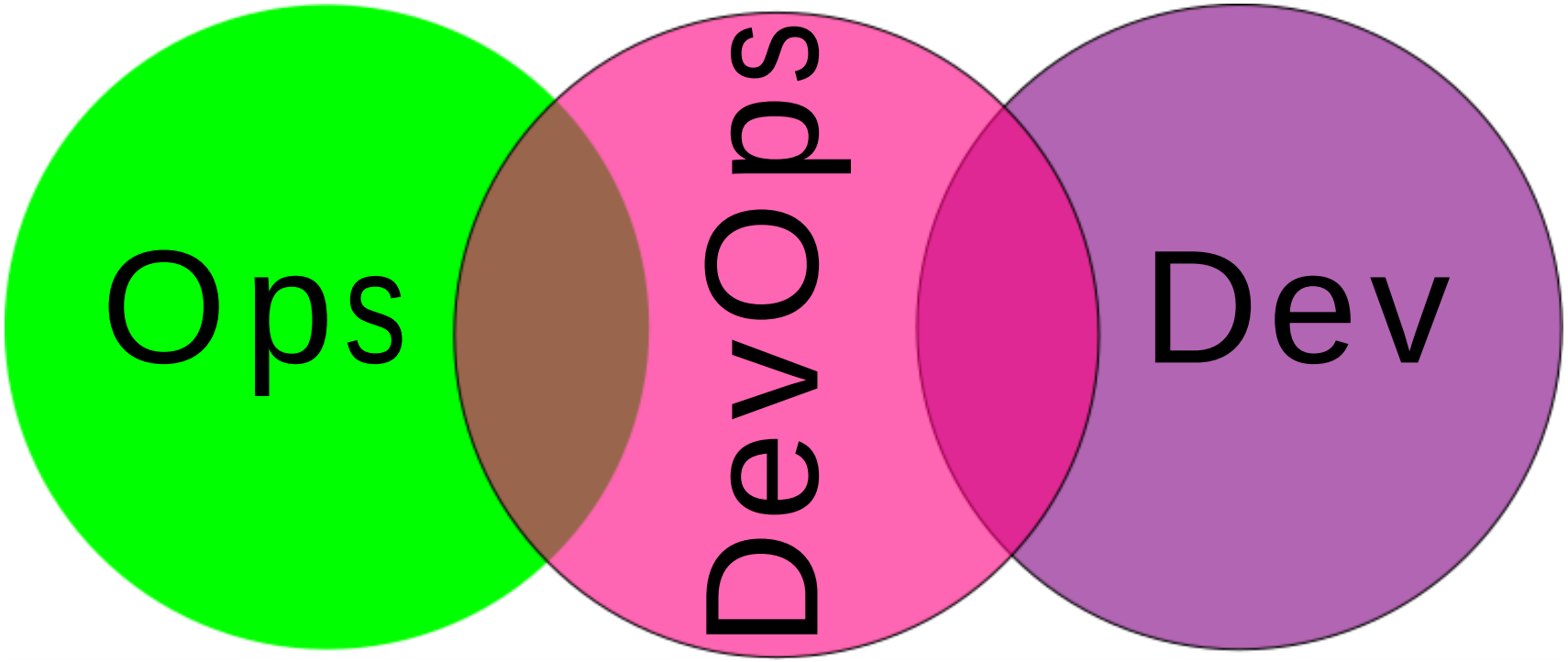
```
static int awtWMNonReparenting = -1;
static boolean isNonReparentingWM() {
    if (awtWMNonReparenting == -1) {
        awtWMNonReparenting = (XToolkit.getEnv("_JAVA_AWT_WM_NONREPARENTING") != null) ? 1 : 0;
    }
    return (awtWMNonReparenting == 1 || XWM.getWMID() == XWM.COMPIZ_WM
            || XWM.getWMID() == XWM.LG3D_WM || XWM.getWMID() == XWM.CWM_WM);
}
```

# Проблема



СБЕРБАНК ТЕХНОЛОГИИ

# Решение



# Кому это может понравиться?

- Администраторы чувствуют, что не делают ничего нового
- Программисты не видят результатов (пишут «в стол»)
- Девопсы постоянно пишут новое, и тут же видят результат

# Логика повествования

- ~~Психологическая часть вопроса~~
- ~~Покажу что девопсы нам реально нужны~~
- Техническая часть вопроса и типичные задачи
- Собрать всё вместе и сделать какое-то заключение

# Что делает девопс на джава-конференции?

Это же конференция для инженеров-разработчиков?

СБЕРБАНК ТЕХНОЛОГИИ

# Что делает девопс на джава-конференции?

Это же конференция для инженеров-разработчиков?

▸▸▸ Девопс и есть инженер-разработчик

СБЕРБАНК ТЕХНОЛОГИИ

# Область определения

Типичное Java-приложение

СБЕРБАНК ТЕХНОЛОГИИ

Фронт: HTML, JavaScript

# Область определения

Типичное Java-приложение

СБЕРБАНК ТЕХНОЛОГИИ

# Область определения

Типичное Java-приложение

Фронт: HTML, JavaScript

Транспорт: HTTP/REST, SOAP

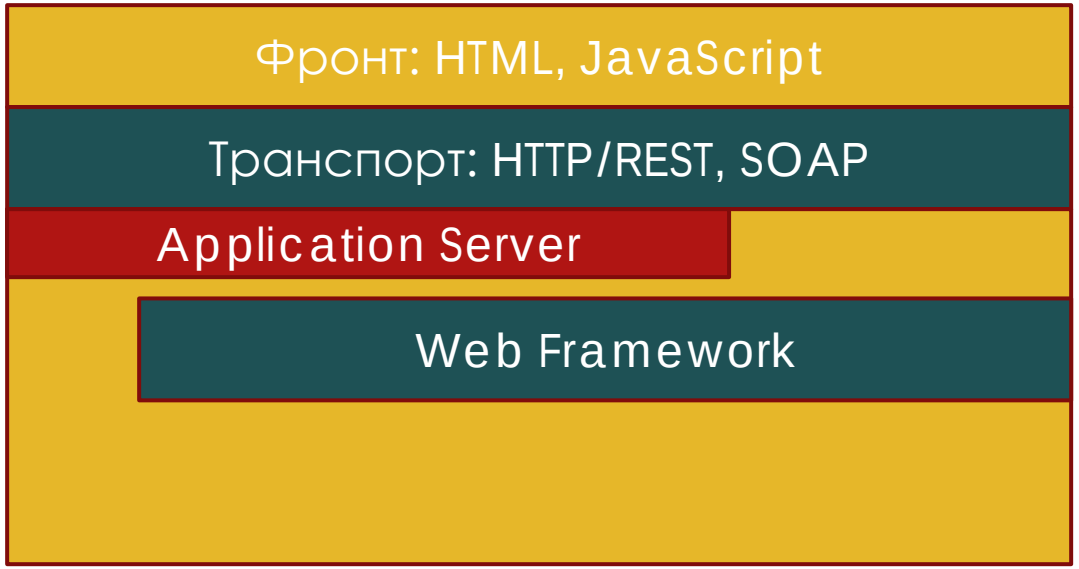
СБЕРБАНК ТЕХНОЛОГИИ

# Область определения

Типичное Java-приложение

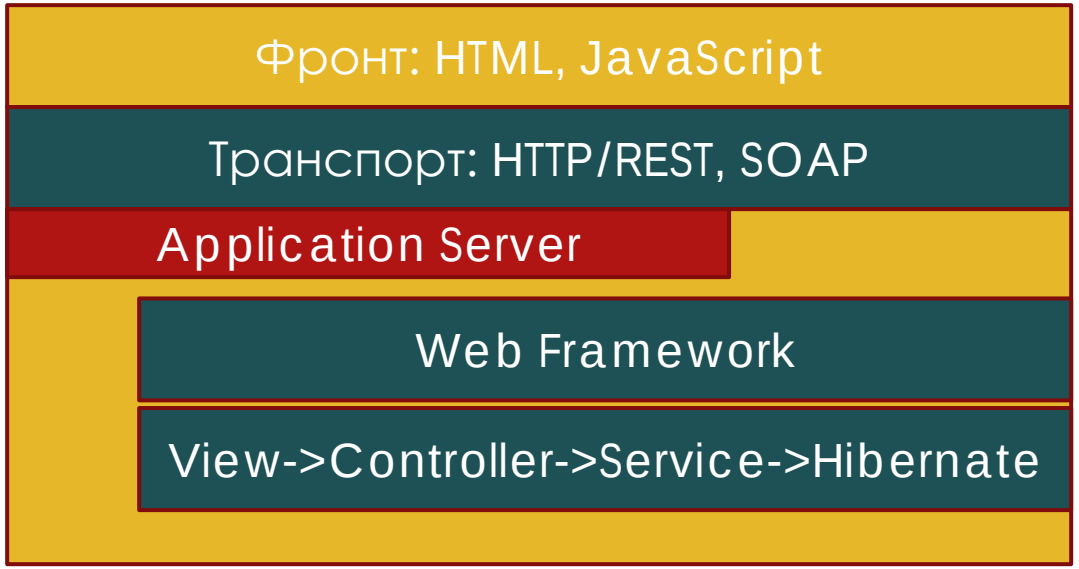
# Область определения

Типичное Java-приложение



# Область определения

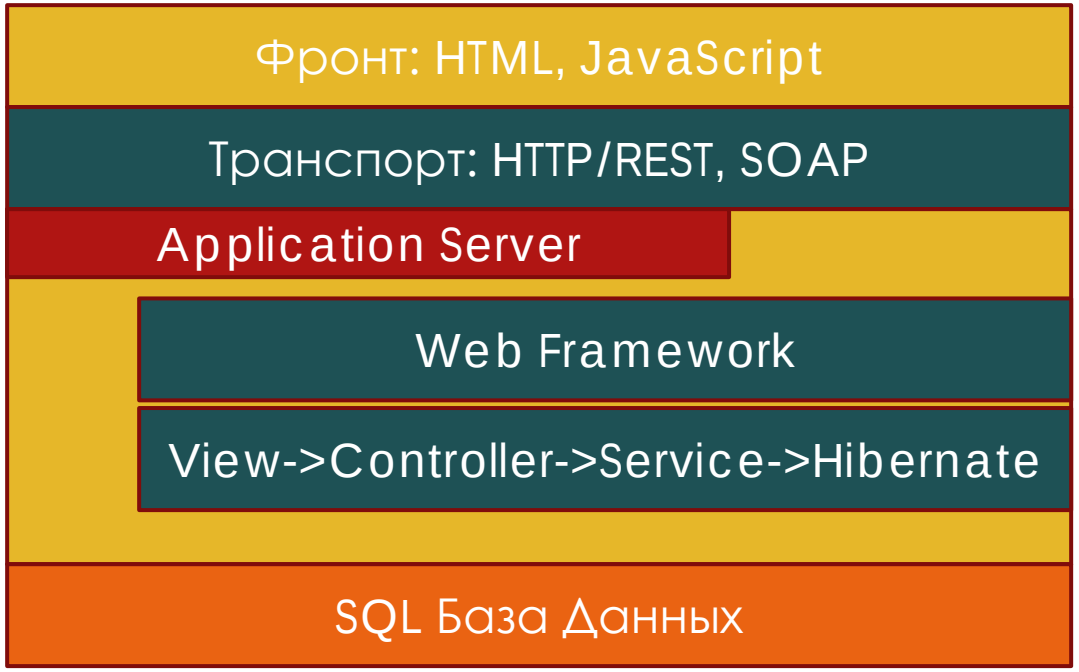
Типичное Java-приложение




СБЕРБАНК ТЕХНОЛОГИИ

# Область определения

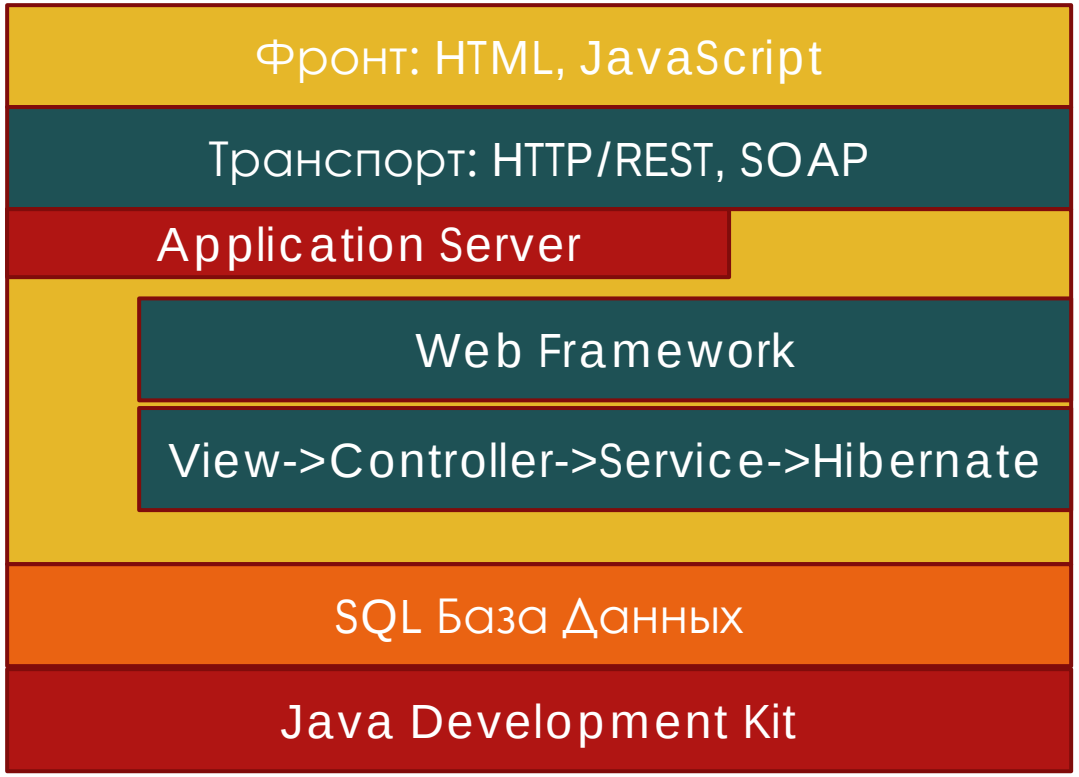
Типичное Java-приложение




СБЕРБАНК ТЕХНОЛОГИИ

# Область определения

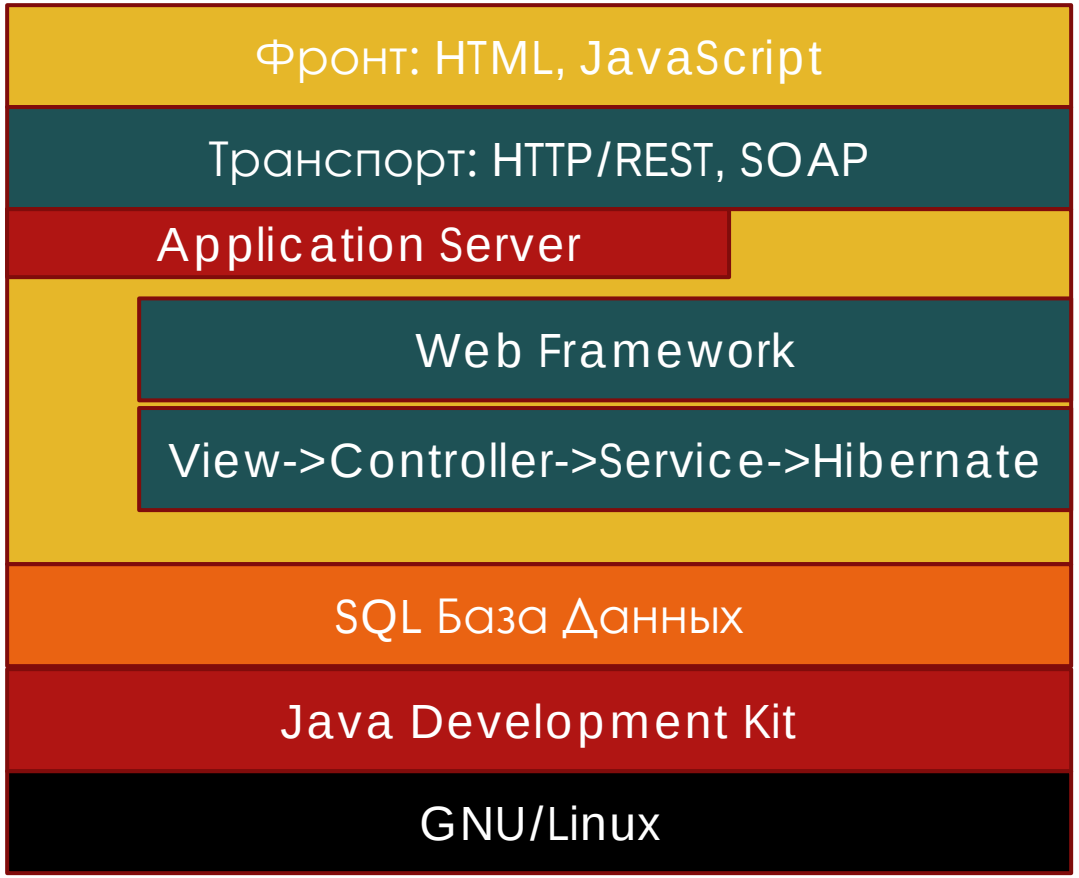
Типичное Java-приложение




СБЕРБАНК ТЕХНОЛОГИИ

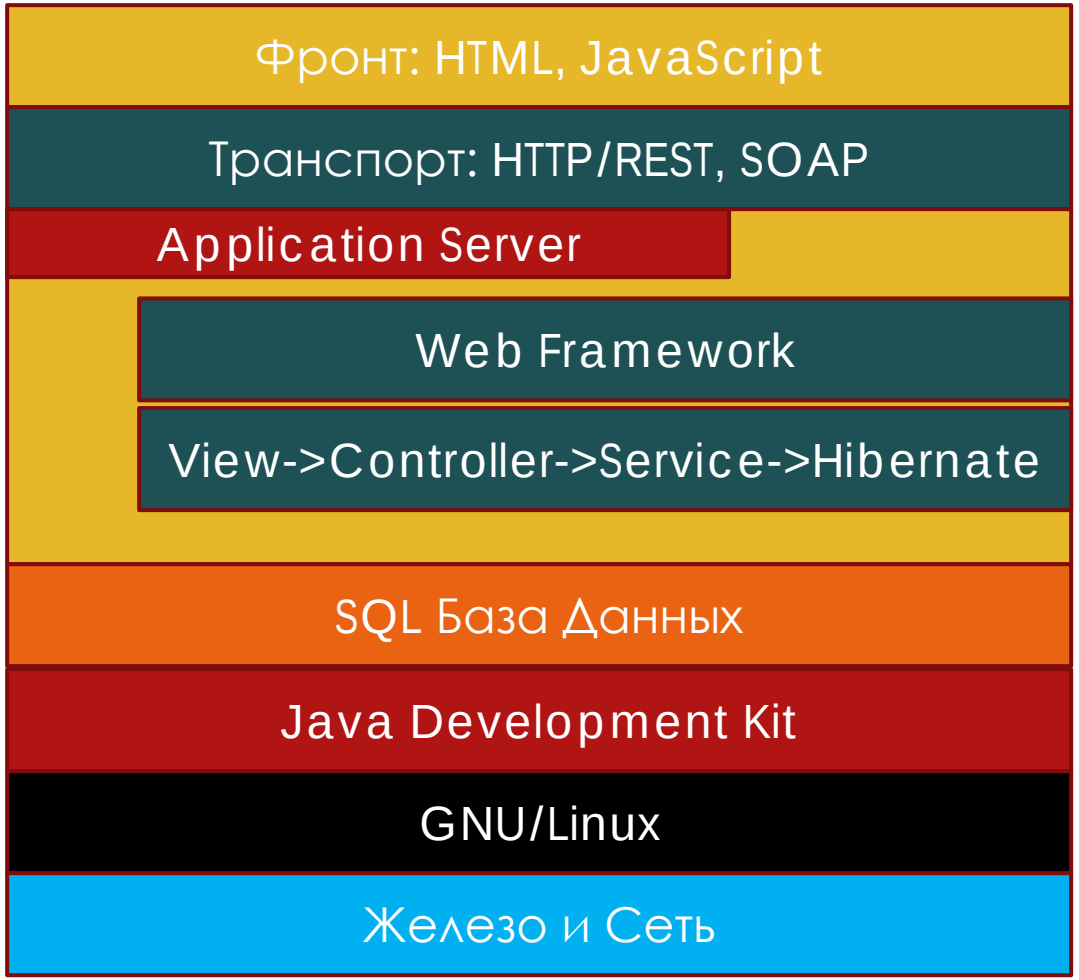
# Область определения

Типичное Java-приложение



# Область определения

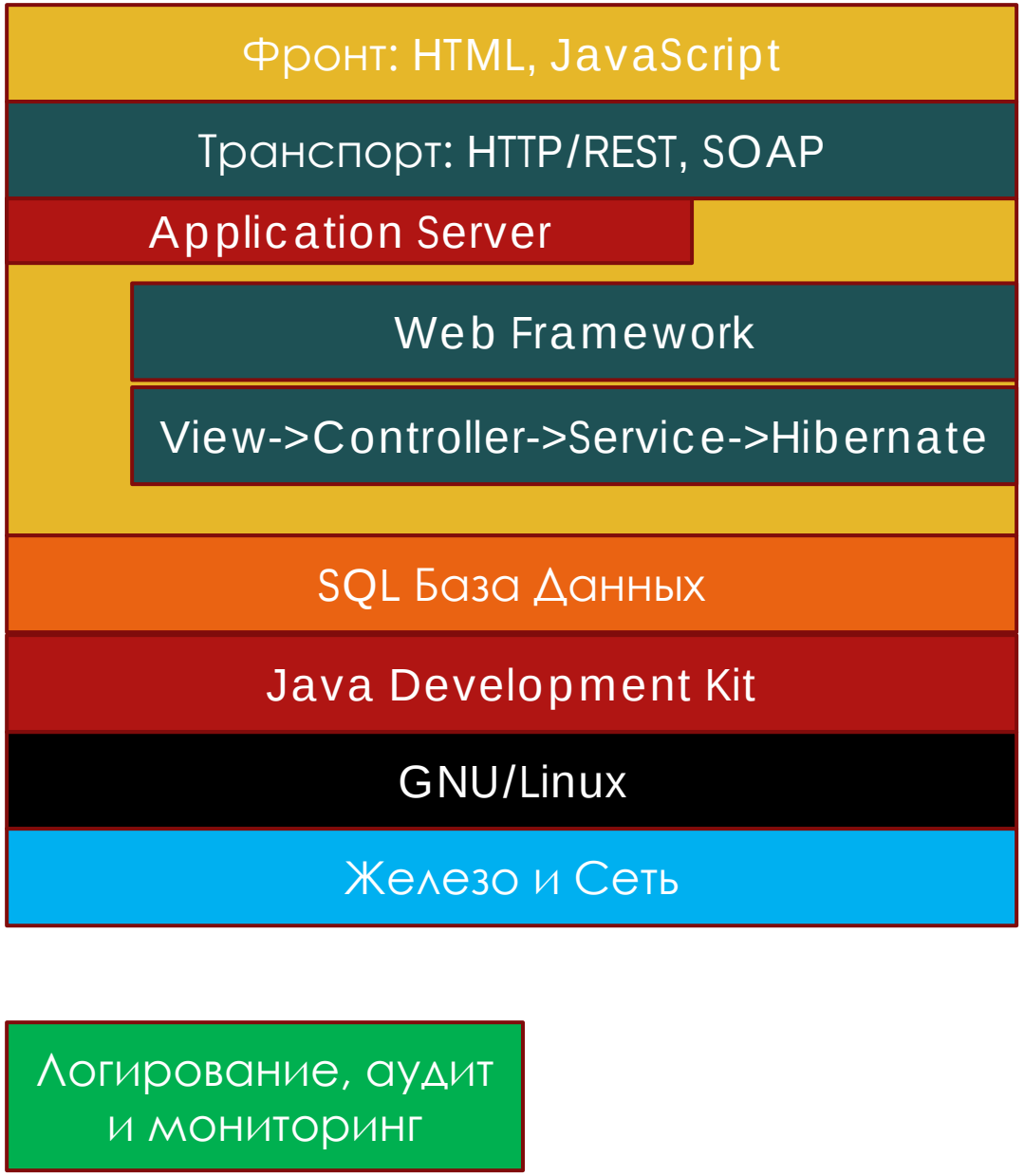
Типичное Java-приложение




СБЕРБАНК ТЕХНОЛОГИИ

# Область определения

Типичное Java-приложение




СБЕРБАНК ТЕХНОЛОГИИ

# Область определения

Типичное Java-приложение

СБЕРБАНК ТЕХНОЛОГИИ

# Кейс 1: <% spoiler alert %>

**Дано**

- Многомодульное приложение (типа порталов, но самописное)
- CI/CD – поставку снапшота можно делать мгновенно

**Проблема**

- Каждое следующее обновление делается чуть дольше

# Кейс 1: Жирная JVM

**Проблема**

1. ~100 модулей
2. Приложение разрослось на 500G хипа
3. Каждое следующее обновление делает его больше

# Кейс 1: Жирная JVM

**Жирная JVM – это хорошо?**

# Кейс 1: Жирная JVM

- Compressed Oops
  https://wiki.openjdk.java.net/display/HotSpot/CompressedOops

# CopmpressedOops

**96 бит**

Integer на 32-битной JVM

**160 бит**

Integer на 64-битной JVM, coops выключены

**128 бит**

Integer на 64-битной JVM, coops включены

# CopmpressedOops

```
[node04] ~ ➜ jdk1.7.0_51/bin/java -Xms31g -Xmx31g -Xmn50m
Memory Total Memory (in bytes): 33279705088
Free Memory (in bytes): 33278908064
Max Memory (in bytes): 33279705088
Elements created and added to LinkedList: 587889429
```

```
[node04] ~ ➜ jdk1.7.0_51/bin/java -Xms32g -Xmx32g -Xmn50m
Memory Total Memory (in bytes): 34353446912
Free Memory (in bytes): 34352649800
Max Memory (in bytes): 34353446912
Elements created and added to LinkedList: 385481085
```

▸▸▸ Хип на 47G меньше хипа на 32G



https://blog.codecentric.de/en/2014/02/35gb-heap-less-32gb-java-jvm-memory-oddities/

# Кейс 1: Жирная JVM

- Compressed Oops
  https://wiki.openjdk.java.net/display/HotSpot/CompressedOops
- Запускается несколько часов

# Кейс 1: Жирная JVM

- Compressed Oops
  https://wiki.openjdk.java.net/display/HotSpot/CompressedOops
- Запускается несколько часов
- Нужно держать hot standby копию на случай отказа (не дешево)

# Кейс 1: Жирная JVM

- Compressed Oops
  https://wiki.openjdk.java.net/display/HotSpot/CompressedOops
- Запускается несколько часов
- Нужно держать hot standby копию на случай отказа (не дешево)
- Если упало что-то одно — надо перезапускать всё

# Кейс 1: Жирная JVM

- Compressed Oops
  https://wiki.openjdk.java.net/display/HotSpot/CompressedOops
- Запускается несколько часов
- Нужно держать hot standby копию на случай отказа (не дешево)
- Если упало что-то одно — надо перезапускать всё
- Full GC по терабайту памяти займет очень долго (10 минут?)

# Кейс 1: Жирная JVM

- Compressed Oops
  https://wiki.openjdk.java.net/display/HotSpot/CompressedOops
- Запускается несколько часов
- Нужно держать hot standby копию на случай отказа  (не дешево)
- Если упало что-то одно — надо перезапускать всё
- Full GC по терабайту памяти займет очень долго (10 минут?)
- Всем нужно покупать лицензию на JRebel

# Кейс 1: Жирная JVM

- Compressed Oops
  https://wiki.openjdk.java.ne CompressedOops
- Запускается несколько
- Нужно держать hot star каза (не дешево)
- Если упало что-то одно всё
- Full GC по терабайту пам го (10 минут?)

# Кейс 1: Жирная JVM: Итоги

- Объяснить и донести, почему плохо
  - Девопс должен знать джава-платформу наравне с разработчиками, чтобы иметь возможность аргументированно отстаивать свое мнение
- Предоставить альтернативные решения
  - Попилить всё на микросервисы?
- Запретить использовать жирную JVM
- **Нужно иметь административные права на вмешательство в архитектуру и код**
  - Забраковать нерачбоее решение, не пустить его в прод
  - Написать тот код, которым никто не хочет заниматься

# Кейс 2: Тормозной Дженкинс

# Кейс 2: Тормозной Дженкинс

**Типичный Java-монстр ☺**
**Пример legacy приложения**

# Кейс 2: Тормозной Дженкинс

**Проблема**:

установка плагинов занимает огромное количество времени

СБЕРБАНК ТЕХНОЛОГИИ

# Кейс 2: Тормозной Дженкинс

**Следствие установило**

- Не умеет на лету устанавливать плагины, нужна перезагрузка. (кривая архитектура)
- Чем больше плагинов, тем дольше перезагрузка (вплоть до часов)
- А в крупной компании очень-очень много плагинов
- Администраторы откладывают перезагрузки насколько можно

# Кейс 2: Тормозной Дженкинс

**Решение**

- Docker (каждой команде по докеру)
- Написание своего зеркала и менеджера плагинов
  - Не умеет работать в закрытой корпоративной сети
- Jenkins as a Service (Docker+Mesos)
  - Доклад Толкачева и Тарасова: https://www.youtube.com/watch?v=xPf32v8kOe0

# Кейс 2: Тормозной Дженкинс: Итоги

- У каждого в шкафу лежат свои легаси приложения
- Нужно иметь скилл, чтобы написать зеркало и менеджер пакетов
  - Именно поэтому девопс – это в первую очередь инженер-разработчик
- **Нужно иметь административные права на вмешательство в администрирование**

# Кейс 3: Логирование исчезло

**Дано**

- На локальном компьютере разработчика логирование работает
- На проде и тесте даже не создаются файлы с логом
- **Подсказка:** Россия
- **Подсказка:** В приложении используется криптография

# Кейс 3: Логирование исчезло

**Проблема**

Использование отладчика вместо логирования

- не альтерантива по смыслу
- занимает очень много времени и сил разработчика
- требует специальных настроек безопасности

# Кейс 3: Логирование исчезло

**Следствие установило**

- В дистрибутиве JDK на проде лежит commons-logging.jar, что сводит с ума логеры
  - jre/lib/ext

# Кейс 3: Логирование исчезло

**Следствие установило**

- В ходе отладки оказалось, что в дистрибутиве JDK на проде/тесте лежит commons-logging.jar
  - jre/lib/ext
- В не-dev окружении используется Крипто Про JCP
- КриптоПро устанавливается подкладыванием JARок в JDK/JRE, и ему принадлежит commons-logging.jar
- Это Extension Classloader, поэтому простым способом ничего не починишь
- **Решение**: самый простой способ – использовать другую систему логирования, т.е. переписать код и доложить в него костылей

# Кейс 3: Логирование исчезло: Итоги

- Девопс должен уметь отлаживать любую часть Java стека, и чинить её на лету
- Для этого девопс должен быть фуллстек-разработчиком
- Как любой фулстек, должен постоянно поддерживать свои знания о непрерывно изменяющемся мире

# Сворачиваемся

У нас есть человек-девопс, который умеет:

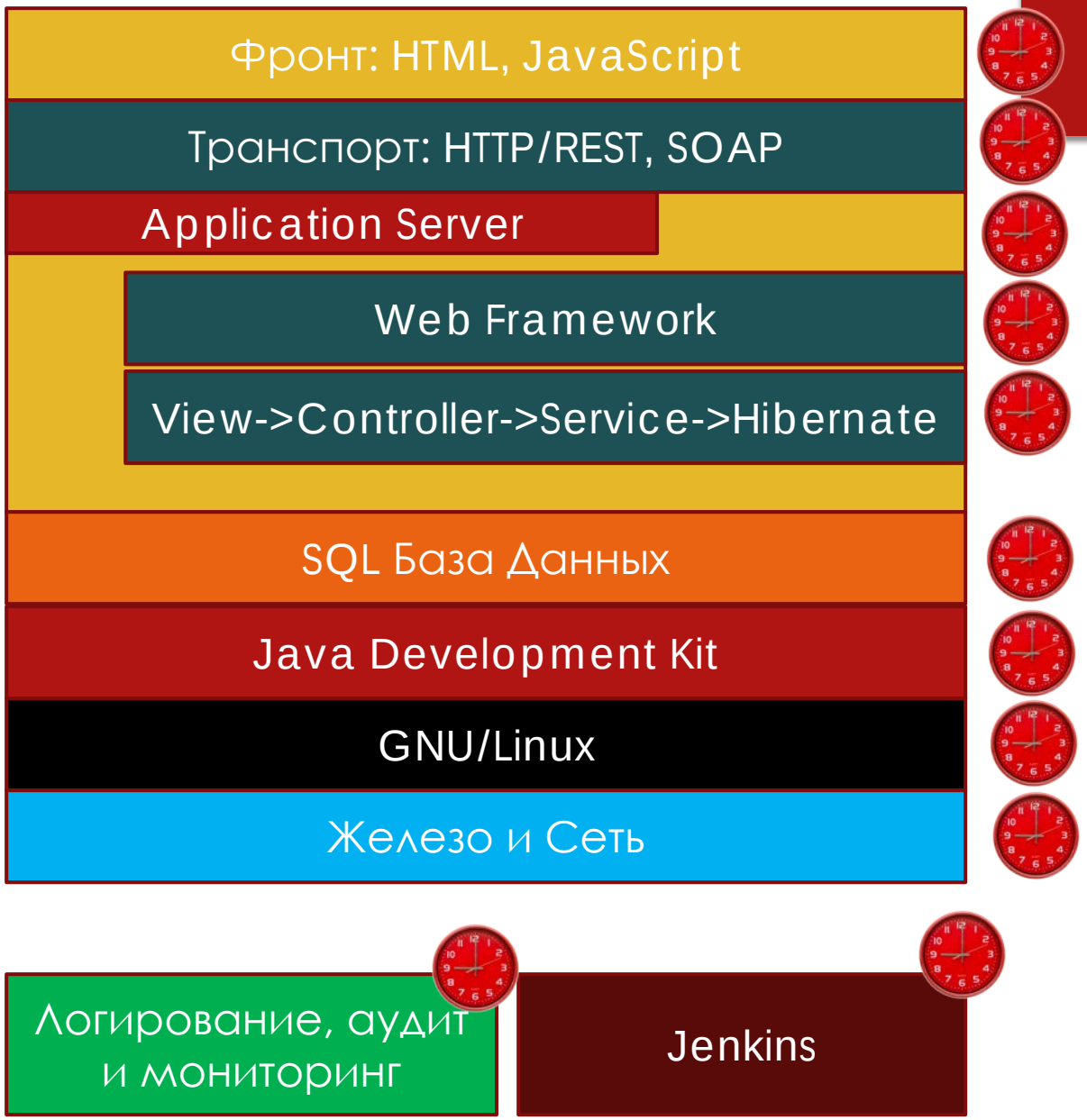
- Писать код
- Отлаживать весь стек технологий
- Администрировать

Обладает административными привилегиями на создание

- Изменение и ревью любого кода
- Вмешательство в работу администраторов

# Область торможения

Если девопс – фулстек разработчик, то какую часть стека ему нужно разгонять?



# Локальное решение



Spring Cloud

+

NETFLIX OSS

# Технологическое ядро

# Логика повествования

- ~~Психологическая часть вопроса~~
- ~~Покажу что девопсы нам реально нужны~~
- ~~Техническая часть вопроса и типичные задачи~~
- Собрать всё вместе и сделать какое-то заключение

# Итоги

- **Девопс – это в первую очередь фулстек инженер-разработчик**
- Роль девопса можно рассматривать как развитие роли программиста или аминистриратора. Это зона роста.
- Девопс везде нужен и важен, и никогда не пропадет
  - Крупные компании типа Google, Facebook, Amazon, Сбербанк-Технологии
  - Особенно в свете того, что нужно поддерживать обновленный Java стек
- Призываю не бояться лезть в код, создавать новые технологии и делать Java лучше