



SPEAKER

DOMINIK KAPUSTA

BIG CHANGES AND NO TESTS

Agenda

Progressive rollout

Fine-grained monitoring

The “Rollout Service”

Consistency checks

Automation

Benefits for everyone

Takeaways

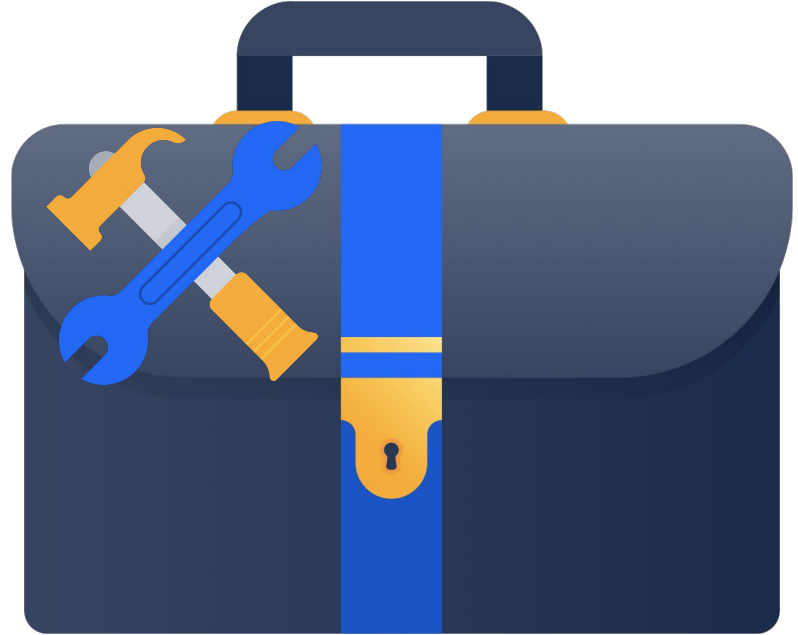
Jira Cloud



hundreds of thousands of companies



No rocket science



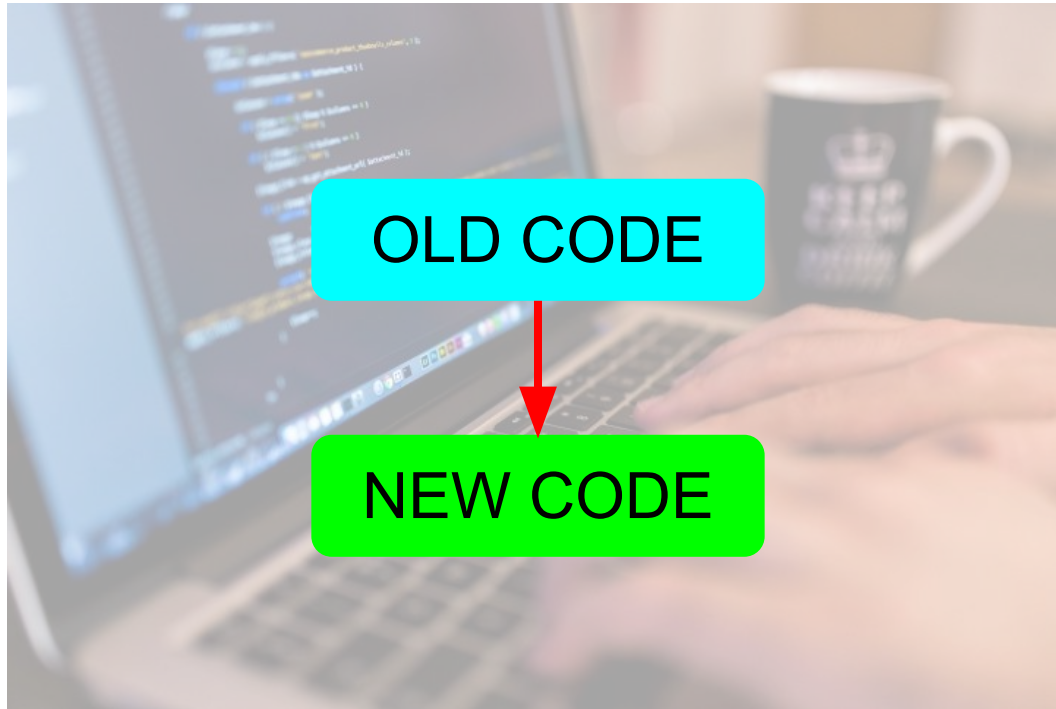
What do we do as devs?



Source: <https://pixnio.com/objects/computer/programming-code-programmer-coding-coffee-cup-computer-copy-hands-computer-keyboard>



What do we do as devs?



Old school

```
if (i > 0) {  
    logMessage.append('\n');  
}  
logMessage.append(deepDiffs.get(i).toString());  
if (i ≥ maxNrOfDiffsToLog - 1) {  
    logMessage.append(REST_SKIPPED_MESSAGE);  
    break;  
}  
}
```

OLD CODE

```
if (logMessage.length() > maxLogMessageLength) {  
    String trimmed = "(trimmed)";  
    logMessage.delete(Math.max(0, maxLogMessageLength - trimmed.length()));  
    logMessage.append(trimmed);  
}  
  
logger.info(logMessage.toString());
```

v. 23

```
if (i > 0) {  
    logMessage.append('\n');  
}  
logMessage.append(deepDiffs.get(i).toString());  
if (i ≥ maxNrOfDiffsToLog - 1) {  
    logMessage.append(REST_SKIPPED_MESSAGE);  
    break;  
}  
}
```

NEW CODE

```
if (logMessage.length() > maxLogMessageLength) {  
    String trimmed = "(trimmed)";  
    logMessage.delete(Math.max(0, maxLogMessageLength - trimmed.length()));  
    logMessage.append(trimmed);  
}  
  
logger.info(logMessage.toString());
```

v. 24



Old school

```
if (i > 0) {  
    logMessage.append('\n');  
}  
logMessage.append(deepDiffs.get(i).toString());  
if (i ≥ maxNrOfDiffsToLog - 1) {  
    logMessage.append(REST_SKIPPED_MESSAGE);  
    break;  
}  
}
```

OLD CODE

```
if (logMessage.length() > maxLogMessageLength) {  
    String trimmed = "(trimmed)";  
    logMessage.delete(Math.max(0, maxLogMessageLength - trimmed.length));  
    logMessage.append(trimmed);  
}  
  
logger.info(logMessage.toString());
```

v. 23



```
if (i > 0) {  
    logMessage.append('\n');  
}  
logMessage.append(deepDiffs.get(i).toString());  
if (i ≥ maxNrOfDiffsToLog - 1) {  
    logMessage.append(REST_SKIPPED_MESSAGE);  
    break;  
}  
}
```

NEW CODE

```
if (logMessage.length() > maxLogMessageLength) {  
    String trimmed = "(trimmed)";  
    logMessage.delete(Math.max(0, maxLogMessageLength - trimmed.length));  
    logMessage.append(trimmed);  
}  
  
logger.info(logMessage.toString());
```

v. 24

AKA “fingers-crossed development”



Agenda

Progressive rollout

Fine-grained monitoring

The “Rollout Service”

Consistency checks

Automation

Benefits for everyone

Takeaways

New school - feature flags

```
if (i > 0) {  
    logMessage.append('\n');  
}  
logMessage.append(deepDiffs.get(i).toString());  
if (i >= maxNrOfDiffsToLog - 1) {  
    logMessage.append(REST_SKIPPED_MESSAGE);  
    break;  
}  
}  
  
if (logMessage.length() > maxLogMessageLength) {  
    String trimmed = "(trimmed)";  
    logMessage.delete(Math.max(0, maxLogMessageLength - trimmed.length()));  
    logMessage.append(trimmed);  
}  
  
logger.info(logMessage.toString());
```

OLD CODE

v. 23

```
if (ffService.isOn(MY_FF)) {  
  
} else {  
  
}
```

NEW CODE

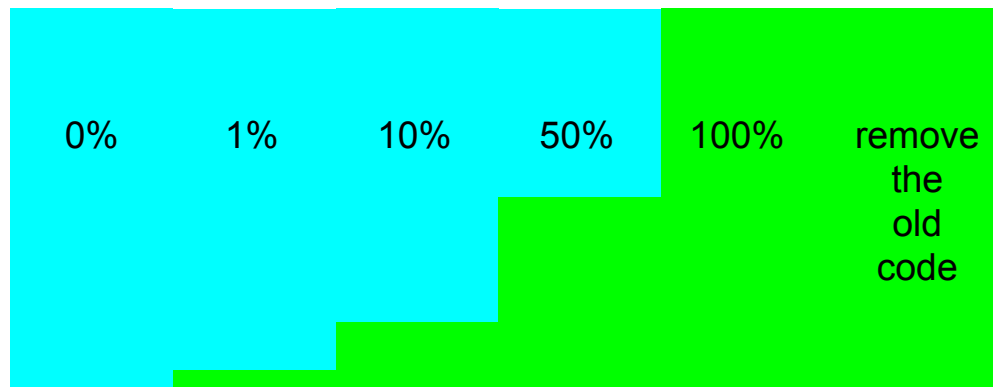
OLD CODE

v. 24



Progressive rollout

```
if (ffService.isOn(MY_FF)) {  
    NEW CODE  
} else {  
    OLD CODE  
}
```



time

1% rollout in



for 10 minutes

=

200k requests from
3k users belonging to
1k tenants



Progressive rollout

```
if (ffService.isOn(MY_FF)) {
```

NEW CODE

```
} else {
```

OLD CODE

```
}
```

=

OLD

NEW



Agenda

Progressive rollout

Fine-grained monitoring

The “Rollout Service”

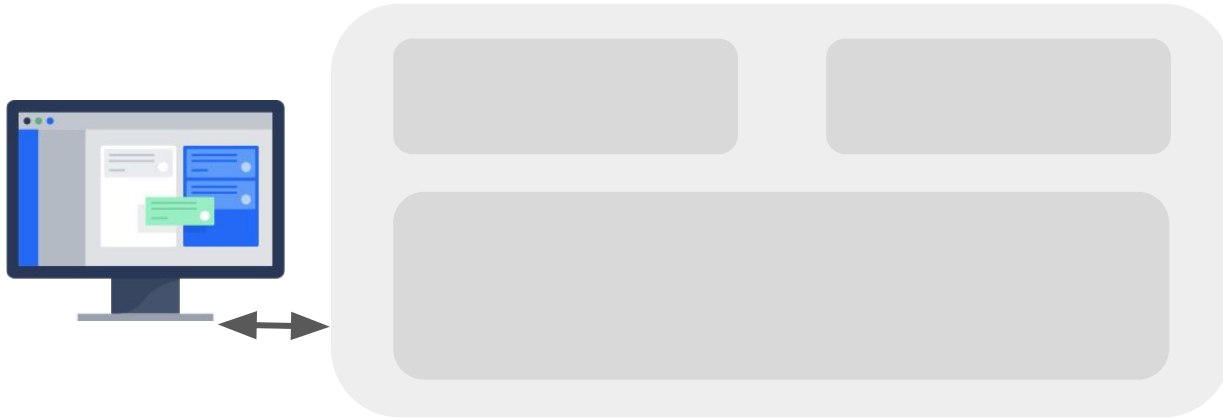
Consistency checks

Automation

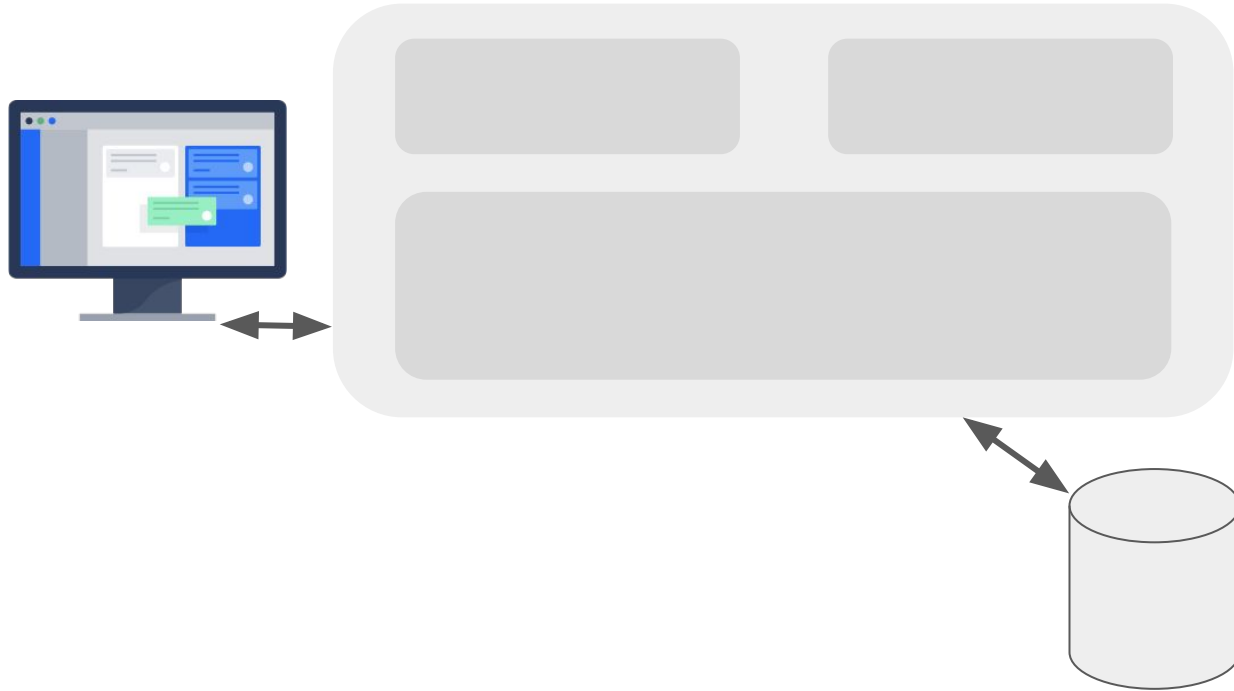
Benefits for everyone

Takeaways

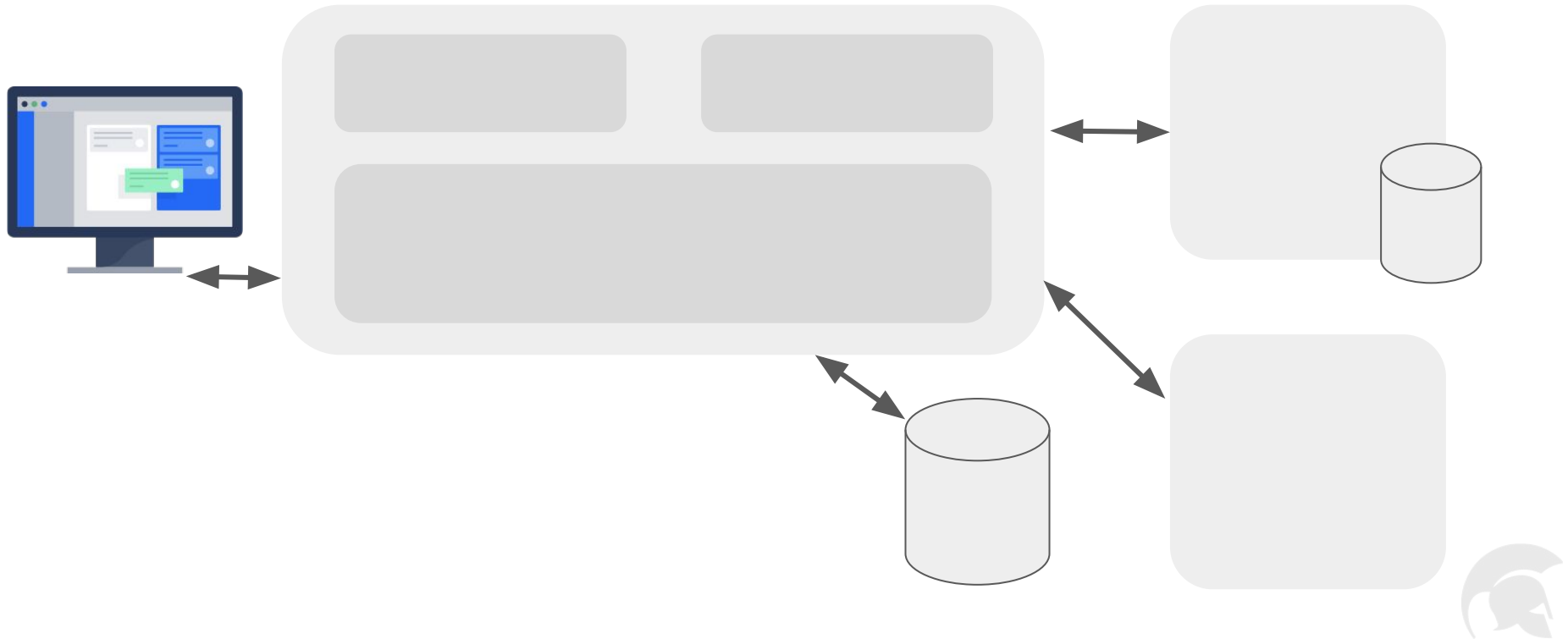
Sample architecture



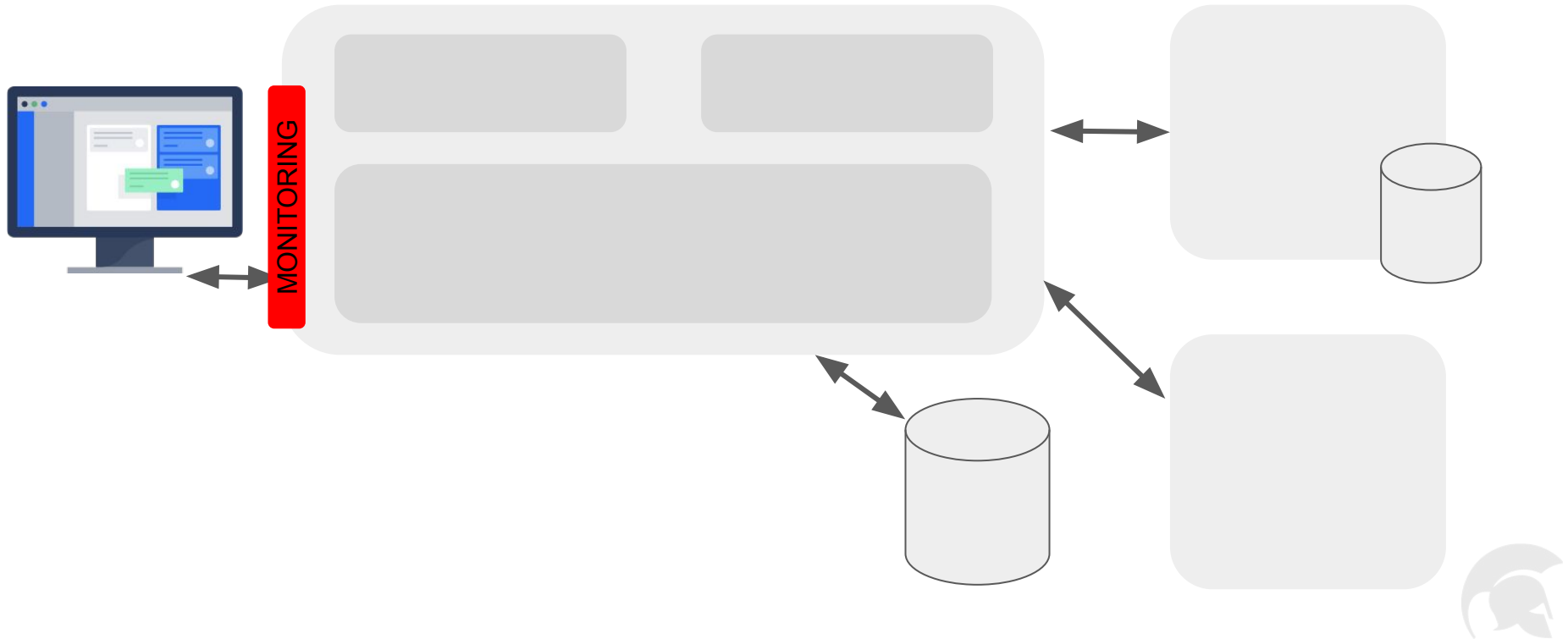
Sample architecture



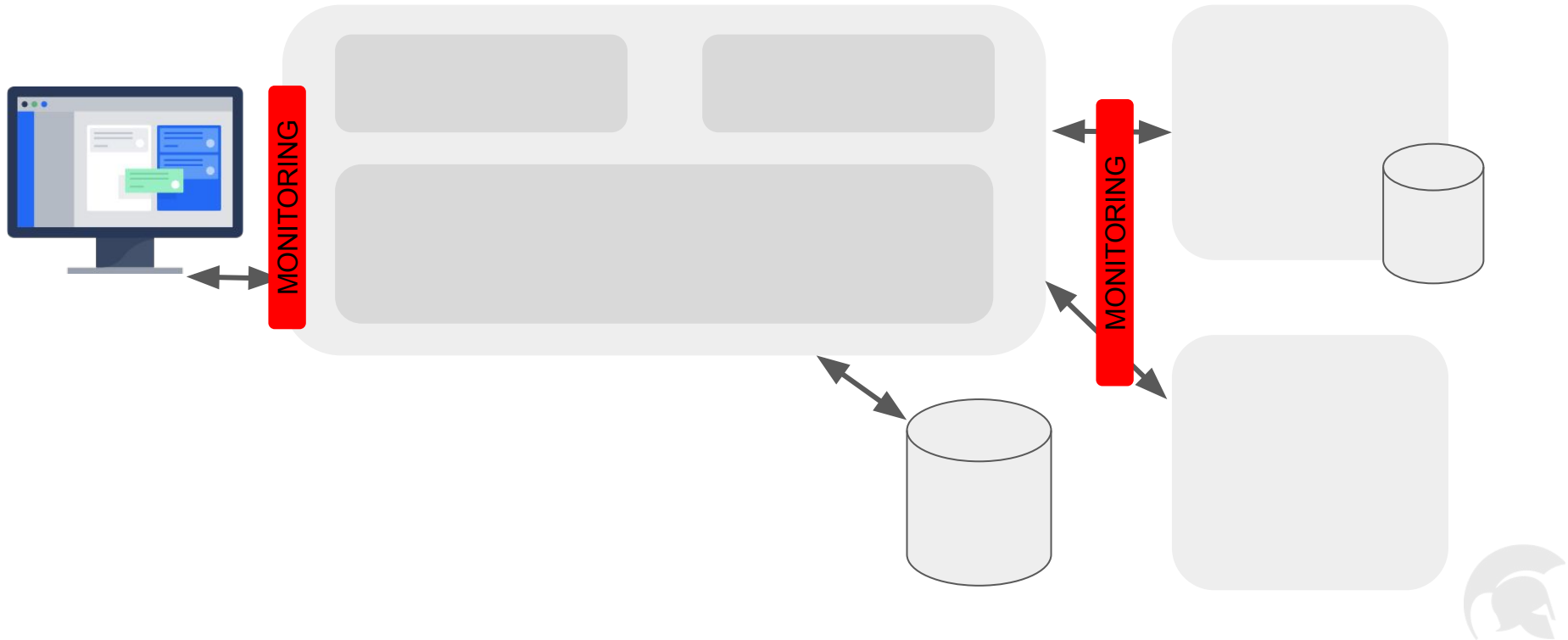
Sample architecture



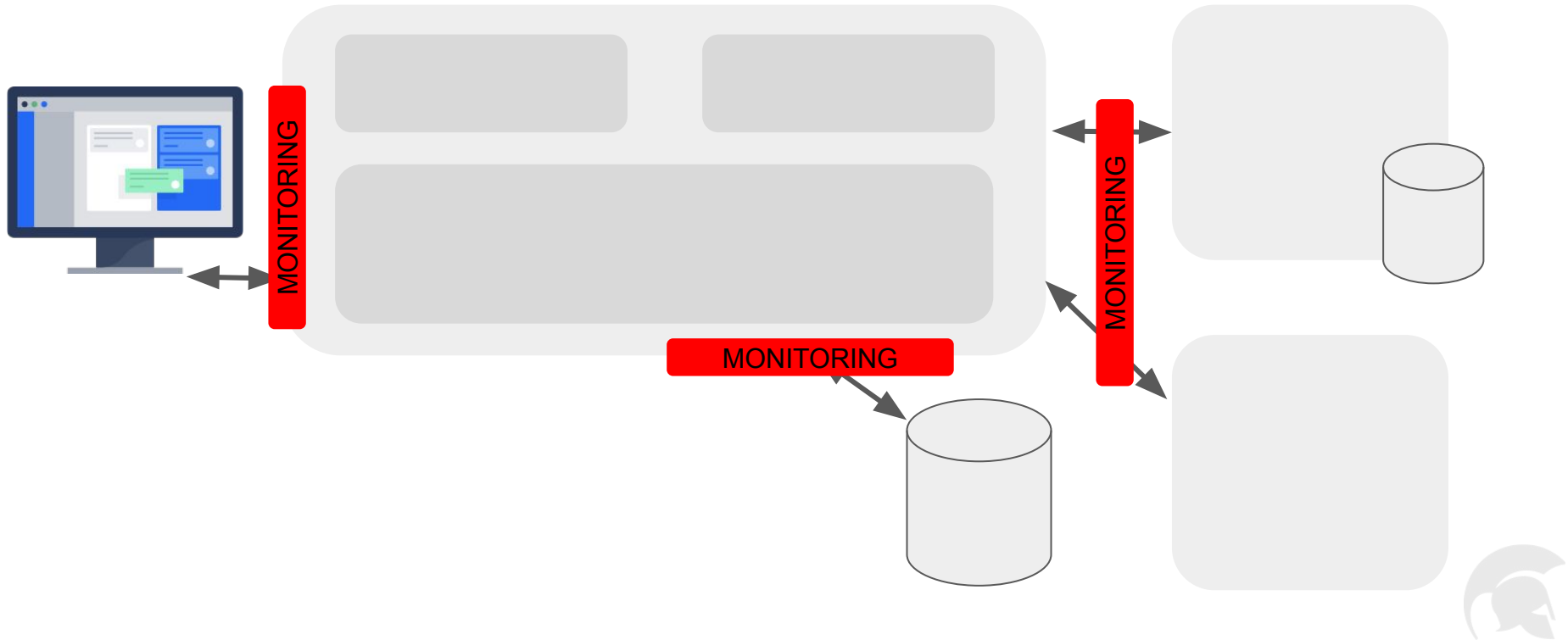
Sample architecture



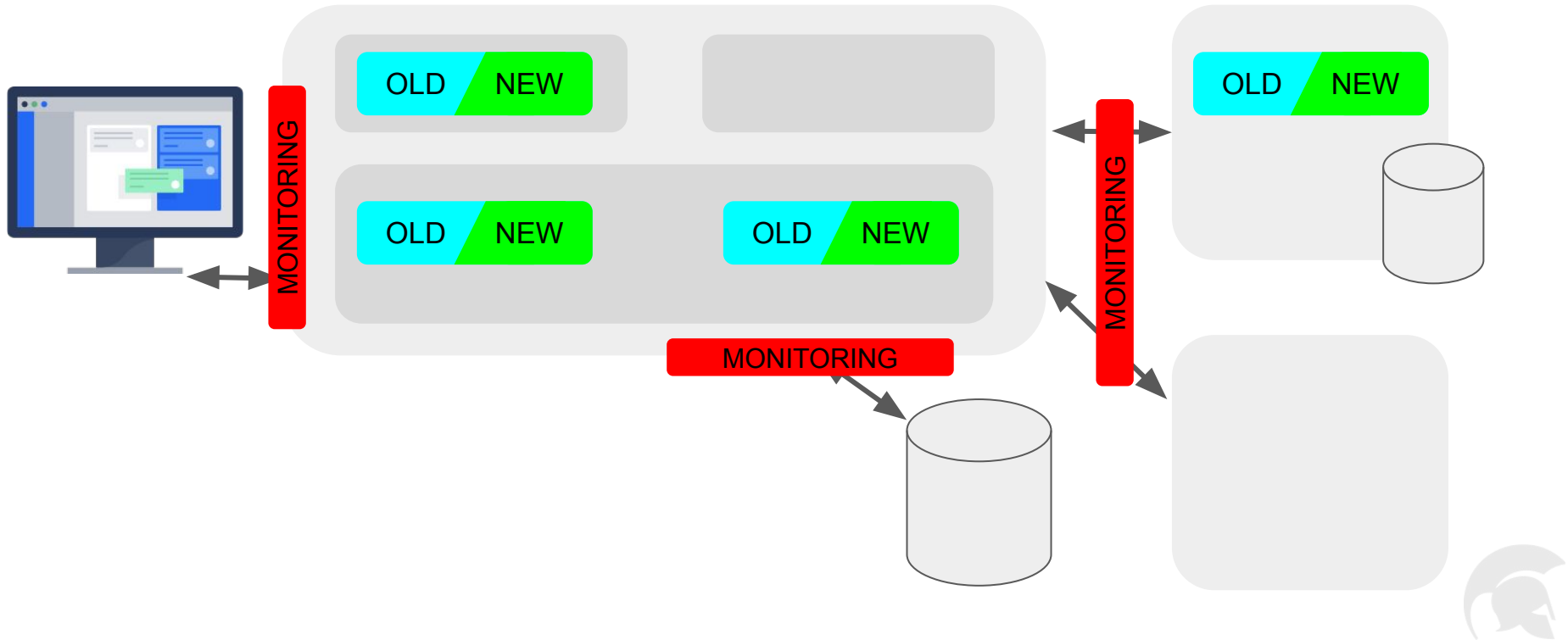
Sample architecture



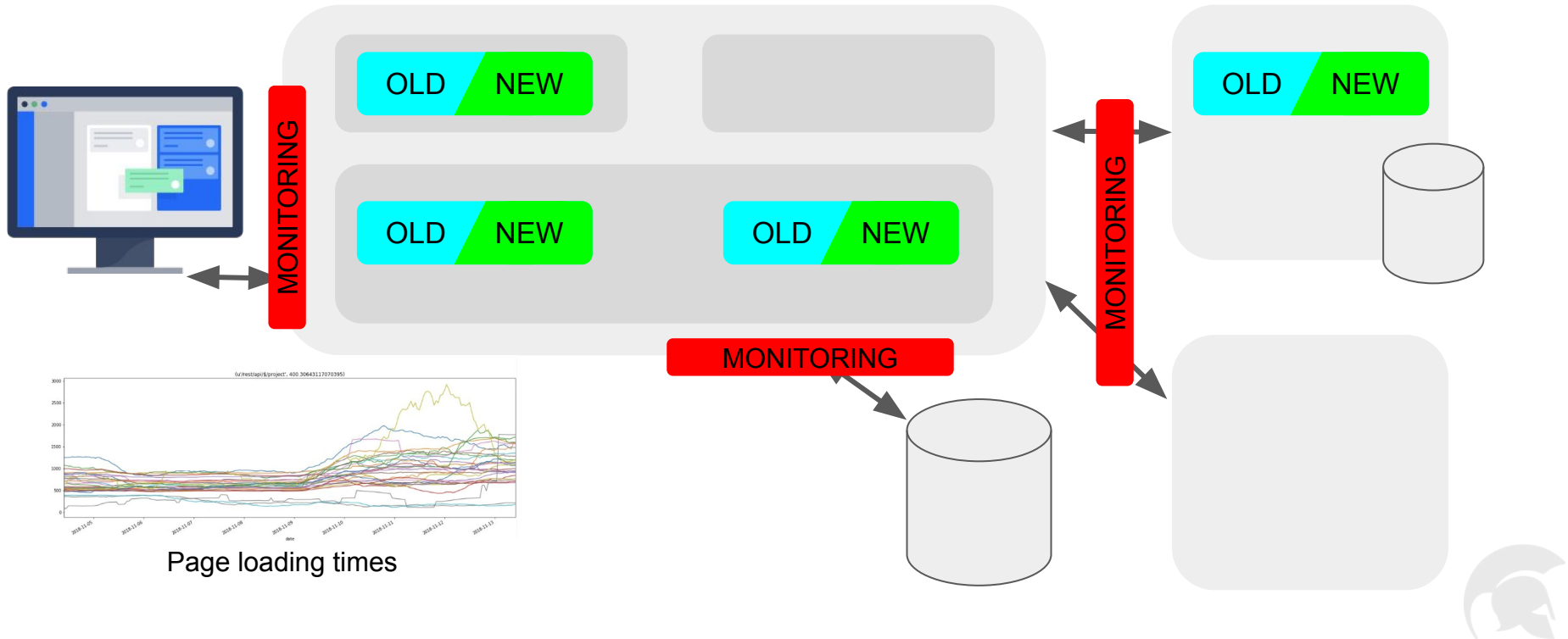
Sample architecture



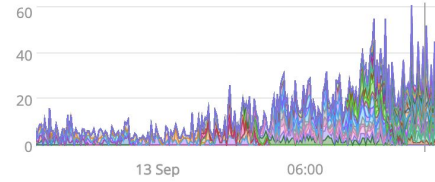
Sample architecture



Sample architecture



Sample architecture



HTTP 500s errors



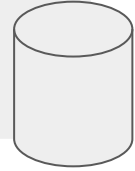
MONITORING

OLD NEW

OLD NEW

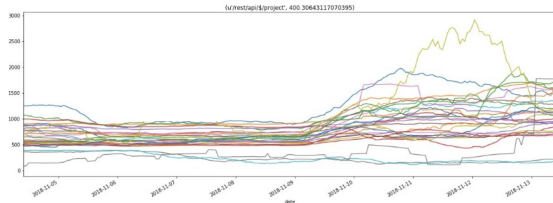
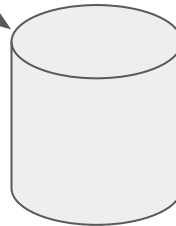
OLD NEW

OLD NEW



MONITORING

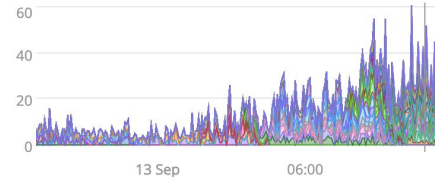
MONITORING



Page loading times



Sample architecture



HTTP 500s errors



MONITORING

OLD NEW

OLD NEW

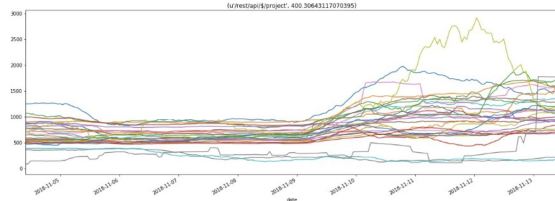
OLD NEW

OLD NEW

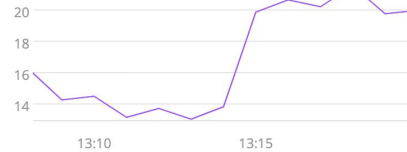


MONITORING

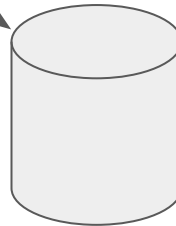
MONITORING



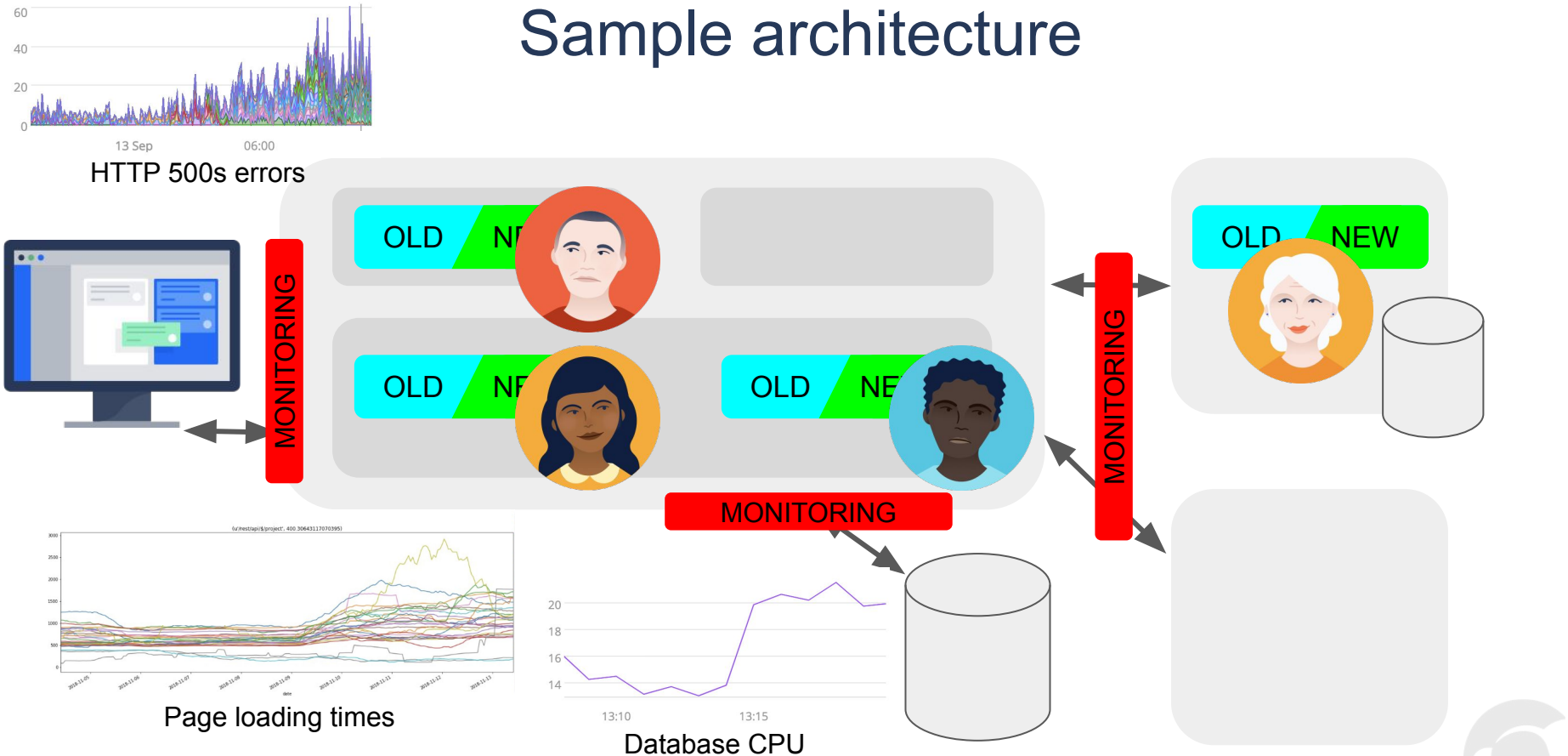
Page loading times



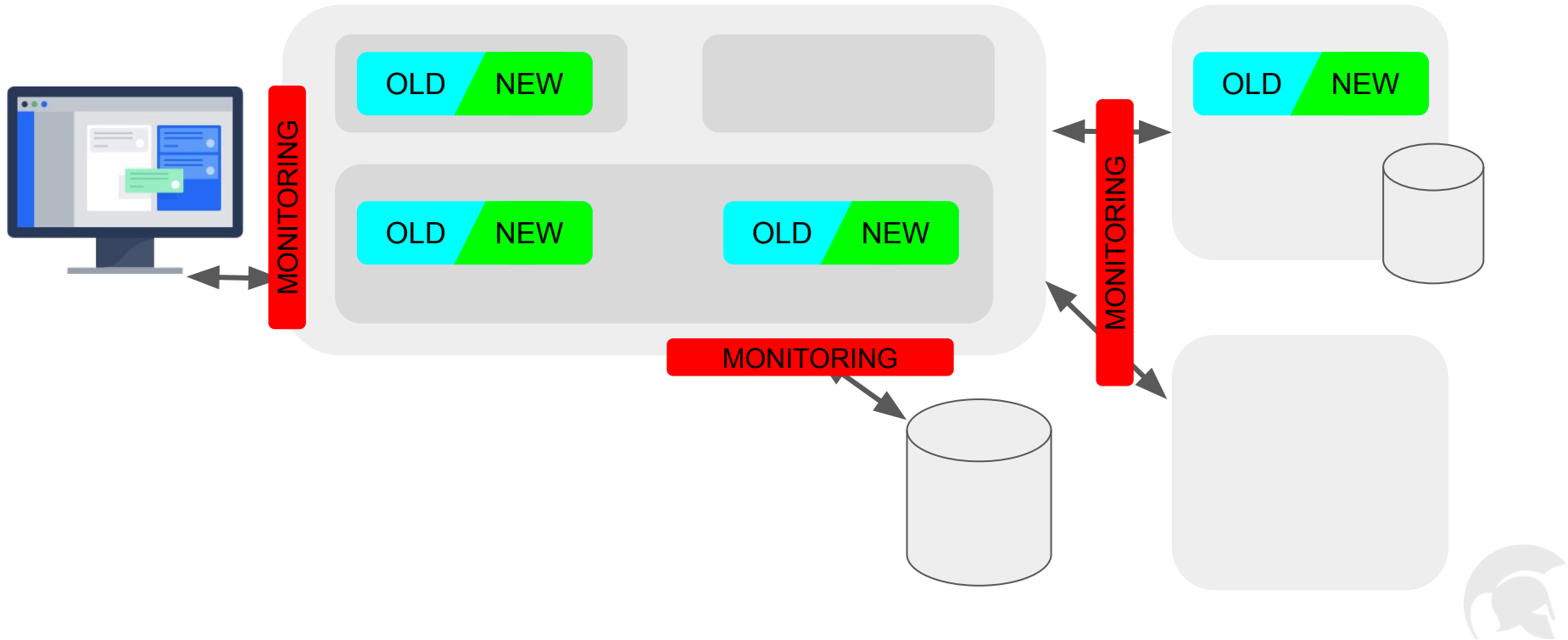
Database CPU



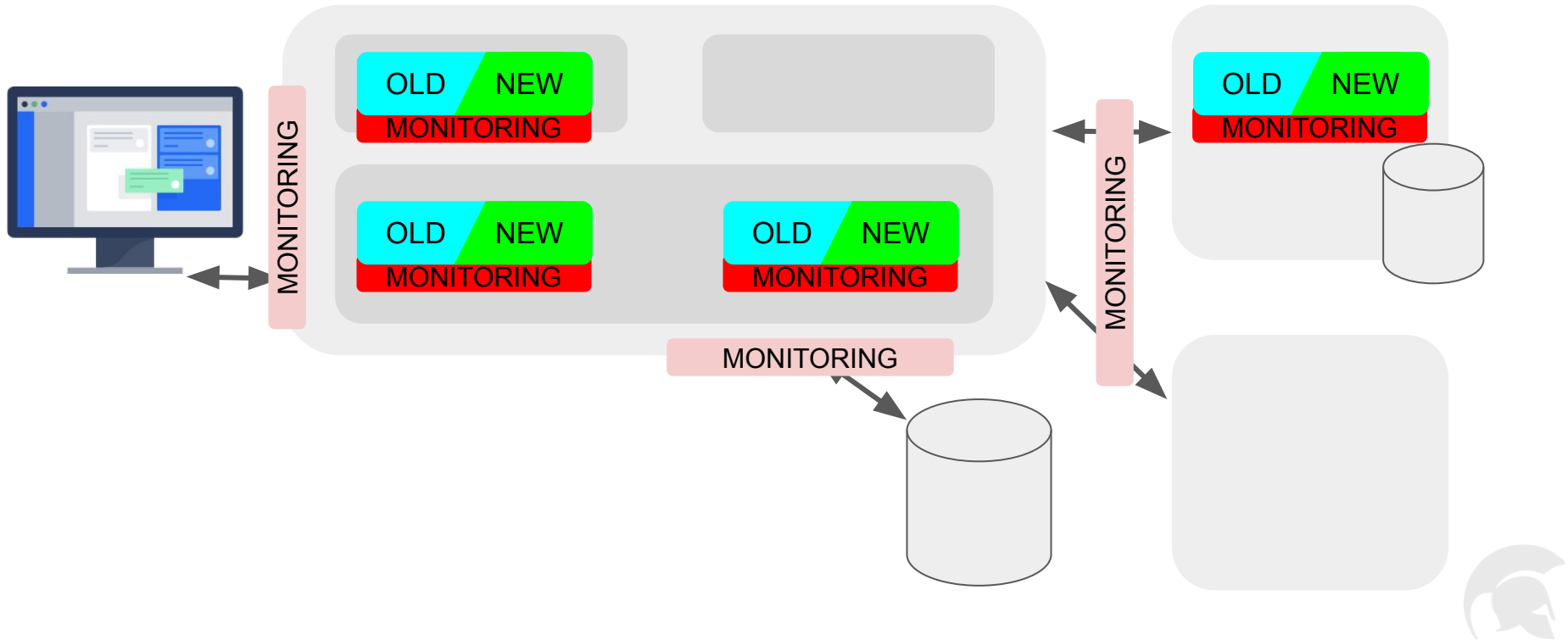
Sample architecture



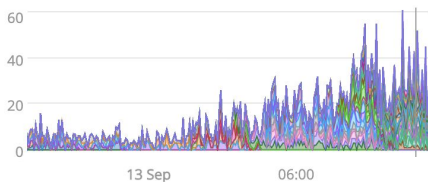
Sample architecture



Fine-grained monitoring

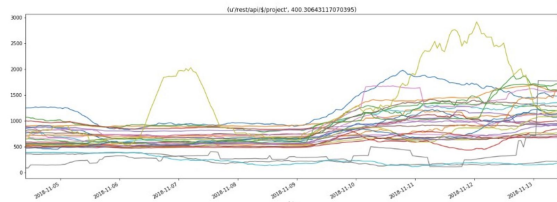
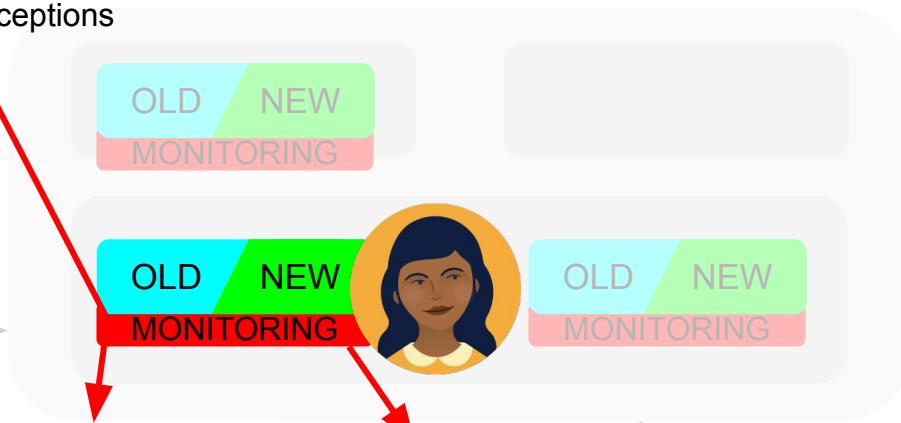


Fine-grained monitoring



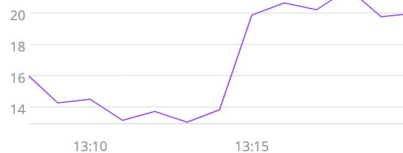
HTTP 500s errors

New vs Old code exceptions

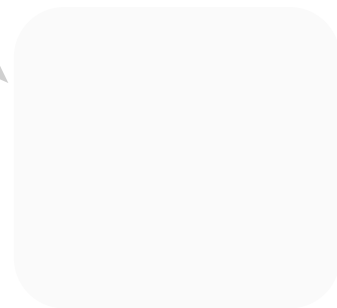
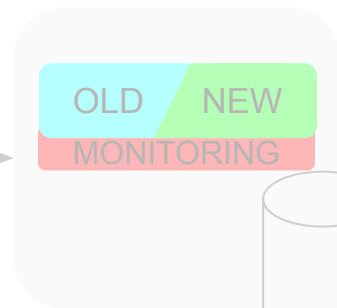


Page loading times

New vs Old code execution times



**Database CPU
Database time**



Agenda

Progressive rollout

Fine-grained monitoring

The “Rollout Service”

Consistency checks

Automation

Benefits for everyone

Takeaways

The “Rollout Service”



The “Rollout Service”

```
if (ffService.isOn(MY_FF)) {
```

NEW CODE

```
} else {
```

OLD CODE

```
}
```



The “Rollout Service” - exceptions

```
try {  
    if (ffService.isOn(MY_FF)) {  
        NEW CODE  
    } else {  
        OLD CODE  
    }  
} catch (Exception e) {  
    log(MY_FF, ffService.isOn(MY_FF), e);  
    throw e;  
}
```

Log:

- feature flag name
- feature flag on / off
- exception thrown by old or new code



*“New code throws 100 times more **NullPointerExceptions** than the old code.
Perhaps we should **rollback** the feature flag?”*



The “Rollout Service” - performance

```
long startTime = System.nanoTime();  
if (ffService.isOn(MY_FF)) {
```

NEW CODE

```
} else {
```

OLD CODE

```
}  
logTime(MY_FF, ffService.isOn(MY_FF),  
        System.nanoTime() - startTime);
```

Log:

- feature flag name
- feature flag on / off
- time taken to execute the old or new code



*“New code executes **2 times slower** than the old code.*

*Perhaps we should **rollback** the feature flag?”*



The “Rollout Service” - DB time

```
try {  
    markDbCalls(MY_FF, ffService.isOn(MY_FF));  
    if (ffService.isOn(MY_FF)) {  
        NEW CODE  
    } else {  
        OLD CODE  
    }  
} finally {  
    unmarkDbCalls(MY_FF, ffService.isOn(MY_FF));  
}
```

Log:

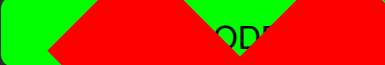
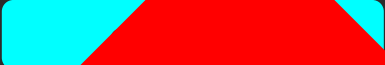
- feature flag name
- feature flag on / off
- time spent in DB
- number of DB calls

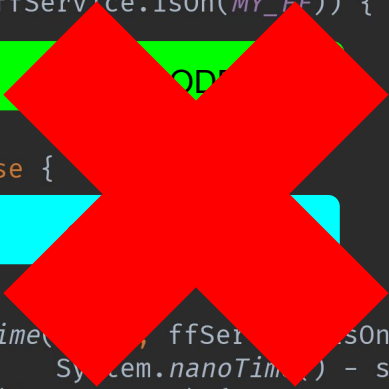


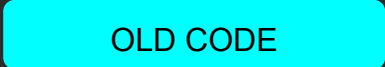

*“New code executes DB queries that are
3 times slower.
Are you serious?”*



The “Rollout Service” - bringing it all together

```
try {  
    markDbCalls(MY_FF, ffService.isOn(MY_FF));  
    long startTime = System.nanoTime();  
    if (ffService.isOn(MY_FF)) {  
           
    } else {  
           
    }  
    logTime(MY_FF, ffService.isOn(MY_FF),  
            System.nanoTime() - startTime);  
} catch (Exception e) {  
    log(MY_FF, ffService.isOn(MY_FF), e);  
} finally {  
    unmarkDbCalls(MY_FF, ffService.isOn(MY_FF));  
}
```



```
rolloutService.rolloutControlledBy(MY_FF)  
    .replacing(  
        () →  OLD CODE  
    )  
    .with(  
        () →  NEW CODE  
    );
```



Agenda

Progressive rollout

Fine-grained monitoring

The “Rollout Service”

Consistency checks

Automation

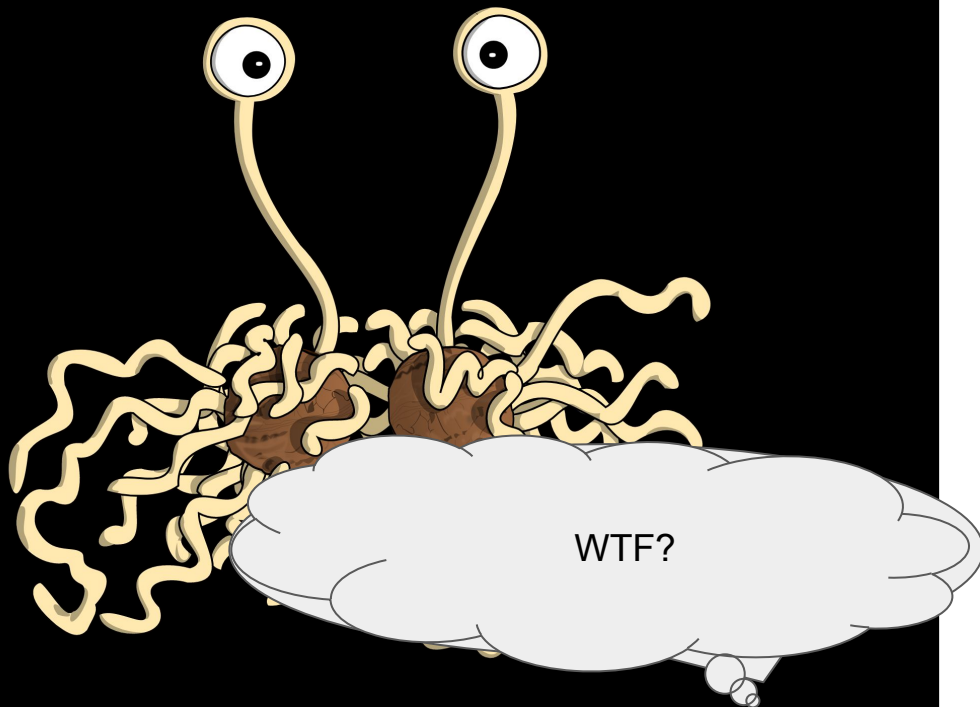
Benefits for everyone

Takeaways

Refactoring of a pricing algorithm

```
ourPrice = calculatePrice(  
    originalPrice,  
    numberOfTravelers,  
    excessiveLuggage,  
    priorityBoarding,  
    extraOptions,  
    travelerAges,  
    flyingDuringTheWeekend,  
    tonsOfOtherThings  
)
```

Can we add one more thing?



Refactoring of a pricing algorithm

`calculatePriceTheOldWay() == calculatePriceRefactored()`



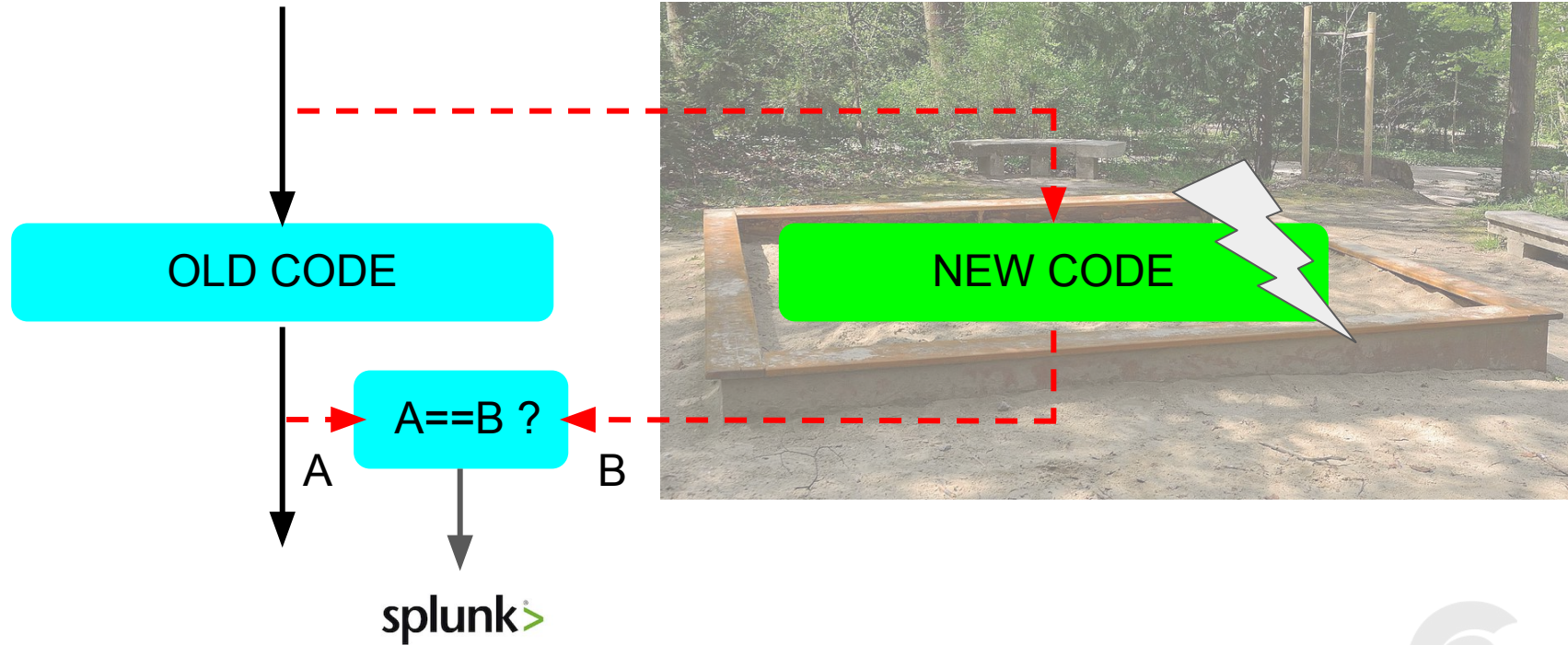
Do some exploratory testing?
Write comprehensive tests?
Copy production data?

Refactoring of a pricing algorithm

```
public Price calculatePrice(PricingData data) {  
    Price oldPrice = calculatePriceTheOldWay(data);  
    try {  
        Price newPrice = calculatePriceRefactored(data);  
        if (!newPrice.equals(oldPrice)) {  
            log("Consistency check failed for {}", data);  
        }  
    } catch (Exception e) {  
        log(e);  
    }  
    return oldPrice;  
}
```



Consistency Check



Back to the Rollout Service...

```
rolloutService.rolloutControlledBy(MY_FF)  
    → .withConsistencyCheck()  
      .replacing(  
          () → OLD CODE  
      )  
      .with(  
          () → NEW CODE  
      );
```



Rollout Service summary



Makes you aware of:

- exceptions
- performance regressions
- expensive DB queries
- expensive microservice calls
- data inconsistencies
- etc.

You

LAYER

App



Agenda

Progressive rollout

Fine-grained monitoring

The “Rollout Service”

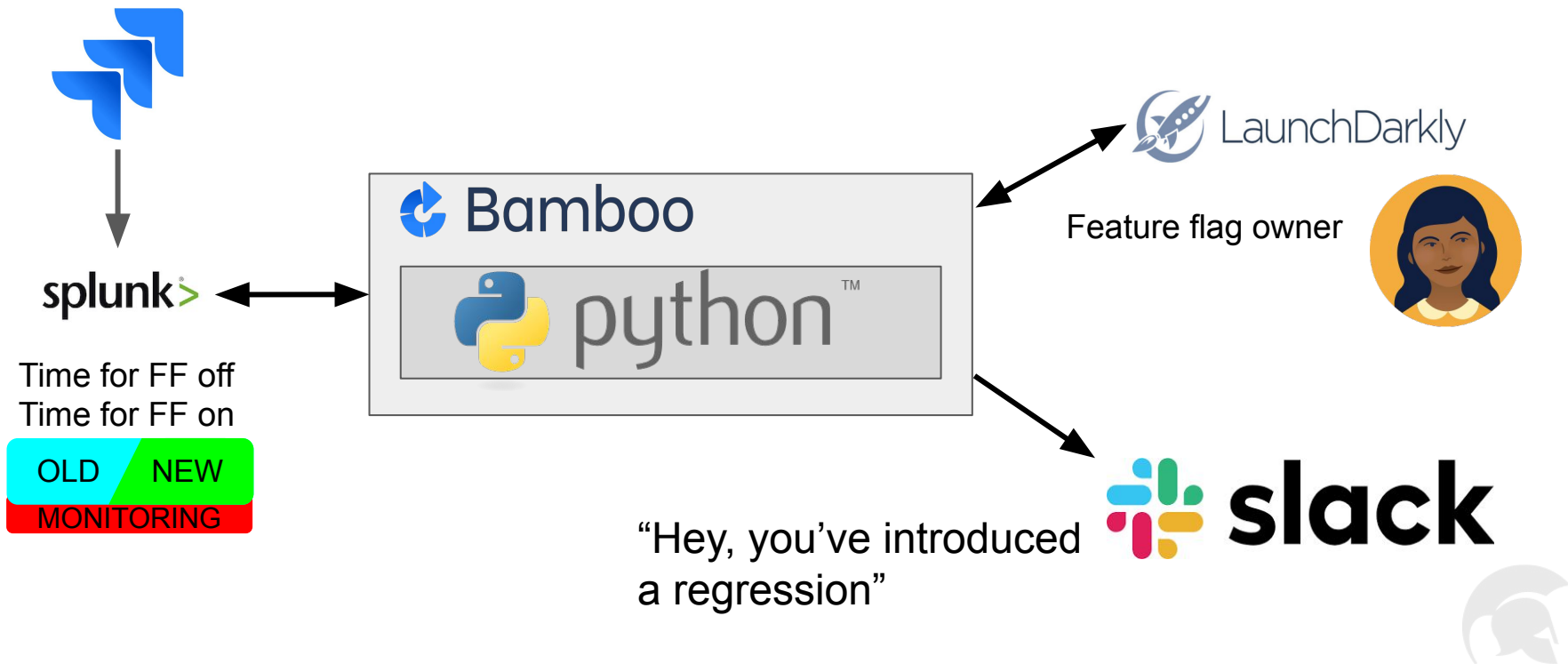
Consistency checks

Automation

Benefits for everyone

Takeaways

Automated Regression Detection



Automated Regression Detection



CloudBoa Regression Detector APP 7:33 AM

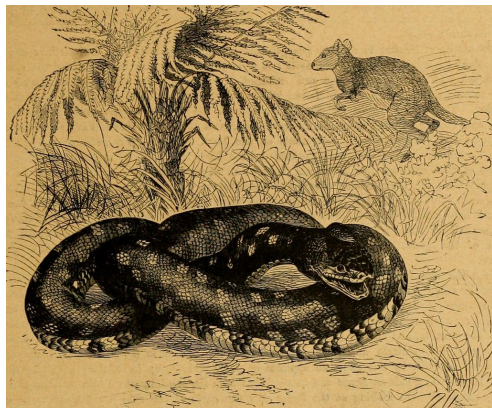
Hello, I'm [Cloud Boa - Jira Cloud Regression Detector](#)



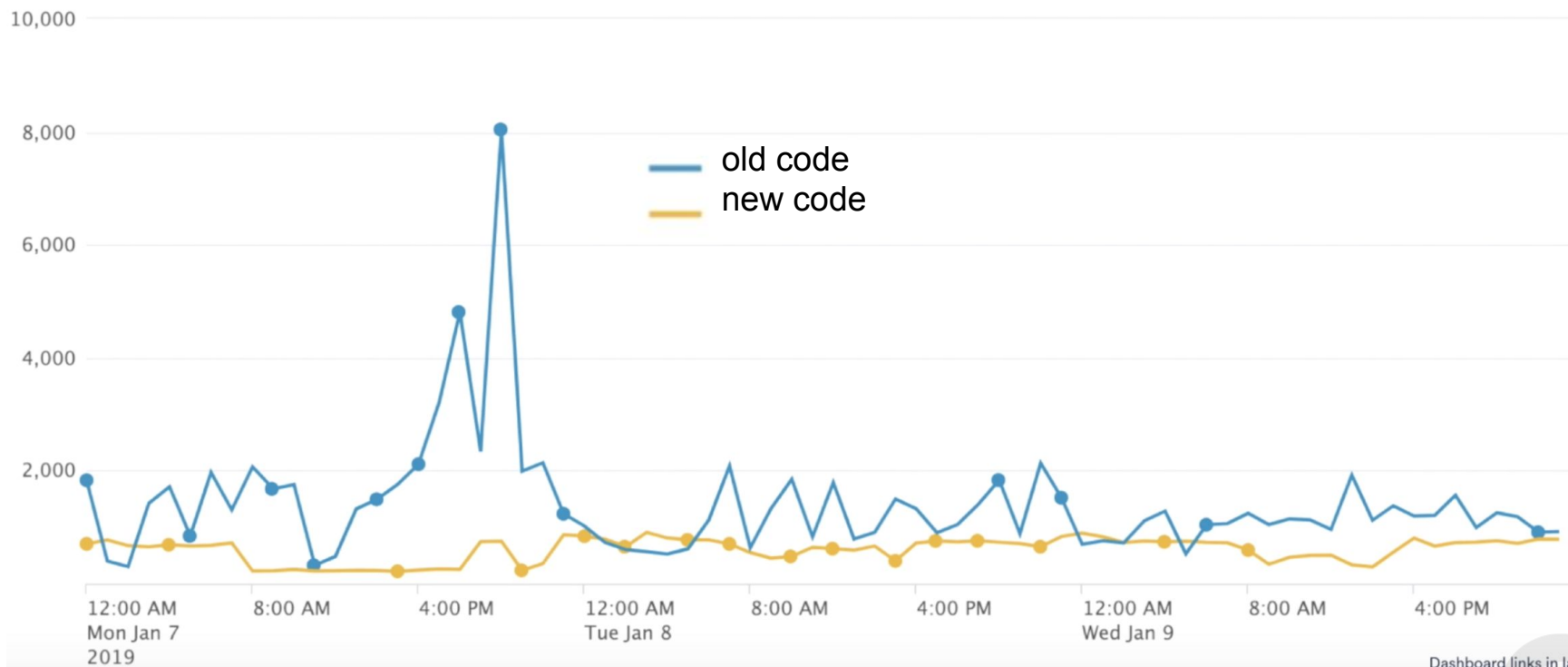
I've found that code enabled on production by your feature flag '[some-feature-flag](#)' introduced a potential regression.

The code regressed by 112% (by 110 ms on 99th percentile) and there're ~ 24463 requests affected by the feature flag per hour

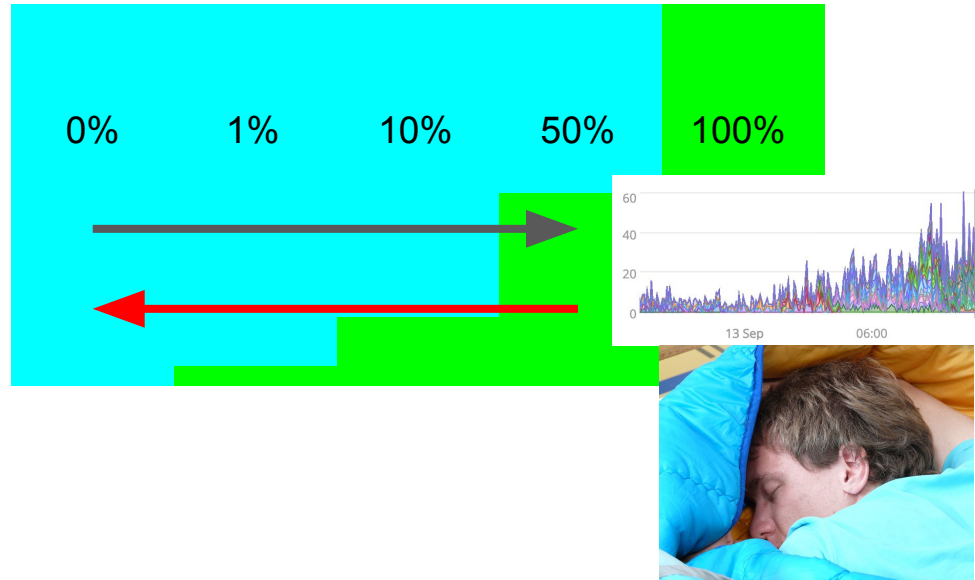
See the [results in Splunk](#)



Automated Regression Detection



Automated Rollback



Agenda

Progressive rollout

Fine-grained monitoring

The “Rollout Service”

Consistency checks

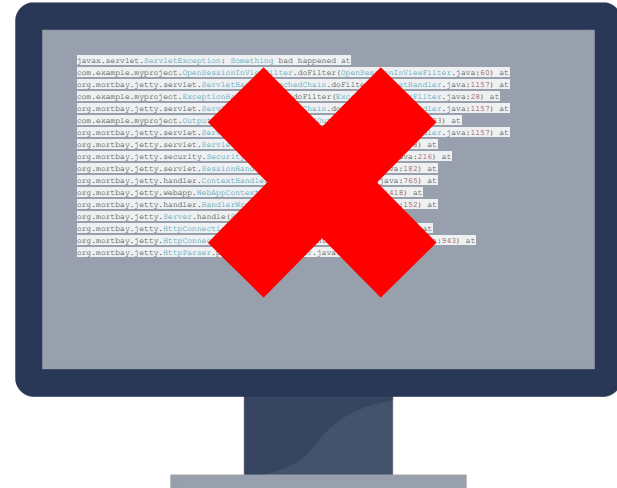
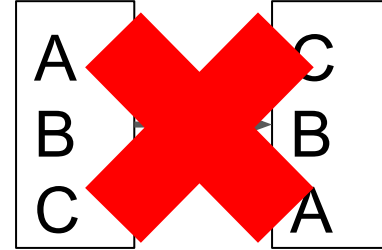
Automation

Benefits for everyone

Takeaways

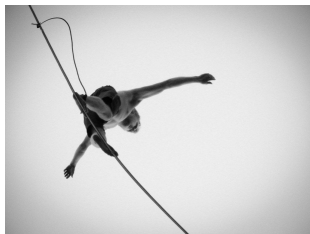
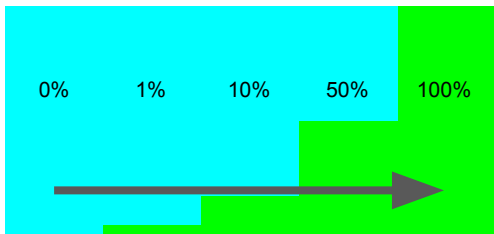


Happy users

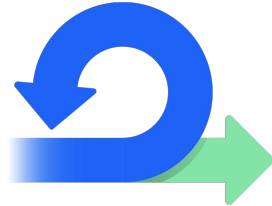
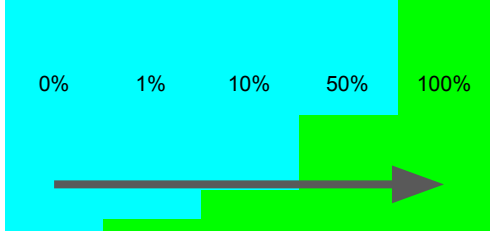




Happy developers



Happy developers

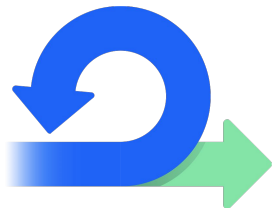
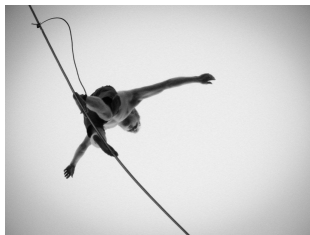
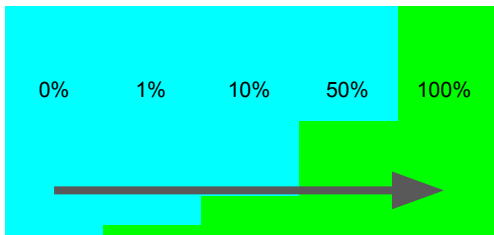
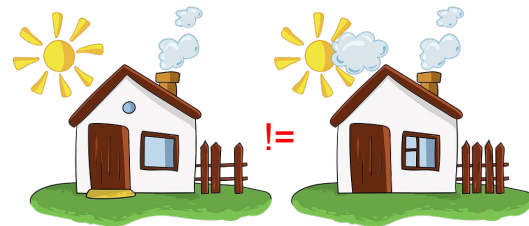


Source: https://commons.wikimedia.org/wiki/File:LEGO_Bits_Box_2.jpg





Happy developers

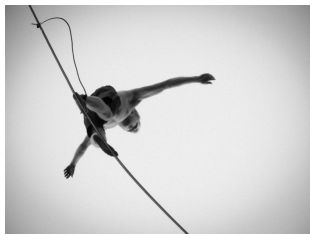
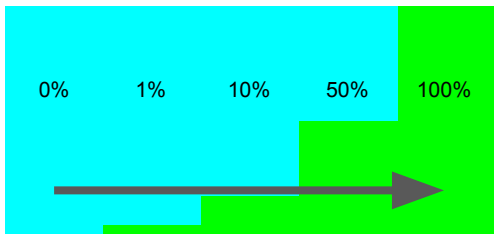
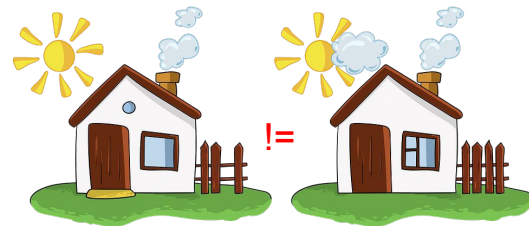


Source: <https://pixabay.com/illustrations/house-houses-spot-the-difference-3208132/>





Happy developers

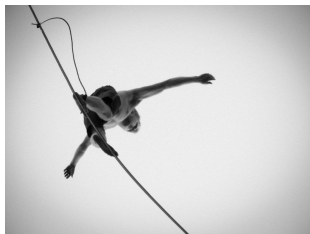
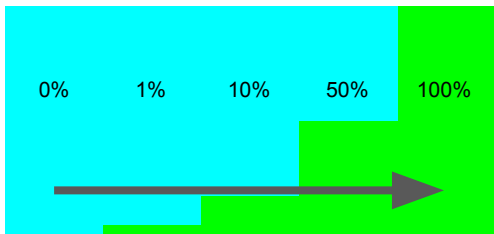
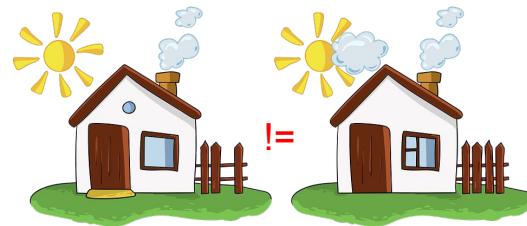


Source: <https://www.pexels.com/photo/woman-on-hammock-reading-book-1097177/>



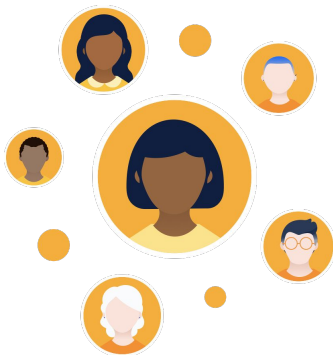
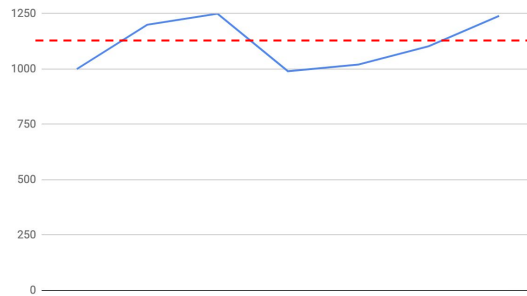


Happy developers



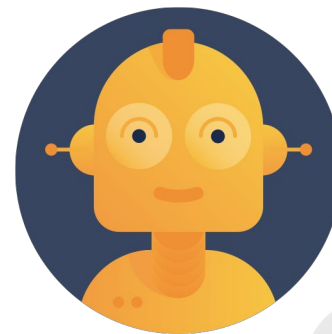
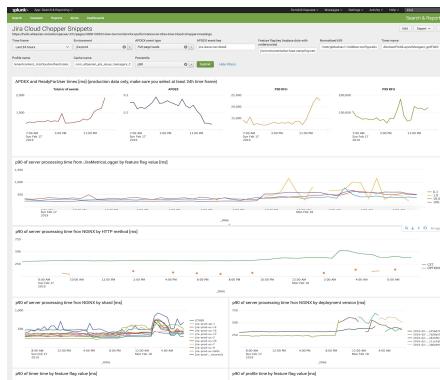
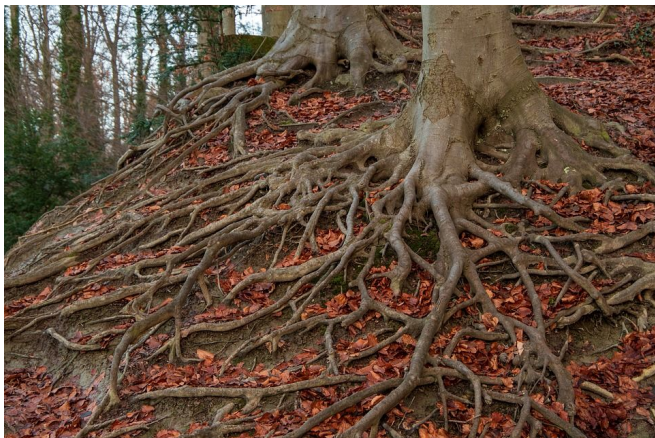
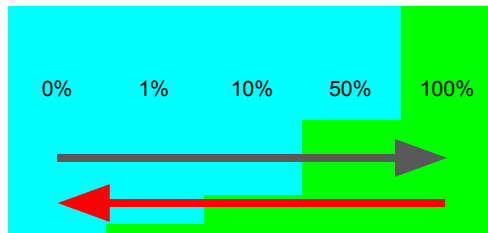


Gotchas for devs

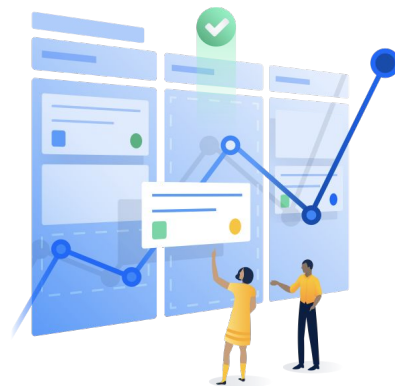
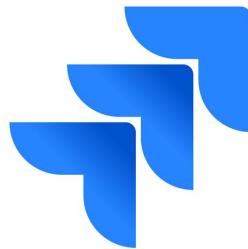




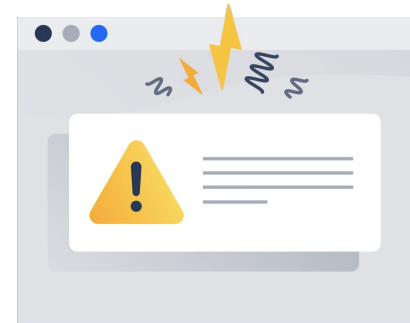
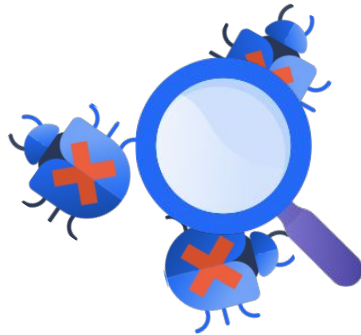
Happy DevOps



Happy PMs



Happy company



“Do you really have no tests?”

?





Source: <https://pixabay.com/vectors/key-vintage-key-lock-old-antique-2824086/>
<https://pixabay.com/pl/vectors/takeaway-chi%C5%84ski-fast-food-pole-154591/>



Key takeaways

- Testing on production doesn't have to hurt
- Simple ideas may reduce risk by a lot
- Automate!
- Embrace cultural change by making it sexy





Questions?

@DominikKapusta4 / LinkedIn: elser

Dominik Kapusta / Site Reliability Engineer

Join our super team!

spartez.com/careers





Thank you

@DominikKapusta4 / LinkedIn: elser

Dominik Kapusta / Site Reliability Engineer