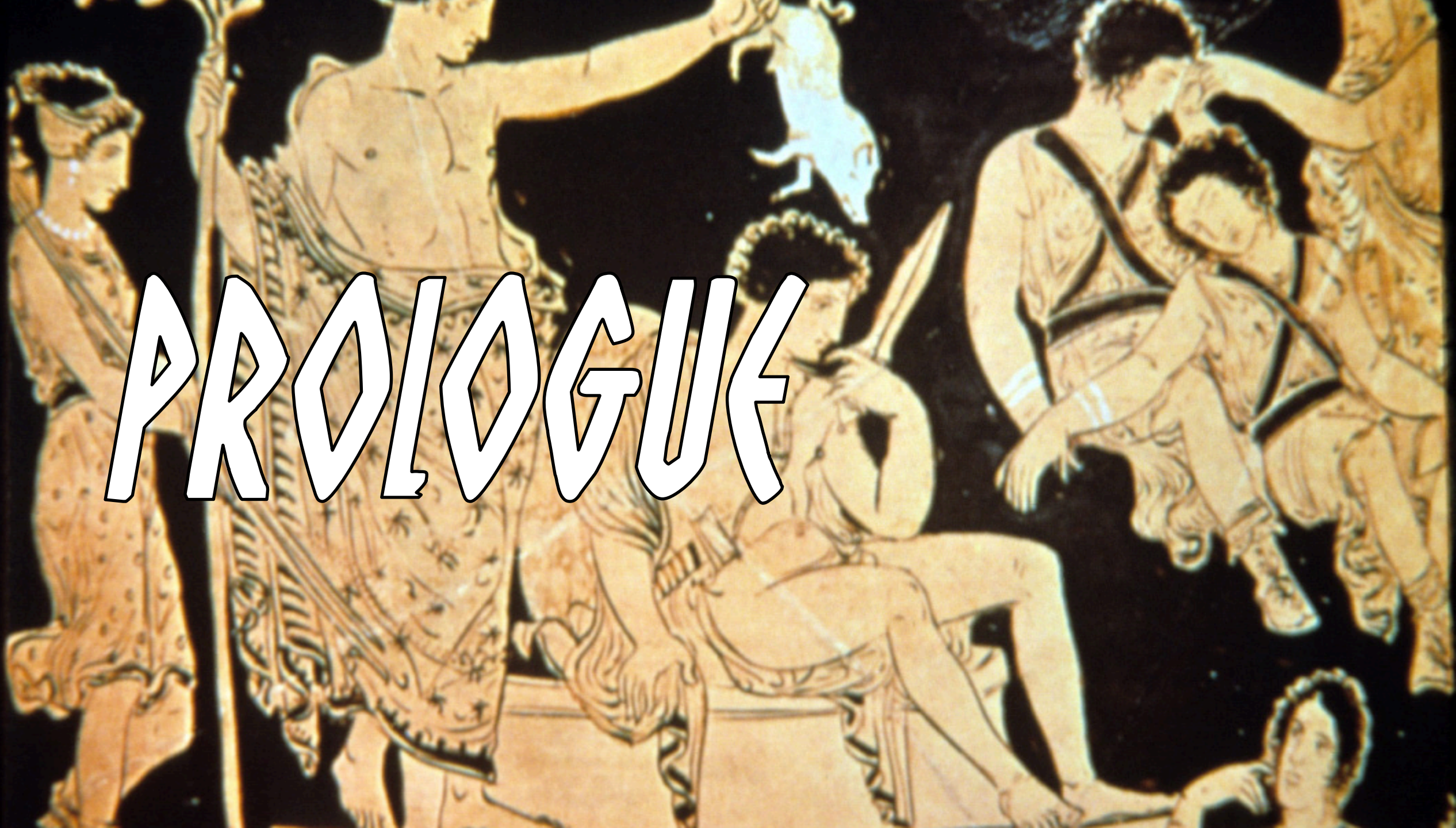




# DEVOPS @SCALE

GREEK TRAGEDY IN THREE ACTS

# PROLOGUE





BARUCH SADOGURSKY

LEONID IGOLNIK

HEAD OF DEVOPS ADVOCACY@JFROG  
@JBARUCH ON THE INTERNETZ

EX VP ENG @CA & SIGNALFX  
@LIGOLNIK ON THE INTERNETZ

# [HTTPS://JFROG.COM/SHOWNOTES](https://jfrog.com/shownotes)

 SLIDES

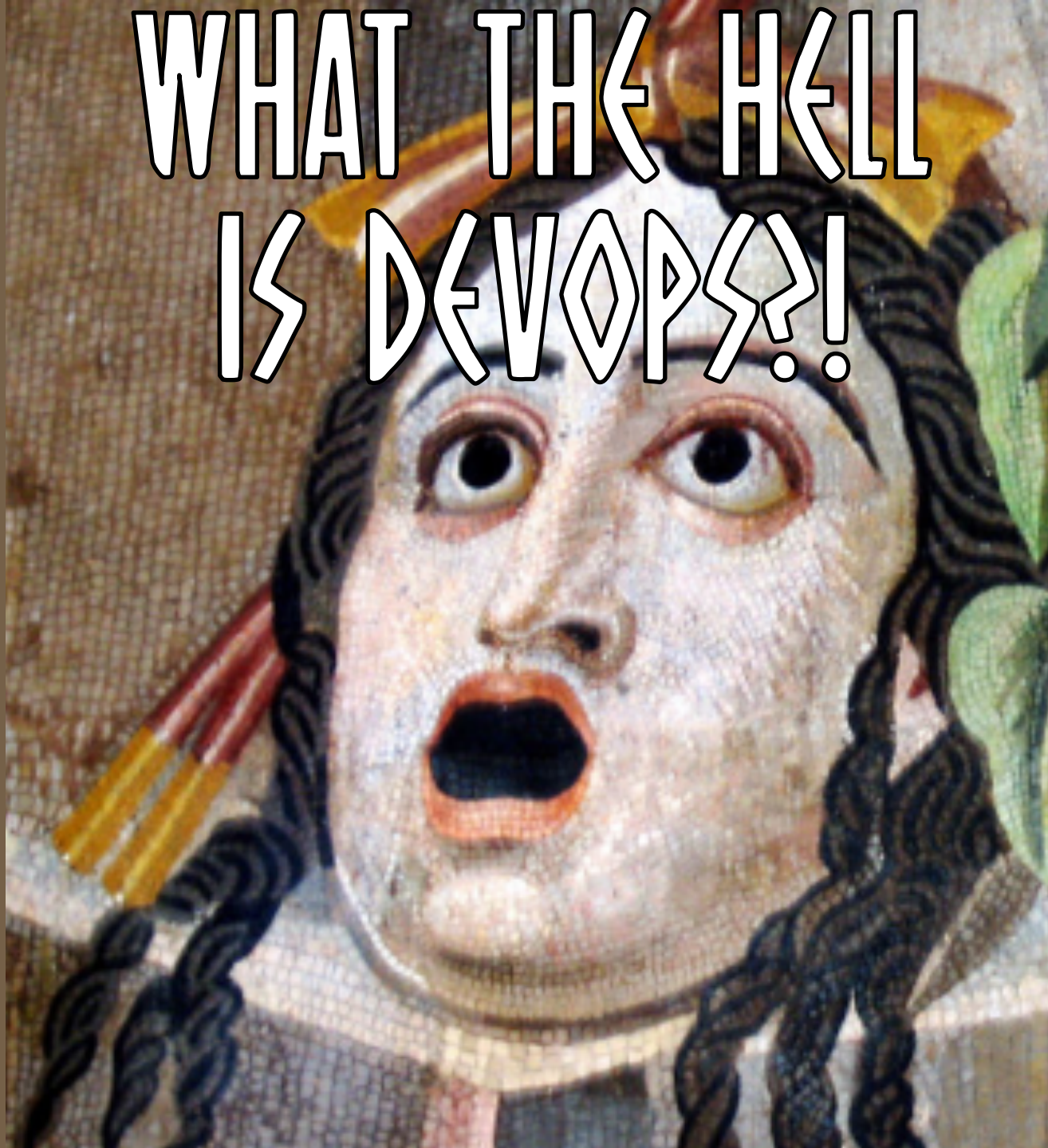
 VIDEO

 ALL THE LINKS

 COMMENTS, RATINGS

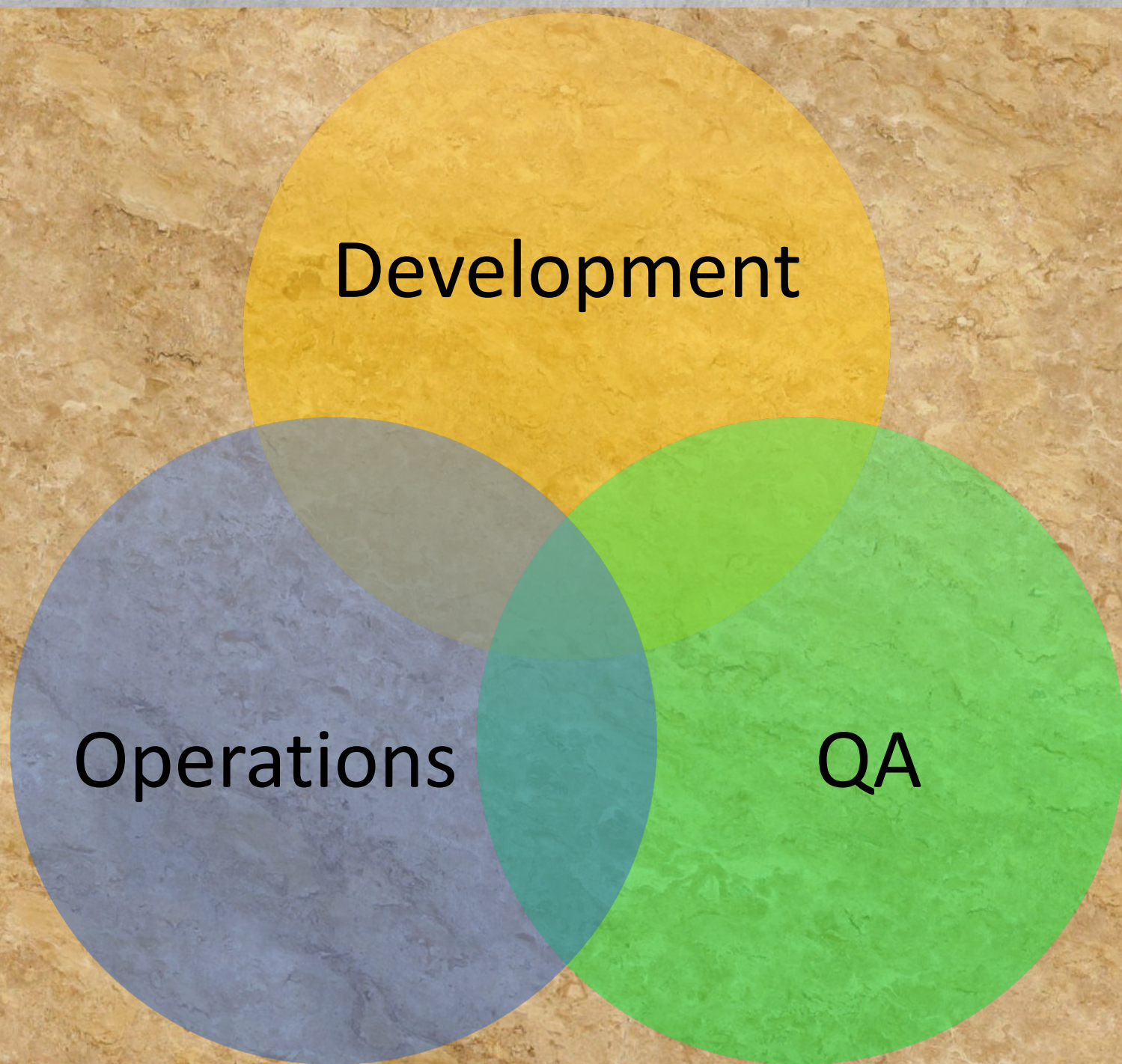
 RAFFLE!

WHAT THE HELL  
IS DEVOPS?!



# GREEKS LOVE VENN DIAGRAMS



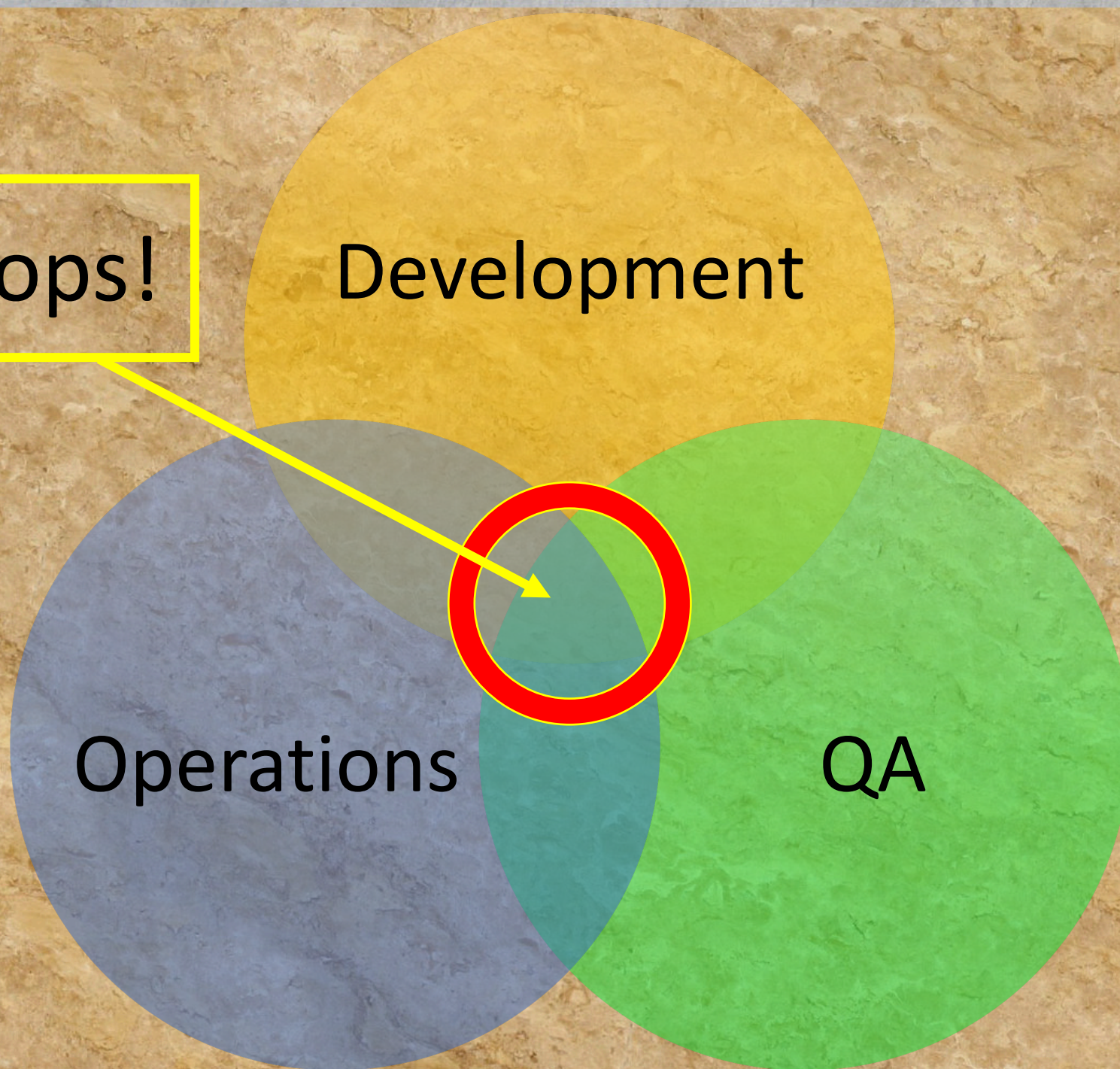


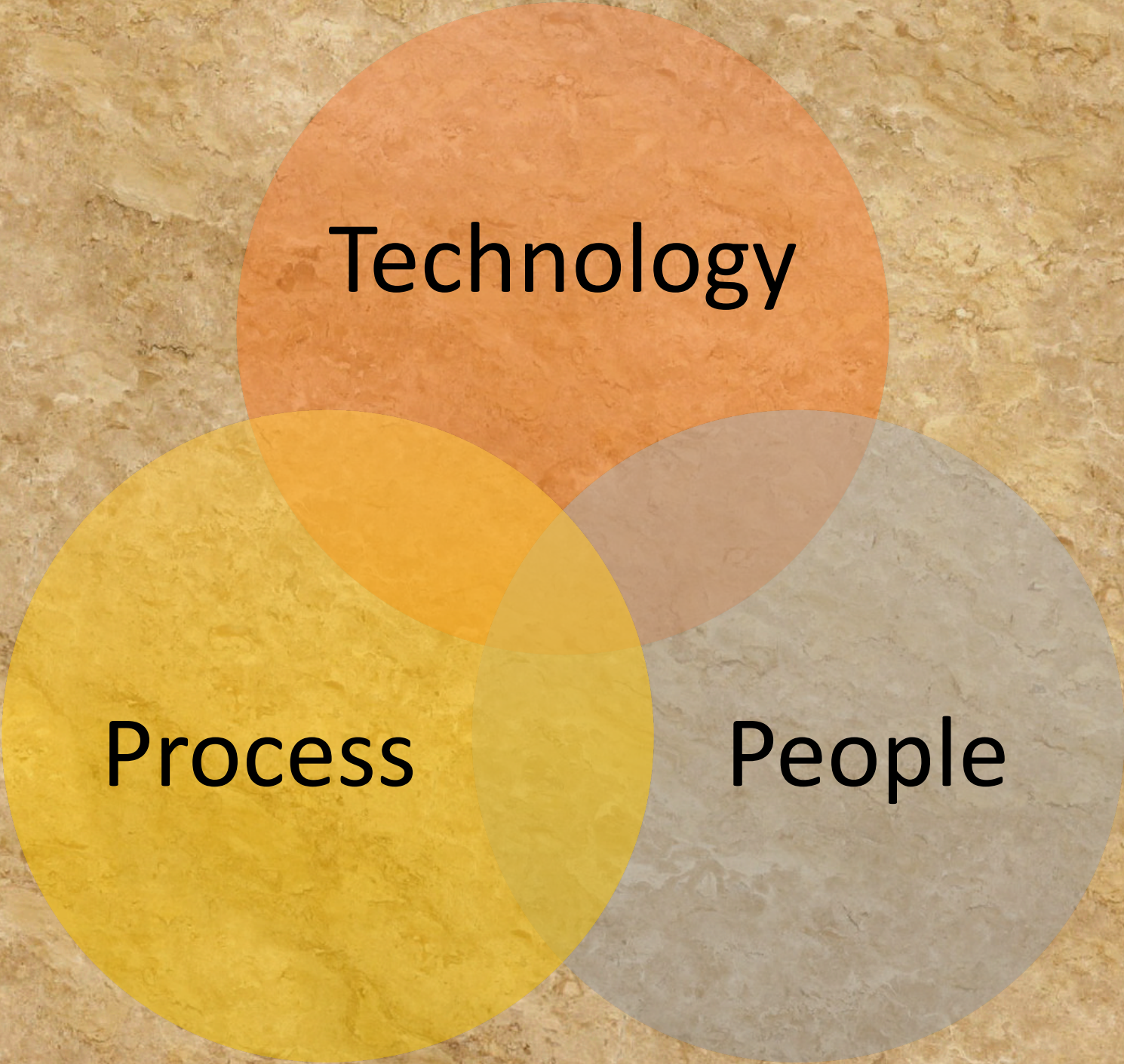
Devops!

Development

Operations

QA





A Venn diagram consisting of three overlapping circles on a textured, light brown background. The top circle is orange and labeled 'Technology'. The bottom-left circle is yellow and labeled 'Process'. The bottom-right circle is grey and labeled 'People'. The circles overlap in various combinations, creating different shades of orange, yellow, and grey in the intersection areas.

Technology

Process

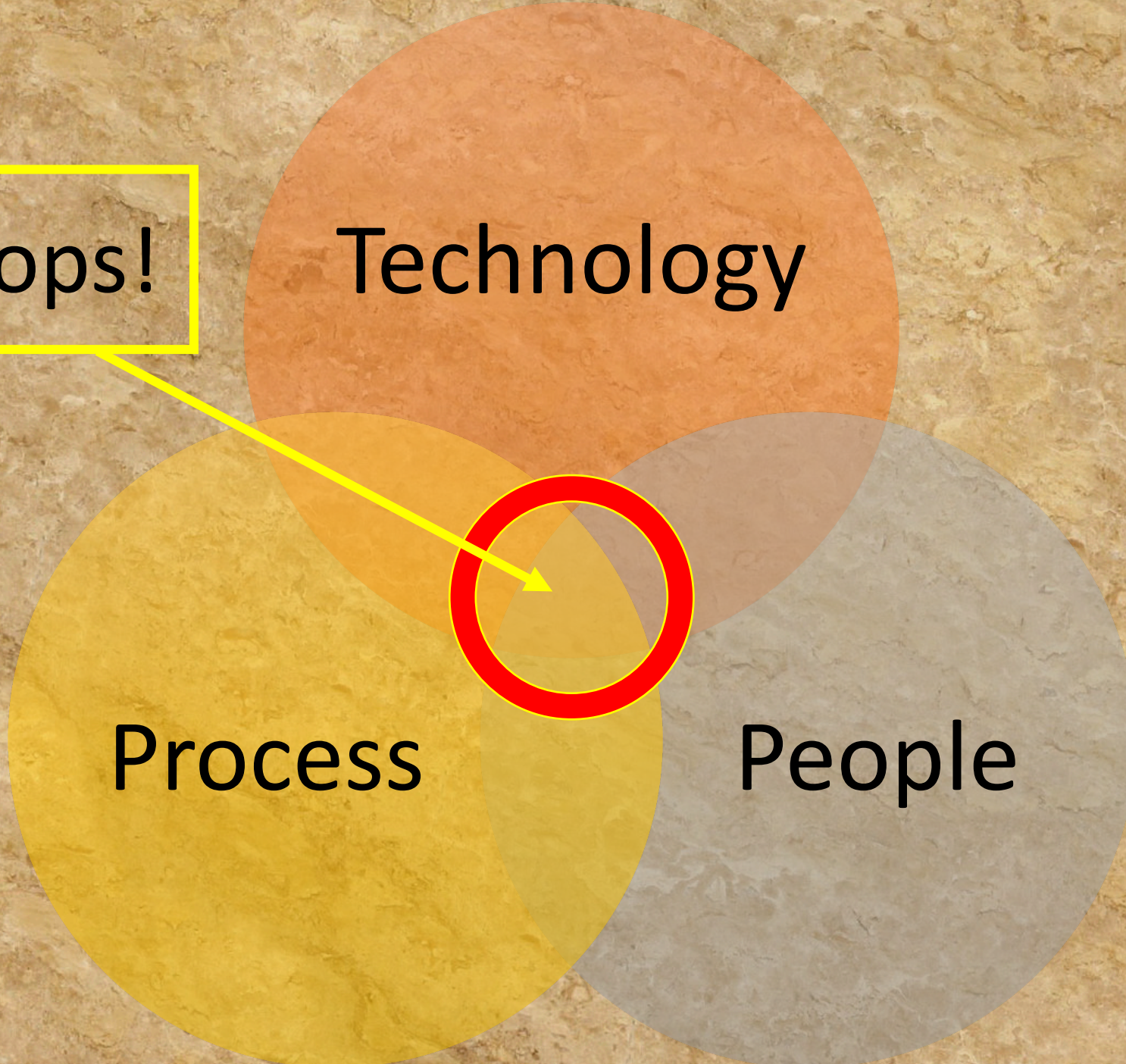
People

Devops!

Technology

Process

People



# PENTAGON INK



# ACT I

*FIRE BRIGADE A.K.A. REACTIVE OPS*



# SETTING THE STAGE

# PEOPLE

🏺 → DEVS


🏺 ON PREM BACKGROUND (DEFENCE!)

🏺 SMART PEOPLE

🏺 HIPSTERS!

🏺 JS, NODE, REACTIVE, MIKROSERVICES

# PROCESS

 DEV - KANBAN

 QA - TDD, UNIT & INTEGRATION AUTOMATED TESTS

 OPS - SERVERLESS NOOPS

# TOOLS



JIRA

GitHub

Travis CI

AWS  
Beanstalk

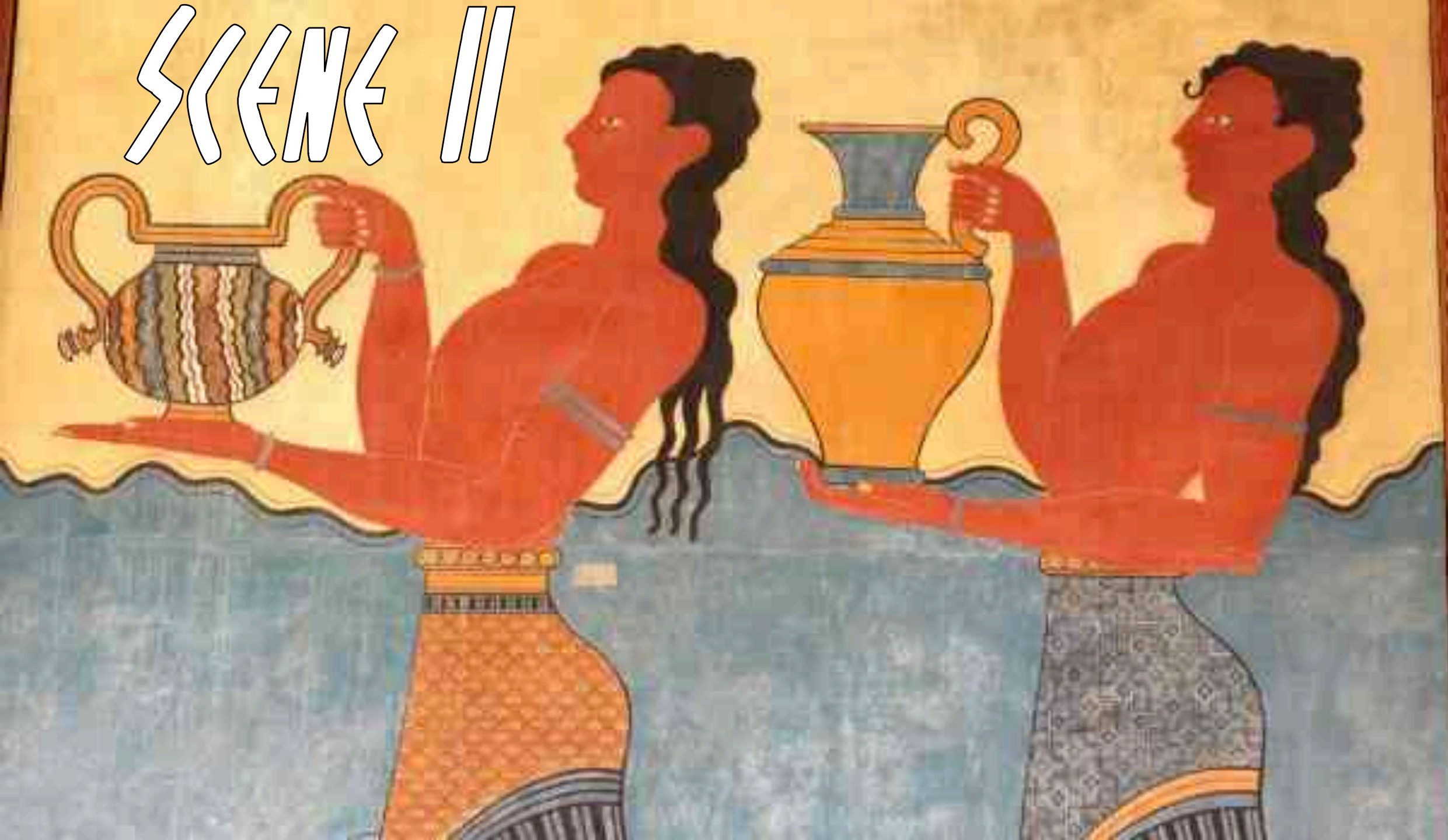
***ACTION***



# SCENE I



# SCENE II



A dramatic painting depicting a city in flames. In the foreground, a body of water reflects the fire, with several boats, some of which are also on fire. The middle ground shows a large, multi-story building with a central section that is heavily engulfed in bright orange and yellow flames. To the left, a tall, dark stone tower with arched windows stands amidst the smoke. To the right, another building with a classical facade is visible. The background is filled with thick, dark smoke and more distant fires, creating a sense of widespread destruction. The overall color palette is dominated by the dark blues and greys of the smoke and buildings, contrasted with the vibrant oranges and yellows of the fire.

# SCENE III



# SCENE IV





Software

# How one developer just broke Node, Babel and thousands of projects in 11 lines of JavaScript

Code pulled from NPM – which everyone was using

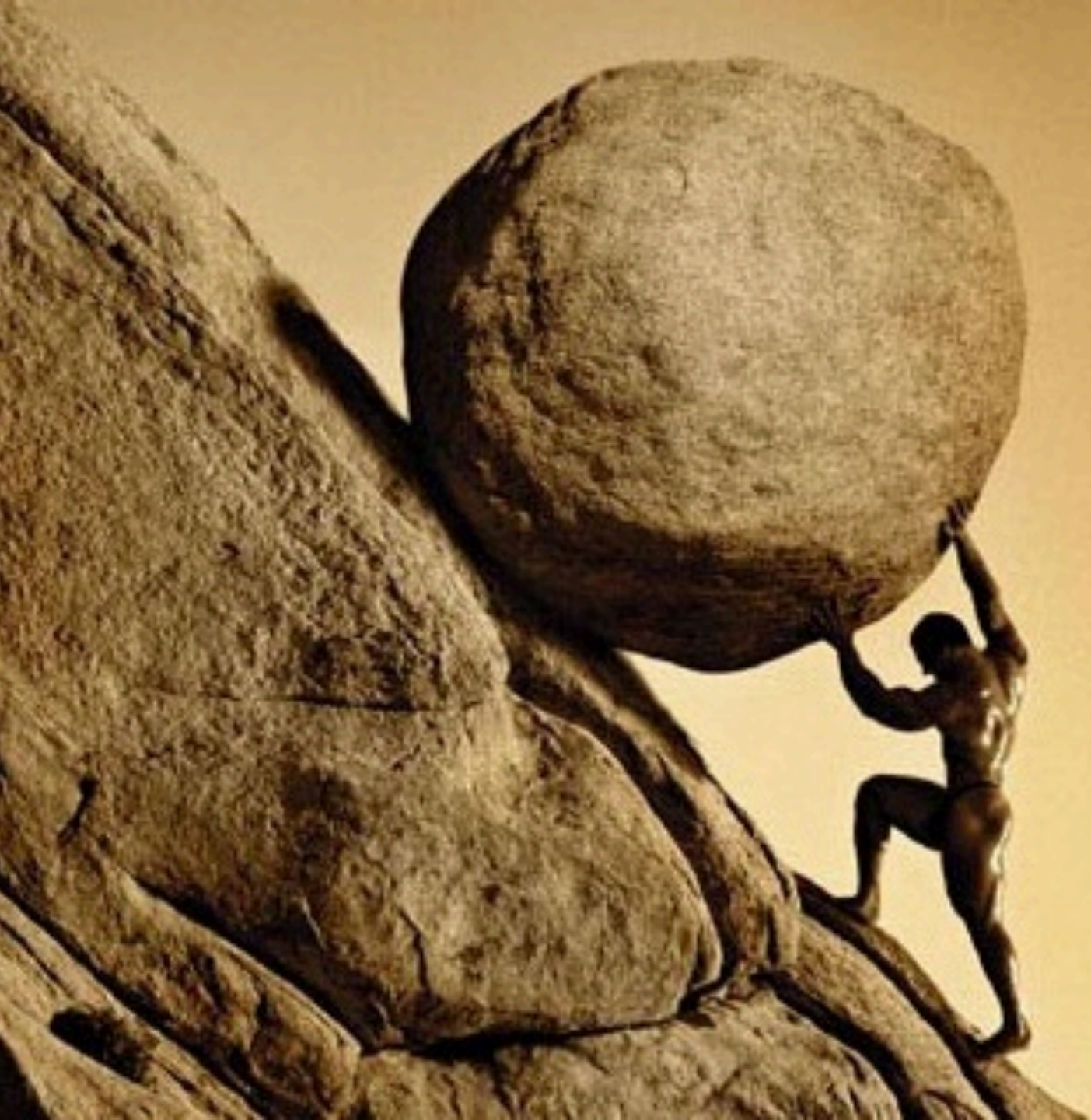
By [Chris Williams](#), US editor 23 Mar 2016 at 01:24

SHARE ▼



Careful, careful ... Don't fumble this like the JS world (Credit: Claus Rebler)

**Updated** Programmers were left staring at broken builds and failed installations on Tuesday after someone toppled the Jenga tower of JavaScript.



# *INTERLUDE*

# ACT II

*SMOKE ALARM INSTALLERS A.K.A.  
REACTIVE IMPROVEMENT*



# SETTING THE STAGE

# NEW IN PEOPLE

🏺 ROUND A: 26 DEVELOPERS

🏺 1 DEVELOPER WITH OPS BACKGROUND

🏺 ~100 CUSTOMERS

🏺 SUPPORT TEAM

# NEW IN PROCESS

🏺 DEV - SCRUM

🏺 QA - EXPLORATORY TESTING

🏺 OPS - WE GOT A GUY THAT KNOWS THIS SH\*T!

🏺 DEVELOPER ON CALL

🏺 LOGS AND CLOCKS IN UTC. FOR \* SAKE!

# TOOLS



JIRA +  
Confluence

GitHub

Travis CI

JFrog  
Artifactory

AWS  
Beanstalk

Sumologic

Pingdom

*ACTION*



# SCENE I



# SCENE II





# SCENE III

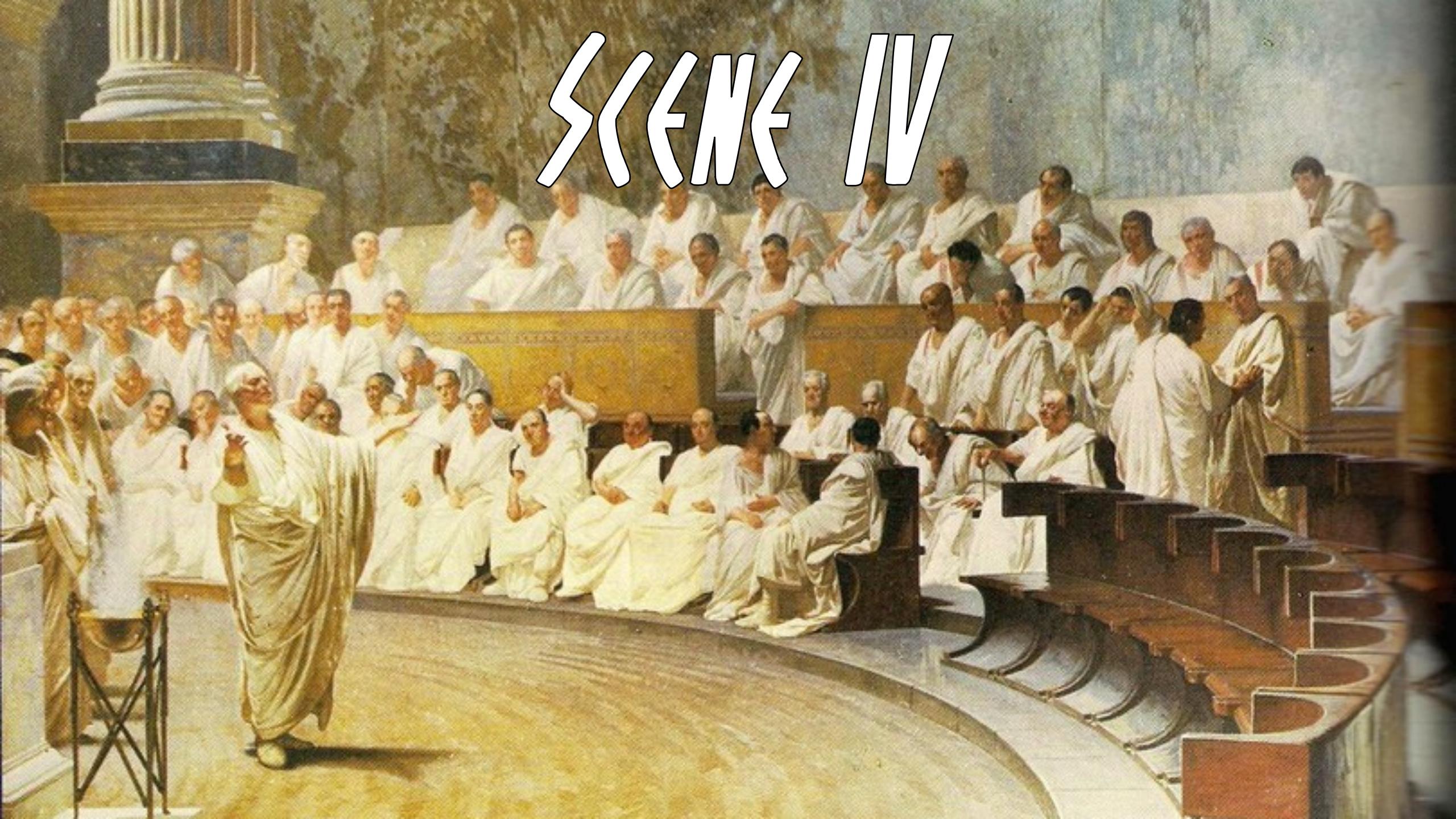


A man with dark hair, wearing a brown coat and a grey scarf, is speaking into a microphone. He is outdoors at a crowded event, with many people in winter clothing visible in the background. A red dot and the word "REC" are visible in the lower-left corner of the frame.

REC

It's Groundhog Day again.

# SCENE IV



# Root Cause Analysis Report

Environment	[REDACTED]	What was affected	Onboarding [REDACTED] Apps
Date RCA Completed	Friday June 17th, 2016	Date issue occurred	Wednesday June 15th, 2016
RCA Team Members	[REDACTED]	Time issue started	Wednesday June 15th, 2016 10:04 PM CST
ReferenceID	DE175504	Time issue stopped	Thursday June 16th, 2016 10:44 PM CST
Any flowdock references		Total time services affected	24:40 Hours

1

## EVENTS:

*describe timeline of events*

June 15, 2016 10:04 PM CST - [REDACTED] fails to onboard in production during smoke tests and verifying hotfix on [REDACTED] account imports

June 15, 2016 10:35 PM CST - [REDACTED] finds logging indicating that there are significant issues with hero, others jump into debugging session for hero

June 16, 2016 12:14 AM CST - [REDACTED] verifies that ServiceNow onboarding still works in staging, [REDACTED] identifies log error in production indicating access token could not be retrieved

June 16, 2016 02:55 AM CST - Kron service is stopped in production, resulting in temporary fix of hero issues. It is verified that [REDACTED] still fails to onboard.

June 16, 2016 03:24 AM CST - After further debugging of logs an environment by the team, it is determined that the production environment being still configured with global Oauth credentials for [REDACTED] is the most likely cause. It is decided to continue with verification in the morning.

June 16, 2016 11:48 AM CST - [REDACTED] are able to verify in a local development environment that defined global oauth credentials is the issue and debugged hero to find root defect, which gets opened separately

June 16, 2016 10:44 PM CST - production is redeployed without global oauth credentials for [REDACTED] configured, resolving the issue

2	<b>SYMPTOMS:</b> <i>describe the symptoms</i>	Onboarding [REDACTED] on production never succeeded. Test connection calls for [REDACTED]s always failed and returned with a 403 response (after temporarily resolving Hero issues that would result in occasional 500s). Hero logs indicate that there was an error acquiring the [REDACTED]ss token and that the User was not authenticated, despite using credentials that worked on other unaffected environments.
3	<b>WHAT HAPPENED:</b> <i>technical description of events</i>	<p>Smoke testing was happening in production after a hotfix deployment to fix [REDACTED] onboarding while importing certain kinds of data. While I was able to verify this on staging, I could not even get past the ServiceNow add page because it could not properly authorize the connection, despite using credentials that allowed for successful onboarding in other environments.</p> <p>At the same time, we were discovering significant issues with hero performance that resulted in more than occasional hero tasks failing to respond properly, resulting in failures with a different error code. After investigation of those issues and a temporary fix implemented via shutting down of Kron, the [REDACTED] onboarding continued to fail, though now consistently with authentication errors.</p> <p>After theorizing that the tenant credentials were not actually being used to perform the auth, [REDACTED] discovered that there were global auth credentials for [REDACTED] still configured in production environment, where the staging environment had no such configs. As that was believed to be the only significant difference between the two environments, we believed that the global auth keys were still being used.</p> <p>After verifying this in a development environment the next morning, the production environment was redeployed later that evening with the hotfix for hero and the global auth tokens for [REDACTED] removed from configuration. This resolved the [REDACTED] onboarding issue.</p>

4	<b>ROOT CAUSE:</b> <i>the root cause</i>	The short term effective cause was that the production environment was configured with global [REDACTED] tokens from the past that did not get cleared out when no longer needed and were no longer valid. However, the [REDACTED] syncer config and hero code is supposed to be set up to explicitly ignore global oauth configs, but a bug in hero still gave those configs precedence. A separate defect was filed for the hero bug.
5	<b>NEXT STEPS:</b> <i>what actions can be taken to eliminate this issue from occuring again</i>	<ol style="list-style-type: none"><li>1. Fix the hero defect causing global oauth keys to always take precedence, even when configured not to [https://rally1.rallydev.com/[REDACTED]</li><li>2. Make an official process for which code changes that require config changes are specially noted and marked such that those configuration changes are also done in the staging and production environments before deployment</li></ol>



# RETROSPECTIVE





**GitLab**  
@gitlab

 Follow



So, what happened yesterday? Here's what went wrong and what we're doing to fix it:



#### **GitLab.com Database Incident**

Yesterday we had a serious incident with one of our databases. We lost six hours of database data (issues, merge requests, users, comments, snippets, etc.) for GitLa...

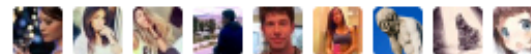
[about.gitlab.com](https://about.gitlab.com)

RETWEETS

**394**

LIKES

**358**



5:05 AM - 1 Feb 2017



36



394

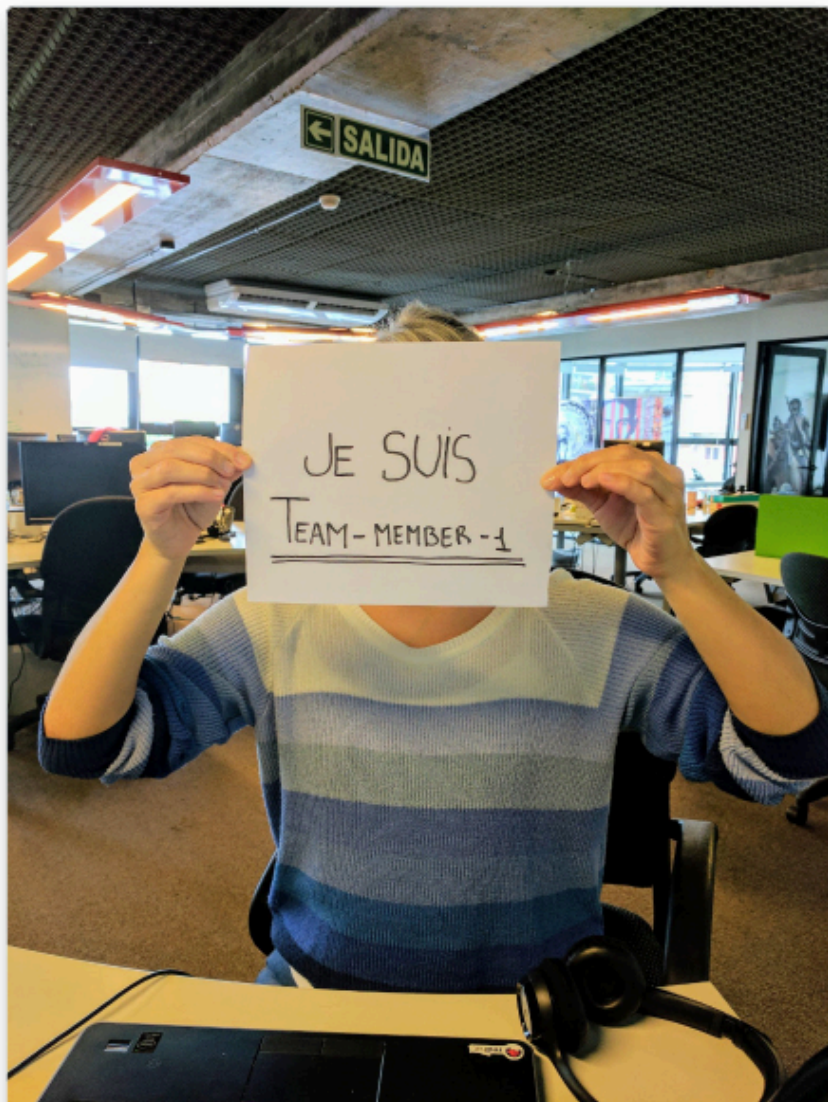


358

# Problems Encountered

- LVM snapshots are by default only taken once every 24 hours. *Team-member-1* happened to run one manually about six hours prior to the outage because he was working in load balancing for the database.
- Regular backups seem to also only be taken once per 24 hours, though *team-member-1* has not yet been able to figure out where they are stored. According to *team-member-2* these don't appear to be working, producing files only a few bytes in size.
- *Team-member-3*: It looks like `pg_dump` may be failing because PostgreSQL 9.2 binaries are being run instead of 9.6 binaries. This happens because omnibus only uses Pg 9.6 if data/PG\_VERSION is set to 9.6, but on workers this file does not exist. As a result it defaults to 9.2, failing silently. No SQL dumps were made as a result. Fog gem may have cleaned out older backups.
- Disk snapshots in Azure are enabled for the NFS server, but not for the DB servers.
- The synchronisation process removes webhooks once it has synchronised data to staging. Unless we can pull these from a regular backup from the past 24 hours they will be lost
- The replication procedure is super fragile, prone to error, relies on a handful of random shell scripts, and is badly documented
- Our backups to S3 apparently don't work either: the bucket is empty
- So in other words, out of five backup/replication techniques deployed none are working reliably or set up in the first place. We ended up restoring a six-hour-old backup.
- `pg_basebackup` will silently wait for a master to initiate the replication progress, according to another production engineer this can take up to 10 minutes. This can lead to one thinking the process is stuck somehow. Running the process using "strace" provided no useful information about what might be going on.





**kiru**  
@karmukis

Follow

#JeSuisTeamMember1 #GitLab

11:31 AM - 1 Feb 2017

17 56



Gift from Google



Gift from Codefresh

Yes, *team-member-1* is doing very well!

Coincidentally, just before the DB incident, *team-member-1* had qualified for a promotion to senior developer. The outage did not change that decision.



Yorick Peterse

@yorickpeterse · Member since August 4, 2015

[yorickpeterse@gmail.com](mailto:yorickpeterse@gmail.com) ·  · [yorickpeterse.com](http://yorickpeterse.com) ·  The Netherlands ·  GitLab

Database (removal) Specialist at GitLab

# ACT III

*TRAGEDY CULMINATION,  
A.K.A. THE PERFECT STORM*



# SETTING THE STAGE

# NEW IN PEOPLE

🏺 ROUND B: 74 DEVS

🏺 5 DEV WITH OPS BG

🏺 1 PERFORMANCE ENG

🏺 CHIEF ARCHITECT

🏺 CUSTOMER SUCCESS TEAM

🏺 LEGAL

🏺 CFO

🏺 ~1000 CUSTOMERS

# NEW IN PROCESS

🏺 DEV - SAFE

🏺 QA - SYSTEM TESTING

🏺 OPS - OPS TEAM (WAT?!)

🏺 ESCALATION PATH: SME AND MANAGER ON CALL

🏺 SOC II

🏺 NONFUNCTIONAL BACKLOG

# TOOLS



JIRA + Confluence

GitHub

Configuration  
Management

Travis CI

JFrog  
Artifactory

AWS  
Beanstalk

Sumologic

APM

C O N G R A T S

On your promotion.

***ACTION***



# SCENE I



# SCENE II

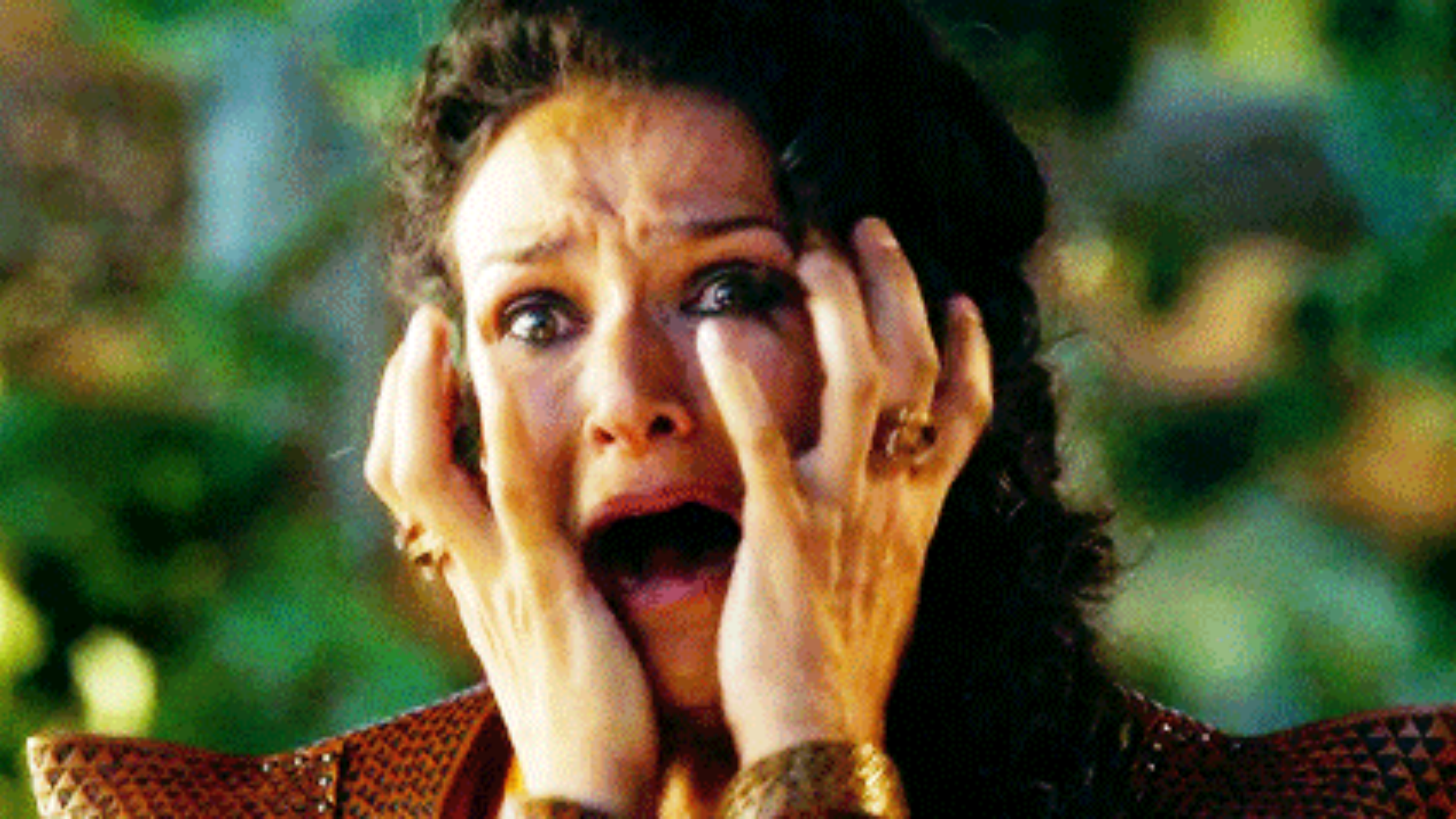






# SCENE III

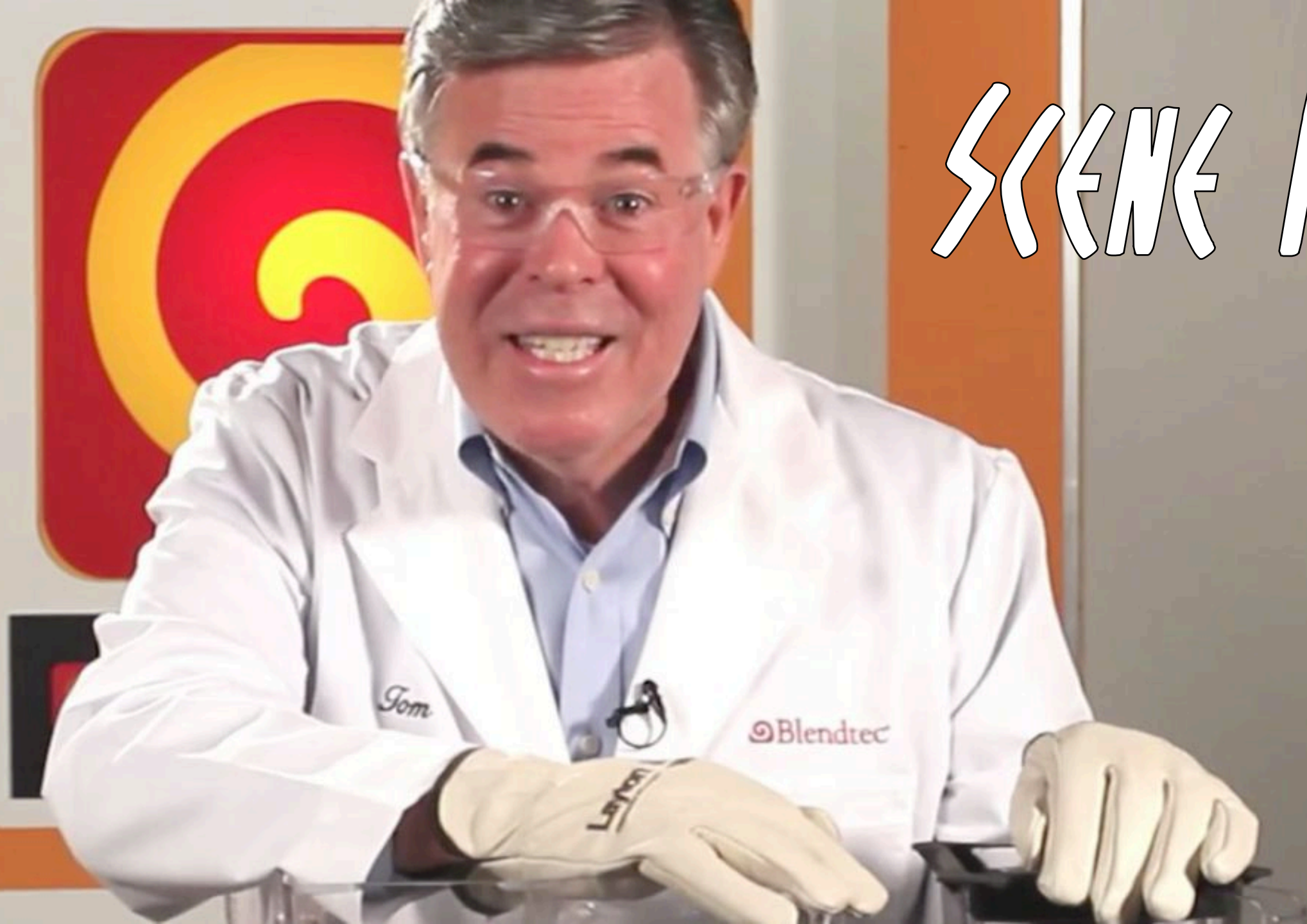




THE



# SCENE IV







This is where we hold them





# EPILOGUE

*SMOKEY BEAR A.K.A. PROACTIVE IMPROVEMENT*



# NEW IN PROCESS

- 🏺 PERFORMANCE AND SCALABILITY TESTING
- 🏺 LICENSE AND SECURITY MANAGEMENT
- 🏺 PROACTIVE PERFORMANCE AND TRENDS REVIEW
- 🏺 NON-FUNCTIONAL DEFINITION OF DONE AND ESTABLISHED PATTERNS

# TOOLS



JIRA + Confluence

GitHub

Configuration  
Management

Travis CI

JFrog  
Artifactory +  
Mission Control

JFrog Xray

Blazemeter

Service  
Virtualization

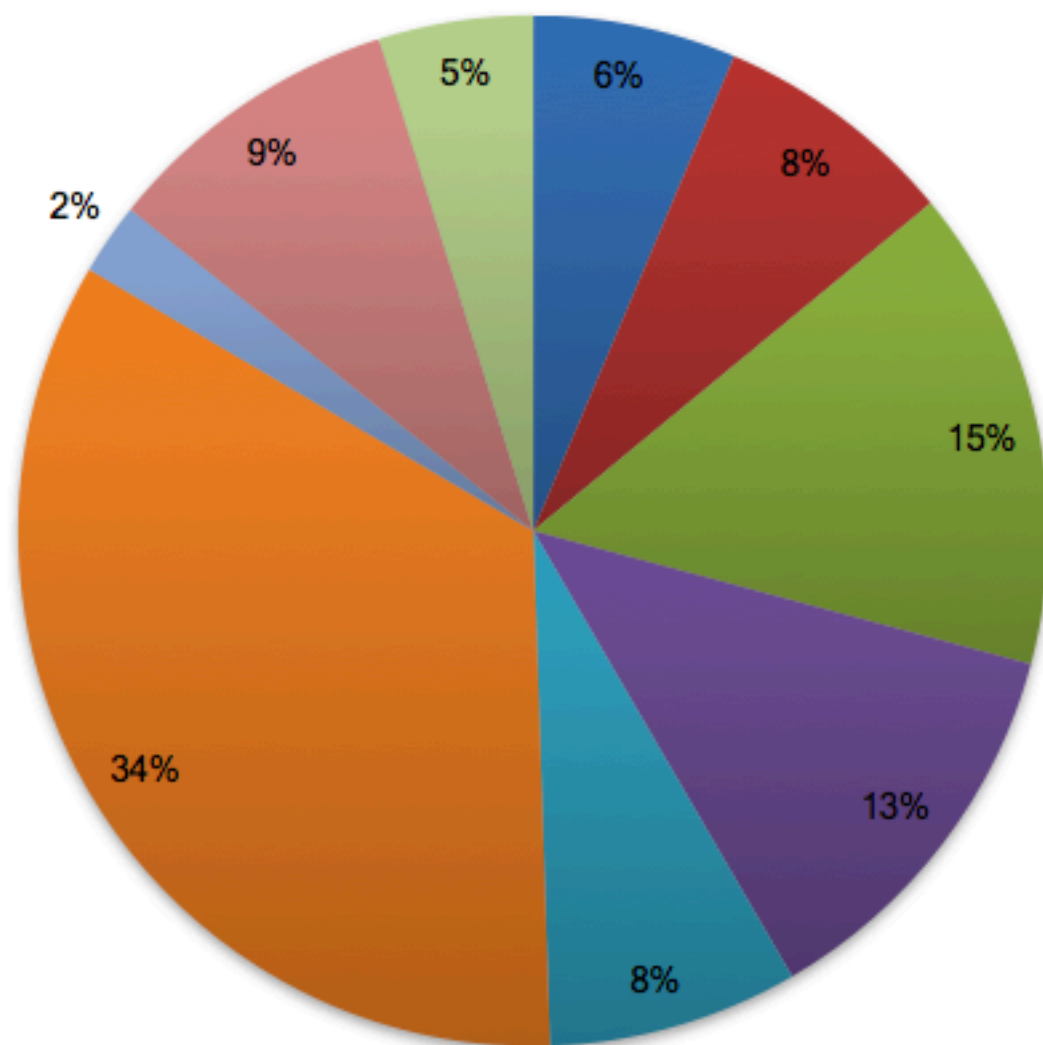
Sumologic

APM

THE



## Engineering effort allocation



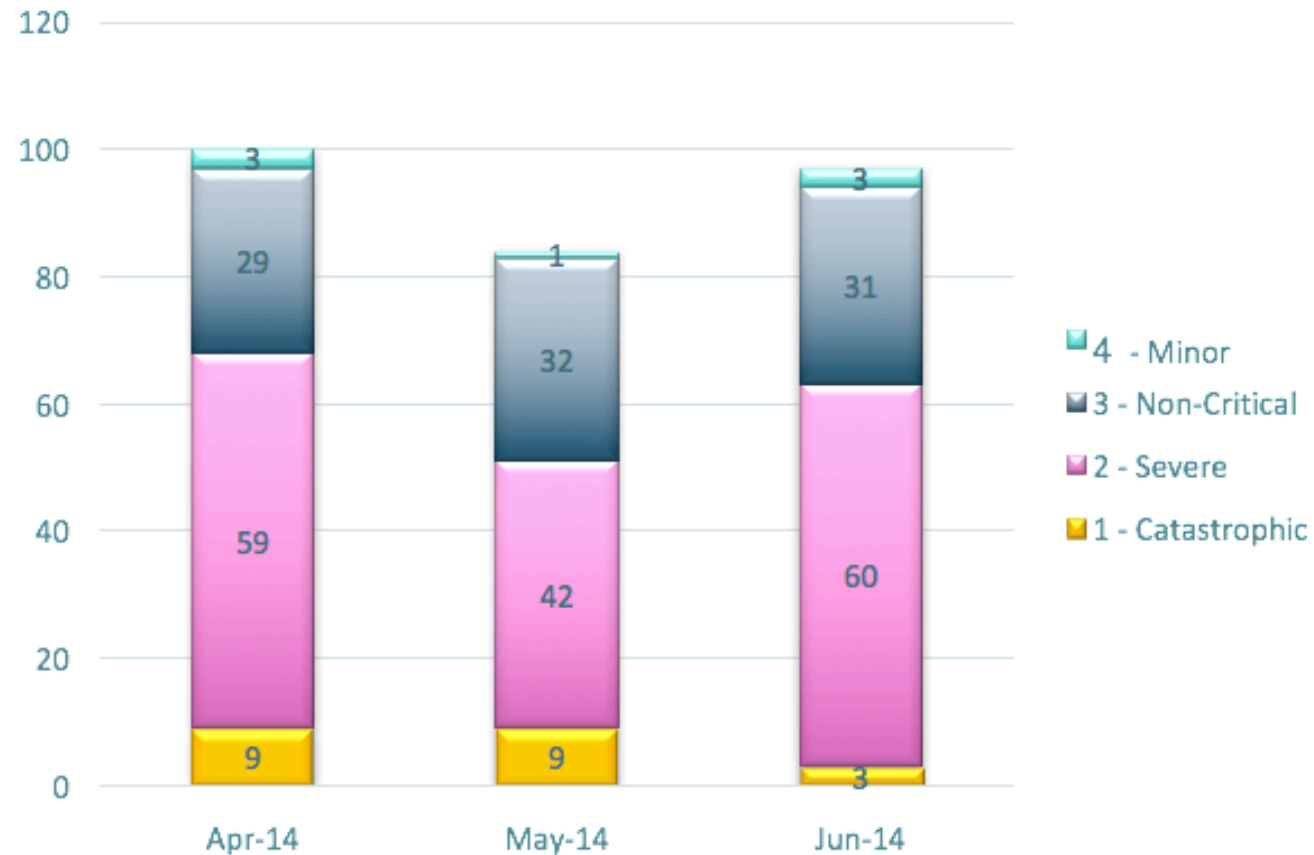
- Improve Eng. Velocity
- Decrease Customer MTTR
- Reduce TCO
- Big Feature A
- Fulfull customer and field commitments
- Keep the lights on
- Corporate Initiatives
- Quality Improvements
- Uncategorized + Research for Future Releases



This is where we hold them

## Q1FY15 Customer Defects by Severity

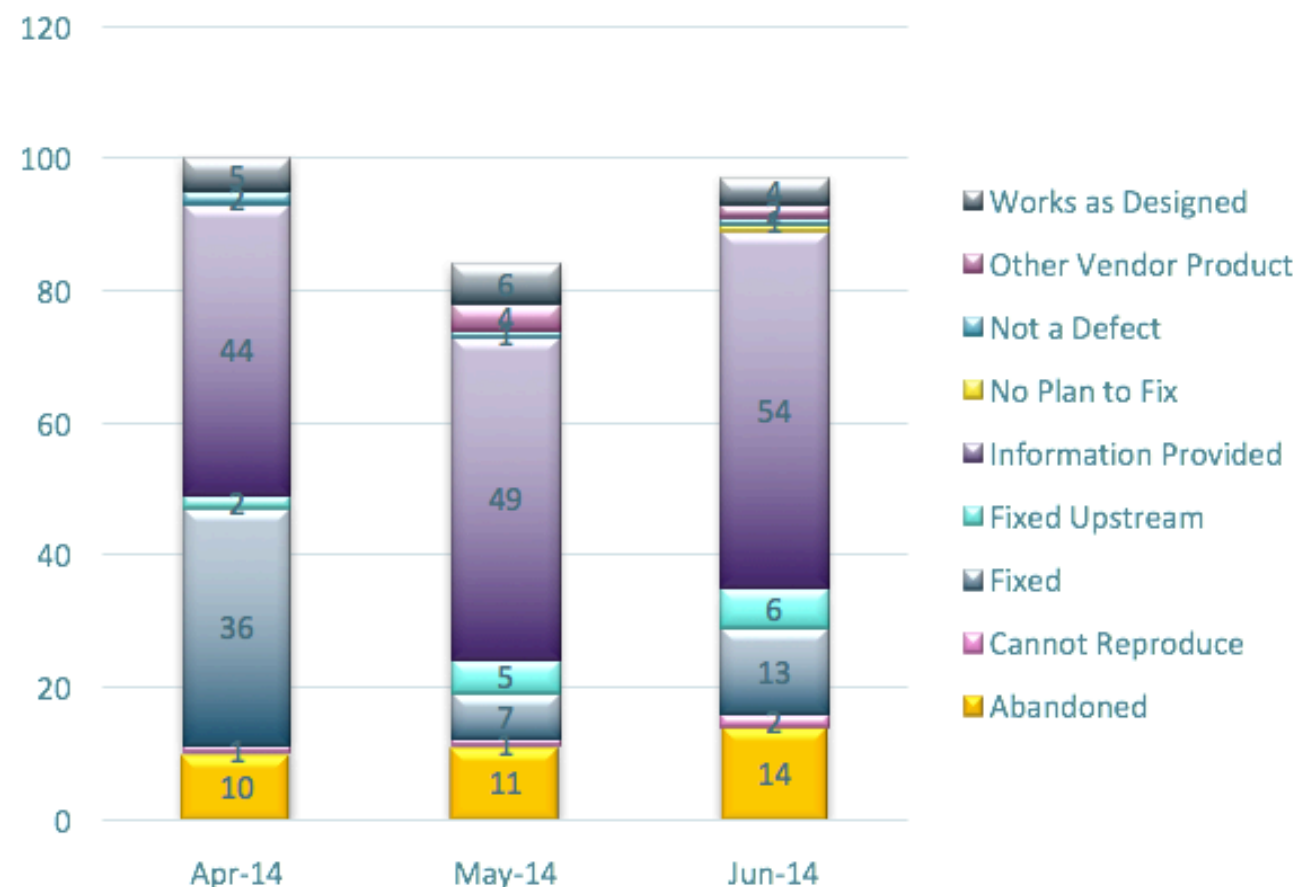
---



**281** – Total Q1FY15 Customer Defects(escalated by Support to Engineering)

**182 (65%)** – S1 & S2 Tickets

# Q1FY15 Customer defects by Resolution



**56 (19.9%)** defects– Resolution = Code Fix

**147 (52.3%)** defects– Resolution = Information Provided



ΕΡΙΜΥΘΗ

SCALE IS A... SCALE!





A Venn diagram consisting of three overlapping circles. The top circle is orange and labeled 'Tools'. The bottom-left circle is yellow and labeled 'Process'. The bottom-right circle is grey and labeled 'People'. The circles overlap in the center and at the intersections of two circles. The background is a textured, light brown surface.

Tools

Process

People

Change!

Technology

YOU BUILD IT YOU  
OWN IT

PAIN IS INSTRUCTIONAL



Process

People

DATA IS THE KEY!  
(EXCEL IS OK)



# Q&A AND TWITTER ADS

 @JBARUKH

 @LIGOLNIK

 #TECHSTRONG

 [HTTPS://JFROG.COM/SHOWNOTES](https://jfrog.com/shownotes)