



# VCU

Virginia Commonwealth University  
**VCU Scholars Compass**

---

Capstone Design Expo Posters

College of Engineering

---

2017

## Digital Analysis of Heartbeats from Remote Machines

Dewey Mitchem

*Virginia Commonwealth University*

Devin Dulay

*Virginia Commonwealth University*

Taylor Kaplan

*Virginia Commonwealth University*

Follow this and additional works at: <https://scholarscompass.vcu.edu/capstone>



Part of the [Computer Engineering Commons](#)

© The Author(s)

---

Downloaded from

<https://scholarscompass.vcu.edu/capstone/205>

This Poster is brought to you for free and open access by the College of Engineering at VCU Scholars Compass. It has been accepted for inclusion in Capstone Design Expo Posters by an authorized administrator of VCU Scholars Compass. For more information, please contact [libcompass@vcu.edu](mailto:libcompass@vcu.edu).



# Digital Analysis of Heartbeats from Remote Machines

CS 321 | Team members: Dewey Mitchem, Devin Dulay, Taylor Kaplan | Sponsor: Dephinitive | Sponsor adviser: Martin Carr

## Data Visualization

### Problem:

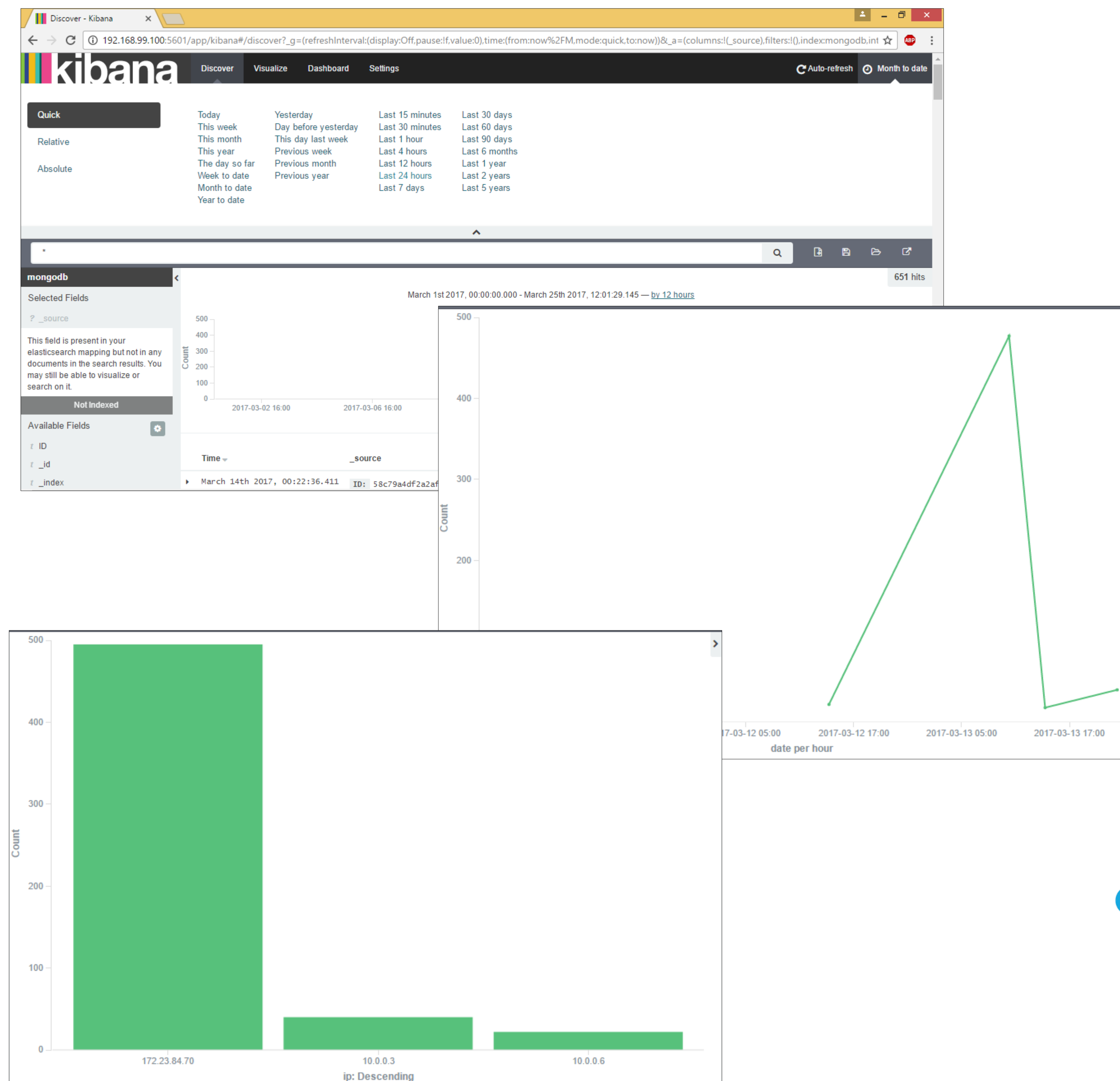
Computers inevitably run into issues and encounter errors. However these issues are not identified until the system stopped entirely.

To solve this problem, we developed a RESTful Web API that will accept JSON Contracts for configured computers to send heartbeats to a central location.

### Purpose

While the API is active on a given host/server, machines that are required to check in will contact our machine with JSON contracts with their status.

The method that was chosen provides database security in the sense that RabbitMQ acts as a broker between the API being called and the deposit of the contract into MongoDB. Thus the database cannot be accessed by bad contracts.



### How it Works

1. Computer system recognizes there is a problem.
2. Makes a call to our Web-Facing API to submit a JSON Contract.
3. System checks the IP and submits the JSON to RabbitMQ.
4. Internal Node program queries RabbitMQ for jobs and cleans the data.
5. Internal Node program then submits contract to MongoDB for storage.
6. Query all documents from MongoDB via ElasticSearch and deposit into Kibana for analysis.

### Technology Used:

