

## Project Development Description:



### [openFDA Enforcement Mapper](http://open-fda-mapper.blueraster.io/)

Blue Raster developed the interactive web mapping application at <http://open-fda-mapper.blueraster.io/> by using openFDA data in response to 18F's RFQ 4QTFHS150004. We are experts at creating custom web applications, helping our clients to analyze data quickly, visualize challenges and opportunities, and make sound decisions using geospatial intelligence. Our agile development strategy emphasizes collaboration and personal attention at every stage.

This application consumes data from <https://api.fda.gov/food/enforcement.json>, and consists of two main components:

- 1) The server-client, used for geocoding enforcement data from the openFDA data and web serving
- 2) The browser-client, used to request data from the openFDA API and matching records with geocoded data from the server-client. We cluster records based on spatial location using the ArcGIS API for JavaScript, chart information using Chart.js, and display the results on an interactive map.

#### A) Assign One Product Manager

Blue Raster assigned one leader, Kevin McMaster, an experience project manager, and held him accountable for the delivery of a quality prototype and supporting documentation within the scheduled period of performance.

#### B) Multi-disciplinary Team

To support this solution development, in addition to the Product Manager, we established the roles of a Technical Architect (Stephen Ansari), a Frontend Web Developer (Albert Marquez), and a Backend Web Developer (Jon Nordling).

#### C) Technology Used

By the project mid-point, we identified and implemented the following modern and open-source technologies:

- 1) Source code pre-processing: Babel, Jade and Stylus
- 2) JavaScript ES6 & 7
- 3) Amazon Web Services / EC2 Instance Docker container
- 4) Node.js / Express
- 5) Data Visualization: Chart.js
- 6) React
- 7) Flux
- 8) Container: Docker
- 9) Map interface: ArcGIS JavaScript API
- 10) Jenkins

#### D) IaaS provider

We deployed the prototype using the following IaaS provider: Amazon Web Services EC2 instance container.

## E) Unit Testing

We wrote both unit tests and functional tests for the code. For example, one test makes API calls to create a reference store and consume geospatial information by referencing openFDA enforcement record data.

## F) Continuous Integration (CI)

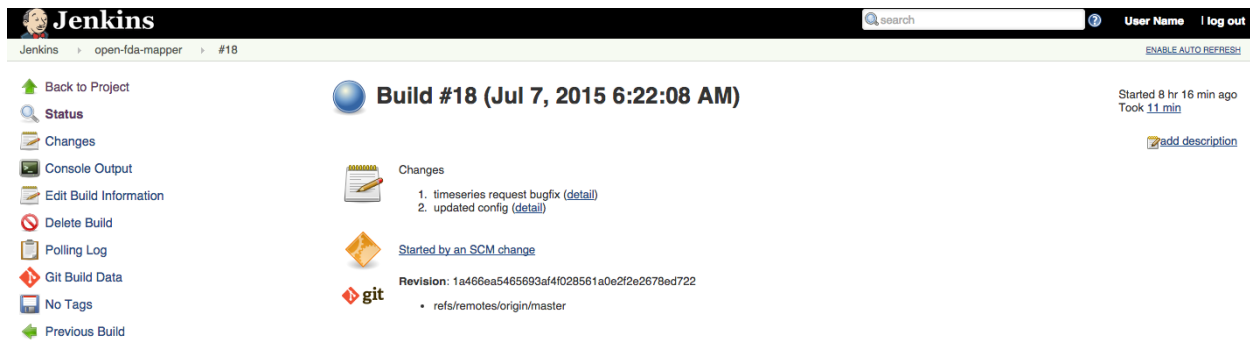
Our team set up Jenkins for CI to monitor the master branch of our GitHub repository <https://github.com/blueraster/open-fda-map-viewer> and deploy automatic builds upon code check-in. We also used the CI to check the API, the application, and make sure the code deployed.

The screenshot shows the Jenkins web interface for the project 'open-fda-mapper'. The top navigation bar includes the Jenkins logo, a search bar, and user information. The left sidebar contains links for 'Back to Dashboard', 'Status', 'Changes', 'Workspace', 'Build Now', 'Delete Project', 'Configure', and 'Git Polling Log'. The main content area displays the project name 'Project open-fda-mapper' and a description: 'OpenFDA Enforcement Mapper Jenkins continues continues integration task'. Below this are links for 'Workspace' and 'Recent Changes'. On the right, there are links for 'edit description' and a 'Disable Project' button. The 'Build History' section shows a list of builds from #1 to #18, with columns for build number, status (indicated by colored circles), and timestamp. The 'Permalinks' section provides links to specific build details, including the last build, last stable build, last successful build, last failed build, and last unsuccessful build.

Build History	trend
#18	Jul 7, 2015 6:22 AM
#17	Jul 7, 2015 5:59 AM
#16	Jul 7, 2015 5:39 AM
#15	Jul 7, 2015 5:37 AM
#14	Jul 7, 2015 5:36 AM
#13	Jul 7, 2015 5:35 AM
#12	Jul 7, 2015 5:34 AM
#11	Jul 7, 2015 5:33 AM
#10	Jul 7, 2015 5:31 AM
#9	Jul 7, 2015 5:27 AM
#8	Jul 7, 2015 5:26 AM
#7	Jul 7, 2015 5:25 AM
#6	Jul 7, 2015 5:09 AM
#5	Jul 7, 2015 5:06 AM
#4	Jul 7, 2015 5:00 AM
#3	Jul 7, 2015 4:59 AM
#2	Jul 7, 2015 4:53 AM
#1	Jul 7, 2015 4:13 AM

**Permalinks**

- [Last build \(#18\), 8 hr 15 min ago](#)
- [Last stable build \(#18\), 8 hr 15 min ago](#)
- [Last successful build \(#18\), 8 hr 15 min ago](#)
- [Last failed build \(#16\), 8 hr 58 min ago](#)
- [Last unsuccessful build \(#16\), 8 hr 58 min ago](#)

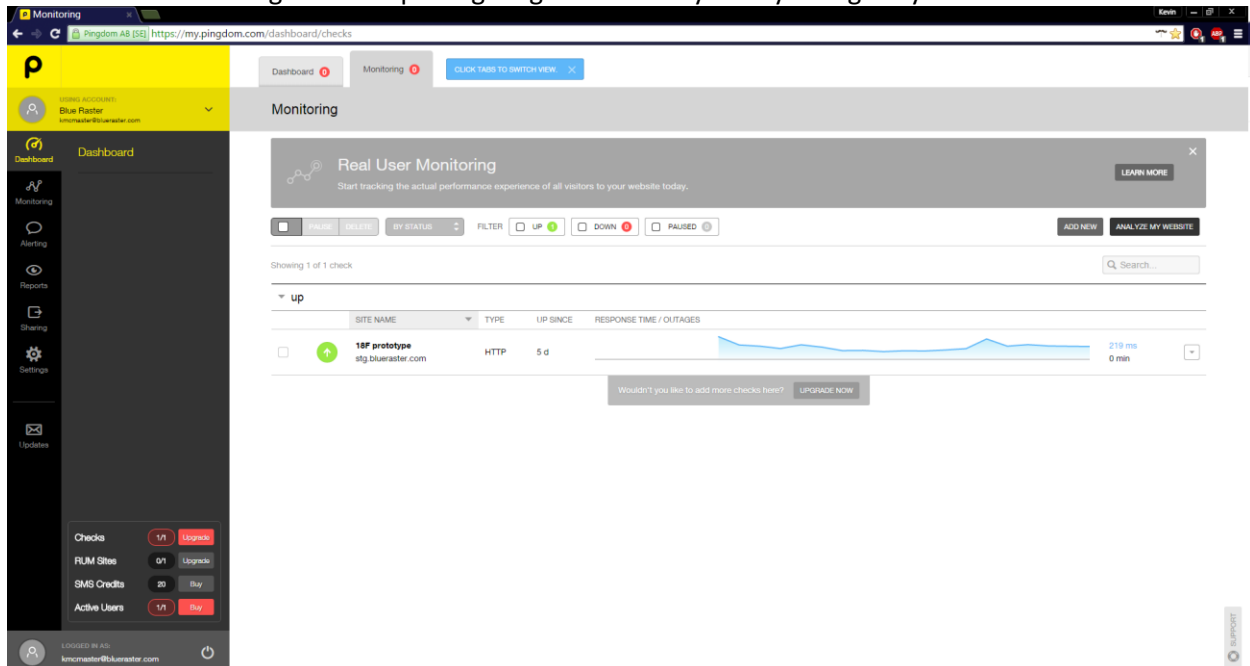


## G) Configuration Management

Configuration management was set up using NPM (node package manager) package .json (one each for server-client and browser-client), a Dockerfile, and build processing using Gulp.

## H) Continuous Monitoring

Continuous monitoring was set up using Pingdom to notify of any outages by email.

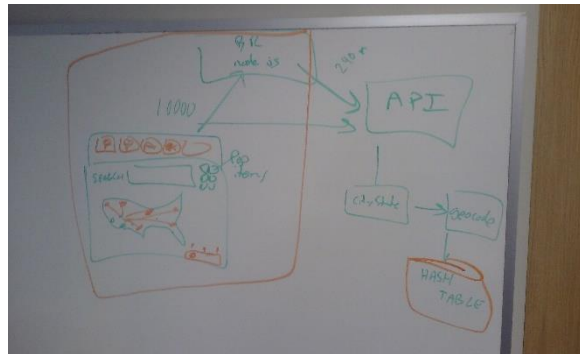


## I) Software Deployment

We deployed the application in a docker container on an AWS EC2 instance.

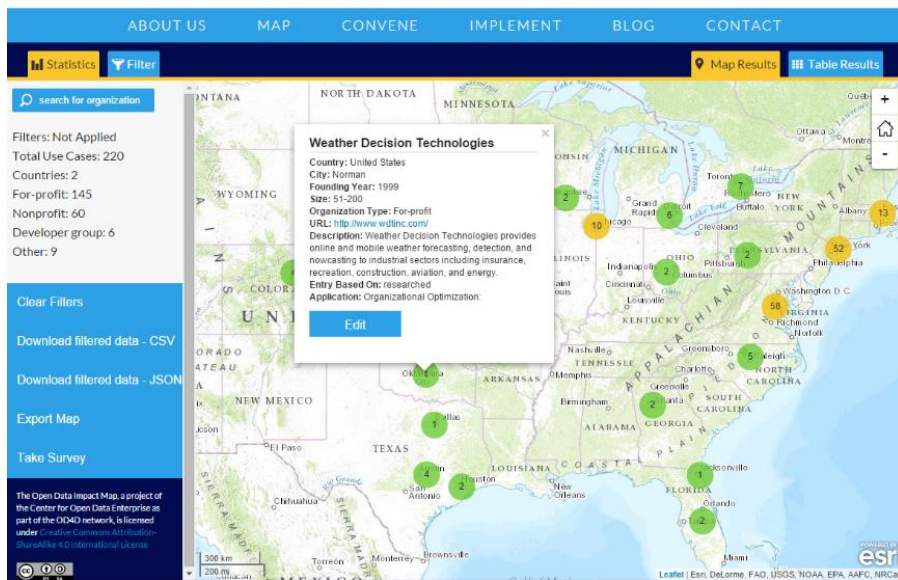
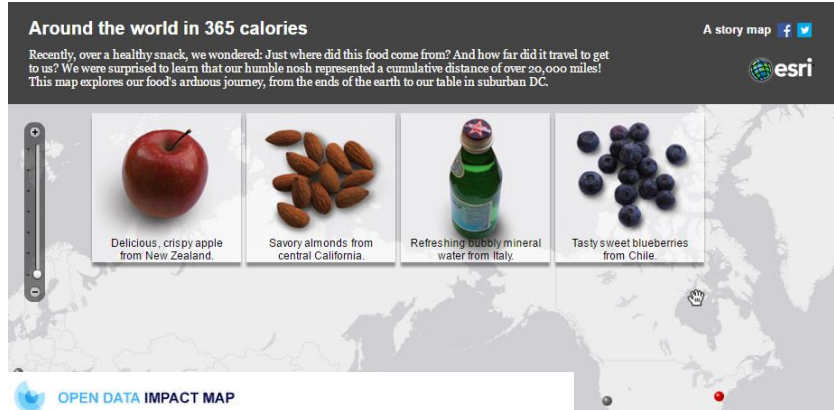
## J) Iterative Approach

The project began with a vision of a mapping application that shows issued enforcement reports and the geography impacted by the recall for various categories of interest. Working from the end goal, we architected the solution of a server client powered by Node.js that pulls data from the openFDA API, geocodes the data based on city and state information using Esri's geocoder restful API, and then stores the geometry. Filtered calls by the browser-client made directly to the openFDA API are returned and matched based on report unique ID to geometry data served by the server-client. A hash system could be used to determine if the enforcement location had been geocoded in the past, so only new locations would require geocoding.



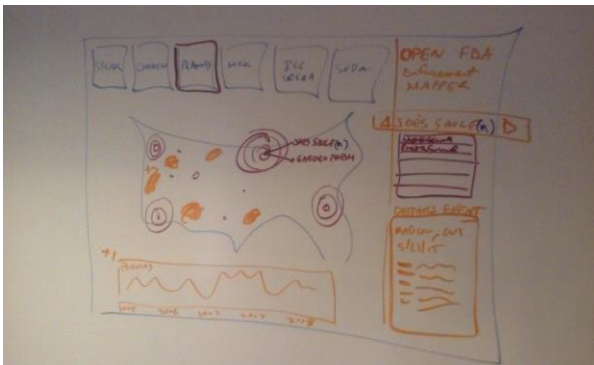
Site design was inspired by a story map at: <http://storymaps.esri.com/snack/>, and cluster map at <http://www.opendataenterprise.org/map/viz/>. We analyzed representative reasons for recall on the enforcement reports and selected the following recall keywords for demonstration purposes:

- 1) Contaminants:
  - a. Chloramphenicol
  - b. Salmonella
  - c. Listeria
- 2) Food
  - a. Ice Cream
  - b. Wheat
  - c. Peanuts
  - d. Salads





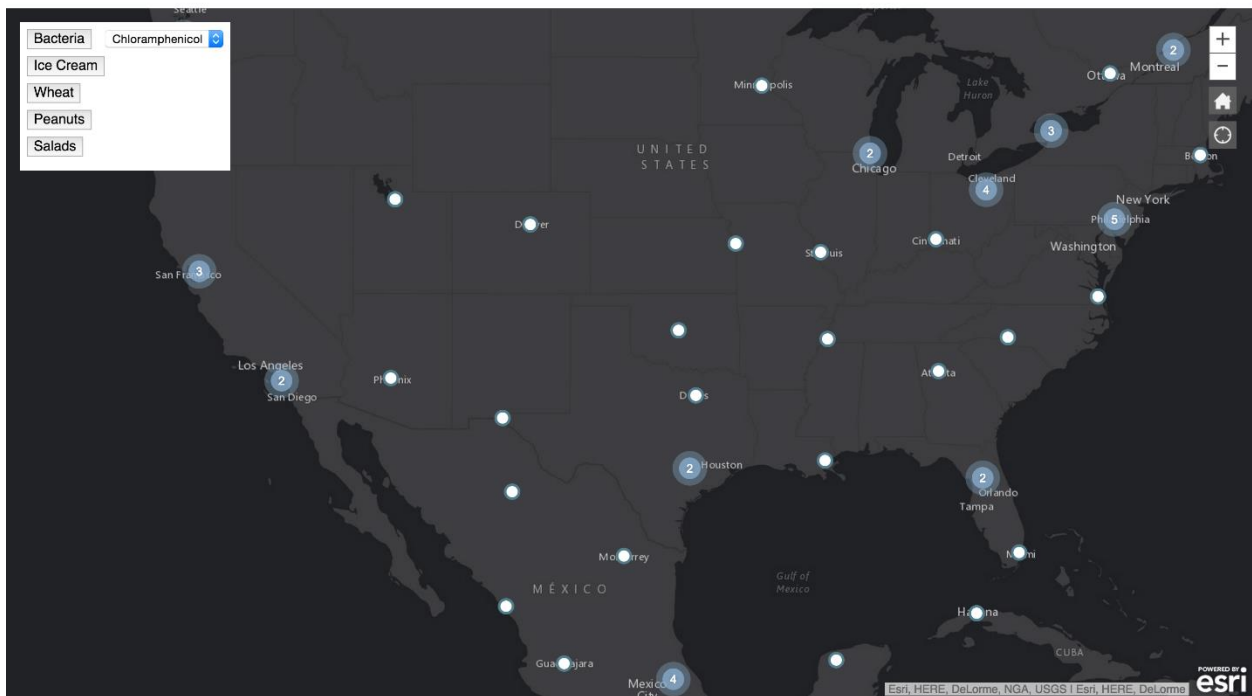
A daily standup meeting to discuss projects, nightly builds, and frequent demonstrations facilitated an iterative development process. Close collaboration between team members ensured a quality, robust prototype deliverable. During progressive demonstrations and application refinement, we modified the wording of the interactive application to make it more accessible to a wider audience, and incorporated the severity of the recall.



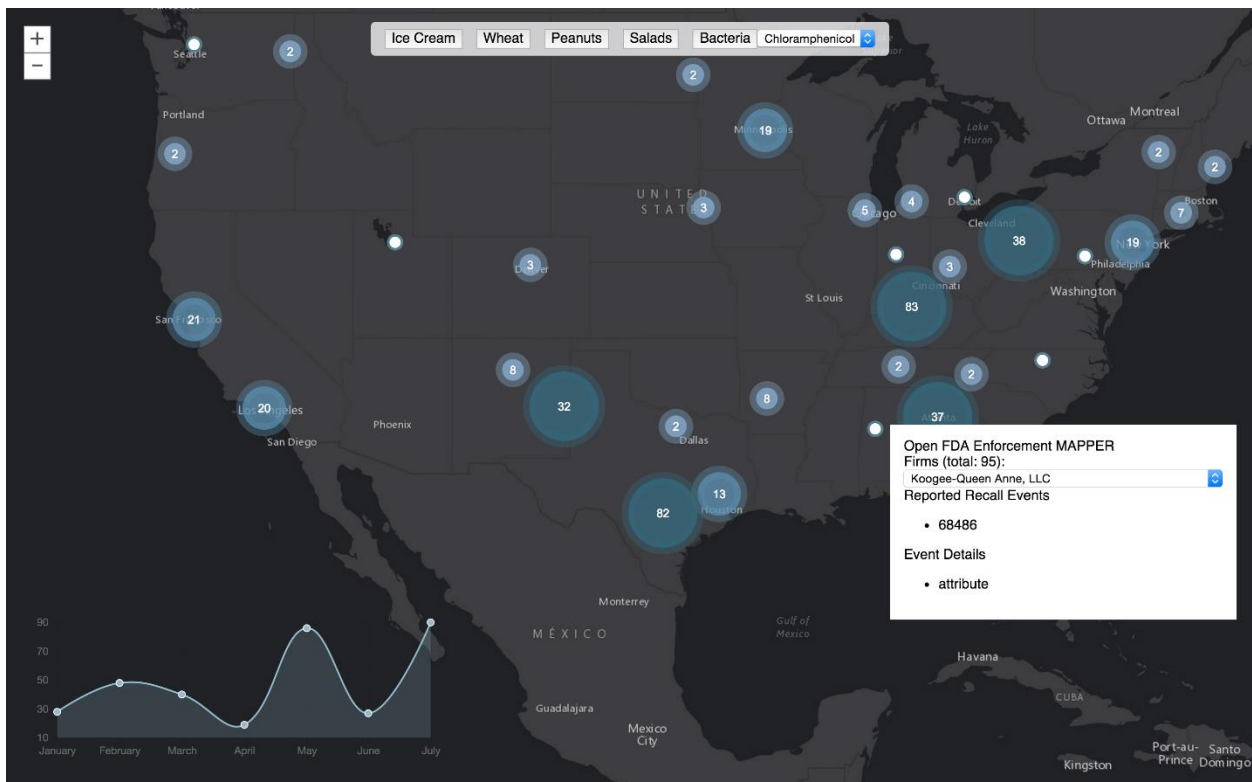
6/26/2015

### Agile Delivery Service Example Using openFDA Data: Geo-Enabling Enforcement Reports

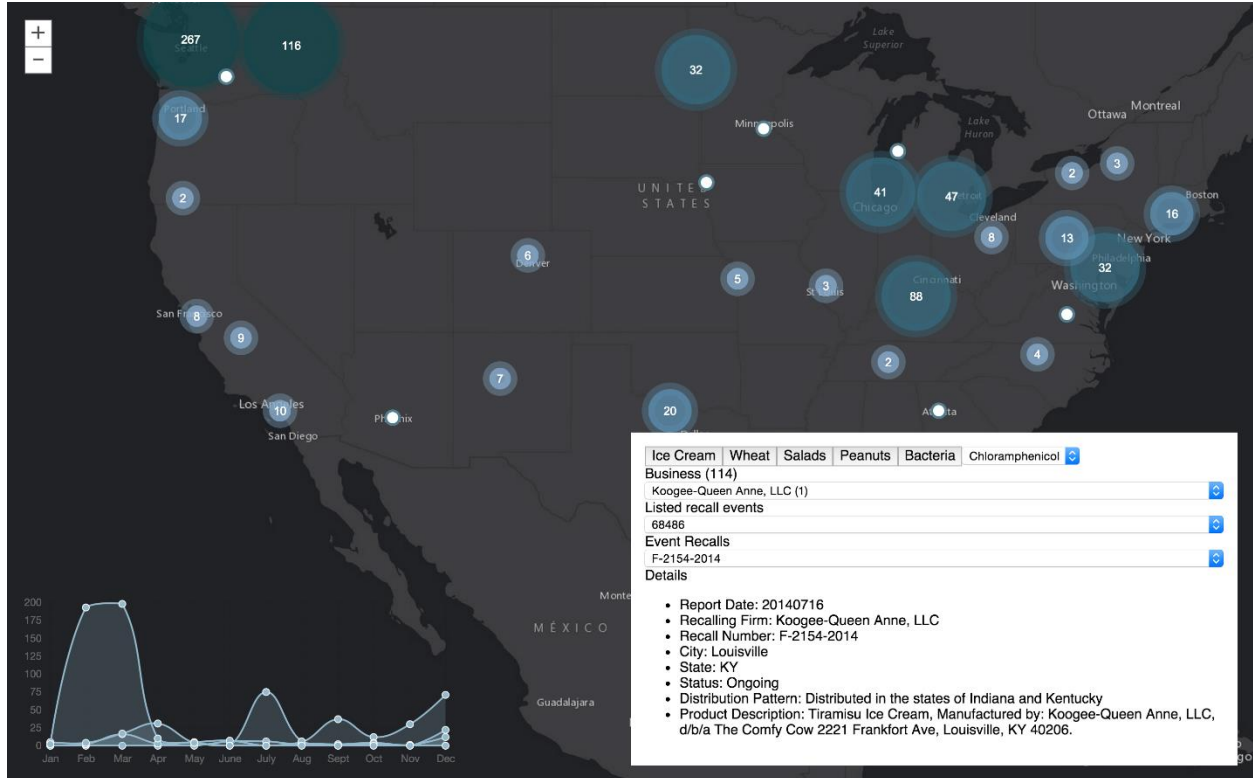
Enforcement reports throughout the U.S. mentioning the items below



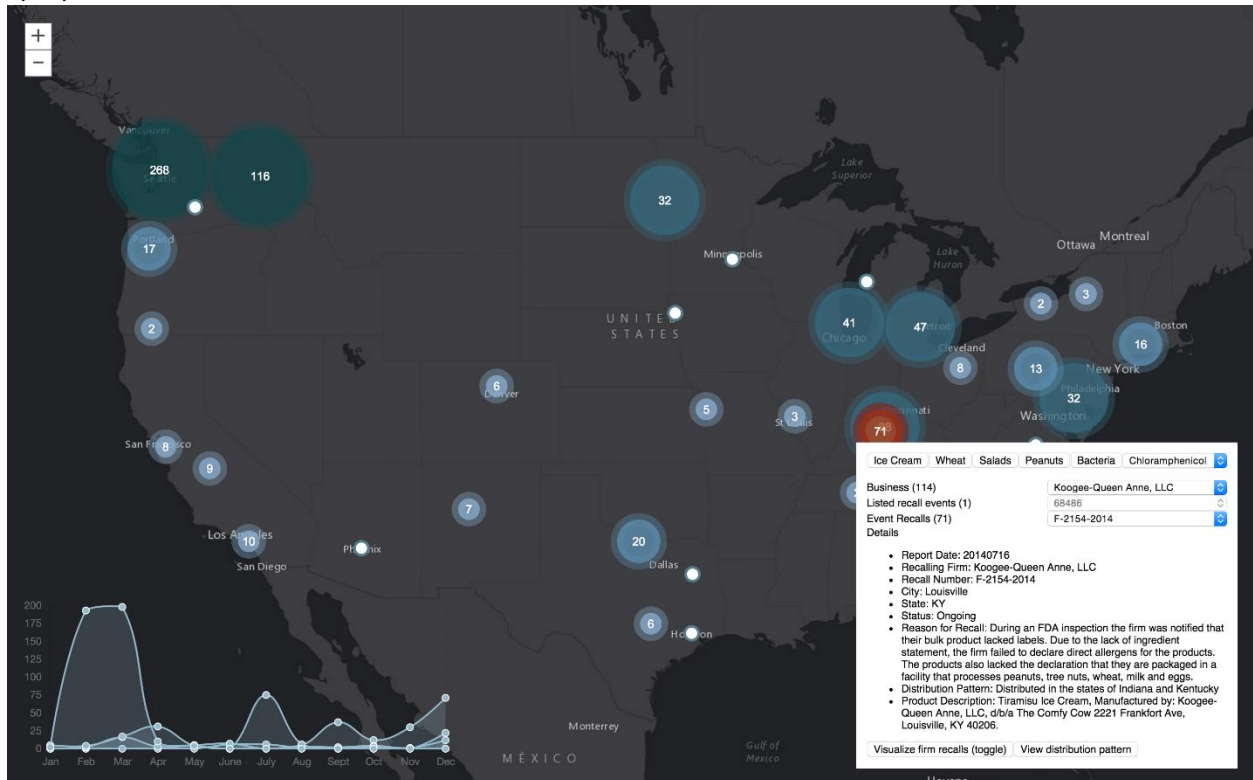
6/27/2015



6/29/2015

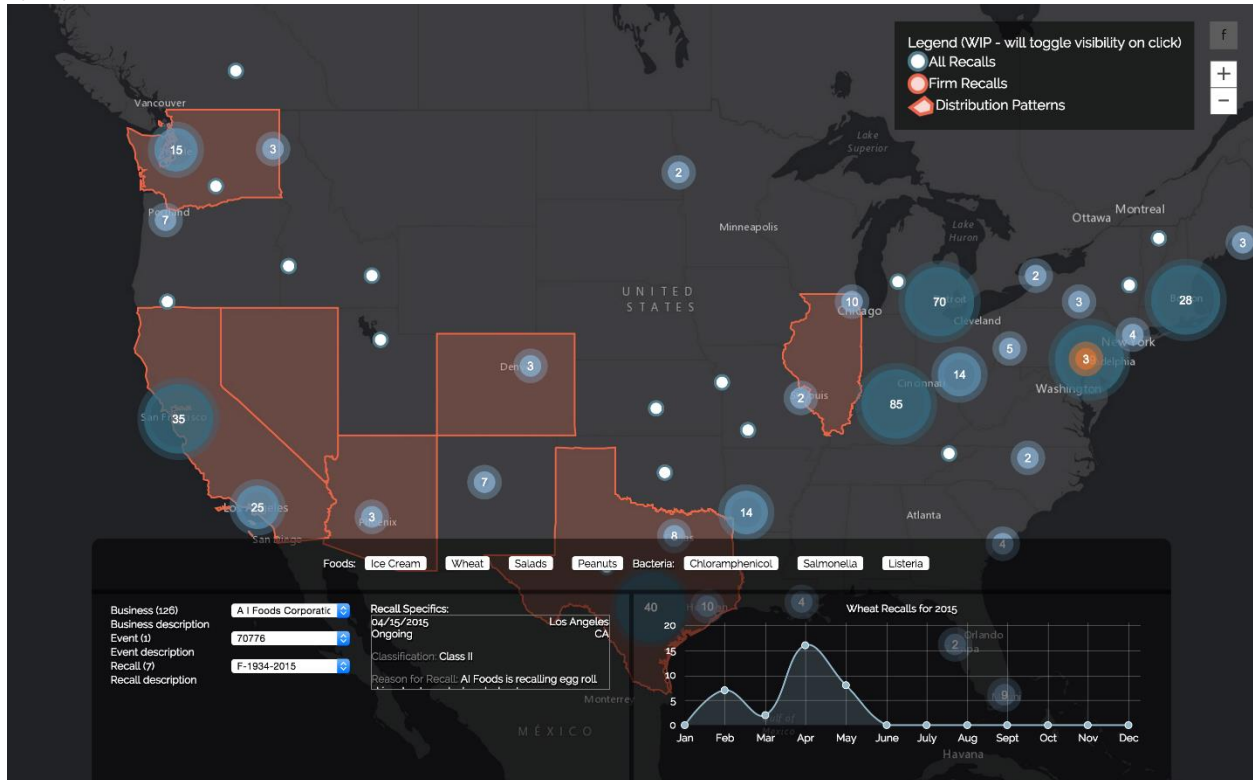


6/30/2015

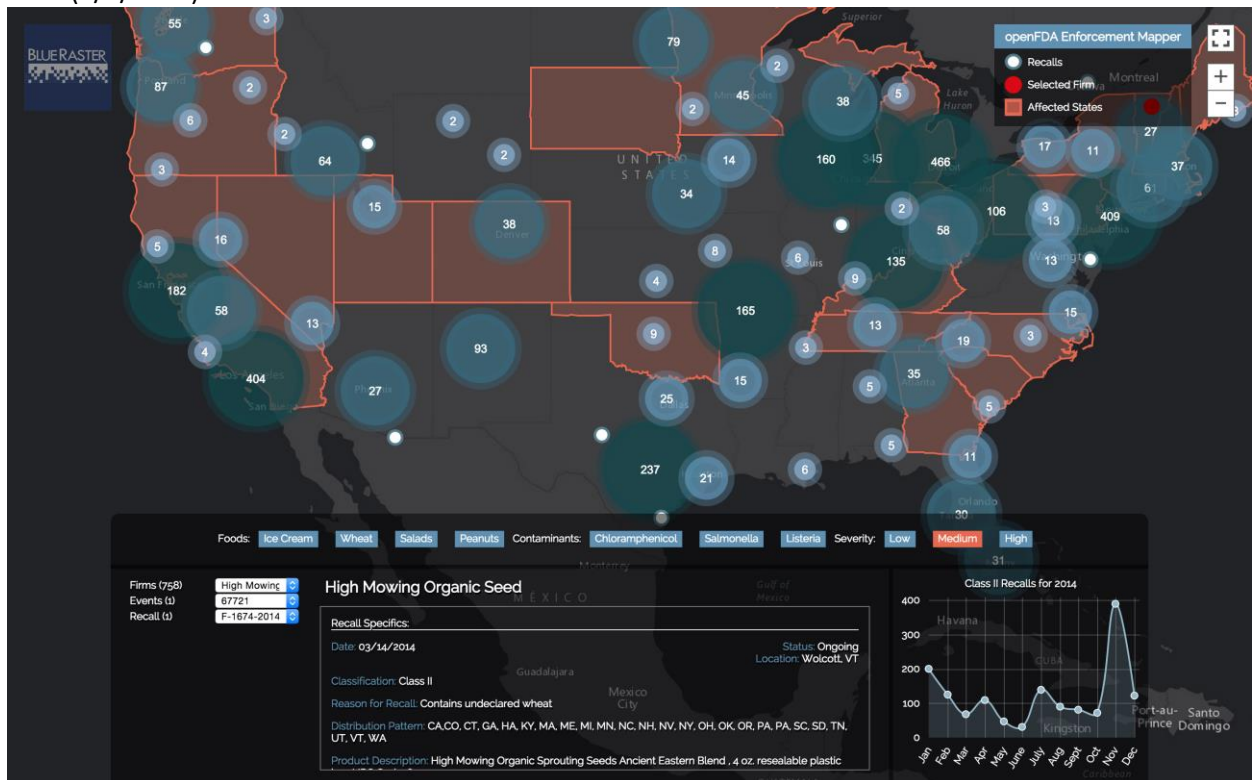




6/31/2015



Final (7/7/2015)



#### K) Documentation

Blue Raster provided sufficient documentation in the readme.md file and a docker container to install and run the prototype on another machine at <https://github.com/blueraster/open-fda-map-viewer>.

#### L) Openly Licensed

The prototype and underlying platforms used to create and run the prototype are openly licensed and free of charge.

To learn more about Blue Raster's work visit [www.blueraster.com/blog](http://www.blueraster.com/blog)