MICHIGAN STATE
UNIVERSITY

Cloud Security Event Processing and Alerting Platform

The Capstone Experience

Team Rook

Bradley Baker
Alex Fall
Jake Fenton
Brian Jones
Kaushik Sridasyam

Department of Computer Science and Engineering
Michigan State University

Fall 2017

From Students…
…to Professionals

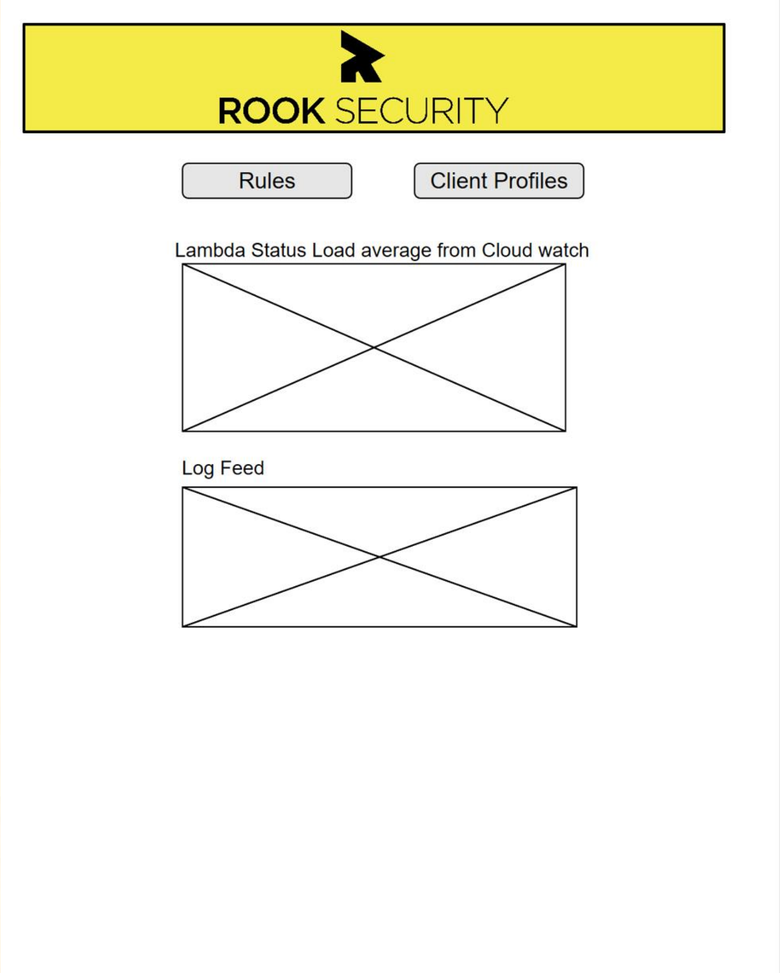
Functional Specifications

- Maintain features from current engines
- Manage and enable rules from web interface
- Manage addition of clients from web interface
  - Lambdas automatically handle new data
- Stretch: Allow testing of new rules

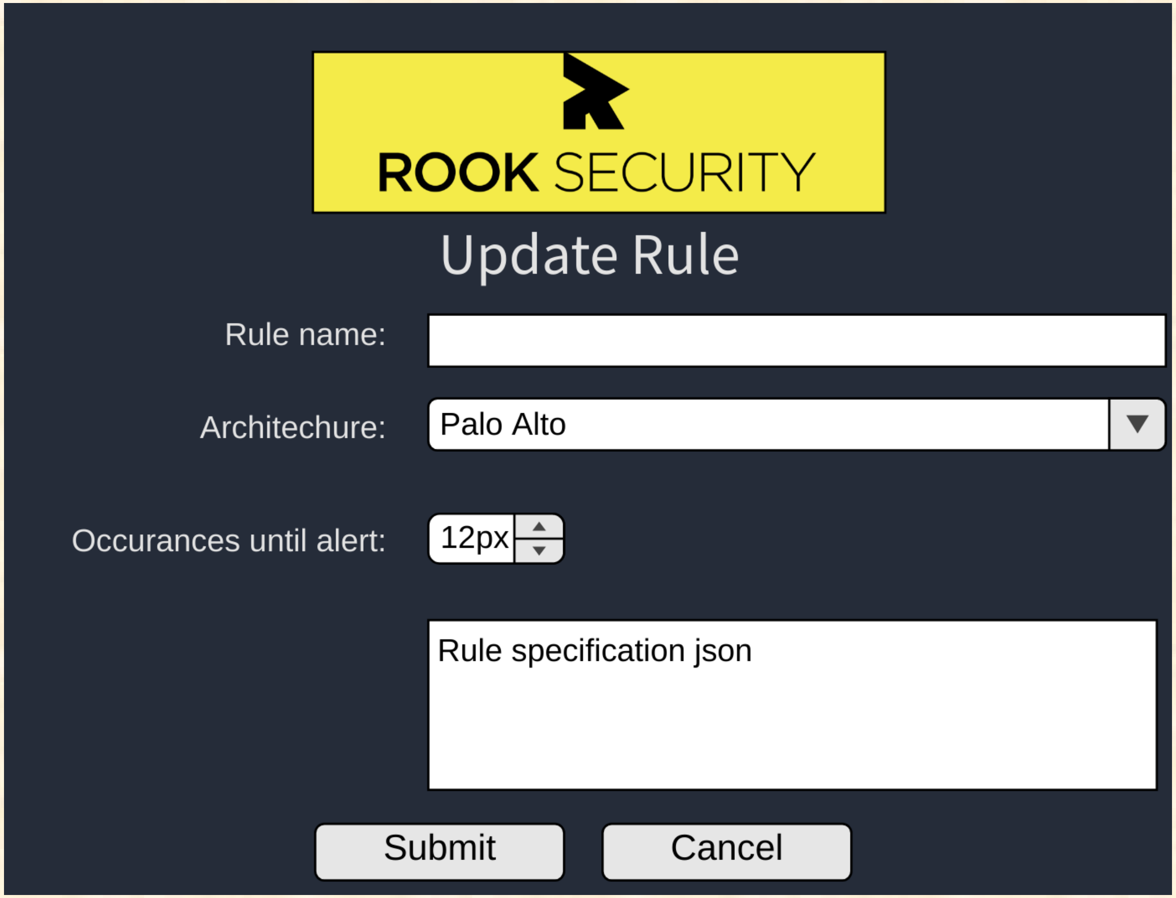
Design Specifications

- Security Analysts arrive at the Dashboard view for status
- Able to add and edit rules (CRUD)
- Can view client profiles
  - See all current rules for clients
- View performance of rule prospect.

Screen Mockup: Status/Landing Page



Screen Mockup: Update Rule



Update Rule

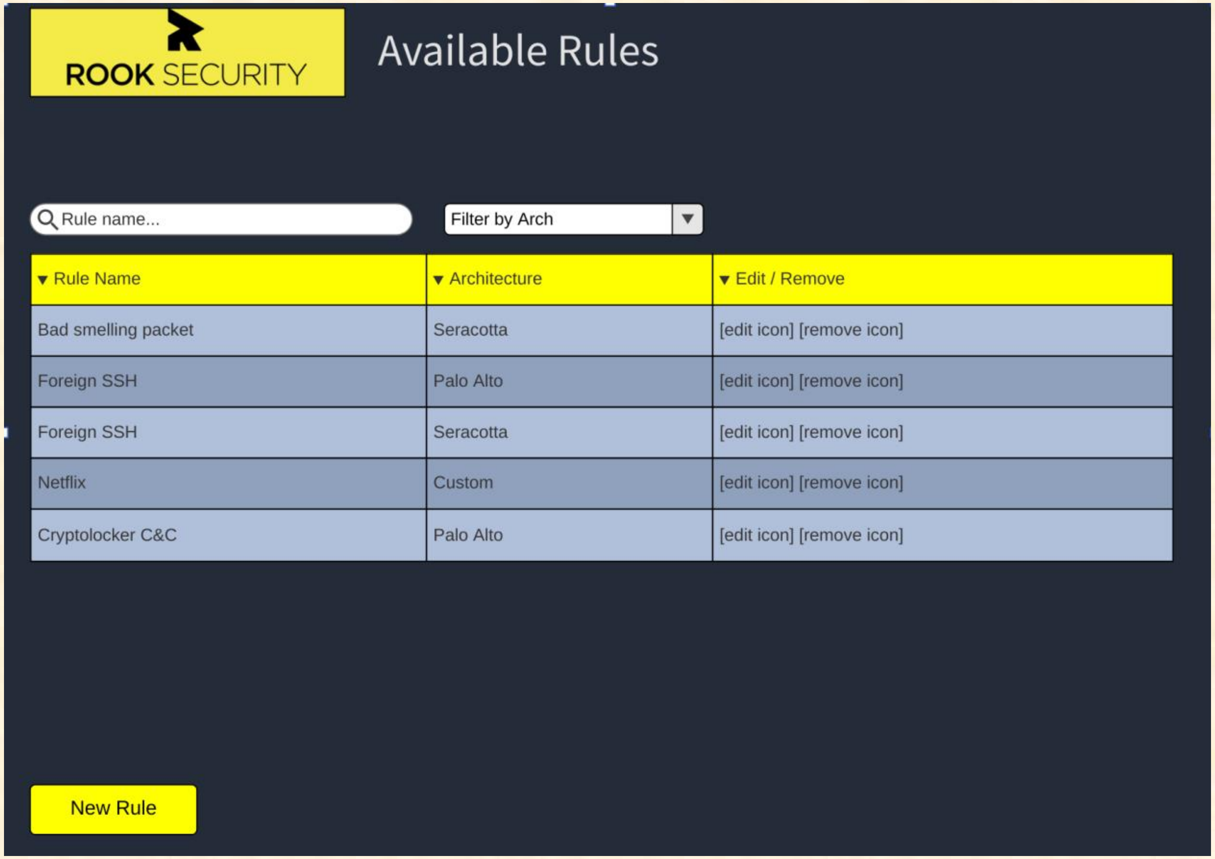
Rule name:

Architechure: Palo Alto

Occurances until alert: 12px

Rule specification json

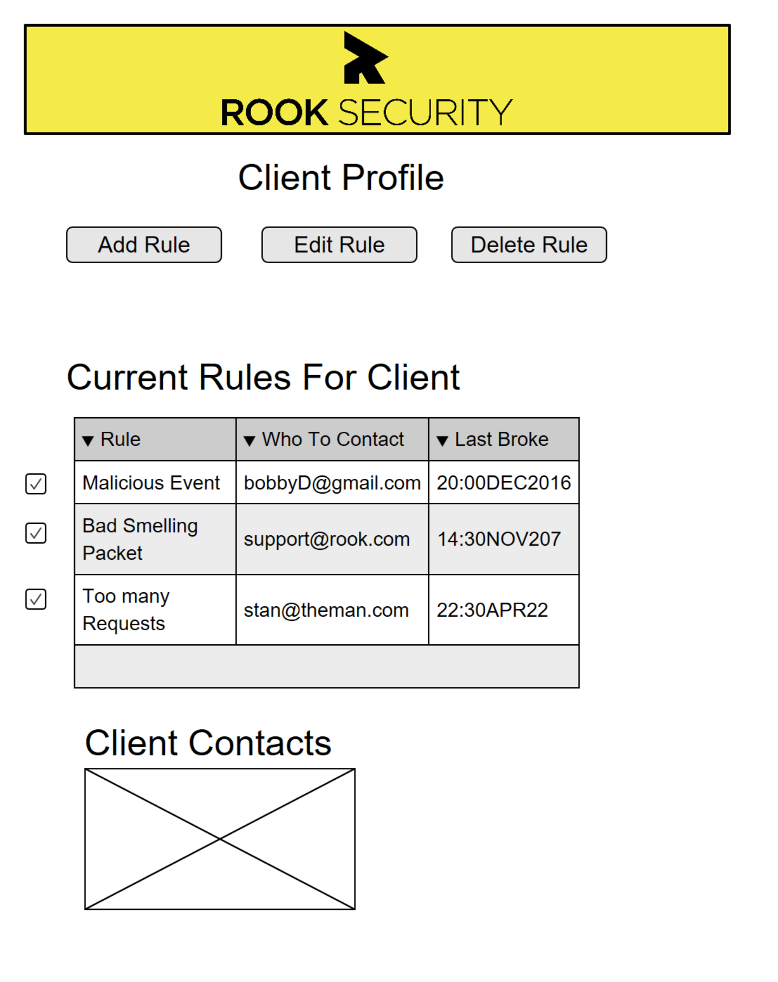
Screen Mockup: Rule List View



Available Rules

▼ Rule Name	▼ Architecture	▼ Edit / Remove
Bad smelling packet	Seracotta	[edit icon] [remove icon]
Foreign SSH	Palo Alto	[edit icon] [remove icon]
Foreign SSH	Seracotta	[edit icon] [remove icon]
Netflix	Custom	[edit icon] [remove icon]
Cryptolocker C&C	Palo Alto	[edit icon] [remove icon]

Screen Mockup: Client Profiles



Client Profile

Current Rules For Client

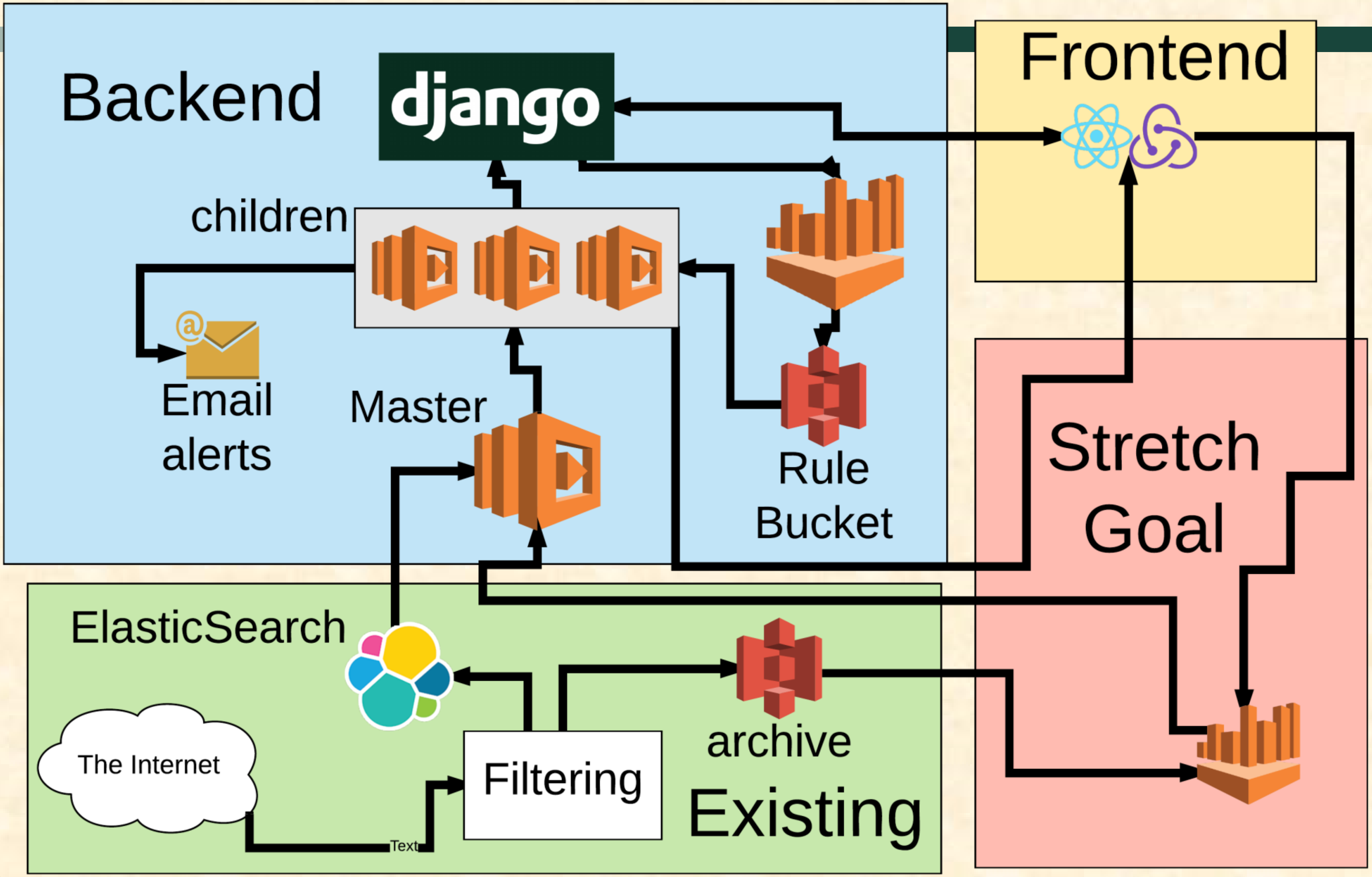
	▼ Rule	▼ Who To Contact	▼ Last Broke
☑	Malicious Event	bobbyD@gmail.com	20:00DEC2016
☑	Bad Smelling Packet	support@rook.com	14:30NOV207
☑	Too many Requests	stan@theman.com	22:30APR22

Client Contacts

Technical Specifications

- Master Lambda function
  - Pull new entries from ElasticSearch
  - Segregate by client and firewall architecture
  - Pass data to spawned child function
- Child Lambda function
  - Process client- and arch- specific events
  - Create tickets and/or send emails
- Python Django API
  - CRuD for the rules

System Architecture



Backend

django

children

Email alerts

Master

Rule Bucket

Frontend

Stretch Goal

ElasticSearch

The Internet

Text

Filtering

archive

Existing

System Components

- Hardware Platforms
  - AWS Lambda Functions
  - AWS EC2 Django Server
  - AWS EC2 Frontend Server
- Software Platforms / Technologies
  - Python Django
  - AWS Lambda Functions
  - ReactJS/Redux
  - SQLite

Testing

- Testing the Correlation Engine
  - Using sanitized archived data to simulate events
- Testing The Front End
  - Jest testing the React portion of our project
- Prototyping via Rook Personnel
- Django API
  - Postman

Risks

- Lambda Timing Out
  - Potential loss of vision on events
- Cannot currently reach Rook demo backend
  - Difficult to integrate UI when we can’t truly move through the state
- Django server needs access to S3 buckets
  - AWS credentials are sensitive
- Sending emails from engine
  - Not sure what our capabilities will be

Questions?