

# Analyzing Real time Activity Streams for Security.

---

Flink to the Rescue

Rashmi Singh  
Senior Software Engineer  
Qualys Inc.

# Qualys Platform Environment

## Security at scale on cloud

18+ products for comprehensive suite of security solutions

12,000+ customers

80+ deployments globally... on-prem, AWS, Azure, GCP



2+ trillion security events annually

3+ billion scans annually

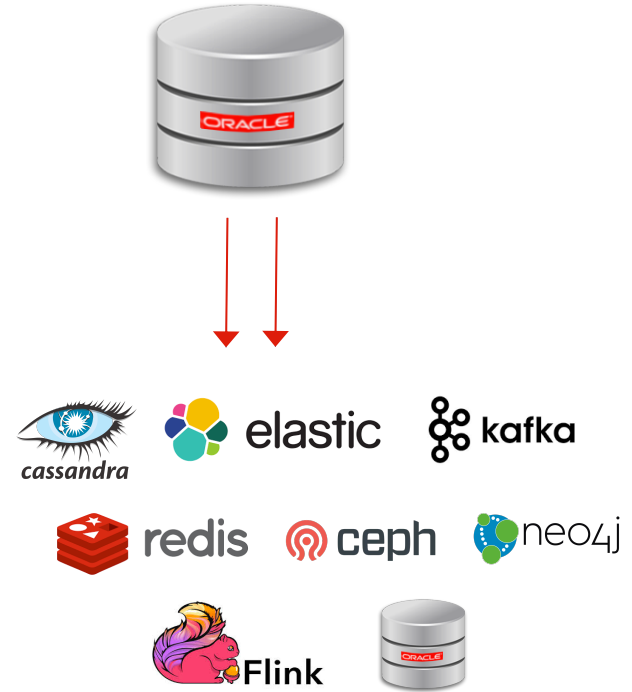
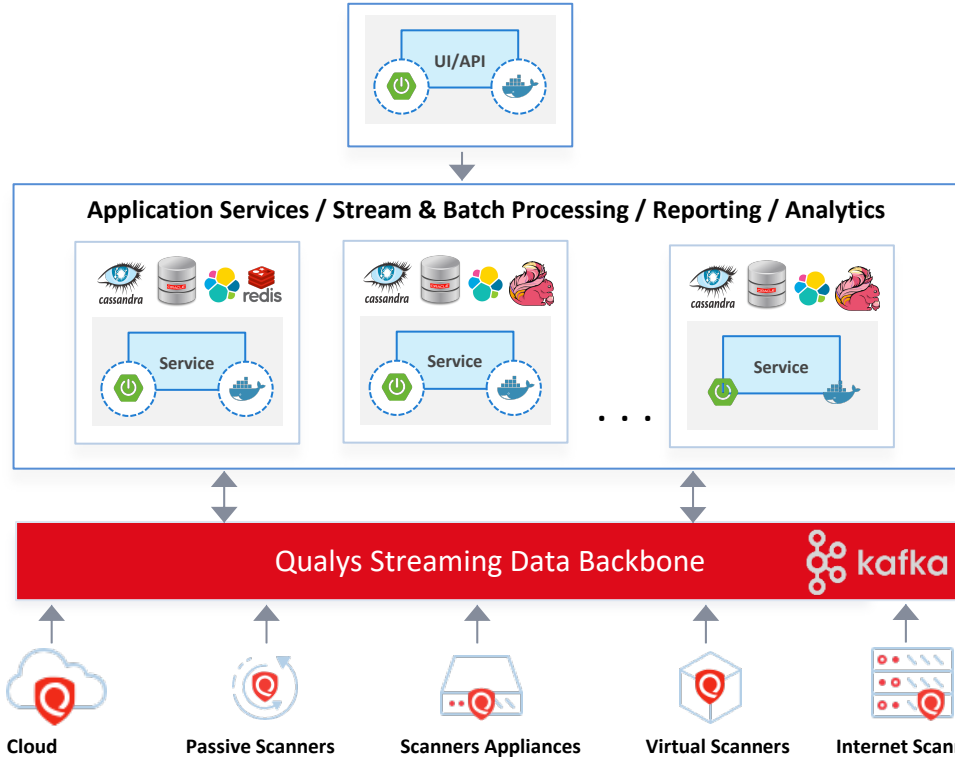
2.5+ billion messages daily across Kafka clusters

16+ PB storage and 16000 cores

**3+ trillion data points indexed in our Elasticsearch clusters**

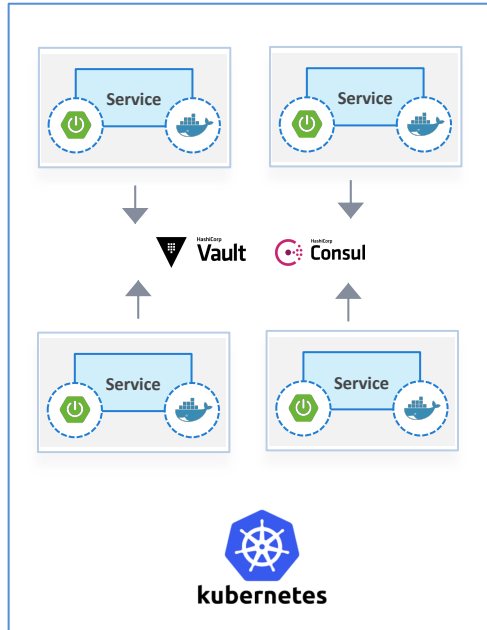
# Qualys Cloud Platform

Highly performant fully distributed microservice based big data stack

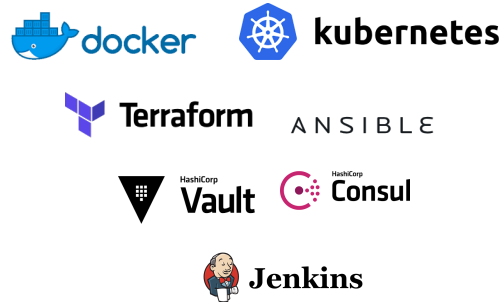


# Microservices & Cloud Native Architecture

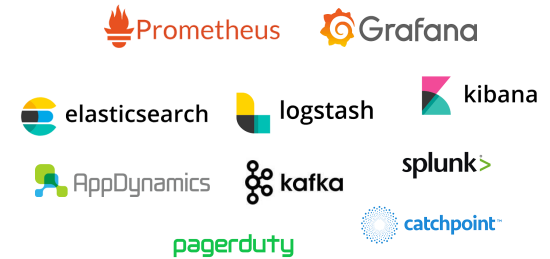
## Distributed Elastic Containers



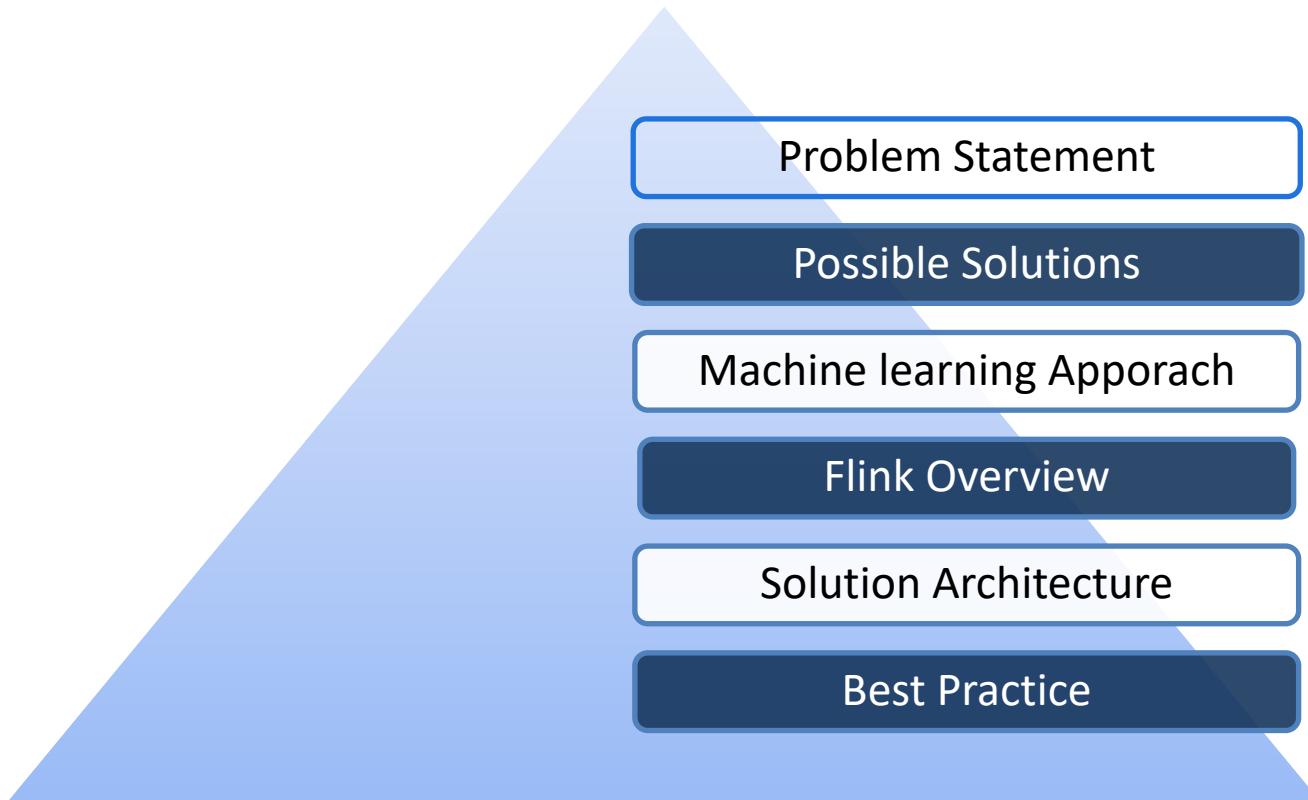
## Best DevOps tooling



## Centralized Monitoring and Analytics

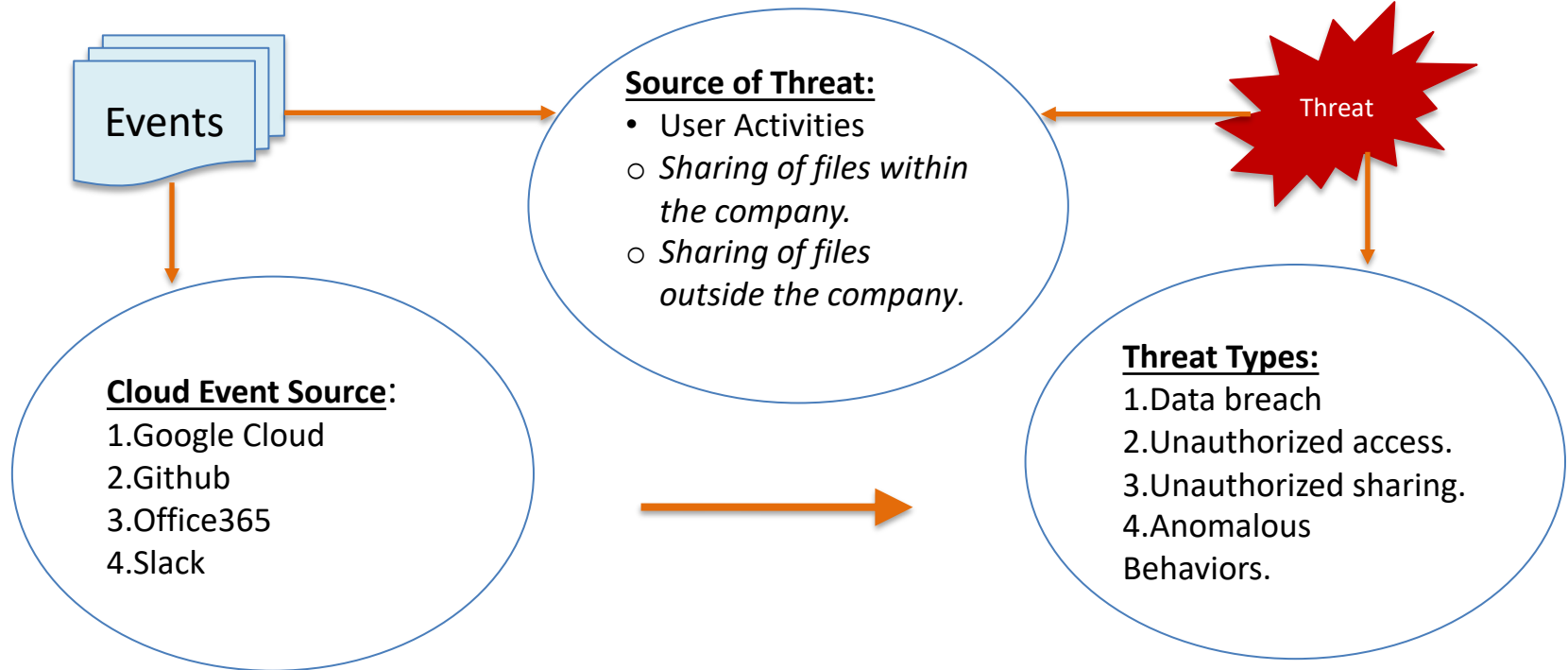


# Agenda



# Problem Statement :

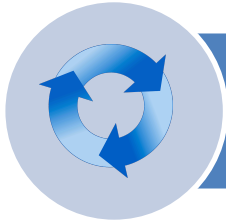
## Threat detection from User Activities on Cloud Applications



# Possible Solutions



User-defined rules



Identifying the repetitive event marked as a Threat previously.



Machine Learning Approach

# Machine learning Approach For Threat Detection

Data Capture, Pre-processing, Feature Extraction

Model Training

Threat Detection







# Flink: Overview

Distributed Stream Processing Engine.

Features:

- |                                 |       |                |
|---------------------------------|-------|----------------|
| 1. Real Time Stream Processing. | ————— | DataStream API |
| 2. Batch Processing.            | ————— | DataSet API    |
| 3. Machine Learning at Scale.   | ————— | Flink ML       |
| 4. Graph Analysis.              | ————— | Gelly          |

Special connectors like Flink Kafka etc



# Flink: Use case

## Event Driven Applications

- Threat Detection and Anomaly Detection

## Data Pipeline Applications

- Continuous search index building in e-commerce

## Data Analytics Applications

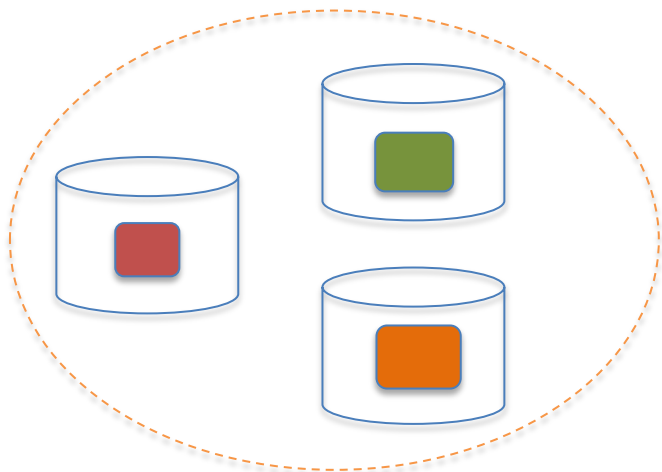
- Performance Metrics, Quality Monitoring



# Flink: State Management

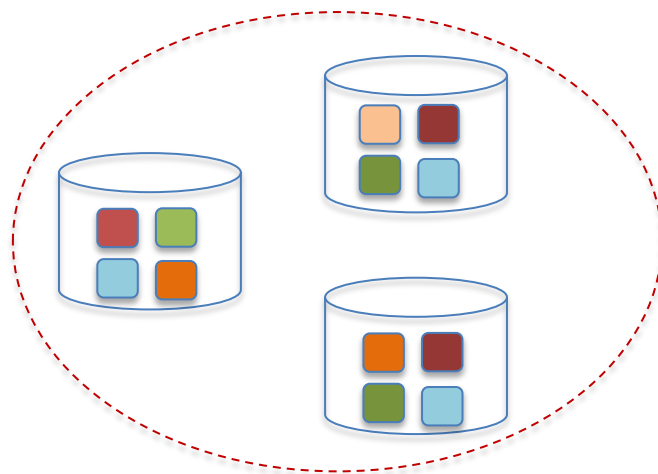
## Operator State

- State corresponding to a sub-task is recorded.
- Only List type is supported.



## Keyed State

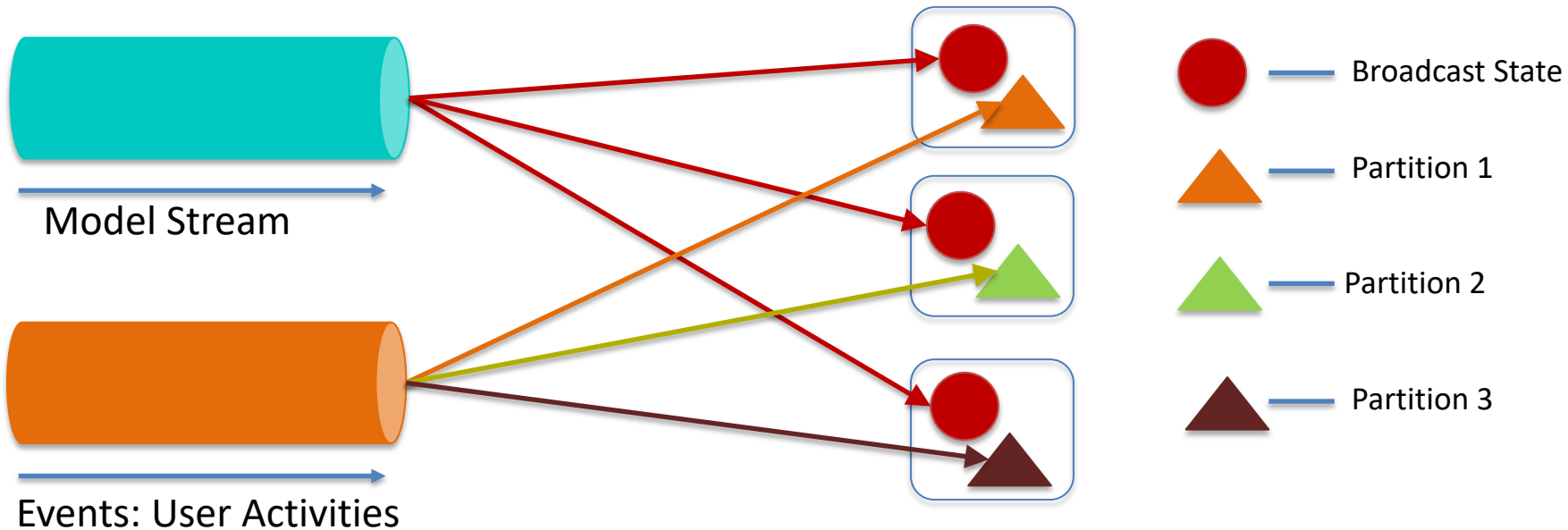
- State corresponding to each key is recorded
- Supports multiple data structure.



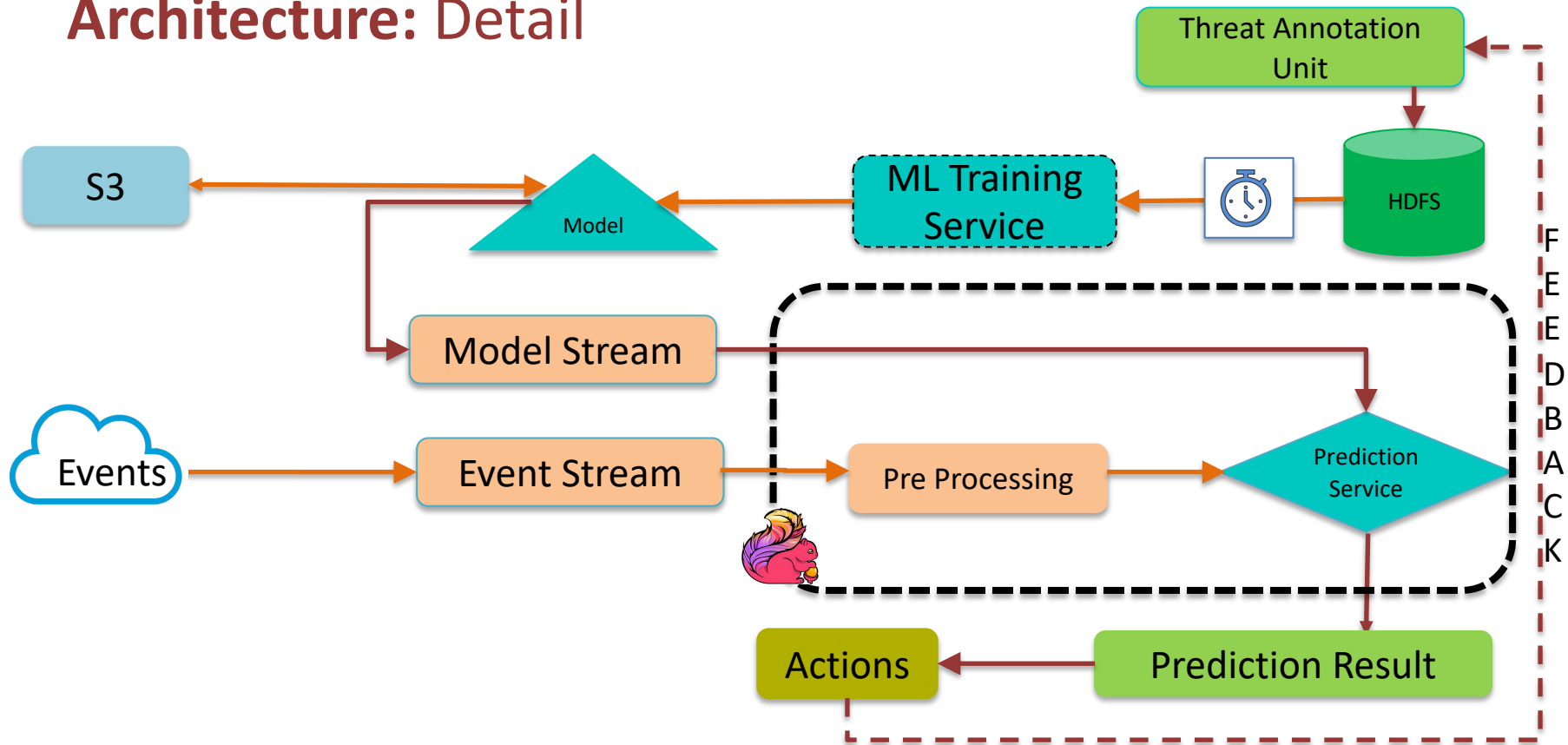


# Flink: Broadcast State

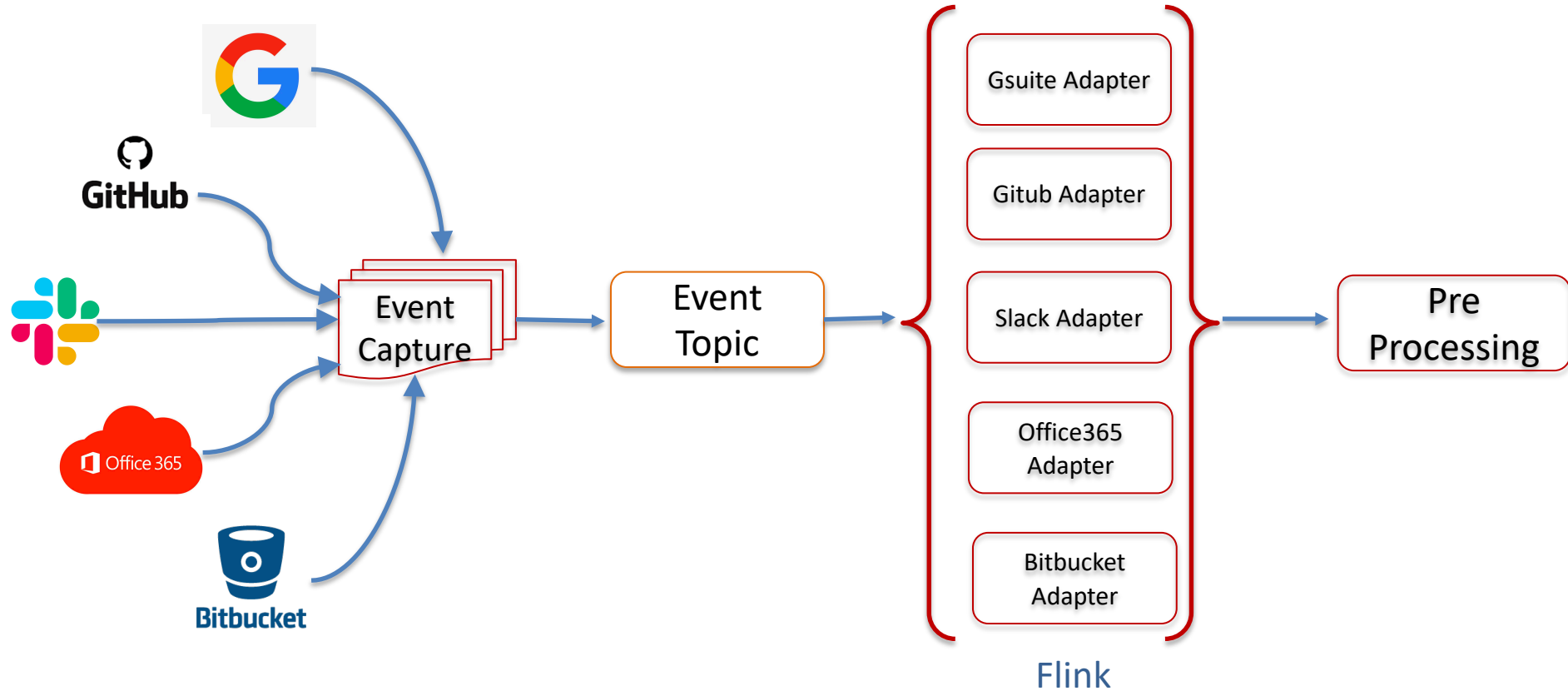
Operator tasks



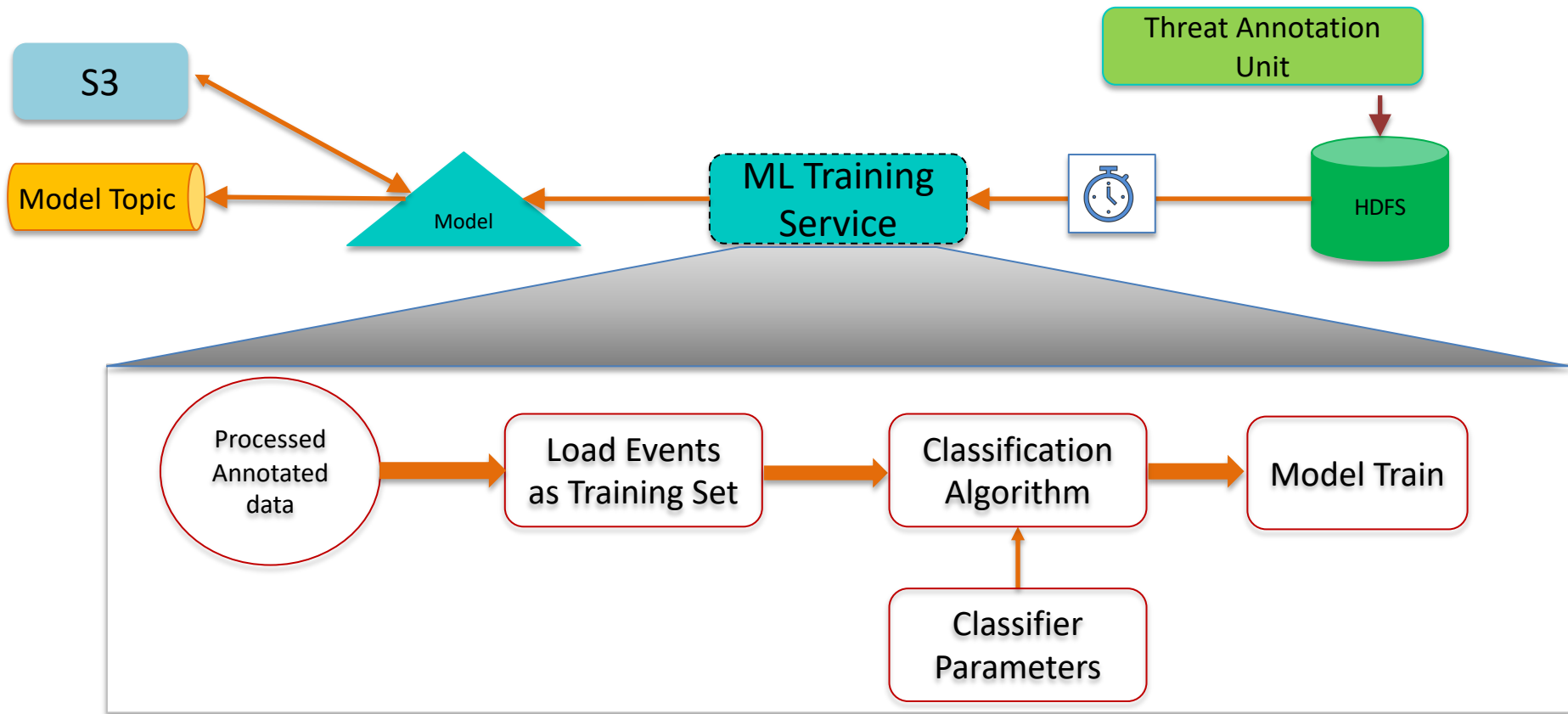
# Architecture: Detail



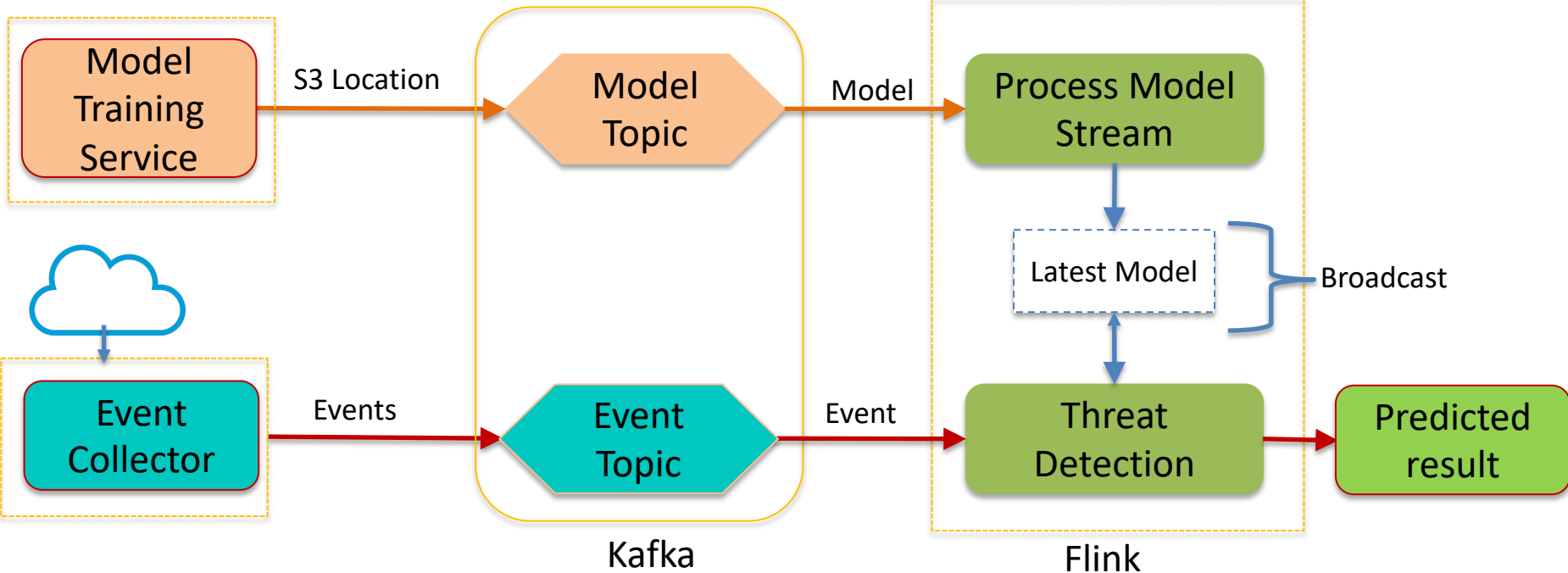
# Architecture: Event Capturing and Pre-Processing



# Architecture: Model Training





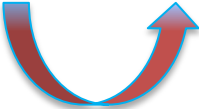
# Architecture: Load and Apply Model






# Threat Actions

1.   Send alert to Admin about the Event

2.  Revert back the Event

3.  Block the user who performed the event

# Best Practice

- Use Kafka as a Message Broker
- Instead of passing the whole model in Kafka topic, we can always pass the S3 location of the model
- Take checkpointing seriously

# References

- FLIP-23 Model Serving Architecture  
<https://cwiki.apache.org/confluence/display/FLINK/FLIP-23+-+Model+Serving>
- Pull request for FLIP-23 Implementation  
<https://github.com/apache/flink/pull/7446>
- Special thanks to **Boris Lublinsky**



**Qualys**<sup>®</sup>

Continuous Security

**Thank You**

Email- [singhrashmi579@gmail](mailto:singhrashmi579@gmail.com)

Twitter- [@Rashmi100Singh](https://twitter.com/Rashmi100Singh)