RSA CONFERENCE 2014
ASIA PACIFIC & JAPAN

Share.
Learn.
Secure.
Capitalizing on
Collective Intelligence

Combat Sophisticated Threats

How Big Data and OpenSOC Could Help

SESSION ID: SEC-T11

James Sirota

Big Data Architect/Data Scientist
Cisco Security Solutions Practice
@JamesSirota

In the next few minutes…

- **Introduction to Data-Driven Security**
- **Overview of OpenSOC**
- **Overview of the Analytics Pipeline**
- **Survey of Algorithms used by OpenSOC**
- **Probabilistic Structures for Stream Estimation**
- **Q&A**

Introduction to Data-Driven Security

Traditional Approach to Security

- Reactive Security Model [1,2,3,4,5]
  - 80% of spending on perimeter defenses
  - Average 40-50 security tools per organization
  - Collectively generate ~20,000 events per second
  - Tools are highly specialized
  - 80% of alerts require additional follow-up

Traditional Approach to Security

- Investigation and Incident Response[5,6]
  - 66% take an hour or longer to identify
  - 91% take a day or longer to discover root cause
  - 81% take a day or longer to fix
  - 84% of fixes take a week or longer to validate

Where things break down…

- Challenges with perimeter defense [2]
  - Most organizations support BYOD
  - 47% of organizations use cloud services
  - Global Workplace
- Challenges with Point Tools [4, 5]
  - Too many tools and manual processes
  - Too many false positive alerts
  - Lack of integrated architecture
  - Not enough data sources

Where things break down…

- Challenges with Adding Data Sources
  - Volume: from Terabytes to Zettabytes
  - Variety: structured to unstructured
  - Velocity: everything is a sensor, generates logs
  - Veracity: data quality issues
  - Value: What to keep? Purge? Summarize?

- Skills Challenge[4,6,7]
  - Staff spend ~40% less time on tasks than ideal
  - 35% of organizations use contractors

Cisco Approach to Security

- Data-Driven Security Model
  - Moving from reactive to predictive
  - Moving from point tools to a unified platform
  - Shifting skills from specialists to data scientists
  - Emphasis on quality and not quantity of alerts
  - Emphasis on closed-loop analytics

Overview of OpenSOC

Introducing OpenSOC

- Supports Data-Driven Security Model
  - Unified platform for ingest, storage, analytics
  - Provides multiple views/access patterns for data
  - Interactive Analytics and Predictive Modeling
  - Provides contextual real-time alerts
  - Rapid deployment and scoring

Emergence of Big Data

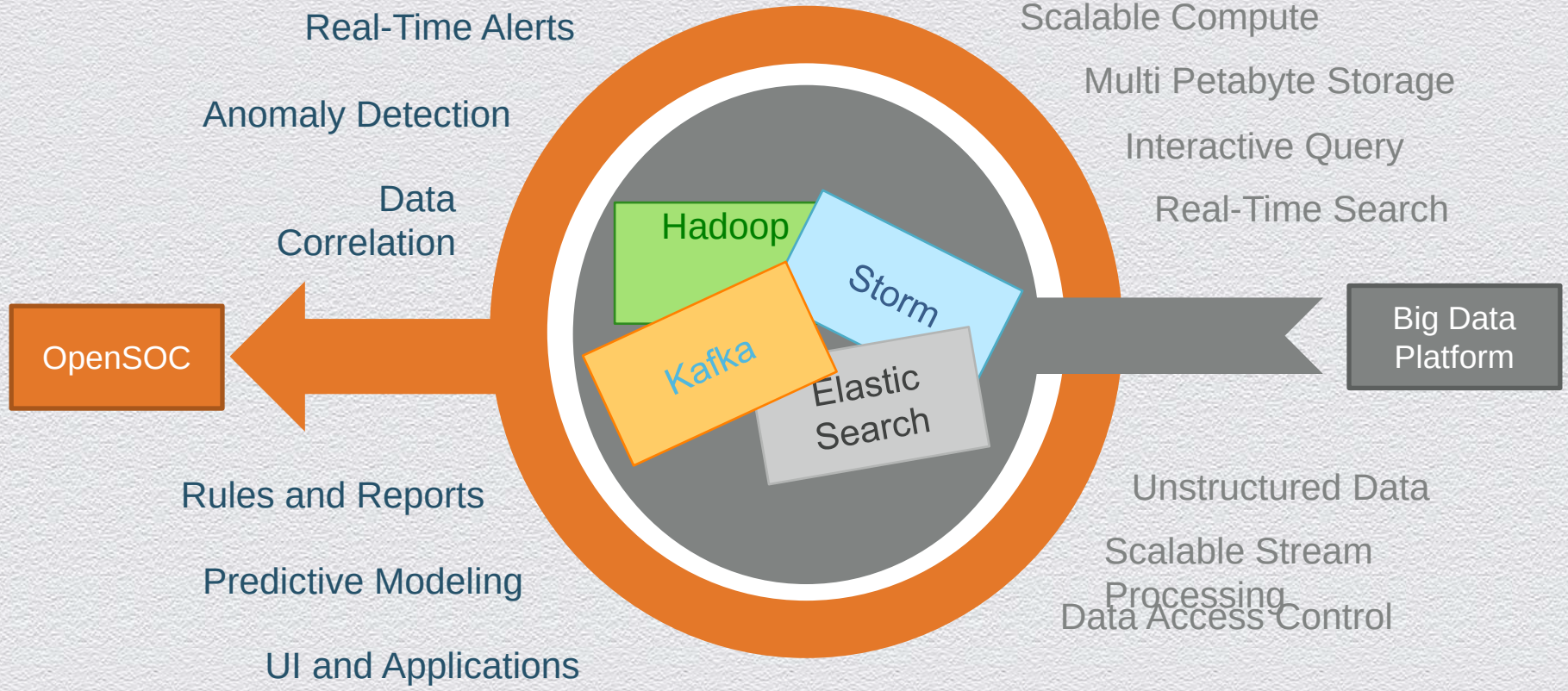
- Systems that scale out, not up
- Parallel and scalable computation tools
- Cheap, massively-scalable storage
- Stream computation + stream analysis
- Scaling + approximation

Technology Behind OpenSOC

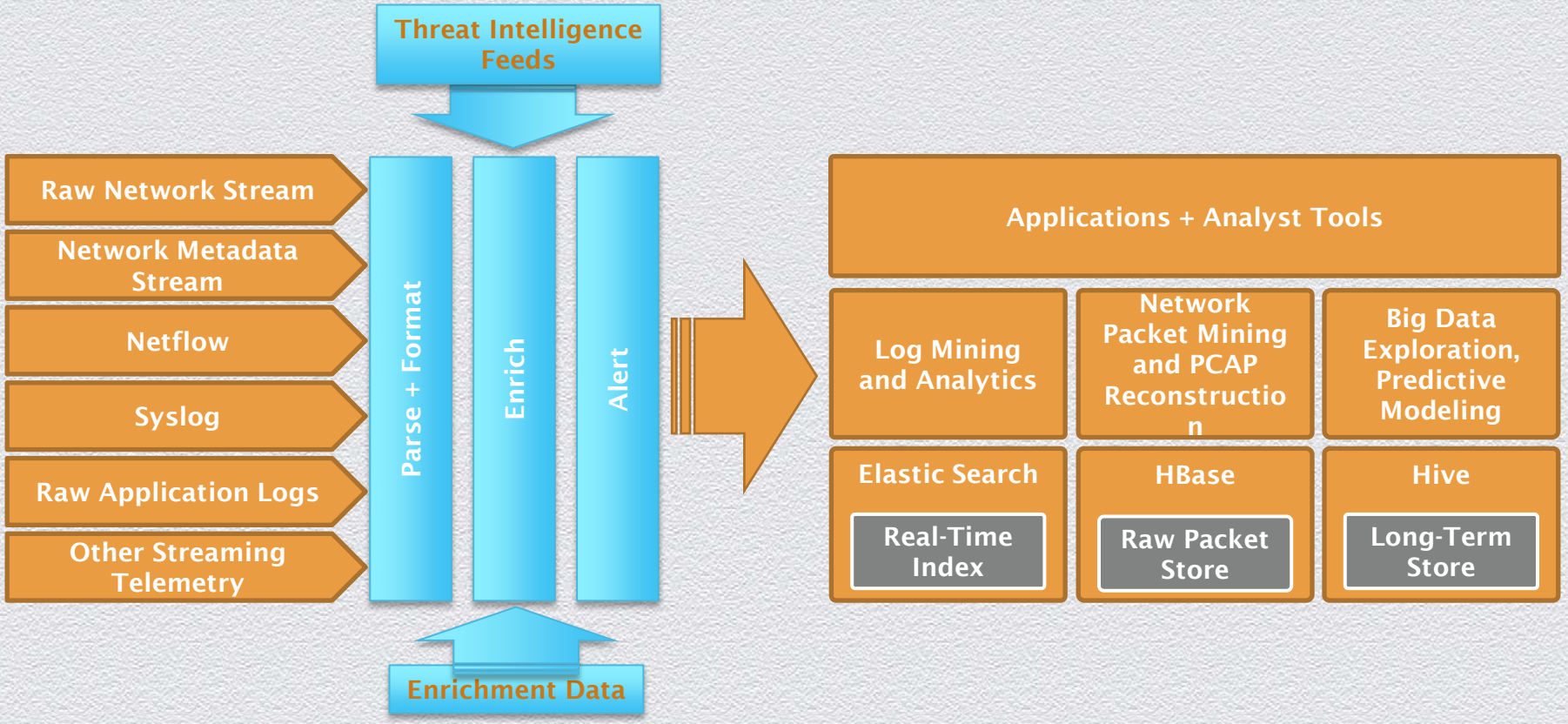
- Telemetry Capture Layer: **Apache Flume**
- Data Bus: **Apache Kafka**
- Stream Processor: **Apache Storm**
- Real-Time Index and Search: **Elastic Search**
- Long-Term Data Store: **Apache Hive**
- Long-Term Packet Store: **Apache Hbase**
- Visualization Platform: **Kibana**

Introducing OpenSOC

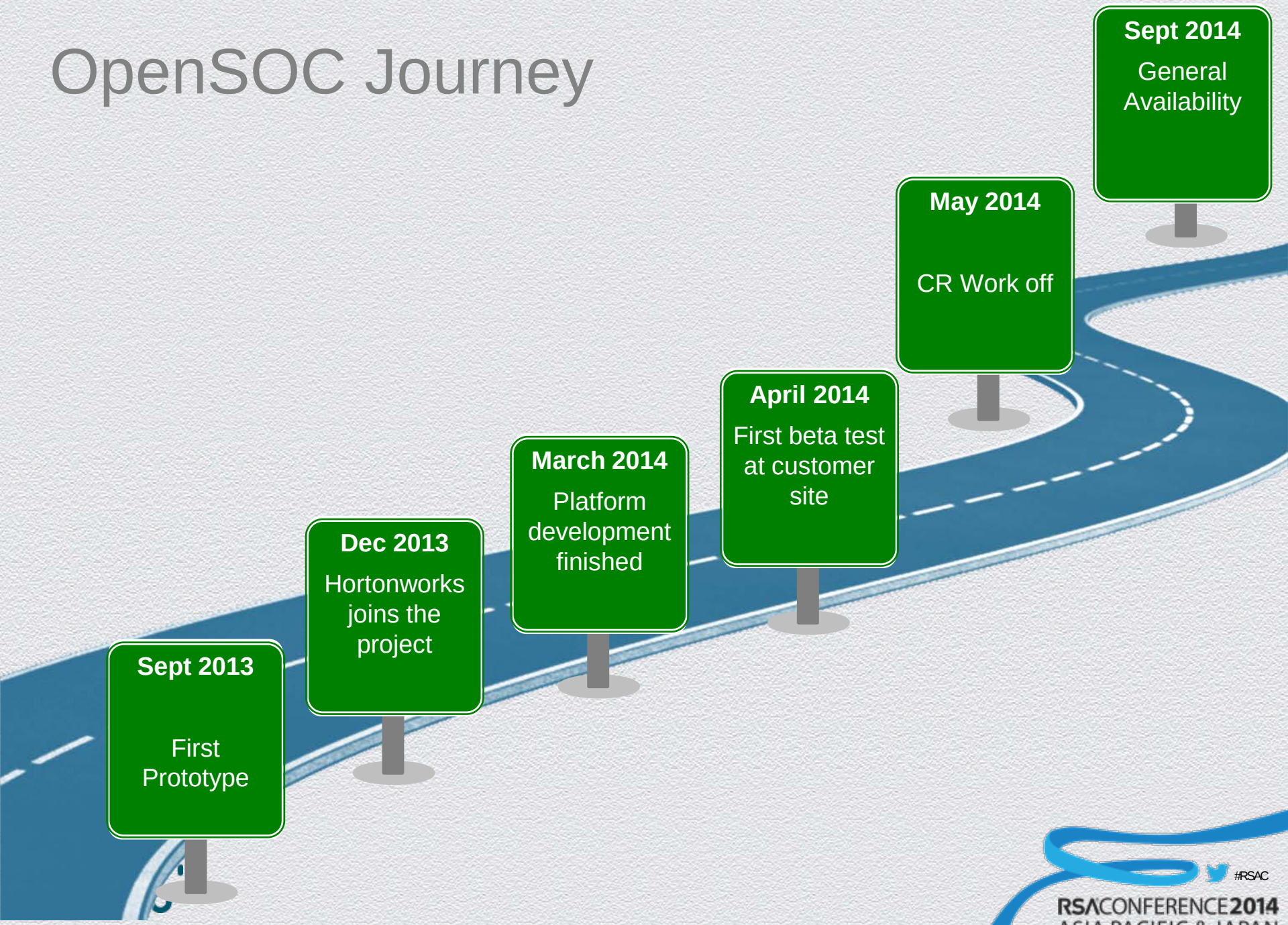
Intersection of Big Data and Security Analytics



OpenSOC in a Nutshell

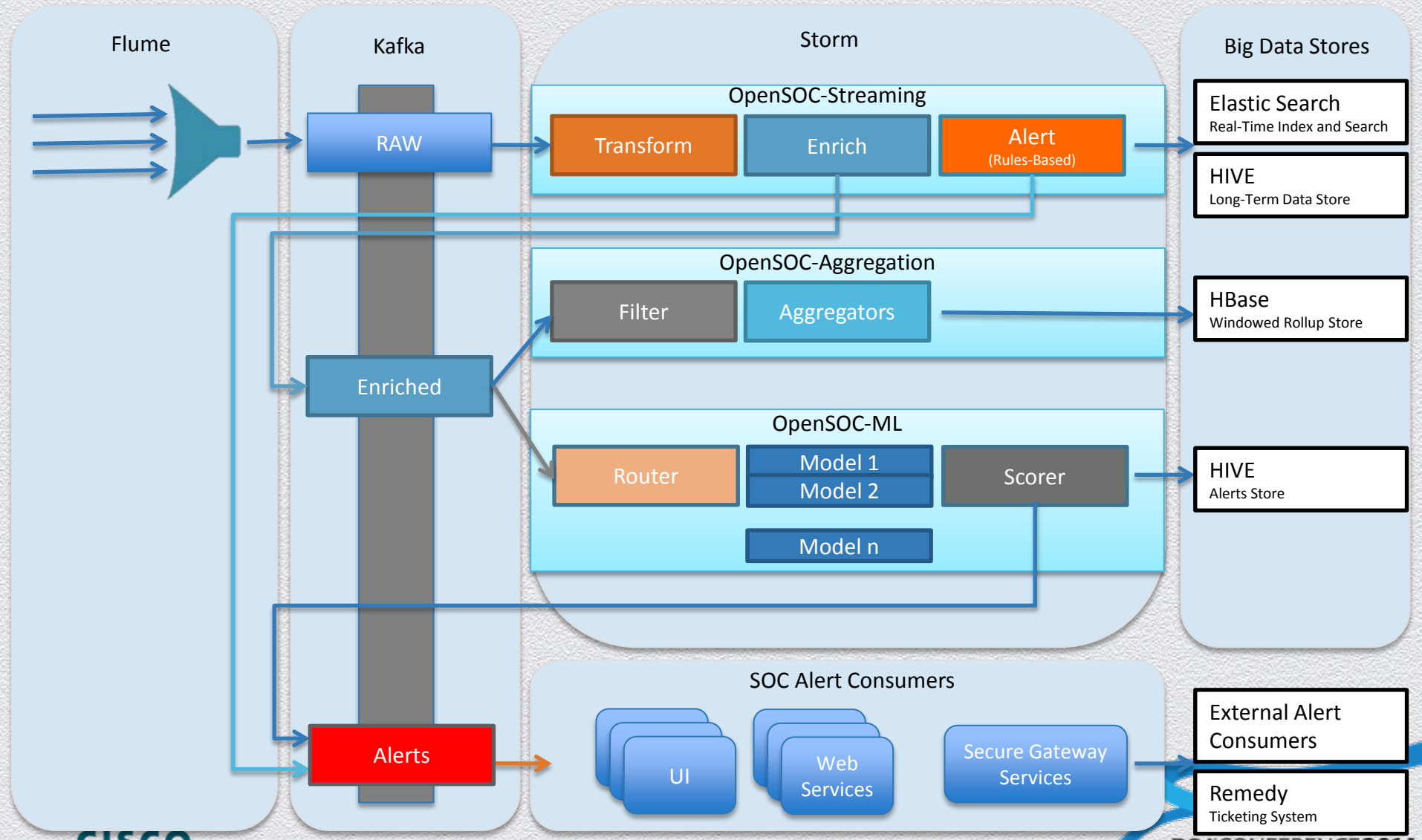


OpenSOC Journey

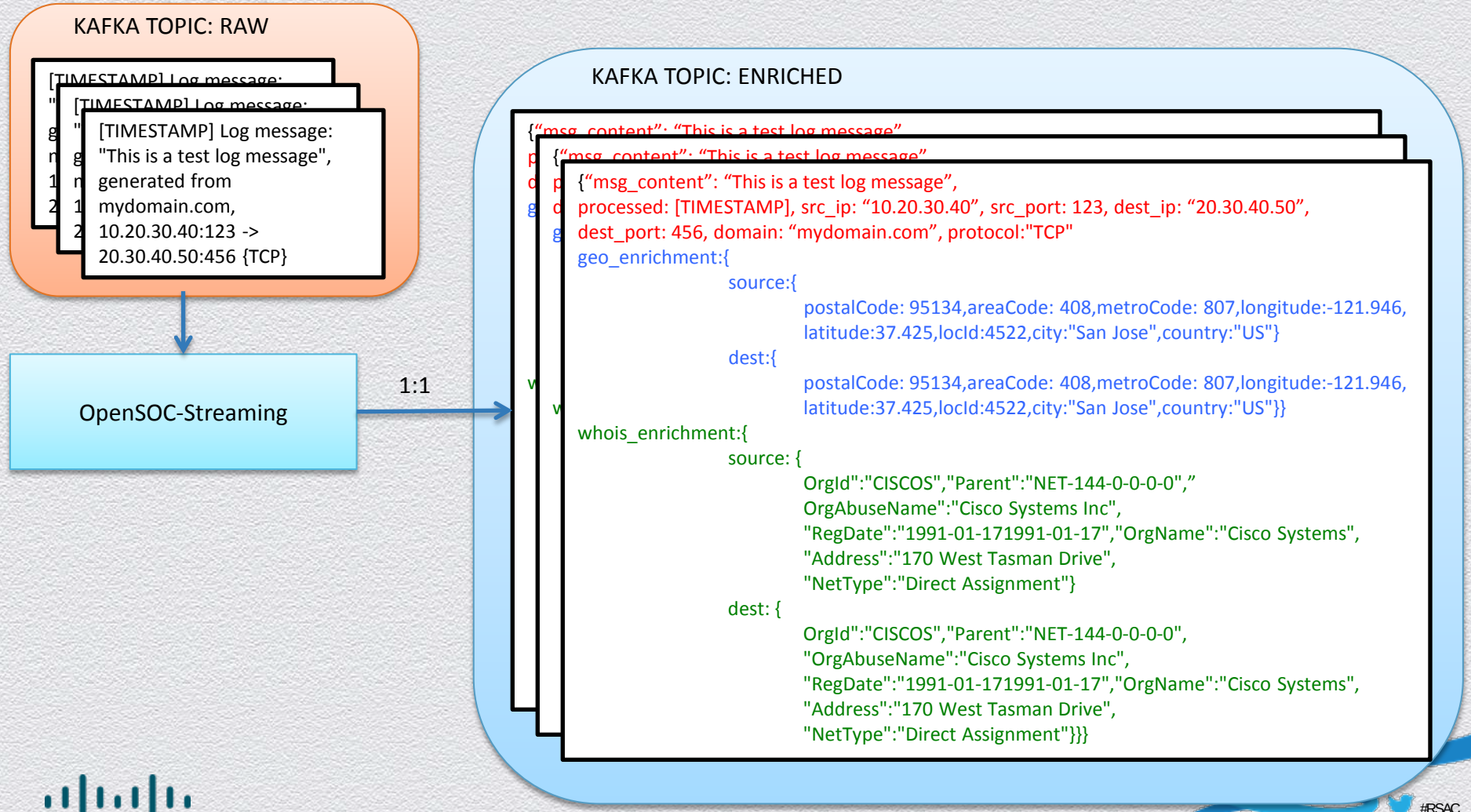


Overview of the Analytics Pipeline

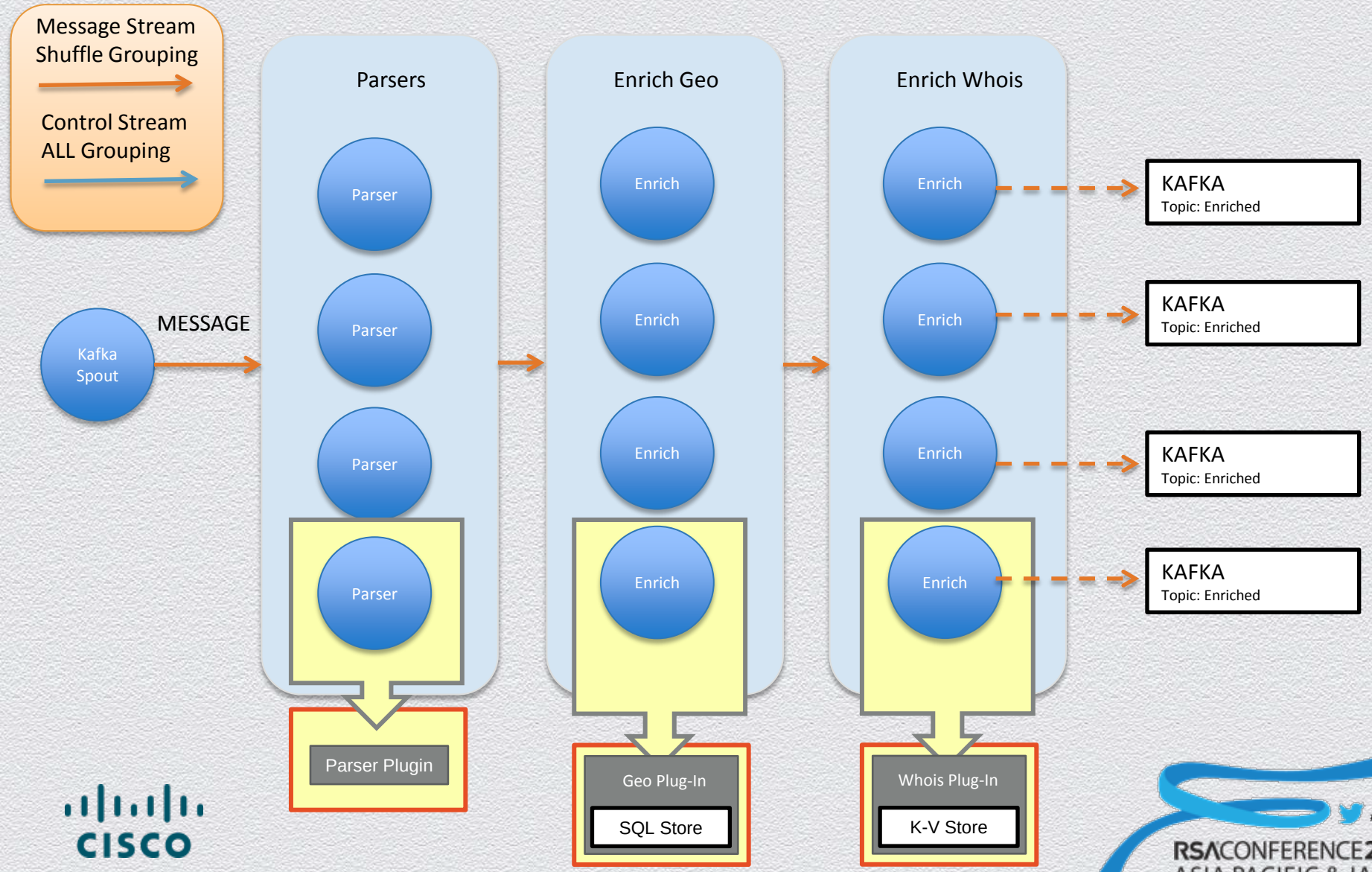
Analytics Pipeline



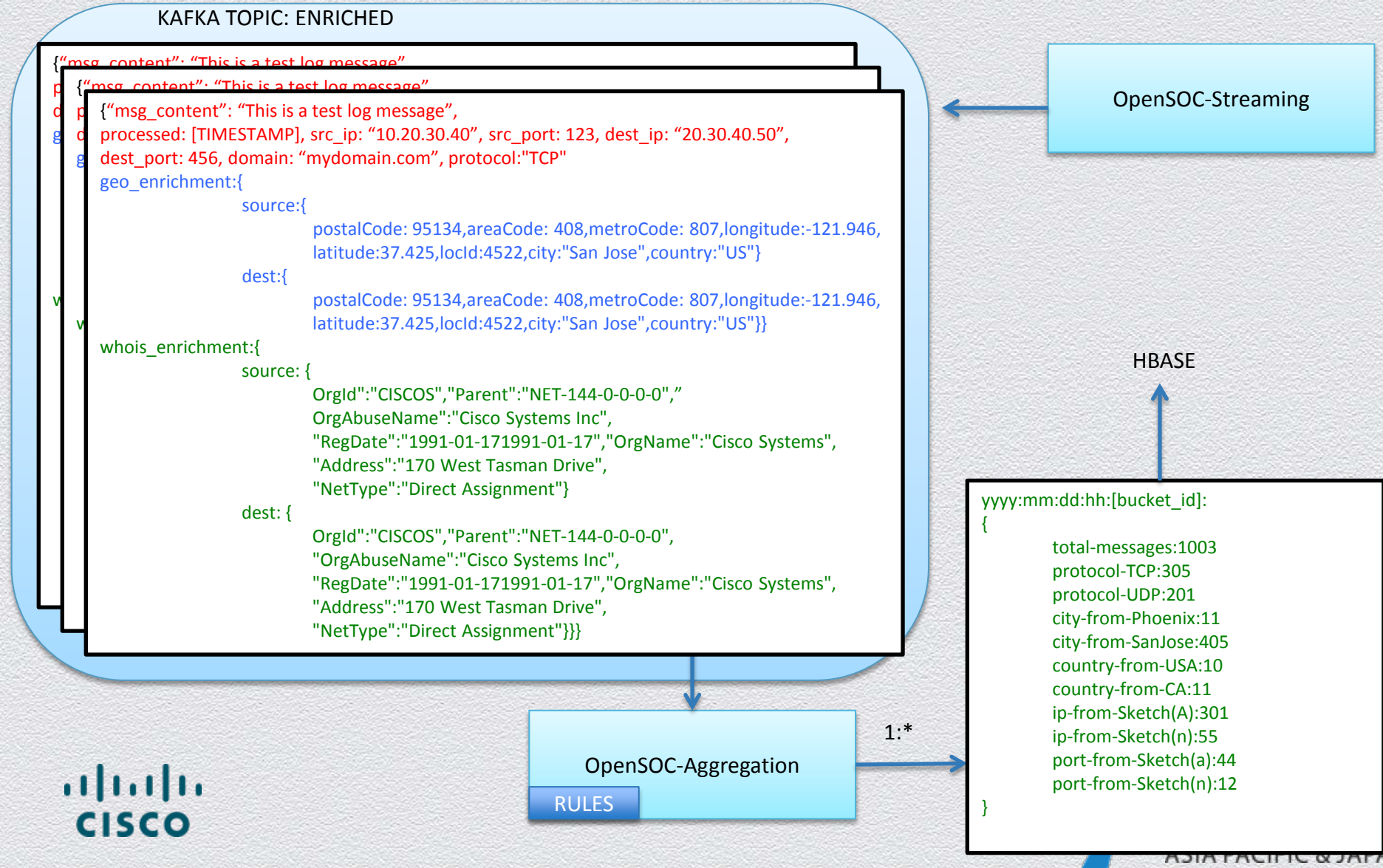
OpenSOC-Streaming



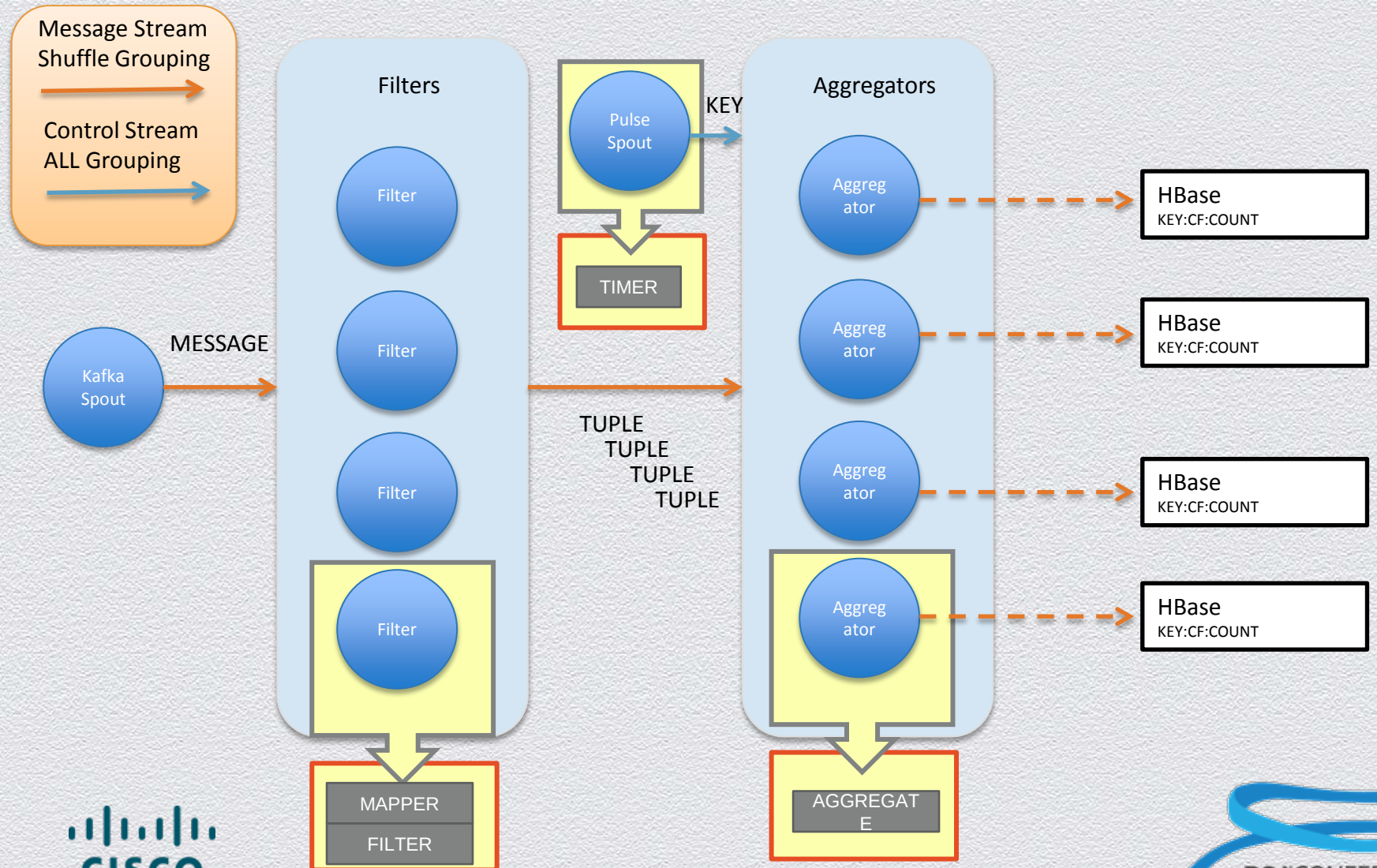
Streaming Topology



OpenSOC-Aggregation



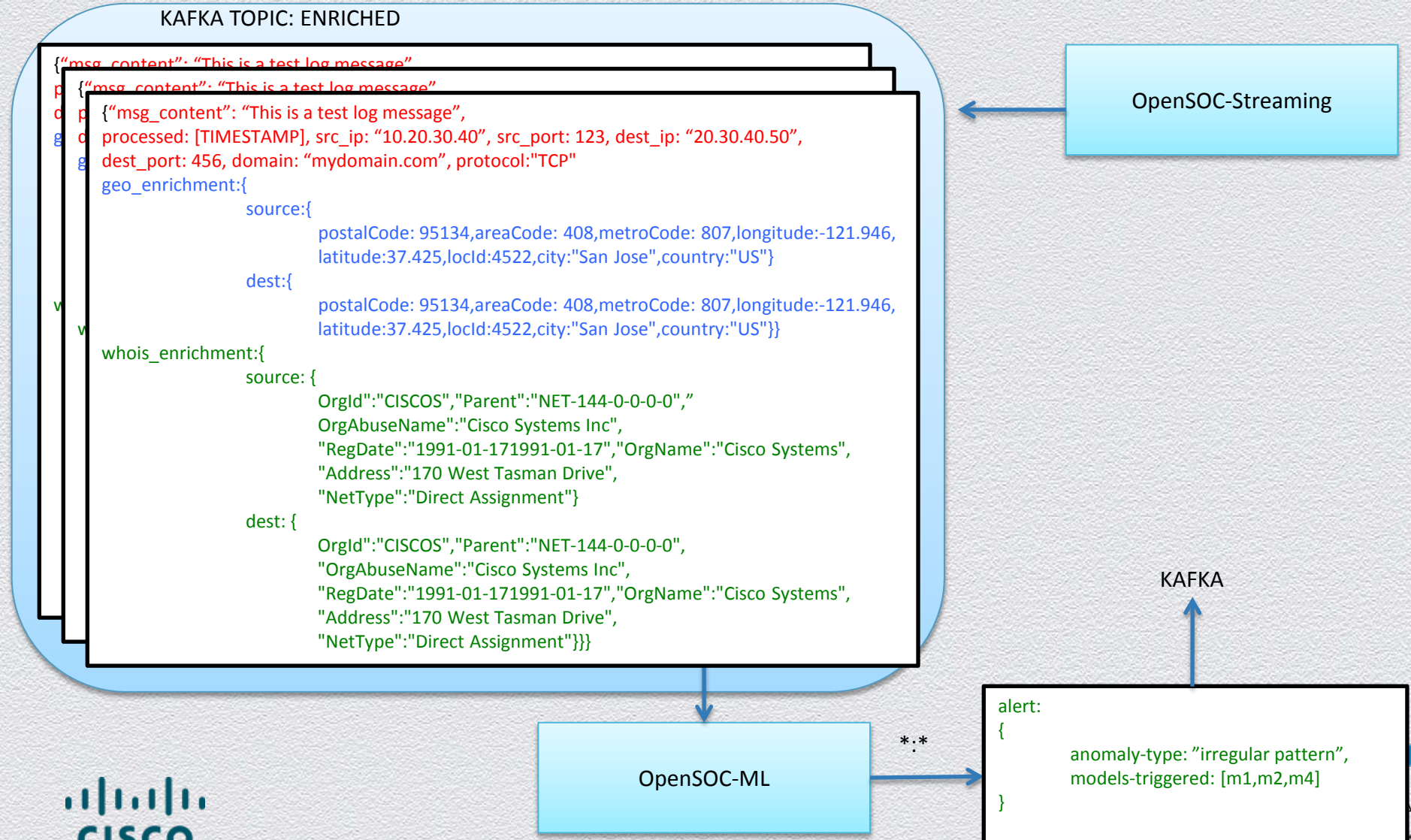
Aggregation Topology



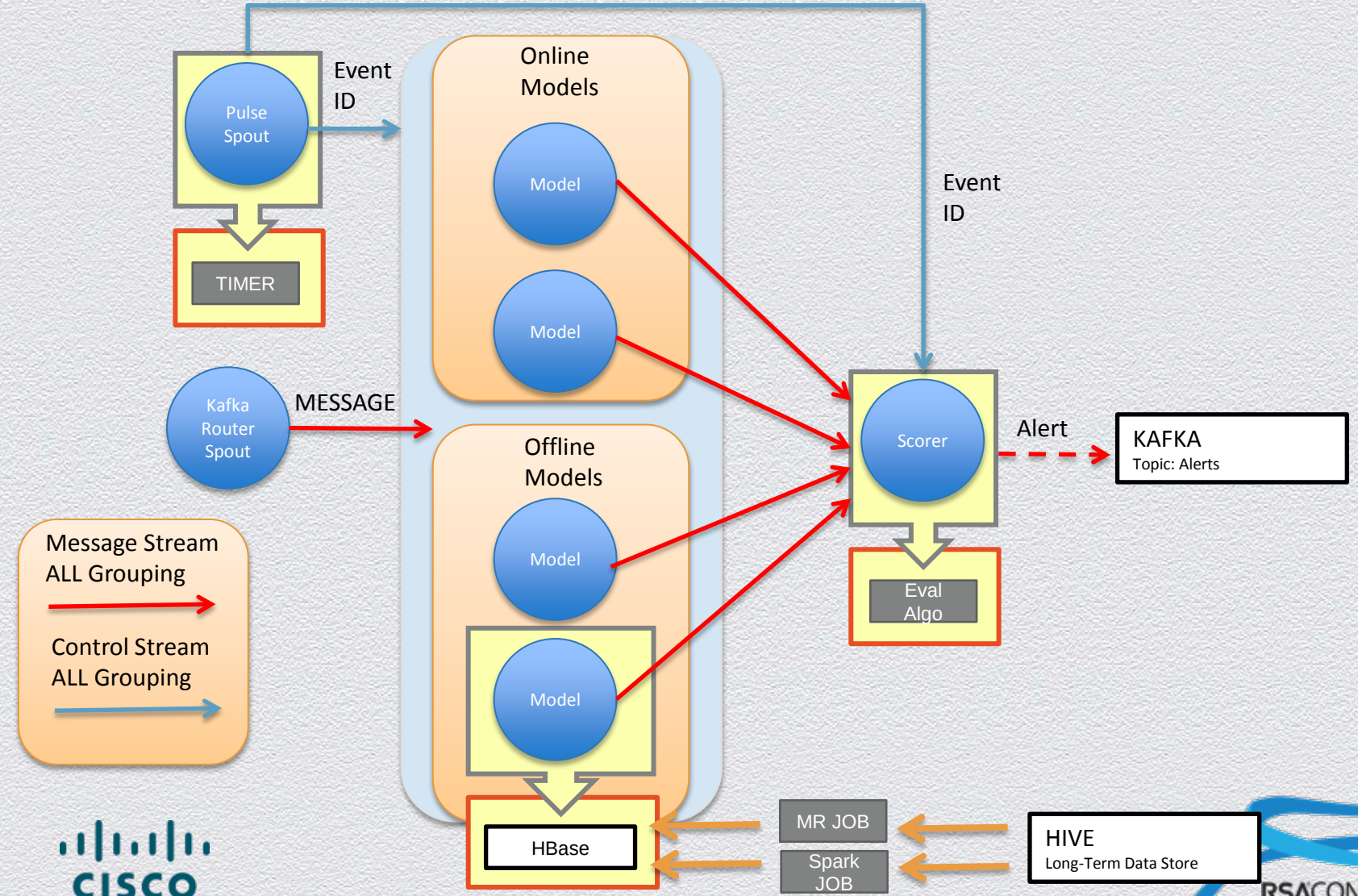
HBase Aggregation Table

Key	Total-Messages	Feature 1	Feature 2	Feature 3	Feature 4	Feature 5	…	Feature n
2004-01-12-15-1	34	5	0	0	14	4		22
2004-01-12-15-2	30	6	2	0	11	5		24
2004-01-12-15-3	33	4	1	1	12	2		20
2004-01-12-15-4	31	5	0	0	14	4		21
2004-01-12-16-1	30	2	4	2	10	5		24
….								

OpenSOC-ML


KAFKA TOPIC: ENRICHED
{“msg_content”: “This is a test log message”
{“msg_content”: “This is a test log message”
{“msg_content”: “This is a test log message”,
processed: [TIMESTAMP], src_ip: “10.20.30.40”, src_port: 123, dest_ip: “20.30.40.50”,
dest_port: 456, domain: “mydomain.com”, protocol:"TCP"
geo_enrichment:{
source:{
postalCode: 95134,areaCode: 408,metroCode: 807,longitude:-121.946,
latitude:37.425,locId:4522,city:"San Jose",country:"US"}
dest:{
postalCode: 95134,areaCode: 408,metroCode: 807,longitude:-121.946,
latitude:37.425,locId:4522,city:"San Jose",country:"US"}}
whois_enrichment:{
source: {
OrgId":"CISCOS","Parent":"NET-144-0-0-0-0",”
OrgAbuseName":"Cisco Systems Inc",
"RegDate":"1991-01-171991-01-17","OrgName":"Cisco Systems",
"Address":"170 West Tasman Drive",
"NetType":"Direct Assignment"}
dest: {
OrgId":"CISCOS","Parent":"NET-144-0-0-0-0",
"OrgAbuseName":"Cisco Systems Inc",
"RegDate":"1991-01-171991-01-17","OrgName":"Cisco Systems",
"Address":"170 West Tasman Drive",
"NetType":"Direct Assignment"}}}
OpenSOC-Streaming
OpenSOC-ML
.
KAFKA
alert:
{
anomaly-type: ”irregular pattern”,
models-triggered: [m1,m2,m4]
}

ML Topology

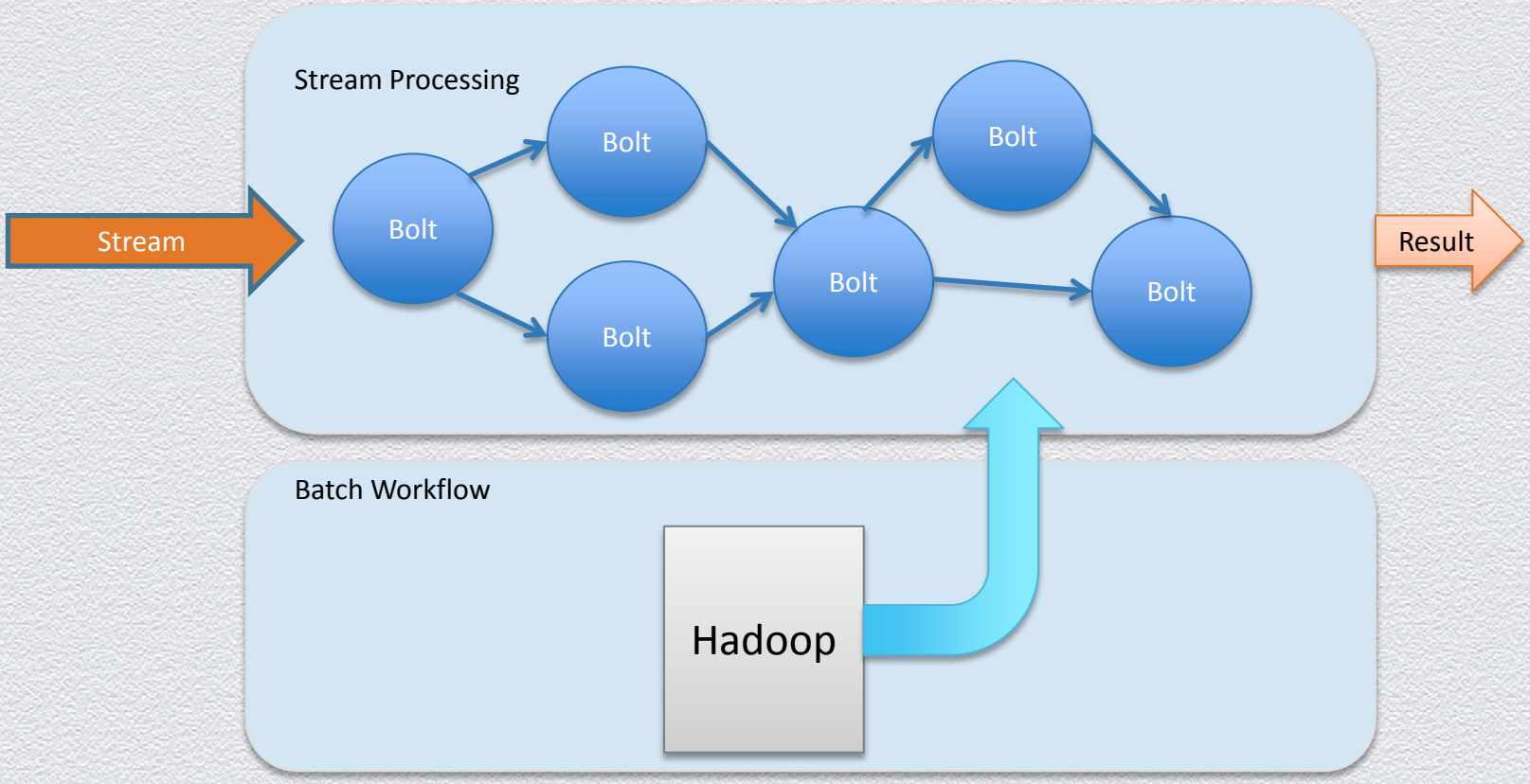


Survey of Algorithms Used by OpenSOC

Types of Algorithms

- **Offline** – analyze entire data set at once
  - Generally a good fit for Apache Hadoop/Map Reduce
  - Model compiled via batch, scored via stream processor
- **Online** – start with an initial state and analyze each piece of data serially one at a time
  - Generally require a chain of Map Reduce jobs
  - Good fit for Apache Spark, Tez, Storm
  - Primarily batch, good for Lambda architectures
- **Online Streaming** – real-time, in-memory, limited analysis
  - Generally a good fit for Apache Storm, Spark-Streaming
  - Probabilistic, random structures, approximations

Online Models



Examples: Stream Clustering

- **StreamKM++**: computes a small weighted sample of the data stream and it uses the k-means++ algorithm as a randomized seeding technique to choose the first values for the clusters.
- **CluStream**: micro-clusters are temporal extensions of cluster feature vectors. The micro-clusters are stored at snapshots in time following a pyramidal pattern.
- **ClusTree**: It is a parameter free algorithm automatically adapting to the speed of the stream and it is capable of detecting concept drift, novelty, and outliers in the stream.
- **DenStream**: uses dense micro-clusters (named core-micro-cluster) to summarize clusters. To maintain and distinguish the potential clusters and outliers, this method presents core-micro-cluster and outlier micro-cluster structures.
- **D-Stream**: maps each input data record into a grid and it computes the grid density. The grids are clustered based on the density. This algorithm adopts a density decaying technique to capture the dynamic changes of a data stream.
- **CobWeb**: uses a classification tree. Each node in a classification tree represents a class (concept) and is labeled by a probabilistic concept that summarizes the attribute-value distributions of objects classified under the node

Example: Stream Classification

- **Hoeffding Tree (VFDT)**
  - incremental, anytime decision tree induction algorithm that is capable of learning from massive data streams, assuming that the distribution generating examples does not change over time

- **Half-Space Trees**
  - ensemble model that randomly spits data into half spaces. They are created online and detect anomalies by their deviations in placement within the forest relative to other data from the same window

Examples: Outlier Detection[10]

- **Median Absolute Deviation**: Telemetry is anomalous if the deviation of its latest datapoint with respect to the median is X times larger than the median of deviations

- **Standard Deviation from Average**: Telemetry is anomalous if the absolute value of the average of the latest three datapoint minus the moving average is greater than three standard deviations of the average.

- **Standard Deviation from Moving Average:** Telemetry is anomalous if the absolute value of the average of the latest three datapoints minus the moving average is greater than three standard deviations of the moving average.

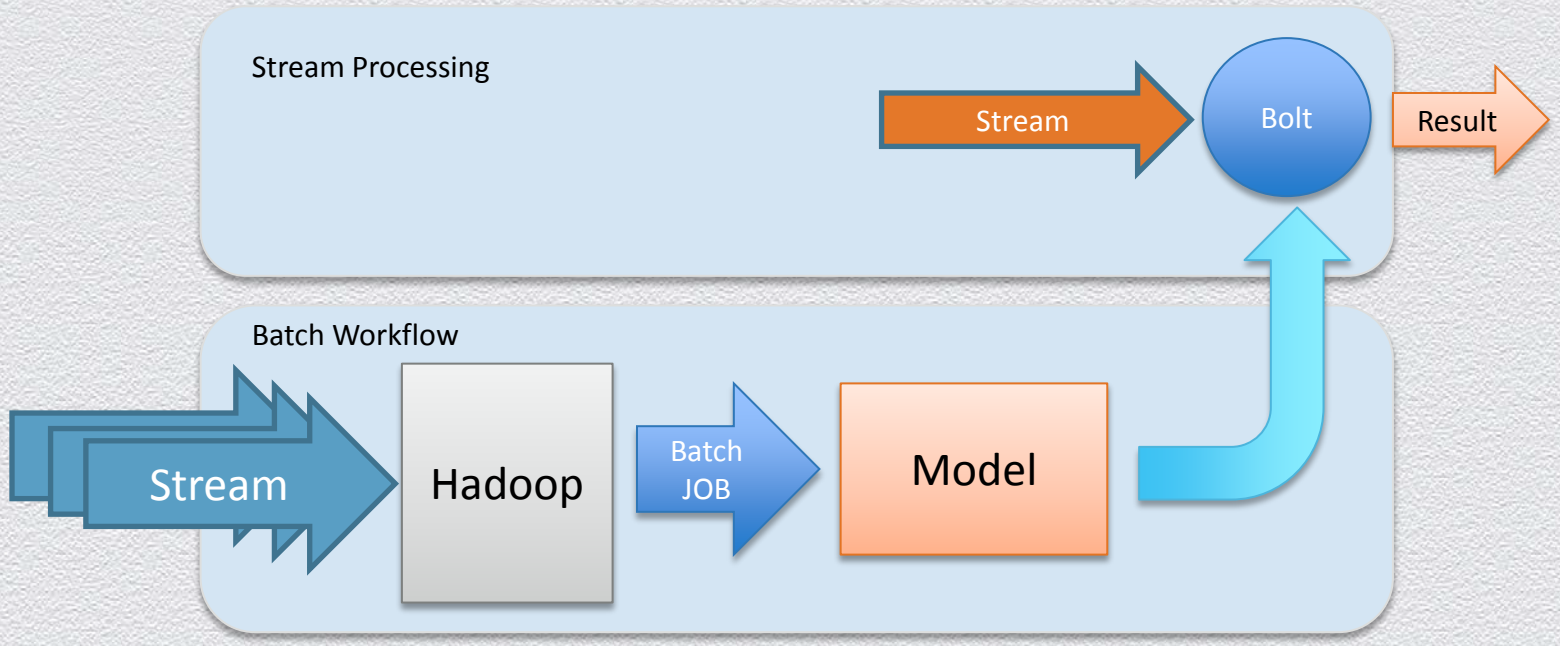
Examples: Outlier Detection [10]

- **Mean Subtraction Cumulation:** Telemetry is anomalous if the value of the next datapoint in the series is farther than three standard deviations out in cumulative terms after subtracting the mean from each data point

- **Least Squares:** Telemetry is anomalous if the average of the last three datapoints on a projected least squares model is greater than three sigma

- **Histogram Bins:** Telemetry is anomalous if the average of the last three datapoints falls into a histogram bin with less than x

Offline Models



Offline Algorithms

- **Hypothesis Tests**
  - **Chi2 Test** (Goodness of Fit): A feature is anomalous if it the data for the latest micro batch (for the last 10 minutes) comes from a different distribution than the historical distribution for that feature
  - **Grubbs Test:** telemetry is anomalous if the Z score is greater than the Grubb's score.
  - **Kolmogorov-Smirnov Test:** check if data distribution for last 10 minutes is different from last hour
  - **Simple Outliers test:** telemetry is anomalous if the number of outliers for the last 10 minutes is statistically different then the historical number of outliers for that time frame
- **Decision Trees/Random Forests**
- **Association Rules (Apriori)**
- **BIRCH/DBSCAN Clustering**
- **Auto Regressive (AR) Moving Average (MA)**

Approximation for Streaming

Approximation for Streaming

- **Skip Lists**
  - Trade memory for lookup time increase
  - Hierarchical layered indexing on top of lists
  - Lookup improvement: O(n) -> O(log n)
  - Best Uses:
    - Search for element in a list
    - Approximate the number of distinct elements in a list

Approximation for Streaming

- **HyperLogLog**
  - Turns each feature to a hash value
  - Keeps track of the longest run of a feature
  - Approximates the number of occurrences of a feature based on it's longest runs
  - Best Uses:
    - Estimate how rare an occurrence of a feature is
    - Approximate count of distinct objects over time
    - Estimate entropy of a data set

Approximation for Streaming

- **Bloom Filters**
  - Trade memory for fast set membership check
  - Hash all incoming elements to hash sets
  - Use multiple hash functions and multiple hash sets of varying sizes
  - Best Uses:
    - Check element membership in a set

Approximation for Streaming

- **Sketches**
  - Top-K: count of top elements in the stream
  - Count-Min: histograms over large feature sets (similar to Bloom Filters with counts)
- **Digest**
  - algorithm for computing approximate quantiles on a collection of integers

Thank You

Follow us on Twitter

@ProjectOpenSOC

MTD Data Science Team

James Sirota @**JamesSirota**

Sam Davis @**samdavis510**

Nate Bitting @**nbitting**

Sources

[1] RSA Incident Response Teams are the New (Security) Black https://blogs.rsa.com/incident-response-teams-new-security-black/

[2] PWC - The Global State of Information Security Survey 2014 http://download.pwc.com/ie/pubs/2013_key_findings_from_the_global_state_of_information_security_survey_2014.pdf

[3] Tripwire 2014 IT SECURITY BUDGET FORECAST ROUNDUP FOR CIOs AND CISOs/CSOs https://www.akat-t.com/wp-content/uploads/2014/01/IT-Security-Budget-Forecast-Roundup-2014-for-CIOs-and-CSO-CISOs.pdf

[4] NetworkWorld: Real-Time Big Data Security Analytics for Incident Detection http://www.networkworld.com/article/2225959/cisco-subnet/real-time-big-data-security-analytics-for-incident-detection.html

[5] CAS Static Analysis Tool Study http://samate.nist.gov/docs/CAS_2011_SA_Tool_Method.pdf

[6] TIBCO Cyber Security Platform http://www.tibco.com/multimedia/wp-cyber-security-platform_tcm8-16777.pdf

[7] Reuters http://www.reuters.com/article/2012/06/13/us-media-tech-summit-symantec-idUSBRE85B1E220120613

[8] Staffing the Informaton Security Organization https://vostrom.com/get/InfoSec_Staffing.pdf

[9] SBT Global Survey us.westcon.com/documents/.../stbglobalsurveywpen11nov2013web.pdf

[10] Etsy Kale http://codeascraft.com/2013/06/11/introducing-kale/