Modern Security Analytics
Finding a Needle in the Hayblower

Martin Rehak, Principal Engineer

July 1, 2014, AIMS 2014, Brno

Stream Security Analytics

- What is malware and why do we need to fight it?
- Step-by-step walkthrough through a security incident
- Security domain considerations, evolutions and particularities
- Stream Analytics: understanding big data in flight
- Ensemble of anomaly detectors and collective classification
- False Positives Analysis

The Advanced Malware Attack Lifecycle

PLAN

Attacker determines possible entry points, formulates a plan of attack

EXPLOIT / ATTACK

Attacker exploits vulnerabilities and delivers its weapon

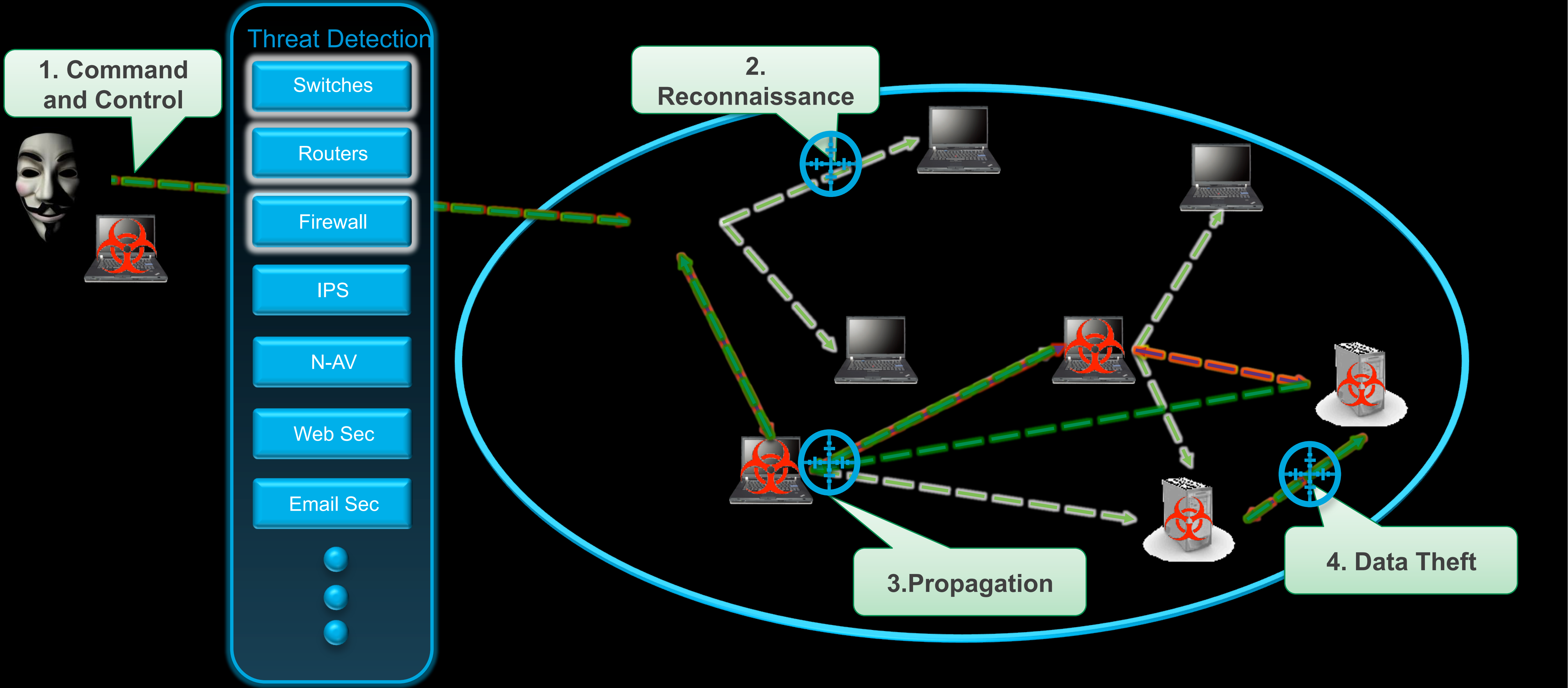
INFECT / SPREAD

Malware moves laterally through the internal network in search of additional resources and data

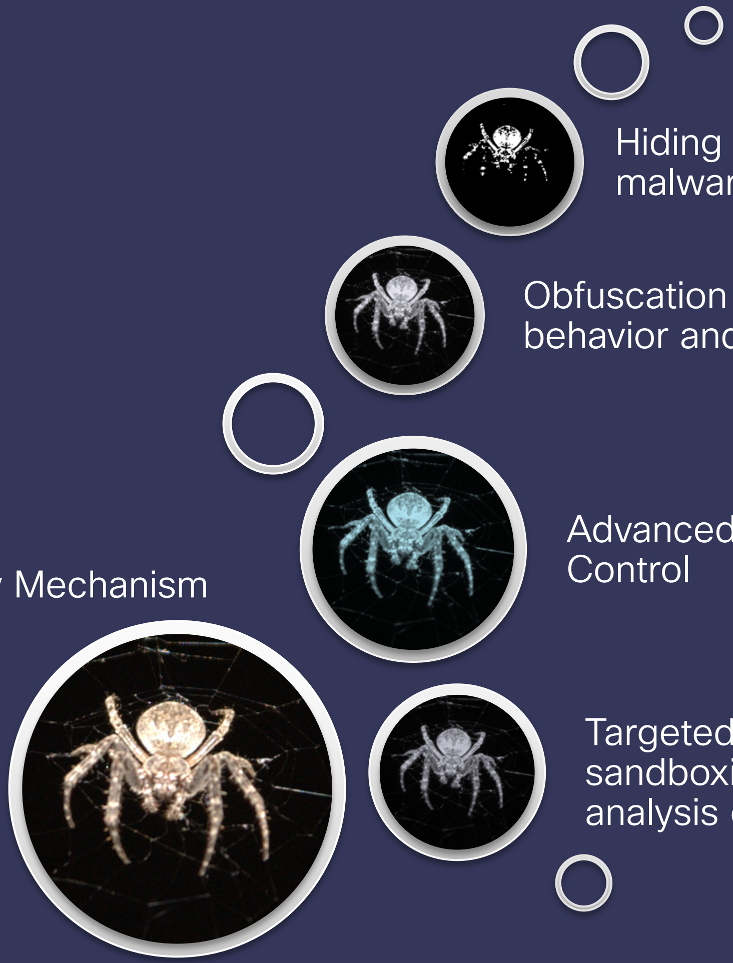
STEAL / DISRUPT

Attacker takes action on its objectives and exfiltrates data or disrupts systems

Attack Kill Chain: Post Breach



Technology Escalation in Advanced Cyber Threats



Hiding between the random malware infections

Obfuscation of content, behavior and mission

Advanced Command & Control

Improved Delivery Mechanism

Targeted execution, limiting sandboxing, detection and analysis options

(Trivial) Incident Walkthrough

(Slides removed)

High-Level View of the System

Cloud Web Security

The Security Perimeter in the Cloud

Collective Security Intelligence

Telemetry Data

Threat Research

Advanced Analytics

The Distributed Perimeter

Cloud Connected Network

Mobile

Router

Firewall

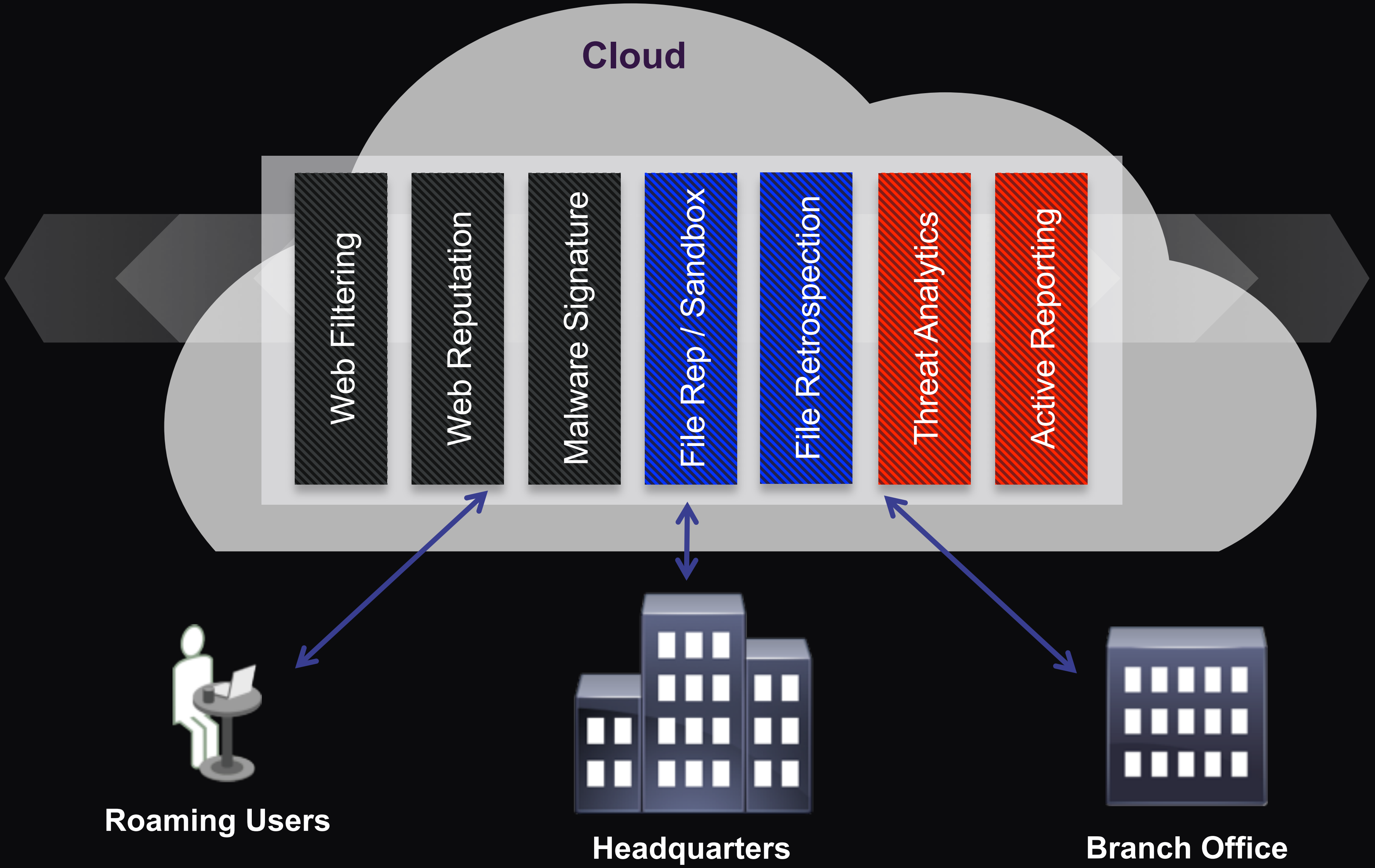
3M+
Cloud Web Security Users

6 GB
Web Traffic Examined, Protected Every Hour

75M
Unique Hits Every Hour

10M
Blocks Enforced Every Hour

Cisco Cloud Delivered Security Capability



Three Worlds of Security Analytics

Pattern Matching

- Finds **known malware** or other attacks by matching the content of communication or files against patterns of known attacks.

Threat Intelligence

- Analyses **known malicious behavior to infer general characteristics** of attacks and uses the model to discover new attacks by their associations with known attacks.

Anomaly Detection

- Analyses the **normal behavior** of the network to build a predictive model and **detects any patterns deviating from the normal behavior** as potentially malicious.

Pattern Matching

- Lowest False Alerts
- Very specific and verifiable convictions
- Proven and traditional industry standard…
- … that needs to be complemented by other techniques in order to cope with advanced attacks.

Threat Intelligence

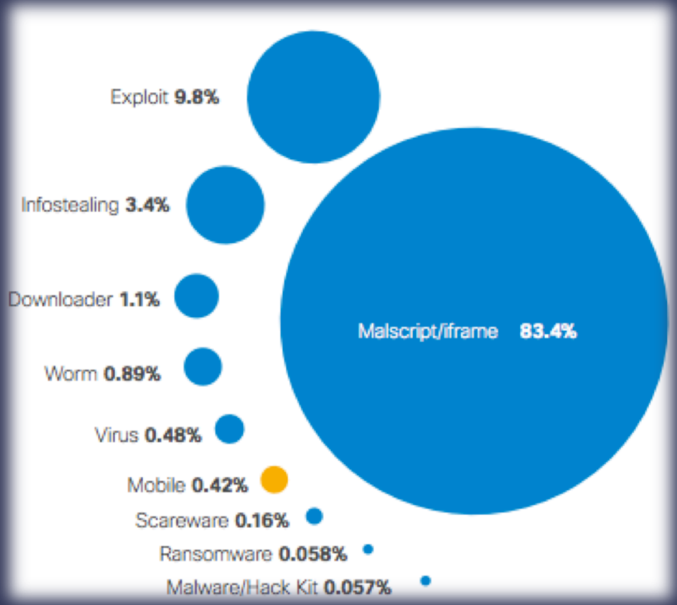
- Medium false alerts
- Convictions are not direct, but still understandable by humans
- Allows the detection of malware by exploiting the operational and technical imperfections of the attackers…
- … not committed by the advanced organizations.

Anomaly Detection

- Higher false alerts
- Convictions are not specific and are difficult to explain
- Allows the detection of broadest scope of malware, including the advanced attacks…
- … provided that we can separate the malware from random statistical fluctuations

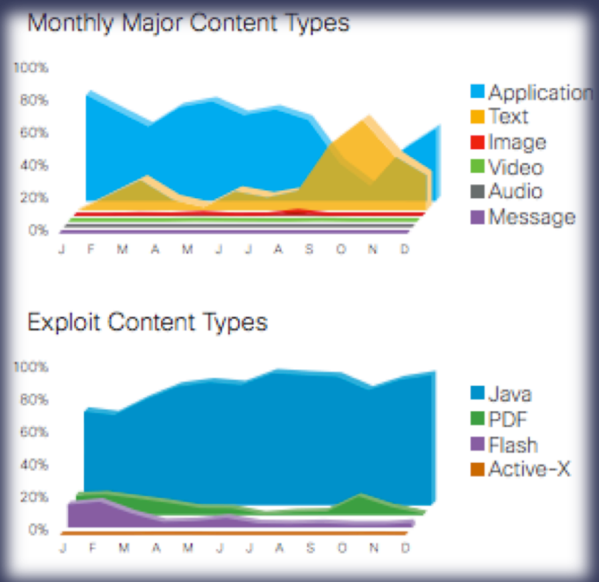
Pattern Matching

- Complexity: Proportional to the number of known and described unique samples/traits that need to be described



Threat Intelligence

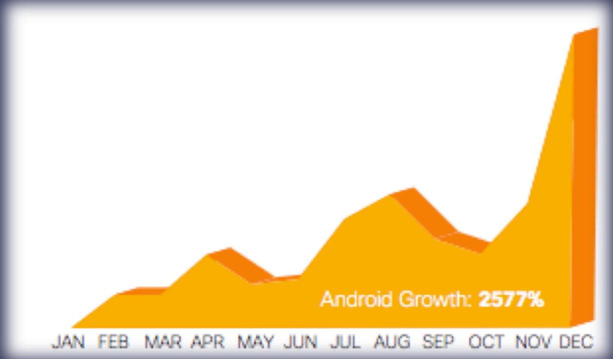
- Complexity: Proportional to the number of known botnets, malicious infrastructures,


Monthly Major Content Types

Exploit Content Types

Anomaly Detection

- Complexity: proportional to the number of users and organizations protected



Anomaly Detection – A Machine Learning Problem

Anomaly detection requires us to build a predictive model of

1. global internet traffic
2. each customer’s network, and
3. each host or user in each customer’s network

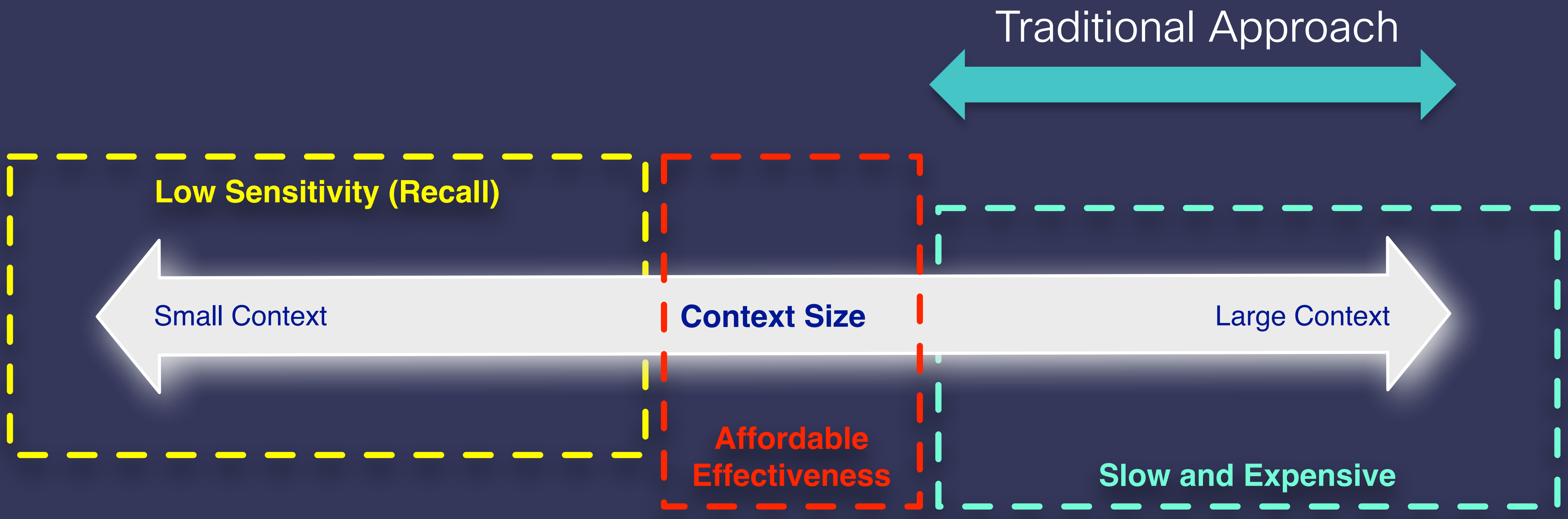
while respecting strict privacy and economic constraints.

“Traditional” Big Data Workflow

1. Receive the data
2. Perform an ETL (Extract-Transform-Load) process
   - Check correctness, formatting, deduplication
   - Extract Features
   - Schedule for writing into DB/Storage
3. Store the transformed data in data store
4. Run the analytics process on the data to find value
5. Display the results

Cost Breakdown

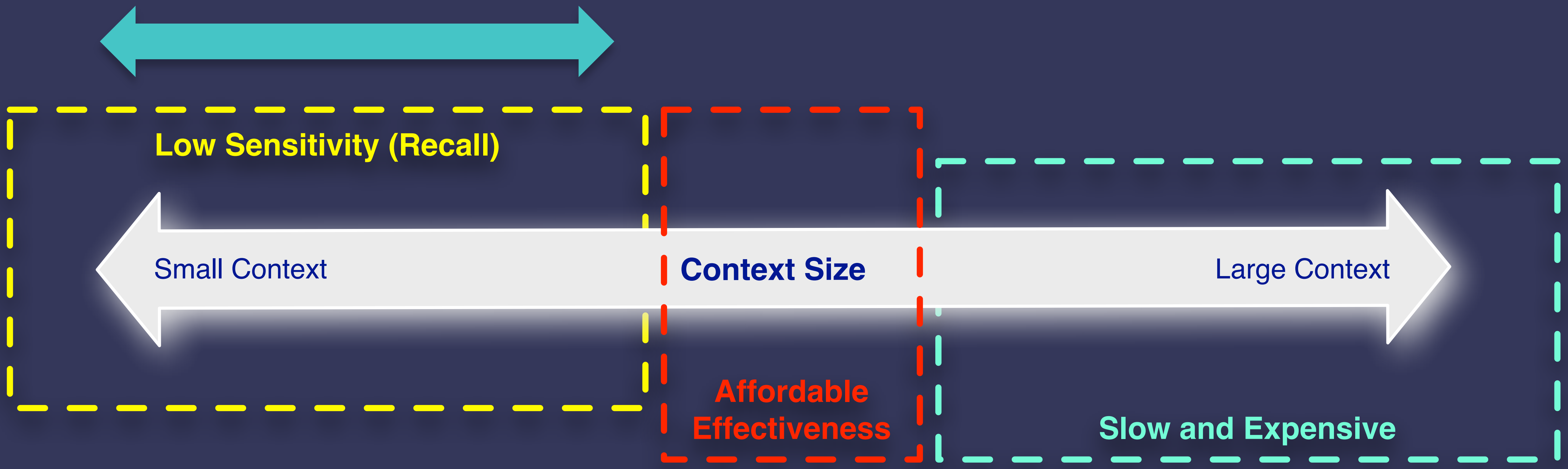
1. Receive the data
2. Perform an ETL (Extract-Transform-Load) process
    - Check correctness, formatting, deduplication
    - Extract Features
    - Schedule for writing into DB/Storage
3. Store the transformed data in data store
4. Run the analytics process on the data
5. Display the results


Traditional Approach
Low Sensitivity (Recall)
Small Context
Context Size
Large Context
Affordable Effectiveness
Slow and Expensive

Stream Analytics

1. Receive the data
2. Perform an ETL (Extract-Transform-Load) process
   - Check correctness, formatting, deduplication
   - Extract Features
   - Schedule for writing into DB/Storage
3. Run fast and effective analytics process on the data
4. Store the transformed and filtered data in data store
5. Run (second) analytics process
6. Display the results

Simple Stream Analytics



Low Sensitivity (Recall)

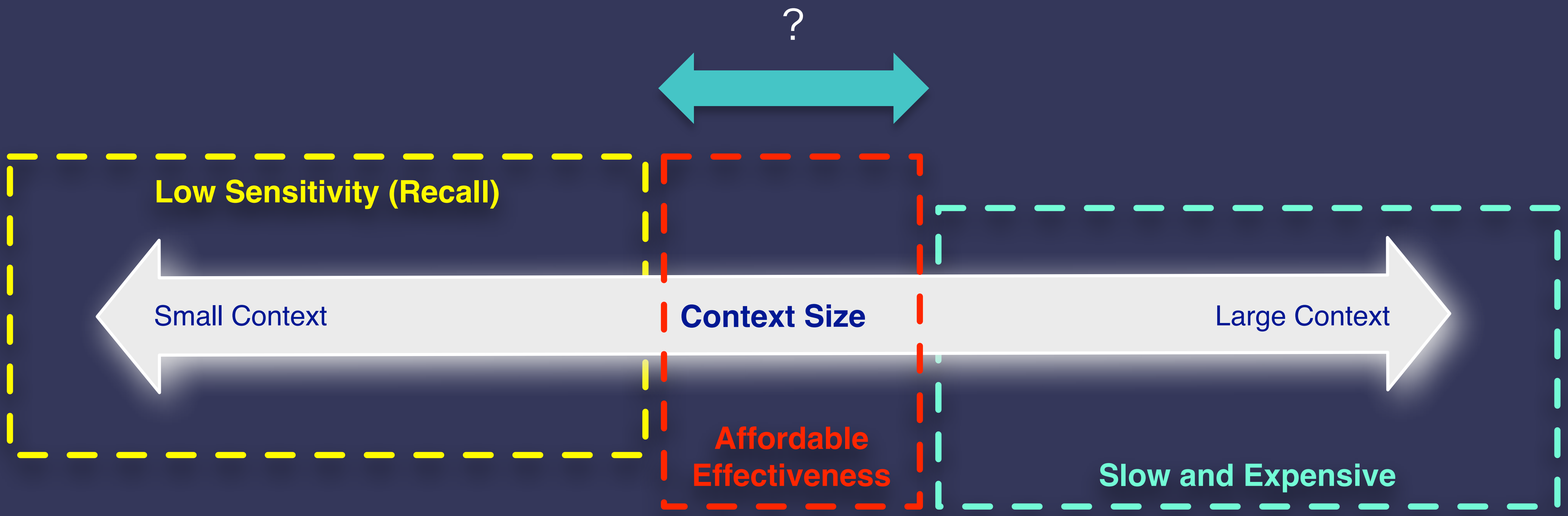
Small Context

Context Size

Large Context

Affordable Effectiveness

Slow and Expensive



?

Low Sensitivity (Recall)

Small Context

Context Size

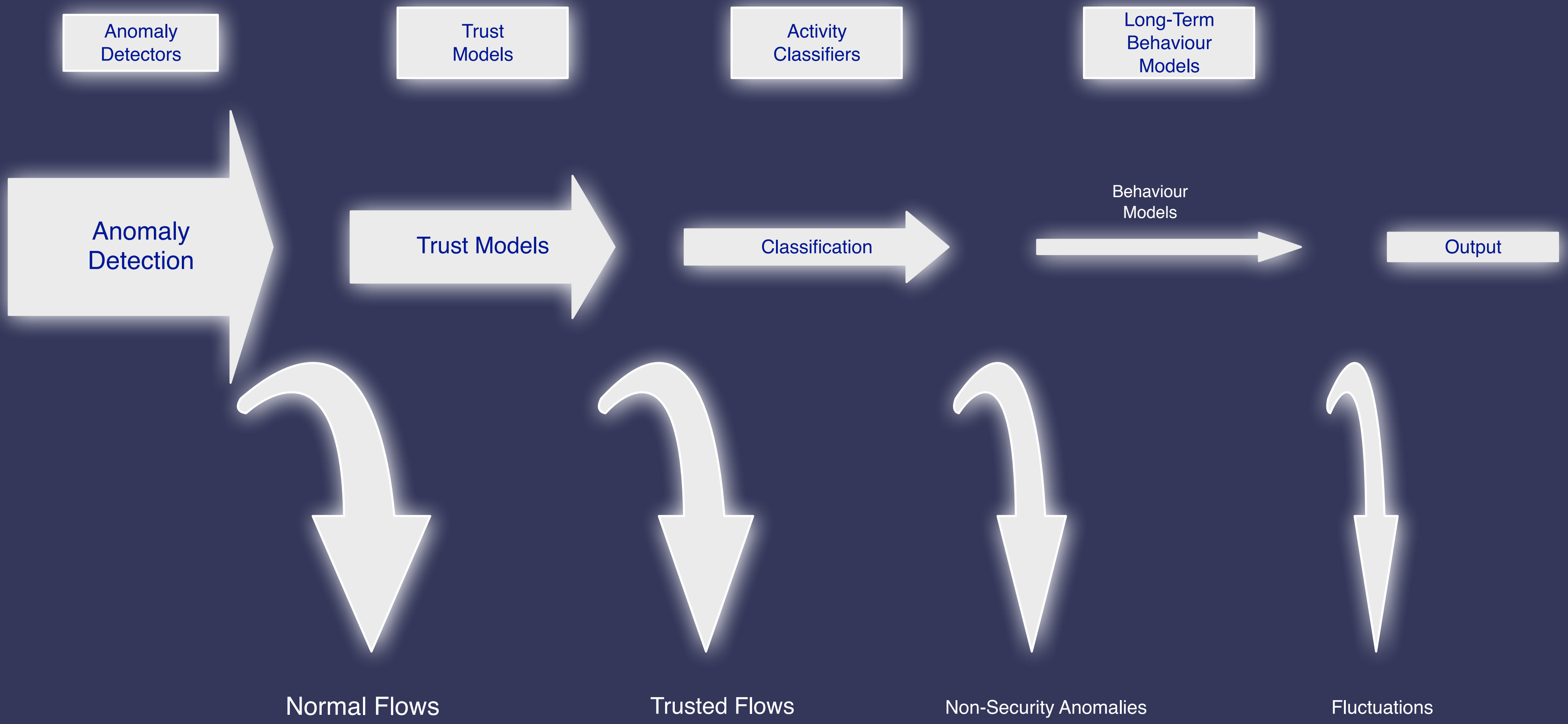
Large Context

Affordable Effectiveness

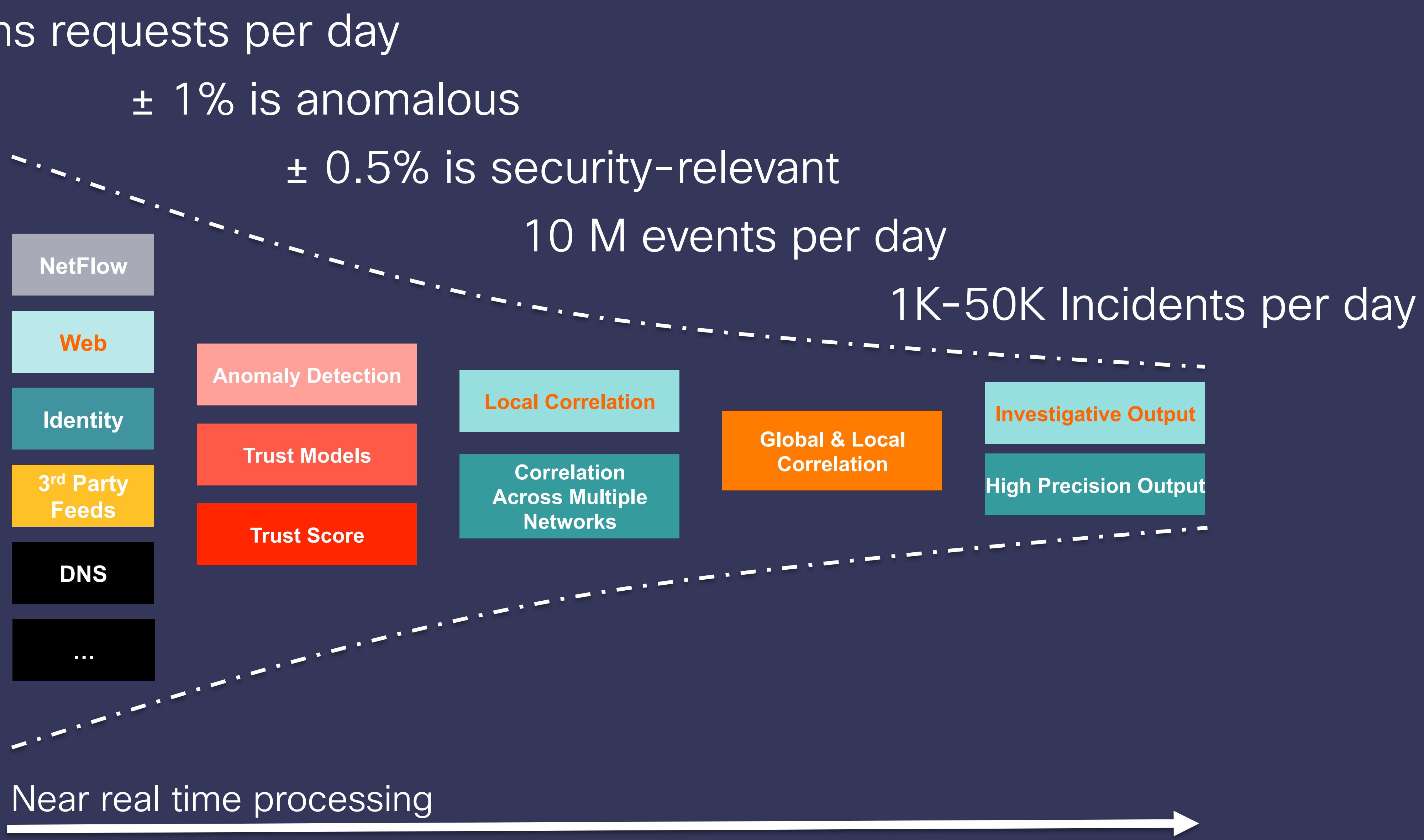
Slow and Expensive

Inside View

Model Decomposition – Hierarchical and Distributed



- M Rehak, M Pechoucek, M Grill, J Stiborek, K Bartos, P Celeda. Adaptive multiagent system for network traffic monitoring. Intelligent Systems, IEEE 24 (3)



5 billions requests per day

± 1% is anomalous

± 0.5% is security-relevant

10 M events per day

1K-50K Incidents per day

- NetFlow
- Web
- Identity
- 3rd Party Feeds
- DNS
- …

- Anomaly Detection
- Trust Models
- Trust Score

- Local Correlation
- Correlation Across Multiple Networks

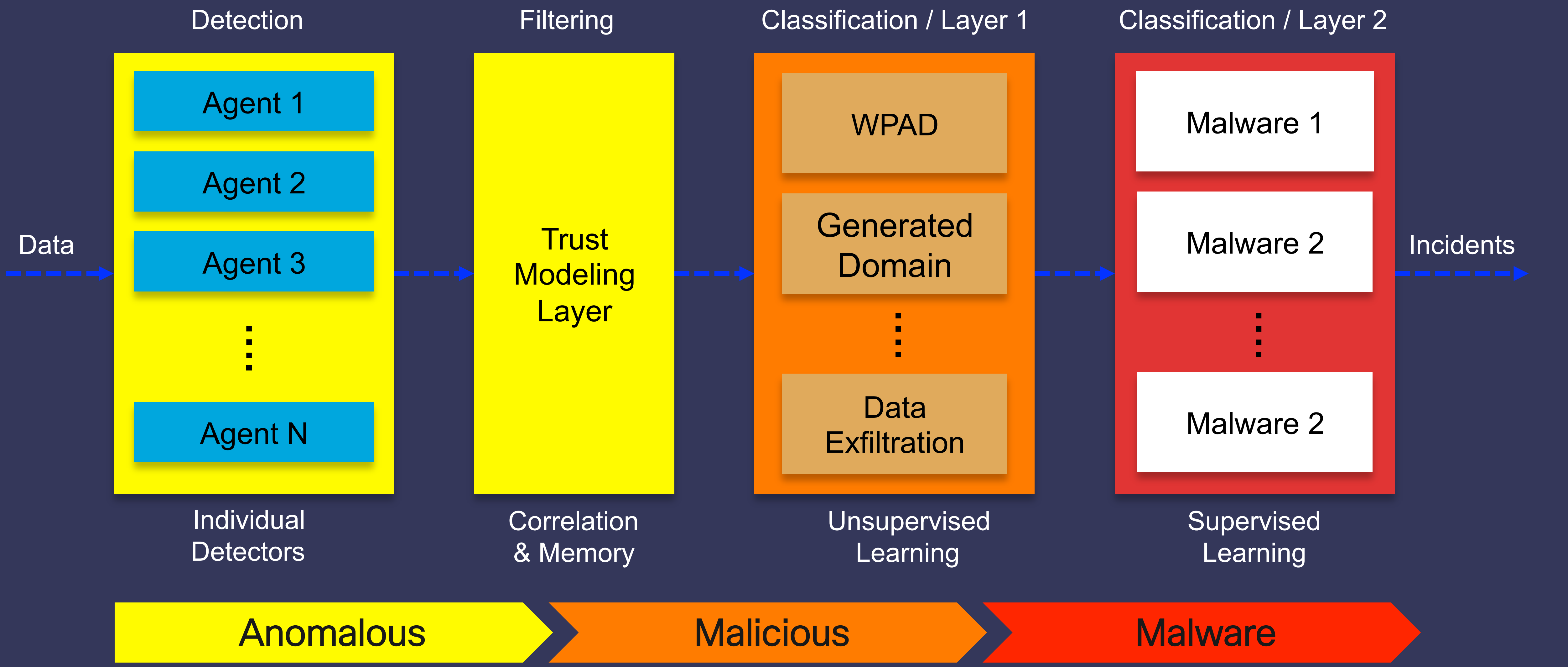
- Global & Local Correlation

- Investigative Output
- High Precision Output

Near real time processing

Cognitive Threat Analytics

Layered Detection Engine



Detection

Filtering

Classification / Layer 1

Classification / Layer 2

Data

Agent 1

Agent 2

Agent 3

⋮

Agent N

Trust Modeling Layer

WPAD

Generated Domain

⋮

Data Exfiltration

Malware 1

Malware 2

⋮

Malware 2

Incidents

Individual Detectors

Correlation & Memory

Unsupervised Learning

Supervised Learning

Anomalous

Malicious

Malware

Examples of AD output (HTTP, real and synthetic malware)

Typical anomaly detection algorithm

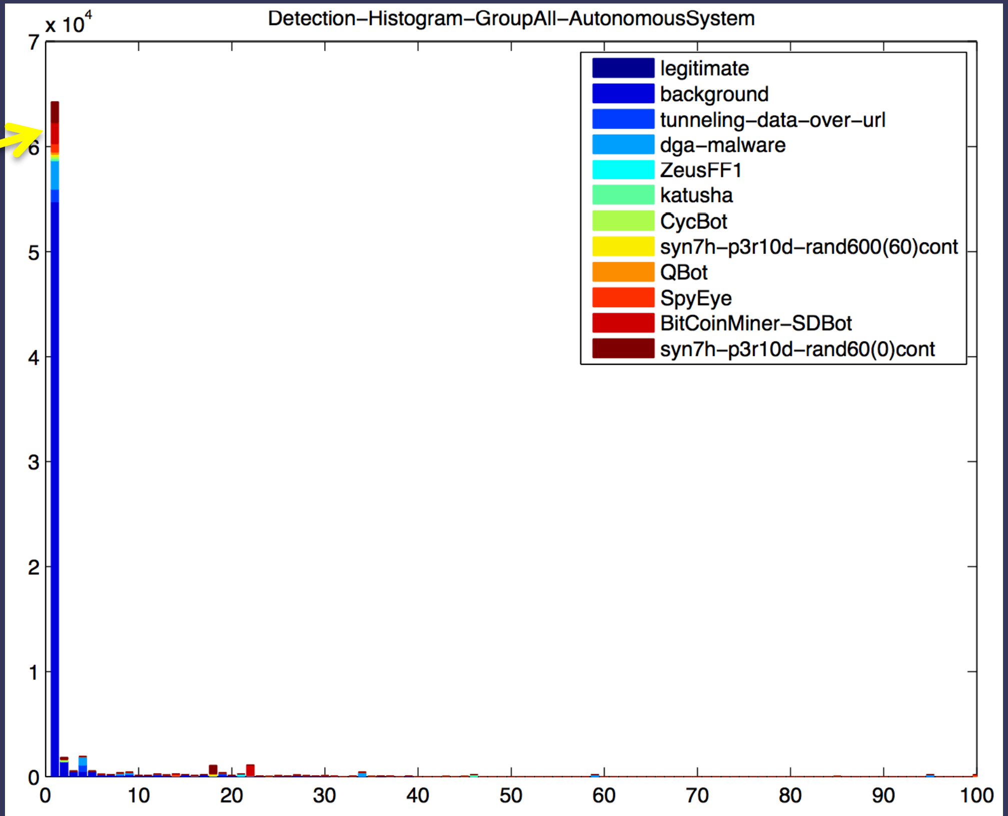
does not quite work

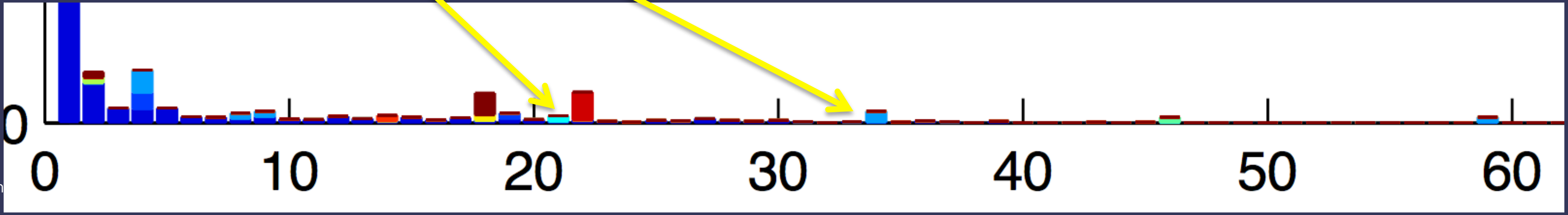
Problems:

False negatives

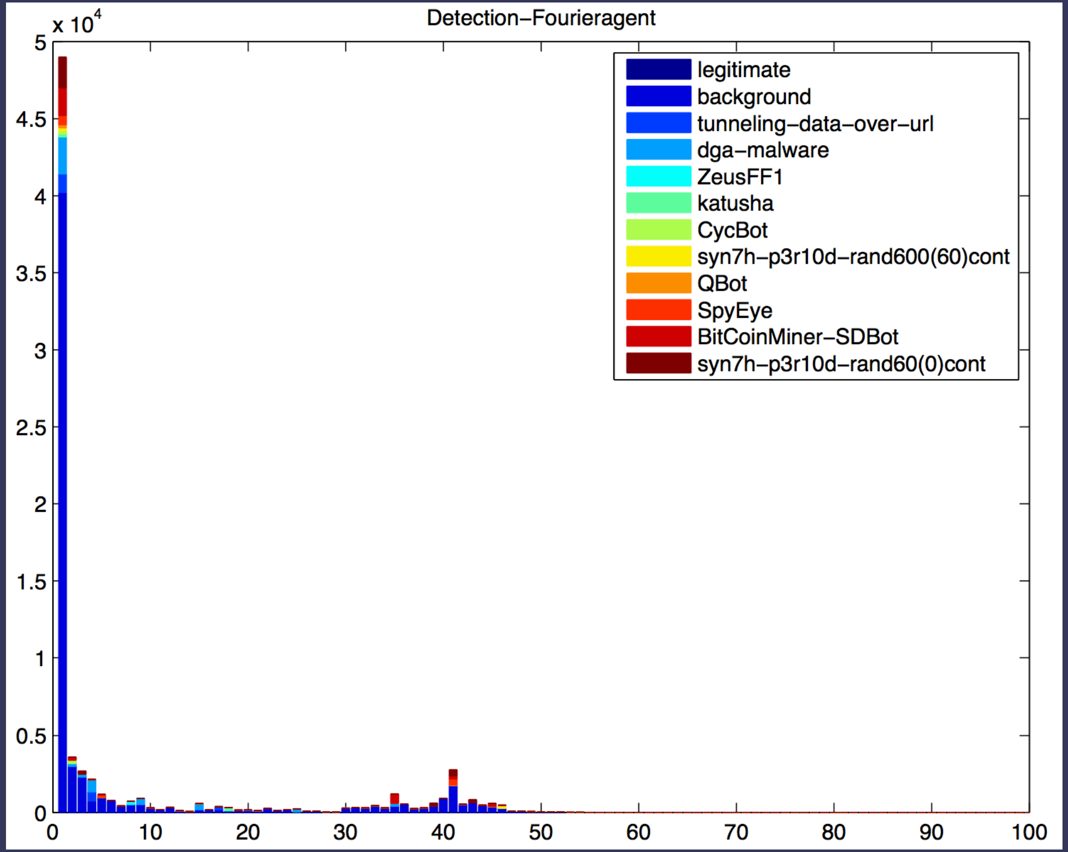
False positives

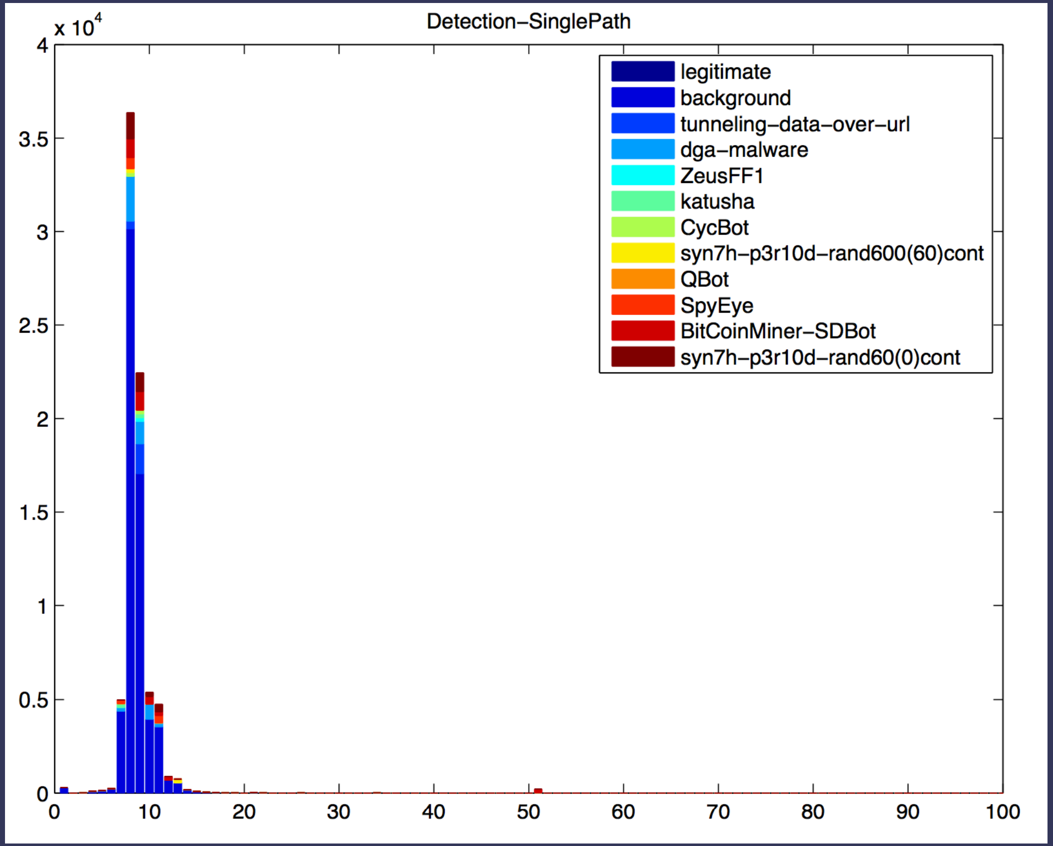
AUC around 0.7-0.8 not an exception

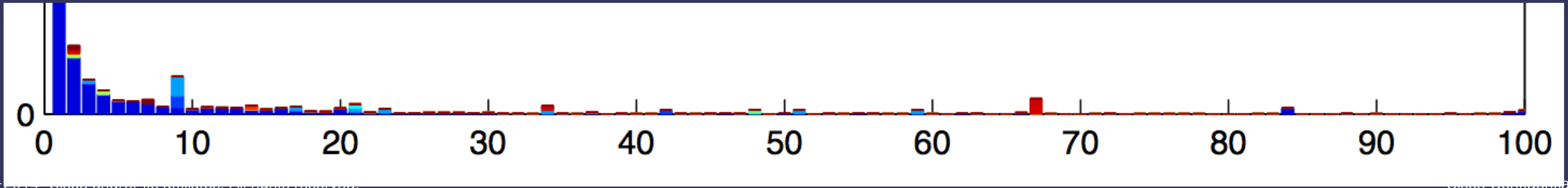




More AD output samples







Anomaly Detection Approaches

Statistics and Empirical Distribution, Information Theory:

- M Rehak, M Pechoucek, K Bartos, M Grill, P Celeda. Network intrusion detection by means of community of trusting agents - IAT 2007 - IEEE/WIC/ACM I.C. Intelligent Agent Technology

Principal Components Analysis:

- T Pevný, M Rehák, M Grill. Identifying suspicious users in corporate networks. Proceedings of workshop on information forensics and security, 2012, 1-6

Graph Theory:

- Jusko, J. and Rehak, M. (2014), Identifying peer-to-peer communities in the network by connection graph analysis. Int. J. Network Mgmt.. doi: 10.1002/nem.1862
- Jusko, J, Rehak, M. (2012) Revealing Cooperating Hosts by Connection Graph Analysis, Securecomm 2012

Anomaly Detectors

Individual AD models can detect malicious behavior as an outlier

- Making them better would compromise their generality

Base-Rate Fallacy as a fundamental limitation

- Not enough attacks in the traffic

Precision and Recall are not sufficient for direct use

Can we make the Precision better?

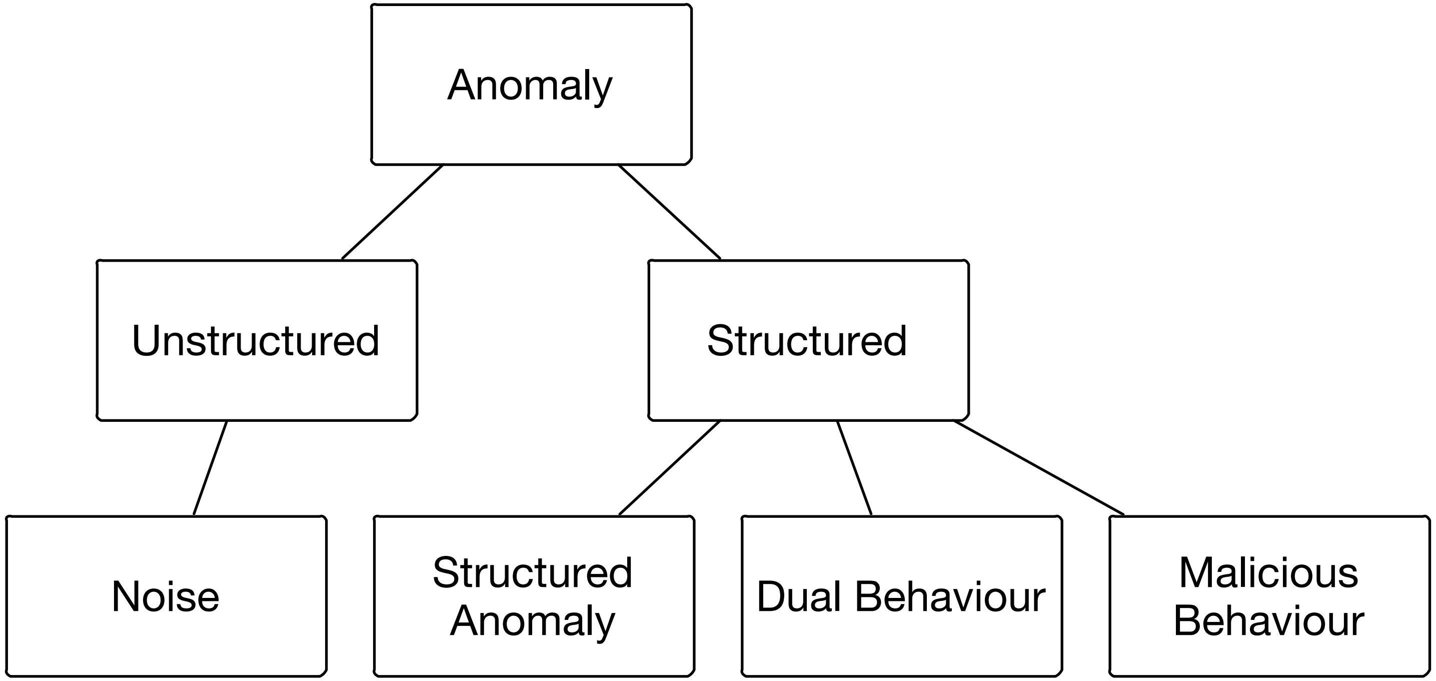
Unstructured

- Short-term artifacts
- Uniformly distributed over users
- Proportional to traffic volumes
- Frequently associated with novelty, rather than anomaly

Structured

- Long-term behaviors
- Confined to subset of hosts/users
- No direct relationship to background
- Structured and explainable

High-Level False Positive Breakdown



Anomaly

Unstructured

Structured

Noise

Structured Anomaly

Dual Behaviour

Malicious Behaviour

Unstructured Anomalies - Noise

- Unstructured, short-term anomalies
- Evenly distributed over hosts with the distribution proportional to the background traffic
  - Triggered by widespread, uniformly distributed behaviors (such as web browsing or SW updates).
  - Small proportion of high-volume behavior will be anomalous.
- Same anomaly instance generated by one or small number of hosts only, but noise categories and types can spawn over many or almost all hosts.
  - One specific favicon anomaly will be there for one user, **but every user will have a favicon anomaly with some probability**
- Examples:
  - **Novelty**: Users visiting unusual discussion forums or unusual documents (how to repair a dishwasher X)
  - Random failures: Redirects, dead links, incorrect retries
  - **Fragments**: Fragments of legitimate browsing behavior (ads from less known sites invoked by js)
  - **Multimedia**: Flash games & other multimedia fragments

- T Pevny, M Komon, M Rehak. Attacking the IDS learning processes. Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE

(Slides removed)

Structured Anomaly

- Understandable outlier behavior with internal structure of each instance
- Long-term behavior
- Specific to one host or a small subset
  - The type of the anomaly will be restricted to a small subset of hosts
  - Triggered by a specific application (software update/software usage), context (wpad) or by combination of both factors (Skype running in some context)
- Additional evidence allows its (ex-post) separation from malicious behavior
- Examples:
  - Software updates
  - Skype/P2P over HTTP(S)
  - Configuration scripts, remote log creation, regular calls of unusual APIs

(Slides removed)

Dual Anomalies

- Major characteristics are identical with Structural Anomalies
  - Understandable behavior with internal structure of each instance
  - Long-term behavior
  - Specific to one host or a small subset
- Events, where the maliciousness can only be decided on the level of intent, as the identical events would appear in malicious and legitimate scenarios alike.
  - Features necessary for separation are not not be accessible (judging the HTTPS by the endpoint) or the behaviors are inherently inseparable.
  - Direct effect is the same in case of attack or legitimate activity. Intent can be dramatically different.
- Example:
  - Large file download/upload
  - Long-lasting HTTPS connection to a single host outside the company network

(Slides removed)

Malicious Behavior

- Major characteristics are close to Structural Anomalies or Dual Behaviors
  - Understandable behavior with internal structure of each instance
  - Long-term behavior
  - Specific to one host or a small subset
- Our experience, correlated evidence or the behavior itself allow us to conclusively confirm the behavior as malicious upon examination.
- Prioritization within the malware category is not trivial!

(Slides removed)

Unstructured Noise

- Caused by inherent unpredictability & randomness of the modeled system, computational limitations imposed on anomaly detectors and predictive precision of the anomaly detection algorithms.
- Can be reduced, but never fully eliminated.

Structured Anomaly

- Caused by the limited capability of anomaly detectors that are unable to model whole classes of behavior. Capability can be either a modeling problem (such as wrong timescale), or a feature selection problem.
- Separable, but not separated…

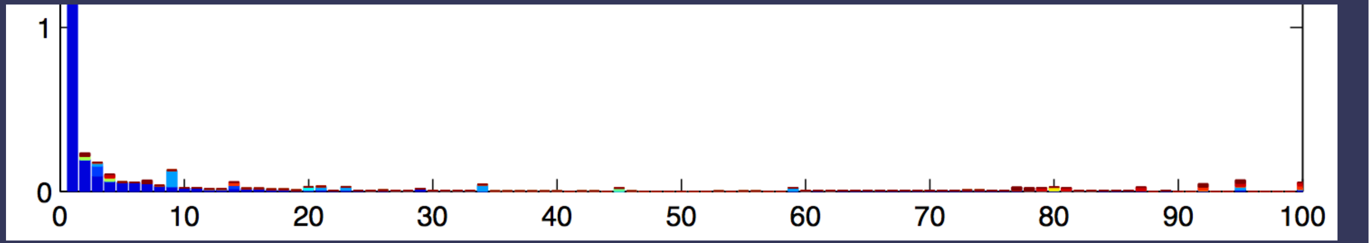
Dual Behavior

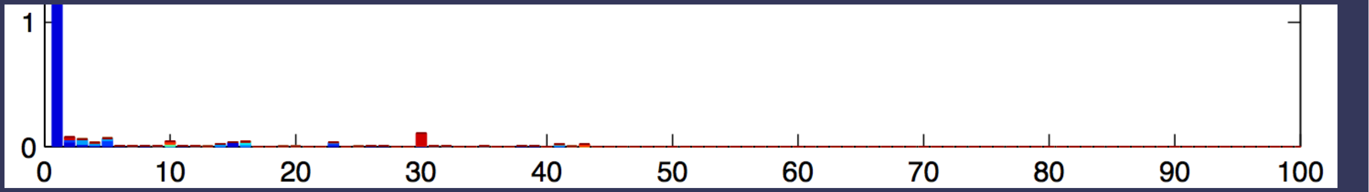
- The accessible features do not allow separation between the legitimate and malicious behavior of this class without a qualitatively different model.
- Opens exciting new research areas:
  - Intent modeling
  - Strategic correlation
  - Plan recognition
  - …

Assumptions for Ensemble Approach

Model Diversity: We assume that the methods and features used by the same model ensure that the fact that one model has singled out a legitimate object as a false positive is not correlated with other models doing so

Model Alignment: We assume that malicious behavior detection is correlated between the models that we combine





And should the assumptions hold…

Averaging the outputs of several AD’s should allow you to dramatically increase the precision of the system without the negative impact on recall

But the assumptions hold only partially…

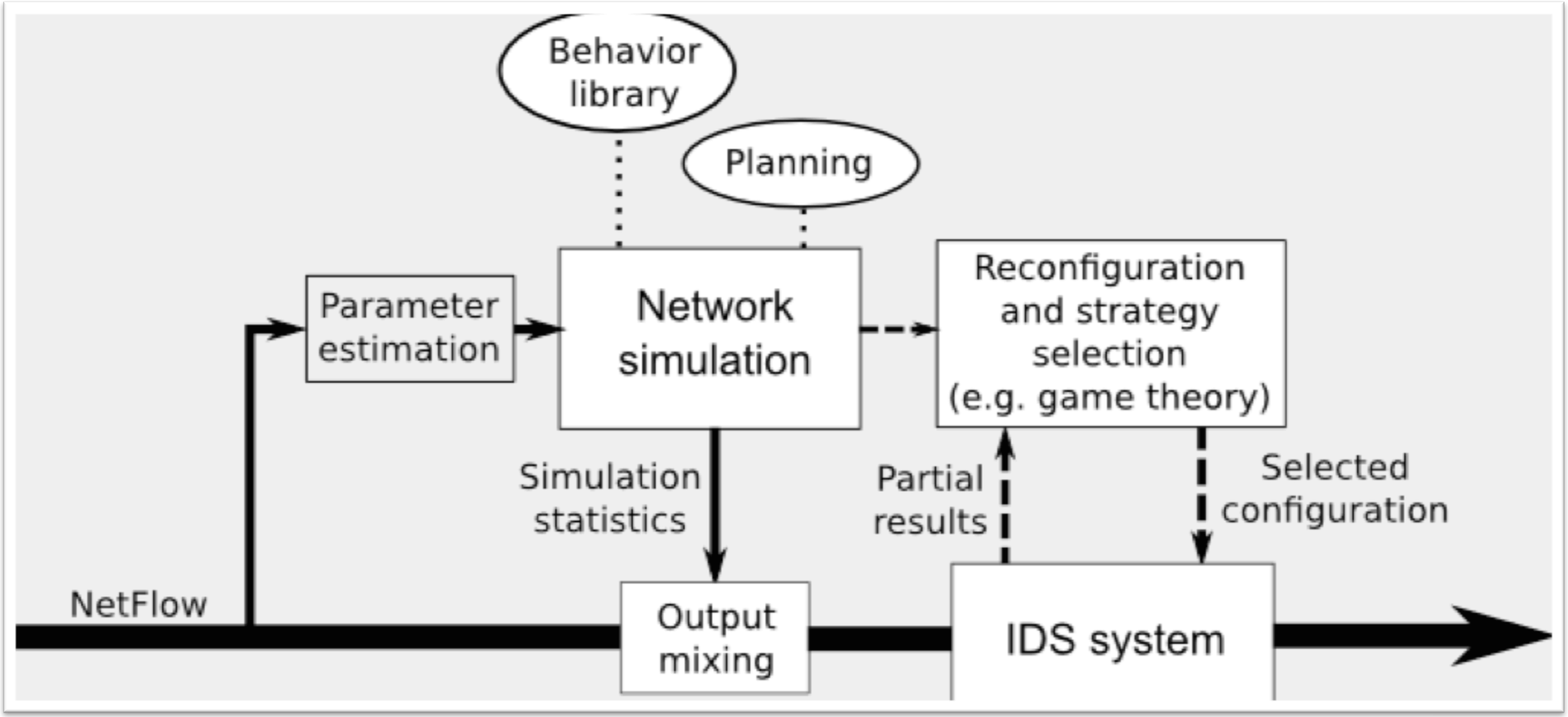
Limitations

Question: Actually, as system designers, we ought to be able to design the detectors with enough diversity by careful detection of features and AD methods – right?

Short answer: Sort of. Assuming the attackers are nice/stupid enough, or the AD methods are almost flawless in terms of generality.

Better answer:

- Diversity assumptions are relatively easy to satisfy and verify, as they rely on the background traffic characteristics. They also tend to carry over between the contexts (networks) reasonably well.
- General Alignment is far more difficult to guarantee due to the diversity of malicious behaviors. In reality, we don't average all the detectors, but select subsets of 4-10 detectors to optimize the recall.



- M Rehák, E Staab, V Fusenig, M Pěchouček, M Grill, J Stiborek, K Bartoš, Thomas Engel. Runtime monitoring and dynamic reconfiguration for intrusion detection systems, RAID 2008 – Recent Advances in Intrusion Detection
- J Stiborek, M Grill, M Rehak, K Bartos, J Jusko. Game Theoretical Adaptation Model for Intrusion Detection System, PAAMS 2012

(Slides removed)

User-Level Perception of the System

- **Precision alone** defines the user experience & operations
    - Government – more recall sensitive in top security areas
    - Multinational corporation – precision determines the real usability
    - Smaller enterprises and companies – even higher precision requirements

- Recall is (almost) always argued on the "compared to the state without the system X deployed" basis

- Improving recall is our professional duty, but the business rationale and user perception is less clear than in case of precision
    - But there is a catch…precision and recall are directly related through the **user bottleneck**