SECURITY MATTERS

Technische Universiteit Eindhoven University of Technology

Where innovation starts

From Intrusion Detection to Software Design

A position paper

Sandro Etalle

Why me

- Intrusion Detection in Industrial Control Systems

- First in academia
- Then, in our spin-off
  - CEO for 4 years+
  - I talked to customers

- SecurityMatters
  - Large install base
  - 40 people, and growing
  - Healthy and reputable
  - We must have done something right

The problem: attacks

We believe that fighting crime should be easy: we provide effective, easy-to-use offensive technology to the worldwide law enforcement and intelligence communities.

- striking is how easy it is to break into a system.

I believe that today the single most important reason why attacks are so difficult to counter is that

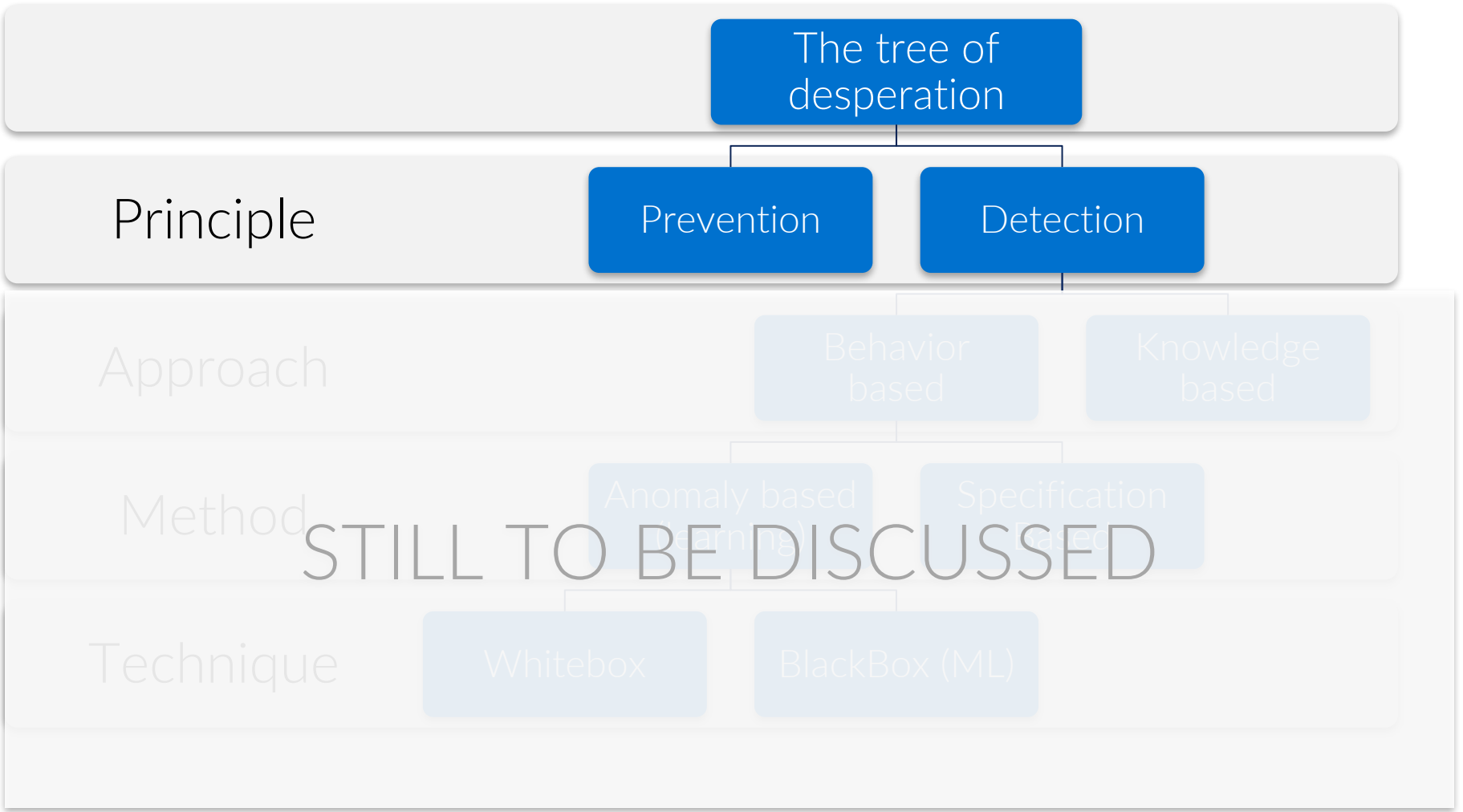
present systems are so hard to monitor

The source: Attackers

- Interesting types
  - Criminals (Cost < Benefit)
  - Hacktivists (Cost < fixed limit)
  - Nation states (no constraints)
  - Occasional (typically: insiders)

- Not everything hackable will be hacked, see e.g.
  - Where Do All The Attacks Go?, by Dinei Florencio and Cormac Herley https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/WhereDoAllTheAttacksGo.pdf

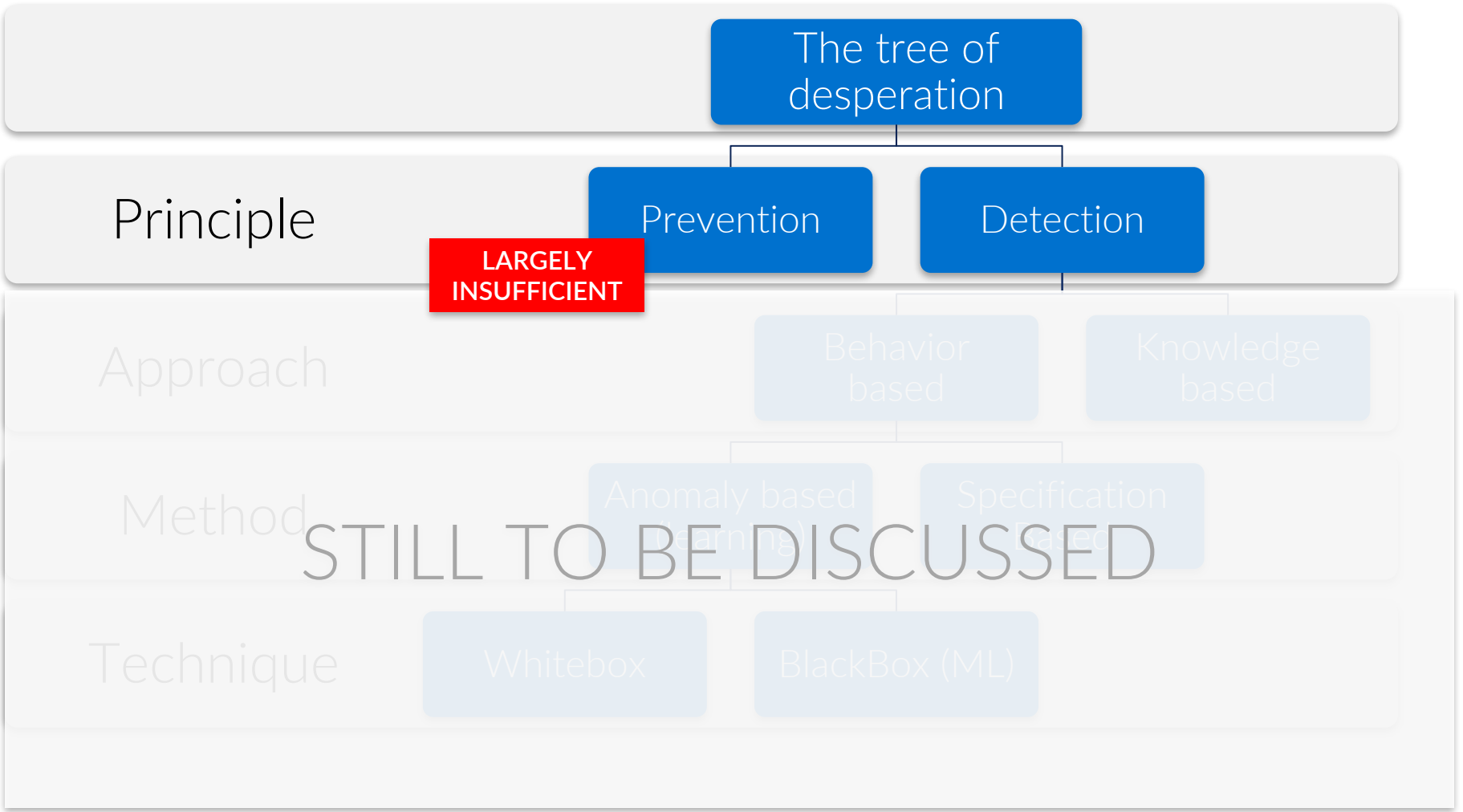
Two Ways of Dealing with Attacks



The Solution: Prevention?

- SW will never be 100% bug-free
- and even if it were 100% bug-free, it would be used in an insecure way
- and even if it were used in a secure way, something else will eventually spoil the system. There are too many connections
- And even then ….

The possibilities (in my opinion…)



So what is *Intrusion Detection?*

- **An area with a large gap between research and applications**

- **"despite extensive academic research one finds a striking gap in terms of actual deployments of such systems"**
  - Robin Sommer, Vern Paxson: Outside the Closed World: On Using Machine Learning for Network Intrusion Detection. S&P 2010
  - They are talking about machine-learning based IDS,

- **Evidence indicates that this is a general problem of IDS**
  - Why?
  - Next: the evaluation parameters of IDS

When do we have a GOOD IDS?

- Research papers look at (only) two parameters
  - Low **False Negatives** (high **detection rate)**: effectiveness
  - Low **False Positives** rate. High FP => High Usage Costs

- IMHO
  - Regarding the detection rate, papers usually indicate 90%+, but 50% detection rate would be more than sufficient, if *it was for real attacks (attacks are multistep anyhow)*
  - False positive rate is very important and my rule of thumb is that it should be < 0,01% to be viable.
  - BUT : these parameters are not enough to evaluate an IDS

When evaluating an IDS we should also look at:

- **Actionability**: how much information does the IDS give the user to prepare the response? No information => Very High Usage Costs

- **Adaptability**. Most IT systems change continuously (even SCADA systems, for that matter). The IDS operational costs are heavily affected by the cost of adapting it to the system changes.

- **Scalability.** How much does it cost to install and operate the IDS when deployed on 2, 200 or 2000 networks.

- IMHO:
  - lack on these fronts are the reason why "despite extensive academic research one finds a striking gap in terms of actual deployments of such systems"
  - Of course these parameters are difficult to evaluate in an academic setting
  - Did I mention it is a "horrible" research area?

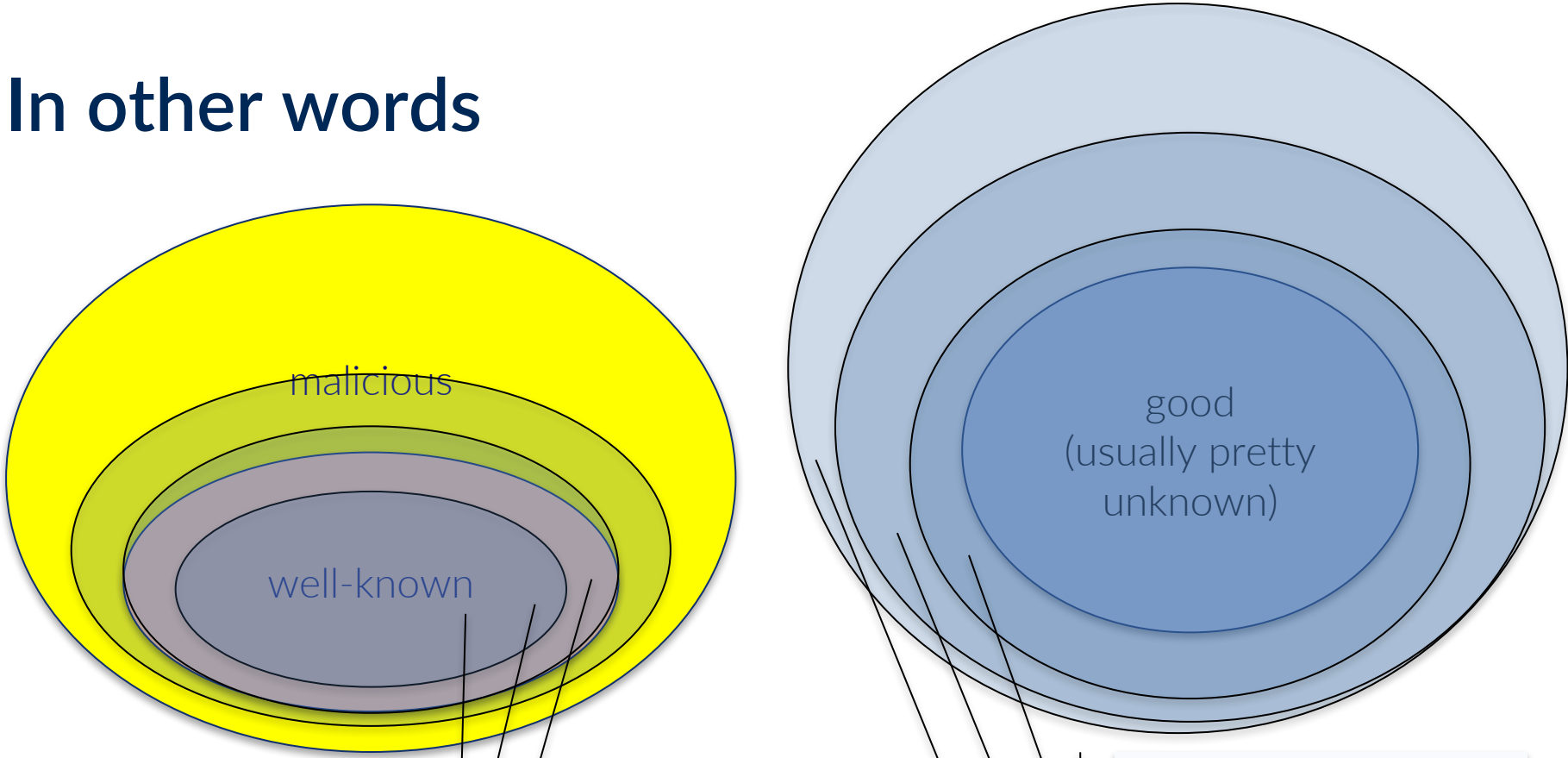
Let’s start digging into IDSs

How can you detect an attack.

- **Knowledge-Based**
  - Negative model aka blacklisting
  - You recognize the attack
  - Anti-viruses, Blacklisting, Signatures, etc…

- **Behavior Based**
  - Positive model: you recognize the normal behavior
  - what is not normal, is an attack, or in any case it is worth looking at
  - e.g. firewalls, whitelisting systems,

In other words



very specific signatures

less specific rules, emulation

generic experimental stuff

- the size of circles is arbitrary
- these are just examples

very specific whitelisting

generic whitelisting (e.g. WA firewalls)

anomaly detection

Let's take care of knowledge-based systems

- **They detect a fraction of the attacks**.
  - Too bad, because they score very well on the the other criteria
- For a lot of systems you don't have the knowledge
- … or it is not cost effective to process it
- Too easy to evade

They Named it — Einstein, But $6 Billion Firewall Fails to Detect 94% of Latest Threats

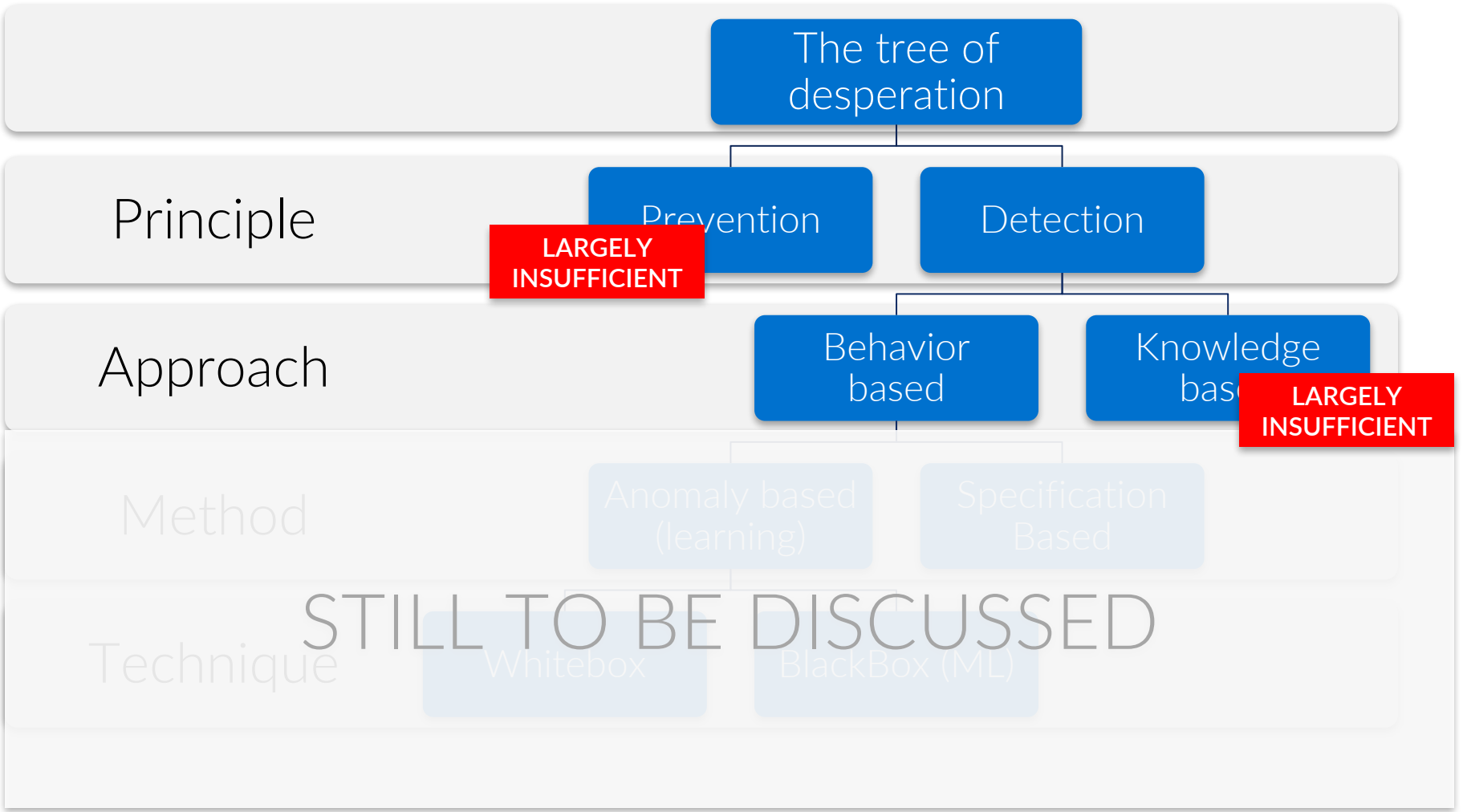
Tuesday, February 02, 2016 Swati Khandelwal

G+1 158 Like 1k Share 686 Tweet 259 Share 42 Share 1105

The US government's $6 Billion firewall is nothing but a big blunder.

Dubbed **EINSTEIN**, the nationwide firewall run by the US Department of Homeland Security (DHS) is not as smart as its name suggests.

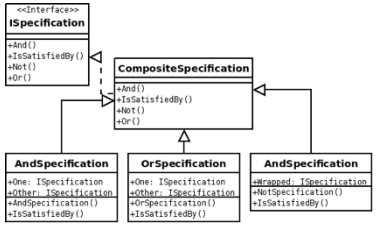
The possibilities (in my opinion…)



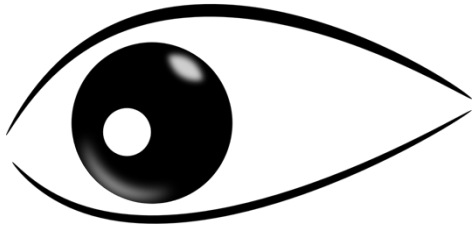
So we are left with behavior-based systems

- Where do we get the knowledge about the system?

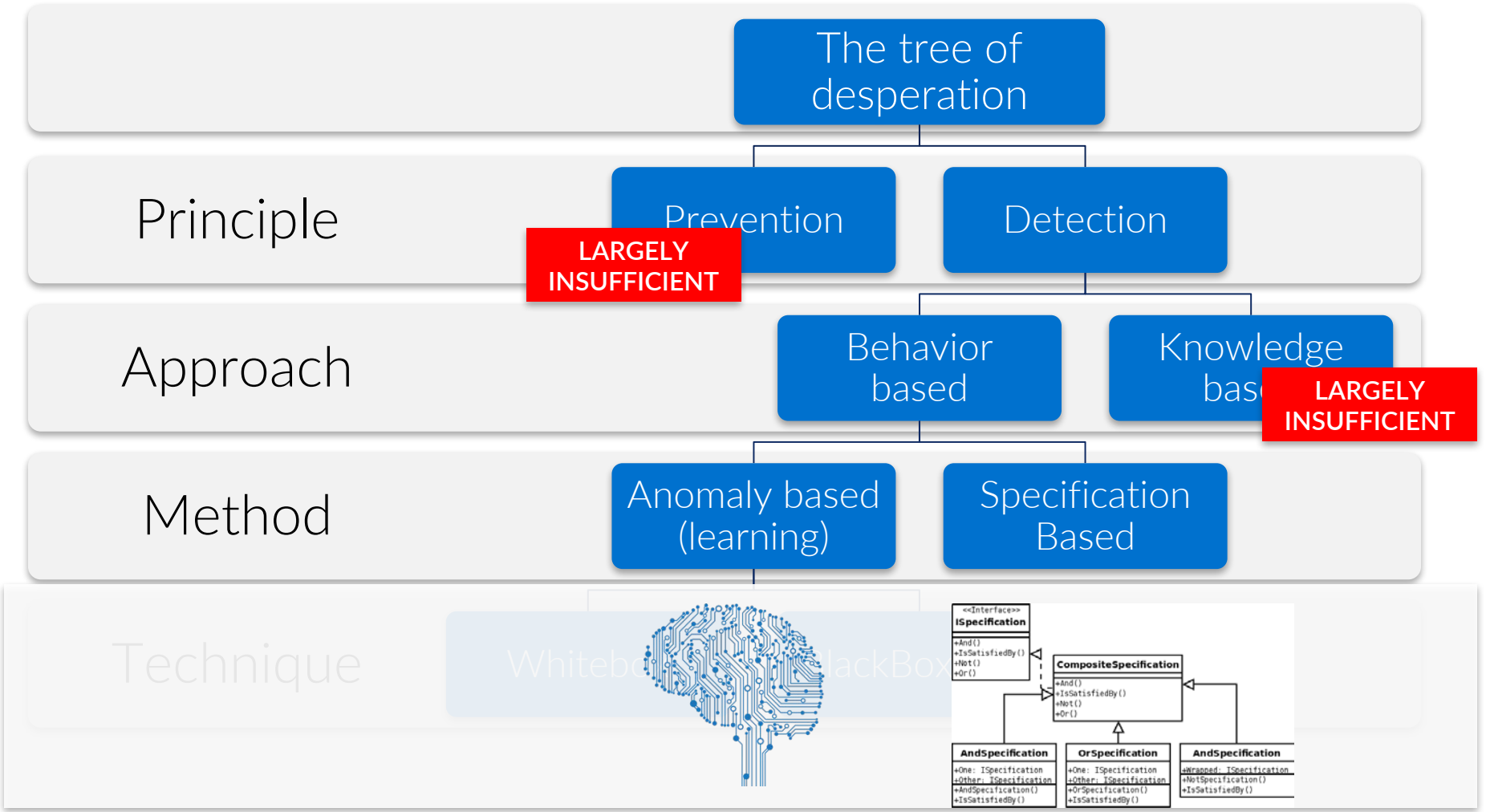
- From a specification,
  - (specification-based systems)



- We learn it automatically
  - (“anomaly-based systems”)



To build the model of the system, we have two options



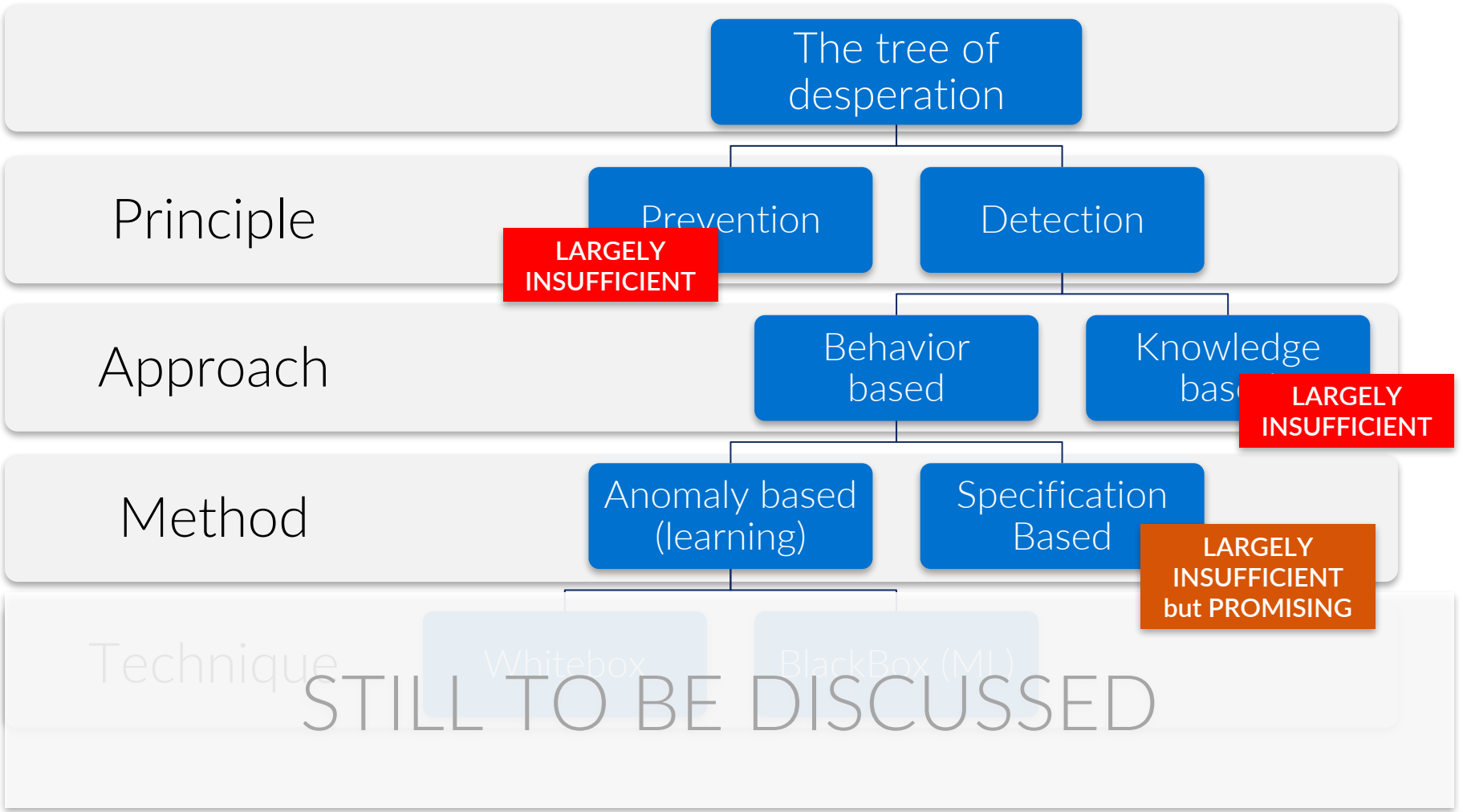
Specification-based systems are not the solution

- *This is all "in my opinion"*

- Two crucial features they do not satisfy "by definition"
  - **Adaptability**. Most IT systems change continuously (even SCADA systems, for that matter).
  - **Scalability**. How much does it cost to install and operate the IDS when deployed on 2, 200 or 2000 networks.

- Disclaimer
  - I love the principle of specification-based systems
  - I think it will become increasingly popular
  - But applied only to specific subparts of a system of systems (think of IoT….)

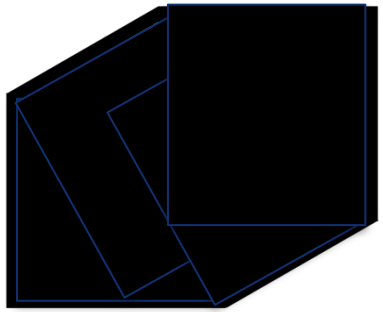
The possibilities (in my opinion…)



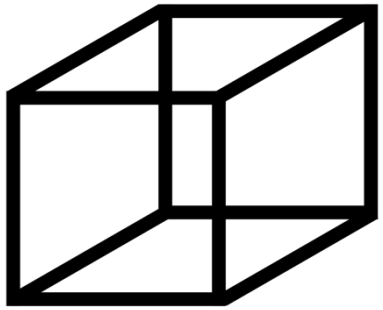
And now we are left with anomaly-based systems

- Another splitting, in two flavors:

  - **BlackBox**, using machine learning approaches, like neural networks.
    - The semantics used by the detection system is "unrelated" to the semantics of the target system



  - **WhiteBox**, in which we try to *explain* the semantics of the target system
    - The semantics used by the detection system is related to the semantics of the target system
    - Based on e.g. understanding the communication protocol, extracting command and setpoints and whitelisting them.

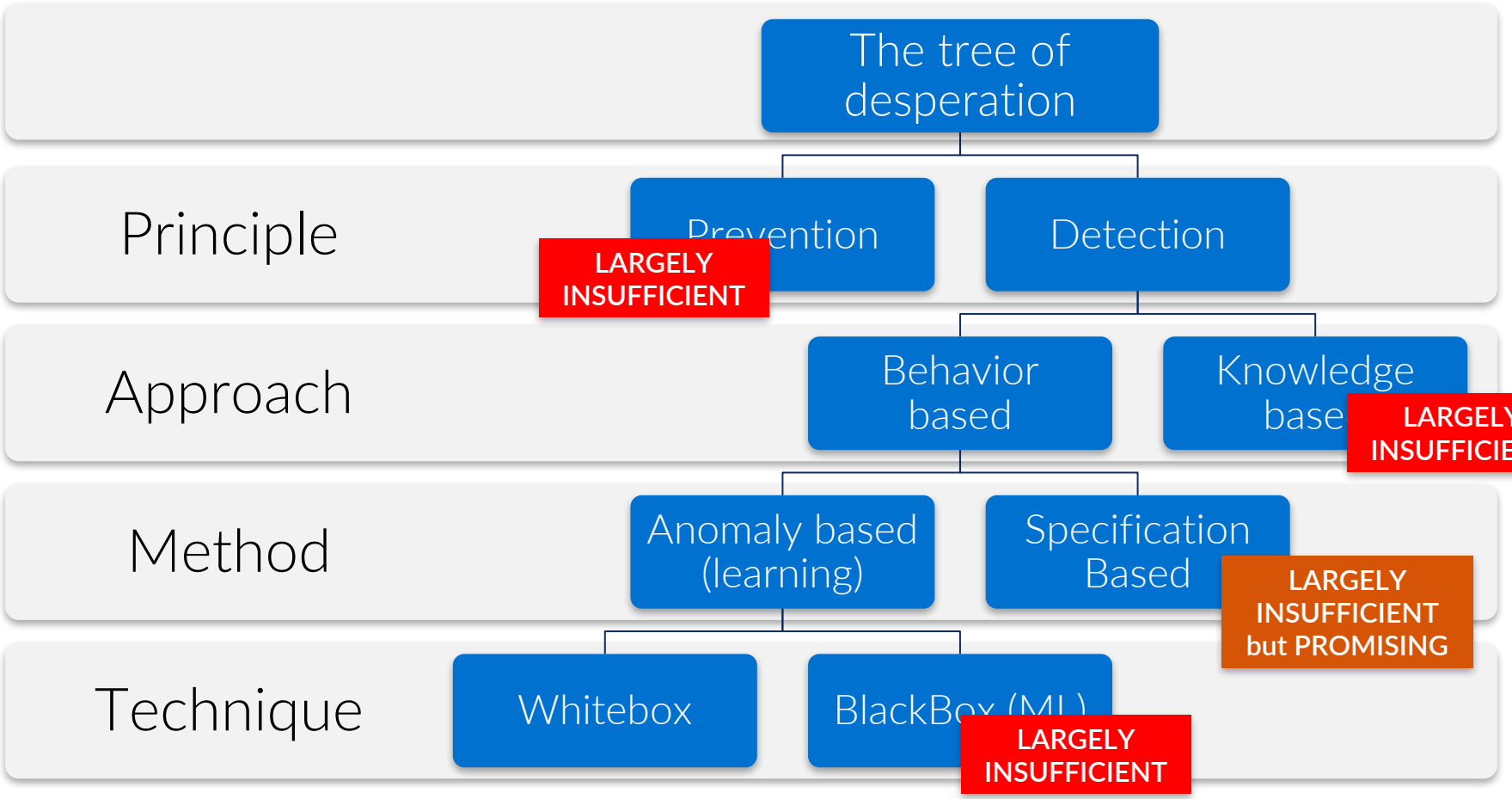


BlackBox Systems are not the solution

- **Personal Opinion 1**
- **I believe that blackbox anomaly-based intrusion detection systems are of very limited use for security.**
  - Actionability is the main problem
  - But also FPs…

- Sommer and Paxson (S&P 2010)
  - "we deem it crucial for any effective deployment to acquire deep, semantic insight … rather than treating the system as a black box as unfortunately often seen. "
  - "the better we understand the semantics of the detection process, the more operationally relevant the system will be."
  - [blackbox] anomaly detection systems face a key challenge of transferring their results into *actionable* reports …. In many studies, we observe a lack of this crucial final step.

The possibilities (in my opinion…)



Whitebox IDS should better be working

- **It works! But: on specific systems**
  - even on some large-scale systems.
  - very good usability results on SCADA/ICS
  - a solution for all problems? No
  - definition: there is not a one-size fits all.

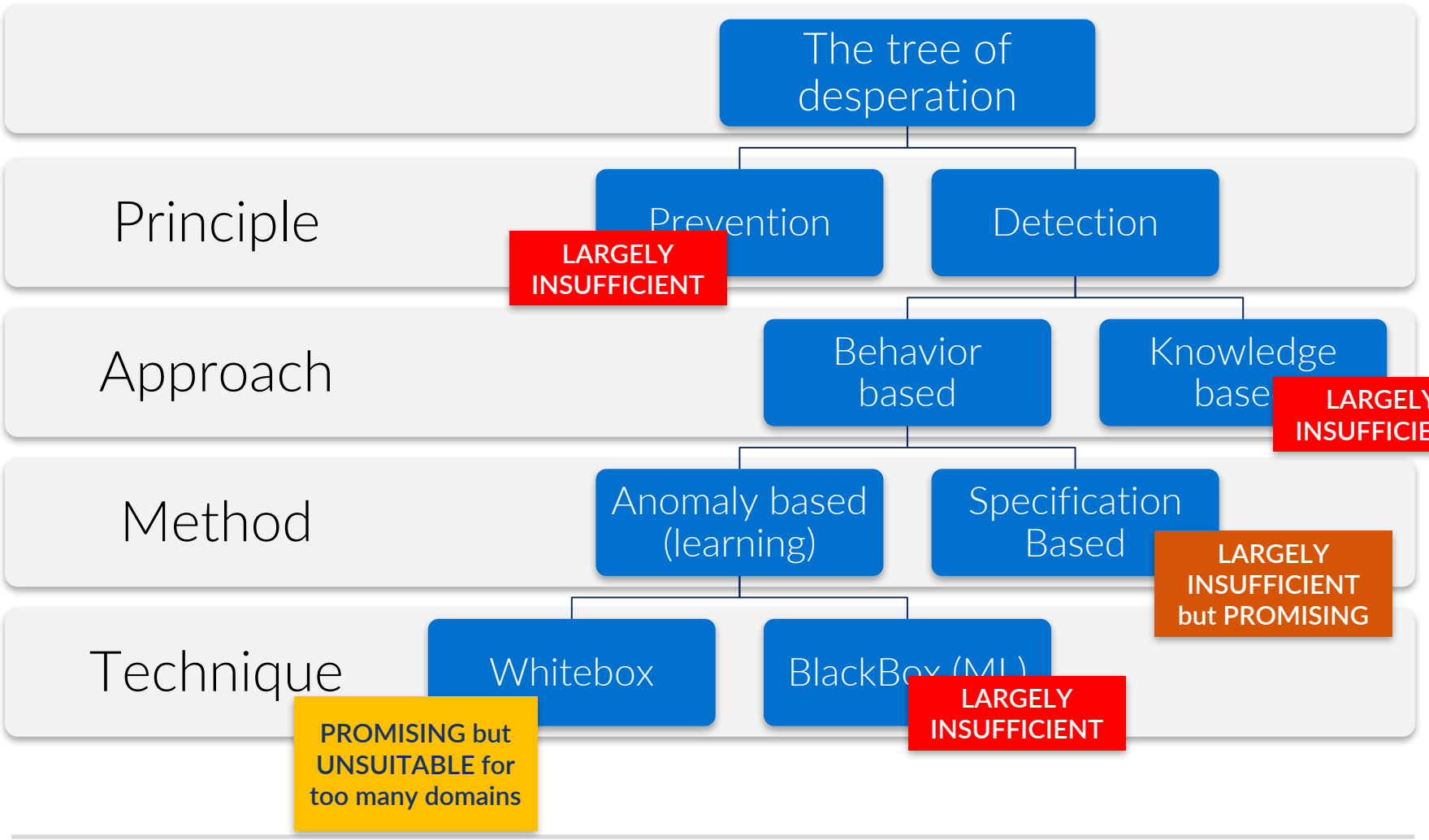
- **Personal Opinion 2**
- "Useful" anomaly-based intrusion detection **is not quite about intrusion detection; it is about being able to understand what happens in the target system** and being able to monitor its integrity.

Where Whitebox Anomaly Detection Fails

- ***most IT systems are simply not understandable***
  - Too complex, too dynamic too much of a mess.
  - Try to do anomaly detection on the first picture…

- **Personal Opinion 3**
- **There cannot be a one-size-fits-all anomaly-based network intrusion detection system** that works equally well on all domains.

WE GOT STUCK


The tree of desperation
Principle
Prevention
Detection
LARGELY INSUFFICIENT
Approach
Behavior based
Knowledge base
LARGELY INSUFFICIENT
Method
Anomaly based (learning)
Specification Based
LARGELY INSUFFICIENT but PROMISING
Technique
Whitebox
BlackBox (ML)
LARGELY INSUFFICIENT
PROMISING but UNSUITABLE for too many domains

What should we do?

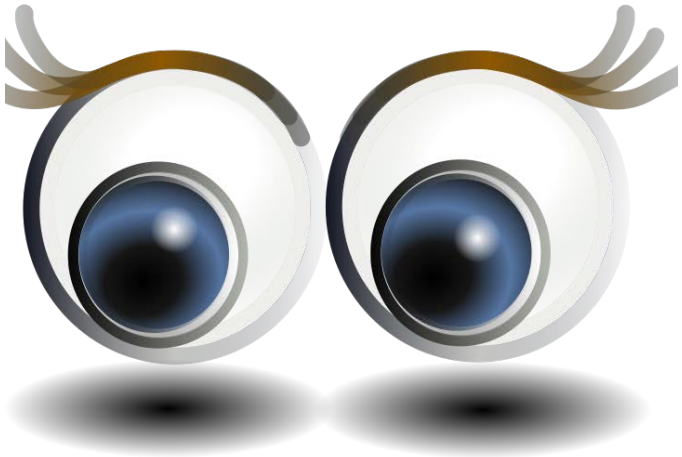
- Change the way we write software to make it more amenable to monitoring
- We have no other choice

- This is basically Personal Opinion 4

What is supervisable software?

- **The short answer: I don't know**

- The long answer is: I really really really don't know.



- SW allowing people who monitor it to understand what it is doing.
- It should be easier than writing secure software.

What about privacy?

- **Supervisability certainly does not help privacy.**
- a *very serious concern*.
  - There is a tendency to obfuscate the working of software to "guarantee privacy"
  - There is also the tendency to obfuscate the working of software to "guarantee security" – as if we hadn't done that mistake a million times already

- **Personal Opinion 5**
- **Trying to achieve privacy by making the software not supervisable is in my opinion (almost) as wrong as trying to achieve security by obscurity.**

Supervisable *and Privacy-Preserving*

- The obvious way is to separate
  - the observables regarding the working of the artifact, and
  - the private data

- This is not always possible: the working may reveal private information.

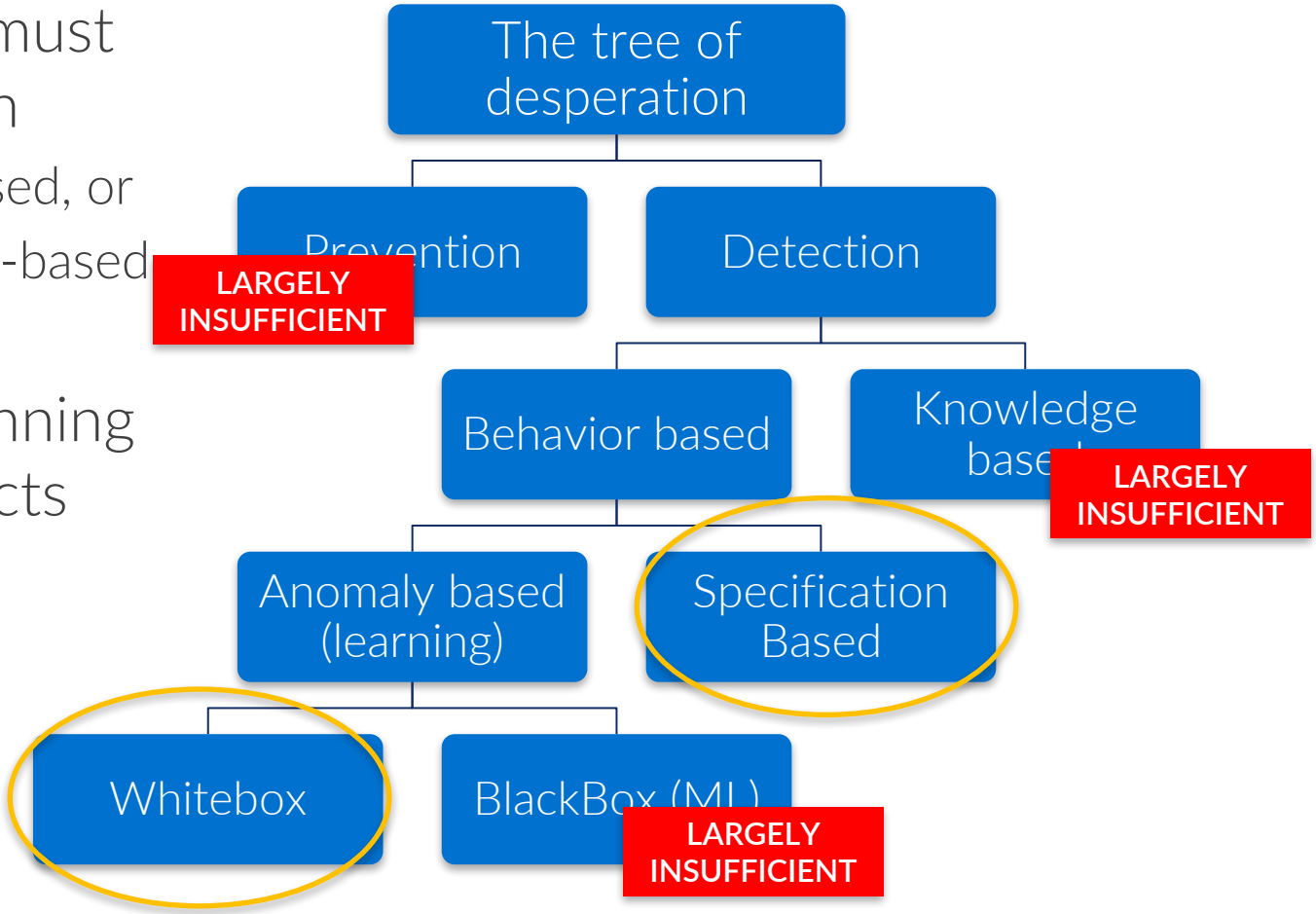
- However, consider
  - There are *many* sectors in which this is possible
  - There are many sectors in which we have lost that privacy anyhow
  - And there are many sector in which separating the working and the private data is not going to be possible.

The path to supervisability

- Supervisability
  - Could not find a precise definition
  - An art more than a science
- Writing supervisable SW: easier than writing secure SW
- There are fields (IoT) where this finds a natural application
- Unfortunately market forces do not help, I believe at the end of the day regulations will be necessary.

I believe there is no other way

- Software Eingineering must help detection
  - Anomaly-based, or
  - Specification-based
- The rest is running behind the facts



I believe that today the single most important reason why attacks are so difficult to counter is that

present systems are so hard to monitor

I believe the only practical way towards making more secure systems goes through

making software more supervisable

Questions?