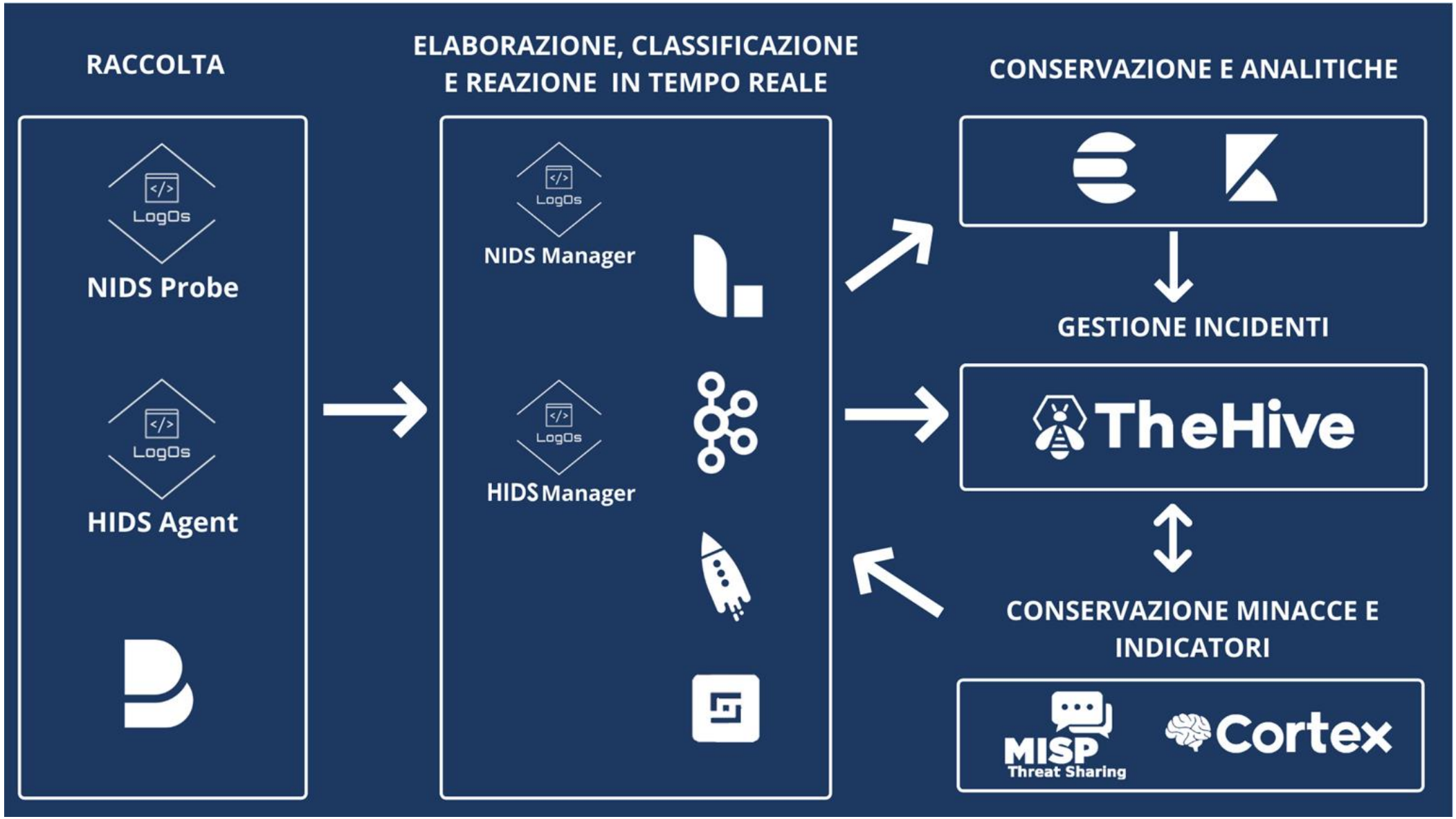
Machine Learning in LogOs

Enrico Polesel
System engineer, Seacom srl

#OSW2021

RIOS RETE ITALIANA OPEN SOURCE

LogOs


RACCOLTA
ELABORAZIONE, CLASSIFICAZIONE
E REAZIONE IN TEMPO REALE
CONSERVAZIONE E ANALITICHE
LogOs
NIDS Probe
LogOs
HIDS Agent
LogOs
NIDS Manager
LogOs
HIDS Manager
GESTIONE INCIDENTI
TheHive
CONSERVAZIONE MINACCE E
INDICATORI
MISP
Threat Sharing
Cortex

Eventi

Raccogliamo:

- log
- eventi applicativi
- modifiche dei file
- metriche
- risultati di scansioni e simili

Ma anche:

- dati di inventario
- dati di reputazione

Classificazione degli eventi

Dato **un evento** possiamo classificarlo in base:

- al significato semantico:
  - individuato malware
  - login fallito
  - login ok
  - aperta una nuova connessione TCP
- ai suoi campi “interni”:
  - ruolo del server
  - tipo di utente
- ai suoi campi “esterni”:
  - reputazione della sorgente

Classificazione degli eventi

RIOS RETE ITALIANA OPEN SOURCE

Guardare un singolo evento non basta!

Regole di correlazione, tipicamente sequenze di eventi:

- più di 5 eventi falliti
- più di 5 eventi falliti seguiti da un login OK
- connessioni verso porte non standard dopo un login
- più di 1000 query al database al secondo

Alcuni utilizzano **aggregazioni**

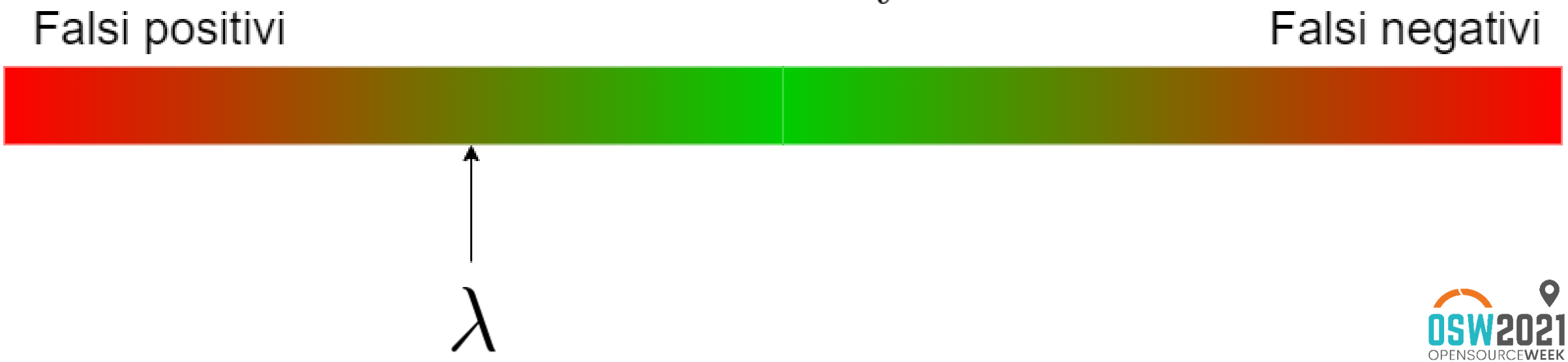
Molti sono basati su **soglie**

OSW2021 OPENSOURCEWEEK

Soglie

Molti sono basati su **soglie**

$$f(\{evento_i\}_i) > \lambda$$

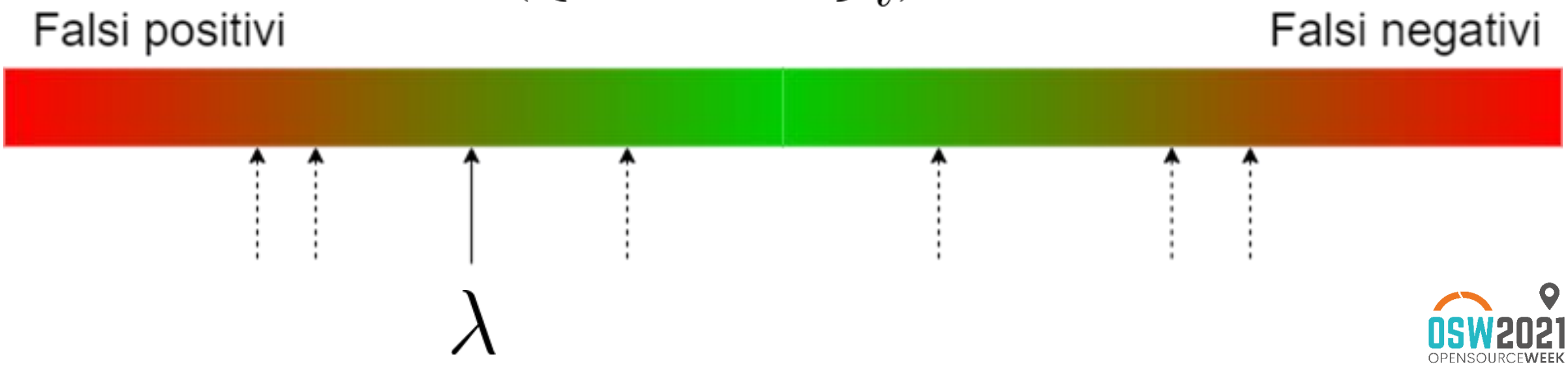


Soglie

Molti sono basati su **soglie**

Soglie che vanno **scelte** e **mantenute**

$$f(\{evento_i\}_i) > \lambda$$

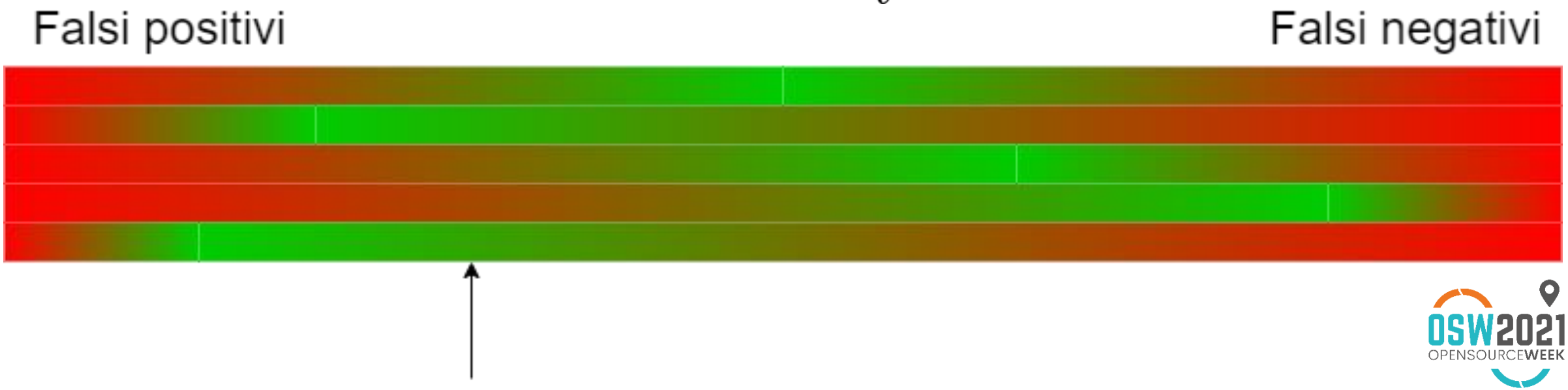


Soglie

Molti sono basati su **soglie**

La soglia ottimale potrebbe dipendere da molti fattori come il **tipo di server** o l'**orario dell'evento**

$$f(\{evento_i\}_i) > \lambda$$

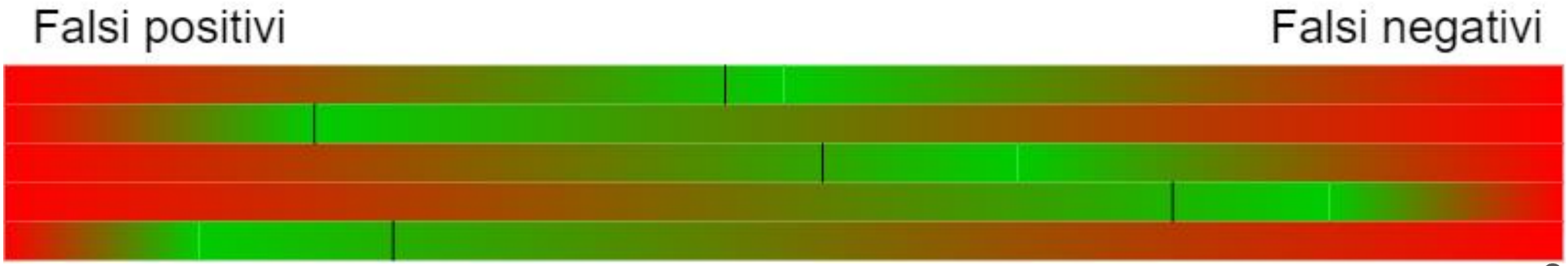


Soglie

Molti sono basati su **soglie**

La soglia ottimale potrebbe dipendere da molti fattori come il **tipo di server** o l'**orario dell'evento**

$$f(\{evento_i\}_i) > \lambda\,(time, server)$$

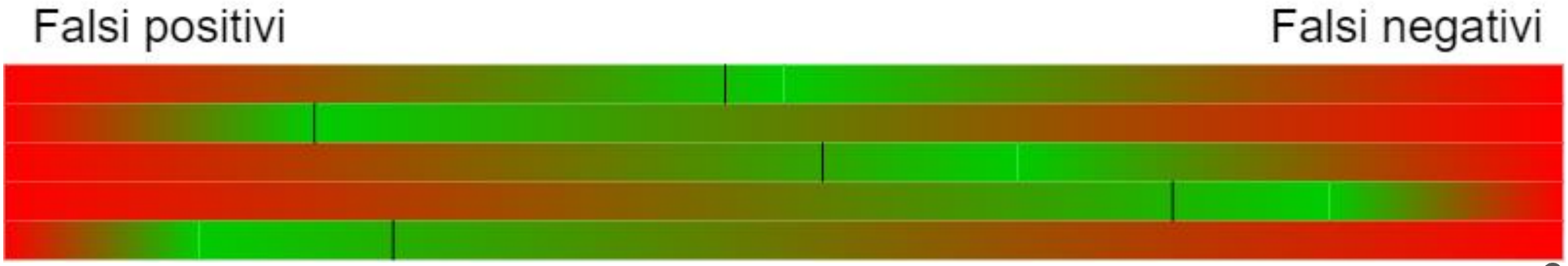


Soglie

Molti sono basati su **soglie**

Ma potrebbe non essere **mantenibile**.

$$f(\{evento_i\}_i) > \lambda\,(time, server)$$



Machine Learning

Vogliamo un sistema in grado di analizzare **automaticamente** il comportamento dei nostri sistemi (**machine learning**), creare un **modello** e avvertirci quando il comportamento non soddisfa tale modello (**anomaly detection**).

Con in machine learning definiamo:

- quale comportamento monitorare (e.g. numero login falliti)
- su quale entità aggregare i risultati (e.g. globale, oppure per utente)

In particolare non dobbiamo più definire manualmente le soglie e molte dipendenze (tra cui l'andamento temporale) viene riconosciuto automaticamente.

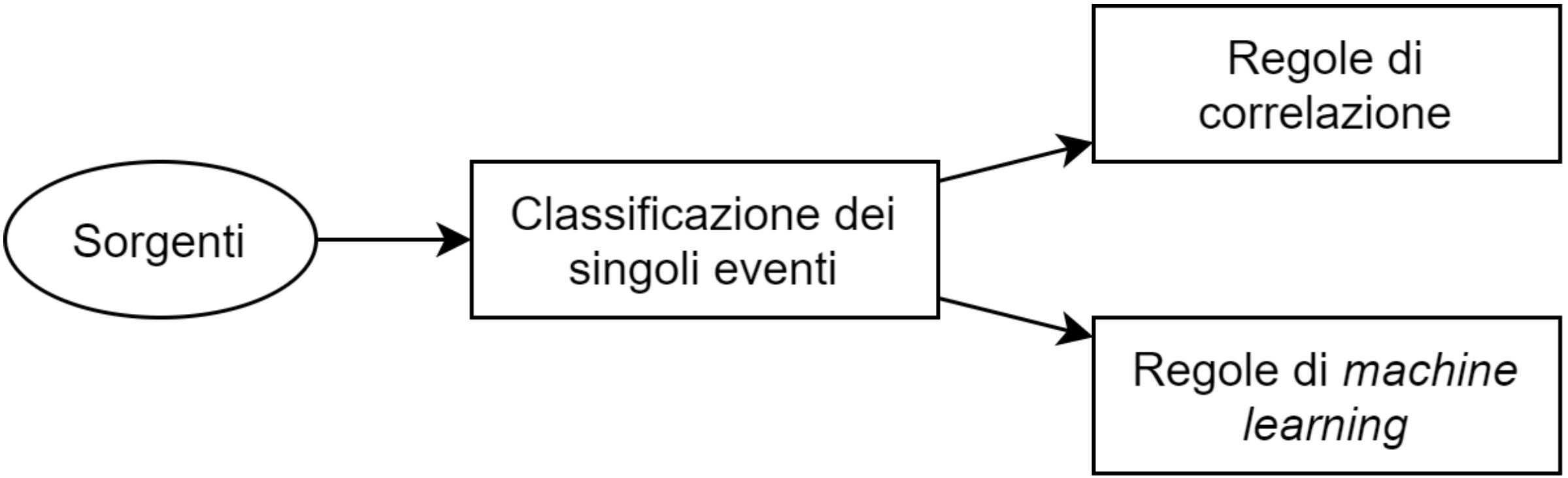
Machine Learning

Quando usiamo delle aggregazioni su un criterio (e.g. per server) possiamo cercare anomalie:

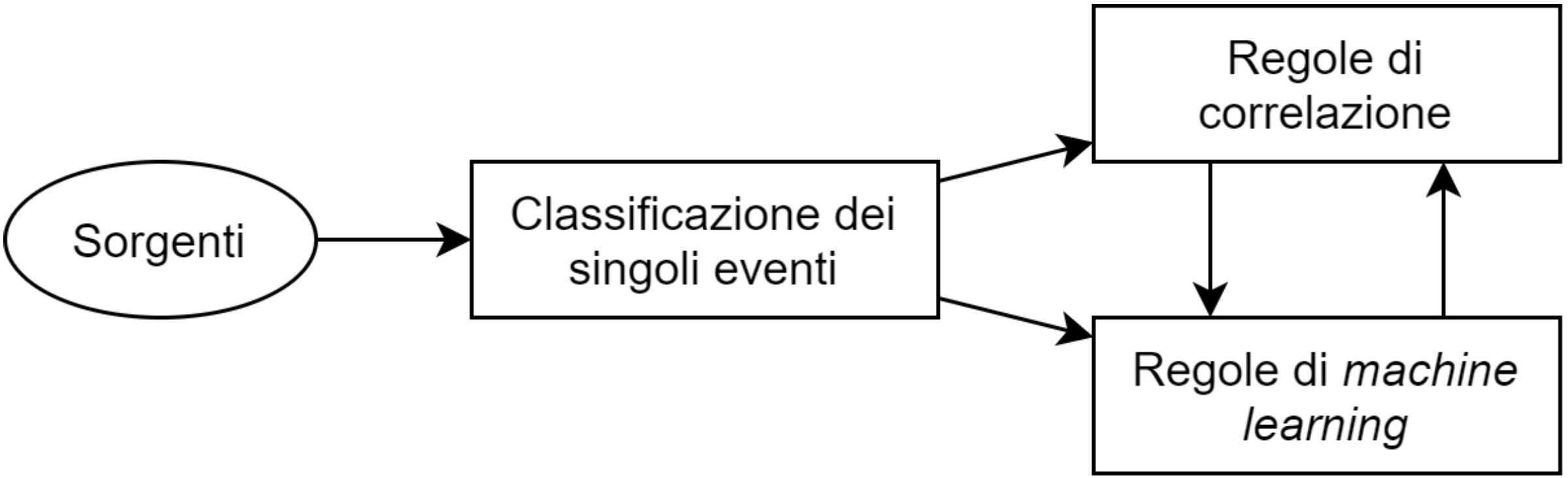
- anomalia rispetto alla storia di tale chiave (e.g. un server cambia il comportamento rispetto **al proprio passato**)
- anomalia rispetto alle altre chiavi (e.g. un server si comporta in **modo diverso dagli altri**)

Il risultato in generale non sarà un booleano (vero/falso) ma un livello di gravità (quanto il dato di discosta dal modello).

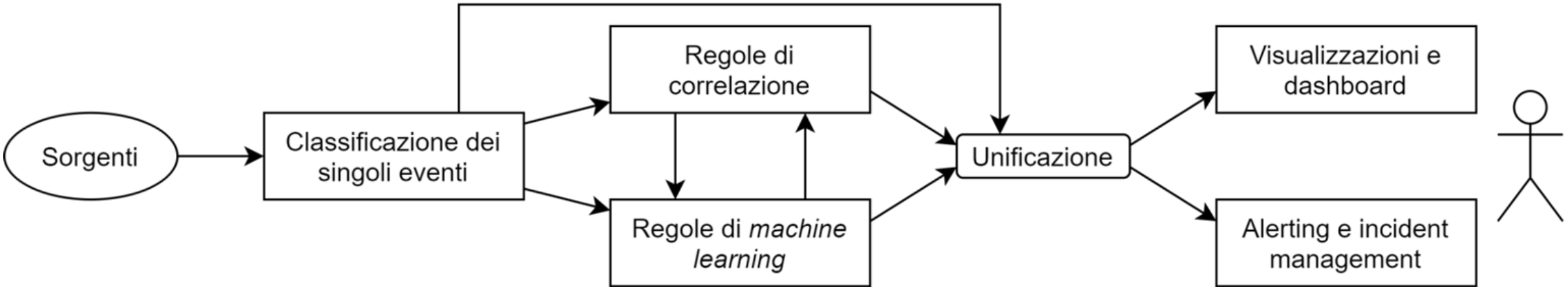
Flusso degli eventi



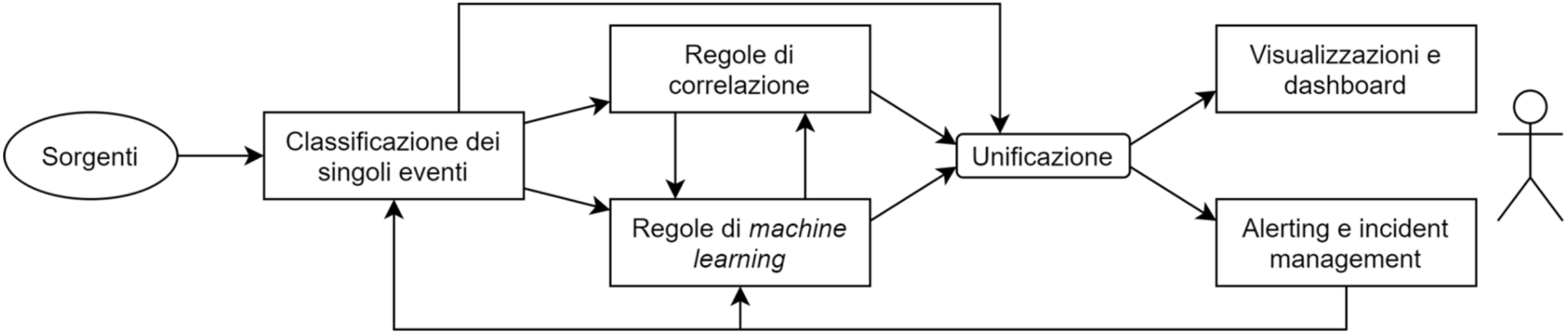
Flusso degli eventi



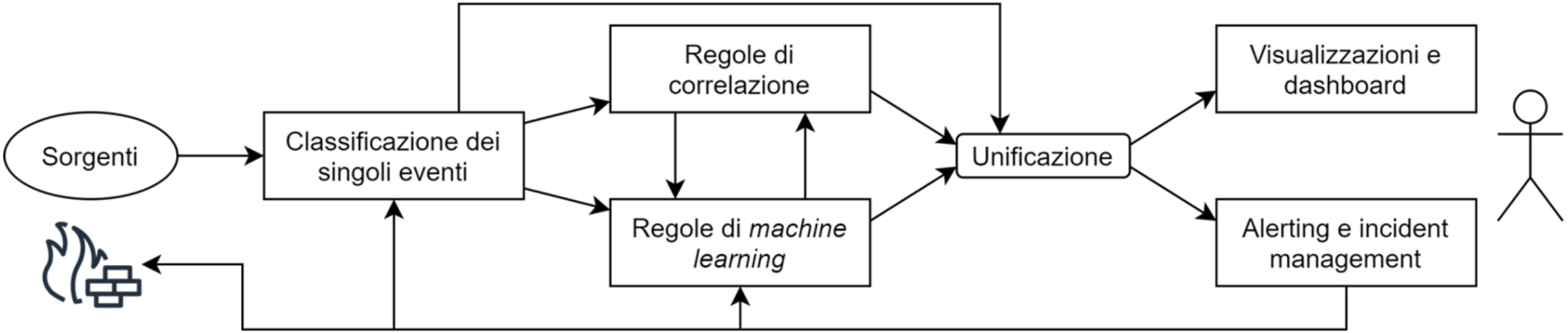
Flusso degli eventi



Flusso degli eventi



Flusso degli eventi



OSW2021
OPENSOURCEWEEK

Grazie per l'attenzione!