Forward-Looking Statements

This presentation may contain forward-looking statements regarding future events, plans or the expected financial performance of our company, including our expectations regarding our products, technology, strategy, customers, markets, acquisitions and investments. These statements reflect management's current expectations, estimates and assumptions based on the information currently available to us. These forward-looking statements are not guarantees of future performance and involve significant risks, uncertainties and other factors that may cause our actual results, performance or achievements to be materially different from results, performance or achievements expressed or implied by the forward-looking statements contained in this presentation.

For additional information about factors that could cause actual results to differ materially from those described in the forward-looking statements made in this presentation, please refer to our periodic reports and other filings with the SEC, including the risk factors identified in our most recent quarterly reports on Form 10-Q and annual reports on Form 10-K, copies of which may be obtained by visiting the Splunk Investor Relations website at www.investors.splunk.com or the SEC's website at www.sec.gov. The forward-looking statements made in this presentation are made as of the time and date of this presentation. If reviewed after the initial presentation, even if made available by us, on our website or otherwise, it may not contain current or accurate information. We disclaim any obligation to update or revise any forward-looking statement based on new information, future events or otherwise, except as required by applicable law.

In addition, any information about our roadmap outlines our general product direction and is subject to change at any time without notice. It is for informational purposes only and shall not be incorporated into any contract or other commitment. We undertake no obligation either to develop the features or functionalities described, in beta or in preview (used interchangeably), or to include any such feature or functionality in a future release.

splunk> .conf22

Muffle the |

OBS1251B

Peter Zumbrink

Sr. System Engineer | IKEA IT

Alexander Lindeskär

Sr. System Engineer | IKEA IT

Peter Zumbrink

Sr. System Engineer | IKEA IT

Alexander Lindeskär

Sr. System Engineer | IKEA IT

splunk> .conf22

IKEA® IT

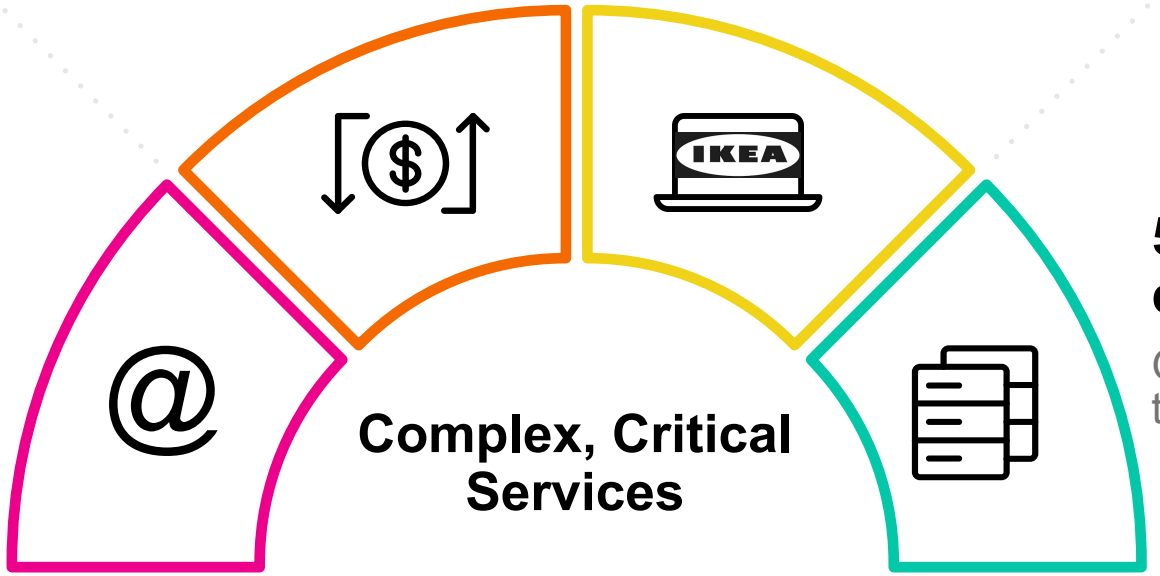
By the numbers

300 million transactions

13 check-outs per second

4 billion visits

to ikea.com



25,000 servers

100k+ personal computers, 200k co-workers worldwide

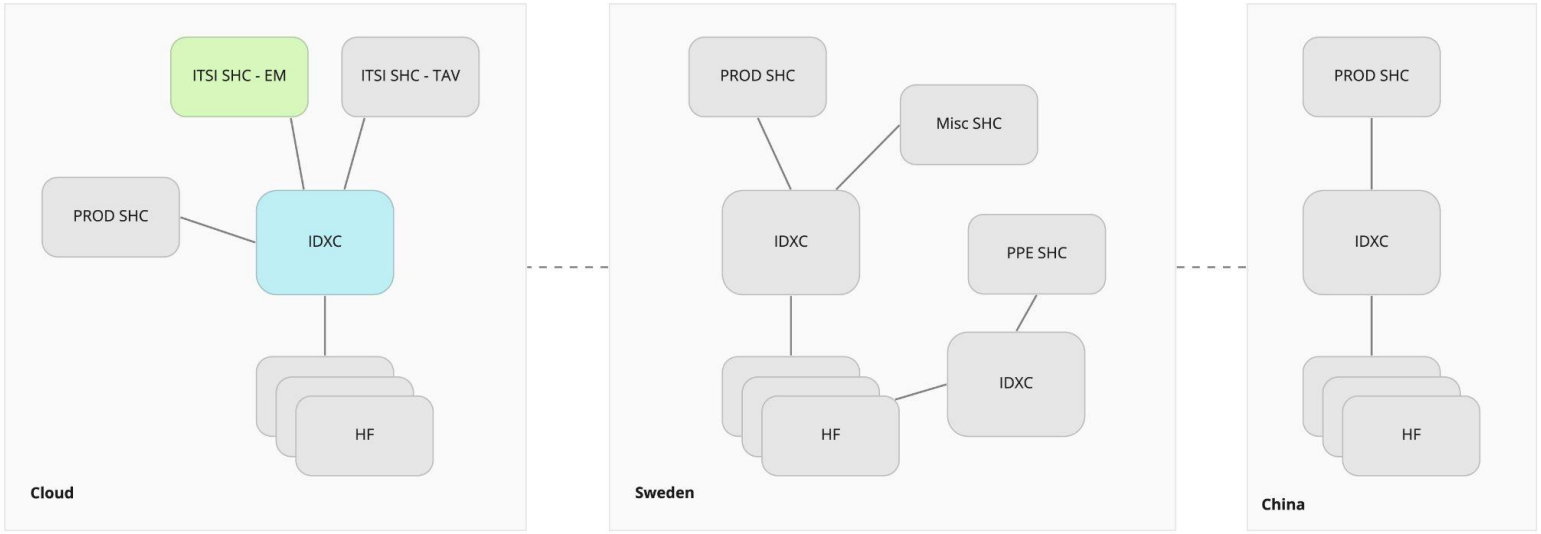
5 data centers + cloud regions

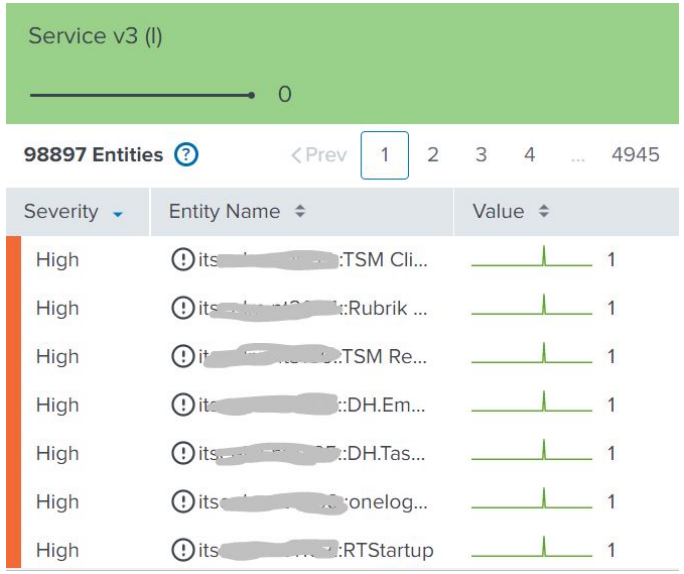
Green Data Centre of the Year 2014

The Volume in and Out the |

might explain the why

- ~20 terabytes data input
- ~300 million events searched by ITSI
- ~2500 episodes and incidents





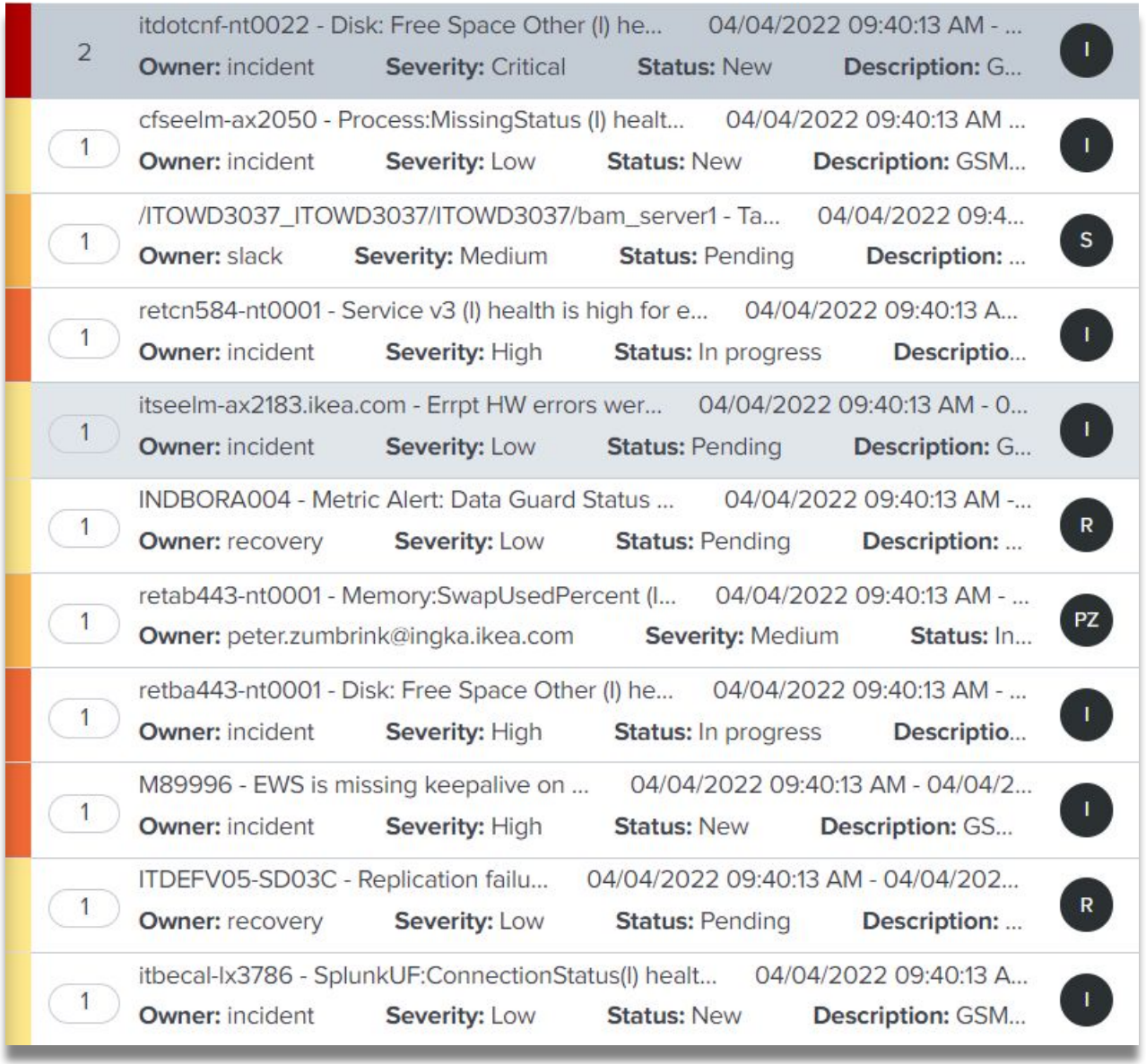
splunk> .conf22

Some Requirements

for Monitoring and Event Management with Splunk® ITSI

Wanted Position

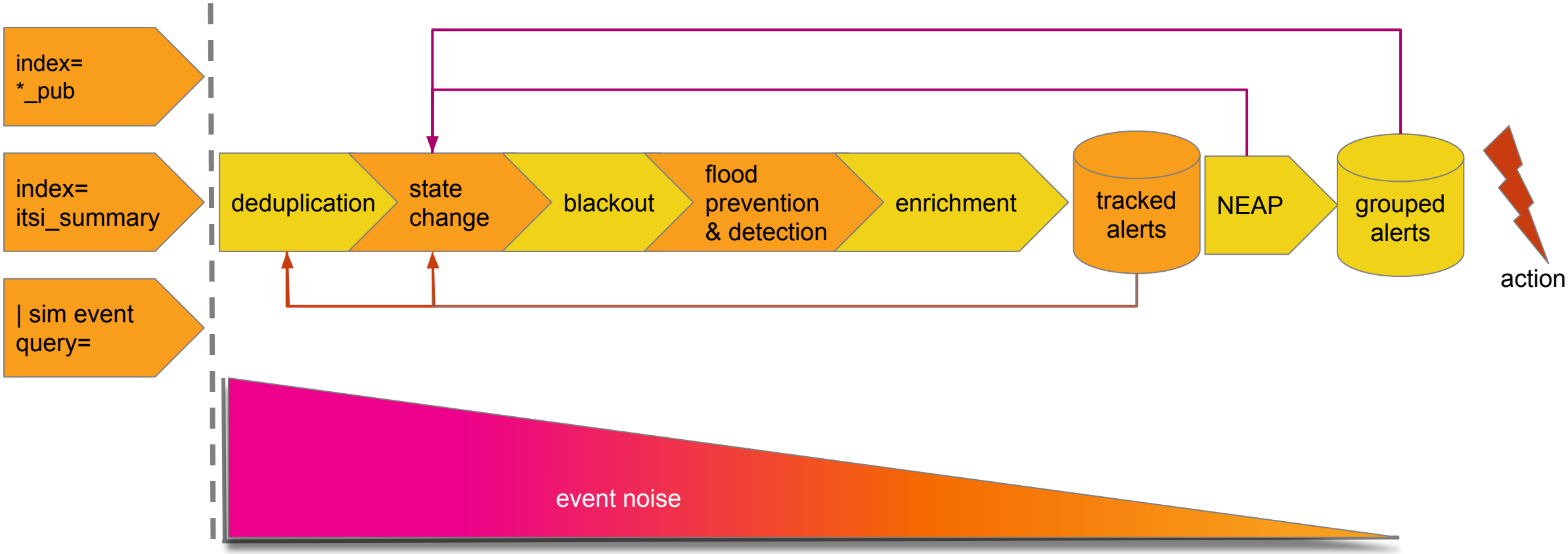
One event management solution for all the different sources automatically linked to an alert action with escalation mechanism and almost no false positives supporting agreed processes.



splunk> .conf22

At a Glance

the cool stuff we are going to talk about



Field Engineering

in a different way

field name	purpose
correlation_key	Drives grouping into episodes. This is the main “split events by” field.
event_key	Makes the events unique and is used for deduplication.
flood_key	Second stage correlation_key to prevent Episode flooding. Another “split events by” field.
action	Defines which Notable Event Aggregation Policy (NEAP) should pick up the Notable Event.
trigger	Triggers internal or external Alert Actions in the NEAP (e.g. recovery, slack, incident).

Harvesting Public Indexes

events from summary search

```
`itsi_not_summarized("test_pub","test")`
 | eval
  itsi_entity=source_hostname,
  itsi_correlation_key=source_hostname."~".inc_support_group,
  itsi_event_key=source_event_id,
  itsi_message=source_message_text,
  itsi_summary=source_event_name,
  itsi_impact="High",
  itsi_urgency="Medium",
  itsi_assignment=inc_support_group,
  itsi_business_service=source_service,
  itsi_eventtype=source_eventtype,
  itsi_tag=mvappend("NowIT", "ITSI")
`comment("# Use itsi_collect(3) macro to dump search result to _pub index")`
`itsi_collect("test_pub", "test", f)`
```

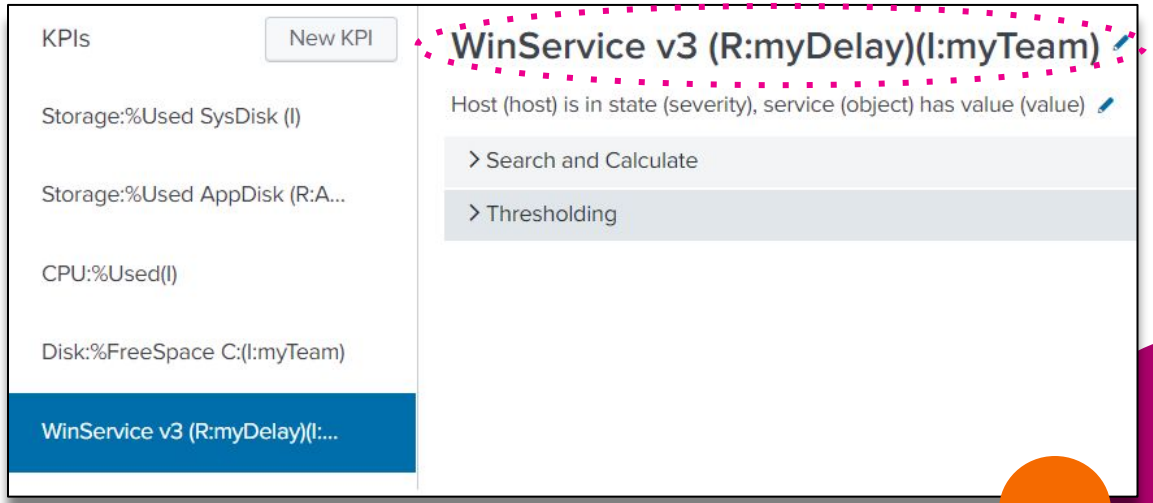
```
index=*_pub
(itsi_tag{}=ITSI AND itsi_tag{}=NowIT)
```

💡 **TIP!**
To connect events from summary searches to ITSI services just populate a field with the service_id and i.e. read service tags via rest call:

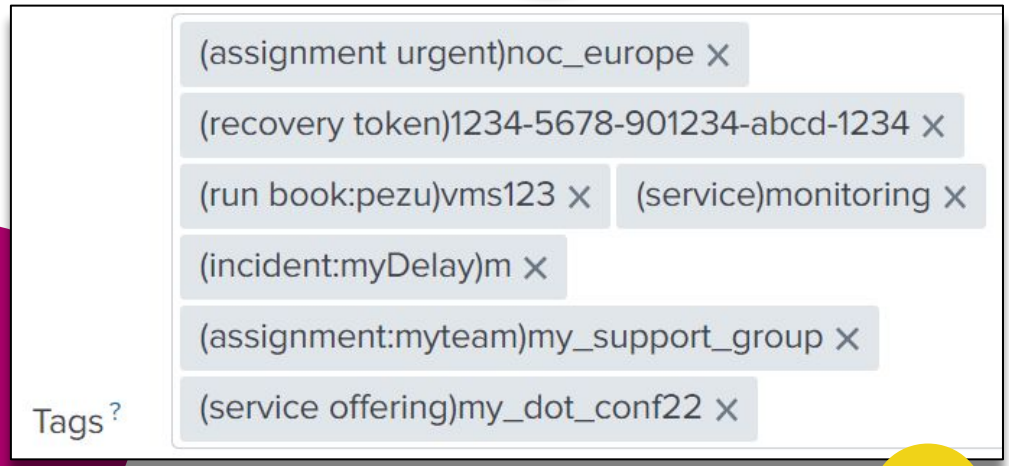
```
rest /servicesNS/nobody/SA-ITOA/
itoa_interface/service
```

The speciality with KPIs

and the stuff from Splunk® ITSI services




```
index="itsi_summary"
(indexed_is_service_aggregate::0
 indexed_is_service_max_severity_event::0)
AND
( kpi="*(I:*)*" OR kpi="*(R:*)*" OR
  kpi="*(I)*"   OR kpi="*(R)*")
```



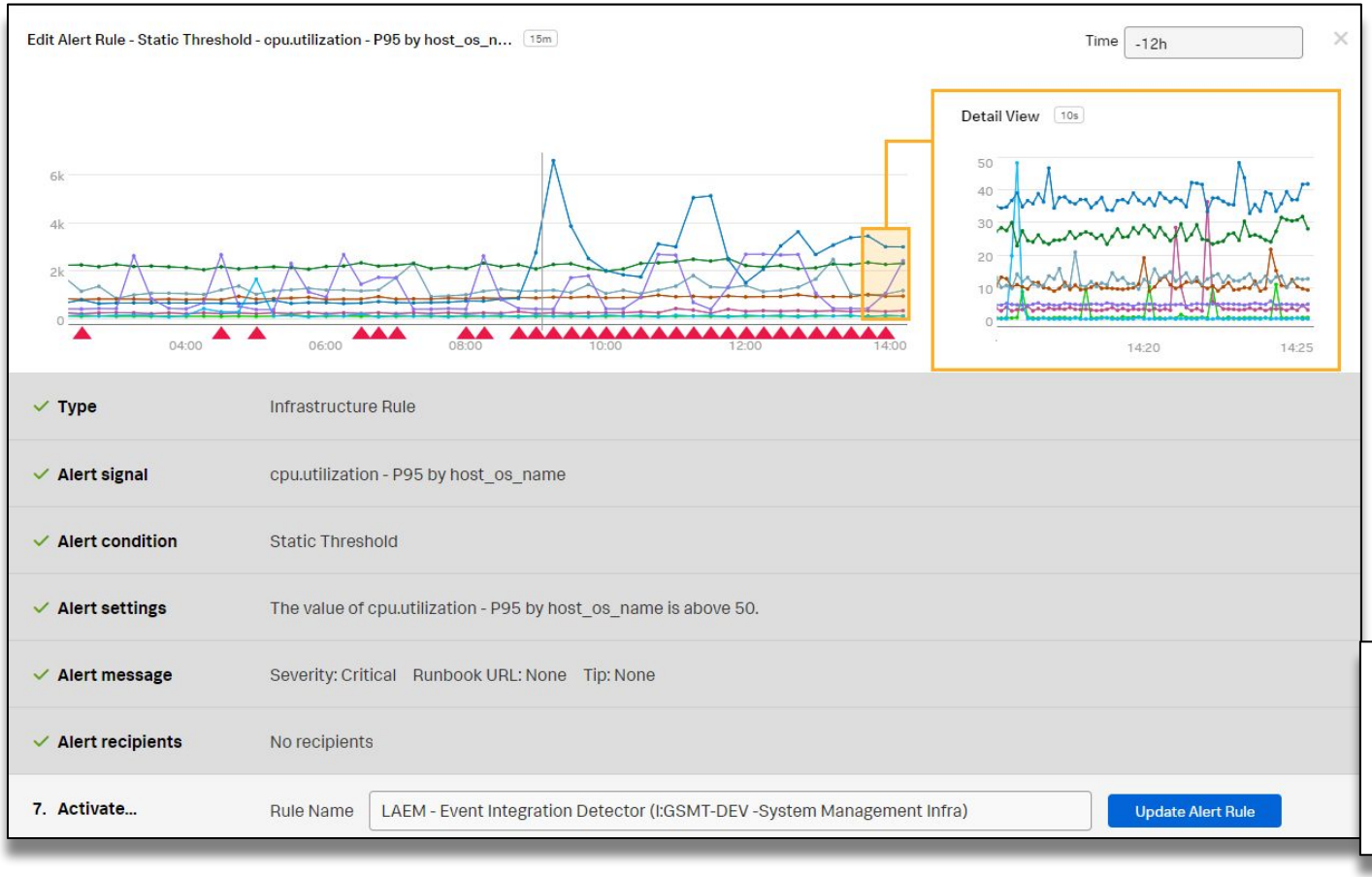

```
| rest splunk_server=local
"/servicesNS/nobody/SA-ITOA/
  itoa_interface/service?
  fields=title,description,service_tags,
_key,kpis._key,kpis.title,kpis.description"
```

Event
Deduplication
State change
Blackout
Flood prevention and detection
Enrichment
Aggregation

splunk> .conf22

What about detectors

or the integration with Splunk® Infrastructure Monitoring (Sfx)



```
| sim event query="
sf_notificationString:*(I:*)*
AND
sf_eventCategory: ALERT"
```

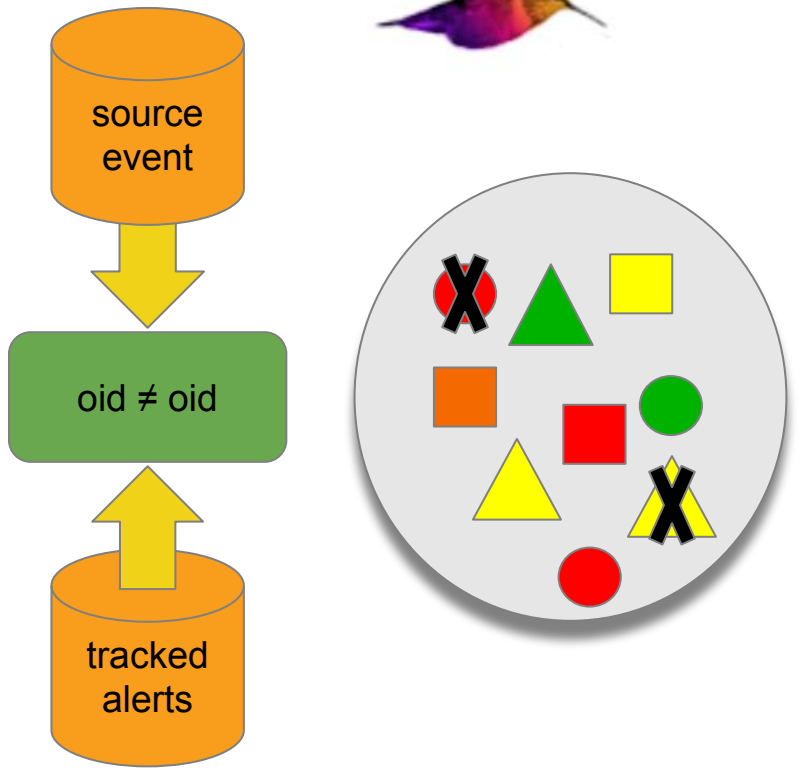
splunk> .conf22

Event
Deduplication
State change
Blackout
Flood prevention and detection
Enrichment
Aggregation

No duplicates please

it is all about the news

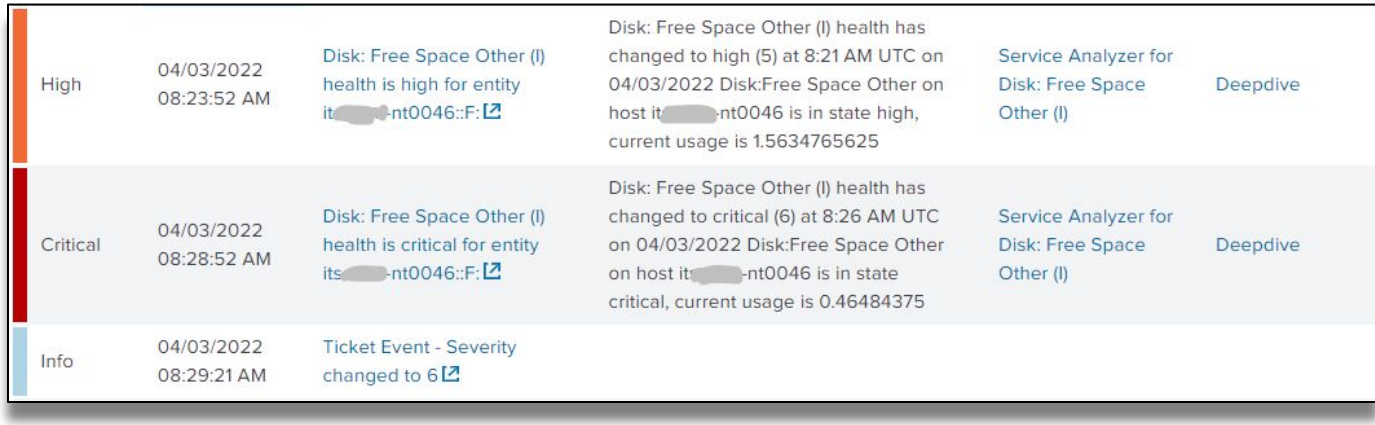
```
| eval oid="C:"._time.correlation_key
| append [|search index=itsi_tracked_alerts oid="C:*" earliest=-10m@m
          | stats delim=";"
              latest(_time) as _time
              latest(severity) as ta_severity
              latest(event_key) as ta_event_key
              values(oid) as oid_list
              by correlation_key
         ]
| eval unique_key=event_key."~".severity
| stats latest(_time) as _time latest(*) as * by unique_key
| where isnotnull(oid)

| makemv delim=";" oid_list
| eval oid_found=if(in(oid,oid_list),"true","false")
| where oid_found="false" AND severity!=ta_severity
```



State change

- Severity Normal not to create new Episode
- New Notable Event (NE) only if severity or event key differs from previous one
- Severity Normal to resolve Episode
- Severity higher than Episodes severity to escalate Episode

High	04/05/2022 11:03:59 AM	Disk: Free Space Other (I) health is high for entity ret[redacted]-nt0001::D:	Disk: Free Space Other (I) health has changed to high (5) at 11:01 AM UTC on 04/05/2022 Disk:Free Space Other on host ret[redacted]nt0001 is in state high, current usage is 1.662109375	Service Analyzer for Disk: Free Space Other (I)	Deepdive
Normal	04/05/2022 11:18:56 AM	Disk: Free Space Other (I) health is normal for entity ret[redacted]-nt0001::D:	Disk: Free Space Other (I) health has changed to normal (2) at 11:16 AM UTC on 04/05/2022 Disk:Free Space Other on host ret[redacted]-nt0001 is in state normal, current usage is 2.935546875	Service Analyzer for Disk: Free Space Other (I)	Deepdive
Info	04/05/2022 11:19:23 AM	Ticket Event - Resolved			



High	04/03/2022 08:23:52 AM	Disk: Free Space Other (I) health is high for entity it[redacted]-nt0046::F:	Disk: Free Space Other (I) health has changed to high (5) at 8:21 AM UTC on 04/03/2022 Disk:Free Space Other on host it[redacted]-nt0046 is in state high, current usage is 1.5634765625	Service Analyzer for Disk: Free Space Other (I)	Deepdive
Critical	04/03/2022 08:28:52 AM	Disk: Free Space Other (I) health is critical for entity its[redacted]-nt0046::F:	Disk: Free Space Other (I) health has changed to critical (6) at 8:26 AM UTC on 04/03/2022 Disk:Free Space Other on host it[redacted]-nt0046 is in state critical, current usage is 0.46484375	Service Analyzer for Disk: Free Space Other (I)	Deepdive
Info	04/03/2022 08:29:21 AM	Ticket Event - Severity changed to 6			

State changes

is of interest, the rest is just noise

The <<split_by_hash>> field

split event by field name	:	split event by field value	:

```
| eval hash="correlation_key:".correlation_key.":"
| join type=left hash [ |search index=itsi_grouped_alerts_em source="this_search" earliest=-15d
   | stats latest(severity) as ga_severity by itsi_group_id
   | lookup itsi_notable_group_system_lookup _key as itsi_group_id | where is_active=1
   | rename split_by_hash as hash start_time as earliest_event_time
   | fields itsi_group_id ga_severity hash earliest_event_time
   ]
| lookup itsi_notable_group_user_lookup _key as itsi_group_id OUTPUT mod_time status as
ep_status
| where (ep_status=5 AND severity>2)
     OR (ep_status<5 AND severity!=ga_severity)
```

Simple blackout

noise cancelling on the fly

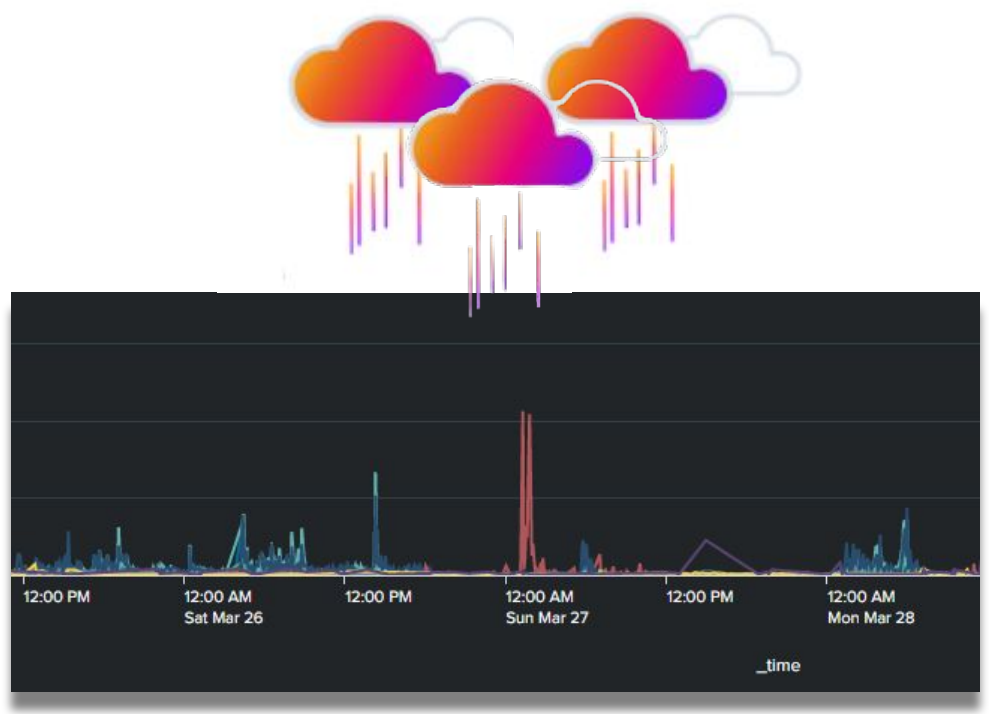
```
curl --silent --insecure --show-error \
--header 'Authorization: Splunk 1234567-abcd-1234-cmyk-my_pub_secret' \
--data '{"event":{"resource": "myhost", "duration": 300}}' \
https://splunk.foo.net:443/services/collector
```

```
| makeresults
| eval
  resource="myhost", duration=300,
  comment="Sample blackout for .conf22"
| collect index="my_pub" source="http:event"
  sourcetype="blackout:external" testmode=f
```

```
| eval resource=entity
| join type=left resource [
  search index="my_pub"
         sourcetype="blackout:external"
         earliest=-24h
  | eval starttime=_time
  | stats latest(*) as *
  | where starttime+duration>now()
  | eval blackout="true"
  | fields resource blackout
]
| where NOT blackout="true"
```

Flood detection & prevention

- Detect ITSI Service KPIs or other events about to create too many unique Episodes
- Group events with same receiver into a dedicated Episode
- Allow users to set own flood key and flood threshold
- Automatically go back to normal event flow after a given time, e.g. create flood Episodes with a short lifetime



Flood detection & prevention

because you don't want to be hated

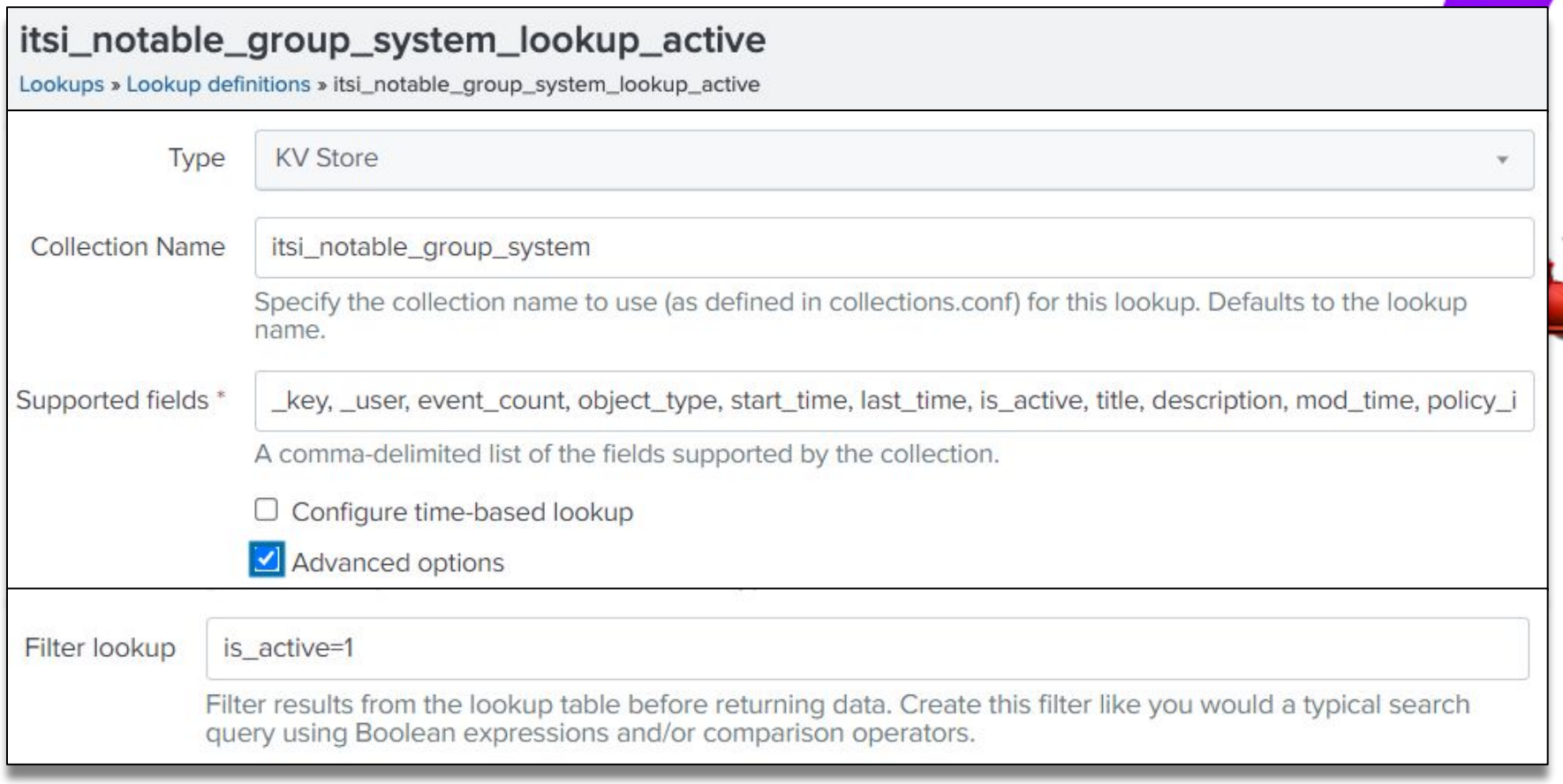
```
| eval threshold=coalesce(user_threshold,50)
| eventstats count(flood_key)
    as fc by flood_key
| eventstats count(correlation_key)
    as cc by correlation_key
| eval value=if(fc>0,fc/cc,0)
| eval action=if(value>threshold,"flood",action)
```

💡 **TIP!**
Find a suitable key for the flood detection calculation, i.e. ITSI service name plus kpi name or the receiver name such as support group

```
| eval flood="false"
| eval hash="flood_key:".flood_key.":"
| lookup itsi_notable_group_system_lookup_active split_by_hash as hash output _key as flood_id
| eval flood    =if(isnotnull(flood_id),"true"   ,"false")
| eval flood    =if(   isnull(group_id),flood    ,"N/A"  )
| eval flood_key=if(   isnull(group_id),flood_key,null() )
| eval action   =if(flood="true","flood",action)
| eval group_id =coalesce(group_id,flood_id,null())
```

Side note

maybe this is worth to be mentioned

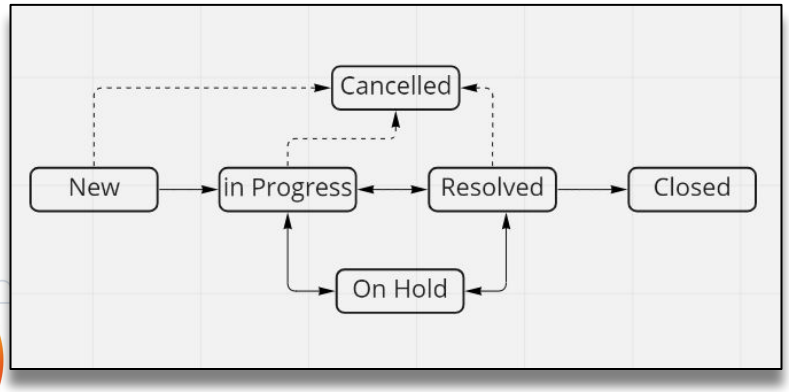


Other enrichment & filtering

from our action endpoints

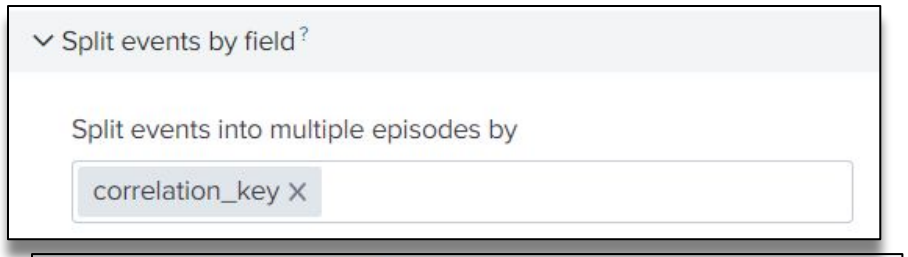
```
| eval episodeid=itsi_group_id
| join type=left episodeid [
    | search index=snow_pub sourcetype=snow:incident
      dv_x_splu2_splunk_ser_splunk_url="*event.com*"
      endpoint="incident.com"
      episodeid=*
      earliest=-30m@m
    | stats values(dv_incident_state) as state
      by episodeid
  ]
| where NOT state="Resolved"
```

Example:
Have the incident process steering your notable event flow.
This is used to apply a KPI grace period when an incident was resolved recently.

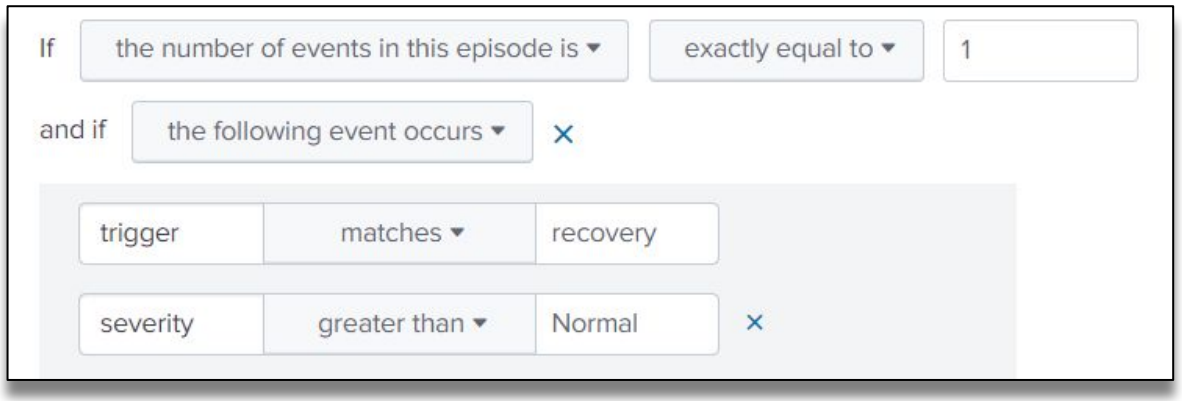


The aggregator

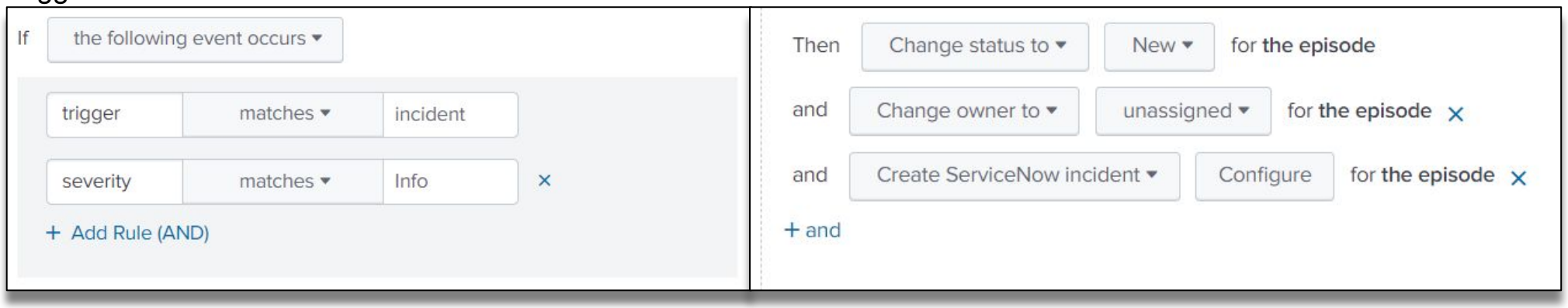
or what is named the Notable Event Aggregation Policy (NEAP)



Trigger recovery



Trigger incident



splunk> .conf22

Special use case correlation

for the things behind the scenes

```
…
| join type=inner itsi_group_id
    [| search index=_internal host=sh*em* sourcetype=ta_snow_ticket
    | rex "^(\d{4}-\d{2}-\d{2} \d{2}:\d{2}:\d{2},\d{1,3}) (?<severity>\w+)"
    …
    | table itsi_group_id ta_snow_error_msg ]
| search ta_snow_error_msg IN ("*502*", "*socket.timeout*", "*Return*429*", "*Re
…
| eval title="Tick
| eval instructior
| eval incident_r
```

```
| inputlookup itsi_notable_group_system_lookup_active
| rename _key as itsi_group_id

| lookup itsi_notable_group_user_lookup _key as itsi_group_id .
| where ((600 + mod_time) < now())
| where (status == `STAT_RESOLVED`)
…
| eval
    break="1",
    ep_title="EpisodeCloseResolved",
    …
| episodeupdate group=itsi_group_id policy=policy_id …
| where 1=2
```

```
`itsi_event_management_group_index` ...
...
| eval alive_time=now()-start_time
...
| eval alive_threshold=case(
    incident_delay="xs",360,
    incident_delay="s",660,
    incident_delay="m",960,
    incident_delay="l",1860,
    incident_delay="xl",3660,
    true(),360
)
...
| where alive_time >= alive_threshold
...
```

splunk> .conf22

Muffle the |

Average noise reduction compiled over 24 hours

splunk> .conf22

Thank You

splunk> .conf22