

SMARTE ZABBIX TEMPLATES

AM BEISPIEL: KUBERNETES MONITORING IN ZABBIX 6

CHRISTIAN ANTON

CLOUD NATIVE INFRASTRUCTURE & SERVICES, SECADM GMBH

DRIVEN BY DEDICATION.

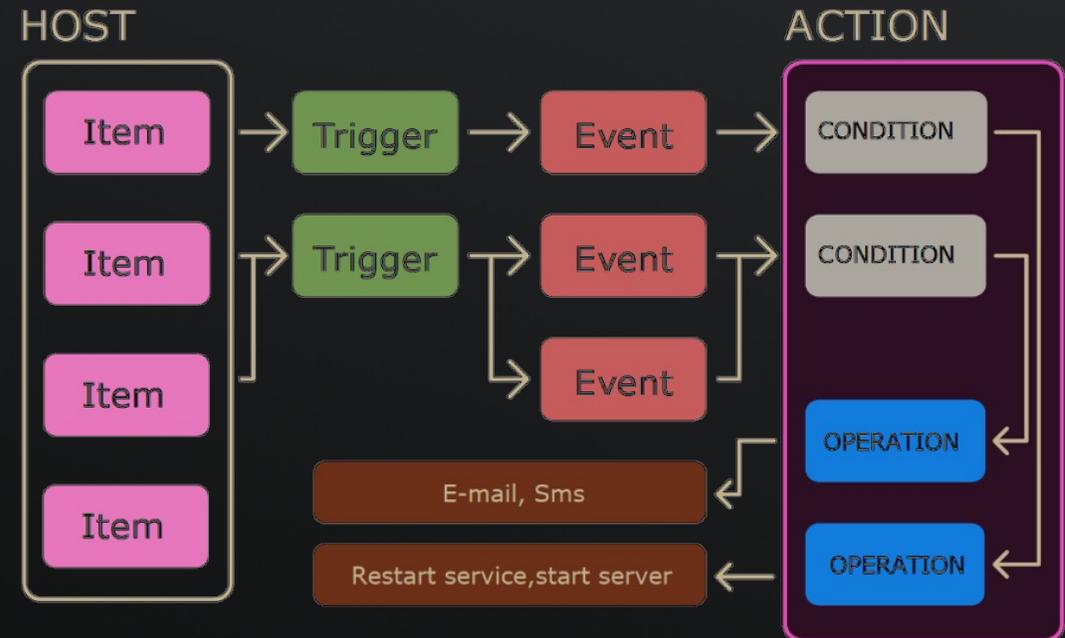
01 Entwicklung des Data Gathering

02 Anwendungsbeispiel

03 Kubernetes Monitoring

DAS KONZEPT ZABBIX

- „Vollständiges“ Monitoring Tool
- So vielseitig wie Monitoring-Anforderungen unterschiedlich sind
- möglichst viele Anwendungsszenarien abdecken
- Standardisiertes Monitoring
- Transportierbare, “self-contained“ Monitoring Templates
- Einheitliche Definition von Problemzuständen
- Vielfältige Datenerhebungsmethoden

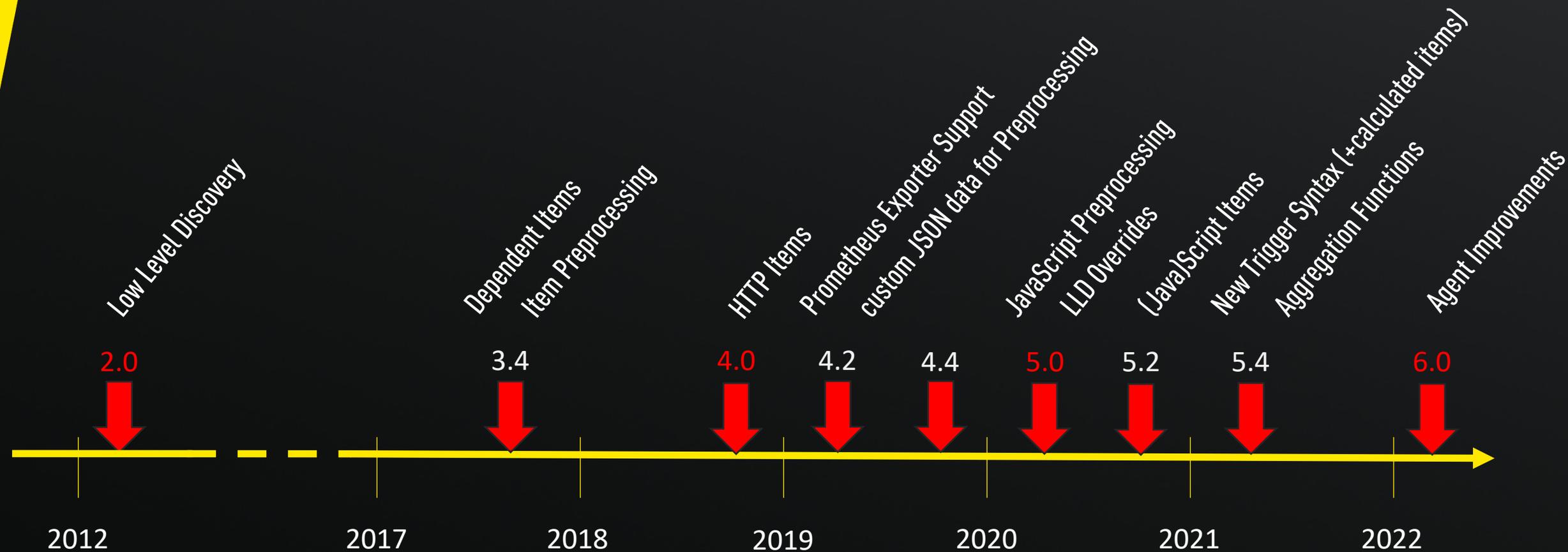


HERAUSFORDERUNGEN / ZIELE

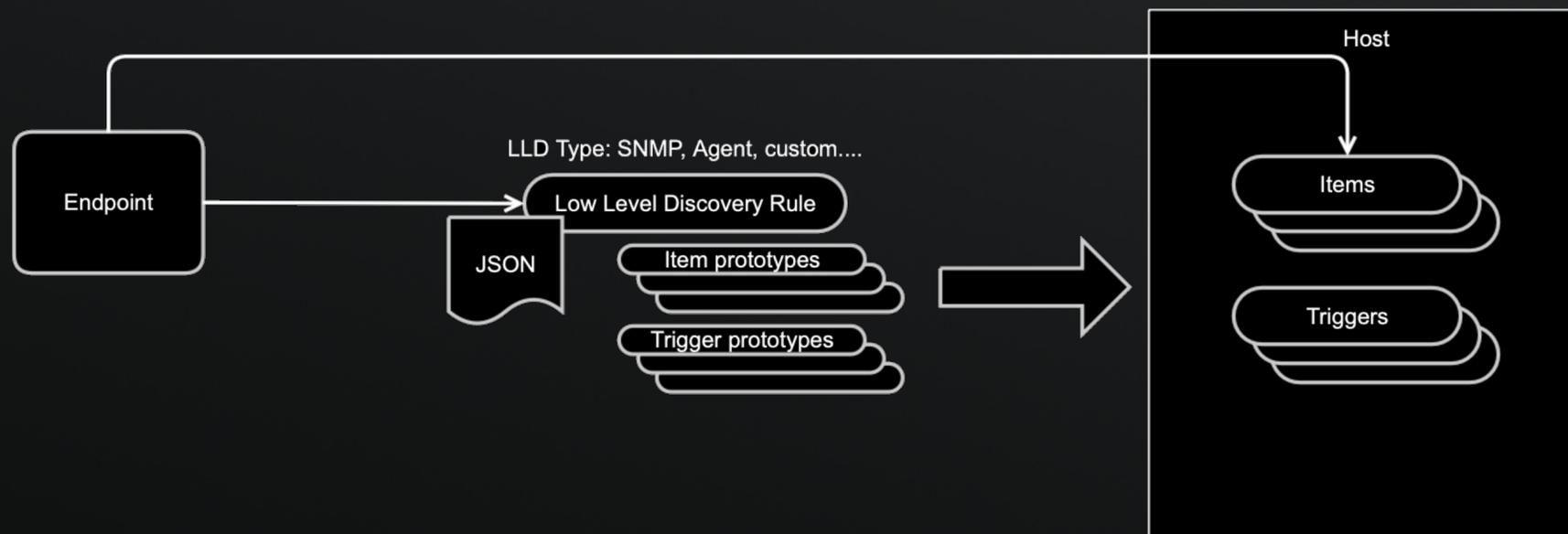
- Extraktion von Monitoring-Daten aus beliebigen Quelldaten
 - SNMP
 - Agenten
 - Agentless
 - APIs, ...
- Verzicht auf externe Skripte
- Flexibilität, Dynamik



FEATURE EVOLUTION



LOW LEVEL DISCOVERY

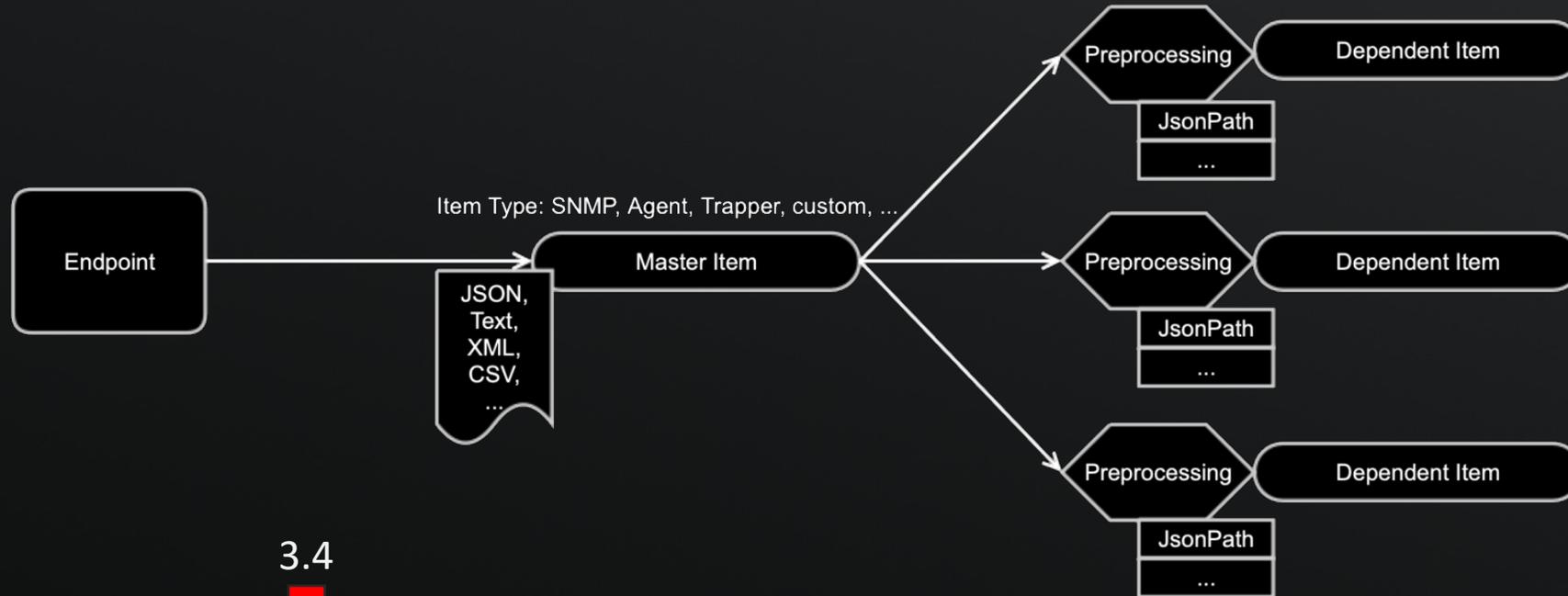


2.0

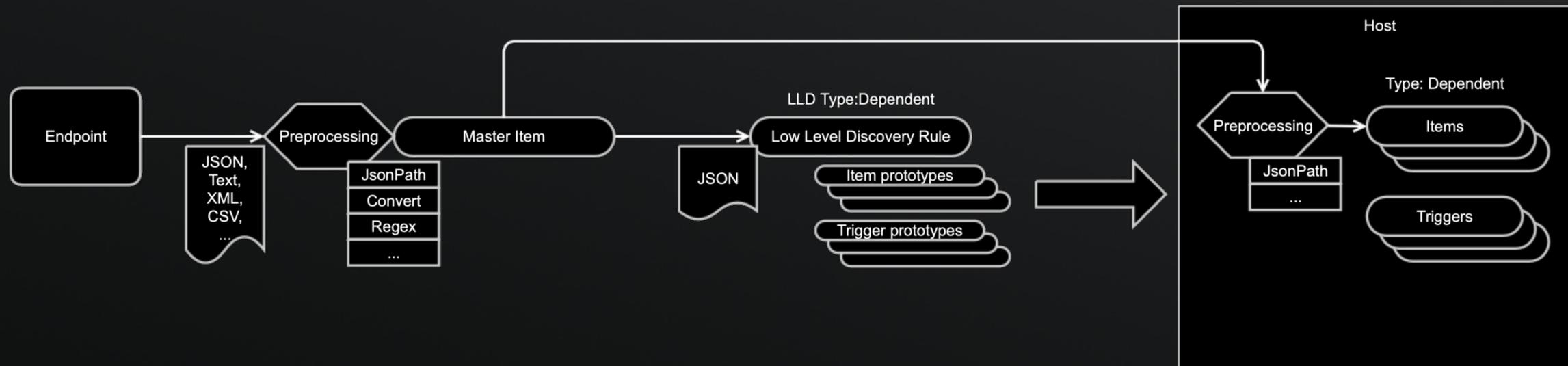


2012

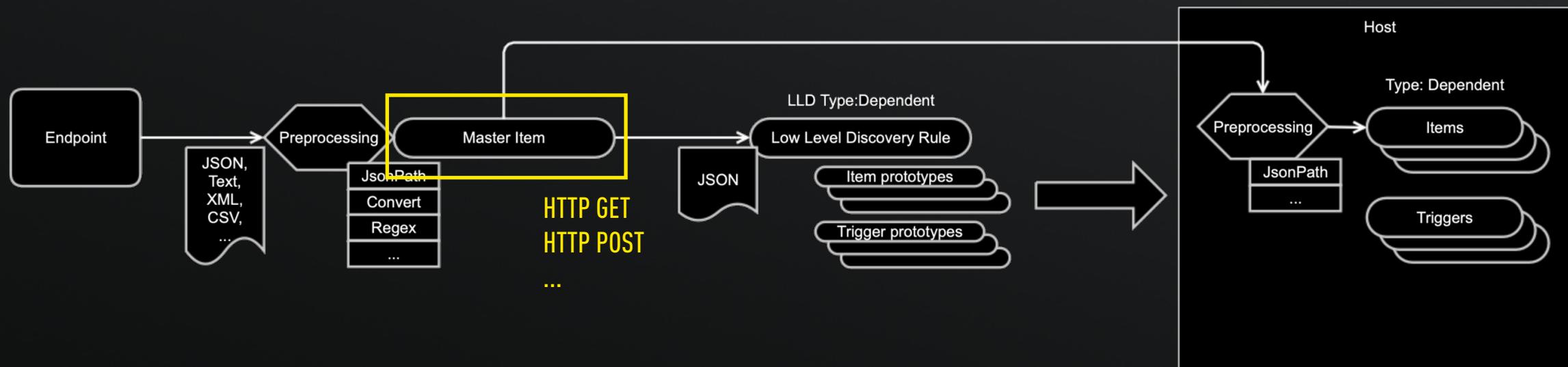
DEPENDENT ITEMS & PREPROCESSING



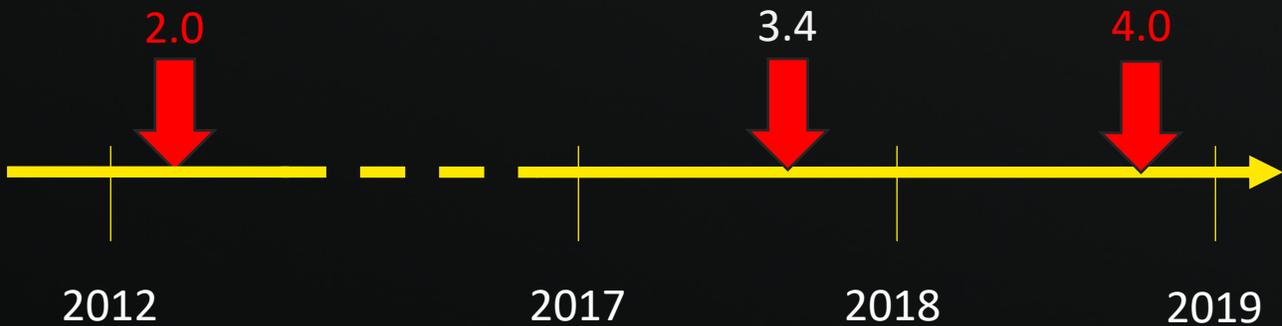
DEPENDENT ITEMS & PREPROCESSING



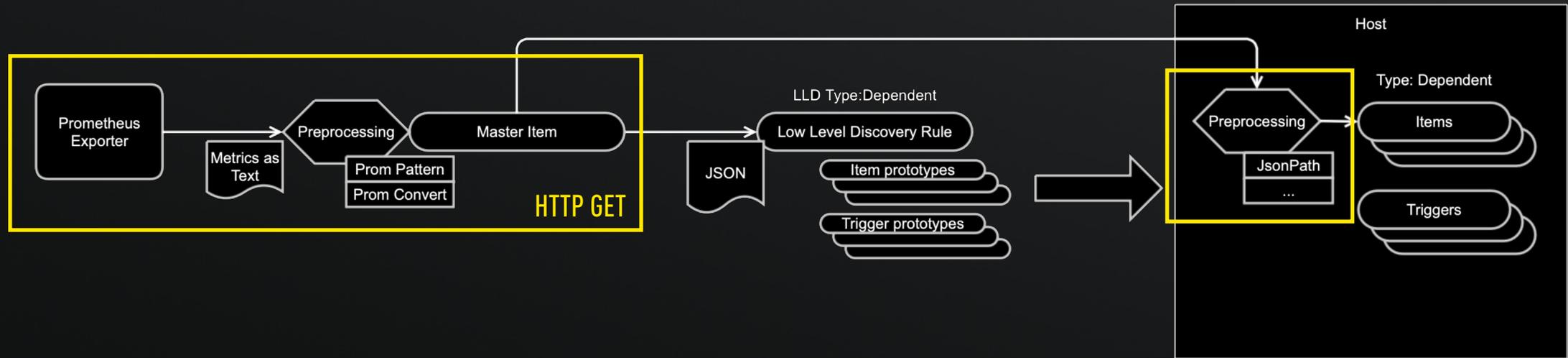
HTTP AGENT ITEMS



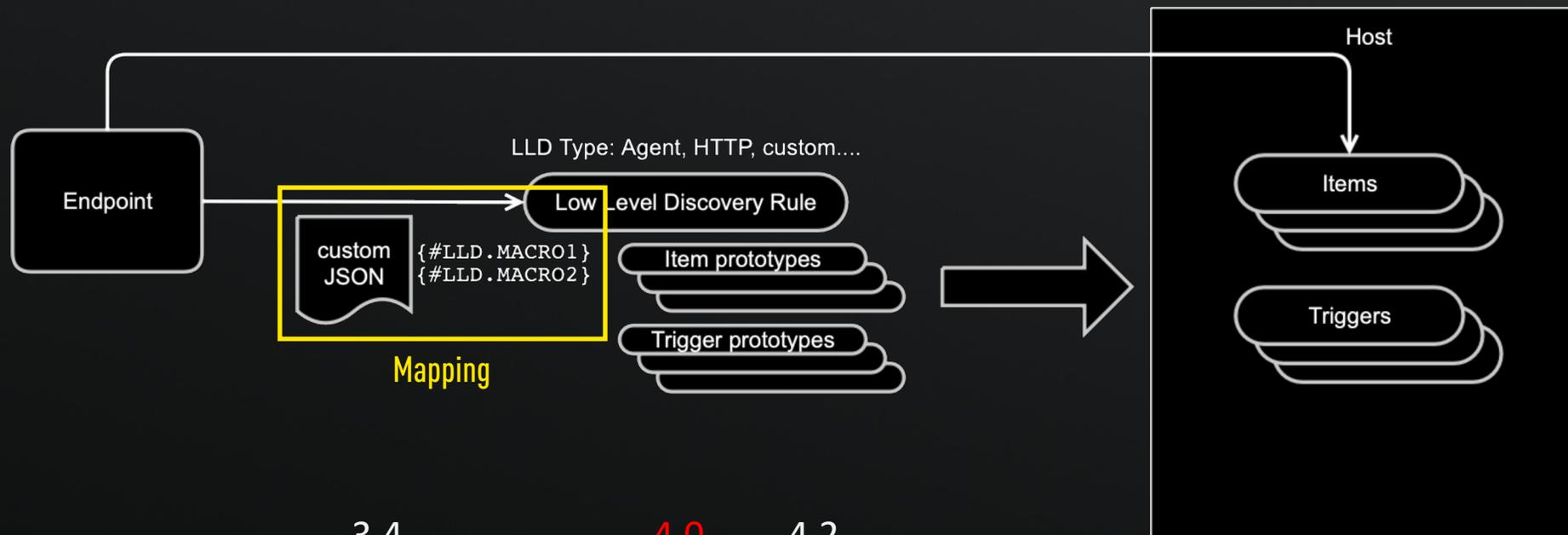
HTTP GET
HTTP POST
...



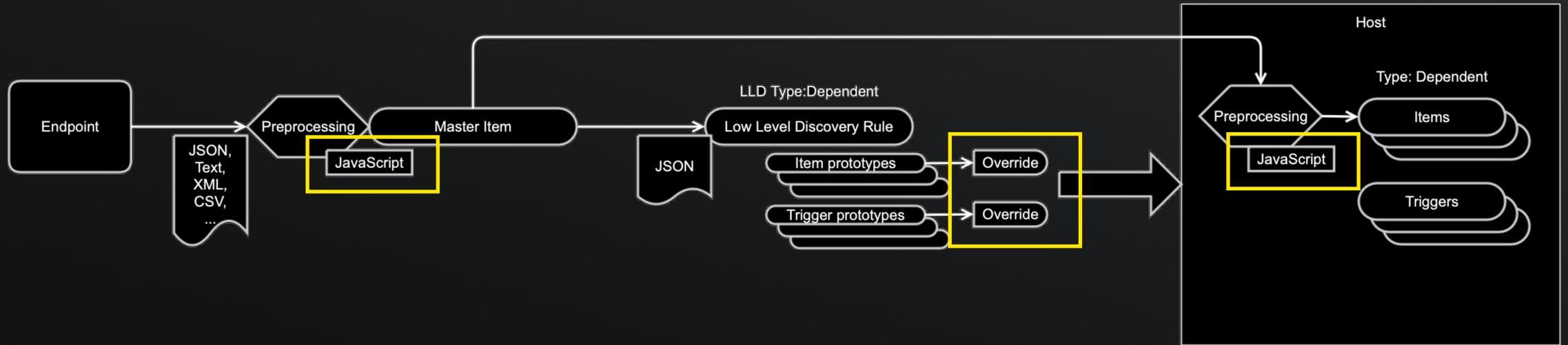
PROMETHEUS EXPORTER SUPPORT



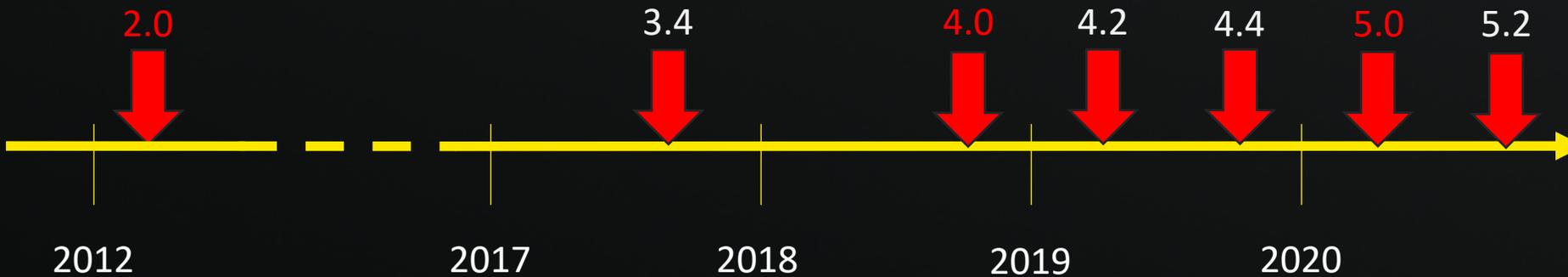
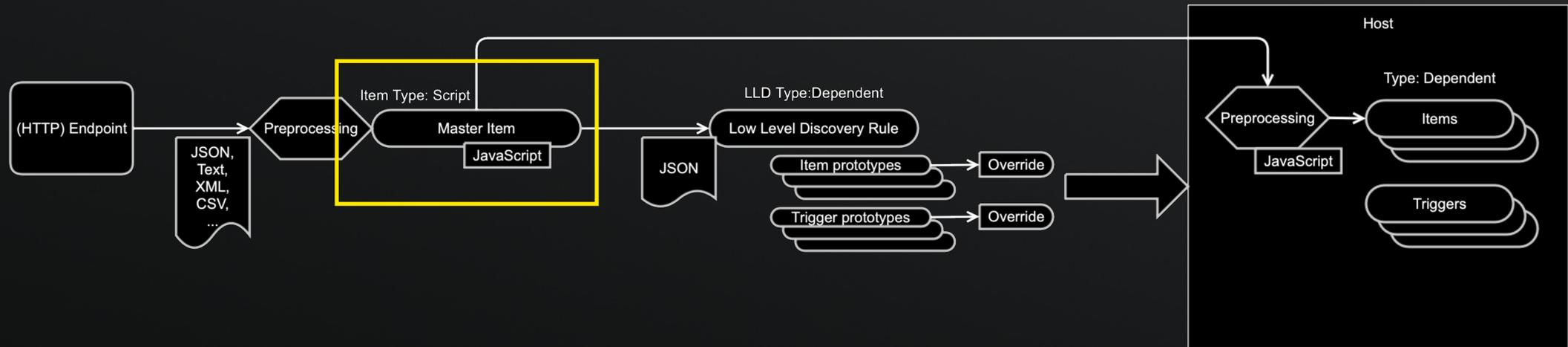
CUSTOM JSON FOR LOW LEVEL DISCOVERY



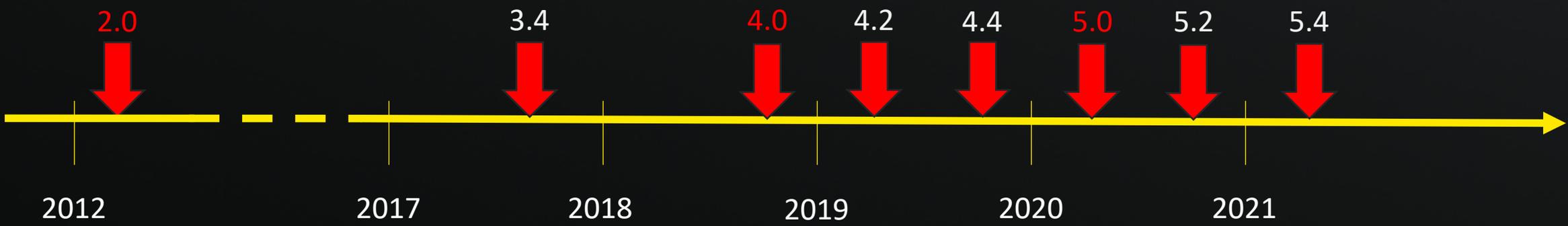
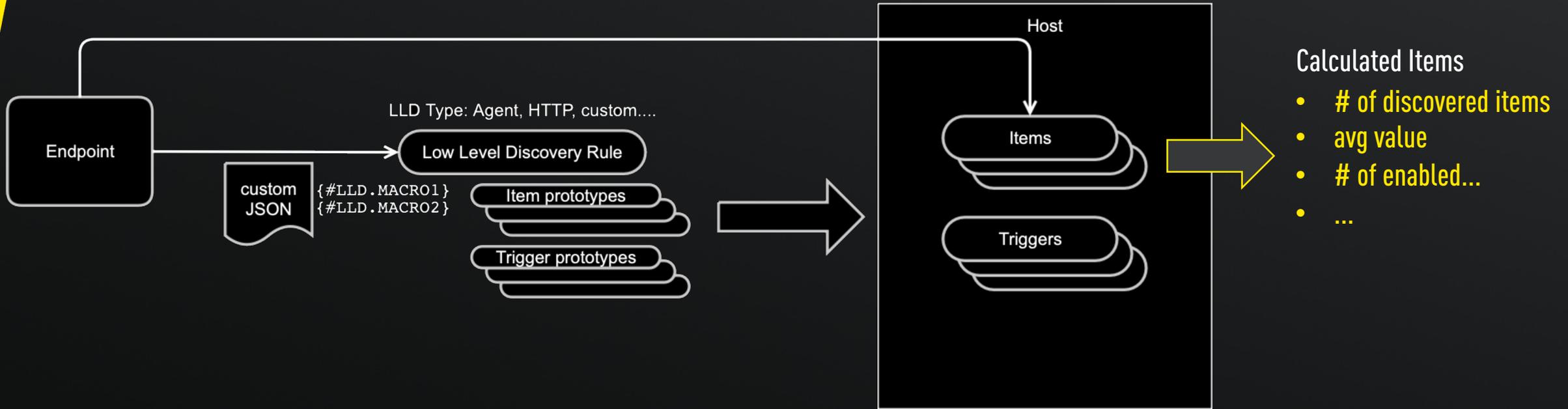
JAVASCRIPT PREPROCESSING + LLD OVERRIDES



JAVASCRIPT ITEMS



NEW TRIGGER SYNTAX + AGGREGATION



FEATURE EVOLUTION

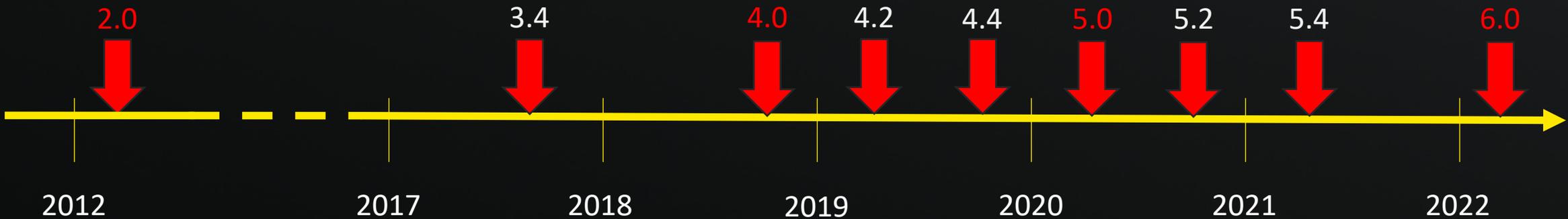
+ mehr!

Certificate Monitoring

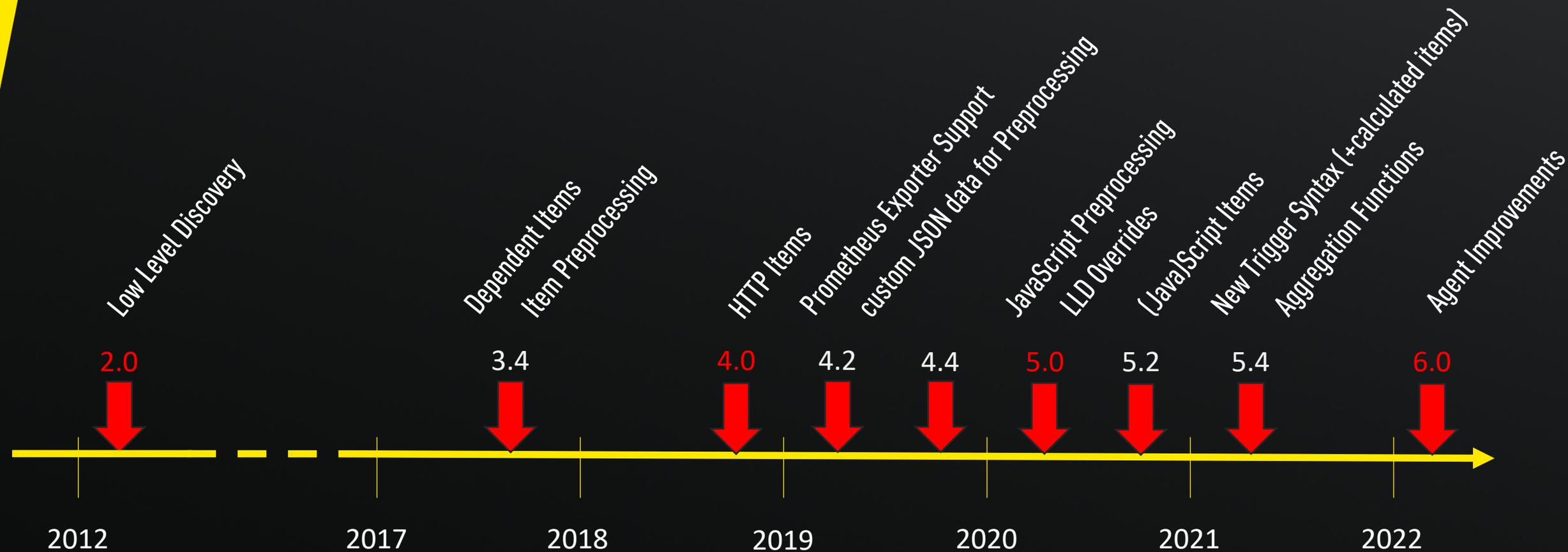
SD-WAN Velocloud Template

InfluxDB Template

Kubernetes Monitoring



FEATURE EVOLUTION



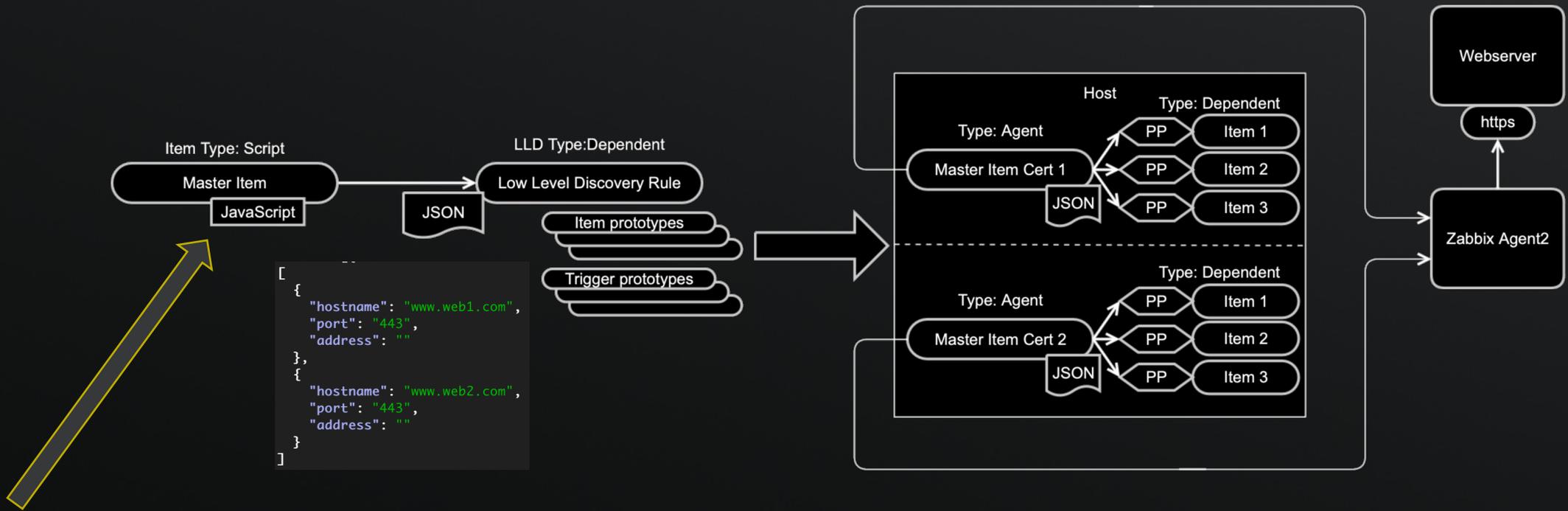
01 Entwicklung des Data Gathering

02 Anwendungsbeispiel

03 Kubernetes Monitoring

ZERTIFIKATSMONITORING - ERWEITERT

Mehrere Zertifikate auf einem Host



```
{ $MULTICERT.CONFIG }
```

```
HOSTNAME , <PORT> , <ADDRESS> ; HOSTNAME , <PORT> , <ADDRESS> ; ...
```

01 Entwicklung des Data Gathering

02 Anwendungsbeispiel

03 Kubernetes Monitoring

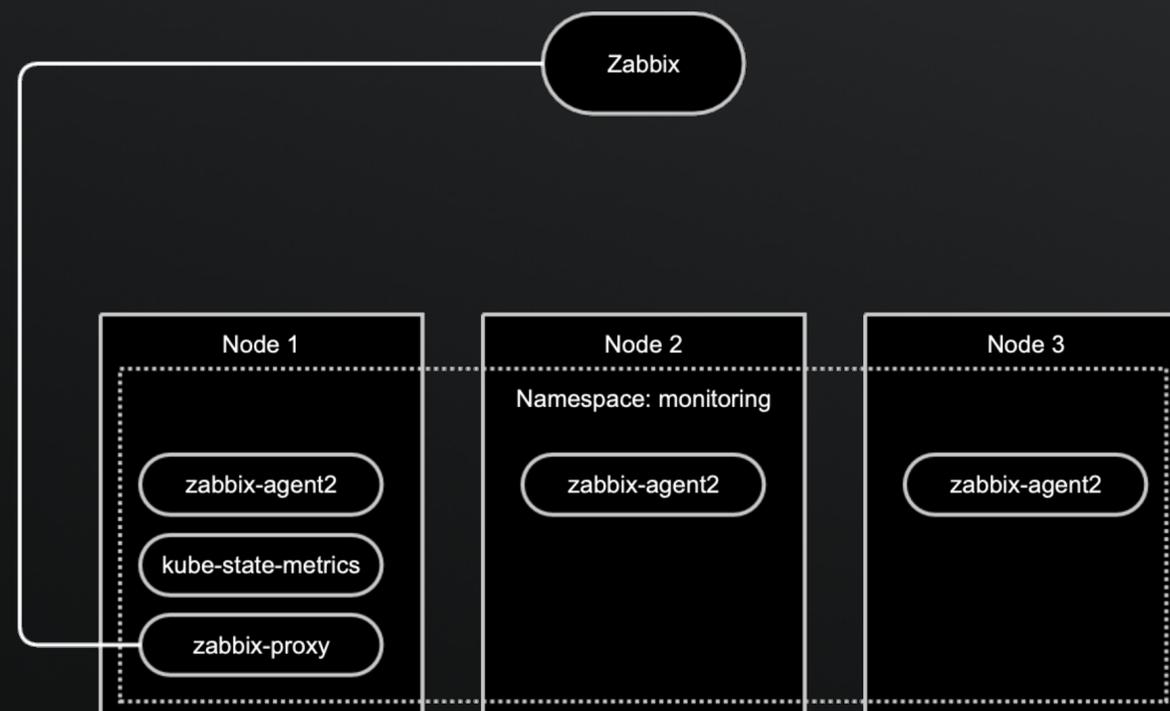
CLOUD NATIVE INFRASTRUCTURE

- Micro Services
- Hochgradig dynamisch
- vollständig API basiert – “alles ist ein Objekt“
- „Serverless“ und Servicezentrisch
- Metriken und Status via API
- Zusätzliche Metriken via Prometheus Exporter



VORAUSSETZUNGEN (EMPFEHLUNGEN)

- Zabbix Proxy im Cluster installiert
- Zabbix Agent auf jeder Node aktiv (DaemonSet)
- kube-state-metrics „Agent“



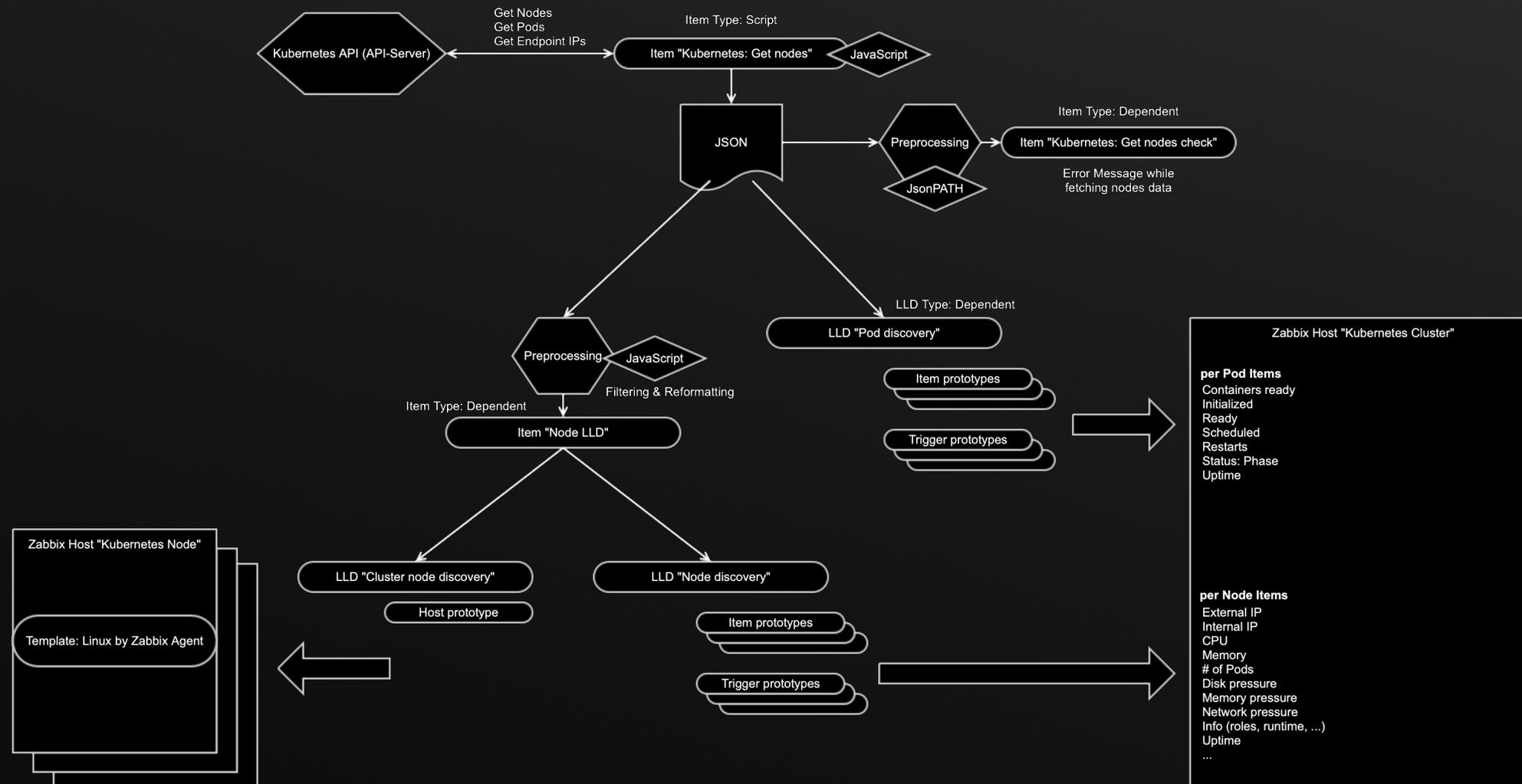
KUBERNETES MONITORING TEMPLATES

Kubernetes nodes by HTTP

- Interaktion mit dem K8S API Server
- „Dummy“-Host
- Cluster Nodes als Host Prototypes
 - Template: Linux by Zabbix Agent
- (Worker) Nodes (Items pro Host auf dem Dummy Host)
 - Memory, CPU, Disk pressure, Health
- Pods (Items auf dem Dummy Host)
 - **Filter nach Namespace verwenden!**
 - Restarts, Conditions, Scheduled, ...

```
{ $KUBE.LLD.FILTER.POD.NAMESPACE.MATCHES }  
{ $KUBE.LLD.FILTER.POD.NAMESPACE.NOT_MATCHES }
```

TEMPLATE: KUBERNETES NODES BY HTTP



TEMPLATE: KUBERNETES NODES BY HTTP

DEMO

KUBERNETES MONITORING TEMPLATES

Kubernetes cluster state by HTTP

Kubernetes API server by HTTP

Kubernetes Controller manager by HTTP

Kubernetes kubelet by HTTP

Kubernetes Scheduler by HTTP

```
{ $KUBE.LLD.FILTER.NAMESPACE.MATCHES }  
{ $KUBE.LLD.FILTER.NAMESPACE.NOT_MATCHES }
```

➤ Clusterkomponenten als Host Prototypes

- API Servers
- K8S Controller Managers
- Kubelets
- K8S Schedulers

➤ Daemonsets, Deployments, Endpoints, PVCs, ...

- **Filter nach Namespace verwenden!**

➤ Pods

- Mehr Informationen als über das „Nodes“ Template
- **Filter nach Namespace verwenden!**

TEMPLATE: KUBERNETES CLUSTER STATE

DEMO

FAZIT KUBERNETES MONITORING

- Zeigt, was derzeit alles möglich ist
- Sehr flexibel und umfangreich
- Customizing empfohlen/erforderlich (Filter, Duplikate, ...)
- Umfasst NICHT das Workload-Monitoring an sich (ist jedoch auch durch Zabbix supported)
 - Abfragen von durch Pods bereitgestellte Metriken
 - Auto-Discovery von Prometheus Exporter Endpoints
- Insbesondere in heterogenen Infrastrukturen eine wertvolle Erweiterung von Zabbix



07 SUPPORT

VIELEN DANK!

christian.anton@secadm.de

MCL

