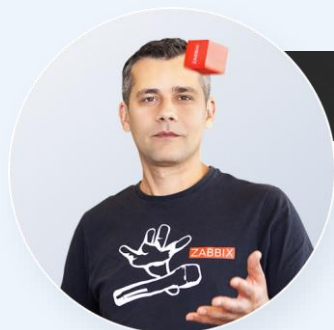


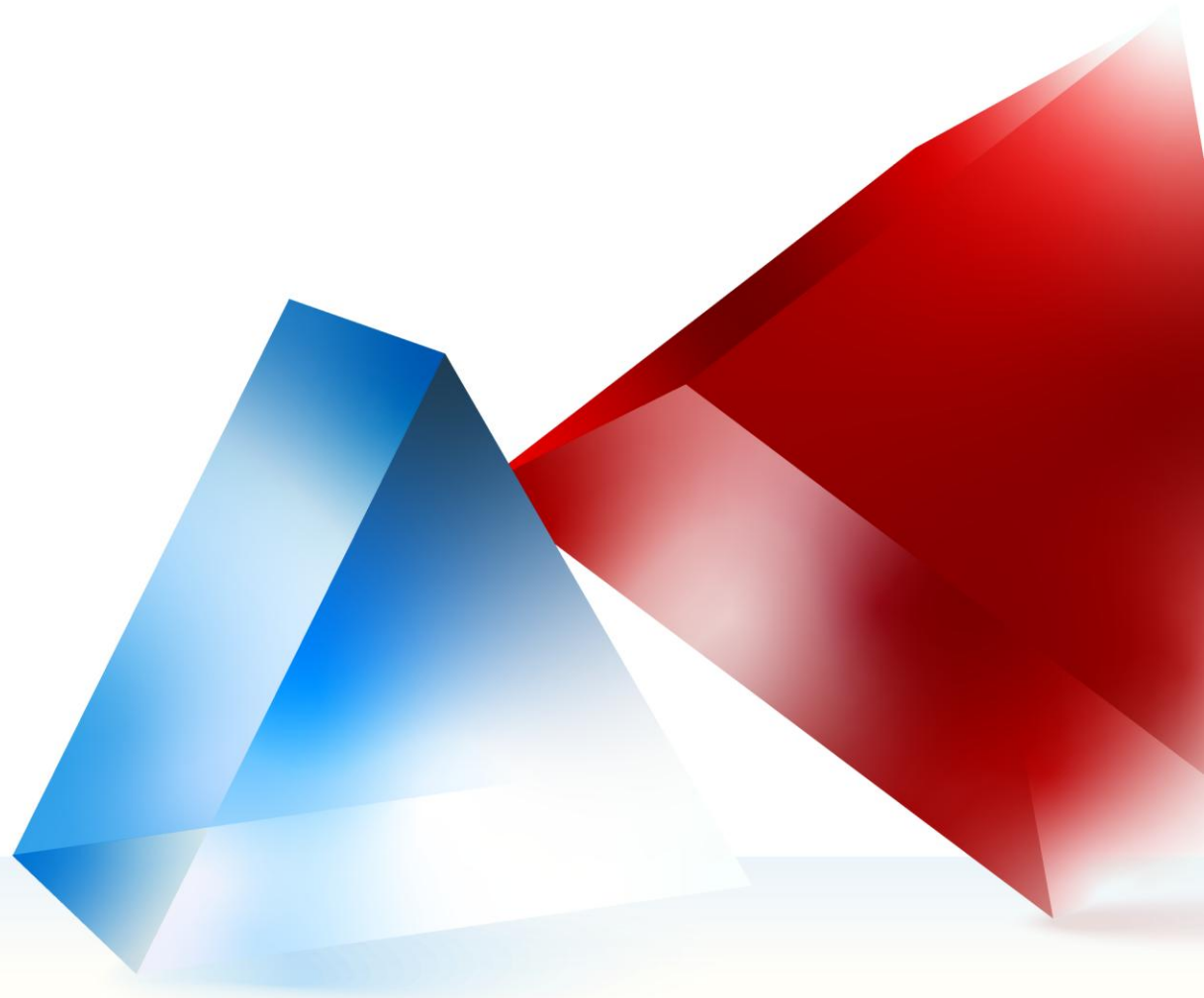
From Basics to Brilliance: LLD in action



Kaspars Mednis

Training Project Manager

What is LLD?



What is Low Level Discovery?

LLD is used to automatically discover entities **inside a host**:

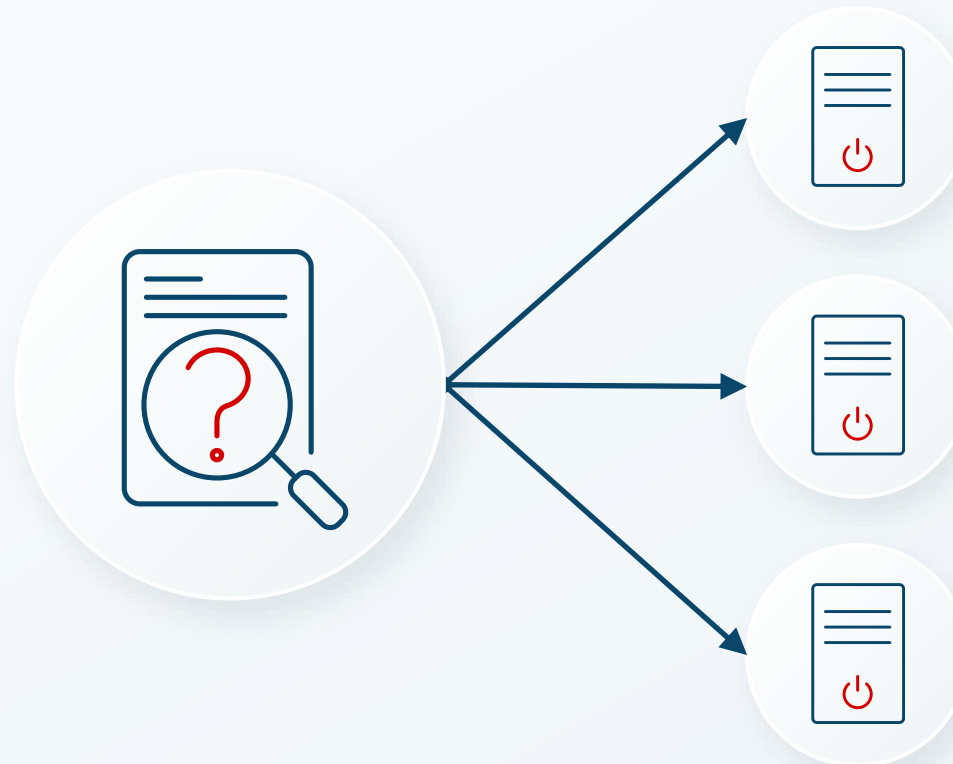
- ▶ Network interfaces
- ▶ Filesystems
- ▶ CPU cores
- ▶ Windows and Linux services
- ▶ Physical disk drives
- ▶ Database tables
- ▶ And much more...



Low Level Discovery Hosts

LLD can even create **new hosts** from the discovery results:

- ▶ Virtual machines
- ▶ Docker containers
- ▶ Kubernetes pods
- ▶ Cloud resources
- ▶ And much more...



LLD in Zabbix

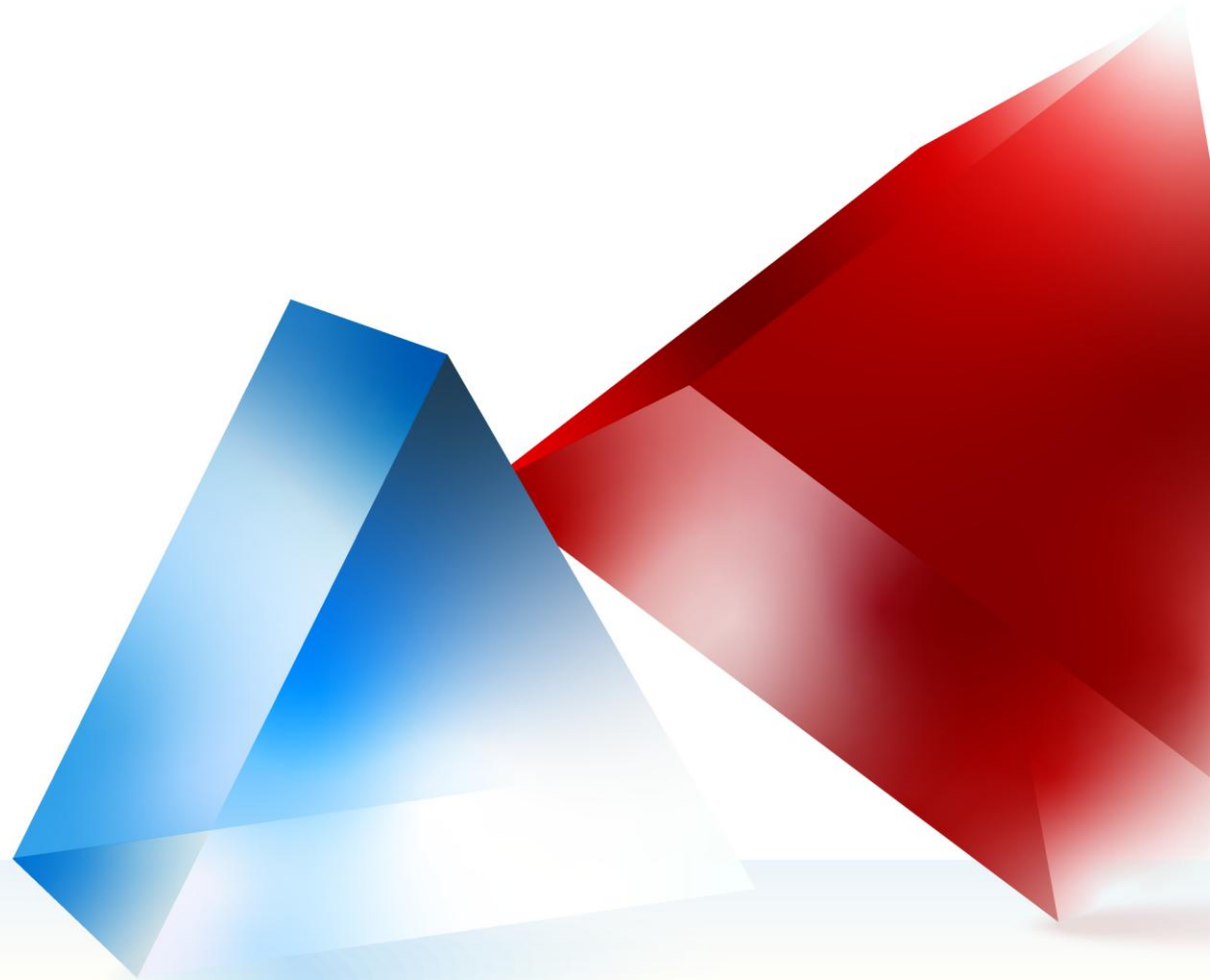
Most **official Zabbix templates** use LLD rules:

- ▶ Operating systems Linux, Windows, HP-UX
- ▶ SNMP devices Switches, routers, UPS, iLO
- ▶ Databases MySQL, PostgreSQL, Oracle
- ▶ Cloud providers AWS, Azure, GCP
- ▶ VMWare Hypervisors, Virtual Machines, Datastores

LLD in templates

							LLD rules			
<input type="checkbox"/> Name ▲	Hosts	Items	Triggers	Graphs	Dashboards	Discovery	Web	Vendor	Version	
<input type="checkbox"/> Apache Tomcat by JMX	Hosts	Items 1	Triggers 1	Graphs	Dashboards 1	Discovery 4	Web	Zabbix	7.4-0	
<input type="checkbox"/> APC Smart-UPS 3000 XLM by SNMP	Hosts	Items 26	Triggers 22	Graphs 3	Dashboards 1	Discovery 6	Web	Zabbix	7.4-1	
<input type="checkbox"/> AWS by HTTP	Hosts	Items	Triggers	Graphs	Dashboards	Discovery 7	Web	Zabbix	7.4-4	
<input type="checkbox"/> Cisco IOS by SNMP	Hosts	Items 23	Triggers 8	Graphs	Dashboards 1	Discovery 8	Web	Zabbix	7.4-2	
<input type="checkbox"/> DELL PowerEdge R820 by SNMP	Hosts	Items 28	Triggers 10	Graphs 1	Dashboards 1	Discovery 12	Web	Zabbix	7.4-3	
<input type="checkbox"/> FortiGate by SNMP	Hosts	Items 53	Triggers 11	Graphs 5	Dashboards 3	Discovery 9	Web	Zabbix	7.4-2	
<input type="checkbox"/> HP iLO by SNMP	Hosts	Items 16	Triggers 9	Graphs	Dashboards	Discovery 13	Web	Zabbix	7.4-2	
<input type="checkbox"/> Linux by Zabbix agent	Hosts 1	Items 43	Triggers 15	Graphs 8	Dashboards 3	Discovery 3	Web	Zabbix	7.4-2	
<input type="checkbox"/> VMware FQDN	Hosts	Items 7	Triggers 2	Graphs	Dashboards 1	Discovery 5	Web	Zabbix	7.4-2	
<input type="checkbox"/> Windows by Zabbix agent	Hosts	Items 34	Triggers 13	Graphs 5	Dashboards 3	Discovery 4	Web	Zabbix	7.4-2	

How LLD Works



Prototypes

Low level discovery rules use the entity prototypes (blueprints)



Item prototypes



Trigger prototypes



Graph prototypes



Host prototypes

LLD rules and prototypes



Prototypes

All templates / Linux by Zabbix agent Items 43 Triggers 15 Graphs 8 Dashboards 2 **Discovery rules 3** Web scenarios Filter

<input type="checkbox"/> Template	Name ▲	Items	Triggers	Graphs	Hosts	Key	Interval	Type	Status
<input type="checkbox"/>	Linux by Zabbix agent Block devices discovery	Item prototypes 9	Trigger prototypes 1	Graph prototypes 3	Host prototypes	vfs.dev.discovery	1h	Zabbix agent	Enabled
<input type="checkbox"/>	Linux by Zabbix agent Linux: Get filesystems: Mounted filesystem discovery	Item prototypes 6	Trigger prototypes 5	Graph prototypes 1	Host prototypes	vfs.fs.dependent.discovery		Dependent item	Enabled
<input type="checkbox"/>	Linux by Zabbix agent Network interface discovery	Item prototypes 9	Trigger prototypes 4	Graph prototypes 1	Host prototypes	net.if.discovery	1h	Zabbix agent	Enabled

Displaying 3 of 3 found

Item prototypes

?

Create item prototype

All hosts / Zabbix server Enabled **ZBX** Discovery list / Network interface discovery **Item prototypes 9** Trigger prototypes 3 Graph prototypes 1 Host prototypes Discovery prototypes

<input type="checkbox"/>	Name ▲	Key	Interval	History	Trends	Type	Create enabled	Discover	Tags
<input type="checkbox"/>	... Linux by Zabbix agent: Interface {#IFNAME}: Bits received	net.if.in["{#IFNAME}"]	15s	31d	365d	Zabbix agent	Yes	Yes	component: network interface: {#IFNAME}
<input type="checkbox"/>	... Linux by Zabbix agent: Interface {#IFNAME}: Bits sent	net.if.out["{#IFNAME}"]	15s	31d	365d	Zabbix agent	Yes	Yes	component: network interface: {#IFNAME}

Item vs Item prototype

Item

Item Tags 2 Preprocessing 2

Discovered by [Network interface discovery](#)

* Name Interface eth1: Bits received

Type Zabbix agent

* Key net.if.in["eth1"]

Type of information Numeric (unsigned)

* Host interface demo.example.com:100

Units bps

* Update interval 15s

Item prototype

Item prototype Tags 2 Preprocessing 2

* Name Interface {#IFNAME} Bits received

Type Zabbix agent

* Key net.if.in "{#IFNAME}"

Type of information Numeric (unsigned)

Units bps

* Update interval 15s

LLD rule



Traffic in on {#IFNAME}



Traffic out on {#IFNAME}



High traffic in on {#IFNAME}



High traffic out on {#IFNAME}



Traffic graph on {#IFNAME}



Traffic in on **eth0**



Traffic out on **eth0**



High traffic in on **eth0**



High traffic out on **eth0**



Traffic graph on **eth0**



Traffic in on **eth1**



Traffic out on **eth1**



High traffic in on **eth1**



High traffic out on **eth1**



Traffic graph on **eth1**

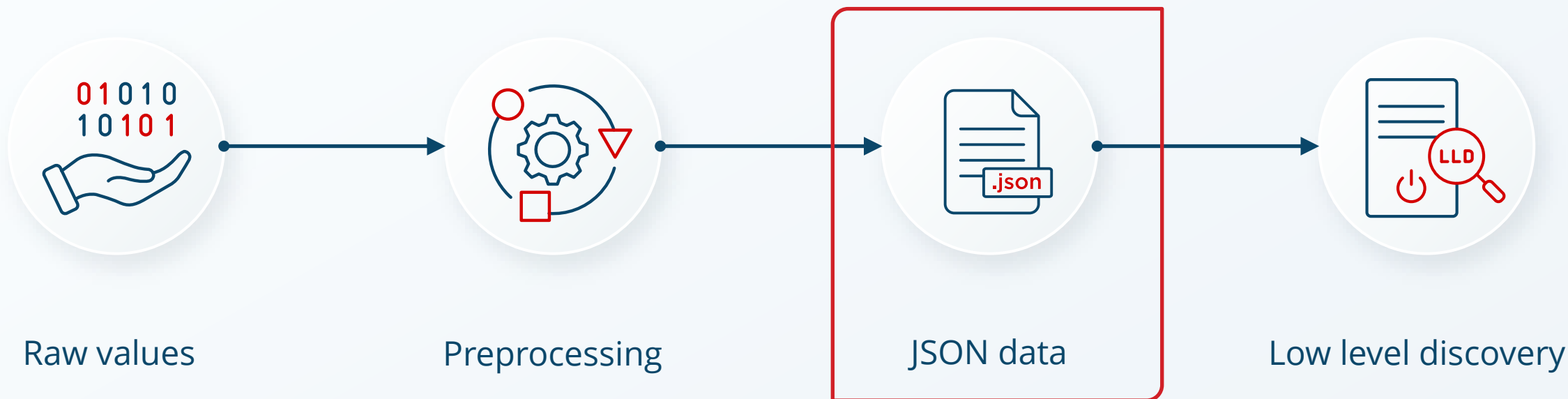
Network interface
eth0

Network interface
eth1

LLD data flow

LLD is based on data in **JSON format**:

- ▶ Data collector collects data for discovery (any format)
- ▶ Preprocessing transforms raw data to JSON
- ▶ Zabbix creates entities based on the JSON structure



JSON data structure

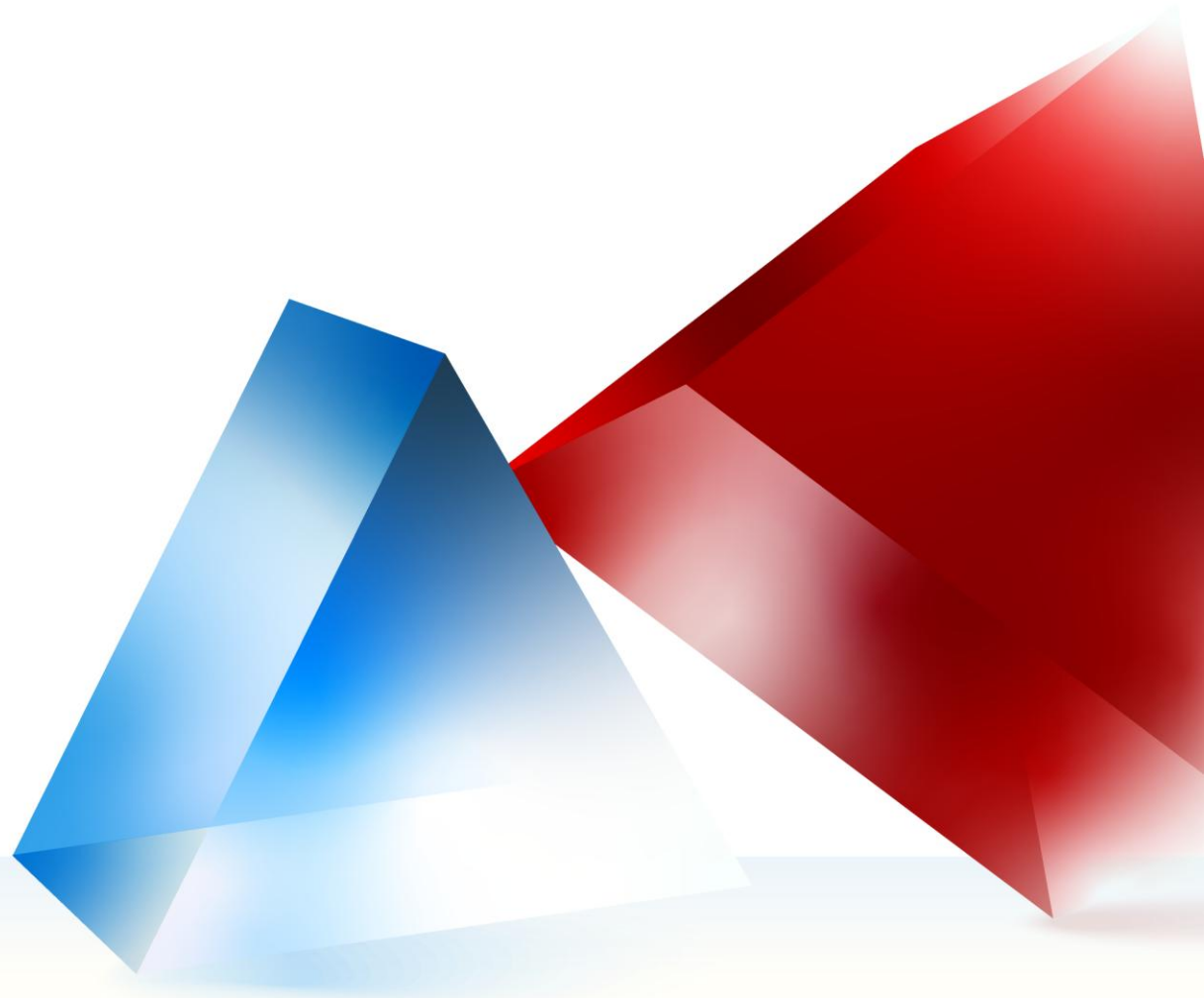
JSON data in Zabbix format

```
[
  {
    "{#FSNAME}": "/sys",
    "{#FSTYPE}": "sysfs"
  }, {
    "{#FSNAME}": "/",
    "{#FSTYPE}": "xfs"
  }, {
    "{#FSNAME}": "/root",
    "{#FSTYPE}": "rootfs"
  }
]
```

Free-form JSON data

```
[
  {
    "TABLE_NAME": "acknowledges",
    "DATA_LENGTH": "16384",
    "INDEX_LENGTH": "49152"
  },
  {
    "TABLE_NAME": "actions",
    "DATA_LENGTH": "16384",
    "INDEX_LENGTH": "32768"
  }
]
```

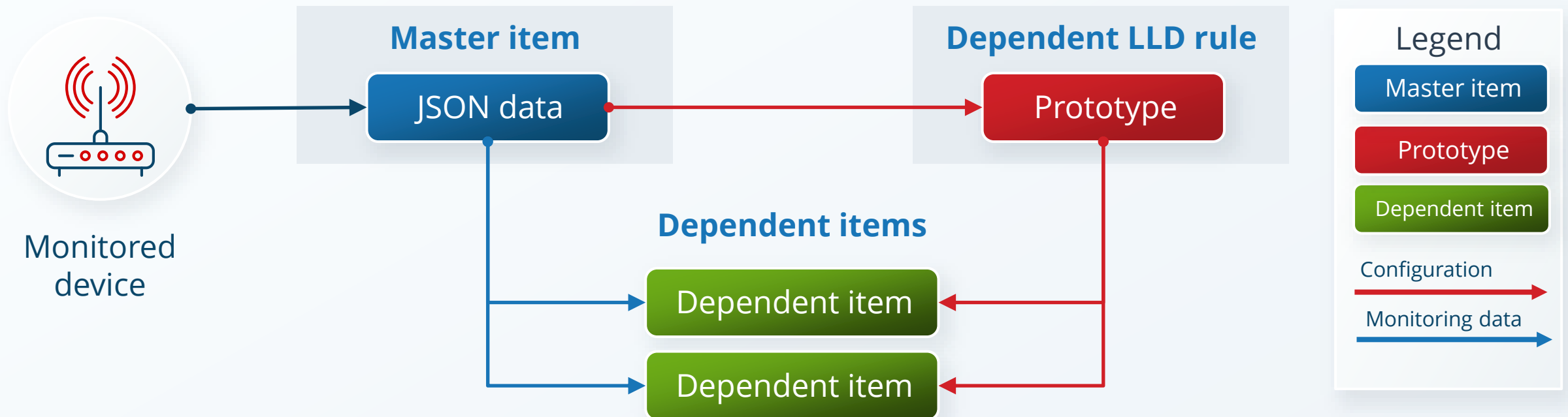
Dependent LLD



Dependent LLD

Dependent LLD utilizes **the same JSON data** for two purposes:

- ▶ Configuration of items and triggers discovered by LLD rule
- ▶ The actual monitoring data collected by items



JSON Example



```
{
  "fsname": "/",
  "fstype": "xfs"    "bytes": {
    "total": 53616816128,
    "free": 46228865024,
    "used": 7387951104,
    "pfree": 86.22083212408087,
    "pused": 13.779167875919129
  },
  "inodes": {
    "total": 26212800,
    "free": 26163440,
    "used": 49360,
    "pfree": 99.81169504974669,
    "pused": 0.18830495025331137
  },
  "options": "rw,relatime,attr2,inode64,logbufs=8,logbsize=32k,noquota"
},
```

Legend

Configuration data

Item values

How it looks in Zabbix



Discovery rule

Preprocessing 2

LLD macros 2

Filters 4

Overrides 1

* Name

Mounted filesystem discovery

Type

Dependent item

* Key

vfs.fs.dependent.discovery

* Master item

Linux by Zabbix agent: Get filesystems

Select

* Delete lost resources

Never

Immediately

After

7d

* Disable lost resources

Never

Description

The discovery rule will discover the following resources:

Enabled

☒

Name ▲	Key
<input type="checkbox"/> ... Get filesystems: FS [{#FSNAME}]: Get data	vfs.fs.dependent[{#FSNAME},data]
<input type="checkbox"/> ... FS [{#FSNAME}]: Get data: FS [{#FSNAME}]: Inodes: Free, in %	vfs.fs.dependent.inode[{#FSNAME},pfree]
<input type="checkbox"/> ... FS [{#FSNAME}]: Get data: FS [{#FSNAME}]: Option: Read-only	vfs.fs.dependent[{#FSNAME},readonly]
<input type="checkbox"/> ... FS [{#FSNAME}]: Get data: FS [{#FSNAME}]: Space: Available	vfs.fs.dependent.size[{#FSNAME},free]
<input type="checkbox"/> ... FS [{#FSNAME}]: Get data: FS [{#FSNAME}]: Space: Total	vfs.fs.dependent.size[{#FSNAME},total]
<input type="checkbox"/> ... FS [{#FSNAME}]: Get data: FS [{#FSNAME}]: Space: Used	vfs.fs.dependent.size[{#FSNAME},used]
<input type="checkbox"/> ... FS [{#FSNAME}]: Get data: FS [{#FSNAME}]: Space: Used, in %	vfs.fs.dependent.size[{#FSNAME},pused]

BENEFITS



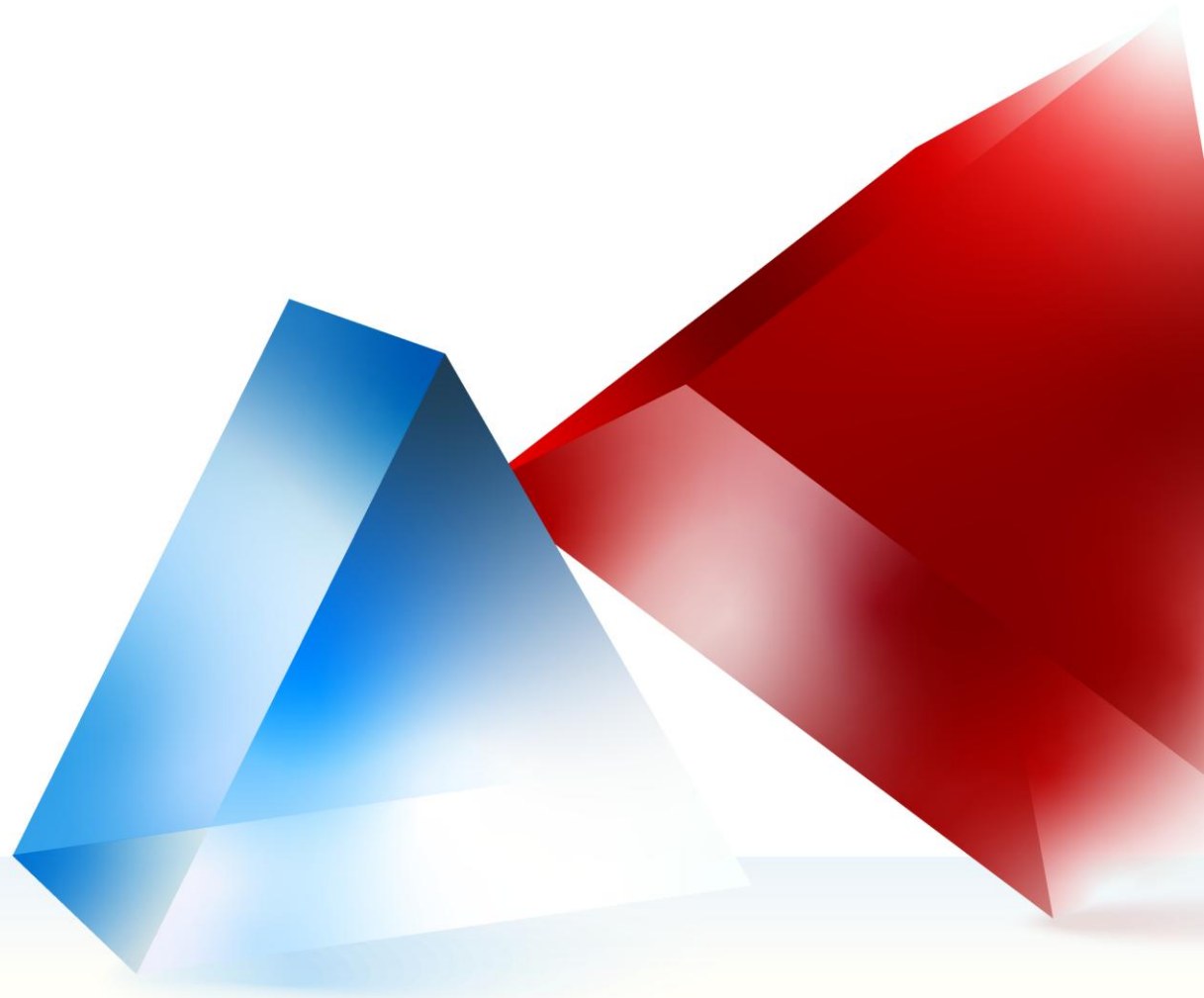
Dependent LLD rules provides lots of benefits:

- ▶ All required data is collected in a **single request**
- ▶ **Fewer connections** are opened from Zabbix to monitored devices
- ▶ All monitoring data are **collected at the same time** and are consistent

Keep in mind:

- ▶ Dependent items **use preprocessors** to extract data from a master item
- ▶ Dependent LLD rule executes **every time data is collected** by the master item
- ▶ Use **discard unchanged data with heartbeat** preprocessing step on LLD rules

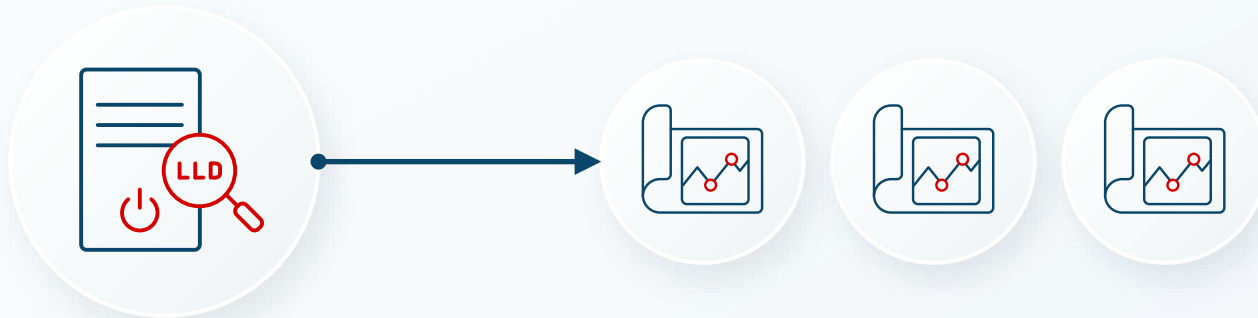
Nested LLD



Pre-7.4 limitations

Previous Zabbix versions were limited to a single LLD level:

- ▶ Enough in most simple use cases
- ▶ Impossible to implement more complex scenarios



Nested LLD

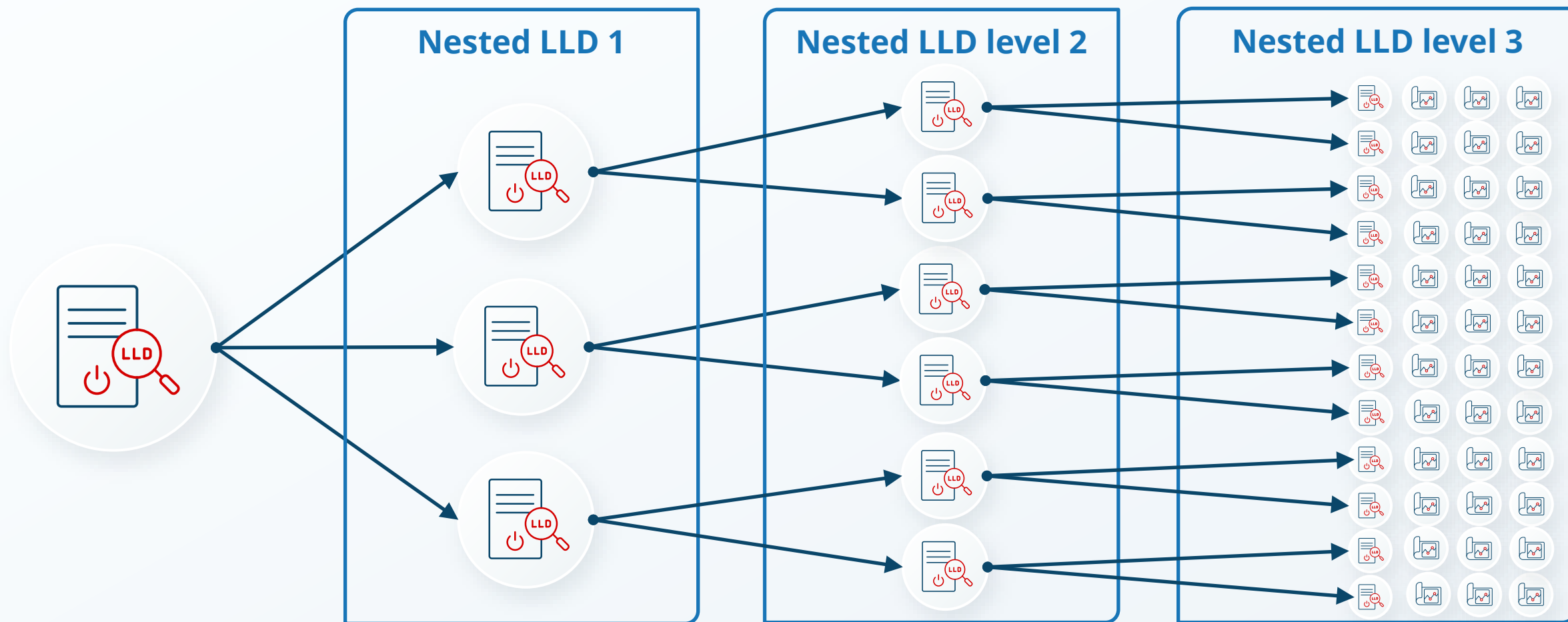
A parent LLD rule discovers entities and spawns **child LLD rules**:

- ▶ More **complex scenarios** can be implemented
- ▶ Allows users to fully utilize **multi-level JSON** data



Multiple levels of nested LLD

Unlimited levels of nested LLD rules are supported:



How nested LLD works

Each nested LLD level inherits its portion of JSON data from the previous level:

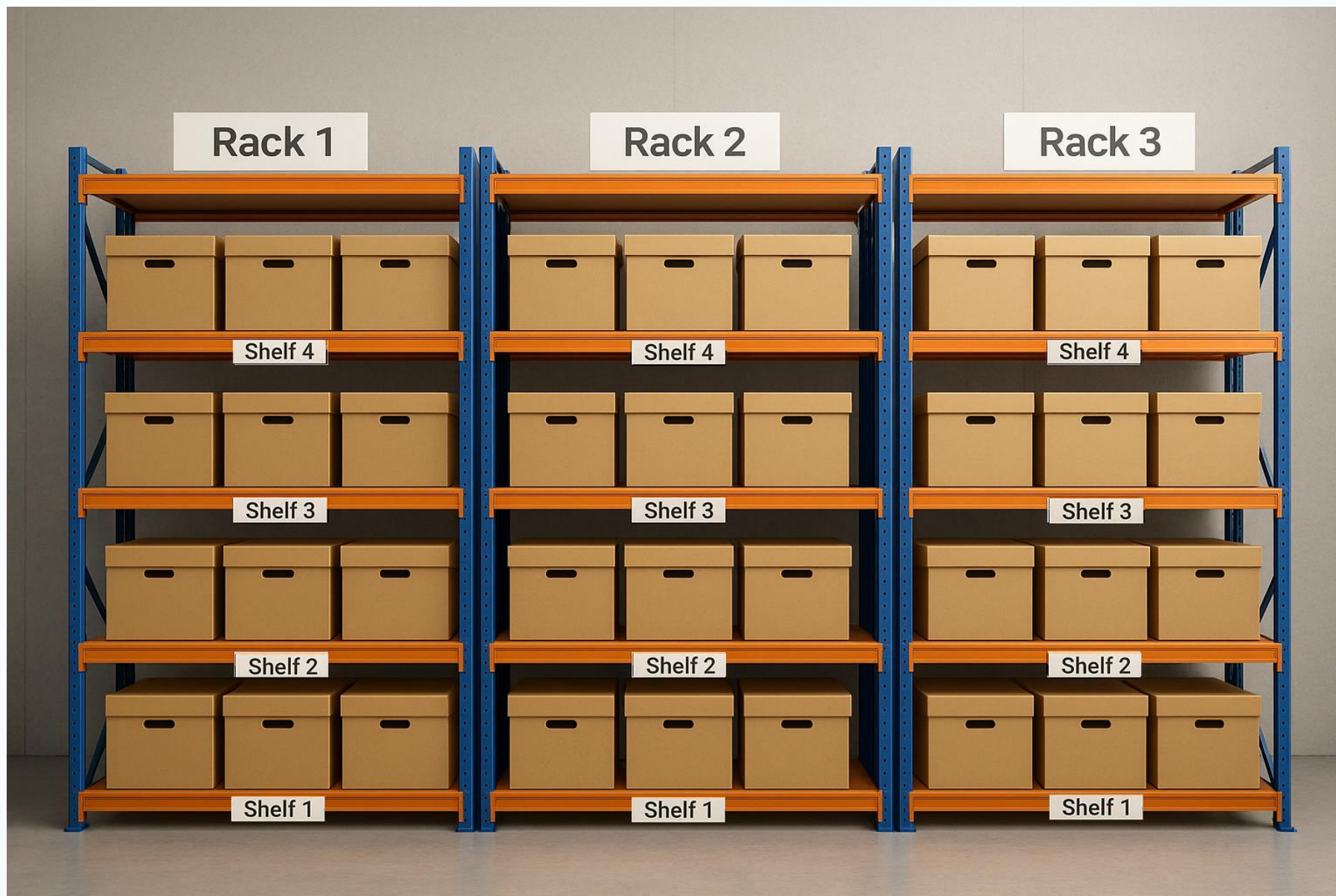
- ▶ Nested LLD rules are created using **discovery prototypes**
- ▶ Discovery prototype have a new "**nested**" type

The screenshot shows the Zabbix web interface for configuring a discovery prototype. The breadcrumb navigation at the top includes 'All hosts / Warehouse Enabled ZBX Discovery list / ... / Shelves discovery Item prototypes Trigger prototypes Graph prototypes Host prototypes' and 'Discovery prototypes 1' (highlighted with a red box). Below the breadcrumb, the 'Discovery prototype' tab is active, with sub-tabs for 'Preprocessing 1', 'LLD macros 2', 'Filters', and 'Overrides'. The form contains the following fields and controls:

- Name:** Text input field containing 'Boxes discovery'.
- Type:** Dropdown menu with 'Nested' selected (highlighted with a red box).
- Key:** Text input field containing 'warehouse.bboxes.discovery[{#RACK},{#SHELF}]'.
- Delete lost resources:** Radio buttons for 'Never', 'Immediately', and 'After' (selected), followed by a text input '7d'.
- Disable lost resources:** Radio buttons for 'Never', 'Immediately' (selected), and 'After'.
- Description:** Text area.
- Create enabled:** Checkmark.
- Discover:** Checkmark.
- Buttons:** 'Update' (blue), 'Clone' (blue), 'Test' (grey), 'Delete' (blue), and 'Cancel' (blue).

Example LLD scenario

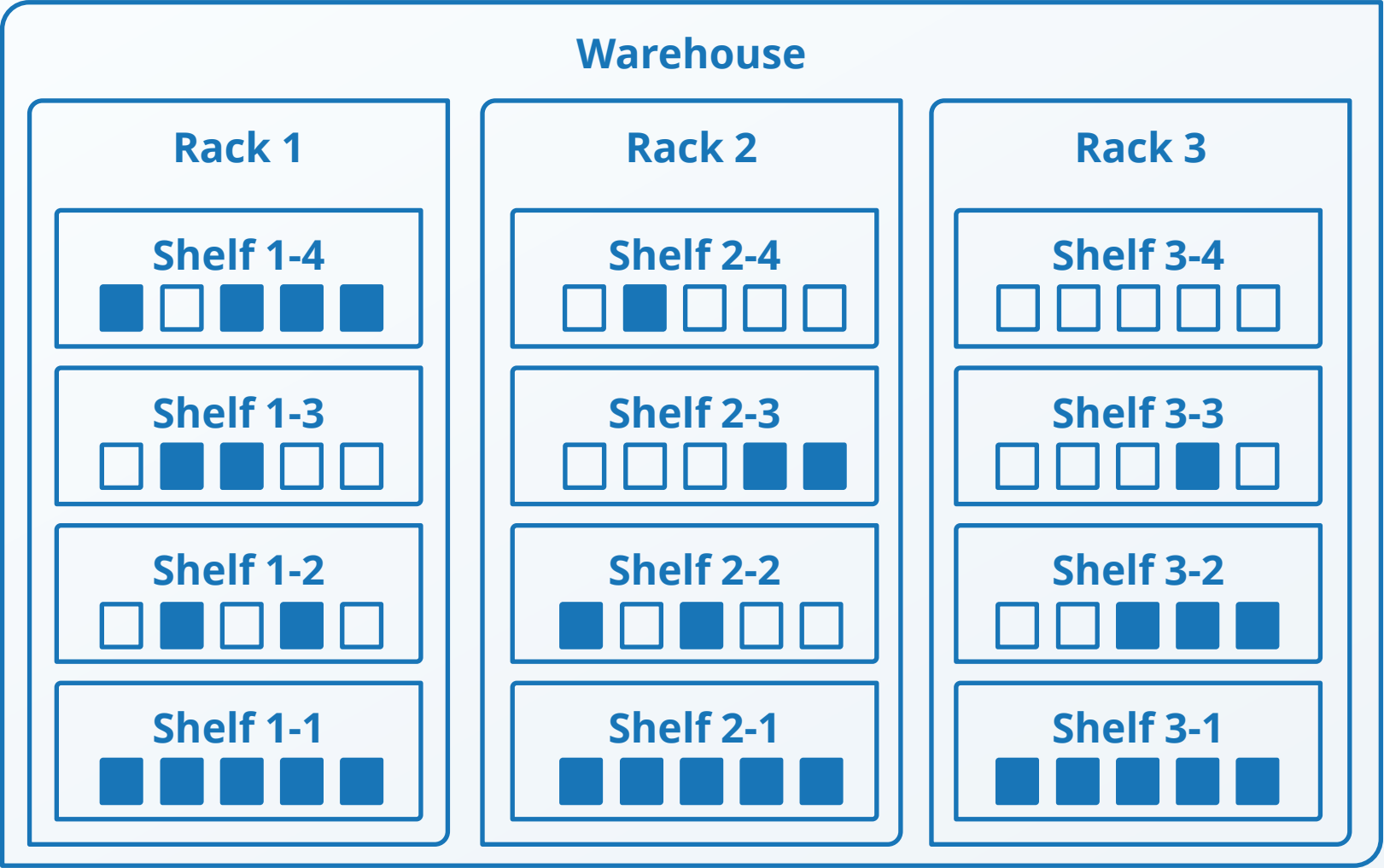
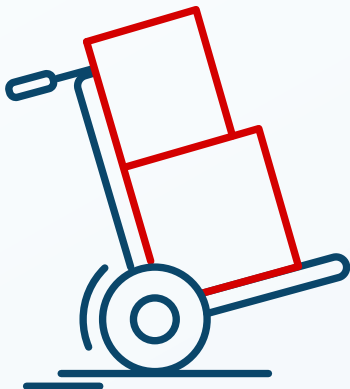




Example scenario

Warehouse

- ▶ LLD Level 1: Racks
- ▶ LLD Level 2: Shelves
- ▶ LLD Level 3: Boxes



Example scenario JSON

Data is represented as a **hierarchical JSON**:

- ▶ Level 1: **Racks**
- ▶ Level 2: **Shelves**
- ▶ Level 3: **Boxes**



```
{
  "warehouse": {
    "racks": [
      {
        "rack": "Rack 1",
        "shelves": [
          {
            "shelf": "Shelf 1-4",
            "boxes": [
              {
                "box": "Box 1-4-1",
                "content": 1
              },
              {
                "box": "Box 1-4-2",
                "content": 0
              },
              {
                "box": "Box 1-4-3",
                "content": 1
              },
              {
                "box": "Box 1-4-4",
                "content": 0
              },
              {
                "box": "Box 1-4-5",
                "content": 0
              }
            ]
          }
        ]
      }
    ]
  }
},
```

Nested LLD Screenshot



Warehouse > Racks discovery > Shelves discovery > Boxes discovery

<input type="checkbox"/> Host	Name ▲	Items	Triggers	Graphs	Hosts	Discovery rules	Key	Interval	Type
<input type="checkbox"/> Warehouse	Shelves discovery: Boxes discovery	Item prototypes 1	Trigger prototypes 1	Graph prototypes	Host prototypes	Discovery prototypes	warehouse.bboxes.discovery[Rack 1,Shelf 1-1]		Nested
<input type="checkbox"/> Warehouse	Shelves discovery: Boxes discovery	Item prototypes 1	Trigger prototypes 1	Graph prototypes	Host prototypes	Discovery prototypes	warehouse.bboxes.discovery[Rack 1,Shelf 1-2]		Nested
<input type="checkbox"/> Warehouse	Shelves discovery: Boxes discovery	Item prototypes 1	Trigger prototypes 1	Graph prototypes	Host prototypes	Discovery prototypes	warehouse.bboxes.discovery[Rack 1,Shelf 1-3]		Nested
<input type="checkbox"/> Warehouse	Shelves discovery: Boxes discovery	Item prototypes 1	Trigger prototypes 1	Graph prototypes	Host prototypes	Discovery prototypes	warehouse.bboxes.discovery[Rack 1,Shelf 1-4]		Nested
<input type="checkbox"/> Warehouse	Shelves discovery: Boxes discovery	Item prototypes 1	Trigger prototypes 1	Graph prototypes	Host prototypes	Discovery prototypes	warehouse.bboxes.discovery[Rack 2,Shelf 2-1]		Nested
<input type="checkbox"/> Warehouse	Shelves discovery: Boxes discovery	Item prototypes 1	Trigger prototypes 1	Graph prototypes	Host prototypes	Discovery prototypes	warehouse.bboxes.discovery[Rack 2,Shelf 2-2]		Nested
<input type="checkbox"/> Warehouse	Shelves discovery: Boxes discovery	Item prototypes 1	Trigger prototypes 1	Graph prototypes	Host prototypes	Discovery prototypes	warehouse.bboxes.discovery[Rack 2,Shelf 2-3]		Nested
<input type="checkbox"/> Warehouse	Shelves discovery: Boxes discovery	Item prototypes 1	Trigger prototypes 1	Graph prototypes	Host prototypes	Discovery prototypes	warehouse.bboxes.discovery[Rack 2,Shelf 2-4]		Nested
<input type="checkbox"/> Warehouse	Shelves discovery: Boxes discovery	Item prototypes 1	Trigger prototypes 1	Graph prototypes	Host prototypes	Discovery prototypes	warehouse.bboxes.discovery[Rack 3,Shelf 3-1]		Nested
<input type="checkbox"/> Warehouse	Shelves discovery: Boxes discovery	Item prototypes 1	Trigger prototypes 1	Graph prototypes	Host prototypes	Discovery prototypes	warehouse.bboxes.discovery[Rack 3,Shelf 3-2]		Nested
<input type="checkbox"/> Warehouse	Shelves discovery: Boxes discovery	Item prototypes 1	Trigger prototypes 1	Graph prototypes	Host prototypes	Discovery prototypes	warehouse.bboxes.discovery[Rack 3,Shelf 3-3]		Nested
<input type="checkbox"/> Warehouse	Shelves discovery: Boxes discovery	Item prototypes 1	Trigger prototypes 1	Graph prototypes	Host prototypes	Discovery prototypes	warehouse.bboxes.discovery[Rack 3,Shelf 3-4]		Nested
<input type="checkbox"/> Warehouse	Warehouse data: Racks discovery	Item prototypes	Trigger prototypes	Graph prototypes	Host prototypes	Discovery prototypes 1	warehouse.racks.discovery		Dependent item
<input type="checkbox"/> Warehouse	Racks discovery: Shelves discovery	Item prototypes	Trigger prototypes	Graph prototypes	Host prototypes	Discovery prototypes 1	warehouse.shelves.discovery["Rack 1"]		Nested
<input type="checkbox"/> Warehouse	Racks discovery: Shelves discovery	Item prototypes	Trigger prototypes	Graph prototypes	Host prototypes	Discovery prototypes 1	warehouse.shelves.discovery["Rack 2"]		Nested
<input type="checkbox"/> Warehouse	Racks discovery: Shelves discovery	Item prototypes	Trigger prototypes	Graph prototypes	Host prototypes	Discovery prototypes 1	warehouse.shelves.discovery["Rack 3"]		Nested

Nested LLD result



<input type="checkbox"/> Host	Name ▲	Last check	Last value	Change
<input type="checkbox"/> Warehouse	Box content [Rack 1/Shelf 1-1/Box 1-1-1]	16s	0 Box	-1 Box
<input type="checkbox"/> Warehouse	Box content [Rack 1/Shelf 1-1/Box 1-1-2]	16s	0 Box	
<input type="checkbox"/> Warehouse	Box content [Rack 1/Shelf 1-1/Box 1-1-3]	16s	1 Box	
<input type="checkbox"/> Warehouse	Box content [Rack 1/Shelf 1-1/Box 1-1-4]	16s	0 Box	
<input type="checkbox"/> Warehouse	Box content [Rack 1/Shelf 1-1/Box 1-1-5]	16s	1 Box	
<input type="checkbox"/> Warehouse	Box content [Rack 1/Shelf 1-2/Box 1-2-1]	16s	0 Box	
<input type="checkbox"/> Warehouse	Box content [Rack 1/Shelf 1-2/Box 1-2-2]	16s	1 Box	
<input type="checkbox"/> Warehouse	Box content [Rack 1/Shelf 1-2/Box 1-2-3]	16s	0 Box	
<input type="checkbox"/> Warehouse	Box content [Rack 1/Shelf 1-2/Box 1-2-4]	16s	1 Box	
<input type="checkbox"/> Warehouse	Box content [Rack 1/Shelf 1-2/Box 1-2-5]	16s	0 Box	
<input type="checkbox"/> Warehouse	Box content [Rack 1/Shelf 1-3/Box 1-3-1]	16s	1 Box	
<input type="checkbox"/> Warehouse	Box content [Rack 1/Shelf 1-3/Box 1-3-2]	16s	0 Box	
<input type="checkbox"/> Warehouse	Box content [Rack 1/Shelf 1-3/Box 1-3-3]	16s	1 Box	
<input type="checkbox"/> Warehouse	Box content [Rack 1/Shelf 1-3/Box 1-3-4]	16s	0 Box	
<input type="checkbox"/> Warehouse	Box content [Rack 1/Shelf 1-3/Box 1-3-5]	16s	1 Box	+1 Box
<input type="checkbox"/> Warehouse	Box content [Rack 1/Shelf 1-4/Box 1-4-1]	16s	0 Box	
<input type="checkbox"/> Warehouse	Box content [Rack 1/Shelf 1-4/Box 1-4-2]	16s	1 Box	
<input type="checkbox"/> Warehouse	Box content [Rack 1/Shelf 1-4/Box 1-4-3]	16s	0 Box	
<input type="checkbox"/> Warehouse	Box content [Rack 1/Shelf 1-4/Box 1-4-4]	16s	1 Box	+1 Box

Official Zabbix template



The first official Zabbix template with nested LLD has been developed:

- ▶ Template **Proxmox VE by HTTP**
- ▶ Will be officially available with Zabbix 8.0
- ▶ <https://support.zabbix.com/browse/ZBXNEXT-10051>

<input type="checkbox"/> Template	Name ▲	Items	Triggers	Graphs	Hosts	Discovery rules	Key	Interval	Type	Status
<input type="checkbox"/>	Proxmox VE by HTTP	Cluster: Get resources: Proxmox LXC discovery	Item prototypes 18	Trigger prototypes 7	Graph prototypes 6	Host prototypes	Discovery prototypes	proxmox_ve.node.lxc.discovery[{#NODE.ID}]	Dependent item	Enabled
<input type="checkbox"/>	Proxmox VE by HTTP	Cluster: Get node data: Proxmox nodes discovery	Item prototypes 20	Trigger prototypes 9	Graph prototypes 4	Host prototypes	Discovery prototypes 3	proxmox_ve.nodes.discovery	Dependent item	Enabled
<input type="checkbox"/>	Proxmox VE by HTTP	Cluster: Get resources: Proxmox QEMU discovery	Item prototypes 17	Trigger prototypes 7	Graph prototypes 4	Host prototypes	Discovery prototypes 3	proxmox_ve.node.qemu.discovery[{#NODE.ID}]	Dependent item	Enabled
<input type="checkbox"/>	Proxmox VE by HTTP	Cluster: Get resources: Proxmox shared storage discovery	Item prototypes 5	Trigger prototypes	Graph prototypes	Host prototypes	Discovery prototypes	proxmox_ve.shared.storage.discovery	Dependent item	Enabled

All templates / Proxmox VE by HTTP

Discovery list / Proxmox nodes discovery

Item prototypes 20

Trigger prototypes 9

Graph prototypes 4

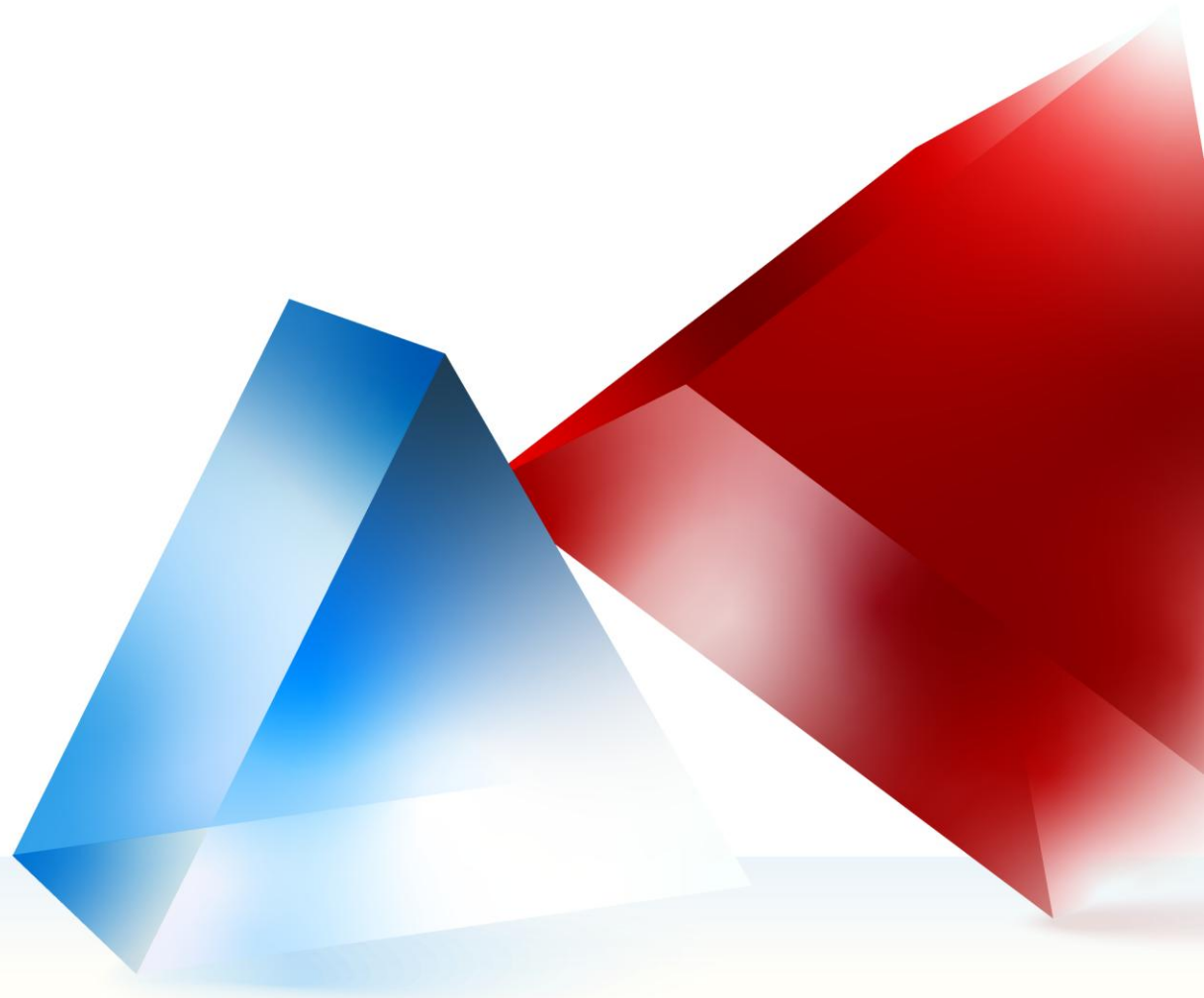
Host prototypes

Discovery prototypes 3

<input type="checkbox"/> Name ▲	Items	Triggers	Graphs	Hosts	Discovery rules	Key	Interval	Type	Create enabled	Discover
<input type="checkbox"/> Node [{#NODE.NAME}]: Certificate discovery	Item prototypes 6	Trigger prototypes 1	Graph prototypes	Host prototypes	Discovery prototypes	proxmox_ve.node.certificates.discovery[{#NODE.ID}]	{ \$PVE.PARAMS.INTERVAL.CERT }	HTTP agent	Yes	Yes
<input type="checkbox"/> Node [{#NODE.NAME}]: Disks discovery	Item prototypes 6	Trigger prototypes 1	Graph prototypes	Host prototypes	Discovery prototypes	proxmox_ve.node.disks.discovery[{#NODE.ID}]	{ \$PVE.PARAMS.INTERVAL.DISK }	HTTP agent	Yes	Yes
<input type="checkbox"/> Node [{#NODE.NAME}]: Storage discovery	Item prototypes 6	Trigger prototypes 2	Graph prototypes 1	Host prototypes	Discovery prototypes	proxmox_ve.node.storage.discovery[{#NODE.ID}]	{ \$PVE.PARAMS.INTERVAL.STORAGE }	HTTP agent	Yes	Yes

Displaying 3 of 3 found

Workshop info



Workshop



13:00

Nested LLDs

60 m

Aleksandrs Petrovs-
Gavrilovs, Zabbix Trainer,
Zabbix

Alfa

Technical

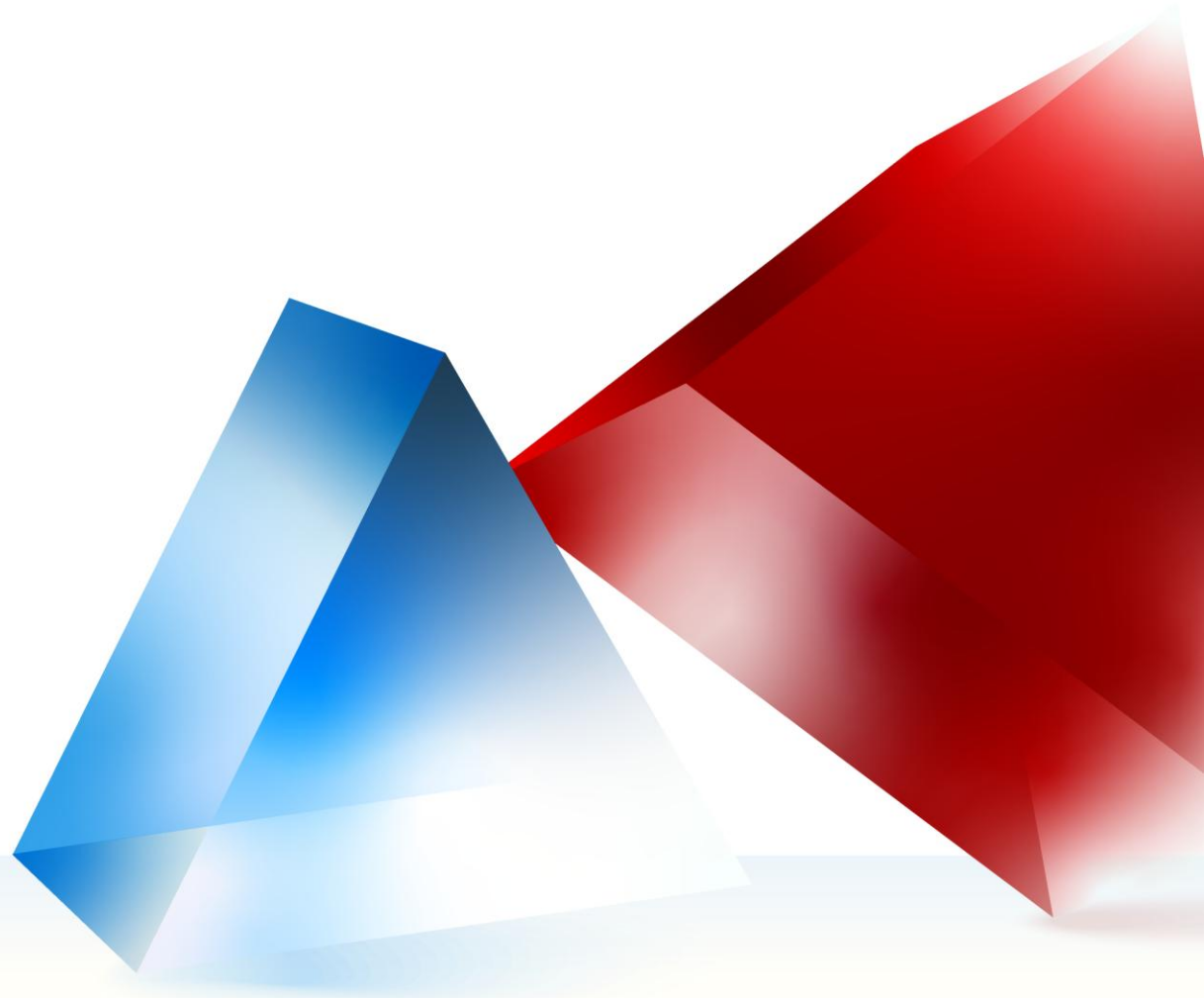
We'll guide you through a scenario focused on discovering and monitoring a warehouse, including:

- ▶ Multiple racks
- ▶ Multiple shelves per rack
- ▶ Multiple storage boxes per shelf

You must **bring your own laptop** to the workshop:

- ▶ Individual virtual machines will be provided by Zabbix

Training info



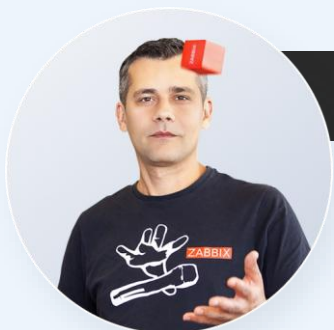
New one-day course



A new **one-day course** on Low Level Discovery (LLD) is coming soon!

- ▶ Available as both [live training](#) and a [Zabbix Academy](#) course
- ▶ Covers every aspect of LLD, from the [basics](#) to [dependent LLD](#) and [nested LLD](#)
- ▶ Learn how to create hosts from [host prototypes](#)
- ▶ Discover how to transform [raw data](#) into [JSON](#) for LLD discovery of nearly anything
- ▶ Includes live examples with [Zabbix Agent](#), [SNMP](#), and [database](#) discovery

Thank You!



Kaspars Mednis

Training Project Manager