

The ZABBIX logo consists of the word "ZABBIX" in white, uppercase, sans-serif font, centered within a solid red rectangular background. The background of the entire slide is a dark blue gradient with a faint, glowing network map of the world, where nodes are connected by thin lines and some nodes are highlighted with small blue circles.

ZABBIX

JSON item value type

Artjoms Rimdjonoks

C Developer

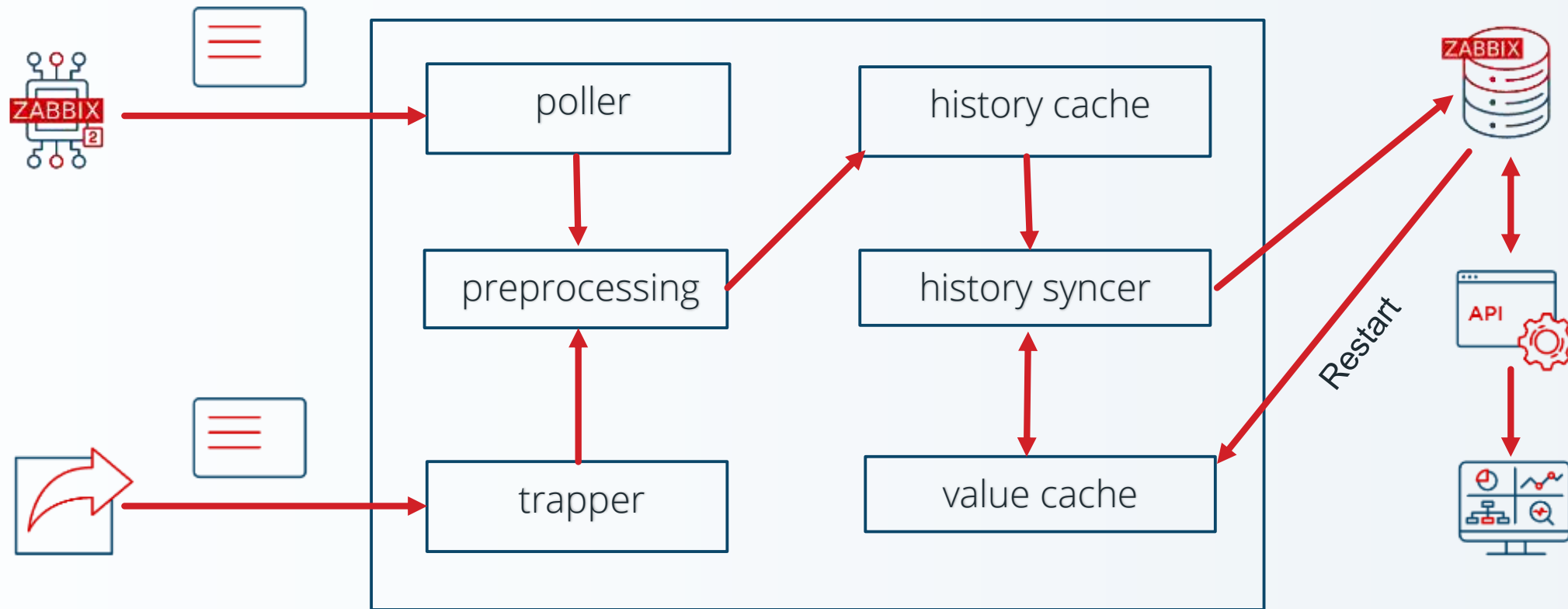
Existing value types

- Numeric (unsigned)
- Numeric (float)
- Character - short text (255 chars limit)
- Text - long text (65536 bytes/characters limit)
- Log - same as Text, but extra logic/metadata in code for log processing (to avoid rereading entire file)

+ Binary item value type

- since 7.0
- very specific use-case – web monitoring
- dependent items only
- stores screenshots of web-pages
- useful for reporting the rendered web-page
- if one of the checks failed (e.g. no certain html element)
- BLOB (~1GB)

Why there are different item value types in Zabbix?



Which components really use value types ?

Zabbix agent ?



- does not really need value type
- validation is done **after** preprocessing
- if value type is uint, but text "badger" is received, this could be translated to value 6, (if preprocessing counts the number of letters)
- for Log item value type – agent needs to use **last log line** (instead of rereading the file over again)

Which components really use value types ?

preprocessing (Zabbix server/Zabbix proxy)

- validation **after** preprocessing (including the max size limit check)

history syncer (Zabbix server)

- evaluate trigger expressions

DB

- store history in dedicated tables

Which components really use value types ?

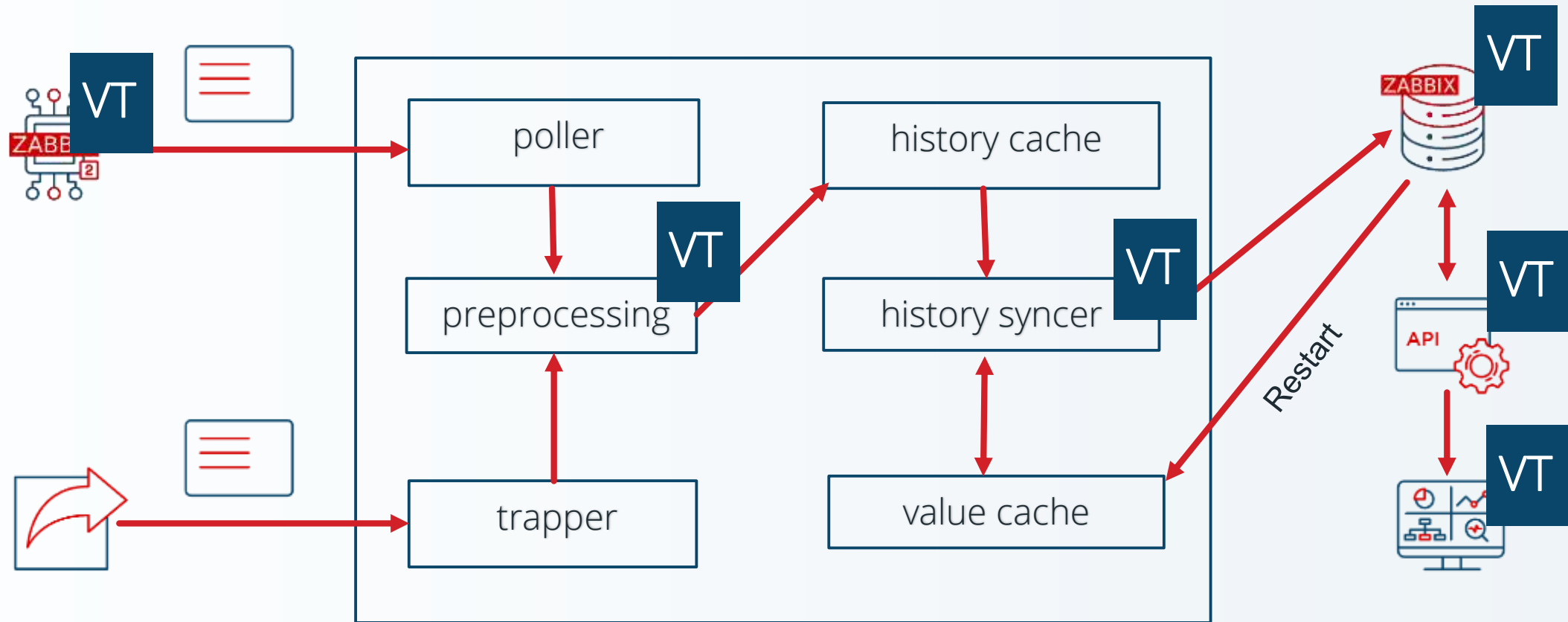
API

- validation (history.push)

UI

- vizualization (need to know if history is numeric to draw a graph)

Why there are different item value types ?



Monitoring vs Observability

Monitoring

- collect and analyze what is explicitly configured
- configuration can be extended with automation like LLD

Observability

- collect much more data than it is immediately needed
- store it efficiently (aggregate, index etc.)
- analyse it further later when necessary

Monitoring vs Observability

- Key distinction – **amount of data** collected (lots of JSON)
- ~100MBs (APM or NetFlow)
- JSON is well integrated into DBs and Programming languages

- Application Performance Monitoring server
- returns JSON (part ElasticSearch stack)
- ~10MB

```
{
  "@timestamp": "2025-11-20T12:15:00Z",
  "service": { "name": "orders-api" },
  "host": { "hostname": "node-1" },
  "system": {
    "process": {
      "cpu": { "total.norm.pct": 0.47 }
    }
  },
  "metricset": { "name": "system.process" }
}
...
```

Netflow collector

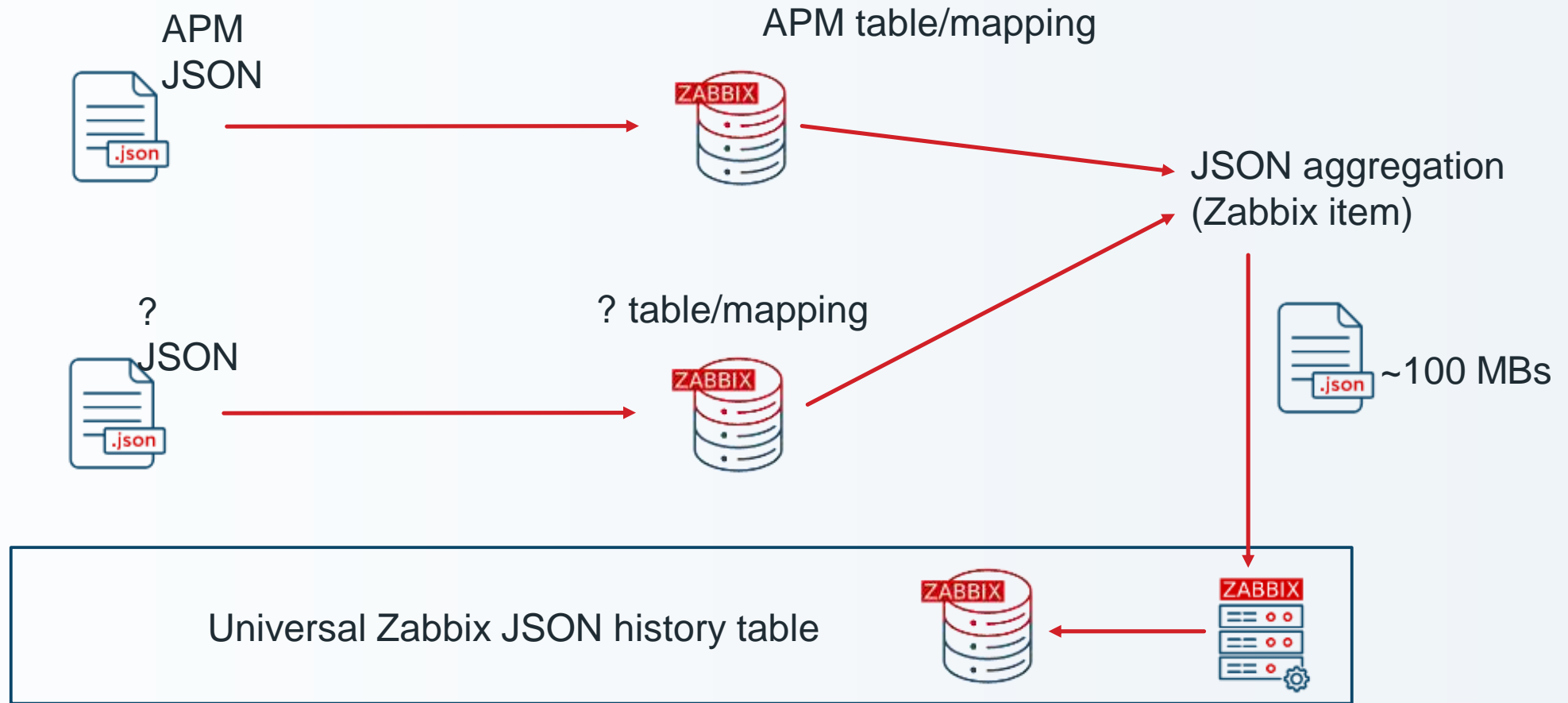
- traffic analysis
- returns JSON, to make it easier to manipulate or store in Elastic
- ~100 MBs

```
{  
  "src_ip": "192.168.1.10",  
  "dst_ip": "8.8.8.8",  
  "src_port": 55321,  
  "dst_port": 53,  
  "protocol": "UDP",  
  "bytes": 240,  
  "packets": 3,  
  "timestamp": "2025-11-20T12:00:01Z"  
}  
...
```

- there is no universal and efficient JSON storage
- observability JSON collectors provide their own **unique** JSON mapping
- need to index data efficiently (~schema for JSON)
- **dynamic mapping** has its own limitations (mapping explosion)

Planned Observability flow

- Zabbix needs to aggregate data from different observability points.
- And then store the resulting JSON in history.



Does zabbix support large JSON ?

- Numeric (unsigned)
- Numeric (float)
- Character (255)
- Text (65536)
- Log (65536)
- Binary (BLOB, only dependent items)

We also want to validate JSON input history (and discard invalid entries ASAP)

Be available for all Zabbix item types – (agent, script, trapper etc.)

Does zabbix support large JSON ?

Text

update the limits

+

add preprocessing step manually for JSON ?

Binary

expand to other item types

+

add preprocessing step manually for JSON ?

DB JSON type

- build-in, dedicated, native
- PostgreSQL – **jsonb**
- MySQL – **json**
- In-build JSON validation (not that important to us)
- Binary (efficient storage)
- Efficient search within JSON (better than text)



DB JSON type

cpu_data table with text hostname and jsonb metrics

```
id | hostname |          metrics
----+-----+-----
 1 | server1 | {"idle": 64.3, "user": 25.5, "system": 10.2, ...many MBs of JSON... }
 2 | server2 | {"idle": 5.0, "user": 85.0, "system": 10.0, ...many MBs of JSON... }
...
```

```
zabbix=> select hostname from cpu_data where (metrics->>'user')::numeric > 80;
hostname
-----
server2
```

Elastic JSON type

```
"mappings": {  
  "dynamic": false,  
  "properties": {  
    "itemid": {  
      "type": "long"  
    },  
    "clock": {  
      "format": "epoch_second",  
      "type": "date"  
    },  
    "value": {  
      "type": "object"  
      "dynamic": true,  
      "properties": {}  
    }  
  }  
}
```

New JSON item value type

ZBXNEXT-10069

- entirely new Item Value type
- validates JSON automatically, (after preprocessing)
- has increased limits (preprocessing validation)
- stores data into DB as JSON native

JSON item value type

New item

Item **Tags** Preprocessing

* Name

Type

* Key

Type of information

Parameters

* Script

* Update interval

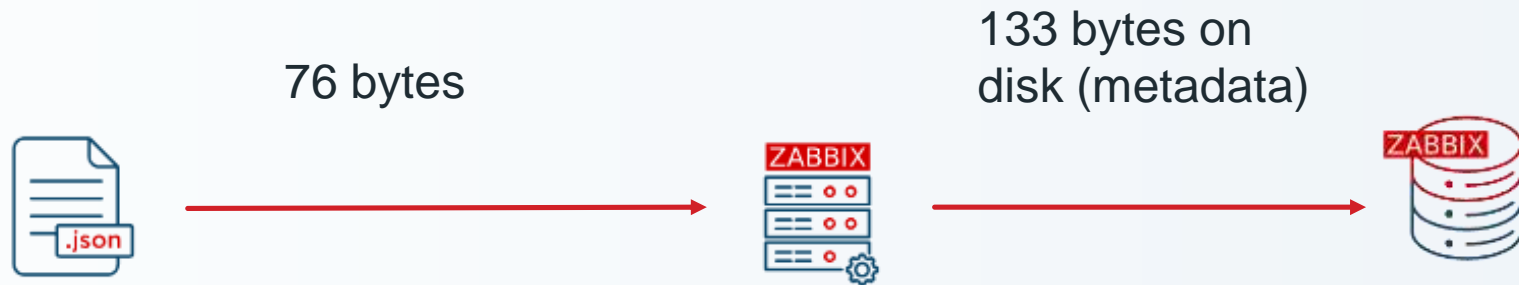
Value

- Numeric (unsigned)
- Numeric (float)
- Character
- Log
- Text
- JSON

Limit

How much maximum history limit should be increased ?

- jsonb on disk takes more space for smaller values



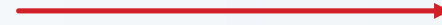
Limit

However, on larger scale, it is the opposite:

183 MB



28 MB



So, can we store 1GB of JSON ?

Limit PostgreSQL

No..

- there seems to be no **official** PostgreSQL maximum supported limit for jsonb
- the limitation factor is not the space that jsonb takes on disk - but the **PostgreSQL internal limits (element array limit)** when processing values
- **224MB** of raw JSON generates the following error:
2025-07-22 11:11:55.911 UTC [16138] zabbix@zabbix ERROR: total size of jsonb array elements exceeds the maximum of 268435455 bytes at character 85
- **223MB** works without issues..

Limit PostgreSQL

- the closest real limit will be 128 MB
- entries above it will be discarded
- note, agent items that read files - `vfs.file.contents` may have their own limits (this is not updated)

Limit MySQL

- MySQL and MariaDB should handle 128 MB as well
- but its `/etc/mysql/my.cnf` may need to be updated:

```
[mysqld]  
max_allowed_packet=128M
```

```
[mysqldump]  
max_allowed_packet=1G
```

Notes

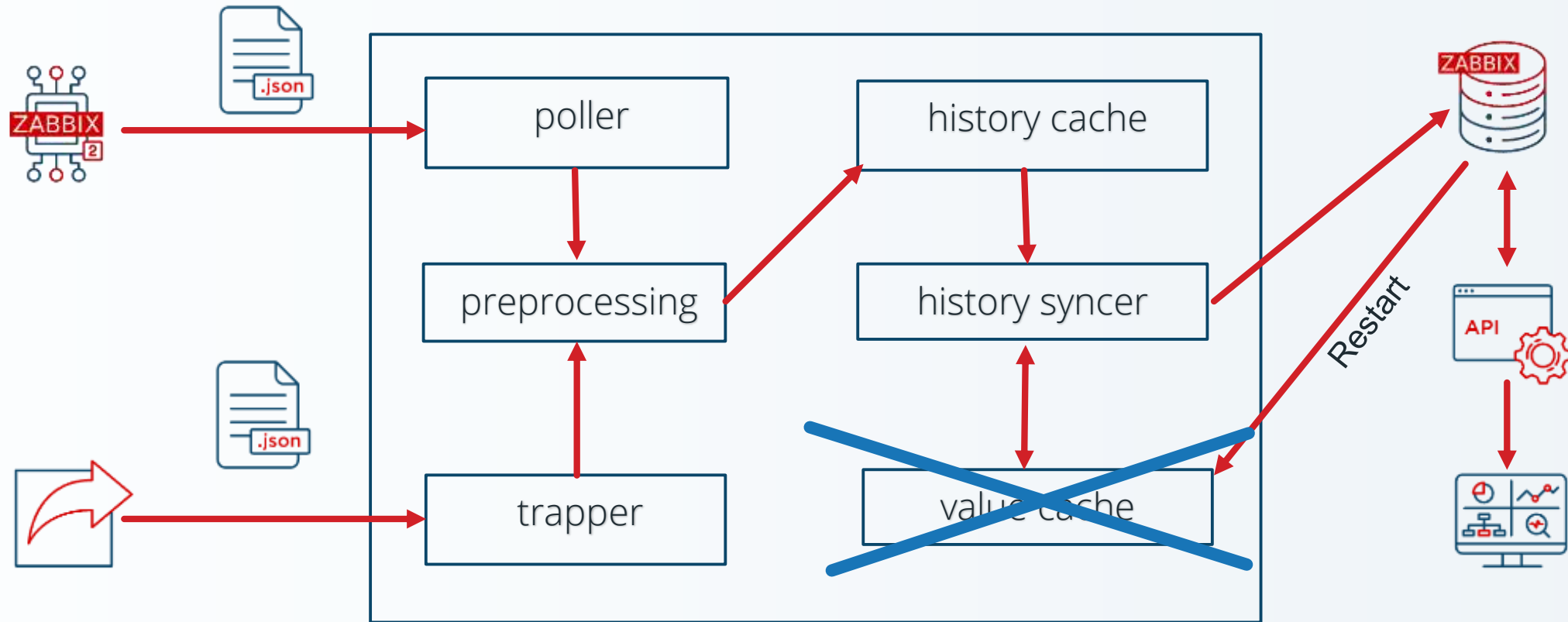
Value-cache

Throttling

MariaDB

Note 1 – Value cache is bypassed

- JSON item value type cannot be used in triggers or calculated items
- for triggers/calculated items – dependent items will need to be used



Note 2 – Throttling

Discard identical history value preprocessing step.

The screenshot shows the 'Preprocessing 1' configuration window in Zabbix. It features a table for 'Preprocessing steps' with one entry: '1: Discard unchanged'. Below the table is an 'Add' button and a 'Type of information' dropdown menu set to 'JSON'. On the right side, there are checkboxes for 'Custom on fail' (unchecked) and 'Actions' (with links for 'Test', 'Remove', and 'Test all steps'). At the bottom, there are buttons for 'Update', 'Clone', 'Execute now', 'Test', 'Clear history and trends', 'Delete', and 'Cancel'. Below these buttons is a 'Discovered' section with 'All', 'Yes', and 'No' options. At the very bottom, there are 'Apply' and 'Reset' buttons.

- for preprocessing to understand that it needs to discard the newly arrived value - it needs to compare it with previous one
- so, it needs to store the latest value in memory

Note 2 - Throttling

- 100 items, each returns 100MB of data, that means ~ 10GB of memory..
- we can store in memory only the hash and compare new values against it, drastically reducing the memory required
- will be done as a separate ticket.
<https://support.zabbix.com/browse/ZBXNEXT-10270>
"Improve throttling by using SHA hash"
- because the change needs to be applied to other types
lots of benchmarking needed

Note 3 – MariaDB

- MariaDB developers are convinced they do not need proper JSON support (<https://jira.mariadb.org/browse/MDEV-9144>)
- json in MariaDB is just text
- MariaDB developers suggest their JSON functions should be fast enough to convert text to json on the fly if JSON structure is needed..

Contact us

USA

Phone +1 877-4-ZABBIX
+1 877-4-922249 (Toll-free)
Email sales@zabbix.com

JAPAN

Phone +81 3-4405-7338
Email sales@zabbix.co.jp

LATIN AMERICA

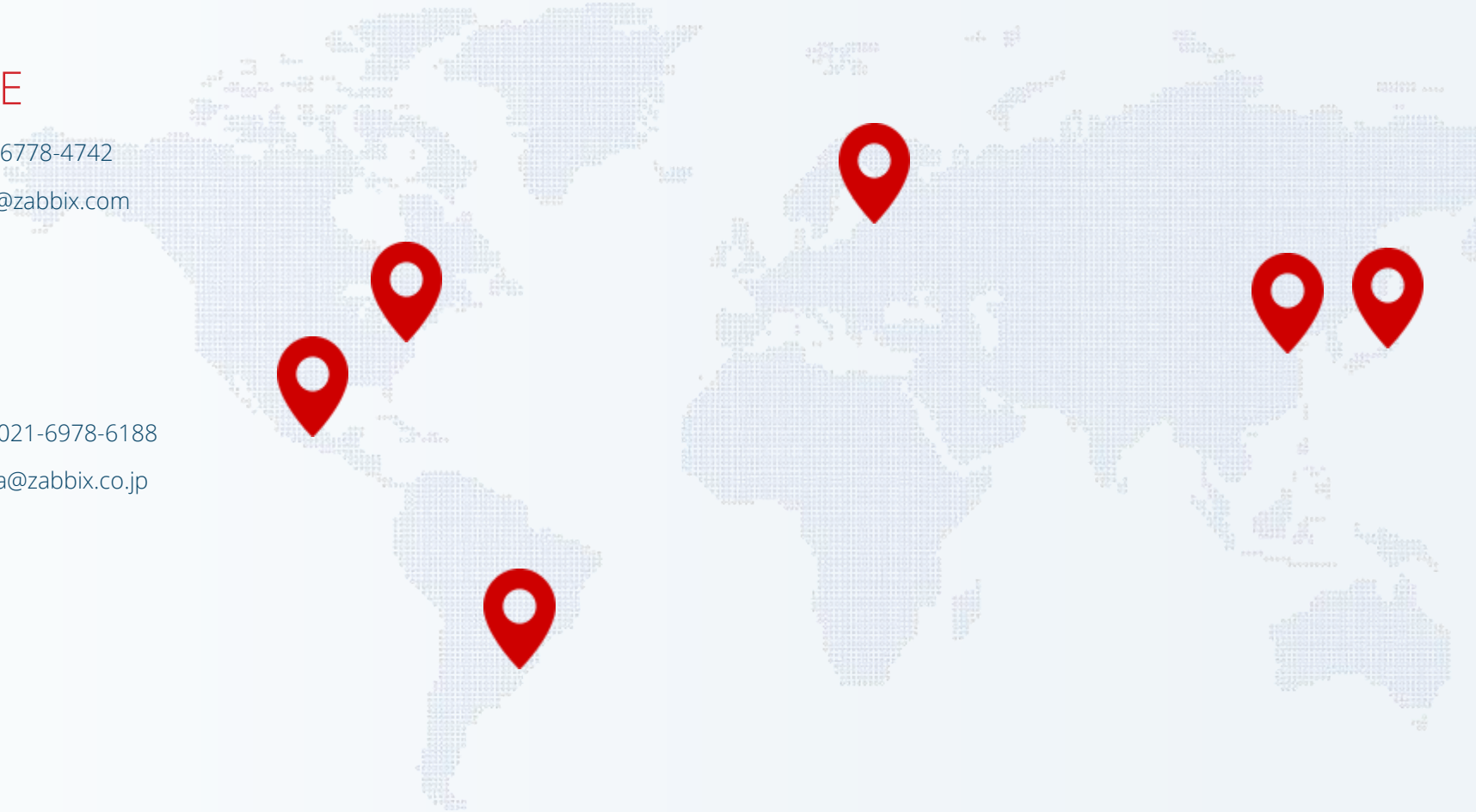
Phone Argentina | Buenos Aires +54 113989-4060
Brazil | San Paulo +55 11 4210-5104
Chile | National +56 44 890-9410
Colombia | Bogota +57 1 3819310
Mexico | Mexico city +52 55 8526-2606
Email sales.latam@zabbix.com

EUROPE

Phone +371 6778-4742
Email sales@zabbix.com

CHINA

Phone +86 021-6978-6188
Email china@zabbix.co.jp



Stay updated on Zabbix news:



Zabbix



zabbix_official



@zabbix



Zabbix

The ZABBIX logo consists of the word "ZABBIX" in a bold, white, sans-serif font, centered within a solid red rectangular background. The background of the entire slide is a dark blue gradient with a faint, glowing network map of the world, where nodes are connected by thin lines and some nodes are highlighted with small blue circles.

ZABBIX

Artjoms

Rimdjonoks