



The Enterprise class Monitoring Solution for Everyone

[WWW.ZABBIX.COM](http://WWW.ZABBIX.COM)

# Zabbix 3.0 Тренинг Сертифицированный Профессионал

День 1

# РАСПИСАНИЕ НА КАЖДЫЙ ДЕНЬ

## Четверг

**09.00-11.30 Zabbix 3.0  
Сертифицированный  
Профессионал**

11.30-11.45 Перерыв

**11.45-13.00 Zabbix 3.0  
Сертифицированный  
Профессионал**

13.00-14.00 Перерыв

**14.00-15.30 Zabbix 3.0  
Сертифицированный  
Профессионал**

15.30-15.45 Перерыв

**15.45-17.50 Zabbix 3.0  
Сертифицированный  
Профессионал**

## Пятница

**09.00-11.30 Zabbix 3.0  
Сертифицированный  
Профессионал**

11.30-11.45 Перерыв

**11.45-14.00 Zabbix 3.0  
Сертифицированный  
Профессионал**

14.00-15.00 Перерыв и  
время Q/A

**15.00-16.00  
Дополнительные темы**

**16.00-17.50  
Сертификация и выдача  
сертификатов**

# ПОВЕСТКА

Расширенн  
ый сбор  
данных



Мониторинг  
VMware



Пользователь  
ские макросы



Низкоуровне  
вое  
обнаружение



Распределен  
ный  
мониторинг



Дополнительн  
ые темы





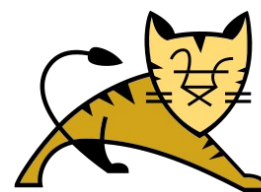
JAVA  
МОНИТОРИНГ

# МОНИТОРИНГ JAVA ПРИЛОЖЕНИЙ

Zabbix сервер



Zabbix Java gateway



Apache Tomcat



GlassFish



# JAVA GATEWAY

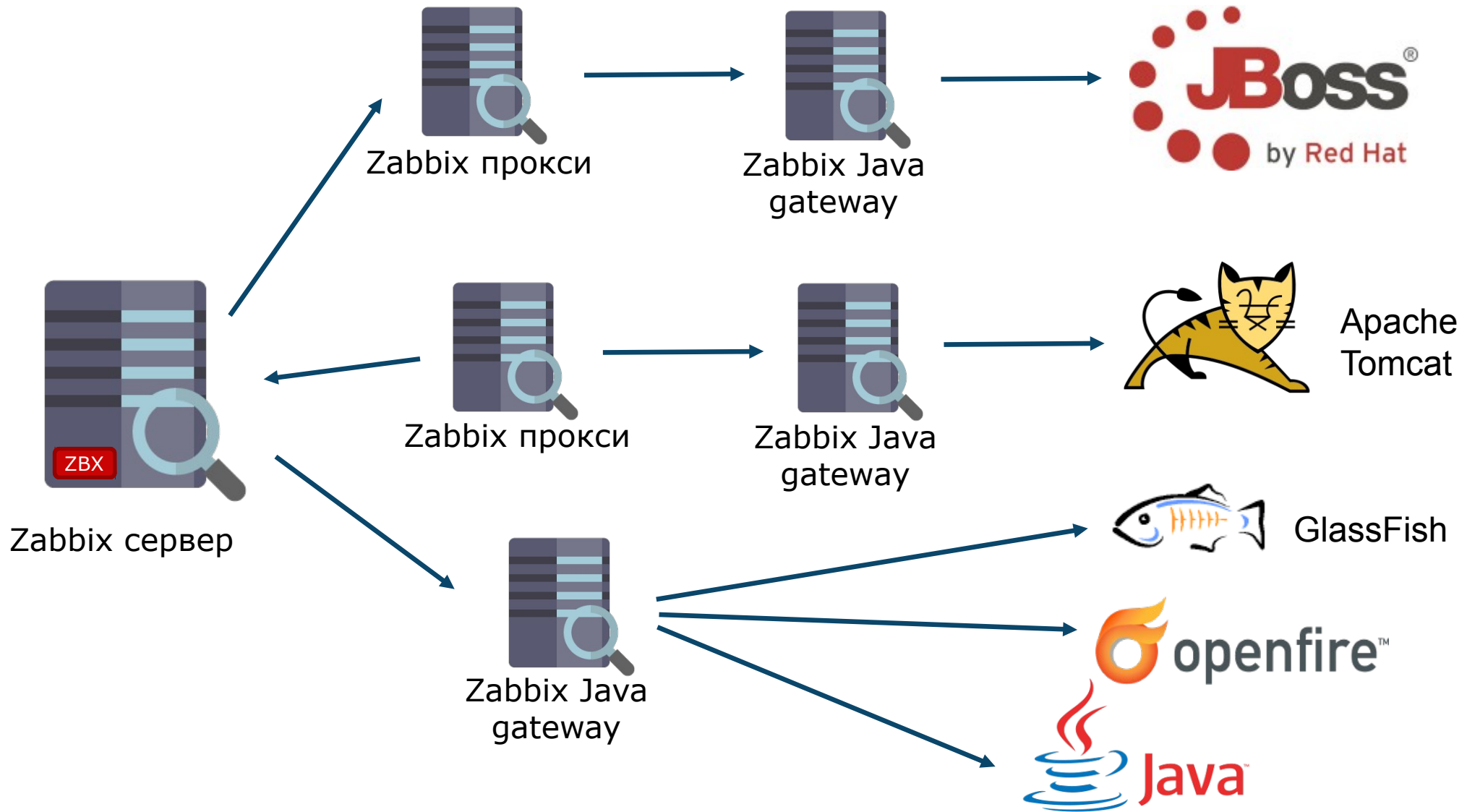
Требуется Java

Можно запустить на одной или на отдельной системе

Обрабатывается сервером

Пакетные запросы по причине производительности

# НЕСКОЛЬКО И УДАЛЕННЫЕ GATEWAY



# ПРАКТИЧЕСКАЯ УСТАНОВКА

Установите Tomcat

Установите Zabbix Java Gateway

Настройте Java Gateway для мониторинга Tomcat при помощи "Template JMX Generic"

Проверьте иконку состояния JMX агента и "Последние данные" на наличие значений





ODBC  
МОНИТОРИНГ

# ПОЛУЧЕНИЕ ДАННЫХ ИЗ ВНЕШНИХ БАЗ ДАННЫХ

Требуется подключение ODBC

Зависит от UnixODBC

`db.odbc.select[<уник. короткое описание>,<dsn>]`

Name	<input type="text" value="Number of online banking customers"/>		
Type	<input type="text" value="Database monitor"/>		
Key	<input type="text" value="db.odbc.select[num_customers,oracle02_onl_bank]"/>	<input type="button" value="Select"/>	
User name	<input type="text" value="zabbix_odbc"/>		
Password	<input type="text" value="Nmd4O!u6"/>		
SQL query	<input type="text" value="SELECT count(*) FROM customers"/>		
Type of information	<input type="text" value="Numeric (unsigned)"/>		



SNMP  
ТРАПЫ

# SNMP ТРАПЫ

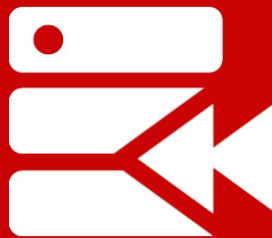
Работает с snmptrapd и одним из встроенных механизмов для передачи трапов - perl скрипт или SNMPTT

## Синтаксис

```
snmptrap[regex]  
snmptrap.fallback
```

## Примеры

```
snmptrap["SNMPv2-MIB::coldStart"]  
snmptrap[LineVoltageProblem]  
snmptrap["IF-MIB::(linkDown|linkUp)"]
```



SSH/TELNET  
МОНИТОРИНГ

# ПОЛУЧЕНИЕ ДАННЫХ С ВНЕШНЕГО УСТРОЙСТВА

Аутентификация на основе пароля или публичного ключа  
([SSHKeyLocation](#))

Возможность выполнения любой команды и получения результата выполнения на Zabbix

`ssh.run[<уник. короткое описание>,<ip>,<порт>,<кодировка>]`

Name	Status of MySQL server
Type	SSH agent ▼
Key	ssh.run[mysql.status]
Host interface	192.168.56.101 : 10050 ▼
Authentication method	Password ▼
User name	zabbix_ssh
Password	K1v4#au7
Executed script	/usr/bin/mysqladmin ping   grep -c alive

# ПРАКТИЧЕСКАЯ УСТАНОВКА

Добавьте SSH элемент данных с аутентификацией по паролю в "Template Basic":

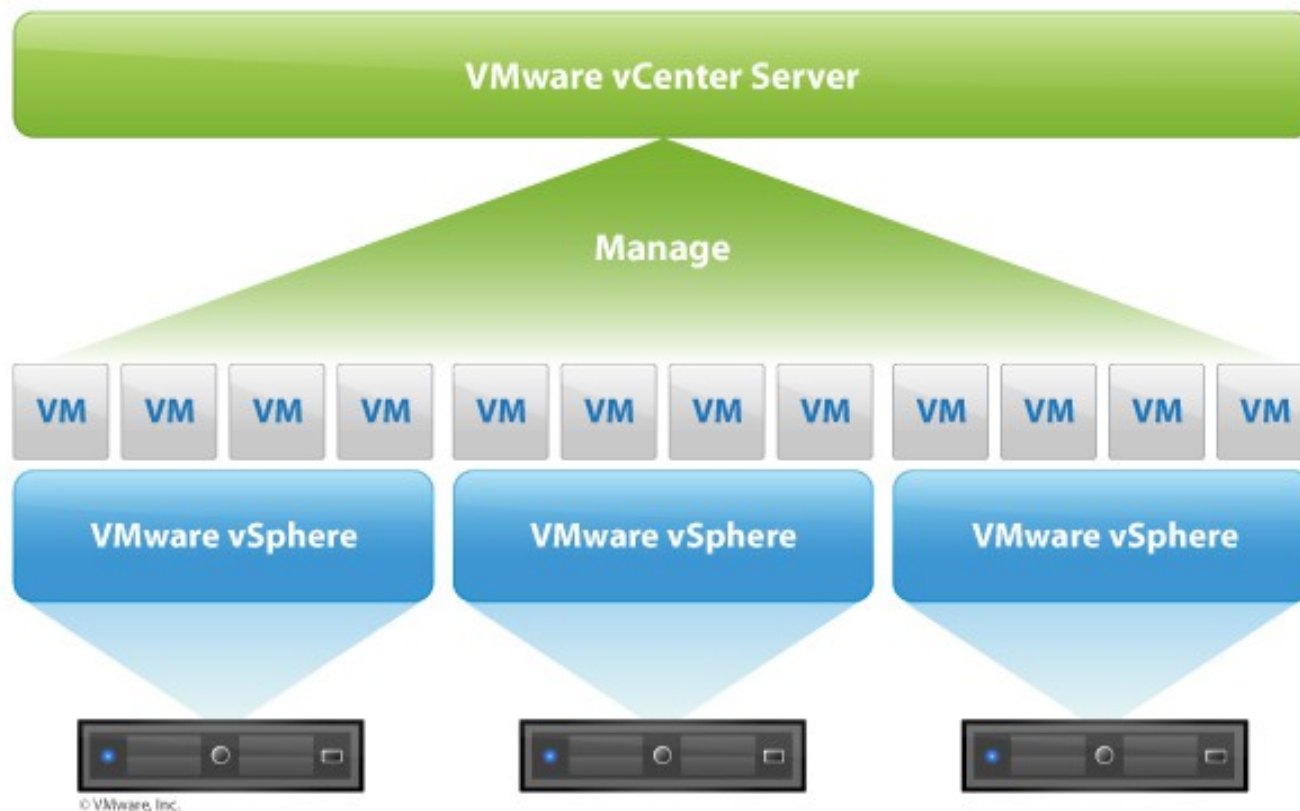
"Zabbix server daemon version"

Убедитесь, что элемент данных получает данные





# МОНИТОРИНГ VMWARE



Мониторинг vCenter и vSphere

Автоматическое обнаружение гипервизоров и гостей VM

Поддержка прототипов узлов сети; возможно расширить для XEN, KVM, Linux Containers и т.п.

# МОНИТОРИНГ VMWARE

- ✓ Работает из коробки
- ✓ Не требует какие-либо сторонние утилиты
- ✓ Основан на родном VMWare API
- ✓ Оптимизирован на выполнение как можно меньших запросов API насколько возможно
- ✓ Данные настроек и производительности являются двумя отдельными запросами (2.4)

# ГОТОВЫЕ К ИСПОЛЬЗОВАНИЮ ШАБЛОНЫ

Template Virt **VMWare**

Template Virt **VMWare Guest**

Template Virt **VMWare Hypervisor**



ПОЛЬЗОВАТЕЛЬСКИЕ  
МАКРОСЫ

# ФУНКЦИОНАЛ ПОЛЬЗОВАТЕЛЬСКОГО МАКРОСА

## Лёгкое обслуживание – один шаблон и:

разные параметров ключей элементов данных

`net.tcp.service[ssh,{$SSH_PORT}]`

разные значения выражений триггеров

`{server:system.cpu.load[,avg1].last(0)} > {$CPU_LOAD}`

## В порядке убывания – приоритет:

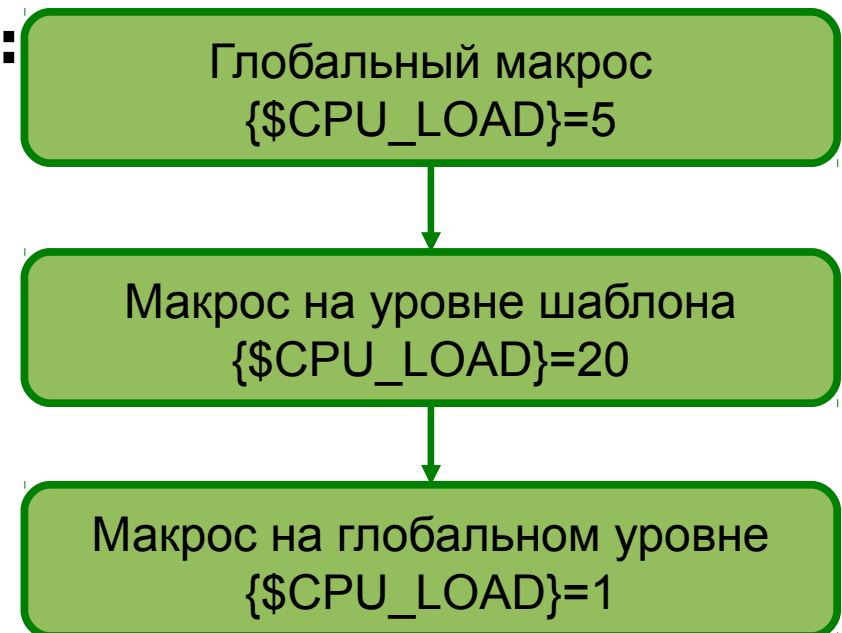
Макрос на уровне узла сети

Макрос на уровне шаблона

Макрос на глобальном уровне

## Синтаксис:

`{$ИМЯ}`



# НАСТРОЙКА

## ПОЛЬЗОВАТЕЛЬСКИХ МАКРОСОВ

Глобальный: Администрирование → Общие → Макросы

Macros

MACRO		VALUE	
<input type="text" value="{ \$SNMP_COMMUNITY }"/>	⇒	<input type="text" value="public"/>	<a href="#">Remove</a>
<input type="text" value="{ \$CPU_LOAD }"/>	⇒	<input type="text" value="5"/>	<a href="#">Remove</a>
<input type="text" value="{ \$FREE_DISK_SPACE }"/>	⇒	<input type="text" value="10"/>	<a href="#">Remove</a>
<a href="#">Add</a>			

Узел сети: Свойства узла сети и шаблона

Hosts

All hosts /  Enabled ZBX SNMP JMX IPMI Applications 4 Items 6 Triggers Graphs Discovery rules 3 Web scenarios

Host Templates IPMI Macros Host inventory Encryption

Host macros Inherited and host macros

MACRO		EFFECTIVE VALUE		TEMPLATE VALUE		GLOBAL VALUE (CONFIGURE)
<input type="text" value="{ \$SNMP_COMMUNITY }"/>	⇒	<input type="text" value="Hk32yT@o#s"/>	<a href="#">Remove</a>			⇐ "public"
<input type="text" value="{ \$SSH_PORT }"/>	⇒	<input type="text" value="22"/>	<a href="#">Change</a>	⇐ Template SNMP OS Linux: "22"		
<a href="#">Add</a>						

# ПОДДЕРЖКА КОНТЕКСТА В ПОЛЬЗОВАТЕЛЬСКИХ МАКРОСАХ

Контекст макроса это текстовое значение

Примером использования контекстов макросов может быть использование значения макроса LLD

Пример:

```
{ca_001:vfs.fs.size[{#FSNAME},pfree].last()} <  
{ $LOW_SPACE_LIMIT: {#FSNAME} }
```

где:

```
{ $LOW_SPACE_LIMIT } 10  
{ $LOW_SPACE_LIMIT: "/opt" } 25
```

События будут созданы, когда "/" и "/home" будут иметь меньше чем 10% свободного места или "/opt" файловая система меньше чем 25%.



НИЗКОУРОВНЕВОЕ  
ОБНАРУЖЕНИЕ



# ВСТРОЕННЫЕ ОБЪЕКТЫ LLD

- ✓ Агент – файловые системы
  - ✓ Агент – сетевые интерфейсы
  - ✓ Agent – службы Windows
  - ✓ SQL объекты
  - ✓ SNMP объекты
- ...всё-что угодно при использовании скриптов

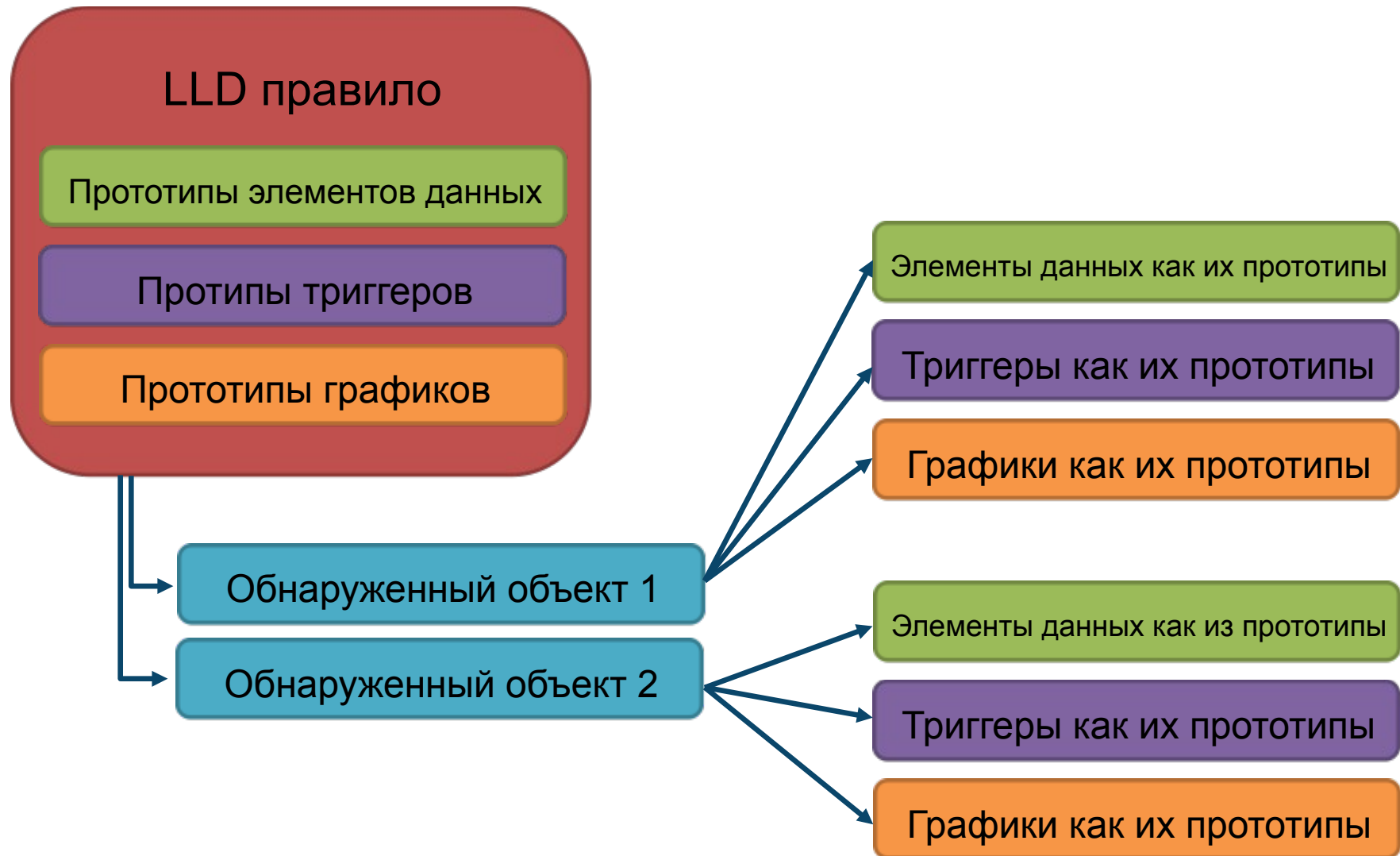
# КОМПОНЕНТЫ LLD

## LLD правило

- ✓ Прототипы элементов данных
- ✓ Прототипы триггеров
- ✓ Прототипы графиков
- ✓ Прототипы узлов сети

<input type="checkbox"/> NAME ▲	ITEMS	TRIGGERS	GRAPHS	HOSTS	KEY	INTERVAL	TYPE	STATUS	INFO
<input type="checkbox"/> <a href="#">Template OS Linux: Mounted filesystem discovery</a>	<a href="#">Item prototypes 5</a>	<a href="#">Trigger prototypes 2</a>	<a href="#">Graph prototypes</a>	<a href="#">Host prototypes</a>	vfs.fs.discovery	1h	Zabbix agent	Enabled	
<input type="checkbox"/> <a href="#">Template OS Linux: Network interface discovery</a>	<a href="#">Item prototypes 2</a>	<a href="#">Trigger prototypes</a>	<a href="#">Graph prototypes 1</a>	<a href="#">Host prototypes</a>	net.if.discovery	1h	Zabbix agent	Enabled	
Displaying 2 of 2 found									
0 selected <input type="button" value="Enable"/> <input type="button" value="Disable"/> <input type="button" value="Delete"/>									

# КОМПОНЕНТЫ LLD



# ОБНАРУЖЕНИЕ СЕТЕВЫХ УСТРОЙСТВ

## Правило обнаружения сетевых интерфейсов

Прототип элемента данных:  
входящий трафик

Прототип элемента данных:  
исходящий трафик

Прототип триггера:  
высокий входящий трафик

Прототип триггера:  
высокий исходящий трафик

Прототип графика:  
трафик на интерфейсе

Обнаружен eth0

Элемент данных:  
входящий трафик

Элемент данных:  
исходящий трафик

Триггер:  
высокий входящий трафик

Триггер:  
высокий исходящий трафик

График:  
трафик на интерфейсе

Обнаружен eth1

Элемент данных:  
входящий трафик

Элемент данных:  
исходящий трафик

Триггер:  
высокий входящий трафик

Триггер:  
высокий исходящий трафик

График:  
трафик на интерфейсе

# СОЗДАНИЕ ПРОТОТИПОВ

Правила LLD возвращают данные в переменных (макросах):

Диски: `{#FSNAME}`, `{#FSTYPE}`

Интерфейсы: `{#IFNAME}`

CPU: `{#CPU.NUMBER}`, `{#CPU.STATUS}`

SNMP: `{#SNMPINDEX}`, `{#SNMPVALUE}`, ...

ODBC: имена колонок становятся именами макросов

Службы Windows: `{#SERVICE.NAME}`, `{#SERVICE.STATE}`, ...

Пример ключа:

`vfs.fs.size[{#FSNAME},free]`

LLD макросы можно использовать в выражениях триггеров (2.2)

`{Template_OS_Linux:vfs.fs.size[{#FSNAME},pused].last(0)}>{#MY_CUSTOM_MACRO}`

# СВОЙСТВА LLD ПРАВИЛА

Можно использовать любой тип "элемента данных" для сбора данных

Учитывается интервал обновления

Фильтрация по регулярному выражению

Discovery rule

Filters

Name

Network interface discovery

Type

Zabbix agent

Key

net.if.discovery

Update interval (in sec)

3600

Custom intervals

TYPE	INTERVAL	PERIOD	ACTION
<div>Flexible</div> <div>Scheduling</div>	50	1-7,00:00-24:00	<a href="#">Remove</a>
<a href="#">Add</a>			

Keep lost resources period (in days)

30

Description

Discovery of network interfaces as defined in global regular expression "Network interfaces for discovery".

Enabled

☒

# ЗАВИСИМОСТИ МЕЖДУ ПРОТОТИПАМИ ТРИГГЕРОВ

## Trigger prototypes

[All templates](#) / [Template OS Linux](#) [Discovery list](#) / [Mounted filesystem discovery](#) [Item prototype](#)

☐ SEVERITY NAME ▲

☐ Average Free disk space is less than 10% on volume {#FSNAME}

☐ Warning Free disk space is less than 20% on volume {#FSNAME}  
**Depends on:**  
Template OS Linux: Free disk space is less than 10% on volume {#FSNAME}

☐ Warning Free inodes is less than 20% on volume {#FSNAME}

0 selected

Enable

Disable

Mass update

Delete

# СОЗДАННЫЕ ОБЪЕКТЫ

Обозначаются в конфигурации

<u>Mounted filesystem discovery</u> : Free disk space on /boot		vfs.fs.size[/boot,free]
<u>Mounted filesystem discovery</u> : Free disk space on /		vfs.fs.size[/,free]
<u>Mounted filesystem discovery</u> : Free disk space on /boot (percentage)	Triggers 1	vfs.fs.size[/boot,pfree]
<u>Mounted filesystem discovery</u> : Free disk space on / (percentage)	Triggers 1	vfs.fs.size[/,pfree]

<input type="checkbox"/>	Average	<u>Mounted filesystem discovery</u> : Free disk space is less than 20% on volume /	{Zabbix server:vfs.fs.size[/,pfree].last(0)}<20
<input type="checkbox"/>	Average	<u>Mounted filesystem discovery</u> : Free disk space is less than 20% on volume /boot	{Zabbix server:vfs.fs.size[/boot,pfree].last(0)}<20



# ОБЪЕКТЫ К УДАЛЕНИЮ

Обозначаются в конфигурации у элемента данных и узла сети

net.if.in[eth3]	5m	7d	365d	Zabbix agent	Enabled	!
The item is not discovered anymore and will be deleted in 29d 23h 41m (on 2016-03-12 at 10:58).						

# Присоединение к группам элементов данных на основе значений обнаружения

New application	<input type="text"/>
Applications	<div><div>-None-</div><div>CPU</div><div>Filesystems</div><div>General</div><div>Memory</div><div>Network interfaces</div></div>
New application prototype	<input type="text" value="SQL Server: {#DBNAME}"/>
Application prototypes	<div><div>-None-</div></div>



LLD

СЛУЖБ WINDOWS

## Новый LLD для служб Windows: `service.discovery`

LLD правило возвращает данные в макросах:

```
{#SERVICE.NAME}  
{#SERVICE.DISPLAYNAME}  
{#SERVICE.DESRIPTION}  
{#SERVICE.STATE}  
{#SERVICE.STATENAME}  
{#SERVICE.PATH}  
{#SERVICE.USER}  
{#SERVICE.STARTUP}  
{#SERVICE.STARTUPNAME}
```

Новый ключ элемента данных:  
`service.info[service,<парам>]`

Пример:  
`service.info[{#SERVICE.NAME},state]`

# ПОЛЬЗОВАТЕЛЬСКИЕ ДАННЫЕ LLD

```
{
  "data": [
    { "#FSNAME": "/", "#FSTYPE": "rootfs" },
    { "#FSNAME": "/sys", "#FSTYPE": "sysfs" },
    { "#FSNAME": "/proc", "#FSTYPE": "proc" },
    { "#FSNAME": "/dev", "#FSTYPE": "devtmpfs" },
    { "#FSNAME": "/dev/pts", "#FSTYPE": "devpts" }
  ]
}
```



LLD ПО  
SNMP OID

# Синтаксис для правил обнаружения по SNMP:

SNMP OID до 3.0: **SNMP OID**

SNMP OID после 3.0: **discovery[{#SNMPVALUE}, SNMP OID]**

## Пример:

```
discovery[{#IFDESCR}, IF-MIB::ifDescr, {#IFALIAS}, IF-MIB::ifAlias]

{
  "data": [
    {"{#SNMPINDEX}":1, "{#IFDESCR}": "Interface #1", "{#IFALIAS}": "eth1"},
    {"{#SNMPINDEX}":2, "{#IFDESCR}": "Interface #2", "{#IFALIAS}": "eth2"},
    {"{#SNMPINDEX}":3, "{#IFALIAS}": "eth3"},
    {"{#SNMPINDEX}":4, "{#IFDESCR}": "Interface #4"},
    {"{#SNMPINDEX}":5, "{#IFALIAS}": "eth5"}
  ]
}
```



LLD C

ИСПОЛЬЗОВАНИЕМ

SQL



# LLD ПРИ ПОМОЩИ SQL ЗАПРОСОВ

LLD с SQL запросами:

`db.odbc.discovery[<описание>,<dsn>]`

Результаты автоматически преобразовываются в JSON

Имена колонок становятся именами макросов и выбранные строки становятся значениями этих макросов

Используйте алиасы к колонкам, чтобы обозначить имена макросов:

```
mysql> SELECT c.name, c.loc AS location FROM customers c;
```

**Имейте в виду:** правило обнаружения станет неподдерживаемым, если имя макроса недопустимо

# ПРАКТИЧЕСКАЯ УСТАНОВКА

Настройте unixODBC для MySQL

Получите список таблиц базы данных Zabbix, используя SQL запрос:

```
show tables
```

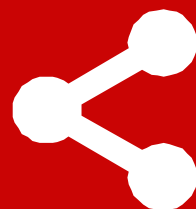
Используйте имена таблиц в прототипах групп элементов данных

Наблюдайте размер данных и индексов по каждой таблице:

```
SELECT data_length FROM information_schema.partitions  
WHERE table_name LIKE '<TABLE_NAME>'
```

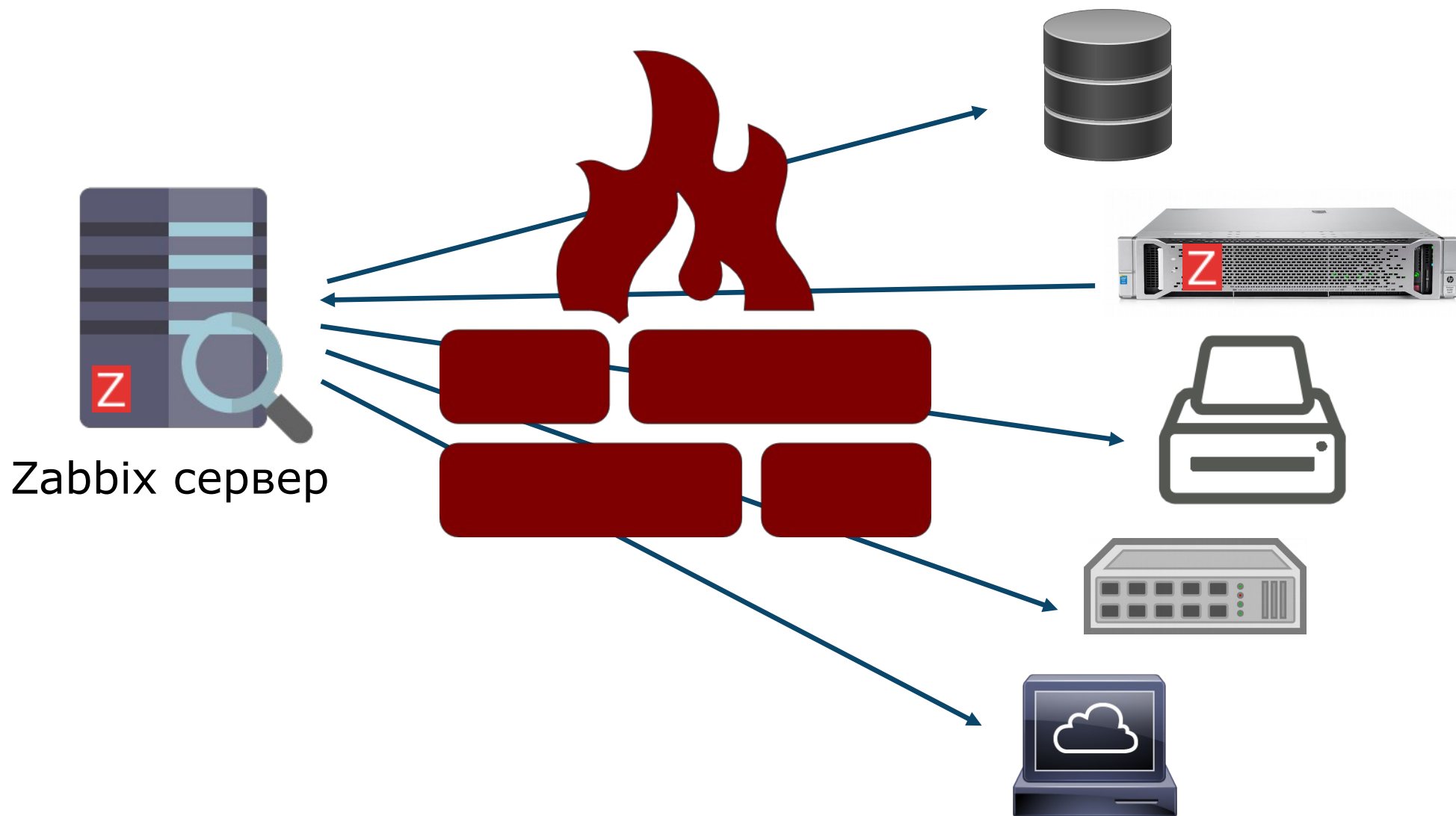
```
SELECT index_length FROM information_schema.partitions  
WHERE table_name LIKE '<TABLE_NAME>'
```

Убедитесь, что элементы данных получают данные

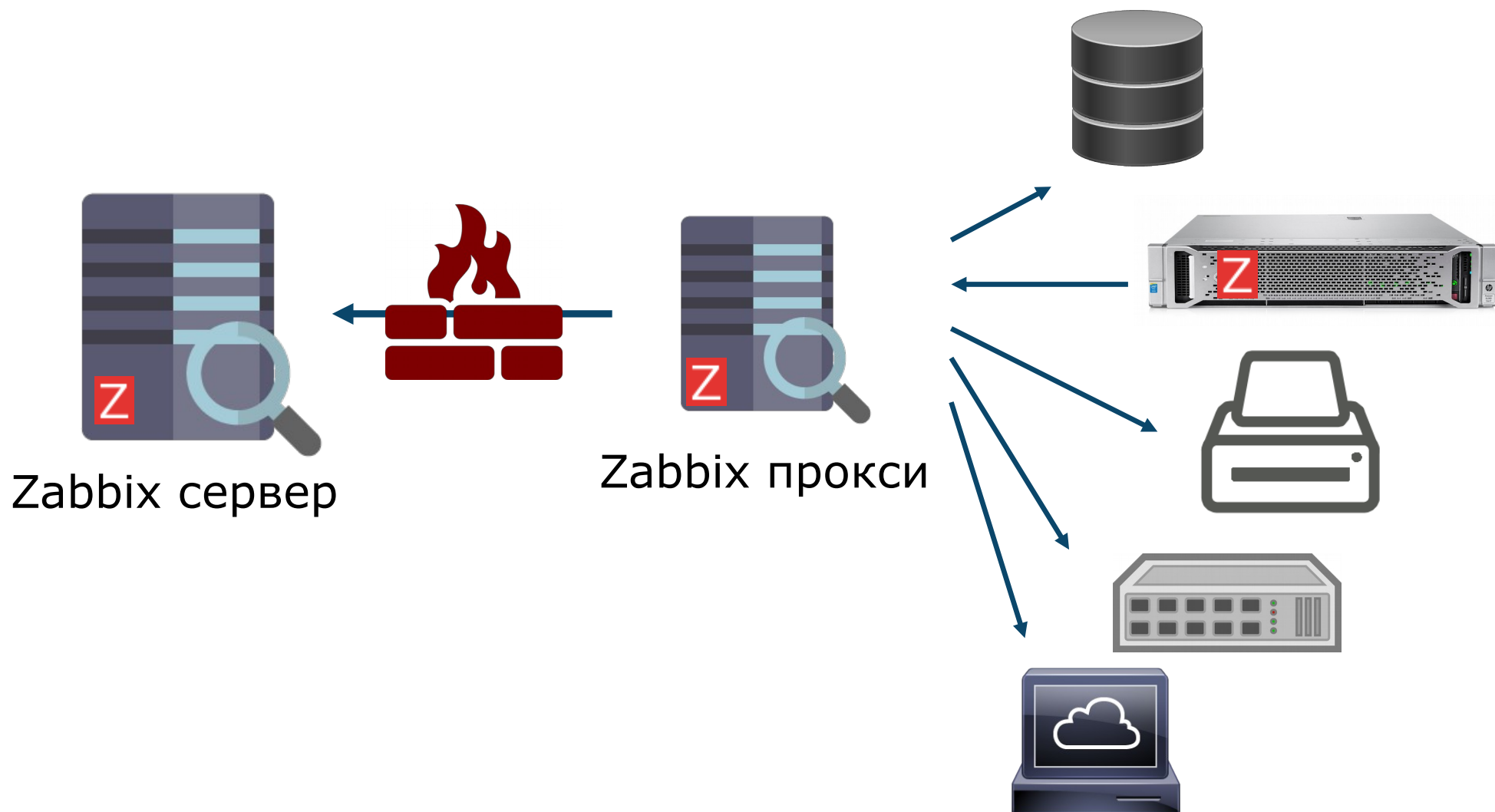


РАСПРЕДЕЛЕННЫЙ  
МОНИТОРИНГ

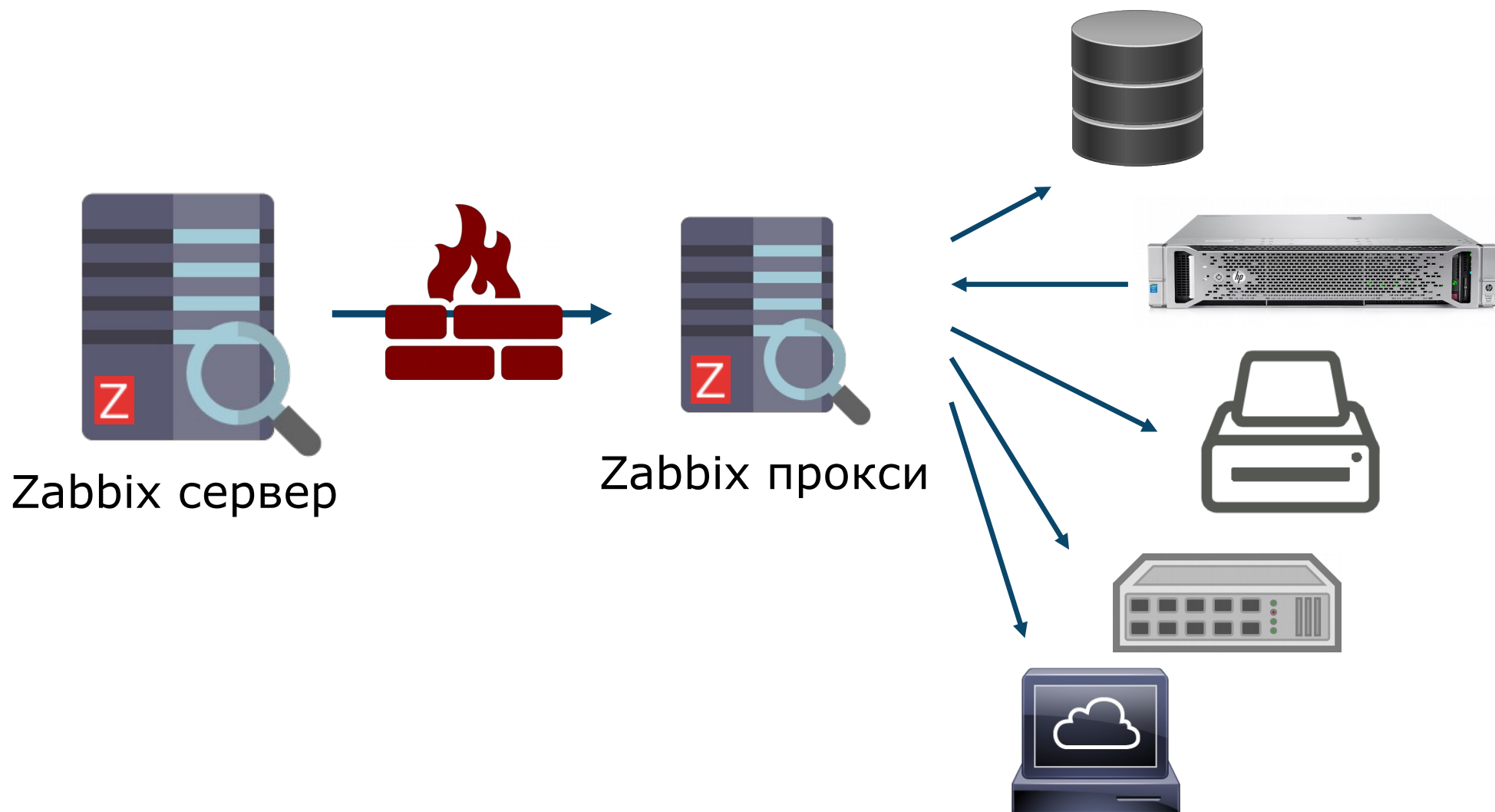
# ПРОБЛЕМА



# РЕШЕНИЕ - ZABBIX ПРОКСИ



# ПАССИВНЫЙ ПРОКСИ



# ОБЗОР ПРОКСИ

- ✓ Централизованный мониторинг
- ✓ Zabbix сервер контролирует конфигурации на всех прокси
- ✓ Поддерживает любую платформу, что и сервер
- ✓ Поддерживает любой тип базы данных, что и сервер
- ✓ Может создать БД на SQLite автоматически
- ✓ Может буферизировать данные в случае проблем со связью
- ✓ Выбор направления соединения
- ⚠ Не используйте ту же самую БД для прокси, что и для сервера
- ✓ Можно использовать разные типы БД на сервере & прокси

# ИЗНАЧАЛЬНАЯ УСТАНОВКА

Скомпилируйте бинарные файлы (`--enable-proxy`)

Создайте базу данных прокси (опционально для SQLite)

Обновите файл конфигурации

Запустите прокси

Добавьте прокси в веб-интерфейсе  
(Администрирование → Прокси)

Настройте узлы сети, чтобы они наблюдались через прокси



# ПРОКСИ

## Список прокси в веб-интерфейсе

- ✓ Отображается режим прокси
- ✓ Шифрование
- ✓ Когда был виден в последний раз
- ✓ Количество узлов сети, элементов данных
- ✓ Требуемая производительность

Proxies								Create proxy
<input type="checkbox"/> NAME ▲	MODE	ENCRYPTION	LAST SEEN (AGE)	HOST COUNT	ITEM COUNT	REQUIRED PERFORMANCE (VPS)		HOSTS
<input type="checkbox"/> New York proxy	Passive	PSK	1s	3	63	1.14		Switch 2250-SFP, Switch 2626, Tomcat
<input type="checkbox"/> Tallinn proxy	Passive	NONE	1s	4	201	6.73		Active Directory Server, Core Router, ESXi, MS Exchange Server 2010
<input type="checkbox"/> Tokyo proxy	Passive	CERT	1s	1	82	1.19		WIN-BS768P0N4TA
<input type="checkbox"/> Vilnius proxy	Passive	NONE	1s	2	115	1.04		Oracle DB, Sybase DB

# КОНФИГУРАЦИЯ АКТИВНОГО ZABBIX ПРОКСИ

**ProxyMode** должен быть задан равным 0 (активный)

**Hostname** должно совпадать с именем прокси указанным в веб-интерфейсе

**ProxyOfflineBuffer** контролирует как долго данные должны храниться в локальном буфере в случае неудачного подключения к серверу (по умолчанию, один час)

**ProxyLocalBuffer** позволяет сохранять данные в базе данных прокси для последующей обработки

**ConfigFrequency** контролирует как часто прокси будет запрашивать информацию о конфигурации с Zabbix сервера

**DataSenderFrequency** контролирует как часто данные будут отправляться на Zabbix сервер

**HeartbeatFrequency** выполняет соединение с Zabbix сервером, если нет данных для отправки на сервер

# КОНФИГУРАЦИЯ ПАССИВНОГО ZABBIX ПРОКСИ

Файл конфигурации Zabbix прокси:

**ProxyMode** должен быть задан равным 1 (пассивный)

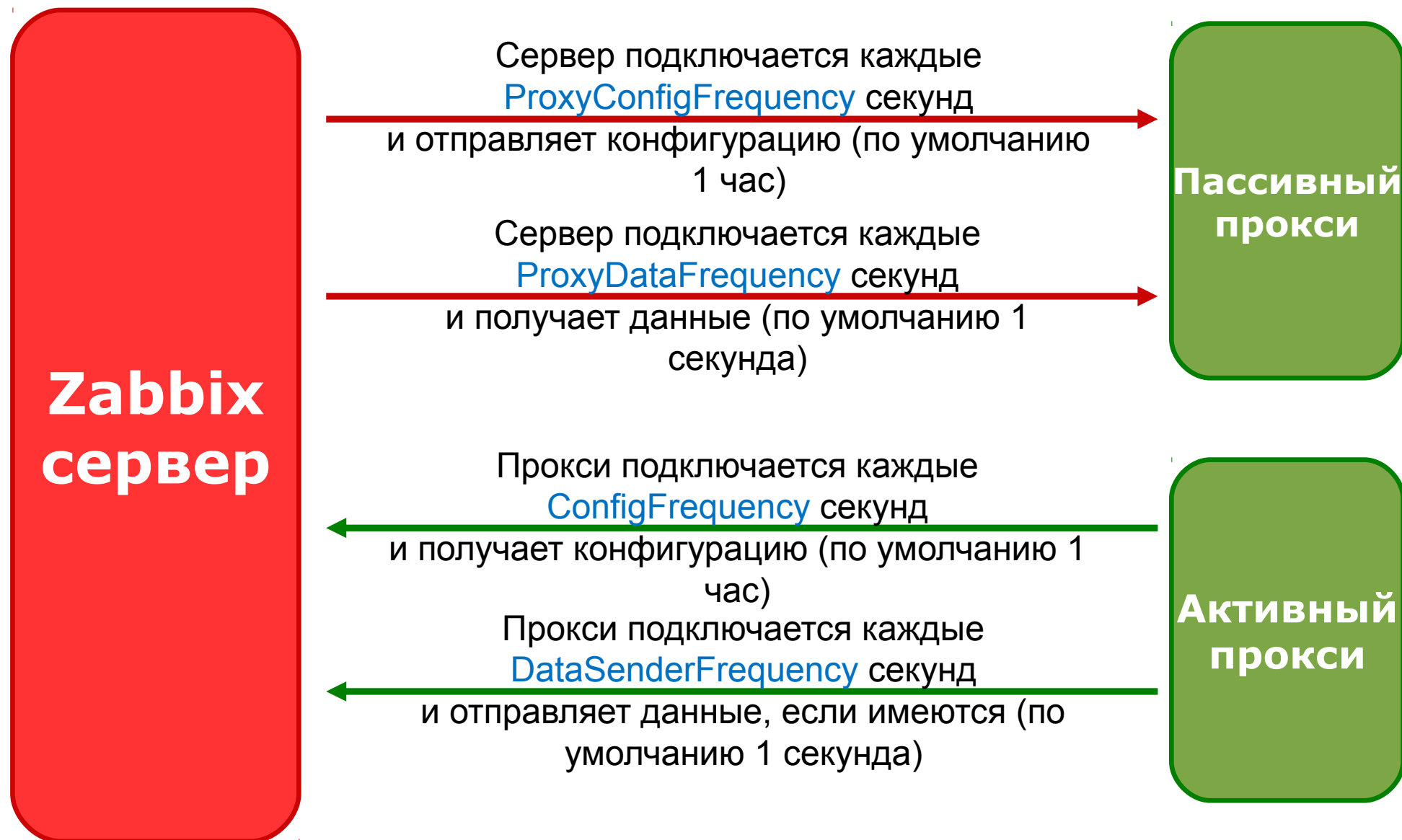
Файл конфигурации Zabbix сервера

**StartProxyPollers** контролирует количество поллеров, которые будут контактировать с прокси

**ProxyConfigFrequency** – как часто Zabbix сервер будет отправлять изменения конфигурации на пассивные прокси

**ProxyDataFrequency** – как часто будут запрошены данные с пассивных прокси

# ВЗАИМОДЕЙСТВИЕ С ПРОКСИ



# ОБНОВЛЕНИЕ КОНФИГУРАЦИИ АКТИВНОГО ПРОКСИ

```
# zabbix_proxy --runtime-control config_cache_reload
```

Отправка сигнала текущему прокси на перезагрузку кэша конфигурации

Также выполняется запрос конфигурации с сервера активным прокси

Игнорируется пассивным прокси

# МОНИТОРИНГ ДОСТУПНОСТИ ПРОКСИ

`HeartbeatFrequency` гарантирует, что сервер будет видеть прокси даже, если нет данных на отправку. По умолчанию раз в минуту.

Внутренний элемент данных `zabbix[proxy,"Proxy name",lastaccess]`

Триггер на основе функции `fuzzytime`:

```
{server:zabbix[proxy,"Proxy name",lastaccess].fuzzytime(180)}=0
```

# ОЧЕРЕДЬ ПО КАЖДОМУ ПРОКСИ

Отобразится производительность Zabbix сервера и каждого прокси отдельно

Нет деталей по категориям элементов данных

Queue of items to be updated							Overview by proxy ▼
PROXY	5 SECONDS	10 SECONDS	30 SECONDS	1 MINUTE	5 MINUTES	MORE THAN 10 MINUTES	
New York	0	0	0	0	0	0	
Tokyo	0	0	0	0	0	0	
Server	0	0	0	0	0	83	
Total: 3							

# ВОПРОСЫ ПРОИЗВОДИТЕЛЬНОСТИ

## Zabbix сервер

- ✓ Быстрый CPU
- ✓ Быстрое хранилище данных для обработки исторической информации
- ✓ Количество процессов траппер должно быть больше количества активных прокси
- ✓ Производительность после недоступности

## Proxu

- ✓ Низкие требования к аппаратному обеспечению
- ✓ Можно использовать встраиваемые аппаратные решения



# ОГРАНИЧЕНИЯ

Прокси не поддерживает перенаправление удаленных команд

Автоматическое создание базы данных доступно только для SQLite

Нет реакции на триггеры, прокси используются только для сбора данных

# ПРАКТИЧЕСКАЯ УСТАНОВКА

Настройте один активный прокси с SQLite

Настройте один активны прокси с MySQL (по желанию)

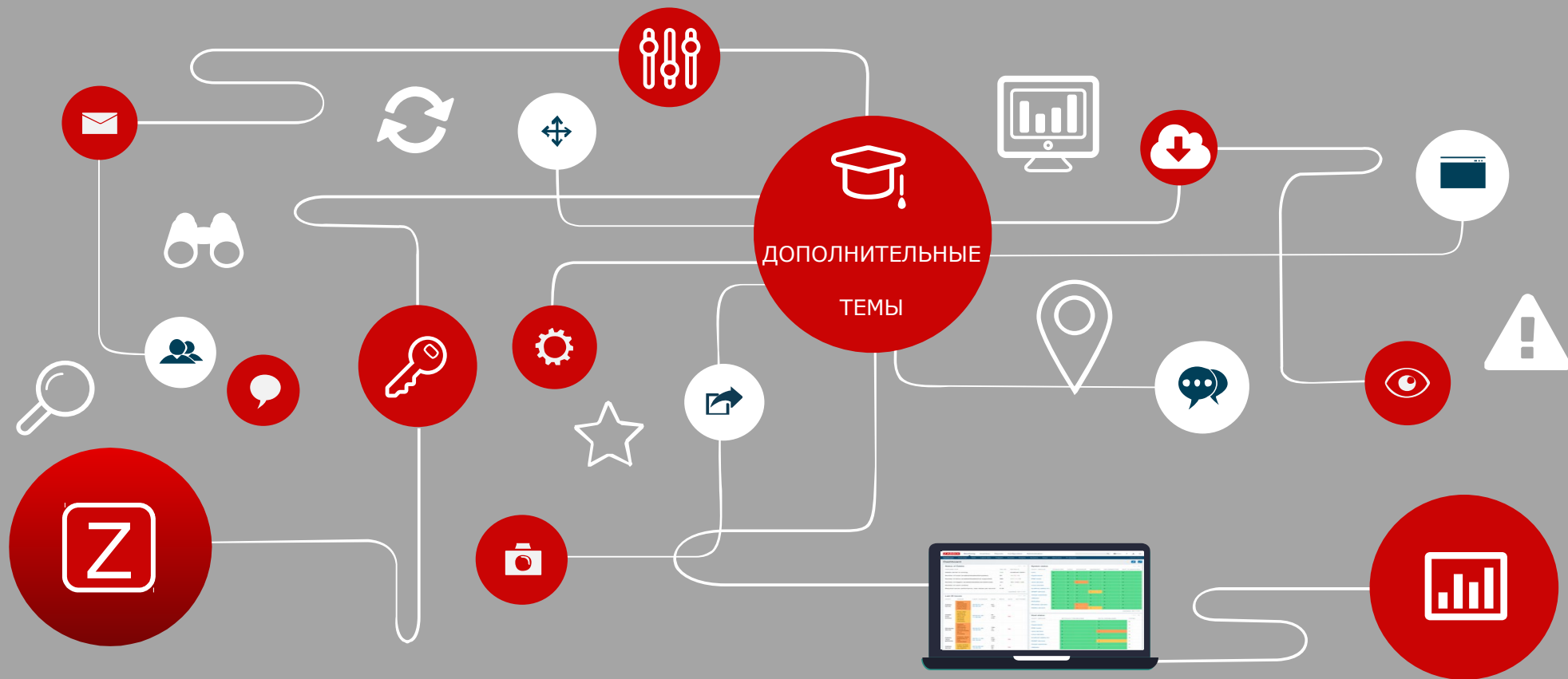
Настройте один пассивный прокси с SQLite (по желанию)

Мониторьте доступность всех прокси

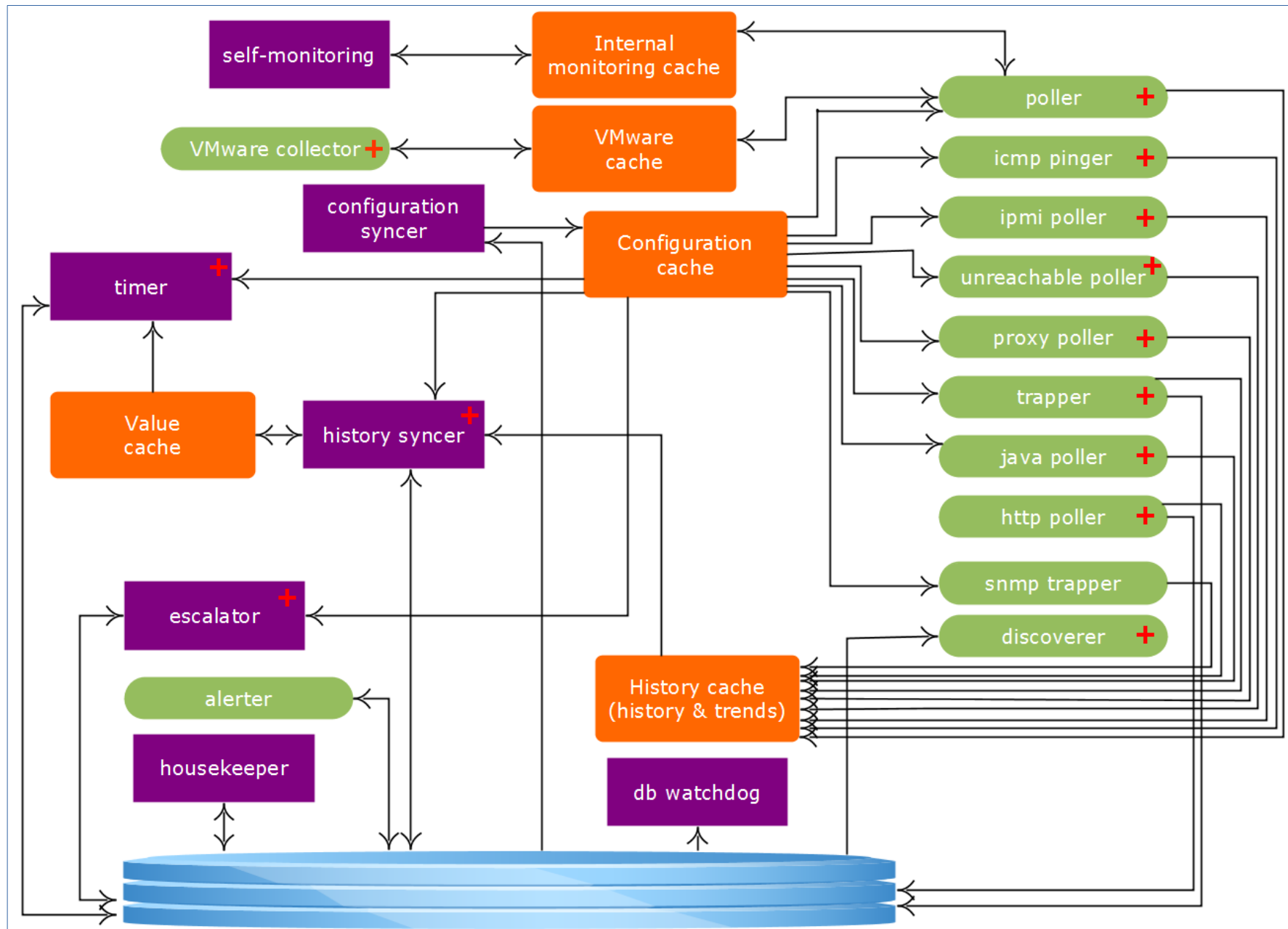
Мониторьте по крайней мере одну систему через прокси

Отправьте значение на элемент данных Zabbix траппер

Убедитесь, что все элементы данных получают значения



# КОМПОНЕНТЫ ZABBIX СЕРВЕРА



# НАСТРОЙКИ ВЕБ-ИНТЕРФЕЙСА

## В `include/defines.inc.php`

Защита от перебора паролей

`ZBX_LOGIN_ATTEMPTS`

`ZBX_LOGIN_BLOCK`

Умолчания & лимиты графиков

`ZBX_MIN_PERIOD`

`ZBX_MAX_PERIOD`

`ZBX_PERIOD_DEFAULT`

`GRAPH_YAXIS_SIDE_DEFAULT`

Ограничение количества строк во всплывающих окнах

`ZBX_WIDGET_ROWS`

Округление

`ZBX_UNITS_ROUNDOff_THRESHOLD`

`ZBX_UNITS_ROUNDOff_UPPER_LIMIT`

`ZBX_UNITS_ROUNDOff_LOWER_LIMIT`

Последние данные и обзор элементов данных

`ZBX_HISTORY_PERIOD`

# КОНФИГУРАЦИЯ В ДЕТАЛЯХ

Параметры конфигурации Zabbix сервера  
[zabbix\\_server.conf](#)

Параметры конфигурации Zabbix агента  
[zabbix\\_agentd.conf](#)

# ZABBIX ПРОТОКОЛЫ

Смотрите <https://zabbix.org/wiki/Docs/protocols>

```
{
  "request" : "history data",
  "host" : "Zabbix proxy",
  "clock" : 1423755919,
  "ns" : 909069388,
  "data" : [
    {
      "host": "Zabbix server",
      "key": "web.test.in[example.com,index,bps]",
      "clock": 1423755915,
      "ns": 146648341,
      "value": "0.000000"
    },
    {
      "host": "Zabbix server",
      "key": "web.test.time[example.com,index,resp]",
      "clock": 1423755915,
      "ns": 146648341,
      "value": "0.259713"
    },
    {
      "host": "Zabbix server",
      "key": "web.test.rspcode[example.com,index]",
      "clock": 1423755915,
      "ns": 146648341,
      "value": "200"
    }
  ]
}
```

# ВОПРОСЫ?

**ZABBIX**

The Enterprise class Monitoring Solution for Everyone

[WWW.ZABBIX.COM](http://WWW.ZABBIX.COM)





ВРЕМЯ ДЛЯ ОТДЫХА :)