OTUS

ОНЛАЙН-ОБРАЗОВАНИЕ

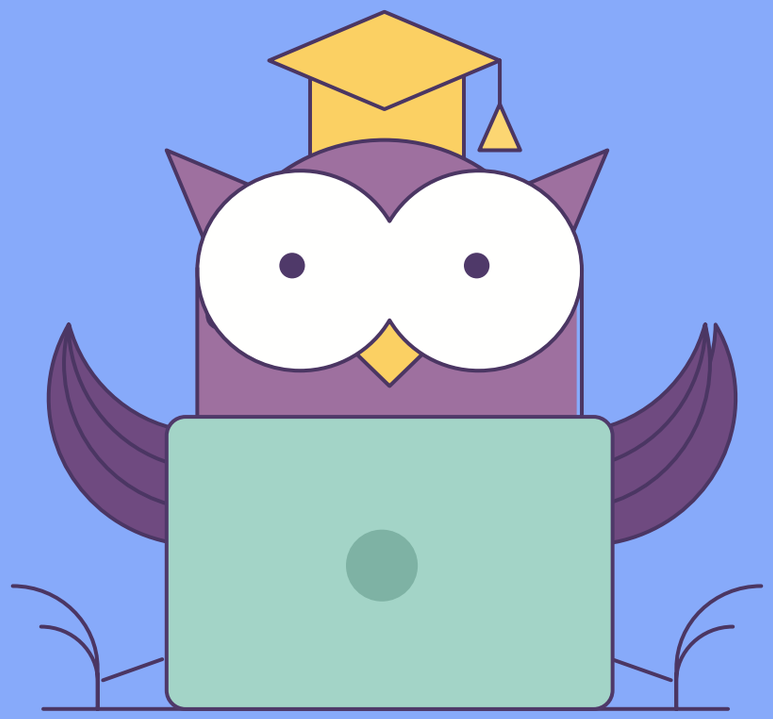
OTUS

Мониторинг

Курс «Data Engineer»

Владимир Дроздецкий

Меня хорошо слышно && видно?



Напишите в чат, если есть проблемы!

Ставьте + если все хорошо
Ставьте - если есть проблемы

Цели занятия

- Задачи мониторинга
- Типы мониторинга
- Метрики
- Хранение данных
- Сбор данных
- Алертинг
- Инструменты (Zabbix, NetData, etc...)

Ожидания от мониторинга

- Алертинг о сбоях в системе => Сбор информации о текущем состоянии
- Предупреждения о возможных сбоях и проблемах
- Отражение состояния системы/сервера/сервиса/компонента сервиса
- Сбор статистики и визуализация
- Агрегация и группировка
- Предоставление информации для мониторинга “второго порядка” или
- “мониторинга вперед” - trend monitoring.
- Отчеты
- Красивые дашборды

Whitebox мониторинг

Мониторинг, базирующийся на метриках, которое дает само приложение/сервис:

- через логи
- интерфейсы
- API
- SNMP

Blackbox мониторинг

Мониторинг, оценивающий внешнее состояние сервиса/системы:

- ping
- HTTP request
- Открытый порт
- Наличие процессов

Метрики

Основные метрики системы:

- LoadAvg, CPU, Net (bps/pps), DISK Load
- потребление Mem/DISK
- “чистота” системных логов: dmesg, messages
- Актуальность состояния и резервных копий

Метрики процесса

- Наличие процесса и правильное количество этих процессов
- Открытый сокет/порт процесса
- возможность получить статус процесса (где применимо)
- некоторые параметры статуса

Метрики сервиса

- “диагностический запрос”, который задействует все или большинство компонентов системы.
- время отклика/обработки запроса
- количество обращений в единицу времени
- количество одновременных обращений

Исторические данные (TimeSeries)

С историческими данными возникает несколько вопросов:

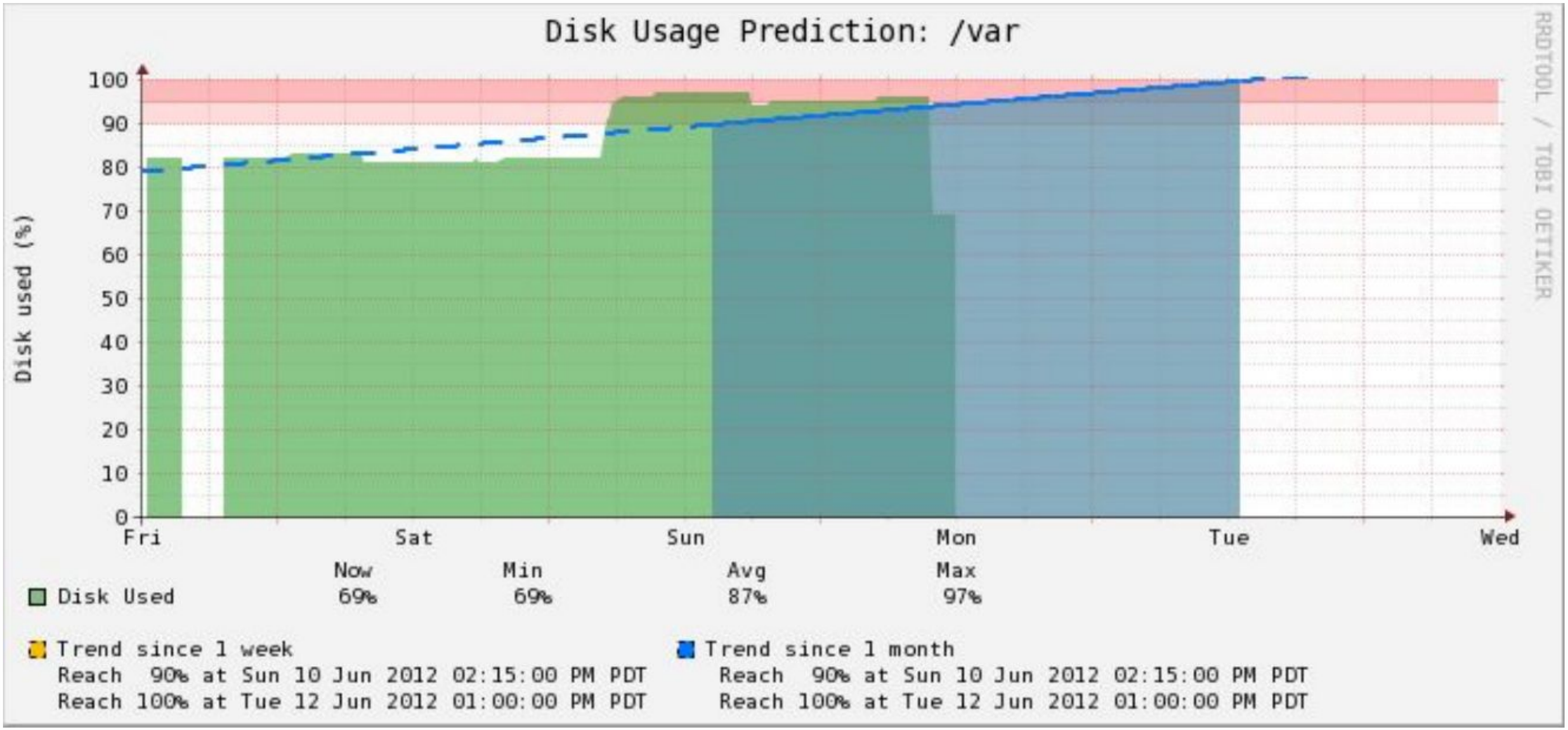
1. Как их собирать?
2. Где их хранить?
3. Сколько их хранить?
4. Как с ними работать?

Для чего нужны исторические данные

1. Анализ проблем
2. Гадание на графиках или Trend monitoring.

Тут высокое разрешение не важно, т.к. временные рамки в которых идет “гадание” это недели, месяцы и годы.

Пример


Disk Usage Prediction: /var
RRDTOOL / TOBI OETIKER
Disk used (%)
100
90
80
70
60
50
40
30
20
10
0
Fri
Sat
Sun
Mon
Tue
Wed
Now
Min
Avg
Max
Disk Used
69%
69%
87%
97%
Trend since 1 week
Reach 90% at Sun 10 Jun 2012 02:15:00 PM PDT
Reach 100% at Tue 12 Jun 2012 01:00:00 PM PDT
Trend since 1 month
Reach 90% at Sun 10 Jun 2012 02:15:00 PM PDT
Reach 100% at Tue 12 Jun 2012 01:00:00 PM PDT

Для чего нужны исторические данные

Что из себя представляют исторические данные?

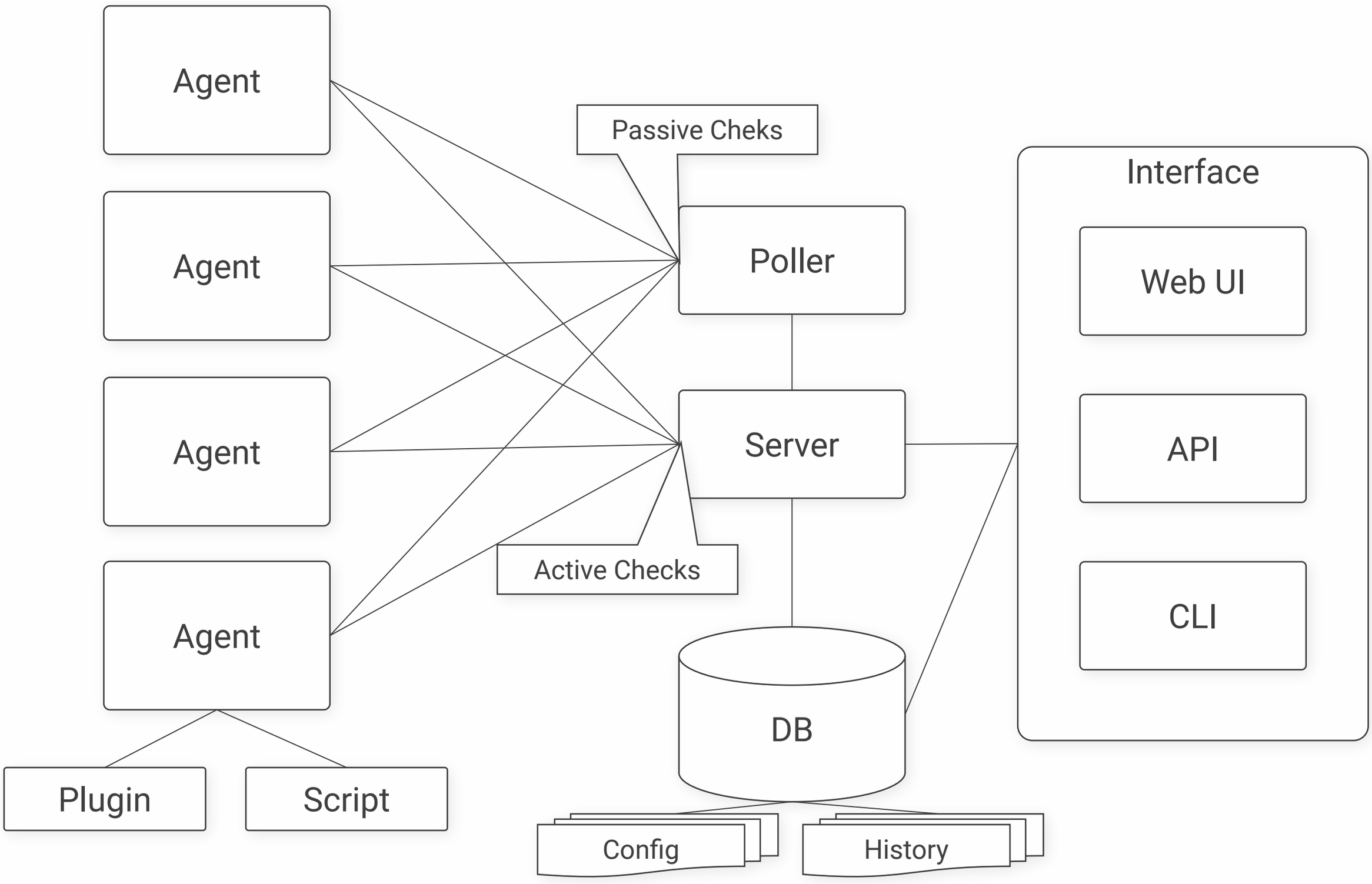
Пару дата_время+значение, все это привязано к некоему имени метрики и должно храниться какое-то время, возможно по-дороге усредняться в “меньшее” разрешение (1/s -> 1/m -> 1/10m -> 1/h).

Такие данные называются time series и есть несколько хранилищ для таких данных:

 rrdtool
Whisper (Graphite)
Influxdb ()
Prometheus

Также некоторые инструменты (Zabbix, Nagios, Cacti) используют SQL базы данных

Основные компоненты мониторинга



Инструменты мониторинга

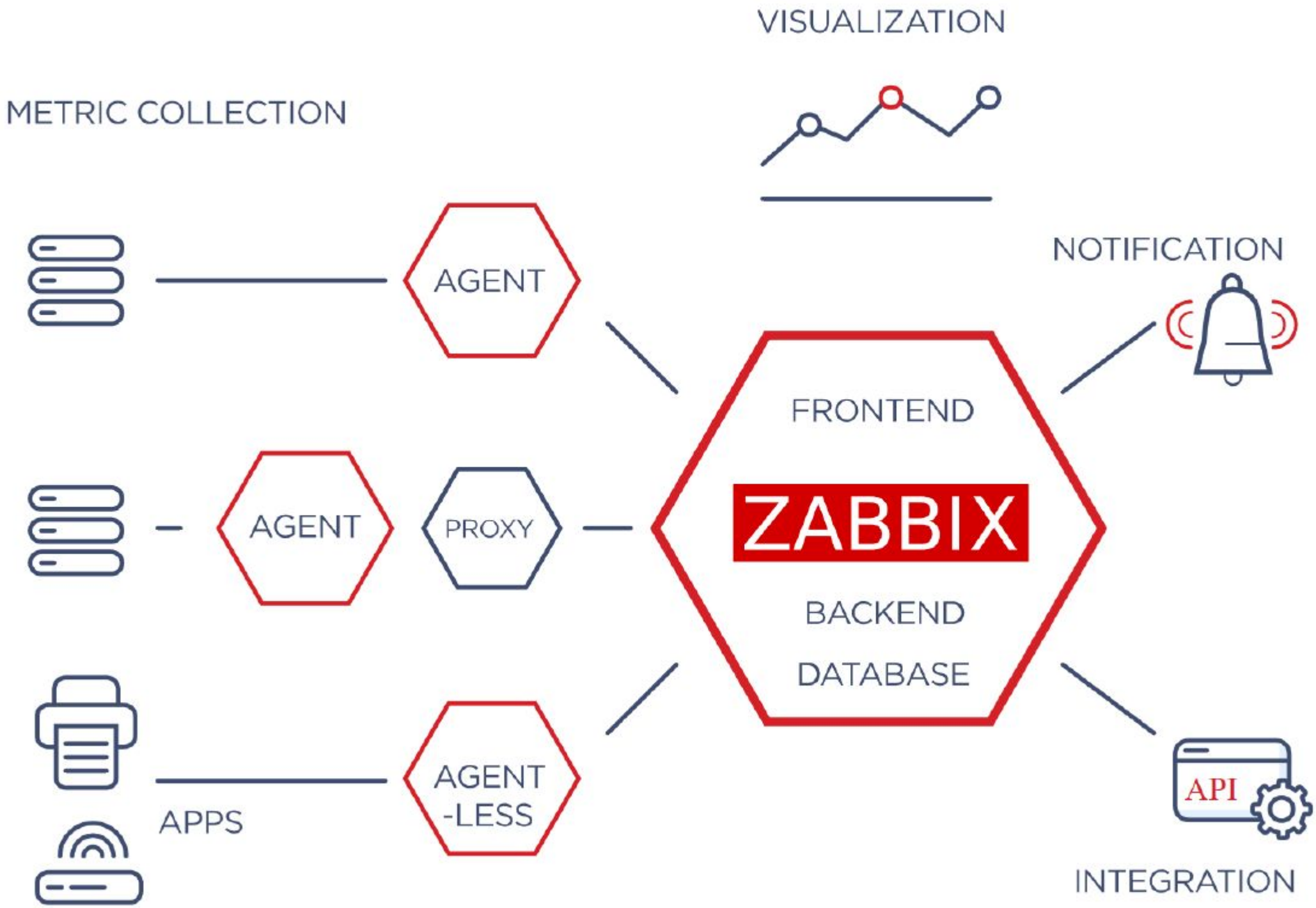
- Zabbix - свой агент, свой веб, SQL database
- nagios/icinga - свой агент, свой веб, SQL database
- Graphite - carbon (агент), whisper (tsdb), graphite-web
- cacti, ganglia, collectd, mrtg, munin - база rrdtool
- Netdata - агент и веб
- Prometheus - node_exporter (агент), prometheus (база, веб), alertmanager
- InfluxData - telegraph (агент), influxdb, chronograph (web), kapacitor (alerts)
- StatsD - набор инструментов для сбора метрик
- Graphana - интерфейс для дашбордов и алертинга

COMBO!!!

- carbon (агент) -> whisper (бд) -> grafana (интерфейс)
- netdata (в качестве агента) -> null / influxdb / elasticsearch / prometheus / graphite (в качестве бд) -> grafana (интерфейс)
- node_exporter (агент) -> prometheus (в качестве бд) -> grafana (интерфейс)
- collectd (агент) -> influxdb (бд) -> grafana (интерфейс)
- zabbix (агент+сервер) -> mysql -> grafana (интерфейс)
- telegraf (агент) -> elasticsearch (бд) -> kibana (интерфейс)


ZABBIX

Arch


VISUALIZATION
METRIC COLLECTION
NOTIFICATION
AGENT
FRONTEND
ZABBIX
AGENT
PROXY
BACKEND
DATABASE
AGENT
-LESS
API
APPS
INTEGRATION

Требования

ОС:

- Linux
- Solaris
- AIX
- HP-UX
- FreeBSD
- OpenBSD

База данных:

- MySQL
- Форки MySQL
- PostgreSQL
- Oracle
- IBM DB2
- Elasticsearch

Доп. библиотеки:

- Compression: zlib
- SNMP: Net-SNMP
- Web: libcurl
- SSH: libssh2
- IPMI: OpenIPMI
- Jabber: lib-iksmel
- ODBC: unixODBC
- Encryption: OpenSSL

Суть установки

- Пакеты самого приложения
- База данных - создание структуры
- Фронтенд - Apache, lighthttpd, nginx
- PHP-FPM

Общение с узлами

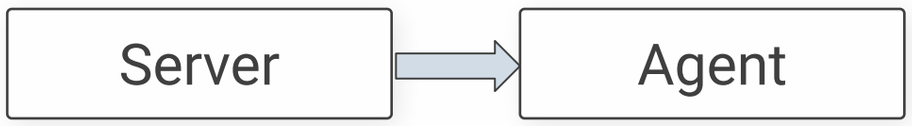
- Zabbix Agent
  - Собственно агент устанавливаемый на целевые хосты
- SNMP
  - Протокол использующийся для управления сетевыми устройствами
- JMX
  - Мониторинг JMX можно использовать для наблюдения за счетчиками JMX в Java приложениях.
- IPMI
  - интеллектуальный интерфейс управления платформой. В контексте Zabbix используется для мониторинга аппаратной платформы серверов

Узлы сети и группы

- Старайтесь группировать хосты по типам и задачам (хост может быть включен в любое кол-во групп)
- Давать понятные имена для хостов
- Впоследствии разделение прав на просмотр и управление

Типы элементов данных

Опрашиваемые

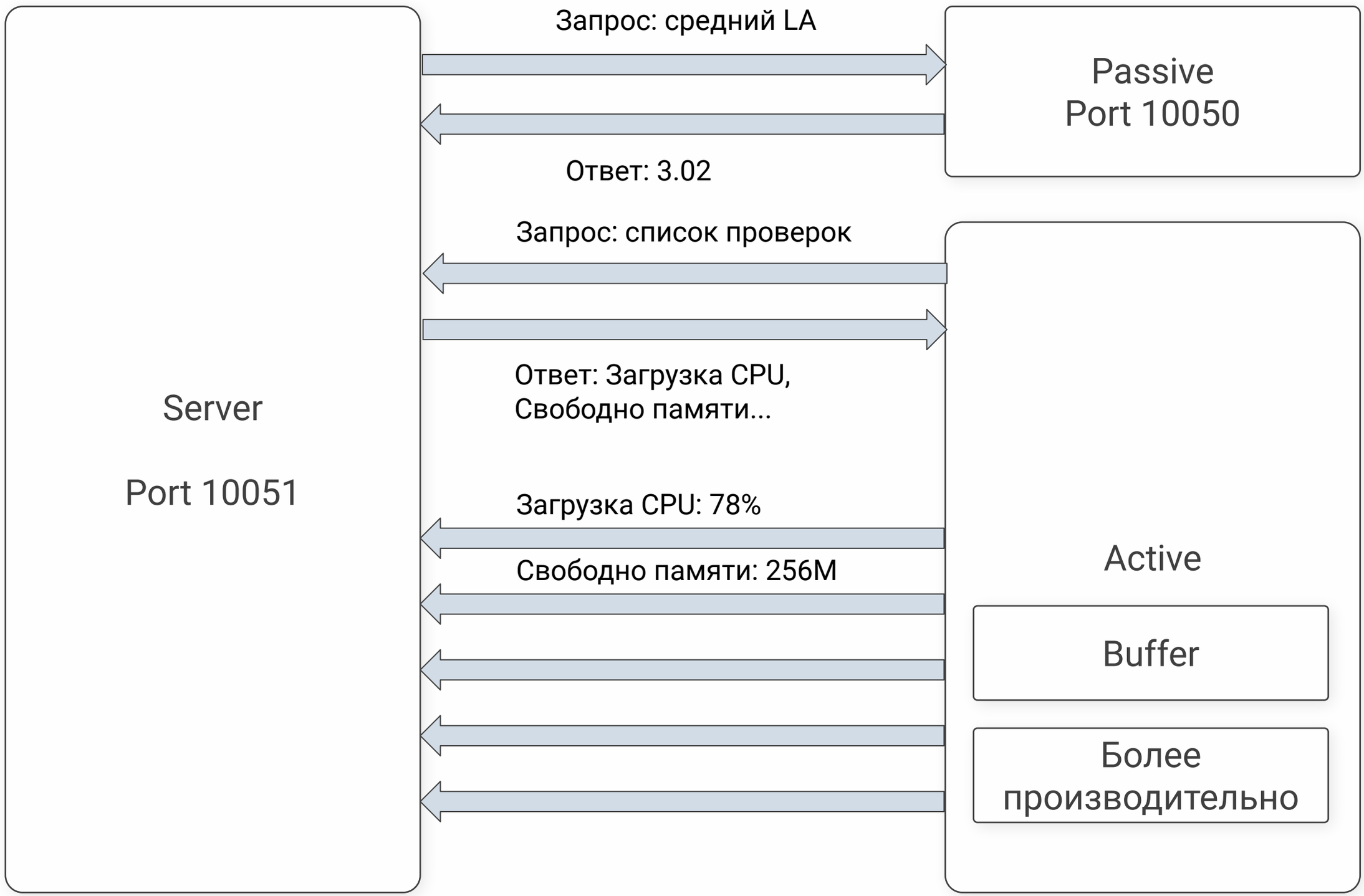


- Zabbix agent
- Безагентные проверки (simple checks)
- SNMP агент
- IPMI
- JMX
- HTTP
- SSH
- Telnet
- Базы данных

Трапы

- Zabbix agent (активный)
- SNMP трапы
- Zabbix траппер

Active/Passive checks



Триггеры

- Понятное имя триггера
- Устанавливайте важность триггера
- Синтаксис:
  - {host:key.function(param)}=0
  - vm.memory.size[available].last(0)}<20M
- Тестируйте триггеры (встроенная функция в Zabbix)

SOS!!!

- Mail
- SMS
- SIP
- Jabber
  - Slack
  - Telegram
  - Discord

Хранение истории

Вы можете указать как много дней история будет храниться:

- в диалоге свойств элемента данных
- при массовом обновлении элементов данных
- при настройке задач очистки истории

Любые более старые данные будут удалены с помощью автоматической очистки базы данных (Housekeeper).

Мониторинг БД

Два пути:

- Zabbix Agent
  - сбор стандартных метрик
- ODBC (другие коннекотры)
  - кастомные запросы
  - не нужен Агент

Мониторинг VMWare

Как:

- Работает из коробки
- Не требует сторонних утилит. Основан на родном API VMWare

Что умеет:

- Авто обнаружение гипервизоров в кластере и гостевых машин
- Можно расширить использовать и для XEN, KVM, LXC
- Есть готовые шаблоны

Шифрование

Зачем:

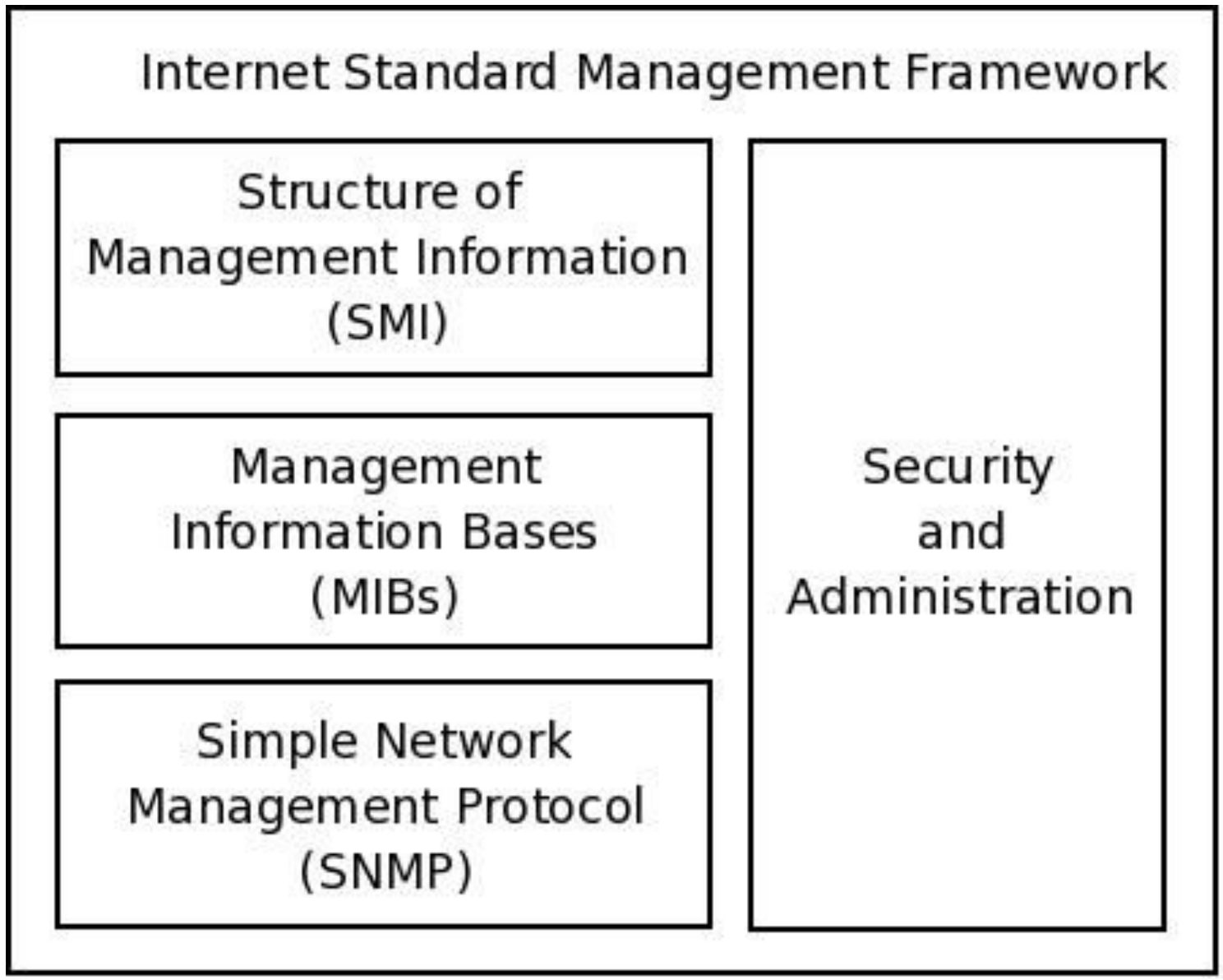
- защита деликатных данных (например, данные конфигурации от Zabbix сервера на прокси могут содержать учетные данные для доступа к наблюдаемым узлам сети)
- доверие “другой стороне”
- предотвращение отправки сфальсифицированных данных на Zabbix

Как:

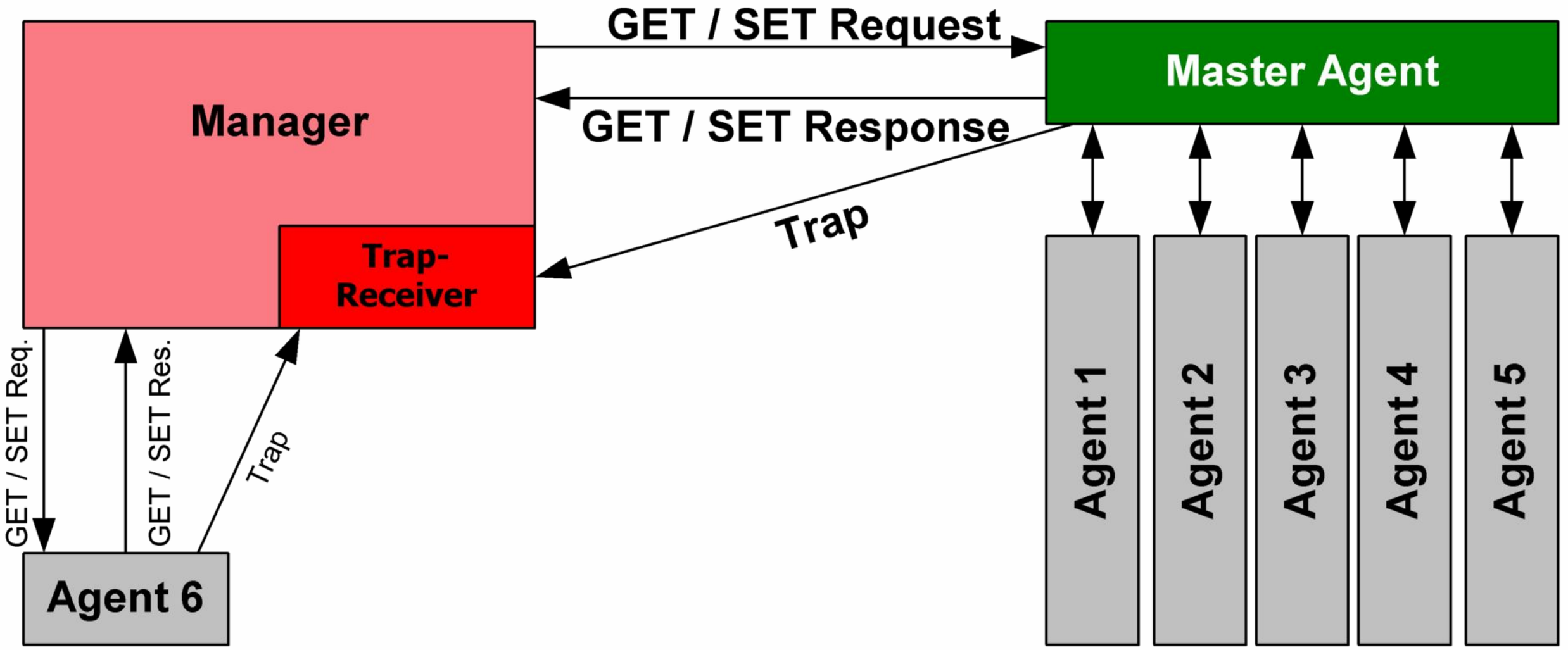
- Протокол TLS 1.2
- Библиотеки:
  - OpenSSL с 1.0.1 (--with-openssl)
  - GnuTLS с 3.1.18 (--with-gnutls)
  - mbed TLS (PolarSSL) с 1.3.9 – 1.3.x, не 2.0.0 (--with-mbedtls)

SNMP

SNMP (Simple Network Management Protocol) — протокол, который используется для управления сетевыми устройствами. С помощью протокола SNMP, программное обеспечение для управления сетевыми устройствами может получать доступ к информации, которая хранится на управляемых устройствах (например, на коммутаторе). На управляемых устройствах SNMP хранит информацию об устройстве, на котором он работает, в базе данных, которая называется MIB.


Internet Standard Management Framework
Structure of Management Information (SMI)
Management Information Bases (MIBs)
Simple Network Management Protocol (SNMP)
Security and Administration

SNMP


GET / SET Request
GET / SET Response
Trap
Manager
Trap-
Receiver
Master Agent
Agent 1
Agent 2
Agent 3
Agent 4
Agent 5
GET / SET Req.
GET / SET Res.
Trap
Agent 6

Ваши вопросы?

Заполните, пожалуйста, опрос в ЛК о занятии

Спасибо
за внимание!
До встречи в Slack и на вебинаре