

# Описание автоматизируемых функций платформы "Центральный Пульс"

2019

**ЦЕНТРАЛЬНЫЙ ПУЛЬС**

<b>1.</b>	<b>Исходные данные</b>	<b>2</b>
1.1	Материалы и документы, используемые при разработке функциональной части АС .....	3
1.2	Особенности объекта управления .....	3
1.3	Системы управления, взаимосвязанные с разрабатываемой АС .....	3
1.4	Описание информационной модели объекта .....	4
<b>2.</b>	<b>Цели АС и автоматизируемые функции</b>	<b>5</b>
<b>3.</b>	<b>Характеристика функциональной структуры</b>	<b>7</b>
3.1	Перечень подсистем АС .....	8
3.2	Пояснения к разделению автоматизированных функций .....	9
3.3	Требования к временному регламенту и характеристикам .....	10
<b>4.</b>	<b>Типовые решения</b>	<b>13</b>

# Исходные данные



## 1 Исходные данные

### 1.1 Материалы и документы, используемые при разработке функциональной части АС

---

При разработке функциональной части проекта были использованы следующие материалы и документы:

- концепция АС «Центральный Пульт»;
- техническое задание по разработке пилотного проекта АС «Центральный Пульт»;
- бизнес-требования от потенциальных пользователей;
- данные, полученные в результате опроса лиц, ответственных за разработку системы;
- схема функциональной структуры.

### 1.2 Особенности объекта управления

---

При разработке АС «Центральный Пульт» учитывался ряд особенностей, которые повлияли на проектные решения. К этим особенностям относится увеличение партнёров из различных сфер деятельности, что влечёт за собой регулярное усовершенствование функционала, интеграцию с локальными и вспомогательными АС партнёров, введение в эксплуатацию новых способов сбора данных. В связи с этим закладываемые в АС «Центральный Пульт» проектные решения должны обладать достаточной степенью гибкости и масштабируемости.

### 1.3 Системы управления, взаимосвязанные с разрабатываемой АС

---

В качестве подсистем, взаимосвязанных с АС «Центральный Пульт», могут выступать следующие подсистемы:

- SAYMON-Server;
- СУБД (MongoDB, OpenTSDB);
- SAYMON Agent;

- Клиент (Web, Android, IOS).

## 1.4 **Описание информационной модели объекта**

---

Информационное обеспечение АС «Центральный Пульт» включает в себя внутримашинное и внешнее информационное обеспечение.

В состав внешнего информационного обеспечения входит система документации.

В состав внутримашинного информационного обеспечения входят:

- центральное хранилище данных;
- центр обработки запросов клиента;
- набор NodeJS-приложений;
- сетевое журналируемое хранилище данных «Redis»;
- система управления базами данных «MongoDB»;
- база данных временных рядов «OpenTSD».

# Цели АС и автоматизируемые функции

ЦЕНТРАЛЬНЫЙ ПУЛЬТ

## 2 Цели АС и автоматизируемые функции

Цели создания платформы:

1. Упростить сбор и анализ информации;
2. Ускорить процесс обработки данных и сделать его автоматизированным;
3. Визуализировать полученные и обработанные данные;
4. Обеспечить непрерывное хранение информации;
5. Автоматически уведомлять пользователей о состояниях объектов удобным для них способом;
6. Автоматически исправлять аварийные ситуации;
7. Управлять объектами мониторинга.

Перечень автоматизируемых функций включает в себя:

1. Представление практически любого объекта окружающего мира в качестве объекта мониторинга;
2. Смена состояния объекта в соответствии с заданными условиями;
3. Хранение оригинальных значений показателей за промежутки времени;
4. Обеспечение анализа в табличной и графической формах;
5. Быстрое реагирование на изменение статуса объекта;
6. Представление объектов, согласно их географическому месторасположению;
7. Использование гибкого механизма оповещений;
8. Группировка объектов по заданным общим критериям;
9. Импорт объектов и метрик из Zabbix;
10. Экспорт метрик в Grafana;
11. Преобразование данных от агента в компактный вид и их экспорт;
12. Прикрепление контекстной документации и отображение свойств объектов.

# Характеристика функциональной структуры



**ЦЕНТРАЛЬНЫЙ ПУЛЬТ**



## 3 Характеристика функциональной структуры

Программное обеспечение платформы «Центральный Пульт» имеют открытые API-интерфейсы, которые обеспечивают информационную совместимость системы и возможность интеграции с другими автоматизированными системами.

### 3.1 Перечень подсистем АС

---

Система состоит из следующих логических подсистем:

#### 1. SAYMON-Server

Подсистема обеспечивает выполнение следующих функций:

- получение уведомлений об ошибках передачи данных;
- настройка и контроль доступа к объектам системы;
- контроль целостности данных;
- управление работой других подсистем;
- реагирование на возникновение аварийных ситуаций;
- преобразование данных в требуемый системой формат;

#### 2. СУБД (MongoDB, OpenTSDB)

Подсистема обеспечивает выполнение следующих функций:

- хранение данных;
- журналирование переданной и полученной информации от сервера к клиенту и наоборот;
- загрузку полученных данных в систему;
- резервное копирование;
- восстановление базы данных после сбоев.

#### 3. SAYMON Agent

Подсистема обеспечивает выполнение следующих функций:

- сбор информации на выбранном объекте мониторинга;

- анализ полученной информации;
- выполнение пользовательских скриптов;
- осуществление как пассивного, так и активного мониторинга;
- отправка обработанных данных серверу.

## 4. Клиент (Web, Android, IOS)

Подсистема обеспечивает выполнение следующих функций:

- настройка условий мониторинга;
- создание и выбор объектов;
- настройка объектов мониторинга;
- управление пользователями;
- кастомизация интерфейса;
- просмотр и изменение текущих состояний;
- построение графиков.

## 3.2 Пояснения к разделению автоматизированных функций

### 3.2.1 Подсистема SAYMON-Server

SAYMON-Server — набор NodeJS-приложений, которые взаимодействуют между собой и со всеми остальными компонентами.

Выделяются следующие составляющие подсистемы:

- Веб-сервер - это HTTP-сервер, обслуживающий запросы клиента. В качестве Веб-сервера используется Nginx.
- REST-сервер — серверный компонент, обрабатывающий REST-запросы от JavaScript-компонентов на тонком клиенте. Через REST-API пользователь получает всю информацию об инфраструктуре, а также производит манипуляции с ней. REST-сервер реализован в виде набора NodeJS-приложений.
- Сервер данных — производит анализ данных, поступивших от агентов. В частности, управляет логикой смены состояний у объектов и связей. Сервер данных реализован в виде набора NodeJS-приложений.

- Кэш в памяти или In-Memory кэш — хранит базу данных оперативной памяти. В качестве In-Memory кэша используется Redis.

## 3.2.2 Подсистема СУБД

Данная подсистема отвечает за хранение и передачу данных между участниками обмена и включает в себя два компонента:

1. MongoDB — система управления базами данных, классифицированная как NoSQL;
2. OpenTSDB — база данных временных рядов (Time series).

## 3.2.3 Подсистема SAYMON-Agent

«SAYMON-Agent» - компонент системы, осуществляющий мониторинг на выбранном узле и расположенных на нём объектах, а также связей данного узла. Данные, собранные агентом, периодически отправляются в In-Memory кэш и затем анализируются сервером. Агент реализован в виде набора Java-приложений.

## 3.2.4 Подсистема Клиент

Данная подсистема является основным инструментом конечного пользователя и может быть представлена двумя способами:

1. Web-Client — работа осуществляется внутри браузера Chrome версии не ниже 58.0.
2. Мобильное приложение на Android и IOS.

## 3.3 Требования к временному регламенту и характеристикам

Требования к временному регламенту и характеристикам процесса реализации автоматизированных функций соответствует общим требованиям к автоматизированной системе, изложенным ниже.

При проектировании и разработке подсистем учитывались следующие общие требования:

- работа с программным обеспечением должна осуществляться пользователями и администраторами системы. Доступ к функциям должен осуществляться в соответствии с выделенными правами и уровнем доступа пользователей;
- происходящие события должны фиксироваться в системном журнале с указанием типа событий, времени его выполнения и имени учётной записи пользователя, инициировавшего его.

При выполнении проектирования программного обеспечения учитывалась теоретическая нагрузка, приведённая в Таблице 1.

Тестируемый процесс	Показатель	Средняя величина
SNMP-трапы	Максимальная пропускная способность агента	12 300/sec
	Максимальная пропускная способность сервера	3800/sec
	Задержка	12 ms
Запись пришедших данных	Максимальная пропускная способность	45 000/sec
	Задержка	1 ms
Уведомления о данных датчиков на удаленный агент	Максимальная пропускная способность	2150/sec
	Задержка	12 ms
Исторические данные	Пропускная способность	50 000 metrics/sec
	Скорость отображения данных	83 000

Тестируемый процесс	Показатель	Средняя величина
		metrics/sec
	Среднее время отклика при записи данных	201 ms
	Среднее время отклика при отображении данных	2 ms
	Размер записи	6 bytes
	Объём исторических данных за один год по одной метрике	3 MB
	Объём использованной памяти агента	55 MB
Rest-сервер	Скорость отображения данных	700 requests/sec
	Среднее время отклика	2 ms

Таблица 1

# Типовые решения



**ЦЕНТРАЛЬНЫЙ ПУЛЬТ**

## 4 Типовые решения

В процессе разработки системы были использованы следующие типовые решения:

- использование архитектуры Клиент-Сервер-Агент для построения системы мониторинга;
- использование архитектуры MVC (Model-View-Controller) для построение Web-приложения;
- использование формата JSON для передачи данных между клиентом, сервером и агентом;
- использование шаблона проектирования Message Bus для обмена сообщениями между модулями сервера.

**РОССИИНО**  
**[www.rossinno.net](http://www.rossinno.net)**  
**2019**