

# **ТЕХНИЧЕСКОЕ ПРЕДЛОЖЕНИЕ**

на разработку и внедрение  
универсальной системы диспетчеризации

# Оглавление

ВВЕДЕНИЕ.....	3
ОСНОВНАЯ ЧАСТЬ.....	4
1 Объект.....	5
2 Регистратор.....	6
2.1 Сервер-регистратор.....	7
2.2 СУБД.....	8
2.3 Web-сервер.....	9
2.4 Связь с GSM-модемом.....	10
2.5 Связь с АСКУЭ.....	10
2.6 Связь с клиентами.....	11
3 Облачный сервер.....	12
ПРИЛОЖЕНИЯ.....	15
ПРИЛОЖЕНИЕ А.....	15
Базовая модель.....	15
ПРИЛОЖЕНИЕ В.....	17
Базовая модель + Облачный сервер.....	17
ПРИЛОЖЕНИЕ Г.....	19
Примеры экранов проекта WebНМІ.....	19

## **ВВЕДЕНИЕ**

Данный документ содержит описание универсальной системы диспетчеризации.

Для построения системы используются:

- недорогие, доступные, взаимозаменяемые комплектующие,
- программное обеспечение с открытым исходным кодом.

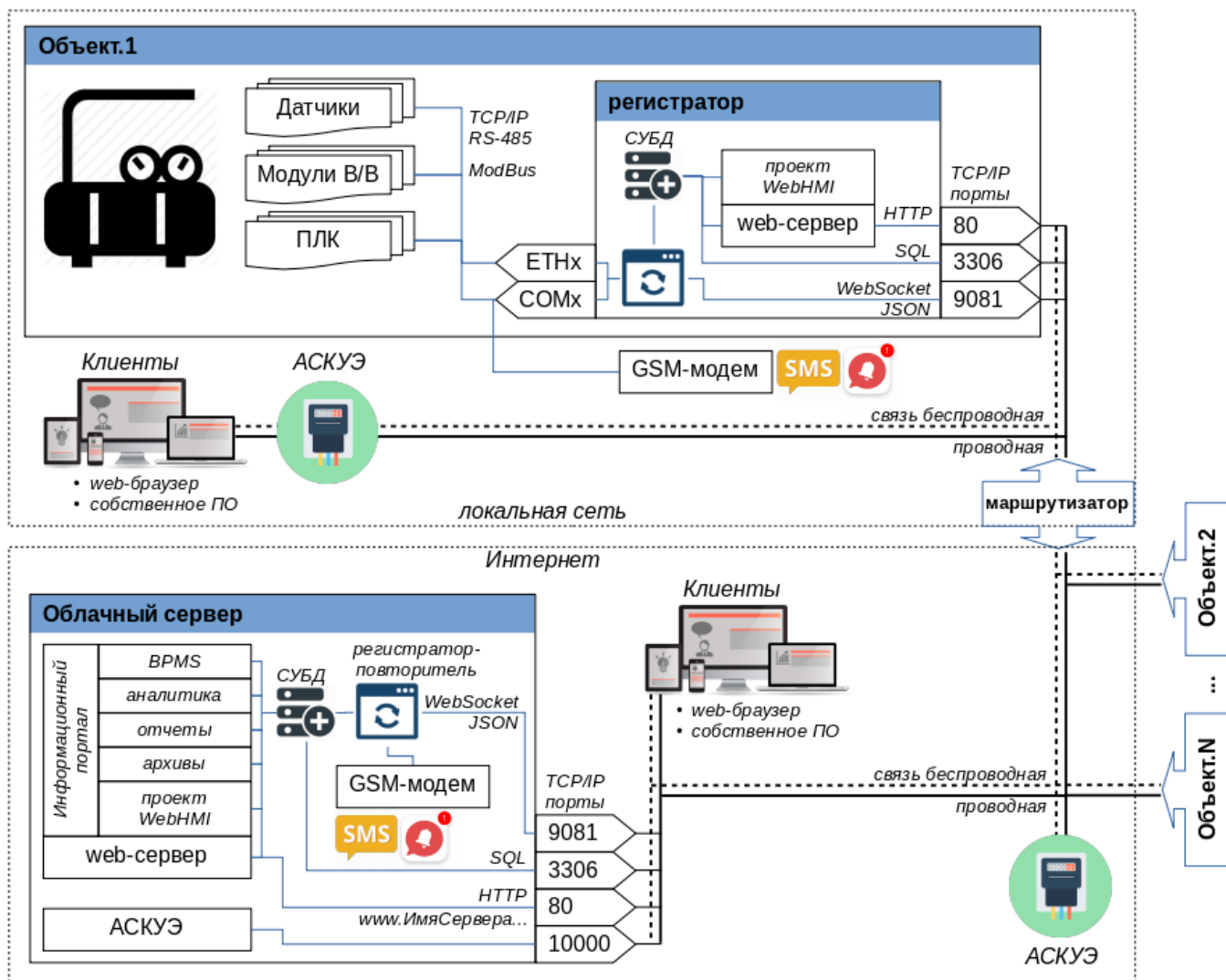
В основной части документа рассматривается полная функциональная схема универсальной системы диспетчеризации.

В Приложении А приводится вариант исполнения системы «Базовая модель».

В Приложении Б приводится вариант исполнения системы «Базовая модель + Облачный сервер».

В Приложении В приводятся примеры экранов проекта WebHMI.

# ОСНОВНАЯ ЧАСТЬ



функциональная схема универсальной системы диспетчеризации

Основные функции системы диспетчеризации:

- 1) сбор данных с целевого Объекта,
- 2) обработка полученных данных,
- 3) предоставление средств взаимодействия Клиента с целевым Объектом (визуализация, человеко-машинный интерфейс)

## 1 Объект



Под Объектом понимается аппаратная, программная или аппаратно-программная система, которая может предоставить доступ к своим компонентам по различным каналам передачи данных.

Аппаратные и аппаратно-программные системы — это системы, реализующие какой-либо технологический процесс (например, котельная, система вентиляции, компрессорная, термообработка и так далее). Подобные системы могут быть автоматизированными как частично, так и полностью. Автоматизация здесь, как правило, строится на базе аппаратно-программных регуляторов или программируемых логических контроллеров (ПЛК).

Программные системы — это готовые системы диспетчеризации, базы данных, системы учета технико-коммерческих показателей.

Объект может включать в себя:

- цифровые (дискретные), аналоговые и интеллектуальные датчики
- станции ввода/вывода
- систему управления (например, на базе ПЛК)
- исполнительные устройства
- сетевые каналы передачи данных
- хранилища, базы данных

Поддерживаемые каналы связи

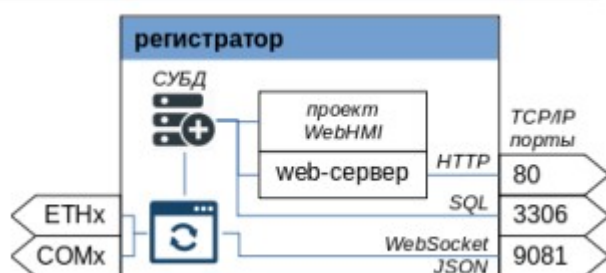
№ п.п.	Интерфейс
1	RS-232 / USB
2	RS-485
3	ETHERNET

Поддерживаемые протоколы передачи данных

№ п.п.	Протокол
1	ModBus RTU / TCP

Поддержка иных интерфейсов и/или протоколов связи может быть добавлена (реализована) по возможности.

## 2 Регистратор



Регистратор — это программная система (программа), осуществляющая с одной стороны связь с объектами, с другой — связь с клиентами.

### Компоненты регистратора

№ п.п.	Компонент	Особенности
1	<b>Сервер-регистратор</b>	
1.1	наличие:	обязательное
1.2	сетевой доступ:	есть
1.3	поставка:	исполняемое ПО + настройки
2	<b>Сервер баз данных (СУБД)</b>	
2.1	наличие:	необязательное
2.2	сетевой доступ:	есть
2.3	поставка:	исполняемое ПО + настройки
3	<b>Web-сервер</b>	
3.1	наличие:	необязательное
3.2	сетевой доступ:	есть
3.3	поставка:	исполняемое ПО + настройки
4	<b>Связь регистратора с GSM-модемом</b>	
4.1	наличие:	необязательное
5	<b>Связь регистратора с клиентами</b>	
5.1	наличие:	необязательное
6	<b>Связь регистратора с АСКУЭ</b>	
6.1	(автоматизированная система контроля и учета энергоресурсов) наличие:	необязательное

Все компоненты регистратора являются кроссплатформенными - имеются сборки под различные процессорные архитектуры и операционные системы.

## Требования к платформе регистратора:

№ п.п.	Требование
1	Аппаратная платформа
1.1	• Сервер
1.2	• Персональный компьютер
1.3	• Одноплатный компьютер (контроллер)
2	Процессорная платформа
2.1	• x86 (32-битный процессор)
2.2	• x64 (64-битный процессор)
2.3	• ARMv6, ...
3	Операционная система
3.1	• Linux
3.2	• Windows
4	Оперативная память (ОЗУ)
4.1	• от 512 Мб
5	Жесткий диск или FLASH-память
5.1	• от 8 Гб
6	Для доступа к компонентам регистратора из сети Интернет
6.1	• «белый» IP-адрес или прозрачный доступ в Интернет для регистратора
6.2	• проброс портов регистратора на маршрутизаторе
6.3	• использование Облачного сервера

### 2.1 Сервер-регистратор



Является обязательным программным компонентом Регистратора.

Программа реализована на языке C++ с использованием фреймворка Qt. Реализация открытая и кроссплатформенная: есть сборки под различные процессорные архитектуры и операционные системы, есть возможность передачи исходных кодов. Имеется гибкая система настроек (в виде файлов формата JSON).

Сервер-регистратор может быть запущен на любом персональном компьютере, сервере или одноплатном компьютере (контроллере), которые базируются на процессорных архитектурах x86, x64 или ARMv6 и выше, с операционными системами Windows, Linux.

#### Функции сервера-регистратора

№ п.п.	Функция
1	Периодический опрос объектов (получение данных) по поддерживаемым каналам связи

2	Обработка полученных данных
3	Передача данных в локальное или удаленное хранилище (СУБД)
4	Предоставление данных по сети:
4.1	• TCP/IP (Ethernet),
4.2	• сетевой порт (по-умолчанию, 9081),
4.3	• многопользовательский режим сервера,
4.4	• протокол WebSocket,
4.5	• формат JSON.
5	Связь с GSM-модемом:
5.1	• событийное оповещение (SMS),
5.2	• периодическое оповещение (SMS).
6	Связь с АСКУЭ:
6.1	• передача данных.

Поддержка иных архитектур, ОС и функционала может быть добавлена (реализована) по возможности.

Сервер-регистратор поставляется заказчику в виде исполняемой программы со всеми необходимыми настройками, инструкцией и описанием формата JSON для передачи данных. Возможна передача исходных кодов.

## 2.2 СУБД



СУБД (сервер баз данных) - готовая программная система, предназначенная для хранения данных и обработки запросов от множества клиентов. Работа с СУБД осуществляется стандартными запросами SQL.

Регистратор может как содержать в своем составе локальную СУБД, так и работать с внешними СУБД.

Локальная СУБД может использоваться для временного или постоянного хранения данных (как правило, данные целевого объекта). Объем локальной СУБД зависит от доступного объема памяти.

Связь с внешними СУБД осуществляется по протоколу TCP/IP (Ethernet).



## Поддерживаемые СУБД

№ п.п.	СУБД
1	MySQL
2	MariaDB

Поддержка иных СУБД может быть добавлена (реализована) по возможности.

СУБД поставляется заказчику в виде настроенной исполняемой программы. Исходные коды СУБД, в случае их открытости, доступны на сайте разработчика СУБД.

## 2.3 Web-сервер

проект WebHMI
web-сервер

Web-сервер — это готовая программная система, предоставляющая клиентам данные (web-контент, в том числе человеко-машинный интерфейс) по протоколу HTTP.

Клиенты — это web-браузеры.

Web-контент — информация, отображаемая в web-браузере.

Регистратор содержит в своем составе локальный web-сервер, который предоставляет клиентам web-контент (проект WebHMI) касательно контролируемого объекта.

## Поддерживаемые web-серверы

№ п.п.	web-сервер
1	Apache

## Функции web-сервера

№ п.п.	Функция
1	Предоставление данных по сети:
1.1	• TCP/IP (Ethernet),
1.2	• сетевой порт (по-умолчанию, 80),
1.3	• многопользовательский режим сервера,
1.4	• протокол HTTP (HTTPS), WebSocket.

Проект WebHMI (web human-machine interface) — web-интерфейс контролируемого объекта. Реализуется средствами web-разработки: JavaScript, AJAX, WebSocket, jQuery, jQuery UI, CSS5, HTML5.

## Функции проекта WebHMI

№ п.п.	Функция
1	Ограничение доступа
1.1	• без ограничений
1.2	• логин и пароль
2	Таблицы
2.1	• текущие данные,
2.2	• архивные данные
3	Мнемосхемы
3.1	• текущие данные,
3.2	• архивные данные
4	Графики
4.1	• текущие данные,
4.2	• архивные данные
5	Доступ к данным
5.1	• просмотр,
5.2	• изменение (дистанционное управление объектом)

### 2.4 Связь с GSM-модемом



Программная система, поддерживающая возможность рассылки SMS посредством GSM-модема.

GSM-модем может подключаться к аппаратной платформе Регистратора по одному из последовательных интерфейсов (зависит от модема и платформы регистратора): RS-232, USB, RS-485.

Отправка SMS:

- по настраиваемым событиям  
(например, получен сигнал закрытия газового клапана)
- по настраиваемому периоду  
(например, отправлять показание температуры котла раз в час)

### 2.5 Связь с АСКУЭ



АСКУЭ — автоматизированная система контроля и учета энергоресурсов (например, ЛЭРС Учет, Энергомера и т. п.).

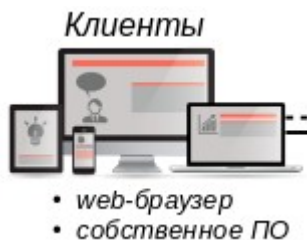
Регистратор может периодически отправлять данные в АСКУЭ. Связь с АСКУЭ осуществляется по протоколу TCP/IP (Ethernet).

## Поддерживаемые АСКУЭ

№ п.п.	АСКУЭ
1	ЛЭРС Учет
2	Энергомера

Оплата лицензии АСКУЭ осуществляется заказчиком отдельно по соответствующим тарифам.

### 2.6 Связь с клиентами



Клиент — персональный компьютер, ноутбук, нетбук, планшет, смартфон. Для контроля за объектом клиенту достаточно web-браузера, который связывается с web-сервером регистратора объекта. В браузере нужно ввести web-адрес регистратора.

Если заказчику недостаточно функционала web-браузера, то можно разработать собственное ПО — например, специализированную программу для смартфона или планшета (ОС Android), работающую с Регистратором по одному из поддерживаемых протоколов.

### 3 Облачный сервер

Характеристики предлагаемого облачного сервера

№ п.п.	Характеристика
1	Аппаратная платформа
1.1	<ul style="list-style-type: none"><li>Одноплатный компьютер Orange Pi PC2</li></ul>
2	Процессорная платформа
2.1	<ul style="list-style-type: none"><li>ARMv8 (64-битный процессор H5 Cortex-A53, 4-ядра)</li></ul>
3	Операционная система
3.1	<ul style="list-style-type: none"><li>Armbian Linux (64-битная, ядро 4.19)</li></ul>
4	Оперативная память (ОЗУ)
4.1	<ul style="list-style-type: none"><li>1 Гб</li></ul>
5	FLASH-память
5.1	<ul style="list-style-type: none"><li>32 Гб</li></ul>
	Жесткий диск для СУБД
5.2	<ul style="list-style-type: none"><li>1 Тб</li></ul>
6	Интернет-провайдер
6.1	<ul style="list-style-type: none"><li>прозрачный доступ в Интернет,</li></ul>
6.2	<ul style="list-style-type: none"><li>безлимитный тариф</li></ul>

В качестве облачного сервера может выступать любое устройство с характеристиками, поддерживаемыми компонентами платформы Регистратора.

Функции облачного сервера

№ п.п.	Функция
1	Регистратор-повторитель <ul style="list-style-type: none"><li>связь с внешними регистраторами</li><li>связь с внешними АСКУЭ</li><li>доступ из Интернет по выделенному доменному имени и порту</li></ul>
2	GSM-модем
2.1	<ul style="list-style-type: none"><li>2 СИМ-карты</li></ul>
2.2	<ul style="list-style-type: none"><li>отправка SMS по данным от регистратора-повторителя:</li></ul>
2.2.1	<ul style="list-style-type: none"><li><ul style="list-style-type: none"><li>по настроенным событиям,</li></ul></li></ul>
2.2.2	<ul style="list-style-type: none"><li><ul style="list-style-type: none"><li>периодически</li></ul></li></ul>
3	СУБД MariaDB 10.3
3.1	<ul style="list-style-type: none"><li>доступ из Интернет по выделенному доменному имени и порту</li></ul>
4	Web-сервер Apache 2.4
4.1	<ul style="list-style-type: none"><li>дистанционный web-доступ к СУБД через клиент phpMyAdmin,</li></ul>
4.2	<ul style="list-style-type: none"><li>предоставление контента web-проектов,</li></ul>
4.3	<ul style="list-style-type: none"><li>доступ из Интернет по выделенному доменному имени и порту</li></ul>

5 5.1	Проекты WebNMI удаленных регистраторов <ul style="list-style-type: none"> <li>• объединены в один «Информационный портал предприятия»</li> </ul>
6 6.1 6.2	Сервер бизнес-отчетности, аналитики и интеграции данных Pentaho BI <ul style="list-style-type: none"> <li>• web-отчеты с экспортом в печатные форматы,</li> <li>• web-аналитика,</li> <li>• интеграция, обработка данных</li> <li>• доступ из Интернет по выделенному доменному имени и порту</li> </ul>
7 7.1 7.2	Сервер бизнес-процессов ProcessMaker 3 <ul style="list-style-type: none"> <li>• моделирование бизнес-процессов в нотации BPMN 2.0,</li> <li>• исполнение бизнес-процессов,</li> <li>• web-клиент</li> <li>• доступ из Интернет по выделенному доменному имени и порту</li> </ul>
8	АСКУЭ

Функционал облачного сервера может быть расширен.

### **Когда нужен облачный сервер?**

#### **Когда регистратор объекта находится в сети провайдера с «непрозрачным» доступом в Интернет:**

- а) есть свободный доступ в Интернет со стороны локальной сети регистратора  
*пример: регистратор может свободно подключиться к удаленному облачному-серверу в сети Интернет*
- б) нет доступа из сети Интернет к сетевым компонентам регистратора  
*пример: web-браузер клиента не может подключиться из сети Интернет к web-серверу удаленного регистратора*

Как это работает:

1. В настройках регистратора прописывается доменное имя (или IP-адрес) и номер сетевого порта регистратора-повторителя облачного сервера;
2. Регистратор, при работе, автоматически подключается к повторителю облачного сервера;
3. Регистратор, при появлении данных от объекта, автоматически пересылает их копию повторителю облачного сервера  
ИЛИ  
После того, как регистратор подключился к повторителю облачного сервера, повторитель сам опрашивает регистратор по установленному каналу связи;
4. Повторитель облачного сервера принимает данные от регистратора и дальше может делать с ними тоже самое, что и обычный регистратор:
  - а) обрабатывать по заданным алгоритмам,

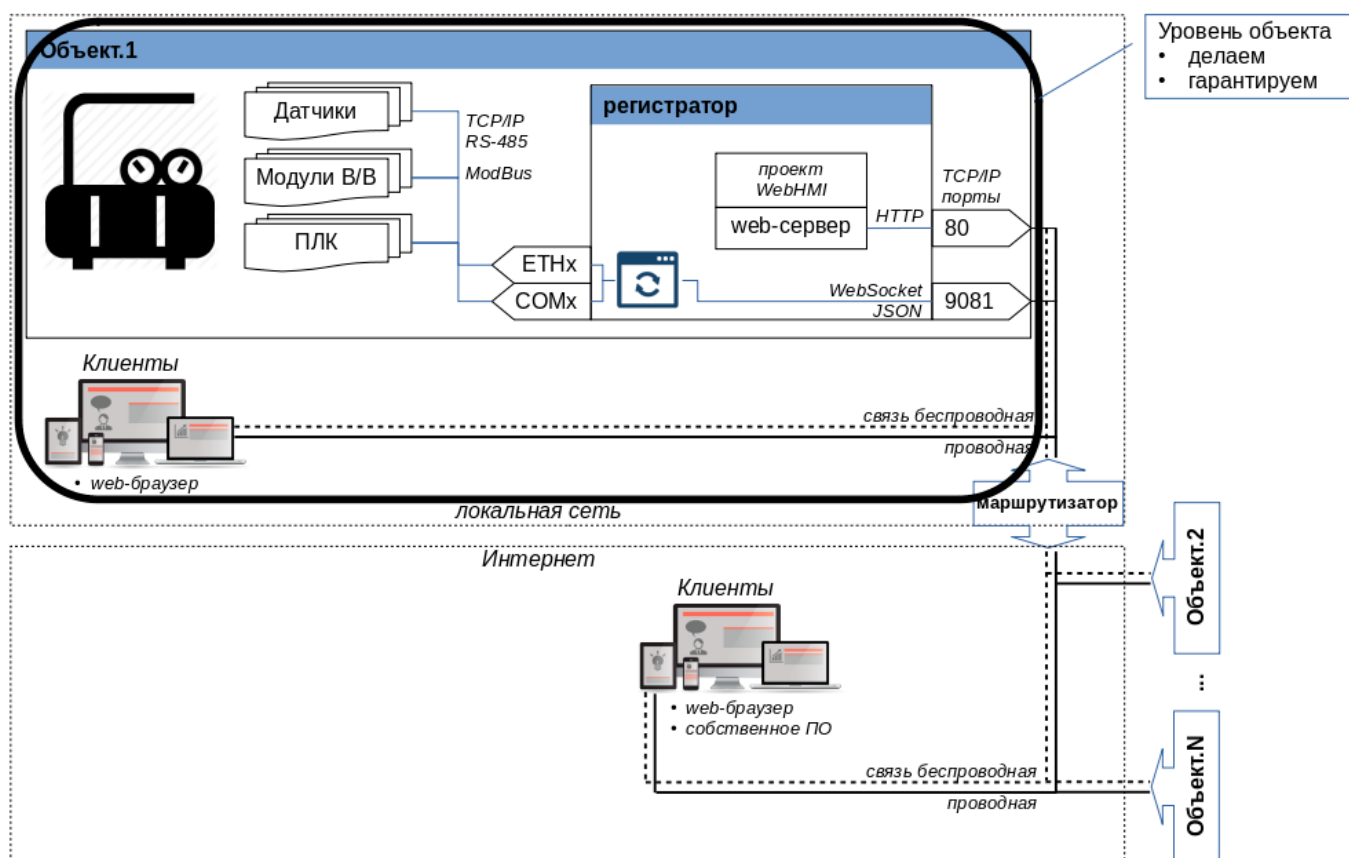
- б) передавать в хранилище данных,
  - в) предоставлять интернет-клиентам данные по протоколу WebSocket или через web-сервер,
  - г) отправлять SMS,
  - д) и так далее.
5. Один повторитель облачного сервера может работать с несколькими удаленными регистраторами;
6. Повторителей на облачном сервере может быть несколько (для каждого может быть выделен свой сетевой порт).

**Когда у заказчика нет возможности своими силами сопровождать сетевые компоненты регистратора объекта (например, СУБД, АСКУЭ, GSM-модем).**

# ПРИЛОЖЕНИЯ

## ПРИЛОЖЕНИЕ А

### Базовая модель



Базовая модель системы диспетчеризации

Функции базовой модели системы для одного объекта:

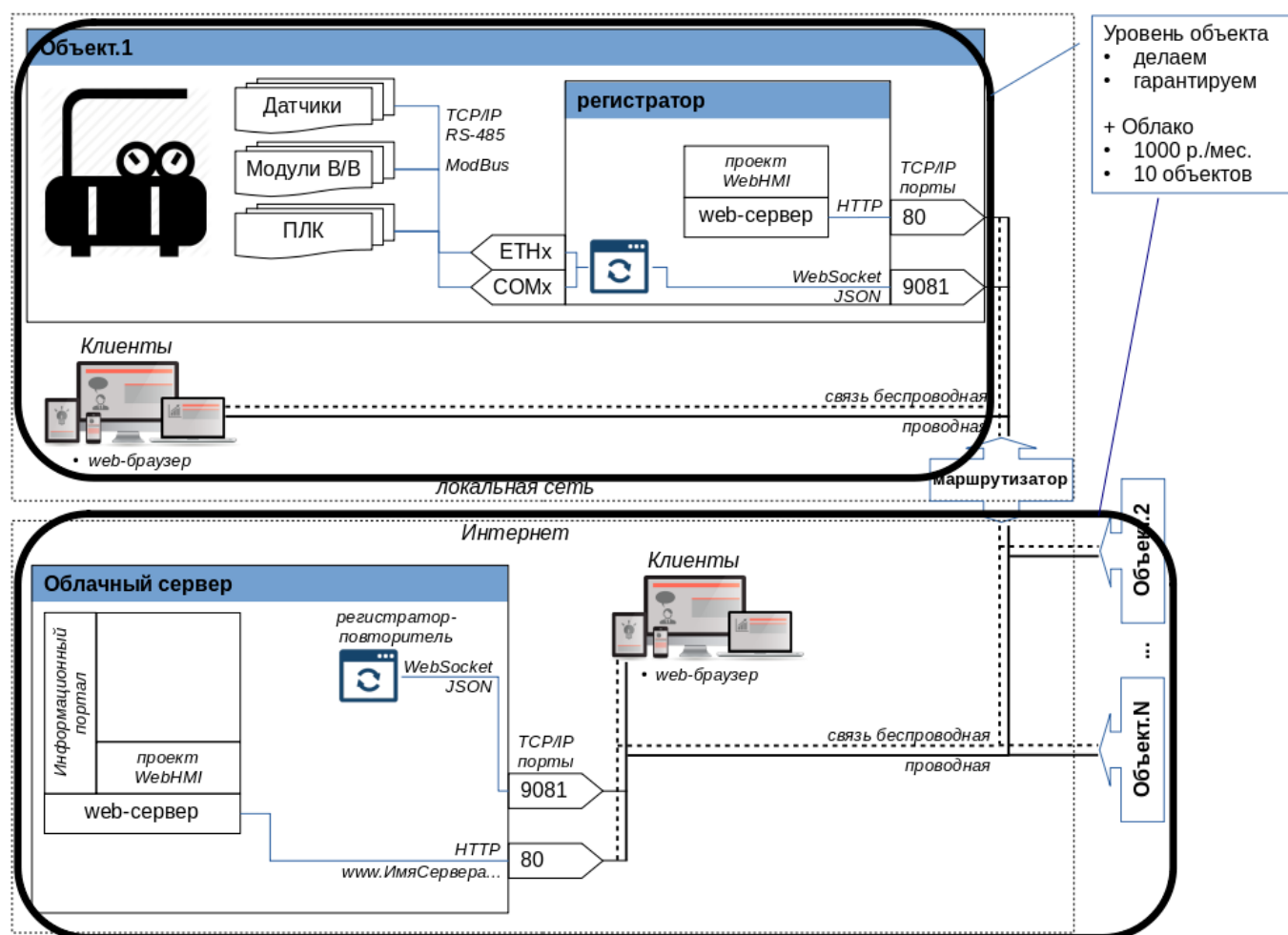
№ п.п.	Функция
1	Объект
1.1	• может содержать:
1.2	○ датчики
1.3	○ модули ввода/вывода
1.3	○ систему управления (ПЛК)
	• дополнительно могут потребоваться:
1.5	○ датчики
1.6	○ модули ввода/вывода
	• поддерживаемые сети:
1.7	○ RS-485
1.8	○ TCP/IP (Ethernet)
	• поддерживаемые протоколы:
1.9	○ ModBus RTU / TCP

N п.п.	Функция
2 2.1 2.1.1 2.1.2  2.2 2.2.1  2.3 2.3.1	Регистратор <ul style="list-style-type: none"> <li>• сервер-регистратор <ul style="list-style-type: none"> <li>◦ сбор данных с объекта</li> <li>◦ предоставление данных клиентам <ul style="list-style-type: none"> <li>▪ TCP/IP (Ethernet)</li> <li>▪ протокол WebSocket</li> <li>▪ сетевой порт 9081</li> </ul> </li> </ul> </li> <li>• web-сервер <ul style="list-style-type: none"> <li>◦ предоставление web-доступа к проекту WebHMI <ul style="list-style-type: none"> <li>▪ TCP/IP (Ethernet)</li> <li>▪ протокол HTTP</li> <li>▪ сетевой порт 80</li> </ul> </li> </ul> </li> <li>• проект WebHMI <ul style="list-style-type: none"> <li>◦ web-интерфейс <ul style="list-style-type: none"> <li>▪ без ограничения доступа</li> <li>▪ страница 1: таблица основных значений</li> <li>▪ страница 2: мнемосхема</li> <li>▪ только текущие значения</li> <li>▪ только просмотр</li> </ul> </li> </ul> </li> </ul>
3 3.1	Клиенты <ul style="list-style-type: none"> <li>• контроль объекта через web-браузер (только просмотр)</li> </ul>



## ПРИЛОЖЕНИЕ В

### Базовая модель + Облачный сервер



Базовая модель системы диспетчеризации и аренда облачного сервера

Функции базовой модели системы для одного объекта:

№ п.п.	Функция
1	Объект
1.1	• может содержать:
1.2	○ датчики
1.3	○ модули ввода/вывода
1.5	• дополнительно могут потребоваться:
1.6	○ датчики
1.7	○ модули ввода/вывода
1.8	• поддерживаемые сети:
1.7	○ RS-485
1.8	○ TCP/IP (Ethernet)

N п.п.	Функция
1.9	<ul style="list-style-type: none"> <li>поддерживаемые протоколы: <ul style="list-style-type: none"> <li>ModBus RTU / TCP</li> </ul> </li> </ul>
2 2.1 2.1.1 2.1.2  2.2 2.2.1  2.3 2.3.1	Регистратор <ul style="list-style-type: none"> <li>сервер-регистратор <ul style="list-style-type: none"> <li>сбор данных с объекта</li> <li>предоставление данных клиентам <ul style="list-style-type: none"> <li>TCP/IP (Ethernet)</li> <li>протокол WebSocket</li> <li>сетевой порт 9081</li> </ul> </li> </ul> </li> <li>web-сервер <ul style="list-style-type: none"> <li>предоставление web-доступа к проекту WebHMI <ul style="list-style-type: none"> <li>TCP/IP (Ethernet)</li> <li>протокол HTTP</li> <li>сетевой порт 80</li> </ul> </li> </ul> </li> <li>проект WebHMI <ul style="list-style-type: none"> <li>web-интерфейс <ul style="list-style-type: none"> <li>без ограничения доступа</li> <li>страница 1: таблица основных значений</li> <li>страница 2: мнемосхема</li> <li>только текущие значения</li> <li>только просмотр</li> </ul> </li> </ul> </li> </ul>
3 3.1	Клиенты <ul style="list-style-type: none"> <li>контроль объекта через web-браузер (только просмотр)</li> </ul>
4 4.1 4.2  4.3	Облачный сервер <ul style="list-style-type: none"> <li>регистратор-повторитель <ul style="list-style-type: none"> <li>поддержка до 10 регистраторов объектов</li> </ul> </li> <li>web-сервер <ul style="list-style-type: none"> <li>предоставление web-доступа к проекту WebHMI из сети Интернет <ul style="list-style-type: none"> <li>TCP/IP (Ethernet)</li> <li>протокол HTTP</li> <li>сетевой порт 80</li> <li>собственное доменное имя <i>например: www.my-cloud.no-ip.org</i></li> </ul> </li> </ul> </li> <li>проекты WebHMI всех поддерживаемых регистраторов объектов, объединенные в один «Информационный портал»</li> </ul>

## ПРИЛОЖЕНИЕ Г

### Примеры экранов проекта WebNMI

Котельная

Объект #1

отопление

Т подачи	75.0	С°
Р подачи	4.8	бар
Т возвр.	45.0	С°
Р возвр.	4.1	бар

местное

Т	31.6	С°
---	------	----

состояние

норма

[подробно](#)

Объект #2

отопление

Т подачи	75.0	С°
Р подачи	4.7	бар
Т возвр.	44.9	С°
Р возвр.	4.2	бар

местное

Т	31.4	С°
---	------	----

состояние

норма

[подробно](#)

Объект #3

отопление

Т подачи	74.8	С°
Р подачи	4.7	бар
Т возвр.	45.0	С°
Р возвр.	4.1	бар

местное

Т	31.2	С°
---	------	----

контур котлов

Т	79.5	С°
---	------	----

ГВС

Т подачи	61.2	С°
Т возвр.	45.0	С°
Р	4.0	бар

состояние

норма

[подробно](#)

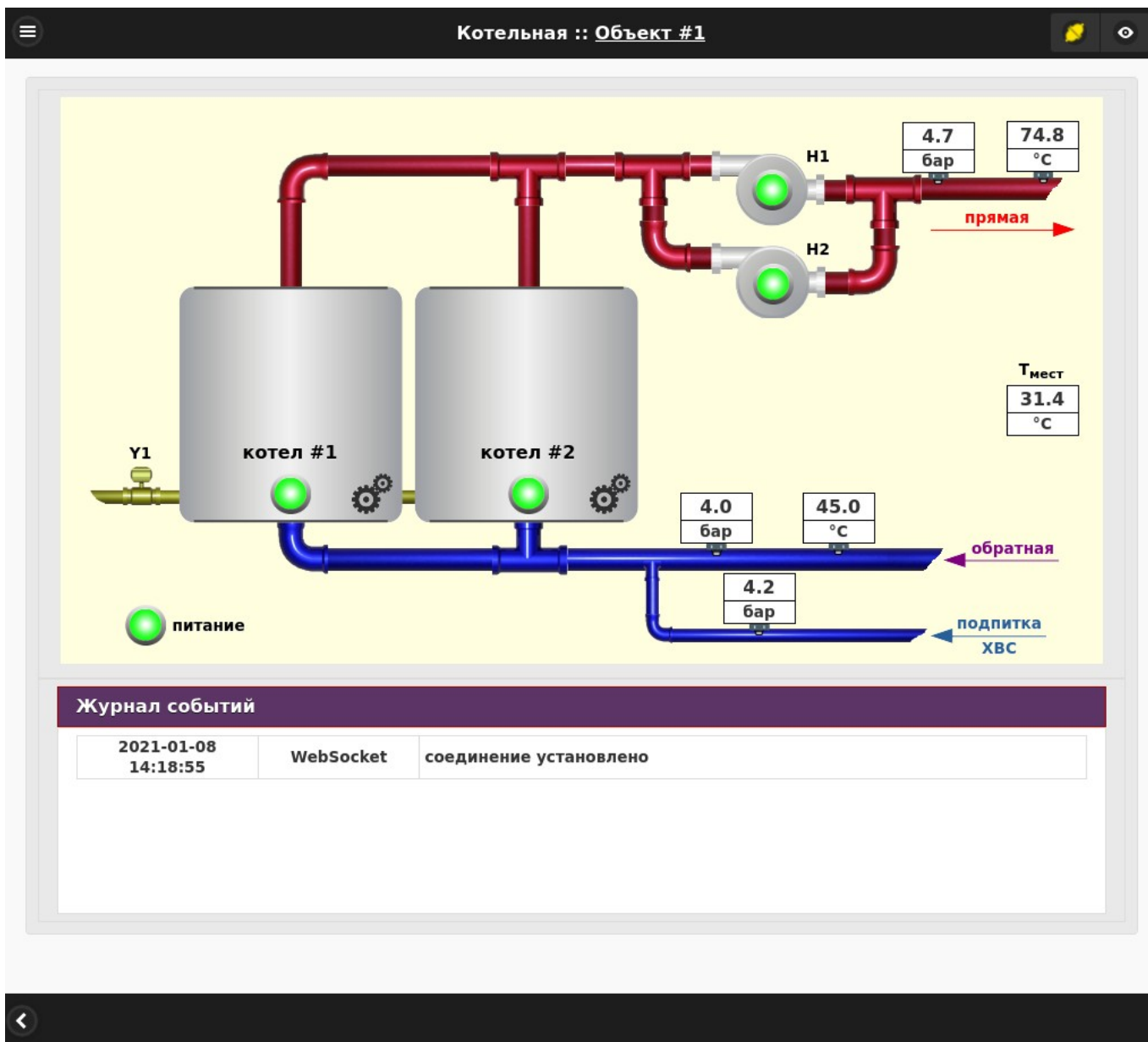
Журнал событий

2021-01-08 14:21:19

WebSocket

соединение установлено

Таблицы основных значений трех объектов



Мнемосхема объекта