

ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ «АЙ-НОВУС»  
(ООО «АЙ-НОВУС»)

**СЕРВИС УПРАВЛЕНИЯ НАСТРОЙКАМИ  
«АЙГАЧ»**

**ОПИСАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

На 10 листах

## **АННОТАЦИЯ**

Данный документ предназначен для описания функциональности и особенностей Сервиса управления настройками «Айгач». В документе рассматриваются основные функции и преимущества сервиса, включая хранение и управление конфигурационными данными в формате «ключ-значение».

## СОДЕРЖАНИЕ

Термины и сокращения .....	4
1 Общие сведения .....	6
1.1 Обозначение и наименование программного обеспечения .....	6
1.2 Минимальные системные требования .....	6
1.3 Языки программирования, на которых написан Сервис.....	6
2 Функциональное назначение .....	7
3 Структура и функционирование Сервиса.....	8
4 Взаимодействие с Сервисом .....	10

## ТЕРМИНЫ И СОКРАЩЕНИЯ

Перечень принятых терминов и сокращений приведен в таблице 1.

Таблица 1 – Перечень принятых терминов и сокращений

Термин/ Сокращение	Определение/Расшифровка
API	(англ. Application Programming Interface – программный интерфейс приложения) – набор готовых классов, процедур, функций, структур и констант, предоставляемых приложением (библиотекой, сервисом) для использования во внешних программных продуктах
CRUD	(англ. Create, Read, Update, Delete) – акроним, обозначающий четыре базовые функции, используемые при работе с базами данных: создание, чтение, модификация, удаление
HTTP	(англ. HyperText Transfer Protocol – протокол гипертекстовой передачи) – протокол прикладного уровня передачи данных, изначально — в виде гипертекстовых документов в формате HTML, в настоящее время используется для передачи произвольных данных
HTTP API	(англ. Hypertext Transfer Protocol Application Programming Interface) – интерфейс программирования приложений, который позволяет программам взаимодействовать между собой через протокол HTTP
JSON	(англ. JavaScript Object Notation – обозначение объектов JavaScript) – текстовый формат обмена данными, основанный на JavaScript
RESTful	Способ взаимодействия сайтов и веб-приложений с сервером
Агент	Долго работающий фоновый процесс, который выполняет системную задачу на каждом элементе кластера Сервиса. Он запускается путем запуска Агента Сервиса. Агент может работать как в клиентском, так и в серверном режиме. Поскольку на всех узлах должен быть запущен агент, то проще ссылаться на узел как на клиент или сервер, но существуют и другие экземпляры агента. Все агенты могут запускать интерфейсы DNS или HTTP и отвечают за выполнение проверок и синхронизацию служб
Гб	Гигабайт, единица измерения количества информации
Гц	Гигагерц, единица частоты периодических процессов
Кб	Килобайт, единица измерения количества информации
Клиент	Агент, который пересылает все RPC на сервер. Клиент не фиксирует текущее состояние. Единственная фоновая активность, которую выполняет клиент – участие в кластере LAN Gossip. Это приводит к минимальному потреблению ресурсов и потребляется лишь небольшая часть пропускной способности сети
ОЗУ	Оперативная память, энергозависимая часть системы компьютерной памяти, в которой во время работы компьютера хранится выполняемый машинный код, а также входные, выходные и промежуточные данные, обрабатываемые процессором
Сервер	Агент с расширенным набором функций, включая участие в кворуме Raft, поддержание состояния кластера, реагирование на запросы

<b>Термин/ Сокращение</b>	<b>Определение/Расшифровка</b>
	RPC, обмен WAN Gossip с другими центрами обработки данных и пересылку запросов лидерам или удаленным центрам обработки данных
Сервис, Айгач	Сервис управления настройками «Айгач»
Токен	Уникальная последовательность символов (букв, цифр и символов), содержащая исчерпывающие данные для идентификации конкретного пользователя или системы
Хранилище ключей и значений	База данных ключей и значений, представляет собой модель данных, в которой каждый ключ связан с одним и только одним значением в наборе
Центр обработки данных	Сетевая среда, которая является частной, с низким временем отклика и высокой пропускной способностью

## **1 ОБЩИЕ СВЕДЕНИЯ**

### **1.1 Обозначение и наименование программного обеспечения**

Полное наименование системы: Сервис управления настройками «Айгач».

Краткое наименование системы: Сервис, Айгач.

### **1.2 Минимальные системные требования**

Для обеспечения применения Сервиса необходимо наличие следующих минимальных технических средств:

- процессор с количеством ядер в диапазоне от 2 до 4;
- ОЗУ не менее 8 Гб;
- диск объемом не менее 100 Гб;
- для диска не менее 3000 операций ввода-вывода в секунду;
- пропускная способность диска не менее 75 МБ/с.

### **1.3 Языки программирования, на которых написан Сервис**

Язык программирования: Go (Golang) и JavaScript.

## 2 ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ

Сервис управления настройками предназначен для управления хранилищем, содержащим настройки распределенных сервисов. Хранилище представляет собой базу «ключ-значение», которое позволяет хранить параметры конфигурации, метаданные, индексированные объекты и прочие настройки.

Задачами Сервиса являются:

- 1) обеспечение единого хранилища настроек распределенных сервисов;
- 2) обеспечение простого и универсального доступа к хранилищу настроек распределенных сервисов.

Сервис управления настройками предоставляет следующие функциональные возможности:

- хранение древовидной базы данных настроек в формате «ключ-значение»;
- управление базой данных настроек;
- взаимодействие с базой данных настроек через командную строку для выполнения команд верхнего уровня;
- взаимодействие с базой данных настроек через пользовательский интерфейс;
- взаимодействие с базой данных настроек через HTTP API;
- создание токена с привилегиями для доступа к базе данных настроек.

### 3 СТРУКТУРА И ФУНКЦИОНИРОВАНИЕ СЕРВИСА

Структура Сервиса представлена на рисунке 1.

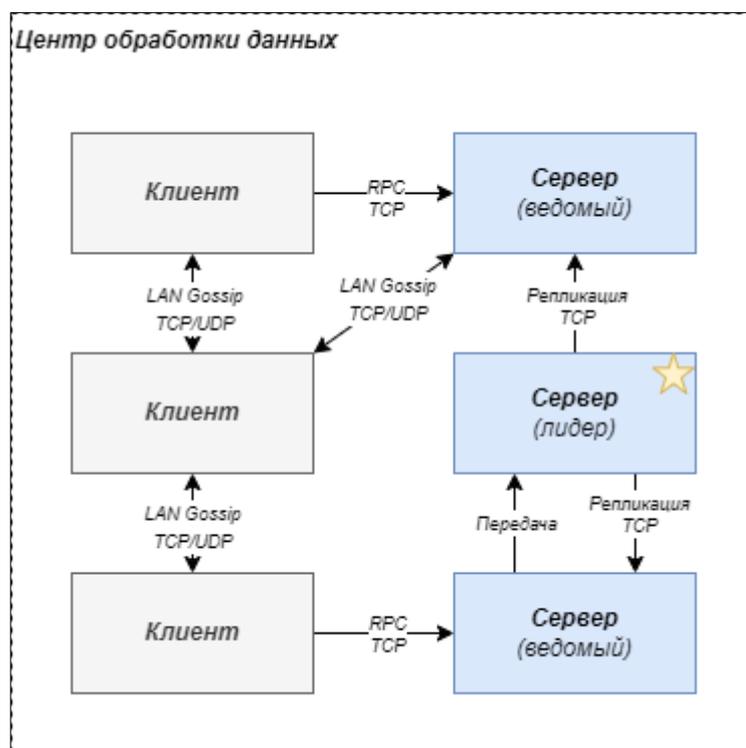


Рисунок 1 – Структура сервиса управления настройками

Центр обработки данных – закрытая сетевая среда с низкой задержкой и высокой пропускной способностью. Это исключает связь, которая будет проходить через общедоступный Интернет.

Коллекция агентов Сервиса, которые знают друг о друге, называется кластером Сервиса.

Агент – это фоновый процесс, запущенный на каждом узле кластера Сервера. Агент может работать как в режиме клиента, так и в режиме сервера. Поскольку на всех узлах должен быть запущен агент, проще называть узел Клиентом или Сервером.

Клиент – это агент, который перенаправляет все вызовы RPC на сервер.

Сервер – это агент с расширенным набором обязанностей, включая участие в распределении лидера и ведомых серверов, поддержание состояния кластера, ответ на запросы RPC.

Агент Сервиса может работать в двух различных режимах – Сервер и Клиент.

Агент Сервера хранит всю информацию о состоянии, включая IP-адреса узлов, проверки работоспособности и конфигурацию.

Хранение объектов в «Айгач»:

- Непрозрачность (opacity): «Айгач» не накладывает ограничений на тип объекта, хранящегося в записи ключ-значение, обеспечивая полную гибкость.

- Ограничение по размеру: основное ограничение на объект – это его размер, максимальный размер составляет 512 КБ.
- Экономия ресурсов: благодаря ограничению по размеру и основным вариантам использования, обычно не требуется дополнительное хранилище.

#### Организация данных в «Айгач»:

- Отсутствие ограничений по типу: ключи, как и объекты, не ограничены по типу и могут включать любые символы.
- Рекомендации по использованию символов:
  - Использовать URL-безопасные символы: для безопасности и совместимости рекомендуется использовать URL-безопасные символы – [a-zA-Z0-9-.\_~], за исключением /, который может использоваться для организации данных.
  - Избегать проблемных символов: Для предотвращения проблем при использовании API и в shell-скриптах рекомендуется избегать использования символов \*, ?, ' и %.

## 4 ВЗАИМОДЕЙСТВИЕ С СЕРВИСОМ

Сервис предоставляет несколько способов вызова (доступа к данным), что позволяет пользователям и приложениям взаимодействовать с системой хранения конфигурационных данных. Доступ можно ограничить с помощью настройки привилегий по токену.

Основные способы вызова Сервиса с соответствующими носителями данных:

- 1) RESTful HTTP API позволяет выполнять операции над парами «ключ-значение». Это основной способ обращения к сервису. API может выполнять базовые операции CRUD на узлах, сервисах, проверках, конфигурации и т. д. По умолчанию вывод всех запросов HTTP API осуществляется в виде JSON;
- 2) командная строка. Сервис предоставляет клиентский интерфейс командной строки, который позволяет выполнять команды для взаимодействия с хранилищем;
- 3) пользовательский веб-интерфейс позволяет взаимодействовать с хранилищем ключей и значений, обеспечивая легкий доступ к функциям управления настройками.