

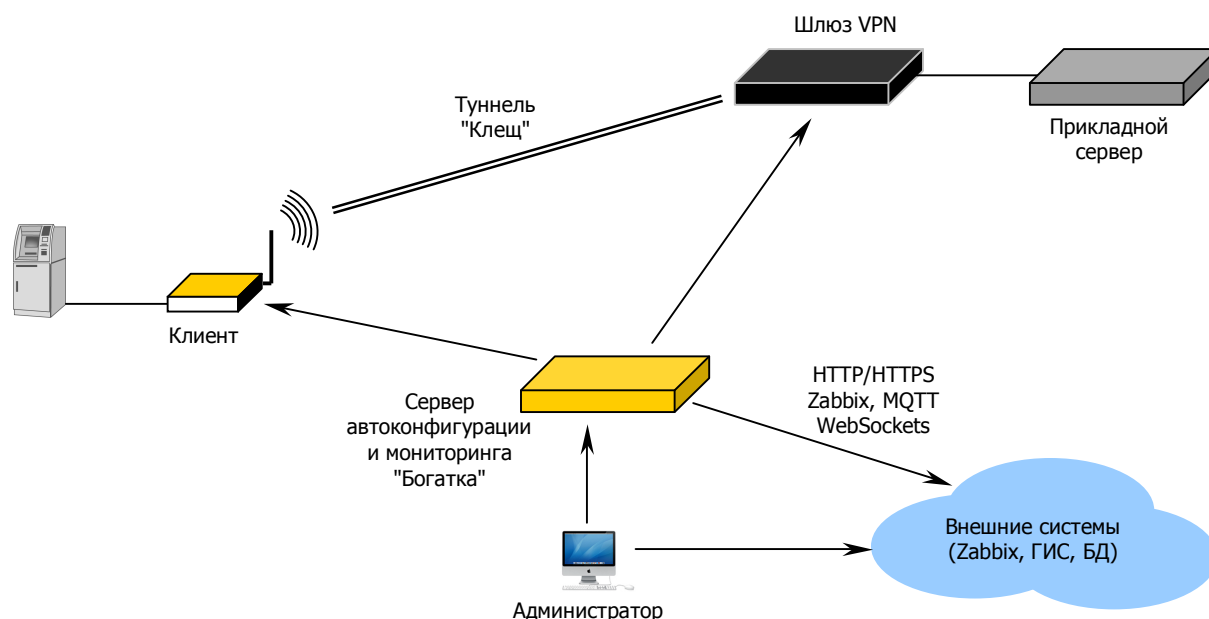
БОГАТКА

Система SD-WAN для массового подключения удалённых объектов



"Богатка" — программно-аппаратный комплекс для массового подключения однотипных удалённых объектов, таких как банкоматы, базовые станции сотовых сетей и сетей IoT, технологические площадки в различных отраслях промышленности и ЖКХ, и т.п. Решение построено на принципах программно-определяемых глобальных сетей (Software-Defined WAN, SD-WAN) и обеспечивает:

- Автоматическое конфигурирование клиентских устройств и ядра сети по принципу Zero Touch Provisioning (в переводе на русский — "хочу, чтобы оно всё настроилось само");
- Бесперебойную доставку данных по неустойчивым каналам связи;
- Защиту данных при передаче по сети с использованием современных протоколов и алгоритмов безопасности;
- Централизованный мониторинг и управление.



Система предназначена для работы с большим числом удалённых узлов (*клиентов*) — тысячи и десятки тысяч. Предполагается, что узлы оснащены оборудованием одного или нескольких типов, с несколькими типовыми аппаратными конфигурациями, программными настройками и способом подключения. "Богатка" берёт на себя все типовые операции и настройки, выполняемые единообразно на многих устройствах, и оставляет администратору только то, что относится индивидуально к каждому клиенту.

Автоматическая настройка клиентских узлов

Процедура подключения клиентских устройств к сети "Богатка" упрощена в максимально возможной степени. На самой площадке при этом никакие настройки не выполняются — вся конфигурация производится сервером и, отчасти, администратором в центре. Установка не требует высокой квалификации персонала на местах, поскольку

вообще не предполагает доступа к интерфейсу управления устройством. Задача техника состоит в том, чтобы физически установить устройство и подключить к нему нужные кабели. В самом крайнем случае от него может потребоваться сбросить устройство в заводскую конфигурацию путём определённых манипуляций с кнопками и питанием. (Процедура искусственно усложнена, чтобы её нельзя было выполнить случайно.)

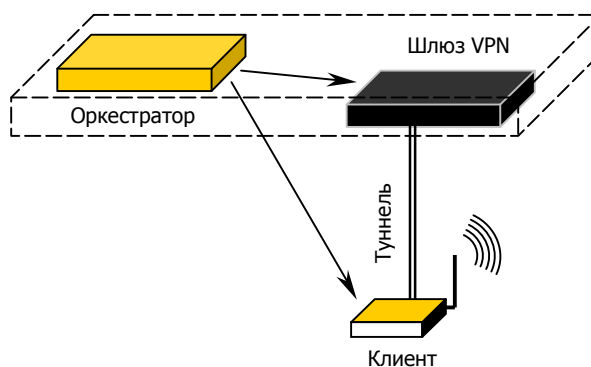
Устройства поставляются заказчику с подготовленной ("кастомизированной") заводской конфигурацией. На сервере заранее составляется список клиентов и типовых шаблонов для них. После включения устройство выходит в Интернет без дополнительных настроек, ищет доступные сервера по заданному списку, и запрашивает рабочую конфигурацию по HTTPS.

Сервер генерирует индивидуальную конфигурацию клиента на основе заданных шаблонов и уникальных параметров данного клиента (в простейшем случае — модели и заводского номера). При этом максимально возможное число параметров вычисляется алгоритмически. (Например, рекомендуется иметь для сети единый адресный план, в котором все IP-адреса клиентов формируются однозначно по их заводским номерам.) Это не только упрощает работу администратора системы, но и исключает возможные источники человеческих ошибок. Вручную следует вводить только те параметры, которые не могут быть известны системе по существу.

Создание туннелей

Построение VPN — это, как правило, согласованная настройка как клиентской, так и серверной стороны каждого туннеля. Сложность и трудоёмкость этой работы оставляет широкий простор для человеческих ошибок.

"Богатка" осуществляет настройку обеих сторон в рамках одной процедуры регистрации клиента. Строго говоря, первичными объектами управления в системе являются не устройства — клиенты и шлюзы VPN — а туннели как целое.



Каждый сервер "Богатка" сочетает в себе две функции: собственно оркестратора "Богатка" для управления системой и шлюза VPN для туннелей "Клещ". Принципиально возможно и разнести эти две функции на различные физические или виртуальные машины, но в данном случае это не имеет смысла, поскольку "Клещ" нецелесообразно настраивать иными способами.

Бесперебойные туннели L3/L2

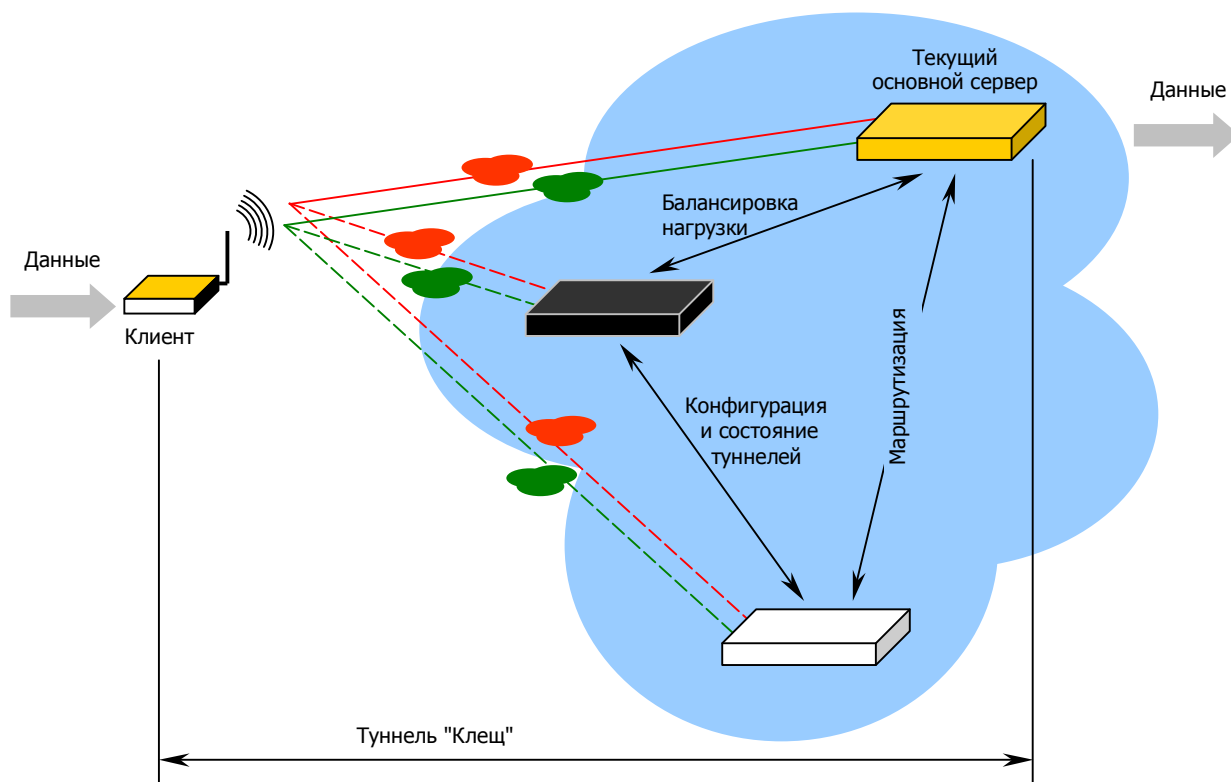
Система "Богатка" изначально создавалась для использования совместно с другой разработкой NSG — фирменной VPN "Клещ" для критически важных приложений. Туннели "Клещ" обеспечивают бесперебойную передачу данных по нескольким неустойчивым каналам связи, в том числе без потерь при переходе с одного канала на другой. Число и тип каналов связи не ограничены, это могут быть, например, несколько сотовых операторов, или наземный плюс сотовый, и т.д. Клиент "Клещ" устанавливает соединения с сервером по всем доступным каналам связи, следит за их работоспособностью, при необходимости рестартует сотовые модули (не прерывая при этом работу устройства в целом).

Дисциплина работы с несколькими каналами предусматривает горячее резервирование с автоматическим выбором лучшего канала.

С точки зрения сетевой топологии, туннели "Клещ" представляют собой виртуальные соединения типа "точка-точка". В зависимости от конфигурации, они могут работать как интерфейсы типа *tun* (Layer 3) или типа *tap* (Layer 2). Эти интерфейсы пригодны для передачи любых типов данных и доступны для всех стандартных сетевых процедур, таких как маршрутизация или QoS.

Распределённое ядро системы

Архитектура "Богатки" изначально подразумевает, что в системе имеется несколько серверов, для резервирования и распределения нагрузки, притом установлены они на территориально разнесённых площадках. Доступ к системе может осуществляться с любого сервера, одновременно несколькими пользователями. Сервера используют единую распределённую базу данных о туннелях, их конфигурациях (для обеих сторон) и текущем состоянии. Эта база синхронизируется между ними в реальном времени, без участия оператора, и в каждый момент содержит на каждом сервере актуальную информацию о состоянии всей системы.



С точки зрения VPN, клиент устанавливает и поддерживает соединения по каждому из N каналов связи к каждому из M серверов — итого $N \times M$ соединений. Данные передаются только через один из серверов, назначенный системой в качестве основного. Использование нескольких каналов связи к одному и тому же серверу регламентируется выбранной дисциплиной, см. выше. Вне зависимости от дисциплины и фактического состояния каналов связи, туннель "Клещ" представляется всегда как одно соединение "точка-точка" между клиентом и его основным сервером.

Если основной сервер выходит из строя или планово выключается (например, перезагружается для обновления ПО), то клиенты переводятся с него на другие сервера.

Между серверами работает внутренний протокол маршрутизации, определяющий маршруты на клиентов. База этих маршрутов, как и база "Богатки", синхронизируется автоматически и непрерывно. Благодаря этому, пакет в сторону клиента может быть передан через любой из шлюзов — тот самостоятельно определит основной сервер для данного клиента и переадресует ему пакет.

Безопасность данных

Все данные, передаваемые внутри туннелей "Клещ", защищены с помощью современных алгоритмов, соответствующих TLS 1.3.

Возможно также полностью заменить это шифрование стандартными типами VPN на туннельных интерфейсах.

Масштабируемость и резервируемость

Эффективные программные средства позволяют полностью задействовать потенциал современных многоядерных серверных платформ. Благодаря этому сервера "Богатки" способны обслуживать большое число клиентов даже в одиночку*, хотя этого, очевидно, недостаточно с точки зрения резервирования и отказоустойчивости.

Распределённая архитектура серверного ядра позволяет расширять систему практически неограниченно — путём добавления новых серверов. Для подключения нового сервера администратору не приходится прилагать больших усилий: распределённая база данных автоматически переносит на него всю информацию о других серверах, шлюзах VPN, клиентах, их текущих состояниях, а также о пользователях системы.

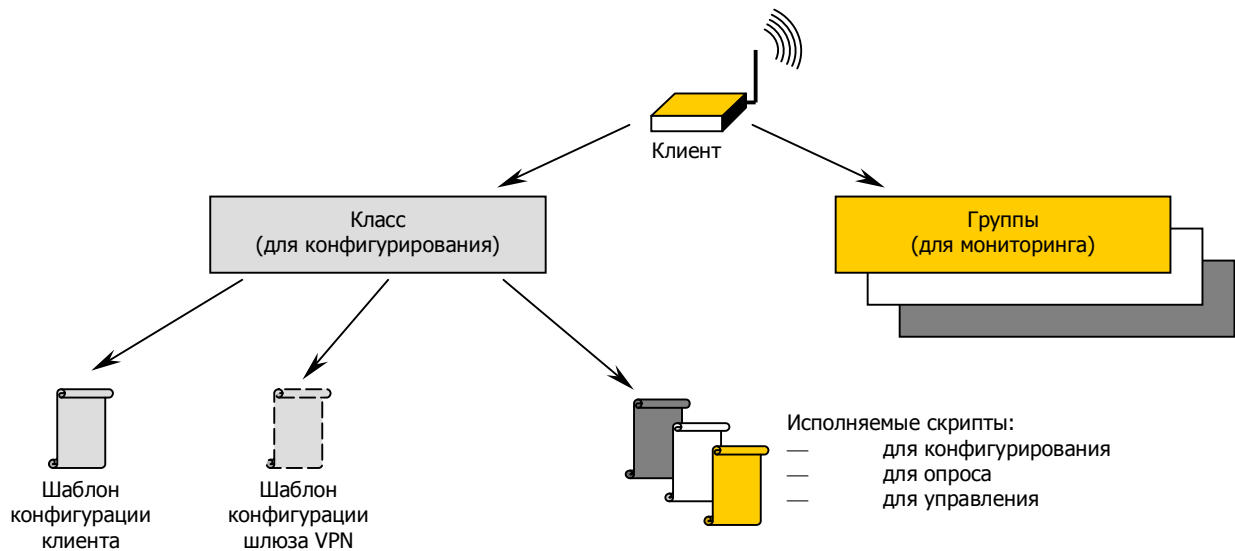
В функции "Богатки" в распределённой системе входит также балансировка нагрузки между серверами. Если какой-то сервер включается в работу, выбывает из строя, испытывает чрезмерную или недостаточную нагрузку, система автоматически перераспределяет клиентов между серверами. Благодаря этому общее число серверов в системе, необходимое для её отказоустойчивости, определяется по принципу N+1.

Конфигурация: классы и группы

С точки зрения конфигурации, все клиенты разбиваются на *классы*. Класс имеет один шаблон конфигурации для всех туннелей и один набор скриптов для управления и мониторинга. При этом каждый клиент системы принадлежит к одному и только одному классу. Как правило, в качестве класса целесообразно определять устройства одной аппаратной модели с одинаковой схемой подключения.

С точки зрения мониторинга и управления, клиенты разбиваются на *группы*. Группы могут создаваться по любому принципу: территориальному, по административному подчинению или внутрикорпоративной структуре, по важности и ответственности конкретных площадок, по устойчивости работы (например, "новые" и "проблемные", требующие повышенного внимания) и т.п. Один и тот же клиент может принадлежать к нескольким группам одновременно. При использовании *фильтров* создаются *временные группы* (см. ниже), с которыми можно производить какие-то действия отдельно от остальных.

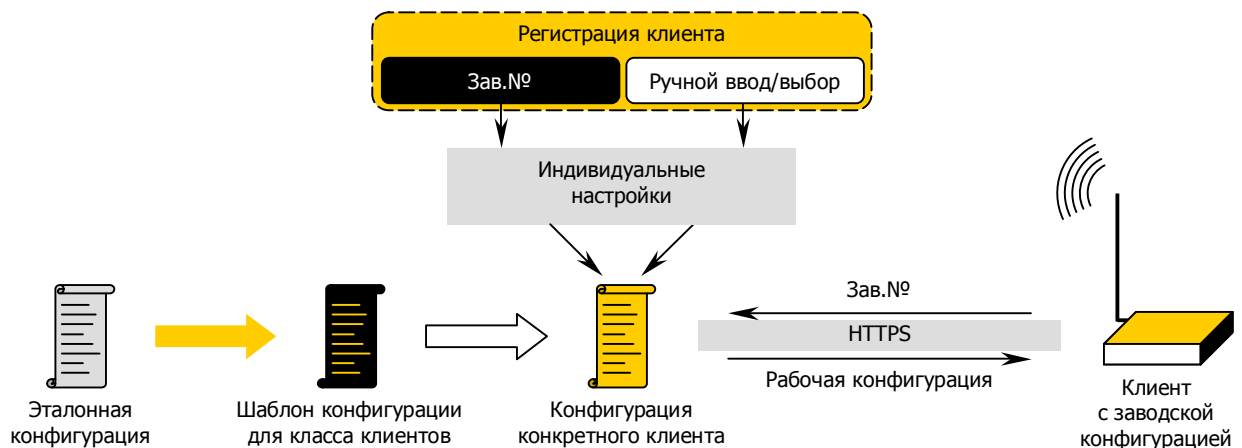
* На момент написания данного документа — в опытных инсталляциях до 2000 клиентов на 1 сервер.



Конфигурация: скрипты и шаблоны

Для управления клиентами и серверной стороной туннелей используются *скрипты* и *шаблоны конфигурации*. Шаблон — это текстовый файл конфигурации в формате конкретной модели оборудования, в котором все уникальные параметры заменены макроподстановками. Макроподстановки могут вычисляться автоматически из уникального идентификатора устройства (заводского номера и т.п.), выбираться пользователем вручную из заданного списка (например, режим LTE/UMTS/GSM для конкретного оператора на конкретной площадке) или вводиться в свободной форме (например, название площадки). Комбинация шаблонов со значениями макроподстановок создаёт файл конфигурации для конкретного клиента и его серверной стороны.

Скрипты — это исполняемые файлы, предназначенные для выполнения тех или иных действий на клиенте. Например, скрипт первоначальной настройки передаёт на клиента файл конфигурации (сгенерированный индивидуально для него), сохраняет его и вводит в действие. Другие скрипты предназначены, в общем случае, для операций мониторинга и управления клиентами.



Скрипты исполняются в контексте командной оболочки, доступной на данной модели оборудования. В частности, для устройств NSG можно использовать как фирменный язык команд NSG, так и стандартную для Linux оболочку *bash*. При редактировании скриптов и шаблонов может выполняться их синтаксическая и семантическая проверка. История изменения шаблонов сохраняется, что помогает оперативно восстанавливать их, если администратором допущена ошибка.

Редактировать клиента

Имя: NSG1810_1410001259

Описание: ATM23

Адрес: Москва, ул. Волыная, 35

Группы: NSG-1810, NSG-1810, Больные, Проблемные, Серые

Класс: Test 1810

Параметры:

m1 mode: uulo

m2 mode: uulo

Местоположение: Ш: 55.117635010/49.71 Л: 37.138031768966784

Сохранить

К списку

Мониторинг и управление в ручном режиме

Начальное конфигурирование клиентов — лишь одна из функций "Богатки". В процессе работы система непрерывно контролирует состояние всех туннелей. В пользовательский интерфейс выводятся текущее состояние туннелей и оперативные сообщения о падении и восстановлении соединений с серверами. Помимо этого, подсчитывается суммарная статистика по системе: число клиентов в разных состояниях, распределение клиентов по серверам, поставщикам услуг и т.п. Вся статистика представляется в виде таблиц, графиков и диаграмм.

Клиенты Группа "Все"

Группы: Карта Обновить

Экспорт CSV Печать Столбцы

Показать 10 записей Поиск:

ID	Имя	Описание	Адрес	Сервер	Link UP	Link DOWN	N
1	NSG1810_1410001549	ATM14		Север		m1@Юг, m1@Север	2
2	NSG1810_1410001494	ATM34		Север	m2@Север, m2@Юг	m1@Юг, m1@Север	4
4	NSG1810_1410001354	ATM43		Юг	m1@Север, m2@Север, m2@Юг, m1@Юг		4
4	NSG1810_1410001357	ATM12		Север	m1@Юг, m2@Юг, m1@Север, m2@Север		4
4	NSG1810_1410001382	ATM15		Север	m2@Юг, m1@Юг, m1@Север, m2@Север		4
4	NSG1810_1410001356	ATM46		Юг	m2@Юг, m1@Север, m1@Юг, m2@Север		4
4	NSG1810_1410001376	ATM16		Север	m1@Юг, m1@Север, m2@Север, m2@Юг		4
4	NSG1810_1410001421	ATM41		Юг	m2@Север, m2@Юг, m1@Юг, m1@Север		4
4	NSG1810_1410001397	ATM25		Юг	m1@Север, m1@Юг, m2@Север, m2@Юг		4
4	NSG1810_1410001547	ATM11		Юг	m1@Юг, m2@Юг, m1@Север, m2@Север		4

Записи с 1 до 10 из 19 записей

Предыдущая 1 2 Следующая

Со своей стороны, оператор может обратиться к любому из клиентов и выполнить на нём операции из заранее определённого набора. По существу, это могут быть любые скрипты, заданные администратором системы. В простейшем случае это операции ручного мониторинга и инвентаризации: можно запросить, например, версию ПО, состояние каждого из сотовых интерфейсов, идентификаторы сотовых модулей и SIM-карт и т.п.

Следующий уровень работы с клиентом — это скрипты, содержащие те или иные команды управления. С их помощью можно, например, рестартовать клиента целиком или заданный порт/службу на нём, загрузить новую конфигурацию, обновить программное обеспечение и т.п.

В случаях, требующих изучения ситуации на месте, оператор может подключиться через туннель непосредственно к клиентскому устройству и выполнить необходимые настройки вручную, используя CLI клиента.

The screenshot displays the NSG management interface for a client named NSG1810_1410001494. The interface is divided into several sections:

- Client Information:** Client ID, name (ATM34), class (Test-1810), and primary server (Север).
- Monitoring Section:**
 - Состояние соединений (Connection Status):** A table showing connection details for 'm1@Север' and 'm2@Юг' on both 'Север' and 'Юг' servers, including status (UP), uptime, and failure rates.
 - Выполнить скрипт на клиенте (Execute script on client):** A section for running scripts like 'uptime' on the client, with a 'Выполнить' (Execute) button.
 - Выполнить скрипт на сервере (Execute script on server):** A section for running scripts on the server, currently showing 'Скрипт не определен' (Script not defined).
 - Датчики (Sensors):** A table showing sensor data for 'csq[m1]' and 'csq[m2]' with their values and last update times.
 - Отключить терминал (Disconnect terminal):** A section for connecting to the client's CLI, showing a terminal window with system information.
- Configuration Section:** Includes options for users, servers, groups, classes, templates, and clients.
- Registration and Scheduling:** Options for client registration and task scheduling.
- Audit and Chat History:** Sections for auditing actions and viewing chat history.

Совокупность средств мониторинга и диагностики позволяет не только оперативно *устранять* возникающие проблемы — неработающих клиентов, но и *превентивно выявлять* те проблемы, которые могут возникнуть в будущем, и своевременно *предотвращать* их. Например, например, рестарты клиента без видимой причины могут свидетельствовать о проблемах с электропитанием на площадке. Или, если у клиента не работает один из каналов связи, то это можно увидеть и исправить до того, как откажет и другой канал тоже и клиент станет недоступным.

Помимо клиентских устройств, аналогичные сервисы предусмотрены и для серверной стороны системы — например, обновить конфигурацию серверной стороны туннеля на всех узлах.

Групповые операции и фильтры

Все операции, предусмотренные для одиночных клиентов, могут выполняться на большом числе клиентов одновременно. Пользователь системы может выбрать в качестве объекта некоторый класс и/или группу клиентов, или всех клиентов системы. "Богатка"

рассылает им заданные скрипты и контролирует их исполнение. Клиенты, от которых результат исполнения скрипта не получен (например, они были на недоступны в этот момент), легко идентифицируются, и при необходимости скрипт выполняется для них повторно.

The screenshot displays the NSG web interface for group operations. The main content area is titled "Групповые операции" and shows a filter: "script.m1_temp ~="" and script.m1_temp > 60". Below the filter, there are buttons for "Посмотреть" and "Применить". The selected script is "m1 Температура". A table below shows the execution results for the "m1 Температура (m1_temp)" script. The table has columns for "Имя", "Описание", "Запрос", "Ответ", "Данные", and "Статус".

Имя	Описание	Запрос	Ответ	Данные	Статус
NSG1810_1410001354	ATM43	2019-11-11 16:43:09	2019-11-11 16:43:19	64	выполнен
NSG1810_1410001364	idle	2019-11-11 16:43:09	2019-11-11 16:43:19	79	выполнен
NSG1810_1410001421	ATM41	2019-11-11 16:43:09	2019-11-11 16:43:19	62	выполнен
NSG1810_1410001451	ATM44	2019-11-11 16:43:09	2019-11-11 16:43:18	61	выполнен

On the right side, there is a "Сообщения:" panel with three entries:

- 17:35:38 NSG1810_1410001407: Соединение m2@Ю.Порт упало
- 17:34:54 NSG1810_1410001382: Порт m2 не работает
- 17:33:36 NSG1810_1410001270: Порт m2 не работает

Процедура работы с большими группами клиентов может быть сделана ещё более избирательной с помощью *фильтров*. Система может проконтролировать ответы, полученные от клиентов, и отфильтровать их по заданным правилам. Из клиентов, попавших под эти условия, можно образовать *временную группу* и далее работать с ними отдельно. Например, можно выбрать класс устройств одной и той же модели, проверить на них версию ПО, отобрать все устройства с версией ПО ниже некоторой, и выполнить на них обновление.

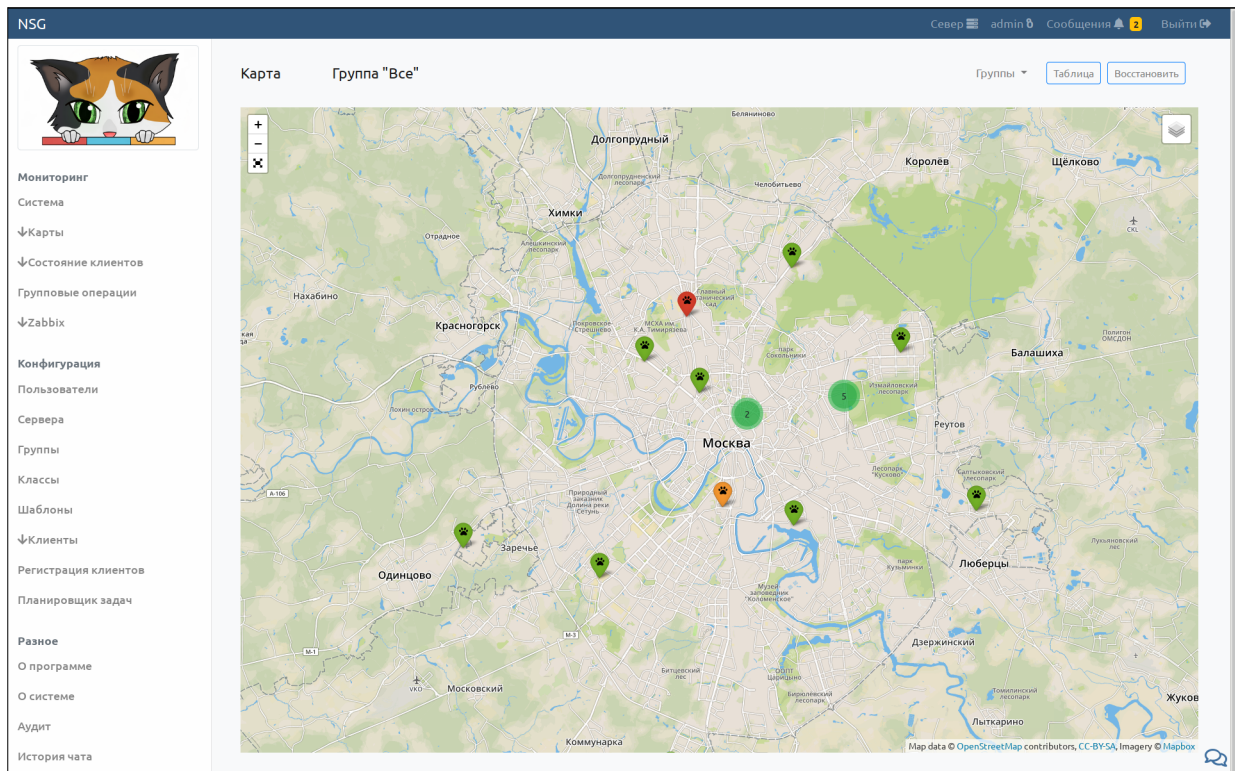
Локальный мониторинг на клиентах

Помимо ручного мониторинга со стороны оператора, на клиентских устройствах NSG могут в постоянном режиме исполняться скрипты, собирающие любую интересующую пользователя информацию. Это могут быть, например, уровни сигнала сотовых сетей, характеристики используемых каналов связи (в простейшем случае, результаты *ping*), статистика сетевых интерфейсов и многое другое — практически любая информация, доступная локально на устройстве. При наличии внешних датчиков устройство может контролировать также любые параметры физического окружения: температуру, влажность, освещённость, вскрытие помещений или аппаратных шкафов, параметры электропитания и т.п.

Эта информация может передаваться в централизованные системы мониторинга и базы данных различными способами, включая протоколы MQTT и Zabbix. Клиент также способен анализировать её самостоятельно и реагировать различными способами, например, при превышении установленных критических значений посылать уведомления по SMS и электронной почте.

Интеграция с ГИС

"Богатка" позволяет учитывать информацию о местонахождении клиентов и отражать её на картах от различных систем общего пользования, таких как Open Street Maps или Яндекс.Карты. Местонахождение клиентов может вводиться вручную в виде адреса или указываться мышью на карте — одно автоматически разрешается в другое.



Если клиентское устройство оснащено приёмником геонавигационных систем, то его положение на карте устанавливается автоматически. В этом случае можно видеть не только сам факт, что, например, банкомат "сделал ноги", но и куда именно он этими ногами пошёл.

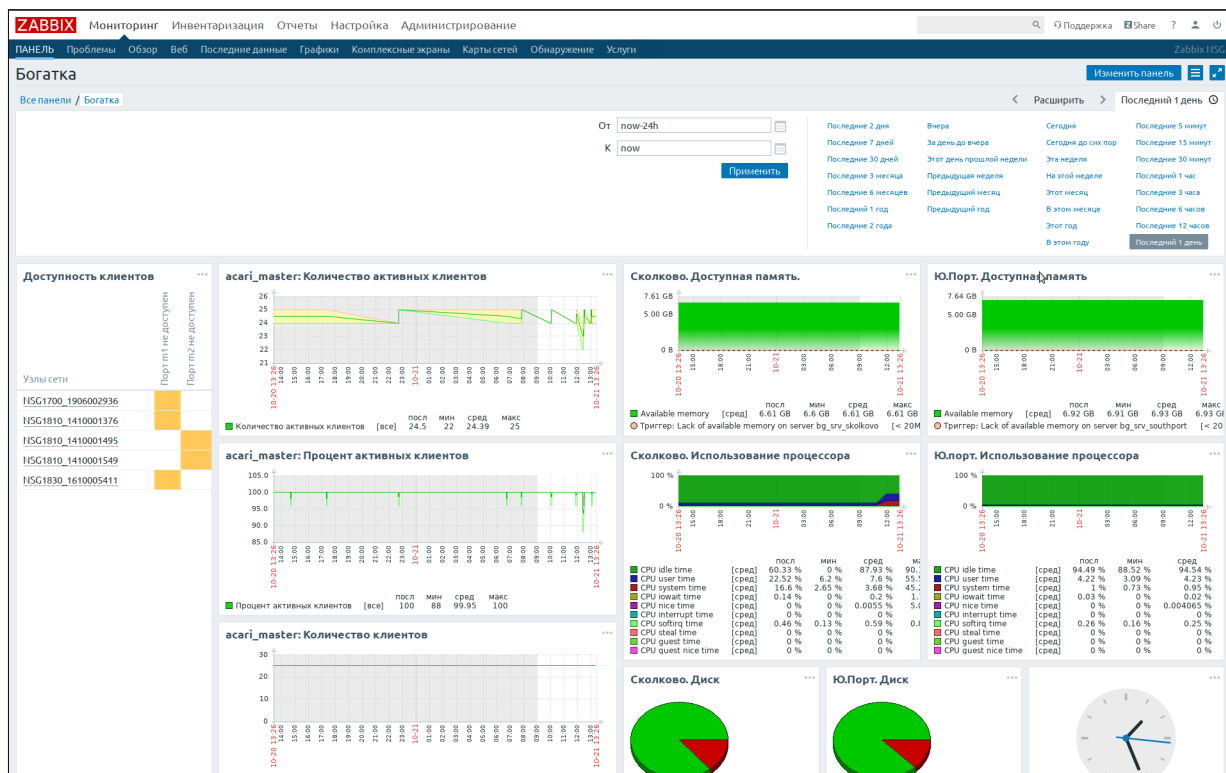
Интеграция с Zabbix

Сама по себе "Богатка" не предполагает накопления и анализа истории и долгосрочной статистики работы системы, генерации сложных отчётов и т.п. Она предназначена для оперативного контроля работы клиентов и системы в целом. Для работы с "медленной" долгосрочной статистикой существует ряд готовых решений и продуктов. Одним из наиболее популярных среди них является Zabbix.

"Богатка" предусматривает взаимодействие с Zabbix и существенно расширяет его возможности. Клиенты передают свои данные серверам "Богатка", а те ретранслируют их на сервер Zabbix. В процессе этой передачи система выполняет дополнительные действия, не предусмотренные в Zabbix, в т.ч.:

- автоматически создаёт узлы для новых клиентов на сервере Zabbix (и удаляет их при удалении из системы);
- обобщает данные клиентов независимо от того, на каком сервере каждый из них находится в данный момент;
- регулирует поток сообщений на сервер Zabbix, чтобы он успевал их обрабатывать;
- генерирует сводную статистику по системе и передаёт её в Zabbix как самостоятельные данные.

С другой стороны, дополнительный набор скриптов позволяет запросить в Zabbix требуемые данные за некоторый период времени, обработать их и представить желаемым образом. Это удобный инструмент, если такую обработку проблематично выполнить средствами Zabbix, или если требуется представить все данные непосредственно в "Богатке".



Ролевая модель доступа

Архитектура "Богатки" предусматривает построение большой системы, которая ни физически, ни административно не может контролироваться одним человеком. В ней организовано гибкое разделение прав пользователей и их областей ответственности.

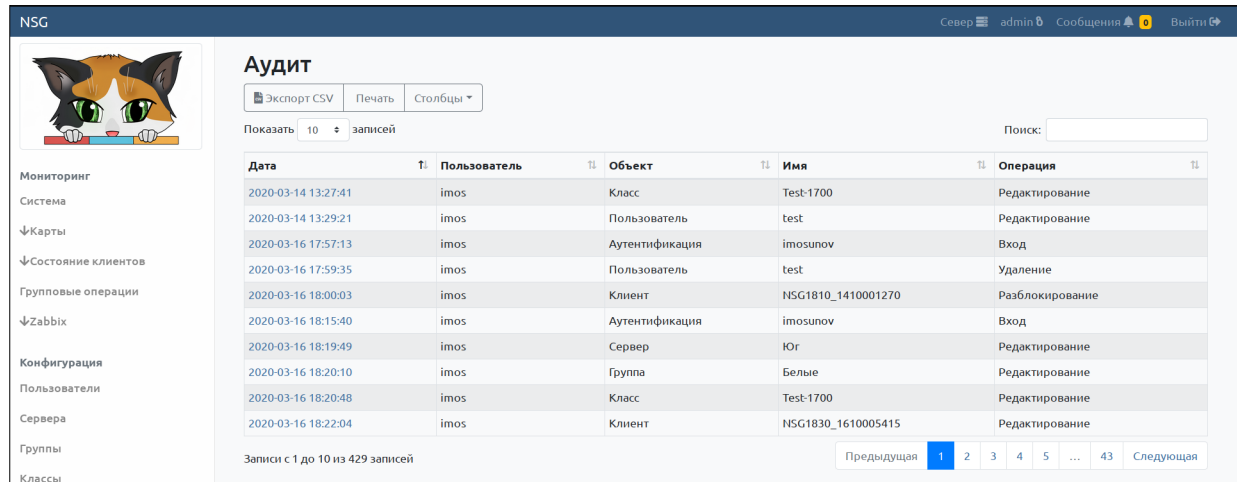
Администраторы системы имеют доступ ко всем её функциям, но основная область их ответственности — это настройки, которые влияют на работу всей системы: классы, шаблоны и скрипты, а также группы, пользователи и их права.

Каждому рядовому пользователю (оператору) системы назначаются группы, за которые он отвечает. Пользователь непосредственно управляет клиентами, входящими в его группы, и не влияет на работу других групп.

Группа	Нет	Ограниченный	Полный
NSG-1810	Нет	Ограниченный	Полный
NSG-1830	Нет	Ограниченный	Полный
Белье	Нет	Ограниченный	Полный
Серые	Нет	Ограниченный	Полный

Пользователь может иметь *полный* или *ограниченный доступ*, причём эти понятия существенно более гибкие, чем традиционное *read-write/read-only*. Набор правил для исключений позволяет сочетать жёсткое регламентирование с требованиями реальной жизни. Например, неотложные простые операции, такие как рестарт проблемного

устройства или блокировка похищенного клиента, можно доверить любому оператору, имеющего хотя бы ограниченный доступ к этому клиенту. С другой стороны, если в конфигурации клиента содержатся ключи и пароли, то можно разрешить её просмотр только операторам с полным доступом.



The screenshot shows the 'Аудит' (Audit) page in the NSG interface. It features a sidebar with navigation options like 'Мониторинг', 'Система', 'Карты', etc. The main area displays a table of audit records with columns for 'Дата', 'Пользователь', 'Объект', 'Имя', and 'Операция'. Below the table, there are controls for 'Показать' (10 records) and 'Экспорт CSV', 'Печать', 'Столбцы'. A search bar is also present.

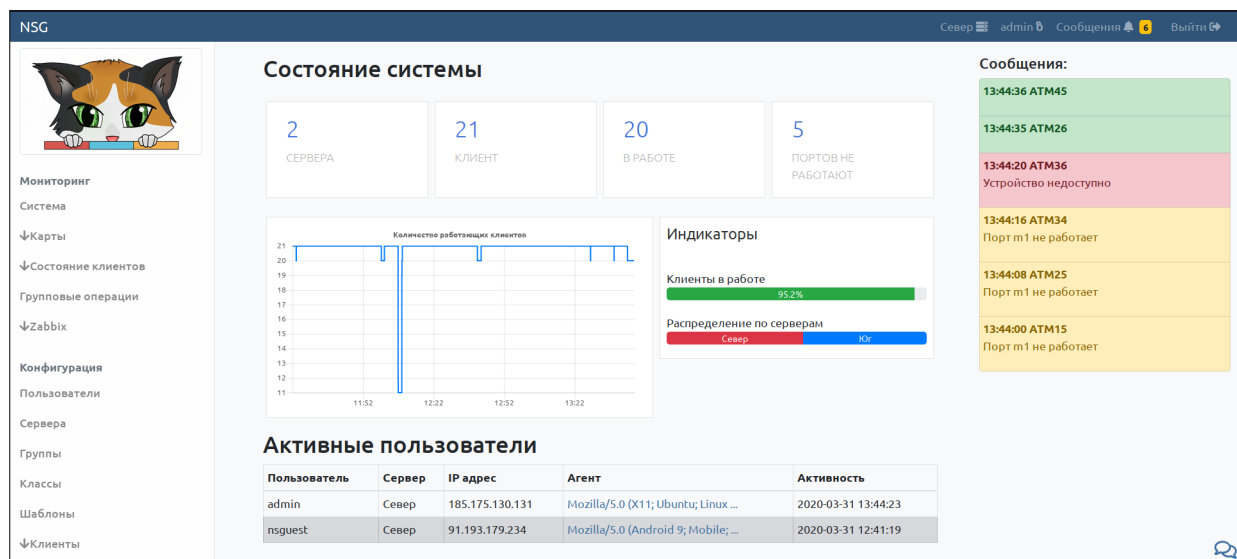
Дата	Пользователь	Объект	Имя	Операция
2020-03-14 13:27:41	imos	Класс	Test-1700	Редактирование
2020-03-14 13:29:21	imos	Пользователь	test	Редактирование
2020-03-16 17:57:13	imos	Аутентификация	imosunov	Вход
2020-03-16 17:59:35	imos	Пользователь	test	Удаление
2020-03-16 18:00:03	imos	Клиент	NSG1810_1410001270	Разблокирование
2020-03-16 18:15:40	imos	Аутентификация	imosunov	Вход
2020-03-16 18:19:49	imos	Сервер	Юг	Редактирование
2020-03-16 18:20:10	imos	Группа	Белые	Редактирование
2020-03-16 18:20:48	imos	Класс	Test-1700	Редактирование
2020-03-16 18:22:04	imos	Клиент	NSG1830_1610005415	Редактирование

Все действия пользователей регистрируются в системе *аудита* и могут быть восстановлены и проанализированы впоследствии.

Web-интерфейс

Управление всей системой "Богатка" и "Клещ" производится из одного Web-интерфейса, построенного с использованием современных технологий и принципов Web-дизайна.

Динамическая генерация страниц позволяет формировать информацию об объектах, интересующих пользователя, и в удобном ему виде. Информация представлена в виде таблиц, графиков и диаграмм. Таблицы можно разбивать на страницы с заданным числом строк, выбирать видимые колонки, сортировать по различным колонкам, искать в них заданные записи. Данные таблиц можно распечатывать или экспортировать в формате CSV.



The screenshot shows the 'Состояние системы' (System Status) page. It includes a sidebar, a top navigation bar, and several sections: 'Состояние системы' with four status cards (2 servers, 21 clients, 20 working, 5 ports not working), 'Индикаторы' with a bar chart for 'Клиенты в работе' (95.2%) and a pie chart for 'Распределение по серверам' (Север, Юг), 'Сообщения' (Messages) with a list of alerts, and 'Активные пользователи' (Active Users) with a table of current users.

Пользователь	Сервер	IP адрес	Агент	Активность
admin	Север	185.175.130.131	Mozilla/5.0 (X11; Ubuntu; Linux ...)	2020-03-31 13:44:23
nsguest	Север	91.193.179.234	Mozilla/5.0 (Android 9; Mobile; ...)	2020-03-31 12:41:19

Адаптивный дизайн страниц обеспечивает удобство работы на любых категориях устройств, от видеостен до смартфонов. Страницы автоматически масштабируются и переформируются под ширину экрана.

Сообщения о падении/поднятии соединений, а также о других значимых системных событиях, выводятся в окно браузера с помощью *push*-уведомлений. Благодаря этому, пользователь видит состояние системы в реальном времени без необходимости обновлять страницу вручную.

Для оперативного взаимодействия пользователей в системе имеется *встроенный чат*. С его помощью можно, например, предупредить коллег о своих действиях, разослать информацию для общего сведения, или спросить совета.

Искусственный интеллект

Встроенные механизмы искусственного интеллекта позволяют развёртывать и эксплуатировать систему без применения естественного. Интеллект "Богатки" активно развивается и умеет самостоятельно диагностировать основные типы неполадок, исправлять человеческие ошибки, а также руководить пользователями, способными выполнять несложные механические действия, например: "заменить SIM-карту".

Рекомендации системы по устранению неисправностей доступны как операторам, так и инженеру на площадке. При этом не обязательно, чтобы инженер был зарегистрирован в системе в качестве пользователя. Для него генерируется разовая диагностическая страница, на которую он может зайти со смартфона и видеть состояние портов, указания к действиям и не может никаким образом вмешаться в работу системы. Ссылка на эту страницу может быть передана в виде SMS или QR-кода.

Виртуализация

Серверная часть "Богатки" и "Клеца" поставляется в виде готовых контейнеров Docker и может быть легко развёрнута на любой имеющейся платформе с архитектурой x86_64, локально или в облачной инфраструктуре. Клиентская сторона представляет собой самостоятельное приложение для Linux и может быть установлена, теоретически, на любую открытую Linux-машину с достаточными аппаратными ресурсами и возможностями (CPU, RAM, ROM, аппаратное управление сотовыми интерфейсами). Коммерческие аспекты такой инсталляции выходят за рамки данного документа.

Документация и самодокументация

Интерактивный интерфейс системы интуитивно понятен и написан на русском языке. Благодаря этому он не требует многостраничных руководств пользователя. Для работы в системе достаточно знать общие принципы и сущности, в ней наличествующие, в объёме данного документа. Полные руководства администратора и оператора предоставляются отдельными документами.

Разнообразные поля для описаний и комментариев ко всем объектам делают инсталляцию самодокументируемой. Благодаря этому она не нуждается в отдельной сопроводительной документации. Любой сотрудник может легко видеть, что представляет собой тот или иной объект или почему для него сделаны некоторые индивидуальные настройки. Даже когда тот, кто делал эти настройки, уйдёт в отпуск, уволится из организации или покинет сей бренный мир — система остаётся живой и понятной для его преемников.

