МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

СОГЛАСОВАНО
Генеральный директор
ЗАО «АйТи»

_____ Бакиев О.Р.
"29" декабря 2011 г.

УТВЕРЖДАЮ
Ректор НИУ ИТМО

_____ Васильев В.Н.
"29" декабря 2011 г.

МНОГОПРОФИЛЬНАЯ ИНСТРУМЕНТАЛЬНО-ТЕХНОЛОГИЧЕСКАЯ ПЛАТФОРМА СОЗДАНИЯ И УПРАВЛЕНИЯ РАСПРЕДЕЛЕННОЙ СРЕДОЙ ОБЛАЧНЫХ ВЫЧИСЛЕНИЙ CLAVIRE

ПРОГРАММНЫЙ КОМПОНЕНТ КОНТРОЛЯ ДОСТУПА CLAVIRE/GATEKEEPER

ОПИСАНИЕ ПРОГРАММЫ

ЛИСТ УТВЕРЖДЕНИЯ

RU.СНАБ.80066-06 13 26-ЛУ

Представители
Организации-разработчика

Руководитель разработки,
профессор НИУ ИТМО

_____ Бухановский А.В.
"29" декабря 2011 г.

Ответственный исполнитель,
с.н.с. НИУ ИТМО

_____ Луценко А.Е.
"29" декабря 2011 г.

Нормоконтролер
ведущий инженер НИУ ИТМО

_____ Позднякова Л.Г.
"29" декабря 2011 г.

2011

Инв.№ подл.	Подп. и дата	Взам.инв.№	Инв.№ дубл.	Подп. и дата

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

УТВЕРЖДЕН
RU.СНАБ.80066-06 13 Ошибка! Источник ссылки не найден.6-ЛУ

МНОГОПРОФИЛЬНАЯ ИНСТРУМЕНТАЛЬНО-ТЕХНОЛОГИЧЕСКАЯ ПЛАТФОРМА СОЗДАНИЯ И УПРАВЛЕНИЯ РАСПРЕДЕЛЕННОЙ СРЕДОЙ ОБЛАЧНЫХ ВЫЧИСЛЕНИЙ CLAVIRE

ПРОГРАММНЫЙ КОМПОНЕНТ КОНТРОЛЯ ДОСТУПА CLAVIRE/GATEKEEPER

ОПИСАНИЕ ПРОГРАММЫ

RU.СНАБ.80066-06 13 ОШИБКА! ИСТОЧНИК ССЫЛКИ НЕ НАЙДЕН.**6**

Подп. и дата	Инв.№ дубл.	Взам.инв.№	Подп. и дата	Инв.№ подл.

ЛИСТОВ 18

2011

АННОТАЦИЯ

Документ содержит описание программного компонента контроля доступа CLAVIRE/GateKeeper RU.СНАБ.80066-06 01 26. Компонент предназначен для задания и обеспечения корректного выполнения политик общесистемной безопасности платформы CLAVIRE (при работе МИТП в многопользовательском режиме), направленной на разграничение прав доступа пользователей, и защиты компонентов и ресурсов системы от несанкционированного использования. Основными функциональными задачами данного компонента являются аутентификация и авторизация пользователей на доступ к конкретным ресурсам и сервисам комплекса.

Программный компонент контроля доступа разработан в ходе выполнения проекта «Создание распределенной вычислительной среды на базе облачной архитектуры для построения и эксплуатации высокопроизводительных композитных приложений» (Договор № 21057 от 15 июля 2010 г., шифр 2010-218-01-209) в рамках реализации постановления Правительства РФ № 218 «О мерах государственной поддержки развития кооперации российских высших учебных заведений и организаций, реализующих комплексные проекты по созданию высокотехнологичного производства».

СОДЕРЖАНИЕ

1. ОБЩИЕ СВЕДЕНИЯ

Программный компонент контроля доступа CLAVIRE/GateKeeper RU.СНАБ.80066-06 01 26 предназначен для задания и обеспечения корректного выполнения политик общесистемной безопасности, направленной на разграничение прав доступа пользователей, и защиты компонентов и ресурсов системы от несанкционированного использования при работе МИТП в многопользовательском режиме.

Политики безопасности, задаваемые в системе, различаются своими объектами применения: сервисы и ресурсы. Права на сервисы МИТП имеют организационный смысл: пользователи разделены на группы, имеющие разный уровень доступа к компонентам, – обычные пользователи, администраторы ресурсов, администраторы пакетов, администраторы безопасности, системные операторы и др. Права на ресурсы определяют исключительно возможность использования пользователями определенных вычислительных ресурсов.

Компонент состоит из трех модулей: аутентификации, авторизации и управления правами. Все они представляют собой сервисы Windows Communication Foundation, реализованные на языке программирования C# версии 4.0.

Компонент функционирует на аппаратных системах с архитектурой процессора x86, x86_64 и IA64. Для его работы необходимо следующее системное программное обеспечение: ОС семейства Windows NT (версии старше Windows 2000), .NET 4.0, Microsoft Internet Information Services (с версией старше 6.0). Также необходимо наличие установленной и корректно настроенной базы данных MongoDB для хранения учетных данных пользователей.

2. ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ

Компонент решает следующие задачи.

1) Аутентификация пользователя в системе.
2) Задание прав пользователей на сервисы и ресурсы системы.
3) Проверка прав пользователя на использование сервисов и ресурсов системы.

Для решения поставленных задач компонент контроля доступа взаимодействует со следующими компонентами МИТП: обеспечения доступа к инфраструктуре, исполнения WF, взаимодействия с пользователем и информационного портала.

Таким образом, компонент обеспечивает следующие функциональные возможности.

1) Предоставление пользователю доступа к МИТП по учетным данным (вместе с компонентами пользовательского интерфейса и обеспечения доступа к инфраструктуре).
2) Ведение баз данных прав пользователей на ресурсы и сервисы.
3) Изменение прав пользователей на ресурсы и сервисы администраторами прав.
4) Аудит изменений политик распределения прав.
5) Ведение журнала действий пользователя внутри комплекса.

3. ОПИСАНИЕ ЛОГИЧЕСКОЙ СТРУКТУРЫ

3.1. Архитектура компонента

Модуль аутентификации взаимодействует с компонентами пользовательского интерфейса и компонентом администрирования для выполнения первичной аутентификации пользователей в системе.

Модуль авторизации вместе с компонентом обеспечения доступа к инфраструктуре отвечает за допуск пользователя к управляющим частям компонентов МИТП – системным сервисам. За авторизацию на вычислительные ресурсы комплекса отвечает модуль управления правами, который тесно работает с компонентом исполнения WF.

Все три модуля имеют двухуровневую структуру: уровень логики безопасности и уровень хранения данных. Хранение данных централизованно для всех модулей и реализовано как коллекции базы данных MongoDB.

Как видно из рис. 3.1, уровень хранения данных компонента состоит из хранилища данных пользователей (реализовано как коллекция базы данных MongoDB) и журнала аудита (специальный log-файл сервиса). Кроме собственно учетных данных (имени и пароля) хранилище содержит описание ролей и прав пользователя, а также информацию о пользователе для компонента учета использования ресурсов (баланс, квоты) и дополнительную информацию – дата последнего посещения, дата регистрации и т.п.

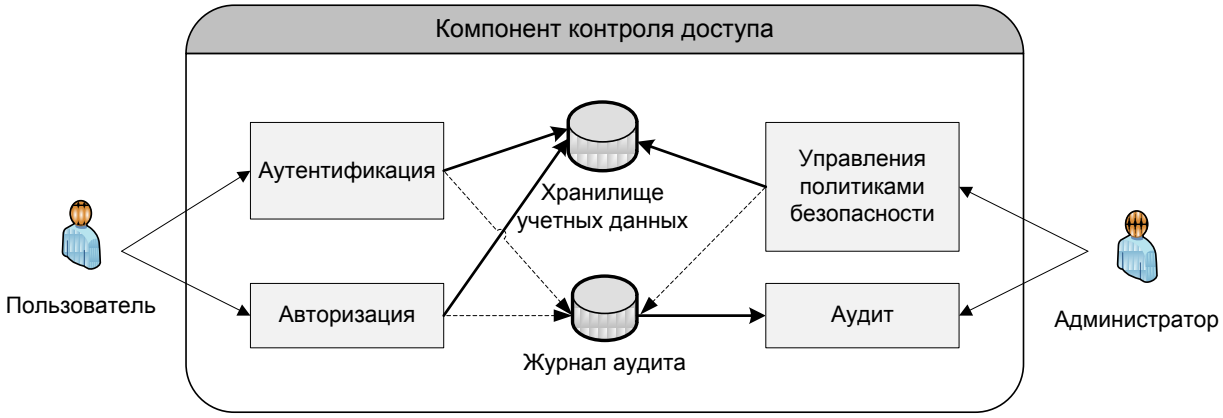

Рисунок 3.1 – Архитектура модуля обеспечения безопасности

3.2. Модуль аутентификации

При обращении пользователя к комплексу через любой компонент взаимодействия (CLAVIRE/Ginger, CLAVIRE/AdminTool) формируется запрос аутентификации, состоящий из имени пользователя и его пароля, и направляется к модулю аутентификации компонента GateKeeper. Модуль аутентификации проверяет правильность предоставленных данных, в случае их некорректности пользователю отказывается в дальнейшем обслуживании. В случае получения корректной пары имя/пароль соответствующий компонент создает пользовательскую сессию (что предотвращает повторное отправление запросов аутентификации) и ведет всю дальнейшую работу от имени данного пользователя.

Безопасность паролей пользователей обеспечивается хранением не самих паролей, а значений хеш-функций от них. Если точнее, каждая запись пароля содержит идентификатор используемой хеш-функции и «соль» – случайный набор символов. «Соль» дописывается к паролю и именно от этой строки берется хеш-функция, это позволяет разрушить статистические закономерности в множестве паролей пользователей и снизить уязвимость к переборным атакам на базу паролей.

Таким образом, процедура проверки корректности учетных данных делится на две части: поиск пользователя и проверка пароля. Если поиск пользователя выполняется запросом к базе данных, то пароль проверяется сравнением двух хешей – хранящегося в базе и полученного из запроса аутентификации с добавленной «солью».

Также модуль аутентификации имеет некоторые вспомогательные функции – проверка существования пользователя в комплексе, изменение пароля и др.

3.3. Классы модуля аутентификации

3.3.1. Перечисление HashType

Перечисление HashType определено для того, чтобы различать алгоритмы хеширования, которые могут применяться при хранении учетных данных пользователей МИТП. Перечисление содержит следующие значений:

- MD5 –хеш вычислялся 128-битовым алгоритмом MD5.
- SHA1 –128-битовый хеш-алгоритм SHA-1.
- SHA256 – 256-битовый хеш-алгоритм SHA-256.
- SHA384 –384-битовый хеш-алгоритм SHA-384.
- SHA512 – 512-битовый хеш-алгоритм SHA-512.

По умолчанию для хеширования ключа используется алгоритм SHA512, являющийся самым устойчивым из этого списка. Не рекомендуется использовать 128-битные алгоритмы в связи с их недостаточной безопасностью. Алгоритм SHA-384 является сокращенной 512-битовой версией алгоритма, а поэтому имеет худшие показатели по соотношению производительность/безопасность.

3.3.2. Класс SafePassword

Класс SafePassword предназначен для хранения в базе MongoDB хеша-значения от пароля пользователя, а также для удобного получаемого открытого пароля и хеш-значения, хранимого в базе.

Свойства класса:

- HashType (тип HashType) – определяет, каким хеш-алгоритмом был обработан пароль пользователя.
- Salt (string) – «соль» для пароля пользователя, т.е. случайная строка, необходимая для нарушения статистических закономерностей в значениях хеш функции. «Соль» добавляется к паролю, и только от получившейся строки вычисляется хеш.
- HashedPassword (string) – значение хеш-функции от пароля пользователя и «соли», по этому полю и проверяется правильность предоставляемых паролей.

Методы класса

- GenerateSalt – создает случайную строку заданной длинны. Входной параметр: ByteNamber – длина строки. Выходной параметр: случайная строка длиной ByteNamber символов.

- Compare – сравнивает предложенный пароль с хранимой версией. Входные параметры: строка Password – пароль для проверки. Выходные параметры: bool – определяет совпадение паролей.
- HashPassword – статический, получает из пароля, «соли» и идентификатора алгоритма хеширования результат применения этого хеш-алгоритма к паролю и «соли». Входные параметры: пароль, «соль», HashType – алгоритм хеширования. Выходные параметры: строка – результат хеширования.

3.3.3. Класс UserData

UserData – класс для предоставления информации о пользователе. Этот класс достаточно большой и используется в четырех модулях двух компонентов комплекса, так что здесь приведем только необходимые для аутентификации части этого класса.

Поля

- UserId (Guid) – уникальный идентификатор пользователя.
- Name (string) – имя пользователя.
- Email (string) – адрес электронной почты пользователя для альтернативного входа.
- Password (SafePassword) – хеш-значение пароля.

Методы

- GetCorrectId – приводит идентификатор к виду строки со строчными символами, для устроения неоднозначности представления идентификатора. Входные параметры: строка – представление идентификатора. Выходные параметры: корректное представление идентификатора.
- GetCorrectId – приводит имя к виду строки со строчными символами для устроения неоднозначности представления. Входные параметры: строка –имя. Выходные параметры: имя со всеми строчными символами.
- GetUserEntryFromUserData – статический, получает сокращенную информацию о пользователе для графического интерфейса. Входные параметры: UserData – вся пользовательская информация. Выходные параметры: UserEntry – сокращенная информация (имя и идентификатор пользователя).

3.3.4. Интерфейс IAuthService

Описывает контракт сервиса аутентификации.

Методы

- GetUserDataById – метод получения данных пользователя по его идентификатору. Входные параметры: Id – идентификатор пользователя. Выходные параметры: полные данные пользователя.
- GetUserDataByName – метод получения данных пользователя по его имени. Входные параметры: Name – идентификатор пользователя. Выходные параметры: полные данные пользователя.
- GetUserList – возвращает список идентификаторов всех пользователей МИТП. Входные параметры: нет. Выходные параметры: список идентификаторов пользователей платформы.
- Login – метод, проверяющий пользователя и сохраняющий новое значение в поле LastVisit. Входные параметры: имя пользователя и пароль. Выходные параметры: UserEntry – сокращенные данные пользователя, если он существует и привел правильный пароль. В случае ошибки – генерирует исключение и передает вызывающей стороне отказ в доступе.
- Validate – метод, проверяющий пользователя и правильность пароля, не изменяет поле LastVisit. Входные параметры: имя пользователя и пароль. Выходные параметры: bool – определяет правильность приведенных учетных данных.

3.3.5. Класс AuthService

Реализация интерфейса IAuthService, он имеет несколько дополнительных внутренних методов.

- CheckUniquenessAndGet – проверяет существование и уникальность записи в таблице базы данных. Входные параметры: список полученных записей. Выходные параметры: единственная запись, приведенная к заданному типу, или исключение при дублировании или отсутствии записи.
- GetUserCollection – получает таблицу пользователей МИТП из базы данных. Входные параметры: нет. Выходные параметры: MongoCollection – таблица базы данных пользователей.

3.4. Модуль авторизации

Двухуровневая структура компонента контроля, которая была представлена на рис. 3.1., также относится и к модулю авторизации. Уровень логики безопасности модуля

обеспечивает авторизацию пользователей, управление политиками безопасности и аудит использования системных сервисов.

При каждом обращении пользователя к компонентам комплекса (системным сервисам) происходит формирование запроса к модулю авторизации, который проверяет, разрешается ли данному пользователю выполнение этого сервиса. Авторизация основана на ролевой модели доступа – каждый пользователь комплекса имеет одну или несколько заранее предписанных ему ролей, а каждый сервис имеет список ролей, имеющих к нему доступ. Также от роли пользователя зависит его пользовательский интерфейс, если пользователь входит в какую-либо группу администраторов, он может использовать в работе инструменты, соответствующие его специализации.

Вся информация по ролям для пользователей хранится в хранилище учетных данных, и изменяться она может только через интерфейс модуля авторизации. Сервисы хранят роли доступа с помощью стандартных средств WCF. Хранилище учетных данных основано на документно-ориентированной системе управления базами данных MongoDB и, кроме того, содержит информацию для модуля управления правами и компонента учета использования ресурсов.

3.5. Классы модуля авторизации

3.5.1. Интерфейс ISecurityService

Интерфейс ISecurityService – это формальный контракт модуля авторизации, описывающий основные методы поведения сервиса. Также на сервис авторизации возложена административная функциональность по добавлению и удалению пользователей, эти методы доступны только пользователем с привилегированными правами.

Административные методы

- AddUser – метод добавления пользователя МИТП. Входные параметры: имя пользователя и пароль. Выходные параметры: строка – идентификатор нового пользователя, в случае невозможности выполнения операции генерируется исключение.
- ChangePassword – изменяет пароль пользователя. Входные параметры: id – идентификатор пользователя, newPass – новый пароль. Выходные параметры: bool – удостоверяет смену пароля.

- RemoveUser – удаляет пользователя МИТП, эта функция доступна только администратору МИТП. Входные параметры: строка id – идентификатор пользователя. Выходные параметры: bool – подтверждение удаления пользователя.
- IsUserExists – метод, проверяющий существование пользователя платформы МИТП по его имени. Входные параметры: строка – имя пользователя. Выходные параметры: bool – подтверждение/опровержение существования пользователя МИТП
- GetUserName – возвращает имя пользователя по его идентификатору. Входные параметры: строка id – идентификатор пользователя. Выходные параметры: строка – имя пользователя.
- GetUserId – возвращает идентификатор пользователя по его имени. Входные параметры: строка Name – имя пользователя. Выходные параметры: строка – идентификатор.

Методы авторизации

- AddRoles – метод, добавляющий новые роли пользователю комплекса. Входные параметры: строка id – уникальный идентификатор пользователя, список строк Roles – список добавляемых ролей. Выходные параметры: bool – подтверждение добавления ролей пользователю.
- RemoveRoles – метод, удаляющий роли пользователя платформы. Входные параметры: строка id – уникальный идентификатор пользователя, список строк Roles – список удаляемых ролей. Выходные параметры: bool – подтверждение удаления ролей.
- GetUserRoles – метод, предоставляющий список ролей заданного пользователя. Входные параметры: срока Id – идентификатор пользователя. Выходные параметры: список ролей пользователя.

3.5.2. Класс SecurityService

Класс, реализующий контракт ISecurityService, является реализацией всей функциональности модуля авторизации. Он содержит все методы, продекларированные в интерфейсе и такие же вспомогательные методы, как модуль аутентификации. К ним добавляется очень полезный вспомогательный метод FindUserOrCreate, который ищет данные пользователя по имени в базе данных пользователей и создает его, не найдя. Входные параметры: name – имя пользователя. Выходные параметры: класс UserData – полная информация о пользователе.

3.6. Модуль управления правами

Основной задачей модуля управления правами является авторизация пользователей на вычислительные ресурсы МИТП. В отличие от модуля авторизации, этот модуль использует простое дискреционное управление доступом, где явным или неявным образом для каждой пары пользователь/ресурс задается всего один показатель: возможность использования данного ресурса данным пользователем.

Ограничение доступа к ресурсам осуществляется в точке запуска вычислений, т.е. в компоненте исполнения WF. После того как задача попадает в компонент CLAVIRE/Executor и перед планированием выполнения, вызывается метод модуля управления правами для определения ресурсов, на которых данный пользователь может выполнять вычисления. Дальнейшее планирование происходит уже только для разрешенных ресурсов. Таким образом, пользователь не может запустить задачу на ресурсе, на который он не имеет права.

В хранилище учетных данных для каждого пользователя хранится набор ресурсов с указанием возможности его использования. По умолчанию пользователь не может получить доступ ни к одному ресурсу, задавать права на использование ресурса может только администратор прав.

3.7. Классы модуля управления правами

3.7.1. Интерфейс IRightsProvider.

Интерфейс RightsProvider описывает методы только получения данных о правах на ресурсы для пользователей платформы, но не методы задания и управления правами.

Методы

- CanAccessResource – метод, определяющий возможность пользователю производить вычисления на заданном ресурсе. Входные параметры: name – строка, имя пользователя, resource – строка, идентификатор ресурса. Выходные параметры: bool – подтверждение/отказ в использовании ресурса.
- GetUserRights – предоставляет список всех ресурсов, на которых выбранный пользователь может производить свои вычисления. Входные параметры: name – строка, имя пользователя. Выходные параметры: список строк – список идентификаторов ресурсов, доступных пользователю.
- GetUserRightsList – предоставляет список ресурсов, доступных выбранному пользователю из заданного списка ресурсов. Входные параметры: name – строка,

имя пользователя, resources – список строк, список предлагаемых ресурсов. Выходные параметры: список строк – список идентификаторов ресурсов, доступных пользователю.

3.7.2. Интерфейс IRightsManager

Интерфейс IRightsManager описывает множество средств управления правами пользователей платформы.

- AddRight – метод добавления нового права пользователю, т. е. добавление ресурса, на котором он может проводить свои вычисления. Входные параметры: name – строка, имя пользователя, resource – строка, идентификатор ресурса. Выходные параметры: нет.
- AddRights – метод добавления нескольких новых ресурсов, на которых он может проводить свои вычисления. Входные параметры: id – строка, идентификатор пользователя, resources – список строк, список идентификаторов ресурсов. Выходные параметры: bool – подтверждение выполнения операции. *Внимание! У этих методов похожие сигнатуры, но тем не менее различные параметры – у метода множественного добавления используется идентификатор пользователя, в то время как у первого метода – имя.*
- RemoveRight – метод удаления права пользователя. Входные параметры: name – строка, имя пользователя, resource – строка, идентификатор ресурса. Выходные параметры: нет.
- AddRights – метод удаления нескольких ресурсов. Входные параметры: id – строка, идентификатор пользователя, resources – список строк, список идентификаторов ресурсов. Выходные параметры: bool – подтверждение выполнения операции. *Внимание! У этих методов похожие сигнатуры, но различные параметры – у метода множественного удаления используется идентификатор пользователя, в то время как у первого метода – имя.*

3.7.3. Класс RightsProviderService

Несмотря на свое название этот класс реализует оба интерфейса и IRightsProvider и IRightsManager, а соответственно и все их методы. Кроме того, он имеет несколько вспомогательных классов для работы с базой данных и конвертирования информации.

- CheckUniquenessAndGet –метод для проверки существования и уникальности записей в таблице базы данных. Входные параметры: список полученных записей.

Выходные параметры: единственная запись, приведенная к заданному типу, или исключение при дублировании или отсутствии записи.

- FindUserByName – ищет данные пользователя по имени в базе данных пользователей. Входные параметры: name – имя пользователя. Выходные параметры: класс UserData – полная информация по пользователю.
- OpenOrCreateCollection – открывает базу данных MongoDB или создает, если ее еще не существует. Входные параметры: строка – имя коллекции. Выходные параметры: класс MongoCollection – искомая коллекция.

4. ИСПОЛЬЗУЕМЫЕ ТЕХНИЧЕСКИЕ СРЕДСТВА

Компонент контроля доступа предназначен для функционирования на аппаратных системах с характеристиками:

- архитектура процессора – x86, x86_64, IA64;
- минимальный объем оперативной памяти – 1 ГБ;
- минимальный объем свободного пространства на жестком диске – 1 ГБ;
- минимальная тактовая частота процессора – 1 ГГц.

Также компонент требует для своей работы наличия следующего системного ПО: ОС семейства Windows NT (версии старше Windows 2000), .NET 4.0, IIS (с версией старше 6.0), сервер баз данных MongoDB.

5. ВЫЗОВ И ЗАГРУЗКА

Все модули реализованы как WCF-сервисы, для их запуска был выбран вариант размещения в структуре сервера IIS, таким образом, для установки компонента требуется web-сервер, поддерживающим технологию ASP .NET Web-Services (IIS 6.0, XSP) и какая-либо реализация платформы .NET 4.0.

Компилировать и размещать сервисы данного компонента можно независимо друг от друга. Компиляция компонента возможна при помощи компилятора csc.exe платформы .NET 4.0 или dmcs – платформы Mono. Для запуска сервиса используется стандартный механизм используемого web-сервера. Загрузка сервиса в этом случае выполняется web-сервером автоматически по мере поступления запросов от клиентов сервиса.

После запуска компонента доступ ко всем модулям можно получать по адресам, заданным при развертывании, например, адрес может выглядеть так: http://SampleServer.ru:8008/SampleService.

Вызов сервиса производится стандартным для технологии WCF способом: по опубликованному описанию сервиса (WSDL) необходимо создать прокси-класс, через который осуществляется взаимодействие с сервисом путем вызова необходимых операций (методов). Процедура создания прокси-класса зависит от того, на базе каких технологий строится клиентское приложение. Если выбран язык программирования C# и платформа .NET, построение клиента производится за счет вызова служебного программного средства svcutil.exe, распространяемого в составе платформы .NET, либо за счет создания ссылки на сервис в среде Microsoft VisualStudio.

Компонент не имеет сильных зависимостей от других компонентов комплекса, критичным при установке является только наличие компонента обеспечения доступа к инфраструктуре.

6. ВХОДНЫЕ ДАННЫЕ

Сервисы компонента являются классическими web-сервисами, реализованными на основе каркаса, предоставленного Windows Communication Foundation. Разработка такого рода компонентов состоит из двух частей: написание ядра сервиса, удовлетворяющего контракту и обеспечивающего выполнение необходимой логики, и расширение поведения сервиса для предоставления дополнительной функциональности. Особенностью такой реализации является скрытие внутренних потоков данных как от пользователя, так и от разработчиков. Таким образом, программный компонент контроля доступа не требует специальных входных данных кроме стандартных конфигурационных файлов, с помощью которых можно менять некоторые настройки компонента.

7. ВЫХОДНЫЕ ДАННЫЕ

Для компонента GateKeeper можно выделить два вида выходных данных. Данными первого типа можно считать ответы на запросы пользователей в форме SOAP-пакетов и скрытую в них информацию, требуемую пользователем. Вторым видом можно условно считать данные, записываемые в БД.

В листинге 7.1 приведен формат данных, хранимых в базе компонента GateKeeper.

Листинг 7.1. Пример формата данных ролей и прав пользователя на вычислительные ресурсы

```
{
  "_id": "2dfe6bc1-e5b0-44e1-9eb5-59edc6b42851",
  "Roles": [ "User" ],
  "Name": "testuser",
  "Password": {
    "Salt": "KpHE4oo/j3c=",
    "HashedPassword":
"3eZvIV1fDdxwrt9WgOORO5OD6Sq3frgM0DbUC7uy7BH3+yvueuJBl9P0VIA+axw3nvAv2Dm3iPG+fAedCFcKC
Q==",
    "HashType": "SHA512"
  },
  "Rights": {
    "cluster_niinkt_1": "true",
    "b4": "true",
    "hp-cl (grid nnn)": "true",
    "virtualpc": "true",
    "tb06 (grid nnn)": " false "
  },
  "RegistrationDate": "Thu, 22 Dec 2011 16:46:02 GMT +03:00",
  "LastVisit": "Thu, 29 Dec 2011 18:31:51 GMT +03:00",
}
```

Все данные в MongoDB хранятся в формате JSON, уникальными здесь являются только поля записи. Значения полей для этого формата приведены в табл. 7.1.

Таблица 7.1

Поля формата данных сервиса регистрации

Название поля	Описание поля
_id	Внутренний идентификатор объекта в MongoDB
Name	Имя пользователя
Rights	Список ресурсов и прав пользователя на них
Roles	Список ролей пользователя
Password	Описание пароля пользователя с параметрами хеширования
RegistrationDate	Дата регистрации пользователя
LastVisit	Дата и время последнего входа пользователя в систему

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

JSON	JavaScript Object Notation (текстовый формат обмена данными, основанный на JavaScript)
MEF	Managed Extensibility Framework
SOAP	Simple Object Access Protocol (простой протокол доступа к объектам)
WCF	Windows Communication Foundation (программный фреймворк)
WF	Workflow (поток заданий)
БД	База данных
МИТП	Многопрофильная инструментально-технологическая платформа
ОС	Операционная система
ПО	Программное обеспечение

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

Лист регистрации изменений									
Изм.	**Номера листов (страниц)**				**Всего листов (страниц) в докум.**	**№ документа**	**Входящий № сопроводительного докум. и дата**	**Подп.**	**Дата**
	измененных	**замененных**	**новых**	**аннулированных**					