

**МИНИСТЕРСТВО ПО РАЗВИТИЮ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И КОММУНИКАЦИЙ РЕСПУБЛИКИ УЗБЕКИСТАН  
КАРШИНСКИЙ ФИЛИАЛ ТАШКЕНТСКОГО УНИВЕРСИТЕТА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ**

**«Утверждаю»**

*Зав.кафедры "ИТ"*

А.Э.Чупонов\_\_\_\_\_

«\_\_\_» \_\_\_\_\_ 2016 год

**ВЫПУСКНАЯ РАБОТА**

на тему:

***«Разработка и внедрение облачных вычислений  
на основе SaaS»***

Выпускник: \_\_\_\_\_ О.О. Очилов  
подпись

Руководитель: \_\_\_\_\_ А. Тургунов  
подпись

Рецензент: \_\_\_\_\_  
подпись

Консультант по БЖД: \_\_\_\_\_ О.Д.Рахимов  
подпись

**Карши 2016**

**КАРШИНСКИЙ ФИЛИАЛ ТАШКЕНТСКОГО УНИВЕРСИТЕТА  
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ**

**факультет**

**КОМПЬЮТЕРНЫЙ ИНЖИНИРИНГ**

*Направление: Информатика и информационные технологии*

**«Утверждаю»**

*Зав.кафедры "ИТ"*

А.Э.Чупонов \_\_\_\_\_

« \_\_\_\_ » \_\_\_\_\_ 2016 год

**ЗАДАНИЕ**

***По выпускной работе***

**Студента:**

**Отабек Очилов**

**1. Тема ВР:** *«Разработка и внедрение облачных вычислений на основе SaaS»*

**2. Утверждено указом № \_\_\_\_\_** Каршинского филиала ТУИТ \_\_\_\_\_

**3. Срок сдачи ВР:** *2.06. 2016 год*

**4. Начальная информация для квалификационной работы:** Изучение основ облачного вычисления. Обзор и исследование программных средств ориентированные на разработку веб технологий и его видов, методы и технологии создания веб сайтов, языки программирования ориентированные на создании SaaS приложения, требования и этапы разработки облочных решений.

**5. Содержание расчётно-пояснительной записки** (перечень подлежащих к разработке вопросов) Научно – методические основы разработки веб систем, облочные решения, программная и техническая поддержка веб систем, безопасность жизнедеятельности.

**6. Графические материалы, слайды:** всего \_\_\_\_\_ рисунка

**7. Дата задания выпускной работы: 18.01.2016**

Руководитель по ВР: \_\_\_\_\_ А. Тургунов

Студент: \_\_\_\_\_ О.Очилов

**8. График консультаций по каждому разделу:**

Раздел	Руководитель	Дата Подпись	
		Дал задание	Принял задание
Введение	А. Тургунов	20.03.16	20.04.16
Глава 1. Основы Вычислительного Об-лака	А. Тургунов	18.01.16	20.03.16
Глава 2. Постановка Задачи	А. Тургунов	23.03.16	01.04.16
Глава 3. Программное Решения На Ос-нове Saas	А. Тургунов	01.04.16	05.05.16
Глава 4. Безопасность Жизнедеятель-ности	О.Д.Рахимов	28.05.16	01.06.16
Заключения	А. Тургунов	23.05.16	31.05.16
Литература	А. Тургунов	01.04.16	01.06.16

**9. График выполнения работы:**

№	Раздела ВР	Срок выполнения	Подпись руководителя
1.	Введение	20.03-20.04.2016	
2.	Глава 1. Основы Вычислительного Облака	18.01-20.03.2016	
3.	Глава 2. Постановка Задачи	23.03-01.04.2016	
4.	Глава 3. Программное Решения На Основе Saas	01.04-05.05.2016	
5.	Глава 4. Безопасность Жизнедеятельности	28.05-01.06.2016	
6.	Заключения	23.05-31.05.2016	
7.	Литература	01.04-01.06.2016	

Студент            О.Очилов            « \_\_\_\_\_ » \_\_\_\_\_ 2016г.  
(подпись)

Руководитель    А. Тургунов            « \_\_\_\_\_ » \_\_\_\_\_ 2016г.  
(подпись)

ВВЕДЕНИЕ .....	7
Глава. 1 ОСНОВЫ ВЫЧИСЛИТЕЛЬНОГО ОБЛАКА.....	
1.1 Модели развертывания.....	
1.2 Модели обслуживания.....	
1.2.1 IaaS (Infrastructure-as-a-service).....	
1.2.2 PaaS (Platform-as-a-service).....	
1.2.3 SaaS (Software-as-a-service).....	
1.3 Платформы для облачных вычислений.....	
1.3.1 Microsoft Azure.....	
1.3.2 Google Apps.....	
Глава. 2 ПОСТАНОВКА ЗАДАЧИ.....	
Глава. 3 ПРОГРАММНОЕ РЕШЕНИЕ НА ОСНОВЕ SAAS.....	
3.1 Теоритическая часть.....	
3.2 Используемые технологии.....	
3.3 Установка и настройка Wowza Streaming Engine.....	
3.4 Разработк ПО.....	
Глава. 4 БЕЗОПАСНОСТЬ ЖИЗНЕДЕЯТЕЛЬНОСТИ.....	
4.1 Освещения компьютерных помещений.....	
4.2 Принципы защиты населения в ЧС.....	
ЗАКЛЮЧЕНИЕ.....	
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ.....	
ПРИЛОЖЕНИЕ 1. Установка Wowza Streaming Engine.....	
ПРИЛОЖЕНИЕ 2. Обзор программы.....	

## **АННОТАЦИЯ**

Данная выпускная квалификационная работа посвящена изучению основ облачного вычисления и создания веб службы для организации видео стриминга. Целью выполнения выпускной работы является разработка веб приложения на основе облачной технологии SaaS для передачи видео файлов в реальном времени.

## **SUMMARY**

This graduation qualification work is devoted to learning the basics of cloud computing and the creation of web service organization for video streaming. The aim of this work is the development of Web applications based on SaaS cloud technology for the transmission of video files in real time

## ВВЕДЕНИЕ

В современном мире человек повсеместно окружён вычислительными устройствами. На протяжении всего периода развития вычислительной техники, её мощности постоянно возрастают. Согласно закону Мура, количество транзисторов, а, следовательно, и мощность вычислительных устройств растёт экспоненциально с течением времени. И даже несмотря на этот неумолимый рост, всегда существовали и досех пор существуют задачи, которым текущих мощностей одиночных компьютеров оказывается недостаточно. Например, одними из самых трудоёмких и в то же время распространённых и востребованных задач и по сей день остаются задачи потоковой обработки мультимедийной информации. Для эффективного решения таких вычислительно-трудоёмких задач компьютеры объединяются в различные вычислительные сети или кластеры. Огромные распределённые вычислительные кластеры часто именуют как – «облачные системы», а вычисления, выполняемые в них, соответственно, - «облачными вычислениями». К большой популярности таких распределённых платформ привели содержащийся в них огромный вычислительный потенциал и удобство для пользователя, который получает необходимые ему вычислительные ресурсы по требованию, избавлен от необходимости разбираться в технологических особенностях реализации и инфраструктуры используемых средств, и платит исключительно за то, чем пользуется. Всё это послужило причиной активного развития данной отрасли информационных технологий.

Существующие решения чаще всего предоставляют либо традиционную сервис-ориентированную модель, либо возможности по пакетной обработке заданий. При этом задача, отправляемая на решение в «облако», обычно описывается в виде некоторого запроса конечной длины и должна быть полностью известна до непосредственного начала выполнения вычислений, связанных с её решением. Это серьёзно ограничивает спектр задач, доступных для решения в подобных системах. Ярким примером задач, не решаемых

в существующих «облаках» общего назначения, является различная потоковая обработка данных, например, обработка «живых» мультимедийных потоков: транскодирование, наложение видео-фильтров, применение алгоритмов распознавания образов в видеопотоке и т.д.

Стремительное развитие информационных технологий за последние несколько десятков лет приучили людей к удобству вычислительной техники, которая сейчас используется повсеместно. Однако вычислительные ресурсы стоят денег, и, столкнувшись с очередной задачей, требующей применения подобных благ цивилизации, современный человек обладает целым рядом альтернативных возможностей.

В последнее время широкое распространение получили так называемые «облачные вычисления» - предоставление пользователю абстрактных вычислительных мощностей, которые физически распределены на многих удалённых устройствах, образующих так называемое «облако». Для пользователя данный подход значительно удобнее предыдущего, так как он платит не за покупку или пользование конкретными аппаратными средствами, а лишь за непосредственно выполненные для него расчёты. Стоит отметить, что в настоящее время термин «облачные вычисления» трактуется разными людьми по-разному. Обобщая, под «облачными системами» будем понимать огромные распределённые платформы. На самом деле, такие платформы совсем не обязательно должны быть вычислительными – существуют, например, многочисленные облачные хранилища данных (Amazon Simple Storage Service (Amazon S3), Dropbox, Google Drive, Yandex Disk).

**Цель и задачи работы.** Целью данной выпускной квалификационной работы является изучения основ **Облачного вычисления** и создания программного продукта для организации онлайн видео трансляции по интернету на основе технологии SaaS

**Объектом исследования** является технологии, средства разработки и языки программирования для создания WEB-системы. Методы доступа к базам данных средствами SQLite. Возможность создания WEB-сервиса, реали-

зующего задачи предметной области на основе технологий Java, Flash, PHP, MySql, CSS,HTML и Cloud Computing.

**Практическая значимость.** В наш век обмен информацией немыслим без современных средств связи. Одно из таких средств – современные глобальные компьютерные сети. Сети - важная часть группового взаимодействия, так как они позволяют быстро и эффективно обмениваться информацией.

Данная WEB-система позволяет организовать видео трансляцию. Данная система не требует особых оборудований для видео стриминга, видео потоки можно передавать с любого устройства имеющий видео камеру и доступ в интернет.

**Структура и объем работы.** Данная выпускная квалификационная работа состоит из 71 страниц, включающих в себя введение, 4 глав, заключение и приложения 1-2, инструкция по администрированию программный оболочки являющегося основой выполненной выпускной квалификационной работы.

Введение.

Глава 1. Основы Вычислительного Облака. В этой главе рассматриваются основы вычислительного облако и SaaS технологии.

Глава 2. Постановка Задачи.

Глава 3. Программное Решения На Основе Saas. Рассматриваются архитектура SaaS приложения, основы видео стриминга и создается ПО для организации онлайн видео вещания.

Глава 4. Безопасность Жизнедеятельности.

## Глава 1. ОСНОВЫ ВЫЧИСЛИТЕЛЬНОГО ОБЛАКА

В этой главе мы подробно познакомимся основами облачного вычисления. Сперва рассмотрим архитектуру облачно вычислительных систем на основе модели CCRA (Cloud Computing Reference Architecture) разработанных в **NIST** (National Institute of Standards and Technology).

NIST является, по сути, обобщённой концептуальной моделью высокого уровня и служит эффективным инструментом при формировании требований, структур и действий облачных вычислений. Архитектура определяет состав участников, деятельность и функции, которые могут быть реализованы в процессе разработки архитектур облачных вычислений и устанавливает взаимоотношения между участниками облачных вычислений. Она содержит некоторые примеры и описания, являющиеся основой при обсуждении требований и стандартов для облачных вычислений. На рисунке 1 показаны сервисы, с точки зрения задачи данной работы представляет интерес элемент «переносимость/интероперабельность».

Однако, вычисления являются только одной из областей применения «облаков». Второй областью является хранение информации. И, наконец, третьей областью является гибридное использование облаков: для сбора, хранения данных и в качестве средства обработки данных. Мы остановимся на проблеме интероперабельности (способность к взаимодействию) облаков с целью раскрыть содержание Рис.1.



Рис. 1 Эталонная архитектура облачных вычислений NIST.

На рисунке 2 показаны примеры сервисов для потребителей SaaS, PaaS и IaaS. Как следует из рисунков 1 и 2, все проблемы обеспечения потребителей требуемыми сервисами решает провайдер (поставщик). Следовательно, вопросы интероперабельности тоже решает провайдер.

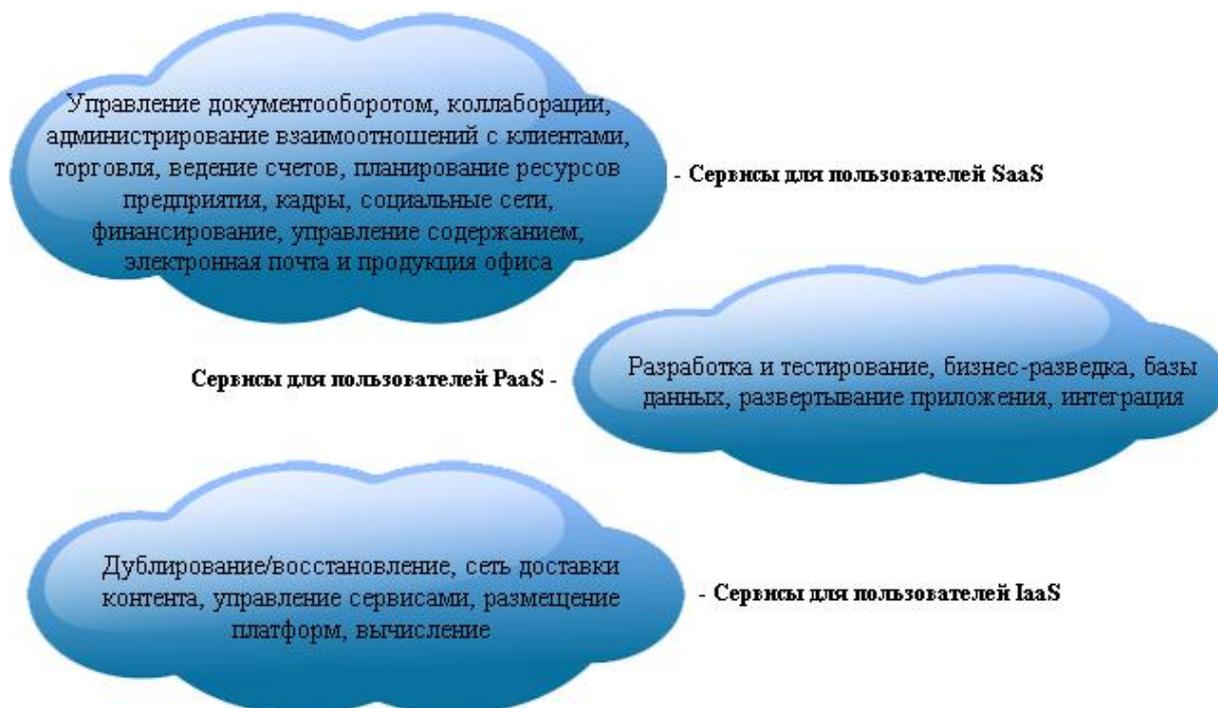


Рис. 2 Примеры сервисов для пользователей облачных вычислений.

На рисунке 3 показаны основные функции провайдера: развёртывание сервиса, оркестровка сервиса, облачный сервис-менеджмент, безопасность и конфиденциальность.



Рис.3 Основные функции провайдера облака.

Развертывание сервиса состоит в сборке ресурсов.

Оркестровка сервиса показана на рисунке 4.



Рис. 4 Сервис оркестровки провайдера облака.



Рис. 5 Сервис управления облаком провайдера облака.

Облачный сервис-менеджмент включает все сервис-ориентированные функции, которые необходимы для администрирования и действия затребованных сервисов или которые могут оказаться затребованными клиентами облака. Как показано на Рис. 5, сервис управления облака включает описание с точки зрения поддержки бизнеса, сборки и конфигурирования и с точки зрения удовлетворения требований переносимости и интероперабельности. Это подчёркивает ответственность провайдера за обеспечение интероперабельности между потребителем облачных сервисов и собственно провайдером. Интерфейсы, предоставленные потребителям облака, могут быть разделены на два главных типа, в которых интероперабельность задаётся отдельно для каждого типа. Как показано на рисунке 6, каждый вид облачной модели предоставления услуг содержит интерфейс к каждому типу.



*Рис. 6 Структура сервиса облака.*

Функциональный интерфейс (ФИ), который представляется к (или самой) части (частью) облака, выполняет главную функцию сервиса облака. В этом его отличие от интерфейса, который применяется для управления использованием сервиса облака (API).

### **IaaS-интерфейс**

В случае IaaS (см. рис.7), ФИ представляет собой виртуализованные центральный процессор (ЦП), память и пространство ввода/вывода, обычно применяемые операционной системой, и стек ПО, работающий с этой операционной системой.

Пользователь облака использует интерфейс администратора, чтобы управлять самому сервисом облака при старте, остановке и манипулировании образом виртуальной машины и подключенными к ней ресурсами.



*Рис. 7 IaaS-интерфейс.*

API IaaS является кандидатом на то, чтобы стать стандартным средством обеспечения интероперабельности. Примером стандарта интерфейса управления ресурсами IaaS является интерфейс, предложенный рабочей группой Open Cloud Computing Interface (OCCI) из международной организации Open Grid Forum. Стандарт интерфейса администрирования данных – Cloud Data Management Interface (CDMI) является примером интерфейса, как для администрирования хранения, так и функциональным интерфейсом хранилища.

### **РaaS-интерфейс**

Для PaaS, как показано на рис. 8, мы снова видим необходимость в разделении интерфейсов на два типа.



*Рис. 8 PaaS-интерфейс.*

ФИ PaaS представляет собой среду выполнения программ с использованием библиотек и компонентов, необходимых для приложения. API PaaS, однако, может быть очень похож на интерфейс управления IaaS модели. Интероперабельность интерфейсов управления самообслуживания может быть достигнута отдельно от интероперабельности функциональных интерфейсов PaaS.

### **SaaS-интерфейс**

В таком случае, когда SaaS-приложение заказывается с помощью Web-программы (рис. 9), возникает множество стандартов, которые используются для достижения интероперабельности между (что существенно) Web-

сервером и программой пользователя, например, IP (v4, v6), TCP, HTTP, SSL/TLS, HTML, XML, REST, Atom, AtomPub, RSS и JavaScript/JSON. Ни один из этих стандартов не специфицирован под облако, и, тем не менее, эти стандарты применяются в Web-браузерах, положенных в основу API.

В случае, когда SaaS-приложение заказано другой системой в качестве сервиса, облака или чем-то подобным, существуют различные стандарты, как на содержание данных, так и на интерфейсы. Самыми важными для интероперабельности являются канонические форматы данных, обычно сегодня представляемыми стандартами XML.

Существуют различные стандарты, отвечающие различным прикладным областям (т.е. OAGi BODs для бизнес-документов или ODF и OOXML для офисных документов). Также важным является комплект стандартов интероперабельности на интерфейсы, упаковку и транспортировку данных, примером которых служат SOAP, WS-\* и ebXML.



*Рис. 9 SaaS-интерфейс.*

Этих облачных сервисов (IaaS, PaaS, SaaS) будем подробно рассматривать далее.

## **2.1 Модели развёртывания**

Рассмотрим более подробно различные подходы к построению распределённых вычислительных сетей и Модели развёртывания

## **Частное облако**

Частное облако (*private cloud*) - концепция построения ИТ, направленная на реализацию принципа «ИТ, как услуга». Серверы объединяются в пул, из которого на решение определенных задач выделяются вычислительные ресурсы, не привязанные к конкретным физическим серверам. Предоставление мощностей из такого динамического пула осуществляется по запросам в службу Service Desk или через специальный портал - «витрину сервисов». При этом могут быть выделены виртуальные серверы под задачи бизнес-приложений или тестирования, предоставлены отдельные приложения или целые виртуальные рабочие места пользователей, а ПК на столах сотрудников могут быть заменены «тонкими клиентами» - небольшими устройствами, тихими и экономичными.

## **Публичное облако**

Публичное облако (*public cloud*) — облачная инфраструктура, используемая одновременно множеством компаний. Публичное облако физически располагается и находится в собственности, управлении и эксплуатации провайдера. Публичное облако наиболее простой способ реализации концепции IaaS (Infrastructure as a Service) — предоставления компьютерной инфраструктуры как услуги. В отличие от *частного облака*, публичное облако позволяет получить готовую инфраструктуру без первоначальных затрат. Публичное облако обладает практически неограниченными возможностями масштабирования.

## **Общественное облако**

Общественное облако (*community cloud*) — вид инфраструктуры, предназначенный для использования конкретным сообществом потребителей из организаций, имеющих общие задачи (например, миссии, требований безопасности, политики, и соответствия различным требованиям). Общественное облако может находиться в кооперативной (совместной) собственности,

управлении и эксплуатации одной или более из организаций сообщества или третьей стороны (или какой-либо их комбинации), и оно может физически существовать как внутри, так и вне юрисдикции владельца.

## **Гибридное облако**

Гибридное облако (*hybrid cloud*) — это комбинация из двух или более различных облачных инфраструктур (частных, публичных или общественных), остающихся уникальными объектами, но связанных между собой стандартизованными или частными технологиями передачи данных и приложений (например, кратковременное использование ресурсов публичных облаков для балансировки нагрузки между облаками).

Корпоративные ИТ-службы и государственные организации используют гибридные облака, чтобы упростить процессы, расширить возможности выбора, повысить оперативность, сократить затраты, улучшить соблюдение соглашений об уровне обслуживания, усовершенствовать средства визуализации и управления.

## **2.1 Модели обслуживания**

### **2.1.1 IaaS (Infrastructure-as-a-service)**

Infrastructure as a Service (**IaaS**) - инфраструктура как услуга. Модель предоставления облачных вычислений, при которой потребитель получает возможность управлять средствами обработки и хранения, а также и другими фундаментальными вычислительными ресурсами (виртуальными серверами и сетевой инфраструктурой), на которых он может самостоятельно устанавливать операционные системы и прикладные программы под собственные цели. По сути, потребитель арендует абстрактные вычислительные мощности (серверное время, дисковое пространство и пропускную способность сетевых каналов) или использует услуги аутсорсинга ИТ-инфраструктуры. Потребитель

тель не управляет основной инфраструктурой облака, но управляет операционными системами, хранилищем и развернутыми им приложениями.

IaaS - это предоставление пользователю компьютерной и сетевой инфраструктуры (servers, storage, networking) и их обслуживание как услуги в форме виртуализации, т.е. **виртуальной инфраструктуры**. Другими словами, на базе физической инфраструктуры дата-центров или ЦОД поставщик (провайдер) создает виртуальную инфраструктуру, которую предоставляет пользователям как сервис. Средства виртуализации позволяют преобразовать физическую инфраструктуру data centers в виртуальную и таким образом создать первый слой облачных услуг - IaaS.

*Что такое виртуализация?* Технология виртуализации ресурсов позволяет физическое оборудование (серверы, хранилища данных, сети передачи данных) разделить между пользователями на несколько частей, которые используются ими для выполнения текущих задач. Например, на одном физическом сервере можно запустить сотни виртуальных серверов, а пользователю для решения задач выделить время доступа к ним. Реализация виртуализации может быть осуществлена как на программном уровне, так и на аппаратном.

Таким образом, пользователю предоставляются абстрактные эластичные вычислительные мощности, например, не сервер или серверы, а серверное время для обработки его задач. Не диски для хранения данных, а необходимое дисковое пространство, не каналы связи, а требуемая для решения задач сетевая пропускная способность каналов коммутации.

Кроме виртуализации для создания IaaS используется автоматизация, которая обеспечивает динамическое распределение ресурсов без участия персонала поставщика услуг, т.е. система автоматически может добавлять или уменьшать количество виртуальных серверов, дисковое пространство

для хранения данных, или изменять сетевую пропускную способность каналов связи. Виртуализация и автоматизация обеспечивают эффективность использования вычислительных ресурсов и снижение стоимость аренды облачной услуги IaaS.

Как правило, IaaS предоставляется в аренду (предоставляется IaaS-сервис на условиях аутсорсинга) корпоративным пользователям. То есть пользователи получают интегрированные ресурсы для создания своей собственной вычислительной инфраструктуры. В этом случае пользователь должен сам установить и настроить OS и необходимые программы для выполнения производственных задач или для разработки приложений.

Концепция IaaS позволяет пользователю покупать только те вычислительные мощности, которые необходимы ему для выполнения конкретных задач. В состав дополнительных услуг IaaS может входить подключение любого физического оборудования пользователя к облачной платформе и его размещение в сети дата-центров.

Инфраструктура как сервис - это решение корпоративного уровня для предприятий разного масштаба. Инфраструктура может быть размещена как в центре обработки данных предприятия, так и во внешнем дата-центре. Услуги IaaS предназначены для создания и использования защищенных частных, публичных и гибридных облачных сред. Поставщики могут обеспечить построение гибридных облачных конфигураций, при которых объединяются локальные сети в офисе заказчика с сетями облачной платформы.

Кроме того, к IaaS-услугам облачных вычислений относится облачный хостинг (Cloud Hosting). Облачный хостинг - это хостинг, который может обеспечить динамическое распределение ресурсов, обладает возможностью автоматического масштабирования ресурсов и имеет повышенную отказоустойчивость. Облачный хостинг является существенной альтернативой вир-

туальному хостингу, хостингу на виртуальном выделенном сервере VPS / VDS и хостингу на физическом выделенном сервере.

Провайдер Cloud Hosting предоставляет владельцам сайтов только необходимые сайту ресурсы: виртуальные серверы, количество оперативной памяти и объем жесткого диска, а также возможности для управления инфраструктурой хостинга (например, выбор операционной системы, количества RAM, объема и типа HDD, числа ядер CPU, тактовой частоты и скорости доступа). Оплата за аренду облачного хостинга производится только по факту уже потребленных ресурсов: количества процессорного времени, объема дискового пространства, количества потребляемой оперативной памяти и скорости доступа к сайту.

При необходимости арендатор (владелец сайта) облачного хостинга может изменить ресурсы хостинга или настроить его на автоматическое увеличение ресурсов при росте нагрузки, но он всегда будет оплачивать только уже потребленные ресурсы. Облачный хостинг имеет повышенную отказоустойчивость, так как сайт, размещенный на нем, одновременно находится на нескольких виртуальных серверах и отказ одного из них не повлияет на работу сайта.

В настоящее время хостеры предлагают в аренду облачные хостинги с предустановленной CMS. Хостинг-провайдеры для организации таких облачных хостингов могут, например, развернуть на своих серверах платформу-как-инфраструктуру Jelastic с предустановленными CMS. Jelastic предоставляет платформу-как-инфраструктуру в виде полного стека, позволяющего развертывание облачных хостингов на физической инфраструктуре ЦОД хостинг-провайдера.

Функционал платформы Jelastic позволяет в один клик установить встроенные в нее CMS с оптимизированным веб-окружением, например,

Jelastic на Infobox. Jelastic является продуктом, который включает в себя функциональность PaaS и легко конфигурируемую инфраструктуру IaaS. Jelastic – это платформа для запуска Java- и PHP-приложений и может быть использована не только хостерами для организации облачных хостингов, но и корпорациями для создания среды разработки (частных или гибридных облаков) веб-приложений.

На облачных хостингах, размещают облачные сайты – это современные облачные приложения. В облачных сайтах (облачных приложениях) данные хранятся в облачных БД, серверные приложения сайтов хранятся, и выполняется на облачных виртуальных серверах, а клиентская часть сайта выполняется в браузере пользователя.

Среду облачных вычислений можно создавать на основе облачных решений Amazon EC2, IBM x86, Microsoft Azure, EMC, VMware, на базе open-source решения OpenStack, RackSpace на базе OpenStack и др., которые позволяют преобразовать центр обработки данных в динамичную ИТ-среду. Кроме того, для облачных вычислений используются облачные базы данных, т.е. базы данных, которые запускаются на платформах облачных вычислений. В качестве облачных баз данных применяются как SQL-ориентированные модели, так и модели данных NoSQL.

### **2.1.2 PaaS (Platform-as-a-service)**

Platform as a Service (**PaaS**) - платформа как услуга. Модель предоставления облачных вычислений, при которой потребитель получает доступ к использованию программной платформы: операционных систем, СУБД, прикладного ПО, средств разработки и тестирования ПО. Фактически потребитель получает в аренду компьютерную платформу с установленной операционной системой и специализированными средствами для разработки, разме-

щения и управления веб-приложениями. Потребитель не управляет основной инфраструктурой облака, включая сеть, серверы, операционные системы или хранилища данных, но управляет развернутыми приложениями и возможно параметрами настройки конфигурации среды окружения.

Сервис PaaS предоставляет программную платформу и ее обслуживающие как сервис в составе:

- OS - сетевая операционная система (Unix-системы, включая Ubuntu Server, BSD/OS Family, Solaris/SunOS и т.д. или Windows Server),
- Database - система управления базой данных СУБД (MySQL, Microsoft SQL, SQL Database, PostgreSQL, Oracle и др.),
- Middleware - программное обеспечение среднего слоя или связывающее (промежуточное) программное обеспечение, которое предназначено для обеспечения взаимодействия между различными приложениями, системами и компонентами,
- Software development tools and testing - инструментальное программное обеспечение для разработки веб-приложений и их тестирования (среда разработки ПО: программные фреймворки, библиотеки и т.д. для создания веб-приложений на языках программирования: Python, Java, PHP, Ruby, JS для Node.js и т.д.),
- App server - сервер приложений для разработки, тестирования, отладки и работы веб-приложений.

Итак, PaaS предлагает разработчикам ПО средства разработки, тестирования, развертывания и поддержки различных приложений. Кроме того пользователю предоставляются инструменты администрирования и управления. В основном PaaS используется для разработки и размещения веб-приложений (например, связанных распределенных приложений - SaaS mashup, облачных сайтов и т.д.).

### 2.1.3 SaaS (Software-as-a-service)

Software as a Service (**SaaS**) - программное обеспечение как услуга. В этой модели предоставления облачных вычислений потребитель использует приложения поставщика, запущенные в облачной инфраструктуре, которые доступны клиенту через интерфейс (web-браузер) или интерфейс программы. Потребители не могут управлять и контролировать лежащую в основе облака инфраструктуру, включая сеть, серверы, операционные системы, хранилища данных или даже изменять параметры настройки конкретного приложения.

По схеме SaaS поставляются следующие типы облачных приложений и их обслуживание: Business Apps, Office Web Apps, Management Apps, Communications, Security и др. Наиболее востребованными облачными приложениями являются: CRM (система управления взаимоотношениями с клиентами), HRM (система по работе с персоналом, т.е. с кадрами), ERP (система планирования ресурсов предприятия, например 1С), офисные приложения, средства коммуникаций и др. Salesforce.com является крупнейшим в мире поставщиком облачных приложений CRM.

Под средствами коммуникаций подразумевается электронная переписка (например, Gmail), аудио и видео чаты (например, Microsoft Lync Online), Cloud PBX или облачная АТС (например, виртуальная АТС Манго-Офис), облачный сервис MDM (Mobile Device Management - управление мобильным устройством). Облачный сервис MDM предназначен для работы с корпоративными системами при помощи мобильных устройств.

На различные мобильные устройства, которые работают под управлением облачной системы MDM, устанавливаются приложения, так называемые агенты. Эти приложения обеспечивают централизованную настройку мобильных устройств и доступ к корпоративной сети предприятия в виде об-

лачной услуги SaaS. Как правило, облачные средства коммуникаций интегрируются с другими услугами SaaS, например, CRM+MDM, Office Web Apps+Lync Online, Google Docs+Gmail + Hangouts и т.д.

Основными потребителями SaaS являются предприятия малого и среднего бизнеса. Большинство SaaS-приложений предназначены для поддержки взаимодействия между сотрудниками, совместно работающими над решением общих задач (Collaboration). Архитектура SaaS-приложений, в которой единый экземпляр приложения, запущенный на сервере, обслуживает множество потребителей, является мультиарендной (Multi-tenant), т.е. каждому потребителю в процессе выполнения задач предоставляется свой экземпляр виртуального приложения.

Основные application software:

- Business Apps (CRM, FRM, IBM® B2B Cloud Services, Axway Cloud B2B, amoCRM SaaS сервис для B2B, Google Apps for Business),
- Business Intelligence (PowerBI в составе Office 365/Microsoft, Oracle Business Intelligence Managed Cloud Service, Anaplan/ADE Professional Solutions, Brand Analytics),
- Office Web Apps (Google Docs, Office Online/Microsoft OneDrive, Office Web Apps/Microsoft, Zoho Docs, IBM SmartCloud Docs и т.д.),
- Management Apps (ERP/АРЕНДА 1С, HRM, SCM, MRP),
- Communications (Gmail, Google Hangouts, Microsoft Lync Online, Cloud PBX или облачная АТС, MDM),
- Security (Panda Cloud Email Protection, Panda Cloud Internet Protection, McAfee SaaS Email Protection & Continuity, Comfortway Mobile Security и т.д.), и др.
- Collaboration and Multi-tenant (Google Docs, Google Sites, Microsoft Office Online, Office 365, Office Web Apps).

Существуют и множество других SaaS-услуг облачных вычислений, например, Cisco WebEx - облачный сервис для проведения web-конференций; CMS на основе модели SaaS (например, SaaS-платформа UMI.CLOUD); E-Commerce B2B/B2C по модели SaaS; Маркетинг SaaS-решений; услуга «Антивирус Dr.Web» по модели SaaS; SugarCRM — коммерческая CRM-система с открытыми исходными кодами; BPMonline CRM с инструментами моделирования и автоматизации бизнес-процессов и т.д.

## 2.2 Платформы для облачных вычисления

### 2.2.1 Microsoft Azure

**Microsoft (Windows) Azure** — название облачной платформы Microsoft. Предоставляет возможность разработки и выполнения приложений и хранения данных на серверах, расположенных в распределённых дата-центрах. Microsoft Azure реализует три облачные модели — платформы как сервиса (PaaS), программа как сервис (SaaS) и инфраструктуры как сервиса (IaaS). Работоспособность платформы Windows Azure обеспечивает сеть глобальных дата-центров Microsoft.

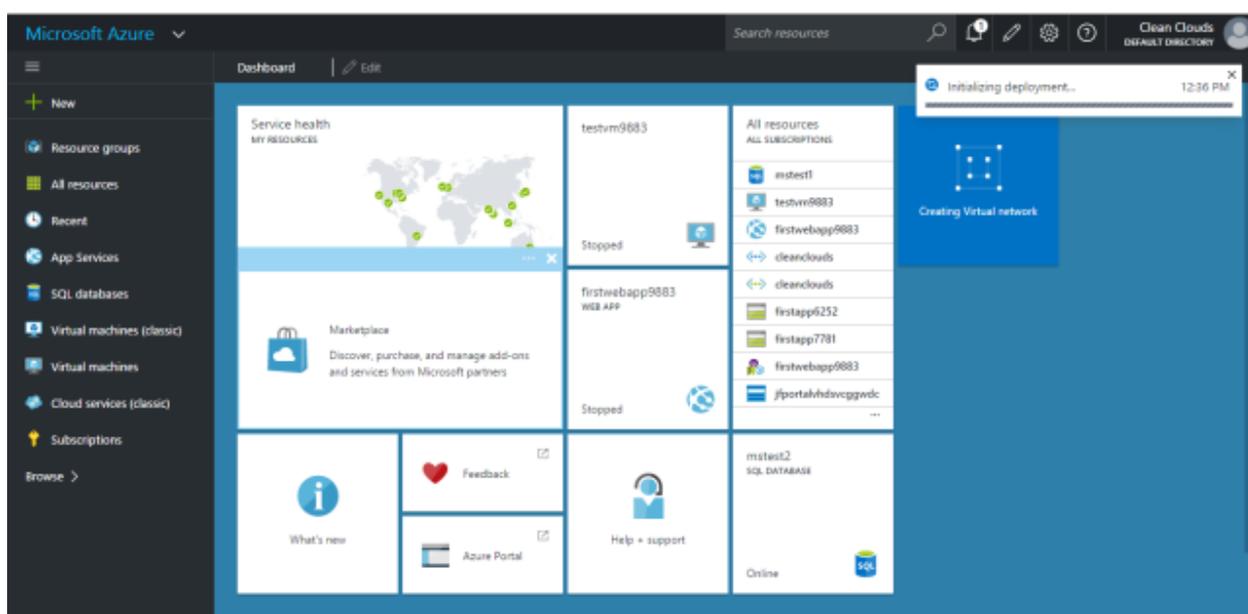


Рис. 10 Microsoft Azure интерфейс пользователя

Основные особенности данной модели:

- оплата только потребленных ресурсов;
- общая, многопоточная структура вычислений;
- абстракция от инфраструктуры.

В основе работы Microsoft Azure лежит запуск виртуальной машины для каждого экземпляра приложения. Разработчик определяет необходимый объём для хранения данных и требуемые вычислительные мощности (количество виртуальных машин), после чего платформа предоставляет соответствующие ресурсы. Когда первоначальные потребности в ресурсах изменяются, в соответствии с новым запросом заказчика платформа выделяет под приложение дополнительные или сокращает неиспользуемые ресурсы дата-центра.

Microsoft Azure как PaaS обеспечит не только все базовые функции операционной системы, но и дополнительные: выделение ресурсов по требованию для неограниченного масштабирования, автоматическую синхронную репликацию данных для повышения отказоустойчивости, обработку отказов инфраструктуры для обеспечения постоянной доступности и многое другое.

Microsoft Azure также реализует другой тип сервиса — инфраструктуру как сервис. Модель предоставления инфраструктуры (аппаратных ресурсов) реализует возможность аренды таких ресурсов, как серверы, устройства хранения данных и сетевое оборудование. Управление всей инфраструктурой осуществляется поставщиком, потребитель управляет только операционной системой и установленными приложениями.

В галерее образов доступны образы следующих операционных систем: Windows Server (2008, 2012, Technical Preview), CoreOS, Ubuntu Server, CentOS, openSUSE, SUSE Linux Enterprise Server, Oracle Linux.

В 2013 году было представлено новое хранилище образцов виртуальных машин — VM Depot — это проект для сообщества Windows Azure, за-

пущенный командой Microsoft Open Technologies. Содержимое портала, а также настроенные для разных задач виртуальные машины, будут создаваться и публиковаться силами сообщества.

Microsoft Azure состоит из:

- *Compute* — компонент, реализующий вычисления на платформе Windows Azure.
- *Storage* — компонент хранилища предоставляет масштабируемое хранилище. Хранилище не имеет возможности использовать реляционную модель и является альтернативной, "облачной" версией SQL Server.
- *Fabric* — Windows Azure Fabric по своему назначению является «контролёром» и ядром платформы, выполняя функции мониторинга в реальном времени, обеспечения отказоустойчивости, выделения мощностей, развертывания серверов, виртуальных машин и приложений, балансировки нагрузки и управления оборудованием.

Практически все сервисы Microsoft Azure имеют интерфейс взаимодействия API, построенный на основе ограничений для распределённых гипер-систем REST, что позволяет разработчикам использовать «облачные» сервисы с любой операционной системы, устройства и платформ

### 2.2.2 Google Apps

**Google Apps for Work** (ранее Google Apps для бизнеса) — пакет облачных сервисов и приложений для совместной работы, предлагаемый компанией Google по подписке.

В пакет входят популярные веб-приложения Google, в том числе Gmail, Google Диск, Google Hangouts, Google Календарь и Google Документы. В сервисах Google Apps for Work есть те же функции, что и в бесплатных вер-

сиях продуктов. Помимо этого, они предоставляют дополнительные возможности для корпоративных клиентов, такие как адреса электронной почты в домене компании, 30 ГБ и более для хранения писем и документов, а также круглосуточная поддержка по телефону и электронной почте. Поскольку Google Apps является облачным решением, то все пользовательские данные хранятся в защищенных дата-центрах Google, а не на локальных серверах клиента, как при использовании стандартного специализированного ПО.

## **Глава 2. ПОСТАНОВКА ЗАДАЧИ**

В рамках данной дипломной работы передо мной были поставлены следующие задачи:

1. Основы видео стриминга и обработки видео потоков в удаленных серверах
2. Спроектировать и реализовать архитектуру видео стриминга на основе технологии SaaS
3. Реализовать возможность решения задач потоковой обработки данных
4. Реализовать возможность общения по произвольным протоколам передачи данных
5. Реализовать возможность взаимодействия сервисов путём обмена произвольными сериализуемыми данными
6. Разработать средства для развёртывания и поддержания работы видео стриминга

## **Глава 3. ПРОГРАММНОЕ РЕШЕНИЯ НА ОСНОВЕ SAAS**

### **3.1 Теоритический часть**

Потоковое мультимедиа (от. англ. stream media) — это мультимедиа, которое непрерывно получается пользователем от провайдера потокового вещания. Это понятие применимо как к информации, распространяемой через телекоммуникации, так и к информации, которая изначально распространялась посредством потокового вещания (например, радио, телевидение) или непотоковой (например, книги, видеокассеты, аудио CD).

Первые попытки отображения мультимедиа информации на компьютерах начались в середине XX века. Однако, прогресс в этой сфере был очень малым, вследствие высокой стоимости и ограниченных возможностей компьютеров тех времён.

С конца 1980-х и до 1990-х компьютеры, доступные потребителям, уже были способны отображать различные виды информации. Основными техническими проблемами потокового вещания были:

- наличие достаточно производительного CPU и шины для передачи мультимедиа необходимого битрейта
- создание ОС, при работе которых гарантируется высоконадёжная передача данных.

Тем не менее, компьютеры сети оставались ограниченными, а потоковое мультимедиа уступало традиционному (CD-ROM).

В период с 1990 до 2000 пользователи интернета получили:

- высокую пропускную способность сетей, в частности, на последней миле
- возросло количество абонентов сетей, особенно Интернета
- стали использоваться стандартизованные протоколы и форматы, такие как TCP/IP, HTTP и HTML

- появилась коммерция в Интернете

Эти достижения в области сетей в совокупности с высокопроизводительными домашними компьютерами и современными операционными системами сделали потоковую мультимедийную информацию доступной широкому кругу простых пользователей. Автономные приёмники интернет-радио предлагали пользователям возможность прослушивания потокового звука без наличия компьютера.

В основном, мультимедиа информация занимает большие объемы, так что затраты на хранение и передачу подобной информации всегда велики; поэтому, в большинстве случаев, передаваемая в поток информация сжимается при передаче в сеть вещания.

Мультимедиа потоки бывают двух видов: по запросу или живыми. Потоки информации, вызываемой по запросу пользователя хранятся на серверах продолжительный период времени. Живые потоки доступны короткий период времени, например, при передаче видео со спортивных соревнований.

Размер, необходимый для хранения потоковой мультимедиа информации (в большинстве файловых систем выражается в мегабайтах, гигабайтах, терабайтах и т. д.) вычисляется в зависимости от скорости передаваемой информации и продолжительности информации по следующей формуле (для одного пользователя и файла): размер хранилища (в мегабайтах) = продолжительность (в секундах) \* битрейт (в кбит/с) / (8 \* 1024)

Пример: Один час видео, закодированного со скоростью 300 кбит/с (типичное видео транслируемое в интернете, имеющее размер 320×240 пикселей) будет занимать: (3,600 с \* 300 кбит/с) / (8\*1024) порядка 130 Мб места на диске

Если файл, хранимый на сервере с режимом передачи по запросу будет просматриваться 1000 людей одновременно по протоколу Unicast (1 клиент — 1 соединение), то сервер должен иметь следующую пропускную способность:  $300 \text{ кбит/с} * 1,000 = 300,000 \text{ кбит/с} = 300 \text{ Мбит/с}$  сетевого интерфейса

Это эквивалент порядка 125 Гб информации в час. Разумеется, при использовании протокола Multicast нагрузка на сервер намного ниже, так как для передачи информации всем клиентам используется единственный поток. Следовательно, такой поток будет занимать всего 300 кбит/с сетевого интерфейса сервера.

### **Протоколы потокового вещания**

Разработка сетевых протоколов потокового вещания вызывает следующие проблемы:

- Датаграмные протоколы, такие как User Datagram Protocol (UDP), отправляют поток медиаинформации как поток отдельных маленьких пакетов. Он прост и эффективен; в то же время, в спецификации протокола нет гарантии доставки данных получателю. Это очень сильно затрудняет поиск и исправление получаемых данных принимающим информацию приложением. При потере данных поток может быть отключен.
- Протоколы RTSP, RTP и RTCP специально разрабатывались для передачи мультимедийной информации по сети. Последние два построены на основе UDP.
- Надежные протоколы, такие как TCP, гарантируют корректность получаемых данных клиентов потокового вещания. Однако при большом количестве ошибок при соединении/подтверждении получаемой информации передаваемая информация может стать неактуальной. Это также может вызвать значительные задержки при передаче информации на время, затраченное на

пересылку поврежденной информации. Одним из решений данной проблемы является буферизация информации на стороне клиента.

- Протоколы Unicast отправляют отдельную копию данных каждому клиенту. Unicast подходит для большинства пользователей сети Интернет, но сильно затрудняет масштабирование сервера для большего количества клиентов.

- При широковещательной передаче одна копия данных передается всем клиентам сервера.

Протоколы Multicast разработаны для снижения нагрузки с серверов на подключения/ширину канала при получении потокового мультимедиа большим количеством клиентов. Эти протоколы отсылают одну порцию данных целой группе клиентов. В зависимости от типа сетевой инфраструктуры, групповая передача данных может быть доступна, а может и не быть. Одним из потенциальных недостатков групповой передачи является отсутствие возможности реализовать функцию видео по запросу. Непрерывное вещание потоковой информации также делает невозможным управление воспроизведением пользователем. Однако, эта проблема может быть решена внедрением в сеть передачи данных кэширующих серверов и буферизирующего принимаемый поток программного обеспечения.

- Multicast позволяет передавать один поток информации группе клиентов по сети. Одной из проблем при реализации подобной схемы потокового вещания является корректная настройка маршрутизаторов для передачи широковещательных пакетов из одного сегмента сети в другой. Если организация, предоставляющая потоковое вещание, имеет контроль над сетью между сервером и клиентами (например, в образовательной, правительственной или корпоративной сети), то протоколы маршрутизации, такие как IGMP и PIM,

могут быть использованы для доставки мультимедиа нескольким клиентам из различных сегментов LAN.

- Протоколы P2P могут использоваться при распространении предварительно записанной мультимедиа между компьютерами. Это снимает нагрузку с сервера, однако сеть передачи данных между сервером и одним из клиентов становится узким местом данного варианта реализации потокового вещания информации.

Так как пропускная способность каналов ограничена и все крупные сервера, раздающие медиаконтент обычным способом очень сильно перегружены. Распределенные файлообменные сети существенно снижают нагрузку, однако реальная скорость передачи данных у них чрезвычайно низка.

Компромиссной технологией раздачи медиаконтента является онлайновое вещание по технологии Multicast, обеспечивающей одновременную доставку идентичного контента всем запросившим его пользователям, что существенно разгружает каналы передачи данных. Это также ограничивает свободу пользователей в выборе контента, поскольку если к серверу подключились сто тысяч пользователей и каждый из них выбирает свой файл, то никакого выигрыша владелец сервера не получит. С другой стороны, можно иметь несколько независимых Multicast-каналов, передающих различные файлы, к которым может подключаться кто угодно. Разница между обычным скачиванием файла с сервера в том, что трансляция не позволяет слушателем/зрителем управлять потоком, и они вынуждены слушать/смотреть файл с момента подключения к серверу, который к тому времени мог проиграть половину файла. В некоторых случаях это приемлемо, в некоторых - нет. Как показывает практика, достаточно большой аудитории пользователей совершенно неважно, что именно играет в данный момент - главное, чтобы что-то вообще играло.

## Обзор мультимедиа серверов

С развитием интернет технологий, потоковое вещание мультимедиа вышло на новый уровень. Сегодня с легкостью можно найти тысячи ссылок ведущих на множества потоков музыки или видео. Для организации серверов, с которых ведется потоковое вещание разработано множество программного обеспечения. Большая часть из которого предназначена для вещания аудио данных в форматах mp3 или ogg. Для видео данных набор программного обеспечения ничуть не меньше, но серверов, которые могли бы полностью покрыть потребности медиасервисов довольно немного и основная часть является коммерческими проектами. Наиболее популярными на сегодняшний момент серверами для потокового видео можно назвать TVersity, QuickTime Broadcaster, VideoLAN, Windows Media Services, FFserver (FFmpeg), Red5, Wowza, Adobe Media Services и т.д.

FFmpeg — набор свободных библиотек с открытым исходным кодом, которые позволяют записывать, конвертировать и передавать цифровое аудио и видео в различных форматах. Он включает libavcodec — библиотеку кодирования и декодирования аудио и видео и libavformat — библиотеку мультиплексирования и демуплексирования в медиаконтейнер. Название происходит от названия экспертной группы MPEG и FF, означающего fast forward.

FFmpeg разработан под ОС на основе Linux, однако может быть скомпилирован под многие другие операционные системы. Разработчики не выпускают релизов и рекомендуют использовать последнюю версию из Subversion. Распространяется под лицензиями GNU LGPL или GNU GPL.

Серверы на основе FFmpeg часто организуют на вебхостинге, создавая различные видео порталы. Но зачастую данную библиотеку используют лишь как конвертер для медиафайлов при загрузке их на сервер.

Wowza - серверное программное обеспечение для организации как вещания потокового аудио и видео, так и доставки видео по запросу. Может осуществлять вещание на различные типы устройств и клиентов, включая Adobe Flash, Microsoft Silverlight, Apple QuickTime и устройства, под управлением iOS (iPad, iPhone, iPod Touch), мобильные телефоны стандарта 3GPP (Android, BlackBerry OS, Symbian), устройства IPTV (Amino, Enseio, Roku и другие), игровые консоли (Wii, PS3).

В течении этой дипломной работы в качестве сервера Видео вещания мы будем использовать Wowza Streaming Engine 4.4.1.

### **3.2 Используемые технологии**

Первым решаемым в данной работе вопросом стал выбор технологий для разработки видео вещания. В силу необходимости обеспечить кроссплатформенность создаваемого программного продукта и внедрить облачное решение на основе SaaS, выбор был ограничен языками программирования PHP и технологии FLASH, а в качестве сервера для обработки и вещания видео в реальном времени был выбран Wowza Streaming Engine.

Язык программирования PHP хорошо подходит для решения задач, связанных с потоковой обработкой данных. В то время как FLASH позволяет читать и отправлять видео потоки в реальном времени.

Использование языков PHP не требует от программиста больших временных затрат, однако получаемые при этом средства для глубокой системной интеграции и возможности написания высокопроизводительного программного кода, оправдывают их выбор в качестве главного инструмента для решения поставленных передо мною задач.

Для ускорения разработки некритичных к производительности фрагментов создаваемого решения активно использовались готовые библиотеки, входящий в состав Wowza Streaming Engine и предоставляющие огромное ко-

личество программных средств для решения многих задач общего назначения, как работа с файловой системой, сетью, потоками исполнения, сериализацией/десериализацией данных.

Унифицированный пользовательский веб-интерфейс распределённой вычислительной среды создавался с использованием языков HTML5 , CSS и JavaScript.

### **3.3 Установка и настройка Wowza Streaming Engine**

Чтобы организовать видео стриминг первым делом мы должны установить и настроить сервер для передачи и обработки видео потоков.

Медиа сервер Wowze является платным программным обеспечением но в официальном сайте можно скачать и установить пробную версию ПО. Установка ПО займет около 10 минут. Полное руководство по установке можно посмотреть в Приложение 1.

После установки запустим сервер и в браузере откроем ссылку <http://127.0.0.1:8088/enginemanager/ftu/>.

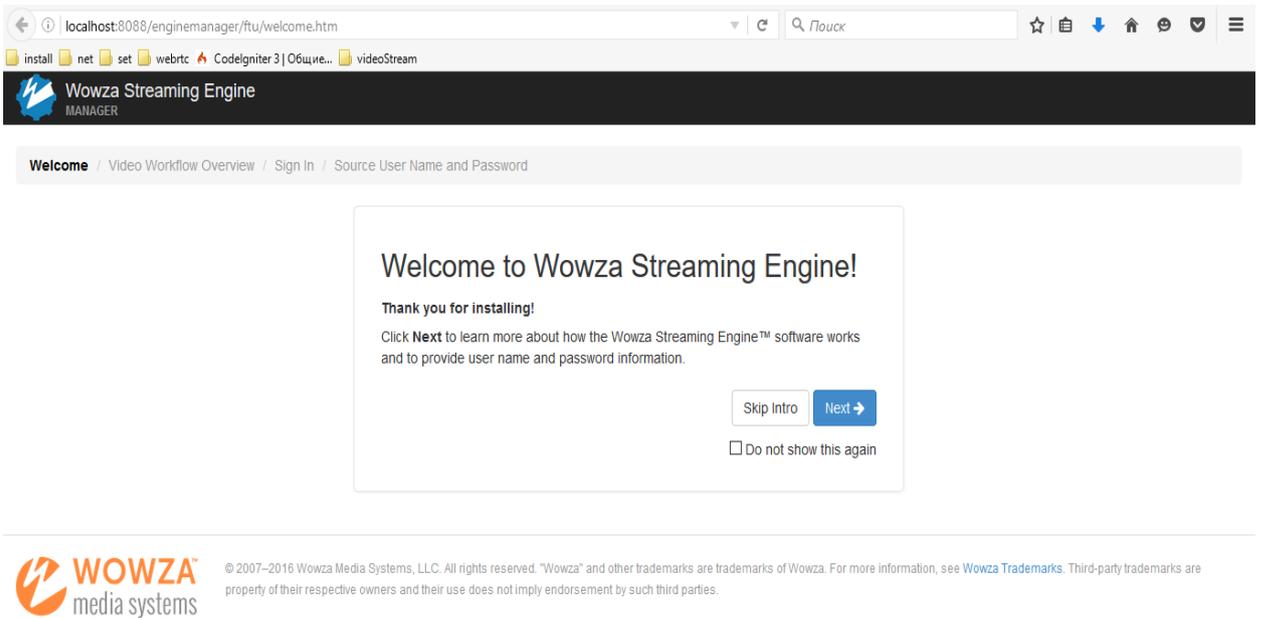


Рис. 11 Нажмем кнопку Next

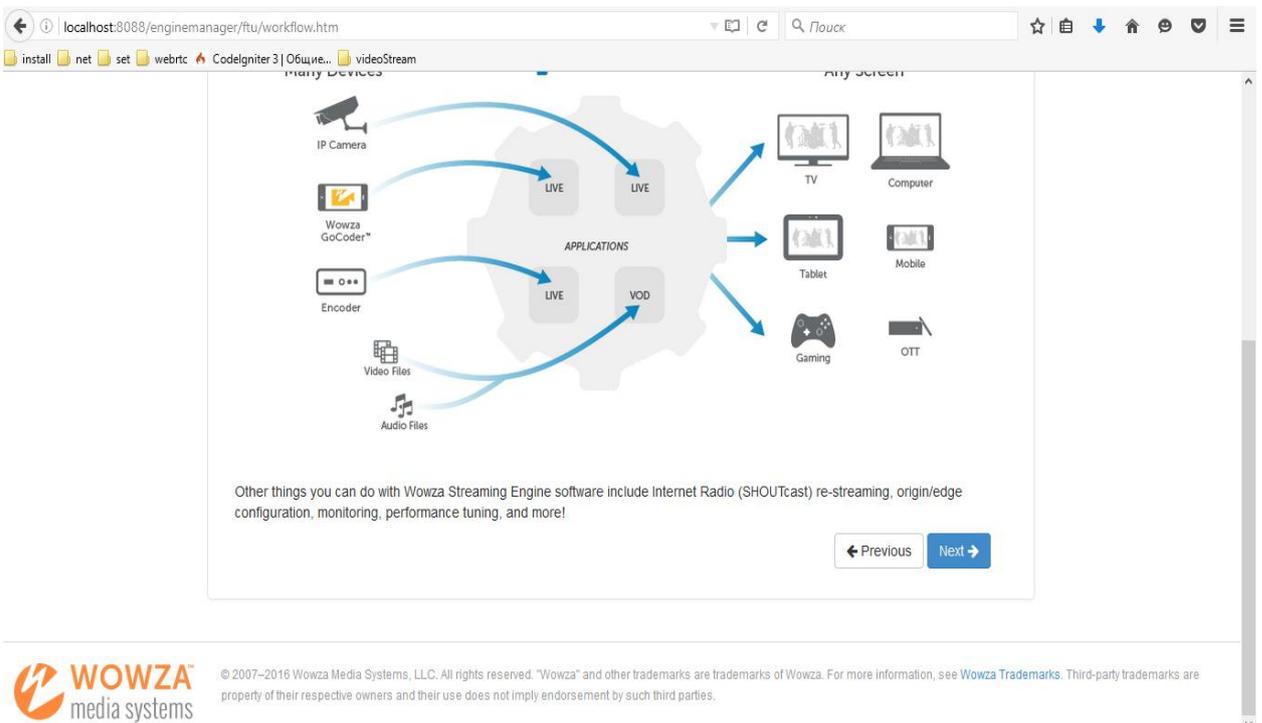
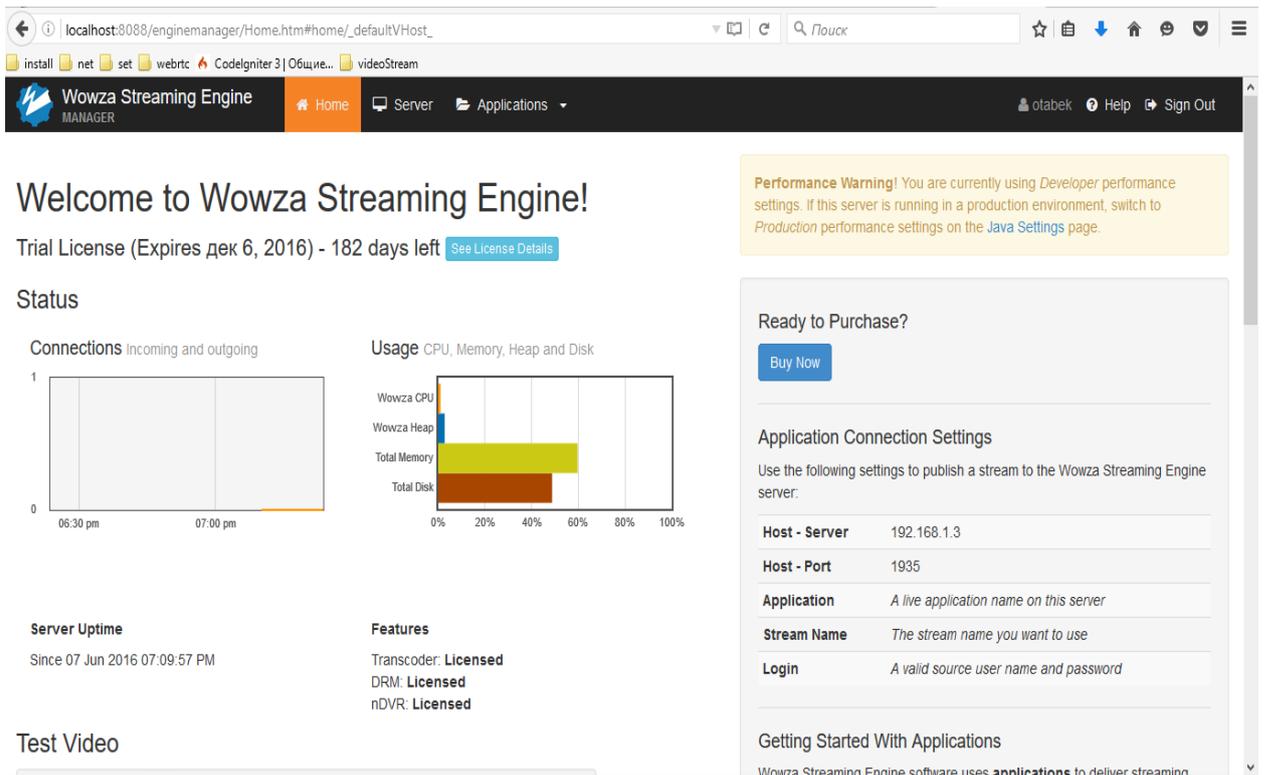
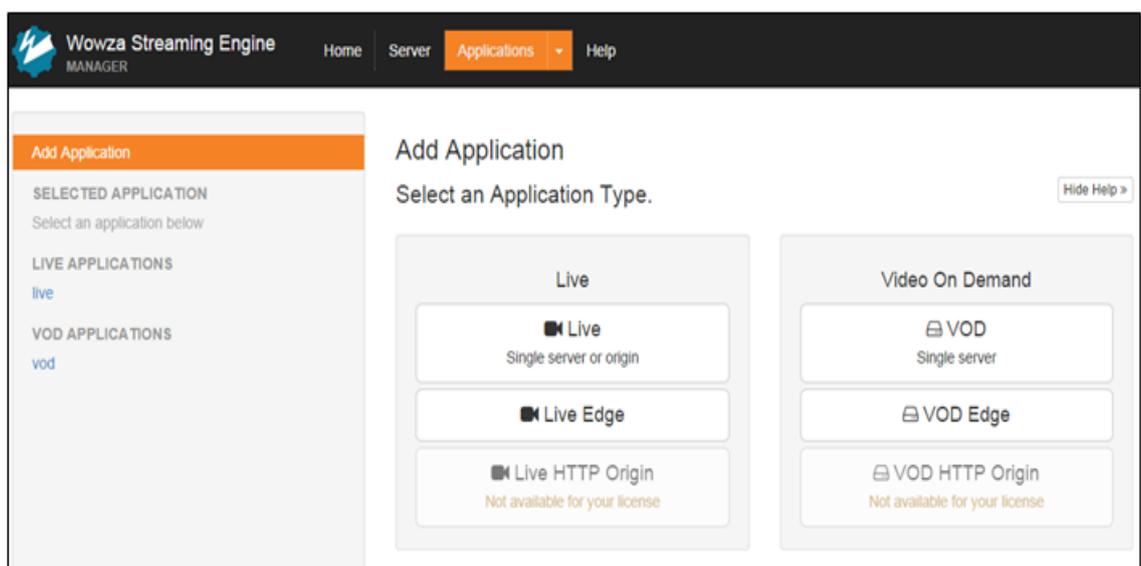


Рис. 12



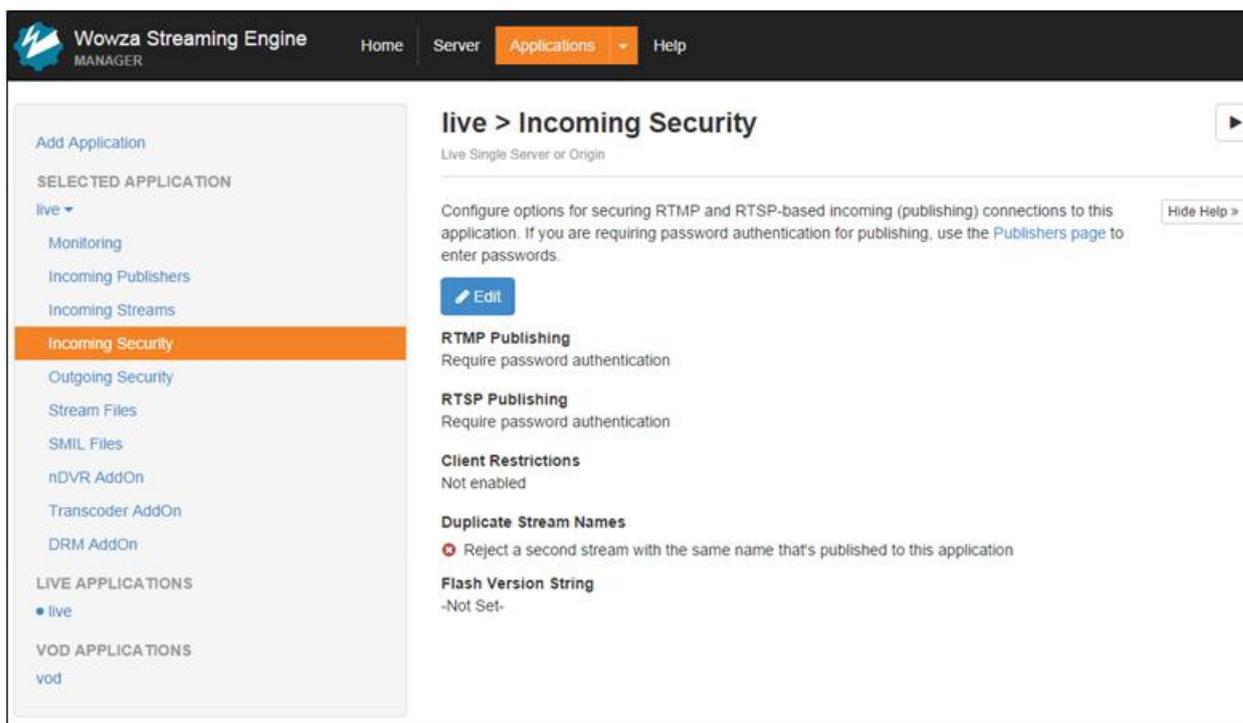
*Рис. 13* Главный окно медиа сервера

Перейдем на вкладку Applications. Выберем приложение live в списке **LIVE APPLICATIONS**



*Рис. 14*

Далее настроим параметры аутентификации потоков. Выбираем пункт **Incoming Security** и нажимаем **Edit**



*Рис. 15*

Отметим опцию **Require password authentication** (Необходима аутентификация паролем) в **RTMP /RTSP Publishing** и нажмем **Save**. Теперь медиасервер будет принимать только видеопотоки подписанные учетной записью издателя.



Рис. 16

Далее перейдем к пункту **Outgoing security**

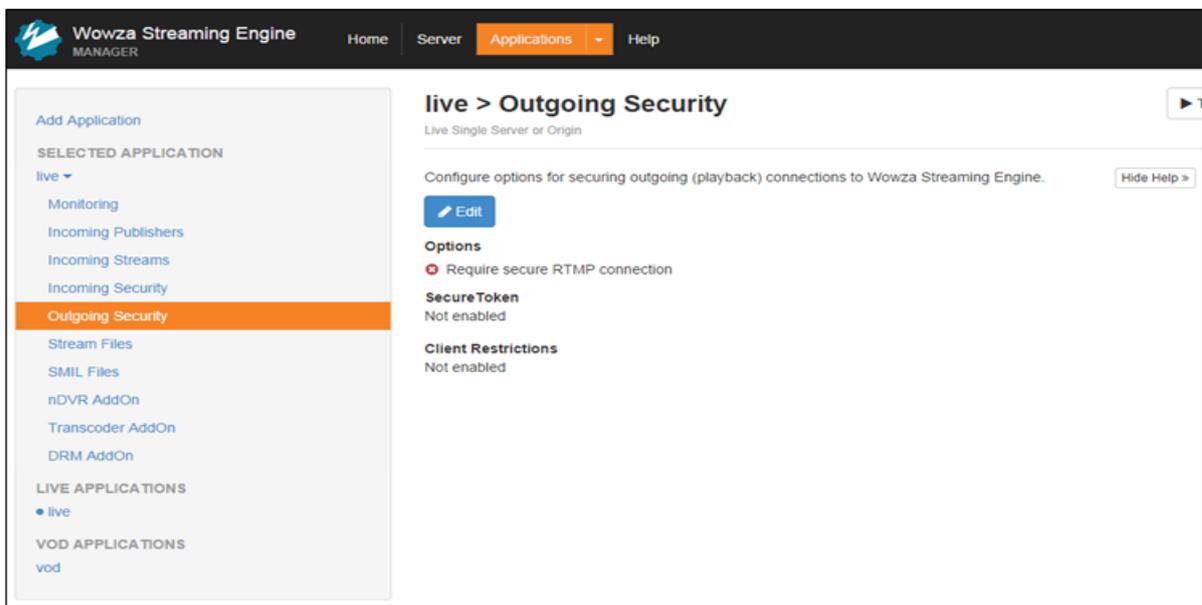


Рис. 17

Чтобы позволить медиасерверу отдавать потоки без шифрования снимаем галочку с опции **Require secure RTMP connection**, если она стоит

**live > Outgoing Security**  
Live Single Server or Origin

\* = Required field Hide Help >

**Options**

Require secure RTMP connection

**SecureToken**

Enable SecureToken content protection

Enter or generate a string of characters to use as your shared secret.

**Client Restrictions**

No client restrictions

Only allow playback from the following IP addresses

Do NOT allow playback from the following IP addresses

Comma-separated list of IP addresses

*Рис. 18*

Теперь сервер готов принимать входящие потоки по протоколу RTMP. Потоки следует отправлять на адрес **rtmp://[ip-сервера]:1935/live**. Активные потоки можно посмотреть в **Incoming Streams**

**live > Incoming Streams** ▶ Test  
Live Single Server or Origin

Active Streams Streams available for recording via this application. Hide Help >

**Default Instance (\_definst\_)**

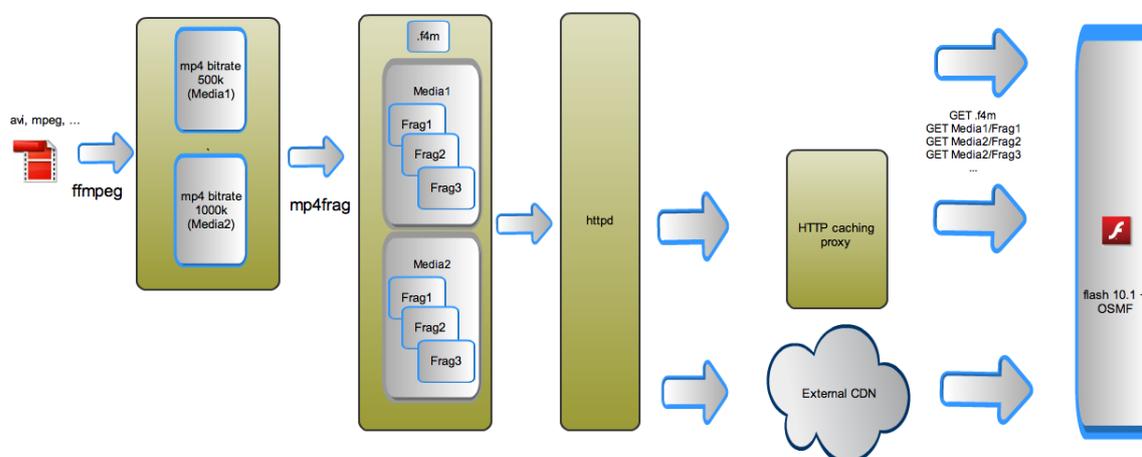
Stream	Status	Actions
→ myStream rtmp://10.0.165.174:50744	Active	

*Рис. 20*

### 3.4 Разработка ПО

Видео стриминг в течении этой книги создается на основе технологии Flash а в качестве языка программирования используется PHP и Java.

ПО разворачивается на OpenServer а в качестве медиа сервера используется Wowza Stream Engine. Ниже на рис 21 показана схема взаимодействия HTTP и Медиа Сервера с технологий Flash.



create and share your own diagrams at gliffy.com



Рис. 21

Adobe Flash – программа кодировщик, которая захватывает видео и аудио сигнал, кодирует и отправляет его на сервер. Сервер принимает эти данные обрабатывает и отправляет нужным пользователям. А приложения на PHP вступает в качестве посредника.

Первым делом создадим PHP приложения а в качестве Flash будем использовать готовый скрипт который позволит захватывать изображения с камеры пользователя.

PHP приложения состоит из нескольких страниц и flash скриптов. А так же в себя включает css и js файлы.

```
<object width="100%" height="500" id="videowhisper_livebroadcast" type="application/x-shockwave-flash" data="<?=$swfurl?>">
  <param name="movie" value="<?=$swfurl?>"></param>
  <param bgcolor="<?=$bgcolor?>" /></param>
  <param name="scale" value="noscale" > </param>
  <param name="salign" value="lt"></param>
  <param name="allowFullScreen" value="true"></param>
  <param name="allowscriptaccess" value="always"></param>
  <param name="base" value="<?=$baseurl?>"></param>
  <param name="wmode" value="<?=$wmode?>" ></param>
</object>
```

Рис. 22

Таким образом к php файлу подключается flash скрипт который захватывает камеру и возвращает видео поток, этот поток отправляется в медиа сервер.

Ниже приведены настройки медиа сервера.

```
$rtmp_server = "rtmp://localhost:1935/videowhisper-live";
$rtmp_amf = "AMF3";
$rtmfp_server="rtmfp://stratus.adobe.com/f1533cc06e4de4b56399b10d-1a624022ff71/";
$httpstreamer="http://localhost:1935/videowhisper/";
```

Рис. 23

В данном случае медиа сервер размещен по адресу <http://localhost:1935>. Выше приведенные четыре строки кода позволяют нам подключаться к медиа серверу.

Теперь нам осталось настроить медиа сервер таким образом чтобы он смог принимать данные от нашего ПО.

Для этого первым делом создадим пустую папку с именем нашего ПО в папке application медиа сервера. В этой папке будут сохранены временные видео файлы.

Далее создадим конфигурационный файл в формате XML. Создадим в папке conf папку с названием ПО и разместим файл в эту папку.

```
<?xml version="1.0" encoding="UTF-8"?>
<Root version="1">
  <Application>
    <Name>videowhisper</Name>
    <AppType>Live</AppType>
    <Description>Live Video Streaming</Description>
    <Connections>
      <AutoAccept>>true</AutoAccept>
      <AllowDomains></AllowDomains>
    </Connections>
    <Streams>
      <StreamType>live</StreamType>
      <StorageDir>${com.wowza.wms.context.VHostConfigHome}/content</StorageDir>
      <KeyDir>${com.wowza.wms.context.VHostConfigHome}/keys</KeyDir>
      <LiveStreamPacketizers>cupertinostreamingpacketizer, mpegdashstreamingpacketizer,
      sanjosestreamingpacketizer, smoothstreamingpacketizer</LiveStreamPacketizers>
      <Properties>
      </Properties>
    </Streams>
  </Application>
</Root>
```

Рис. 24

```
<RTP>
  <Authentication>
    <PublishMethod>digest</PublishMethod>
    <PlayMethod>none</PlayMethod>
  </Authentication>
  <AVSyncMethod>senderreport</AVSyncMethod>
  <MaxRTCPWaitTime>12000</MaxRTCPWaitTime>
  <IdleFrequency>75</IdleFrequency>
  <RTSPSessionTimeout>90000</RTSPSessionTimeout>
  <RTSPMaximumPendingWriteBytes>0</RTSPMaximumPendingWriteBytes>
  <RTSPBindIpAddress></RTSPBindIpAddress>
  <RTSPConnectionIpAddress>0.0.0.0</RTSPConnectionIpAddress>
  <RTSPOriginIpAddress>127.0.0.1</RTSPOriginIpAddress>
  <IncomingDatagramPortRanges>*</IncomingDatagramPortRanges>
  <Properties>
  </Properties>
</RTP>
```

Рис. 25

Теперь медиа сервер по умолчанию будет слушать адрес 127.0.0.1 и обрабатывать все медиа потоки с этого адреса.

Нам только осталось написать сценарии на языке Java чтобы медиа сервер СМОГ ПОНЯТЬ НОВЫЕ НАСТРОЙКИ.

PhpSerialize.class	13.10.2014 1:36	Файл "CLASS"	4 КБ
SerializedPhpParser\$1.class	13.10.2014 1:36	Файл "CLASS"	1 КБ
SerializedPhpParser\$PhpObject.class	13.10.2014 1:36	Файл "CLASS"	1 КБ
SerializedPhpParser.class	13.10.2014 1:36	Файл "CLASS"	5 КБ
VideoWhisper\$1FirstResult.class	13.10.2014 1:47	Файл "CLASS"	2 КБ
VideoWhisper\$1ResultObj.class	13.10.2014 1:47	Файл "CLASS"	4 КБ
VideoWhisper\$vwClient.class	13.10.2014 1:47	Файл "CLASS"	3 КБ
VideoWhisper\$vwRoom.class	13.10.2014 1:47	Файл "CLASS"	2 КБ
VideoWhisper\$webStatusTask.class	13.10.2014 1:47	Файл "CLASS"	1 КБ
VideoWhisper.class	13.10.2014 1:47	Файл "CLASS"	39 КБ

Рис. 25 Все Java классы

Теперь запустем ПО (находится по адресу <http://vs/>)

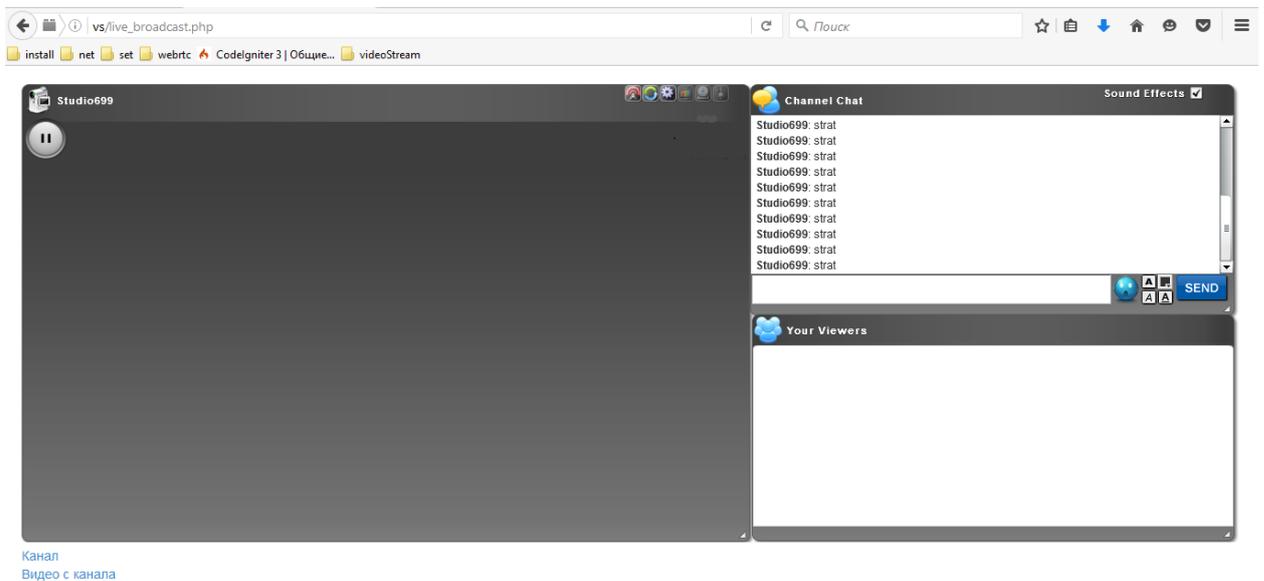


Рис. 26

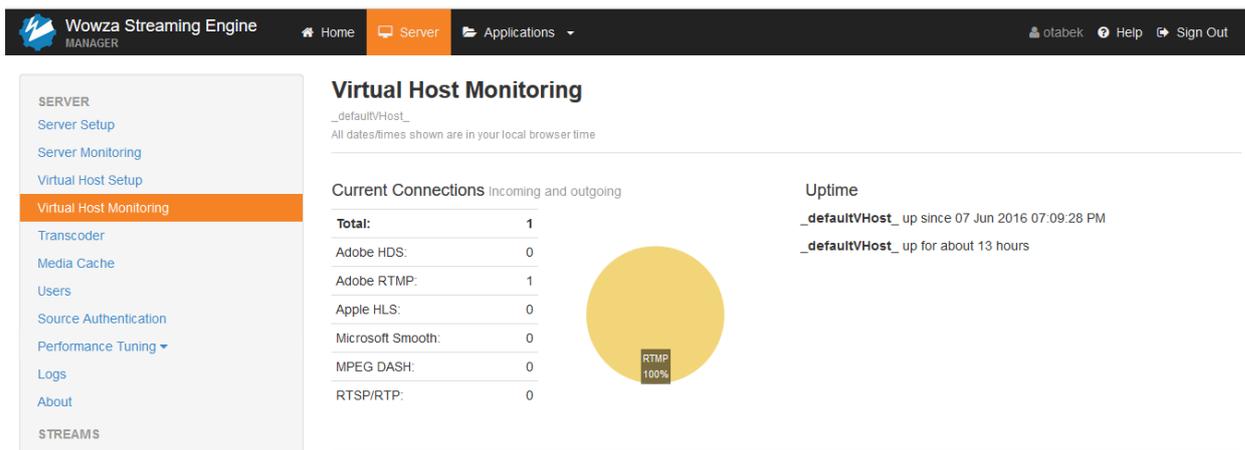


Рис. 26

С рис 26 видно что появилось новое соединение это означает что видео стриминг успешно работает. По адресу <http://vs/channel.php?n=Studio699> можно подключиться к видео стримингу и смотреть онлайн трансляцию.

## **Глава 4. БЕЗОПАСНОСТЬ ЖИЗНЕДЕЯТЕЛЬНОСТИ**

Специалист высшей квалификации должен знать основы дисциплины “Безопасность жизнедеятельности”. А именно, будущий специалист специальности АТПШ должен знать инженерные основы “Безопасности жизнедеятельности”.

Как известно, дисциплина “Безопасность жизнедеятельности” изучает опасности, угрожающие человеку, закономерности их проявления и способы, методы и средства защиты от них. При проектировании производственной среды и обеспечении безопасности труда будущий инженер должен учитывать человеческий фактор. Данная глава дипломной работы носит своей целью закрепить навыки проектирования производственной среды с учетом человеческого фактора.

Повышенное внимание к проблеме БЖД во всех средах обитания объясняется целым рядом факторов. Например, резким ростом вероятности несчастных случаев на производстве. Учет всех вредностей позволит инженеру правильно спроектировать производственную среду и обеспечить БДЖ всех работающих в этой среде.

В настоящее время не менее резко изменились условия труда. Во-первых, наблюдается повышение интенсивности воздействующих на человека, как уже было сказано выше, неблагоприятных санитарно-гигиенических факторов. Растет интенсивность акустических шумов, достигающая на ряде производств болевого порога. Механизированный ручной инструмент, ряд машин и механизмов создают высокие уровни вибраций. В ряде отраслей промышленности трудовая деятельность выполняется в помещениях с избытками явного тепла

Учет неблагоприятных факторов является необходимым условием обеспечения требуемой эффективности деятельности и сохранения здоровья работников. В условиях НТП постоянно растет перечень неблагоприятных физических и химических факторов внешней среды. Безопасный ранее, например, фоновый уровень проникающей ионизирующей радиации в усло-

виях современной технологии стал реальной причиной одного из опасных профессиональных заболеваний - лучевой болезни. Высокие мощности сверхвысокочастотного электромагнитного излучения современных радиолокационных установок, некоторых технических средств, базирующихся на использовании этого излучения, а также электромагнитных излучений от оргтехники оказываются опасными для здоровья человека, вызывая изменения в психическом и физиологическом состояниях.

## **5.1 Освещения компьютерных помещений**

В компьютерных залах должно быть естественное и искусственное освещение. Естественное освещение обеспечивается через оконные проемы с коэффициентом естественного освещения КЕО не ниже 1,2% в зонах с устойчивым снежным покровом и не ниже 1,5% на остальной территории. Световой поток из оконного проема должен падать на рабочее место оператора с левой стороны.

Искусственное освещение в помещениях эксплуатации компьютеров должно осуществляться системой общего равномерного освещения.

Освещенность на поверхности стола в зоне размещения документа должна быть 300-500 лк. Допускается установка светильников местного освещения для подсветки документов. Местное освещение не должно создавать бликов на поверхности экрана и увеличивать освещенность экрана более 300 лк. Прямую блескость от источников освещения следует ограничить. Яркость светящихся поверхностей (окна, светильники), находящихся в поле зрения, должна быть не более 200 кд/м<sup>2</sup>.

Отраженная блескость на рабочих поверхностях ограничивается за счет правильного выбора светильника и расположения рабочих мест по отношению к естественному источнику света. Яркость бликов на экране монитора не

должна превышать 40 кд/м<sup>2</sup>. Показатель ослепленности для источников общего искусственного освещения в помещениях должен быть не более 20, показатель дискомфорта в административно-общественных помещениях не более 40. Соотношение яркости между рабочими поверхностями не должно превышать 3:1 — 5:1, а между рабочими поверхностями и поверхностями стен и оборудования 10:1.

Для искусственного освещения помещений с персональными компьютерами следует применять светильники типа ЛПО36 с зеркализированными решетками, укомплектованные высокочастотными пускорегулирующими аппаратами. Допускается применять светильники прямого света, преимущественно отраженного света типа ЛПО13, ЛПО5, ЛСО4, ЛПО34, ЛПО31 с люминисцентными лампами типа ЛБ. Допускается применение светильников местного освещения с лампами накаливания. Светильники должны располагаться в виде сплошных или прерывистых линий сбоку от рабочих мест параллельно линии зрения пользователя при разном расположении компьютеров. При периметральном расположении — линии светильников должны располагаться локализованно над рабочим столом ближе к его переднему краю, обращенному к оператору. Защитный угол светильников должен быть не менее 40 градусов. Светильники местного освещения должны иметь непросвечивающийся отражатель с защитным углом не менее 40 градусов.

Для обеспечения нормативных значений освещенности в помещениях следует проводить чистку стекол оконных проемов и светильников не реже двух раз в год и проводить своевременную замену перегоревших ламп.

## **5.2 Принципы защиты населения в ЧС**

**Защита населения в чрезвычайных ситуациях** — одна из главных задач гражданской обороны. Объем и характер защитных мероприятий определяются особенностями отдельных районов и объектов, а также вероятной

обстановкой, которая может сложиться в результате химического, бактериологического (биологического) и других видов заражения. Защита населения при возникновении чрезвычайных ситуаций в условиях мирного и военного времени организуется и осуществляется в соответствии со следующими принципами:

1. Осуществление постоянного руководства мероприятиями по защите населения со стороны руководителей министерств, ведомств и объектов народного хозяйства.
2. Заблаговременное планирование мероприятий по защите населения и проведение их во всех городах, населенных пунктах и на всех объектах народного хозяйства страны.
3. Проведение дифференцирования с учетом политического, экономического и оборонного значения экономических районов, городов и объектов народного хозяйства.
4. Планирование и проведение мероприятий по защите населения во взаимодействии с вооруженными силами страны.
5. Планирование и осуществление мероприятий по защите населения в соответствии с планами экономического и социального развития республики, края, области, города, объекта народного хозяйства.

Под режимом защиты понимается применение средств и способов, максимально снижающих вероятность заражения, отравления либо облучения людей в зоне поражения.

**Способами защиты населения являются:**

1. Своевременное оповещение населения.
2. Мероприятия по противорадиационной и противохимической защите (ПРиПХЗ).
3. Укрытие людей в защитных сооружениях.

4. Использование средств индивидуальной защиты.
5. Проведение эвакуационных мероприятий (рассредоточение и эвакуация населения из городов в загородную зону).

Помимо этого организуется и проводится всеобщее обязательное обучение населения. Также осуществляется защита продовольствия, систем водоснабжения и водозабора от заражения радиоактивными, отравляющими веществами и бактериальными средствами; осуществляются радиационная, химическая и бактериологическая разведка, устанавливаются режимы защиты рабочих, служащих и производственной деятельности объектов, а также дозиметрический и лабораторный (химический и бактериологический) контроль. Планируются профилактические противопожарные, противоэпидемические и санитарно-гигиенические мероприятия, спасательные и неотложные аварийно-восстановительные работы (СНАВР) в очагах поражения, санитарная обработка людей, обеззараживание техники, одежды, обуви, территории и сооружений.

### **Своевременное оповещение населения**

Оно организуется по радио- и телевещанию. Чтобы привлечь внимание населения, используют сигналы транспортных средств, а также включают гудки на предприятиях.

Услышав сигналы «Внимание всем!», надо немедленно включить теле- и радиоприемники и ждать сообщения от местных органов власти или штаба ГО. Все дальнейшие действия определяются их указаниями.

При аварии на химически опасном объекте содержание информации может быть следующим: «Внимание! Говорит штаб ГО. Граждане! На мясокомбинате произошла авария с утечкой аммиака. Зараженное облако движется в направлении населенного пункта Знаменка. В связи с этим населению, проживающему по улицам Некрасова, Кузнечной, Заводской, следует нахо-

даться в своих квартирах и домах и провести дополнительную герметизацию помещений.

Населению, проживающему по улицам Заречной, Зеленой и Ямской, необходимо немедленно покинуть жилые дома и выйти к Лысой Горе. О полученной информации сообщите соседям. В дальнейшем нужно действовать в соответствии с указаниями штаба ГО».

После сигнала «Внимание всем!» может последовать и другая информация, например, о надвигающейся угрозе радиоактивного или бактериологического заражения. И в этих случаях будет передано краткое сообщение о порядке действий и правилах поведения.

### **Укрытие населения в защитных сооружениях**

Для защиты населения от ядерного, химического и бактериологического (биологического) воздействия существуют специальные сооружения. Они в зависимости от защитных свойств подразделяются на убежища и противорадиационные укрытия (ПРУ). За счет прочности ограждающих конструкций и перекрытий убежища обеспечивают наиболее надежную защиту людей от всех поражающих факторов — отравляющих веществ и бактериальных средств, от высоких температур и вредных газов в зонах пожаров, от обвалов, происшедших в результате взрыва.

Люди могут находиться в таких сооружениях длительное время. В убежищах предусмотрены надлежащие санитарно-гигиенические условия, обеспечивающие нормальную жизнедеятельность людей.

Наиболее распространены встроенные убежища, оборудованные в подвальных или полуподвальных помещениях производственных, общественных и жилых зданий.

Возможно также строительство убежищ в виде отдельных сооружений, полностью или частично углубленных в грунт. Под них могут быть приспособлены различные подземные переходы и галереи, метрополитены.

Убежища должны располагаться в местах наибольшего сосредоточения людей, для укрытия которых они предназначены.

Убежище состоит из основного помещения, предназначенного для размещения людей, и вспомогательных помещений — входов, фильтровентиляционной камеры, санитарного узла. В убежище большой вместимости могут быть выделены помещения под кладовую для продуктов питания и под медицинскую комнату.

В помещении, предназначенном для размещения людей, на одного человека предусматривается не менее 0,5 м<sup>2</sup> площади пола и 1,5 м<sup>3</sup> внутреннего объема. Большое помещение разбивается на отсеки вместимостью по 50 — 75 человек. В помещениях (отсеках) оборудуются двух- или трехъярусные нары — скамейки для сидения и полки для лежания; места для сидения устраиваются размером 0,45х0,45 м, а для лежания — 0,55 х 1,8 м.

Для того чтобы в помещения, где располагаются люди, не проникал зараженный воздух, они хорошо герметизируются. Убежище обычно имеет не менее двух входов, расположенных в противоположных сторонах. Кроме того, встроенное убежище должно иметь аварийный выход.

Входы в убежище в большинстве случаев оборудуются в виде двух шлюзовых камер (тамбуров), отделенных от основного помещения и имеющих плотно закрывающиеся двери. Снаружи устраивается прочная защитно-герметическая дверь, способная выдержать давление ударной волны ядерного взрыва. Вход может иметь предтамбур.

Аварийный выход представляет собой подземную галерею с выходом на незаваливаемую территорию через вертикальную шахту. (Незаваливаемой считается территория, расположенная от окружающих зданий на расстоянии, равном половине высоты ближайшего здания плюс 3 метра). Аварийный выход закрывается защитно-герметическими ставнями, дверями или другими устройствами, способными выдержать давление ударной волны.

В убежище оборудуются различные инженерные системы: электро-снабжения (трубы с электропроводкой окрашены в черный цвет), водоснабжения (трубы окрашены в зеленый цвет), отопления (трубы окрашены в коричневый цвет). В нем также устанавливаются радиотрансляционная точка (громкоговоритель) и телефон. В помещениях убежища предусмотрено наличие дозиметрических приборов и приборов химической разведки, защитной одежды, средств пожаротушения, аварийного запаса инструментов, средств аварийного освещения, запаса продовольствия и воды. Также должны быть паспорт и план убежища, правила содержания и табель его оснащения, журнал проверки состояния убежища.

### **Рассредоточение рабочих, служащих и эвакуация населения**

Рассредоточение и эвакуация населения — один из способов защиты от оружия массового поражения. Под рассредоточением понимают организованный вывоз из городов и других населенных пунктов и размещение в загородной зоне свободной от работы смены рабочих и служащих, продолжающих работу в военное время. Рабочие и служащие после расселения в загородной зоне посменно выезжают в город для работы на своих предприятиях, а по окончании работы возвращаются. Эвакуация представляет собой организованный вывоз или вывод из городов и других населенных пунктов и размещение в загородной зоне остального населения, а также вывоз или вывод населения из зон возможного затопления. В отличие от рассредоточен-

ных эвакуированные постоянно проживают в загородной зоне до особого распоряжения.

Загородная зона представляет собой территорию, расположенную за пределами зон возможных разрушений в городе. Каждому предприятию, учреждению, учебному заведению города, из которого планируются рассредоточение и эвакуация, в загородной зоне назначается район размещения населения, который в зависимости от количества рабочих, служащих и членов их семей может включать один или несколько расположенных рядом населенных пунктов.

Районы расселения рабочих и служащих в загородной зоне должны находиться на таком удалении от города, которое обеспечило бы их безопасность, а на переезд людей для работы в город и их возвращение в загородную зону для отдыха затрачивалось бы минимальное время.

Районы расселения целесообразно также располагать вблизи железнодорожных станций и автомобильно-дорожных магистралей.

Расселяют рабочих, служащих и членов их семей с соблюдением производственного принципа. При этом облегчаются отправка рабочих смен в город и обеспечение людей питанием, медицинское обслуживание.

Рабочих и служащих объекта, переносящего свою производственную деятельность в загородную зону, размещают вблизи имеющихся или вновь создаваемых производственных баз за районами размещения рабочих и служащих предприятий, продолжающих работать в городе. Эвакуированное население, не связанное с производством и не являющееся членами семей рассредоточиваемых рабочих и служащих, размещают в более отдаленных районах загородной зоны, а население, эвакуированное из зон возможного затопления, — в населенных пунктах, находящихся вблизи этих зон.

Эвакуация населения, не связанного с производством, осуществляется по территориальному принципу по месту жительства — через домоуправления, жилищно-эксплуатационные конторы и дирекции эксплуатации зданий (ДЭЗ). Дети обычно эвакуируются вместе с родителями, но не исключается возможность вывоза их со школами и детскими садами.

Непосредственно организацией и проведением эвакуационных мероприятий занимаются начальники штабов гражданской обороны объектов и эвакуационные комиссии, создаваемые в городах (городских районах). Рассредоточение и эвакуацию организуют и проводят после получения распоряжения об их проведении.

Для проведения рассредоточения и эвакуации используются все виды общественного транспорта (железнодорожный, автомобильный, водный), не занятого военными и неотложными производственными и хозяйственными перевозками, а также транспорт индивидуального пользования.

Для вывоза населения железнодорожным и водным транспортом используются пассажирские железнодорожные составы и пассажирские теплоходы, а при их недостатке — и товарные вагоны, грузовые суда и баржи. Предусматриваются более уплотненная загрузка вагонов и судов, а также увеличение длины поездов.

Вывоз рассредоточиваемого и эвакуируемого населения автотранспортом обычно планируется на небольшие расстояния. Для этого используются автобусы, приспособленные для перевозки людей грузовые автомобили. Определенная часть населения, подлежащего эвакуации, может выводиться пешим ходом. Вывод населения организуется колоннами по дорогам, не занятым другими перевозками или по обозначенным маршрутам и колонным путям.

Для организованного движения пешеходных колонн разрабатывают схему, на которой указаны состав колонны, маршрут движения, исходный пункт, пункты регулирования движения и время их прохождения; районы и продолжительность привалов; медицинские пункты и пункты обогрева; промежуточный пункт эвакуации (ППЭ), порядок и сроки вывода (вывоза) колонны из этого пункта в район постоянного размещения; сигналы управления и оповещения.

Часто используют комбинированный способ, который позволяет провести эвакуацию в кратчайшие сроки. Сущность его состоит в том, что массовый вывод населения из городов пешим ходом сочетается с вывозом всеми видами имеющегося транспорта. Этот способ является основным.

Все работы по организации и проведению рассредоточения и эвакуации населения осуществляются в соответствии с планом и указаниями начальника ГО объекта. Для руководства рассредоточением и эвакуацией населения на объекте создается эвакуационная комиссия, а на крупных объектах кроме того могут создаваться сборные эвакуационные пункты (СЭП). Приказом начальника ГО объекта создается администрация СЭП. Председателем объектовой эвакуационной комиссии назначается один из заместителей руководителя объекта.

Рассредоточение и эвакуация населения проводятся через сборные эвакуационные пункты, которые обычно организуются в школах, клубах и других общественных зданиях. Предназначаются СЭП для сбора, регистрации и отправки населения, эвакуируемого транспортом, на станциях, пристанях и других пунктах посадки, а эвакуируемого пешим ходом (пешими колоннами) — на исходных пунктах пешего движения. СЭП организуют обычно вблизи железнодорожных станций, платформ, портов, пристаней, а для населения, выводимого пешим ходом, — вблизи маршрутов вывода в назначенные районы.

На территории СЭП и вблизи него должны быть подготовлены убежища и укрытия из расчета размещения в них людей, которые могут одновременно находиться на пункте. Каждому СЭП присваивают номер; к нему приписывают ближайшие объекты народного хозяйства, а также ДЭЗ, ЖЭК, население которых будет эвакуироваться через данный СЭП.

Получив распоряжение о проведении рассредоточения и эвакуации, начальник ГО объекта организует выполнение эвакуационных мероприятий в соответствии с планом и распоряжением старшего начальника.

О начале эвакуации население оповещается через предприятия, учреждения, учебные заведения, ДЭЗ, милицию, а также радиотрансляционную сеть и местное телевидение. Получив извещение о начале рассредоточения и эвакуации, люди должны немедленно подготовить и взять с собой документы, деньги, необходимые вещи и запас продуктов и явиться на сборный эвакуационный пункт в строго определенное время.

В случае если рабочих и служащих разместить вместе с семьями не представляется возможным, членов их семей эвакуируют отдельно в более отдаленные районы по направлению рассредоточения (эвакуации); время их явки на сборный эвакуационный пункт будет установлено особо.

Прибывших на сборный эвакуационный пункт людей регистрируют, группируют по вагонам железнодорожного эшелона или по автомашинам (судам) и в назначенное время выводят к пунктам посадки на транспорт. Представители объекта совместно с администрацией станции (пункта, пристани) осуществляют посадку людей в вагоны, автомашины, на суда и поддерживают установленный порядок.

В каждом железнодорожном эшелоне (на судне) назначается начальник эшелона (судна), в автомобильной колонне — старший автоколонны, в каждом железнодорожном вагоне — старший вагона. Начальники железнодо-

рожных эшелонов и старшие автоколонн должны принимать все меры к тому, чтобы выдержать график движения транспорта и прибыть на соответствующую станцию (пункт) в загородной зоне в установленное время.

Вывод населения объекта пешим ходом осуществляется по заранее установленным маршрутам, рассчитанным, как правило, на один суточный переход, совершаемый за 10 — 12 часов движения. Численность пеших колонн может колебаться от 500 до 1000 человек. Для удобства управления колонну следует разбивать на группы по 50 — 100 человек, а во главе групп ставить старших. Старшие групп обязаны проверять численность личного состава, не допускать в группы посторонних лиц и следить, чтобы не было отстающих. Скорость движения колонн на маршруте следует выдерживать в пределах 4 — 5 км/час, а дистанцию между колоннами соблюдать до 500 метров.

При совершении марша через каждые 1 — 1,5 часа движения делают малые привалы продолжительностью 10 — 15 минут, а в начале второй половины суточного перехода устраивают большой привал на 1 — 2 часа, как правило, за пределами зоны возможных разрушений. На малых привалах людям оказывают необходимую медицинскую помощь, проверяют состав колонн (групп) оказывают помощь отстающим, предоставляют людям кратковременный отдых. За время привала отставшие колонны подтягиваются. На большом привале кроме этого организуют прием горячей пищи.

При проведении рассредоточения и эвакуации обеспечивается обслуживание. С этой целью на СЭП, станциях (пристанях, в портах) посадки развертывают медицинские пункты. На каждый поезд (судно) назначают одного или двух медицинских работников, а в состав пешей колонны включают двух медицинских работников и двух сандружинниц.

Не допускается употребление загрязненных продуктов питания и воды соблюдаются и другие меры предосторожности.

Эвакуация производится из тех районов, где пребывание населения может привести к заражению выше допустимых пределов и где нельзя обеспечить его защиту другими способами. Решение об эвакуации принимается начальником ГО области (края, республики).

Эвакуация проводится после тщательной подготовки людей, транспорта, изучения маршрутов движения с учетом радиационной обстановки. Население заранее предупреждается о времени и порядке эвакуации, транспорт подается к местам нахождения населения (к подъездам домов), погрузка и перевозка людей производятся на крытых транспортных средствах, в короткие сроки, по маршрутам с наименьшими уровнями радиации. Во время передвижения ведутся радиационная разведка и дозиметрический контроль. Население доставляется до границы зоны загрязнения, а затем пересаживается на незагрязненный РВ транспорт и доставляется в места размещения.

При выезде на незагрязненную территорию производится контроль зараженности людей и выводимого транспорта. При необходимости производится санитарная обработка людей, дезактивация одежды, имущества и транспорта.

В зонах загрязнения проводятся мероприятия по дезактивации территории, сооружений, оборудования, техники и других объектов, ведется контроль загрязненности сельскохозяйственной продукции. Организуется охрана зданий и имущества.

## **ЗАКЛЮЧЕНИЕ**

В результате проделанной работы были получены следующие результаты:

Была спроектирована и реализована архитектура кроссплатформенной децентрализованной сервис-ориентированной вычислительной среды на основе SaaS. Конкретно, на языке программирования PHP было написано SaaS приложения, реализованы передачи видео потоков в реальном времени.

Были предоставлены возможности решения в «облаке» задач, связанных с потоковой обработкой данных, проведена демонстрация данной возможности на примере обработки «живых» мультимедийных потоков.

Была спроектирована и реализована концепция транспортов и транспортных каналов, позволившая поддержать возможности общения по произвольным протоколам передачи данных. Были представлены примеры сервисов, работающих по протоколам TCP, HTTP, RTC и UDP.

Для обеспечения возможности активного взаимодействия сервисов, в том числе сервисов различных видов, решающих разные задачи, была спроектирована и реализована система передача произвольных сериализованных данных между сервисами, что привело к архитектуре, в которой различные сервисы совместно способствуют решению поставленных перед ними задач.

Полученная система была продемонстрирована в качестве веб-сервиса для реализации видео стриминга, а также сервиса видеоконференций.

## **СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ**

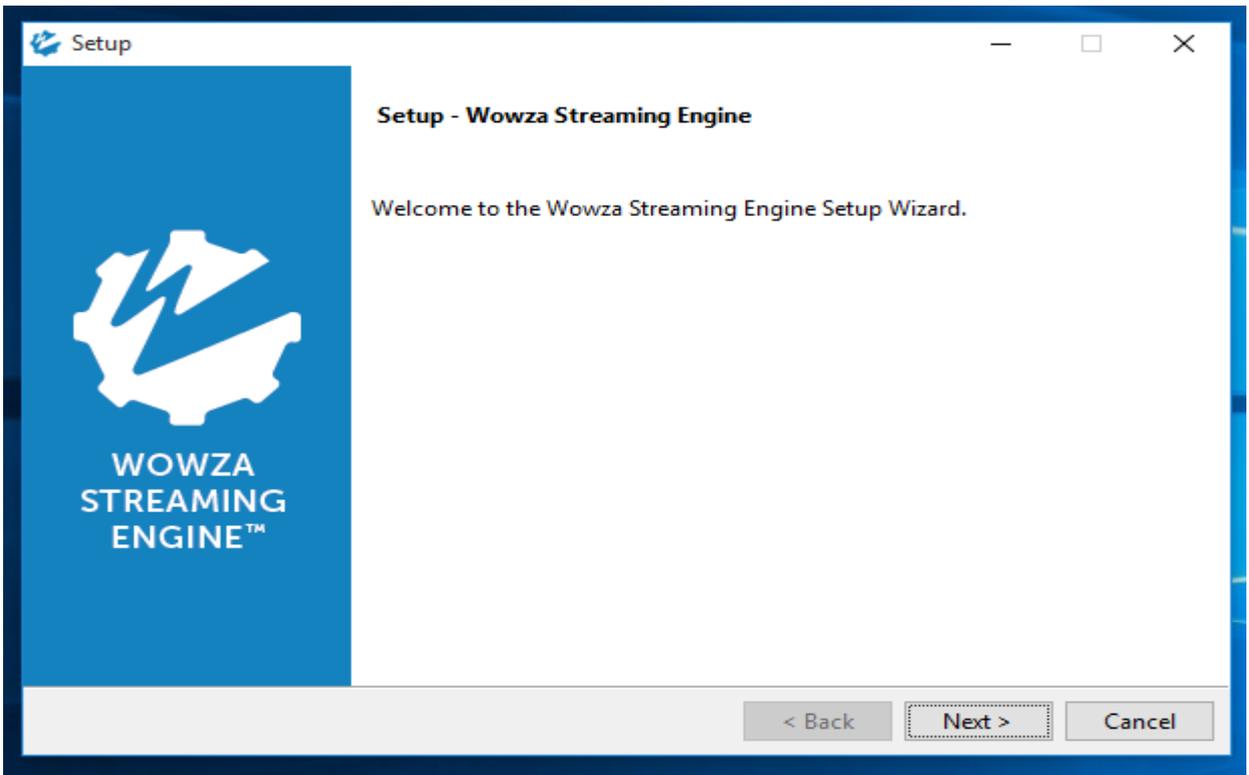
1. Доклад Президента Республики Узбекистан Ислама Каримова на заседании Кабинета Министров, посвященном итогам социально-экономического развития в 2015 году и важнейшим приоритетным направлениям экономической программы на 2016 год.
2. И.А. Каримов «Мировой финансово-экономический кризис, пути и меры по его преодолению в условиях Узбекистана», Ташкент, Узбекистон, 2009.
3. Ўзбекистон Республикаси қонун ҳужжатлари тўплами, 2012 й., 5-сон, 47-модда. [<http://lex.uz>]
4. А.Кудратов, Т.Ганиев и др. «Безопасность жизнедеятельности». Ташкент, Алокачи, 2005.
5. Харатишвили, Д. Utility-компьютинг и "облачные" вычисления // Компьютерпресс, 2009, N 9, С. 14-19.
6. Cloud Application Architectures [Джордж Риз](#) 2011
7. «SQL. Сборник рецептов», Молинаро
8. «jQuery. Подробное руководство по продвинутому JavaScript», Бибо Б., Кац И
9. «HTML и XHTML. Подробное руководство», Муссиано, Кеннеди
10. «Мастерская CSS. Профессиональное применение web-стандартов», Энди Бадд
11. «Совершенный код. Практическое руководство по разработке программного обеспечения», Макконнелл С.
12. «Разгони свой сайт. Методы клиентской оптимизации веб-страниц», Николай Мацевский
13. «Базы данных. Проектирование, реализация и сопровождение. Теория и практика», Томас Коннолли, Каролин Бегг
14. Рекс Блэк - Ключевые процессы тестирования 2014
15. Максим Кузнецов, Игорь Симдянов, Сергей Гольшев. РНР 5 на примерах. Серия: На примерах. Издательство: БХВ-Петербург, 2005 г., 576 стр.

16. Джо Мараско - The Software Development Edge: Essays on Managing Successful Projects / IT-проекты. Фронтальные очерки 2010

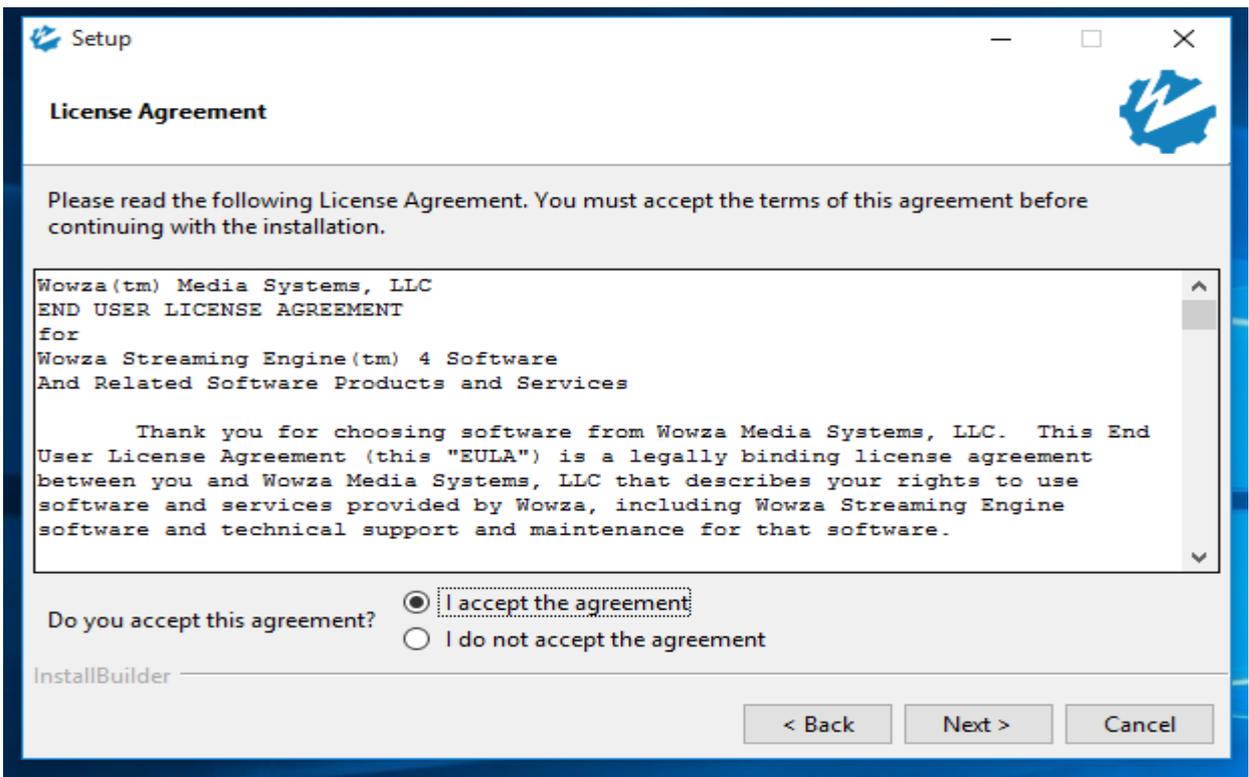
### **Ссылки на использованные источники в Интернете.**

1. Adobe AIR Platform <http://www.adobe.com/ru/products/air.html>
2. Adobe Flash Media Server <http://www.adobe.com/ru/products/flashmediaserver/>
3. Adobe Flash Platform <http://www.adobe.com/products/flash.html>
4. Adobe Flash Video (F4V/FLV) File Format Specification  
[http://download.macromedia.com/f4v/video\\_file\\_format\\_spec\\_v10\\_1.pdf](http://download.macromedia.com/f4v/video_file_format_spec_v10_1.pdf)
5. Amazon Elastic Compute Cloud (Amazon EC2) <http://aws.amazon.com/ec2/>
6. Amazon Simple Storage Service (Amazon S3) <http://aws.amazon.com/s3/>
7. Apache Subversion <http://subversion.apache.org/>
8. Dropbox File Hosting <https://www.dropbox.com/>
9. FFmpeg Libraries <http://www.ffmpeg.org/>
10. Google App Engine <http://appspot.com/>
11. Google Drive <https://drive.google.com/start>
12. Google: Cluster Computing and MapReduce <http://code.google.com/intl/ru-RU/edu/submissions/mapreduce-minilecture/listing.html>
13. Microsoft .NET Framework <http://www.microsoft.com/net>
15. Motion JPEG (MJPEG) [http://en.wikipedia.org/wiki/Motion\\_JPEG](http://en.wikipedia.org/wiki/Motion_JPEG)
16. MPEG Audio Layer III (MP3) Format <http://en.wikipedia.org/wiki/MP3>
19. Platform as a Service (PaaS)  
[http://en.wikipedia.org/wiki/Platform\\_as\\_a\\_service](http://en.wikipedia.org/wiki/Platform_as_a_service)
20. Red5 Media Server <http://www.red5.org/>

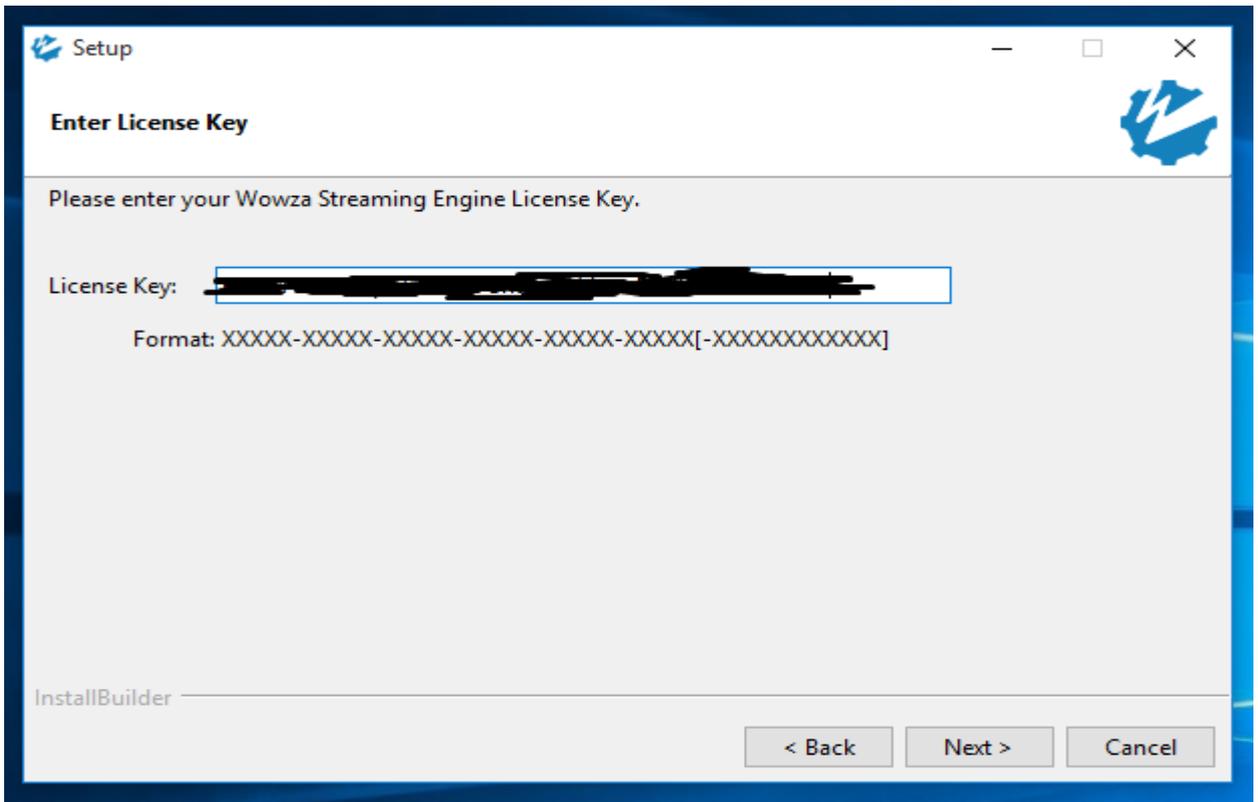
ПРИЛОЖЕНИЕ. 1 Установка Wowza Streaming Engine



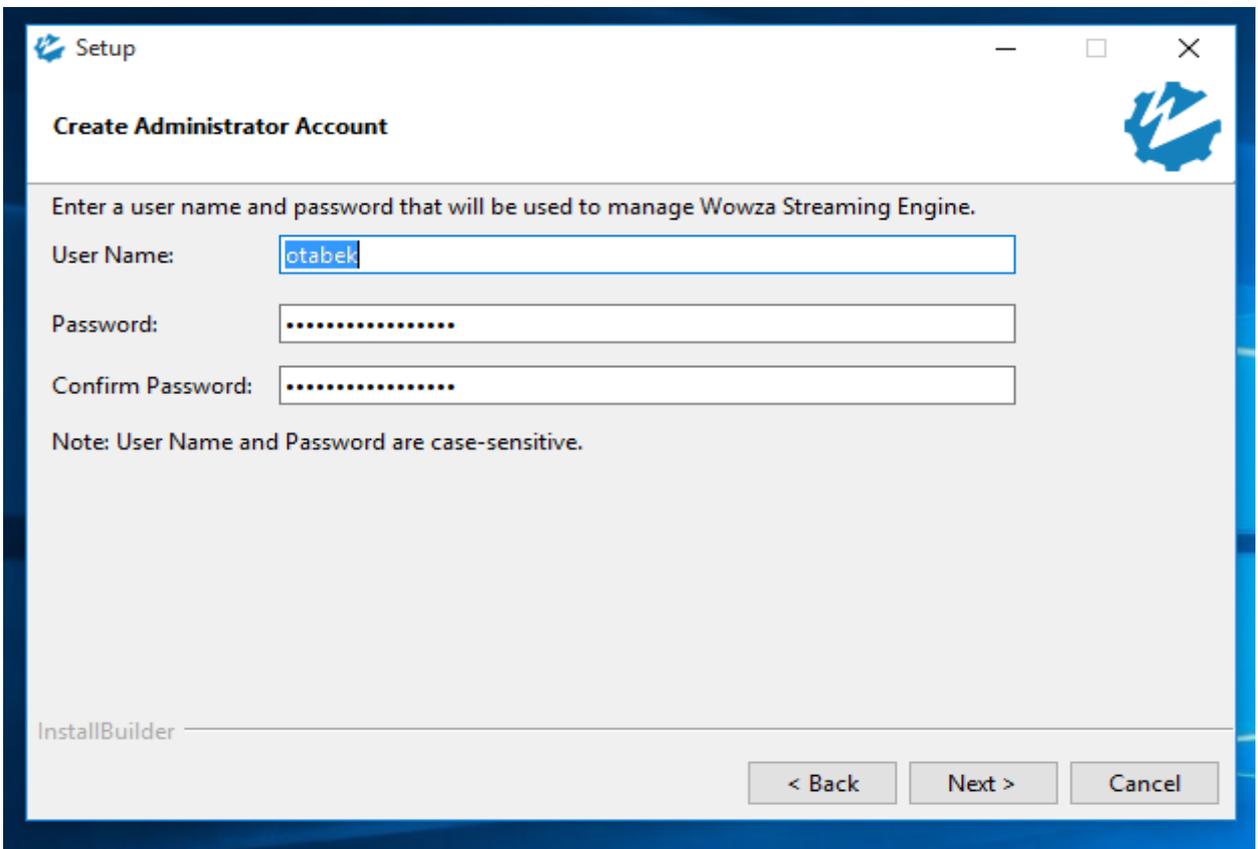
*Рис. 1 Запустил инсталлятор*



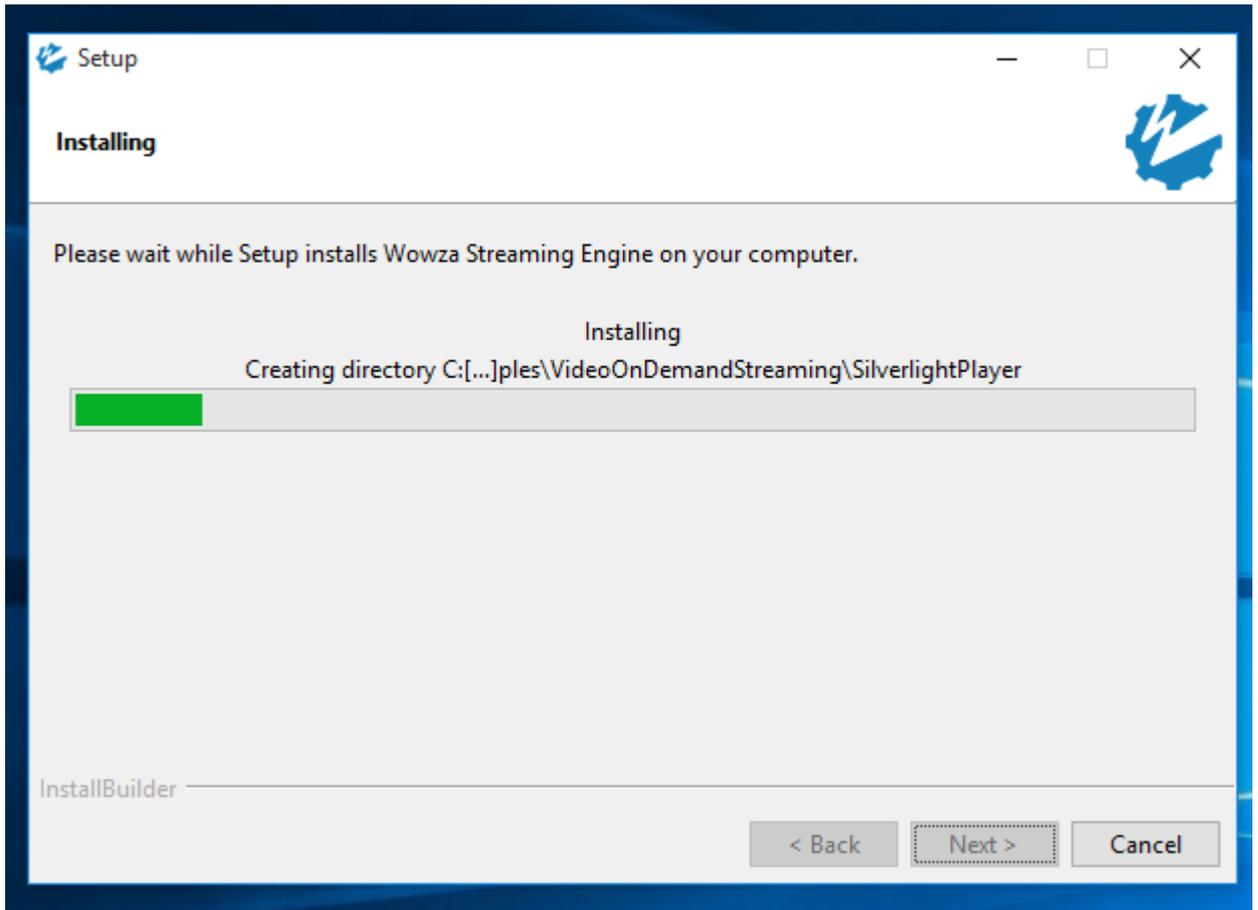
*Рис. 2 Принимаем пользовательские соглашения*



*Рис. 3 Вводим ключ продукта*



*Рис.4 Вводим данные аккаунта*



*Рис. 5 После установки запустим ПО*

## ПРИЛОЖЕНИЕ. 2 Обзор программы

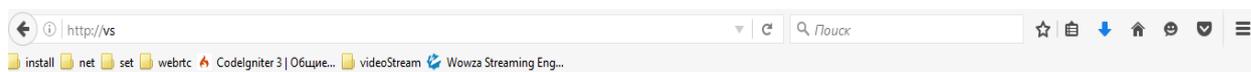


Рис. 1 Создания видео канала

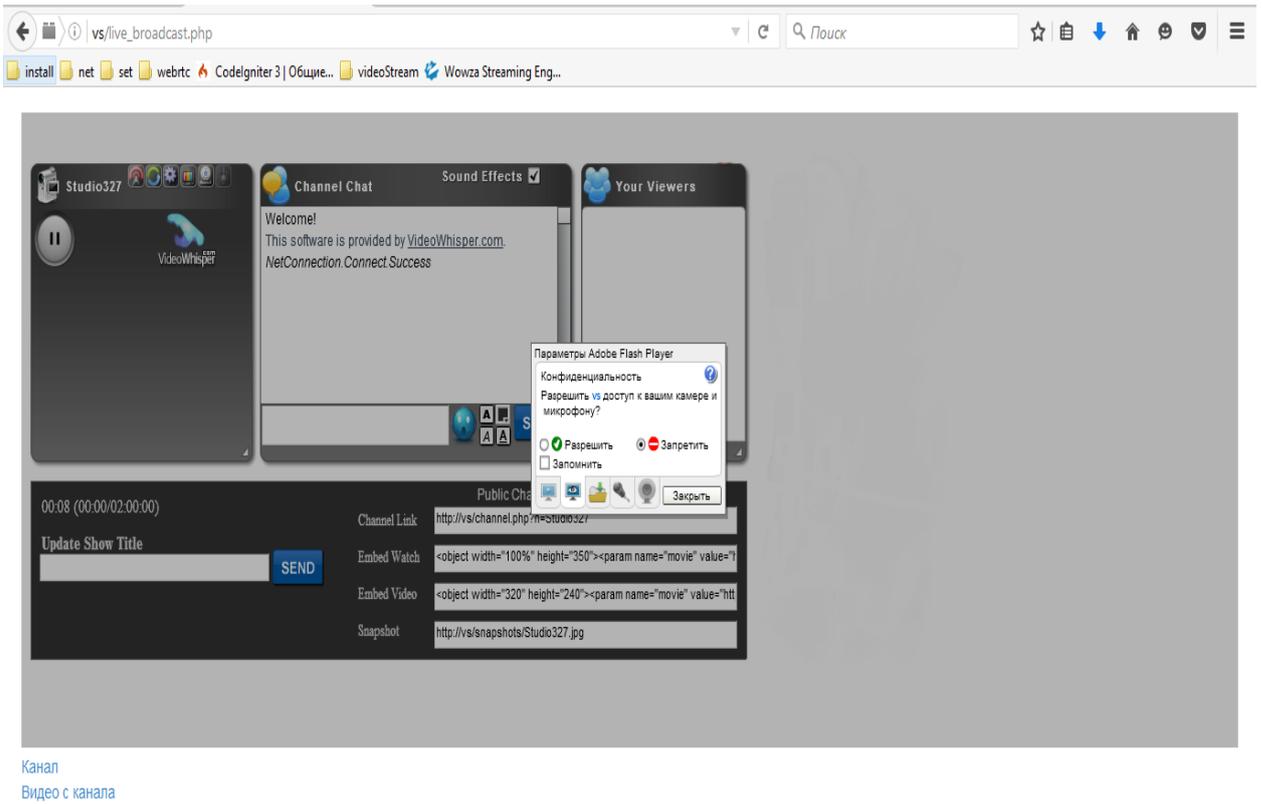


Рис.2

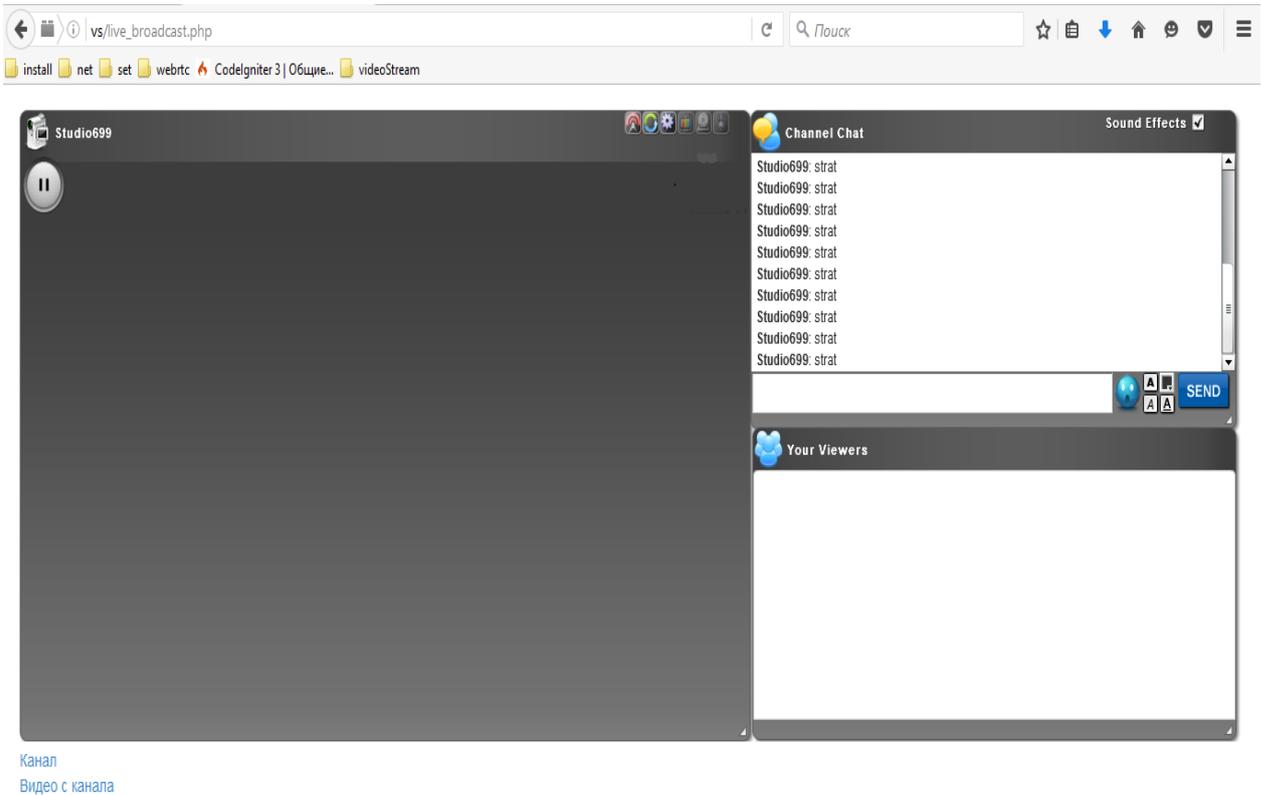
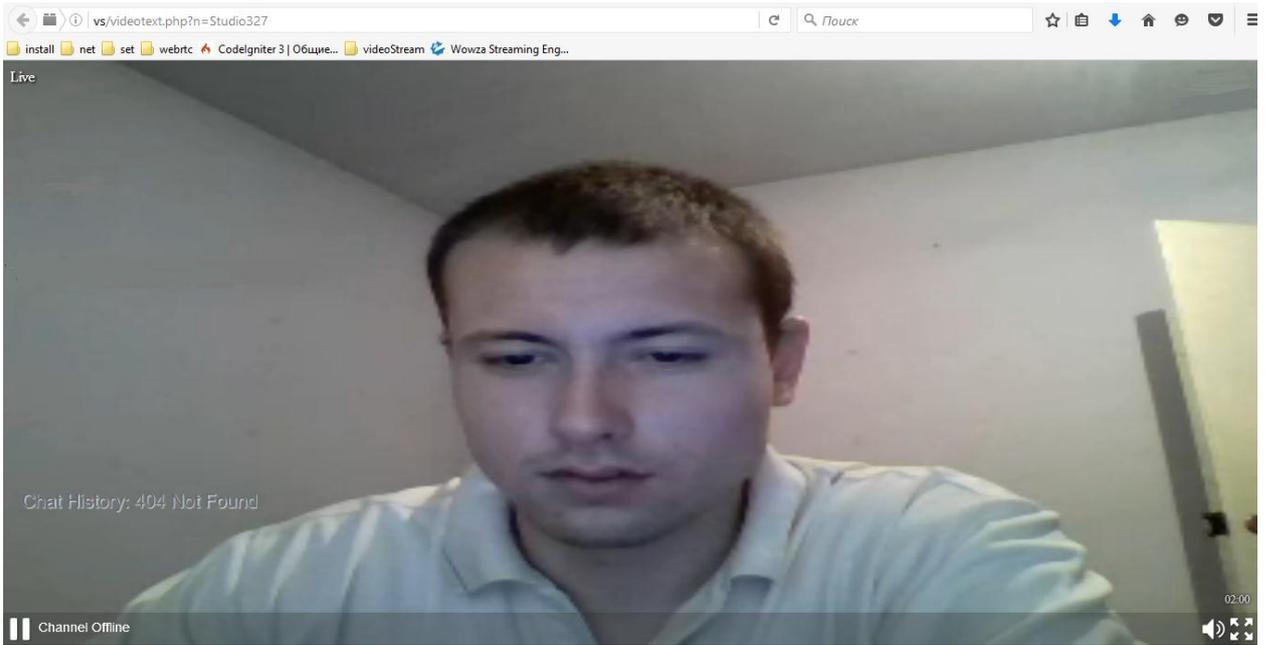
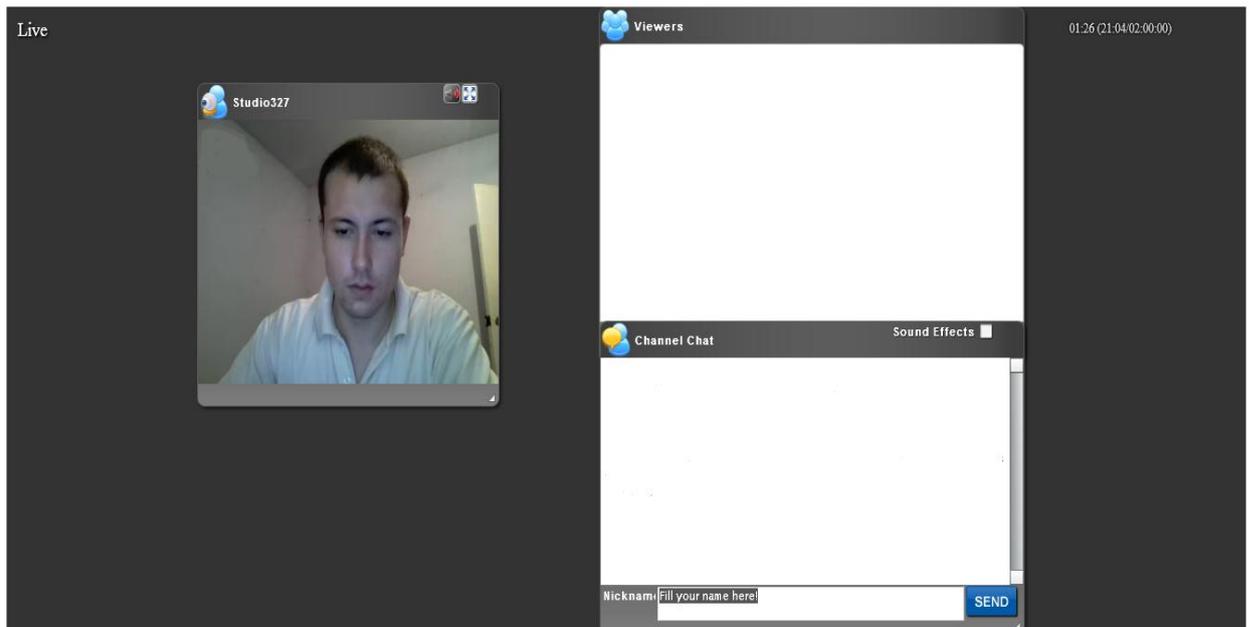


Рис. 3



*Рис. 5 Live Видео стриминг*



*Рис.6*