

## **Тема 1.6. Виды клиентского программного обеспечения**

### **1. Клиентские приложения**

Клиентской называется часть приложения, с которой напрямую взаимодействует конечный пользователь. Это может быть либо приобретенное компанией серийное коммерческое программное обеспечение, либо прикладная программа, разработанная внутри компании с помощью инструментальных средств третьих фирм.

Наличие клиентских приложений способствует упрощению работы конечного пользователя базы данных. От него скрыты объекты базы данных, содержащие реальную информацию, программный код, а также происходящие внутри нее события. Современные технологии делают работу с прикладными программами для конечного пользователя более интуитивной: они дают ему возможность сосредоточиться на выполнении своих прямых обязанностей и, тем самым, способствуют повышению производительности его труда.

В целом, клиентские приложения обеспечивают выполнение следующих стандартных функций:

- обеспечение сеанса связи с сервером, при этом одно приложение может одновременно поддерживать несколько соединений;
- формирование запросов и передача их серверу;
- получение результатов выполнения запросов и их отображение при помощи стандартных интерфейсных элементов;
- обеспечение редактирования, дополнительной обработки, преобразования данных и т. д.

### **2. Стратегия клиентских приложений**

Доступ с любого компьютера предприятия к информации, поступающей от производственного технологического процесса, от любой подсистемы становится необходимостью. Одним из основных компонентов АСУТП (автоматизированная система управления технологическим процессом), способным поставлять технологическую информацию, является приложение SCADA (Supervisory Control and Data Acquisition). А различного типа клиентские приложения могут предоставлять соответствующие производственному процессу в огромном объеме данные в приемлемом для пользователя виде

Традиционно SCADA-системы выполняют следующие функции:

- сбор данных с контроллерного уровня, в том числе на основе стандартных протоколов DDE, OPC;
- отображение данных с использованием графических анимированных объектов (простых и сложных);
- обработка данных с использованием встроенных языков программирования.
- алармирование данных;
- архивирование, хранение данных.

Самым простым и распространенным клиентским приложением являются клиенты в локальной сети.

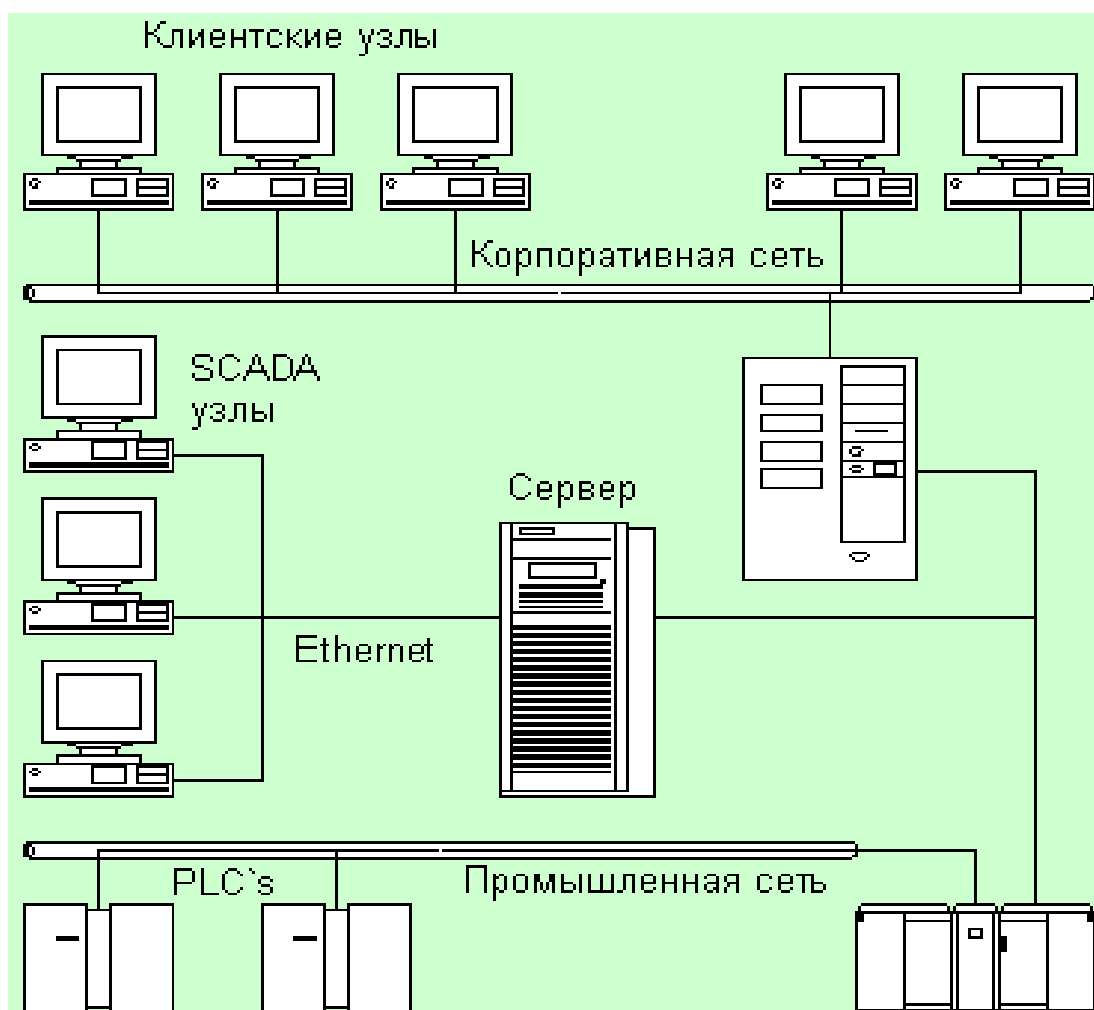


Рис.1

Клиент-серверная организация SCADA-систем предполагает наряду с серверными конфигурациями применение клиентских компонентов двух типов: с возможностью передачи управляющих воздействий с клиентского приложения и чисто мониторинговые приложения. Такие клиентские компоненты SCADA-систем традиционно объединяются с серверными приложениями с помощью протоколов локальных сетей (TCP/IP, NetBEUI). Однако развивающиеся технологии Internet/Intranet не оставили безучастными разработчиков SCADA-систем, баз данных реального времени и т.п. программных продуктов и привели к появлению следующих типов клиентских приложений:

- клиентские приложения в режиме сервер/терминал;
- бедные и богатые Internet/Intranet-клиенты.

Основой рассматриваемых решений для клиентских приложений являются новые технологии Microsoft, реализованные в структуре Windows DNA (Distributed Internet Architecture). Поэтому знакомство с ними предлагается начать с краткого изложения особенностей этой структуры.

### 3. Структура Windows DNA

Структура Windows DNA это, в первую очередь, реализация трехуровневой модели приложения, включающей следующие уровни:

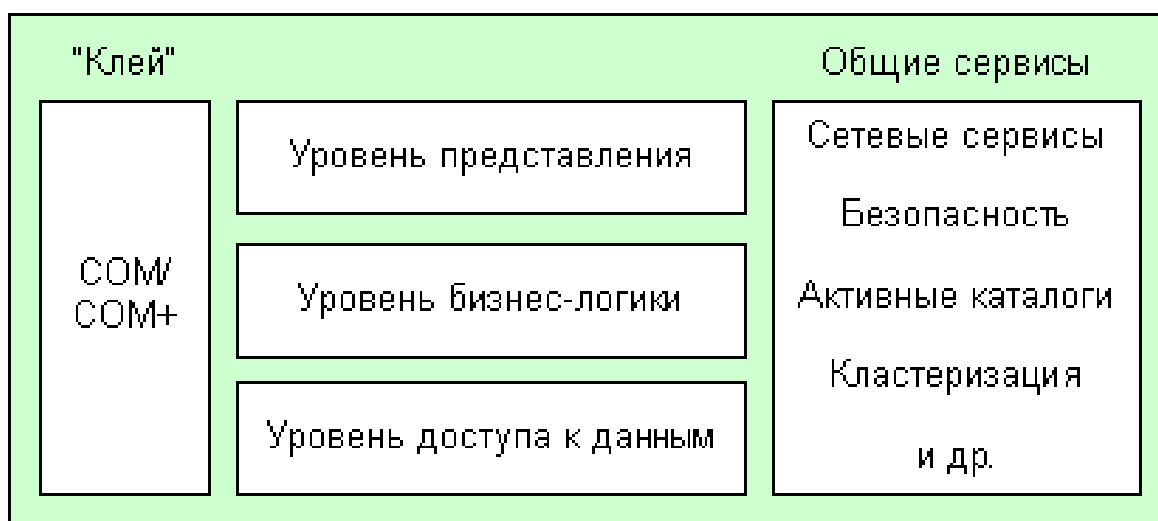


Рис.2

- уровень представления;
- уровень бизнес-логики;
- уровень доступа к данным.

Кроме технологий, привязанных к уровням, применяются технологии, представляющие общие сервисы, а также склеивающие технологии. В программном обеспечении Microsoft роль склеивающих технологий играют архитектуры COM и COM+. Архитектура компонентных объектов COM (Component Object Model) это объектно-ориентированная технология. Приложение с компонентной организацией конструируется из COM-объектов, используя готовые наборы этих объектов.

Слои Windows DNA. Технологии Microsoft и относящийся к ним инст-

рументарий предназначены для разработки и реализации трехуровневых приложений.

### **3.1 Уровень представления**

На этом уровне есть два обширных вида клиентов, называемых бедными (thin) и богатыми (rich) клиентами. В отличие от толстого (fat) клиента богатый клиент в большей степени ссылается на используемые при создании пользовательского интерфейса технологии, чем на то, какое количество кода выполняется на стороне клиента. Богатые клиенты похожи на обычные приложения Win32, но являются клиентской частью трехуровневого приложения.

Бедные клиенты не одинаково бедны. Примером бедного клиента служит давно известный терминал. Компания Microsoft предложила технологию Windows Terminal Server, в которой приложение Windows работает на центральном сервере и передает графический интерфейс пользователю-клиенту. При этом требуется дорогостоящий сервер и широкая полоса пропускания между клиентом и сервером. Чаще всего понятие бедный клиент обозначает приложение, работающее на Web-сервере и передающее пользовательский интерфейс с помощью HTML-страниц на Web-браузер.

Далее появилась идея обогащения Web-приложений различными компонентами, которые могут использоваться браузером, управляющие элементы ActiveX, апплеты Java и т.д. Различной оснащенности бедные клиенты предлагаются и компаниями-поставщиками SCADA-систем.

### **3.2 Уровень бизнес-логики**

Этому уровню свойственны три сервиса: сервисы компонентов (COM), сервер очередей сообщений Microsoft Message Queue (MSMQ) и сервер Internet Information Server (IIS). Сервер IIS это полнофункциональный Web-сервер Microsoft, интегрированный в Windows 2000/2008 Server. Сервер IIS

является сервером приложений, поддерживающим бедных клиентов, которые подключаются к нему через протокол HTTP.

Microsoft Transaction Server и COM+. Фундаментальной структурной концепцией, которая обеспечивает разработку сложных многопользовательских приложений для работы с данными, является транзакция. Главное свойство транзакции атомарность. Именно концепция транзакции обеспечивает выполнение ряда операций получения данных из разных СУБД и позволяет рассматривать их как единую операцию.

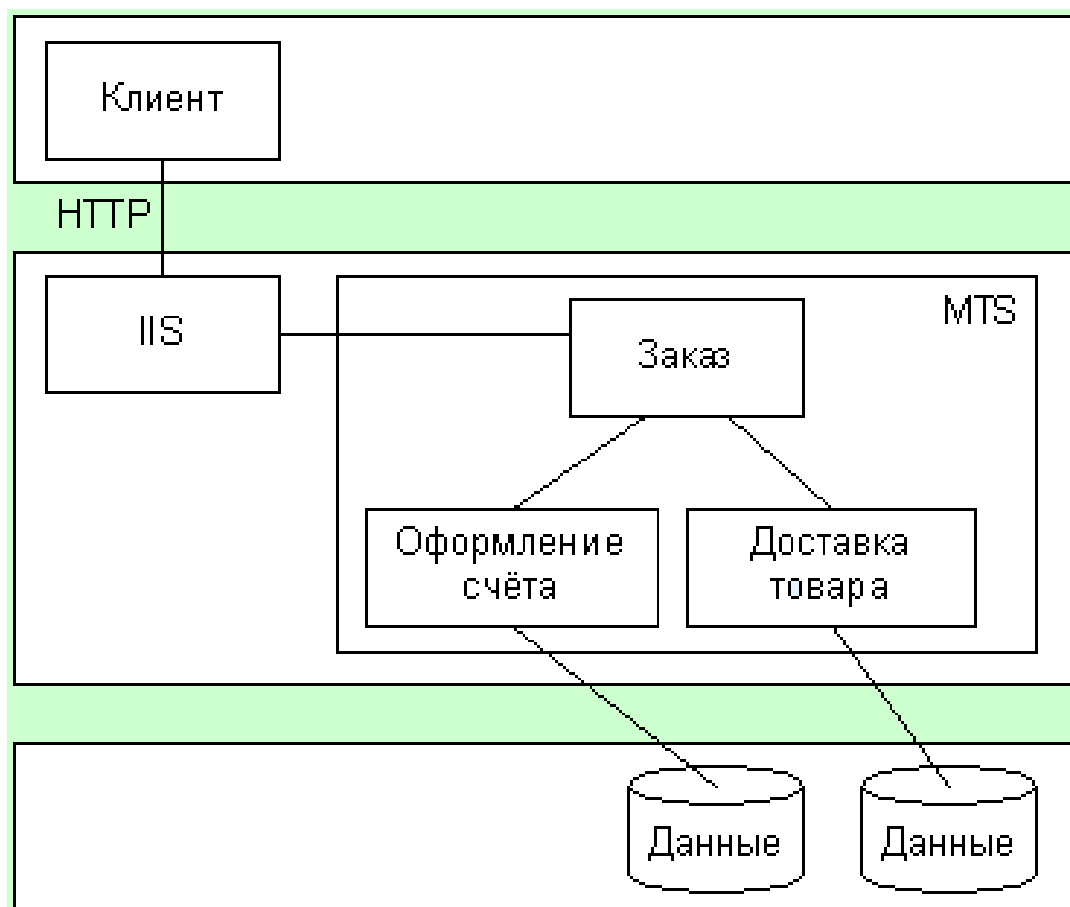


Рис.3

Message Queue асинхронная однонаправленная связь, ориентированная на сообщения. Протоколы DCOM и HTTP являются синхронными. Они возвращают результат, и до получения ответа от сервера работа клиента блокируется. Асинхронность MSMQ означает, что вызов сервиса осуществляется помещением сообщения в очередь. При этом возврат управления клиенту происходит немедленно (и возврат свидетельствует о постановке в очередь) и

клиент продолжает работать (нет блокировки).

### **3.3 Уровень доступа к данным**

Фундаментальной технологией доступа к данным является OLE DB гибкий низкоуровневый интерфейс COM.

Структура Windows DNA, особенно уровня представления данных, является основой клиентских приложений, предлагаемых поставщиками SCADA-систем.

## **4. Новая реализация клиентского приложения в режиме сервер/терминал**

С появлением Windows NT/2000/2008 Terminal Services вновь стала возможна организация клиентских сессий, когда каждый клиент функционирует независимо от других. В этом случае каждый пользователь получает свой ресурс: память, время центрального процессора, доступ к дискам сервера и приложениям. Когда клиент запускается, терминальный сервер его регистрирует, предоставляя доступ к ресурсам сервера. ОС Windows создает также виртуальный дисплей, который затем передается клиенту и отображается на локальном мониторе. Операции ввода, активизируемые клиентом с клавиатуры и мыши также обслуживаются сервером. Добавление новых клиентов сводится к встраиванию нового терминала.

Для организации взаимодействия между сервером и клиентом используются стандартные протоколы Microsoft RDP (Remote Desktop Protocol) и Citrix ICA (Independent Computing Architecture), что допускает реализацию клиентов в виде супер-тонких бездисковых рабочих станций на платформах Linux/CE, от Windows 3.11.95.98 до рабочих станций Windows NT или 2000/2008.

Используя новые архитектурные возможности, компании-разработчики SCADA-систем имеют возможность предложить терминальные сервисы,

поддерживающие выполнение SCADA-приложений в режиме сессии. Так компания Wonderware уже предоставляет Terminal Services (терминальные сервисы) для SCADA-системы InTouch версии 7.1, что позволяет установить исполняющую систему InTouch один раз на центральном сервере и затем запускать InTouch-приложения много раз. Клиентские узлы необходимо подключать в режиме терминальной сессии InTouch. Бедный клиент может быть в этом случае терминалом персонального компьютера или встроенным терминальным устройством с вышеперечисленными операционными системами.

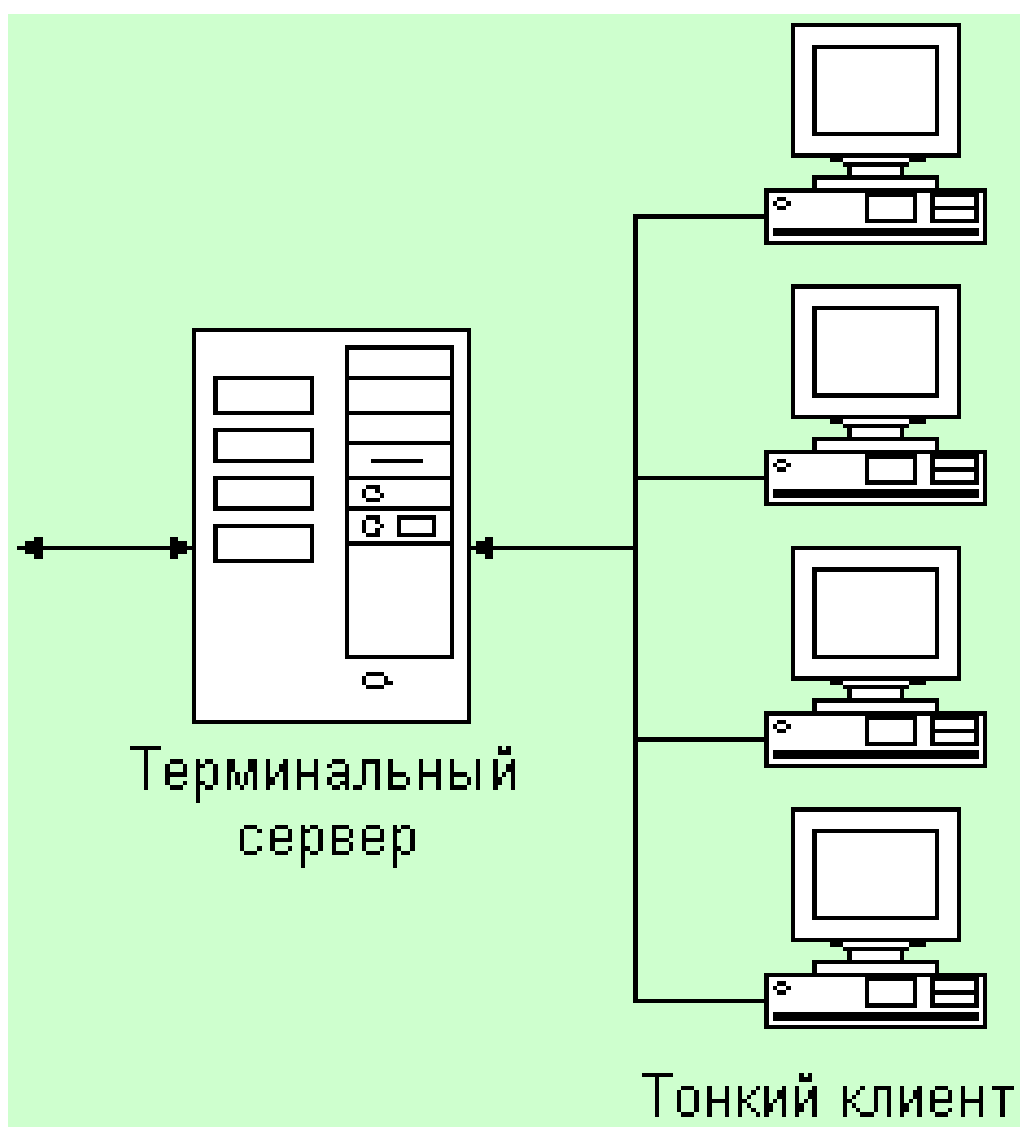


Рис.4

Терминальные пользователи имеют доступ к данным, графическим мнемосхемам с возможностью обмена информации в реальном времени без необходимости установки InTouch на локальном клиентском компьютере.



Применение терминал-серверной модели позволяет создавать более экономичные решения за счет того, что приложение устанавливается и поддерживается инженерами только на сервере, при этом можно использовать клиентские узлы на различных платформах. Следует заметить, что на клиентских узлах может просматривать как одно и то же приложение, так и разные приложения.

## 5. Бедные и богатые Internet/Intranet-клиенты

В Internet/Intranet решениях в обмене данными, кроме технологического сервера, как поставщика данных, и клиента, как получателя информации, задействован Web-сервер (рис. 5).

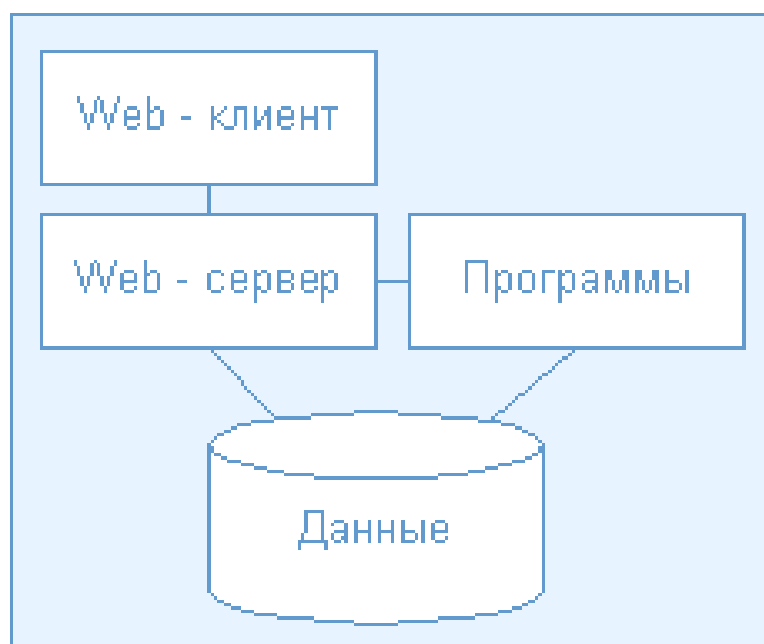


Рис.5 Клиенты и серверы Web

Информация на сервере хранится в виде страниц, на которых кроме текста могут находиться разные объекты: графические изображения, аудио- и видеоролики, формы для ввода данных, интерактивные приложения и т.д.

Страницы сервера WWW могут содержать не только статическую, но и динамическую информацию. Страница может содержать формы для выполнения запросов к базе данных. Результат такого запроса будет динамически сформирован в виде страницы, которая появится на экране пользователя. Сервер WWW может решать любую задачу, принимая любые данные от удаленного пользователя, обрабатывая их и передавая обратно. Для обработки на сервере WWW запросов, поступающих от клиентских приложений SCADA и требующих получения данных из БДРВ или других источников информации реального времени (РВ), разрабатывается специальное серверное расширение, которое, с одной стороны, получает и обрабатывает дина-

мические запросы от Web-клиентов, а с другой, обеспечивает взаимодействие с Microsoft Internet-серверами. Взаимодействие между Web-сервером и клиентами осуществляется на основе протокола HTTP (HyperText Transfer Protocol протокол передачи гипертекста.). Так компанией Wonderware предлагается FactorySuite (FS) Web-сервер, который обеспечивает динамическими данными клиента Web, реализованного в виде SCADA-приложения InTouch (см. врезку Серверное расширение FactorySuite Web Server ).

На рисунке 6 показаны возможности разработки Internet-приложений и запуск их в реальном времени на примере SCADA-системы InTouch.

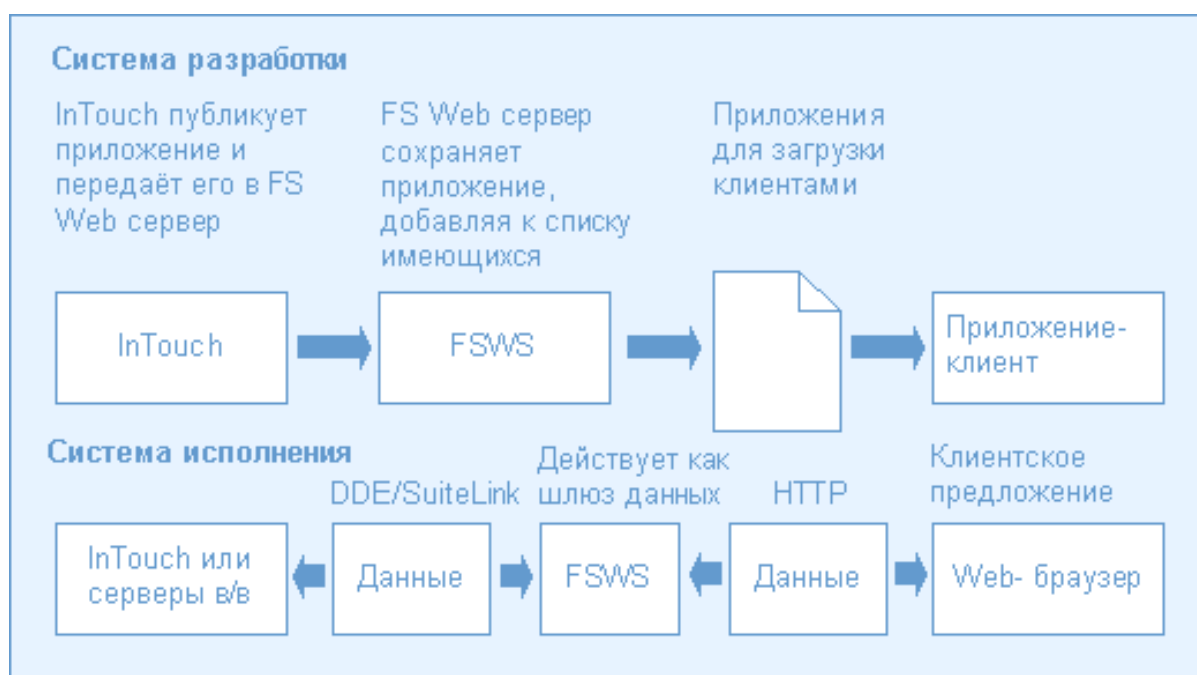


Рис. 6 Web сервер для обмена данными между приложениями InTouch.

Для просмотра приложения web-клиентом могут использоваться навигатор Microsoft Internet Explorer (MIE) или исполняющая система InTouch.

Интернет приложение позволяет собирать данные с многих FS Web-серверов (рис.7). В таких случаях каждый Web-сервер адресуется специально именем или IP-адресом.

Чтобы подписаться на приложение, необходимо загрузить его из текущего FS Web-сервера и выделить его в локальную директорию на клиентской машине.

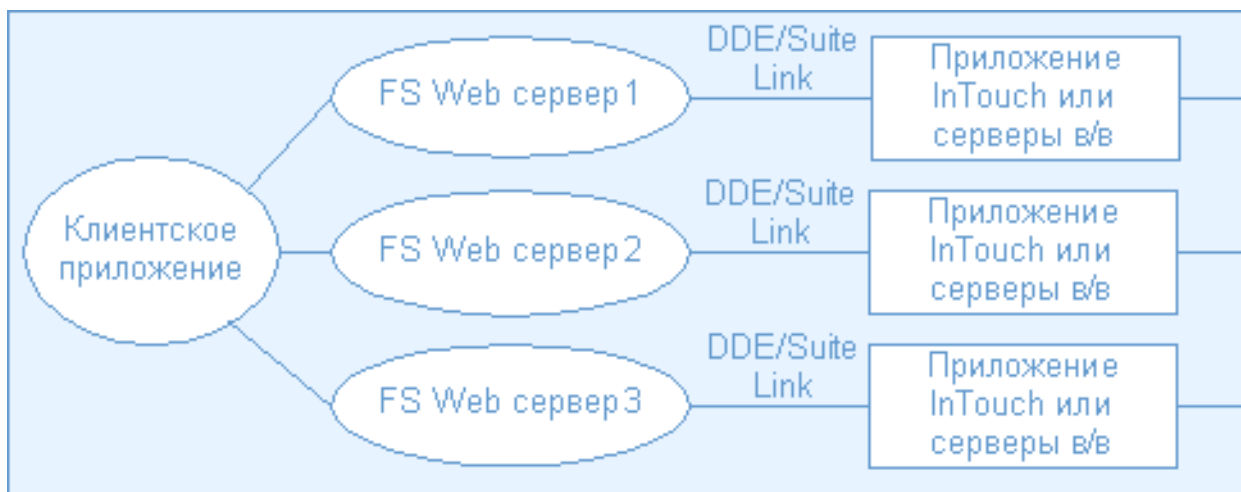


Рис. 7 Получение данных от нескольких Web-серверов.

Опубликованные приложения могут редактироваться на компьютерах-клиентах. Для запуска приложения InTouch на клиентском узле должна быть установлена исполняющая система InTouch. При этом web-клиентам передаются только данные, и данные следующих типов:

- удаленные InTouch переменные;
- распределенные алармы;
- архивные данные InTouch;
- данные с серверов ввода-вывода;
- данные из IndustrialSQL Server.

Публикация приложения InTouch возможна в двух режимах: с исходными файлами, так что приложение может модифицироваться в среде разработки в дальнейшем, и в режиме исполнения только.

Таким образом, приложения некоторых SCADA-систем могут поддерживать функцию толстого или богатого Internet-клиента. Преимущество применения такого клиента в том, что способ разработки клиентского приложения остается традиционным (обычное SCADA-приложение), возможно использование режима управления. А недостатком, безусловно, является то, что для каждого клиентского узла оплачивается лицензия.

Если клиент является бедным, то обработка любого запроса клиентского приложения выполняется на сервере. Клиенту предоставляется только Web-страница. Рассмотрение такого типа клиентов начнем с клиентов к базам данных (БД).

## 6. Базы данных реального времени (БДРВ) и Internet-решения

Поскольку БДРВ поддерживают SQL-запросы, то для организации доступа с технологической информации возможен стандартный подход как к обычным реляционным БД. Традиционный подход позволяет получать данные из БД и БДРВ, используя уже ставшие стандартными SQL-объекты, доступные, практически, из любого браузера. Этот подход требует программистского опыта разработки web-сайтов и использования специальных SQL-объектов, но является типичным примером бедного клиента.

Теперь рассмотрим более простую, с точки зрения пользователя-разработчика сайта, процедуру доступа к БДРВ на примере IndustrialSQL Server от Wonderware. Программа IndustrialSQL Server использует трехуровневую клиент-серверную архитектуру (рис. 8), которая позволяет создавать приложения Интернет/интранет. Обработка запроса на получение данных, сделанного клиентским объектом к IndustrialSQL Server, поддерживается с помощью специальных объектов Business Objects. Специальные объекты являются объектами COM (Component Object Model), которые размещаются либо на локальном компьютере, либо на Microsoft Internet Information Server (IIS), и в этом случае он доступен через Интернет и отвечает за получение данных из БДРВ.

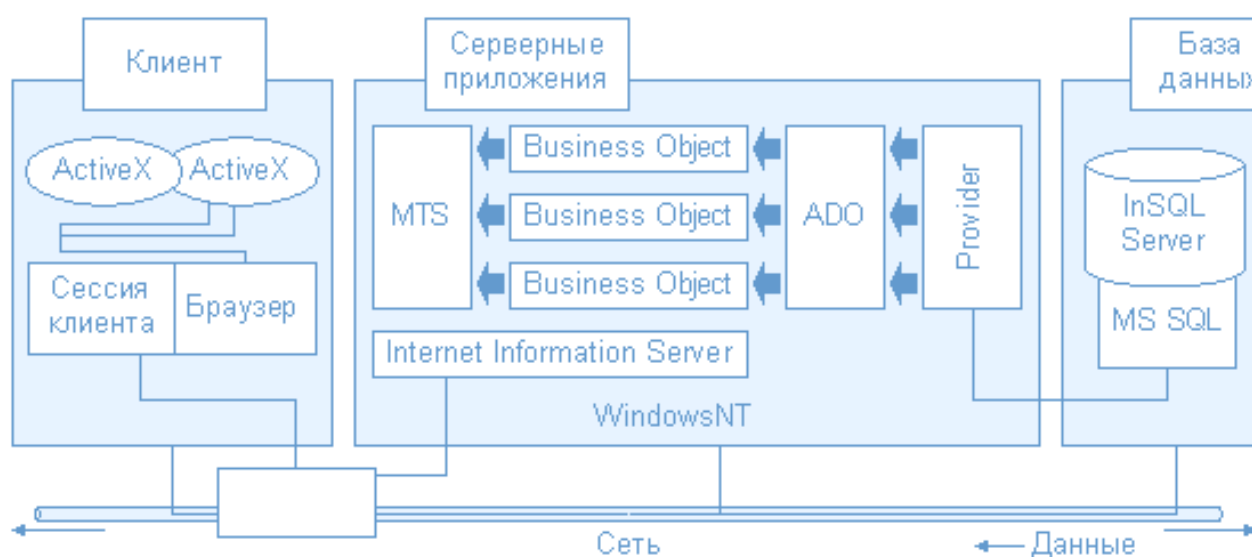


Рис. 8 Трехуровневая клиент/серверная архитектура

Клиентские приложения. Формат таблиц базы данных в БДРВ, в основном,

предопределен. И клиентские приложения, учитывая предопределенный формат таблиц, обеспечивают доступ к данным для визуализации и анализа. Клиентские приложения не требуют от пользователя знания языка SQL-запросов, что расширяет класс пользователей. Так, для продуктов Plant2SQL (CiTechnologies) и IndustrialSQL Server (Wonderware) компании предлагают специальные приложения, ориентированные на получение данных из БДРВ. С технологической точки зрения часть приложений реализована как независимые приложения, другая часть представляет объекты ActiveX.

Как независимые приложения, встроенные в программы Microsoft Office, так и объекты ActiveX предназначены для создания текущих и архивных трендов, для создания параметрических графиков X-Y и для табличного отображения текущих и архивных данных. На врезке Список объектов ActiveX из FactoryOffice описаны объекты ActiveX из пакета FactoryOffice компании Wonderware.

Объекты ActiveX могут встраиваться в приложения InTouch, Visual Basic, Visual C, и в HTML-страницы Internet Explorer. А специальные серверные компоненты Business Objects обеспечат получение данных, запрошенных в объекте ActiveX или SQL-запросе. Использование технологии ActiveX, с точки зрения клиентских приложений, сводится к настройке на Интернет-обмен при конфигурировании соответствующего ActiveX-объекта: для этого активизируется свойство Use Internet Server (Использовать Internet сервер) и определяется имя и/или IP-адрес сервера в форме HTTP://имя сервера.

Использование объектов ActiveX оснащает бедных клиентов новыми возможностями, т.е. бедные клиенты не одинаково бедны.

## **7. Специальный инструментарий для создания Internet/Intranet-клиентов**

Если Вы не используете готовых приложений клиентов Web, то для того чтобы создать свой Web-сайт и при этом разработать не просто бедного клиента, а клиента, оснащенного ActiveX-объектами, Java-апплетами и т.п., целесообразно рассмотреть применяемый для этого инструментарий. Он бывает разнородным традиционный инструментарий общего назначения и инструментарий, ориентированный на особенности механизмов обмена, используемых в АСУТП. Специализированный инструментарий характеризуется тем, что сейчас его поставляют следующие компании:

- независимые компании (Intuitive Technology), предлагающие поддержку характерных для АСУТП протоколов (DDE, OPC, OLE DB), обеспечивая, таким образом, клиентские приложения данными в реальном

времени;

- компании-разработчики SCADA-систем. Их инструментарий поддерживает не только ставшие стандартными протоколы обмена, но частнофирменные протоколы, конвертацию приложений SCADA в HTML, языки XML. Как пример такого инструментария ниже будет рассмотрен программный пакет SuiteVoyager от Wonderware.

Создание собственного или редактирование существующего web-сайта. Пользователь устанавливает соединение с сервером Web через сеть с помощью специальной программы просмотра страниц Web браузера, например, навигатора MIE и Netscape Navigator. При установке соединения пользователь указывает адрес сервера Web. Дополнительно он может указать путь к файлу страницы web, которая должна отображаться сразу после подключения к серверу. К серверу может подключаться несколько Web-сайтов. Web-сайт это не просто набор отдельных Web-страниц, а иерархическая система HTML-документов, файлов, графических изображений, апплетов на языке Java, текстовых видео- и аудиофайлов, а также сценариев на языке CGI или ином языке. Для обеспечения целостности сайта используются гипертекстовые связи (hyperlink) врезка Гипертекстовая связь .

Для создания сайтов сейчас на рынке предлагается огромное разнообразие инструментальных средств, и их выбор зависит в первую очередь от решаемых задач. Для создания сайтов, ориентированных на мониторинг и управление технологическим процессом, предлагается использовать пакеты Microsoft InterDev или FrontPage. Рассмотрим особенности последнего. Итак, FrontPage обладает следующими свойствами:

- он используется как визуальное средство, позволяющее непрограммистам реализовать web-публикацию в среде клиент/сервер;
- он используется для обслуживания Web-сервера и web-сайтов на этом сервере;
- Web-страница с FrontPage поставляется с 16- и 32-разрядными версиями собственного сервера Personal Web Server, который может использоваться с ОС Windows.

Перед установкой пакета FrontPage должно быть установлено программное обеспечение Web-сервера, ответственное за обработку полученных от клиен-

та данных, динамическое формирование HTML-документа и возврат его пользователю. Серверные расширения FrontPage поддерживают стандарты HTTP и CGI, обеспечивая совместимость с существующими HTML-документами и CGI-сценариями (см. врезку Назначение интерфейса CGI ).

Текстовые файлы страниц готовятся с использованием специального языка разметки гипертекста HTML (Hyper Text Markup Language); Взаимодействие пользователя с сервером WWW осуществляется через формы. Сервер, получив данные из полей формы, запустит созданное специально для этой формы программное расширение для обработки полученных данных, динамически сформирует документ HTML и возвратит его пользователю (нет ограничений на вид выполняемой обработки или вид сформированного документа HTML).

Сервер, содержащий наряду со статическими документами динамические, называют активным интернет-сервером. Активные серверы создаются и используются программных расширений сервера WWW-приложений CGI, ISAPI (врезку Программные расширения ISAPI ).

Данные, полученные через запросную форму, передаются программному расширению CGI или ISAPI. Эти расширения могут обратиться, например, к СУБД через интерфейс ODBC или через интерфейс этой СУБД, а результат запроса оформить в виде документа HTML и вернуть удаленному пользователю.

Возможности языка HTML ограничены. Часто требуется обрабатывать содержимое локальных файлов, отображать данные в графическом виде или выполнять другую нетривиальную работу. Создав орган управления ActiveX и расположив его на сервере WWW, можно сделать ссылку на этот орган в документе HTML.

Код ActiveX загружается из сервера WWW в адресное пространство удаленного компьютера и поэтому имеет доступ ко всем его ресурсам. Это позволяет организовать сложные алгоритмы обработки и отображения любых локальных данных, что невозможно при использовании программных расширений CGI и ISAPI. Но ActiveX представляет и потенциальную угрозу в смысле распространения вирусов. Для уменьшения этой угрозы компания Microsoft предложила сертификацию органов управления ActiveX. Когда пользователь попадает на страницу со ссылкой на ActiveX, ему выдается изображение сертификата фирмы-разработчика. Если пользователь доверяет сертификату, он может согласиться на загрузку и запуск ActiveX, если нет можно отказаться.

Язык HTML допускает использование языков программирования Java, JavaScript и VBScript. Язык программирования Java разработан фирмой Sun на основе языка Oak, как платформно-независимый интерпретируемый, объектно-ориентированный язык. Ссылки на созданные программы Java размещаются в документах HTML. Такие Java-программы называются апплетами (applets). Программы Java, расположенные на сервере WWW, обладают большими возможностями по обработке и отображению данных. По сравнению с объектами ActiveX они более безопасны, поскольку не могут выполнять запись на локальные диски и читать с них.

Для создания программ на языке Java могут использоваться следующие средства разработки: Java Development Kit (JDK) от Sun, SDK-Java или Visual J++ от Microsoft.

Исходный текст программ, составленных на языке программирования JavaScript и VBScript, вставляется непосредственно в документ HTML, поэтому для их разработки не нужны специальные средства. Интерпретатор JavaScript и VBScript встроен непосредственно в навигатор Microsoft Internet Explorer (Netscape с языком VBScript не работает).

Страницы сервера WWW содержат ссылки на другие страницы, реализованные в виде специальных текстовых строк, либо в виде графических объектов или органов управления. Страницы могут ссылаться на страницы, расположенные и на других серверах в сети Интернет, включая серверы FTP, Gopher, конференции, электронные почтовые адреса.

Следует сказать и о языке XML (Extensible Markup Language), имеющем общего предка с HTML стандартного обобщенного языка описания документов SGML (Standard Generalized Mark-up Language), но обладающего более строгим синтаксисом. Отмечается тенденция использования языка HTML для представления данных, а не для обмена ими, в то время как публикация данных происходит в формате XML. Производители Oracle, Sybase, Informix скоро начнут выдавать результаты запросов в формате XML и импортировать XML-данные в свои таблицы.

Таким образом, используя инструментальные средства, подобные FrontPage, можно создавать собственные, ориентированные на решение конкретных задач web-сайты. Технологии, предлагаемые Microsoft, позволяют применять как технологию ActiveX, так и технологию доступа к реляционным базам данных. Их использование допускает встраивание:



- рассмотренных ранее объектов ActiveX для доступа к данным IndustrialSQL Server (ActiveTagBrowser, ActiveDataGrid, ActiveGraph, ActiveTimeSelector);
- рассмотренных ранее объектов ActiveX для доступа к данным IndustrialSQL Server (ActiveTagBrowser, ActiveDataGrid, ActiveGraph, ActiveTimeSelector);
- стандартных форм SQL-запросов ряда навигаторов (прежде всего Microsoft Internet Explorer).

Сервер WWW может решать любую задачу, принимая любые данные от удаленного пользователя, обрабатывая их и передавая обратно.

Пакет SuiteVoyager. Специальный пакет SuiteVoyager от Wonderware поставляет масштабируемое, расширяемое средство разработки информационных порталов. Портал это просто Web-сайт, который предоставляет пути доступа к дополнительной информации по определенным темам. Пакет SuiteVoyager является набором интегрированных программ, поддерживающих удобный способ получения технологической информации (рис. 9).

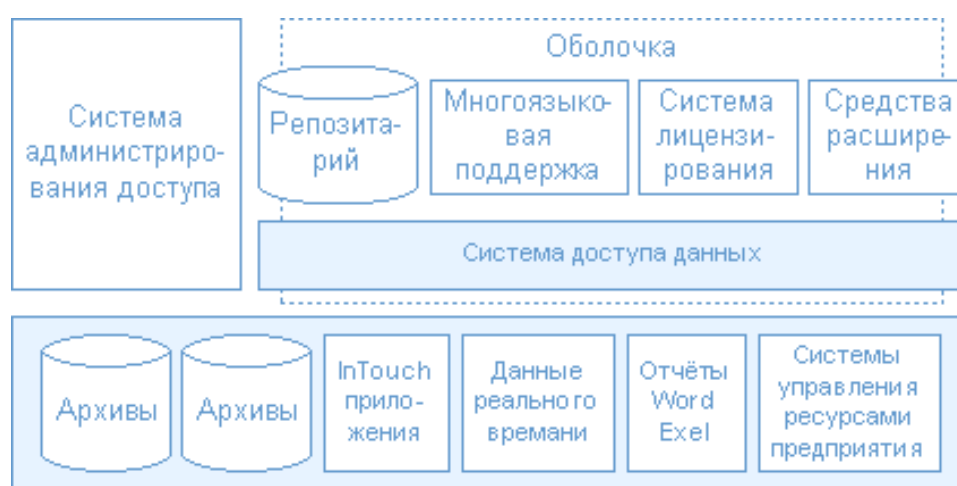


Рис. 9 Структура портала SuiteVoyager

Пакет представляет набор средств для просмотра, подготовки отчетов на основе технологических данных. Традиционно передача графической информации требует доставки файлов большого размера и длительных периодов времени для загрузки. Чтобы преодолеть это ограничение, SuiteVoyager по-

ставляет интерактивные HTML-страницы, преобразуя существующие графические окна InTouch (и ассоциированную с ними анимацию) в формат XML (рис. 10).

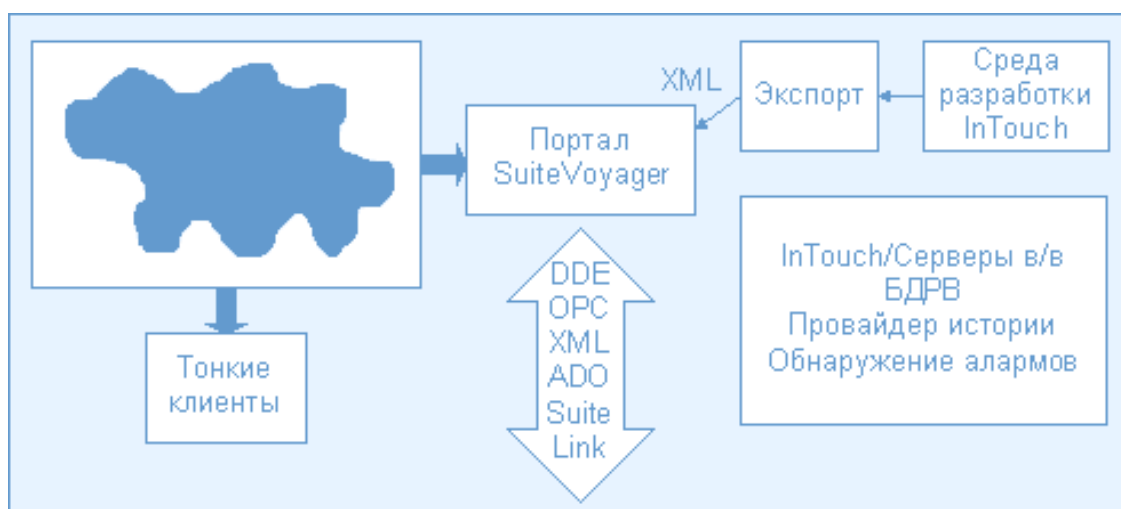


Рис. 10 Решение на основе SuiteVoyager

Использование технологии XML уменьшает объем передаваемой между клиентом и сервером информации почти на 80%. Пакет SuiteVoyager позволяет пользователям визуализировать технологическую информацию, поступающую из серверов ввода-вывода, SCADA-приложений, БДРВ через Internet/Intranet, используя Internet Explorer 5+. Пакет поддерживает новые, ориентированные на web (made-for-the-Web) технологии, такие как XML (eXtensible Mark-up-Language).

## 8. Взаимодействие серверного и клиентского программного обеспечения.

За последние годы применение ПК эволюционировало от централизованных интерактивных систем на базе крупных ПК к распределенной архитектуре «клиент-сервер». Это стало возможным с распространением LAN на основе ПК. В основе «клиент-серверной» архитектуры лежит распределенная модель вычисления. В самом общем случае под «клиентом» и «сервером» понимают два взаимодействующих процесса, один из которых является поставщиком некоторого сервиса для другого.

Сервер – логический процесс, который обеспечивает некоторый сервис по запросу от клиента. Обычно сервер не только выполняет запрос, но и управляет очередностью запросов, буферами обмена, извещает своих клиентов о выполнении запроса и т.д.

Клиент – процесс, который запрашивает обслуживание от сервера. Процесс не является клиентом по каким-то параметрам своей структуры, он является процессом по отношению к серверу.

При взаимодействии клиента и сервера инициатором диалога с сервером, как правило, является клиент. Сервер сам не инициирует собственную работу. Это не исключает, однако, того, что сервер может извещать клиентов о каких-нибудь зарегистрированных им событиях. Инициирование взаимодействия, запрос на обслуживание, восприятие результатов от сервера, обработка ошибок – это обязанности клиента.

В «клиент-серверной» системе программа сервера стартует первой и пассивно ожидает запросов от клиента и при получении обрабатывает их при помощи различных механизмов. Таким образом, в модели «клиент-сервер» различают запросы и ответы.



В данном случае сеть изображена в виде овала, при разговоре об аспектах сети будем раскрывать этот овал. Программирование в этой среде бывает весьма разнообразным вследствие разнообразных сетевых ОС (NOS – Network Operating System), которые объединяют ПК в ЛВС. Каждая такая NOS имеет свой собственный интерфейс прикладного программирования (API – Application Programming Interface) на высоком уровне и поддерживает различные интерфейсы прикладного программирования более низкого уровня для осуществления связи.

Сетевое приложение «файл-серверной» архитектуры отличается от предыдущей тем, где происходит обработка данных. Данные в виде одного или нескольких файлов размещаются на файловом сервере. Сервер принимает запросы от ПК в сети и передает им требуемые данные. Основная обработка данных происходит в процессе клиента. Сервер следит лишь за тем, чтобы не возникало конфликтов при одновременном обращении к файлам.

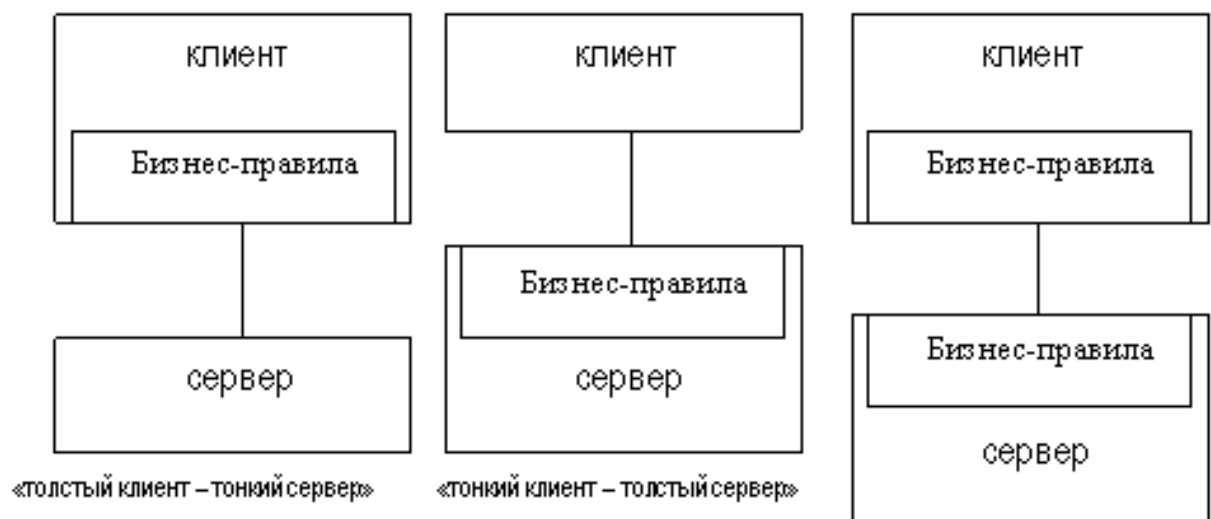
В «клиент-серверной» архитектуре сервер не только обеспечивает пересылку необходимой информации, но и берет на себя часть или всю обработку этих данных.

Недостатки «файл-серверной» архитектуры очевидны и вытекают из того, что данные хранятся в одном месте, а обрабатываются в другом. Это означает, что данные в больших количествах необходимо передавать по сети, а это приводит к большой нагрузке на сеть и, как следствие, к снижению производительности при увеличении количества одновременно работающих пользователей.

«Клиент-серверная» архитектура позволяет устранить эти недостатки путем распределения вычислительной нагрузки между клиентом и сервером. Именно это распределение и является достаточно сложной задачей при проектировании программного обеспечения сетевых вычислительных систем.

Основная задача клиентского приложения – это обеспечение интерфейса с пользователем, т.е. ввод данных и предоставление результатов в удобном для пользователя виде и управление сценариями работы приложения.

Основная задача серверного приложения – это обеспечение надежности, согласованности и защищенности данных, управление запросами клиентов, быстрая обработка запросов и предоставление ответов.



Сначала большую часть функций приложения выполнял клиент, и такая архитектура получила название «толстый клиент – тонкий сервер». Когда появилась возможность хранить на серверы исполняемые процедуры, то это

привело к тенденции большую часть приложения хранить на сервере. Такое решение имело очевидные преимущества, например его легче поддерживать, так как изменения нужно вносить только в одном месте – на сервере. Но язык описания хранимых процедур не был достаточно гибким по сравнению с языками программирования клиентских приложений. Поэтому возникли трехзвенные и многозвенные архитектуры «клиент-сервер», сочетающие в себе достоинства всех перечисленных выше систем.

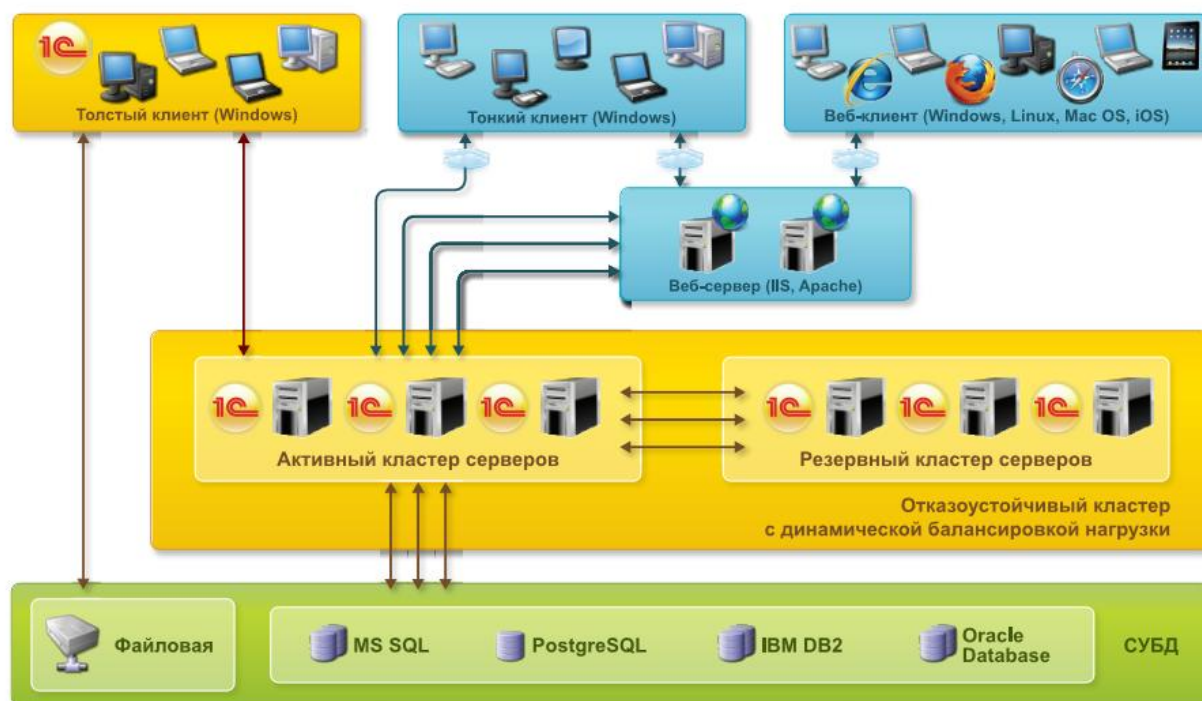
## **9. Три клиентских приложения: толстый клиент, тонкий клиент, веб-клиент**

«**Толстый**» клиент – приложение, исполняющее практически всю функциональность программного продукта. Данное приложение требует значительного количества ресурсов на компьютере пользователя и может «общаться» с базой данных только посредством файлового доступа или по локальной сети

«**Тонкий**» клиент – приложение, исполняющее ограниченный набор функциональности продукта. Тонкий клиент оперирует ограниченным набором типов встроенного языка, предназначенным лишь для отображения и изменения данных в памяти. Вся работа с базой данных, объектными данными, исполнение запросов выполняются на стороне сервера. Тонкий клиент только получает готовые данные, подготовленные для отображения.

**Веб-клиент** – приложение, не требующее предварительной установки на компьютер. В отличие от толстого и тонкого клиентов оно работает через интернет-браузер: Windows Internet Explorer, Mozilla Firefox, Google Chrome или Safari. Пользователю достаточно запустить свой браузер, ввести адрес веб-сервера, на котором опубликована информационная база, и начать работать.

**Пример схемы(1С):**



## 10. Средства разработки клиентских приложений

Клиентская часть приложения может создаваться как при помощи специализированных программных сред, поставляемых совместно с сервером, так и при помощи обычных средств разработки программ.

Для того, чтобы воспользоваться многочисленными новейшими инструментальными средствами, предназначенными для создания клиентской части приложений, которые доступны сегодня на рынке программного обеспечения, разработчики должны уметь программировать на таких языках, как C++ и HTML, или на одном из множества других процедурных языков программирования, предназначенных для разработки Web-приложений. Раньше для разработки пользовательских корпоративных программ, работающих в основном в символьном режиме, использовались такие языки программирования, как ANSI C, COBOL, FORTRAN и Pascal. Сегодня большинство вновь разрабатываемых клиентских прикладных программ является GUI-приложениями - они содержат графический интерфейс пользователя. Большинство из доступных сегодня инструментальных средств являются дружелюбными по отношению к пользователю и объектно-ориентированными. В них широко используются пиктограммы, различного рода мастера, а также технология drag-and-drop. Наиболее популярными средствами для создания Web-приложений являются C++-Builder и IntraBuilder фирмы Borland, а также Visual J++ и Visual C++ компании Microsoft. Другие популярные средства разработки корпоративных приложений для локальных вычислительных се-

тей - PowerBuilder компании Powersoft, Developer/2000 корпорации Oracle, Visual Basic компании Microsoft и Delphi фирмы Borland.

Использование приложений для браузеров позволяет широко использовать все возможности WWW, поскольку приложение фактически становится составной частью гипертекстового документа. Это приводит к существенному упрощению системы поддержки – клиентское ПО автоматически обновляется при подключении к серверу, документация и система помощи легко реализуется с помощью стандартных средств HTML, а некоторые учебные курсы, к примеру, могут быть реализованы практически полностью средствами HTML.

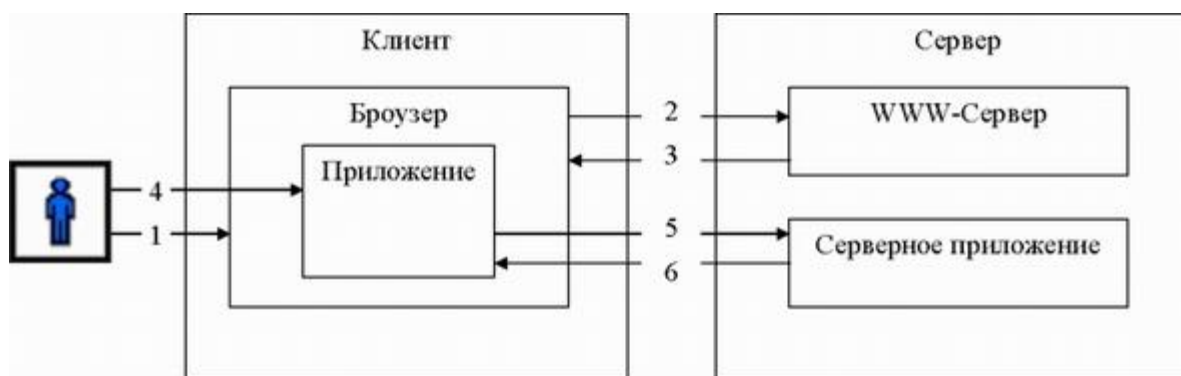


Рис. 3. Схема взаимодействия пользователя, сервера и серверных приложений

Пользователь взаимодействует с браузером (1), который запрашивает (2) и получает (3) от веб-сервера страницу с включенным приложением. После этого начинается взаимодействие пользователя с приложением (4), которое при необходимости может связаться с серверным приложением (5,6) для взаимодействия с СУБД и т.п.

В то же время возможностей HTML недостаточно для реализации сложных клиентских приложений, что приводит к необходимости использования дополнительных средств и написания вспомогательных программ.

Отметим основные проблемы в разработке клиентских средств для WWW.

1. Проблема многоплатформенности. Поскольку к WWW подключены системы, основанные на различных аппаратных и программных платформах, возникает необходимость разработки для каждой платформы (которую предполагается поддерживать) специализированной версии ПО. Это приводит либо к ограничению поддерживаемых платформ, либо к возрастанию затрат на разработку и поддержку ПО. Для корпоративных закрытых систем с однородной архитектурой аппаратных и программных средств эта проблема менее актуальна.

2. Проблема безопасности. Здесь имеют место два аспекта проблемы – возможные сбои в системе из-за ошибок в программном обеспечении и целенаправленные действия внешнего программного обеспечения, направленные на нарушение функционирования системы, на копирование конфиденциальной информации и т.д.
3. В традиционных языках отсутствуют встроенные средства для выполнения наиболее часто встречающихся (при работе с Web) операций – взаимодействие с удаленными серверами, загрузка файлов, работа с изображениями и т.д.
4. Трудности передачи по сети из-за значительных размеров исполняемых файлов, генерируемых традиционными средствами разработки.

Наибольшую популярность завоевали следующие подходы к реализации вспомогательных приложений для клиентской стороны:

- реализация подключаемых модулей (plug-ins) Netscape;
- использование элементов ActiveX;
- использование Java-приложений;
- средства подготовки сценариев JavaScript, VBScript.;
- Macromedia Flash.

Рассмотрим эти подходы более подробно.

Использование подключаемых модулей получило широкое распространение в связи с популярностью браузера Netscape Navigator, предоставляющего такую возможность. Однако сейчас это путь является тупиковым, поскольку не дает адекватного решения ни по одной из перечисленных проблем.

К достоинствам использования элементов ActiveX относится реализация на основе OLE/COM-технологии, что позволяет достаточно легко перевести в эту форму традиционное ПО, реализованное для Win32-платформ. Среди недостатков – поддержка в настоящее время только Win32-платформы и поддержка только браузером Internet Explorer, хотя для Netscape Navigator существует plug-in, позволяющий использовать элементы ActiveX. Размер элементов ActiveX минимизирован за счет использования разделяемых динамических библиотек, поставляемых вместе с браузером Internet Explorer, которые также предоставляют все необходимые для сетевого взаимодействия функции. Проблема безопасности решается с помощью введения института сертификатов – при загрузке пользователю предъявляется сертификат производителя, заверенный независимой организацией (например, VeriSign Inc.), после чего он может решить, доверять или нет этому компоненту. Защита от сбоев не предусмотрена. Таким образом, проблема безопасности решается административными средствами.



Данная технология наиболее пригодна для разработки корпоративных приложений для внутреннего пользования – особенно в случае существования готовых наработок в этой области.

Основным достоинством Java-приложений является их независимость от клиентской платформы. В отличие от традиционных приложений, транслирующихся в исполняемые коды процессора, Java-приложения транслируются в так называемый байт-код, интерпретируемый в дальнейшем виртуальной Java-машиной. При этом байт-код независим от платформы, на которой он в дальнейшем будет выполняться – достаточно, чтобы для этой платформы была реализована Java-машина. Поскольку большая часть основных функций реализована на уровне виртуальной Java-машины, это приводит к существенному уменьшению размеров байт-кода. Это является как достоинством, так и недостатком Java-приложений – поскольку байт-код интерпретируется виртуальной машиной, производительность Java-приложений уступает производительности традиционных откомпилированных программ. Частично с этим удастся бороться, применяя компиляторы времени исполнения (JIT – just in time compilers), осуществляющие компиляцию приложения при его загрузке в «родной» для данного процессора код. Также возможен вызов функций, реализованных на других языках программирования (таких как C, C++) и откомпилированных для данной платформы – так называемый native code. Это применяется при реализации наиболее критичных к времени исполнения фрагментов кода.

Другим достоинством Java-приложений является их защищенность:

- **с точки зрения программирования:** из языка исключены средства, наиболее часто приводящие к ошибкам при программировании, такие как указатели, перегрузка операторов и т.д.; язык является строго объектно-ориентированным, в него встроена «сборка мусора» и т.д.;
- **с точки зрения исполнения:** значительно ограничена возможность работы с файлами на локальных машинах, а с установлением сетевых соединений и др. программа выполняется в отдельном адресном пространстве.

Это позволяет работать с приложениями, полученными из сети, не опасаясь наличия в них опасного кода. Таким образом, проблема безопасности полностью решена на уровне архитектуры.

Второй недостаток – необходимость существования для данной платформы виртуальной Java-машины. Java-машины реализованы для всех наиболее распространенных платформ, но они остаются достаточно ресурсоемкими, а зачастую довольно нестабильными системами. Кроме того, остаются проблемы несовместимости – поскольку язык Java изначально проектировался для написания многоплатформенных приложений, в него преимуществен-

но входили элементы, доступные на всех платформах, что привело к некоторой аскетичности доступных средств. Некоторые разработчики виртуальных машин расширяли их возможности для конкретной платформы, что может привести к тому, что Java-приложение, использующее все эти возможности, утратит способность запускаться на других платформах. Ошибки в реализации виртуальной машины могут также привести к снижению безопасности системы, чему за последние несколько лет было немало примеров.

Существенным достоинством Java является ее объектная ориентированность. Программа на Java представляет собой набор взаимодействующих между собой классов. С помощью классов осуществляется и доступ к основным сервисам виртуальной машины. Стандартная библиотека классов достаточно обширна и включает в себя классы для работы с сетевыми протоколами – как на низком, так и на прикладном уровне, с графикой, графическим пользовательским интерфейсом, базами данных, строками и т.д.

Перечисленные достоинства делают Java-приложения лучшим выбором в гетерогенных системах, для которых безопасность имеет большее значение, чем возможные потери в производительности.

JavaScript, VBScript и другие подобные средства представляют собой упрощенные языки подготовки сценариев, код которых встраивается непосредственно в html-файл и выполняется браузером. Они непригодны для реализации серьезных приложений, в них отсутствуют средства для работы с файлами, сетевого взаимодействия и т.д. Но они широко используются во вспомогательных целях, в качестве средства первоначальной обработки результатов, для оформления, «оживления» html-документов (так называемый Dynamic HTML) и т.д.

Macromedia Flash разрабатывалось как средство анимации, основанное на векторной графике, и в последнее время практически заняло ту нишу в web-дизайне, на которую еще несколько лет назад претендовала Java. Имеет достаточно ограниченные возможности по программированию, но весьма широкие – по построению мультимедийных приложений в рамках заданной модели (реакция на нажатие кнопок/движение мыши, анимация, векторная графика, слои). Для работы flash-приложений необходимо наличие соответствующего проигрывателя, существующего для большинства распространенных платформ (в том числе, в виде ActiveX-модуля и Java-апплета).