

УДК 004.75.05

К.А. Бохан<sup>1</sup>, Д.В. Гринев<sup>2</sup><sup>1</sup>Национальный аэрокосмический университет имени Н.Е. Жуковского «ХАИ», Харьков<sup>2</sup>Харьковский национальный экономический университет, Харьков

## ИНФОРМАЦИОННАЯ БИЗНЕС СИСТЕМА В CLOUD-СРЕДЕ

В статье проведен анализ основных архитектур клиент-сервер для построения корпоративных информационных систем. Рассмотрены концептуальные уровни и архитектурные звенья распределенных информационных систем. Предложен подход к построению информационных бизнес систем на основе сервис-ориентированной архитектуры SOA с развертыванием в Cloud-среде.

**Ключевые слова:** информационные бизнес системы, архитектура клиент-сервер, сервис-ориентированная архитектура (SOA), Web-сервис, облачные вычисления (Cloud Computing).

### Введение

**Постановка проблемы.** В последнее время растет доля бюджета предприятия, выделяемая под интеграционные проекты, имеющие целью предоставить более широкую и унифицированную поддержку для бизнес-процессов. Однако, зачастую эти проекты не оправдывают ожиданий из-за жесткости, присущей интеграции на аппаратном уровне: изменение одного-единственного приложения вызывает необходимость последующих изменений во всей ИТ-структуре, что приводит к дорогой и трудноизменяемой инфраструктуре. Подавляющее большинство отечественных распределенных информационных бизнес систем реализованы в рамках клиент-серверной архитектуры с «толстым» клиентом, в которых на рабочие места пользователей устанавливается ресурсоемкое специальное программное обеспечение (ПО), реализующее интерфейс пользователя. Такая архитектура ПО имеет ряд ограничений, связанных со сложными процессами развертывания (внедрения), сопровождения (обновления) ПО, а также интеграцией программного обеспечения корпоративных информационных систем (ИС) с другими ИС [1]. Перечисленные ограничения являются проблемой в развитии современных информационных бизнес систем.

Решение преодолеть названные ограничения привело к развитию архитектур и технологий создания программного обеспечения, связанных с размещением логики приложений (компонентов ПО) и всех данных на серверных вычислительных комплексах, что особенно важно для информационных бизнес систем ввиду наличия большого объема данных и ресурсоемкости операций их обработки [1, 2].

Наиболее перспективной архитектурой разработки программного обеспечения, в рамках которой возможно эффективное решение задачи оптимизации развертывания, сопровождения, и интеграции ПО с другими ИС, является сервис-ориентированная архитектура (Service-Oriented Architecture – SOA) [2, 3]. Поэтому целью статьи является анализ способов программно-аппаратной реализации архитекту-

ры информационных бизнес систем.

### Изложение основного материала

Основной задачей распределенных ИС является облегчение пользователям доступа к удаленным ресурсам и контроль совместного использования этих ресурсов.

Концептуально распределенные ИС строятся в несколько уровней (слоев), представленных на рис. 1. Эти уровни могут быть только абстракциями, существуя лишь на этапе проектирования, но могут быть четко видимы в ПО, когда их реализуют в виде отдельных подсистем. Программное обеспечение ИС разбивается, как правило, на три основных физических компоненты архитектуры клиент-сервер (ПО клиента, промежуточное обеспечение и ПО сервера), которые распределяются между несколькими концептуальными уровнями.

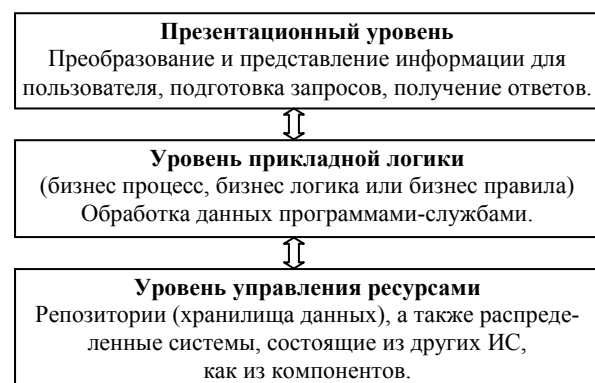


Рис. 1. Концептуальные уровни распределенной ИС

Программное обеспечение первых корпоративных информационных систем представляло собой монолитные модули, включающие в себя все компоненты ПО, такие как СУБД, логика приложения и логика взаимодействия с пользователем, которые размещались на серверах. Доступ к таким системам осуществлялся посредством «тонких» клиентов, представляющих собой программу-клиент, реализующую на сервере большую часть задач по обра-

ботке информации. В силу концентрации ПО на сервере, такой тип архитектуры может быть как однозвенной клиент-серверной архитектурой, так и многозвенной. На рис. 2 и 3, а представлена одно- и двухзвенная архитектура с «тонким» клиентом.

С начала 90-х годов начинает преобладать двухзвенная клиент-серверная архитектура (с «толстым» клиентом), в рамках которой на серверах размещаются данные и СУБД, а остальные компоненты программного обеспечения, такие как логика приложений и графический интерфейс пользователя размещаются на компьютерах пользователей, так называемых «толстых» клиентах, обеспечивающих расширенную функциональность независимо от центрального сервера (рис. 3, б). Для многих типов ресурсоемких приложений, например, приложений обработки видео, графики, геоинформационных систем и др. этот тип архитектуры остается распространенным и сегодня.

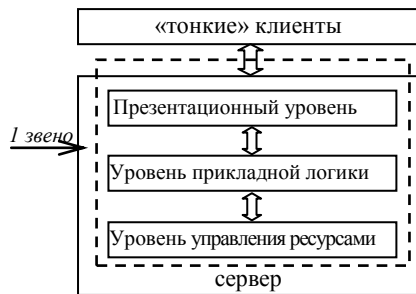


Рис. 2. Однозвенная клиент-серверная архитектура

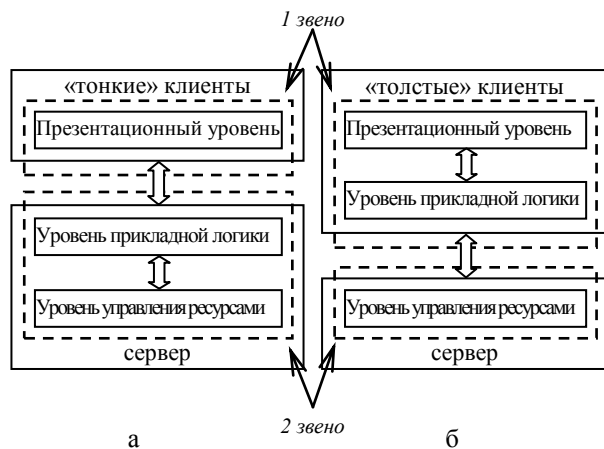


Рис. 3. Двухзвенная клиент-серверная архитектура

Достоинства двухзвенной клиент-серверной архитектуры с «тонким» клиентом заключаются в следующем:

- относительная простота эксплуатации за счет отсутствия инсталляции, обновления и сопровождения клиентских мест;
- хорошая масштабируемость.

Основным недостатком является сильная зависимость интерфейса от качества работы сетевой инфраструктуры. Основным достоинством двухзвенной клиент-серверной архитектуры с «толстым» клиентом заключается в независимости интерфейса от качества работы сетевой инфраструктуры. Недостатки:

- высокие программно-аппаратные требования к клиенту;
- ресурсоемкие процессы внедрения и сопровождения ПО;
- сложность интеграции с другими ИС (низкая интероперабельность);
- низкая масштабируемость;
- привязка к одной платформе;
- низкая актуальность данных.

Однако двухзвенные архитектуры не справились с требованиями локальных информационных сетей. С их помощью клиенты могли общаться только со своими серверами, не будучи в силах взаимодействовать с другими.

Трехзвенные архитектуры клиент/сервер сложнее и разнообразнее двухзвенных архитектур. Все уровни в них четко разделены (рис. 4).

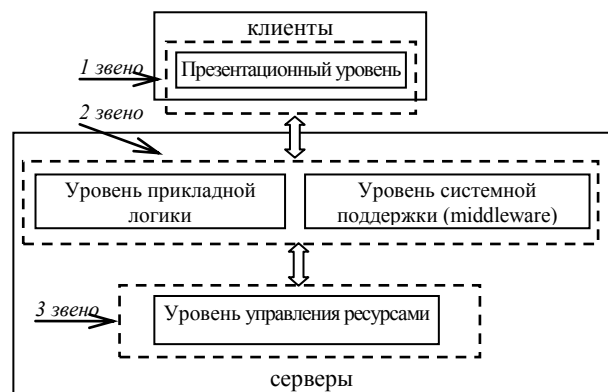


Рис. 4. Трехзвенная клиент-серверная архитектура

Презентационный уровень может размещаться в клиенте, как в двухзвенной архитектуре, однако в последнее время на стороне клиента происходит только рендеринг – процесс получения изображения, а генерирование интерфейса осуществляется на стороне middleware. Прикладная логика размещается в среднем звене и называется уровнем системной поддержки или промежуточным уровнем программного обеспечения (middleware). Уровень управления ресурсами располагается на третьем звене и состоит из всех серверов, которые интегрируются в архитектурном решении. С точки зрения подсистемы управления ресурсами программы, работающие на уровне прикладной логики, это просто клиенты.

Хотя трехзвенные архитектуры разрабатывались преимущественно для интеграции, их можно использовать точно так же, как и двухзвенные. Их преимуществом при этом будут возросшие возможности по масштабированию. Каждый уровень может работать на отдельном узле. В частности, прикладной уровень может быть распределен по разным компьютерам. Прикладная логика при этом может быть сделана существенно более независимой от управления ресурсами, ее переносимость существенно возрастает.

Трехзвенные системы дополнили и развили концептуально важные понятия, выдвинутые двухзвенными системами. Стало очевидно, что управление ре-

сурсами должно подчиняться четким интерфейсам, которыми должны пользоваться программы прикладной логики, находящиеся в промежуточном уровне. Если двухзвенные архитектуры потребовали определения интерфейсов прикладного уровня, то трехзвенные привели к стандартизации интерфейсов уровня управления ресурсами. Особенно четко преимущества трехзвенной архитектуры проявляются при интеграции разнородных ресурсов. Современное ПО промежуточных уровней включает в себя функциональность, необходимую для введения в эти уровни дополнительных свойств: транзакционных гарантий для различных видов ресурсов, балансировки загрузки оборудования, возможностей по регистрации событий, репликация, сохранности данных и многого другого. Потери в производительности компенсируются распространением модели промежуточного уровня на разные сетевые узлы, что существенно влияет на масштабируемость и надежность систем. Ограниченность модели трехзвенных систем проявилась при попытках интегрировать несколько трехзвенных систем, а также при выходе распределенных систем на уровень Интернета, что связано недостаточной стандартизацией этих систем.

Многозвенные архитектуры это обобщение двух и трехзвенных моделей, с учетом объединения разнородных систем. Серверное приложение ИС в рамках многозвенной архитектуры, как правило, представляет собой монолитный программный модуль, и реализация взаимодействия внешних информационных систем с отдельными логическими компонентами этого приложения является трудоемкой задачей без разбиения серверного ПО на физические модули.

Эти ограничения преодолеваются в рамках сервис-ориентированной архитектуры [2, 3]. Единые межплатформенные механизмы взаимодействия отдельных частей программного обеспечения, отсутствующие в двух и трехзвенной клиент-серверных архитектурах, и декомпозиция ПО ИС на набор сервисов – слабосвязанных исполняемых и готовых к использованию группы компонент – позволяют эффективно решать задачи интеграции корпоративных ИС с другими информационными системами путем предоставления межплатформенного сетевого доступа к функциональным возможностям сервисов ИС клиентам и другим информационным системам.

Сервис-ориентированная архитектура – это метод создания бизнес-приложений, использующих общие сервисы для поддержки бизнес-функций.

Отличие SOA от более общей модели клиент-сервер заключается в явном акценте на слабой связанности между компонентами ПО (бизнес сущностями) и в характерном для нее использовании самых разных интерфейсов. Первичным структурным элементом приложений SOA является сервис, а не приложение. С точки зрения объектно-ориентированного программирования сервисы, как правило, состоят из наборов или библиотек классов, доступ к которым осуществляется с помощью программных средств через платформу-независимые интерфейсы.

С точки зрения элементов приложения сервисы состоят из компонент или бизнес сущностей.

Общий вид программной архитектуры SOA представлен на рис. 5. Присутствующие уровни в программной архитектуре SOA тесно перекликаются с концептуальными уровнями распределенной ИС (см. рис. 1) и реализуются в виде трех- и многозвенной архитектуры клиент-сервер. Использование «тонких» клиентов позволяет выделить клиентскую часть SOA в презентационный уровень, развернутый в одном архитектурном звене.

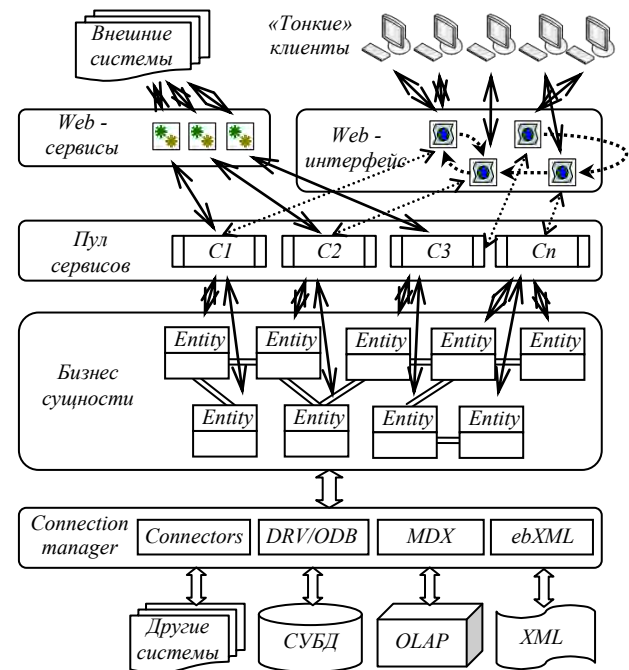


Рис. 5. Программная архитектура SOA

Web-сервисы вместе с Web-интерфейсами концептуально могут находиться на уровне прикладной логики и системной поддержки, с реализацией в одном архитектурном звене вместе с пулом сервисов и бизнес сущностями. Встречается также и разделение на несколько промежуточных уровней. Уровень управления ресурсами представлен нижним уровнем в архитектуре SOA.

В рамках SOA Web-сервисы это технология, которая позволяет приложениям взаимодействовать друг с другом независимо от платформы, на которой они развернуты, а также от языка программирования, на котором они написаны. Другими словами Web-сервисы это система, идентифицируемая строкой URI, чьи общедоступные Web-интерфейсы (совокупность средств, методов и правил взаимодействия между Web-сервисами) определены на языке XML. Описание этой системы может быть найдено другими системами, которые могут взаимодействовать с ней согласно этому описанию посредством сообщений, основанных на XML, и передаваемых с помощью интернет-протоколов. Web-сервис является единицей модульности при использовании сервис-ориентированной архитектуры приложения.

В настоящее время на предприятиях большинство SOA технически развернуты в сети в виде централизованной архитектуры (Dedicated Server Network) с выделенным сервером или на основе кластера, представленные на рис. 6.

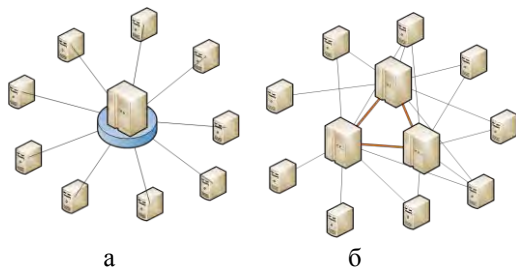


Рис. 6. Централизованная архитектура Dedicated Server Network: а – с выделенным сервером; б – на основе кластера

В централизованной архитектуре с выделенным сервером вся информация стекается на один центральный узел, на котором реализуются хранения основной информации и базовые сервисы системы. Таким образом, серверные функции не рассредоточены по сети, а централизованы на одном сервере. В централизованной архитектуре на основе кластера все ресурсы системы связаны центральным кластером с помощью диверсных каналов. На центральном кластере хранится вся информация системы и реализуются базовые сервисы.

Достоинства сети с выделенным сервером:

- высокая управляемость;
- высокая функциональная масштабируемость;
- надежная система защиты информации;
- относительно недорогая архитектура;
- относительная простота эксплуатации.

Недостатки сети с выделенным сервером:

- зависимость быстродействия, устойчивости и надежности клиентов от сервера (одна точка сбоя центрального узла, выводит из строя всю систему);
- низкая нагрузочная масштабируемость.

Достоинства сети на основе кластера:

- высокая управляемость;
- высокая функциональная и нагрузочная масштабируемость;
- надежная система защиты информации;
- высокая надежность и доступность.

Недостатки сети на основе кластера:

- низкая устойчивость к катастрофам при локальном размещении узлов кластера;
- относительно высокая стоимость реализации.

Общие недостатки построения информационных бизнес систем на основе централизованной архитектуры:

- рост нагрузки на корпоративные сервера, приводит к необходимости достаточно частого обновления серверов;
- необходимость обслуживания серверов приводит к росту совокупной стоимости владения информационными технологиями (ИТ) и др.

Стремление устранить перечисленные недостатки привело к появлению мощных центров обработки данных (data centers) предназначенных для распределенной и удаленной обработки и хранения данных, так называемых облачных вычислений (Cloud Computing) [3 – 5]. Услуги и решения Cloud Computing можно разделить на три категории.

Software as a Service (SaaS) – программное обеспечение как сервис. Под данным определением понимается предоставление доступа к программам, запущенным на серверах, через Web-браузер. В качестве примера можно привести Web-интерфейс к серверам электронной почты, форумы, социальные сети, а также программы, ранее доступные только посредством установки их на компьютер. Известнейшим разработчиком офисных программ, использующих Web-браузер, является компания Google.

Применимо к корпоративным нуждам, аренда приложений дает следующие преимущества:

- низкие первоначальные инвестиции в ИТ;
- оптимизация расходов;
- снижение рисков (сервис-провайдер несет ответственность за бесперебойную работу услуги);
- масштабируемость решений;
- простота поддержки (отсутствие необходимости стандартизации ПО, обучения сотрудников ИТ новым версиям и т.д.)

Platform as a Service (PaaS) – платформа как сервис, ориентирована прежде всего на разработчиков и предоставляет возможность гибкого и широкого выбора настроенных под бизнес задачи виртуальных вычислительных ресурсов и программ для построения бизнес решения. Хорошими примерами могут служить интегрированные среды разработки и выполнения программ Google App Engine, Windows Azure, AptanaCloud.

Infrastructure as a Service (IaaS) – инфраструктура как сервис предназначена компаниям, нуждающимся в мощных вычислительных ресурсах. Виртуальная инфраструктура позволяет сэкономить на аппаратном обеспечении и на услугах ИТ, а также она рассчитана на масштабируемость вычислительных ресурсов. Одной из разновидностей IaaS стала услуга Data Storage as a Service (dSaaS) - хранение данных как сервис. Самыми известными представителями IaaS и dSaaS являются разработки компании Amazon: Elastic Compute Cloud (EC2) и Simple Storage Service (S3), а также продукты других компаний: GoGrid, ElasticHosts и др.

Программную архитектуру Cloud Computing можно разделить на шесть основных уровней [3 – 5]:

- уровень клиента – это клиентское ПО, используемое для доступа к облачным сервисам, например, Web-браузер;
- уровень сервисов – это сами сервисы, используемые через облачную модель;
- уровень приложений – это программы, доступные через облако и не требующие инсталляции на клиенте (одно из главных преимуществ облачной модели);

- уровень платформы – это программная платформа, объединяющая полный набор инструментов для развертывания и использования облачных вычислений без инсталляций на клиенте, например, Microsoft.NET Azure Services Platform;

– уровень памяти – поддержка хранения данных пользователя и доступа к ним через облако;

– уровень инфраструктуры – предоставление полной виртуализированной платформы через облако, например, Amazon EC2.

Основные достоинства Cloud Computing:

– предоставление вычислительных ресурсов по требованию (on Demand) (высокая масштабируемость);

– снижение затрат на приобретение, обслуживание и обновление ПО и техники (экономия средств, вложенных в инфраструктуру);

– распределённость вычислительных ресурсов (в одном месте сложно размещать большие технические мощности);

– независимость от платформы (не требуется обновление программно-аппаратных средств на стороне клиента);

– высокая эффективность совместной работы клиентов (синхронизация данных различных клиентов, доступность с различных технических устройств);

– высокая отказоустойчивость и надёжность хранения данных.

Таким образом, построение информационных бизнес систем на базе сервис-ориентированной архитектуры приложения с размещением программно-аппаратных мощностей в Cloud-среде совмещает достоинства SOA и Cloud Computing.

Общая схема предлагаемой корпоративной ИС представлена на рис. 7.

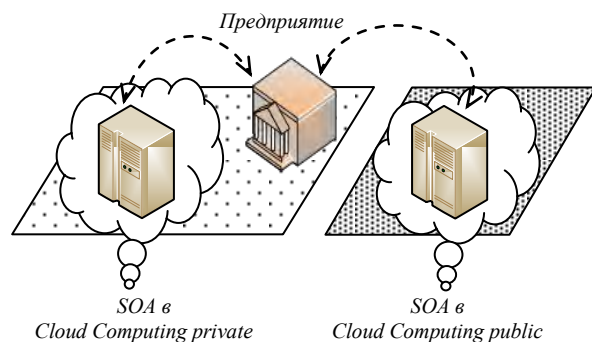


Рис. 7. Построение корпоративной ИС на базе SOA и Cloud Computing

Возможно три варианта использования Cloud Computing на предприятии с точки зрения размещения технических мощностей: *Cloud Computing public* – размещение облака вне предприятия; *Cloud Computing private* – размещение облака внутри предприятия; *Cloud Computing hybrid* – совмещение открытого и закрытого размещения.

Достоинством закрытого (private) размещения всех программно-аппаратных средств внутри пред-

приятия в первую очередь является высокая защита информации от несанкционированного доступа. Однако такая реализация приводит к большим финансовым затратам. Преимущество открытого (public) размещения заключается в минимальных финансовых затратах на реализацию Cloud Computing, а недостаток – наличие непрерывного высокоскоростного доступа к Интернет. Золотой серединой построения корпоративной ИС является развертывания SOA предприятия с гибридной (hybrid) реализацией Cloud Computing. Предложенная реализация предполагает размещение баз корпоративных данных в зоне Cloud private, а исполнение базовых сервисов в зоне Cloud public.

## Выводы

Построение корпоративной информационной системы на базе сервис-ориентированной архитектуры SOA с реализацией в Cloud-среде требует минимальных затрат и дает предприятию следующие преимущества:

– сокращается время на реализацию бизнес проектов;

– ускоряется интеграция, разработка и внедрение приложений;

– снижается риск, связанный с внедрением проектов в область автоматизации услуг и процессов;

– появляется возможность непрерывного улучшения предоставляемых услуг и др.

В отличие от других технологий, архитектура SOA обеспечивает полную независимость от платформы реализации посредством определения набора интегрирующих Web-стандартов, в рамках которых осуществляется взаимодействие, облегчая тем самым процесс межсистемной интеграции и уменьшая трудности взаимодействия для пользователей.

Cloud-среда предоставляет вычислительные ресурсы по требованию (on Demand) тем самым обеспечивает увеличение доступных вычислительных мощностей при минимальных финансовых затратах.

## Список литературы

1. Карпов Л.Е. Архитектура распределенных систем программного обеспечения / Л.Е. Карпов. – М.: МАКС Пресс, 2007. – 130 с.
2. Гладцын В.А. Сервис-ориентированная архитектура. Стандарты, алгоритмы, протоколы / В.А. Гладцын, К.В. Кринкин, В.В. Яновский. – СПб.: СПбГЭТУ "ЛЭТИ", 2004. – 108 с.
3. David S. Linthicum. Cloud Computing and SOA Convergence in Your Enterprise / David S. Linthicum. – Boston: Addison-Wesley, 2010. – 265 p.
4. George Reese. Cloud Application Architectures / George Reese. – Cambridge: O'Reilly, 2009. – 206 p.
5. Introduction to Cloud Computing architecture. – White Paper Sun Microsystems. – 2009. – 40 p.

Поступила в редколлегию 5.04.2011

Рецензент: д-р техн. наук, проф. В.А. Краснобаев, Харь- ковський університет Воздушних Сил ім. І. Кожедуба, Харьков.

### **ІНФОРМАЦІЙНА БІЗНЕС СИСТЕМА В CLOUD-СЕРЕДОВИЩІ**

К.О. Бохан, Д.В. Гриньов

*У статті проведений аналіз основних архітектур клієнт-сервер для побудови корпоративних інформаційних систем. Розглянуті концептуальні рівні і архітектурні ланки розподілених інформаційних систем. Запропоновано підхід до побудови інформаційних бізнес систем на основі сервіс-орієнтованої архітектури з розгортанням в Cloud-середовищі.*

**Ключові слова:** інформаційні бізнес системи, архітектура клієнт-сервер, сервіс-орієнтована архітектура (SOA), Web-сервіс, хмарні обчислення (Cloud Computing).

### **INFORMATION BUSINESS SYSTEM IN CLOUD-ENVIRONMENT**

К.А. Бохан, D.V. Grinyov

*In the article the analysis of basic architectures is conducted client-server for the construction of the corporate informative systems. Conceptual levels and architectural links of the distributed informative systems are considered. Offered approach to the construction informative business of the systems on the basis of сервис-ориентированной архитектуры with development in Cloud-environment.*

**Keywords:** informative business of the system, architecture is a client-server, service-oriented architecture (SOA), Web-service, cloudy calculations (Cloud Computing).