

МЕТОДИ ЗА ПРОЕКТИРАНЕ НА БИЗНЕС ИНФОРМАЦИОННИ СИСТЕМИ НА БАЗАТА НА WEB 2.0 И CLOUD COMPUTING¹

Валентин Кисимов, Димитър Велев*

1. СЪЩНОСТ НА WEB 2.0

Терминът Web 2.0 се отнася за група известни технологии, които надживяват големия провал на Интернет-базираните начинания и фирми през 2001 г, тъй като те продължават развитието на предишното поколение Интернет технологии (Web 1.0), еволюирало в периода 1995–2001 г. Често като по-разширено тълкуване на Web 2.0 може да се срещне определението „социален уеб“. Web 2.0 е термин, отразяващ промяна. Интернет се превръща в платформа, а новопоявяващите се Уеб технологии представляват инструменти за изследване на Интернет. Тази промяна има особено силно въздействие и влияние върху културните, социалните и поведенските взаимоотношения. През последните няколко години се появиха Уеб приложения, които са подчинени на изискванията на обществото и бизнеса, например като YouTube, MySpace и др. Еволюцията на приложения за Web 2.0 не е ограничена само до масовите приложения, но също така навлиза много по-сериозно и дълбоко в корпоративните приложения.

Терминът първоначално е дефиниран от Tim O'Reilly от корпорацията O'Reilly Media [20, 27]. Той описва Web 2.0 като „бизнес революцията в компютърната индустрия“. Независимо от множеството дефиниции за Web 2.0, които могат да се срещнат в специализираната литература, най-общо Web 2.0 може да се характеризира със следните компоненти и особености [21, 22]:

- **Силна обвързаност на множество Интернет потребители.** Повечето Web 2.0 софтуерни приложения разчитат на използването им на множество свързани в глобалната мрежа потребители. Колкото повече потребители са свързани в Интернет, толкова повече ценни стават продуктите и услугите, чрез които те взаимодействат помежду си. Създава се дори специален софтуер, който да управлява колективната интелигентност на множество потребители.
- **Потребителите се превръщат в активни участници,** тъй като те са престанали да бъдат обикновени пасивни читатели на материали в мрежата, а се превръщат в писатели и съзидатели и целенасочено допринасят за увеличаване стойността на използваните от тях софтуер и услуги.
- **Децентрализация.** Независимо от факта, че общуването на множество Интернет потребители може да бъде преднамерено или непреднамерено, комбираното

¹ Настоящата студия обобщава резултати от работата по два проекта по НИД на УНСС – 21.03–11.2009 и 21.03–44.2009.

* Валентин Кисимов е доктор на икономическите науки, доцент в катедра “Информационни технологии и комуникации” на УНСС; тел.: 81-95-203, e-mail: vkisimov@gmail.com; Димитър Велев е доктор по икономика, доцент в катедра “Информационни технологии и комуникации” на УНСС; тел.: 81-95-694, e-mail: dvelev@unwe.acad.bg

въздействие от действията им може да има огромно влияние и въздействие. Същото се отчита и в организациите, където отделните индивиди могат да допринесат за увеличаване на общото ниво на експертност на служителите без да е необходимо спазването на строги йерархични правила и изисквания на работните потоци.

- **Отвореност.** Множество компоненти на Интернет, като например данни и софтуер, се предлагат на потребителите, които желаят да ги използват почти безплатно, а в много случаи и напълно безплатно. Все повече организации могат да споделят своите активи, използвайки отворени технологични стандарти.
- **Възможност за многократно съчетаване на различни софтуерни компоненти и технологии.** Поради факта, че Web 2.0 технологията е свободна и отворена, разработчиците на софтуер могат да използват компоненти, създадени от други разработчици.
- **Край на цикъла за предоставяне на софтуер.** Софтуерът се предоставя като услуга и трябва да се поддържа, преглежда и подобрява ежедневно. Това обуславя необходимостта потребителите да се разглеждат като съзработчици. Следователно възможността за многократно използване на софтуерни компоненти, уеб услуги и mashup играят важна роля.
- **Свобода на използването.** Докато по-рано софтуерът се състоеше от предварително дефинирани набори от действия и процеси, то Web 2.0 софтуерът позволява осъществяването на по-свободни структури и се разчита повече на крайните потребители, а не на разработчиците, за начините за неговото използване.
- **Софтуерът е над нивото на единично устройство.** Приложенията са независими от устройствата, които осъществяват достъп към тях. В частност, мобилните приложения не са приложения с намалени възможности.
- **Наситеност на преживяванията при работа.** Съвременните уеб сайтове са изпълнени с разнообразни фотоматериали, анимации, мултимедия, аудио и видео съдържание, сложни визуални среди, които още по-силно приближават потребителите към поведение и изживяване, съответстващи на тези в естествена за човека среда. Многофункционалните интернет приложения (RIA) са важни за удовлетвореността на потребителите, тъй като те пренасят възможностите на приложенията за настолна среда в средата на браузърите.
- **Web 2.0 е платформа.** Само преди няколко години софтуерът, изпълняван автономно в настолните компютри, беше много по-бърз и богат на функционалности от софтуера, изпълняван в браузърите. Изпълнението на софтуер в Web 2.0 обаче протича много по-гладко сега и в голяма степен се отъждествява с настолния, поради която причина все повече разработчици на софтуер реализират своите разработки за работа изключително в Интернет среда.

Web 1.0 е статичен и по-скоро е среда, в която може да се намери информация, отколкото да е форум за размяна на идеи или място за съвместно създаване на нови продукти. Web 2.0 е среда в Интернет, в която потребителите могат активно да взаимодействат и участват в различни процеси, а не само пасивно да четат публикувани от други участници материали.

Таблица 1

Разлики между Web 1.0 и Web 2.0 [23, 25]

	Web 1.0	Web 2.0
Действия	Четене	Четене и добавяне
Основно съдържание	Web страница	Публикувано съобщение / Запис
Състояние	Статично	Динамично
Инструменти за работа	Браузъри	Браузъри, RSS – четци
Архитектура	Клиент – Сървър	Уеб услуги
Автор на съдържанието	Уеб разработчици	Всички
Обитател на средата	Специалисти	Всички

Еволюцията на Web 2.0 приложения се управлява от редица ключови фактори [19, 21]:

- *Социални изисквания* – при двупосочните комуникации в Интернет всички са едновременно потребители и доставчици на информация, на които е необходим идеален инструмент за взаимодействие и изпълнение на няколко различни дейности като например четене на новини, изпращане и четене на поща, преглед на борсови показатели и др., които всички произтичат от едно място. Това изисква обединението на множество източници и безпроблемното им общо споделяне по удобен начин.
- *Изисквания на пазара*. Пазарите се развиват във всички промишлени сектори, изискващи взаимодействие на ниво бизнес-към-бизнес. Това изисква преминаването към нови технологии и предоставянето на Уеб услуги.
- *Конкуренция*. Конкурентите използват нови приложения, които са предназначени за работа в Web 2.0 среда, принуждавайки останалите участници да се пренасочат към такива условия на конкурентна борба.

Основните Web 2.0 приложения могат да се сведат до следните основни групи [20, 21, 26]:

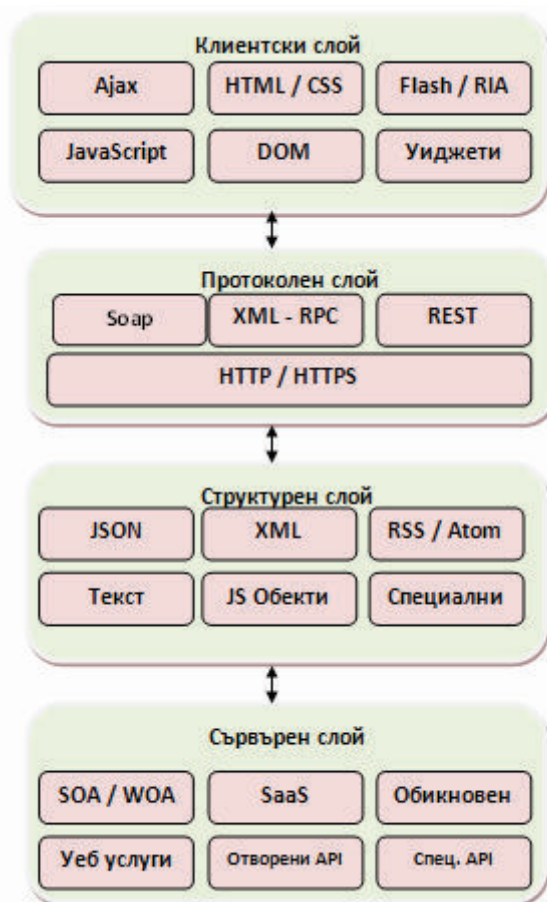
- *Блогове (blog)* – представляват опростени Web базирани издателски системи, които позволяват на хора, които не притежават специални технически умения, да създават собствени Уеб сайтове под формата на личен дневник. Основна характерна черта на този софтуер е, че позволява на читателя да отговаря на създателя на блога. Обикновено блогът се концентрира около определена тема (лична, професионална или др.) и натрупаната там информация по формата на общуване между автора и читателите може в последствие да се използва с определена цел (например, разрешаване на определен професионален проблем).
- *Социални мрежи* – специални сайтове, които позволяват създаването на лични профили, общуване с приятели, създаване на определена идентичност, обмяна на частни съобщения, публикуването на информационни бюлетени.
- *Обмен на видео и фотоматериали* – сайтове, които позволяват лесното публикуване и обмен на видео и фото информация.

- *Уики (wiki)* – Уеб сайтове, изградени със специално предназначение, при които група от децентрализирани потребители, свързани чрез Интернет, могат съвместно да работят като една голяма йерархично структурирана организация.
- *Онлайн карти* – Уеб сайтове за публикуване на географски данни под формата на карти. Картите могат да разкрият мощни шаблони и взаимовръзки, които по-рано са били скрити в огромните обеми данни.
- *Виртуални светове* – триизмерни представления на реалността и фантазията. Въпреки че първоначално могат да изглеждат като игри, те са места, в които се разкрива истинския социален живот. Там хората закупуват собственост, строят домове, срещата се с други хора, осъществяват бизнес и на практика живеят там. Всички тези Web 2.0 приложения, реализирани чрез съответни Уеб сайтове, имат следните общи характеристики [20, 25, 27]:
- *Инструменти за опростено публикуване* – Уеб сайтовете за споделяне на информация елиминират почти всички затруднения и спънки от техническа гледна точка като форматиране, зареждане и кодиране на Уеб страници за извеждане на съответната информация за споделяне, както и специални технически умения.
- *Социални елементи* – Уеб сайтовете предлагат множество методи за споделяне на информация и нейното обсъждане, като зони за коментар и средства за изпращане на информация в различни формати към приятели.
- *Публикуване в персонални сайтове*. Осигурени са опростени методи за публикуване под формата на профили в социални мрежи, блогове, уики и др.
- *Ниска себестойност* – повечето Уеб сайтове са безплатни или имат специален план, предвиждащ и безплатно използване на част от възможностите му.

Web 2.0 представлява съвкупност от множество различни нови технологии. От своя страна, те са причината за създаване на приложения от по-ново поколение. Появата на архитектурата на Web 2.0 е свързана с тези технологии. Те са разположени върху различни логически разделени слоеве, които са показани на фиг. 1.

Клиентски слой. По същество този слой е слоя на Web браузърите, които за крайните потребители представляват входната точка към Интернет. От гледна точка на Web 2.0 в този слой се създават условията за наситеност с многоформатни носители на информация за крайния клиент. Технологиите в този слой са комбинация от стари технологии и някои нови елементи. Ajax и Flash са основните компоненти за Web 2.0 приложенията. Тези технологии се вграждат в into HTML, JavaScript, Document Object Model (DOM) и Cascading Style Sheets (CSS).

Протоколен слой. На базата на HTTP технологията Web 2.0 въвежда някои нови протоколи в този слой, които поддържат технологиите клиент – сървър. Web 2.0 приложенията използват HTTP като среда за пренос на различните протоколи. Едно от основните ограничения на HTTP е, че потоците не могат да се изпращат от браузъра към сървъра. HTTP използва GET и POST за предаване на информация към сървъра. Тази информация трябва да се изпраща като двойка „име-стойност”. Това ограничение е елиминирано с технологията Ajax, която позволява информационните потоци да се предават от и към сървъра. Както се вижда от фигурата SOAP, XML-RPC, and REST представляват основни компоненти на протоколния слой.



Фиг.1. Технологична архитектура на Web 2.0

Структурен слой. Информационните структури са важни компоненти на комуникационните канали. За разлика от предишните приложения, които използват само чист HTML, Web 2.0 използва много по-добри и ефикасни структури. Web 2.0 приложенията използват компонентите на многофункционални приложения, като например Flash и Ajax, с цел осъществяване на предаването и получаването на не-HTML структури в мрежата. Тази възможност предоставя няколко нови начина за осъществяване на достъп до информация и изпращането ѝ към сървър. Този логически слой на Web 2.0 приложения се запълва с множество различни структури, съдържащи данни и информация, които се изпращат към приложението и които се получават от браузъра. JSON, RSS, XML са основни компоненти в този слой.

Сървърен слой. В този слой Web 2.0 въвежда няколко нови технологии за усиляване на мрежата в качеството на платформа, както и за осъществяване на поддръжката на взаимодействието между приложенията. Сървърният слой поддържа всички клиенти от ново поколение, използващи Ajax и RIA, протоколите XML-RPC, SOAP,

както и структурите `such JSON` и `JS-Objects`. Съществуват няколко нови начина и технологии за изграждане на Web 2.0 приложения от гледна точка на сървъра. С въвеждането на Ajax и многофункционалните интернет приложения голяма част от разработването на софтуер е прехвърлена към брауъра, като повече внимание се отделя върху бизнес логиката. За да се изпълни тази задача, приложната архитектура е изместена към архитектурата, ориентирана към услуги (SOA). Технологията SOA е един от ключовите елементи на цялостната архитектура. SOA осигурява различни набори от Уеб услуги, които могат да се използват от съответния брауър или от друго приложение. Уеб услугите се изпълняват в приложна сървърна среда и могат да осъществяват достъп до бази данни или до други критични сървърни компоненти. Освен това тези услуги могат да осъществят достъп през Интернет до приложения, предоставяни от трети страни, като по този начин допълнително подпомагат обединението на различни приложения в едно място.

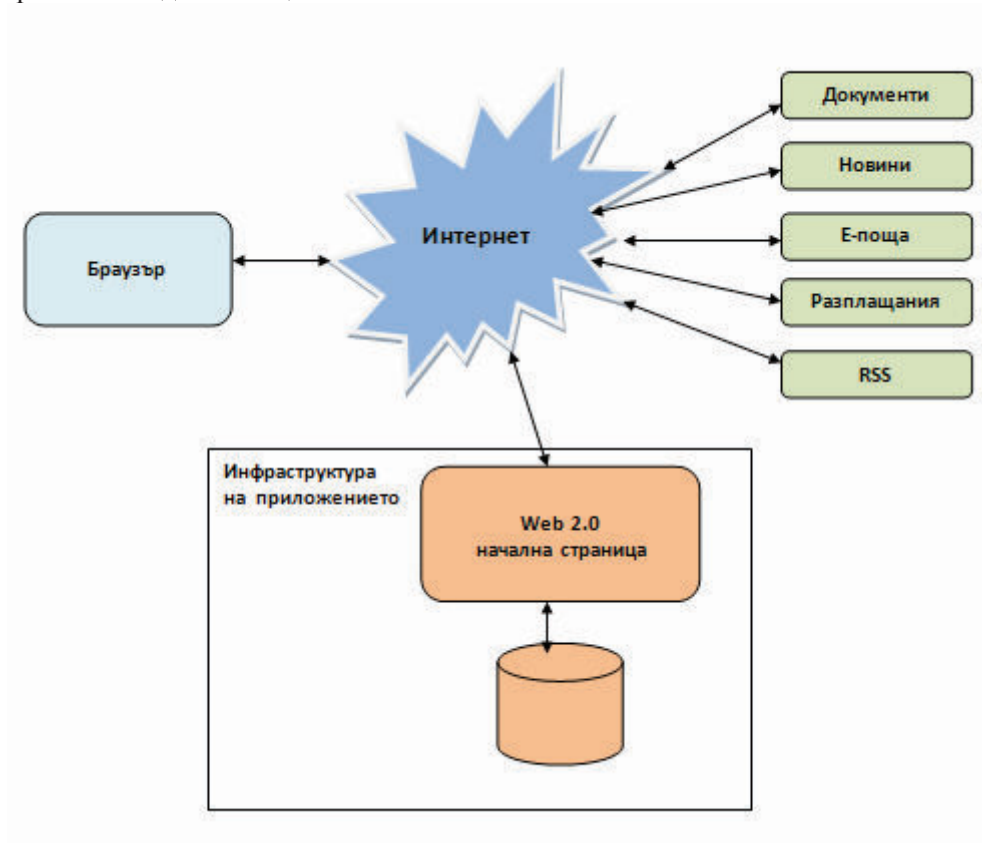
На базата на този модел могат да се дефинират някои от най-важните предимства на архитектурата на Web 2.0 [21, 25, 26]:

- Ajax и Flash предоставят методи за асинхронна комуникация, така че на крайния потребител не се налага да изчака за опресняване и презареждане на съответните страници. Тези методи за асинхронна комуникация правят целия процес на брауване многозадачен и многонишков.
- Многофункционалният клиентски интерфейс удовлетворява повечето потребности от настолно приложение. Брауърът може да играе ролята на настолен компютър за тези нови приложения.
- Постига се ефективно осъществяване на комуникации чрез опростени, гъвкави и леки информационни структури.
- Универсално възприетите протоколи на базата на XML като SOAP, XML-RPC и REST могат съществено да улеснят комуникациите между различните нива.
- Web услугите и SOA осигуряват механизъм за комуникация между различните приложения и позволяват подаването на данни и информация към отделни приложения. Това спомага за създаването на mashup приложения.
- Осъществява се възможността за комуникация от брауъра или Уеб приложението при известна крайна точка за изпълнение от дадено приложение.

Една от основните разлики между Web 2.0 приложения и приложенията от по-старата генерация е използването на информационни ресурси. Web 2.0 приложенията се основават на заложените технологии и приложен програмен интерфейс (API), които се поддържат от различни други приложения. Това позволява на приложенията да използват информационни ресурси, които се намират в други сървъри, като ефективно улавят и представят съответната информация на крайния потребител. На фиг. 2 е представена примерна начална страница на Web 2.0 приложение [22].

Приложението разполага със собствен сървър за база данни. При достъп на крайния потребител към началната страница посредством брауър, приложението зарежда няколко компоненти, създадени на базата на Ajax и Flash, в брауъра, което позволява на потребителя да осъществи достъп до данните през една единствена страница. От своя страна, началната страница на приложението осъществява достъп до няколко Интернет информационни ресурса с помощта на протоколите

SOAP, XML-RPC, REST. Чрез тези протоколи приложението може да осъществи достъп до информацията на потребителя относно неговите банкови активи, време, документи, новини и т.н. Цялата тази информация се преобразува в единична страница и на крайния потребител не е необходимо да се обръща към други приложения. Цялата обработка се извършва посредством mashup, при което мрежата играе ролята на платформа, а съответните приложения играят ролята едновременно на потребители и доставчици.



Фиг. 2. Информационни потоци в Web 2.0 приложение.

Разбира се Web 2.0 не е лишен от недостатъци, като най-големият от тях е свързан с проблемите на сигурността на Web 2.0, които произтичат от особеностите на тази нова архитектура и съответните й слоеве и компоненти [24, 27]:

- **Клиентски слой** – браузърите представляват традиционно място за атака от страна на различни нарушители и вируси. Целта е да се открадне ценна персонална информация като например кукита (cookie). Нарушаването на защитата на браузъра е нова заплаха и уязвимите Уеб приложения, които обслужват Ажак приложенията могат да представляват слабо място в системата. Приложенията, разработени с Flash, са също изложени на риск поради възможността за обратно проектиране на кода.

- **Сървърен слой** – Web 2.0 приложенията използват интензивно XML и производните му протоколи като XML-RPC, SOAP и REST за пренос на съобщения и е възможно да бъдат подложени на някои от съществуващите атаки, но по нов начин, в които случаи механизмите за подаване на информация могат да се променят. Това най-вече може да се дължи на някои вътрешни проблеми или лоша реализация на тези протоколи.
- **Информационни ресурси и обработка** – Web 2.0 приложенията използват информационни ресурси с или без необходимата степен на защита – блогове, RSS и електронна поща. Съответното съдържание се обработва в сървър или в браузър, което може да доведе до потенциален проблем.
- **Информационна структура** – информационните структури са критични компоненти на Web 2.0 приложенията. Тези приложения включват RSS, Atom, XML, JSON или други структури, които могат успешно да бъдат атакувани. Особено внимание при създаването на съответните приложения трябва да бъде обърнато на механизмите за обмен на информационни структури, информационните източници и обработка.
- **SOA и Уеб услуги** – Уеб услугите, които са ключов компонент на SOA, се предлагат от фирмите и организациите за съвместно споделяне на критична информация с клиенти или с други потребители и представляват нова входна точка към приложна инфраструктура. Тези услуги, заедно с изброяването им, представляват потенциална заплаха, особено ако са неподходящо проектирани, тъй като атаката може да доведе до пряк достъп до базите данни и всякаква друга информация, която се намира в сървъра.

2. МЕТОДИ ЗА ПРОЕКТИРАНЕ НА БИЗНЕС ИНФОРМАЦИОННА СИСТЕМА НА БАЗАТА НА WEB 2.0

Основната идея на бизнес информационните системи на базата на Web 2.0 е способността им да събират и споделят информация от и между различни потребители.

Преди да се пристъпи към проектиране на бизнес информационна система на базата на Web 2.0 трябва да се изгради съответния бизнес план за работа в Web 2.0 среда. Той може да включва следните основни компоненти [19, 23, 27]:

- **Анализ на пазара** – основните въпроси са свързани с това колко бързо потенциалните клиенти ще реагират на новата услуга и колко време ще им отнеме да я изпробват и започнат да я използват.
- **Анализ на конкуренцията** – поради факта, че Web 2.0 бизнес моделът предполага съвместна дейност между потребителите, то трябва да се съберат подробни отчети за това как фирменият модел създава, улавя и разпределя стойност от множество други източници (като колективна потребителска стойност; взаимодействие на базата на P2P (peer-to-peer) мрежи със свободно съдържание и общности, споделящи съдържание; стойност от социални мрежи; екосистеми, които генерират допълнителни взаимоотношения и отношения между трети страни, реализиращи модел бизнес към бизнес.
- **Продукти и услуги** – тъй като генерираното от потребители съдържание често

много трудно може да се прогнозира, трябва да се дефинира как неговият материал трябва да се интегрира в съответната система; как да се лицензира и разпределя това съдържание; какво се полага на потребителя като възнаграждение и др.

- **Времеви показатели** – трябва да се дефинира количеството съдържание, което даден потребител може да генерира; скоростта на генериране; колко често трябва да се променя приложението, с което се обработва генерираното съдържание и т.н.
- **Ръководство** – умения за съвместна работа като взаимодействие, бърза реакция, вслушване и др., служещи като основа на динамичните възможности за съгласуване и управление на големи онлайн съобщества и групи. Уменията излизат на преден план като предпоставка за управление на Web 2.0 приложения с много потребители.
- **Финанси** – определяне на стойността на всеки единичен потребител, стойността на общността потребители и стойността на екосистемата, които са от съществено значение за определяне целесъобразността от въвеждане на Web 2.0 приложения.

Винаги трябва да се помни, че много Web 2.0 проекти могат да се изградят на базата на ниски инвестиции, но възвращаемостта на вложените средства може многократно да надвишава инвестициите.

Създаването на нов софтуер от самото начало ще се превръща във все по-рядко срещано явление, тъй като е по-лесно да се комбинират съществуващи вече модули и сегменти програмен код. Това е пряко следствие от обектно ориентирания подход за създаване на софтуер, при който енкапсулирани програмни сегменти могат да се използват многократно в целия жизнен цикъл на създаване на информационни системи. Архитектурата, ориентирана към услуги, разширява тази тенденция и открива нови възможности, преодолявайки функционални и технически граници на действие.

Web 2.0 е възможност за изграждане на бизнес информационни системи на базата на Web базирани софтуерни услуги, чрез които може да се изгражда или нов Web 2.0 софтуер, или пък които могат да се вграждат в собствените информационни системи на фирмите и организациите. Многократното прилагане на Уеб и mashup услуги предоставят възможности за пълноценно изграждане на бизнес информационни системи. Уеб сайтовете на такива информационни системи вече не представляват интерфейс за работа с единствен потребител, а отварят своята функционалност и данни за всички, които биха ги използвали като собствени. Това позволява на потребителите многократно да пренастройват и използват чужда функционалност като собствен софтуер по свое желание и за своите цели. Бъдещето на бизнес информационните системи на базата на Web 2.0 ще се свежда до комбинирането на предоставяните и съществуващи Уеб услуги в нови, иновативни по характера си приложения.

За да могат тези услуги да се предоставят за използване, трябва да се създаде съответен набор от приложен програмен интерфейс (API), в който процес трябва да се дефинират подходящи методи за тяхното създаване и поддържане [21, 23]:

- **Простота и мащабируемост** – опростените интерфейси предоставят облекчена поддръжка и мащабируемост в сравнение със сложни системи, изградени на множество нива, които трябва да гарантират предаването на съобщения през множество слоеве преди да се изведе заключение и вземе решение за тяхното използване. Интерфейсите, които се основават на HTTP протоколи, както и на техните производни (например, REST), могат да се окажат твърде подходящи за относително дълъг период на използване, но не винаги са най-подходящите. Съществуват P2P архитектури, с голям коефициент на мащабируемост, които изискват собствени протоколи и много по-сложни информационни структури. Облекчаването на информационната структура на корпоративните системи се превръща в задължително условие със засилващата се тенденция за размиване на границите между бизнес информационните системи, Интернет и мобилните устройства. Мащабируемостта е най-критичната точка за Web 2.0 базирани бизнес информационни системи, тъй като трябва да се изградят планове на много ранен етап за радикално децентрализиране
- **Простота на използване** – обикновено разработчиците предпочитат работа с данни, които вече са обработени по някакъв начин. Текстовите формати са доказали своите предимства в изградени информационни системи. В Интернет среда понякога HTML код може да се окаже полезен. XML предлага по-ясна структурираност и читаемост, а най-лесен за интегриране е форматът AMF (Action Message Format) за многофункционални интернет приложения, изграждани чрез Flash и Flex. Решения с протоколите WS* и SOAP могат да се окажат подходящи, ако се изисква тяхното прилагане специално от потребителите на съответните информационни системи.
- **Сигурност** – въпреки че най-критичните решения могат да се обезпечат чрез използването на SOAP-базирани методи, не са за пренебрегване десетки съществуващи решения на базата на доказали се средства за криптиране на информацията. Осигуряването на необходимото ниво на сигурност увеличава както общата работа по обработката на данните, така и информационния трафик, което води до допълнителни въпроси относно интеграцията от клиентска страна.
- **Инструментална поддръжка** – огромното множество програмни библиотеки с широка функционалност и софтуерни инструменти могат успешно да спомогнат за свързването на клиенти, сървъри и информационни системи. Особено разпространени са такива на базата на XML, както и непрекъснато се увеличава поддръжката на REST.

Методите за изграждане на Web 2.0 базирани бизнес информационни системи се основават на новия клиент-сървър модел за Web 2.0. Шаблоните за моделиране на взаимодействието [21, 22, 25] в такива системи са много по-сложни от взаимодействието по обикновения шаблон заявка – отговор на първоначалния модел при Web 1.0. Еволюцията на стария модел в петслоен модел за Web 2.0 бизнес информационна система може в бъдеще да бъде продължена с добавянето на нови шаблони и функционалност, фиг. 3.

Този модел отразява начините, по които Web 2.0 свързва потребителите и възможностите в множество реализации. Един вариант на такава реализация може да

представлява корпорация или организация, в която корпоративната информация се споделя с потребителите, а друга различна реализация може да представлява P2P мрежа.



Фиг. 3. Модел за архитектура на Web 2.0 бизнес информационна система

Слоят „Възможности” дефинира функционалността, която има реално измерение. Не е необходимо тази функционалност да бъде разположена на компютър и моделът не е свързан с определена реализация. Възможностите могат да са притежание или да се предоставят от държавни учреждения, различни фирми или от отделни лица, предоставящи функционалност, която те желаят да споделят.

Слоят „Услуги” осигурява използването на предлаганите услуги чрез общ набор от протоколи и стандарти. От потребителите на услугите не се изисква да познават как са изградени тези услуги. Услугите притежават модел на данните, а данните се подават или приемат по време на взаимодействието с потребителите. Архитектурният модел, който е основата на този слой, е архитектурата, ориентирана към услуги (SOA). Въпреки че SOA представлява основна концепция при проектирането на бизнес информационни системи на основата на Web 2.0, SOA е и способ за организиране и използване на разпределените услуги, които могат да са под управлението на различни притежатели. SOA осигурява мощна рамка за идентифициране на услуги, които трябва да обслужват определени потребности, и за съгласуване на услугите и потребителите им.

Взаимодействието може да се осъществява на базата на два вида действия:

- **Публични действия** – водят до изменения в състоянието на текущо изпълняваните процеси;
- **Частни действия** – не са познати за другата страна, въпреки че могат да се стартират чрез публично видими интерфейси на услугите. Поддръжката за частните действия е една от основните характеристики на услугите. Услуга, изпълняваща

частни действия или функции, които потребителите не могат да видят, скрива реализацията на възможностите ѝ от потребителите. Тази непрозрачност осъществява при необходимост смяна на заложените технологии без да се окаже въздействие върху слоя на потребителите.

Слоят „Свързаност/Достъпност” трябва да осигурява определена степен на съгласуваност между различните набори протоколи, стандарти и технологии.

Слоят „Клиентски приложения/Софтуер” трябва да отразява и реализира концепцията за преминаване на взаимодействието в модела клиент-сървър към онлайн и офлайн режим на работа с постигане на висока степен на интерактивност чрез многофункционални Интернет приложения. Множество клиентски приложения и софтуер могат да се реализират като хибридни приложения – комбинации от настолни и интернет приложения.

Слоят “Потребители” трябва да отразява потребителя като стереотип, който включва различни реализации. Потребителят трябва да се разглежда както като консуматор на услуги и съдържание, което информационната система предлага, така и като доставчик на услуги и съдържание, които система консумира. Освен това потребителите могат съвместно да използват и споделят резултатите от взаимодействието с други потребители и с това да се осъществи колективната интелигентност на потребителите. Конкретните начини за постигане на тази интелигентност могат да се сведат до следните позиции [20, 25, 27]:

- Потребителите трябва да се разглеждат като източник на данни, които по други начини трудно могат да се събират.
- Съсредоточаване и събиране на колективната интелигентност от безкрайните информационни ресурси в Интернет, чието съдържание трябва да се анализира, класифицира и разпредели.
- Генериране на мащабен мрежови ефект вследствие на колективната интелигентност на потребителите в Интернет.
- Осигуряване на фолксономия – самоорганизирането на потребителите гарантира възможността за предоставяне и използване на съдържанието в сайтовете, или социалният софтуер да бъде използван по много по-добър начин.
- Създаване на обратен филтър за интелигентност – хипервръзки и други указатели към информационни ресурси и съдържание могат да се съберат и анализират, за да се определи какво е важно и ценно без по-нататъшно вмешателство и обработка от страна на потребителите. Комбинирането с различни филтри и техники може да доведе до създаването на сложни оценяващи механизми.

По своята същност шаблоните са повтарящи се решения за повтарящи се проблеми и те представляват фундаментална структурна организационна схема за софтуерни системи. Шаблонът предоставя набор от предварително дефинирани подсистеми, определя техните задължения и включва правила и инструкции за организиране на взаимодействията между тях. Шаблоните са съставени от три основни елемента [22, 23]:

- **Проблем** – набор от всички причини и обстоятелства, довеждащи до повторение на действия. Изисква се подробно описание на проблема съгласно определени изисквания. Допълнително могат да се дефинират изискванията, органиченията

и необходимите действия за осигуряване на решение. Възможен пример за проблем може да бъде изграждането на Web 2.0 бизнес информационна система, която да събира и поддържа в актуално състояние личните и финансовите данни на всички свои клиенти.

- **Съдържание** – условията, при които възниква проблемът. Съдържанието може да бъде обобщено с цел гарантирането на максимална приложимост на шаблона или пък може да бъде силно специализирано при определени специфични условия.
- **Общо решение** – удовлетворява повтарящ се бизнес проблем. Такова решение има статична и динамична част. Статичната част може да се изрази чрез езици за описание на архитектурата, включващи прости схеми. Динамичната част указва как обектите и компонентите взаимодействат помежду си.

Чрез съществуващите шаблони могат да се дефинират най-общо следните методи за проектиране на бизнес информационни системи на базата на Web 2.0 [20, 21, 23, 27]:

Метод за проектиране чрез шаблон SOA

SOA представлява архитектурна парадигма, която се използва за организиране на възможностите и функционалностите, управлявани от различни притежатели и която позволява различни видове взаимодействие между потребителите на дефинираните възможности. Също така може да се дефинира като абстрактна граница на действията, която улеснява взаимодействията между потребители с нужди и потребители с възможности. Използването на тази стратегия преобразува основната функционалност, която се споделя между потребителите, в непрозрачни услуги. Непрозрачността изолира потребителя на услугите от вътрешните детайли как те са изградени, като същевременно разхлабва силната връзка между компонентите на системата. При работа със SOA трябва да се разрешат основните проблеми, свързани с протоколите и стандартите, сигурността и управлението. SOA инфраструктурата е много гъвкава и нови допълнителни услуги могат относително леко да се добавят. Системата няма да прекъсне своята работа, ако определено приложение трябва да се замени при условие, че новото приложение поддържа съществуващия интерфейс на услугите. SOA архитектурата позволява елиминирането на зависимостите между системите и дублирането на системите може в значителна степен да се намали, ако услугите се пренастроят за потребностите на множество клиенти.

Метод за проектиране чрез шаблон SaaS

Шаблонът SaaS (Софтуер като услуга) може да се използва като метод за проектиране в случаите, в които се изгражда информационна система, чиито модел, управление или текущи резултати от обработката на данни трябва да се актуализират съгласно динамични изменения или в случаите, при които трябва да се генерира специална функционалност в приложението. SaaS е модел за предоставяне на функционалност, а не на доставка на софтуер. Повечето функционалност може да се

предостави по Интернет, така че потребителят да взаимодейства със съответното приложение с цел получаване на желаната функционалност, без необходимост от инсталиране на софтуера в собствената му система. Методът може да се приложи, когато потребителите се нуждаят от повече компютърни ресурси, отколкото те могат да поддържат в своите локални информационни системи в онези случаи, в които потребителите се нуждаят от специфични изчислителни възможности.

Метод за проектиране чрез шаблон Участие-Съвместна работа

Методът позволява реализирането на платформа за съвместна работа, с помощта на която крайните потребители или приложения могат да подават знания, програмен код, факти или друг материал, който е свързан с определена тема. Може да се оперира с произволен тип съдържание, като методът не е ограничен само до текст, подаван от Уеб издателски системи и уики. Могат да се обработват и други информационни формати като аудио и видео материал. Платформите за динамично публикуване могат да осигурят на притежателите на съдържание възможност да включват такова, което е предоставено от потребителите им. Към съществуващите системи могат да се причислят блоговете, блики (блогове с уики поддръжка), блокове (съвместно написани книги), моблогове (мобилни блогове) и вблогове (видео блогове).

Метод за проектиране чрез шаблон mashup

Методът използва приложение, което да зарежда определена карта и шаблон в системата на потребителя при отправена заявка. След това приложението открива отдалечено разположени информационни източници, функциониращи в онлайн режим, и предоставя на потребителя бутони за управление, чрез които той да обработи наборите данни в съответствие с изискванията на картата. Данните се извличат съгласно съществуващи Уеб базирани протоколи и стандарти за свързване към услуги, които подават данни. Наборите данни могат да бъдат кеширани локално при клиента с цел осигуряване на по-ефективно действие в клиентската част. Mashup позволяват адаптирането им и по този начин могат да се пригодят да обслужват множество приложения. Някои от тях позволяват на потребителите да постигат пълно управление върху потребителския интерфейс, така че да бъдат показани само онези детайли, които представляват интерес.

Метод за проектиране чрез шаблон Синхронизиран Уеб

Синхронизираният Уеб обработва ситуациите, при които определено приложение разполага с локален компонент (разположен в определен компютър) и компонент в свързан с мрежа сървър, и от двата компонента се изисква изпълнението на определена функционалност в един и същ момент. Методът може да се използва и в случаите, при които информацията съществува в множество точки и е с възможност за изменение. Ако дори само в едно физическо местоположение информацията се

промени, другите копия на тази информация трябва своевременно да се синхронизират. Терминът синхронизация не се използва по начина, който е приет в терминологията на традиционните бази данни, а по-скоро за целите на Web 2.0 и се отнася за репликацията.

Метод за проектиране чрез шаблон Структурирана информация

Същността на този метод е в прилагането на определен синтаксис за означаване на данните в съответствие с правилата на специфичен речник, синтаксис и таксономия. Езикът за маркиране (Markup Language) може по такъв начин да маркира съдържанието, че то логически да бъде разбито на по-малки сегменти, всеки един от които да е обозначен с етикет, позволяващ идентифицирането на фрагмента съдържание. Фрагментите трябва да бъдат по-малки от HTML страници. В идеалния случай синтаксиса на такъв език трябва да бъде сходен със стандартните езици, а, от друга страна, твърде опростен, за да може да бъде възприет от потребители и разработчици. Таксономията ще използва езика за създаване на речници за специфични цели. Таксономията ще включва различни тагове, имащи съответните анотации, което ще подпомогне другите потребители да разберат тяхната семантика и предназначение.

Метод за проектиране чрез шаблон Потребителска многофункционалност

За реализиране на потребителска многофункционалност разработчиците на бизнес информационни системи трябва да моделират процесите, които текат и се използват в реалния свят, и да създадат работни потоци, които да улеснят сходни и успешни взаимодействия с уеб сайтове и приложения. Тези процеси могат да имат различни начални и крайни състояния, както и различни точки за вземане на решения. Даден процес може да започне със заявка от потребителя за осигуряване на достъп до определен Уеб сайт. В такъв случай разработчикът трябва да узнае максимално добре предпочитанията и бъдещите действия на потребителя преди той да премине на следващата стъпка. Потребителската многофункционалност ще трябва да включва сложна система, която подава смислена и навременна информация към потребителя на базата на събития, от които той е заинтересован. Реализацията на информационната система ще включва моделиране на цялостната система, включително елементите за случай на употреба, схеми за последователността от действия, модели на данните и т.н., които се използват в приложението. Трябва да се възприемат единен синтаксис и система за моделиране при проектирането на интерфейса и взаимодействията.

Понастоящем в процес на проектиране, внедряване, използване и усъвършенстване са множество Web 2.0 базирани бизнес информационни системи, по важните от които са [22, 25, 27]:

- **Enterprise 2.0 – Корпорация 2.0**, обхваща социалния софтуер, използван в дадена корпорация, фирма или организация в смисъла на осъществяване на съответните бизнес дейности. Включва социални и мрежови модификации на фир-

мения Интранет и други класически софтуерни платформи, използвани от множество потребители за организация на комуникациите помежду им. Функционалността се свежда до такива основни компоненти като: а) търсене, позволяващо на потребителите да търсят други потребители или съдържание; б) връзки за групиране на определени потребители или съдържание; в) публикуване, включващо блогове и уики; г) разширения, осигуряващи публикуване на съдържание на базата на определени профили; д) сигнали, позволяващи на потребителите да се абонират за информационно съдържание с помощта на RSS.

- **Identity 2.0 – Идентичност 2.0**, управлява поведението на потребителите в Интернет – дефинира начините, по които те се идентифицират и получават достъп до мрежови информационни ресурси, осигурява защита на идентичността им, технологиите, гарантиращи тяхната защита като мрежови протоколи, цифрови сертификати, пароли, авторизация, автентикаци и т.н. Този процес обхваща три компонента: а) чиста идентичност – създаване, управление или изтриване на идентичност; б) достъп за потребителя; в) услуги – системи, които предоставят персонализирани услуги на потребители и съответните им устройства.
- **Government 2.0 – Правителство 2.0**, е неологизъм, отразяващ опитите за прилагане на социалните мрежи и интегрирането на Web 2.0 в услуга на правителството. Правителство 2.0 е система за по-ефективно предоставяне на услуги от страна на държавата за отделните граждани и за бизнеса. Интегрира такива инструменти и технологии като уики, разработка и внедряване на социални мрежови сайтове на държавата, включва използването на блогове и др. Системата спомага на правителството да предоставя информация на гражданите по начин, който се оказва полезен за потребителите. Голяма част от предоставяната информация е структурирана по такъв начин, че да се обработва с mashup.
- **Democracy 2.0 – Демокрация 2.0**, е система, позволяваща на гражданите да внесат своя принос в създаването на нови закони посредством Web 2.0.
- **Office 2.0 – Офис 2.0**, представлява група синхронизирани приложения, които осигуряват една и съща функционалност на потребители, които съвместно работят върху един и същ документ. Документите и приложенията, които се използват за създаването и показването им, в повечето случаи оперират в онлайн режим и всички потребители разполагат с актуалното състояние на документа.
- **Travel 2.0 – Пътешествие 2.0**, представлява разширение и адаптиране на концепциите на Web 2.0 за трансформиране на онлайн предложенията за отдых и туризъм до нова функционалност, изразяваща се в осигуряването на по-голяма прозрачност, съвместна работа, скорост и предсказуемост.
- **Library 2.0 – Библиотека 2.0**, управлява процесите по двупосочен информационен обмен между потребителите на библиотеки и самите библиотеки като такива.
- **Media 2.0 – Медия 2.0**, представлява реализацията на множество модели, осигуряващи предоставянето на съдържание от вестници, списания и други печатни издания. Тази технология по същество обединява много елементи като демокрация, разпределено предоставяне, идентичност, контекстуализация и др.
- **Advertising 2.0 – Реклама 2.0**, е подход за мащабируема Web 2.0 базирана рек-

ламна дейност. Тя реализира множество базови концепции на Web 2.0 – идентифициране на потребителите и обединяване на колективната интелигентност. Дори базовите модели на телевизионна реклама се променят от последователно излъчване на реклами към паралелното им излъчване със съответното телевизионно предаване.

- **Music 2.0 – Музика 2.0**, представлява революционен начин за разпръскване и слушане на музикални произведения, обединяващ и реализиращ повечето концепции и технологии на Web 2.0.

3. СЪЩНОСТ НА CLOUD COMPUTING (КОМПЮТЪРЕН ОБЛАК)

Терминът „Cloud computing”, който може да се представи на български като „Компютърен облак”, представлява организиране на ресурси от изчислителна мощност, данни и приложенията, използвани чрез Уеб достъп. Под „Уеб достъп” не се разбира използване на TCP/IP протокол, а използване на HTTP протокол (който е надстройка над TCP/IP протокола) и неговите производни – надстройки над HTTP. Поточно казано, се разбира достъп до Уеб сървъри (и Портали) и Уеб базирани услуги. Думата „облак” се прилага като еквивалент на Уеб мрежата, наричана понякога неформално „Интернет”, защото в графичното изобразяване на Уеб достъпа („WWW”), се използва графика, напояняща облак. Тук би трябвало да се направи разликата между „Интернет” и „WWW”. Докато в понятието „Интернет” се включват всички видове дейности и услуги, базирани на използването на протокол TCP/IP, то под понятието „WWW” се разбира съвкупността от Уеб сървъри и Уеб базирани услуги. Под „Уеб базирани услуги” не трябва да се разбира само Уеб услугите (Web services), а и всички видове услуги, които може да се предоставят, използвайки Уеб достъп, такива като Уеб 2.0 и Уеб 3.0.

Ресурсите в Компютърния облак са достъпни онлайн, от всеки тип устройство, което поддържа Уеб достъп. Всеки потребител на Интернет има Уеб достъп, може да бъде индивидуална личност или фирма и може да има достъп до тези ресурси [2]. Данните и приложенията в Компютърния облак са отделени от потребителя и са централизирани във виртуално пространство, което представлява самия Компютърен облак. Потребителят осигурява само потребителски интерфейс (например браузър), чрез който може да ползва всичките компютърни услуги, които се предлагат от Компютърния облак. Аналогия на Компютърния облак може да се направи с Големите машини (mainframes), при които потребителят ползва неинтелигентен терминал – стандарт IBM3280, и посредством този терминал предава всичките заявки и получава всичките резултати. Компютърният облак осигурява централизирано управление на всичките изчислителни ресурси, като използва предимствата на централизираните компютърни системи. Това обаче не означава, че всички сървъри в един Компютърен облак са централизирани; те могат да са разпределени, но трябва да имат централизирано управление. Същевременно Компютърният облак започва да променя изцяло начинът за създаване и използване на информационни системи. При Компютърния облак се използва принципът на поделените ресурси – компютърни и човешки (за управление на компютърните ресурси), който води до

намаляване цената на използване на компютърните ресурси. В Компютърния облак се прилага също така и принципът „плащаш само каквото си ползвал”, т.е. не се плаща стойността на изчислителните системи, съставляващи Компютърния облак, а се плаща за ползваните ресурси от него.

Компютърният облак често се свързва с технологии с онлайн достъп, такива като „Софтуер като услуги” – Software as a Service (SaaS) или „Хардуер като услуги” – Hardware as a Service (HaaS). Тези две услуги – SaaS и HaaS, са най-типичните услуги, които Компютърният облак предоставя, и те са едни от първите услуги, осигурявани от Компютърния облак при неговото първоначално формиране като подход за работа. По принцип SaaS предоставя възможност приложните програми, които са заредени някъде в Компютърния облак, да се ползват, като потребителят се концентрира не как да състави тези програми, не как да ги настрои, не как да ги поддържа, а само как да ги ползва, като плаща толкова, колкото е ползвал.

Услугата HaaS предоставя възможност да се ползват хардуерни ресурси – сървъри, големи машини, дискови системи и пр., на които потребителят да зарежда свои програми за изпълнение. При HaaS потребителят не се грижи за купуване на хардуер, за доставка, за инсталиране, за поправка и поддържане, а се концентрира върху ползването на хардуерните ресурси, като плаща само толкова, колкото е ползвал.

Съществуват фирми и организации, които притежават много комплексни изчислителни системи, за които да се намери Компютърен облак, за да реализира техните изисквания за изчислителна система, което не е лесен въпрос. Същевременно има множество фирми и организации, например като банките, които искат техните системи да са зад Защитни стени, които те лично управляват, защото те отговарят със собствените си пари за това, ако някой влезе в тези изчислителни системи и нанесе изменения в данните. Подобен въпрос за лични изчислителни системи, а не поделени системи, възниква, когато трябва да се прави одит на оперирането на изчислителни системи – къде да се одитира, кои компютърни системи да се изследват и как да проследи транзакционността на бизнес процесите при поделени ресурси. Също така е много важен и въпросът за гарантираното качество на услугите и предвидимата производителност (свързана с времето за отговор и с времето за приключване обработката на един бизнес процес – например като „Край на деня” при банковите системи и пр.). Същевременно съществуват ИКТ системи (системи на базата на Информационни и комуникационни технологии), като на банки и застрахователни компании, които са аутсорсвани на сервизни организации. Но ако една сервизна организация извършва аутсорсване на ИКТ системите на две застрахователни компании, каквито случаи има например в дейността на T-Systems, то тя може да се превърне в Компютърен облак за застрахователни организации, т.е. в Компютърен облак от специализиран частен вид. Множеството практики на аутсорсване в света показват, че каквито и комплексни изчислителни системи да съществуват в дадени фирми, те могат да се аутсорснат, т.е. могат да съществуват организации, които да предоставят изчислителни ресурси за функционирането на тези комплексни информационни системи. От гледна точка на сигурността, някои действия като администрация на бази от данни или поддържане функционалността на някои приложения

програми, могат да останат в ръцете на потребителя, а всички други услуги да се извършват от фирмата, предоставяща услуги за аутсорсинг. По този начин предоставяните услуги могат да се разделят на части – Хардуер като услуга, Операционни системи като услуга, само Някои приложни програми като услуга (за другите приложни програми такава услуга да не се предоставя), Мрежа като услуга и т.н. Ето защо Компютърният облак трябва да може да предоставя както комплексни услуги, така и услуги за по-детайлно ниво изчислителни действия, осигурявайки за всеки клиент нужните видове услуги. Услугата на Компютърния облак не е продукт, а представлява компютърна възможност или компютърна дейност с условия за нейното евентуално ползване, с Уеб достъп, с управление, включващо създаването и спирането на услугата, с потенциална възможност за конфигуриране преди ползването посредством метаданни, с многопотребителска едновременна форма на ползване, с освободена обвързаност към място за изпълнение, с възможност за динамична скалабилност и надеждност, виртуализиране на компютърната система, на която се работи, като заплащането за тази дейност не се прави чрез закупуване на съответни технологии, а се заплаща само ползването на самата услугата.

За да се ползват услуги на Компютърния облак, не са нужни капитални разходи, т.е. потребителят не е нужно да си купува първоначално сървъри, операционни системи, приложни програми, мрежови устройства и пр. Потребителят заплаща само за услугите, които е ползвал. По този начин вниманието на потребителите се измества от заплащане за технологии към заплащане за това какво технологиите могат да направят. Този подход е много подходящ за малки и средни предприятия. В света съществуват два основни модела за заплащане [1] – месечна равномерна вноска, или според количеството и качеството на ползваните услуги. По този начин финансовият модел за ползване на Компютърен облак е еквивалентен на модела на ползване на електричество – ползваш услуги, които не е важно къде се създават. Услугите имат определени качества и заплащането е според ползваните ресурси и предоставеното качество на изпълнение на тези услуги [3]. В цената на услугата участват и параметри като степен на гарантираност на услугата, надеждност, производителност, гъвкавост за увеличаване на скалабилността, интегрируемост на услугата с други услуги, минимално време за поправка, време за възстановяване и пр. Основата на намаляването на цената се базира на поделеното ползване на ресурси, т.е. няколко потребители ползват един и същ ресурс и заедно заплащат стойността за неговото изграждане и поддържане [5]. Естествено е, че не може да има генерализирани услуги от даден тип, например ползване на изчислителна мрежа на фиксирана цена, независимо от използваната пропускателна способност и независимо от качеството и надеждността на връзките. Би трябвало да има цена за Mbps, осигурявана с определено гарантирано качество и с определена надеждност. Подобна ситуация съществува в момента при работа с Мрежови доставчици на услуги. Например доставчик на MPLS услуги предлага 4 вида качество на услуги, с 20 вида различни скорости и 3 вида надеждност, което прави 240 различни цени; потребите на тази MPLS услуга може да превключва от една скорост на друга, от едно качество на друго, като съответно се променя цената на месечното заплащане. Този пример показва, че Компютърният облак не е нещо съвсем абстрактно за съв-

ременните доставчици на изчислителни услуги, но не е разпространено като явление за всичките възможни услуги, които се изискват за работа на една изчислителна система.

Компютърният облак предполага приемственост на множество от съществуващите изчислителни и комуникационни технологии (ИКТ), но същевременно се създават специални технологии за неговото функциониране. По-долу в настоящата част от студията ще се представят някои от онези специфични технологически решения, ориентирани към Компютърния облак.

Компютърната част на клиента, с която се осигурява достъп до Компютърния облак, може да представлява комбинация на Браузър с Богати интернет приложения - RIA (Rich Internet Applications). Богатите интернет приложения представляват plug-ins към браузър или добавени виртуални машини (като JVM), такива като Adobe Flash, JavaFX, Microsoft Silverlight и пр. Посредством Богатите интернет приложения резултатите, представяни чрез браузър, имат богатата функционалност като на десктоп приложенията. Пример за такава работа е използването на видео резултат, представян от YouTube, които работи посредством Adobe Flash, добавени към браузър. Характеристиките на Богатите интернет приложения, които се добавят към стандартния браузър са:

- Уеб базирано Зареждане и Инсталиране;
- Възможност да работи с няколко операционни системи;
- Автоматичен ъпгрейд и промяна на версии;
- Определен вид сигурност на протокола за обмен на данни с Компютърния облак.

За да се осъществи пълно функциониране от ниво браузър, наподобяващо работа на десктоп приложения, се използват компонентите widgets, gadgets, pipes и blocks. Widgets са визуални елементи на Графическия потребителски интерфейс, чрез които се представят променяеми от потребителя компоненти като бутони, прозорци и правоъгълници / листове с текст. Gadgets са малки технологически обекти, считани като технологическа новост (хардуерни и софтуерни), чрез които също се осъществява диалог с приложения от Компютърния облак – такива като устройства iPhone, SmartPhones и софтуер за търсене чрез Google, представяне на географски карти, и пр. Pipe осигурява преобразуване на два или повече източника от данни в един, под формата на XML, който след това може да се представи визуално на потребителя. Blocks осигурява филтриране на подавана информация са визуализиране.

При разработването на Архитектура за Компютърния облак се използват различни принципи и технологии, но има два принципа, които имат по-утвърдено приложение, защото са създадени отпреди няколко години:

- a) Виртуализация;
- b) Уеб ориентирана архитектура.

Виртуализацията е принцип на абстракцията на компютърния ресурс. Почти всички компютърни ресурси могат да бъдат виртуализирани – сървъри, операционни системи, оперативна памет, дискова памет, изчислителна мрежа, приложения [1]. Тази виртуализация може да се осъществи с хардуерни средства, софтуерни средства и комбинирано. Съществува и така наречената виртуализация на платформата, при която операционната система се освобождава от ангажимента за работа с опре-

делени команди на процесора, т.е. тя става виртуална по отношение на платформата на която се изпълнява. Такава виртуална по отношение на платформата операционна система обхваща и случаите на консолидиране на сървъри, работещи с дадена операционна система. Средно, инфраструктурните ИКТ елементи се използват много неефективно – с коефициент на използване 10%-20% [6]. Въвеждането на виртуализацията осигурява по-добро използване на тези ресурси, като в множество случаи един хардуерен сървър на базата на Интел типа процесори може да извършва ефективно управление на 8-12 виртуални сървъра (все пак зависи от типа приложения които се изпълняват от тези сървъри). Всичко това обяснява икономическата изгода от използването на виртуализацията и защо в Компютърния облак тази технология е една от основните.

Уеб ориентираната архитектура – WOA (WOA – Web Oriented Architecture) е вид софтуерна ИКТ архитектура, представляваща разширение на Архитектурата ориентирана към услуги – АОУ (SOA – Service Oriented Architecture). Съществена част на WOA е използването на принципа на Оперирание с представителна част на ресурс – REST (Representation State Transfer). При REST се оперира с Уеб ресурси, представени чрез техните глобални идентификатори (URI), използвайки стандартния протокол HTTP. АОУ осигурява оркестрирането на тези ресурси, получавайки крайния резултат. Оркестрирането може да се извърши от BPEL (език за изпълнение на бизнес процеси) или от някой друг процедурен език или съвкупност от бизнес правила. Като пример за работа с представителна част на ресурс може да се използва *кръг*, разглеждан като ресурс, а неговата представителна част може да бъде или координати на 3 точки от кръга, или център и радиус. При оркестрирането с такива ресурси, освен адресът на ресурса (URI), е необходимо да се знае в какъв формат на данните се получава тази представителна част на ресурса – текст, фигура, HTML, XML, JSON (JavaScript Object Notation – текстово базиран формат на данни, четим от човек, представляващ структура от данни и обхващаща числа, символи, Булеви променливи, масиви от еднакви величини и обекти, съставляващи колекции от различни величини) [7]. Друга съществена част от WOA използва RSS (Really Simple Syndication) [8], представляващи Уеб ресурси, захранващи със съдържание Уеб екрани, Desktop екрани или екрани на мобилни апарати, и представляващи стандартен XML файл за публикуване на информация. Определен RSS четец (програма) регулярно трябва да извлича информацията от тези Уеб ресурси, за да се визуализира резултатът. Процесът на визуализация може да се извърши с множество програмни средства. Също важна част на WOA е стандартът Atom, който функционира по подобие на RSS и е създаден като алтернатива на RSS, използвайки AtomPub (Atom Publishing Protocol) протокол за захранване с Уеб съдържание [9][10], като неговият формат на данни също е базиран на XML. AtomPub е надстройка на HTTP протокола и се използва за създаване и актуализиране на Уеб ресурси. За разлика от RSS, където процесът по обединение на Уеб ресурсите трябва да извлича новото съдържание на ресурсите, то Atom осигурява предаване на новите изменения в ресурсите на процеса – обединение на Уеб ресурси, т.е. в обединението на Уеб ресурси съществува tag за дефиниране на Atom ресурс, за този ресурс е създаден до-

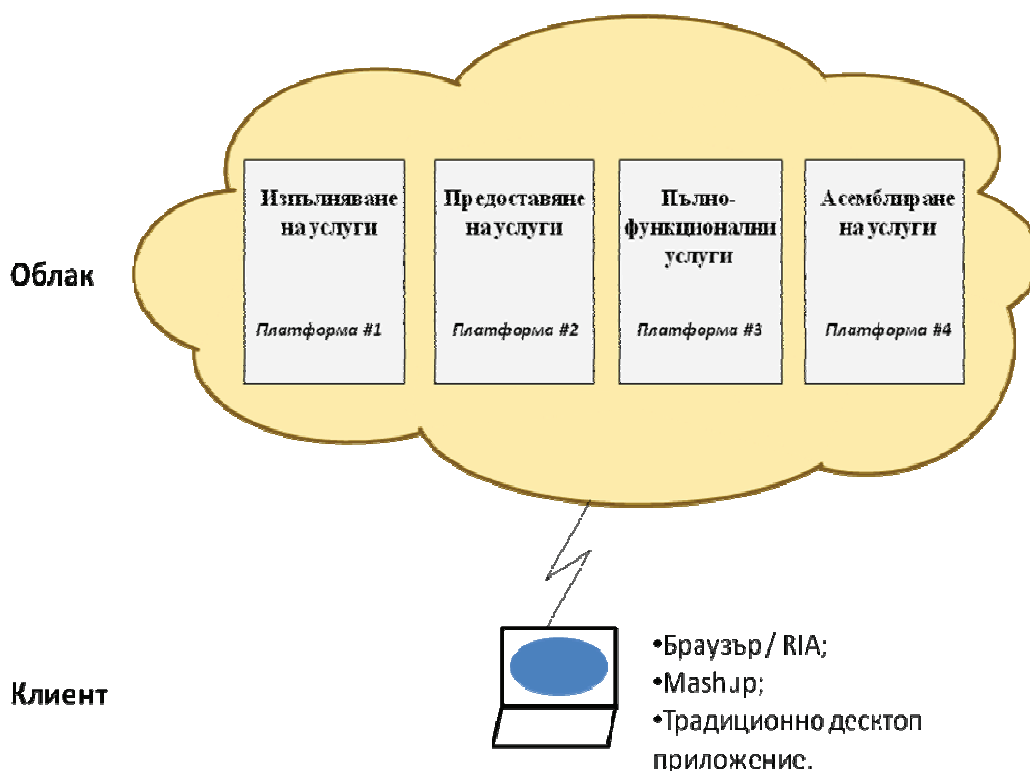
пълнително Atom формат, с което се осигурява експортиране на променено съдържание на Уеб ресурс до обединението на Уеб ресурси.

При създаване на архитектура на Компютърния облак се използва също и технология, наречена "Mashup", която може да се формулира на български „Технология на обединението". Тя има аспекти както при създаването на нови данни, така и при създаването на резултати от различни приложения. Тази технология представлява създаване на Уеб страница или приложение, комбиниращи данни или функции от различни външни съществуващи източници. Това съответства на форма на интеграция без съществена промяна на вида и съдържанието на информацията, идваща от участващия източник. Например при използване на една картографска информация над нея може да се насложи информация за дадена земеделска област, както и различни варианти на състоянието на река според количество валежи. Технология на обединение на данни се отнася за данни от подобен тип, прилагани за създаване на едно представяне (например уеб страница). При Технология на обединението за приложения няколко приложения се активират за създаване на една Уеб страница или Уеб услуга. По принцип, Технологията за обединение и технологията на Портала са технологии за агрегиране, но при Портала се работи на две стъпки – генериране на фрагменти (създадени от компонента portlet) и интегриране на фрагментите в една уеб страница, като заявката за създаването на тази уеб страница се трансформира на заявки до отделните portlets. По този начин Порталът може да се разглежда като интегриране на ниво сървър, осъществявано на ниво 6 на ISO Референтния модел – презентация. При Технологията на обединението може да се извършва агрегиране на две нива – уеб ниво, което може да се нарече клиентско ниво, извършвано от клиентския браузър (браузъра комбинира и реформатира различните данни и тогава ги извежда), и сървърно ниво, извършвано от сървър, разположен в Компютърния облак (анализирането и преформатирането на данните става в сървър, намиращ се в облака, като вследствие от неговата работа се получава обединен резултат като уеб страница, която се изпраща на браузъра за директно визуализиране). Технологията на обединението предполага лесно персонализиране на представяната чрез браузъра информация.

Една от основните характеристики на Компютърния облак е скалабилността, т.е. при необходимост да се увеличава изчислителната мощ на съответните сървъри, без да е нужно допълнително програмиране или зареждане на нови софтуерни услуги. За осъществяване на тази скалабилност се използват добре известните технологии на Load Balancing, Clustering и Multiprocessing [11],[12],[13]. За целите на скалабилността на Компютърните облаци Google създаде нова технология, наречена MapReduce [14], която осигурява разпаралелване една заявка за обслужване от много паралелно работещи сървъра (Map) и след обслужване – обединение на всичките получени резултати от всичките паралелно работещи сървъри в един отговор (Reduce).

Компютърният облак не може да се разглежда като чиста ИКТ архитектура или като изчислителна инфраструктура. В Компютърния облак трябва да съществуват допълнително механизми за създаване на услугите, механизми за управление функционирането на услугите и механизми за заплащане на тяхното ползване. Според

изследователи на Компютърни облаци тяхната архитектура се състои от различни видове ИКТ платформи [3]. В настоящата студия се дефинират 4 вида платформи, в зависимост от това какви услуги се предлагат от Компютърния облак. Тези платформи, заедно с Уеб достъпа до клиенти, използващи специализирани технологии за Компютърни облаци, формират Концептуална архитектура на информационна система, работеща с Компютърни облаци. Една информационна система може да използва един или няколко Компютърни облаци за своето функциониране. В настоящата студия се представя разработена от авторите такава Концептуалната архитектура, която е представена на фигура 4.



Фиг. 4

Разработена е Концептуалната ИКТ на Информационна система, използваща Компютърен облак, която се представя със следните видове платформи на Компютърния облак:

- Изпълняване на услуги* (Платформа-1) – това е основният вид платформа на Компютърния облак, която осигурява изпълнението на отделните видове услуги, такива като SaaS, NaaS, INFaaS (информация като услуга), NaaS (мрежа като услуга), AaaS (приложение сигурност като услуга), STaaS (памет като услуга), SECaaS (информационна сигурност като услуга), BUSPRaaS (бизнес процес като услуга), CONTaaS (съдържание като услуга) и пр. Когато се проектира Компю-

търен облак, тази платформа е първата, която се проектира, за да се осигурят основните функциониращи услуги на Компютърния облак. Тази платформа обикновено се създава и управлява от една фирма притежател на Платформа-1 (представлявайки Едноличен собственик и мениджър на Платформа-1), но също така може да се създава и управлява от няколко фирми, координирайки своите дейности, предлагани от Платформа-1 (представлявайки Асоциативен собственик на Платформа-1). При няколко собственици предлаганите услуги от всеки собственик обикновено са взаимно допълващи се. Независимо от броя собственици на Платформа-1, потребителите са много и от различни организации;

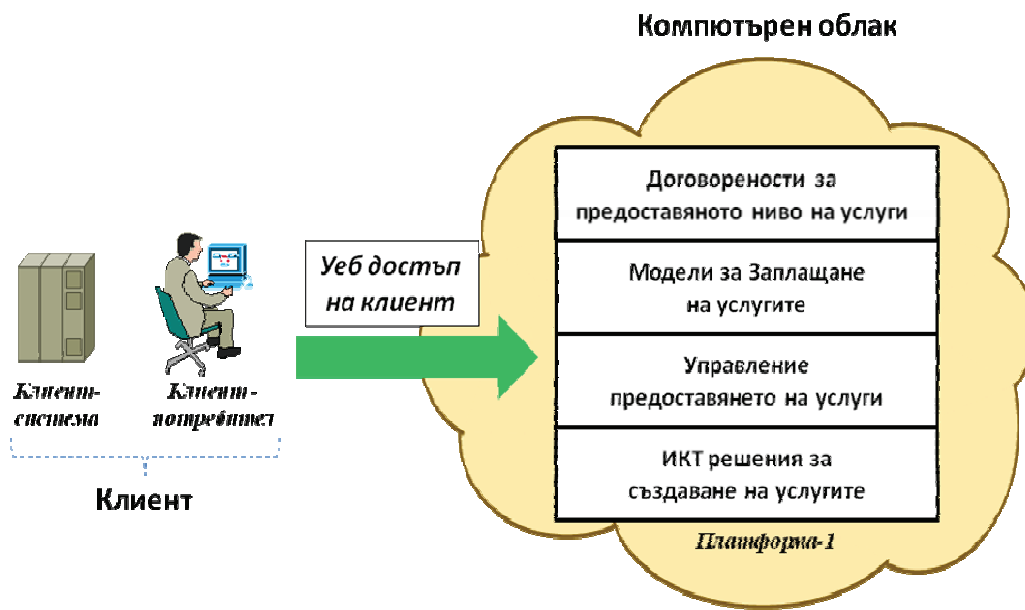
- b) *Предоставяне на услуги* (Платформа-2) – това е платформа, която Доставчик на услуги на Компютърен облак предоставя на отделни клиенти (може да се направи аналогия с Доставчик на Интернет услуги, който предоставя на клиенти различни видове Интернет достъп, скорост и качество на свързвания за различни клиенти). Обикновено доставчикът на тази услуга е трета страна – между Компютърния облак и клиентите. Предоставянето на услуги на Компютърен облак изисква хостване на една или няколко ИКТ системи от типа на Платформа-1, които фактически да изпълняват услугите на Компютърния облак. Доставчикът на Предоставяни услуги: или ползва платформи от типа Платформа-1 на други фирми; или консолидира услуги на няколко фирми, работещи с Платформа-1, като осъществява препредаване на заявки и резултати; или интегрира услуги, осигурявани от една или няколко Платформи-1. За да може да се осъществи процесът на хостване, както и консолидиране и интегриране на услуги, е необходима нов вид платформа – Платформа-2;
- c) *Пълнофункционални услуги* (Платформа-3) – предлагаща определен затворен кръг от специализирани услуги на Компютърния облак, които са насочени: или към определен вид бизнес функционалност на хоризонтално ниво (например CRM – Връзки с клиенти или ERP – Планиране ресурси на предприятието); или изпълняват функции за определен индустриален сегмент на вертикално ниво. Този вид платформа – Платформа-3, изпълнява фиксирано множество от услуги, които не подлежат на модифициране или надстройка. Същевременно различни клиенти могат да ползват тези услуги, например различни вериги за хранителни стоки могат да ползват един тип Компютърен облак с Платформа-3;
- d) *Асемблиране на услуги* (Платформа-4) – извършвано от специален тип Доставчик на услуги на Компютърен облак, който осъществява създаване на нов вид услуги на базата на съществуващи услуги, предоставяни от Платформа-1, и/или Платформа-2, и/или Платформа-3. Този нов вид услуги представлява асемблиране на услуги на Компютърния облак (обвързване на съществуващи услуги с добавяне на бизнес логика за взаимовръзка и оркестриране между тях, както и Mashup на приложения), създавайки нов вид интерфейс с клиента, нов вид SaaS, нов вид BUSPRaaS, нов вид CONTaaS и пр. За да се осъществят тези функции на асемблиране, е необходима нов вид платформа – Платформа-4. Тази платформа може да извършва Mashup на приложения, използвайки резултати от Платформа-1, Платформа-2 и Платформа-3, както и бизнес оркестриране на техните услуги.

Представените платформи могат да се използват за създаване на:

- Компютърен облак от частен тип – работещ със заплащане на всяка ползвана услуга;
- Компютърен облак от обществен тип – при който не се заплаща ползването на услугите, като покриването на разходите на доставчика на тези услуги се осигурява от приходи от реклами.

4. МЕТОДИ ЗА ПРОЕКТИРАНЕ НА БИЗНЕС ИНФОРМАЦИОННА СИСТЕМА НА БАЗАТА НА CLOUD COMPUTING

При проектирането на Cloud Computing (Компютърен облак), основното е да се създаде Платформа-1. Тази платформа осигурява предоставянето на основните оперативни услуги, които предлага един Компютърен облак. При проектирането на Платформа-1 се дефинират услугите, които ще се предоставят, като за всяка услуга се създава съответно решение. Всички останали платформи – 2, 3 и 4, представляват надстройка над Платформа-1. За да се извърши проектиране на Платформа-1, от авторите е създадена „Рамка за проектиране на Платформа-1“, която има за цел да осигури всичките необходими компоненти на Платформа-1, за да може тя да функционира като Компютърен облак. Тази Рамка е представена на фигура 5.



Фиг. 5

Представената рамка се състои от 4 нива:

- *ИКТ решение за създаване на самата услуга.* На това ниво се изгражда самата ИКТ архитектура на решението, чрез която съответният компютърен ресурс се

оформя като услуга за ползване чрез Уеб достъп. Това ниво е същинската част на предоставянето на услугата;

- *Управление предоставянето на услуги.* Това ниво извършва конфигуриране на услугата според специфичните изисквания на клиента, създава нужната скалабилност и надеждност и наблюдава функционирането на предлаганите услуги;
- *Модели за заплащане на услугите.* Това ниво предлага различни модели за заплащане на услугата, например еднаква стойност на месец за услуга, заплащане на ползването на услугата на час, заплащане според количеството ползван ресурс (обем дискова памет, пропускателна способност на мрежа), модифицируемо заплащане според предоставяното качество на извършената услуга (в рамките на допустими диапазони на качество) и пр.;
- *Договореност за предоставянето ниво на услуги.* Това ниво осигурява регистриране и следене на предоставянето на услуги на базата на SLA (Service Level Agreement).

При работа на Компютърния облак за предоставяне и ползване на услуга в паралел работят и четирите нива на Платформа-1. Освен това те си взаимодействат едно с друго. В настоящата студия ще се представят основните методи за създаване на Компютърен облак, като ще се акцентира главно върху ИКТ решенията за Създаването на услугите и Управлението на предоставяните услуги, т.е. върху изграждането на първите две нива на представената Рамка за проектиране на Платформа-1. Архитектурното решение, представящо принципите на работа на тези две нива на Компютърния облак, е показано на фигура 6.

При него отделните компютърни ресурси (такива като сървъри, мрежи, системи, програми и бизнес процеси) първоначално се трансформират в Компоненти, а след това в Услуги. Превръщането на компютърните ресурси в компютърни компоненти е предмет на системната интеграция, позната добре в науката и литературата [15][16], за която съществуват множество технологии и продукти. Възможно е един ресурс да се представи като самостоятелен компонент или като част от компонент. Например един рутер, който е ресурс, може да се представи без никакви промени като компонент (в случай, ако е обособен логически като отделна единица, например Граничен рутер), а може да се представи като част от компонент (в случай на PE компонент на MPLS мрежа, в който участват допълнително втори рутер и комутатор). Превръщането на ресурс в компонент е логически процес, от една страна, а, от друга, е процес на извършване на системна интеграция. Логическото обединяване на ресурси в компонент зависи от типа услуга, която ще се формира в Компютърния облак, както и от физическата ИКТ архитектура на самия ресурс. Интегрирането на ресурс е въпрос на ИКТ решение, за осъществяването на което не са нужни нови научни достижения, а съществуващите световни добри практики са достатъчни за извършване на най-прецизните и сложни интеграции. По тази причина обособяването на ресурс като компонент или интегрирането на ресурси в компонент не е предмет на настоящата студия.



Фиг.6

Звено „Превръщане на компонентите от Компютърни ресурси в услуги“ използва един или няколко компонента от Компютърни ресурси и ги превръща в услуга. Основното различие на услуга от компонент е, че услугата може да се състои от агрегирани компоненти, че може да се адресира самостоятелно чрез Уеб достъп и че е на едно по-високо абстрактно ниво, отделяйки ненужните технически детайли и насочвайки фокуса към бизнес използването на услугата. Превръщането на компонента в услуга става на основата на Архитектура, Ориентирана към Услуги на Междинно Ниво (АОУМН). Това е стандартна АОУ, която в 7-слойната архитектура на ISO Референтния модел обслужва не 7-о ниво, а 4-то ниво, т.е. на междинно ниво според този модел. При използване на АОУМН се създават ИКТ решения (хардуерни и софтуерни) за представяне на всеки компонент като отделна Уеб услуга, както и за нейното модифициране съобразно конкретни клиентски изисквания. При този подход, за всеки отделен компонент се прилагат различни хардуерно-софтуерни решения, чрез които се подпомага неговото разпознаване от АОУМН като услуга. По тази причина АОУМН е един от елементите на това преобразуване. Друга роля на АОУМН е да извърши оркестриране на получените услуги, провеждайки съответно модифициране според конкретни клиентски изисквания. За реализиране на АОУМН може да се използват различни АОУ, например такива с отворен

код, които допускат BPEL или други процедурно ориентирани бизнес езици или бизнес правила, и които да осъществяват оркестрирането. Това дава възможност отделният клиент да използва стандартни бизнес езици за кustomизиране на своите услуги.

След като се формира услугата от компютърни компоненти, тя се предава на друга Архитектура, Ориентирана към Услуги (АОУ), която да извършва самото „Управление на предоставяните услуги“. По принцип това ниво на Рамката на Платформа-1 се осъществява с АОУ, но частта за модифициране и кustomизиране на услугите според клиентски изисквания може да се извършва от АОУ, от АОУМН или от двете заедно, в зависимост от характера на услугата. За практическа реализация на АОУ може да се използва също АОУ с отворен код, допускащи работа с BPEL или други процедурно ориентирани бизнес езици или бизнес правила. В зависимост от конкретната реализация, при отчитане на нужната скалабилност и производителност, АОУМН и АОУ може да се изпълнят чрез една системна реализация, а не чрез две. Същевременно за АОУМН и АОУ могат да се използват не само системи с отворен код, а и такива на крупни софтуерни фирми, например на IBM и Oracle.

Ползването на услугите става чрез стандартен достъп до Уеб услуги, адресиран до „Управление на предоставяните услуги“, което е фактически АОУ, в която всяка услуга на Компютърния облак представлява Уеб услуга.

4.1. Метод за проектиране на Мрежа като услуга – NaaS

В изчислителните мрежи управлението се състои от манипулиране на мрежови съобщения и от управление на мрежовите устройства, на които да се зададе как мрежовите съобщения да се обработват. За да може да се осъществи управление на изчислителната мрежа като услуга, трябва да се управляват посочените двата главни ресурса на мрежата – мрежовите съобщения и мрежовите компоненти. Управлението на тези мрежови ресурси трябва да се изведе до ниво услуга, управляема като Уеб услуга, т.е. мрежовите компоненти да се адресират и управляват със средствата на Уеб услуги, разпознаваеми в АОУМН като услуги.

Принципът на управлението на изчислителните ресурси третира мрежовите съобщения по начин, който представлява надстройка на третирането на мрежови съобщения, съществуващо в съвременните изчислителни мрежи, т.е. от един или няколко IP пакета се формира приложното бизнес съобщение (във вид, в който приложната програма го разпознава). Това се прави с оглед разпознаваемостта на мрежовите съобщения от ниво „услуга“, която да се управлява в Компютърния облак. Мрежовите устройства са физическите устройства, съставлящи изчислителната мрежа – рутери, комутатори, кабели, радио вълни и пр. По принцип мрежовите устройства не могат да имат динамично поведение, но взаимосвързани комбинации от такива устройства могат да предлагат динамично поведение, ако динамично се променят техните конфигурационни файлове. Точно този аспект на динамизъм е залегнал в настоящата разработка за управление на мрежовите устройства като

„услуги”. На тази база ИКТ решението за превръщане на мрежовите ресурси в услуги е представено на фигура 7.



Фиг.7

Получаването на мрежовите ресурси като услуги се състои от два етапа – „Подготовка на мрежови компоненти” (специфична част за мрежовите ресурси) и „Превръщане на компонентите от Компютърни ресурси в услуги” (общо за всички предлагани услуги в Платформа-1). Подготовката на мрежови компоненти се състои също от две части – Бизнес интелигентни комуникации, предназначени за опериране със съобщенията, и Динамично управляеми мрежови устройства (извършващи управлението на мрежовите устройства за формиране на мрежови ресурси).

Блокът „Бизнес интелигентни комуникации” селективно избира IP пакети, отправени към някоя бизнес система (например SAP), и формира съответното приложно съобщение (например SAP съобщение). Имайки XML формата на съобщението, блокът „Бизнес интелигентни комуникации” анализира съдържанието на съобщението и взема съответни мрежови решения, базирани на това съдържание. Главното решение, което блокът взема, е изборът на онова приложно съобщение, чиято структура е описана чрез XML. След като приложното съобщение е избрано, то този блок може да преобразува съобщението от един тип в друг, вземайки неговите данни, като формира съобщение от друг тип, използвайки същите приложни данни. За възможна реализация на това решение би могло да се използва Cisco AON устройството, защото то подпомага селектиране на IP пакети за последващо преобразуване като приложни съобщения. AON е устройство, обикновено вградено

в рутер (например Cisco 2860 на фирмата Cisco) или комутатор. В предлаганото решение може да се използва като АОУМН продуктът с отворен код ServiceMix.

За да работи блокът „Динамично управляеми мрежови устройства” според принципите, посочени по-горе, той трябва да съдържа хранилище на конфигурациите, където се пази информация за наличните и възможни конфигурации на мрежовите устройства, както и за начините и ограниченията, които съществуват, за възможни бъдещи модификации на тези конфигурации. „Автоматичният конфигуратор” е част от този блок и осигурява промени на съществуващи конфигурационни файлове или генериране на чисто нови файлове, като всеки новополучен конфигурационен файл се проверява за валидност. Съществуват фирмени решения за проверката на валидност на конфигурационния файл в рамките на едно устройство, докато Автоматичният конфигуратор проверява за валидност на последователност от връзки между устройства, правещи една бизнес комуникация. Когато новият или модифициран конфигурационен файл е готов, той се зарежда в съответното устройство според режимите на работа на мрежовите устройства. Получените бизнес комуникации, представляващи комбинация от една или няколко мрежови връзки с интегрирани характеристики, се оформят като мрежови услуги АОУМН.

4.2. Метод за проектиране на Изчислителен ресурс като услуга – COMPaas

Изчислителни ресурси са необходими за изпълнение на различни бизнес приложения – например Unix сървър е необходим за работа на Файл сървър, Linux сървър е необходим за Уеб портал и т.н. Предоставяните ресурси, които се превръщат в услуги, могат да бъдат:

- Файлово пространство;
- Таблици и пространство в база от данни;
- Процес – програма за изпълнение;
- Изчислителна мощност на десктоп или сървър, в която в последствие да се зареди програма за изпълнение.

За да се предоставят ресурси, в десктопа и сървъра трябва да има специализирана програмна система, която да предоставя ресурси, несъздавайки конфликт в работата на другите приложения на десктопа / сървъра, които са техни основни приложения. За целта в предложената система е използвана специална Виртуална машина (VM) за Доставка на Изчислителни Ресурси. Тя използва съществуващи Виртуални машини на пазара, за които е създадено допълнително програмно осигуряване. Цялостният процес на формиране на услуги от изчислителни ресурси е показан на фигура 8.



Фиг.8

От представената фигура се вижда сходството с осигуряване на услуги от изчислителна мрежа, описано по-горе. Блокът „Подготовка на изчислителни ресурси“ може да работи с няколко системни услуги, извършвани от системния блок „Управление доставянето на изчислителни ресурси“:

- Приемане на Заявка за изчислителен ресурс;
- Предоставяне на изчислителен ресурс;
- Освобождаване на изчислителен ресурс.

Този системен блок може да се реализира централизирано като обикновен Java обект – РОЈО за АОУМН. Функциите, които може да изпълняват „VM за доставяне на изчислителни ресурси“, могат да бъдат:

- Предоставяне на данни за възможни типове ресурси за използване;
- Предоставяне на Ограничения на всеки достъпен ресурс;
- Изготвяне на информация за налични ресурси в средностатистически план;
- Данни за Налични ресурси в определен момент (периодично генерирана информация);
- Приемане на заявка за заемане на ресурси;
- Информация за Наличност на нужните ресурси по време на получаване на заявката;
- Изготвяне на информация за използваните ресурси по време на периода на използване;
- Използване на ресурс – работа с База от данни;
- Използване на ресурс – работа с Файлова система;

- Използване на ресурс – работа с Процес (програма) чрез нейния интерфейс – API;
- Определяне на дисково пространство, в което в последствие да се зареди някаква програмна система за изпълнение (това фактически е заделяне на изчислителни ресурси, понеже специално заделяне на процес, памет и периферни устройства при Виртуалната машина не се извършва);
- Приключване ползването на ресурси и освобождаване на ресурса

По този начин предлаганият метод за създаване на услуга от Изчислителен ресурс обединява в себе си създаването на услуги като NaaS, INFaaS, STaaS.

4.3. Метод за проектиране на Приложение като услуга – AaaS

Приложенията, които са проектирани като Уеб услуги, могат да се използват без изменения като услуги в Компютърния облак. Програмните компоненти, които не са под формата на Уеб услуги, но разполагат с приложен програмен интерфейс – API или могат да се обслужват чрез приложен комуникационен адаптер (адаптер в термините на J2EE Java Connection Architecture – JCA) са една значителна група приложения, които в настоящата студия се предлага да се преобразуват в услуги за Компютърен облак. Тази услуга е наречена „Приложение като услуга” – AaaS. Настоящият метод използва два подхода:

- а) Подход с използване на **Web Services Invocation Framework** за осигуряване на връзка със съществуващи програмни компоненти;
- б) Подход с използване на **Адаптери** за осигуряване на връзка със съществуващи програмни компоненти.

Подходът за използване на Web Services Invocation Framework (WSIF) [17], работи с WSIF binding, който предоставя на програмната компонента възможността отвън да се разпознава чрез WSDL, т.е. като Уеб услуга, а отвътре да се прави обръщение към нея чрез нейните приложни протоколи – API. По този начин програмните компоненти не се преправят, а им се добавя само WSIF binding, което може да е или просто XML описание, или се създава елементарна Java програма.

WSIF може да се разгледа като надстройка на Уеб услугите, както и като инструмент за превръщане на програмни компоненти в Уеб услуги. WSIF е натурален Java API – програмен интерфейс на езика Java, предоставяйки възможност на програмна компонента да бъде извиквана чрез WSDL, все едно че е Уеб услуга. Използвайки WSIF, клиентът на програмната компонента работи с параметрите на WSDL (тип порт, операции и обменяни съобщения), без да отделя внимание на използвания протокол към програмната компонента. Тази абстракция работи, понеже тя е подкрепена от програмни кодове на реални протоколи, наречени „доставчици”. Доставчикът извършва реалното доставяне на съобщението според конкретния протокол. Тази абстракция, предоставяна от WSIF, осигурява:

- динамично извикване на програмна компонента,
- представяне на програмна компонента като Уеб услуга.

Наличието на поддръжката на различни протоколи дава възможност да се свържат множество съществуващи програмни компоненти и да се оформят външно като

услуги за използване от клиентите на Компютърния облак. WSDL binding – частта от WSIF за свързване на програмната компонента, направи съответствие между типа порт (PortType) на WSDL с реалния програмен интерфейс на програмната компонента. Стандартно WSDL използва SOAP като протокол за връзка към съответната услуга. Същевременно WSIF може да работи освен със SOAP и със специални обвързващи разширения за директно обвързване на обикновени Java класове, EJB обекти, програмни компоненти, достъпни чрез опашкови съобщения (message queuing), и компоненти, предлагащи достъп чрез JCA адаптери. Съществуват програмни продукти, например като Oracle BPEL Process Manager, в които „доставчиците” са разширени с добавени нови протоколи, такива като HTTP Get / Post, TCP sockets и др. В структурата на WSIF съществуват посочените „доставчици”, така че обвързването с изброените програмни компоненти се явява прозрачно от гледна точка на програмиране, т.е. не се изисква програмиране.

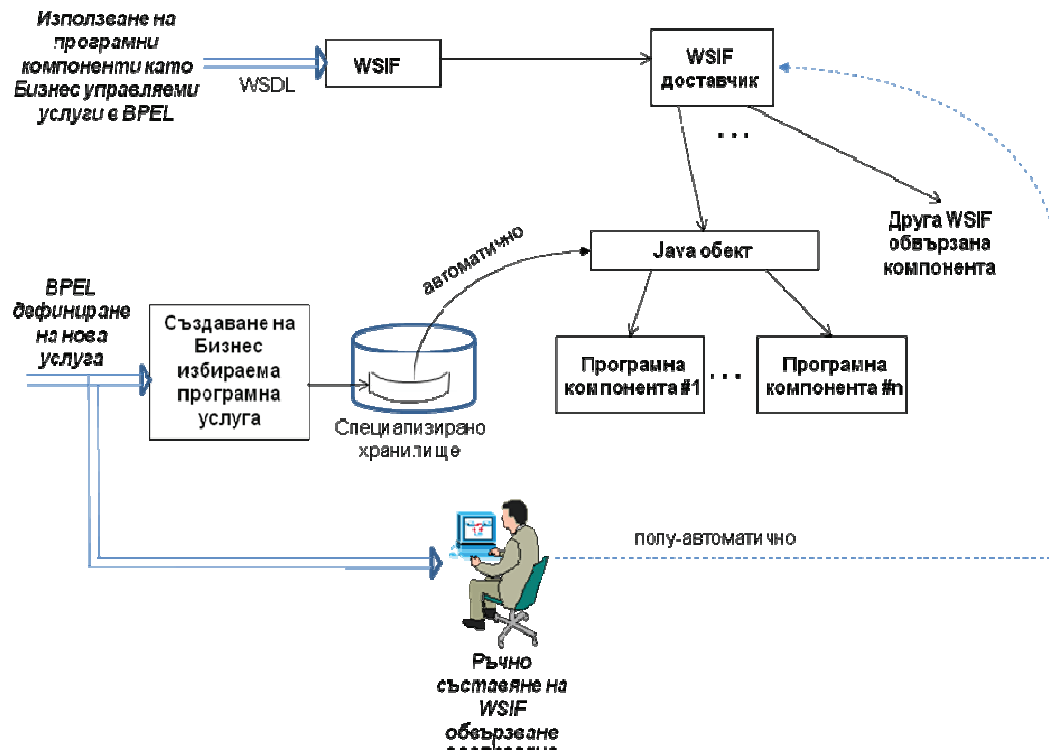
Извикването на обикновен Java клас (Plain Old Java Object – POJO) е многократно по-ефективно, отколкото извикването на Уеб услуга описана чрез SOAP [18]. Това се дължи на обстоятелството, че Java класът е зареден в Приложния сървър на АОУМН и извикването на този клас от външен клиент става чрез локалните средства на Приложния сървър. По този причина предложеното в настоящата студия обвързване на програмни компоненти чрез WSIF и превръщането им в Уеб услуги е с POJO, което е по-ефективен начин за работа.

Адаптерите са елементи на J2EE-архитектурата за връзка на J2EE-Приложен сървър със съответна програмна система, например ERP, База от данни, система, работеща с опашки от съобщения, и т.н. В момента съществуват стотици адаптери за връзки със стотици съществуващи програмни компоненти и протоколи. Архитектурната част на J2EE за управление на тези адаптери се нарича Java Connection Architecture – JCA, и осигурява използването на адаптер за връзка със съществуваща програмна система, използвайки нейния естествен протокол, и премахвайки необходимостта от програмиране на връзки и интерфейси към този програмен компонент. В някои интегрирани системи за проектиране, например като Oracle JDeveloper и Escape, откън стартирана услуга може да има директна връзка с много адаптери и оттам със съответния програмен компонент, без да се изисква допълнително програмиране. За други адаптери, за които няма директна връзка между услугата и тях, може да се използва връзка чрез обикновени Java класове, които от, една страна, са директно адресируеми от АОУМН, а, от друга, могат да ползват всеки един от стотиците адаптери на J2EE-Приложния сървър. По този начин става възможно използването на всеки програмен компонент като услуга, за която има съответен адаптер.

Предложеният метод за бизнес управление на програмни компоненти посредством адаптери не изисква никакви допълнителни ИКТ решения, когато адаптерът на програмната компонента има адаптер за директна връзка. При това положение услугата специфицира работата с дадения адаптер и се посочва местоположението на програмната компонента, която да се използва посредством този адаптер – всичко това е конфигуриране. При положение, че програмната компонента използва адаптер, който не се явява адаптер за директна връзка в АОУМН, то се прилага Java

клас, който да извършва преадресиране на съобщението, идващо от АОУМН до съответния нестандартен адаптер. Създаването на такъв Java клас е рутинна Java програмистка работа.

За предлагания метод за Бизнес управление на програмни компоненти, които не са създадени като Уеб услуги, е разработено ИКТ архитектурно решение, което е показано на фигура 9.



Фиг.9

Показаното ИКТ решение подпомага главно подхода за Бизнес управлението на програмни компоненти посредством използване WSIF, както и частично подхода за Бизнес управлението на програмни компоненти посредством използване на адаптери на JCA. Частичното подпомагане на втория подход се обосновава от факта, че за използването на адаптери, дефинирани в АОУМН, не се изисква никакво допълнително техническо ИКТ решение, а за работа с нестандартни адаптери се изисква съответна Java програма. Когато АОУМН няма дефиниран адаптер, който е необходим за съответната програмна компонента, то тогава трябва да се използва предложеното ИКТ решение – в режим на полуавтоматична работа.

Литература

1. В Кисимов, Динамично бизнес-упрвляеми информационни системи, ISBN 978-954-323-444-8, 2008
2. Cloud computing: The Evolution of a Software-as-a-Service, 2008, <http://knowledge.wpcarey.asu.edu/article.cfm?articleid=1614>
3. Web platforms: Business in the Cloud, Gartner conference, 2008, Barcelona, Spain
4. Cloud computing Myths Magic and Mayhem, Gartner conference, 2008, Barcelona, Spain
5. Forrester's advice to CFOs: Embrace Cloud computing to cut costs, 10.2008, <http://www.eweek.com/c/a/Enterprise-Applications/Forresters-Advice-to-CFOs-Embrace-Cloud-Computing-to-Cut-Costs/>
6. S.Prentice, Infrastructure 2015, Gartner Symposium ITxpo 2005, Conference proceedings, Johannesburg, RSA, 2005.
7. JSON RFC 4627, <http://tools.ietf.org/html/rfc4627>
8. RSS 2.0 Specification, <http://cyber.law.harvard.edu/rss/rss.html>
9. The Atom syndication format, RFC 4287, <http://tools.ietf.org/html/rfc4287>
10. The Atom Publishing Protocol, RFC 5023, <http://tools.ietf.org/html/rfc5023>
11. Load balancing, [http://en.wikipedia.org/wiki/Load_balancing_\(computing\)](http://en.wikipedia.org/wiki/Load_balancing_(computing))
12. Cluster, [http://en.wikipedia.org/wiki/Cluster_\(computing\)](http://en.wikipedia.org/wiki/Cluster_(computing))
13. Multiprocessing, <http://en.wikipedia.org/wiki/Multiprocessing>
14. MapReduce, <http://en.wikipedia.org/wiki/MapReduce>
15. F. Cummins, Enterprise integration – An architecture for Enterprise applications and System Integration, OMG Press, 0-471-40010-6, USA, 2002
16. C.Hsu, Service Enterprise Integration – An Enterprise Engineering Perspective, Springer's Integrated series on Information systems, Springer, 978-0-387-46361-2, USA, 2007
17. Welcome to WSIF: Web Services Invocation Framework, <http://ws.apache.org/wsif/>
18. M. Juric, Using WSIF for Integration, Oracle Technology network, http://www.oracle.com/technology/pub/articles/bpel_cookbook/juric.html
19. Amy Shuen, Web 2.0: A Strategy Guide, O'Reilly, 2008, ISBN-13: 978-0-596-52996-3, 266 pp.
20. Ben Rigby, A Practical Guide to Using Web.2.0 Technologies, Jossey Bess Publishing, 2008, ISBN 978-0-470-22744-2, 268 pp.
21. Jason Whittaker, Producing for Web 2.0, Routledge, 2009, ISBN 13: 978-0-203-88203-0, 272 pp.
22. Gottfried Vossen & Stephan Hagemann, Unleashing Web 2.0, Morgan Kaufmann Publishers, 2007, ISBN: 978-0-12-374034-2, 368 pp.

23. Miltiadis D. Lytras, Web 2.0 - The Business Model, Springer, 2009, ISBN-13: 978-0-387-85894-4, 326 pp.
24. Shreeraj Shah, Web 2.0 Security: Defending AJAX, RIA, and SOA, Course Technology, 2008, ISBN-10: 1-58450-550-8 , 385 pp.
25. Susan Rice Lincoln, Mastering Web 2.0, Kogan Page, 2009, ISBN 978 0 7494 5466 1, 225 pp.
26. P. Candace Deans, Social Software and Web 2.0 Technology Trends, Information Science Reference, 2008, ISBN 978-1-60566-122-3, 250 pp.
27. Vince Casarez et al., Reshaping Your Business with Web 2.0, McGraw-Hill, 2009, ISBN 0-07-164366-4, 290 pp.

**МЕТОДИ ЗА ПРОЕКТИРАНЕ НА БИЗНЕС ИНФОРМАЦИОННИ СИСТЕМИ НА
БАЗАТА НА WEB 2.0 И CLOUD COMPUTING****Резюме:**

Настоящата студия предлага методи за проектиране на бизнес информационни системи на базата на най-актуални и авангадни в момента информационни технологии – Web 2.0 и Cloud Computing. Представени са предназначението, същността, архитектурните компоненти и характеристиките на тези две технологии, като специално внимание е отделено на приложението им в проектирането на бизнес информационни системи. Изяснени са изискванията към софтуера, необходим за разработване на такъв вид приложения. Предложени са методи за проектиране на бизнес информационни системи съответно на базата на Web 2.0 (Методи за проектиране чрез различни шаблони - SOA, SaaS, Участие – Съвместна работа, Mashup, Синхронизиран Уеб, Структурирана информация, Потребителска многофункционалност) и Cloud Computing (Метод за проектиране на Мрежа като услуга, Метод за проектиране на Изчислителен ресурс като услуга, Метод за проектиране на Приложение като услуга).

**DESIGN METHODS FOR BUSINESS INFORMATION SYSTEMS
ON WEB 2.0 AND CLOUD COMPUTING BASIS****Abstract:**

The paper proposes methods for designing business information systems on the basis of the most actual and sophisticated at the present moment information technologies – Web 2.0 and Cloud Computing. The purpose, essence, architectural components and the characteristics of those two technologies are presented with a special emphasis on their application in the design of business information systems. The requirements to the software needed for the design of business information systems are defined. Methods for designing business information systems are proposed correspondingly on the basis of Web 2.0 (Design methods through different patterns – SOA, SaaS, Participation – Mutual activity, Mashup, Synchronized Web, Structured Information, User multifunctionality) and Cloud Computing (Design methods for Network as a Service, Computing as a Service, Application as a Service).