Знание и бизнес, 2015

Икономика и Компютърни науки

Научно списание

бр.2 / 2015

Издателство Знание и бизнес
Варна

ИКОНОМИКА И КОМПЮТЪРНИ НАУКИ

Научно списание

Брой 2 / 2015

Редакционна колегия „Знание и бизнес“

проф. д-р ик. н. Петко Щерев Илиев – главен редактор

доц. д-р Светлозар Димитров Стефанов – зам. гл. редактор

доц. д-р Юлиан Андреев Василев – зам. гл. редактор

доц. д-р Анастасия Стефанова Кондукторова – научен секретар

маг. Елис Адемова Салиева – технически секретар

проф. д-р Марин Тодоров Нешков

доц. д-р Павел Стоянов Петров

доц. д-р Събка Димитрова Пашова

доц. д-р Десислава Бориславова Серафимова

гл. ас. д-р Тодор Костадинов Дянков

© НАУЧНО СПИСАНИЕ ИКОНОМИКА И КОМПЮТЪРНИ НАУКИ

брой 2, 2015г.

ЦЕНТЪР ЗА ПРОФЕСИОНАЛНО ОБУЧЕНИЕ „ЗНАНИЕ И БИЗНЕС“

Главен редактор: проф. д-р ик. н. Петко Илиев – тел: 0893 492 350

Зам. гл. редактор: доц. д-р Светлозар Стефанов – тел: 0878 362 245

Зам. гл. редактор: доц. д-р Юлиан Василев – тел: 0882 164 711

Научен секретар: доц. д-р Анастасия Кондукторова – тел: 0882 164 846

Технически секретар: Елис Салиева – тел: 0893 492 352

9000, гр. Варна, бул. „Сливница“ №2, офис 209

email: eknigibg@gmail.com

СЪДЪРЖАНИЕ

Автор: доц. д-р Юлиан Василев
ПРЕДОСТАВЯНЕ НА ЛОГИСТИЧНА ИНФОРМАЦИЯ ЧРЕЗ УЕБ УСЛУГИ..... 7

Автор: ас. Миглена Стоянова
АРХИТЕКТУРА НА СИСТЕМА ЗА ИГРОВИЗАЦИЯ 18

Автор: докторант Невян Нейков
АРХИТЕКТУРА НА ОБЛАЧНИ ОПЕРАЦИОННИ СИСТЕМИ 35

Автор: гл. ас. д-р Тодор Дянков
ОСНОВНИ КОНСТРУКТИВНИ ЕЛЕМЕНТИ ПРИ УПРАВЛЕНИЕ НА ВЗАИМООТНОШЕНИЯТА С КЛИЕНТИ В ТУРИСТИЧЕСКИЯ БИЗНЕС 54

Уважаеми читатели имаме удоволствието да Ви информираме, че по инициатива на редакционния съвет на „Знание и бизнес“ ООД е лицензирано издателство „Знание и бизнес“ на научно списание Икономика и компютърни науки.

Нашата теза е, че информационните технологии намират приложение във всички области на човешката дейност. Ето защо и научното списание Икономика и компютърни науки е отворено за публикации от различни области на науката и практиката.

Убедени сме, че публикациите от списанието ще съдействат за обогатяване на научните знания, както и повишаване на ефективността от използване на информационни технологии.

С уважение: проф. д-р ик. н. Петко Щ. Илиев

Юлиан Василев е доцент д-р в катедра „Информатика” към Икономически университет – Варна.

Завършил е специалност „Информатика“, магистър в ИУ-Варна (1998). В университета постъпва като асистент през 2000 г. Академичен растеж: доктор (2008), доцент (2011). Докторска дисертация: „Информационно осигуряване на производствени логистични процеси“ (2008). Гост лектор в Университета в Марибор (2006), Приморски университет в Копер (2007), Технически университет Вилнюс (2008), Икономически университет в Катовице (2009).

Води лекции и упражнения по „Информатика“, „Информационна логистика”, „Управление на проекти“, „Електронна търговия“, „Счетоводен софтуер“, „Информационни технологии в продажбите“, „Компютърни комуникационни системи“, „ERP systems“, “Business management systems” и “Geographic Information Systems”.

Консултирал е редица фирми в областта на приложението на информационни технологии и системи в логистиката. Автор е на над 70 научни публикации. Разработил е няколко уеб базирани информационни системи за нуждите на учебния процес в Икономически университет – Варна. Интересите му са в областите: информационната логистика, географски информационни системи, бизнес интелигентни системи, счетоводен софтуер и електронна логистика. Участва в няколко национални и международни проекта.

Доц. Василев участва в научния съвет на няколко международни конференции. Доц. Василев е член на редколегиите на няколко международни списания. Доц. Василев е член на експертния съвет към Центъра за социални изследвания при ИУ-Варна

ПРЕДОСТАВЯНЕ НА ЛОГИСТИЧНА ИНФОРМАЦИЯ ЧРЕЗ УЕБ УСЛУГИ

доц. д-р Юлиан Василев
Икономически университет – Варна
E-mail: vasilev@ue-varna.bg

Резюме

Предоставянето на логистична информация може да бъде организирано по различни начини от информационна гледна точка. Цел на настоящата статия е представяне на две методики. Първо, на методика за изграждане на уеб услуги в търговско предприятие, което предоставя онлайн информация за приети поръчки. Второ, методика за използване на уеб услугата в средата на различни софтуерни системи от страна на корпоративни клиенти. В статията е даден програмният код както на сървърната част, така и на клиентската част. Представените методики могат да се използват от софтуерни фирми за изграждане на други уеб услуги в логистиката. Използването на уеб услуги в логистиката води до улесняване на информационните процеси по обмен на логистична информация в реално време между хетерогенни софтуерни системи.

Ключови думи: вериги за доставка, Delphi XE

1. Въведение

Съвременните условия на работа предполагат обмен на информация между участниците във веригите за доставка. Логистичната информация е свързана с движението на материалните потоци. Материалният поток естествено е свързан с информационен поток, който описва неговото движение. Традиционните начини за комуникация чрез хартиени документи, телефонни обаждания и електронна поща съществуват в днешната действителност. Нарастването в интензивността на обменяната информация във веригите за доставка обуславя приложението на нови логистични подходи и информационни технологии за обмен на данни.

Обменът на данни е сложен процес, защото предполага стандартизация на документи, стандартизация на съобщенията, които се изпращат и получават, стандартизация на софтуерните системи, които се използват от отделните участници. В този смисъл следва да се отбележи, че промяната както в логистичната технология за пренос на данни, така и в информационната технология изискват значителни усилия на специалисти от различни области.

Развитието на информационните технологии може да способства за подобряване организацията на логистичните дейности. Следва да се търсят адекватни подходи за адаптация на последните достижения от областта на информатиката в управлението на веригите за доставка. Считаме, че интердисциплинарният подход по отношение на адаптация на информационни технологии в логистичната практика следва да доведе до по-бърз и точен пренос на логистична информация между участниците във веригите за доставка.

Логистичните технологии и системи са насочени предимно към организация на материалните потоци, която да осигури бързото им придвижване и минималното използване на складови площи. Логистичните технологии са свързани с управление на операциите вътре в предприятията, планиране на потребността от материали, планиране на производствените ресурси, планиране на транспорта и дистрибуцията. Наличието на добри практики по управление на логистиката на корпоративно ниво води до съществено подобряване в логистичната дейност. Голяма част от индустриалните стандарти са публикувани за широка употреба и от други предприятия, на пример Just-in-time и Kanban.

Логистичните технологии, касаещи обмен на информация между участниците във веригите за доставка, са насочени предимно към електронен обмен на данни. Основно се прилагат стандартите EDIFCT и XML за обмен на логистична информация между партньорите във веригите за доставка. Стандартът EDIFACT намира широка приемственост в Европейския съюз. Той се използва за описание на документи в електронен формат.

Информационните технологии за пренос на логистична информация са твърде разнообразни. Следва да отбележим електронната поща като един от най-често използваните инструменти за обмен на логистична информация.

Електронната поща се използва както за предаване на структурирана информация (на пример фактури), така и за неструктурирана информация (на пример продуктова спецификация и запитвания за гаранционен сервиз).

Стандарти за електронен обмен на данни (като на пример EDI) намират широка приемственост в стопанската практика. Приложението на стандарта EDI е именно по отношение на обмен на логистична информация между партньорите във веригите за доставка.

Приложението на информационни технологии в логистиката на микро равнище се нарича „информационна логистика“. Приложението на информационни технологии в логистиката на макро равнище се нарича „електронна логистика“. В областта на електронната логистика комуникацията между партньорите се извършва чрез приложение на бизнес модела „бизнес-към-бизнес“ (Business-to-business – B2B).

За реализация на този модел се препоръчва използването на уеб услуги. Уеб услугите са специфична информационна технология, която позволява онлайн комуникация между две или повече разнородни софтуерни системи, инсталирани в различни предприятия. Ето защо в настоящата статия извеждаме необходимостта от използването на уеб услуги за предоставяне на логистична информация.

2. Създаване на уеб услуга от страна на предприятието, което предоставя логистична информация

За илюстриране предоставянето на логистична информация предлагаме да се разработи уеб услуга, която проверява статуса на изпратена поръчка и връща отговор с актуалния статус на поръчката. Предвиждаме уеб услугата да се инсталира в предприятието, което приема поръчки, а отделните клиенти да я използват. Отделните клиенти могат да доразвият своите софтуерни системи, така че да ползват уеб услугата. Използването на уеб услугата спестява разходи, защото отделните корпоративни клиенти имат достъп до онлайн информация, без да е нужно да се логват в онлайн платформата на търговеца.

Създава се база от данни Statuses.accdb. В нея се създава една таблица „Statuses“ с две колони: „DocumentNumber“ и „Status“. Уеб услугата, която създаваме, предвиждаме да прочете статуса за конкретен документ от таблица Statuses и да върне като резултат неговия статус.

Таблица 1.

Съдържание на таблица „Statuses“

DocumentNumber	Status
111	1
222	2
333	3

Използвани са примерни данни. Отделните статуси могат да бъдат „поръчката е приета“, „поръчката е отхвърлена“, „поръчката е изпълнена“, „поръчката е в ход на изпълнение“. В таблица „Statuses“ се използват кодирани значения на статусите. Предвиждаме резултатът от уеб услугата да се дава като число, за да се унифицира изходът на уеб услугата и за да се ускори преносът на данни.

За създаването на сървърната част на уеб услугата се създава папка „WS Check status SERVER“, в която да се съхрани проектът на сървърната част.

Стартира се Delphi XE. В средата на Delphi XE се изпълняват следните стъпки:

1) Създава се ново SOAP сървърно приложение (File/New/Other/Delphi projects/Web Services/SOAP server application). Избира се „CGI-Standalone executable“.

2) Създава се сървърният интерфейс на SOAP услугата (File/New/Other/Delphi projects/Web Services/SOAP server interface). Задава се име на интерфейса „CheckStatus“.

3) Отваря се модулът, описващ сървърния интерфейс (CheckStatusIntf.pas), за да се въведе декларацията на сървърната функция.

След редовете:

ICheckStatus = interface(IInvokable)

['{BD0C49D7-6EDF-4AAF-941B-6C5007E5A60D}']

се добавя декларацията на разработваната сървърна функция:

function Check_status(aDocument_Number : Integer) : Integer; stdcall;

4) Въвежда се описанието на услугата. Отваря се модулът, описващ програмния код на сървърната функция (CheckStatusImpl.pas).

След реда „public" отново се декларира разработваната функция.

function Check_status(aDocument_Number : Integer) : Integer; stdcall;

За да се генерира автоматично процедурата, където да се напише програмния код, се избира команда <Ctrl> + <Shift> + <C>.

```
function TCheckStatus.Check_status(aDocument_Number: Integer): Integer;
begin

end;
```

Между редовете „begin" и „end" трябва да се въведе програмният код, който извлича статуса на поръчката от базата от данни на търговеца.

```
function TCheckStatus.Check_status(aDocument_Number: Integer): Integer;
var
  ADOConnection : TADOConnection;
  ADOQuery      : TADOQuery;
  DataSource    : TDataSource;
  SQL           : string;
begin
  Result := 0;
  //
  ADOConnection := TADOConnection.Create( nil );
```

```
//
ADOQuery      := TADOQuery.Create( nil );
//
DataSource    := TDataSource.Create( nil );
//
try
 //
 ADOConnection.ConnectionString                                    :=
'Provider=Microsoft.ACE.OLEDB.12.0;Data  Source=Statuses.accdb;Persist  Security
Info=False';
 //
 ADOQuery.Connection := ADOConnection;
 //
 DataSource.DataSet := ADOQuery;
 //
 SQL := format( 'Select status from statuses where DocumentNumber = %d ',
[ aDocument_number ] );
 //
 ADOQuery.SQL.Add( SQL );
 //
 Try // Опитвам се да изпратя заявката до базата от данни
  ADOQuery.Open; //
 except // Ако възникне изключение
  on E : Exception do
    begin //
     // labMessage.Caption := E.Message;
     Exit; // Излизам от процедурата
    end;
 end; // try - except

 // функцията връща като резултат статуса на поръчката
```

```
    Result := DataSource.DataSet.Fields[ 0 ].AsInteger;

  finally
    ADOConnection.Free;
    ADOQuery.Free;
    DataSource.Free;
  end; // try-finally
end; // procedure
```

Сървърният модул „WS_Check_status_SERVER“ се компилира (<Ctrl> + <F9>).

След това компилираният модул „WS_Check_status_SERVER.exe“ и базата от данни „Statuses.accdb“ се копират в папка „cgi-bin” на Apache сървър.

За да се провери дали уеб услугата е инсталирана на сървъра, в браузър трябва да се напише:

http://localhost/cgi-bin/WS_Check_status_SERVER.exe

За да се използва уеб услугата в приложения от тип клиент-сървър, вместо „localhost“ трябва да се напише IP адреса на сървъра.

В уеб браузър може да бъде разгледано описанието на уеб услугата като XML документ.

http://localhost/cgi-bin/WS_Check_status_SERVER.exe/wsdl/ICheckStatus

Освен стандартизираното описание на уеб услугата с езика WSDL, в браузъра се вижда и декларацията на функцията:

```
int      Check_status(int aDocument_Number)
```

Това, което вижда програмистът (който може да използва функцията), е, че тя приема като вход променлива от числов тип и връща като резултат число. Самата бизнес логика не е видима за програмиста, който пише клиентската част – софтуерното приложение, което ползва уеб услугата.

3. Създаване на софтуерно приложение, което да използва създадената уеб услуга

Създадената уеб услуга следва да бъде качена на сървър с достъп до интернет. Всяко предприятие, което желае да провери статуса на своята поръчка, може да доработи собственото си софтуерно приложение посредством използването на стандартизиран подход за достъп до уеб услугата. Този факт е изключително важен, защото всяко едно предприятие използва собствена софтуерна система. Следва да се подчертае, че всяка една софтуерна система е изградена с различна среда за разработка. Различните софтуерни системи могат да се интегрират чрез различни информационни и комуникационни технологии. В настоящата статия се предлага един от множеството възможни подходи, а именно – за интеграция на хетерогенни софтуерни системи (които обменят логистична информация) чрез използване на уеб услуги.

Уеб услугата, описана в настоящата статия, е създадена в средата на Delphi XE. За илюстриране на използването на уеб услугата отново използваме Delphi XE, въпреки че клиентската част може да бъде изградена с всяка една среда за разработка.

Последователността на работата е следната:

1) Създава се папка „WS Check status CLIENT“. Създава се нов проект (File/New/VCL forms – Application Delphi). Проектът се съхранява като „WS_Check_status_CLIENT.dproj“.

2) Вмъква се описанието на уеб услугата.

(File/New/Other/Delphi projects/Web Services/WSDL importer)

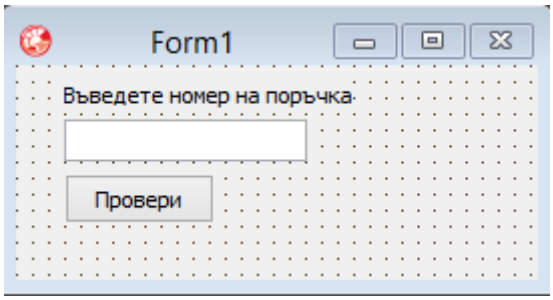
В полето за WSDL файл или URL се записва:

http://localhost/cgi-bin/WS_Check_status_SERVER.exe/wsdl/ICheckStatus

Delphi XE създава нов модул „ICheckStatus1.pas“.

3) Изграждане на потребителския интерфейс.

Върху формуляра (Form1) се поставя един етикет (TLabel), една текстова кутия (TEdit) и един бутон (TButton).



Фиг. 1. *Изглед на клиентската част в режим на проектиране*

4) Създаване на събитийна процедура, която да се изпълни при натискане на бутон „Провери“.

```
procedure TForm1.Button1Click(Sender: TObject);
var
  aDocument_number,
  Status : Integer;
  St     : String;
begin
  // конвертирам текстово поле в число
  aDocument_number := StrToIntDef( Edit1.Text, 0 );
  // проверявам статуса на конкретна поръчка
  Status := GetICheckStatus().Check_status( aDocument_number );
  // подготвям съобщението за крайния потребител
  St := format( 'Статусът на поръчката е %d.', [ Status ] );
  // извеждам съобщението
  ShowMessage( St );
end;
```

Клиентската част на софтуерното приложение се компилира и стартира.

Заключение

Използването на уеб услуги в логистичната практика създава предпоставки за усъвършенстване на корпоративните софтуерни системи. В настоящата статия са показани две методики. Първо, за създаване на уеб услуга, която проверява статуса на конкретна поръчка. Уеб услугата се инсталира на сървър, на който е инсталиран уеб сървър Apache. Второ, дадена е методика за надграждане на съществуващо софтуерно приложение, което да използва уеб услугата.

Представените методики могат да се използват както от други изследователи, така и от софтуерни фирми за изграждане на други уеб услуги в логистиката. Двете методики могат да се адаптират в широк кръг от фирми, което води до улесняване на информационните процеси по обмен на логистична информация в реално време между хетерогенни софтуерни системи.

Литература

1. Vasilev, J. Web service for calculating the probability of returning a loan – design, implementation and deployment. Informatica Economica Journal, Vol. 18, No 4, 2014, p. 111-119. <http://www.revistaie.ase.ro/content/72/10%20-%20Vasilev.pdf>.

2. Василев, Ю. Електронна комуникация във веригите за доставка. Икономиката в променящия се свят: национални, регионални и глобални измерения. Сборник с доклади от международна научна конференция, том III, с. 148-154. Available: http://conference.ue-varna.bg/2015/bg/Tom_3.pdf

3. Embarcadero Technologies. www.embarcadero.com

Ас. Миглена Стоянова е преподавател от катедра „Информатика“, факултет „Информатика“ при Икономически университет – Варна. Родена е през 1977 г. в гр. Шумен. Завършва спец. „Информатика“, магистър, ШУ „Еп. Константин Преславски“, Шумен, 2001 г. Работи като преподавател в ШУ „Еп. Константин Преславски“, Шумен (2001 – 2006) и като хоноруван преподавател в Икономически университет – Варна (2006 – 2009).

Академичен научен растеж: Докторант редовно обучение, научна специалност 05.02.08 „Приложение на изчислителната техника в икономиката“ (2006 – 2009), Икономически университет - Варна. Асистент към катедра „Информатика“, Икономически университет – Варна (2009).

Научни интереси: игровизация, електронно обучение, уеб технологии.

АРХИТЕКТУРА НА СИСТЕМА ЗА ИГРОВИЗАЦИЯ

ас. Миглена Стоянова
катедра "Информатика", Икономически университет – Варна
e-mail: m_stoyanova@ue-varna.bg

1. Въведение

Игровизацията е сравнително нов подход в областта на информационните системи, който се стреми към постигане на по-голяма ангажираност на участниците в даден процес, промяна на поведението им и стимулиране на иновациите. Тя се дефинира като използване на елементи от игровия дизайн в неигрови контекст[1]. Така например даването на бонус точки на лоялни клиенти се счита за вид игровизация. В контекста на модела бизнес-към-потребител (B2C), целта на адаптирането на игрови модели в съществуващи софтуерни системи е да се повиши ангажираността на клиентите към компанията. В последно време, този подход предизвиква интерес и в сферата на предприятията, където въвеждането на игрови механики в работата води до по-високи нива на ангажираност на служителите, а оттам и до положителни резултати за самата организация. Игровизацията се прилага като допълнителен аспект към софтуерни продукти, базирани на концепциите ERP, CRM или SCM, които по настоящем са широко използвани на работното място. Тези системи осигуряват подходяща среда за пренос и посредничество на игрово преживяване за крайните си потребители. Редица скорошни изследвания показват, че игровизацията води до постигане на желаните подобрения в удовлетвореността на клиентите от психологическа гледна точка ([2],[3]).

Според прогнози на някои консултантски и изследователски компании, глобалният пазар за игровизирани приложения и услуги се очаква да нарасне

[1]Deterding, S., Dixon, D., Khaled, R. and Nacke, L. From Game Design Elements to Gamefulness: Defining Gamification. In MindTrek '11 Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments. ACM, 2011.
[2]Herzig, P., Strahringer, S. and Ameling, M. Gamification of ERP Systems – Exploring Gamification Effects on User Acceptance Constructs. Multikonferenz Wirtschaftsinformatik. GITO, 2012.
[3]Thom, J., Millen, D. and DiMicco, J. Removing Gamification from an Enterprise. Proceedings CSCW, 2012.

през следващите години[4]. Тези предвиждания са причина за появата на множество технологични решения и софтуерни проекти с отворен код, предлагащи платформи за игровизация. Те могат да се използват за добавяне на мотивиращи игрови елементи като мисии, класации или нива в уеб сайтове. Въпреки това, тези решения имат някои недостатъци при прилагане в големи корпоративни системи или бизнес-към-бизнес (B2B) сценарии за интеграция.

Настоящата статия представя архитектура на система за игровизация за корпоративни системи. За тази цел са използвани изследвания от областта на системните архитектури. Въз основа на предложената архитектура са изведени основни насоки, които могат да бъдат разгледани в бъдещи изследвания.

2. Съществуващи изследвания в областта на игровизацията

Изследванията по отношение на платформите за игровизация са все още малко на брой. Независимо от това, съществуват реални подходи за адаптация на игрови модели в съществуващи софтуерни системи, които се използват от специалистите в практиката.

В свое изследване, Xu[5] представя две платформи с отворен код (Mozilla Open Badges, Userinfuser) и множество комерсиални платформи (Bunchball, Badgeville, BigDoor и др.). Първата група платформи е предназначена за програмно въвеждане на прости игрови механики като значки или точки в уеб сайтове. По тази причина, тази група не отговаря на изискванията на предприятията. Втората група платформи (комерсиалните) са цялостни платформи за игровизация, предназначени предимно за големи клиенти. Те също са фокусирани предимно върху уеб сайтове и B2C пазара. За разлика от решенията с отворен код, доставчиците на платформи като Badgeville или Bunchball предлагат допълнителни приложно-програмни интерфейси (ППИ) като уеб услуги и инструменти за анализ на данните от игровизацията. Това води до големи софтуерни проекти за интеграция, при които съществуващото приложение трябва да бъде значително променено, за да може игровият модел да

[4]A Brief History of Gamification: Part VI – The Predictions. <http://edulearning2.blogspot.bg/2014/05/a-brief-history-of-gamification-part-vi.html>, (13.10.2015).
[5]Xu, Y. Literature Review on Web Application Gamification and Analytics. <https://csdl-techreports.googlecode.com/svn/trunk/techreports/2011/ 11-05/11-05.pdf>, (13.10.2015)

бъде част от съществуващия приложен софтуер. Освен това, тези усилия за интегриране в редица случаи водят до изолирани (несъвместими) приложения, които е желателно да бъдат избягвани в една корпоративна информационна система. При това, платформи като Badgeville и Bunchball са достъпни само чрез абонаментен модел (SaaS), т.е. доставчикът на услугата притежава данните от игровизацията, което е много критично в някои предприятия.

Както отбелязахме, BigDoor е една от платените платформи за игровизация. BigDoor прилага по-различен подход спрямо Badgeville и Bunchball. Тази платформа предлага адаптивна JavaScript странична лента за уеб сайтове. Въпреки че този подход намалява времето за интегриране на интерфейса, интегрирането на персонализиран потребителски вход не е предвидено. По този начин, игровите механики са много ограничени (например точки за споделяне на съдържание или за посещения на уеб сайт) и подходяща бекенд или мидълуер интеграция е извън обхвата на тази платформа. Изводът, който може да се направи, е че всяка платформа е разработена на базата на уникална теоретична концепция. Ето защо е трудно да се изведе обща теоретична концепция за игровизацията само на базата на емпирично разглеждане на няколко софтуерни платформи за игровизация. В този смисъл е трудно да бъде направена обобщена оценка на ефективността на платформа за игровизация, тъй като всяка платформа обикновено е ориентирана към решаване на специфични задачи в конкретно предприятие.

3. Изисквания за системата за игровизация

Нефункционалните изисквания за информационните системи в предприятията се различават в голяма степен от десктоп или уеб приложенията, поради тяхната тясна връзка с бизнес процесите, организационната структура или бизнес модела, който поддържат[6]. Една организационна архитектура трябва да бъде проектирана с висока степен на гъвкавост и приспособимост, за да реагира бързо на променящите се изисквания на бизнеса. Някои от основните

[6]Krafzig, D., Banke, K. and Slama, D. Enterprise SOA - Service-oriented Architecture Best Practices. Prentice Hall International, 2005.

задачи на проектирането са простота, повторна употреба и отделена функционалност ([7],[8]).

В този контекст, концепциите за **Архитектура ориентирана към услуги (Service Oriented Architecture, SOA)** и **Архитектура управлявана от събития (Event Driven Architecture, EDA)** осигуряват най-добрите общоприети практики и модели за постигане на поставените задачи[9].

Архитектурата, ориентирана към услуги, се изгражда в предприятията с оглед въвеждане на автономни и оперативно съвместими услуги, които предлагат бизнес функционалност за многократна употреба чрез стандартизирани интерфейси. Тези услуги следва да бъдат свободно интегрирани и осигуряват, от своя страна, оркестрация на бизнес процесите. SOA обикновено е изградена с помощта на синхронни модели от вида заявка-отговор. Освен това, местоположенията на услугите и интерфейсите трябва да бъдат предварително известни. При EDA, всички участници комуникират посредством събития през модул за обмен на съобщения (message broker). Корпоративната система действа или като „издател“ на събития (източник на събития) или като „слушател“ на събития (приемник на събития). Модулът за обмен на съобщения в архитектурния модел EDA изпраща публикуваните събития до всички заинтересовани абонати. Освен това, бизнес правилата могат да бъдат изведени от корпоративната информационна система чрез децентрализирани модули за правила или процесори за комплексни събития (Complex Event Processors). За разлика от SOA, участниците (отделните софтуерни системи в рамките на EDA) са напълно отделени един от друг, което води до по-голяма автономност на отделните софтуерни системи и по-добра възможност за интеграция на данни между тях.

Въпреки това, при EDA липсва ясно дефиниране на структурата на компонентите, интерфейсите и договорите за данни. Принципите на SOA могат да помогнат в EDA контексти за успешното структуриране на всички системни

[7]Пак там.
[8]Erl, T. SOA Design Patterns. Prentice-Hall, 2009.
[9]Levina, O. and Stantchev, V. Realizing Event-Driven SOA. Int. Conf. on Internet and Web Applications and Services. Venice/Mestre: IEEE, 2009, pp. 37-42.

инстанции като услуги. Поради това, някои изследователи предлагат комбинирането на двата подхода в т.нар. Event – Driven SOA (ED – SOA)[10].

Този подход може да се използва и при предлаганата от нас система за игровизация. Нейната цел е да въведе игровизацията в съществуващи или нови бизнес приложения по възможно най-лесен и икономичен начин. Предвиждаме системата за игровизация да включва компонентите **система за управление на бизнес правила (СУБП), компонент за игрови правила и механики, модул за правила, модул за анализ** и **хранилище за игровизация.** Те могат да бъдат проектирани в смисъла на услуги, с предварително дефинирани и структурирани интерфейси и договори за обмен на данни. Комуникацията помежду им се предвижда да се осъществява с помощта на събития.

Общата цел на организационните архитектури и софтуер е да осигурят бързодействие, гъвкавост и ефективност, които са необходими, за да отговорят на постоянните промени в една организация. Въз основа на своите наблюдения, група автори[11] формулират основни изисквания за корпоративни системи: простота, гъвкавост, повторна употреба, разделяне на функционалност от технология. Системата за игровизация, като вид софтуерна система, е желателно да отговаря на тези изисквания.

Въз основа на направения преглед на литературата, са идентифицирани допълнителни изисквания, които са специфични за областта на игровизацията. На първо място, една система за игровизация трябва да осигурява незабавна обратна връзка, цели / предизвикателства и правила за своите потребители[12]. Докато второто и третото условие са свързани с функционалните изисквания, изготвени от бизнес потребителите във връзка с бизнес процесите, първото изискване е нефункционално и е основна цел при проектирането на EDA. На второ място, системата за игровизация следва да предоставя категории / нива, времеви ограничения, работа в екип, виртуални стоки, подаръци, пазари и

[10]Luckham, D. SOA, EDA, BPM and CEP are all Complementary - Part 1, 2012, <http://complexevents.com/wp-content/uploads/2007/05/SOA_EDA_ Part_1.pdf>, (10.07.2015).
[11]Krafzig, D., Banke, K. and Slama, D. Enterprise SOA - Service-oriented Architecture Best Practices. Prentice Hall International, 2005.
[12]McGonigal, J. Reality is Broken: Why Games Make Us Better and How They Can Change The World. The Penguin Press, New York, 2011.

състезания между участниците[13]. Други научни трудове[14] разглеждат игрови механики като точки, класации, значки, социални компоненти или персонализиране на интерфейса. Важно е да отбележим, че тези механики описват главно контекстуалната част на системата за игровизация, която може да се използва от дизайнер на игровизация или бизнес потребител. Дизайнерът на игровизацията е отговорен за проектирането на концепцията за игровизация. Той използва психологическите модели и методи, както и методологиите и средствата на игровия дизайн.

В т. 4 „Архитектура на система за игровизация“ е представена архитектурата на системата за игровизация, въз основа на посочените изисквания.

4. Архитектура на системата за игровизация

На софтуерния пазар се предлагат редица платформи за игровизация. Те реализират само отделни изисквания от представените в т. 3 и не са толкова гъвкави, колкото се изисква от предприятията. Предлаганата от нас архитектура на системата за игровизация се базира на комбинирането на двата архитектурни модела SOA и EDA. Тя е показана на фигура 1, без заобикалящата я среда.

Всички действия, които са свързани с игровата обработка в системата за игровизация, могат да бъдат представени чрез събития. Например, ако даден потребител / играч **u** в предприятието успешно завърши стъпка **x** в процес **p**, тогава съществува събитие **E**, което представя и описва всички параметри **E=(u, x, p)**.

Модулът за правила съхранява **n** игрови правила $R=(r_1, ..., r_n)$. Предвиждаме той да регистрира и оценява комплексни събития (например времеви оператори като *after* или *before*).

Игровите правила се управляват от Система за управление на бизнес правила (СУБП). Тя може да получава правила от източник, външен за системата за игровизация и да ги управлява от нейно име. Всяко игрово правило е *if-then*

[13]Reeves, B. and Read, J. Total Engagement: Using Games and Virtual Worlds to Change the Way People Work and Businesses Compete. Harvard Business Press, Boston, MA, 2009.

[14]Zichermann, G and Cunningham, C. Gamification by Design: Implementing Game Mechanics in Web and Mobile Apps. O'Reilly Media, 2011.

условие от вида $P \Rightarrow C$, където следствието C се изпълнява веднага, след като предпоставката P стане истина (е изпълнена). Модулът за правила оценява всяко събитие, в съответствие с предварително дефинираните си правила. Например, ако има правило, което награждава потребител с пет точки за завършване на дадена стъпка в един процес, модулът за правила изпраща събитие награда за този потребител през модула за обмен на съобщения. Това събитие се обработва в хранилището за игровизация и се съхранява в неговата база от данни.

Фиг. 1. *Архитектура на системата за игровизация*

Компонентът за игрови правила и механики предоставя различни игрови механики за поддържане на играта. Те могат да включват инструменти за създаване на обекти в играта и придружаващата логика за взаимодействие с тях. Игровите механики осигуряват средства за награждаване на играчите (бизнес потребителите, които участват в игровизацията) с точки, за осъществяване на търговия, за комуникация между играчите и т.н. Основните възможности на игровите механики са мониторинг на играчите в играта, например проследяване на прогреса на играч, обявяване на постижения (например играч X е победител на месеца) и предоставяне на класации.

Модулът за анализ може да се използва за анализиране на поведението на играчите, с цел подобряване на игровите правила и оптимизиране на дългосрочното ангажиране.

Всички услуги на системата за игровизация са представени чрез модел на услуги[15], за да се осигури независим договор за обмен на данни между всички комуникиращи си софтуерни системи. Хранилището за игровизация представлява регистър за състояния, който позволява реализацията на услуги без запазване на състоянието. Модулът за правила е свързан с модела за централизация на правила[16]. Шината на услуги в предприятието свързва системата за игровизация със софтуерните системи на предприятието. Тя дава възможност за обмен на данни с различен формат между отделните софтуерни приложения[17].

Следва да представим заобикалящата среда на системата за игровизация. Предполагаме, че наследените системи (legacy systems) принадлежат към един от следните архитектурни типове или, по-вероятно, към техни комбинации:

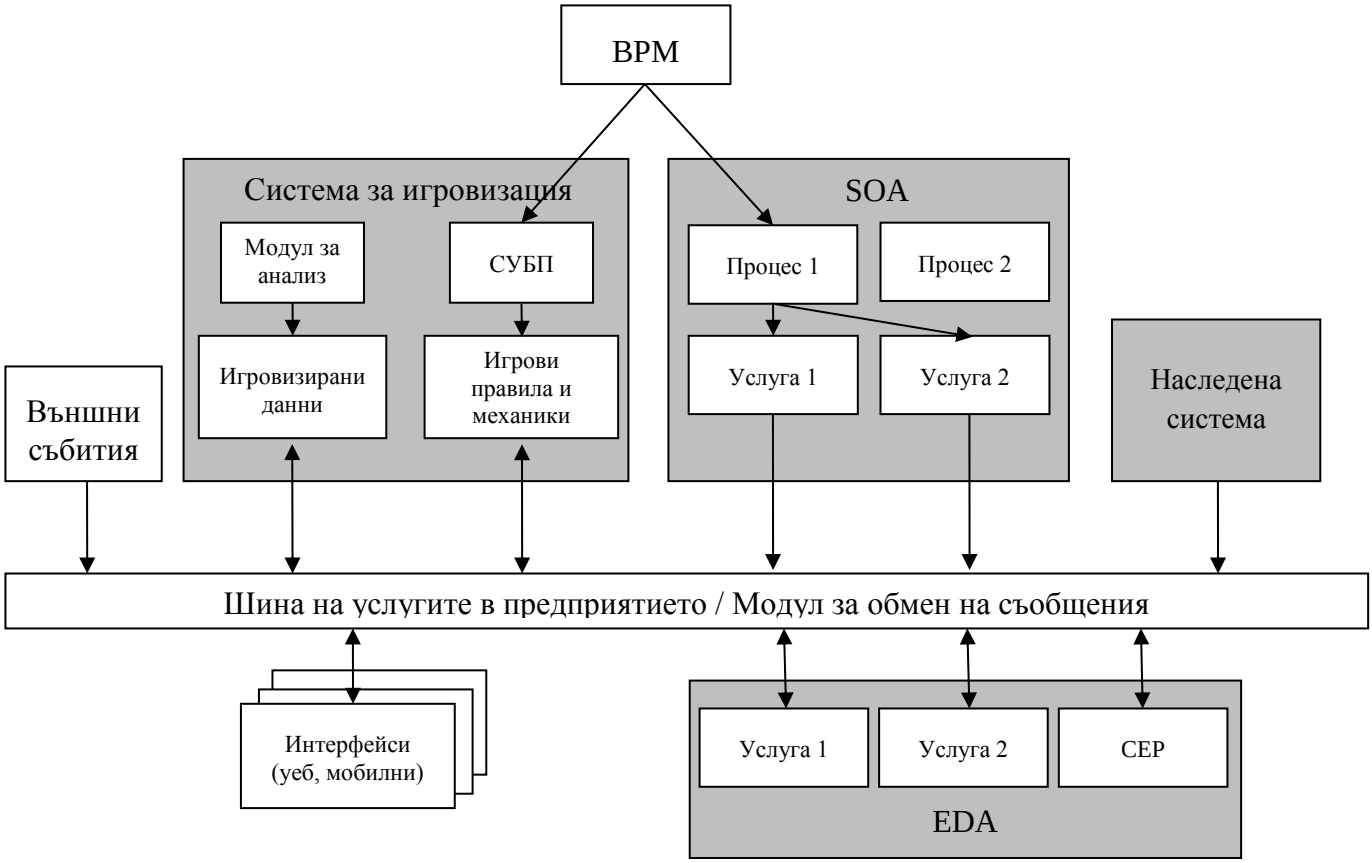
- SOA, където услугите са оркестрирани от решение за управление на бизнес процеси (BPM) и са представени от своите интерфейси;
- Наследена система, която е монолитен блок с произволно сложна функционалност;

[15]Erl, T. SOA Design Patterns. Prentice-Hall, 2009, p. 333.
[16]Пак там, p. 216.
[17]Пак там, p. 671-693.

- Архитектура управлявана от събития, в която CEP контролира хореографията между услуги и/или интерфейси.

На фигура 2 е изобразен примерен вариант за интеграция на системата за игровизация с различни архитектури на корпоративни информационни системи. Фигура 2 представя общ случай на употреба, в който системата за игровизация е интегрирана в хипотетично предприятие, в което са внедрени корпоративна информационна система (ИС) със SOA архитектура, корпоративна ИС с EDA архитектура и корпоративна ИС с общо предназначение. Като се имат превид горните предположения, интеграцията може да се осъществи чрез модула за обмен на съобщения или шината на услуги в предприятието. Освен това, наследените системи (съществуващите софтуерни системи в предприятието) могат да бъдат интегрирани с наследена обвивка (legacy wrapper)[18].

[18]Наследената обвивка (обвивка на наследен модул) е код, който капсулира или „обвива“ съществуващ софтуерен модул и му дава възможност да бъде достъпен като услуга. Кодът на обвивката работи като интерфейс към функциите на съществуващия софтуерен модул.



Фиг. 2. *Интеграция на системата за игровизация с различни архитектури на корпоративни информационни системи*

В допълнение, шината на услуги може да прихване произволни външни събития, идващи от софтуерни системи извън предприятието. Като специализирана форма на тези външни събития сме представили интерфейсите на съответните корпоративни информационни системи. Те могат да бъдат уеб, десктоп или мобилни клиенти, пренасящи например събития на потребителския интерфейс към целевата система. Освен че са „издатели", тези интерфейси също са „абонати" на съответни събития, генерирани от игровия модул, например, за показване на съобщение, че потребителят е достигнал ново ниво. Освен това, е необходимо интегриране на BPM на предприятието със СУБП на системата за игровизация, за да се осигури висока гъвкавост и приспособимост на бизнес

потребителя към игровизацията. Правилата на играта трябва да могат да се променят често от бизнес потребителите, във връзка с техните изисквания.

Всички събития се публикуват в шината на услуги в предприятието, която се грижи за маршрутизацията, посредничеството на услуги и сигурността. Тя е от полза не само при интегрирането с целевата система (системата, в която се въвежда игровизация), но и най-вече когато трябва да се игровизират процеси в множество предприятия. Разработчиците могат да дефинират събитията, които се предават в рамките на предприятието, събитията, приложими при проектирането на игровизацията и управлението на сигурността.

5. Насоки за бъдещи изследвания

Въз основа на представената архитектура на системата за игровизация, можем да определим следните насоки, които могат да бъдат изследвани.

1) **Интеграция на интерфейсите**. Предложената архитектура се фокусира върху безпроблемна, повторно използваема и свободна интеграция на игровизацията със съществуващи корпоративни софтуерни приложения. Това обаче се отнася само за мидълуера и бекенда. Представянето на игровата динамика (например потребителския статус) в съществуващите интерфейси изисква значително време за интегриране. Както беше споменато, системата за игровизация трябва да работи с голямо разнообразие от хетерогенни устройства и технологии. Предвиждаме този проблем да бъде разгледан, чрез предлагане на динамично и платформено независимо генериране на библиотеки, които дават възможност на разработчиците за бързо въвеждане на игровизация от страна на клиента. Представеното съдържание трябва да се променя динамично, когато моделът на данни или правилата се променят в бекенда, за да отговори на високата приспособимост на игровизацията в предприятията.

2) **Мобилна игровизация**. Освен традиционните уеб или десктоп интерфейси, мобилните устройства също трябва да се поддържат от системата, тъй като преносимите компютри и игровизацията осигуряват огромни възможности за взаимодействия. Контекстът на използване на подобни устройства обаче значително се различава от традиционните устройства. По-

точно, устройствата могат да бъдат в три състояния: *свързани*, където скоростта на връзката обикновено е ниска; *отделени*, когато няма връзка с бекенда; *синхронизиращи се*, когато междинните данни се синхронизират с бекенда. Използването на централизиран модул за правила във всички случаи може да се окаже неподходящо, тъй като това нарушава изискването за незабавна обратна връзка. Поради това, може да се изследва подход за мобилен модул за правила, при който конкретен избор на подмножество от всички правила от бекенда се изпълнява на мобилното устройство веднага, а последствията се синхронизират асинхронно с бекенда по-късно. Освен това, трябва да се има предвид, че при този подход изчислителната мощ за правилата е ограничена при подобни устройства и е възможна само за правила, които могат да бъдат оценени локално на устройството.

3) **Дизайн на правила и BPM интегриране**. За да се постигне добър баланс между бизнеса и ИТ, правилата трябва да могат да се създават и променят от бизнес потребители без технически познания. Това е важно изискване, тъй като правилата на играта се променят често, за да се подобри мотивационния дизайн на приложението. Следователно, трябва да съществува един специфичен език за областта, който да е лесно разбираем и разширяем за специфични бизнес сценарии. В литературата се срещат различни класификации на игрови механики, което води до неоднозначно дефиниране на термини. Но за да се опише такъв език, преди това формално трябва да се дефинират действителните игрови механики и тяхната семантика.

Правилата за игровизация трябва да се прилагат в контекста на самия етап на моделиране на бизнес процеси. Затова стремежът е към тяхното интегриране с BPM. Но някои автори[19] отбелязват, че интегрирането на управлението на бизнес правила за игровизация с BPM може да намали възможността за тяхната промяна, тъй като бизнес процесите е малко вероятно да се променят много често, за разлика от бизнес правилата. Освен това, включването на бизнес правила, насочени към игровизация, в бизнес процесите води до трудно

[19]Kharbili, M. E. and Keil, T. Bringing Agility to Business Process Management: Rules Deployment in an SOA, Emerging Web Services Technology - Volume III, 2010, p.157-170.

разбираеми процеси. Затова, учените предлагат бизнес правилата да се формализират като услуга, която се извиква от BPM модула, когато е необходимо.

4) **Модул за анализ**. До тук определихме изисквания и насоки само за фазите на планиране и изпълнение на един проект за игровизация. Според нас, за да се подобри концепцията за игровизация, е необходимо постоянното измерване на степента на ангажираност на потребителите към системата за игровизация. Един аналитичен инструмент следва да може да измерва прости характеристики, например степен на участие или повторни посещения, както и понятия от по-високо ниво, като например ангажираност на потребителите. Системата може да помогне на бизнес потребителите да разберат причините за намаляване на участието на хората или за тяхното напускане на системата.

5) **Конфиденциалност**. Конфиденциалността е основен проблем когато става въпрос за игровизация, поради високите нива на прозрачност на потребителски данни. Всяко потребителско действие в системата за игровизация може да бъде проследено и евентуално възстановено. Въпреки че това може да бъде от полза за топ мениджмънта, в много страни събирането на конфиденциални данни за служителите е забранено. В системата следва да съществува механизъм за сигурност, който позволява санкциониран достъп до данни само на определени лица. Другите могат да имат достъп само до анонимни или обобщени данни.

В повечето случаи на игровизация, играчите трябва да си сътрудничат или да се конкурират в системата. По тази причина, следва да има определена видимост между потребителите. В предложеното технологично решение предвиждаме потребителите да имат три възможности за осигуряване на тяхната конфиденциалност.

Първо, да могат да изберат аватар, отразяващ тяхната виртуална идентичност, който се използва при показване на потребителски данни от игровизацията на други хора. Вътрешно, виртуалната самоличност е свързана с тази от реалния свят на потребителя.

На второ място, да могат да изключат игровизацията, т.е. да не се събират повече данни, свързани с нея.

Трето, при получаването на първото събитие от игровизацията за всеки потребител, системата да генерира съобщение към интерфейсите, задействащо покана, която да информира потребителя за събирането на данни. Ако потребителят откаже тази покана, системата за игровизация спира събирането на данни.

6) **Мултиарендност**. За да се намалят общите разходи за притежаване, системата за игровизация трябва да бъде интегрируема и с облачни среди. Този въпрос е свързан с управлението на мултиарендността и най-вече с оценяването на игровите правила в модула за правила, поради факта, че базата винаги запазва състоянието. Маркираният проблем дори се задълбочава, ако правилата съдържат времево поведение (например правила за закъснение). В тази област могат да се намерят единични изследвания, които да оценяват подходите за управление на бизнес правила по заявка, които отговарят на изискванията за приспособимост на SOA към нуждите на конкретно предприятие. Допълнително изследване може да адаптира принципите за мултиарендност на базите от данни, за да опише съответни подходи за решение на поставения проблем.

В настоящата система за игровизация, множество ползватели (различни предприятия) използват един единствен модул за правила, тъй като всички те си сътрудничат при едни и същи правила. Предлагаме всеки наемател да може да дефинира свой собствен набор от правила, отделно от друг общ набор от правила, който се прилага за всички наематели. В този случай, конфликтите могат да бъдат разрешени в съвместни или конкурентни сценарии. Освен това, могат да съществуват няколко екземпляра (инстанции) на модула за правила, за да се подобри производителността.

6. Заключение

Прилагането на игровизацията в корпоративните информационни системи предлага широк кръг от теми с практическа значимост за изследователи и практици. Те са изведени от състоянието на организационните архитектури и

изискванията за игровизация. В настоящата статия е предложена обща архитектура на система за игровизация и нейното интегриране в произволна корпоративна информационна система. Бъдещи изследвания могат да проучат мобилната сфера, както и сценарии за интеграция на интерфейсите.

Използвана литература

1. A Brief History of Gamification: Part VI – The Predictions. <http://edulearning2.blogspot.bg/2014/05/a-brief-history-of-gamification-part-vi.html>, (13.10.2015).

2. Deterding, S., Dixon, D., Khaled, R. and Nacke, L. From Game Design Elements to Gamefulness: Defining Gamification. In MindTrek ’11 Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments. ACM, 2011.

3. Erl, T. SOA Design Patterns. Prentice-Hall, 2009.

4. Herzig, P., Strahringer, S. and Ameling, M. Gamification of ERP Systems – Exploring Gamification Effects on User Acceptance Constructs. Multikonferenz Wirtschaftsinformatik. GITO, 2012.

5. Kharbili, M. E. and Keil, T. Bringing Agility to Business Process Management: Rules Deployment in an SOA, Emerging Web Services Technology - Volume III, 2010.

6. Krafzig, D., Banke, K. and Slama, D. Enterprise SOA - Service-oriented Architecture Best Practices. Prentice Hall International, 2005.

7. Levina, O. and Stantchev, V. Realizing Event-Driven SOA. Int. Conf. on Internet and Web Applications and Services. Venice/Mestre: IEEE, 2009.

8. Luckham, D. SOA, EDA, BPM and CEP are all Complementary - Part 1, 2012, <http://complexevents.com/wp-content/uploads/2007/05/SOA_EDA_Part_1.pdf>, (10.07.2015).

9. McGonigal, J. Reality is Broken: Why Games Make Us Better and How They Can Change The World. The Penguin Press, New York, 2011.

10. Reeves, B. and Read, J. Total Engagement: Using Games and Virtual Worlds to Change the Way People Work and Businesses Compete. Harvard Business Press, Boston, MA, 2009.

11. Thom, J., Millen, D. and DiMicco, J. Removing Gamification from an Enterprise. Proceedings CSCW, 2012.

12. Xu, Y. Literature Review on Web Application Gamification and Analytics. <https://csdl-techreports.googlecode.com/svn/trunk/techreports/2011/ 11-05/11-05.pdf>, (13.10.2015).

13. Zichermann, G and Cunningham, C. Gamification by Design: Implementing Game Mechanics in Web and Mobile Apps. O'Reilly Media, 2011.

Невян Нейков, се занимава със сигурност на софтуер и компютърни технологии. Завършва Икономически университет - Варна специалност Информатика.

Следва практически стажове по уеб дизайн и програмиране в Лийдс - Великобритания и Берлин - Германия. Работил е като Интранет уеб програмист и дизайнер на официални документи в Агенция по права на човека във Виена - Австрия и като програмист в Европейски парламент Брюксел - Белгия. Интересува се от тематика и е разработил собствен уеб портал за осъществяване на електронно IT обучение. В последствие е участвал в разработването на система за е-обучение към водеща компания за уеб обучение в Антверпен - Белгия. Продължава обучението си, като докторант към Икономически университет - Варна, специалност Информатика, където изследва темата за сигурност на облачните технологии.

АРХИТЕКТУРА НА ОБЛАЧНИ ОПЕРАЦИОННИ СИСТЕМИ

докторант Невян Нейков, Икономически университет – Варна
e-mail: nevyan_neykov@ue-varna.bg

1. Въведение

Облачните изчисления подпомагат създаването на големи, еластични системи за обработка на данни. В случаите, когато изискванията на потребителите са ниски, системите автоматично намаляват консумираните от тях ресурси, като при нужда ги разрастват за да отговорят на необходимите възникнали потребности. Обикновено практическото използване на облачни ресурси изисква създаването на специална мрежа (Grid) от виртуални системи. Изграждането на подобни системи се извършва, чрез използване на процеса виртуализация, с помощта на технологията хипервайзор, позволяваща изграждането и стартирането на множество виртуални системи на един и същи физически сървър. Облачните изчислителни системи консумират ресурси само при необходимост, които в последствие биват освободени. Това ги прави удобни за употреба на разнороден хардуер, като минимизирайки загуби произтичащи от излишък на ресурси. Все повече доставчици на облачни услуги предлагат предварително тестване на продуктите, което дава на потребителите възможност за тяхното опознаване.

2. Същност на облачните изчисления

Виртуализация – софтуерна технология, която чрез споделяне на ресурси, позволява създаването на виртуални версии на: хардуерни устройства и софтуер. Това прави възможно стартирането на множество операционни системи и приложения на един сървър по едно и също време.

Виртуална машина – софтуерна абстракция на хардуер, управлявана от операционна система, използваща в основата си хипервайзори. Различават се два

вида виртуални машини – гост и домакин. Домакина представлява първоначално инсталираната(първична) операционна система, госта е виртуалната машина, която е инсталирана върху първичната операционна система.

Хипервайзор – форма на виртуализация, емулираща ресурси на хардуерно ниво, даваща възможност да се създават и стартират виртуални машини, на които се инсталират множество операционни системи. Тези операционни системи споделят виртуализирани хардуерни ресурси. Примери: Xen, VirtualBox, KvmM, Wmware, Hyper-V.

Контейнер – форма на виртуализация, емулираща ресурси на ниво на операционна система(ядро). По този начин се използват по-малко ресурси, което от своя страна позволява опаковането и работата на много приложения на един сървър. Контейнерите не правят абстракция на целия хардуер, като използват само нужните за приложението ресурси, останалите остават налични за другите контейнери. Това води до повишаване на тяхната скорост на стартиране и работа. Примери: Docker, LXD

Основни разлики между понятията клъстер, грид и облак: Клъстерът представлява група от еднородни с подобни хардуерни конфигурации компютри свързани посредством локална мрежа. При грида и облака тази мрежа може да бъде разнородна, там машините са разпространени в по-широк мащаб и са географски разпръснати. Облачните изчисления имат централизиран модел на разпределение на ресурсите, докато при грида модела е децентрализиран измежду множество административни домейни. При облачните изчисления компютрите се притежават от една организация, докато при в грида те са собственост на много лица.

Облачните изчисления включват в себе си, както приложенията, предоставяни като услуги, така и хардуерните и софтуерните ресурси намиращи се в центровете за данни. Обикновено всички тези услуги се предлагат публично и се наричат Софтуер като услуга (SaaS). В тях се включват облачно-базирани операционни системи, които представляват вид операционна система,

конструирана за работа в облачни и виртуализирани среди. Тя управлява действието и изпълнението и процеси на виртуалните машини, виртуални сървъри и виртуална инфраструктура, както и хардуерни и софтуерни ресурси. Примери за подобни облачно-базирани ОС са: Microsoft Windows Azure и Google Chrome OS. Техните ресурси са достъпни за потребителите. Облачните операционни системи се наричат още Web OS, защото с помощта на браузър могат да бъдат управлявани, да се стартират апликации или дори цялостни операционни системи. Данните им се съхраняват в Интернет и са налични по всяко време и място. Обикновено публичните облачно базирани операционни системи предоставят параметри за ползване като: 1GБ до 30GB свободно място и възможност за връзка с други доставчици на услуги за съхранение като: DropBox, Google Drive, SkyDrive, 4Shared и др. От приложенията са налични инструменти за редакция на текст, снимки, документи, таблици и съобщения, календар, бележки, презентации, файлов мениджър, ftp клиент, аудио плеър, емейл. Допълнително се предлага синхронизация на файлове с водещи операционни системи. Облачно базираните ОС са полезни за хора, които са в постоянно движение и работят с различни компютърни и мобилни устройства. Самата web операционна система не изисква много ресурси за стартирането си, като това я прави приложима за използване от нетбуци, таблети и по-стари компютри, поради извършването на операции в облака и връщането на техният резултат. Примери за публични облачно базирани системи са:

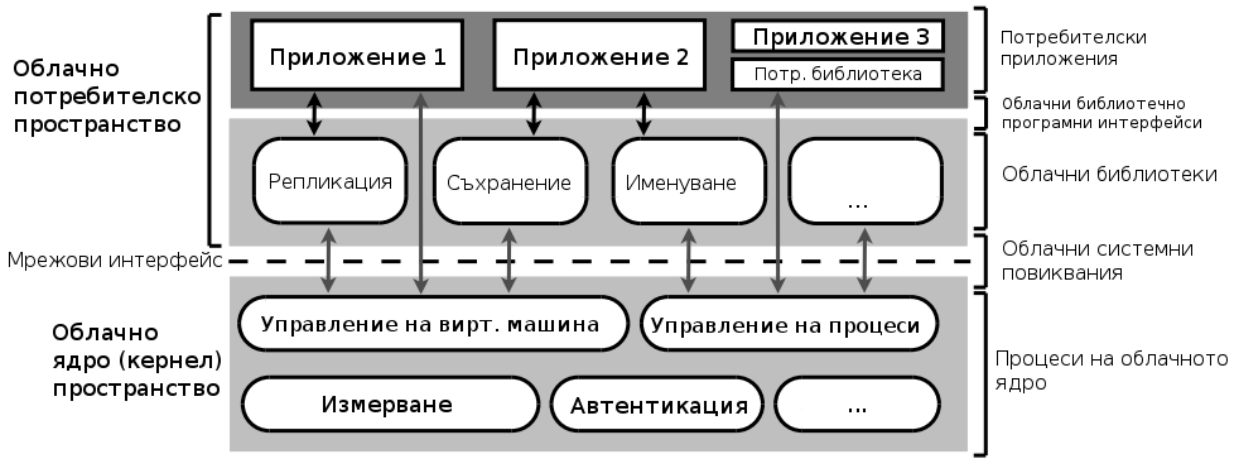
MS Office online: Word, Excel, PowerPoint, OneNote запис на данни в OneDrive

Google: Gmail, Calendar, Docs, Sheets, Slides – запис на данни в Google Drive

Apple iCloud: mail, contacts, calendar: Pages, Numbers., Keynopte – запис на данни в iCloud

хибридни – DropBox – локално и облачно съхранение на данни

Ще разгледаме подробно архитектурата на една облачно базирана операционна система и нейното функциониране – фиг. 1. При облачните операционни системи дадена компютърна машина се обозначава с понятието възел и се идентифицира с произволен идентификатор с цел минимизиране на риска й от колизии с други машини.



Фиг. 1. *Логическа архитектура на облачна операционна система*

Облачен обект – съвкупност от локални процеси принадлежащи на операционната система.

Облачен процес – набор от облачни обекти, от които в последствие се изгражда едно приложение. Трябва да се има в предвид, че в една облачна операционна система изчислителните облачни процеси са винаги активни и работещи.

В рамките на пространството на ядрото се управляват процеси като: физическа алокация, контрол на достъп и измерване на ресурси. Всички останали процеси се обхващат от потребителското пространство.

Облачни процеси, изпълнени директно от потребители се наричат потребителски приложения. Облачните библиотеки от своя страна представляват

облачни процеси, извиквани от потребителски приложения или от други библиотеки.

Приложенията комуникират с библиотеките и ядрото на облачните процеси чрез мрежа, посредством стандартни интерфейси наречени облачни системни повиквания. Обектите намиращи се в облачното потребителско пространство имат достъп до управлението на тези облачни системни повиквания, като прихващат сигнали от облачната операционна система и са достъпни през мрежови интерфейс за извършване на процеси по управление.

Ядрото на облачните процеси и по-конкретно управлението на процеси и виртуални машини управляват връзките между имената на обектите, техните мрежови адреси и портове, като резултатната информация става достъпна из целия облак посредством библиотека на имената. Тя проследява връзките между процесите на потребителските облачни приложения и съставните им обекти.

Автентикационното ядро на облачните процеси раздава и проверява правата за достъп необходими на операциите по управление.

Измерване на ресурси: OS поддържа броя на наличните облачни ресурси, чрез извършване на локални измервания на всеки един облачен възел. Вземат се в предвид променливи като: латентност, ширина на мрежова лента, загуба на пакети, които са ценни ресурси на знание и се използват директно от приложенията.

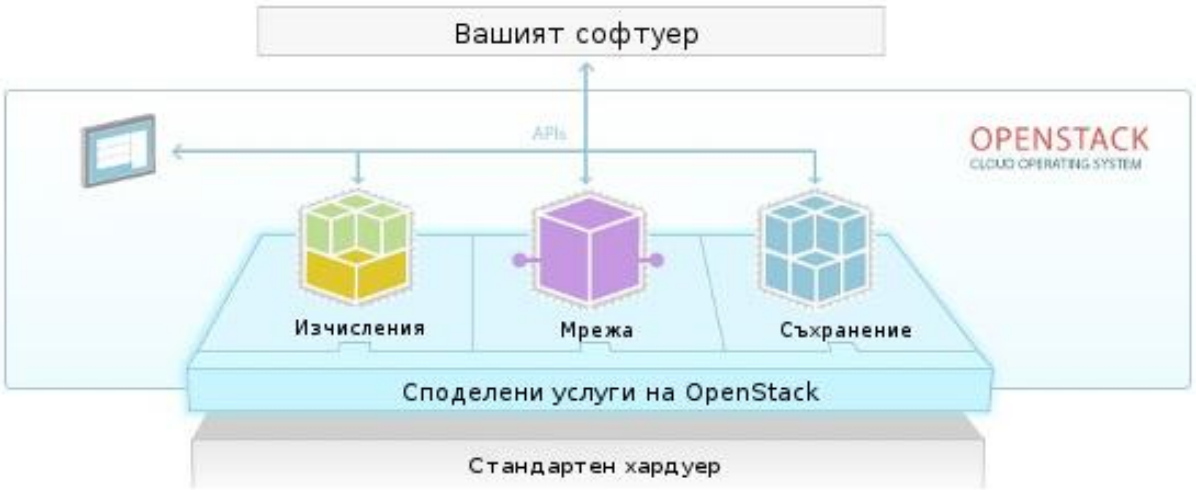
Разпределено управление на процеси и приложения: Облачната операционна система създава и управлява всеки един съществуващ обект в облачните възли. За тази цел, като обща практика се използват виртуални машини, доставящи абстракция, която гъвкаво отделя логическите изчислителни ресурси от изграждащите ги физически облачни възли. Виртуализацията предоставя характеристики, нужни за облачната среда, като: поддръжка на множество операционни платформи работещи върху един и същ компютър и до известна степен изолация между процеси стартирани от различни виртуални машини, намиращи се на един и същи хардуер. Оптимизационните изисквания,

като: изчислителна еластичност и балансиране на товар внасят нужда от динамично алокиране на ресурси. Например прехвърлянето на един работещ процес между два облачни възела, се осъществява или на ниво на облачни процеси – мигриране на единични процеси между възли или на ниво виртуални машини – съхранение и възстановяване на цялостното състояние на виртуалната машина върху друг възел. Операционната система 'гост' се намира на слой над хипервайзора, който посредством файлово-системен интерфейс достъпва функциите предоставяни от операционната система 'домакин'.

Облачната операционна система, допълнително предоставя на потребителите интерфейс за управление на процеси, посредством абстракция с цел обединение на всички различни изчислителни ресурси в едно. Облачните библиотеки предоставят стандартни API интерфейсни характеристики като: достъп до обекти из целия облак, именуване на процеси посредством DNS, разпределена и надеждна функционалност по съхранение, автоматизирано изграждане на облачни приложения, хоризонтално скалиране и управление на жизнен цикъл, висока достъпност, поддръжка при грешки с възможност за репликиране на изпълнение на процеси. Облачните библиотеки дават на разработчиците контрол върху нужното им ниво на репликация, консистентност и достъп, както и начини за справяне в случай на провал, когато изискванията на приложението не са задоволени. По този начин разработчика на приложения концентрира вниманието си върху определени черти на приложението, знаейки, че системата ще направи най-доброто за да осигури заявените изисквания. Ползите са двустранни: от една страна разработчика, не е задължен да конструира собствени библиотеки специално за приложението, а само да настройва ниво на поддръжка от което има нужда посредством API интерфейс на облачната библиотека. По този начин се намалява сложността и настъпването на грешки при създаването на приложения. От друга страна може повторно да се използват компоненти на облачни приложения, като се адаптират за задоволяване на различни спецификации, обновявайки много малка част от параметрите на самата облачна библиотека.

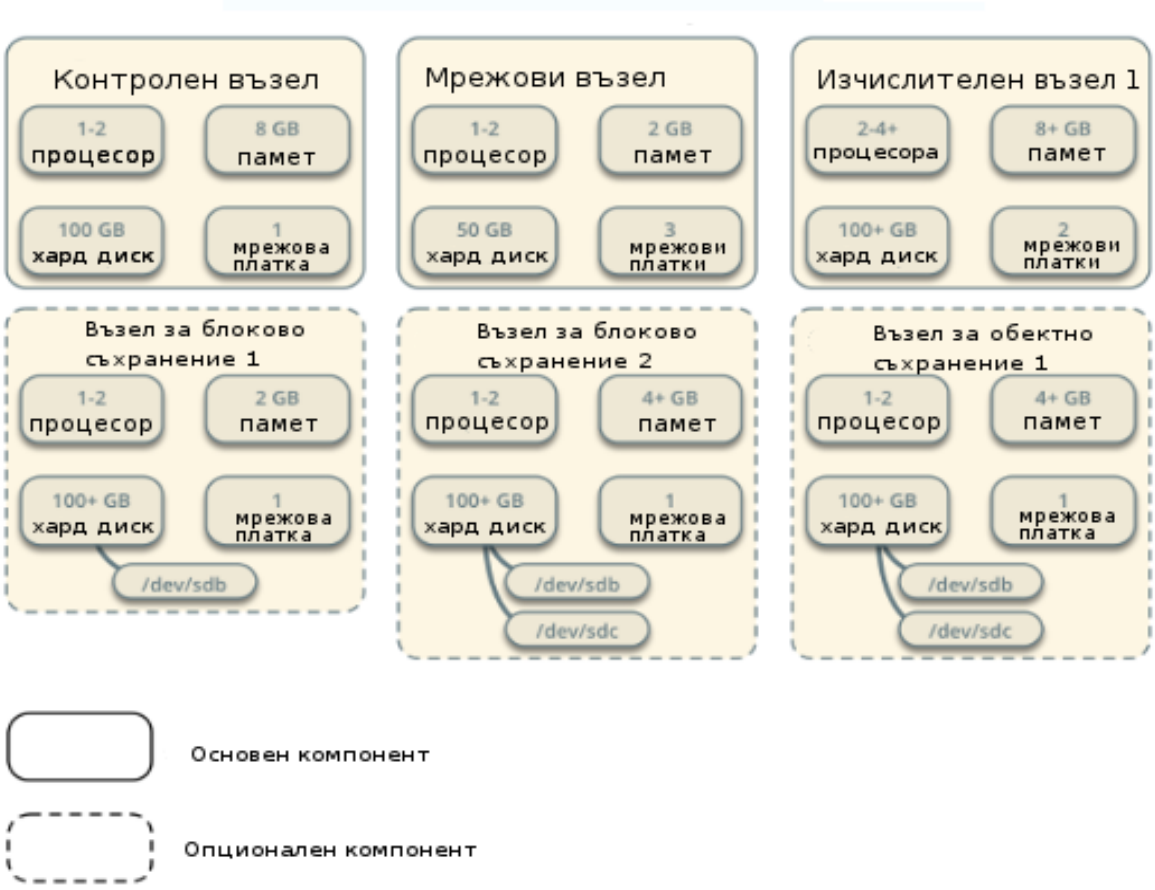
3. OpenStack – инфраструктурен софтуер за облачни изчисления, използващ Apache лиценз

OpenStack е основан от NASA и RackSpace, представлява софтуер управляващ облачни изчисления, който се използва от големи фирми, университети и институции. Факт е, че 55% от OpenStack продукционните облаци по света работят под Линукс Убунту. OpenStack не доставя само услуги, обслужвайки системи или центрове за данни, а представлява софтуер с отворен код за създаване на публични, частни и хибридни облаци. По този начин продуктът позволява лесно преместване на данни и услуги от една към друга фирма, както и разпределяне и използване на общи услуги конкурентно от различни фирми. Концептуалната архитектура на софтуерния продукт е представена на фиг. 2.



***Фиг. 2** Концептуална архитектура на OpenStack*

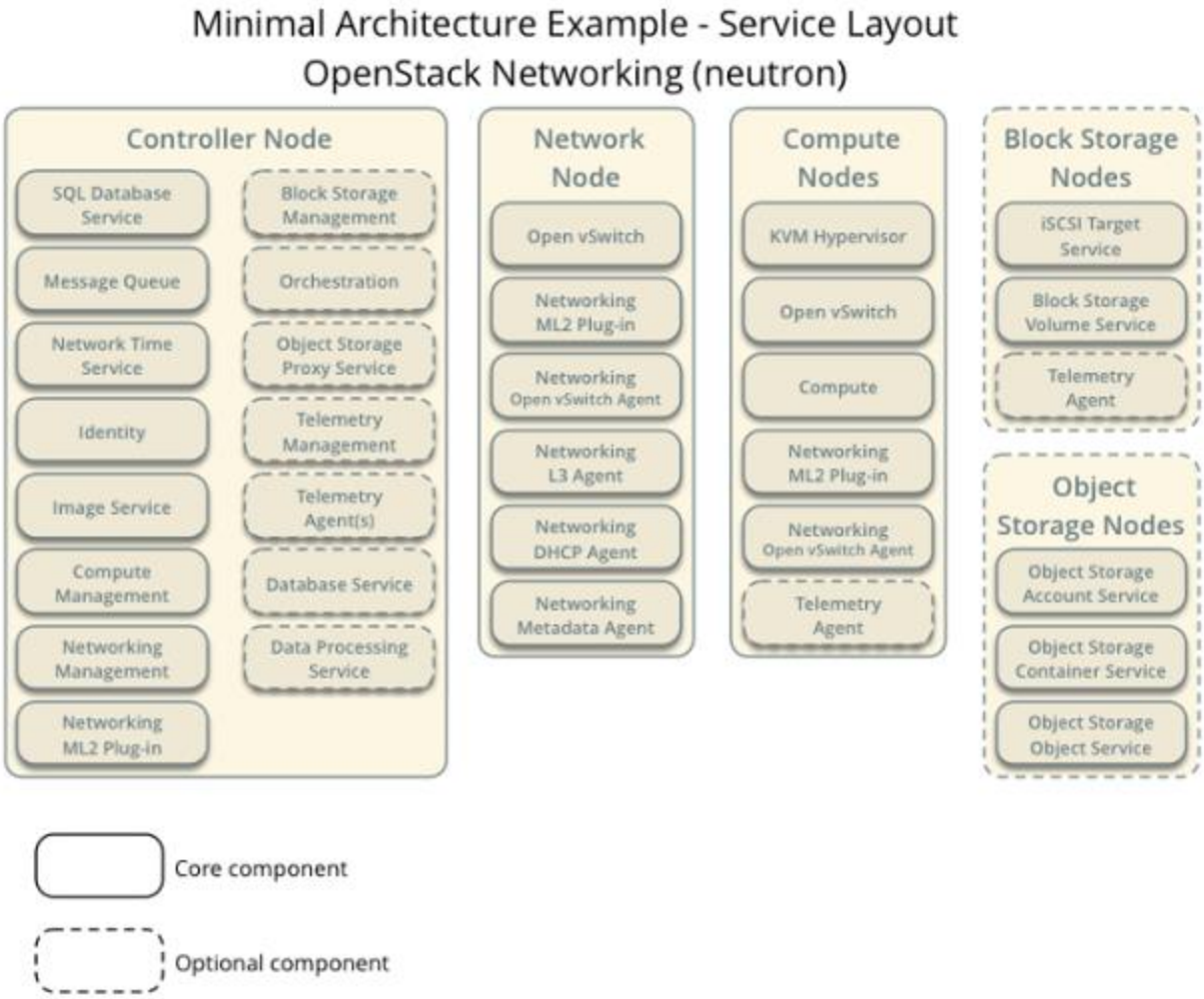
Следва минималната хардуерна конфигурация, която се изисква за стартиране на продукта, включваща по един компютър за контролен, мрежови и



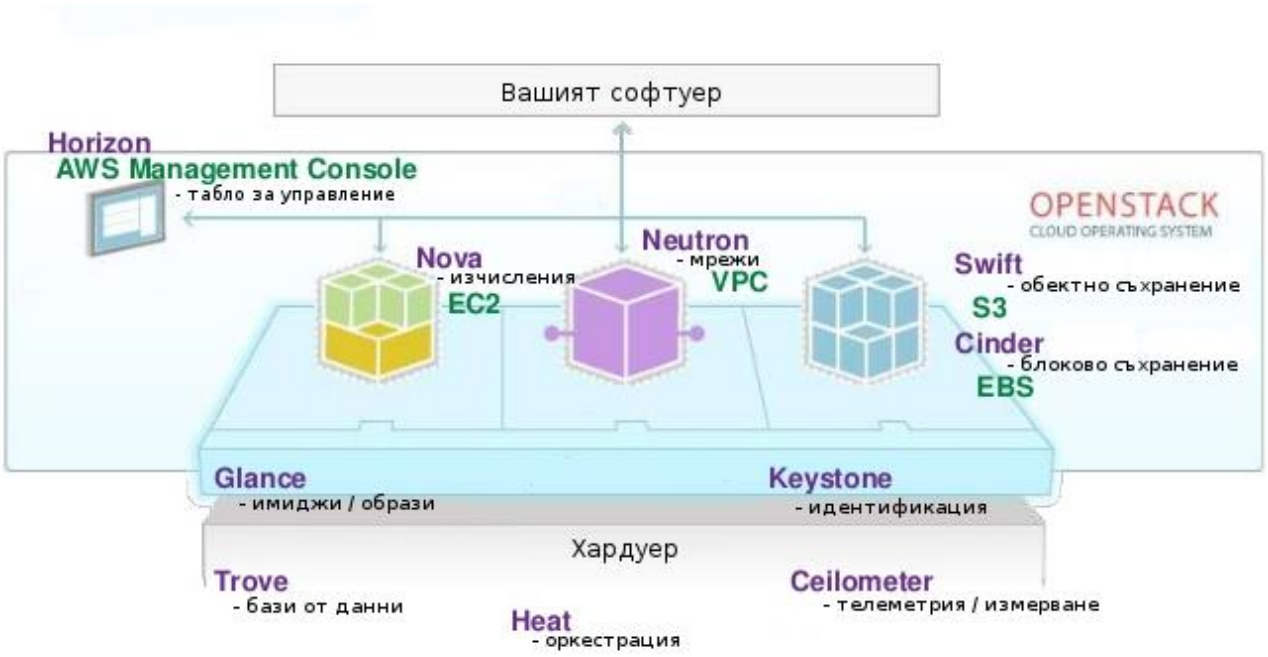
изчислителен възел - фиг. 3.

Фиг. 3 *Минимални хардуерни изисквания на архитектурата на Neutron*

Проектът OpenStack доставя облачни решения от тип Инфраструктура като услуга (IaaS) чрез различни услуги. Всяка една услуга се предлага за достъп чрез включен софтуерен програмен интерфейс (API), който улеснява интеграцията им. OpenStack е конфигуруем, за да може да посрещне нуждите от различни компютърни, мрежови и дискови опции. Стартирането на виртуална машина или т.нар. инстанция включва в себе си взаимодействия измежду множество услуги. Следва описание на основните услуги предлагани от неговата софтуерна платформа представени на фиг. 4 и фиг.5.



Фиг. 4. *Компоненти и услуги на OpenStack базиран на Neutron*



Фиг. 5. *OpenStack услуги*

Навигационна среда: Horizon: Доставя уеб базиран портал за самообслужване, който взаимодейства с услугите предлагани от OpenStack, като стартиране на инстанция, назначаване на IP адреси и конфигуриране на контрол на достъп.

Изчисления: Nova: Управлява цикъла на живот на отделните изчислителни инстанции (компютри). Отговаря за създаването, разпределението и разрушаването на виртуални машини според предварително определени изисквания. *Компютърният възел* стартира хипервайзорната (KVM) част от Compute, която от своя страна стартира виртуални машини или инстанции. Тук се стартира също и мрежовия плъгин, който не само свързва мрежи, но и доставя услуги, като защитни стени. Може да се стартира повече от един Compute възел. Допълнително, Compute възела включва агент по Телеметрия, който събира метрики, както и трети мрежови интерфейс към отделна мрежа, за да подобри производителността на услугите по съхранение на информация.

Мрежова свързаност: Neutron: Позволява свързаност, като услуга към други услуги на OpenStack. Доставя API за потребителите за определяне на мрежи и прикачвания към тях. Има pluggable архитектура, която поддържа множество мрежови технологии. *Мрежовия възел* включва в себе си мрежови плъгин и различни агенти, които дават възможност за превключване, рутиране, преадресиране, както и услуги по динамично раздаване на IP адреси. Този възел също се занимава с външната свързаност между отделни виртуални инстанции.

Съхранение на информация

Запазване на обекти: Swift: Запазва и извлича произволно неструктурирани обекти посредством REST HTTP API. Доставя висока толерантност при репликация на грешки и мащабиране на архитектурата. Опционалния Object Storage възел включва дискове за съхранение на акаунти, контейнери и обекти. Минималната архитектура изисква съществуването на повече от два възела.

Блоково съхранение: Cinder: Доставя непрекъснато блоково съхранение към стартираните инстанции. Неговият драйвер ускорява създаването и управлението на блокови устройства за съхранение на информация. Опционалния Block Storage възел включва дискове, които се използват от включените виртуални машини. Може да се стартира повече от един от този възел.

Споделени услуги

Идентифицираща услуга: Keystone: Доставя автентикация и оторизация към други услуги на OpenStack, както и каталог от крайни точки съдържащ всички предлагани услуги.

Имидж услуга: Glance: Запазва и извлича дисковите имиджи на виртуалните машини. Използва се от Compute при стартирането на услуги.

Телеметрия: Celiometar: Прави мониторинг и измерва облака за производителност, мащабиране, заплащане и статистически цели.

Услуги от високо ниво

Оркестрация: Heat: Оркестрира множество съставни облачни апликации, като използва шаблонни формати захранвайки REST API на OpenStack

Бази от данни: Trove: Доставя мащабиращи и надеждни услуги предаващи функционалност по бази от данни.

Концепция за идентификация

Идентификацията представя пред OpenStack даден човек, система или

Фиг. 6. *Процес на идентификация в OpenStack – Keystone*

услуга, използващи облачните услуги, като валидира входящите заявки на потребителя, който твърди че прави запитването. Услугата по идентификация проследява потребителите и техните позволения, като доставя каталог съдържащ наличните им услуги и техните крайни API точки за достъп. Всяка една услуга е нужно да се инсталира посредством Identity service, която дава възможност да се проследят всички инсталирани услуги и тяхното месторазположение в мрежата. Към потребителите се добавят тоукъни, за да осъществят достъп до ресурси. Допълнително потребителите, могат директно да се присъединяват, към определен ползвател за придобиване на поведение (пример: достъп до ресурси), присъщо за този тип ползвател. Следва описание на използваните понятия предлагани от платформата за идентификация:

Права за достъп

Това да данни, които потвърждават идентичността на даден потребител. Например: потребителско име, API ключ и парола, или автентикиращ токън доставен от Услугата по Идентичност.

Автентикация

Процес при който се потвърждава идентичността на даден потребител. Услугата Identity потвърждава входящите заявки, валидирайки определен набор от credentials подадени от потребителя, които биват: потребителско име и парола или потребителско име и API ключ. След тяхната валидация, OpenStack Identity издава автентикационен токън, който потребителя представя в последващи заявки – фиг. 6 .

Токън

Низ, състоящ се от букви и цифри, с който се осъществява достъп до API и ресурсите на OpenStack. Той може да бъде премахнат по всяко едно време и е валиден за определен краен момент от време.

Ползвател

Контейнер, който се използва за да групира или изолира ресурси или обекти по идентичност. В зависимост от оператора на услугата даден ползвател може да бъде прикачен към клиент, акаунт организация или проект.

Услуга

Услугите в OpenStack са: Compute (nova), Object Storage (swift), Image Service (glance). Те доставят една или повече крайни точки, през които потребителите имат достъп до ресурси и могат да извършват операции.

Крайна точка

Достъпен през мрежата адрес, от където може да извика услуга, обикновено представлява URL адрес.

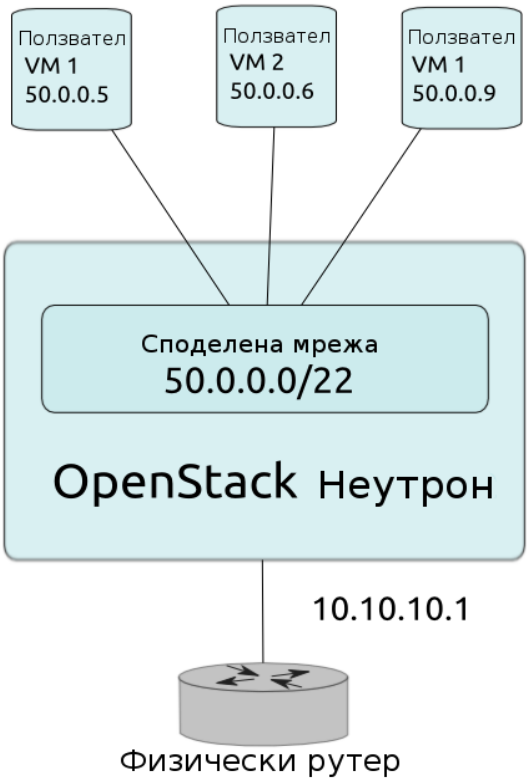
Роля

Персоналност с дефиниран набор от правила или привилегии, служеща за изпълнението на определен брой операции. Услугата Identity, издава токън за даден потребител, който включва в себе си списък с роли. Услугите, които се извикват от даден потребител определят: как да интерпретират неговия набор от роли и по-точно за кои операции и ресурси всяка роля му дава достъп.

Мрежови концепции

Мрежовата свързаност на OpenStack Neutron управлява всички мрежови аспекти от виртуалната и нивата на достъп на физическата мрежова инфраструктури – фиг. 7. Тази свързаност дава възможност на ползвателите да създават сложни виртуални мрежови топологии, включващи услуги, като: защитни стени, балансиране на натоварване и виртуални частни мрежи – фиг. 8.

Фиг. 7. *Мрежова архитектура управлявана от Neutron*



Фиг. 8. *Примерна мрежова архитектура - Neutron*

Мрежовата свързаност предоставя абстракции на мрежи, под-мрежи, рутери и обекти. Всяка една подобна абстракция притежава приличаща на физическата функционалност, където от мрежи се създават подмрежи, а чрез рутери се пренасочва трафик между различни подмрежи. Всеки един рутер има gateway, който е свързан към дадена основна мрежа, както и множество интерфейси свързани към подмрежи. По този начин подмрежите имат възможност за достъп до машини от други подмрежи, към които рутерът е свързан.

Всяка една инсталирана мрежа съдържа в себе си поне една външна мрежа. Важно е да се знае, че тя не е виртуална, а представлява гледка към реална физическа, външна мрежа, достъпна извън инсталацията на OpenStack. Поради това, че нейните IP адреси са достъпни от всеки, който се намира извън

инсталацията на OpenStack, при нея протокола за раздаване на динамични адреси DHCP е забранен.

Освен това, всяка една мрежова свързаност съдържа една или повече вътрешни мрежи. Към тези софтуерно-дефинирани мрежи директно се свързват виртуални машини (ВМ). За достъп от външна мрежа до виртуалните машини и обратно са нужни рутери между мрежите. Виртуалните машини от една мрежа са директно достъпни единствено от ВМ на всяка вътрешна мрежа или от подмрежи свързани през интерфейси на рутери. Всеки един рутер притежава gateway, който е свързан към мрежата и множество интерфейси, свързващи подмрежите. Подобно на физически рутер, ВМ от една подмрежа имат достъп до ВМ на друга подмрежа, свързана към същия рутер. Чрез gateway-а интерфейса на рутера ВМ имат достъп до външни мрежи.

Всяка една връзка към дадена подмрежа се нарича 'порт'. Допълнително, може да се разпределят IP адреси от портове на външни към вътрешни мрежи, както и да се асоциират външни мрежови IP адреси с определени портове към виртуални машини. По този начин компютри от външната мрежа могат да имат достъп до виртуалната машина.

Групи по сигурност

Те дават право на администраторите да дефинират правила на защитни стени и да ги запазват в групи. Една виртуална машина може да принадлежи към една или повече групи за сигурност. Частта Neutron Networking прилага правила в тези групи по сигурност, които блокират или отварят портове или тип трафик специфичен за тази виртуална машина.

4. Заключение

Облачните услуги навлизат в живота на всеки Интернет потребител. До този момент те представляват предизвикателство не само към ползвателите, но и към хората разработващи подобни системи, които желаят с тяхна помощ плавно да заменят стандартните съществуващи архитектури. Описаната примерна архитектура на облачно-изчислителна инфраструктура е изградена на база на софтуерната платформа OpenStack. Възприетия в нея архитектурен модел позволява функционален достъп и обработка на данни при поискване, използвайки ресурси в зависимост от възникнали потребности. Това от своя страна води до оптимизация на цялостната производителност и употреба на дадена услуга, наред с намаляване на съпътстващите я финансови разходи.

Използвана литература

1. Hemke M., Ubuntu Unleashed 2015 Edition: Covering 14.10 and 15.04, SAMS, Pearson Education
2. OpenStack Installation Guide for Ubuntu 14.04, OpenStack Foundation, docs.openstack.org/juno/install-guide/install/apt/openstack-install-guide-apt-juno.pdf, 7-08-2015
3. Doyle J., Shorten S., O'Mahony D., Stratus: Load Balancing the Cloud for Carbon Emissions Control, IEEE: Transactions on Cloud Computing, http://www.tara.tcd.ie/bitstream/handle/2262/67834/Stratus_v2.pdf
4. Poelker C., http://www.computerworld.com/article/2473557/data-center/will-cloud-computing-kill-the-storage-area-network-.html, 11-12-2012
5. Ubuntu in the cloud, http://www.ubuntu.com/cloud
6. Cloud computing for business: What is cloud, http://www.opengroup.org/cloud/cloud/cloud_for_business/what.htm
7. Matthews R., How Environmentally Sustainable is Cloud Computing and Storage?, 09-12-2013,

http://globalwarmingisreal.com/2013/09/12/sustainable-cloud-computing/

8. Pillai S., Difference between Hypervisor Virtualization and Container Virtualization http://www.slashroot.in/difference-between-hypervisor-virtualization-and-container-virtualization, 10-21-2014
9. Pianese, Fabio, et al. "Toward a cloud operating system." Network Operations and Management Symposium Workshops (NOMS Wksps), IEEE, 2010, http://www.moritzsteiner.de/papers/CloudOS.pdf

Тодор Константинов Дянков

1995 г. - езикова гимназия – В. Търново

1999 г. – бакалавър по „Международен туризъм” – ИУ – Варна

2000 г. - бакалавър по „Международни икономически отношения” – ИУ – Варна

2001 г. – магистър по „Туризъм” - ИУ – Варна

2012 г. – магистър по „Право” - ИУ – Варна

2006 г. – доктор по шифър 05.02.18 „Икономика и управление на туризма”

От 2008 г. до сега – главен асистент към катедра „Икономика и организация на туризма” – ИУ – Варна

Основни научни интереси в областите: **маркетинг на взаимовръзки с клиенти, маркетингови комуникации, управление на рекламации от клиенти.**

ОСНОВНИ КОНСТРУКТИВНИ ЕЛЕМЕНТИ ПРИ УПРАВЛЕНИЕ НА ВЗАИМООТНОШЕНИЯТА С КЛИЕНТИ В ТУРИСТИЧЕСКИЯ БИЗНЕС

Гл.ас. д-р Тодор Дянков
Икономически университет – Варна
E-mail: todordyankov@abv.bg

Резюме. Настоящата публикация има за цел откроването и преосмислянето на съвременните конструкти за успешното изграждане и развиване на устойчиви взаимоотношения между туристическите фирми и техните клиенти. Спецификите на взаимоотношенията с клиенти в туризма са изведени на фона на сравнително добре познатите транзакционни взаимодействия в контекста на съвременните бизнес управленски концепции, модели и практики.

Ключови думи. взаимоотношения, транзакционни взаимодействия, доверие, ангажираност, позволение, удовлетвореност, лоялност;

В съвременния туристически бизнес все по-невралгични стават проблемите относно успешното изграждане и развиване на фирмените взаимоотношения с крайни клиенти. Туристът на 21 – ви век се отличава със своята силно информираност, гъвкавост, мобилност, самостоятелност, непредсказуемост както в своя избор на туристическа дестинация, така и в потреблението на конкретен туристически продукт. Златните години на масовите организирани туристически пътувания с обща цена, като че ли изглеждат безвъзвратно отминали с оглед настъпилите трансформационни промени и преформатиране на туристическите пазари като резултат от развитието на интернет средата и информационните технологии. Туристическото търсене е безпрецедентно фрагментирано в нишови и персонализирани продукти, както никога досега. Междуфирменото съперничество за клиенти се преобразува от предлагането на силно стандартизирани туристически продукти към незабравими изживявания и дълготрайни взаимоотношения между доставчика на услугите и непрекъснато изискващия потребител. В този ред на мисли, целта на настоящата публикация е да се открояят и критично преразгледат основните конструкти, които е необходимо да се вземат под внимание от фирмите за ефективно управление на взаимоотношенията си със своите клиенти.

В своята същност, изграждането на всяко взаимоотношение между фирма и клиент носи в себе си определени елементи от транзакционен характер Транзакционността неминуемо стои в икономическата основа на изграждане на

взаимоотношенията. От друга страна не всяко транзакционно взаимодействие може да се превърне във взаимоотношение, както и не всяко изграждане на взаимоотношение е необходимо да стъпи върху предходен транзакционен обмен. Последната позиция се защитава от автори, които виждат в приложението на директния маркетинг задължително възможността за подобна трансформация – а именно от транзакции към взаимоотношения. Дори е налице отиване по-далече в тази посока, като директният маркетинг се причислява към маркетинга на взаимоотношения с клиенти (*Желев, С и Г. Младенова, 2012*). Много често практически това е не само неизгодно за едната или двете страни във взаимодействието, но е дори и нелогично от бизнес гледна точка. От друга страна е вече станало класическо асболютизирането на разликите между транзакции и взаимоотношения с клиенти, което не бива да омаловажава единият или да възвеличава другия вид социално-икономическо взаимодействие. Ето защо в основата на изграждане и развитие на взаимоотношения с клиенти в туризма могат да се систематизират някои основни концепции и конструктивни елементи.

Дефинициите за маркетинг на взаимоотношения с клиенти са изключително много и разнородни по съдържание и същевременно взаимно допълващи се. В качеството на отправна база може да се използва едно широко определение, което гласи, че: „Маркетингът на взаимоотношенията обхваща всички маркетингови дейности насочени към установяването, развиването и поддържането на успешни връзки с клиентите" (*Conway, T, 2000*). Някои автори добавят, че в основата на тази философия стои задържането на клиента, като взаимовръзките трябва печелившо да удовлетворяват целите на всички участници, а дори когато е необходимо, тези връзки да могат да бъдат и прекратявани *(Bowie, D. and F. Buttle, 2004)*. Развиването на дългосрочни връзки с клиента се извежда в сърцевината на концепцията за маркетинга на взаимоотношенията в туризма (*Cavagnaro, E and F. Bosker, 2007)*. Котлър добавя създаването на дълготрайна стойност за клиента, като според него измерител за успех представлява дългосрочната клиентска удовлетвореност *(Kotler, Ph. and others, 1996).*

Маркетингът на взаимоотношенията с крайни клиенти има благоприятна почва за развитие изключително в сферата на услугите и по-точно в сливането (интегрирането) на производствения процес на продукта със самия процес на обслужване *(Crandall, R. and W. Crandal, 2008).* Дори при стандартизираното обслужване, маркетинговата дейност продължава в процеса „производство – потребление". Неслучайно тези възможности за развитие са присъщи за много дейности от туристическата индустрия, независимо, че при нея липсва една от важните възможности за развиване на взаимоотношения с клиенти – а именно възможността за допълнително обслужване след края на процеса на потребление

на продукта. Ето защо е необходимо маркетингът на взаимоотношенията да бъде откроен, като алтернативна парадигма на фона на класическия маркетинг, без да се абсолютизира мястото на първия, който получава и не малко критика в теорията и практиката *(Palmer, A., 1996).*

Процесът на изграждане и развиване на взаимоотношения с клиенти в туризма включва маркетингови дейности *преди, по време на* и *след потреблението* на туристическия продукт. В тази връзка, центърът на самото взаимодействие се измества от реализирането на предварително заложен за производство продукт според неговата фаза на развитие към реализирането на взаимоотношения с клиенти в техния специфичен стадий на развитие. С други думи, фирменото намерение за продажба на конкретен продукт към който и да е клиент от сегмента се трансформира в намерение за персонализирано удовлетворяване нуждите на въпросния клиент с адаптирани към него продуктови решения.

Доверието е централен конструктивен елемент при изграждане и развиване на взаимоотношения с клиенти. Най-обща дефиниция за него дава Мурман, а именно като ясно изразено желание да се разчита на партньор в когото вярваме *(Moorman, R and others, 1992)*. Доверието се изгражда върху стабилността (коректност, точност, искреност) на партньора, като и върху неговите експертни качества в полза на поддържане и развиване на взаимоотношенията. От друга страна, този, който се доверява, е потенциално уязвим, било то и в състояние на несигурност при поддържане на взаимоотношенията с другия партньор.

Друг основен конструктивен елемент в развиване на взаимоотношенията между фирма и клиент е така нареченото продължаване на *ангажираността* във връзката на поне една от страните (предимно фирмата). Под ангажираност се има предвид „намерението да се продължава поетия курс на действие или желанието за поддържане на взаимоотношения" *(Altan, M. and K. Eringa, 2007).* Тук фирмената ангажираност излиза отвъд времевите рамки на договорното (юридическо) обвързване между страните. В тази насока фирмата е необходимо активно да поддържа своята ангажираност към клиента, стига последният да даде своето *позволение*. Относително на обратния полюс стои въпросът с фирмената ангажираност при реализиране на транзакционни взаимодействия между страните. Обикновено ангажираността се затваря в рамките на предмета и обсега на сделката (договора) относно всяка конкретна транзакция. В тази връзка се счита, че ангажираността се подновява (револвира) с възникването на следващата по ред транзакция между същите страни. Оттук следва, че и фирмената ангажираност не е задължително условие за поддържане на транзакционни отношения между страните

Позволението от страна на клиента е ключово условие за изграждане на устойчивост във взаимоотношенията му с фирмата. Най-висока степен на доверие клиентът оказва към фирмата при т. нар. *интравенозно позволение* в сферата на услугите *(Годин, С., 2010)*. Интравенозното позволение е рядко срещано по своята същност вид пряко упълномощаване на фирмата от клиента в определена област на получавани услуги (например почистващи, транспортни, логистични и др.). Тук клиентът не само дава пълното си позволение фирмата да го обслужва и задължава към трети страни, но и последната свободно да извършва разплащания от неговия бюджет. В туризма, клиентското упълномощаване на фирми за интравенозна дейност като цяло е нетипична практика. Трудностите за нейното изпълнение произлизат от причини като: нерегулярното потребление на туристически услуги от страна на клиента (дори да става въпрос за стандартизирани продукти); все по-лесните и удобни комуникационни начини за достигане и договаряне на услугите от клиента; степента на интензивна конкуренция между доставчиците на туристическите услуги; необходимостта фирмата повторно да иска позволение от клиента за осъществяване на сделката или в краен случай за нейното потвърждаване за да може да влезе в юридическа сила. Последното основание е особено характерен резултат от динамично променящите се пазарни условия. В този ред на мисли, може да се твърди че в туристическата дейност клиентското позволение рядко има упълномощителен за фирмата характер, а по-скоро последната действа като класически медиатор (посредник) със специализирани консултантски функции – например пътническите агенции, които продават единични услуги на крайни клиенти предоставяни от техни контрагенти. Ето защо позволението от клиента за изграждане на взаимоотношения с конкретна туристическа фирма обикновено произлиза от по-ниско ниво – например като даване на свободен достъп на фирмата за отправяне на бизнес предложения; безплатен абонамент за информираност; регулярно получаване на конкретна персонализирана информация; напомняне за предстоящи събития; допускане на неформално и непосредствено активно фирмено комуникиране от общ характер; дори допускане на фирмата до личния (интимен) живот на клиента. Искането на позволение е приоритетна фирмена активност, посредством която последната има за цел най-малкото да задържи контакта с клиента на ниво ползотворна комуникация и последващо развиване на устойчиви бизнес взаимоотношения. Искането на позволение отключва възможността за фирмата да поддържа определено комуникационно равнище с клиента, дори без да е необходимо по същото време да отправя оферти или да осъществява скрити опити за продажби *(Джей, Р., 2001)*. Позволението е необходимо и дори за ефективно запознаване с потенциален клиент в рамките на самия първи контакт с него. Някои автори правят аналогия между исканото клиентско позволение и позволението за

започване процеса на ухажване в интимните отношения между хората (*Cavagnaro, E and F. Bosker, 2007)*. Тук се изхожда от презумпцията, че ако обектът на ухажване (клиентът) предостави своите координати за контакт, то възможността да се стартира връзка, която да прерасне в интимни, партньорски взаимоотношения се увеличава. Ето защо е изкушаваща аналогията между взаимоотношенията при хората и тези, които протичат по време на целия жизнен цикъл на клиента с фирмата. Имат се предвид етапите от привличането на клиента (неговото ухажване, въвличане във връзката) – транзакционен период, през развиване на взаимоотношения (получаване на дял от клиентския бюджет чрез задържането му във фирмата), към изграждане на стратегическо състояние на грижа за клиента („брачно партньорство") или така наречения дял от живота на клиента. Естествено взаимоотношенията могат да се прекратят както по взаимно съгласие, така и при инициатива на една от двете страни.

Степента на *лоялност* на клиента е следващата централна категория разграничаваща маркетинга на взаимоотношенията от характеристиките на транзакционните взаимодействия. Кратка характеристика за клиентска лоялност я определя като „поет значим ангажимент от страна на клиента да купува повторно или подкрепя покупките на предпочитан продукт или услуга последователно и за в бъдеще, въпреки ситуационните влияния и маркетингови усилия, които имат потенциала да предизвикат промяна на неговото поведение" *(Oliver, R., 1997).* Така поставената дефиния е ограничаваща тъй като лоялността е мултидеминсионална категория.

В едно от направленията на изследване на лоялността, акцентът пада върху процеса на повишаването й според движението на клиента по т. нар. „стълба на лоялността" в маркетинга на взаимоотношенията *(Bowie, D. and F. Buttle, 2004).* Според това градиране, клиентите за фирмата се разглеждат условно на няколко равнища – от намиращи се на най-ниско равнище на лоялност „*потенциални клиенти"* – тези които представляват обект на фирмената дейност за първи път; през „*купувачи"* – вече осъществили определена бизнес дейност с фирмата; „*текущи клиенти"* – осъществили повтарящи се покупки от фирмата, но отношението спрямо последната може да е негативно или в най-добрия случай неутрално; *„клиенти поддръжници"* – харесват фирмата и покупките от нея, но все още я подкрепят пасивно; *„клиенти защитници"* – активно препоръчват фирмата и нейната дейност, като развиват всякакви форми на вирусен маркетинг в нейна полза без да е необходимо да бъдат стимулирани за тези активности; *„клиенти партньори"* – разчитат на утвърждаването на партньорски взаимоотношения с фирмата и се определят на най – високото равнище на лоялността. Всяко следващо ниво по стълбата на лоялността може да отразява еволюционното развитие на клиент или неговото

директно попадане на съответното стъпало в процеса на взаимоотношенията му с фирмата.

Според други изследователски виждания, лоялността е необходимо да се отчита при използването най-малко на два метрични индикатора – *степен на потребление (покупки)* и *клиентска удовлетвореност (Brown, S., 2000)*. Във фазата на привличане (придобиване) на клиент, обемът на неговите покупки (приходите от тях за фирмата) е от ключово значение за оценка на транзакционния характер на лоялността му. Във фазите на задържане на клиента и стратегически развиваната грижа за него се счита, че по-точен измерител е неговото равнище на удовлетвореност. Клиентската удовлетвореност условно може да се раздели на три нива от гледна точка на неговите възприятия и интерпретация на резултата от потреблението – *„неудовлетворен клиент”*, при който качествата на полученият от фирмата продукт не покриват предварителните му очаквания; *„удовлетворен клиент”* – резултатът от потреблението му отговаря напълно на очакванията му, *„свръхудовлетворен клиент”* – полученото качество от продукта надхвърля очакванията му.

Основна слабост при класифицирането на клиентската удовлетвореност е, че информацията за нея се получава чрез събирането на качествена информация от полеви изследвания. Зададените от фирмата качествени измерители и интервали за степен на удовлетвореност може да не отразяват прецизно вижданията на всеки анкетиран клиент за нивото му на лична потребителска удовлетвореност. От друга страна множеството емпирични изследвания по въпроса не доказват наличието на стабилна висока корелационна връзка между клиентската удовлетвореност и равнище на лоялност. В тази връзка не е нетипично да се очаква, че дори високо удовлетворени клиенти могат да имат намерение за смяна на доставчика на услугата. Относно честата смяна на доставчици в туризма, високо удовлетворените клиентите отбелязват като типични причини склонността им към експериментиране, честата смяна на дестинацията на посещение, бягство от клишетата и модата, избягване на социалните условности и конюнктурни утвърждавания както и други техни специфични черти от потреблението им.

В този аспект много вероятно е смяната на доставчика на туристическите услуги да се развива паралелно и със смяната на туристическата дестинация. Обратната ситуация на клиентска удовлетвореност като че ли има по-логично обяснение. Ниско удовлетворените клиенти обикновено не прекратяват взаимовръзките си с доставчика на услугите поради някои от причините, като: високи разходи за смяна на доставчика; липса на достатъчни алтернативи пред клиента, изключително ниски ценови равнища на закупуваните продукти и др. При измерване на клиентската лоялност в стойностен и в качествен аспект, е необходимо да се изследва *референтната точка* от която клиентът ще стартира

при изграждане на взаимоотношенията си с фирмата. Референтната точка се свързва с предходното състояние на индивида спрямо което той оценява евентуалните ползи и загуби от свое последващо транзакционно поведение. От друга страна референтна точка може да бъде статуквото на индивида въобще, резултатът който той очаква или до който може да си позволи да действа *(Канеман, Д., 2012).* Референтната точка включва историята на транзакциите на индивида, промени във вкусовите му възприятия, потребности, желания, доходи и възнаграждения, промените в личния статус, личен мироглед и др. подобни. Историята на транзакционното поведение на индивида е и в основната логика на приложение на RFM[20] модела за анализ и оценка на взаимовръзки с клиенти. От гледна точка на своята референтна точка, за купувача е болезнено да прави отстъпки спрямо сегашната си позиция. Изключение правят т.нар. бедни купувачи, които са склонни да водят социален живот и под референтната си точка на потребление. Тогава изборите им през транзакционни отношения се определят като избори между по-малката загуба, вследствие на което и икономическото им поведение е доста по-рационално от това на богатите. В допълнение може да се отбележи, че за фирмата е от особена важност да разбере до колко значим може да бъде така наречения *клиентски компромис*. Последният представлява всичко, което клиентът е необходимо да пожертва за осъществяване на бизнес с фирмата чрез изграждане на взаимоотношения. С други думи клиентският компромис допълва в концепцията за удовлетвореност разликата между желаното от клиента и предлаганото от фирмата в качествено – стойностно измерение. Основна цел пред фирмата е да елиминира клиентският компромис за да спечели висока лоялност от страна на клиента.

Лоялността е категория, която трудно подлежи на точни качествено-стойностни измервания. Обикновено меродавна информация за степента на лоялност се извежда по индиректни начини в съответното проучване по темата. В тази насока, фирмата е необходимо да фокусира своята дейност в повишаване на клиентската лоялност в няколко направления. Първото от тях е *сегментиране според стойността на взаимоотношенията с клиента*, което да послужи като база за последващо *сегментиране според конкретните потребности на клиента*. Сегментирането според стойността на взаимоотношенията с клиента се основава на оценката относно това, което фирмата инвестира във взаимоотношенията с клиента и съответно това, което получава и / или очаква да получи до нейното изчерпване. Обикновено в процеса на взаимоотношения между фирма и клиент, колкото първата се фокусира по-успешно върху

[20] RFM (recency – изминало време от последната покупка; frequency – честота на покупки; money – обща стойност на извършените покупки) – модел за оценка и прогнозиране развитието на клиентското поведение относно фирмата с цел успешно управление на взаимоотношенията с него.

клиентските нужди, толкова по-успешно постига същата и дори по-висока степен на лоялност с по-малко инвестиции.

Въпросът за *стойността за клиента* се явява също основно предизвикателство при изграждане и управление на взаимоотношенията от страна на туристическите фирми. Някои автори виждат стойността за клиента като свързваща категория между възприетите качество и ценови равнища от една страна и равнищата на удовлетворение и лоялност от друга *(Gallarza, M. and others, 2012).* Други автори фокусират своите изследвания върху процеса на трансформация от така наречената *разменна стойност* на туристическия продукт, до *съвместно създавана стойност* между фирма и клиент *(Goldsmith, R. and others, 2012)*. Последната категория получава и своята крайна интерпретация в отделни изследвания под наименованието *стойност на клиентско изживяване (Gao, L., 2012)*. Тази стойност представлява предмет на изследване на база припокриващите се (споделяни) възприятия за създавани материални елементи, процес на обслужване и значение на цялостния продукт между фирмите и клиентите.

Друго направление за повишаване на лоялността е разработване от фирмата на прогностични модели на *отпадане на фирмени клиенти*. Организацията е необходимо да открои най-податливите на отпадане клиенти преди да вземе решение за инвестиране в тях на средства по програми за лоялност. Непрекъсваемостта на контактите с клиента, неговото задържане в бизнеса с фирмата, слабата му ценова чувствителност са допълнителни предпоставки за очаквана дългосрочна клиентска рентабилност в полза на фирмата. Оттам и метриките за измерване на клиентската рентабилност отварят нови стратегически виждания за продължаване на взаимоотношенията с различни видове клиенти, както на индивидуално, така и на групово равнище. Постигането на нарастваща дългосрочна клиентска рентабилност е логичен целеви резултат при развиването на взаимоотношения.

Сложността при измерването на лоялността в туризма произтича и от още няколко условности. Предполага се, че щом клиент осъществи повторна покупка от една и съща фирма, то неговата най-ниска степен на лоялност е вече налице т.е. пред фирмата се открива възможност за нейното повишаване в зависимост от поставените бизнес цели. Лоялността на клиента може да се резюмира и в контекста на повторните му посещения в дестинацията, независимо доколко използва услугите на един и същ доставчик. Отделен е въпросът дали покупките ще продължат и какви са истинските клиентски мотиви относно тези покупки. Мотивите за достигане до високо състояние на лоялност обикновено стоят извън тези за придобиване на икономически ползи. В този аспект рационалното поведение на индивида трудно може да се характеризира като водещо при търсене на варианти за повишаване на неговата лоялност. В тази връзка много

често в теорията и практиката се смесват категориите *лоялни* клиенти с *регулярни*. Последните се отличават с висок интензитет на покупки за период от време, ценово чувствителни са, с ниски разходи за смяна на доставчика, преднамерено осъществяват предимно транзакционни взаимодействия с фирмата и дори избягват нейните опити към последващо развиване на взаимоотношения. От друга страна фирмите обслужващи регулярни клиенти предлагат предимно стандартизирани, лесно копируеми продукти, на относително ниски ценови равнища. Регулярният клиент се интересува от икономически ползи и е готов лесно да смени доставчика на продукта, ако получи по-изгодни бизнес предложения от конкурентите. Регулярността на един клиент е предпоставка за развиването му в по-високи степени на лоялност, но далеч не е достатъчна за постигането на най-високо равнище на лоялност в стратегически аспект. Всъщност повечето от така наречените „програми за лоялност" в туризма (например – акумулиране на въздушни мили, точки за престой в бизнес заведения за пребиваване и др. подобни) реално утвърждават ниските нива на лоялността, при които клиентът има чертите по-скоро на регулярен потребител със силно транзакционни мотиви на потребление.

В заключение може да се отбележи, че основните конструктивни елементи за управление на взаимоотношения с клиенти в туризма се отличават с висока степен на вариативност при тяхното приложение. Ето защо техническите метрики на измерване в областта на маркетинга на взаимоотношения с клиенти е необходимо да отстъпят централно място на качествените. Оттук и копирането на емблематичен управленски опит и възприемането на рецепти в управление на взаимоотношенията с клиенти може силно да подведе заинтересованата фирма. Необходимостта от стратегическо установяване на основните конструктивни елементи за изграждане и развиване на взаимоотношения с клиенти е ключова предпоставка за устойчиво маркетингово управление на фирмената рентабилност.

ЛИТЕРАТУРА

1. Годин, С. Маркетинг на позволението, „Локус публишинг", София, 2010, с. 99
2. Джей, Р. Хитрините на клиентите, „Фокус", София, 2001, с. 123
3. Желев, С. и Г. Младенова. Маркетинг., Издателски комплекс – УНСС, София, 2012 г.
4. Канеман, Д. Мисленето, „Изток - Запад", София, 2012, с. 367
5. Altan, M. and K. Eringa. Service management and marketing: How to create value for the customer. // Services and Sustainability. A travellers guide', CHN University, 2007, pp. 3 - 30

6. Bowie, D. and F. Buttle. Hospitality Marketing. An introduction. Elsevier Butterworth – Heinemann, 2004, p. 298
7. Brown, S. Customer Relationship Management. A Strategic Imperative in the World of e- Business, "John Wiley & Sons", 2000, p. 60
8. Cavagnaro, E and F. Bosker, Services and sustainability. A travellers'guide, CHN University, Netherlands, p. 16
9. Conway, T. International relationship marketing.The importance of physical distance. European Journal of Marketing, vol. 34, 11/12, 2000
10. Crandall, R. and W. Crandal. New methods of competing in the global marketplace. Critical success factors from service and manufacturing., "Taylor and Francis Group", 2008, p. 1 – 17
11. Gallarza, M. and others. Customer value in tourism services: meaning and role for a relationship marketing approach, Strategic marketing in tourism services, Emerald Group Publishing, Bingley, UK, 2012, pp. 147 – 159
12. Gao, L. Tourist experience development: designed attributes, perceived experiences and customer value, Strategic marketing in tourism services, Emerald Group Publishing, Bingley, UK, 2012, pp. 215 - 230
13. Goldsmith, R. and others. Introduction to experiential marketing, Strategic marketing in tourism services, Emerald Group Publishing, Bingley, UK, 2012, pp. 207 - 213
14. Kotler, Ph. and others.Marketing for hospitality and tourism. PrenticeHall, 1996, p.350
15. Moorman, R and others. Relationships between providers and users of marketing research: The dynamics of trust within and between organizations. // Journal of Marketing Research, vol. 29 (3), 1992, p. 314 – 329
16. Oliver, R. Satisfaction: A Behavioural Perspective on the Consumer, McGraw Hill, New York, 1997
17. Palmer, A. Relationship marketing: a universal paradigm or management fad?, The Learning Organization, Volume 3, Number 3, 1996