

ТЕХНИЧЕСКИ УНИВЕРСИТЕТ – СОФИЯ
Факултет по приложна математика и информатика
Катедра Информатика

**МОДЕЛИ И СОФТУЕРНИ АСПЕКТИ ПРИ
РЕАЛИЗАЦИЯТА НА КОМПЛЕКСНИ
АДМИНИСТРАТИВНИ УСЛУГИ**

ДИСЕРТАЦИЯ

за присъждане на образователна и научна степен “доктор”
по докторантската програма
„Информатика“

Направление 4.6. Информатика и компютърни науки

маг. инж. Виктор Венциславов Главев

Научен ръководител
доц. д-р Моско Ашер Аладжем

СОФИЯ • 2023

СЪДЪРЖАНИЕ

СПИСЪК НА ИЗПОЛЗВАНИ АБРЕВИАТУРИ И ОЗНАЧЕНИЯ	7
ВЪВЕДЕНИЕ.....	10
СТРУКТУРА НА ДИСЕРТАЦИОННИЯ ТРУД	11
ЦЕЛ И ЗАДАЧИ НА ДИСЕРТАЦИОННИЯ ТРУД.....	12
ГЛАВА 1. ПРЕДПОСТАВКИ ЗА РАЗВИТИЕ НА КОМПЛЕКСНО АДМИНИСТРАТИВНО ОБСЛУЖВАНЕ	13
1.1. Базисен модел на КАО	14
1.2. Проблеми при реализация на КАО	17
1.2.1. Неструктурирани данни, които не подлежат на междурегистров обмен	17
1.2.2. Анализ на структурирани данни и изготвяне на отговор	17
1.3. Информационни системи в администрацията	18
1.3.1. Административни информационни системи.....	19
1.4. Системи на електронното управление	21
1.4.1. Централизирани системи за електронно управление	23
1.4.2. Интеграционни шини.....	24
1.5. Технологии за обмен на данни в администрацията	26
1.5.1. Протоколи и стандарти при реализацията на уеб услуги.....	29
1.6. Нормативна рамка, регламентираща изпълнението на комплексно административно обслужване	31
ГЛАВА 2. ОСНОВНИ ИЗИСКВАНИЯ ПРИ ПРОЕКТИРАНЕ НА РЕФЕРЕНТНА СОФТУЕРНА АРХИТЕКТУРА НА АДМИНИСТРАТИВНА ИНФОРМАЦИОННА СИСТЕМА, ОРИЕНТИРАНА КЪМ ПРЕДОСТАВЯНЕ НА КОМПЛЕКСНИ АДМИНИСТРАТИВНИ УСЛУГИ.....	34
2.1. Информационна сигурност	34
2.1.1. Количествена оценка на сигурността в софтуерната архитектура	35
2.2. Оперативна съвместимост.....	38
2.3. Нормативни фактори.....	40
2.4. Разширяемост	40

2.4.1.	Гъвкавост	41
2.4.2.	Настройваемост.....	41
2.4.3.	Възможност за персонализиране на процеси.....	41
2.4.5.	Възможност за надграждане	41
2.5.	Съвместимост (съвместимост напред и обратна съвместимост).....	42
2.6.	Надеждност.....	44
2.7.	Възможност за поддръжка.....	46
2.8.	Възможност за тестване и проверка	47
 ГЛАВА 3. АРХИТЕКТУРНИ МОДЕЛИ.....		49
3.1.	Архитектурен модел Клиент-сървър	50
3.1.1.	Анализ на предимства и недостатъци на архитектурата Клиент – сървър.....	52
3.2.	Многослойна архитектура.....	52
3.2.1.	Основни концепции при многослойните архитектури	53
3.2.2.	Анализ на предимства и недостатъци на многослойната архитектура	55
3.3.	Архитектура, ориентирана към услуги	56
3.3.1.	Анализ на предимства и недостатъци на архитектурата, ориентирана към услуги	58
3.4.	Архитектурен модел Микро-ядро	59
3.4.1.	Анализ на предимства и недостатъци на архитектура Микро-ядро	60
3.5.	Архитектурен модел Микро-услуги.....	62
3.5.1.	Топологии при архитектура Микро-услуги.....	64
3.5.2.	Анализ на предимства и недостатъци на архитектурата Микро-услуги	66
 ГЛАВА 4. АРХИТЕКТУРА НА АДМИНИСТРАТИВНА ИНФОРМАЦИОННА СИСТЕМА, ОРИЕНТИРАНА КЪМ ПРЕДОСТАВЯНЕ НА КОМПЛЕКСНИ АДМИНИСТРАТИВНИ УСЛУГИ.....		68
4.1	Унифициран модел на процес за предоставяне на комплексни административни услуги	68
4.1.1	Заявяване на комплексна административна услуга.....	70
4.1.2	Регистрация на документ – заявление за комплексна административна услуга	70

4.2.2. Обработка на заявление за комплексна административна услуга.....	70
4.2.3 Изготвяне на резултатен документ.....	72
4.2.4. Запис на данни и обстоятелства в регистри от данни.....	72
4.2.5 Регистриране и предоставяне на резултатния документ на заявителя на услугата. .	73
4.2 Модули, изграждащи архитектурата на административната информационна система.....	73
4.2.1 Модули в ядрото на административната информационна система.....	74
4.2.2. Основни модули, необходими за предоставяне на комплексно административно обслужване.....	82
4.2.3. Допълнителни и спомагателни модули.....	91
4.2.4. Примерна реализация на Модул за електронно удостоверение.....	92
4.3. Основни интерфейси за комуникация между модулите в предлаганото архитектурно решение.....	99
4.3.1 Интерфейс Document Management.....	99
4.3.2 Интерфейс Workflow Provider.....	101
4.3.3 Интерфейс E-certificate Processor.....	101
4.3.4 Интерфейс Task Management.....	102
4.3.5 Интерфейс Register Management.....	102
4.3.6 Интерфейс RegiX Request.....	102
4.3.7 Интерфейс СЕОС Request.....	103
4.3.8 Интерфейс ССЕВ Request.....	103
4.3.9 Интерфейс Е-Разп. Request.....	103

ГЛАВА 5. НАДГРАЖДАНЕ НА ПРОТОКОЛ, ИЗПОЛЗВАН ОТ СЕОС ЗА РЕАЛИЗИРАНЕ НА ИНТЕР-АДМИНИСТРАТИВНИ РАБОТНИ ПРОЦЕСИ..... 105

5.1. Надграждане на централния регистър на участниците в СЕОС.....	106
5.1.1 Надграждане на регистъра с предоставяните по електронен път услуги от администрацията.....	106
5.1.2 Надграждане на регистъра с данни за определени служители, длъжности или отдели в администрацията.....	107
5.2. Надграждане на протокола на СЕОС.....	107

ГЛАВА 6. ВЕРИФИКАЦИЯ НА ИЗГРАДЕНИТЕ МОДЕЛИ.....	116
6.1. Архитектура на АИС АКСТЪР ОФИС	116
6.1.1. Модул за електронно заявяване на комплексна административна услуга	118
6.1.2. Модул за управление на документи.....	118
6.1.3. Модул за управление на задачи и работни процеси	119
6.1.4. Модул за управление на електронни регистри	132
6.2. Реализация на интер-административни, споделени работни процеси в АИС АКТСЪР ОФИС	133
ИЗВОДИ	138
ЗАКЛЮЧЕНИЕ.....	139
ПРИНОСИ	140
ИЗПОЛЗВАНА ЛИТЕРАТУРА.....	141
ПРИЛОЖЕНИЕ 1 СПИСЪК НА АДМИНИСТРАЦИИ, ПРЕДОСТАВЯЩИ ЧАСТИЧНО И ПЪЛНО КОМПЛЕНСНО АДМИНИСТРАТИВНО ОБСЛУЖВАНЕ С АИС АКСТЪР ОФИС	146
ПРИЛОЖЕНИЕ 2 ПУБЛИКАЦИИ ПО ТЕМАТА НА ДИСЕРТАЦИОННИЯ ТРУД	148
ПРИЛОЖЕНИЕ 3 УДОСТОВЕРЕНИЯ ЗА УСПЕШНО ВНЕДРЯВАНЕ НА РАЗРАБОТЕНИТЕ МОДЕЛИ ЗА КОМПЛЕКСНО АДМИНИСТРАТИВНО ОБСЛУЖВАНЕ И ИНТЕР-АДМИНИНИСТРАТИВНИ РАБОТНИ ПРОЦЕСИ.....	149

БЛАГОДАРНОСТИ

Изказвам своята благодарност на своя научен ръководител доц. д-р Моско Аладжем за всичко, на което ме научи. Благодаря за непрестанните усилия, ценните съвети и напътствия, знанията и опита, които сподели и продължава да споделя с мен.

Изказвам своята благодарност и към Гергана Герова за подкрепата, разбирането и търпението, което проявява към мен всеки ден, и за помощта при оформянето на този труд. Без нейната подкрепа настоящата работа би била невъзможна!

Не на последно място искам да благодаря и на доц. д-р Весела Пашева (1946-2023 г.), за подкрепата, която ми оказа.

СПИСЪК НА ИЗПОЛЗВАНИ АБРЕВИАТУРИ И ОЗНАЧЕНИЯ

Абревиатури и означения на кирилица

Абревиатура	Значение
АИС	Административна информационна система
АМ	Архитектурните модели
АО	Административен орган
АПК	Административнопроцесуален кодекс
АУ	Административна услуга
ВАУ	Вътрешна административна услуга
ВЕАУ	Вътрешна електронна административна услуга
ЕИСУЧРДА	Единна информационна система за управление на човешките ресурси в държавната администрация
ЕПДЕАУ	Единен портал за достъп до електронни административни услуги
ЕРОС	Европейска рамка за оперативна съвместимост
е-управление	Електронно управление
ЗЕУ	Закон за електронно управление
ИКТ	Информационни и комуникационни технологии
ИС _{иг}	Информационна сигурност
ИС _{ис}	Информационна система
ИТ	Информационни технологии
КАО	Комплексно административно обслужване
КАУ	Комплексна административна услуга
МЕУ	Министерство на електронното управление
НАО	Наредбата за административно обслужване

НОИИСРЕАУ	Наредба за общи изисквания към информационните системи, регистрите и електронните административни услуги
ООП	Обектно ориентирано програмиране
ОС	Оперативна съвместимост
ПАД	Първичен администратор на данни
ПИ	Потребителски интерфейс
СЕОС	Система за електронен обмен на съобщения
СРЗИ	Столична регионална здравна инспекция
ССЕВ	Система за сигурно електронно връчване
СУДП	Системата за управление на документи и преписки

Абревиатури и означения на латиница

Абревиатура	Значение
API	Application Programming Interface
BPEL	Business Process Execution Language
HTTP	Hypertext transfer protocol
HTTS	Secure Hypertext State Transfer
IEEE	Institute of Electrical and Electronics Engineers
ISO	International Organization for Standardization
JSON	JavaScript Object Notation
MVC	Model-View-Controller
OASIS	Organization for the Advancement of Structured Information Standards
ORM	Object Relational Mapper
OSI	Open Systems Interconnection
RMI	Remote Method Invocation

SOA	Service-oriented architecture
SOAP	Simple Object Access Protocol
W3C	World Wide Web Consortium
WSDL	Web Services Description Language
XML	eXtensible Markup Language
XSD	XML Schema Definition

ВЪВЕДЕНИЕ

Развитието на информационните и комуникационните технологии (ИКТ) през последното десетилетие предоставя нови възможности за усъвършенстване на процесите, реализирани в публичната администрация. Такъв процес е предоставянето на комплексни административни услуги (КАУ) на гражданите и бизнеса. По същество комплексните административни услуги представляват усъвършенстван модел на административни услуги, при който заявителят на услугата делегира правото на дадена администрация да изиска необходимите за изпълнението на услугата документи и данни от други администрации.

Основните проблеми, свързани с ефективната реализация на комплексно административно обслужване (КАО) в администрациите е ниското ниво на развитие на електронно управление в Република България. Под електронно управление (е-управление) се разбира управление в електронна среда на нормативните взаимоотношения, административните процеси и обслужване и взаимодействието с потребителите с използване на информационни, статистически и математически модели и методи на обработка на данни, информация и знания, които осигуряват много по-високо ниво на ефективност на управлението. Е-управлението може да се разглежда като инструмент за повишаване на ефективността на процесите, изпълнявани в една администрация, но и за облекчаване на процесите при взаимодействие между администрации, граждани и бизнес [2]. Именно поради тази причина развитието на е-управление чрез внедряване на съвременни ИКТ и софтуерни решения е от ключово значение за предоставяне на КАО на гражданите и бизнеса. И ако през последните години, с развитието на междуминистерски обмен на документи, всички администрации разполагат със съвременни системи за регистрация на документи, то един от основните проблеми произхожда от факта, че голяма част от регистрите, поддържани от администрациите, първични администратори на данни, се съхраняват в неелектронен, хартиен вид. Дори и при наличието на достъп до структурирани данни, съхранявани в електронни регистри, ИС на администрациите не разполагат с функционалност за автоматизирано извличане на данни от тези регистри и последващия им анализ.

Дисертационният труд разглежда важен за обществото проблем, защото осигуряването на възможност за предоставяне на КАО от администрациите е основен приоритет, тъй като то гарантира:

- предоставяне на качествени услуги с по-висока ефективност и ефикасност;
- повишаване удовлетвореността на гражданите и бизнеса;
- намаляване на административната тежест;
- намаляване на възможностите за корупционни практики.

СТРУКТУРА НА ДИСЕРТАЦИОННИЯ ТРУД

Дисертационният труд е организиран във въведение, цел и задачи, шест глави и заключение, съдържащо направени изводи и обобщени приноси.

Глава първа описва базисен модел на КАО, проблеми, свързани с реализацията на КАУ и предпоставки за неговото развитие. Разгледани са централизирани и хоризонтални системи на е-Управление в Република България, които са необходими предпоставки за това развитие.

В глава втора са представени основните нефункционални изисквания, известни още като атрибути на качеството, които трябва да бъдат взети под внимание при проектиране на софтуерна архитектура на АИС, осигуряваща изпълнение на процесите по предоставяне на КАО. Описани са изискванията : информационна сигурност, оперативна съвместимост, нормативни фактори, разширяемост, съвместимост и надеждност, както и защо те са от изключителна важност при проектиране на ИС, използвани от администрацията.

Глава трета представлява преглед на основните архитектурни модели, които са релевантни към изискванията, разгледани в глава втора. Представени са основните им характеристики и анализ за приложението им в предлаганото архитектурно решение.

В глава четвърта е представен унифициран модел на процес за предоставяне на КАО. В детайли са разгледани основните етапи от него. На базата на предложения модел е представен референтна софтуерната архитектура на АИС, чрез която се имплементират всички процеси от унифицирания модел.

Глава пета разглежда възможността за надграждане на протокола на СЕОС за нуждите на реализация на интер-административни работни процеси между две или повече администрации, които подпомагат процесите по предоставяне на КАО.

Глава шеста представя резултати от имплементирането на създадения модел на процеси за КАО и референтна софтуерна архитектура за КАО в АИС, внедрена в множество държавни администрации в Р. България, както и представители на частния бизнес.

ЦЕЛ И ЗАДАЧИ НА ДИСЕРТАЦИОННИЯ ТРУД

Въз основа на анализ на текущото състояние на внедряване на ИКТ в процесите по предоставяне на услуги в държавната администрация и отчитайки съвременните тенденции в разработката на водещи решения в предметната област, целта на дисертационния труд е:

Да се създаде модел на процесите, обхващащи всички етапи по предоставяне на комплексно административно обслужване и да се разработи референтна софтуерна архитектура на административна информационна система, ориентирана към предоставяне на комплексни административни услуги.

За постигане на поставената цел са дефинирани следните задачи за изпълнение:

1. Да се създаде модел на процеса по предоставяне на КАО, обхващащ всички етапи от заявяване на услугата до предоставянето на резултантен документ. Този модел трябва да води до:
 - опростяване на работните процеси от гледна точка на служителите на администрацията чрез автоматизирано извличане на данни, техния анализ, изготвяне на резултантни документи и/или автоматизирано поддържане на регистри до степен, минимизираща ежедневната рутинна работа;
 - възможност за съвместна работа между служителите на няколко администрации чрез възможност за изпълнение на колаборационни, интер-административни работни процеси за създаване и съгласуване на документи, общи за администрациите, участващи в работния процес;
2. На базата на създадения модел да се проектира архитектурен модел на АИС, ориентирана към КАО, чрез дефиниране на основните компоненти на системата., Моделът ще има следните характеристики:
 - Различните компоненти да бъдат максимално независими един от друг, така че да могат да бъдат доставени от различни производители на софтуер;
 - Отделните компоненти да бъдат лесно надграждаеми от гледна точка на непрекъснатото развитие на е-Управлението в Р. България;
 - Да позволява гъвкавост от гледна точка на изменението на процесите по предоставяне на КАО и нормативната уредба.
3. Разработените модели да бъдат верифицирани чрез приложение в реална АИС

ГЛАВА 1. ПРЕДПОСТАВКИ ЗА РАЗВИТИЕ НА КОМПЛЕКСНО АДМИНИСТРАТИВНО ОБСЛУЖВАНЕ

Дисертационната работа разглежда и анализира основните проблеми, свързани с възможността за реализиране и предоставяне на КАО от гледна точка на съществуващите софтуерни решения, използвани в администрациите в Р. България.

За да бъде разбран процесът по предоставяне на КАО, е необходимо да се дефинират следните понятия:

- **Административно обслужване** е всяка дейност по извършване на административни услуги от структурите на администрацията и от организациите, предоставящи обществени услуги;
- Предоставяне на **административна услуга (АУ)** е издаване на индивидуален административен акт, с който се удостоверяват факти с правно значение, признава се или се отрича съществуването на права или задължения, както и се извършват други административни действия, които представляват законен интерес за физическо или юридическо лице;
- **Вътрешна административна услуга (ВАУ)** е административна услуга, която един административен орган предоставя на друг при осъществяване на неговите правомощия;
- **Комплексно административно обслужване (КАО)** е административно обслужване, при което административната услуга се предоставя от компетентните административни органи, лицата осъществяващи публични функции и организациите, предоставящи обществени услуги, без да е необходимо заявителят да предоставя информация или доказателства, за които са налице данни, събирани или създавани от административни органи, първични администратори на данни (ПАД), независимо дали тези данни се поддържат в електронна форма или на хартиен носител;
- **Вътрешна електронна административна услуга (ВЕАУ)** е вътрешна административна услуга, която може да се заяви и/или предостави от разстояние чрез използването на електронни средства;
- **Административна информационна система (АИС)** е система, която осигурява поддържането и обработката на данни за обработката на електронни документи и документи на хартиен носител при предоставяне на административни услуги и изпълнение на административни процедури [4].

От определението за комплексно административно обслужване може да се направи заключение, че за реализацията му от съществено значение е възможността административните органи да обменят информация помежду си. Тази информация може да бъде във вид както на структурирани данни, така също и в неструктурирани документи. Това означава, че информационните системи на администрацията трябва да обменят данни и документи, и да анализират обменените данни като по този начин улесняват работата на служителите. За реализация на КАО е необходимо е да бъде изграден софтуерен архитектурен модел, който от една страна трябва да бъде базиран на съвременни информационни и комуникационни технологии, а от друга - трябва да се отчете и фактът, че голяма част от данните в администрациите не са дигитализирани, т.е. информацията в тях е в неструктурирани документи.

1.1. Базисен модел на КАО

На Фиг. 1 по-долу е представен базисен модел на КАО. Той описва процесите, изпълнявани от административните органи при предоставяне на комплексна административна услуга (КАУ).

При представяне на КАУ могат да бъдат разграничени следните етапи:

1. **Заявяване на КАУ** - клиентът (физическо или юридическо лице), наречен заявител на КАУ, попълва заявление за комплексна административна услуга. Заявлението може да бъде изпратено по електронна поща, чрез уеб приложение (като електронна административна услуга), да бъде попълнено на гише в съответната администрация или заявено чрез устна форма посредством телефонен разговор;
2. **Обработка на заявлението** - администрацията, приела заявлението, стартира процес по обработка, наречен вътрешно-административен процес. В този момент, служителят на администрацията проверява списъка с приложените от заявителя документи и ако някои от необходимите за изпълнението на услугата документи или удостоверителни факти не е предоставен, се инициира една или повече вътрешно-административни услуги. Целта на тези услуги е извършване на проверка и събиране на данни от първични регистри, най-често поддържани от други администрации (Администрация Б и Администрация В на фиг. 1).
3. **Събиране на факти и обстоятелства** - администрацията изчаква завършването на всяка от инициализираните вътрешно-административните услуги. В резултат от тяхното изпълнение, администрацията получава документи или данни, удостоверяващи факти, извършва анализ върху тях и определя дали може да изпълни заявената административна услуга;

4. **Издаване на документ-резултат** - ако в резултат от заявената АУ се издава удостоверение, администрацията изготвя документ-резултат, който се предоставя заявителя на услугата;
5. **Съхранение и обработка на резултата** - издаденото удостоверение се съхранява в регистрите на администрацията. Възможно е и инициране на процес по служебно уведомяване, при който администрацията уведомява други заинтересовани администрации за обстоятелства по промяна на данни.

Всеки един от гореописаните етапи подлежи на оптимизация чрез внедряване на подходящи информационни технологии. И докато повечето администрации предоставят възможност за електронно заявяване на административни услуги, то много малка част от тях разполагат със средства за изпълнение на автоматизирана електронна обработка на заявлението за комплексно обслужване [4].



Фиг. 1 Базисен модел на комплексно административно обслужване

1.2. Проблеми при реализация на КАО

1.2.1. Неструктурирани данни, които не подлежат на междурегистров обмен

Изпълнението на КАУ изисква от администрациите да обменят данни помежду си. Тези данни най-често се съхраняват и обработват от администрации, наречени ПАД.

В зависимост от физическия начин на тяхното съхранение могат да бъдат разграничени два типа данни:

- *Структурирани данни* – за съхранението им се използва електронен регистър, най-често в база данни, отговаряща на изискванията за съвременни информационни системи;
- *Неструктурирани данни* – те се съхраняват на хартиен носител или в електронен вид, но неподходящ за автоматизиран обмен. Най-често това са регистри, съдържащи данни за населението преди 2000 година.

Проблемът, който съществува при обмена на неструктурирани данни, се състои в това, че когато дадена администрация А се нуждае от данни или обстоятелства, които се съхраняват в неструктуриран вид от администрация Б, неминуемо е необходимо участието на човек в подготовката на данните, които ще се изпратят като отговор към администрация А. Към момента, в този случай, най-често те се изпращат като неструктуриран документ, чрез който се удостоверяват определени факти, от които администрация А се нуждае за да изпълни своята административна услуга. Така получения неструктуриран документ в администрация А не подлежи на машинен анализ, което отново налага човешка намеса.

Този проблем може да бъде решен с разработката на програмен интерфейс, с помощта на който администрация Б директно да предостави данните в структуриран вид. За решаването на този проблем би могло да бъде използван централизиран подход, подобен на средата за междурегистров обмен RegiX или локално решение, предоставено от АИС на администрацията, изискваща необходимите ѝ данни – администрация А.

1.2.2. Анализ на структурирани данни и изготвяне на отговор

Когато една администрация получи всички данни, необходими за изпълнението на дадена КАУ, специалистите в администрацията извършват анализ на тези данни и определят дали услугата да може да бъде изпълнена или не. В по-голямата част от случаите, тези анализи могат да бъдат извършени машинно, чрез подходящ софтуерен

модул, който да определи дали всички предпоставки за изпълнението на услугата са налице. Ако това е изпълнено, най-често администрацията издава документ от вида на удостоверение, което се предоставя на заявителя. Този документ има предварително дефинирана структура и точно определени данни, които се вписват в него. По този начин, издаването на този документ може да бъде извършено машинно, отново чрез софтуерен модул. Така, работата на служителите в администрацията ще се оптимизира и те ще могат да насочват вниманието си към други процеси и задачи, изискващи нестандартен подход за изпълнението им.

1.3. Информационни системи в администрацията

Информационна система (ИС_{ис}) е организирана съвкупност от данни, дефинирани дейности по въвеждане, обработка и анализ на информация, методи и процедури, които осигуряват функционирането на административния орган (АО) с оглед постигането на предварително набелязани цели. Информационните системи са основата на автоматизираните системи за управление, които се реализират на базата на високотехнологично хардуерно и софтуерно оборудване. Информационната система се определя чрез данните и/или функциите, които поддържа [3]. Всеки АО, независимо от това дали е представител на централна, областна или общинска администрация използва информационни системи, които могат да бъдат групирани в следните категории:

- **Административна информационна система**, чрез която се регистрират всички входящи, изходящи и вътрешни документи, в това число заявления за предоставяне на услуги, независимо от начина на заявяване – на място в администрацията или по електронен път чрез уеб портал или мобилно приложение за заявяване на ЕАУ. Освен поддържане на регистри с всички документи, получавани и създавани от администрацията, АИС управлява и самите административни процеси, които се реализират в конкретната администрация. Изпълнението на КАО е административен процес и поради тази причина АИС има съществена роля при предоставянето на КАУ.
- **Системи за управление на регистри** – това са ИС_{ис}, чрез които всеки АО, представляващ първичен администратор на данни, извършва своята дейност по събиране и управление на данни за граждани, организации или обекти. Тези ИС_{ис} съхраняват информация под формата на структурирани данни (записи) за обекти или субекти и връзките между тях. Основните операции, които се поддържат са добавяне, редактиране или изтриване на запис и различни видове справки. Обикновено един АО поддържа множество регистри. Внедряването на

ИС_{ис} за управление на регистри е от изключително значение за развитието на КАО [13].

1.3.1. Административни информационни системи

Внедряването и използването АИС от АО се определя от чл. 4 на [10], според който ръководителите на администрациите осигуряват разработването и внедряването на ИС, които осигуряват поддържането и обработката на данните за оборота на електронни документи и документи на хартиен носител при предоставянето на административни услуги и вътрешни електронни административни услуги. Алинея 5 от същия член определя, че създаването на електронни документи в администрацията се осигурява със средства на АИС или външно програмно приложение.

Наредбата определя също така и функционалните изисквания на АИС, както и структурите на информационните обекти, които тя трябва да поддържа. Дефинирани са следните видове обекти:

- Обект *Потребител* – това е обект, който съхранява данни за идентификация и автентикация на потребителите на системата;
- Обект *Документ* – това е обект, който съхранява данни за получените или създадени документ в администрацията, независимо от това дали са електронни или на хартиен носител. Основните данни, включени в състава на обекта са: създаден от (идентификатор на потребителя, създал документа), наименование, указател за връзка към файлово съдържание на документа.
- Обект *Задача* – за изпълнение на процедури по контрол на административната дейност, включва описание на задачата както и етап от административна услуга или процедура, изпълнител на задачата, дати свързани с започването и завършването на задачата
- Обекти *Физическо лице* и *Юридическо лице* с идентификатори съответно ЕГН и ЕИК. Тези обекти представляват кореспонденти – получател или подател на документа [10].

Регистрацията на документи в администрацията се извършва в официален документен регистър, представляващ база от данни в състава на АИС. Всяка администрация може да има повече от една АИС и само един официален документен регистър, освен ако не е указано друго в нормативен акт. Регистрацията на документи, свързани с предоставянето на административно обслужване на граждани и комуникацията между отделните администрации (включително вътрешни административни услуги, намиращи се в основана на КАО) се извършва винаги в официалния документен регистър. Член 25 от [10] определя правото на администрацията да създава

специфични ИС_{ис}, в които да регистрира документи, свързани с предоставяне на административни услуги. За всеки регистриран документ, вписан в официалния документен регистър, в АИС се съхранява следната информация:

- време на вписване на документа;
- потребител, извършил вписването;
- уникален регистров идентификатор.

АИС поддържа набори от тематично свързани документи, наречени преписки. Всяка преписка се състои от три раздела – официален, вътрешен и контролен. В АИС се създава нова преписка при регистрация на документ, с който е заявена електронна административна услуга или документ, с който се инициира процедура по обработка. В официалния раздел на преписката се съхраняват документи, приети от администрацията, издадени от АО или създадени в администрацията със завършен цикъл на обработка. Във вътрешния раздел на преписката се съхраняват документи, създадени от администрацията, но с все още незавършен цикъл на обработка и други помощни документи. В контролен раздел на преписката се съхраняват документи, които са изведени от официалния ѝ раздел, поради допуснатата грешка при създаването им или в съдържанието им. Всяка преписка представлява обект със следните данни: наименование, уникален идентификатор, време на създаване, идентификатор за персонализиран достъп през Интернет. [10] определя набор от ведомствени номенклатури, които се поддържат със средствата на АИС, в това число номенклатура на видовете документи, номенклатура на етапи от услуги или процедури, номенклатура на услуги и процедури (услугите, включени в номенклатурата са регистрирани в Регистъра на услугите), номенклатура на схемите за съхранение (определя как се съхранява документът и за какъв период от време). АИС обезпечават всички процеси, свързани с обработката на документа – регистрация, сканиране, извеждане, архивиране, групово редактиране, следене на измененията и версиите, управление на достъпа, както до описанието на документа, така и до неговото файлово съдържание. Всички процеси по възлагане на задачите, тяхното приемане от изпълнителя, отчитането на изпълнението и контрола им се извършват с помощта на системата.

В Раздел III на [11] и по-специално в чл.18 е записано, че обменът на електронни документи, съдържащи изявления между администрациите се извършва по технически протокол, определен от Министъра на електронното управление. АО са задължени да използват само АИС, които изпълняват този протокол. Тъй като възможностите и ограниченията при обмена на документи между администрациите е

от съществено значение за предоставяне на КАО, особеностите на протокола ще бъдат разгледани по-надолу.

1.4. Системи на електронното управление

Според стратегията за развитие на електронното управление в Р. България за периода 2014-2020 г., е-управление е такова управление, при което административните процеси, обслужването и взаимодействието с граждани се извършва с използване на информационни, статистически и математически модели и методи за обработка на данни, информация и знания, осигуряващи по-високо ниво на ефективност на управлението. Е-управление е средство за всеобхватно повишаване на ефективността на процесите в администрацията, както и облекчаване на процесите при взаимодействията между администрация, служители, граждани и бизнес чрез използването на е-услуги [6].

Е-управлението обхваща процесите по комуникация и предоставяне на услуги между следните субекти:

- **Административен орган → гражданин** – предоставяне на е-услуги на гражданите чрез използването на съвременни ИКТ, най-често чрез уеб приложения, достъпът до които се извършва чрез уеб портала на администрацията или друг централизиран портал, спазвайки стандарти за електронна сигурност и електронна идентичност;
- **Административен орган → бизнес** – предоставяне на съвременни решения, които оптимизират процесите и деловите отношения между администрацията и различните икономически субекти чрез внедряване на ИКТ;
- **Административен орган → административен орган** – развитие на ИКТ, които предоставят възможност за ефективно взаимодействие между администрациите при изпълнение на процесите, свързани с предоставяне на административно обслужване на гражданите и бизнеса;
- **Вътрешно-административни процеси** – подобряване на ефективността на работа на отделни звена в една администрация, чрез използване на съвременни ИКТ, включвайки софтуерни продукти за:
 - управление на документи;
 - административни процеси;
 - регистри, за които администрацията е първичен администратор;
 - и други допълнителни дейности, изпълнявани от администрацията.

Предоставянето на КАО е зависимо и от четирите процеса на е-управление, защото включва както процеси на комуникация между администрация и граждани или

бизнес, така и комуникация между две или повече администрации. Оттук може да се направи извод, че развитието на КАО е пряко свързано със степента на развитие на е-управление. Според дефиницията си е-управление е зависимо от внедряването на съвременни ИКТ, следователно и предоставянето на КАО е неразривно свързано със степента на използване на ИКТ. Според [9], стратегията за електронно правителство в Р. България датира още от 2002 г., с която се инициират процеси по създаване на ЕАУ за граждани и бизнеса, както и се полага основата вътрешна административна информационна и комуникационна инфраструктура. Там е записано още, че визията за е-управление в периода 2010-2015 г. включва предоставяне на граждани и бизнеса на качествено административно обслужване по електронен път по всяко време, на всяко място и чрез алтернативни канали за достъп. Посочени са и основните принципи, свързани с ИТ, които биха довели до реализацията на описаната визия, а именно:

- технологична неутралност;
- максимално използване на наличните технологии;
- въвеждане и прилагане на принципите на оперативна съвместимост – технологична, семантична и организационна.

Под технологична съвместимост се разбира отделните софтуерни компоненти и системи, използвани от администрацията да обменят данни, използвайки отворени интерфейси, общоприети стандарти и протоколи. Семантична оперативна съвместимост означава обменните данни между АО да имат едно и също семантично значение за всички участници в обмена, а организационната съвместимост се отнася до оптимизиране на административните процеси и съвместна работа между администрациите [35]. В актуализираната стратегия за ЕУ за периода 2019-2025 г. [2] са застъпени принципите за еднократно събиране и създаване на данни от АО, въвеждане на вътрешните електронни административни услуги, които трябва да положат началото на реално КАО. Документът се позовава и на архитектурата на е-управление [3], според която за да бъдат изпълнени тези принципи ИС_{ис} на АО трябва да бъдат интегрирани с хоризонталните и централните ИС_{ис} на ЕУ.

Хоризонталните системи на ЕУ включват следните по-важни системи:

- **Единен портал за достъп до електронни административни услуги (ЕПДЕАУ)** – порталът представлява единна точка за достъп до ЕАУ.
- **Система за е-Автентикация** - реализира процеси, свързани с еднократна идентификация и автентификация на заявители на предоставяните от администрациите е-услуги или други уеб приложения. Системата предоставя програмен интерфейс за интеграция на други ИС_{ис} с изпълняваните процеси по автентикация
- **Система за е-Оторизация** - реализира контрол на достъпа до системите и ресурсите на Е-Управление;

- **Система за електронни плащания към доставчици на услуги** – предоставя сигурен и проследим начин за създаване на платежен документ от тип „платежно нареждане“ и предаването му за изпълнение от доставчик на платежни услуги. Системата предоставя програмен интерфейс за интегриране на АИС на АО за при предоставяне на ЕАУ на граждани и бизнес.
- **Система за сигурно електронно връчване (ССЕВ)** - реализира изпращане, получаване и съхраняване на електронни документи за/от публични органи, физически и юридически лица, които са регистрирани в системата и имат персонални профили. Чрез тази система се предоставя услугата „електронна препоръчана поща“. Комуникацията чрез системата за е-Връчване замества класическия метод за доставка на писма;
- **Система за е-Форми** - осигурява създаване, редактиране и съхранение на електронни форми с възможност за последваща визуализация. Използва се за попълване на заявления за ЕАУ и последващото им изпращане към изпълнителя на АУ чрез ССЕВ [13].

1.4.1. Централизираните системи за електронно управление

Централизираните системи за ЕУ са информационни системи, които се предоставят за споделено ползване на участниците в електронното управление:

- **Система за общински облачни услуги** – тази системата предоставя възможност за автоматизиране на процесите по заявяване, обработка и предоставяне на услуги. Системата е неработоспособна към началото на 2021 г.. На практика нейната функционалност се дублира от Система за е-Форми.
- **Единна информационна система за управление на човешките ресурси в държавната администрация (ЕИСУЧРДА)** – тази система е управленческа, аналитична и оперативна система, чрез която се: извършват ежедневните оперативни процедури по управление на човешките ресурси в държавната администрация; акумулира в реално време в централна база данни пълна информация за цялостното състояние на човешките ресурси и заетостта в държавната администрация; способства за взимане на управленски решения на база анализи, изготвени въз основа на натрупаните данни; подпомагат органите на изпълнителната власт, както и ръководителите на отделни звена, при преглед и анализ на политиката в областта на управление на човешките ресурси в държавната администрация.
- **Система за управление на собствеността** – предоставя възможност за поддръжка на регистри и бази данни, свързани с управлението на общинската и държавната собственост.

1.4.2. Интеграционни шини

Интеграционните шини предоставят съвременни средства за контрол на достъпа и управление на натоварването на информационни ресурси чрез стандартизирани протоколи за взаимодействие, както между ресурсите на електронното управление, така и с външни за него ресурси [56].

По важните компоненти, които трябва да се разгледат, тъй като осигуряват обмен на документи и данни между администрациите, ключов фактор за КАО са Интеграционната шина за обмен на електронни документи и Интеграционната шина за обмен на удостоверителна и справочна информация

1.4.2.1. Интеграционна шина за обмен на удостоверителна и справочна информация – RegiX

Интеграционната шина за обмен на удостоверителна и справочна информация е реализирана чрез софтуерната система за автоматизиран достъп до данни от регистри, наречена още среда за междурегистров обмен – RegiX. Основната функция на системата е заявяване от страна на правоимащи лица (консуматори) на удостоверителна и справочна информация от регистрите и базите данни на ПАД. Удостоверяването на консуматорите се извършва чрез генерирани от системата и предоставените им цифрови сертификати. Заявяването и предоставянето на информация може да се извършва в синхронен или асинхронен режим в зависимост от работните процеси при доставчика. Генерираната информация се удостоверява чрез електронен и времеви печат от ПАД. Използването на системата от лицата по чл. 1, ал. 1 и 2 от ЗЕУ е задължително при обмен на удостоверителна и справочна информация. Шината дава необходимата функционалност за разработка на комплексни административни услуги и предоставя ефективен начин за еднократно събиране и многократно използване на информация съгласно принципа на „Служебното начало“.

Достъпът на консуматорите може да се реализира посредством два сценария на използване:

- чрез интегриране системата RegiX с ИС_{ис} на консуматора – това е правилният подход, който ще бъде разгледан в този дисертационен труд;
- чрез използване на уеб базирано решение, инсталирано и поддържано от МЕУ.

Средата за междурегистров обмен поддържа журнал на събитията, в който се записва информация за заявителя, правно основание, момент на заявяване и идентификатор на обекта, за който е заявена информацията. RegiX предоставя програмен интерфейс под

формата на Simple Object Access Protocol (SOAP) електронна услуга, която може да бъде интегрирана с всяка АИС с цел извличане на данни от регистри на ПАД. По този начин всеки АО посредством своята АИС може да реализира КАУ, използвайки вече събрани и съхранени данни за граждани, фирми и други субекти.

RegiX въвежда следната йерархична организация – администрациите предоставят достъп до данните, които събират, администрират и поддържат под формата на регистри, а всеки регистър предоставя набор от справки. Това означава, че достъпът до данните на ПАД се осъществява чрез “извикване” на конкретна справка от конкретен регистър. За целта АИС на администрацията-консуматор формира заявка към RegiX, която се състои от контекст и същински данни на заявката. Контекстовите данни винаги имат една и съща структура, а същинските данни зависят от конкретната справка, която ще се изпълни. Всяка заявка описва входните си данни и резултата от нейното изпълнение чрез XSD. Заявката за изпълнение на конкретна справка, както и резултата от нейното изпълнение се представя чрез XML документ. RegiX публикува списък с всички АО, които предоставят първични данни в уеб приложение RegiXInfo [26], което е достъпно за разработчиците на АИС. То предоставя следната информация – администрации ПАД и техните регистри и справки, които се предоставят посредством RegiX. За всяка справка е представен нейния идентификатор, структура на заявката и отговора, както и примерни данни.

1.4.2.2. Интеграционна шина за обмен на електронни документи (СЕОС)

Шината осигурява условия за обмен на документи между участниците, вписани в регистъра на участниците, улеснява технологията на деловодната кореспонденция и спестява време и ресурси. Решението се характеризира с висока степен на ефективност, надеждност, сигурност и гъвкавост, което позволява обмен на електронни документи директно между АИС на две администрации. Всеки участник в обмена представлява крайна точка под формата на софтуерен компонент, който може да изпраща и получава електронни документи. Регистърът на участниците поддържа актуални данни за участниците в обмена на съобщения, в т.ч. информация за участниците и техните крайни точки. Регистърът предоставя на участниците програмен интерфейс за извличане на информация за други участници, вписани в него, и потребителски уеб интерфейс за актуализиране на информацията, за себе си. Така изградената платформа за обмен на документи се нарича Система за електронен обмен на съобщения. Според техническата спецификация [14] обменът на документи между АО се извършва чрез размяна на съобщения, като за целта подателят на съобщението го изпраща до крайната точка на получателя, описана в Регистъра на участниците. След

осъществяване на връзка с крайната точка на получателя, подателят следва да изпрати подготвеното съобщение в определен формат, определен от спецификацията.

Съобщенията, които се поддържат от участниците са:

- Заявление за регистриране на документ;
- Заявление за проверка на състоянието на документ;
- Известие за състояние на документ;
- Известие за грешка.

1.5. Технологии за обмен на данни в администрацията

Съвременните ИС_{ис}, внедрени в администрацията се характеризират с особености като софтуерна архитектура, избрана платформа за реализация (MS .NET Framework, Java, PHP и др.), операционни системи, под управлението на които те работят (Windows, Unix). Поради голямото разнообразие от внедрени технологии при реализацията им, от съществено значение при обмена на данни между тях е използването на добре дефинирани, платформено независими стандарти за обмен на данни. Според [11] ИС_{ис} на администрациите осъществяват комуникации и други функции съгласно стандарти, описани в Регистъра на стандартите или други стандарти и препоръки на някои от следните организации - Българския институт по стандартизация, International Organization for Standardization (ISO), World Wide Web Consortium (W3C); Institute of Electrical and Electronics Engineers (IEEE).

Според [56], част от изискванията, които трябва да се спазват при разработване и надграждане на ИС_{ис} в администрацията са:

- ИС_{ис} да бъдат изградени спазвайки основните принципи, определени от архитектурния шаблон “архитектура, ориентирана към услуги”
- ИС_{ис} за изпълнение на административни услуги да бъдат достъпни през уеб услуги, като се спазват изискванията за оперативна съвместимост по отношение на стандарти, стандартизирани интерфейси и протоколи, вписани в Регистъра на стандартите;

Ориентираната към услуги архитектура (Service-Oriented Architecture (SOA) е архитектурен шаблон за проектиране на компютърен софтуер, в който отделните компоненти, наречени услуги, предоставят дадена функционалност на други компоненти чрез предварително определен комуникационен протокол, обикновено базиран на мрежова свързаност. Принципите на SOA предполагат независимост от доставчици, продукти или технологии [5]. Всяка услуга в SOA може да се разглежда като комбинация от изпълним код и данни, реализираща конкретна дискретна

функционалност – например идентификация на потребител, създаване на документ и др. Всяка услуга предоставя програмен интерфейс за нейното използване, като по този начин се постига ниско ниво на обвързаност между отделните компоненти, изграждащи системата. Достъпът до услугите се осъществява посредством стандартни Интернет протоколи като HTTP[14]/HTTPS[49] и добре утвърдени формати за представяне на данни като SOAP [55] и JSON [46]. Всяка услуга се описва чрез метаданни, които се публикуват по начин, който позволява на разработчиците лесно да интегрират, разработвания от тях софтуер с нея. Често софтуерните компоненти-услуги се разработват започвайки отначало, но в голяма част от случаите те служат за интегриране на вече създадени системи [52]. Целта на SOA е да се даде възможност на потребителите да комбинират заедно компоненти с определени функционалности, за да формират приложения, изградени почти изцяло от съществуващи софтуерни услуги [5].

В SOA отделните софтуерни компоненти могат да имат следните роли:

- *Доставчик на услуга* – компонентът предоставя своята функционалност като веб услуга. За да бъде достъпна тази функционалност за други системи и компоненти, той публикува и описание на програмен интерфейс, което представлява метаданни да услугата. За целта съществуват различни стандарти и езици като Web Services Description Language (WSDL) – за SOAP базираните веб услуги и OpenAPI за REST базираните услуги;
- *Клиент на веб услуга* – това е компонент, който използва (консумира) функционалност, предоставена от доставчик на услуга. За целта клиентът анализира описанието на програмния интерфейс на услугата за да може да “извика” нейната функционалност.

Понякога определени компоненти могат да имат и двете роли.

Предимствата на концептуалния модел на SOA, а именно слабата обвързаност между отделните компоненти, използването на широк набор от добре утвърдени стандарти за комуникация и представяне на данни, правят този архитектурен шаблон изключително подходящ при създаването на софтуерни решения за администрацията.

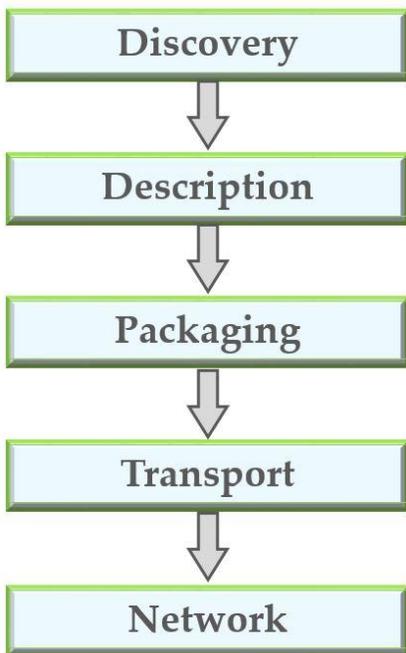
Основни градивни единици на SOA са веб услугите. По същество една веб услуга може да се разглежда като изпълним модул, от който отдалечено може да бъде извикана определена функция, наречена операция. Извикването се осъществява чрез изпращане на съобщение, формирано чрез използване на определен протокол. Структурата на веб услуга е представена на Фиг. № 2.



Фиг. 2 Структура на веб услуга

Кодът на приложението представлява изпълнима програмна единица – функция или метод, наречена операция в термините на веб услугите, която имплементира конкретна бизнес логика (идентификация на потребител, добавяне на запис в регистър, справка по данни от даден регистър и т.н.). Приемащата част на услугата (listener) получава входящи заявки под формата на съобщения от транспортния слой (HTTP, SOAP). Прокси на услугата анализира получените данни и ги преобразува в обръщания към кода на приложението. В резултат от изпълнението на операцията прокси на услугата може да формира отговор за приемащата част, с който тя да отговори на извикващата страна. Тази стъпка може да се пропусне, ако операцията не връща резултат. Веб услугите могат да се изпълняват в контекста на веб сървър, подобен на Internet Information Services, Apache или като самостоятелно приложение. Във втория случай операционната система изпраща получените съобщения директно към приложението.

Веб услугите представляват стек от протоколи и стандарти, подобен на TCP/IP протокола.



Фиг. 3 Стек от протоколи на уеб услугите

Слоеве за пакетиране, описание и откриване в него са съществени за предоставяне на възможността за свързване по време на изпълнение.

1.5.1. Протоколи и стандарти при реализацията на уеб услуги

Според [45,48] при разработката на ИС_{ис}, базирани на SOA архитектурата, за предоставяне на услуги от администрацията на граждани и бизнес, или интегриране на вече съществуващи системи, от изключително значение е постигането на оперативна съвместимост (Interoperability). Тя се дефинира като възможността на различните ИС_{ис} да работят заедно чрез обмен на данни и функционалност по време на изпълнението им. В [45] е направен извод, че съществуват много подходи за реализация на SOA, но с оглед постигане на добри нива на оперативна съвместимост е необходимо използване на набор от стандарти и протоколи като езика за описание на уеб услуги (WSDL), XML Schema Definition за описание на структури от данни, SOAP (Simple Object Access Protocol) за форматиране на обменяните съобщения, HTTP/HTTPS за обмен на съобщенията и BPEL (Business Process Execution Language) за описание на последователността на извикване на отделните уеб услуги, процес известен още като оркестрация на услуги.

WSDL [56] е XML [57] базиран език за описание и дефиниране на уеб услуги, като колекция от крайни мрежови точки (endpoints) или портове. Всяка крайна точка

реализира една или повече операции (функционалности), като за целта получава входни данни под формата на съобщение и опционално връща резултат, отново като съобщение. Изпълняваните операции и съобщенията, свързани с тях са описани абстрактно от WSDL без конкретизиране на мрежови протокол и формат на съобщението. Това означава, че WSDL е разширяем и позволява описание на крайни точки и техните съобщения без оглед на това какви формати на съобщенията и мрежовите протоколи се използват за комуникация. WSDL разделя абстрактните дефиниции на крайни точки и съобщенията от тяхната конкретна мрежова реализация или формат на данните. Това позволява повторно използване на следните абстрактните дефиниции: **съобщенията** представляват абстрактно описание на данните, които се обменят и **типове портове (port types)**, които представляват абстрактни колекции от **операции**. Конкретният протокол и формат на данните за даден вид порт образуват повторно използваемо **свързване (binding)**. Порт се дефинира чрез асоцииране на мрежови адрес с повторно използваемо свързване, а колекция от портове дефинира **услуга**.

Оттук WSDL документът използва следните елементи за дефинирането на веб услуги:

- **Типове (Types)** – контейнер за дефинициите на типовете данни, които са приложими към обменяните съобщения. За дефиниране на тип се използва система за дефиниране на типове данни (например XSD);
- **Съобщение (Messages)** – абстрактна типова дефиниция на обменяните данни. Служат за представяне на входните и изходящи параметри на операциите;
- **Операция (Operations)** – абстрактно описание на действие (функционалност), реализирана от услугата;
- **Тип порт (Port types)** – абстрактна колекция от операции, поддържани от една или повече крайни точки;
- **Свързване (Binding)** – избор на конкретен мрежови протокол и формат на данните за даден тип порт;
- **Порт (Port)** – крайна точка, дефинирана като комбинация от свързване и мрежов адрес;
- **Услуга (Service)** – комбинация от свързани помежду си крайни точки.

Веб услугите се дефинират чрез използване на шестте основни елемента: **типове, съобщение, операция, тип порт, свързване, порт и услуга**, които формират WSDL документ с описанието на услугата. Посредством информацията в този документ веб услугата може да бъде консумирана от всяка друга система, поддържаща необходимите протоколи. Въпреки, че WSDL не определя еднозначно използвания стандарт за формиране на обменяните съобщения между веб услугата и нейния

клиент, той дефинира така наречения SOAP binding, използващ SOAP [55] като протокол за формиране на тези съобщения. Оттук веб услугите често биват реферирани като SOAP услуги. Характерно за тях е, че най-често използват HTTP като преносен протокол, въпреки, че това не е задължително. SOAP е базиран на XML комуникационен протокол, с чиято помощ различни приложения и софтуерни компоненти могат да комуникират чрез базови Интернет протоколи като HTTP. За да изпълни предназначението си, той е разработен така, че да бъде платформено и езиково независим. SOAP съобщенията представляват XML документи с определена структура, състоящи се от заглавна част (header) и тяло (body). В заглавната част могат да бъдат добавяни потребителски инструкции, които определят правила за обработка на SOAP съобщенията. В зависимост от вида на SOAP заявката, в тялото се поставят структурираните данни, необходими за извикването на конкретна операция на веб услугата (SOAP-RPC заявка) или цял XML документ, който служи за пренос на обект (SOAP Document) [40].

HTTP е протокол от най-високото приложното ниво на Open Systems Interconnection (OSI) модела за комуникация в Интернет, използван за обмен на данни. Комуникацията се осъществява посредством заявки и отговори. Клиент изпраща заявка към даден сървър и очаква да получи отговор от него. Всяка заявка включва – метод, заглавна част и опционално тяло на заявката. Отговорът от страна на сървъра съдържа резултантен код, който се използва за да се определи дали заявката е изпълнена успешно, заглавна част и тяло, в което се намира резултата от изпълнението на заявката [47]. HTTP over TLS, известен като HTTPS е надграждане на HTTP, който предоставя възможност за криптиране на обменяните данни, така че те да не могат да бъдат достъпни или променяни от трети лица. Горепоисаните протоколи се използват най-често за реализация на веб услуги.

1.6. Нормативна рамка, регламентираща изпълнението на комплексно административно обслужване

През 2002 г. е поставено началото на оптимизиране на процесите в административните органи с Концепция за подобряване на административното обслужване, приета чрез решение № 878 на Министерски съвет. Целта на концепцията е внедряване на модел на обслужване на граждани и фирми, известен като “едно гише”. Този модел има за цел създаване на един “вход” и “изход” за документите, консолидиране на действията по подаване на заявление за извършване на административна услуга, получаване на

всякаква информация за хода на осъществяването ѝ и получаване на искания документ като краен резултат от услугата [48,57].

Следващата стъпка в развитието на административното обслужване е приемането на АПК през 2006 г., който модернизира отношенията между административните органи и гражданите, като определя основните принципи, които трябва да бъдат спазвани от страна на административните органи. Именно към тези принципи през 2014 г., с чл. 13а е добавен и принципът *“Административните органи прилагат комплексно административно обслужване”*, а чрез допълнителна разпоредба е дадена и точна дефиниция за КАО, гласяща: *“Комплексно административно обслужване”* е това обслужване, при което административната услуга се извършва от административни органи, от лица, осъществяващи публични функции, или от организации, предоставящи обществени услуги, без да е необходимо заявителят да предоставя информация или доказателствени средства, за които са налице данни, събирани или създавани от извършващия административната услуга първичен администратор на данни, независимо дали тези данни се поддържат в електронна форма или на хартиен носител. Задължението администрациите да прилагат КАО се определя от § 12. (1) *Административните органи въвеждат комплексно административно обслужване не по-късно от една година от влизането в сила на този закон.* [5]. Важни допълнения, касаещи КАО, могат да бъдат открити и в други нормативни актове, като Закона за електронно управление, който определя задължителен характер за еднократно събиране и създаване на данни от страна на администрациите. Чл. 2 гласи *Административните органи, лицата, осъществяващи публични функции, и организациите, предоставящи обществени услуги, не могат да изискват от гражданите и организациите представянето или доказването на вече събрани или създадени данни, а са длъжни да ги съберат служебно от първичния администратор на данните.* Дадена е и дефиниция на първичен администратор на данни – *това е административният орган, който по силата на закон събира и създава данни за гражданин или организация за първи път и изменя или заличава тези данни* [8]. С оглед на това, че един от подходите за подобряване на административното обслужване, е създаване и внедряване на информационни системи в администрацията, чл. 58а на ЗЕУ определя като задължително изискването при изготвяне на технически спецификации за провеждане на обществени поръчки да разработка, надграждане и внедряване на такива, включването на изискване за реализация на служебен интерфейс за автоматизиран онлайн обмен на данни и предоставяне на вътрешни електронни административни услуги. Есенцията на КАО е описана и в шеста алинея на чл. 31 от [1], която указва, че когато искането се отнася за комплексно административно обслужване, то може да бъде подадено до всеки административен орган, който участва в него. Административният орган, пред който е подадено искането, образува производството.

Взаимодействието с другите административни органи се осъществява по реда на наредбата по чл. 5а, ал. 1 от [7]. Крайният положителен ефект за гражданите и бизнеса от въвеждането на КАО, който реално намалява административната тежест върху тях е описан в чл. 36, ал. 4 от [1], а именно, че административните органи нямат право да изискват предоставяне на информация или документи, които са налични при тях или друг административен орган, а трябва да ги изискат служебно за нуждите на текущото производство.

Изводи

Процесите по предоставяне на КАО са свързани с обмен на структурирани данни между администрациите и ПАД. Липсата на структурирани данни е основен проблем при изпълнение на КАУ. Централизираните и хоризонтални системи на е-Управление предоставят определена функционалност за извършване на КАО, но АИС, използвани от администрациите не разполагат с подходящи интеграционни модули, които да оптимизират работата на служителите.

ГЛАВА 2. ОСНОВНИ ИЗИСКВАНИЯ ПРИ ПРОЕКТИРАНЕ НА РЕФЕРЕНТНА СОФТУЕРНА АРХИТЕКТУРА НА АДМИНИСТРАТИВНА ИНФОРМАЦИОННА СИСТЕМА, ОРИЕНТИРАНА КЪМ ПРЕДОСТАВЯНЕ НА КОМПЛЕКСНИ АДМИНИСТРАТИВНИ УСЛУГИ

2.1. Информационна сигурност

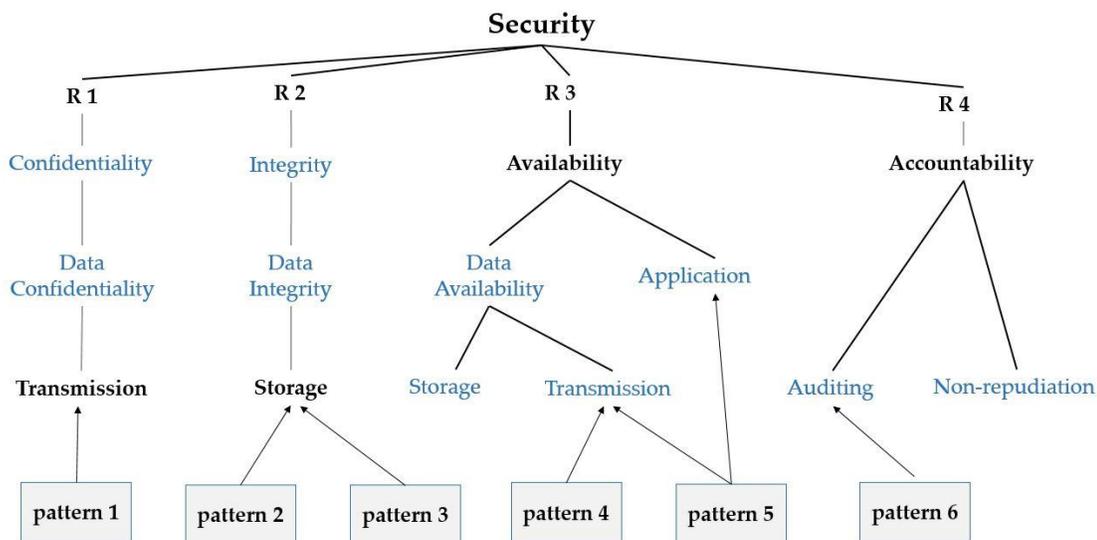
Съществуват няколко дефиниции за информационна сигурност [16], но когато става дума за информационни системи – сигурността може да се определи като свойството на една система да противостои на външни или вътрешни дестабилизиращи фактори, които могат да доведат до нейното нежелателно състояние или поведение или да предоставят нерегламентиран достъп до данните, които тя съхранява. Под сигурност се разбират и мерки, предприети с цел предпазване от шпионаж или саботаж, атака или изтичане на информация. Осигуряването на информационната сигурност е едно от големите предизвикателства към съвременните ИКТ. За решаването на проблемите в това направление е необходим цялостен и задълбочен анализ на всички реални и потенциални заплахи и на въз основа на този анализ трябва да се направи подходящ избор на софтуерна архитектура, която да удовлетвори изискванията за ИС_{инт.}. Според [45] целта на информационната сигурност е защита на ценните ресурси (информация, компютърен хардуер и софтуер) на дадена организация, в частност - администрация. Чрез подбор и прилагане на подходящи предпазни мерки, информационната сигурност подпомага дейността на администрацията чрез предпазване на нейните най-важни ресурси – данните, които тя съхранява и обработва под формата на регистри. Много често данните в тези регистри са такива, че осъществяването на неправомерен достъп до тях би довело да загуба на активи или средства от физически или юридически лица. Според нормативната уредба на Р. България информационните системи, които са внедрени в администрацията трябва да бъдат изградени и да функционират по всички правила и изисквания, заложиени в [6,54]. При внедряване на информационни системи, всеки административен орган трябва да спазва изискванията за оперативна съвместимост и мрежова и информационна сигурност, определени от [8] и от Закона за киберсигурност. Основни изисквания към АИС, са определени в [11].

2.1.1. Количествена оценка на сигурността в софтуерната архитектура

Проблемът с оценката на нивото на сигурност в една информационна система е много актуален. Това се поражда от степента на сложност на самите системи, тъй като много често те на практика са изградени от отделни компоненти, които взаимодействат и обменят данни по между си. Това например е характерно за архитектурите, ориентирани към услуги. В повечето случаи оценката на ИС_{иг} се извършва след като системата е внедрена, което не е правилно. Дисциплината изучаваща софтуерните архитектури е в състояние да противодейства на споменатите фактори с цел определяне на сигурността, която би имала една информационна система, която използва един или друг вид архитектура. Известно е, че софтуерната архитектура е ключов фактор за атрибутите на качеството, които системата ще притежава. В [44] е описан подход за количествена оценка на сигурността на една софтуерна архитектура, който е базиран на принципа на изграждане на дърво, показан на Фиг. 4, което представлява декомпозиция на основните цели, свързани със сигурността.

Системата, използвана за изследването има няколко основни изисквания, свързани със сигурността, а именно:

- R1 – предотвратяване на “изтичане” на данни;
- R2 – предотвратяване на изменения на данните;
- R3 – предоставяне на постоянен достъп до системата на нейните потребители;
- R4 – потребителите да бъдат отговорни за действията, които са извършили.



Фиг. 4 Дървовидна структура на изискванията към сигурността

Всяко от тези изисквания, намиращи се на най-високо ниво, се декомпозира до съответните области, в които е приложимо. Това позволява да бъдат определени подходящи модели (шаблони), свързани със сигурността, които могат да бъдат използвани за постигане на съответните цели. На практика дървото представлява връзка между изискванията (R1, R2, R3, R4), целите и използваните модели. След това се изготвя количествена оценка на сигурността, която отделните модели (шаблони) предоставят. За целта е необходимо да се изгради съответствие между заплахите и поставените цели. Това означава, че за да се измери защитата срещу различни заплахи трябва да се изгради списък от потенциални заплахи. За целта се използва STRIDE – добре известен модел за идентифициране на заплахи в компютърна сигурност, разработен от Microsoft®, който разделя заплахите в шест основни категории, посочени в табл.1. Всяка категория заплаха, на практика нарушава даден атрибут на качеството на системата, свързан със сигурността.

Табл. 1 Категории заплахи в компютърната сигурност

Вид заплаха	Нарушавано качество/свойство на системата
Spoofing	Authenticity
Tampering	Integrity
Repudiation	Non-repudiability
Information disclosure	Confidentiality
Denial of Service	Availability
Elevation of Privilege	Authorization

Всяка заплаха притежава различна степен на тежест. Например, уязвимостите, водещи до някои заплахи, могат да бъдат по-лесни за откриване от други. По същия начин изпълнението на конкретен сценарий на заплаха може да бъде по-скъп от други. Необходимо е да се присвоят тегла на всяка елементарна заплаха, за да се оцени тази разлика. Теглата могат да бъдат определени чрез метод за оценка на риска, който определя сериозността на заплахите. Такъв метод е DREAD [53], разработен от Microsoft®.

Подходът DREAD разглежда пет фактора, допринасящи за сериозността на заплахата или атаката:

- **Щета (Damage)** – определя колко лоша би била една заплаха или атака;
- **Възпроизводимост (Reproducibility)** – колко лесно се възпроизвежда атаката;

- **Възможност за използване (Exploitability)** – колко скъпо е стартирането на атаката от гледна точка на ресурси;
- **Засегнати потребители (Affected users)** – колко потребители или каква част от системата ще бъде засегната;
- **Откриваемост (Discoverability)** – колко лесно се открива заплахата.

Сериозността s_i на конкретна заплата t_i от общ клас заплахи t се изчислява като нормализираната сума на възпроизводимостта R_i , възможността за използване E_i и откриваемостта D_i

$$s_i = (R_i + E_i + D_i) / \sum_{t_i \in t} (R_i + E_i + D_i) \quad (1)$$

Определяне на ниво на защита от заплата – моделите за защита се съпоставят със заплахи, за да се определи нивото на защита, което те предлагат. Това е градивният показател, на който се основава останалата част от подхода. Нивото на защитата се изчислява като предотвратяване на набор от заплахи, свързани с целите на сигурността за чието постигане допринася за модела.

Често един модел осигурява разумна защита срещу заплата, но не я покрива изцяло. В някои случаи моделът може да предложи само частична защита от заплата. Това обикновено се случва, когато основният принос на модела е да изпълни дадена цел, но в същото време указва и страничен ефект върху второстепенна цел.

Нивото на защита на модела срещу заплата се определя по полукачествен начин чрез присвояване една от следните стойности: 0, 0.1, 0.5, 0.9, 1. Смисълът на тези стойности е степента на атаките, които ще бъдат предотвратени, ако се прилагат моделите. Тези стойности означават процент за смекчаване на заплахите, като 0 е „изобщо“ и 1 е „напълно“. Стойностите 0,1 и 0,9 се използват за означаване на странични ефекти на някои модели и невъзможност да се задоволят напълно някои цели (напр. наличност). В случай, че няколко модела допринасят за една и съща заплата, техният принос трябва да се разглежда като вероятност поне един от независимите защитни механизми да е в състояние да защити системата срещу тази заплата, известен още като принципът на „Защита в дълбочина“.

С други думи, ако няколко модела защитават системата срещу една и съща заплата t_i с покритие p_{i1}, p_{i2}, \dots , тогава цялостната защита срещу заплата p_i ще бъде:

$$p_i = 1 - \prod_j (1 - p_{ij}) \quad (2)$$

Изчисляване на общото покритие – индивидуалните стойности на защитата, присвоени на отделните заплахи чрез моделите, могат да бъдат обобщени, за да се определи цялостно покритие за дървото на заплахите. Агрегирането може да се разглежда като претеглена функция. Изчислението разглежда всички приложени модели, поддържащи дадена цел (защита срещу съответната основна заплаха) и създава цялостно покритие за целта. Защитните стойности се умножават по сериозността на заплахите и след това се сумират:

$$c_i = \sum_{i \in I} p_i * s_i \quad (3)$$

Според [36] моделите за сигурност, представени като независими от предметната област експертни знания и опаковани в многократно използваем формат, са в състояние да предложат на софтуерния инженер значителни насоки при разработването на сигурни системи. Въпреки това, изобилието от публикувани модели за сигурност усложнява процеса по намиране на правилния модел за решаване на проблема. Това се дължи на следните три причини:

- първо, не всички модели на сигурност са подходящи за софтуерния инженер;
- второ, независимостта на моделите от предметната област понякога усложнява намирането на решение на специфичен за предметната област проблем.
- трето, моделите съществуват на различни нива на абстракция. Не всички модели могат да бъдат приложени към всяка стъпка в процеса на разработка на система.

2.2. Оперативна съвместимост

Тук ще бъде разгледана оперативната съвместимост (ОС) от нейните технологична и семантична гледна точка. В контекста на е-управление [35] оперативната съвместимост се определя като възможността АИС на отделните администрации да обменят данни по между си или да изпълняват общи административни процеси. За да се постигне това изискване е необходимо още на ниво на проектиране АИС да бъдат заложени такива изисквания, които да гарантират възможността за ОС. Най-често това се свързва с използване на стандарти, технологии и протоколи, които са платформено независими. Според [39] основните препятствия пред ОС са несъвместими технологии и не добре дефинирани структури данни, които водят до семантична несъвместимост. Основните ползи от ОС са оптимизираното управление на данни, както и подобряване на информационната инфраструктура.

Когато става въпрос за ОС [42], една от архитектурните парадигми, която се откроява е SOA – архитектурата, ориентирана към услуги. Това, с което SOA, подпомага ОС е че всеки компонент в тази архитектура, наречен услуга, предоставя програмен интерфейс който може да бъде изпълнен чрез използването на определен конектор. В термините на ОС под конектор се разбира съвкупност от протокол и структура на данните, които потенциалните ползватели на услугата трябва да имплементират. ОС се постига чрез поддръжката на общи протоколи и формати на данните от доставчиците на услуги и техните клиенти. Различията в отделните платформи – операционни системи, програмни езици и софтуерни библиотеки, се преодоляват с използването на общо приети стандарти. Пример за това е представянето на типове данни в различните програмни езици (C++, Java, C#) чрез общ формат, определен от W3C, известен като XSD [58]. SOA е в основата на съвременните архитектурни модели като уеб услуги и микро-услуги.

Европейската рамка за оперативна съвместимост (ЕРОС) [32] определя основните аспекти, свързани технологичната ОС, а именно:

- Добре дефинирани програмни интерфейси;
- Взаимосвързани услуги;
- Интегриране на данни, включващо извличане, трансформиране и последващо използване на данни от други системи;
- Обмен на данни между системи;
- Използване на сигурни комуникационни протоколи.

Според ЕРОС основните проблеми пред ОС, назряват от остарели информационни системи. От историческа гледна точка АИС са разработени използвайки подход отдолу-нагоре, стремейки се да се решат конкретни локални проблеми, без да се отчита необходимост от бъдеща съвместна работа с други ИС. В резултат на това са създадени информационни острови, които са останали невзаимосвързани, а постигането на ОС би било трудна задача. Именно всичко казано до тук е доказателство, че правилния избор на архитектурен модел и подход при проектирането на една система определя бъдещите ѝ възможности за постигане на ОС. Препоръката дадена в ЕРОС е използването на отворени спецификации при проектирането на ИС_{ис}, което означава, че трябва да се използват общоприети стандарти, като по този начин се намалява рискът от определени технологични зависимости, които са в противовес с ОС.

Семантичната оперативна съвместимост гарантира, че точният формат и значение на обменяните данни и информация се запазват по време на обмена, т.е. данните се интерпретират от двете страни по еднакъв начин или с други думи „това, което се

изпраща, е това, което се разбира“. В ЕРОС семантичната оперативна съвместимост обхваща както семантични, така и синтактични аспекти.

Семантичният аспект се отнася до значението на отделните елементи в обменяните данни и връзката между тях. Той включва разработване на речници и схеми за описване на обменяните данни и гарантира, че елементите им се разбират по един и същи начин от всички комуникиращи страни;

Синтактичният аспект се отнася до описването на точния формат на информацията, която трябва да се обменя, по отношение на използваната граматика. Граматиката представлява съвкупност от лексеми (служебни думи) и правила за описания на структурата на обменяните данни.

Основната препоръка, дадена от ЕРОС, за насърчаване на семантичната ОС е данните и информацията да бъдат възприети като актив на обществото, който трябва да бъде грижливо създаден, съхраняван, управляван и споделян [49].

2.3. Нормативни фактори

Нормативните фактори включват различни закони и наредби, които определят изисквания към информационните системи, свързани с тяхната работа в няколко насоки:

- Структура и начин на обработване на данните;
- Информационна сигурност (ИС_{ит});
- Оперативна съвместимост (ОС).

Докато последните две са стандартни нефункционални изисквания, които до определена степен зависят от избрания архитектурен модел, то структурата на данните е функционално изискване, което не зависи от модела. Въпреки това трябва да се има в предвид, че честата промяна на законите и наредбите води до множество изменения в структурата на съхраняваните и обработвани данни. Ето защо е добре да се изгради такава архитектура, която да позволява висока степен на гъвкавост по отношение на изменчивостта на данните и работните процеси в информационните системи.

2.4. Разширяемост

В софтуерното инженерство разширяемостта се определя като качеството на една система да бъде така проектирана, че да позволява добавянето на нови възможности или функционалности. Това е мярка за способността за развитие на системата и нивото на усилие, необходимо за извършване на това развитие.

Идеалните характеристики на една разширяема софтуерна система са гъвкавост, настойваемост, възможност за персонализиране на процеси и възможност за надграждане:

2.4.1. Гъвкавост

Гъвкавостта е мярка за способността за бързо адаптиране към нарастващите нужди и променящите се изисквания. Всяка софтуерната платформа трябва да бъде достатъчно гъвкава, за да поддържа изменения в изпълняваните от нея работните процеси чрез разширяване на съществуващата функционалност.

2.4.2. Настройваемост

Това е характеристика, която гарантира, че една система ще може да променя поведението си, без да е необходимо изменение в програмния ѝ код. Системите, които могат да променят поведението си без промяна на кода им или с много малка такава, дават възможност на по-широка аудитория – бизнес анализатори, експерти в дадена област, без наличието на програмисти – за бързо изграждане и конфигуриране на персонализирани процеси.

2.4.3. Възможност за персонализиране на процеси

Предизвикателството да се промени модела сложни процеси с помощта на изменение в конфигурацията е огромно. Една система трябва да може лесно да се персонализира чрез специализиран код, така че разработчиците и програмистите да могат да реализират различни сценарии на употреба. Това води до повишаване на качеството на системата и удовлетвореността на нейните потребители.

2.4.5. Възможност за надграждане

Една системата трябва да може да бъде лесно изменима, така че да ползва най-актуалните технологии без това да влияе върху вече изградените работни процеси. Възможността за постоянно надграждане на системата води до това, че тя никога няма да се превърне в “остарял” програмен продукт.

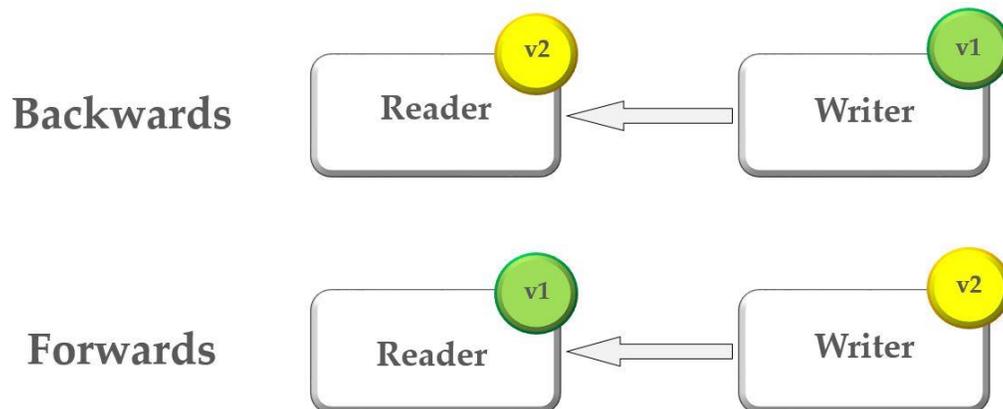
Разширяемостта на една система може да се обхваща следните аспекти:

- разширяване на модел на данни – съществуващият модел на данни се разширява с нови типове данни, включително потребителски типове данни;
- разширява на процеси – от съществено значение е да се автоматизират ежедневните, повтарящи се задачи, за да се позволи на крайния потребител да се съсредоточи върху задачата с висока стойност и да се

минимизира възможността за грешки. Процесите трябва да бъдат силно адаптивни и поддържани чрез инструменти, които специалисти, различни от програмисти, могат лесно да използват.

2.5. Съвместимост (съвместимост напред и обратна съвместимост)

Съвместимостта е важна характеристика на софтуерните продукти, особено когато те са изградени от множество модули, предоставени от различни доставчици, което често е характерно за ИС_{ис} на администрациите. За разлика от оперативната съвместимост (interoperability), съвместимостта (compatibility) се определя като възможността модулите на една ИС_{ис} да работят безпроблемно, дори и версията на някои от тях да се промени. Под промяна на версия се разбира изменение в предоставения от модул или система програмен интерфейс, а именно промяна на предоставяните за изпълнение операции или структурата на обменяните данни. Съществуват два вида съвместимост, наречени съвместимост напред и обратна съвместимост, които са представени на Фиг. 5.



Фиг. 5 Видове съвместимост

Пример за двата вида съвместимост може да се представи чрез два програмни модула. Първият от тях притежава ролята на “генератор” на данни, а вторият използва тези данни, т.е. е “консуматор”. Когато модулът “консуматор” притежава по-нова версия на предоставения програмен интерфейс или на структурата от данни, която консумира, но все още може да обменя успешно данни с по-стара версия на модула “генератор”, тогава се говори за обратна съвместимост, т.е. по-новата версия на консуматора е обратно съвместима с по-старата версия на генератора. Когато текущата версия на

модула “консуматор” е съвместима с по-нова версия на модула “генератор” на данни, се говори за съвместимост напред.

Ако разгледаме типичната Клиент-сървър архитектура [48] обратната съвместимост е важна защото:

- могат да бъдат добавени входни параметри на операциите, изпълнявани от сървъра, без да е необходимо да се променя клиента;
- за типа на връщаните данни – може да бъде надграден клиентът, без да е необходимо да се измени сървъра.

Примери за обратна съвместимост са следните изменения в структурата на обменяните данни:

- добавяне на поле със стойност по подразбиране. По-старите генератори на данни няма да знаят за това поле, така че вместо него ще се използва стойността по подразбиране;
- добавяне на незадължително поле. По-старите генератори няма да знаят за това поле, така че вместо него ще се използва празна стойност (null).
- разширяване на числов тип (например надграждане на поле от тип целочислен до тип с плаваща запетая). По-старите генератори винаги ще използват целочисления тип, чийто стойности са подмножество от стойностите на типа с плаваща запетая;
- добавяне на нова стойност към изброен тип. По-старите генератори ще използват една от съществуващите изброени стойности, за които те знаят.
- премахване на поле. По-новите консуматори ще пренебрегнат всичко, което е написано в премахнатото поле.

Съвместимостта напред от гледна точка на архитектурата Клиент-сървър е важна защото:

- в случай на необходимост от надграждане на входните параметри, клиентите могат да бъдат промени без да е необходимо да се прави изменение по сървърния модул
- за типа на връщаните данни – сървърът може да бъде надграден, без да е нужно изменение в клиента.

Примери за съвместимост напред са следните изменения в структурата на обменяните данни:

- добавяне на ново задължително поле. В този случай по-новите консуматори ще игнорират неговата стойност;

- ограничаване на числов тип (например от тип с плаваща запетая до целочислен тип). В този случай по-новите консуматори ще приемат целочислената стойност, която е подмножество на стойност с плаваща запетая;
- премахване на стойност от изброен тип – по-старите консуматори могат да работят и с типа, който е с по-ограничени стойности.

Ако едно изменение е както съвместимо напред, така и обратно съвместимо, то то се нарича пълна съвместимост. Това означава, че могат да бъдат използвани всякакви комбинации от по-стари или по-нови модули “генератори” и “консуматори”. Пример за такъв вид съвместимости от гледна точка на структурата на обменяните данни са:

- добавяне на поле със стойност по подразбиране;
- добавяне на незадължително поле.

2.6. Надеждност

Надеждността е една от ключовите характеристики, която засяга както потребителите на една система, така и нейните разработчици. Тя силно зависи от софтуерната архитектура и затова изборът на правилна архитектура за конкретен вид ИС_{ис} води до добра или посредствена надеждност. Прогнозирането на надеждността, описано в [38], на един софтуерен продукт е сложно поради взаимозависимостта и взаимодействията между компонентите и трудността това поведение да се представи чрез добре дефинирани модели. Обикновено моделите, които са разработени чрез опростяване на предположенията за софтуерната структура са лесни за използване, но получените от тях резултат е далеч от това, което се случва в действителност. Обратно, доближаването на предположенията до реалността, което позволява описание на сложни взаимодействия и взаимозависимости, води до модели, които са твърде сложни за използване. Получените резултати могат да бъдат сложни за интерпретация.

Проблемът, свързан с прогнозирането на надеждността се влошава от липсата на точна информация за поведението на компонентите на една ИС_{ис}, както и за техните взаимодействия, а това е информация, която е от значение за моделирането на надеждността. Обикновено взаимодействията не са напълно известни поради малки недокументирани странични ефекти. Без точна и прецизна информация, дори и правилни математически модели няма да дадат точни прогнози за надеждност. От друга страна, извличането на необходимата информация от вече завършен продукт, тоест от изходния му код, е не само непрактично, но дори и невъзможно. Това е така, защото кодът съдържа прекалено много подробности за изпълнението, за да бъде

полезен за създаването на проследим модел. Поради тази причина е необходимо да се изгради оптимален подход, който да може да оцени каква ще бъде надеждността на една ИС_{ис} още в процеса на проектиране на нейната архитектура. Според [26,32,40], надеждността се определя като вероятността една ИС_{ис} да изпълнява своите функции или да предоставя определени услуги правилно (безотказно) за определено време в определена среда. Отказ на системата може да бъде дефиниран като неправилен резултат или неочаквано поведение, което нарушава нейните функционални или нефункционални изисквания. В зависимост от жизнения цикъл на измерване на надеждността, мерки могат да бъдат предприети рано в процеса на разработка или по-късно в тестването или след внедряването [34]. Основната цел на моделите за надеждност, създавани в етапа на проектиране, е да се предскаже надеждността на ИС_{ис}, свързана със софтуерната архитектура [35]. Ключови потенциални предимства, които могат да бъдат получени чрез включване на артефакти на архитектурния дизайн, като входни данни при изграждането на модела за надеждност, са възможността за оценка и сравнение на избраните архитектурни решения и при необходимост – възможност за тяхното ревизиране. Като допълнение, по този начин се определя чувствителността на надеждността на ИС_{ис} от надеждността на отделните ѝ компоненти. Такива модели подпомагат за по-доброто разбиране на изискванията за надеждност и как те се отнасят към архитектурния дизайн. Моделите за оценка на надеждността, използвани в ранните етапи на проектиране могат да бъдат разделени на два основни типа – сиви кутии (gray-box) [43] и бели кутии (white-box) [41]. При gray-box моделите, системната надеждност се предсказва на базата на надеждността на всеки един от изграждащите я компоненти, без да се познава тяхното вътрешно поведение. Оценката на надеждността на отделните компоненти се извършва чрез техники от тип черна кутия (black-box). За разлика от gray-box моделите, при white-box моделите, за оценка на надеждността на отделните компоненти се използва информация от техния архитектурен дизайн, който описва вътрешното им поведение. [26,32,40] предлагат осъвременен модел за оценка на надеждността на комплексни ИС_{ис}, изградени от множество отделни компоненти, който е базиран на изграждане на сценарии, използвайки специализиран език за тяхното описание [21]. Авторите представят разработения от тях модел като gray-box, базиран с отчитане характера на изпълняваните операции (последователни или паралелни) и представят различни подходи за определяне на крайната надеждност.

2.7. Възможност за поддръжка

Възможността за поддръжка на комплексна ИС_{ис} е от съществено значение, особено когато тя се състои от множество компоненти и в същото време има многократни внедрявания. [43] определя възможността за поддръжка като *“Лекотата с която една ИС или нейн компонент могат да бъдат променяни с цел коригиране на грешки, подобряване на производителността или друг атрибут на качеството, както и адаптиране към промени в средата”*. Също така тя е многостранно изискване за качество, тъй като често се свързва с възможността за тестване, тъй като след изменение на ИС_{ис} е необходимо нейната функционалност да бъде валидирана отново. В [25] е представен модел, в който възможността за поддръжка притежава следните седем свойства – модулност, четимост, програмен език, стандартизация, ниво на валидация и тестване, комплексност и проследимост. Моделът може да бъде използван за да подчертае нуждата от подобряване на качеството на един софтуерен продукт, така че той да се характеризира с добра възможност за поддръжка.

Гореизброените свойства могат да бъдат описани по следния начин:

- **Модулност** – степента на декомпозиция на ИС_{ис} на малки функционални единици (модули), изградени йерархично на базата на функционалността, която предоставят. Модулността има за цел да осигури както желаната степен на абстрактност, така и възможността за самостоятелна разработка и поддръжка на отделните компоненти на една ИС_{ис};
- **Четимост** – степента, до която външно лице може бързо да се ориентира и да разбере изходния код на модул или система;
- **Програмен език** – определя адекватността на избора на програмния език за разработка на ИС_{ис} за съответната предметна област;
- **Стандартизация** – степента на използване на стандарти за създаване на изходен код с цел избягване на самостоятелни нестандартни решения от страна на разработчиците;
- **Ниво на валидация и тестване** – отношението на вложените усилия по валидация на дизайна и тестване към цялостното усилие по разработка;
- **Комплексност** – определя как сложността на решавания проблем се отразил на създадените софтуерни артефакти (модули);
- **Проследимост** – отразява възможността да се проследи дадено решение взето по време на проектиране обратно до изискванията, които са го породили, както и обратния процес.

От гореописаните свойства, характеризиращи възможността за поддръжка, става ясно, че те могат да бъдат обобщени в следните три категории:

- Структурни – модулност и комплексност;
- Свързани с процеса на разработка – програмен език, стандартизация, ниво на валидация и тестване и проследимост;
- Четимост – асоциира се с човешките възможности и със стандартизацията;

Според тази категоризация може да се направи извод, че малка част от степента на възможността за поддръжка на една ИС_{ис} се определя в процеса на проектиране на нейната архитектура и единствения фактор, който може да бъде отчетен е модулността. От тук може да се направи извод, че при проектиране на софтуерна архитектура на комплексна ИС_{ис} от съществено значение е и изборът на правилна гранулярност на проектираните модули. Това ще доведе до по-лесна поддръжка на системата по време на нейната експлоатация.

2.8. Възможност за тестване и проверка

Възможността за тестване на комплексна ИС_{ис}, състояща се от множество модули и компоненти е от съществено значение. Тук не точно изборът на конкретен архитектурен шаблон е важен, а по-скоро проектирането на самата архитектура, включващо декомпозиране на компонентите на системата на отделни модули. Правилния подход в процеса на декомпозиция и спазването на добре познатите принципи при проектиране ще доведе до възможност за качествено и лесно тестване на готовия продукт. Пример за такъв принцип е Single responsibility (принцип за единична отговорност), в който е заложена идеята, че един структурен елемент (независимо дали е клас, модул или компонент) трябва да има точно една определена отговорност. Спазвайки този принцип драстично се намаляват и причините за корекции в този структурен елемент. На практика тук важи голяма част от казаното за възможността за поддръжка, тъй като както тя, така и тестването и валидацията са част от последния етап от жизнения цикъл на един софтуерен продукт, време за което е характерно, че етапите по проектиране и дизайн са завършили отдавна.

Изводи

Основните нефункционални изисквания, които софтуерната архитектура на АИС, ориентирана към предоставяне на КАО трябва да гарантира са:

- **Информационна сигурност** на съхраняваните и обменяни данни

- **Оперативна съвместимост**, без която обмена на данни между ИС_{ис} на администрациите е невъзможен.
- **Възможност за разширяемост на АИС**, свързана с чести промени в нормативна уредба и породени промени в административните процеси
- **Съвместимост**, гарантираща правилната работа на АИС при надграждане на част от нейните модули или системи, с които тя е интегрирана.

ГЛАВА 3. АРХИТЕКТУРНИ МОДЕЛИ

Архитектурните модели (АМ, архитектурни шаблони) подпомагат определянето на основните характеристики и поведение на една ИС_{ис} [33]. Някои АМ осигуряват по естествен път възможност за създаването на лесно скалируеми системи, докато други предоставят възможност създаване за по-гъвкави приложения. Поради тази причина, познаването на характерните особености, силните и слабите страни на всеки АМ, е от съществено значение за правилния избор на модел, който ще осигури изпълнение на специфични изисквания и цели на една ИС_{ис}. Според [29] архитектурата на една ИС_{ис} почти никога не се ограничава до един единствен АМ. Сложните системи често притежават архитектура, в която са вплетени идеи от няколко АМ. Типичен пример са уеб базирани системи, които по същество погледнати от високо ниво на абстракция реализират клиент-сървър архитектура, но “вътре” в този модел е реализирана Model-View-Controller (MVC) архитектура, а преминавайки на по-детайлно ниво се вижда, че е използван и принципът за разпределяне на функционалността в слоеве. В процеса на проектиране, софтуерният архитект често избира един или повече АМ за да създаде първоначалната структура на системата. Архитектите избират АМ, базирайки се на възможността на всеки от шаблоните да осигури по-голяма част от изискванията, поставени пред ИС_{ис}. Избраните модели възплъщават структурата на високо ниво. Въпреки това е възможно избраните АМ, в техния стандартен вид, да не могат да осигурят изпълнение на пълния набор от изисквания. За да бъдат изпълнени и останалите изисквания, свързани с качествените характеристики на ИС_{ис}, които не са адресирани от избраните АМ, и за да се премахне нуждата от компромиси, могат да бъдат реализирани различни архитектурни стратегии. Те се имплементират на по-високо архитектурно ниво, като целта е да осигурят локално определена качествена характеристика.

Стратегиите са специални видове операционализации (измерване на явление, което не е пряко измеримо), които служат за пресечна точка между атрибутите на качеството и софтуерната архитектура. Според [27] архитектурната стратегия е един вид архитектурна трансформация, която засяга модела, чрез който избраната архитектура осигурява съответния атрибут на качество. В [24,51] авторите правят заключение, че структурата и поведението на стратегията е по-скоро локално, т.е. на по-ниско ниво от АМ и следователно е лесно стратегията да бъде приложена към по-общата структура. Имплементирането на определена стратегия в рамките на даден АМ може да доведе до неговата промяна чрез изменение в неговите компоненти, премахване на част от тях

или добавяне на нови. Това може да доведе до там, че крайната структура на архитектурата да се различава от първоначално избраната. Най-често използваните архитектурни модели, използвани при проектиране на ИС_{ис}, с изисквания близки до тези на АИС, са:

- **Клиент-сървър** – един от първите архитектурни модели, който сам по себе си не може да бъде използван за проектиране на цялостна система и не може да гарантира изисквания, свързани с нейните качества. Този модел предоставя общи насоки при реализацията на дадена ИС_{ис};
- **Многослойна архитектура** – подход, който разделя проектираната ИС_{ис} на отделни независими слоеве, йерархично разположени един над друг;
- **Микроядро** – разделя проектираната ИС_{ис} на два вида компоненти – ядро на системата, което реализира основна базова функционалност и плагин модули, които реализират специализирани функционалности.
- **Архитектура, ориентирана към услуги** [42] – тук се реализира идеята за предоставяне на изпълнимите програмни модули на ИС_{ис} като услуги;
- **Микроуслуги** – съвкупност от малки приложения (услуги), които се комбинират за да се изгради ИС_{ис}.

3.1. Архитектурен модел Клиент-сървър

При този вид архитектура, клиентите (приложенията, които се нуждаят от услуга) и сървърите (приложенията, които предоставят изискваните услуги) са самостоятелни логически обекти, които комуникират посредством мрежа с цел да изпълняват задачи заедно. Клиентът прави заявка за услуга и получава отговор на тази заявка. Сървърът получава и обработва заявката и изпраща обратно необходимия отговор на клиента [50].

Основни характеристики на Клиент-сървър АМ са:

- **асиметрични протоколи** – съществува отношение едно към много между сървъра и клиентите. Винаги клиентите инициират комуникация със сървърите чрез изпращане на заявка за услуга. Сървърите изчакват пасивно тези заявки;
- **капсулация на услугите** – подобно на принципа на капсулацията в Обектно ориентираното програмиране (ООП), сървърът скрива от клиента детайлите по изпълнение на изисканата от него услуга. По този начин сървърите могат да бъдат надградени, без това да повлияе на

клиентите, спазвайки условието определения за комуникация интерфейс да не бъде променян;

- **прозрачност по отношение на местоположението на сървъра** – за клиентите не е от съществено значение къде се намира сървърът, стига той да е физически достъпен. На практика сървърът може да е разположен на същата хардуерна платформа където се намира и клиента или да е отдалечен, но достъпен през някакъв вид мрежа. В частния случай едно приложение може да бъде както клиент, така и сървър;
- **ниска степен на обвързаност** между клиентите и сървърите, които са напълно самостоятелни процеси, комуникиращи помежду си чрез обмен на съобщения;
- **модулен/разширяем дизайн** – модулният дизайн на клиент-сървър системите позволява те да бъдат устойчиви на откази, което означава, че отказ на един клиент или сървър няма да доведе до неработоспособност на цялата система. Модулният дизайн определя също и възможността за лесно разширение или скалиране при необходимост чрез увеличаване на броя на сървърите предоставящи дадена услуга;
- **платформена независимост** – при Клиент-сървър АМ, клиентът и сървърът могат да използват различни хардуерни и софтуерни платформи. Също така, отделните клиенти могат да използват различни операционни системи, оптимизирани с цел постигане на максимална ефективност;
- **повторно използване на кода** – сървърните приложения могат да бъдат инсталирани на множество сървъри;
- **скалируемост** – клиент-сървърните системи могат да бъдат скалирани както хоризонтално така и вертикално. При хоризонталното скалиране се добавят нови клиенти с малко влияние върху производителността. При вертикалното скалиране се добавят по-високопроизводителни сървърни конфигурации или се увеличава техният брой;
- **разделяне и разпределяне** на функционалните характеристики на клиентите и сървърите – клиент-сървър е отношение между процеси, които се изпълняват на една и съща или различни хардуерни системи, като всеки от тях има своите функционалности;
- **споделяне на ресурси** – един сървър може да предоставя услуги на много клиенти в даден момент и да управлява достъпът до споделени ресурси.

Класически пример за клиент-сървър архитектура е системата браузър – уеб сайт. Множество клиенти (уеб браузъри) могат да изпратят заявка за отваряне на уеб съдържание до даден сървър. Сървърът обработва получените заявки от всички клиенти (браузъри) и те получават заявеното уеб съдържание.

3.1.1. Анализ на предимства и недостатъци на архитектурата Клиент – сървър

Общата гъвкавост, която определя възможността за бърза реакция при постоянно променяща се среда, се определя като висока. Това е породено от факта, че клиентите и сървърите не са взаимосвързани и са платформено независими.

Внедряване – опростен процес, тъй като измененията в сървъра могат да бъдат скрити за клиентите.

Възможност за тестване – определя се като добър АМ по отношение на възможността за тестване, тъй като клиентите и сървърите могат да бъдат тествани самостоятелно.

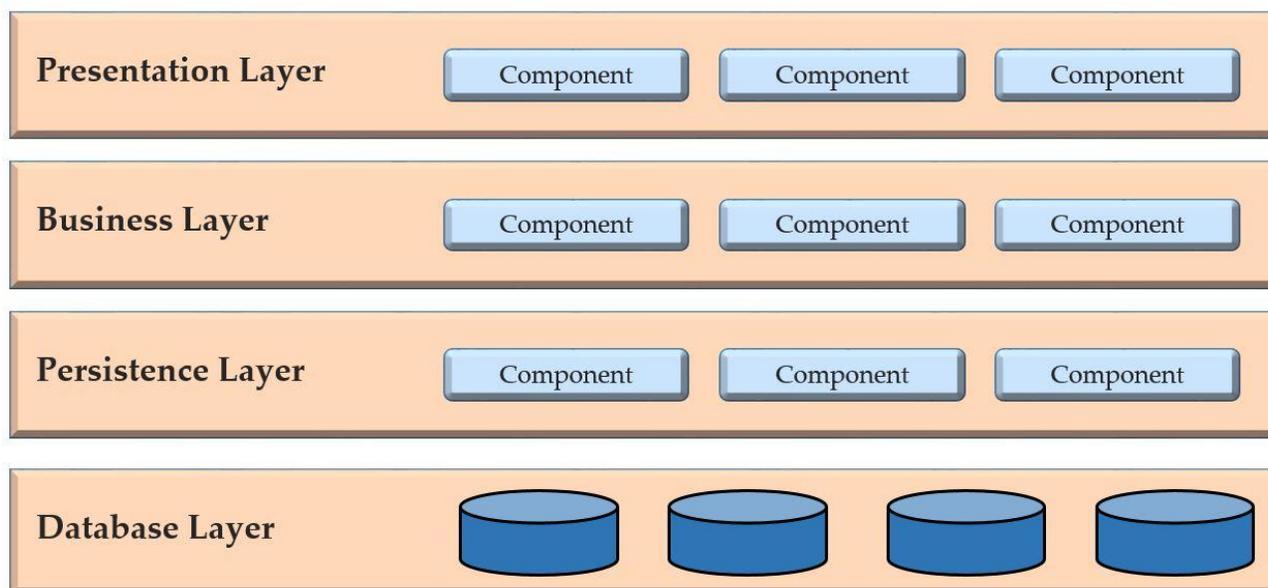
Производителност – този АМ се характеризира с добра производителност, тъй като системите могат лесно да бъдат скалирани. Проблем в производителността може да бъде налице, когато комуникацията между клиентите и сървърите бъде нарушена.

Мащабируемост – този АМ се характеризира с относително висока възможност за мащабируемост.

Разработка – този АМ предоставя възможност за лесна разработка, тъй като сървър и клиент компонентите могат да бъдат разработвани и тествани самостоятелно.

3.2. Многослойна архитектура

В този АМ, показан на фиг. 6, архитектурните компоненти са организирани в хоризонтални слоеве, като всеки слой изпълнява определена специфична роля в приложението. Въпреки че този модел не определя броя и вида на слоевете, които трябва да бъдат включени в архитектурата на ИС_{ис}, повечето многослойни архитектури включват следните слоеве: презентационен, бизнес, слой на данните и база данни. Понякога бизнес слоя и слоя на данните се обединяват, като в този случай логиката за достъп да данните се включва в бизнес слоя. Оттук по-малките приложения се характеризират с трислойна архитектура, а по-мащабните с четири или повече слоя. Всеки слой има точно определени роли и отговорности в рамките на приложението – например презентационния слой е отговорен за предоставяне на потребителски интерфейс и логиката, свързана с него, докато бизнес слоя изпълнява специфични бизнес правила, определени от предметната област, за която е създадено приложението.

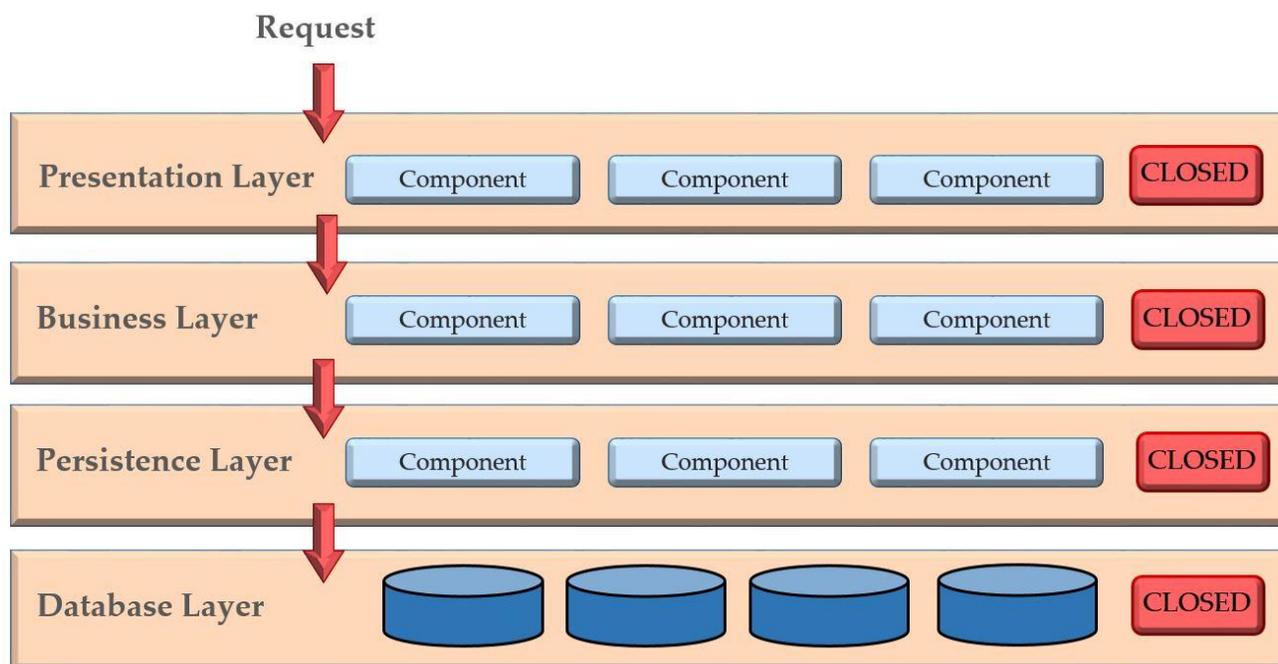


Фиг. 6 Многослойна архитектура

Основното предимство на многослойната архитектура е разделяне на отговорностите между отделните слоеве. Всеки слой използва слоя или слоевете под него без да се интересува от реалната имплементация. Това дава възможност да се постигне както ниско ниво на обвързаност между компонентите от отделните слоеве, така и лесна подмяна на цялостен слой. Например – слой за достъп до данните може да бъде променен, така че да използва друга Object Relational Mapper (ORM) технология или друг механизъм за достъп до данните. От друга страна, при необходимост от промяна на слоя с базата данни (използване на друга система за управление на релационна база данни) в най-критичната ситуация ще се наложи да се направят промени само по слоя за достъп до данни (persistence), тъй като единствено той комуникира директно с базата данни [33].

3.2.1. Основни концепции при многослойните архитектури

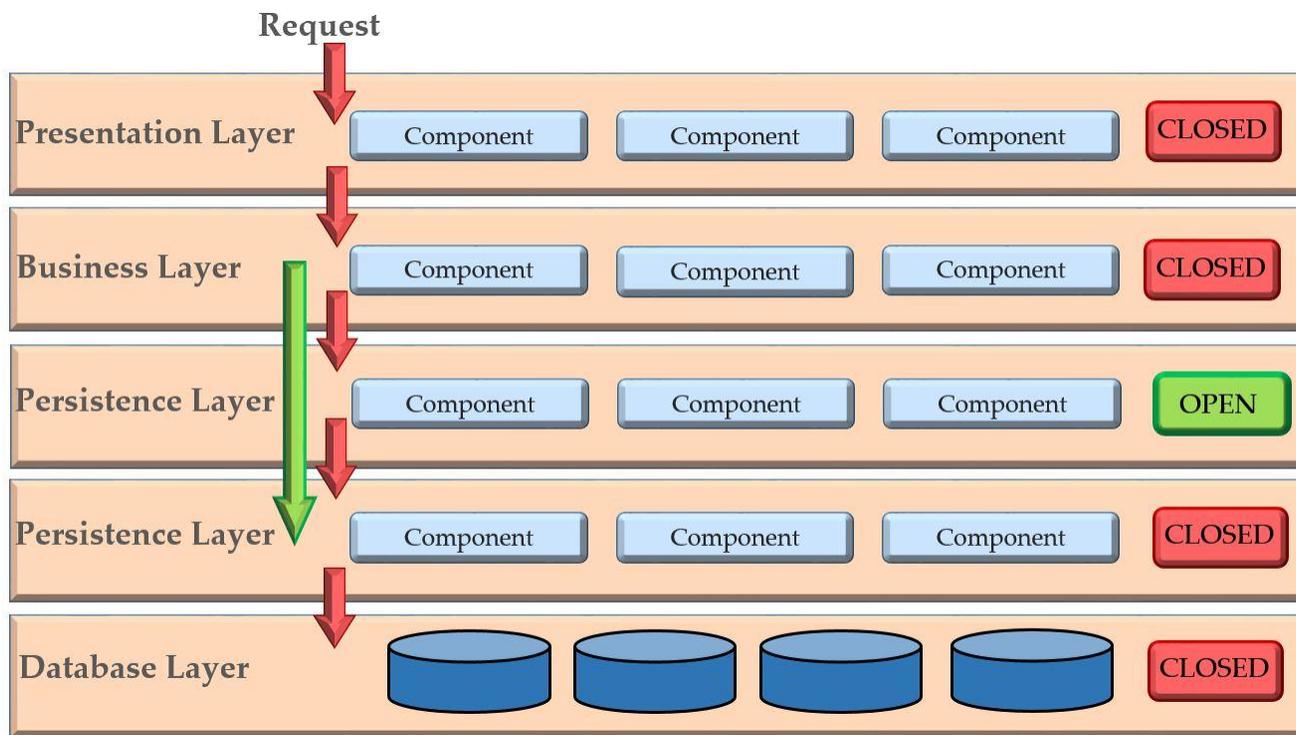
На Фиг.7 е представена концепцията за “затворени” слоеве. Всеки един от тях е маркиран като “затворен”, което означава, че при обработка на заявка, възникнала в по-горен слой, тя се обработва последователно без да се прескачат слоеве. Например, при натискане на бутон в презентационния слой, той се обръща към бизнес слоя, той от своя страна към слоя за достъп до данните, а той към базата данни. Този принцип налага концептуалния модел за изолираност на слоевете. Чрез него се елиминира или драстично намаля броя на промените, които е нужно да се имплементират в останалите слоеве, когато се извършва изменение в един слой.



Фиг. 7 Затворени слоеве

Въпреки че концепцията за затворените слоеве води до положителни ефекти в реализираната система, понякога има смисъл определени слоеве да са “отворени”, както е показано на Фиг. 8. Създаден е нов слой, наречен слой от услуги, който предоставя определена функционалност на бизнес слоя. Това, че той е “отворен” дава възможност на слоя, съдържащ бизнес логиката да има пряк достъп и до слоя с данните. По този начин бизнес слоя може да използва слоя за услугите при определени обстоятелства, а при други да осъществи директен достъп до слоя за данни.

Правилното документирание на слоевете – включително това дали са “отворени” или “затворени”, причините за това и зависимостите между тях, е от съществено значение, тъй като в противен случай проектираната система ще се характеризира със силно-обвързани компоненти, които са много трудни за тестване, поддръжка и внедряване



Фиг. 8 Отворени слоеве

3.2.2. Анализ на предимства и недостатъци на многослойната архитектура

Общата гъвкавост, която определя възможността за бърза реакция при постоянно променяща се среда, се определя като ниска. Това е породено от факта, че често приложенията, реализиращи този АМ са монолитни с тясно обвързани компоненти. Когато “изолацията” между слоевете не е достатъчна, т.е. променя се програмния интерфейс между слоевете, имплементирането на промени се усложнява.

Внедряване – в зависимост от начина на имплементация на този АМ, внедряването може да стане сложен процес, особено при монолитни системи, тъй като промяна в един малък компонент може да доведе до нужда от внедряване на нова версия на цялата система.

Възможност за тестване – определя се като добър АМ по отношение на възможността за тестване, тъй като компонентите на системата принадлежат на определени слоеве, а останалите слоеве могат да бъдат симулирани (mock).

Производителност – този АМ се характеризира с не особено добра производителност, тъй като изпълнението на една заявка води до това, че се изпълнят определени операции от всеки един слой

Мащабируемост – този АМ се характеризира с относително ниска възможност за мащабируемост, тъй като приложенията, реализираща такъв тип архитектура са монолитни. Възможно е да се извърши скалиране чрез разделяне на отделните слоеве на отделни внедрявания или чрез репликация на цялото приложение.

Разработка – този АМ предоставя възможност за лесна разработка, тъй като компонентите от отделните слоеве могат да бъдат разработвани самостоятелно. Също така отделните слоеве могат да бъдат разработвани от различен вид специалисти – проектантите на потребителски интерфейс, проектантите на базата данни и проектантите на бизнес процесите [33].

3.3. Архитектура, ориентирана към услуги

Този АМ представя развитие на Клиент-сървър архитектурата, представяйки идеята за предоставяне на изпълними програмни модули като услуги. SOA е архитектура, която позволява взаимодействие между различни организации (в това число и администрации) посредством споделяне на ресурси с цел постигане на желаната функционалност, която се предоставя на т.нар. консуматори на услуги [43]. OASIS (Organization for the Advancement of Structured Information Standards) определя SOA архитектурата като парадигма за организиране и използване на разпределени изчислителни ресурси, които могат да бъдат под управлението на различни собственици. Тя осигурява унифицирани средства за предлагане, откриване, взаимодействие и използване на програмни компоненти за постигане на желани ефекти, съответстващи на измерими предпоставки и очаквания [54]. На практика всички технологии за реализация на уеб услуги, които се използват днес, независимо дали са SOAP базирани, или просто предоставят приложен програмен интерфейс (API) посредством REST подхода, могат да бъдат разглеждани като специализирани имплементации на архитектура, ориентирана към услуги.

Основните характеристики на архитектурата, ориентирана към услуги са систематизирани от [22] и могат да бъдат обобщени по следния начин:

- **Откриваемост и динамично свързване** – SOA представя концепцията за откриване на необходимите услуги. Това означава, че когато ползвателят на услуга (клиентът в термина на клиент-сървър архитектурата) установи какви са изискванията му към дадена услуга, то той посредством регистър на услугите, може да открие най-подходящата за него услуга и динамично да консумира нейната функционалност;
- **Самодостатъчност и модулност** – услугите поддържат набор от интерфейси за достъп до тях, които се имплементират от един или

няколко програмни модули. При проектирането на тези интерфейси трябва да се вземе под внимание необходимата степен на гранулярност на изпълняваните операции, така че чрез комбинирано обръщане към отделни услуги да може да бъде изградено пълноценно софтуерно решение. [58] определя няколко критерия, които трябва да бъдат взети под внимание при проектирането на софтуерни компоненти. Те определят дали една услуга се характеризира правилната степен на модулност:

- ❖ Модулна **декомпозируемост** – отнася се до разделяне на приложението на по-малки модули. Всеки модул е отговорен за една единствена функционалност на приложението. Този подход е известен като проектиране *“отгоре-надолу”*. Целта е при проектирането на услугата да се идентифицират най-малките програмни единици, които могат да бъдат преизползвани в различни сценарии на употреба;
- ❖ Модулна **композируемост** – проектантите на услуги трябва да създават услуги, които са достатъчно независими, за да се използват повторно в напълно различни приложения от тези, за които първоначално са били предназначени. Този подход е известен като проектиране *“отдолу-нагоре”*
- ❖ **Яснота** на проектираните модули – способността на човек да разбере функцията на услугата, без да има никакви познания за други услуги;
- ❖ Модулна **непрекъсваемост** – отнася се до въздействието на промяна в една услуга, изискваща промяна в други услуги или в потребителите на услугата. Интерфейс, който не скрива достатъчно детайлите на изпълнението на услугата, създава ефект на доминото, когато са необходими промени;
- ❖ Модулна **защита** – модулната защита на дадена услуга е достатъчна, ако необичайно състояние в услугата не се предава каскадно към други услуги или потребители. Грешките в работата на дадена услуга не трябва да оказват влияние върху работата на клиент или друга услуга или състоянието на техните вътрешни данни или по друг начин да нарушават договора с потребителите на услугата.
- **Оперативна съвместимост** – основна характеристика на SOA, която предоставя възможност програмни системи, използващи различни платформи, езици и среди за разработка да взаимодействат помежду си.

За целта всяка услуга предоставя интерфейс, който е достъпен чрез конектор, който представлява съвкупност от протокол и формат на данните. Всеки клиент на услугата, който желае да я достъпи и използва, е необходимо да имплементира съответния протокол и да подаде необходимите данни в правилния формат;

- **Слаба обвързаност** – SOA предоставя ниско ниво на зависимост между потребителите и доставчиците на услуги. Всички зависимости са предварително известни и добре дефинирани;
- **Местоположението на услугата** – потребителите на услуги не се интересуват от това, къде е разположена конкретна услуга. Нещо повече, те не знаят нейния адрес, до момента на откриване на услугата в регистъра и динамичното ѝ извикване;
- **Отказоустойчивост** – при системите, използващи SOA, общата отказоустойчивост се определя от отказоустойчивостта на отделните услуги и на комуникационната среда, използвана за комуникация между клиента и тях.

3.3.1. Анализ на предимства и недостатъци на архитектурата, ориентирана към услуги

Общата гъвкавост, която определя възможността за бърза реакция при постоянно променяща се среда, се определя като относително добра. Това е породено от факта, че в повечето случаи промените по услугите, не водят до промени в клиентите. Ако се налага изменение в услуга, което би довело до промяна в нейните консуматори, добра практика е да се създаде нова услуга или нова нейна версия, която да предоставя изменената функционалност.

Внедряване – процеса по внедряване е улеснен, тъй като клиентите на услугите могат да достъпват техните функционалности, независимо от физическото им разположение. Те могат да бъдат на същия сървър, в същата локална мрежа или достъпът да се осъществява през глобална мрежа.

Възможност за тестване – процесът по тестване на SOA система от една страна е усложнен, тъй като за успешното му изпълнение е необходимо отделните услуги да бъдат имплементирани. От друга страна тестването на отделните услуги е улеснено, тъй като при подходящо имплементирана модулност, всяка услуга предоставя конкретна функционалност. Съществуват програмни инструменти, които улесняват и двата процеса.

Производителност – този АМ може да гарантира достатъчна производителност на приложението, но като цяло не може да предостави свръхпроизводителност, поради разпределения характер на отделните услуги.

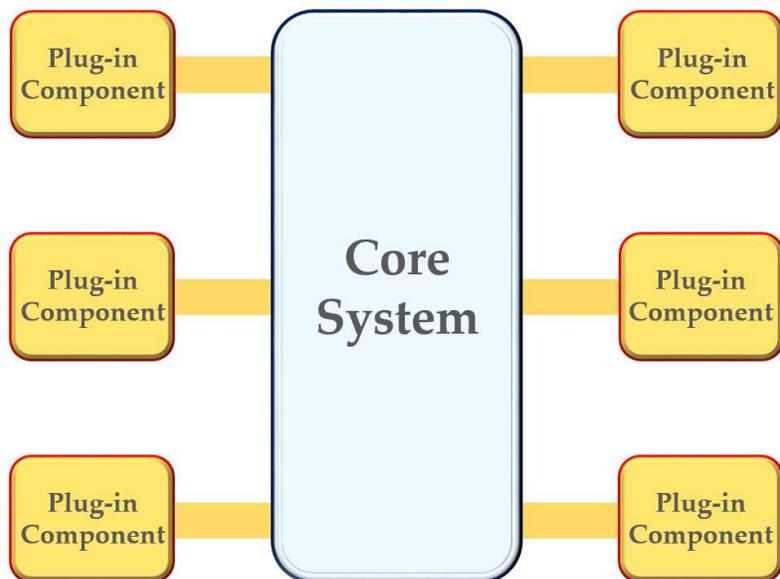
Скарлируемост – този АМ се характеризира с много добра възможност за мащабируемост, тъй като системите, които го имплементират са изградени от самостоятелни компоненти - услуги, всеки от които може да бъде самостоятелно скалируем.

Разработка – при този АМ отделните компоненти могат да бъдат разработвани самостоятелно, защото всеки от тях предоставя изолирана функционалност. Това улеснява процесите по разработка. На практика, при изграждане на система, чиято архитектура е SOA, често се преизползват вече разработени услуги, което допълнително улеснява разработчиците.

3.4. Архитектурен модел Микро-ядро

Микро-ядро е АМ, който е ориентиран към софтуерни решения, които са продуктово-базирани, т.е. такива които крайния потребител може да внедри сам. Въпреки това, много компании, които разработват ИС_{ис} избират този АМ, тъй като той дава възможност за лесно добавяне на нови функционални възможности под формата на така наречените плъгини. Това са допълнителни модули чрез които базовата функционалност на една ИС_{ис} се разширява.

Софтуерните архитектури, които имплементират този АМ се състоят от два вида компоненти – системно ядро и допълнителни плъгин модули. Структурата на АМ е представена на Фиг.9.



Фиг. 9 Архитектурен модел Микро-ядро

Логиката на приложението е разделена между независимите плъгин модули и неговото ядро. Този подход предоставя много добри възможности за разширяемост, гъвкавост и разграничение на основната функционалност на приложението от специализираната обработка, извършвана в допълнителните модули. Традиционно, ядрото съдържа минимална функционалност, нужна за да бъде системата работоспособна. От гледна точка на ИС_{ис}, които изпълняват различни бизнес процеси, характерно за АИС, ядрото имплементира базовата бизнес логика, а плъгин модулите извършват специализирани бизнес процеси. Едно от предимствата на АМ Микро-ядро е, че той може да бъде вграден или използван като част от друг АМ. В процеса на проектиране на големи ИС_{ис}, софтуерният архитект може да използва този модел за да удовлетвори конкретно изискване, но много често не може да имплементира цялата архитектура използвайки го. В този случай АМ Микро-ядро се вгражда в друг модел – например многослойната архитектура.

Чрез подходящо прилагане на модела Микро-ядро могат да бъдат изградени ИС, които еволюират чрез инкрементална разработка, без да са необходими сериозни промени в основното ядро.

3.4.1. Анализ на предимства и недостатъци на архитектура Микро-ядро

Общата гъвкавост, която определя възможността за бърза реакция при постоянно променяща се среда, се определя като висока. Това е породено от факта, че в повечето случаи промените са изолирани в допълнителните плъгин модули, които са слабо обвързани с основното ядро. Като цяло в процеса на разработка на ИС_{ис}, системното

ядро става стабилно в първите итерации и след това не са нужни големи изменения в него.

Внедряване – в зависимост от начина на имплементация на този АМ, допълнителните плъгин модули могат да бъдат добавяни динамично по време на работа на ИС_{ис}, което като цяло улеснява процеса по внедряване.

Възможност за тестване – определя се като много добър АМ по отношение на възможността за тестване, тъй като плъгин модулите на системата могат да бъдат тествани изолирани един от друг или пък да бъдат симулирани от ядрото.

Производителност – този АМ сам по себе си не може да гарантира наличието или отсъствие на добра производителност. Въпреки това, приложенията, които използват такъв вид архитектура се характеризира с добра производителност, тъй като ненужните функционалности могат лесно да бъдат премахнати с деактивиране на допълнителните плъгин модули.

Мащабируемост – този АМ се характеризира с относително ниска възможност за мащабируемост, тъй като приложенията, реализираща такъв тип архитектура са малки и монолитни. Възможно е да се извърши скалиране на ниво плъгин модули, но като цяло този модел не е подходящ, ако мащабируемостта е ключово изискване.

Разработка – при този АМ основното затруднение при разработка се обуславя от необходимостта от създаване на спецификация за интеграция между ядрото и отделните плъгин модули. Тази спецификация задължително трябва да бъде поддържана с версия, особено при необходимост от чести промени на ядрото.

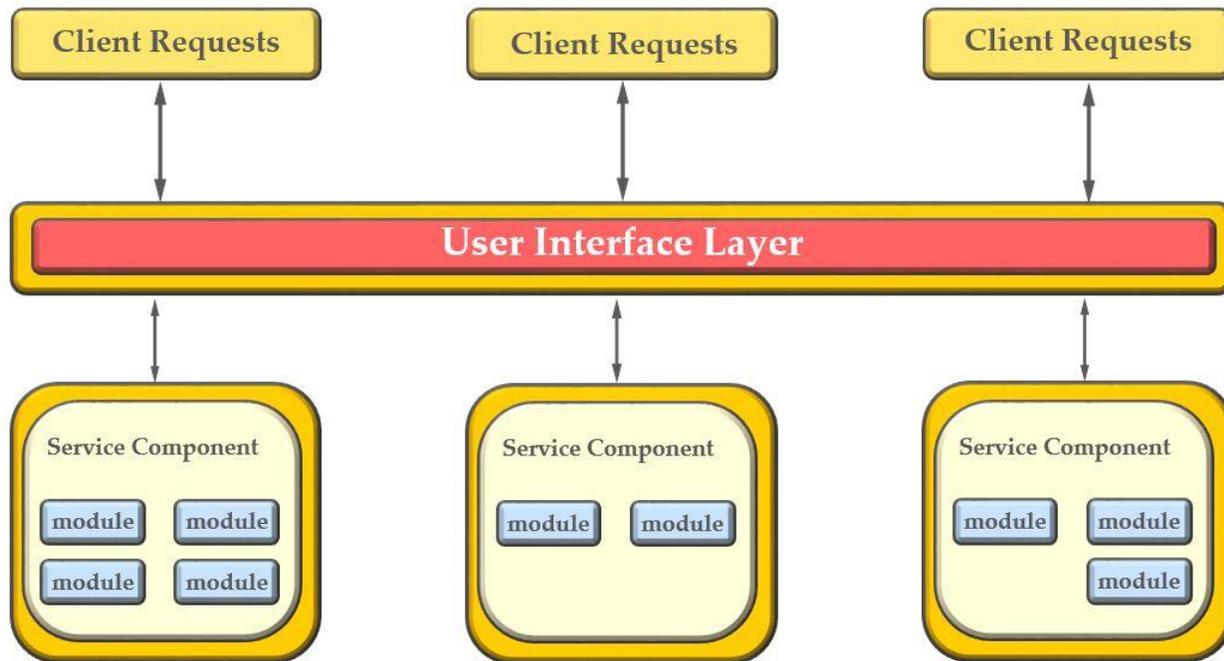
АМ Микро-ядро е приложим при разработка на ИС_{ис}, използвани в администрацията, тъй като той предоставя възможности за изграждане на специфични модули, които се използват от една администрация, а от друга – не. За целта, ядрото на АИС предоставя базова функционалност която се използва от допълнителните плъгин модули, които на практика реализират различните процесите в различните администрации. Ако основното ядро на системата предоставя базовата функционалност, която се характеризира с незначителни разлики в отделните администрации, то в ядрото биха могли да се дефинират т.нар. точки на разширение, в които да се извика функционалност от допълнителен плъгин модул за съответната администрация. По този начин спецификите в административните процеси могат да бъдат проектирани в допълнителните модули [33].

3.5. Архитектурен модел Микро-услуги

Микро-услуги е АМ, който печели бързо популярност като ценна алтернатива на монолитните и сервизно-ориентирани архитектури. Характерно за него е, че все още е развиващ се модел, поради което в софтуерната индустрия има спорове за това как трябва да се прилага.

Независимо от избраната топология или начин на реализация, съществуват няколко ключови идеи, характерни за този по-общ архитектурен стил:

- Идеята за **самостоятелно внедрени програмни единици** – както е изобразено на фигурата по-долу, всеки компонент в АМ Микро-услуги може да се разглежда като самостоятелна програмна единица, което улеснява процеса по внедряване, увеличава възможността за скалируемост, постигайки намаляване на взаимозависимост между отделните модули;
- Идеята за **компоненти от услуги** – отделните компоненти предоставят една единствена функционалност (когато гранулярността на компонентите е по-силно изразена) или относително независима, но все пак голяма част от функционалността на голямо бизнес приложение (при слабо изразена гранулярност на компонентите). Проектирането на правилното ниво на гранулярност на отделните компоненти е от съществено значение за успеха на разработката;
- Идеята за **разпределеност** – тази идея се свързва с разпределеното местоположение на отделните компоненти от архитектурата, което означава, че те са напълно необвързани и независими и се достъпват (използват) чрез протоколи за отдалечен достъп като REST, SOAP, RMI и други. Тази характеристика е предпоставка за лесно постигане на висока степен на скалируемост на проектираната ИС_{ис}.



Фиг. 10 Базова архитектура на модела Микро-услуги

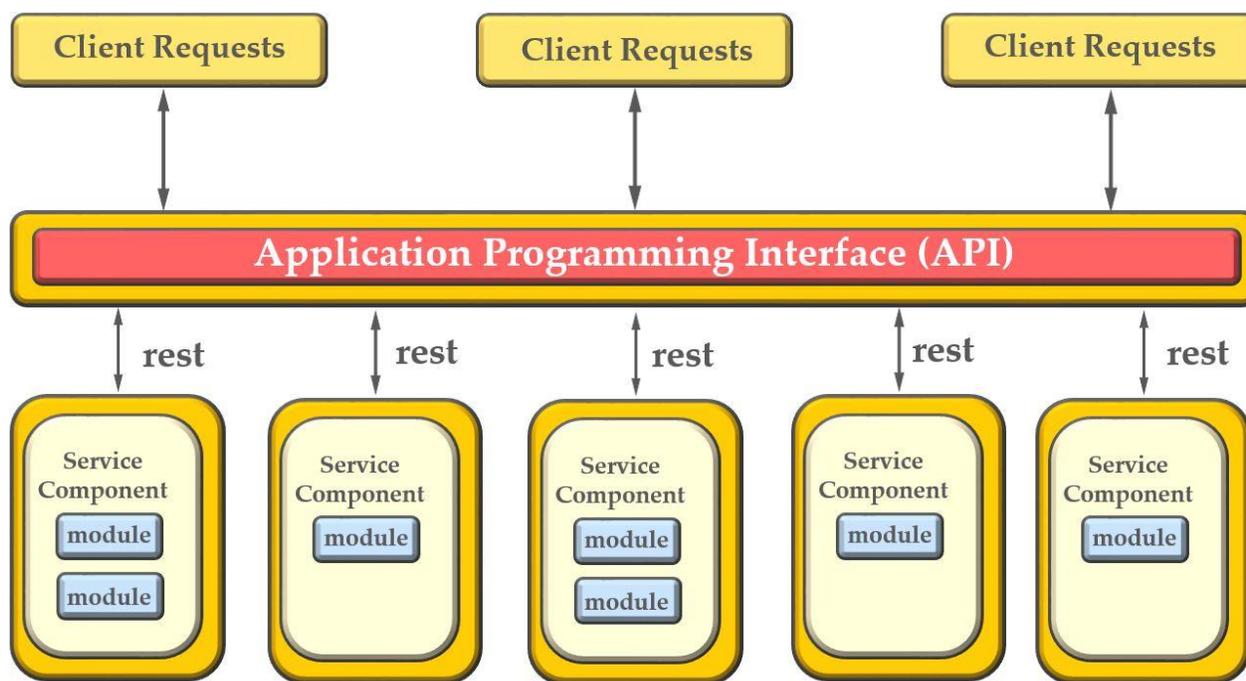
AM Микро-услуги, показан на Фиг.10, се е развил като еволюция на други модели като многослойния и ориентирания към услуги в стремежа за развитие на конвейера (pipeline) на разработка с цел постигане постоянна доставка на усъвършенствани продукти. Монолитните приложения, характерни за многослойния AM са изградени от силно взаимнообвързани компоненти, които са трудни за развитие, тестване и често се внедряват като една самостоятелна единица. Архитектурата на модела Микро-услуги адресира тези проблеми чрез разделяне на големите ИС_{ис} на множество самостоятелни внедрими единици – това са отделните компоненти услуги, които са много по-ефективни от гледна точка на изменения, тестване и разпространение. Друга еволюция, която този модел прокламира е свързана с проблемите, които могат да бъдат открити в архитектурите, ориентирани към услуги. Въпреки че те предоставят неизмерими възможности за моделиране на абстракции, хетерогенна свързаност, оркестрация на услуги и гарантират изпълнение на поставените бизнес цели, обикновено прилагането им е прекалено сложен, скъп и труден за разбиране процес, който обикновено е безсмислен за повечето ИС_{ис}. AM Микро-услуги адресира и тези проблеми чрез опростяване на идеята за услуга, елиминира нуждата от оркестрация и улеснява свързаността и достъпа до отделните компоненти.

3.5.1. Топологии при архитектура Микро-услуги

Въпреки, че съществуват множество подходи за имплементация на този АМ, три основни топологии са се наложили като най-често използвани:

- API REST базирана топология;
- Приложна (application) REST базирана топология;
- Топология с централизиран обмен на съобщения;

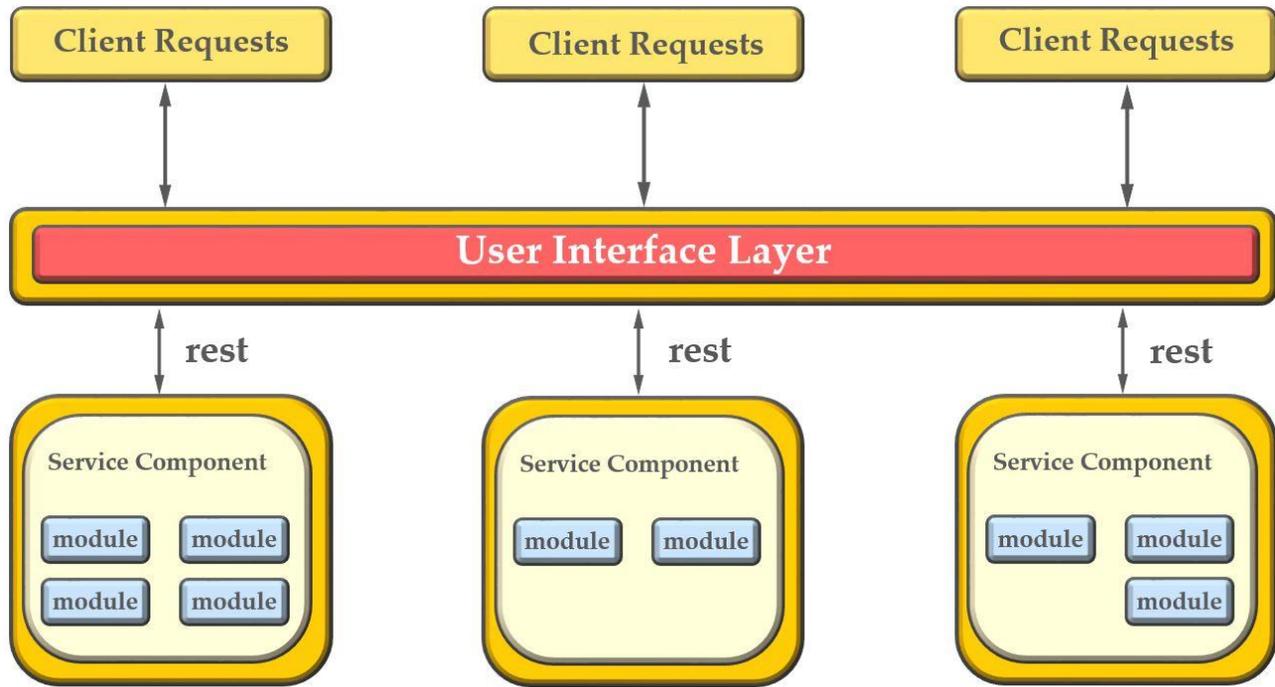
API REST базираната топология е показана на Фиг. 11. Характерно за нея е, че отделните компоненти са с висока степен на гранулярност – голям брой компоненти, изградени от малък на брой модули, които изпълняват специфична функционалност, откъдето произхожда името на архитектурния модел. Отделните компоненти са достъпни чрез REST базиран интерфейс, имплементиран като отделен уеб-базиран API слой.



Фиг. 11 REST API базирана топология

Приложната REST базирана топология се различава от API REST подхода по това, че заявките към компонентите се генерират от стандартен десктоп или уеб базиран потребителски интерфейс. Както е показано на Фиг. 12, характерно за този подход, е че слойът на потребителския интерфейс е отделно уеб приложение, което посредством REST базиран интерфейс извиква отделните компоненти от архитектурата. Друга особеност е, че тези компоненти се характеризират с по-ниска степен на гранулярност

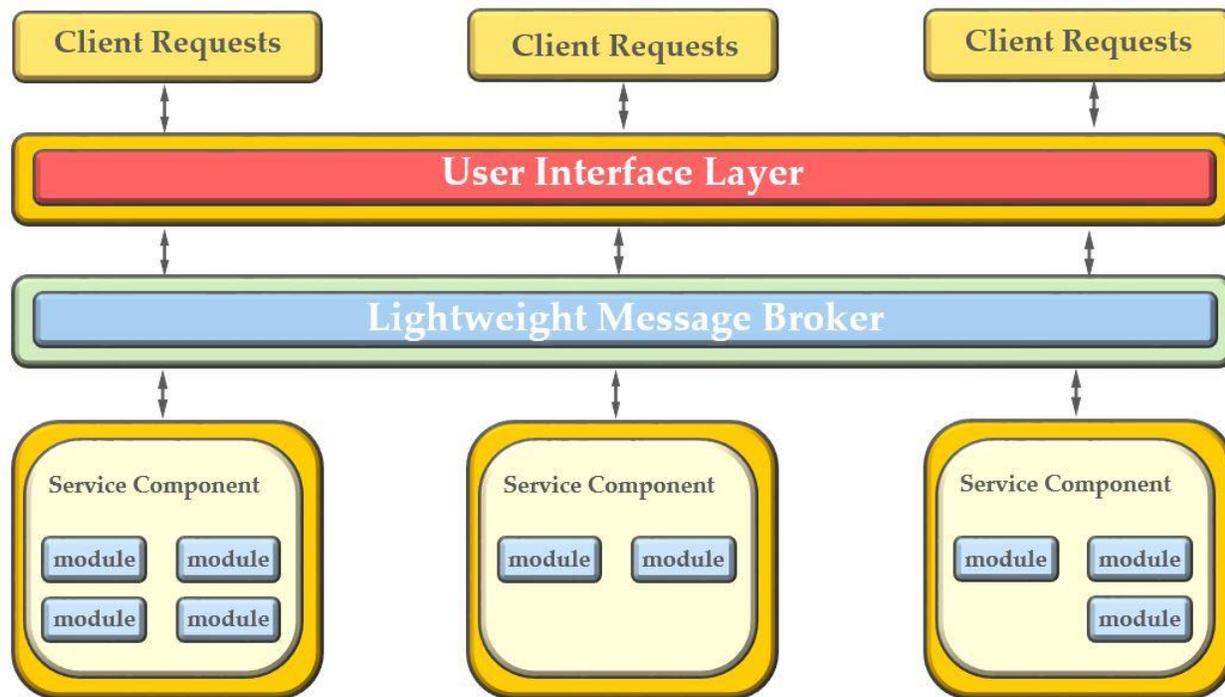
(т.е. те са по-малко на брой, но включват в себе си по-голям брой модули) и имплементират по-съществена част от функционалността на цялата система. Тази топология е подходяща за малки и средни бизнес приложения, които имат относително ниска степен на сложност.



Фиг. 12 Приложна REST базирана топология

Друг често използван подход при изграждане на архитектура, базирана на Микро-услуги, е използването на топология с централизиран обмен на съобщения, показана на Фиг. 13. Тя се различава от предходните топологии, по това че използва централизиран брокер на съобщения (ActiveMQ, HornetQ) вместо REST услуги. Тази топология не трябва да се бърка със стандартната SOA, а по-скоро може да се разглежда като опростена архитектура. Брокерът на съобщенията не извършва сложни действия като оркестрация, трансформация или маршрутизиране. Този подход за имплементиране на AM Микро-услуги е характерен за големи бизнес приложения и приложения, изискващи по-усложнен контрол между потребителския интерфейс и компонентите, реализиращи бизнес логиката.

Предимствата на този подход са – разширени механизми за управление на опашки от заявки, асинхронни съобщения, мониторинг, разширена обработка на грешки и по-добра възможност за скалируемост и управление на натоварването. Единственият проблемен елемент от гледна точка работоспособност и производителност би бил централизираният брокер. Този проблем може да се реши чрез използване на клъстер от брокери на съобщения.



Фиг. 13 Топология с централизиран обмен на съобщенията

3.5.2. Анализ на предимства и недостатъци на архитектурата Микро-услуги

Общата гъвкавост, която определя възможността за бърза реакция при постоянно променяща се среда, се определя като висока. Това е породено от факта, че в повечето случаи промените са изолирани в отделните компоненти за услуги, което позволява бързо и лесно внедряване на нови версии. Приложенията, реализиращи тази архитектура се характеризират с ниска степен на обвързаност на отделните модули.

Внедряване – внедряването на нови версии на компонентите е лесно, заради високата степен на гранулярност и независимостта на отделните компоненти, които се разглеждат като самостоятелно единици.

Възможност за тестване – определя се като много добър АМ по отношение на възможността за тестване, заради изолираността и разделянето на бизнес функционалността. Тестването може да бъде ограничено в рамките на един компонент за разлика от монолитните приложения.

Производителност – този АМ може да гарантира достатъчна производителност на приложението, но като цяло не може да предостави свръхпроизводителност, поради разпределения характер на отделните компоненти.

Скалируемост – този АМ се характеризира с много добра възможност за мащабируемост, тъй като системите, които го имплементират се състоят от множество самостоятелни компоненти, всеки от които може да бъде самостоятелно скалируем.

Разработка – при този АМ отделните компоненти могат да бъдат разработвани самостоятелно, защото всеки от тях предоставя изолирана функционалност. Това улеснява процесите по разработка [33].

Изводи

Не може да бъде определено еднозначно един единствен АМ, който да бъде използван при създаването на софтуерна архитектура на АИС, ориентирана към КАО. Отделните АМ предоставят ефективни решения на различни проблеми, адресирани при проектиране на софтуерна архитектура на АИС.

ГЛАВА 4. АРХИТЕКТУРА НА АДМИНИСТРАТИВНА ИНФОРМАЦИОННА СИСТЕМА, ОРИЕНТИРАНА КЪМ ПРЕДОСТАВЯНЕ НА КОМПЛЕКСНИ АДМИНИСТРАТИВНИ УСЛУГИ

При проектиране на архитектура на АИС, ориентирана към предоставяне на КАО, трябва да се отчетат множеството дейности, които тази ИС_{ис} ще изпълнява. Част от тези дейности могат да се променят в бъдеще в следствие на множество фактори като изменение в нормативна уредба, оптимизиране на работни процеси, развитие на хоризонталните системи за е-управление, навлизането на нови технологии и др. Оттук следва, че АИС, ориентирана към КАО трябва да притежава добре структурирана архитектура, която се характеризира с много добра разширяемост, от гледна точка на добавяне и изменения на функционалности. Правилния подход без съмнение е използването на хибридна софтуерна архитектура, която да се възползва от положителните качества на различните архитектурни модели. Примерна такава архитектура би заимствала идеи от модели като Микро-ядро, Микро-услуги, многослойна архитектура. По този начин изградената система би имала основна функционалност, определена от нейното програмно ядро, а реализирането на специфична функционалност може да се извършва чрез добавяне на допълнителни модули към ядрото. Тези допълнителни модули могат да бъдат на практика консуматори на различни услуги, като по този начин се постигат положителните страни на архитектурите, ориентирани към услуги. Архитектурата на отделните услуги “спазва” препоръките предоставени от многослойните архитектури. Така АИС, ориентирана към КАО се състои от множество модули, всеки от които може да разполага със собствена микро-архитектура. Цялостната архитектура е хибридна, взимайки най-доброто от отделните видове архитектурни модели, спазвайки основните изисквания, отредени от всеки от тях.

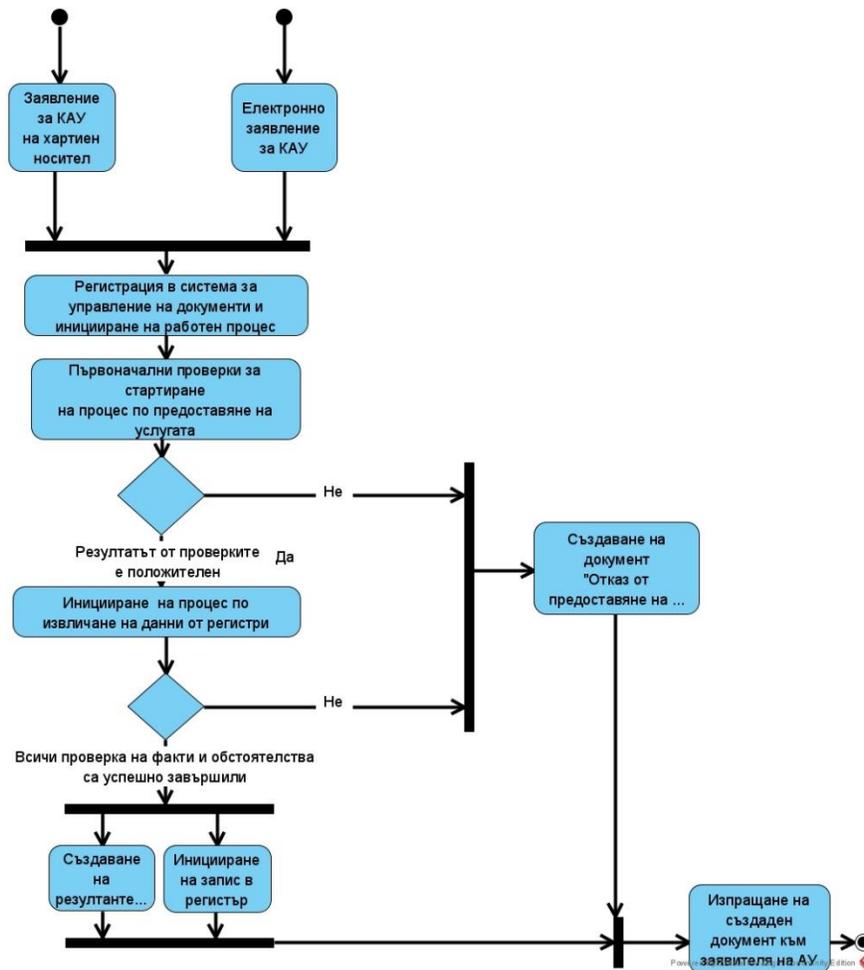
4.1 Унифициран модел на процес за предоставяне на комплексни административни услуги

За да се направи анализ за модулите, които трябва да бъдат включени в архитектурата на АИС за КАО, изключвайки стандартните такива като управление на потребители, роли и права и други характерни за всяка ИС_{ис}, е необходимо да се опишат основните

етапи от процеса по предоставяне на КАУ. Такъв процес включва следните етапи, за всеки от които най-често е отговорен един или няколко модула от архитектурата:

- Заявяване на КАУ;
- Регистрация на документ - заявление за КАУ в АИС на администрацията
- Обработка на заявление за КАУ, включващо:
 - Анализ на предоставените в заявлението данни;
 - Извличане на всички необходими за изпълнението на услугата данни и документи от други системи, най-често регистрови ИС;
 - Анализ на извлечените данни;
- Изготвяне на резултантен документ;
- Запис на данни и обстоятелства в регистри от данни;
- Регистриране и предоставяне на резултантния документ на заявителя на услугата.

Диаграмата на Фиг. 14 представя отделните етапи на процеса по предоставяне на КАУ.



Фиг. 14 Етапи на процеса по предоставяне на КАУ

4.1.1 Заявяване на комплексна административна услуга

Процесът за предоставяне на КАУ започва с попълване на всички необходими за изпълнението на услугата данни в специализирана бланка за заявяване, наречена заявление. Заявлението може да бъде попълнено на хартиен носител или в електронна форма посредством уеб или мобилно приложение. Вторият електронен подход се реализира чрез съвременни уеб технологии, чрез които се идентифицира заявителя на услугата, посредством съвременни средства за автентикация като квалифициран електронен подпис, облачен електронен подпис или централизираната системата “Е-Автентикация”, част от е-Управление. Процесът по електронно заявяване преминава през няколко етапа, през които заявителя попълва данни, прилага документи, данните, в които не могат да бъдат извлечени към момента от регистрирани системи (напр. пълномощно), извършва заплащане на такси за предоставяне на услугата. Когато процесът завърши успешно, всички данни се изпращат към АИС на администрацията, изпълняваща услугата. Ако заявлението се попълва на хартиен носител, то това се извършва на място в администрацията и попълнената бланка се сканира, след което се регистрира директно от служител в АИС посредством следващия етап.

4.1.2 Регистрация на документ – заявление за комплексна административна услуга

Процесът за предоставяне на КАО продължава с регистрацията на заявление за КАУ в АИС на администрацията, независимо дали е попълнено на хартиен носител или в електронна форма посредством уеб или мобилно приложение. Разликата между двата подхода е, че когато заявлението е в електронна форма, в него се съдържат машинно четими структурирани данни, които ще наричаме *“острови от данни”*. Терминът “остров от данни” ще бъде разгледан по-подробно на по-късен етап в дисертационния труд в точка 4.2.4 Примерна реализация на Модул за електронно удостоверение. След регистрацията на заявлението, АИС съхранява всички данни, подадени от заявителя, независимо дали са структурирани или неструктурирани файлове. Като последно действие от този етап, АИС инициира работен процес, чрез който се извършват всички дейности по последващата обработка на заявлението.

4.1.2. Обработка на заявление за комплексна административна услуга

Това е най-сложният етап от процеса по изпълнение на КАУ, който изисква най-много време за изпълнение, особено когато администрацията, изпълняваща КАО не разполага с подходящата функционалност в своята АИС. За изпълнението на този етап в АИС се конфигурира работен процес, специализиран за конкретно заявената КАУ. Всеки работен процес се състои от последователност от стъпки, а във всяка стъпка са

дефинирани едно или няколко действия, които трябва да бъдат изпълнени. Тези действия могат да бъдат:

- Задача към конкретен служител - към определен служител се насочва задача, която той трябва да изпълни в определен срок. Тези задачи най-често са свързани с действия, които не могат да бъдат автоматизирани – например когато заявлението е подадено чрез хартиен носител, част от информацията в него трябва да бъде въведена като структурирани данни в АИС. Така въведените данни ще могат да бъдат използвани при процеса на автоматизирано извличане на данни от регистрите на ПАД.
- Инициране на процес по извличане на данни от регистри системи на ПАД – за изпълнението на всяка КАУ е необходимо да бъдат извлечени данни, доказващи факти и обстоятелства, които заявителя не е доказал, чрез предоставяне на оригинални документи, издадени от други администрации. Този процес се изпълнява от специализиран модул на АИС, който осъществява обмен на регистри данни със ИС на е-Управление и ИС на ПАД. Процесът по извличане на данни е динамичен, което означава, че е възможно конкретните данни, извлечени до момента да определят какви данни от кой ПАД ще бъдат изискани допълнително. Ако дадена операция за извличане на данни от регистър на ПАД е неуспешна се преминава към формиране и изпращане на неструктуриран документ – заявление за ВЕАУ.
- Инициране на процес по анализ и проверка на извлечени от регистри системи данни – след приключване на процеса по извличане на данни от регистрите на ПАД, а понякога и по време на неговото изпълнение е необходимо, вече наличните структурирани данни да бъдат анализирани чрез проверка на стойности на конкретни полета от тяхната структура. Такива проверки могат да определят дали дадено поле има точно определена изискуема стойност, дали стойността е в определен интервал (използва се за полета от тип дата),или дали стойността е равна на стойността на друго поле от вече изтеглените данни. За реализацията на тези проверки се използват логически изрази, на които се оценява стойността. На практика тези логически изрази представляват условия, които определят дали администрацията има право да предостави заявената КАУ.
- Инициране на процес за изпращане на документ по електронен път към друга администрация – когато процеса по извличане на данни от регистрите на ПАД е неуспешен поради различни обстоятелства, (липса

на свързаност, липса на запис в регистър или липса на регистър като цяло) администрацията предоставяща КАУ подготвя неструктуриран документ, заявление за ВЕАУ, който се изпраща по електронен път до ПАД. ПАД е задължен да предостави услугата в определен срок като изпрати изискания документ по електронен път. След предоставяне на изискания неструктуриран документ, служител от администрацията, изпълняваща КАУ попълва липсващата структура от данни, използвайки информацията от неструктурирания документ, за да може процесът по извличане и анализ на данни от други ПАД да продължи.

- Инициране на процес по изпращане на документ към заявителя на услугата – в определени ситуации, в процеса на предоставяне на КАУ, заявителя се известява за необходимост от извършване на допълнителни действия като заплащане на такса или предоставяне на документ, който не може да бъде осигурен по вътрешно административен път.

4.1.3 Изготвяне на резултантен документ

В зависимост от това дали проверките, извършени върху структурираните данни са завършили успешно, т.е. всички условия за предоставяне на КАУ са изпълнени, АИС подготвя неструктуриран документ, който представлява резултата от изпълнението на заявената КАУ. Документът може да бъде от различен вид – удостоверение, лиценз, разрешение и др. Неговото съдържание се създава чрез предварително конфигурирана в работния процес бланка, която съдържа текст, който не се променя и специални маркери, чиято стойност са заменя с конкретни стойности от извлечените структурирани данни. Аналогично, ако условията за предоставяне на КАУ не са изпълнени, АИС създава друг вид документ, който представлява мотивиран отказ за изпълнение на КАУ. Използва се същия подход с предварително конфигурирана бланка.

4.1.4. Запис на данни и обстоятелства в регистри от данни

Ако е изготвен резултантен документ (удостоверение, лиценз, разрешение) вследствие на това, че услугата е предоставена, и АИС поддържа регистър, в който документът и обстоятелства и фактите, описани в него трябва да бъдат включени, то тогава посредством изпълнявания работен процес, АИС иницира действия за записване на структурирани данни от документа в този регистър. Възможно е регистърът, в който трябва да се запишат фактите и обстоятелствата да не под контрола на изпълняващата КАУ администрация. В този случай АИС известява съответния регистър за

настъпилите промени. За целта е необходимо този регистър да предоставя програмен интерфейс за изпълнението на тази операция.

4.1.5 Регистриране и предоставяне на резултантния документ на заявителя на услугата

Създаденият резултантен документ, независимо от това дали услугата е предоставена или нейното изпълнение е отказано, е необходимо да бъде регистриран в официалния документен регистър на администрацията. Този етап завършва с изпращане на създадения и регистриран документ към заявителя на услугата посредством централизирана система на е-Управление за обмен на документ с физически и юридически лица.

Примерна реализация на модул за автоматизирано създаване на резултантни документи е представена в точка 4.2.4, където са дефинирани понятията “остров от данни” и “трансферна схема”

4.2 Модули, изграждащи архитектурата на административната информационна система

Както се вижда от диаграмата на Фиг. 14, началото и крайт на процеса по предоставяне на КАУ се свързва с регистрация и управление на документи. По тази причина, един от основните модули на АИС за КАО е **Модулът за управление на документи и преписки**. Той предоставя функционалност по регистрация, обработка, съхранение и предоставяне на документи в електронен вид. Тъй като всяка администрация разполага със **Модул за управление на документи и преписки**, а и поради факта, че той е входната точка, иницираща процеса по предоставяне на КАО, и изходна точка – генерираща резултантния от реализирането на КАО документ, предлаганото архитектурно решение поставя **Модула за управление на документи и преписки** като основен елемент на АИС за КАО, част от нейното ядро. Втората група модули са свързани с извличане на данни по служебен път, техния анализ и обработка. Те използват съвременни комуникационни технологии за обмен на документи и структурирани данни между администрациите. Тези модули консумират различни услуги, предоставяни от централизираните вертикални системи на е-управление или системи на конкретни администрации. След извличането на данните се извършва анализ, който определя дали услугата ще бъде изпълнена, т.е. дали ще бъде изготвен резултантен документ, а данните от него вписани в регистри. Всички тези процеси се реализират от специализирани модули, управлявани от ядрото на системата.

Ядро на АИС

В ядрото на АИС се включват основни модули, които управляват процеса по предоставяне на КАО. Те са:

- Модул “Управление на документи и преписки”;
- Модул “Управление на работни процеси”;
- Модул “Управление на задачи”;
- Модул “Управление на електронни регистри”
- Модул “Електронно удостоверение”;

Допълнителните модули, необходими за предоставяне на КАО са

- Модул за обмен на документи с други администрации (СЕОС клиент);
- Модул за обмен на документи с граждани и бизнес (ССЕВ клиент);
- Модул за извличане на данни от регистри (RegiX клиент)
- Модул за електронно заявяване за административни услуги, в това число и комплексни, от граждани и бизнес;
- Модул за предоставяне на електронен достъп до документи на физически и юридически лица и други администрации;
- Модул за електронни разплащания.

4.2.1 Модули в ядрото на административната информационна система

4.2.1.1 Модул Управление на документи и преписки

Модулът Управление на документи и преписки предоставя функционалности за регистрация, обработка, управление и съхранение на документи, както в електронен вид, така и пристигнали на хартиен носител. Основните му функции могат да бъдат групирани в следните няколко направления:

- регистрация на документи – това е операция, чрез която документ, пристигнал в администрацията по различни комуникационни канали бива добавен в официален документен регистър. Там той се идентифицира с уникален номер, получен при неговата регистрация;
- управление на преписки – всяка преписка се създава вследствие възникване на инициращ документ. От гледна точка на КАО, това най-често е заявление за предоставяне на АУ. Всяка преписка групира всички документи, възникнали в процеса на предоставяне на АУ. Това могат да бъдат както неструктурирани документи, така и структурирани, които съдържат предварително дефинирана структура от данни, необходими за изпълнение на автоматизирани бизнес процеси, свързани с

предоставянето на КАУ. В преписката задължително се включват заявлението за КАУ и изготвения резултат (удостоверение или друг вид резултатен документ), като могат да бъдат включени и други допълнителни документи, възникнали в процеса на изпълнение на услугата;

- съхранение на всички документи след приключване на процесите по изпълнение на КАУ в архив, който се поддържа определен период от време, определен от класификационна схема, одобрена от Държавна агенция “Архиви”;
- тъй като обработката на документи е неразривно свързана със задачи, изпълнявани от служителите на администрацията (задачите могат да бъдат създавани от други служители или от в резултат изпълнението на автоматизиран бизнес процес, свързан с предоставянето на КАО), модул **Управление на документи и преписки** предоставя контекст за изпълнението на тези задачи.

Модулът за управление на документи и преписки предоставя своята функционалност на останалите модули от АИС, която включва следните групи операции:

- **Регистрация на инициращ документ** (създаване на нова преписка) – чрез тази операция в **Модула за управление на документи и преписки** се създава нов обект от тип документ, който съдържа данни за документа като неговия тип (заявление, сигнал, жалба), вид (входящ, изходящ за администрацията), кореспондент на документа (подател или получател, в зависимост от вида на документа), дата на регистрация в официалния регистър с документи;
- **Регистрация на допълнителен документ към преписка** – резултат от изпълнение на АУ или друг вид документ, необходим за процеса по предоставяне на КАО;
- **Операция, с която се приключва процеса по изпълнение на АУ** (приключване на работата по създадената преписка);
- **Операции за извличане на данни за даден документ**, чрез предоставяне на уникален идентификатор;
- **Операции за извличане на списък от идентификатори на документи** в дадена преписка чрез предоставяне на идентификатор на преписка;
- **Операции за предоставяне на справка** за документи по определени критерии;
- **Абониране за известие** за промяна статуса на документ или преписка.

4.2.1.2. Модул Управление на работни процеси

Модулът Управление на работни процеси дефинира два вида работни процеси – работни процеси, свързани с изготвянето на резултантен документ за заявената услуга и работни процеси, чрез които се извършва процес по съгласуване или утвърждаване на вече изготвени електронни документи. Въпреки че двата вида процеса имат сходства в своите характеристики, те се различават по дейностите, които се извършват при тяхното изпълнение и етапите от предоставянето на услуга, в които те се изпълняват. Характерно и за двата вида процеси е, че те могат да насочват документ или преписка към определени служители в рамките на дадена администрация, последователно или паралелно, а също така и да инициират работни процеси в други администрации. Тези работни процеси се инициират, когато поради обективна причина (напр. липса на данни в регистри, необходимост от изпълнение конкретни, неподлежащи на автоматизация, дейности от служител,) изготвянето на документа – резултат от предоставянето на услугата (акт, удостоверение ли друг вид резултантен документ) не може да бъде напълно автоматизирано и е необходима физическа намеса на служители. На базата на решенията, взети от служителите, изпълнението на процеса може да бъде променено динамично, по подобен начин на този, при който алгоритмите определят операциите, които трябва да бъдат изпълнени на базата на входни и изчислени данни в процеса на изпълнението си.

Един работен процес, свързан с изготвянето на резултат на заявената административна услуга, включва в себе си следните видове действия:

- насочване на документ или преписка към определен служител от администрацията за изпълнение на определена дейност;
- насочване на документ или преписки паралелно към група от служители, като изпълнението на работния процес може да продължи или когато всички служители са изпълнили дейностите, или когато един от тях е изпълнил асоциираната дейност;
- инициране на нов работен процес в текущата администрация, преди текущия процес да е приключил;
- инициране на нов работен процес в друга администрация, преди текущия процес да е приключил.

Работен процес, свързан със съгласуване или утвърждаване на електронни документи, включва в себе си изброените по-долу действия. На практика, този вид работен процес ограничава дейностите, изпълнявани от служителите върху документите до:

- запознаване на даден служител със съдържанието на електронния документ(и), без възможност за редактиране;
- избор на служителя дали е или не е съгласен със съдържанието на документа;
- подписване на съгласувания електронен документ чрез поставяне на електронен подпис, ако служителят е съгласен със съдържанието на документа или отказ от подпис – ако не е.

По-нататък в дисертационния труд, под работен процес ще се разбират и двата вида процеси, а ако е необходимо ще се направи изрично разграничаване между тях. Модулът за автоматизация на работните процеси е тясно обвързан с модула за управление на задачи, който ще бъде разгледан на по-надолу. При изпълнението си работните процеси инициират действия по насочване на документи към служители за изпълнение на определени дейности. Тези дейности се представят чрез задачи, които трябва да бъдат изпълнени от конкретен служител в рамките на определен срок. Работния процес следи изпълнението на поставените задачи и определя как да продължи изпълнението им. Могат да бъдат разграничени самостоятелни задачи, които трябва да бъдат изпълнени еднократно от даден служител, или групови такива – които се изпълняват паралелно от няколко служители. Всеки работен процес може да инициира друг такъв, който се изпълнява в текущата или друга администрация. За целта АИС разполага с функционалност, чрез която преписката, формирана от заявлението за предоставяне на КАУ, се изпраща към друга администрация посредством модул за обмен на електронни документи, като се транспортират допълнителни метаданни за процеса. Този модул ще бъде разгледан в детайли на по-късен етап от представянето на архитектурата.

Модулът за управление на работните процеси предоставя следната основна функционалност на останалите компоненти АИС:

- инициране на работен процес;
- преминаване на следваща стъпка от работен процес чрез изпълнение на текущата;
- прекратяване на работен процес;
- абониране за известие за преминаване към следваща стъпка от работен процес;
- справки за изпълнение на работен процес.

4.2.1.3. Модул Управление на задачи

Модулът Управление на задачи предоставя функционалност на АИС, с която документите и преписките се насочват на вниманието на служителите с цел изпълнение на определена дейност. Всяка задача включва следните атрибути:

- Уникален идентификатор;
- Служител, на когото е поставена задача (изпълнител);
- Опционален служител, който е поставил задача, ако задачата не е генерирана автоматично от работен процес;
- Уникален идентификатор на работния процес, ако задачата е генерирана от работен процес;
- Дата на поставяне на задача и срок за нейното изпълнение;
- Описание на дейността, която трябва да бъде изпълнена;
- Вид на дейността, която трябва да бъде изпълнена (от номенклатура);
- Предефинирани отговори за изпълнение на задачата – те могат да бъдат определени от работния процес;
- Отговор (отчет) на поставената задача, който се попълва от нейния изпълнител;
- Дата на отчитане на поставената задача.

Основните функционални възможности, които **Модулът Управление на задачи** предоставя на останалите компоненти на АИС са:

- създаване на задача с определени атрибути (изпълнител, срок, описание на дейността, която трябва да се изпълни);
- отчитане на изпълнението на задача;
- анулиране на поставена задача, единствено ако са изпълнение определени условия;
- промяна на атрибути на задача, ако са изпълнение определени условия;
- абониране за промяна в статуса на задача;
- справки за създадените задачи.

Двата модула – **Модул Управление на документи и преписки** и **Модул Управление на работни процеси** са тясно свързани с модула за управление на задачи. Това означава, че те са зависими от програмния интерфейс, предоставен от него.

4.2.1.4. Модул Електронно удостоверение

Модулът Електронно удостоверение има за цел да предостави частична или пълна автоматизация на процесите по изготвяне на резултатни документи при изпълнение на АУ. За целта за всяка една АУ се конфигурира набор от справки, които модулът

трябва да изпълни посредством **Модулът за извличане на данни от регистри** и съвкупност от проверки, които трябва да бъдат изпълнени, използвайки получените данни.

Функционалността на **Модул Електронно удостоверение** може да бъде групирана в следните три дейности:

- изпълнение на последователност от справки от първични регистри на данни с цел събиране на необходимите факти и обстоятелства, необходими за изпълнение на заявената АУ;
- анализ на получените данни, който може да включва реализиране на кръстосани проверки в тях;
- изготвяне на резултантен документ в резултат от направения анализ, използвайки предварително зададен шаблон.

4.2.1.4.1. Изпълнение на справки от първичните регистри на данни

Основната задача тук е да изпълни необходимите справки, чрез които да се извлекат данни, необходими за изпълнение на АУ. Всяка справка се състои от следните два типа данни:

- Входни данни – това са данни с определена структура, които трябва да се предоставят на съответния регистър;
- Изходни данни – това е полученият резултат от изпълнението на справката, който също представлява предварително дефинирана структура от данни, която съдържа определени факти и обстоятелства, необходими за изпълнението на АУ.

Изпълнението на справките се извършва последователно, тъй като е възможно изходните данни на една справка да послужат за входни данни на друга справка. По този начин след всяка следваща заявка се увеличава набора от налични данни, необходими за определянето на възможността за предоставяне на заявената АУ. Наборът от данни, който инициира процеса са данните попълнени в заявлението за АУ. Това най-често е идентификаторът на лицето (ЕГН или ЕИК). За определени услуги, последователността от справки, които трябва да бъдат изпълнени може да бъде динамична, което означава, че определянето коя ще бъде следващата изпълнена справка зависи от данните, получени в резултат от изпълнението на предходна. Напр., ако за изпълнението на дадена услуга е необходимо да се определи факт или обстоятелство относно свързано лице – напр. съпруг/съпруга на лицето, за което се изисква издаване на документ, след изпълнението на справка за сключен брак, ако лицето няма такъв, ще отпадне необходимостта от изпълнение на последваща справка.

4.2.1.4.2. Анализ на получените данни

Анализът на получените набори от данни представлява последователност от проверки, които трябва да бъдат описани по структуриран начин. Всяка проверка може да бъде представена като израз, в който участват извлечени стойности от един или повече набори от данни. Когато за потвърждението на даден факт е необходимо да участват два или повече набори от данни може да се говори за кръстосани проверки. Извличането на конкретни стойности от една или повече справки може да се извърши чрез стандартни технологии като XPath, XQuery когато данните са в XML формат или JSONPath, когато данните са представени в JSON формат.

4.2.1.4.3. Изготвяне на резултантен документ

В резултат от извършения анализ е определено дали заявената АУ може да бъде предоставена или е необходимо тя да бъде отказана. Когато в резултат от изпълнението на АУ е необходимо да се издаде физически документ (удостоверение, лиценз) **Модул Електронно удостоверение** ще бъде конфигуриран с бланки, в които определени полета, наречени маркери ще бъдат заменяни със стойности от извлечените набори от данни. По този начин, след замяна на всички маркери със съответните стойности ще се създаде резултантен документ. В случай, че АУ не може да бъде предоставена, ще се създаде документ за отказ от предоставяне на услуга, използвайки същия механизъм.

4.2.1.5. Модул за управление на електронни регистри

Модулът за управление на електронни регистри е основна част от АИС на администрацията, тъй като чрез него тя съхранява, управлява и предоставя достъп до данни на други администрации, относно лица или обекти.

Основните функционални изисквания към този модул са:

- управление на регистри за които администрацията е определена за първичен администратор на данни (ПАД);
- предоставяне на справки върху данните, съхранявани в регистрите на администрацията към други администрации посредством средата за междурегистров обмен RegiX или друг програмен интерфейс.

4.2.1.5.1. Място на регистрите в процеса по предоставяне на КАУ

Всяка администрация е създадена за да поддържа различни данни за физически, юридически лица и обекти, свързани с тях или обекти, които са от съществено значение или интерес за населението или други администрации на национално или локално ниво. На практика всяка таблица в дадена администрация, в която са съхранени данни за списък от обекти или лица може да се разглежда като регистър. От друга страна,

мащабните информационни системи, в които се съхраняват данни за населението, собствеността или други факти и обстоятелства на национално ниво, също представляват регистрови системи. Оттук следва, че независимо от вида и обема на данните в различните регистри, данните в тях трябва да бъдат достъпни по електронен път за останалите администрации. Един от примерите за малък регистър, който в същото време оказва значително влияние върху важни процеси за населението, като прием на деца в държавните детски градини, е регистърът на сертифицирани от СРЗИ частни яслени заведения. Този регистър на практика представлява списък с частните детски заведения, които отговарят на определени критерий и децата, които са били записани в тях, както и периода в който са ги посещавали. Защо е важен този регистър? Защото всяко дете, което е посещавало такова детско заведение за определен минимален период от време, според [12] получава допълнителна точка при кандидатстване в държавна детска градина, където местата са много ограничени и тази допълнителна точка е от изключително значение. Следователно, наличието на електронен регистър, който предоставя справка за посещение на такъв тип детско заведение от дете с определено ЕГН би бил изключително полезен в процеса за кандидатстване в държавните детски градини. За родителите това е облекчение, защото намалява броя документи, които трябва да приложат при кандидатстване в държавно заведение, а за администрацията е облекчение – тъй като справката ще се изпълни автоматично и детето ще получи допълнителната точка без да е необходимо даден служител да прави ръчни проверки.

Този прост реален пример ясно показва, че наличието на електронни регистри е от съществено значение при предоставяне на КАУ, особено когато данните в тях са нужни за предоставяне на определена услуга.

От гледна точка на архитектурата на АИС, ориентирана към КАО, **Модулът за управление на електронни регистри** има важно значение и участието му в процеса на предоставяне на КАУ се обуславя от следните зависимости с останалите модули на АИС:

- **Модул за управление на документи и преписки** – когато системата за управление на документи приеме заявление за АУ, то съдържа структурирани данни, попълнени от заявителя. При някои АУ след завършване на работния процес по нейното предоставяне или по-точно в неговия край, тези структурирани данни трябва бъдат изпратени към модула за управление на регистри с цел създаване или модифициране на запис. Това означава, че всеки регистър трябва да предоставя програмен интерфейс за приемане на структурирани данни. Структурата на тези

данни зависи се определя от самия регистър и тъй като има вероятност данните от заявлението да не са напълно съвместими с данните, очаквани от регистър е възможно да е необходима тяхната трансформация. Тази трансформация може да се изпълни от **Модул Електронно удостоверение** посредством изпълнение на трансферна схема.

- **Модул за извличане на данни от регистри** – този модул, като част от АИС на администрацията, може да изпълнява заявки за изпълнение на справки както от външни регистри (напр. посредством средата за междурегистров обмен RegiX) така и от собствените си регистри. Това означава, че модулът трябва да предоставя програмен интерфейс за извличане на данни на останалите модули на АИС.

4.2.2. Основни модули, необходими за предоставяне на комплексно административно обслужване

Основните модули, необходими за реализацията на КАО са:

- Модул за обмен на електронни документи с администрации (СЕОС клиент);
- Модул за обмен на документи с физически и юридически лица (ССЕВ клиент);
- Модул за извличане на данни от регистри на ПАД (RegiX клиент)
- Модул за електронно заявяване на административни услуги, в това число и комплексни, от физически и юридически лица;
- Модул за предоставяне на електронен достъп до документи на физически и юридически лица и други администрации;
- Модул за електронни разплащания.

4.2.2.1. Модул за обмен на електронни документи с администрации (СЕОС клиент)

Модулът за обмен на електронни документи с администрации е от ключово значение за АИС, ориентирана към предоставяне на КАУ, тъй като предоставя функционалност, която обезпечава както преноса на неструктурирани документи между отделните администрации, така и “прехвърля” процеса по тяхната обработка между отделните администрации. Този модул на практика се състои от два подмодула, чиято функционалност ще бъде разгледана отделно. Това са подмодул за изпращане на документи към администрациите и подмодул за приемане на документи от други администрации. Тъй като всяка администрация задължително изпраща и приема документи, то тя разполага с двата подмодула.

4.2.2.1.1. Подмодул за изпращане на документи

Този модул предоставя функционалност на останалите модули на АИС, когато е необходимо:

- да се изпрати документ (структуриран или неструктуриран) до друга администрация. Самото изпращане може да бъде осъществено в два режима. При първия режим, обработката на документа в администрацията, изпращаща документа, се прекратява временно, докато администрацията получател не изпрати необходимия отговор под формата на нов документ. При втория режим на работа, обработката на документа в администрацията-подател не се прекратява, а продължава (режим на изпращане за сведение), но може да бъде спряна на по-късен етап. Ако администрацията-подател предварително е определила какъв административен (работен) процес е необходимо да бъде изпълнен в администрацията-получател, тя може да го специфицира като добави в изпращаните данни уникален идентификатор на работния процес, който трябва да се инициира от администрацията получател, когато тя регистрира документа и започне неговата обработка. Административният работен процес е предварително дефиниран в получателя и администрацията-подател не е запозната с подробности за него.
- да се изпрати информация за промяна на статуса на обработка на документ в администрацията-получател към администрацията-подател на документа. Когато обработка на документа при администрацията-получател завърши с определен статус, който може да бъде положителен или отрицателен, то отговорност на този модул е да изпрати към администрацията-подател резултата от тази обработка. Резултатът обикновено е отделен документ, част от преписката, свързана с предоставянето на КАУ;
- да се изпрати запитване от администрацията-подател за състоянието на обработка на документи, предходно изпратени към нея. Като резултат администрацията-получател създава отговор за текущия статус на обработката, в частност статуса на изпълнение на административния работен процес;

Подмодулът за изпращане на документи предоставя гореописаните функционалности на останалите модули на АИС и се използва в следните случаи:

- инициране на изпращане на документ към друга администрация от **Модула за автоматизиране на работните процеси**. В този случай работния

административен процес, инициран при изпълнение на КАУ в дадена администрация е конфигуриран така, че при определени обстоятелства да изпрати документа-заявление за КАУ към друга администрация, която трябва да предостави документ или документи, необходими за продължаване на процеса по предоставяне на заявената услуга;

- инициране на изпращане на документ към друга администрация от **Модула за управление на задачи**. Този случай е по-особен, тъй като вследствие на поставена задача към служител на една администрация, документът се изпраща към администрацията-получател, защото конкретния служител физически работи в тази администрация. Казусът се среща често, когато става въпрос за централизирана администрация: определени служители фигурират като ръководни позиции, както в министерства, така и в поверените им агенции.
- инициране на изпращане на документ от **Модула за управление на документи и преписки**. Това е най-простият случай, в който служител определя необходимостта от изпращане на документ към дадена администрация – в този случай документът най-често се изпраща в режим за сведение.

4.2.2.1.2. Подмодул за получаване на документи

Подмодулът за получаване на документи предоставя следните функционалности на останалите модули на АИС:

- Получаване на документи (структурирани и неструктурирани) от други администрации. Процесът по получаване включва валидация на получените данни, тяхната обработка и подготовка на отговор за статуса на получаването. Най-съществения етап е обработката на получените документи, която включва:
 - Съхранение на получения документ в базата данни на АИС;
 - Инициране на работен процес, ако подателят на документа е посочил такъв.
- Получаване на информация за статус на изпълнение на обработката на документа в администрацията-получател на документа – в този случай, ако е бил инициран работен процес в администрацията-получател, тя е изпратила информация относно това как процесът е завършил. Могат да бъдат дефинирани различни “резултати” от изпълнения работен процес, които определят как трябва да продължи работния процес в администрацията-подател. Възможните варианти са:

- процесът да продължи напред, тъй като необходимия документ е изготвен;
- процесът да се върне една стъпка назад, тъй като е необходимо предоставяне на допълнителна информация или корекция;
- процесът да се рестартира в администрацията-подател за да започне отначало;
- процесът да приключи, тъй като изисканият документ не може да бъде издаден поради обективна причина.

Изпълнението на една КАУ, включва съвместна работа между служителите на отделните администрации, когато необходимите данни не са налични в електронен вид под формата на регистрирани ИС_{ис}. Именно **Модулът за обмен на електронни документи с администрации** е този, който обезпечава тази съвместна работа.

4.2.2.2. Модул за обмен на документи с физически и юридически лица (ССЕВ клиент)

Модулът за обмен на документи с физически и юридически лица е интегриран от една страна с централизирана система на електронното управление ССЕВ (Система за сигурно електронно връчване), чрез която физически и юридически лица могат да изпращат и получават структурирани и неструктурирани документи до и от администрациите, а от друга страна – с модула за управление на документи и преписки на администрацията. По този начин **Модулът за обмен на документи с физически и юридически лица** осигурява интеграционна функционалност между ССЕВ и АИС по отношение на получаване и изпращане на документи.

Модулът за обмен на документи с физически и юридически лица предоставя следните функционалности на АИС:

- проверка за наличие на нови документи, несполучени и необработени;
- извличане съдържанието на документ, изпратен от дадено лице;
- изпращане на документ от АИС към неговия получател;
- проверка за наличие на профил в ССЕВ на определено лице, която определя, дали то ще получи изпратения документ.

4.2.2.3. Модул за електронно заявяване на административни услуги, в това число и комплексни, от физически и юридически лица

Модулът за електронно заявяване на административна услуга може да се разглежда като самостоятелна ИС_{ис} от гледна точка на сложността му и обема на функционалните възможности, които той реализира. За целите на този труд в детайли ще бъдат

разгледани само допирните точки с АИС по отношение на предоставянето на КАУ. **Модулът за електронно заявяване на административна услуга** предоставя възможност за създаване на електронни документи-заявления за предоставяне на определена АУ от физически и юридически лица. В резултат от неговата работа се създават структурирани документи, които съдържат информация за заявителя, факти и обстоятелства относно него или обекти, свързани с изисканата услуга включително заплатени такси. Към тези структурирани данни могат да бъдат приложени и неструктурирани документи – такива, които не могат да бъдат използвани директно от АИС без намесата на служител. Когато завърши процеса по електронно заявяване на АУ, модулът предоставя на АИС цялата тази информация посредством интеграционен модул.

На практика **Модулът за електронно заявяване на административна услуга** предоставя уеб базиран потребителски интерфейс на лицата, заявяващи АУ, чрез който те:

- създават ново заявление за АУ като попълват в електронна форма всички изискуеми данни. Ако модулът е интегриран с национални регистри, част от тези данни могат да бъдат попълнени автоматично. Прилагат допълнителни неструктурирани документи;
- проследяват отделните етапи от изпълнение на АУ;
- при необходимост създават коригирано заявления за АУ;
- получават резултата от изпълнението на АУ като електронен документ или отказ от изпълнението на АУ при невъзможност за издаване на изискания документ.

Модулът за електронно заявяване на административна услуга се интегрира към АИС посредством следните програмни интерфейси, предоставяни от **Модула за управление на документи и преписки** и **Модула за автоматизиране на работните процеси**:

- Регистрация на документ-заявление в модула за управление на документи и преписки – като резултат към **Модулът за електронно заявяване на административна услуга** се изпраща регистрационен номер на заявлението. Към заявлението, в **Модула за управление на документи и преписки** се съхраняват и всички структурирани данни, необходими на по-късен етап в процеса на електронно изпълнение на АУ;
- Инициране на работен процес – в зависимост от заявената АУ, интеграцията с **Модула за автоматизиране на работните процеси** позволява инициране на съответния работен процес, който насочва вече

регистрираното заявление към отговорни служители, посредством **Модула за управление на задачи**.

- Промяна на състояние на изпълнение на АУ – необходимо е АИС да поддържа набор от състояния, в които може да се намира дадено заявление за АУ. Той може да включват следните основни състояния:
 - ❖ В процес на обработка;
 - ❖ Очаква предоставяне на допълнителни документи или факти;
 - ❖ Очаква допълнително заплащане;
 - ❖ Приключено - успешно изпълнена АУ;
 - ❖ Отказано изпълнение на АУ.

Всяко едно от състоянията може да бъде детайлизирано с по-конкретни под-състояния. Например: Състоянието *“В процес на обработка”* може да бъде допълнително детайлно със следните под-състояния:

- Извършване на първоначална проверка на предоставените документи;
- Събиране на допълнителни факти и обстоятелства;
- Изготвяне на резултантен документ.

При първоначално постъпване на заявлението за АУ в **Модула за управление на документи и преписки** му се присвоява състояние *“В процес на обработка”*. След извършване на предварителни проверки то може да премине в състоянието *“Очаква предоставяне на допълнителни документи или факти”*, когато от заявителя се очаква да извърши необходимите корекции по заявлението. Ако те бъдат извършени се възобновява работата по коригираното заявление, което преминава отново в състояние *“В процес на обработка”*. Ако е необходимо допълнително заплащане за изпълнението на АУ, процесът преминава в състояние *“Очаква допълнително заплащане”*. Крайните две състояния *“Приключена – успешно изпълнена АУ”* и *“Отказано изпълнение на АУ”* прекратяват работата по предоставянето на заявената АУ. При промяна на статуса на изпълнение на АУ, АИС изпраща към **Модулът за електронно заявяване на административна услуга** допълнителна информация, необходима за неговата работа. Това могат да бъдат както структурирани данни, така и неструктурирани документи, които се предоставят на заявителя посредством уеб интерфейс. Дали заявяваната ЕАУ е комплексна или не, не променя по съществен начин работа на разглеждания модул. Ако заявителят е избрал, че желае изпълнение на заявената АУ като КАУ, то част от данните, които трябва да докаже чрез прилагането на документи, ще бъдат *“събрани”* по служебен път от АИС. Това означава, че в този случай няма да е необходимо да приложи всички изискуеми документи, а само задължителните, ако има такива.

4.2.2.4. Модул за извличане на данни от регистри (RegiX клиент)

Модулът за извличане на данни от регистри е много важен за предоставянето на КАУ. Това се обуславя от факта, че за предоставянето на такъв вид услуга, администрацията трябва да набави по служебен път данни, които се намират в различни регистри на различни администрации. За целта АИС трябва да разполага с модул, който може да бъде надграждан лесно с интеграционни подмодули, обезпечавачи комуникацията със системи от тип регистър. Този вид комуникация може да се разглежда като отношение клиент-сървър, при което АИС на администрацията, изпълняваща КАУ, изпраща заявка към регистровата система на администрацията съхраняваща необходимите данни, а тя от своя страна – генерира отговор на тази заявка, съдържащ необходимите факти и обстоятелства. И заявката и отговорът представляват структура от данни, които трябва да бъдат семантично разпознаваеми и за двете страни. За целта е необходимо да бъдат изпълнение следните изисквания:

- на АИС на администрацията, предоставяща КАУ, е предоставен сигурен и защитен достъп до програмен интерфейс на регистровите системи, съхраняващи фактите и обстоятелствата, от които тя се нуждае. Този програмен интерфейс трябва да отговаря на изискванията за оперативна съвместимост между ИС на администрациите;
- Регистровата система предоставя структури на заявките, които поддържа и отговорите, които се генерират в резултат от изпълнението на всяка заявка. Тази структури трябва да бъде описана в машинно четим формат;
- Трябва да се гарантира, че двете системи са семантично съвместими (семантична оперативна съвместимост) по отношение на обменяните данни – например авторите и на двете системи са вложили еднакъв смисъл в стойността на определен атрибут от обменяните данни.

Средата за междурегистров обмен RegiX, представлява софтуерна фасада, която унифицира достъпа до голяма част от регистри на администрациите. Именно за това тя е полезна, тъй като с малко усилия могат да бъдат добавяни нови регистри, до които АИС на администрациите, предоставящи КАУ, да имат достъп.

Някои регистрови системи предоставят функционалност за абониране към възникване на определено събитие – най-често промяна в данните за обекти, за които се съхраняват факти и обстоятелства. Това означава, че **Модулът за извличане на данни от регистри** трябва да имплементира такъв тип функционалност.

Модулът за извличане на данни от регистри трябва да предоставя следните основни функционалности на останалите модули на АИС:

- Изпълнение на заявка за извличане на данни от регистър
- Абониране за известие за настъпване на промени в определен регистър

4.2.2.5. Модул за предоставяне на електронен достъп до документи на физически и юридически лица и други администрации

Модулът за предоставяне на електронен достъп до документи на физически и юридически лица и други администрации участва индиректно в предоставянето на КАО като предоставя достъп до вече издадени електронни документи на физически и юридически лица или други администрации. Чрез този модул, всеки издаден от администрацията електронен документ, включително документите, създадени в резултат от изпълнение на АУ, може да бъде “публикуван” в Интернет, а достъпът да него да се управлява посредством съвременни технологии като QR код, секретен код или специално генериран URL адрес. На практика функционалността на този модул позволява на собственика на даден електронен документ да го представи на друга администрация или заинтересовано лице директно чрез неговото споделяне. Непосредствено след регистрирането на издадения документ в модула, достъп по него получава единствено неговия адресат получател (физическо или юридическо лице) След като осъществи достъп до издадения документ, получателят може да създаде допълнителен достъп до документа, оказвайки срок на валидност на този достъп, който да предостави на друго лице или администрация. По този начин, ако е необходимо собственикът на електронния документ да го приложи към документи за заявяване на друга административна услуга вместо самия документ, той може да приложи информацията (QR код, URL адрес) за достъп до документа. Този модул е полезен, когато администрацията няма право да предоставя достъп до своите електронни регистри на други администрации поради законово ограничение или притежава регистри, които не са електронизирани. Модулът за предоставяне на електронен достъп до документи на физически и юридически лица и други администрации поддържа различни нива на достъп до документа, които определят каква част от данните в него са достъпни или дали е достъпно за изтегляне неговото файлово съдържание.

Модулът предоставя следния програмен интерфейс:

- Регистрация на документ за предоставяне на достъп;
- Създаване на достъп с определено ниво до определен документ;
- Деактивиране на създаден достъп до документ;
- Справка за осъществени достъпи на документа;

Наличието на този модул не е задължително, но би улеснило гражданите, когато е необходимо да предоставят документи, които са издадени от други администрации. Пример за такива документи са скици на имот, които се създават като електронен документ, съхраняват се в базата данни на АИС на администрацията, която ги издава и имат определен срок на валидност.

4.2.2.6. Модул за електронни разплащания

Модулът за електронни разплащания предоставя възможност да създаване на задължения към физически и юридически лица в процеса на предоставяне на КАО. Той интегрира АИС с централизираната системата за е-управление за разплащателни услуги pay.egov.bg.

Модулът за електронни разплащания предоставя следните услуги на ядрото на АИС:

- Създаване на задължение към конкретно лице, използвайки неговия идентификатор;
- Анулиране на вече създадено задължение;
- Проследяване статуса на създадено задължение.

4.2.2.6.1. Създаване на задължение

Тази операция позволява автоматизиране на процеса по генериране на задължение към заявителя на дадена услуга. АИС е предварително конфигуриран с конкретни суми, които се заплащат за различни административни услуги, а при услуги с плаваща такса, служителят въвежда брой или коефициент, който определя крайната сума. АИС автоматично извлича всички необходими данни от базата си данни, които трябва да се подадат на модул за електронни разплащания, а той от своя страна формира заявка за създаване на задължение към централизираната система на ЕУ pay.egov.bg.

Необходимите данни, подавани на модула за електронни плащания включват:

- Идентификатор на лицето (ЕГН/ЕИК);
- Име на лицето;
- Размер на създаваното задължение;
- Вид и код на плащането, които представляват номенклатури, използвани в националните разплащателни системи;
- Регистрационен номер и дата на регистрация на заявлението за АУ, с което заявителя на услугата е изискал нейното изпълнение;
- Допълнителна причина и коментар.

В резултат от изпълнението на тази заявка, централизираната система за електронни разплащания генерира отговор, в които се съдържа уникален идентификатор на

задължението или код на възникналата грешка. Уникалният идентификатор се използва от АИС за последващи операции.

4.2.2.6.2. Анулиране на задължение

Анулиране на задължение е операция, с която вече създадено задължение може да бъде премахнато чрез посочване на неговия уникален идентификатор. Тази операция не винаги е успешна, тъй като заявителят може да е инициирал процес по заплащане. Тогава централизираната система за електронни разплащания ще отговори със статус неотменимо задължение.

4.2.2.6.3. Проследяване статуса на създадено задължение

Тази операция дава възможност за установяване статуса на създадено задължение чрез посочване на неговия уникален идентификатор. **Модулът за електронни разплащания** следи статусите на задължения, които не са влезли в едно от крайните си състояния (заплатено, отказано). Статусът е важен, тъй като работния процес, свързан с предоставянето на АУ може да продължи когато тя е заплатена. Макар и модул с некритична функционалност, модулът за електронни разплащания има своя роля в процеса по предоставяне на АУ.

4.2.3. Допълнителни и спомагателни модули

4.2.3.1. Модул за идентификация и автентикация на потребители

Модулът за идентификация и автентикация на потребители предоставя своята функционалност като услуга на компонентите от ядрото на системата. Предоставя възможност за регистрация и автентикация както на вътрешни, така и на външни за системата потребители. Външни потребители са такива, които не са служители на администрацията, например заявители на услуги по електронен път. Поради тази причина предоставя функционалност за регистрация, използвайки различни средства за идентификация, включително националните схеми за електронна идентификация, разработена като вертикална система на Е-Управление.

Модулът за идентификация и автентикация на потребители предоставя следните услуги:

- Регистрация на потребители;
- Автентикация на потребители;
- Управление на потребители.

4.2.3.2. Модул за управление на потребителски роли и права

Модулът за управление на потребителски роли и права предоставя услуги на модулите от ядрото на системата по управление на правата на различни потребители до тяхната функционалност. Този модул определя даден потребител, преминал успешна автентикация, какъв достъп до основната функционалност на АИС притежава. За улеснение той дефинира различни потребителски роли, които се асоциират с потребителите. Всеки модул получава информация за това даден потребител какви асоциирани роли притежава.

Модулът за управление на потребителски роли и права предоставя следните услуги:

- регистрация на потребителска роля, асоциирана с определен модул или подсистема;
- асоциация на потребител с потребителска роля;
- извличане на списък от роли за конкретен потребител, асоциирани с определен модул или подсистема.

4.2.4. Примерна реализация на Модул за електронно удостоверение

За да бъде представена примерна реализация на модула е необходимо да бъдат въведени някои основни термини, които ще бъдат разгледани по-долу.

4.2.4.1. Острови от данни

Остров от данни е структура от данни, именуван набор от данни, в който има конкретни, добре описани данни и от/в който лесно ще може да се извличат или записват данни, чрез **трансферна схема**. Трансферната схема представлява скриптова опростена технология, позволяваща да се трансферират данни от един остров в друг, като управлението е динамично извън модула (за целта се използва вграден скриптов език).

Напр. синтаксисът може да бъде от следния вид:

$$\text{Остров } Z = \text{Трансферна схема } 1 (\text{Остров } X, \text{Остров } Y)$$

Скриптовете се извикат служебно и контекстно на определена стъпка в работен процес, асоциирана от системата за управление на документи към обработваното заявление за АУ. При определени АУ изпълнението на тези скриптове е напълно автоматизирано и в резултат от тяхното изпълнение е формиран остров от данни "Резултат", който служи за изготвяне на резултатния документ. При други АУ пълна автоматизация не е възможна най-вече в следствие на липса на структурирани данни, необходими за

формирането на даден остров. В тази ситуация е необходимо намесата на служител, който да извърши допълнителни действия.

Смисълът на понятието остров от данни и трансферна схема е постигане на простота при оформяне на скриптовете.

Основни типове „острови“ от данни са:

- **ОСТРОВ Заявление за АУ, попълнено на хартиен носител** – този остров, съдържа комплект с данни за обработвания документ-заявление за услугата от **Модула за управление на документи и преписки**. Съдържа основни данни като заявителя на услугата, вида на изпълняваната услуга. Липсват структурирани данни от самото заявление. Този остров се използва когато заявлението за АУ е попълнено от заявителя на лист хартия;
- **ОСТРОВ Електронно заявление за АУ** – в него се съдържат структурирани данни за конкретна ЕАУ, въведени от заявителя чрез **Модулът за електронно заявяване на административна услуга**;
- **ОСТРОВ Заявка справка Regix** – това е структура, която се създава за да се извика съответната справка за извличане на данни от Regix или друга система, предоставяща данни от регистри. В тази структура, в зависимост от контекста за заявката, трябва да се запишат различни данни;
- **ОСТРОВ Резултат справка RegiX** – това е структура, която се създава при получаване на отговор от изпълнена RegiX справка;
- **Остров Диалогова форма** – това е структура данни, която позволява на потребителя да въведе определени данни, които не са налични в останалите острови и са външни за системите. Това обаче позволява динамично да се въведат данни, които да се обработят;
- **ОСТРОВ Локални данни за администрацията** – това са данни, които се персонализират за всяка администрация и служи за унифициране документи ѝ. Идеята е те да правят бланките персонални за администрацията чрез смяна на герба, името на администрацията, името на представляващия администрацията, ЕИК и други данни;
- **ОСТРОВ Бланка** - това е структура, в която се съдържат данни, които ще бъдат вмъкнати в конкретна бланка, използвана за създаване на резултантен документ .

Всеки остров има структурно описание, представляващо метаданни за самия остров. Когато островът представлява данни в XML формат, метаданните представляват XSD дефиниция. Инстанция на остров представлява структура с конкретни данни. Когато те са в XML формат, инстанцията на острова представлява XML документ.

Инстанциите на островите от данни се съхраняват в база от данни, до която различните модули на системата имат достъп и могат да добавят нови. Добавянето се извършва чрез програмен интерфейс предоставен от **Модулът за електронно удостоверение**. **Модулът за управление на документи и преписки** на АИС добавя острова **Заявление за АУ, попълнено на хартиен носител**, когато заявлението е подадено на гише. **Модулът за електронно заявяване на административна услуга** добавя остров **Електронно заявление за АУ**, което съдържа структурирани данни на заявлението. По-надолу, под остров от данни ще се има предвид инстанция на остров, ако не е указано друго. В процеса на обработка на заявление за КАУ се създават множество острови, съхранявани в контекста на обработваното заявление, които формират “архипелаг от острови”. Всеки остров може да принадлежи само на един архипелаг.

4.2.4.2. Трансферни схеми

Трансферните схеми са структури, които управляват трансфера на данни между отделните острови. Смисълът на трансферните схеми е следния – да могат да се създават лесно нови острови от вече създадени такива. Този механизъм се използва, когато е необходимо да се създават острови от тип “*Заявка за справка Regix*” или “*Бланка*”. За реализацията на трансферни схеми могат да бъдат използвани утвърдени технологии като XSLT, когато данните са представени в XML формат.

4.2.4.3. Бланки

Бланките представляват шаблони на документи които представят общата структура на даден документ, конкретните стойности са заменени от маркери. При създаване на документ от бланката всеки маркер се заменя с конкретна стойност от вече създаден остров от данни от тип бланка.

4.2.4.4. Скрипт за създаване на електронни документи

Скриптовете са обединяващо звено между островите от данни, трансферни схеми и бланки. Всеки скрипт включва в себе си следните базови операции:

- Зареждане на остров от данни чрез посочване на неговия идентификатор – могат да бъдат заредени острови, за които има създадени инстанции в базата данни на модула;
- Изпълнение на трансферна схема за създаване на нов остров от данни. Тук входните данни за операцията са един или повече заредени острови от данни и трансферната схема, която трябва да бъде приложена към тях. Резултантния остров може да бъде използван за създаване на заявка за справка към регистрова система от типа на RegiX;

- Извличане на данни от регистър – за целта **Модулът за електронно удостоверение** използва програмен интерфейс предоставен от **Модулът за извличане на данни от регистри** като формира необходимата заявка, използвайки даден остров от данни. Резултатът, върнат от регистровата система, се съхранява като нов остров от данни, който, ако е необходимо може да бъде преобразуван към нов с прилагане на трансферна схема;
- Изпълнение на проверки върху данните от един или няколко острова чрез изпълнение на булеви изрази – това са изрази, в които участват операнди, представляващи конкретни стойности от даден остров, връщащи булеви стойности. Чрез тези стойности се определя как продължава изпълнението на скрипта, дали то трябва да се преустанови и какъв е крайният резултат от неговото изпълнение;
- Създаване на документ от бланка (шаблон), използвайки остров от данни. Тази функционалност може да се използва и когато е необходимо да се изпрати документ към друга администрация с искане за изпълнение на вътрешна административна услуга;
- Изпращане на съобщение към друга администрация за попълване на данни в остров от тип диалогова форма. Идеята тук е когато поради различни обстоятелства (най-често липса на електронен регистър) даден остров не може да бъде създаден автоматично чрез извличане на данни от регистрова система чрез това съобщение се предоставя уеб адрес на администрацията първичен администратор да предостави необходимите данни. Неин служител попълва необходимите данни за острова като ги подписва с електронен подпис и времеви печат, удостоверявайки тяхната автентичност. Тази операция временно прекратява изпълнението на скрипта и той преминава в статус *“изчакване”*;
- Извикване на друг скрипт, по подобие на извикване на подпрограма в структурното програмиране.

Скриптовете представляват XML структура, в която всяка операция е представена чрез XML елемент, който съдържа необходимите параметри за нейното изпълнение – входни данни, инструкции за съхранение на резултата от изпълнението на операцията.

4.2.4.5. Механизъм за изпълнение на скриптове

Механизмът за изпълнение на скриптове представлява програмен модул, който може да бъде стартиран със следните входни данни:

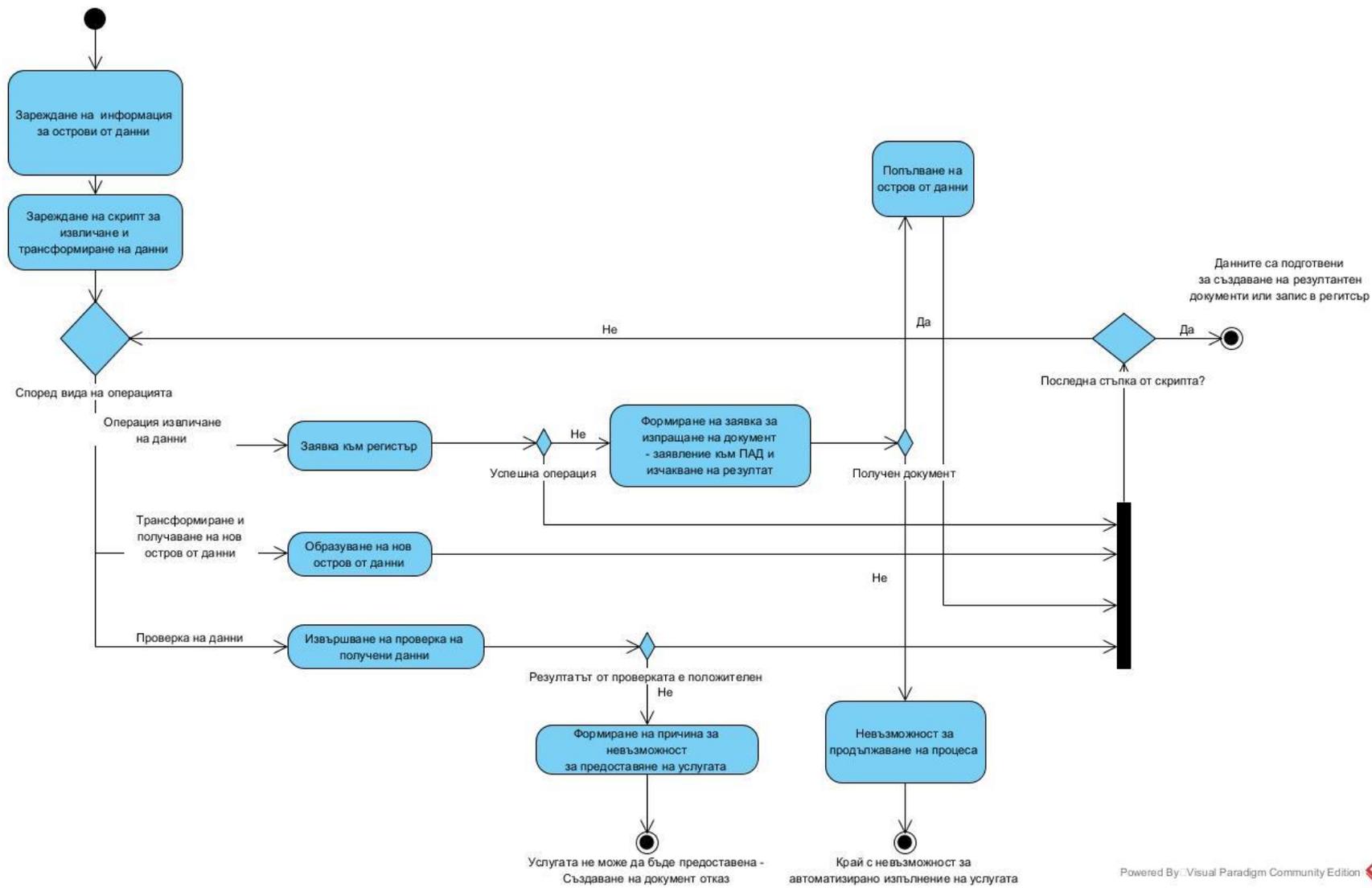
- Идентификатор на скрипт, който трябва да бъде изпълнен
- Контекст на АУ, за която трябва да бъде изпълнен скрипта

Механизмът за изпълнение на скриптове зарежда подадения скрипт като структура от данни и започва да изпълнява последователността от операции, описани в скрипта. С цел лесно разширяване на възможностите на модула, механизмът за изпълнение на скриптове поддържа версии на скриптове, като всяка следваща версия поддържа по-обширен набор от операции, които могат да се изпълняват.

4.2.4.6. Интеграция с работни процеси

При постъпване на заявление за КАУ се инициира работен процес в АИС. **Модулът за електронно удостоверение** разширява стандартните работни процеси като добавя възможности за стартиране изпълнение на скрипт за създаване на електронни документи, обръщайки се към механизма за изпълнение на скриптове. По този начин работен процес инициира процес по изготвяне на резултантен документ. При необходимост от намеса на служител в работния процес обратно се инжектират стъпки или задачи, с които се изпълняват необходимите действия. Тези стъпки най-често са дейности, които са свързани с човешка намеса, обусловена от липсата на възможност за автоматизирано извличане на необходимите данни, т.е. не е възможно формиране на необходим остров от данни.

Процесът по извличане и анализ на структурирани данни е представен в детайли на диаграмата на Фиг. 15.



Фиг.15. Процес по извличане и анализ на структурирани данни

Процесът се имплементира от **Модула Електронно удостоверение** и се описва със следните етапи:

- Зареждане на информация за острови от данни – процесът се инициализира със зареждане на метаданните за всички острови от данни, които са необходими за определяне на резултата от изпълнението на АУ. Съществуващите към момента острови от данни (свързани със заявлението за КАУ) се зареждат в списък *“Инициализирани острови”*. Когато съществуват множество инстанции на даден остров, съдържащи еднакви по структура данни, отнасяща се за различни обекти, всеки от тези острови се съхранява като елемент от списък;
- Зареждане на скрипт за извличане и трансформиране на данни представлява прочитане от базата данни на асоцииран скрипт към предоставяната КАУ, който изпълнява последователност от операции по изпълнение на заявки към определени регистри, създаване на нови острови от данни, извършване на проверка на получените данни. Скриптът се зарежда в оперативната памет като списък от последователни операции, които ще бъдат изпълнени от интерпретатор на операции;
- Изпълнение на описаните в скрипта операции се извършва като цикъл, който продължава докато в скрипта съществуват неизпълнени операции или процесът трябва да извърши поради невъзможност за определяне съдържанието на конкретен остров от данни. Изпълнение на различните видове операции се изпълнява от интерпретатора на скрипта по следния начин:
 - ❖ Операция **Извличане на данни от регистри** – тази операция използва инициализиран остров (чрез посочване на неговото име, а ако островът е част от списък и неговия индекс), чрез който се формира заявка за извличане на данни от регистър. Резултатът от изпълнението на заявката е нов остров от данни, който се добавя към списъка с инициализираните острови. Ако заявката не успее да бъде изпълнена в рамките на няколко опита, се формира заявка за изпращане на неструктуриран документ – заявление за ВЕАУ към ПАД, контролиращ съответния регистър. Ако ПАД не предостави изискания документ, процесът не може да продължи и се приключва със статус *“Край с невъзможност да автоматизирано изпълнение на услугата”*. Ако документът се предостави, чрез информацията в него се създава липсващия остров от данни и процесът продължава;

- ❖ Операция **Образуване на нов остров от данни** – чрез изпълнение на трансферна схема към един или няколко инициализирани острови от данни, се създава нов остров от данни, който се добавя към списъка с инициализираните острови;
- ❖ Операция **Извършване на проверка на получени данни** представлява анализ на данните, съхранени в инициализираните острови от данни чрез изпълнение на логически изрази, представляващи условията необходими за предоставяне на КАУ. Ако резултатът от проверката е положителен, се счита че тя е успешна и процесът продължава със следващата операция. Ако резултатът от проверката е отрицателен, процесът приключва с формиране на причина за невъзможност за предоставяне на заявената КАУ.

4.3. Основни интерфейси за комуникация между модулите в предлаганото архитектурно решение

На диаграмата на Фиг. 16 са представени основните компоненти, участващи в архитектурата на АИС, ориентирана към предоставяне на КАО. Всеки от тях предоставя програмен интерфейс, чрез който останалите модули могат да консумират функционалността, която той имплементира.

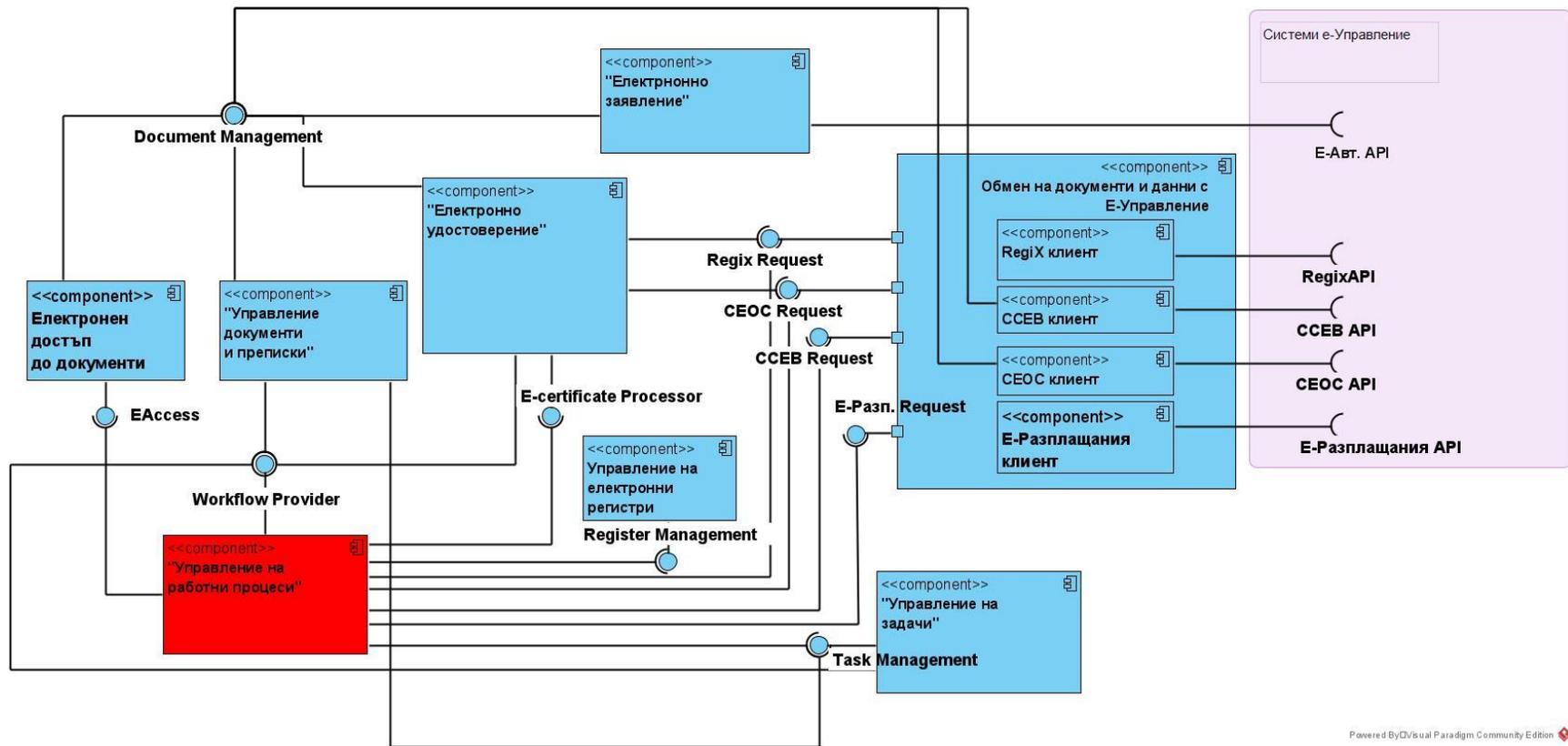
4.3.1 Интерфейс Document Management

Интерфейсът се предоставя от модул “Управление на документи и преписки”. Основните операции, които включва са:

- Регистрация на иницираща преписка документ;
- Регистрация на документ към съществуваща преписка от документи;
- Приключване на документ или преписка;
- Извличане на данни за документ, включително файлово съдържание.

Интерфейсът се използва от следните модули:

- **Модул Електронно заявление** с цел регистрацията на създадено електронно заявление като иницираща документ и създаване на преписка, в която ще се съхраняват всички документи, създадени в процеса на предоставяне на КАУ;
- **Модул Електронен достъп до документи**, който извлича данни за документ, когато се осъществи достъп до него;



Фиг. 16. Основни модули от архитектура на АИС, реализираща КАО

- **Модул Електронно удостоверение**, който регистрира изготвен резултатен документ в преписката на документ-заявление след приключване на процеса по предоставяне на КАУ. Вторият случай на използване е, когато **Модул Електронно удостоверение** подготви заявление за ВЕАУ, което се изпраща до съответния ПАД чрез **Модула за обмен на електронни документи с администрации**.
- **Модулът за обмен на документи с физически и юридически лица – ССЕВ клиент**, който регистрира изпратени чрез ССЕВ документи до администрацията;
- **Модулът за обмен на електронни документи с администрации – СЕОС клиент**, който регистрира изпратени чрез СЕОС документи до администрацията.

4.3.2 Интерфейс Workflow Provider

Интерфейсът се предоставя от **Модул Управление на работни процеси** и включва следните операции:

- Стартиране на работен процес;
- Преминаване на следваща стъпка от работен процес чрез изпълнение на текущата;
- Прекратяване на работен процес;
- Абониране за известие за преминаване към следваща стъпка от работен процес;
- Справки за изпълнение на работен процес.

Интерфейсът се използва от следните модули:

- **Модул Управление на документи и преписки** с цел стартиране на работен процес след регистрацията на инициращ документ-заявление за КАУ;
- **Модул Електронно удостоверение** – когато работен процес е иницирал изпълнение на скрипт за извличане и анализ на данни от **Модул Електронно удостоверение** и изпълнението на скрипта завърши, модулът иницира преминаване към следваща стъпка от работни процес;
- **Модул Управление на задачи** с цел отчитане края на изпълнение на задача, свързана с работен процес.

4.3.3 Интерфейс E-certificate Processor

Интерфейсът се предоставя от **Модул Електронно удостоверение** и включва следните операции:

- Стартиране процес по изпълнение на скрипт за извличане и анализ на структурирани данни от регистрите на ПАД;
- Прекратяване изпълнението на скрипт за извличане и анализ на структурирани данни от регистрите на ПАД.

Интерфейсът се използва от **Модул Управление на работни процеси**. На практика дадена стъпка от работния процес инициира процеса по изпълнение на скрипт за извличане и анализ на структурирани данни от регистрите на ПАД.

4.3.4 Интерфейс Task Management

Интерфейсът се предоставя от **Модул Управление на задачи** и включва следните операции:

- Създаване на задача към служител или група служители;
- Отчитане изпълнението на задача;
- Анулиране на задача.

Модулите, които използват предоставения интерфейс са:

- **Модул Управление на работни процеси**, който чрез предоставената операция създава задачи, насочени към определени служители на администрацията;
- **Модул Управление на документи и преписки**, чрез който потребителите отчитат изпълнението на поставените им задачи.

4.3.5 Интерфейс Register Management

Интерфейсът се предоставя от **Модул Управление на електронни регистри** и включва следните операции:

- Добавяне на запис към даден регистър;
- Модифициране на данните на съществуващ запис в даден регистър.

Модулът Управление на работни процеси използва предоставения интерфейс с цел управление на данните, съхранявани в даден регистър след края на процеса по предоставяне на КАУ.

4.3.6 Интерфейс RegiX Request

Интерфейсът се предоставя от **Модула за извличане на данни от регистри – RegiX клиент** и включва следните операции:

- Изпълнение на RegiX справка за извличане на данни
- Изпълнение на заявка към друга регистрова система

Интерфейсът се използва от следните модули:

- **Модул Електронно удостоверение** с цел автоматизирано изпълнение на справки с предварително създадени заявки, формирана от определен остров от данни;
- **Модул Управление на работни процеси** с цел изпълнение на справка с участието на служител, който въвежда необходимите данни за формиране на заявката.

4.3.7 Интерфейс СЕОС Request

Интерфейсът се предоставя от **Модула за обмен на електронни документи с администрации – СЕОС клиент** и предоставя следните операции:

- Изпращане на документ към дадена администрация чрез СЕОС;
- Изпращане на запитване за състояние на изпратен документ.

Интерфейсът се използва от следните модули:

- **Модул Електронно удостоверение**, в случаите когато процес по извличане на структурирани данни от ПАД не успее да се изпълни или не са открити данни в регистъра. В този случай модулът изготвя заявление за ВЕАУ, което се изпраща до администрацията ПАД;
- **Модул Управление на работни процеси**, в случаите когато определена стъпка от работен процес представлява изпращане на документ към друга администрация.

4.3.8 Интерфейс ССЕВ Request

Интерфейсът се предоставя от **Модула за обмен на документи с физически и юридически лица – ССЕВ клиент** и предоставя следните операции:

- Изпращане на документ към физическо или юридическо лице;
- Проверка за наличие на профил в ССЕВ на определено физическо или юридическо лице.

Интерфейсът се използва от **Модул Управление на работни процеси**, когато е необходимо до заявителя на услугата да се изпрати документ-резултат от изпълнението на КАУ или друг вид документ.

4.3.9 Интерфейс Е-Разп. Request

Интерфейсът се предоставя от **Модула за електронни разплащания – Е-Разплащания клиент**, който предоставя следната функционалност:

- Създаване на задължение към физическо или юридическо лице;
- Анулиране на създадено задължение.

Интерфейсът се използва от **Модул Управление на работни процеси**, който изпълнява стъпка, в която се определя цената на заявена КАУ. След определяне на цената на заявителя на услугата се създава задължение в системата за електронни разплащания pay.egov.bg

Модулите RegiX клиент, ССЕВ клиент, СЕОС клиент и Е-Разплащания клиент са групирани в компонент, изпълняващ функции за обмен на документи и данни със системите на е-Управление посредством предоставени от всяка от тях програмни интерфейси.

ГЛАВА 5. НАДГРАЖДАНЕ НА ПРОТОКОЛ, ИЗПОЛЗВАН ОТ СЕОС ЗА РЕАЛИЗИРАНЕ НА ИНТЕР-АДМИНИСТРАТИВНИ РАБОТНИ ПРОЦЕСИ

Първата версия на протокола на СЕОС е одобрена през 2012 г., когато КАО все още не е задължително да бъде прилагано, тъй като няма изградена законова рамка. На практика в структурата на обменните данни между АИС на отделните администрации освен информация за изпращания документ липсват допълнителни данни, които по някакъв начин да определят защо се изпраща този документ до администрацията получател. Поради множество причини, въпреки успешно изпълнените тестове за обмен на електронни документи между звена на централната администрация, СЕОС не успява да се наложи. През 2017 г. по инициатива на тогавашната Държавна агенция “Електронно управление”, сега Министерство на електронното управление, проектът е възроден, като целта е възможно в най-кратки срокове да се осигури електронен обмен на документи между централна и местна администрация. На практика промени по протокола и структурата на обменните съобщения не е направена, въпреки че към този момент КАО вече е задължително да се прилага.

По-надолу ще бъдат представени допълнения в протокола на СЕОС, които биха довели до значителна оптимизация в процесите по предоставяне на КАО, в случаите когато определени данни не могат да бъдат извлечени автоматично регистрите на дадена администрация, която е ПАД. Това надграждане на СЕОС би било полезно, не само при предоставяне на услуги в режим на КАО, но би улеснило рутинната работа на служителите в администрацията. Измененията, които ще бъдат предложени в този труд включват два основни подхода – надграждане на централния регистър на СЕОС, който съхранява информация за участниците (администрациите) в електронния обмен на документи, и обогатяване на структурата на обменните данни. На практика протокола на СЕОС, дори и в текущата си първа версия, поддържа пренос на допълнителни данни, които обаче не са стандартизирани. Те предоставят възможност всеки доставчик на АИС и в частност на система за управление на документи и преписки (СЕОС се интегрира именно с тази подсистема на АИС) да използва свои собствени допълнителни полета, които са изключени от валидацията на съобщенията.

5.1. Надграждане на централния регистър на участниците в СЕОС

В актуалната версия на СЕОС централния регистър на участниците съхранява минимално количество данни за всяка администрация, включващи:

- Име на администрацията, което трябва да съвпада с данните от административния регистър;
- Единният идентификационен код (ЕИК), с който администрацията е вписана в Търговски регистър на Р. България. Полето е задължително, което създава определени затруднения, когато самостоятелно звено на администрацията има необходимост от регистрация като самостоятелен участник, а не разполага със собствен ЕИК;
- Уникален идентификатор на участника от тип GUID;
- Сериен номер на транспортен сертификат, с които се осъществява защитена комуникация между отделните участници;
- Допълнителни данни за контакт с администрацията;
- Уеб адрес (URL), който се използва от участниците за да изпращат съобщения – архитектурата на СЕОС е архитектура, ориентирана към услуги.

С цел улеснение на процесите по предоставяне на КАО, а от друга страна и останалите административни процеси, надграждането на регистъра на участниците трябва да включва данни за предоставяните по електронен път услуги и опционални данни за структурата на администрацията – определени служители, длъжности и/или отдели. Последното не е задължително, но би улеснило създаването на споделени работни процеси, които се изпълняват между две или повече администрации, особено когато тези администрации са свързани по между си по такъв начин, че е едната е принципал на другата.

5.1.1 Надграждане на регистъра с предоставяните по електронен път услуги от администрацията

За целта, централизираният регистър на участниците трябва да бъде надграден с програмен интерфейс от вида на API, който приема данни за АУ, които администрацията приема да изпълнява по електронен път. На практика това е списък с услуги, които тя изпълнява като ВЕАУ.

Минималния набор от данните, които описват всяка услуга са:

- Наименование на услугата;

- Идентификатор на услуга от административния регистър (всички услуги от даден тип, предоставяни от различните администрации имат еднакъв идентификатор);
- Уникален идентификатор на услугата в АИС на администрацията, която я предоставя (в случай, че поради някаква причина липсва в административния регистър);
- Срокове за изпълнение на услугата като нормална, бърза и експресна (ако са налични);
- Такси за изпълнение на услугата като нормална, бърза и експресна.

Всяка администрация публикува списък с изпълняваните по електронен път услуги посредством регистъра, а АИС на всички останали администрации могат извлекат данните от него и да имат информация относно дадена администрация кои услуги предоставя по електронен път. Тези данни биха били полезни от една страна за предоставяне на КАО, а от друга – и в ситуации, в които физическо или юридическо лице желае да заяви АУ от дадена администрация, посещавайки друга такава. Напр. гражданин заявява АУ, предоставяна от централна администрация, посредством собствената си общинска администрация или кметство.

5.1.2 Надграждане на регистъра с данни за определени служители, длъжности или отдели в администрацията

По този начин дадена администрация може да определи кои конкретни служители, длъжности (напр. главен секретар, главен счетоводител и др.) или отдели ще бъдат “видими” за останалите администрации. Това ще даде възможност когато към администрацията се изпраща документ, той да бъде адресиран автоматично до конкретен служител. Такава функционалност няма пряко влияние върху КАО, но би автоматизирала други административни процеси, особено когато те се изпълняват в две свързани администрации (принципал и подчинена администрация).

5.2. Надграждане на протокола на СЕОС

Надграждането на протокола на СЕОС включва добавяне на нови структури от данни в съобщенията, обменяни между участниците. Някои от тези структури са елементарни и представляват една единствена стойност, която дори може да бъде използвана и като флаг, променящ по някакъв начин обработката на съобщението, а други имат по-сложна йерархична структура или представляват списъци от обекти. Ще бъде разгледано какви допълнителни данни биха били полезни да бъдат добавени към двете

основни съобщения, обменяни между участниците в СЕОС – съобщения “Заявка за регистрация на документ” и съобщение “Справка за състояние на документ”

5.2.1 Съобщение “Заявка за регистрация на документ”

Базовата версия на съобщението, такава каквато е представена в първа версия на протокола на СЕОС, включва само и единствено данни за изпращания документ (№ и дата на регистрация в администрацията-подател, кратко описание – анотация, списък с приложени файлове, които представляват документа, представен в електронен вид).

С цел оптимизиране на работните процеси могат да бъдат добавени следните допълнителни данни към структурата на съобщението:

- Поле **Начин на обработка на съобщението за регистрация на документ в администрацията-получател (ProcessType)**, което има следните две валидни стойности:
 - ❖ **Regular Process** – когато полето има тази стойност, резултатът от обработката на документа в администрацията-получател не е от значение за администрацията-подател. На практика това са стандартните за СЕОС съобщения;
 - ❖ **Complex Service** – когато полето има тази стойност, резултатът от обработката на документа в администрацията-получател е от голямо значение за администрацията-подател. По тази причина, администрацията-получател на съобщението е задължена, когато изпраща съобщение, че документът е приключен да укаже резултата от неговата обработка. Той може да бъде положителен, което означава, че е изготвен изисквания документ или отрицателен, което означава, че изисканата ВЕАУ не е изпълнена и не е изготвен резултатен документ. При този режим на изпращане на документ, неговият работен процес се спира временно в администрацията-подател, докато не се получи резултат от администрацията-получател.
- Поле **Тип на документа спрямо процеса на предоставяне на услуги (DocumentProcessType)** – на това поле се присвояват предварително определени номенклатурни стойности, които определят какъв е типа на документа в контекста на процеса на предоставяне на услуга. Валидни стойности са:
 - ❖ Заявление за АУ;
 - ❖ Резултантен документ;

- ❖ Документ, с който се изискват корекции в заявлението за АУ или прилагане на допълнителни документи, ако данните в тях не могат да бъдат извлечени от регистър (напр. пълномощно);
- ❖ Документ, с който се изисква допълнително заплащане;
- ❖ Документ, удостоверяващ заплатена такса.
- Поле **Вид на услугата спрямо срок за обработка**, което приема следните стойности:
 - ❖ нормална;
 - ❖ бърза;
 - ❖ експресна.
- Поле **Статус на преписка**, което може да приема номенклатурни стойности, които определят в какъв статус се намира процеса по предоставяне на АУ. Валидни стойности са:
 - ❖ В процес на обработка;
 - ❖ Приключена – изпълнена успешно АУ;
 - ❖ Приключена – неизпълнена АУ;
 - ❖ Очакване на допълнителни документи ;
 - ❖ Очакване на допълнително заплащане.
- Структура **Други номера на документа** представлява списък от регистрационни номера на изпращания документ, когато той е бил обработван от няколко администрации. По този начин служителите от всяка следваща администрация могат да го реферират с номера, под който е бил обработван в предходна администрация. Включва следните полета:
 - ❖ регистрационен номер на документ;
 - ❖ дата на регистрация;
 - ❖ наименование на администрация.
- Флагово поле **Възобнови работния процес в последна стъпка** – този флаг се използва за да укаже как трябва да продължи работния процес в администрацията получател на съобщението;
- Поле **Идентификатор на АУ (ServiceIdentity)** – съдържа идентификатор на услуга от централния регистър на СЕОС, предоставяна от администрацията-получател. При обработката на съобщението се регистрира документ, който инициира работен процес, свързан с конкретната АУ;
- Структура **Списък от служители (ExternalUsers)** – представлява списък от служители в администрацията-получател, които трябва да

извършат определено действие – например за съгласуват съдържанието на документи или да поставят електронен подпис;

- Надграждане на структурата, описваща файловото съдържание на документа. Към нея се добавят следните полета:
 - ❖ Поредна версия на файловото съдържание;
 - ❖ Хеш сума на файловото съдържание;
 - ❖ Конкретен вид на документа, за който се отнася файловото съдържание – свободен текст – напр. *“Удостоверение за настоящ адрес”*.

5.2.2 Съобщение “Справка за състояние на документ”

В базовата версия на протокола на СЕОС, в този вид съобщение се съдържат само следните данни:

- статус на документа от **Модула за управление на документи и преписки** (изчаква да бъде регистриран, регистриран, приключен, отказан да бъде регистриран);
- регистрационен номер и дата на регистрация (ако вече е регистриран);
- причина за отказ от регистрация (ако регистрацията му е отказана).

С цел оптимизиране на работните процеси към структурата на съобщението могат да бъдат добавени следните допълнителни данни:

- Поле **Начин на приключване на документа (CloseResult)** – това поле определя начина, по който е приключило изпълнението на иницираната АУ – т.е. дали е издаден резултатен документ или не. Използва се за да може администрацията-получател да определи как да продължи нейния работен процес. Валидните стойности, които може да приема това поле са:
 - *ServiceCompleted*;
 - *ServiceRefused*;
- Поле **Коментар при приключване на документ (CloseComment)** – това поле е с текстово съдържание, което предоставя допълнително пояснение за начина на приключване на документа;
- Поле **Статус на преписка**, което може да приема номенклатурни стойности, които определят в какъв статус се намира процеса по предоставяне на АУ. Валидни стойности са:
 - ❖ В процес на обработка;
 - ❖ Приключена – изпълнена успешно АУ;
 - ❖ Приключена – неизпълнена АУ;

- ❖ Очакване на допълнителни документи;
- ❖ Очакване на допълнително заплащане.
- Флагово поле **Приключи преписка**, което се използва за инициране на операция по приключване на работата по преписка, съдържаща заявление за АУ в администрация получател;
- Структура **Резултат от обработка на документа от всеки служител (ExternalUsersResponse)** – използва се когато при изпращане на съобщение *“Заявка за регистрация на документ”* е бил добавен списък със служители за обработка на документа. В структурата се съхранява информация как всеки от изброените служители е обработил документа (как го е съгласувал, дали го е одобрил).

Така представените изменения в протокола на СЕОС гарантират, че две администрации могат да разменят документ, заявление за конкретна административна услуга, която да бъде иницирана в администрацията-получател на документа, а на по-късен етап, администрацията-получила заявлението да изпрати обратно документ с изготвения резултат от изпълнението на АУ. Администрацията-подател на заявлението ще разпознае този документ като резултат и ще може да продължи своя работен процес по най-добрия начин.

Примерен интер-административен работен процес е представен на Фиг. 17. Работният процес в Администрация 1 (А1) достига стъпка, в която е необходимо да се иницира споделен работен процес в други две администрации – Администрация 2 (А2) и Администрация 3 (А3).

Последователността на обменените съобщения е следната: А1 изпраща съобщение от тип *“Заявка за регистрация на документ”* към А2 (съобщение № 1 на представената на Фиг. 17 диаграма) с следните допълнителни данни:

- **ProcessType**=ComplexService – което ще изиска от А2 да изпрати резултат от изпълнението на услугата и резултантен документ, ако услугата е изпълнена;
- **DocumentProcessType**=Application – функционален вид на документ *“Заявление”*;
- **ServiceIdentity**={c912a66a-df5d-4cба-a864-a27ec72cb9d2} – идентификатор на заявена услуга в А2, определен централизирания регистър на участниците;

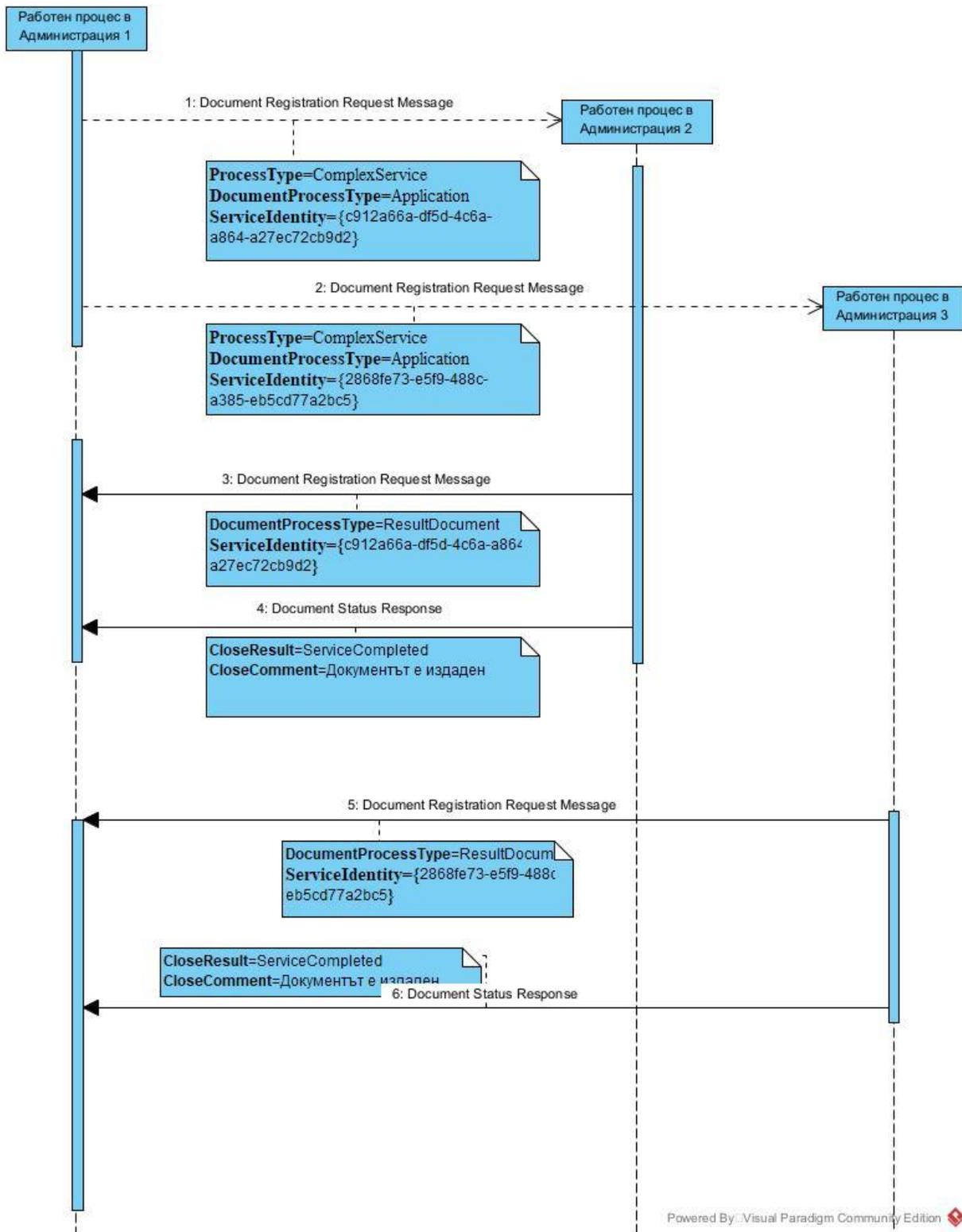
Непосредствено след това се изпраща същото по вид съобщение (съобщение № 2 на диаграмата на Фиг. 17) до А3, със следните допълнителни данни, в които само идентификаторът на услугата в А3 е различен:

- **ProcessType**=ComplexService;
- **DocumentProcessType**=Application;
- **ServiceIdentity**={2868fe73-e5f9-488c-a385-eb5cd77a2bc5}.

След това работният процес в А1 се преминава в състояние на изчакване докато не се получат обратните “съобщения-отговори” от А2 и А3. Когато А2 изпълни изисканата ВЕАУ, тя изпраща две съобщения. Първото представлява “Заявка за регистрация на документ” (съобщение № 3 на Фиг. 17), което съдържа резултатния документ, като за целта са добавени следните допълнително полета:

- **DocumentProcessType**=ResultDocument – изпраща се документ, резултат от изпълнението на ВЕАУ;
- **ServiceIdentity**={c912a66a-df5d-4cба-a864-a27ec72cb9d2} – идентификатор на услугата, за която се изпраща резултатен документ.

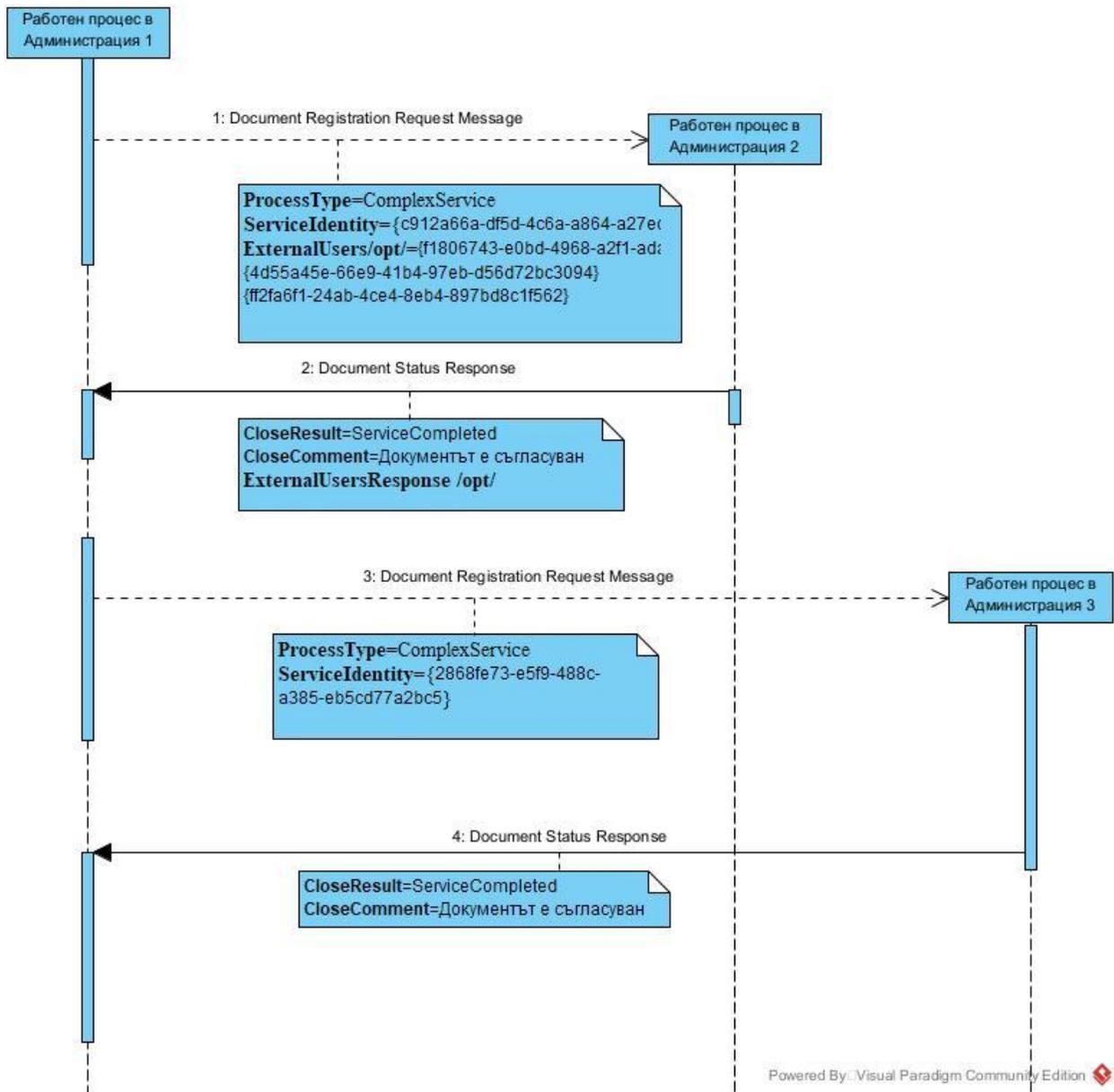
Второто съобщение, което А2 изпраща към А1, е съобщение „Справка за състояние на изпратения документ“, който е изпратен първоначално (съобщение № 4 на диаграмата на Фиг. 17), с което тя “съобщава”, че изисканият документ е предоставен и А1 може да продължи изпълнението на работния си процес, ако всички необходими предпоставки за това са налице. Аналогично А3 изпраща към А1 съобщения с № 5 и №6. На практика едва след тяхното получаване процесът в А1 може да продължи.



Фиг.17. Интер-административен процес, изпълняван в три администрации

Така представеният модел на комуникация може да бъде опростен, когато A2 и A3 не изготвят резултатни документи, а извършват процес на съгласуване (подписване, одобрение) на съдържанието на изпратения документ. В този случай се обменят следните съобщения, представени на диаграмата на Фиг. 18:

- A1 изпраща съобщение № 1, с което инициира работен процес в A2, в който участват определени служители на A2. Изпращането на съобщение към A3 не е синхронно, а изчаква получаване на отговор от A2 (Съобщение № 3).



Фиг. 18 Опростен интер-административен процес за съгласуване на документи в три администрации

- А1 изпраща съобщение № 3, с което инициира работен процес в А3. Изчаква отговор – съобщение № 4 преди да продължи работния си процес.

При този процес, в съобщения от тип *“Справка за състояние на документ”* (съобщения № 2 и № 4) се добавя файловото съдържание на документите, тъй като те са били модифицирани в А2 и А3.

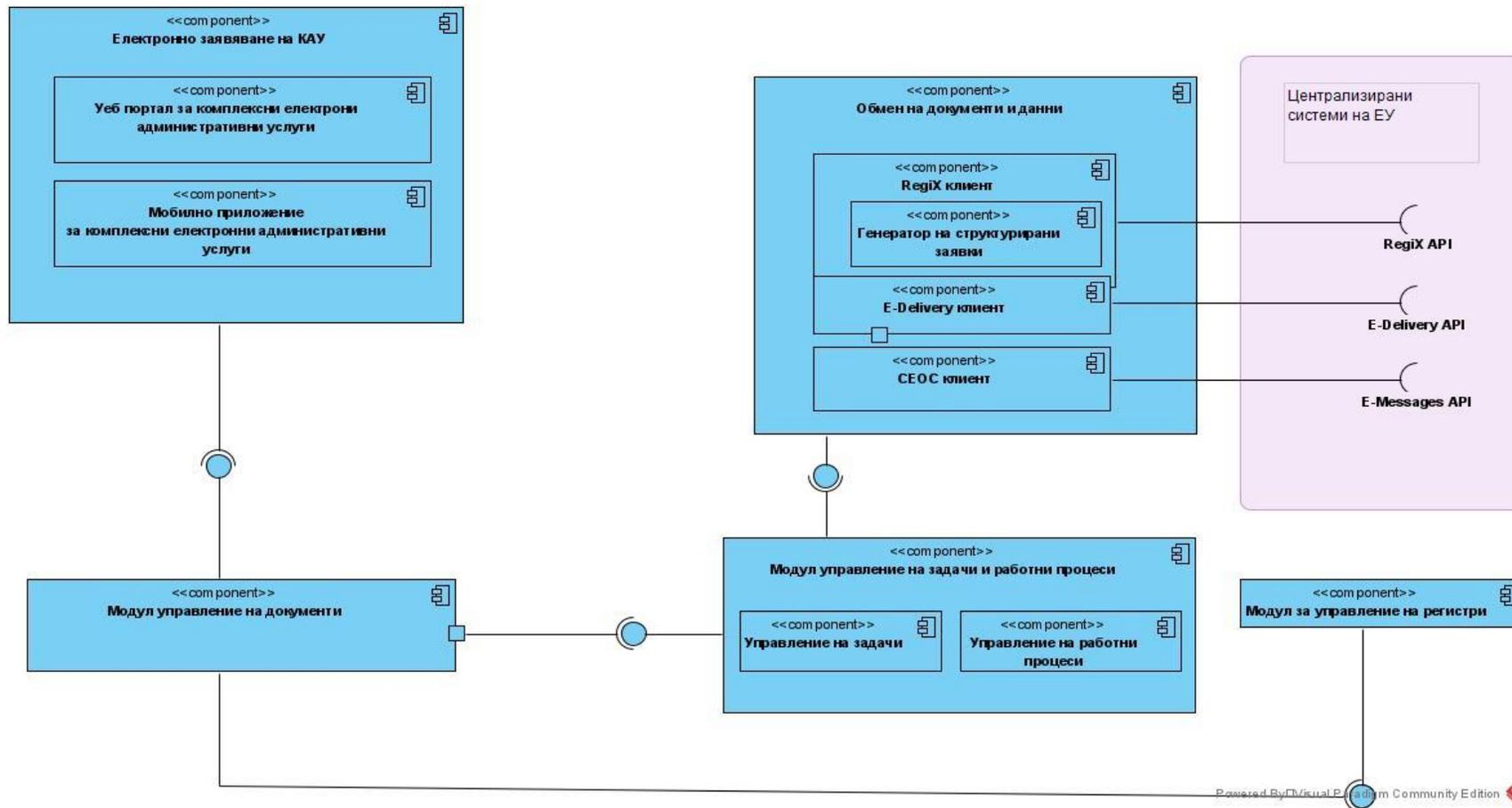
ГЛАВА 6. ВЕРИФИКАЦИЯ НА ИЗГРАДЕНИТЕ МОДЕЛИ

Разработените в този дисертационен труд модели са приложени в реална експлоатация посредством имплементацията им в една от основните АИС, внедрени както в различни държавни администрации (общински, областни, централни) в Р. България, но и в представители на частния сектор. Представената АИС, осигуряваща функционалност за предоставяне на КАО и изпълнение на интер-административни, споделени работни процеси, които се изпълняват в две или повече администрации е проектирана и разработена от Централната лаборатория по Компютърна графика и географски информационни системи към ФПМИ на Технически Университет – София (ЦЛ по КГ и ГИС) и е по-известна под името АКТСЪР ОФИС.

6.1. Архитектура на АИС АКТСЪР ОФИС

Развитието на ЕУ и изграждането на хоризонтални и централизирани ИС за местната и централната власт налага непрекъснато усъвършенстване на софтуерните решения, внедрени и внедрявани в администрацията. Пример за такова решение е представената АИС АКТСЪР ОФИС. На Фиг. 19 е представена нейната архитектура, ориентирана към предоставяне на КАО. Системата заимства аспекти от предложената в този труд референтна архитектура и реализира следните основни модули:

- **Модул за електронно заявяване на комплексна административна услуга;**
- **Модул за управление на документи;**
- **Модул за управление на задачи и работни процеси;**
- **Модулът за обмен на електронни документи с администрации;**
- **Модул за обмен на документи с физически и юридически лица;**
- **Модул за управление на електронни регистри;**



Фиг. 19 Архитектура на АИС АКСТЪР ОФИС

6.1.1. Модул за електронно заявяване на комплексна административна услуга

Модулът се състои от два компонента – уеб портал за заявяване на КАУ и мобилно приложение, което всеки потребител, който желае да заяви КАУ чрез мобилното си устройство, инсталирано на него. Мобилното приложение използва back-end функционалността на уеб портала. По този начин двата компонента използват общи подмодули за интеграция с останалите елементи на АИС. Модулът за електронно заявяване на КАУ представлява потребителски интерфейс под формата на уеб портал или мобилно приложение, с което физическите и юридическите лица могат да попълват, подписват и подават в електронна среда заявления за КАУ. Модулът притежава вградени механизми за автентикация на потребителите, възможност за избор на желаната АУ, която да бъде заявена по електронен път, и извършва множество проверки за валидност на въведените данни. Също така предоставя възможност и за предварително заплащане на заявената услуга. След финализиране на процеса по заявяване на АУ, формира XML структура от данни “остров електронно заявление”, която се предоставя на модул “Управление на документи”, чрез предоставен от него програмен интерфейс.

6.1.2. Модул за управление на документи

Този модул предоставя програмен интерфейс посредством който **Модул за електронно заявяване на комплексна административна услуга** може да регистрира подадените електронни заявления директно в базата данни на подсистемата за управление на документи и преписки. Посредством този интерфейс **Модул за електронно заявяване на комплексна административна услуга** “следи” статуса на изпълнение на заявените услуги в реално време.

Основните операции, предоставени от програмния интерфейс са:

- Регистрация на заявление за КАУ в **Модула за управление на документи**. Създава се преписка, в която се регистрират всички последващи документи, създадени в процеса на изпълнение на КАУ. Резултатът изпълнението на операцията е уникален идентификатор на създадената преписка;
- Извличане на текущо състояние на процеса по изпълнение на КАУ чрез предоставяне на уникален идентификатор на преписка;

- Промяна на състоянието на процеса по изпълнение на КАУ чрез предоставяне на уникален идентификатор на преписка;
- Извличане на списък с всички документи, част от преписка за предоставяне на КАУ
- Извличане на съдържание на документ и неговите атрибути, включен в преписка за изпълнение на КАУ чрез идентификатор на документа

6.1.3. Модул за управление на задачи и работни процеси

Модулът предоставя функционалност за дефиниране и изпълнение на работни процеси за всяка предоставяна чрез АИС АКСТЪР ОФИС административна услуга. Всеки работен процес се състои от отделни стъпки, изпълнението на които се отчита чрез задачи. По този начин всяка стъпка са асоциира с една или повече задачи. Всяка задача може да бъде насочена към конкретен служител от администрацията или към компонент на АИС. По този начин работните процеси могат да създават “задачи” към системата, за изпълнение на дейности като:

- създаване на заявки за изпращане на документи към други администрации;
- създаване на заявки за изпращане на документи към физически и юридически лица;
- изпълнение на справки от регистри чрез средата за междурегистров обмен.

6.1.3.1. Създаване на заявки за изпращане на документи към други администрации

Модулът, имплементиращ управление на задачи и работните процеси предоставя следните възможности:

- автоматизирано изпращане на документ към друга администрация посредством насочване на документ към външен служител;
- автоматизирано изпращане на документ към друга администрация чрез специален вид стъпка в работния процес.

Изпращане чрез насочване към служител – когато модулът създаде задача към служител, външен за администрацията (асоцииран с друга администрация), в този момент се създава заявка за изпращане съдържанието на документа към администрацията на служителя посредством Модул СЕОС клиент.

Изпращане чрез изпълнение на специален вид стъпка в работния процес – при конфигуриране на работните процеси, една от видовете стъпки е наречена *“Външна услуга”*. Тя предоставя възможност да се определят:

- Администрацията, до която да се изпрати документа – възможно е да се определи не конкретна администрация, а правило, с което да се определя администрацията до която ще се изпрати документът;
- Електронна услуга (работен процес), чието изпълнение да бъде иницирано в администрация-получател на документа;

Модулът за управление на задачи и работни процеси създава заявка за изпращане съдържанието на документа към определената администрация.

6.1.3.2. Създаване на заявки за изпращане на документи към физически и юридически лица

В края на работния процес по предоставяне на АУ, може да се добави специален вид стъпка, чрез която резултатния документ от изпълнението на услугата да бъде изпратен до физическото или юридическото лице, заявило услугата. В този случай, тъй като работния процес е асоцииран с документ-заявление, то АИС АКСТЪР ОФИС проверява преписката, сформирана за изпълнение на услугата и търси документ с функционален вид *“Резултат от изпълнение на услуга”*. Ако такъв документ бъде открит, то той се добавя в опашка за изпращане от Модул СЕОС клиент.

6.1.3.3. Изпълнение на справки от регистри чрез средата за междурегистров обмен

Всеки работен процес в АИС АКСТЪР ОФИС се асоциира с набор от справки, изпълнявани чрез средата за междурегистров обмен RegiX, чиито резултати от изпълнението са необходими за предоставяне на конкретна АУ.

В Табл. 2 са представени реални данни с направените асоциации в АИС АКСТЪР ОФИС, внедрена в Община Троян.

Табл. 2. Административни услуги и необходими справки за тяхното изпълнение

УСЛУГА	СПРАВКИ
Разрешение за отсичане и изкореняване над 5 дървета (КАУ)	Нотариален акт - сканирано копие, Справка по персонална партида на юридическо лице, Справка по персонална партида на физическо лице, Справка за семейно положение , Справка за родственоост
Удостоверение за въвеждане в експлоатация по чл.177,ал.1 и ал.3 от ЗУТ на строеж от IV и V категория	Нотариален акт - сканирано копие
Допускане изработването на проекти за изменение на ПУП	Нотариален акт - сканирано копие
Издаване на скица за недвижим имот (КАУ)	Справка за родственоост, Нотариален акт - сканирано копие, Справка по персонална партида на физическо лице
Издаване на удостоверение, че обособените дялове или части отговарят на одобрени за това ИП	Нотариален акт - сканирано копие
Ломец-Адресна карта	Нотариален акт - сканирано копие
Шипково-Адресна карта	Нотариален акт - сканирано копие
Старо село-Адресна карта	Нотариален акт - сканирано копие
Разрешение за достъп до горски територии по чл.148 от ЗГ	Нотариален акт - сканирано копие
Издаване на разрешение за строеж без одобряване на инвестиционни проекти в условията на чл.147 от ЗУТ	Нотариален акт - сканирано копие
Адресна карта за настоящ адрес	Справка по персонална партида на физическо лице
Разрешение за отсичане и изкореняване до 5 дървета	Нотариален акт - сканирано копие, Справка по персонална партида на юридическо лице, Справка по персонална партида на физическо лице, Справка за семейно положение, Справка за родственоост

Категоризация на заведения за хранене и развлечения (КАУ)	Справка по персонална партида на юридическо лице, Справка по персонална партида на физическо лице
Чифлик-Искане за издаване на удостоверение за настоящ адрес след подаване на адресна карта за заявяване или промяна на настоящ адрес	Нотариален акт - сканирано копие
Одобряване на КПИИ по чл. 150 от ЗУТ	Нотариален акт - сканирано копие
Калейца-Искане за издаване на удостоверение за ПА при заявяване на нов	Нотариален акт - сканирано копие
Издаване на констативни протоколи и удостоверение за степен на завършеност на строеж (КАУ)	Нотариален акт - сканирано копие
Дебнево-Искане за издаване на удостоверение за ПА при заявяване на нов	Нотариален акт - сканирано копие
Издаване на разрешение за поставяне на преместваеми обекти	Нотариален акт - сканирано копие
Издаване на виза за проектиране по чл.140, ал.1 от ЗУТ	Нотариален акт - сканирано копие
Одобряване на ПУП	Нотариален акт - сканирано копие
Закупуване на недвижим имот-ОС,който действащия устройствен план е придвидил за придаване към съседен имот	Нотариален акт - сканирано копие
Белиш - Адресна карта	Нотариален акт - сканирано копие
Удостоверение, че имота не е актуван като общинска собственост (КАУ)	Нотариален акт - сканирано копие
Даване на предварително съгласие за прокарване на съоръжения на техническата инфраструктура	Нотариален акт - сканирано копие
Издаване на разрешително за водовземане от повърхностен воден обект-язовир, публична ОС, чрез изградени съоръжения	Нотариален акт - сканирано копие
Издаване на карта за безплатно паркиране на МПС за хора с увреждания	Справка за всички експертни решения на лице,

	Справка за последното експертно решение на лице
Терзийско-Издаване на удостоверение за ПА, при заявяване на нов	Нотариален акт - сканирано копие
Дълбок Дол-Издаване на удостоверение за ПА, при заявяване на нов	Нотариален акт - сканирано копие
Горно Трапе-Издаване на удостоверение за ПА, при заявяване на нов	Нотариален акт - сканирано копие
Издаване на удостоверение за идентичност на УПИ	Нотариален акт - сканирано копие, Справка по персонална партида на физическо лице
Гумощник-Адресна карта	Нотариален акт - сканирано копие
Бели Осъм-Искане за издаване на удостоверение за ПА, при заявяване на нов	Нотариален акт - сканирано копие
Удостоверения по факти и обстоятелства по ТСУ	Нотариален акт - сканирано копие, Справка по персонална партида на физическо лице
Борима-Адресна карта	Нотариален акт - сканирано копие
Разрешително за ползване на плувен басейн през летния период (КАУ)	Справка за актуално състояние(v1), Нотариален акт - сканирано копие, Справка за вписано юридическо лице с нестопанска цел, Справка по персонална партида на юридическо лице, Справка по персонална партида на физическо лице
Одобряване на схема по чл.56 и чл. 57 от ЗУТ за поставяне на преместваеми съоръжения	Нотариален акт - сканирано копие
Бели Осъм-Адресна карта	Нотариален акт - сканирано копие
Издаване удостоверение за търпимост на строеж	Нотариален акт - сканирано копие, Справка по персонална партида на физическо лице, Справка по персонална партида на юридическо лице
Категоризация на места за настаняване (КАУ)	Нотариален акт - сканирано копие, Справка по персонална партида на физическо лице,

	Справка по персонална партида на юридическо лице
Заявление за промяна на постоянен адрес, след подаване на заявление	Справка по персонална партида на физическо лице
Издаване на удостоверение за билки, събирани от култивирани лечебни растения	Нотариален акт - сканирано копие, Справка по персонална партида на физическо лице, Справка по персонална партида на юридическо лице
Издаване на разрешение за отсичане на дървета от зелената система на Община Троян (КАУ)	Нотариален акт - сканирано копие , Справка по персонална партида на юридическо лице, Справка по персонална партида на физическо лице, Справка за валидност на физическо лице, Справка за съпруг/съпруга, Справка за родственост , Справка за семейно положение
Балканец-Адресна карта	Нотариален акт - сканирано копие
Врабево-Издаване на удостоверение за ПА, при заявяване на нов	Нотариален акт - сканирано копие
Заявление за административен адрес на поземлени имоти	Нотариален акт - сканирано копие, Справка по имот
Старо село-Искане за издаване на удостоверение за ПА, при заявяване на нов	Нотариален акт - сканирано копие
Голяма Желязна-Искане за издаване на удостоверение за постоянен адрес за заявяване или за промяна на постоянен адрес	Нотариален акт - сканирано копие
Дебнево-Адресна карта за настоящ адрес	Нотариален акт - сканирано копие
Промяна в обстоятелствата за категоризирани обекти - удостоверение (КАУ)	Нотариален акт - сканирано копие, Справка по персонална партида на физическо лице, Справка за вписано юридическо лице с нестопанска цел, Справка за вписани обстоятелства в Регистъра на производствата по несъстоятелност,

	Справка по персонална партида на юридическо лице
Одобряване на инвестиционен проект при промяна на инвестиционните намерения след издаването на разрешението за строеж	Нотариален акт - сканирано копие
Издаване на пропуски за влизане в зоните и улиците ,ограничени за движение на ППС	Справка за настоящ адрес, Справка по персонална партида на юридическо лице, Справка за МПС по регистрационен номер, Разширена справка за МПС по регистрационен номер
Орешак - Адресна карта	Нотариален акт - сканирано копие
Гумошник-Издаване на удостоверение за ПА, при заявяване на нов	Нотариален акт - сканирано копие
Горно Трапе-Адресна карта	Нотариален акт - сканирано копие
Издаване на разрешение за изработване на ПУП	Нотариален акт - сканирано копие
Издаване на разрешение за строеж в поземлени имоти-зеделски земи без промяна на предназначението	Нотариален акт - сканирано копие
Добродан-Искане за издаване на удостоверение за ПА , при заявяване на нов	Нотариален акт - сканирано копие
Заверка на копие на документ от технически архив	Нотариален акт - сканирано копие
Шипково-Издаване на удостоверение за ПА, при заявяване на нов	Нотариален акт - сканирано копие
Добродан-Адресна карта	Нотариален акт - сканирано копие
Чифлик-Искане за издаване на удостоверение за постоянен адрес за заявяване или за промяна на постоянен адрес	Нотариален акт - сканирано копие
Голяма Желязна-Искане за издаване на удостоверение за настоящ адрес след подаване на адресна карта за заявяване или промяна на настоящ адрес	Нотариален акт - сканирано копие
Ломец-Издаване на удостоверение за ПА, при заявяване на нов	Нотариален акт - сканирано копие

Изменение и/или продължаване или преиздаване на разрешително за водоземане	Нотариален акт - сканирано копие
Калейца-Адресна карта	Нотариален акт - сканирано копие
Борима-Издаване на удостоверение за ПА, при заявяване на нов	Нотариален акт - сканирано копие
Заявление за издаване на разрешение за промяна предназначението на сграда или самостоятелен обект в сграда без извършването на строителни и монтажни работи	Нотариален акт - сканирано копие
Издаване на разрешително за водоземане чрез съществуващи съоръжения от минерална вода	Нотариален акт - сканирано копие
Разрешение за изработване на комплексен проект за инвестиционна инициатива	Нотариален акт - сканирано копие
Черни Осъм - Издаване на удостоверение за постоянен адрес, за заявяване или за промяна на постоянен адрес	Нотариален акт - сканирано копие
Балканец-Издаване на удостоверение за ПА, при заявяване на нов	Нотариален акт - сканирано копие
Белиш-Издаване на удостоверение за ПА, при заявяване на нов	Нотариален акт - сканирано копие
Орешак - Издаване на удостоверение за ПА, при заявяване на нов	Нотариален акт - сканирано копие
Съгласуване и одобряване на ИП, по които се издава РС	Нотариален акт - сканирано копие
Дълбок Дол-Адресна карта	Нотариален акт - сканирано копие
Черни Осъм - Издаване на удостоверение за настоящ адрес след подаване на адресна карта за заявяване или промяна на настоящ адрес	Нотариален акт - сканирано копие
Приемане (поправка) на помощен план по чл.13а, ал.1 и 2 от ППЗСПЗЗ (КАУ)	Нотариален акт - сканирано копие
Разрешение за таксиметров превоз на пътници (КАУ)	Справка за извършен технически преглед по регистрационен номер на автомобил,

	Справка за наличие/ липса на задължения , Справка за актуално състояние на всички/действащите трудови договори
Одобряване на инвестиционен проект	Нотариален акт - сканирано копие
Врабево-Адресна карта	Нотариален акт - сканирано копие
Заявление-декларация за регистриране за упражняване на дейност хотелиерство в стая за гости или апартамент за гости	Нотариален акт - сканирано копие , Справка по персонална партида на юридическо лице, Справка по персонална партида на физическо лице
Терзийско-Адресна карта	Нотариален акт - сканирано копие

От данните в таблицата се вижда, че услугите, които общината предоставя като КАУ са асоциирани с множество справки. Например услугата *Разрешение за таксиметров превоз на пътници (КАУ)* изисква попълване само заявление от страна лицето, което я заявява, без да е необходимо прилагане на допълнителни документи. Необходимите данни се извличат посредством изпълнение на следните справки:

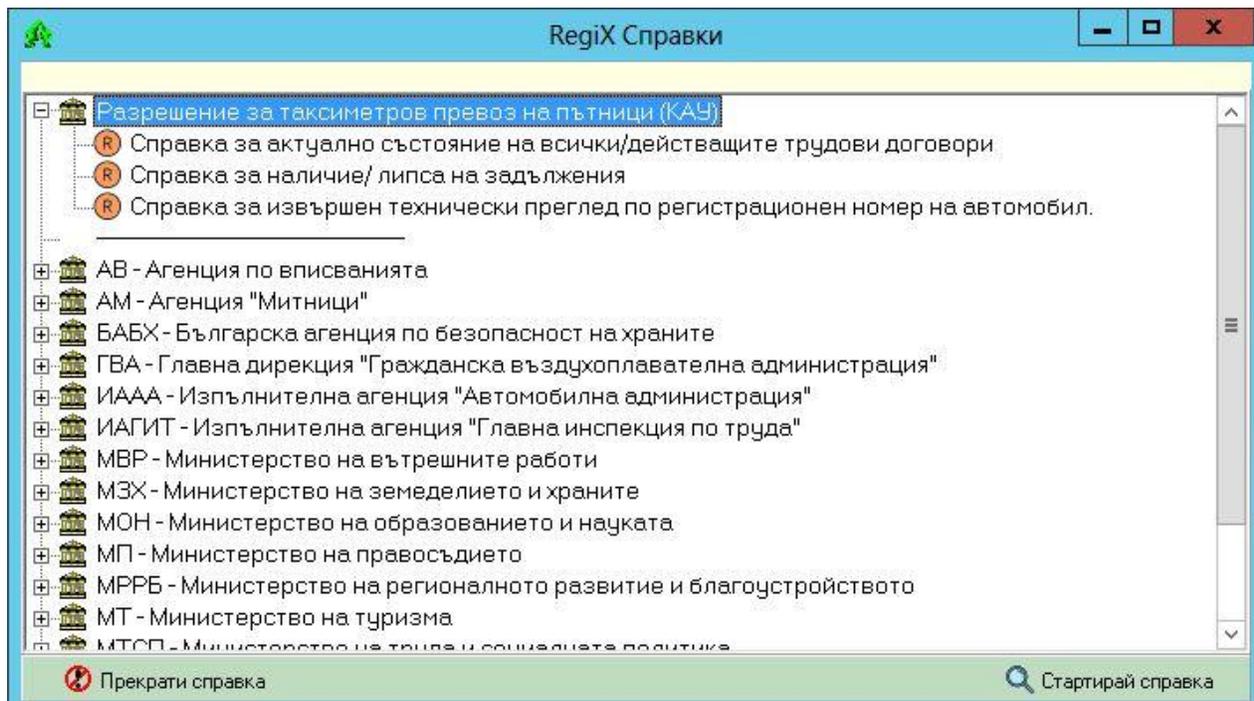
- Справка за извършен технически преглед по регистрационен номер на автомобил;
- Справка за наличие/ липса на задължения;
- Справка за актуално състояние на всички/действащите трудови договори.

За останалите услуги, които не се предоставят като КАУ, за улеснение на заявителите, част от изискуемите документи се изтеглят автоматично като данни или документи от регистрите на ПАД – такъв документ например е сканирано копие на нотариален акт. Чрез тази услуга на Агенцията по вписванията, която предоставя в електронен вид нотариален акт, достъпен за изтегляне по номер и дата на издаване, отпада необходимостта гражданите да предоставят копие на нотариалния си акт, удостоверяващ собственост върху имот, обект на предоставяната услуга.

Интеграцията на АИС със средата за междурегистров обмен RegiX се осъществява със специализирания Модул RegiX клиент, който имплементира в себе си подмодул за генериране на структурирани документи (XML документи), които съгласно описания в настоящия дисертационен труд модел на работа на АИС представляват острови от данни, използвани за създаване на заявки към регистрови системи. Резултатът от изпълнението на справка може също да се разглежда като остров от данни, който в

тази версия на АИС автоматично се трансформира в HTML документ, удобен за разглеждане от служителите на администрацията.

На Фиг. 20 е представен интегриран в потребителския интерфейс на АИС Модул RegiX клиент.



Фиг. 20 Интеграция на Модул RegiX клиент в АИС АКСТЪР ОФИС

От представения диалог се виждат справките, които са асоциирани с избраната административна услуга. В тази версия на АИС справките се инициират от служител и се изпълняват чрез диалоговия прозорец, показан на Фиг. 21.

Фиг.21 Създаване на RegiX заявка за справка

Както се вижда на фигурата, основните данни в горната област на диалога, съпътстващи всяка справка в RegiX, са попълнени автоматично. Данните в бялото поле, които изграждат специфичната структура на заявката се попълват чрез удобен потребителски интерфейс, който се генерира автоматично за всяка справка на базата на метаданни за структурата на заявката ѝ. За целта се използва XSD схема, описваща структурата на заявката, която определя кои полета са задължителни и какви стойности трябва да бъдат попълнени в тях – от номенклатура, свободен текст, цяло, дробно число или дата.

Резултатът от изпълнението на справката се представя чрез трансформиране на получения XML документ в HTML файл, който след това се визуализира на потребителя, както е показано на Фиг. 22.

Заявка Отговор

Трудов договор:

ЕГН/ЕИК/Сл. номер/БУЛСТАТ:
831917834

Наименование:
ТЕХНИЧЕСКИ УНИВЕРСИТЕТ /Висше училище/ гр.СОФИЯ, кв. ДЪРВЕНИЦА ул. КЛИМЕНТ ОХРИДСКИ 8

ЕГН:
[REDACTED]

Имена на лицето:
ВИКТОР ВЕНЦИСЛАВОВ ГЛАВЕВ

Дата на сключване:
2019-09-19

Основание по приложение №1 от Наредба №5:
02

Срок:
2023-09-24

Код КИД:
8542

Код НКЦД:
23107002

Длъжност наименование:
Асистент висше училище

Код по ЕКАТТЕ:
68134

Фиг. 22 Отговор на RegiX заявка

Структурата на заявката, изпратена към средата за междурегистров обмен RegiX, за конкретната справка представлява следния XML документ, показан на Фиг. 23.

```

<?xml version="1.0" ?>
<ServiceRequestData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Operation xmlns="http://tempuri.org/">TechnoLogica.Regix.NRAEmploymentContractsAdapter.
  APIService.INRAEmploymentContractsAPI.GetEmploymentContracts</Operation>
  <Argument xmlns="http://tempuri.org/">
    <EmploymentContractsRequest xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://egov.bg/Regix/NRA/EmploymentContracts/Request">
      <Identity xmlns="http://egov.bg/Regix/NRA/EmploymentContracts/Request">
        <ID>1111111110</ID>
        <TYPE>EGN</TYPE>
      </Identity>
      <ContractsFilter>All</ContractsFilter>
    </EmploymentContractsRequest>
  </Argument>
  <CallContext xmlns="http://tempuri.org/">
    <ServiceURI>B-5035</ServiceURI>
    <ServiceType>Административна услуга</ServiceType>
    <EmployeeIdentifier>grao@troyan.bg</EmployeeIdentifier>
    <EmployeeNames>Acstre Support</EmployeeNames>
    <EmployeePosition>Гл. експерт</EmployeePosition>
    <AdministrationOid>2.16.100.1.С.З.Х.У</AdministrationOid>
    <AdministrationName>Община Троян</AdministrationName>
    <LawReason>чл.24а от ЗАП</LawReason>
    <Remark />
  </CallContext>
  <CitizenEGN xmlns="http://tempuri.org/">1111111110</CitizenEGN>
  <SignResult xmlns="http://tempuri.org/">true</SignResult>
  <ReturnAccessMatrix xmlns="http://tempuri.org/">false</ReturnAccessMatrix>
</ServiceRequestData>

```

Фиг. 23 XML Структура на Regix заявка

Резултатът от изпълнението на избраната справка, с премахнати данни за електронния подпис , е показан на Фиг. 24.

```

<EmploymentContractsResponse xmlns="http://egov.bg/RegiX/NRA/EmploymentContracts/Response">
  <Identity>
    <ID xmlns="http://egov.bg/RegiX/NRA/EmploymentContracts">111111110</ID>
    <TYPE xmlns="http://egov.bg/RegiX/NRA/EmploymentContracts">EGN</TYPE>
  </Identity>
  <EContracts>
    <EContract xmlns="http://egov.bg/RegiX/NRA/EmploymentContracts">
      <ContractorBulstat>8319178340012</ContractorBulstat>
      <ContractorName>НАУЧНО-ИЗСЛЕДОВАТЕЛСКИ СЕКТОР ПРИ ТЕХНИЧЕСКИ УНИВЕРСИТЕТ /поделение/
гр.СОФИЯ, кв. ДЪРВЕНИЦА бл.1 ет.2</ContractorName>
      <IndividualEIK>111111110</IndividualEIK>
      <IndividualNames>ВИКТОР ВЕНЦИСЛАВОВ ГЛАВЕВ</IndividualNames>
      <StartDate>2006-03-06</StartDate>
      <LastAmendDate>2017-04-01</LastAmendDate>
      <EndDate>2019-09-24</EndDate>
      <Reason>03</Reason>
      <TimeLimit>2017-05-31</TimeLimit>
      <EcoCode>7219</EcoCode>
      <ProfessionCode>25116004</ProfessionCode>
      <ProfessionName>ПРОЕКТАНТ ИНФОРМ.СИСТЕМИ</ProfessionName>
      <ЕКАТТЕCode>68134</ЕКАТТЕCode>
    </EContract>
    <EContract xmlns="http://egov.bg/RegiX/NRA/EmploymentContracts">
      <ContractorBulstat>831917834</ContractorBulstat>
      <ContractorName>ТЕХНИЧЕСКИ УНИВЕРСИТЕТ /Висше училище/
гр.СОФИЯ, кв. ДЪРВЕНИЦА ул. КЛИМЕНТ ОХРИДСКИ 8</ContractorName>
      <IndividualEIK>111111110</IndividualEIK>
      <IndividualNames>ВИКТОР ВЕНЦИСЛАВОВ ГЛАВЕВ</IndividualNames>
      <StartDate>2019-09-19</StartDate>
      <Reason>02</Reason>
      <TimeLimit>2023-09-24</TimeLimit>
      <EcoCode>8542</EcoCode>
      <ProfessionCode>23107002</ProfessionCode>
      <ProfessionName>Асистент висше училище</ProfessionName>
      <ЕКАТТЕCode>68134</ЕКАТТЕCode>
    </EContract>
  </EContracts>
  <Status>
    <Code xmlns="http://egov.bg/RegiX/NRA/EmploymentContracts">0</Code>
  </Status>
  <ContractsFilter>All</ContractsFilter>
  <ReportDate>2023-06-27T00:00:00</ReportDate>
</EmploymentContractsResponse>

```

Фиг. 24 XML Структура отговор на RegiX на заявка

6.1.4. Модул за управление на електронни регистри

Модулът за управление на електронни регистри предоставя възможност за създаване и управление на електронни регистри в АИС АКСТЪР ОФИС. Той използва програмен интерфейс за извличане на документи, отговарящи на определени критерии, предоставен от **Модула за управление на документи**. **Модулът за управление на електронни регистри** извършва анализ на структурирани данни, приложени като файлово съдържание към всеки документ част от регистъра. След извършване на анализ, съответния документ се причислява към определен регистър, а неговите структурирани данни се използват за попълване атрибутите на регистъра.

6.2. Реализация на интер-административни, споделени работни процеси в АИС АКТСЪР ОФИС

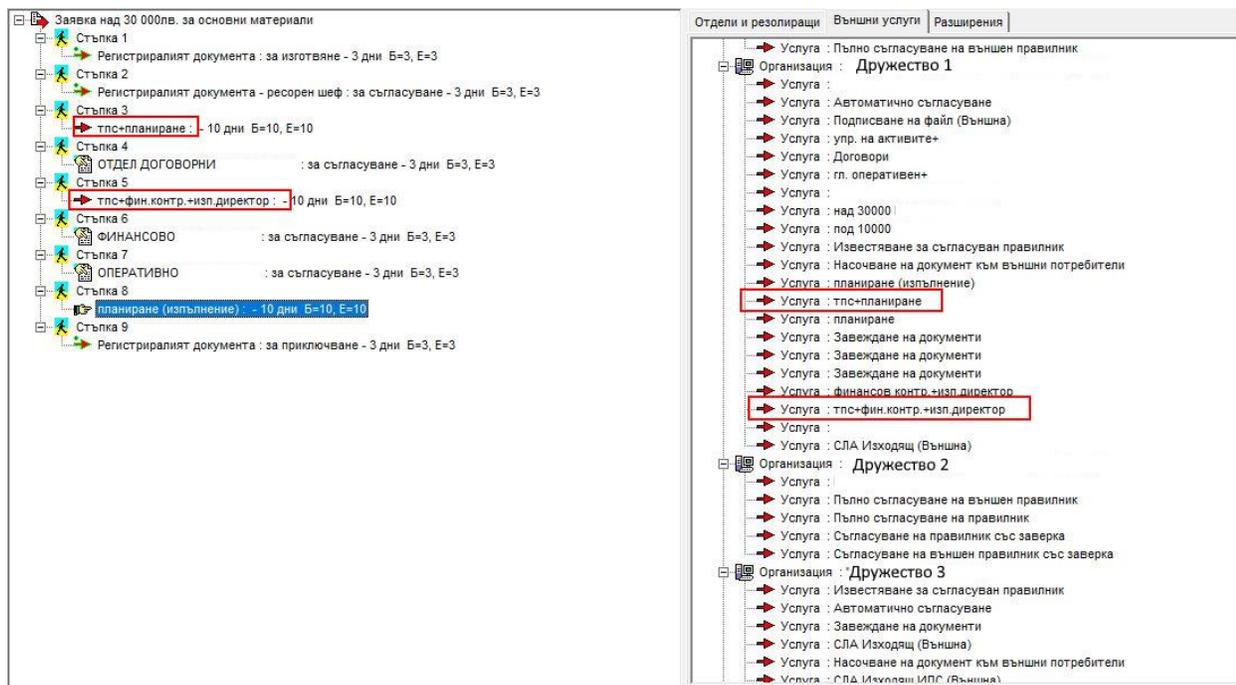
Имплементацията на интер-административни, споделени работни процеси е реализирана с цел удовлетворяване на няколко функционални изисквания, поставени от голяма акционерна компания, оперираща чрез няколко самостоятелни дружества на територията на Р. България. На практика всяко от дружествата може да се разглежда като самостоятелна администрация, в която се изпълняват определени работни процеси и разполага със собствена инстанция на АИС. Всяка инстанция използва самостоятелна база данни.

Поставените изисквания към интер-административните споделени работни процеси са:

- едно дружество Д1 създава електронен документ и инициира работен процес, който насочва документа към определени служители в същото дружество;
- след определена стъпка, работният процес продължава в друго дружество Д2, като служителите в него работят с актуалното съдържание на документа към момента на “излизането” му от Д1. При постъпване на документа в Д2 се реализира интер-административен, споделен работен процес;
- работният процес в Д2 може да завърши по няколко начина:
 - ❖ с цялостно рестартиране на процеса в Д1;
 - ❖ с връщане към последно изпълнената дейност в Д1;
 - ❖ с връщане към Д1 с цел продължаване на работния процес, определен от Д1;
- когато документът се намира в едно дружество, останалите дружества, в които той е бил обработван, не могат да извършват корекции по документа (напр. да редактират файловото му съдържание);
- всяко дружество трябва да може да проследи направените промени по файловото съдържание в друго дружество;
- след връщането си в Д1 документът продължава обработката си от служители в Д1, като в определен момент може да бъде изпратен отново в Д2 или друго дружество, част от акционерната компания. Там обработката му може да завърши отново по един от трите изброени по-горе начина;
- важно условие е, че обработката на един документ не може да се изпълнява паралелно в няколко дружества.

За реализация на така поставените изисквания е използван предложението в този дисертационен труд модел за надграждане на протокола на СЕОС, осигуряващо имплементиране на интер-административни, споделени работни процеси.

Примерна конфигурация на работен процес в едно от дружествата е представена на Фиг. 25



Фиг. 25 Примерна конфигурация на работен процес в АИС на Дружество 1

От представения на Фиг. 25 диалогов прозорец се вижда, че стъпки № 3 и № 5 от показания работен процес **Заявка над 30 000 лв. за основни материали** са специален вид действия, наречени *“Външни услуги”*. Когато **Модулът за управление на задачи и работни процеси** достигне до някоя от тези стъпки, той инициира автоматизирано изпращане на документа към Дружество 1, където са дефинирани представените на Фиг. 25 маркирани в червен цвят работни процеси. Когато процесът **ТПС+Планиране** завърши своето изпълнение в Дружество 1, начинът по който е завършило изпълнението му там ще определи как ще продължи изпълнението на основния процес. Това зависи от избрания отговор на последно изпълнената стъпка от работния процес в Дружество 1. На Фиг. 26 е показана дефиницията на процеса **ТПС+Планиране**.

Избор	Подсказка	Активна	Следваща Стъпка	"И" група	Всички от стъпката	Икона
Съгласувано		<input type="checkbox"/>		0	<input type="checkbox"/>	
Връщане към Изготв		<input type="checkbox"/>		0	<input type="checkbox"/>	
За корекция		<input checked="" type="checkbox"/>	1 за съгласуване	0	<input type="checkbox"/>	

- Изборът е отрицателен = "Отказ"
- Генерирай следваща стъпка без да приключваш текущата (не се ползва) Оригиналите към документа трябва да се подпишат
- Рестартиране на услугата при организацията-изпращач
- Връщане за корекция към организация-изпращач (връща една стъпка назад)
- Връщане на отговор към организация-изпращач (продължава напред)

Фиг. 26 Примерна конфигурация на работен процес в АИС на Дружество 2

От фигурата се вижда се че процесът се състои от две стъпки, които насочват обработката на документа към различни отдели или дирекции от структурата на дружеството Д1. В долната част на екрана е показано, че втората стъпка с текст "За съгласуване" има три възможни отговора, от които служителят, обработващ документа трябва да избере само един:

- Съгласувано;
- Връщане към Изготвил;
- За корекция.

В зависимост от избраната конфигурация от списъка с предварително определени настройки за всеки отговор (областта от прозореца оградена в червен правоъгълник) се определя как ще продължи основния работен процес, в дружество първоизточник на документа, като възможностите са следните:

- ще премине на следваща стъпка 4, където документът ще се насочи за изпълнение от съответния отдел или дирекция;
- ще премине към предходната стъпка 2;
- ще рестартира изпълнението на процеса в стъпка 1.

Структурата на обменените съобщения по разширения протокол на СЕОС в процеса на комуникация между отделните дружества са показани на Фиг. 27 и Фиг. 28.

```

<DocumentRegistrationRequest>
  <Document>
    <DocID>
      <DocumentNumber>
        <DocNumber>XXX-COMM-44-[1]</DocNumber>
        <DocDate>2023-06-13</DocDate>
      </DocumentNumber>
      <DocumentGUID>{165501fb-2cf2-4f8f-857c-4ea6ef9ffb7e}</DocumentGUID>
    </DocID>
    <DocParentID>
      <DocumentNumber>
        <DocNumber>XXX-COMM-44</DocNumber>
        <DocDate>2023-06-13</DocDate>
      </DocumentNumber>
      <DocumentGUID>{165501fb-2cf2-4f8f-857c-4ea6ef9ffb7e}</DocumentGUID>
    </DocParentID>
    <DocKind />
    <DocAbout>sdfds</DocAbout>
    <DocService>
      <ServiceName />
      <ServiceType />
      <ServiceCode />
    </DocService>
    <DocComment />
    <DocAddData>
      <ServiceIdentity xmlns="">7d096212-3894-4d67-ab7f-ccc374755376</ServiceIdentity>
      <ProcessType xmlns="">ComplexService</ProcessType>
      <StartServiceAtLastStep xmlns="">False</StartServiceAtLastStep>
      <DocumentExternalNumbers xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi=
        <DocumentExternalNumber>
          <Comment />
          <Date>2023-06-13T00:00:00</Date>
          <Number>ZZZ-Inter-268</Number>
          <Organization>Дружество 1</Organization>
        </DocumentExternalNumber>
      </DocumentExternalNumbers>
    </DocAddData>
    <DocReqDateClose>2023-06-29</DocReqDateClose>
    <DocAttentionTo />
  </Document>
  <Comment>Опит за изпращане № 1</Comment>
</DocumentRegistrationRequest>

```

Фиг. 27 XML съдържание на съобщение "Заявка за регистрация на документ"

В XML тага **<DocAddData>** се вижда идентификаторът на работния процес, който ще бъде стартиран в получаващото документа дружество, както и инструкцията, че изпълнението на работния процес трябва да приключи с определяне на резултат, който бъде изпратен обратно към изпращащото го дружество.

Съобщение от тип "Справка за състояние на документ", което се изпраща от Дружество 1 към дружеството, първоизточник на документа, след приключване на обработката му в Дружество 1:

```

<DocumentStatusResponse>
  <DocID>
    <DocumentNumber>
      <DocNumber>XXX-COMM-34</DocNumber>
      <DocDate>2023-06-13</DocDate>
    </DocumentNumber>
    <DocumentGUID>{0556037b-130b-478f-a4ce-7d766170dc30}</DocumentGUID>
  </DocID>
  <DocRegStatus>DS_CLOSED</DocRegStatus>
  <DocAddData>
    <CloseResult xmlns="">OK</CloseResult>
    <ReconciliationInfo xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="">
      <ReconciliationSteps>
        <ReconciliatorId>4</ReconciliatorId>
        <RealReconciliatorName>Потребител 1</RealReconciliatorName>
        <IsAnswerPositive>Y</IsAnswerPositive>
        <Comment>Съгласувам: одобрено</Comment>
        <IsSigned>>false</IsSigned>
        <ReconciliationDate>2023-06-13T12:23:14.744</ReconciliationDate>
      </ReconciliationSteps>
      <ReconciliationSteps>
        <ReconciliatorId>3</ReconciliatorId>
        <RealReconciliatorName>Потребител 2</RealReconciliatorName>
        <IsAnswerPositive>Y</IsAnswerPositive>
        <Comment>Съгласувам: </Comment>
        <IsSigned>>true</IsSigned>
        <ReconciliationDate>2023-06-13T12:23:32.111</ReconciliationDate>
      </ReconciliationSteps>
      <ReconciliationSteps>
        <ReconciliatorId>20</ReconciliatorId>
        <RealReconciliatorName>Потребител 3</RealReconciliatorName>
        <IsAnswerPositive>Y</IsAnswerPositive>
        <Comment>Съгласувам: </Comment>
        <IsSigned>>true</IsSigned>
        <ReconciliationDate>2023-06-13T12:23:49.258</ReconciliationDate>
      </ReconciliationSteps>
      <ReconciliationSteps>
        <ReconciliatorId>5</ReconciliatorId>
        <RealReconciliatorName>Потребител 4</RealReconciliatorName>
        <IsAnswerPositive>Y</IsAnswerPositive>
        <Comment>Съгласувам: потвърждавам</Comment>
        <IsSigned>>false</IsSigned>
        <ReconciliationDate>2023-06-13T12:23:58.74</ReconciliationDate>
      </ReconciliationSteps>
      <ReconciliationSteps>
        <ReconciliatorId>2</ReconciliatorId>
        <RealReconciliatorName>Потребител 6</RealReconciliatorName>
        <IsAnswerPositive>Y</IsAnswerPositive>
        <Comment>Съгласувам: </Comment>
        <IsSigned>>false</IsSigned>
        <ReconciliationDate>2023-06-13T12:22:39.87</ReconciliationDate>
      </ReconciliationSteps>
    </ReconciliationInfo>
    <CloseComment xmlns="">Подписано :
  </DocAddData>
</DocumentStatusResponse>

```

Фиг. 28 XML съдържание на съобщение "Справка за състояние на документ"

На Фиг. 28 е представена разширената структура на съобщението, в която е включена следната информация:

- Обработката на документа е приключена;
- Резултатът от обработката е положителен;
- Представена е информация за съгласувателен процес, изпълнен от служителите на Дружество 1.

ИЗВОДИ

1. Извършен е анализ на използваните към момента хоризонтални и централизирани ИС за електронно управление в Република България.
2. Създаден е унифициран модел на процес за предоставяне на комплексни административни услуги, независим от вида на администрацията (общинска, областна, централна), която предоставя КАО.
3. Предложен е референтен модел на софтуерна архитектура на АИС за предоставяне на КАО
4. Предложеният референтен модел на софтуерна архитектура е приложен към реално внедрена АИС в общинска администрация
5. Изследвана е възможността за надграждане на националния протокол за обмен на електронни документи между администрацията (СЕОС)
6. Реализирано е предложение за създаване на следваща версия на протокола на СЕОС с цел предоставяне на възможност за имплементация на интер-административни работни процеси от АИС.
7. Предложената нова версия на протокола е имплементирана успешно в АИС, внедрена в няколко дружества на голяма акционерна компания, оперираща на територията на Р. България

ЗАКЛЮЧЕНИЕ

В дисертационен труд е представено решение на проблем, което оказва положително въздействие върху процесите, изпълнявани в администрациите. Тези процеси засягат всички нас в качеството ни на граждани на Р. България или представители на местни бизнес организации. За представянето на решението е избран последователен подход, при който първо са представени проблемите, свързани с КАО, след това са описани процесите, които то включва и на тяхна база е представено решение за софтуерна архитектура на АИС, подпомагаща по ефективен начин служителите в държавната администрация. Предложеното решение е верифицирано чрез имплементация на включени в него елементи при проектирането на АИС АКСТЪР ОФИС, разработена от специалисти в централна лаборатория по КГ и ГИС към ТУ-София, предоставяща софтуерни решения, внедрявани в различни нива на държавната администрация.

Предстои надграждане на АИС АКСТЪР ОФИС с някои от модулите, представени в референтната архитектура, с цел постигане на максимална ефективност при предоставянето на КАО от администрациите, които я използват.

ПРИНОСИ

1. Научни и научно-приложни приноси

- 1.1. Разработен е унифициран модел на процес за предоставяне на комплексно административно обслужване.
- 1.2. Създаден е референтен модел на софтуерна архитектура на административна информационна система, който позволява оптимизиране на процесите по предоставяне на комплексно административно обслужване.
- 1.3. Създаден е модел за реализация на интер-административни работни процеси.

2. Приложни приноси

- 2.1. Създаденият референтен модел на софтуерна архитектура е приложен при проектирането на модули, осигуряващи КАО, част от реално внедрена в общинска администрация АИС.
- 2.2. Надграденият протокол за обмена на електронни документи между администрациите, осигуряващ изпълнението на интер-административни споделени работни процеси е внедрен успешно в информационните системи на дружествата от структурата на голяма акционерна компания.

ИЗПОЛЗВАНА ЛИТЕРАТУРА

- [1] АДМИНИСТРАТИВНОПРОЦЕСУАЛЕН КОДЕКС
- [2] АКТУАЛИЗИРАНА СТРАТЕГИЯ ЗА РАЗВИТИЕ НА ЕЛЕКТРОННОТО УПРАВЛЕНИЕ В РЕПУБЛИКА БЪЛГАРИЯ 2019 – 2025 г. март 2021 г.(приета с РМС № 298 от 02.04.2021 г.)
- [3] Архитектура на електронното управление в Република България, Кратко описание, Април 2019 г. Версия 1.5, Република България, Държавна агенция „Електронно управление“, Одобрена от Председателя на Държавна агенция „Електронно управление“ със Заповед № ДАЕУ-5040-11.04.2019 г.
- [4] Администрация на Министерски съвет, Базисен модел на комплексно административно обслужване, одобрен от Министерски съвет месец юни 2013 г.
- [5] Българска национална рамка за оперативна съвместимост на информационните системи в изпълнителната власт, 2015 г.
- [6] Европейска рамка за оперативна съвместимост
- [7] Закона за администрацията
- [8] Закон за електронното управление
- [9] Министерство на транспорта, информационните технологии и съобщенията, Концепция за електронно управление в България, 2010 – 2015 г.
- [10] НАРЕДБА за обмена на документи в администрацията
- [11] НАРЕДБА за общите изисквания към информационните системи, регистрите и електронните административни услуги
- [12] НАРЕДБА за прием на деца в общинските самостоятелни детски ясли, детски градини и в подготовителните групи в общинските училища на територията на Столична община
- [13] Отчет за състоянието и годишен план за развитие и обновяване на информационните ресурси в администрацията и информационните ресурси на единната електронна съобщителна мрежа на държавната администрация и за нуждите на националната сигурност, Република България, Министерство на електронното управление 2022 г.

- [14] Система за електронен обмен на съобщения (СЕОС), Техническа спецификация, Версия 1.1, март-май 2017, Република България, Държавна агенция „Електронно управление“
- [15] Стратегия за развитие на електронното управление в Република България 2014 – 2020 г., МС Република България, 2014 г.
- [16] Войноховска В, Цанков С., Въведение и ключови понятия, свързани с информационната сигурност, Научни трудове на Русенския университет - 2014, том 53, серия 6.1, стр. 71-76
- [17] Иванова, Л., Административно обслужване на „едно гише“. Методи, решения, практики“, Институт по публична администрация и европейска интеграция, София, 2003, VIII – 84, стр. 12
- [18] Младенов М., Комплексно административно обслужване: приложни аспекти, Институт по публична администрация, 2015, ISBN: 978-619-7262-05-6
- [19] Спасов К., Проблеми на развитието на системата за Е-управление в България, Лаборатория по електронно управление – БАН, 2014
- [20] Ali, A.; Bashir, M.B.; Hassan, A.; Hamza, R.; Alqhtani, S.M.; Tawfeeg T.M.; Yousif A. Design-Time Reliability Prediction Model for Component-Based Software Systems, Sensors Journal Volume 22, Issue 7, 2022
- [21] Ali. A., Jawawi. D., Isa M.A., Scalable scenario specifications to synthesize component-centric behaviour models. International Journal of Software Engineering and its Applications 9(9), 2015: pp 79-106
- [22] Bieber G. , Carpenter J. Introduction to service-oriented programming
- [23] Griman A., Valdosera L., Mendoza L., Perez M., Mendez E., Issues for Evaluating Reliability in Software Architectures, Proceedings of the Eleventh Americas Conference on Information Systems, Omaha, NE, USA August 11th-14th 2005, pp 2926-2931
- [24] Harrison N. B., Avgeriou P., Zdun U., On the impact of fault tolerance tactics on architecture patterns, SERENE '10: Proceedings of the 2nd International Workshop on Software Engineering for Resilient Systems, April 201, pp 12–21
- [25] Hashim K., Key E., A Software Maintainability Attributes Model, Malaysian Journal of Computer Science, Vol. 9 No. 2, December 1996, pp. 92-97
- [26] Immonen, A.; Niemelä, E., Survey of reliability and availability prediction methods from the viewpoint of software architecture, Softw. Syst. Model. 2008, 7, pp49–65.

- [27] Kassab M., El-Boussaidi G. Towards quantifying quality, tactics and architectural patterns interactions, The 25th international conference on software engineering and knowledge engineering, Boston, 510 MA, USA, 2013 June 27–29, pp 441–446
- [28] Kassab M., El-Boussaidi G., Mili H., A quantitative evaluation of the impact of architectural patterns on quality requirements, Conference Software Engineering Research, Management and Applications 2011 [selected papers from the 9th International Conference on Software Engineering Research, Management and Applications, SERA 2011, Baltimore, MD, USA, August 10-12, 2011
- [29] Kassab M, Mazzara M, Lee J., Succi G., Software architectural patterns in practice: an empirical study, *Innovations in Systems and Software Engineering*, 14, 263–271 (2018)
- [30] Mattsson M., Grahn H., Mårtensson F, , Software Architecture Evaluation Methods for Performance, Maintainability, Testability, and Portability, Department of Systems and Software Engineering School of Engineering, Blekinge Institute of Technology, 2006
- [31] Meyer B. Object Oriented Software Construction, 2nd ed., International Series in Computer Science. Prentice-Hall, 1997
- [32] Musa, J.D.; Iannino, A.; Okumoto, K. (Eds.) *Software Reliability: Measurement, Prediction, Application*; McGraw-Hill: Launches, UK, 1987; 621p.
- [33] Richards M., *Software Architecture Patterns*, 2015 O'Reilly Media
- [34] Roy, B.; Graham, T.N. *Methods for evaluating software architecture: A survey*, Technical Report No.2008-545 School of Computing Queen's University at Kingston Ontario, Canada
- [35] Sanchez A., Basanya R., Janowski T., Ojo A., *Enterprise Architectures – Enabling Interoperability Between Organizations* 8th Argentinean Symposium on Software Engineering, ASEE 2007, Mar del Plata, Argentina,
- [36] Scandariato R., Yskout K., Heyman T., Joosen W. *Architecting software with security patterns*, Report CW 515, April 2008 Katholieke Universiteit Leuven Department of Computer Science Celestijnenlaan 200A – B-3001 Heverlee (Belgium)
- [37] Scholl H., Klischewski R., *E-Government Integration and Interoperability: Framing the Research Agenda*, *Intl Journal of Public Administration*, 30: pp 889–920, 2007
- [38] Semegn A. *Software Architecture and Design for Reliability Predictability*, Cambridge Scholars Publishing, 2011
- [39] Soares D., Amaral L., *Information Systems Interoperability in Public Administration - Identifying the Major Acting Forces through a Delphi Study*, *Journal of Theoretical and Applied Electronic Commerce Research* ISSN 0718–1876

- [40] Suda B., SOAP Web Services, 2003, Thesis for Master of Science, Computer Science, School of Informatics University of Edinburgh
- [41] Tarinejad, A., Izadkhah, H., Ardakani, M.M., Mirzaie, K., Metrics for assessing reliability of self-healing software systems, *Comput. Electr. Eng.* 2021, 90, 106952.
- [42] Valipour M., Amirzafari B, Maleki K., Daneshpour N. A Brief Survey of Software Architecture Concepts and Service Oriented Architecture, Department of Electrical and Computer Engineering Shahid Rajaei University Tehran 1678815811, Iran, 2009
- [43] Yacoub, S.; Cukic, B.; Ammar, H.H., A scenario-based reliability analysis approach for component-based software., *IEEE Trans. Reliab.* 2004, 53, pp 465–480.
- [44] Yautsiukhin A., Scandariato R., Heyman T., Massacci F., Joosen W., Towards a quantitative assessment of security in software architectures, Nordic Workshop on Secure IT Systems (NordSec) 2008/10/01, Copenhagen, Denmark
- [45] White L, Wilde N, Reichherzer T., El-Sheikh E., Goehring G., Understanding Interoperable Systems: Challenges for the Maintenance of SOA Applications, 45th Hawaii International Conference on System Sciences, University of West Florida Pensacola, Florida, U. S. A., 2012
- [46] Bray T., The JavaScript Object Notation (JSON) Data Interchange Format, Internet Engineering Task Force (IETF), Ed., ISSN: 2070-1721, December 2017, <https://tools.ietf.org/html/rfc8259>
- [47] Fielding R., Irvine UC, Gettys J., Compaq/W3C, Mogul J. , Compaq, Frystyk H., W3C/MIT, Masinter L., Xerox, Leach P., Microsoft, Berners-Lee T., Hypertext Transfer Protocol -- HTTP/1.1, Network Working Group, 1999, <https://tools.ietf.org/html/rfc2616>
- [48] Heidel S., Backward vs. Forward Compatibility, 2020 <https://stevenheidel.medium.com/backward-vs-forward-compatibility-9c03c3db15c9>
- [49] E. Rescorla, HTTP Over TLS, Network Working Group, 2000 <https://tools.ietf.org/html/rfc2818>
- [50] BEA Systems, Inc., Anatomy of the Client/Server Model, 2001 https://docs.oracle.com/cd/E13203_01/tuxedo/tux80/atmi/intbas3.htm
- [51] Comparison of architecture patterns <https://www.simform.com/wp-content/uploads/2020/05/Comparison-of-architecture-patterns.png>
- [52] IBM, What is service-oriented architecture (SOA)?, <https://www.ibm.com/topics/soa>
- [53] Microsoft, Improving web application security: Threats and countermeasures,

[https://learn.microsoft.com/en-us/previous-versions/msp-n-p/ff649874\(v=pandp.10\)?redirectedfrom=MSDN](https://learn.microsoft.com/en-us/previous-versions/msp-n-p/ff649874(v=pandp.10)?redirectedfrom=MSDN)

[54] Organization for the advancement of structured information standards,
<http://www.oasis-open.org>

[55] W3C Note 08 May 2000, Simple Object Access Protocol (SOAP) 1.1, 2000
<https://www.w3.org/TR/2000/NOTE-SOAP-20000508/>

[56] W3C Note 15 March 2001, Web Services Description Language (WSDL) 1.1, 2001
<https://www.w3.org/TR/2001/NOTE-wsdl-20010315>

[57] W3C Recommendation 5 April 2012, Extensible Markup Language (XML) 1.0 (Fifth Edition), 2008
<https://www.w3.org/TR/xml/#sec-intro>

[58] W3C Recommendation 5 April 2012, W3C XML Schema Definition Language (XSD) 1.1 Part 2: Datatypes, 2012,
<https://www.w3.org/TR/2012/REC-xmlschema11-2-20120405/datatypes.html>

**ПРИЛОЖЕНИЕ 1 СПИСЪК НА АДМИНИСТРАЦИИ,
ПРЕДОСТАВЯЩИ ЧАСТИЧНО И ПЪЛНО КОМПЛЕНСНО
АДМИНИСТРАТИВНО ОБСЛУЖВАНЕ С АИС АКСТЪР ОФИС**

№	Администрация	СЕОС	ССЕВ	RegiX	Е-Плащания
1	ОА Плевен	да	да	да	—
2	Община Аксаково	да	да	да	да
3	Община Антоново	да	да	да	да
4	Община Банско	да	да	да	да
5	Община Белоградчик	да	да	да	да
6	Община Брусарци	да	да	да	—
7	Община Бургас	да	да	да	да
8	Община Бяла Слатина	да	да	—	да
9	Община Враца	да	да	да	да
10	Община Вълчедръм	да	да	да	да
11	Община Вършец	да	да	да	да
12	Община Дупница	да	да	да	—
13	Община Дългопол	да	да	да	—
14	Община Лом	да	да	да	да
15	Община Марица	да	да	да	да
16	Община Монтана	да	да	да	да
17	Община Панагюрище	да	да	да	да
18	Община Перуцица	да	да	да	—
19	Община Пловдив	да	да	да	да
20	Община Пловдив – р. Източен	да	да	да	да
21	Община Пловдив – р. Северен	да	да	да	да
22	Община Пловдив – р. Южен	да	да	да	да
23	Община Разлог	да	да	да	да
24	Община Севлиево	да	да	да	—
25	Община Сливен	да	да	да	да
26	Община Стамболийски	да	да	да	да
27	Община Сухиндол	да	да	да	да
28	Община Троян	да	да	да	да

29	СО - район Искър	да	да	—	да
30	СО - район Слатина	да	да	да	да
31	СО - район Триадица	да	да	да	да

ПРИЛОЖЕНИЕ 2 ПУБЛИКАЦИИ ПО ТЕМАТА НА ДИСЕРТАЦИОННИЯ ТРУД

1. Glavev V., Integration of document management system with e-government e-payments system, Applications of Mathematics in Engineering and Economics (AMEE'22): Proceedings of the 48th International Conference on Applications of Mathematics in Engineering and Economics, 7–13 June 2022 Sozopol, In press
2. Glavev V., Software Architecture of Administrative Information System Oriented to Complex Administrative Services (One Stop Shop), Applications of Mathematics In Engineering and Economics (AMEE'21): Proceedings of the 47th International Conference on Applications of Mathematics in Engineering and Economics, 7–13 June 2021 Sozopol, Bulgaria, AIP Conference Proceedings 2505, 060007 (2022), <https://doi.org/10.1063/5.0100725>
3. Glavev V., Interoperability and models for exchange of data between information systems in public administration, Applications of Mathematics in Engineering and Economics (AMEE'16): Proceedings of the 42nd International Conference on Applications of Mathematics in Engineering and Economics, 8–13 June 2016 Sozopol, Bulgaria, AIP Conference Proceedings 1789, 060010 (2016) doi: <https://doi.org/10.1063/1.4968502>
4. Victor Glavev, Problems of the Architecture of E-governance in Public Administration of Republic of Bulgaria, Proceedings of Doctoral Conference in Mathematics and Informatics, MIDOC 2015, October 15-18, 2015, Sofia, Bulgaria, pp 50-58

ПРИЛОЖЕНИЕ 3 УДОСТОВЕРЕНИЯ ЗА УСПЕШНО ВНЕДРЯВАНЕ НА РАЗРАБОТЕНИТЕ МОДЕЛИ ЗА КОМПЛЕКСНО АДМИНИСТРАТИВНО ОБСЛУЖВАНЕ И ИНТЕР-АДМИНИСТРАТИВНИ РАБОТНИ ПРОЦЕСИ

ОБЩИНА ТРОЯН  TROYAN MUNICIPALITY
ПЛ. „ВЪЗРАЖДАНЕ“ 1, 5600 ТРОЯН
0670 6 80 09, ФАКС: 0670 6 80 60
MAYOR@TROYAN.BG 1, VAZRZHDANE SQ., 5600 TROYAN, BULGARIA
0670 6 80 09, FAX: 0670 6 80 60
MAYOR@TROYAN.BG

10.7.2023 г.

X

Анита Маринова
И-РД-874/10.07.2023
| Подписано от: Anita Hristova Marinova

РЕФЕРЕНЦИЯ

Община Троян издава настоящата референция на инж. Виктор Главев, в качеството му на проектант и разработчик на модулите за предоставяне на комплексно административно обслужване на АИС АКСТЪР ОФИС, внедрена и използвана успешно от Община Троян за предоставяне на комплексни административни услуги. С помощта на разработената от Технически Университет – София – Технологии ЕООД и Централната лаборатория по компютърна графика и ГИС към ТУ-София административна информационна система, община Троян предоставя ефективно комплексно административно обслужване на физически и юридически лица.

10.7.2023 г.

X

Цветелина Калчева
Секретар на Община Троян
Подписано от: Tsvetelina Rashova Kalcheva

РЕФЕРЕНЦИЯ

„Електрохолд ИКТ“ ЕАД издава настоящата референция на инж. Виктор Главев, в уверение на това, че в качеството му на проектант и разработчик на модулите за изпълнение на интер-административни работни процеси, част от информационната система АКСТЪР ОФИС, внедрена и използвана от дружествата в групата на Електрохолд България е осигурявал услуги по договор номер ICT-CON-58/30.11.2018 г.

С помощта на разработената от Технически Университет – София – Технологии ЕООД и Централната лаборатория по компютърна графика и ГИС към ТУ-София информационна система, дружествата от групата Електрохолд изпълняват общи административни процеси за съгласуване и утвърждаване на електронни документи.

Кирил Петков



Изпълнителен директор на
„Електрохолд ИКТ“ ЕАД

Гр. София,
11.07.2023 г.

ДЕКЛАРАЦИЯ ЗА ОРИГИНАЛНОСТ И ДОСТОВЕРНОСТ
(по чл. 27, ал.2 от ПП ЗРАСРБ)

От маг. инж. Виктор Венциславов Главев, докторант към катедра Информатика към Факултета по приложна математика и информатика на Технически университет - София, декларирам, че представената от мен за защита дисертация на тема: **МОДЕЛИ И СОФТУЕРНИ АСПЕКТИ ПРИ РЕАЛИЗАЦИЯТА НА КОМПЛЕКСНИ АДМИНИСТРАТИВНИ УСЛУГИ** за присъждане на образователната и научна степен "доктор" е оригинална разработка и съдържа оригинални резултати, получени при проведени от мен научни изследвания (с подкрепата и/или съдействието на научния ми ръководител).

Декларирам, че резултатите, които са получени, описани и/или публикувани от други учени, са надлежно и подробно цитирани в библиографията, при спазване на изискванията за защита на авторското право.

Уведомен/а съм, че в случай на констатиране на плагиатство в представената дисертация, комисията по защитата е в правото си да я отхвърли.

Декларирам, че настоящата дисертация не е представяна пред други университети, институти и други висши училища за придобиване на образователна и научна степен.

Дата: 03.07.2023

Декларатор:

/ Виктор Главев /