Anexa nr. 22
la Documentația standard nr. 115
din "15" septembrie 2021

Specificații tehnice

Numărul procedurii de achiziție ***ocds-b3wdp1-MD-1767702087716* din *06 ianuarie 2026***
Obiectul achiziției: ***Sistem informațional de billing și facturare***

Denumirea serviciilor	Denumirea modelului serviciului	Ţara de origine	Produ-cătorul	Specificarea tehnică deplină solicitată de către autoritatea contractantă	Specificarea tehnică deplină propusă de către ofertant	Standarde de referinţă
1	2	3	4	5	6	7
Servicii						
Lotul nr. 1 - Sistem informațional de billing și facturare		RM	RTS	Conform Caietului de Sarcini	Conform Propunerii tehnice anexate la specificații tehnice	
TOTAL						

Digitally signed by Ghiţman Alexandr
Date: 2026.02.03 12:06:09 EET
Reason: MoldSign Signature
Location: Moldova
MOLDOVA EUROPEANĂ

Semnat:_____ Numele, Prenumele: **GHIȚMAN Alexandr** În calitate de: **Administrator**
Ofertantul: **„Real Tech Solutions" S.R.L.** Adresa: **mun. Chișinău, str. Mitropolit G. Bănulescu-Bodoni, 59/B, of. 805**

Anexă
la Specificații tehnice

PROPUNEREA TEHNICĂ

privind dezvoltarea Sistemului informațional de billing și facturare pentru S.A. „ENERGOCOM"

Chișinău 2026

Cuprins

1. INTRODUCERE

1.1. Informații Generale

Documentul dat include descrierea ofertei tehnice pentru dezvoltare și implementare a Sistem informațional de billing și facturare în cadrul S.A. „Energocom", dedicat digitalizării și eficientizării proceselor de evidență a consumatorilor de gaze naturale și generarea facturilor.

La elaborarea sistemului se va ține cont de legislația în vigoare şi standardele naționale în domeniul asigurării securității informaționale şi protecției informației.

1.2. Obiectivele sistemului

1. **Centralizarea și automatizarea proceselor de billing**

Sistemul va centraliza toate procesele legate de evidența consumatorilor, procesarea consumurilor și generarea facturilor, incluzând gestionarea celor 3 tipuri de presiune (joasă, medie, înaltă). Automatizarea completă va elimina introducerea manuală și va reduce timpul necesar pentru procesarea lunară.

2. **Flexibilitate în formule de calcul și gestionare tarife**

Sistemul va permite configurarea flexibilă a formulelor de calcul pentru toate cele 3 tipuri de presiune, calcule bazate pe volum și capacitate calorică. Tarifele și formulele vor putea fi ajustate rapid de administratori fără modificări de cod.

3. **Integrare bancară și validare automată plăți**

Integrare completă cu platformele bancare din RM prin API-uri securizate pentru transmitere date facturi și primire confirmare plăți. Plățile vor fi reconciliate automat cu facturile emise.

4. **Raportare avansată și monitorizare în timp real**

Dashboard-uri interactive cu KPI-uri, rapoarte financiare detaliate, rapoarte ANRE și export date în formate multiple. Managementul va avea acces la date în timp real.

5. **Trasabilitate completă și conformitate GDPR**

Sistemul va înregistra toate operațiile cu autori și date. Conformitate deplină cu GDPR și legislația națională, criptare date sensibile, audit complet.

1.3. Principiile de elaborare ale soluției

Întru asigurarea obiectivelor înaintate sistemului, la proiectarea, dezvoltarea și implementarea acestuia se va ține cont de următoarele principii generale:

- **Principiul Legalității:** sistemul va fi creat și operat în strictă conformitate cu legislația națională și cu normele și standardele naționale în domeniul managementului resurselor umane.
- **Principiul Autenticității și Veridicității Datelor:** datele introduse în sistem vor fi bazate pe informații autentice și verificabile. Toate datele păstrate în sistem vor reflecta fidel situația reală a obiectelor monitorizate, asigurându-se că informațiile sunt corecte și actualizate în permanență.
- **Principiul Securității Informaționale:** sistemul va implementa măsuri riguroase de securitate pentru a proteja datele împotriva pierderii, ștergerii, deteriorării și accesului neautorizat. Sistemul va asigura un nivel înalt de integritate, accesibilitate și eficiență în gestionarea datelor.
- **Principiul Transparenței:** dezvoltarea și operarea sistemului vor respecta standarde transparente în domeniul tehnologiilor informatice.
- **Principiul Expansibilității:** sistemul va fi proiectat astfel încât să permită extinderea și completarea sa cu noi funcții sau îmbunătățirea celor existente.
- **Principiul Scalabilității:** sistemul va fi capabil să mențină performanțe constante chiar și atunci când volumul de date și cerințele operaționale cresc.
- **Principiul independenței față de platforma software:** sistemul va fi dezvoltat pe baza unor module personalizate, fundamentate pe tehnologii open source, pentru a asigura flexibilitate, scalabilitate și costuri reduse de licențiere.
- **Principiul orientării spre utilizator (usability):** structura, conținutul, mijloacele de acces şi navigarea vor fi focalizate spre utilizatorii sistemului. Interfețele vor fi proiectate pentru a asigura o experiență intuitivă, ergonomică și eficientă pentru utilizatori.

1.4. Abrevieri și noțiuni utilizate

Abrevierile principale utilizate în prezentul document sunt delimitate în tabelul 1.1.

Tabelul 1.1. Abrevierile și acronimele utilizate în document

Nr.	Abreviere	Descriere
1.	SBF	Sistemul de Billing și Facturare
2.	Consumator PF	Consumator persoană fizică (casnic)
3.	Consumator PJ	Consumator persoană juridică (non-casnic)
4.	NLC	Numărul Locului de Consum (identificator unic)

5.	IDNP	Identificatorul Numeric Personal (pentru persoane fizice)
6.	IDNO	Identificatorul Numeric al Organizației (pentru persoane juridice)
7.	API	Application Programming Interface
8.	RBAC	Role-Based Access Control
9.	CI/CD	Continuous Integration/Continuous Deployment
10.	ANRE	Agenția Națională pentru Reglementare în Energetică
11.	GDPR	Regulamentul General privind Protecția Datelor
12.	SI	Sistem Informatic
13.	TIC	Tehnologie Informatică și de Comunicație
14.	TLS/SSL	Transport Layer Security
15.	SLA	Service Level Agreement
16.	LAN	Local Area Network
17.	Web	World Wide Web

Noțiuni frecvent utilizate în prezentul document sunt explicate în tabelul 1.2.

Tabelul 1.2. Definiții și noțiuni utilizate în document

Nr.	Noțiune	Descriere
1.	SA "Energocom"	Societatea pe acțiuni, furnizor de gaze naturale în Republica Moldova
2.	Loc de consum	Punct fizic de furnizare a gazelor naturale, identificat prin adresă și contor
3.	Capacitate calorică	Puterea calorifică superioară, exprimată în kWh/m³ (ex: 10,6002 kWh/m³)
4.	Coeficient ANRE	Coeficient de ajustare reglementat de ANRE
5.	Presiune joasă/medie/înaltă	Tipuri de presiune pentru furnizarea gazelor naturale (3 categorii), folosite în formule de calcul pentru consumatorii non-casnici
6.	Clasificatoare	Registre de referință cu date standardizate
7.	Bază de Date	Ansamblu de date organizate conform structurii conceptuale care descrie caracteristicele de baza și relația dintre entități.
8.	Credențiale	Set de atribute ce stabilesc identitatea și autenticitatea utilizatorilor și sistemelor în cadrul sistemelor informaționale.
9.	Modul	Un fragment complet, din punct de vedere funcțional, al unui program, conceput ca un fișier separat cu codul sursă sau ca o parte continuă numită a acestuia.

10.	Utilizator	Angajat DAJ sau oricare altă persoană autorizată care are acces la DAJ Soft, în conformitate cu modelul/tipul de rol al drepturilor de acces.
11.	Rol	Un set de drepturi de acces al utilizatorilor la funcțiile și datele sistemului.
12.	Eveniment	Înregistrare de sistem sub forma unei alerte de informare cu privire la o modificare în sistem: adăugarea unui Dosar, adăugarea unui participant, adăugarea unei sarcini, adăugarea unui eveniment, adăugarea unui dosar, etc.
13.	Single Sign On	Serviciu ce oferă utilizatorilor interni şi externi ai sistemului drepturi de acces la servicii, în baza credenţialelor lor din domeniul de origine, în condiţiile când între domeniul de origine şi domeniul gazdă se află în relaţii de încredere.
14.	Sistem de securitate al infrastructurii	Servicii de securitate concepute pentru a asigura integritatea şi disponibilitatea hardware şi software a sistemelor. Acesta include actualizarea automată a software-ului, administrarea şi instalarea la distanţă, antivirus, firewall, etc.
15.	Fişier	Unitate organizată de înregistrări grupate pe principiul legăturii cu acelaşi subiect, activitate sau tranzacţie. Notă: Aceasta este definiţia noţiunii de fişier în Managementul Înregistrărilor. După cum se vede, această definiţie diferă de cea utilizată în sfera Tehnologiilor Informaţionale.
16.	Meta-date	Date ce descriu contextul, conţinutul şi structura înregistrărilor şi managementul lor în timp. Metadatele sunt în general „date despre date", sau altfel spus, date care descriu alte date, de orice fel și de orice tip. În engleză termenul e numit metadata (la plural). Ca exemple, un element metadată poate descrie: (i) dată individulă; (ii) un element de conţinut; (iii) colecţie de date, incluzând unul sau mai multe elemente de conţinut și de niveluri ierarhice; (iv) așa-numită schemă de baze de date.
17.	Date cu caracter personal	Orice informație cu referire la o persoana fizică identificată sau identificabilă (subiect al datelor cu caracter personal). În acest sens o persoană identificabilă este o persoană care poate fi identificată, direct sau indirect, în special prin referire la un număr de identificare sau la unul sau mai multe elemente specifice, proprii identității sale fizice, fiziologice, psihice, economice, culturale sau sociale.
18.	Integritatea datelor	Stare a datelor, când acestea își păstrează conținutul și sunt interpretate univoc în cazuri de acțiuni aleatorii. Integritatea se consideră păstrată dacă datele sunt corecte și este asigurată încrederea în ele.
19.	Jurnalizare	Funcție de înregistrare a informației despre evenimente. În cadrul sistemelor informaționale înregistrările despre

		evenimente includ detalii despre data și ora, utilizatorul, datele personale identificate, acțiunea întreprinsă.
20.	Veridicitatea datelor	Nivel de corespundere a datelor, păstrate în memoria calculatorului sau în documente, stării reale a obiectelor din domeniul respectiv al sistemului, reflectate de aceste date.

2. ARHITECTURA SISTEMULUI

Arhitectura sistemului de Billing și Facturare (SBF) al SA „ENERGOCOM" va fi bazată pe o structură modernă de microservicii, care asigură scalabilitatea, performanța și flexibilitatea necesare pentru gestionarea unui număr semnificativ de consumatori, procesarea datelor de consum și generarea facturilor.

Sistemul va fi proiectat să automatizeze procesele manuale, reducând semnificativ erorile umane și oferind trasabilitate completă a datelor.

Mai jos este reprezentată diagrama arhitecturală a sistemului de billing și facturare, care ilustrează componentele esențiale ale sistemului și modul în care acestea sunt interconectate.

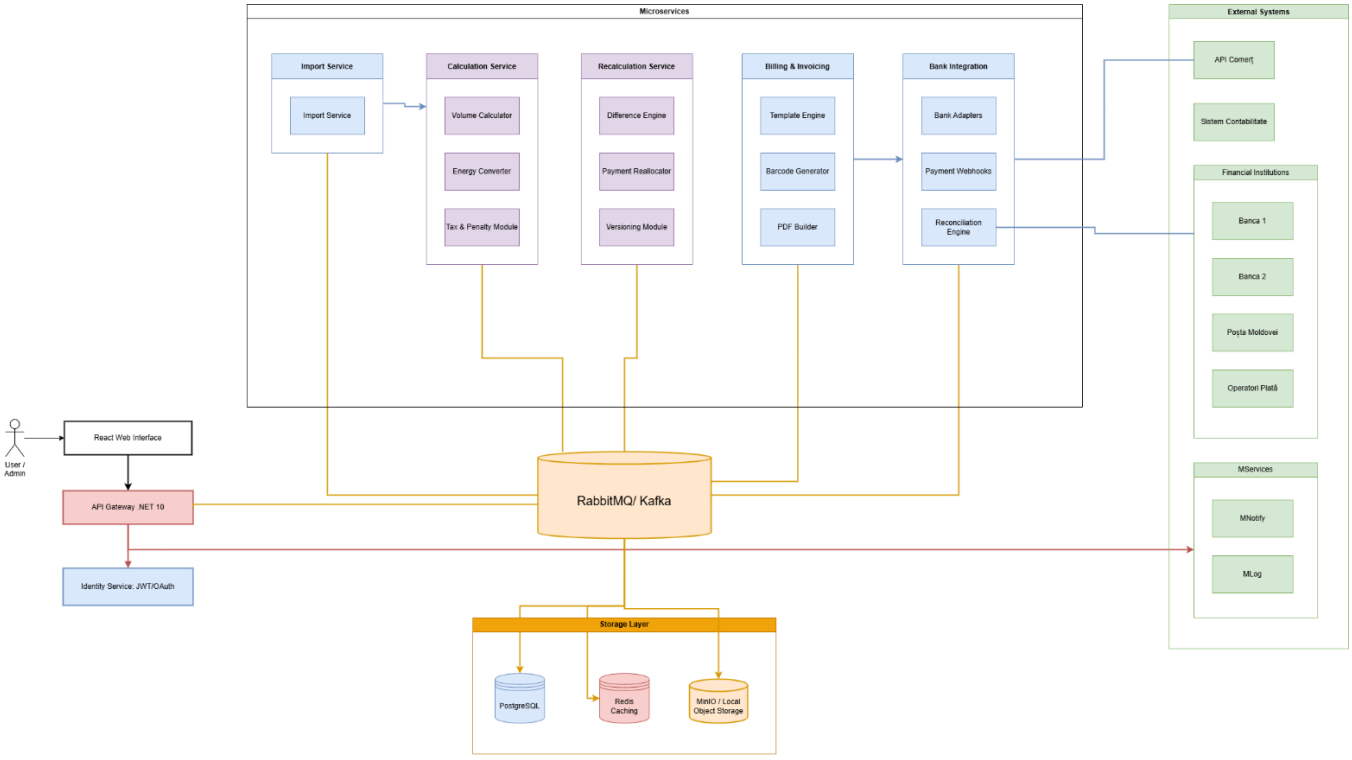

Figura 2.1 Arhitectura sistemului

Arhitectura sistemului va fi împărțită în mai multe straturi și servicii, fiecare având un rol clar definit:

1. **Interfața Utilizatorului (UI Layer)**:

- **React Web Interface**: Este punctul principal de interacțiune între utilizatori (Admin și Operator) și sistemul de facturare. Aceasta oferă o interfață web intuitivă pentru gestionarea și vizualizarea informațiilor.

2. **API Gateway**:
   - **API Gateway - .NET 10**: Asigură gestionarea tuturor cererilor API provenite de la utilizatori sau de la sistemele externe. Se ocupă cu autentificarea (utilizând Identity Service) și permite comunicarea securizată între microservicii și sursele externe.
3. **Microservicii**:
   - **Import Service**: Acest serviciu preia și validează datele de consum de la API-urile externe (de exemplu, API Comerț), importând informațiile despre consumatori și volume.
   - **Calculation Service**: Efectuează calculele pentru facturare, inclusiv calculul volumului de consum, conversia energiei și aplicarea taxelor.
   - **Recalculation Service**: Permite recalcularea facturilor în caz de ajustări sau corecții, având în vedere eventualele modificări de consum sau date financiare.
   - **Invoicing Service**: Este responsabil de generarea facturilor, crearea template-urilor, adăugarea de coduri de bare și exportarea documentelor PDF pentru tipărire.
   - **Bank Integration Service**: Asigură integrarea cu sistemele bancare pentru procesarea plăților și reconcilierea acestora.
4. **Event-driven Communication**:
   - **RabbitMQ / Kafka**: Aceste tehnologii sunt utilizate pentru a permite comunicarea asincronă între microservicii, asigurând un flux continuu de date și evenimente între componentele sistemului.
5. **Stratul de Stocare (Storage Layer)**:
   - **PostgreSQL**: PostgreSQL: Baza de date principală pentru stocarea datelor structurate, cum ar fi informațiile despre consumatori și facturi.
   - **Redis Caching**: Utilizat pentru stocarea temporară a datelor care trebuie accesate rapid, cum ar fi tarifele active și datele de consum.
   - **MinIO**: Este folosit pentru stocarea obiectelor (de exemplu, fișierele PDF generate pentru facturi), asigurându-se astfel că datele sunt păstrate local în infrastructura proprie.
6. **Sisteme Externe**:

- **API Comerț**: Schimbă date despre consumatori și facturi.
- **Sistem Contabilitate**: Permite integrarea financiară cu alte sisteme contabile pentru procesarea plăților și reconcilierea acestora.
- **Platformele Bancare**: Băncile și operatorii de plată (e.g., Banca 1, Banca 2, Poșta Moldovei) sunt integrate pentru a facilita plățile și reconcilierea acestora.

7. **Servicii de Monitorizare și Audit**:
   - **MNotify** și **MLog** sunt serviciile care se ocupă cu gestionarea notificărilor și logării activităților din sistem, asigurând un flux continuu de informații și auditabilitatea acestora.

3. MODULELE SISTEMULUI

În cadrul sistemului de Billing și Facturare (SBF) al SA „ENERGOCOM", fiecare modul va fi responsabil pentru un anumit segment al fluxului operațional. Aceste module vor fi implementate folosind microservicii, iar design-ul lor beneficiază de pattern-uri specifice pentru a asigura eficiența, scalabilitatea și flexibilitatea sistemului.

3.1. Modulul 1: API Gateway

Scop: API Gateway este punctul central de intrare pentru toate cererile externe și interne ale sistemului, gestionând rutarea cererilor, securitatea și autentificarea utilizatorilor.

Componente:

- **Autentificare/Autorizare (JWT/OAuth)**: Va asigura securizarea și gestionarea accesului în sistem.
- **Rate Limiting**: Va previni supraîncărcarea API-ului prin limitarea numărului de cereri pe unitate de timp.
- **Jurnalizare apeluri**: Va monitoriza toate cererile API pentru audit și diagnosticare.

Pattern utilizat: **Facade Pattern**

- **Explicație**: API Gateway va acționează ca o fațadă unificată pentru toate cererile externe, simplificând accesul și interacțiunea utilizatorilor cu sistemele interne și externe.

3.2. Modulul 2: Import Service

Scop: Acest modul va fi responsabil pentru preluarea și validarea datelor provenite de la API-urile externe, cum ar fi **API Comerț**. Importă datele despre consumatori și consumuri într-un format standardizat.

Submodule:

- **Validare date**: Va verifica corectitudinea datelor înainte de a fi procesate.
- **Gestionare asincronă (Queues)**: Va procesa datele într-un mod asincron pentru a nu bloca alte fluxuri operaționale.
- **Prevenire duplicate (Idempotency)**: Va previne procesarea dublă a aceleași date.

Pattern utilizat: **Adapter Pattern**

- **Explicație**: Adapter Pattern va fi folosit pentru a integra API-uri externe diferite într-un format uniform, astfel încât microserviciul să poată comunica cu surse externe fără a depinde de specificitatea fiecărei surse externe.

3.3. Modulul 3: Calculation Service

Scop: Modulul de calcul va răspunde de calcularea valorilor facturabile, inclusiv volum de consum, conversia în energie și aplicarea tarifelor.
Submodule:

- **Calcul volum (tipuri de presiune: joasă, medie, înaltă)**: Va calcula consumul de energie în funcție de presiune.
- **Conversie kWh**: Va converti consumul de gaze în energie electrică echivalentă.
- **Calcul TVA și penalități**: Va aplica TVA-ul și penalitățile corespunzătoare pentru consumurile suplimentare.

Pattern utilizat: **Strategy Pattern**

- **Explicație**: Strategy Pattern va fi utilizat pentru a aplica diferite strategii de calcul în funcție de tipul de consum (de exemplu, presiune joasă vs medie vs înaltă). Fiecare strategie de calcul este implementată ca o clasă separată, iar **Calculation Service** alege strategia potrivită în funcție de datele consumului.

Pattern utilizat: **Factory Pattern**

- **Explicație**: Factory Pattern va fi folosit pentru a crea obiecte de calcul specifice pentru fiecare tip de consum, asigurându-se că procesul de calcul este automatizat și extensibil.

3.4. Modulul 4: Recalculation Service

Scop: Modulul de recalculare va permite ajustarea valorilor facturabile pentru consumuri și corectarea erorilor de facturare, fără a modifica datele istorice.

Submodule:

- **Identificare obiect (factură/perioadă)**: Va identifica facturile care trebuie ajustate.
- **Calcul diferențe (Debit/Credit)**: Va calcula diferențele între sumele de facturat și plățile deja efectuate.
- **Versionare calcule**: Va permite păstrarea unei istorii a recalculărilor pentru a asigura trasabilitatea datelor.

Pattern utilizat: **State Pattern**

- **Explicație**: State Pattern va fi utilizat pentru a gestiona diversele stări ale facturii în timpul procesului de recalculare, cum ar fi facturi corectate, facturi în așteptare etc. Acesta permite trecerea dintr-o stare în alta, păstrând istoricul tuturor modificărilor.

3.5. Modulul 5: Invoicing Service

Scop: Modulul de facturare va crea facturi pentru consumatori, generând fișiere PDF și coduri de bare, gata pentru tipărire.

Submodule:

- **Motor template-uri (PF/PJ)**: Va crea template-uri pentru facturi, diferite pentru persoane fizice și juridice.
- **Generare coduri de bare**: Va genera un cod de bare unic pentru fiecare factură.
- **PDF Builder**: Va crea documentul PDF al facturii pentru a fi tipărit.

Pattern utilizat: **Factory Pattern**

- **Explicație**: Factory Pattern va fi utilizat pentru a crea diferite tipuri de facturi pe baza șablonului (de exemplu, facturi pentru persoane fizice sau juridice), automatizând procesul de creare a documentelor.

Pattern utilizat: **Template Method Pattern**

- **Explicație**: Template Method Pattern va fi utilizat pentru a defini un proces general de creare a facturilor, lăsând pașii specifici (de exemplu, generarea codului de bare) să fie implementați în subclase specifice.

3.6. Modulul 6: Bank Integration Service

Scop: Acest modul va asigura integrarea cu sistemele bancare pentru procesarea și reconcilierea plăților.

Submodule:

- **Adaptoare pentru bănci**: Va permite integrarea cu multiple platforme bancare.
- **Webhook-uri pentru plăți**: Va primi notificări pentru plățile efectuate.
- **Reconciliere automată a soldurilor**: Va actualiza soldurile utilizatorilor pe baza plăților efectuate.

Pattern utilizat: **Bridge Pattern**

- **Explicație**: Bridge Pattern va fi utilizat pentru a decupla logica de procesare a plăților de platformele bancare externe, permițând implementarea mai multor adaptoare pentru diverse bănci și platforme de plată.

Pattern utilizat: **Observer Pattern**

- **Explicație**: Observer Pattern va fi utilizat pentru a implementa notificările în timp real (de exemplu, pentru a anunța sistemul când o plată a fost efectuată).

3.7. Stratul de Interfață (UI Layer)

Scop: Acesta va fi punctul de interacțiune între utilizatorii Admin și Operator și sistem, permițând gestionarea utilizatorilor, controlul proceselor și vizualizarea rapoartelor.

Submodule:

- **Gestiune utilizatori (RBAC/2FA)**: Va permite gestionarea rolurilor utilizatorilor și autentificarea în două etape pentru securitate sporită.
- **Control procese (Start/Stop Import/Calcul)**: Va permite managementul fluxurilor de date și al proceselor de calcul.
- **Modul Tarife**: Va permite gestionarea și vizualizarea tarifelor active.
- **Centru Raportare**: Va oreri rapoarte financiare și statistici ale consumurilor.

3.8. Stratul de Date și Stocare (Data & Storage Layer)

Componente:

- **PostgreSQL**: Va stoca datele structurate, precum consumatorii și facturile.
- **Redis Caching**: Va fi folosit pentru caching rapid al datelor care sunt accesate frecvent.
- **MinIO**: Va fi sistem de stocare a obiectelor pentru fișiere PDF.

Pattern utilizat: **Repository Pattern**

- **Explicație**: Repository Pattern va fi utilizat pentru a abstractiza accesul la date, permițând interacțiunea uniformă cu baza de date și stocarea obiectelor, fără a depinde de detaliile implementării bazei de date.

3.9. Securitatea Aplicației

Securitatea aplicației este o prioritate majoră, având în vedere volumul mare de date sensibile și tranzacțiile financiare efectuate de către cei aproximativ 900.000 de consumatori. Sistemul va implementa o serie de măsuri de protecție pentru a asigura integritatea și confidențialitatea datelor, precum și pentru a preveni accesul neautorizat sau atacurile cibernetice.

Măsuri de Securitate Implementate

1. **Protecția Datelor în Tranzit (SSL/TLS)**:

- Toate comunicațiile între client și server vor fi criptate folosind protocoalele **SSL/TLS**, asigurându-se astfel că datele sensibile (inclusiv informațiile despre consumatori și plăți) nu pot fi interceptate sau modificate în timpul transferului.

2. **Autentificare și Autorizare**:

- **Microsoft Identity Platform** va fi utilizat pentru autentificare și autorizare. Aceasta include suport pentru **OAuth 2.0**, **OpenID Connect** și **JWT (JSON Web Tokens)** pentru gestionarea sesiunilor și asigurarea unui mecanism de autentificare robust.
- Se va implementa **autentificarea multi-factor (MFA)** pentru a adăuga un strat suplimentar de securitate, protejând conturile utilizatorilor printr-un proces de verificare suplimentar.

3. **Protecția împotriva CSRF (Cross-Site Request Forgery)**:

- **CSRF tokens** vor fi utilizate pentru a preveni atacurile de tip **Cross-Site Request Forgery**. Aceste token-uri vor fi generate și validate pentru fiecare cerere de modificare a datelor, asigurându-se că solicitările provin doar de la utilizatori autorizați.

4. **Protecția împotriva XSS (Cross-Site Scripting)**:

- **Sanitizarea input-urilor** va fi implementată pe toate punctele de intrare ale aplicației pentru a preveni injectarea de cod malițios în paginile web. În plus, **Content Security Policy (CSP)** va fi utilizat pentru a limita sursele de scripturi și resurse externe care pot fi utilizate de aplicație.

5. **Testarea de Securitate cu OWASP ZAP**:

- **OWASP ZAP (Zed Attack Proxy)**, un instrument gratuit și open-source pentru testarea aplicațiilor web, va fi utilizat pentru a identifica vulnerabilitățile de securitate. Acesta va efectua scanări automatizate de

securitate pentru a descoperi probleme precum **SQL Injection**, **XSS** și altele.

- **Pentesting**: Testele de penetrare (pentesting) vor fi efectuate periodic pentru a identifica vulnerabilitățile de securitate și pentru a evalua eficiența măsurilor implementate.

6. **Controlul Accesului pe Baza de Roluri (RBAC)**:

- **Role-Based Access Control (RBAC)** va fi folosit pentru a gestiona permisiunile utilizatorilor. Fiecare utilizator va avea un rol definit (e.g., Admin, Operator) și va avea acces doar la funcționalitățile și datele care îi sunt necesare în conformitate cu acest rol.
- În plus, se va utiliza **autentificarea în două etape (2FA)** pentru a adăuga un strat suplimentar de protecție împotriva accesului neautorizat.

7. **Protecția Împotriva Atacurilor DDoS**:

- Se vor implementa măsuri de protecție împotriva atacurilor de tip **Denial of Service (DoS)** și **Distributed Denial of Service (DDoS)**, cum ar fi **Rate Limiting**, **IP Filtering**, și **WAF (Web Application Firewall)** pentru a preveni atacurile de supraîncărcare a serverului.

8. **Criptarea Datelor Sensibile**:

- Toate datele sensibile, inclusiv informațiile financiare și personale ale consumatorilor, vor fi criptate în baza de date folosind tehnici de criptare moderne (de exemplu, **AES-256**). Astfel, chiar și în cazul unui atac asupra bazei de date, informațiile rămân protejate.

9. **Monitorizare și Audit Continu**:

- **Audit Service** va monitoriza și înregistra toate acțiunile efectuate de utilizatori, inclusiv modificările aduse datelor și proceselor critice. Aceste înregistrări vor fi păstrate într-un jurnal securizat pentru a asigura trasabilitatea și a facilita detectarea oricăror activități suspecte.

10. **Politica de Confidențialitate și Conformitate GDPR**:

- Sistemul va respecta pe deplin reglementările **GDPR**, asigurând protecția datelor personale ale utilizatorilor. Se vor implementa proceduri de **anonimizare** și **pseudonimizare** a datelor pentru a proteja confidențialitatea acestora.

Modulul de Securitate (Security Layer)

Pentru a sprijini aceste măsuri, va fi implementat un **modul dedicat de securitate**, care va include:

- **Autentificare și autorizare avansată** (utilizând **Microsoft Identity** și **OAuth 2.0**).
- **Protecția CSRF și XSS** prin utilizarea token-urilor și sanitizarea input-urilor.
- **Testarea continuă de securitate** cu instrumente precum **OWASP ZAP**.
- **Protecția împotriva atacurilor de tip DoS/DDoS**.
- **Criptarea datelor și protecția acestora prin politici de securitate**.

Aceste măsuri vor asigura protecția aplicației împotriva atacurilor cibernetice și gestionarea corectă a datelor sensibile ale utilizatorilor, acoperind standardele OWASP Top 10 și reglementările GDPR privind securitatea și confidențialitatea.

4. STIVA TEHNOLOGICĂ

Stiva tehnologică a sistemului va fi construită pe o combinație de tehnologii moderne și robuste, care asigură performanță, scalabilitate și securitate. Fiecare strat și modul al arhitecturii beneficiază de soluții tehnologice care vor îndeplini cerințele operaționale și de securitate, menținând un ecosistem integrat și eficient.

4.1. Backend

Tehnologie utilizată: **C# .NET 10**

Motivul alegerii:

- **Performanță ridicată**: .NET 10 oferă o platformă performantă și scalabilă, capabilă să gestioneze volume mari de date și cereri simultane, fiind ideală pentru aplicațiile enterprise.
- **Microservicii**: Suport excelent pentru arhitecturi de microservicii, fiind ideal pentru implementarea componentelor sistemului care funcționează independent, dar colaborează între ele.
- **Compatibilitate și integrare**: Se integrează ușor cu soluții de baze de date, sisteme externe și API-uri. Un alt avantaj important al utilizării **C# .NET** este că multe dintre serviciile guvernamentale sunt deja implementate pe această platformă. Astfel, integrarea cu aceste servicii va fi mult mai ușoară și mai eficientă, dat fiind că **C#** este standardul folosit de instituțiile guvernamentale pentru dezvoltarea sistemelor critice.
- **Securitate**: Framework-ul include suport nativ pentru securizarea aplicațiilor și protejarea acestora împotriva atacurilor de tip **SQL Injection**, **XSS**, și altele, prin utilizarea celor mai bune practici de securitate.

4.2. Frontend

Tehnologie utilizată: **React cu TypeScript**

Motivul alegerii:

- **Interfață rapidă și scalabilă**: React permite construirea unor aplicații web interactive, rapide și scalabile, ce pot gestiona eficient date complexe, cum ar fi rapoarte și grafice de consum.
- **TypeScript**: Utilizarea TypeScript adaugă tipuri statice în JavaScript, reducând erorile de dezvoltare și îmbunătățind întreținerea codului.
- **React Ecosystem**: React se integrează ușor cu soluții moderne de frontend, oferind suport pentru componentizarea aplicațiilor și managementul eficient al stării aplicației.

4.3. Baze de Date

Tehnologii utilizate: **PostgreSQL**, **Redis** și **Elasticseach**

Motivul alegerii:

- **PostgreSQL**: Este o bază de date relațională robustă și open-source, ideală pentru gestionarea datelor structurate ale consumatorilor, facturilor și plăților. Suportă tranzacții ACID, integritate referențială și scalabilitate verticală.
- **Redis**: Este o soluție de caching rapid, utilizată pentru a stoca temporar datele care trebuie accesate frecvent, cum ar fi tarifele active și setările utilizatorilor.
- **Elasticsearch/Solr:** Motor căutare pentru indexare și căutări rapide.

4.4. Stocare Obiecte – MinIO

Tehnologie utilizată: **MinIO** (S3-compatible)

Motivul alegerii:

- **Stocare de fișiere**: MinIO este o soluție compatibilă cu S3, care permite stocarea și gestionarea fișierelor de mari dimensiuni, cum ar fi facturile PDF și documentele asociate.
- **Performanță ridicată**: Este o soluție rapidă și scalabilă pentru stocarea obiectelor, ideală pentru aplicațiile care necesită acces frecvent la fișiere mari.
- **Securitate**: Suportă criptarea datelor atât în tranzit, cât și în repaus, asigurând protecția fișierelor sensibile.

4.5. Comunicarea între Microservicii

Tehnologie utilizată: **gRPC**

Motivul alegerii:

- **gRPC**: Este un framework performant pentru comunicarea între microservicii, utilizând protocolul **HTTP/2** pentru un transfer rapid de date. Este ideal pentru comunicațiile sincronizate între microservicii, asigurându-se că mesajele sunt transmise rapid și în siguranță.
- **Kafka** sau **RabbitMQ**: Ambele sunt soluții excelente pentru gestionarea comunicațiilor asincrone între microservicii. Kafka este ideal pentru aplicațiile care necesită procesarea rapidă a unui volum mare de mesaje și pentru sisteme bazate pe evenimente, în timp ce RabbitMQ este o alegere solidă pentru scenarii în care gestionarea fiabilă a mesajelor și a cozii este crucială. Indiferent de alegerea finală, ambele soluții sunt capabile să facă față cerințelor de scalabilitate și performanță ale sistemului.

4.6. Infrastructură

Tehnologii utilizate: **Docker** și **Kubernetes**

Motivul alegerii:

- **Docker**: Permite crearea unor containere izolate pentru fiecare microserviciu, asigurând consistența mediilor de dezvoltare, testare și producție.
- **Kubernetes**: Orchetrează și gestionează containerele Docker într-un mod eficient, asigurând scalabilitatea și disponibilitatea aplicației. Kubernetes permite distribuirea automată a încărcăturii, monitorizarea stării microserviciilor și implementarea acestora într-un mod sigur.

4.7. Monitorizare și Observabilitate

Tehnologie utilizată: **Grafana**

Motivul alegerii:

- **Grafana**: Este o platformă de vizualizare și monitorizare a datelor care se integrează perfect cu soluțiile de monitorizare a aplicației. Permite crearea de dashboard-uri vizuale pentru a monitoriza performanța aplicației și infrastructurii, precum și pentru a urmări indicatorii cheie de performanță (KPIs).

4.8. Testare de Securitate

Tehnologie utilizată: **OWASP ZAP**

Motivul alegerii:

- **OWASP ZAP**: Este un instrument open-source de testare a securității aplicațiilor web. ZAP va fi utilizat pentru a efectua teste automate de securitate pe aplicație, identificând vulnerabilitățile și asigurându-se că aplicația respectă standardele de securitate.

Exemplu de utilizare: Testarea aplicației pentru vulnerabilități de tip **SQL Injection**, **Cross-Site Scripting (XSS)** și **Cross-Site Request Forgery (CSRF)**, în scopul de a proteja datele utilizatorilor și a preveni atacurile cibernetice.

5. ROLURI ȘI PERMISIUNI

Pentru a asigura un control adecvat asupra accesului utilizatorilor și a proteja integritatea și securitatea datelor, sistemul va utiliza un model de Control al Accesului pe Baza de Roluri (RBAC - Role-Based Access Control). Fiecare utilizator al aplicației va avea un rol specific, care determină permisiunile și resursele la care poate accesa. Sistemul va implementa, de asemenea, mecanisme de securitate suplimentare, cum ar fi autentificarea în două etape (2FA), pentru a întări protecția conturilor utilizatorilor și a preveni accesul neautorizat.

5.1. Roluri în Sistem

1. **Administrator (Admin)**:

- **Descriere**: Administratorii sunt utilizatori cu drepturi extinse, având acces complet la toate funcționalitățile aplicației. Aceștia sunt responsabili pentru gestionarea utilizatorilor, configurarea proceselor, administrarea tarifelor și vizualizarea rapoartelor.
- **Permisiuni**:

  - Gestionarea utilizatorilor și rolurilor (adăugare, modificare, ștergere).

  - Configurarea tarifelor și proceselor de calcul.

  - Vizualizarea și generarea rapoartelor financiare și operaționale.

  - Modificarea setărilor de securitate și acces.

2. **Operator**:

- **Descriere**: Operatorii sunt utilizatori care efectuează activități de operare zilnică, cum ar fi importul de date, calcularea facturilor și gestionarea proceselor de reconcilieri financiare.

- **Permisiuni**:

  - Acces la dashboard-ul operațional.

  - Gestionarea proceselor de import și calcul al facturilor.

  - Vizualizarea și actualizarea statusului proceselor în desfășurare.

  - Acces la rapoarte de consum și facturare.

3. **Consumator (opțional, pentru extindere uletrioară)**:

- **Descriere**: Consumatorii sunt utilizatori care ar putea avea acces la propriile date. Aceștia pot vizualiza facturile și istoricul plăților, dar acest rol este **opțional** și depinde de natura aplicației, care poate fi un sistem intern de gestionare a consumatorilor sau un sistem extern de interacțiune cu utilizatorii finali.

- **Permisiuni**:

  - Vizualizarea facturilor și istoricului plăților.

  - Acces la informațiile personale și istoricul consumurilor.

  - Gestionarea opțiunilor de plată și de configurare a contului personal.

4. **Contabil**:

- **Descriere**: Contabilul este un utilizator care are acces doar la informațiile financiare legate de conturi și tranzacții, fără a avea acces la datele de configurare ale sistemului.

- **Permisiuni**:

  - Vizualizarea și gestionarea tranzacțiilor financiare.

  - Generarea rapoartelor financiare și a balanțelor contabile.

  - Acces la funcționalitățile de reconciliere și ajustare financiară.

5.2. Permisiuni Detaliate pe Module

Fiecare modul al aplicației va avea permisiuni specifice, în funcție de rolul utilizatorului. Permisiunile vor fi controlate printr-un sistem RBAC robust, care va asigura că utilizatorii pot accesa doar resursele și funcționalitățile necesare activităților lor.

Modul	Administrator	Operator	Consumator	Contabil
API Gateway	Da	Nu	Nu	Nu
Import Service	Da	Da	Nu	Nu

Calculation Service	Da	Da	Nu	Nu
Recalculation Service	Da	Da	Nu	Nu
Invoicing Service	Da	Da	Nu	Da
Bank Integration Service	Da	Da	Nu	Nu
Dashboard	Da	Da	Nu	Da

- **Administrator** va avea acces complet la toate modulele, inclusiv configurarea acestora și gestionarea utilizatorilor.
- **Operator** va avea acces doar la modulele operaționale, cum ar fi importul datelor, calculul și facturarea.
- **Consumator (opțional, pentru extindere ulterioară)** va avea acces doar la datele proprii, fără posibilitatea de a modifica sau vizualiza alte date.
- **Contabil** va avea acces la modulele financiare pentru generarea rapoartelor și gestionarea tranzacțiilor, dar nu are acces la modulele de configurare.

5. PERFORMANȚA ȘI FIABILITATEA SISTEMULUI

Sistemul va utiliza procesarea asincronă pentru generarea intrărilor și ieșirilor.

Timpul de răspuns al operațiunilor standard va fi până la 3 (trei) secunde.

Sistemul va avea capacitatea de a suporta activitatea concomitentă a cel puțin 100 utilizatori.

Sistemul va fi proiectat pentru a gestiona și procesa volume mari de date, cu un prag de minimum 900.000 de consumatori înregistrați, fără degradarea performanței operaționale.

Sistemul va suporta mecanisme de import masiv de date, asigurând procesarea a cel puțin 500.000 de înregistrări pe oră, cu validare, jurnalizare și tratarea erorilor la nivel de înregistrare.

6. TESTAREA DE ACCEPTANȚĂ A SISTEMULUI

Pentru organizarea testării de acceptanță, vom efectua următoarele activități:

- elaborarea Planului detaliat de testare
- pregătirea scenariilor de teste.
- documentarea erorilor/deficiențelor depistate pe parcursul verificării scenariilor de testare și înlăturarea lor;

- elaborarea unui Raport cu rezultatele finale ale testării sistemului care să conțină inclusiv statutul tuturor erorilor/deficiențelor.

Vor fi elaborate scenariile de testare în colaborare cu Beneficiarul și organizate și documentate 4 tipuri de testări după cum urmează:

- Testarea Funcțională (Functional Testing): Pe parcursul testelor funcționale, toate procesele reale vor fi simulate cap-coadă pentru a verifica dacă sistemul procesează și stochează date în mod corespunzător;
- Testarea Performanței (Load and Stress Testing): Deoarece sistemul urmează a fi exploatat de mai mulți utilizatori autorizați, trebuie efectuate teste de performanță pentru a verifica cum funcționează sistemul în cazul diferitor sarcini. Acest lucru poate necesita optimizare configurațiilor serverului web, a software-ului de aplicație și/sau a serverului de baze de date sau a configurațiilor rețelei;
- Testarea Vulnerabilității (Security Testing): Este necesar să fie efectuate teste detaliate de securitate în conformitate cu cerințele de asigurare a securității informațiilor. Testarea de securitate va verifica existența vulnerabilității sistemului la atacuri cum ar fi: atacul de injecție SQL, XSS, atacuri DDoS, etc. Testarea de securitate va face uz de mijloace software de detectare a amenințărilor și vulnerabilităților de securitate a sistemului.

7. MANAGEMENTUL PROIECTULUI

Procesul de gestionare proiectului de către echipa de implementare a proiectului va fi realizată în cadrul aplicației Redmine (http://redmine.rts.md/) combinată cu metoda de dezvoltare AGILE.

Redmine este o aplicație web de gestionare a proiectelor și de urmărire a progresului. Include funcționalități avansate de planificare, delegare, monitorizare și raportare a sarcinilor în cadrul proiectului, inclusiv vizualizarea în forme grafice - calendarul Gantt.

La realizarea Planului de implementare a proiectului, echipa de proiect va folosi metodologia SCRUM, care va ține cont de următoarele reguli:

1) Definirea tuturor cerințelor de business care să fie acoperite de proiect.

Se va obține în proces de consultare și coordonare cu Autoritatea contractantă, pentru a acoperi toate cerințele de business și tehnice necesare pentru soluția comandată.

2) Crearea listei prioritate de cerințe (product backlog), în baza cerințelor de business definite.

Se va obține prin prezentarea viziunii dezvoltatorului și coordonarea de către Autoritatea contractantă.
3) Divizarea proiectul în iterații (sprint-uri) a câte 2 săptămâni.
4) Lansarea fiecărei iterații va fi realizată de către echipa de proiect printr-o ședință de planificare (sprint planning), în care din product backlog se va crea backlog-ul iterației. Toate cerințele vor fi discutate și detaliate la nivel de sarcini pentru fiecare membru implicat din echipă și documentate în Redmine.
Regula de aur – după lansarea iterației, nu se mai intervine cu schimbări.
5) În etapa de realizare a iterației, echipa de proiect începe ziua de activitate cu desfășurarea unei ședințe zilnice de progres (daily scrum) de cel mult 15 min pentru ca fiecare membru să poată răspunde la următoarele trei întrebări: Ce am făcut ieri? Ce am de gând să fac azi? Cu ce obstacole mă confrunt?
6) La finalizarea unei iterații echipa lansează funcționalitățile dezvoltate într-o versiune nouă de soluție, fac testările necesare și se conving că au fost realizate toate sarcinile trasate în iterație.
7) De asemenea, la finele iterației se organizează și o ședință cu reprezentanții Autorității contractante (sprint review) în care se prezintă cerințele care au fost acoperite și versiunea demo a soluției realizate în cadrul iterației pentru acceptare (prezentare pe platforma de pre-producere). De asemenea, în cadrul acestei ședințe se va coordona cu Autoritatea contractantă prioritățile funcționalităților incluse în următorul sprint pentru echipa de proiect.

Ședințele comune de regulă le desfășurăm în 3 formate:
1. Sediul autorității contractante;
2. Sediul nostru;
3. Online (videoconferințe).

De menționat că, din practica ultimilor 4 ani, cel mai optat format de desfășurare de către beneficiari, este cel online, datorită optimizării timpului și resurselor. Iar toate cele dispuse în cadrul ședinței obligatoriu se consemnează în cadrul unui proces-verbal.

Încheierea unei iterații de către echipa de proiect este realizată printr-o ședință de închidere (sprint retrospective) în care echipa examinează ce a mers bine și ce a mers rău în cadrul iterației și ia decizii de optimizare și îmbunătățire pentru iterația următoare.

Actorii principali de interacțiune dintre dezvoltator și Autoritatea contractantă în procesul de dezvoltare:
1. Product Owner – managerul delegat din partea Autorității contractante, care va răspunde la întrebările apărute la echipa de proiect și va coordona mersul dezvoltării produsului, fără a periclita ritmul de dezvoltare a soluției.
2. Manager de proiect – managerul delegat din partea dezolatorului care va asigura gestionarea proiectului.

3. **Scrum Master** – managerul delegat din partea dezvoltatorului care va asigura implementarea proiectului conform documentației proiectului și respectând regulilor prevăzute de metodologia SCRUM.

8. INSTRUIREA UTILIZATORILOR

Pentru transmiterea cunoștințelor, în colaborare cu Beneficiarul, vor fi organizate sesiuni de instruire pentru utilizatorii Beneficiarului, pentru fiecare rol în parte: administratori, operatori și contabilitate.

Pentru organizarea și desfășurarea sesiunilor de instruire vor fi elaborate materiale de instruire în limba română.

Sesiunile de instruire vor conține prezentarea materialului teoretic, dar și aplicarea în practică în cadrul platformei de pre-producere oferită de noi.

Sesiunile de instruire vor fi desfășurate offline la sediul Beneficiarului sau online prin intermediul sistemelor de videoconferință.

9. GARANȚIA PRODUSULUI DEZVOLTAT

Pentru produsul dezvoltat va fi asigurată o garanție de 6 luni de la momentul lansării în exploatare industrială a sistemului.

În perioada de garanție vom asigura:
- acordarea suportului tehnic;
- eliminarea defecțiunilor raportate de Beneficiar;
- soluționarea tuturor incidentelor raportate de Beneficiar în conformitate cu SL:

Severitate	Timp răspuns	Timp rezolvare
Critică (sistem nefuncțional)	1 oră	6 ore
Ridicată (funcționalitate majoră)	2 ore	15 ore
Medie (funcționalitate minoră)	1 zi	3 zile
Scăzută (îmbunătățiri)	2 zile	5 zile

Vom pune la dispoziția Beneficiarului un serviciu Help Desk pe care beneficiarul îl va putea apela la un număr de telefon național, în limba română sau rusă.

Adițional Beneficiarul va putea semnala problemele tehnice apărute prin email sau canal în comunicatoare online.

Toate solicitările în perioada de garanție vor fi documentate și tratate în carul sistemului Redmine, iar la necesitate vor fi raportate beneficiarului.

10. PRODUSUL FINAL ȘI COMPONENTELE LIVRATE

Produsul final livrat va include toate artefactele software și de documentare ale sistemului, precum și transferul de cunoștințe către Beneficiar, și va cuprinde:

1. Sarcina tehnică a sistemului.
2. Prototipul sistemului.
3. Documentația de testare (planuri de testare, procese-verbale).
4. Documentația de instruire (prezentări, programe, agende, fișe de instruire).
5. Ghidul de instalare, configurare, build și deployment.
6. Manualul de utilizare (divizare pe roluri).
7. Manualul de administrare.
8. Codul sursă.