TEHNIČNE IN FUNKCIONALNE SPECIFIKACIJE ZA TEHNOLOŠKO PRENOVO BALANCING SISTEMA(BS) IN AUCTION PLATFORM (AP)

Vsebina

1. UVOD 7
1.1. Kratice in definicije 7
2. Popis zahtev 9
2.1. Splošne zahteve 9
2.2. Splošne varnostne zahteve 10
2.3. Splošne zahteve uporabe infrastrukturnih strežniških virov 12
2.4. Uporabnost 12
2.5. Razpoložljivost 13
2.6. Zanesljivost 14
2.7. Zmogljivost 14
2.8. Skalabilnost 15
2.9. Komunikacijske zahteve 16
2.10. Upravljanje s popravki/spremembami 17
2.11. Licenciranje in izvorna koda 18
2.12. Produkcijsko, razvojno in testno okolje 19
2.13. Zakonske zahteve in interna pravila 19
2.14. Projekt za izvedbo (PZI) in Projekt izvedenih del (PID) in ostala spremljajoča dokumentacija 20
3. PRIČAKOVANE SPREMEMBE 23
3.1. Tehnološka prenova BS sistema 24
3.1.1. Komunikacijski protokol HTTP/2 24
3.1.2. Microsoft .NET 8/9 24
3.1.3. MSDTC 26
3.2. LOČITEV IN TEHNOLOŠKA PRENOVA AP SISTEMA 27
3.2.1. Dražbe splošno 27
3.2.2. FCR dražbe 31
Definiranje vlog 31
Meje in območja 32
Anonimizacija dodanega PSI 33
Visokonivojski opis procesa FCR dražb 33
3.2.3. Dražbe 42
Sprememba pravil v procesu dražb 42

Set dražb 42
Načini plačila 42
Preprečevanje uhajanja informacij pri izvajanju dražb 42
Tehnična sposobnost in kvalificirane količine PSI 43
Avtomatizem dražb 44
3.2.4. XML transformacija: 46
1.1.1. Dokumenti 47
Dodajanje in urejanje dokumentov 47
Pregled in iskanje dokumentov 47
1.1.2. Sistem za obveščanje 47
Operator Schedules (urniki) 47
Načini obveščanja (alarmi) in filtriranje dogodkov 48
Sestavljanje izhodnih tekstov 49
Opis tipov obveščanja 50
Nov izgled okna sistemskih notifikacij in odgovorov 53
1.1.3. Konfiguracija časovnih serij (Time Series Configuration) 55
1.1.4. Poročila 57
1.1.5. Pomoč uporabniku – Integrirana tehnična dokumentacija 60
1.2. PILOTSKI PROJEKT – kontejnerske tehnologije 60
1.2.1. Pričakovanja in smernice uporabe kontejnerskih tehnologij 61
Docker 62
Razlika med Docker in virtualno napravo (VM) 62
Kreiranje kontejnerja 62
YML datoteka 63
Docker-compose 63
Registri kontejnerjev 63
Shema dockeriziranega okolja 64
Servisi/kontejnerji 64
Inicialna namestitev 65
Nadgradnja sistema 65
Priloge 65
1.2.2. Priprava in analiza obstoječe aplikacije 67
Priprava Docker infrastrukture 67
Testiranje delovanja aplikacije v kontejnerskem okolju 68

Identifikacija težav in ocena izvedljivosti 68
Spremljanje sprememb pri podpori aplikacij 69
Priprava dokumentacije in priporočil 70
Rezultat projekta 70
2. TESTIRANJE 71
Pri izvajalcu 71
Pri naročniku 71

Kazalo slik

Slika 2.9-1 Umestitev IS v Elesovo okolje 17
Slika 2: Shema dokeriziranega razvojnega okolja 64

Kazalo tabel

Nobenega vnosa v kazalo slik ni bilo mogoče najti.

1. UVOD

Prenova BS aplikacije v osnovi zajema prehod iz .NET Framework 4.8 na .NET 8, migracijo iz WCF na gRPC in prehod iz uporabe SQL Stored Procedures na EF. Ti koraki so ključni za izboljšanje zmogljivosti, varnosti in dolgoročne vzdržnosti aplikacije.

Prehod na .NET 8 prinaša več pomembnih prednosti, zlasti na področju varnosti:

- Aktivna podpora in posodobitve: .NET 8 je najnovejša različica z dolgoročno podporo (LTS), kar zagotavlja redne varnostne posodobitve in izboljšave.
- Boljši mehanizmi za zaščito: .NET 8 vključuje napredne varnostne funkcije, kot so izboljšano upravljanje certifikatov, podpora za varnejše šifriranje in integracija z modernimi varnostnimi standardi.

Migracija na gRPC izboljšuje učinkovitost in varnost komunikacij:

- Hitrejša komunikacija: gRPC uporablja protokol HTTP/2, ki omogoča večjo pretočnost podatkov, kompresijo in več sočasnih povezav, kar zmanjšuje zakasnitve.
- Vgrajena varnost: gRPC privzeto podpira TLS, kar zagotavlja šifrirano komunikacijo med odjemalcem in strežnikom.

Prehod na EF pa omogoča izboljšano vzdrževanje in razvoj aplikacije:

- Zmanjšana kompleksnost: EF poenostavi delo z bazo podatkov, kar omogoča lažje spreminjanje in širjenje podatkovnega modela.
- Večja preglednost: Poslovna logika je neposredno v programski kodi, kar izboljšuje preglednost in sledenje spremembam v sistemu.
- Standardizacija: EF omogoča uporabo sodobnih pristopov k delu z bazo podatkov, kot so LINQ poizvedbe, ki povečujejo berljivost in skrajšajo razvojni čas.

1.1. Kratice in definicije

ACER	Agency for the cooperation of energy regulators	
AD	Active Directory	Aktivni imennik
AMQP	Advanced Message Queue Protocol	
AP	Auction Platform	
API	Application Programming Interface	
BPM	Business Process Management	
BS	Balancing System aplikacija	

ECP/EDX	Electronic Communication Protocol/Energy Data Exchange platform	
EE		Električna energija
EES		Elektroenergetski sistem
EF	Entity Framework	
ENTSO-E	European network of transmission system operators for electricity	
EU		Evropska unija
FSSF	File System Shared Folder	
gRPC	Google Remote Procedure Calls	
GUI	Graphical User Interface	
HTTP	Hypertext Transfer Protocol	
LINQ	Language Integrated Query	
ITSM	Information technology service management	
IS	Information system	Informacijski sistem
MS	Microsoft	
MTU	Market Time Unit	Časovni interval izmenjave
ODF	OpenShift Data Foundation	
OPS	Operater Prenosnega Omrežja (TSO)	
PS	DataBase/DataStore	Podatkovno skladišče
PZI		Projekt za izvedbo
RBAC	Role-based access control	
RCV		Republiški center vodenja
RPO	Recovery Point Objective	Čas izgube podatkov ob katastrofi
RSA	Regional security analysis	
RTO	Recovery Time Objective	Čas do povrnitve storitve po katastrofi
SA	Security asessment	Sigurnostna analiza

SAC	Security Assessment Coordinator	Koordinator za sigurnostno analizo
sFTP	Secure file transfer protocol	
SO	System operator	Sistemski operater
SMB	Server Message Block	
TLS	Transport Layer Security	
TP	Transparency Platform	
TSO	Transmission system operator	Sistemski operater prenosnega omrežja
UML	Unified Modeling Language	
WCF	Windows Communication Foundation	

2. Popis zahtev

2.1. Splošne zahteve

Tehnologija mikrostoritev:

Cilj je zagotoviti arhitekturo, ki strukturira aplikacije kot nabor ohlapno povezanih sodelujočih storitev, vsaka storitev pa mora omogočati:

• Visok nivo možnosti vzdrževanja in testiranja, kar omogoča hiter in dinamičen razvoj in uvajanje sprememb

• Ohlapna povezava z drugimi storitvami vzdrževalcem in razvijalcem omogoča neodvisno delo, saj storitve niso soodvisne.

• Možna sočasna neodvisnega uvajanja in vzdrževanja, saj je vsaka storitev izolirana in neodvisna.

Mikro storitev je samostojen kos programske opreme, ki zagotavlja poslovno funkcionalnost z jasnimi vmesniki in lahko prek lastnih notranjih komponent izvaja lastno shranjevanje podatkov, poslovno logiko in uporabniški vmesnik.

Mikro storitev je primerna za neprekinjen proces razvoja programske opreme (CI/CD). Sprememba manjšega dela aplikacije zahteva le vnovično izgradnjo in prerazporeditev le ene ali manjšega števila storitev.

Uporaba mikrostoritev mora mogočati individualno skalabilnost. Podprta mora biti razširitev samo posameznih mikrostoritev, ki podpirajo funkcijo z omejitvami virov. Tako je zagotavljana optimizirana raba virov in stroškovna učinkovitost izvajanja rešitve.

• Upoštevanje modularne gradnje –zagotavlja lažje razumevanje aplikacije, lažji razvoj, testiranje pa postane bolj odporna na erozijo arhitekture.

• Zagotavljanje razširljivosti – mikrostoritve implementirane in razporejene neodvisno druga od druge, kjer tečejo znotraj neodvisnih procesov, je mogoče nadzorovati, prilagajati in širiti neodvisno.

• Zagotovljen porazdeljeni razvoj – omogoča majhnim avtonomnim ekipam, tudi različnih ponudnikov, da neodvisno razvijajo, nameščajo in prilagajajo svoje posamezne komponente. Omogoča pa se tudi nastanek arhitekture posamezne storitve z nenehnim preoblikovanjem. Arhitekture, ki temeljijo na mikrostoritvah, omogočajo neprekinjeno vzdrževanje, dostavo in uvajanje.

Modeliranje:

- Načrtovanje in modeliranje okvirja rešitve mora biti zagotavljano v standardiziranem jeziku za modeliranje in načrtovanje poslovnih procesov BPMN 2.x – Business Process Management (BPM) in zabeleženo UML diagramih.

Sestava:

- Poslovna funkcionalnost mora biti implementirana v aplikacijo z uporabo vzorcev integracije mikrostoritve in mikro vmesnika, ohlapno povezanih prek sporočanja, ki temelji na dogodkih, te mikrostoritve so integrirane v plasti sestave po omejenih kontekstih, definiranih v modelu domene. Ogrodje mora podpirati avtomatizacijo poslovnih procesov, ki se izvajajo v takih okoljih.

Avtonomija:

- V okolju mikro storitev mora biti odvisnost med mikro storitvami omejena na absolutni minimum, to velja tudi z vidika BPM, zlasti če so te mikro storitve nameščene v porazdeljenem okolju. Četudi sama avtonomija nima strogih zahtev glede distribucije, mora biti zagotovljena v kontekstu poslovnih procesov, ki se izvajajo vzdolž omejenih kontekstov, ki prekrivajo več funkcij.

2.2. Splošne varnostne zahteve

- Uporabniški vmesnik mora biti izveden kot spletna aplikacija; objavljena mora biti samo preko varne povezave (https).
- Uporabniški vmesnik mora delovati na standardnih spletnih brskalnikih brez namestitve vtičnikov ipd. (vsaj MS Edge).
- Sistem mora beležiti dnevniške zapise in revizijske sledi.

- Sistem mora beležiti in omogočati pošiljanje dnevniških zapisov in revizijskih sledi v obliki syslog RFC5424. Zahtevana je izvedba integracije s SIEM sistemom ELES.
- Vse povezave, ki se uporabljajo za izmenjavo podatkov znotraj ELES omrežja morajo biti šifrirane.
- Šifriranje podatkov in komunikacij mora biti izvedeno s sodobnimi močnimi šifrirnimi algoritmi in metodami. Uporaba šibkih šifrirnih algoritmov in metod ne sme biti mogoča.
- Prijava v storitev (avtentikacija) mora potekati na način, ki onemogoča nepooblaščeno razkritje ali uporabo avtentikacijskih podatkov (Kerberos avtentikacija in komunikacija z MS AD z LDAPs).
- Uporabniški vmesnik mora podpirati uporabo sistema enkratne prijave (SSO), MS AD.
- Sistem mora podpirati standardne vmesnike za integracijo z MFA (multifaktorska avtentikacija) sistemi.
- Avtentikacijski podatki se ne smejo prenašati ali biti shranjeni v nešifrirani obliki.
- Omogočeno in popisano mora biti spreminjanje gesel vseh računov s privilegiranimi pravicami.
- Sistem mora podpirati standardne vmesnike in protokole (RDP in SSH) za dostop do upravljanja s PAM (upravljanje privilegiranih računov) sistemi, za katero aktualna navodila zagotovi naročnik ob pripravi PZI,
- Sistem mora podpirati standardne vmesnike za integracijo z IM (upravljanje z identitetami) sistemi – najmanj API vmesnik za dostop do vlog, lokalnih uporabnikov in njihovih pravic,
- Zahtevana je izvedba ustrezno globoke integracije z IM sistemom ELES, za katero aktualna navodila zagotovi naročnik ob pripravi PZI,
- Zahtevana je vključitev novega SW/HW v centralne nadzorne sisteme PITK, za katero aktualna navodila zagotovi naročnik ob pripravi PZI,
- Zahtevano je upravljanje dostopov na podlagi vlog (IEC 62351-8 - RBAC) novega IS, ki jih uporabniki/administratorji sistema lahko upravljajo samostojno. Pripadajoče lokalne vloge IS morajo biti dostopne in dodeljene članom uporabniških skupin MS AD aktivnega imenika,
- Za shranjevanje in upravljanje vsebnikovskih slik (Docker images) bo naročnik zagotovil »image repository - registry mirror«, kjer pa bo zagotovljeno tudi varnostno pregledovanje vsebin slik. Ponudnik bo moral zagotoviti prenos in pregled slik na dogovorjeno mesto, hkrati pa tudi uporabo slik za izvajanje kontejenrjev/podov iz registry mirrorja. Aktualna navodila zagotovi naročnik ob pripravi PZI.
- Koda mora biti podpisana z javnim code-signing certifikatom
- Ponudnik pred implementacijo rešitve izvede kibernetsko varnostno analizo rešitve in izvorne kode s strani kvalificirane neodvisne organizacije in prestavi povzetke analize.

2.3. Splošne zahteve uporabe infrastrukturnih strežniških virov

- Prenovljen IS mora delovati na strežniku Microsoft Windows Server 2022 ali novejši,
- Prenovljen IS mora podpirati delovanje na rešitvi za virtualizacijo vmWare 7.0. ali višje.
- Uporabniški del IS – spletni uporabniški vmesnik, mora biti izveden na spletnem strežniku IIS 10.0 (ali novejši),
- PoC IS mora podpirati izvajanje v storitev, oziroma mikrostoritev v infrastrukturi tehnologije RedHat OpenShift,
- PoC IS lahko izrablja za optimalno izvajanje v storitev, oziroma mikrostoritev sposobnosti tehnologije RedHat OpenShift,
- PoC IS lahko izrablja za izvajanje storitev, oziroma mikrostoritev hranjenje podatkov na tehnologiji Red Hat OpenShift Data Foundation (ODF),
- PoC IS lahko za izvajanje v storitev, mikrostoritev, oziroma njihovo interno hrambo podatkov v podatkovnih bazah uporablja tehnologije podatkovnih baz Microsoft SQL server ali PostgreSQL. Obe tehnologiji v kontejnerskih različicah zadnjih aktualnih verzij.
- IS mora za centralno hranjenje in deljenje podatkov uporabljati podatkovno bazo Microsoft SQL 2019. Naslov za povezovanje na MS SQL operativno podatkovno bazo IS. Varovanje podatkov pred izgubo in zagotavljanje visoke razpoložljivosti zagotavlja naročnik z uporabo lastnih tehnologij.
- Uporabnik/Naročnik bo opredelil zahteve po pogostosti (časovni interval) varnostnega kopiranja in čas hranjenja varnostnih kopij, ki jo mora izvajalec tudi upoštevati. Sistem mora podpirati izvedbo varnostnega kopiranja s produktom Veeam Kasten in Veeam Backup & Recovery, ki bo temeljilo na internih pravilnikih naročnika.
- Uporabnik/Naročnik bo opredelil pogostost (časovni interval) arhiviranja podatkov, medtem, ko mora IS omogočati delitev in oblikovanje arhivskih podatkov, kar bo po potrebi definirano pri oblikovanju vsebin PZI,
- Zagotovljeno naj bo ločevanje upravljalskega/administrativnega in uporabniškega vmesnika.

2.4. Uporabnost

IS mora biti zasnovana kot sodobna rešitev, ki omogoča uporabniku na čim bolj enostaven, učinkovit in prijazen način izvesti vse poslovne procese predvidene za zagotavljanje zakonskih zahtev. Rešitev mora omogočati standardizacijo in avtomatizacijo procesov povsod tam, kjer je to mogoče in smotrno.

IS se mora vidno odzivati na uporabnikove akcije. Vsaka akcija povzroči vidno (logično in prepoznavno) spremembo na uporabniškem vmesniku. Ko orodje ni odzivno, uporabniku to vidno prikaže (sprememba kurzorja, prikaz poteka postopka, proženje dogodka s prikazom dogodka v novem oknu »pop-up«, zapis v dnevnik dogodkov in podobno).

Informacijska rešitev mora zagotavljati enostavno (intuitivno) uporabo funkcionalnosti. Kjerkoli je pričakovana takojšnja interakcija z uporabnikom, mora rešitev uporabniku ponuditi takojšen odziv in biti čim bolj prilagojena učinkoviti uporabi (čim manjše število korakov za izvedbo določenega postopka, čim hitrejši dostop do kakovostnih informacij, …).

Ključna zahteva glede funkcionalnosti rešitve je enostavnost uporabe in hitra odzivnost.

Uporabniški vmesnik mora slediti naslednjim smernicam:

- ustrezno segmentiranje in zaporedje vnosnih polj uporabniškega vmesnika, ki uporabniku dajeta jasno informacijo glede statusa in procesa zajema podatkov,
- kontrola pravilnosti in veljavnosti vpisanih podatkov, kjer je to mogoče (ali z znanim algoritmom ali s preverjanjem podatkov v zunanjem viru),
- uporaba pred izpolnjevanja ali izbire možnih vrednosti, kjer je to le mogoče,
- izvedba posameznih aktivnosti v čim manj korakih (optimizirani procesi v uporabniškem vmesniku programske opreme),
- vnos podatkov na enem mestu, ponovna uporaba podatkov, ki so že zajeti v rešitev ali v drugih povezanih informacijskih sistemih, s katerimi je zagotovljena izmenjava podatkov. Enkratni zajem oziroma vnos podatkov v šifrante ter avtomatični prenos teh podatkov v vse module rešitve, ki uporabljajo te šifrante,
- komunikacija z uporabnikom, kjer so na enoten način uporabljeni elementi, kot so sporočila o napakah, obvestila o napredku obdelave in podobno,
- elementi (grafični gradniki) morajo biti oblikovani tako, da v različnih brskalnikih in operacijskih sistemih omogočajo podoben videz in enakovredno funkcionalnost,
- varno skupno delo več uporabnikov na enem zapisu v smislu skupne rabe (vsebine lahko ureja več uporabnikov hkrati) ali pa možnost vsaj hkratnega ogleda vsebine (pri čemer lahko vsebino ureja le en uporabnik hkrati).

2.5. Razpoložljivost

Zagotavljaje neprekinjenega poslovanja produkcijskega IS bo naročnik zagotavljal možnost izvajanja strežnikov na dveh ali treh lokacijah z uporabo tehnologij replikacije virtualnih strežnikov z uporabo z vSphere in vSAN - na nivoju vSphere metro gruče.

Zagotavljaje neprekinjenega poslovanja bo naročnik zagotavljal možnost izvajanja PoC storitev v POD-ih na dveh ali treh lokacijah z uporabo tehnologij kontejnerskih tehnologij RedHat OpenShift. Ponudnik pa mora zagotavljati izvajanje posameznih storitev vsaj v dveh kopijah, ki so medsebojno povezani in sinhrone v svojih nalogah.

Na nivoju MS SQL strežnika zagotovljena replikacija podatkovnih baz in dostopa do njih z uporabo »avilability group« na obeh lokacijah podatkovnega centra. Sicer pa se pričakuje od IS, da sledi zahtevam načrt neprekinjenega poslovanja uporabnikov IS s sledečimi zahtevami:

- RTO: 12 ur
- RPO: 0,25 ure

Ostale zahteve zagotavljanje primerne razpoložljivosti, bo naročnik po podpisu pogodbe zagotovil iz vsebin lasnih pravilnikov zagotavljanja razpoložljivosti delovanja kritične infrastrukture.

2.6. Zanesljivost

Zaradi zahteve po točnosti podatkov, ki se bodo uporabljali v okviru informacijske rešitve, je treba zagotoviti zanesljivost rešitve in njenih podatkov. V tem je treba zagotoviti več varnostnih in kontrolnih mehanizmov, ki bodo omogočali, da so podatki, ki jih uporabnik zahteva, celoviti, točni in odražajo stanje uporabnikovih zahtev.

Zanesljivost delovanja z vidika uporabnika (točke vnosa podatkov) pri vnosih podatkov mora biti zagotovljena z vgrajenimi kontrolami, ki v najvišji meri onemogočajo nepravilen vnos (predvsem morajo biti zagotovljene na nivoju občutljivih podatkov), kot na primer:

- preverjanje dolžin polj, podatkovnega tipa in formata podatka,
- vključitev drugih kontrol (npr. datum začetka del ne sme biti poznejšo kot datum konca del), ki so stvar dogovora in
- odstranjevanje odvečnih presledkov na začetku in koncu polja.

Visoko zanesljivost je treba zagotoviti tudi na nivoju podatkov. V okviru obravnave podatkov mora biti zagotovljeno preverjanje celovitosti podatkov, kjer mora informacijska rešitev zagotoviti, da podatki niso v nobenem trenutku neskladni. V primeru, da pride do napak ali izpada sistema, mora imeti informacijska rešitev zagotovljen mehanizem, ki bo sistemu omogočal prehod v predhodno konsistentno stanje. Prav tako je potrebno v primeru napak zagotavljati mehanizem, ki zagotavlja samodejno zaznavanje napak in proži ustrezne alarme/opozorila.

2.7. Zmogljivost

IS mora zagotavljati:

- nemoteno delovanje in izvajanje funkcionalnosti vsem uporabnikom,
- nemoteno izmenjavo podatkov z drugimi informacijskimi sistemi, kot je opredeljeno v funkcionalnih zahtevah,
- rešitev mora upoštevati, da celotna izmenjava podatkov temelji na Close-to-Real Time in je dopustna latenca pri izmenjavi podatkov 60 s (round trip time (RTT) pošiljatelj-prejemnik),

- v nobenem primeru ob interaktivnem delovanju ne sme programska oprema ponudnika obdelovati podatkov več kot 180 s, po tem, ko je prejela vse podatke iz ostale infrastrukture,
- odzivni čas uporabniškega vmesnika za pregled podatkov in prikaz poročil ter izvajanje obdelav bo natančno opredeljen v okviru priprave PZI. Zahteva naročnika je, da se v okviru priprave PZI opredeli in uskladi tipe vpogleda v podatke in tipe poročil, način njihove priprave (preko spletnega vmesnika, asinhrona priprava poročil v ozadju, predpriprava poročil itd.) ter zahtevane odzivne čase za posamezno kategorijo. Pri tem mora izvajalec zagotoviti takšne odzivne čase za pripravo vpogledov v podatke in poročila, da bodo omogočali zadovoljivo uporabniško izkušnjo,
- posameznih aktivnosti v čim manj korakih, kar zagotavlja optimiziranost procesov v programski rešitvi, vključno z optimalnimi izvajanji uporabe podatkovne baze (optimalni SQL klici, zaklepanja vsebin, primerna indeksiranja in podobno).

V primeru, da izvajalec zahtevanih odzivnih časov ne more doseči, mora identificirati vzrok, ki ga bo obravnaval skupaj z naročnikom. V primeru, da vzrok za performančne težave ne izvira iz drugih sistemov (ki se npr. odzivajo prepočasi pri izmenjavi podatkov itd.) ali iz informacijskih storitev, za katere je zadolžen naročnik, bo moral izvajalec le-tega odpraviti.

2.8. Skalabilnost

Naročnik bo izdela načrt uporabe in števila uporabnikov in transakcij, kar mora izvajalec upoštevati pri načrtovanju in izdelavi IS.

Informacijsko rešitev bodo uporabljali večina zaposlenih pri naročniku. Ne glede na to, da bo informacijsko rešitev na začetku uporabljalo fiksno število uporabnikov, se mora upoštevati, da se bo število uporabnikov v prihodnosti povečevalo. Zato mora biti informacijska rešitev prilagojena na način, ki bo omogočal enostavno povečevanje zahteve po številu uporabnikov. Povečevanje števila uporabnikov oziroma drugih odjemalcev v okviru pričakovanih obsegov uporabe informacijske rešitve ne sme vplivati na poslabšanje zmogljivosti sistema.

Izvajalec mora upoštevati načrtovane količine podatkov in primerno oblikovati upravljanje z uporabljeno podatkovno hrambo, tako na datotečnem prostoru, kot v podatkovnih bazah. Tako mora primerno načrtovati in omogočati izvedbo delitve operativnih in arhivskih vsebin/podatkov, kar se lahko določi ob oblikovanju PZI.

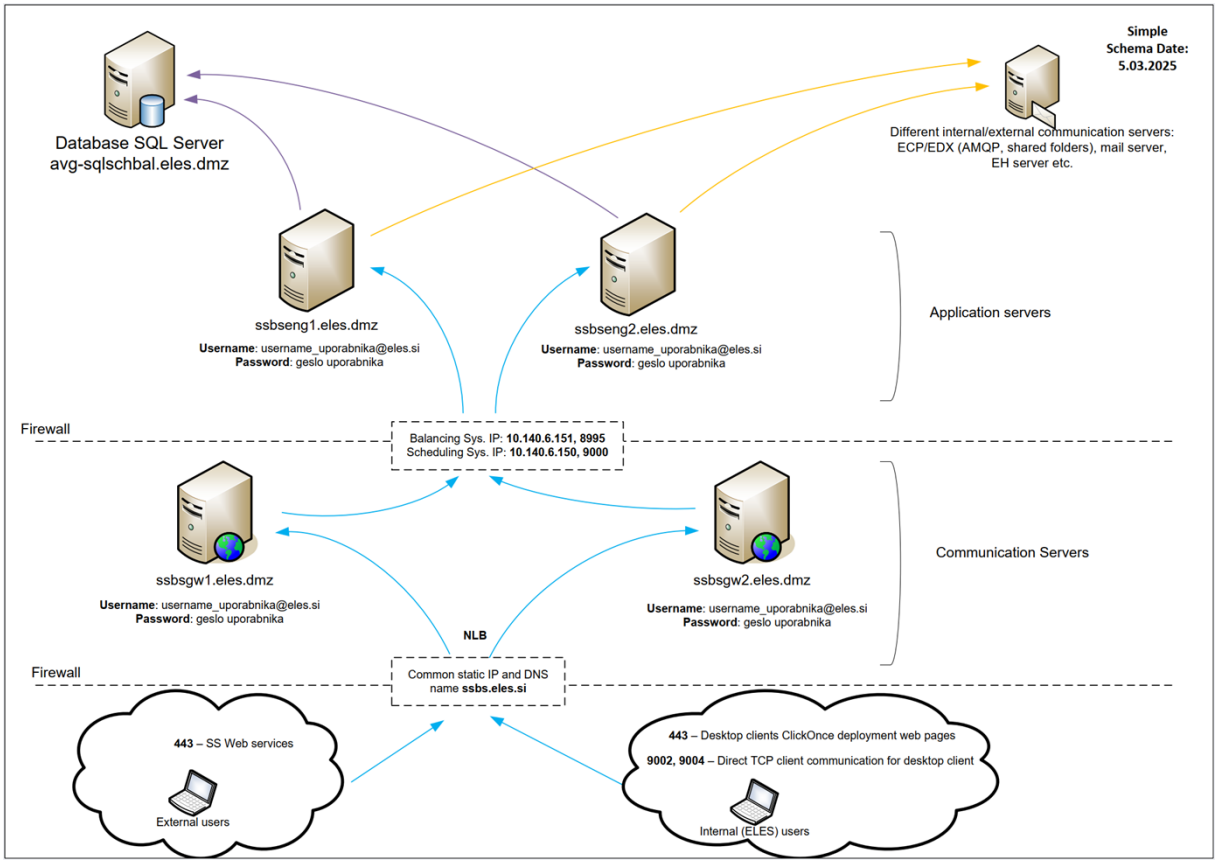
Ocenjena velikost inicialnih podatkov: 10 GB, Letni prirast: 2 GB

2.9. Komunikacijske zahteve

IS mora biti zasnova tako, da omogoča poljubno uporabniško konfiguracijo izmenjave podatkov z ostalimi deležniki. Posamezna izmenjava podatkov (pravilo izmenjave) bo temeljila na izbiri najmanj sledečih vsebin:

- primarne tehnologije/protokola izmenjave,
    - določitve poti do deležnika (v odvisnosti od izbrane tehnologije/protokola izmenjave),
    - določitve uporabnika za dostop do deležnika (v odvisnosti od izbrane tehnologije/protokola izmenjave),
    - določitve vsebine za izmenjavo (v odvisnosti od izbrane tehnologije/protokola izmenjave),
    - določitev pogojne ali brezpogojne izmenjave,
- sekundarne tehnologije/protokola izmenjave.
    - določitve poti do deležnika (v odvisnosti od izbrane tehnologije/protokola izmenjave),
    - določitve uporabnika za dostop do deležnika (v odvisnosti od izbrane tehnologije/protokola izmenjave),
    - določitve vsebine za izmenjavo (v odvisnosti od izbrane tehnologije/protokola izmenjave),
    - določitev pogojne ali brezpogojne izmenjave

Načrtovalna visokonivojska umeščenost IS v IS ELES je predstavljena v Slika 2.9-1.



Slika 2.9-1 Umestitev IS v Elesovo okolje

Podprte/zahtevane tehnologije/protokoli izmenjave so:

- AMQP (ECP/EDX),
- SFTP,
- E-Mail (SMTP, POP3, IMAP),
- FSSF (File System Shared Folder – SMB)
- FSSF (File System Shared Folder – MinIO/S3).

Splošne varnostno komunikacijske zahteve:

- Vse povezave, ki se uporabljajo za izmenjavo podatkov preko javnega omrežij, morajo biti šifrirane.
- Spletni vmesnik mora za svoje delovanje uporabljati izključno varne povezave (TLS 1.2 ali TLS 1.3).
- Za izmenjavo podatkov mora aplikacija uporabljati standardne API vmesnike in šifriranje prometa (najmanj TLS 1.2). Omogočena mora biti menjava API ključev.

2.10. Upravljanje s popravki/spremembami

Redne posodobitve in tehnične izboljšave morajo biti praviloma izvedene tako, da to ne pomeni nedelovanja informacijske rešitve znotraj opredeljenega delovnega časa naročnika. Če se nedelovanju ni mogoče izogniti, mora biti vsaka načrtovana prekinitev delovanja (zaradi

npr. nujnih popravkov programske ali strojne opreme, nadgradenj in drugih nujnih del) naročniku sporočena najmanj 3 delovne dni pred dejansko izvedbo. V sporočilu mora biti naveden razlog za prekinitev ter čas, v katerem bo izvedena prekinitev. Načrtovana prekinitev delovanja se izvede v skladu z dogovorom med izvajalcem in naročnikom.

Upravljanje sprememb:

- Za pravilno izvajanje procesa sprememb bo naročnik po podpisu pogodbe, oziroma pred prehodom v produkcijo zagotovil aktualno verzijo internega navodila »Upravljanje SPREMEB«.
- Naročnik mora za upravljanje s spremembami in popravki zagotoviti informacijski sistem, ki bo omogočal spremljanje aktivnosti na posamezni nalogi, spremembi, popravku in bo neposredno povezljiv in kompatibilen z sistemom Azure DevOps.
- Analiza vseh zahtev naročnika za spremembo delovanja sistema in po potrebi nastavitev pred-produkcijskega ali testenega sistema za simulacijo ter spremljanje delovanja.
- Raznovrstna pojasnila glede delovanja sistemov, analize razlogov za zavračanje sprejetih dokumentov, ročni poseg v bazo mimo aplikacije na zahtevo naročnika.
- Sodelovanje z zaposlenimi v IT službi naročnika v primeru suma ali ugotovitve, da za neustrezno delovanje sistema obstaja vzročna povezava z nekim drugim IT sistemom naročnika.
- Posodobitev in predložitev dokumentacije sistema ob nadgradnjah. To zajema opis funkcionalnosti, tehnična navodila, uporabniška navodila, opis opravljenih del (release notes), popis vplivov izvedenih sprememb na delovanje funkcionalnosti.

Za upravljanje najav in reševanja dogodkov, ter ostalih ITSM procesov pogodbenega razmerja, bo moral dobavitelj/implementator zagotoviti integracijo med dobaviteljevim in naročnikovim okoljem (upravljanje dogodkov, najave in zahteve sprememb, težave v delovanju, spremljanje delotokov,..), kar bo podrobneje določeno v aktualnih navodilih, jih naročnik zagotovi ob pripravi PZI.

2.11. Licenciranje in izvorna koda

- IS mora vsebovati programsko opremo, ki licenčno ni sporna.
- Proizvajalec mora zagotavljati vsaj 5 letno življenjsko dobo programske opreme.
- Proizvajalec ne sme biti s strani države Slovenije ali EU opredeljen kot varnostno nesprejemljiv.
- Dobavitelj / implementator mora za čas življenjske dobe strojne/programske opreme zagotavljati varnostne popravke rešitve in zagotavljati delovanje ob nameščanju rednih varnostnih popravkov OS. (uporabljajo se določila SLA).
- Dobavitelj / implementator programske rešitve IS mora zagotoviti, da se skozi projekt razvoja in implementacije in v času vzdrževanja sistema zagotovi dostop do neobstruirane in verzionirane izvorne kode produkta.

- Dobavitelj / implementator mora zagotoviti uvajanje, spremljanje in ostalo upravljanje z programske rešitve z uporabo naročnikovega sistema. Po potrebi mora biti zagotovljena integracija med dobaviteljevim in naročnikovim okoljem.

2.12. Produkcijsko, razvojno in testno okolje

Vse navedene zahteve v točkah 2.1 do 2.11 morajo biti upoštevana pri načrtovanju vseh treh okolij (Produkcijsko, Razvojno in Testno).

- Produkcijsko okolje IS je komunikacijsko in vsebinsko povezano v produkcijska okolja IS Eles.
- Testno okolje je komunikacijsko in vsebinsko povezano v testna okolja IS Eles. IS mora omogočati vse postopke za prenos produkcijskih podatkov v testno okolje (backup/restore produkcije na test,..), za katere morajo biti pripravljeni in dokumentirani primerni postopki.
- Razvojno okolje IS je komunikacijsko in vsebinsko povezano v testna okolja IS Eles. Način uporabe podatkov se določi ob pripravi PZI.

2.13. Zakonske zahteve in interna pravila

- IEC 62351-8: Role-based access control for power system managementRole - RBAC
- IEC 62325-504: Web Services
- ISO/IEC 27001: Information security, cybersecurity and privacy protection
- ISO/IEC 27031: IT disaster recovery
- ISO/IEC 33061: Software life cycle processes
- RFC 5424: The Syslog Protocol
- MADES: MArket Data Exchange Standard
- PRI 5.3.1 TEMELJNA INFORMACIJSKO KIBERNETSKA VARNOSTNA POLITIKA DRUŽBE ELES D.O.O.
- OP K 7.5.4 P4 Navodilo »UPRAVLJANJE SPREMEMB«
- PRI K 7.5.4 P1_1 Standardna uporabniška strojna oprema
- PRI K 7.5.4 P1_2 Standardna uporabniška programska oprema,
- PRIK 7.5.4 P1_3 Standardna strežniška programska oprema,
- PRIK 7.5.4.1 4 Sistemske nastavitve informacijskega sistema ELES.
- NA K 7.5.4.1 Varovanje in zaščita podatkov poslovno informacijskega sistema,
- NA K 7.5.4.2 Varnostno kopiranje podatkov v elektronski obliki,
- NA K 7.54.8 Metodologija razvoja programske opreme

2.14. Projekt za izvedbo (PZI) in Projekt izvedenih del (PID) in ostala spremljajoča dokumentacija

Vsa dokumentacija PZI, PID in navodila morajo biti v obliki, ki jo lahko naročnik nadalje ureja in spreminja.

Dokumentacija mora biti v slovenskem jeziku, medtem, ko so dovoljeni strokovni izrazi v angleškem jeziku.

Dokument Projekta za Izvedbo (PZI) je namenjen natančnemu popisu in specifikacijam bodočega informacijskega sistema/aplikacije/modula. Na podlagi analize in specifikacij zahtev se v PZI določi seznam potrebnih gradnikov, poslovnih procesov, spletnih servisov in ostalih integracij. Na podlagi teh specifikacij se določi potrebne tehnološke standarde in tehnološke specifikacije za izvedbo ter arhitekturo sistema. Na ta način želi naročnik optimizirati arhitekturo in implementacijo sistema ter ob tem v največji smiselni meri vzpostaviti standardizacijo tehnoloških elementov ter s tem znižati skupne stroške lastništva centralne informacijske infrastrukture (v največji možni meri izogniti situaciji »vendor lock in«).

Naročnik pričakuje izdelan in s strani naročnika potrjen PZI pred pričetkom vpeljave rešitve, ki mora vsebovati najmanj:

- shemo celotnega sistema,
- seznam vse programske opreme, nameščene v okviru vpeljave,
- osnoven opis arhitekture in delovanja sistema (“High level design”).
- funkcionalne specifikacije in funkcionalno dekompozicijo,
- seznam procesov z opisi,
- seznam gradnikov z opisi,
- specifikacije podatkovnih struktur,
- specifikacijo struktur XML oziroma XSD sheme,
- specifikacije spletnih storitev (API),
- specifikacijo aplikacije za prikaz podrobnosti delovanja vseh vključenih komponent,
- arhitekturo sistema za implementacijo (predlog uporabe vzorcev, topologija strežnikov, uporabljeni tehnološki standardi, tehnologija podatkovnih zbirk),
- načrt spletnih mest (žični okvir) s specifikacijami vsebin in funkcionalnostmi za uporabnike (npr. katere informacije vsebuje, kako naj zgleda uporabniški vmesnik, kakšno uporabniško okolje bo ponujeno uporabniku, kako bodo organizirane vsebine, kakšne oblike iskanj bodo omogočene, do katere mere je mogoče doreči politiko umeščanja novih vsebin v portal, da se še zagotavlja preglednost),
- dokumentacijo v zvezi z ravnanjem s podatki (npr. definicijo sklopov podatkov), način obravnave podatkov skozi celoten življenjski cikel podatka (format hranjena, način

izvajanja revizijske sledi, način ščitenja pred vpogledom, način umika na alternativen medij), način brisanja (ročno, batch obdelava, frekvenca),
- varnostne in zaščitne mehanizme,
- navedene in natančno popisane predvidene integracije z zunanjimi sistemi,
- terminski načrt.

Predvideno je, da izvajalec dokument PZI izdela po izvedeni analizi zahtev.

Dokument PZI potrdi naročnik.

Po zaključku del, mora izvajalec izdelati projekt izvedenih del (PID), skupaj z upravljavskimi, vzdrževalnimi in uporabniškimi navodili. Po izvedenem šolanju za upravljanje, administracijo in uporabo, naročnik potrdi PID in predana navodila.

Izvajalec po implementaciji vseh procesov izdelal Business blueprint ("use cases"), ki vsebuje podroben opis izvedbe posameznih procesov.

- Navodila za uporabo povezovanja z drugimi IS:

Navodila za izvedbo integracij novega sistema z ostalimi IS in vključitvijo v IS Eles morajo biti na voljo pred prehodom v produkcijo in morajo zagotavljati vse potrebne informacije za uspešno povezovanj in izmenjavo podatkov z vsemi deležniki,

- Navodila za uporabo – uporabniška

Navodila morajo biti na voljo pred pričetkom šolanja. Navodila morajo biti izdelana na osnovi, ki je vpeljana pri naročniku in ne generična. Predana morajo biti v obliki, ki jo lahko naročnik nadalje ureja (.docx). Opremljena morajo biti vsaj s sledečimi vsebinami:

- Administratorska navodila za glavne administrativne procese. Navodila morajo biti izdelana na osnovi, ki je vpeljana pri naročniku in ne generična od proizvajalca. Predana morajo biti v obliki, ki jo lahko naročnik nadalje ureja (.docx), sheme pa v formatu Visio (.vsdx) ali Draw.io (.drawio). Opremljena morajo biti z vsemi za uporabo potrebnimi informacijami, poleg tega pa tudi z naslednjimi relevantnimi vsebinami:

  - seznam vseh administratorskih uporabniških imen in gesel v ločenem dokumentu ob administratorskih navodilih,
  - postopki zamenjave vseh gesel v sistemu,
  - postopki dodajanja virov, računov,
  - postopki kreiranja, brisanja in vzdrževanja pravil,
  - postopek izdelave poročila z opisom načina dostopa do urejevalnika poročil in osnovnega ustvarjanja enostavnega poročila,
  - postopki za arhiviranje konfiguracije in za ponoven uvoz v sistem,
  - postopki za izvoz dnevnikov revizijskih sledi za namen forenzičnih analiz
  - seznam uporabljenih shem in pomen posameznih polj,

  - primeri uporabljanih xml datotek.

- Primeri delovanja različnih vpeljanih uporabniških scenarijev s stališča različnih vlog.
- Postopek za obnovitev sistema (Disaster Recovery), ki mora vsebovati naslednje:
  - postopek za obnovitev sistema v primeru izpada naprave oziroma vzpostavitev sistema v primeru katastrofe,
  - navodila za ukrepanje v primeru izrednih dogodkov,
  - postopek mora biti uspešno testiran in zapisniško potrjen pred podpisom primopredajnega zapisnika.

3. PRIČAKOVANE SPREMEMBE

Predmet javnega naročila je tehnološka prenova Balancing (BS) in Auction Platform (AP) sistema, lastnik katerega vključno z izvorno kodo je naročnik (ELES).

Naročnik v sklopu tega javnega naročila naroča naslednje nadgradnje:

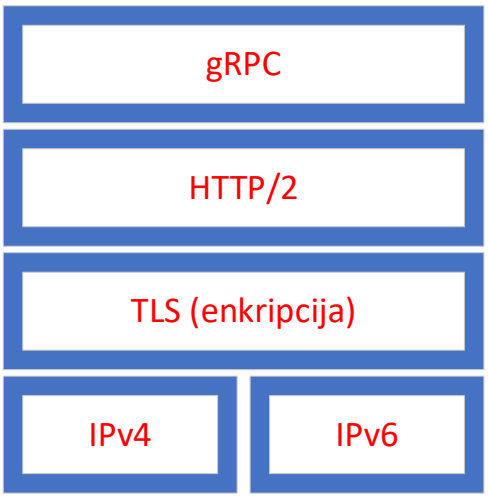
- **Tehnološko prenovo BS sistema**, in sicer:
    - Prehod iz .NET Framework 4.8 na .NET 8/.NET 9, migracijo iz WCF na gRPC in prehod iz uporabe SQL Stored Procedures na EF. Ti koraki so ključni za izboljšanje zmogljivosti, varnosti in dolgoročne vzdržnosti sistema. Prehod na .NET 8/.NET 9 prinaša več pomembnih prednosti, zlasti na področju varnosti:
        - Aktivna podpora in posodobitve: .NET 8/.NET 9 je najnovejša različica z dolgoročno podporo (LTS), kar zagotavlja redne varnostne posodobitve in izboljšave.
        - Boljši mehanizmi za zaščito: .NET 8/.NET 9 vključuje napredne varnostne funkcije, kot so izboljšano upravljanje certifikatov, podpora za varnejše šifriranje in integracija z modernimi varnostnimi standardi.
    - Migracijo na gRPC izboljšuje učinkovitost in varnost komunikacij:
        - Hitrejša komunikacija: gRPC uporablja protokol HTTP/2, ki omogoča večjo pretočnost podatkov, kompresijo in več sočasnih povezav, kar zmanjšuje zakasnitve.
        - Vgrajena varnost: gRPC privzeto podpira TLS, kar zagotavlja šifrirano komunikacijo med odjemalcem in strežnikom.
    - Prehod na EF, s čemer se omogoča izboljšano vzdrževanje in razvoj sistema:
        - Zmanjšana kompleksnost: EF poenostavi delo z bazo podatkov, kar omogoča lažje spreminjanje in širjenje podatkovnega modela.
        - Večja preglednost: Poslovna logika je neposredno v programski kodi, kar izboljšuje preglednost in sledenje spremembam v sistemu.
        - Standardizacija: EF omogoča uporabo sodobnih pristopov k delu z bazo podatkov, kot so LINQ poizvedbe, ki povečujejo berljivost in skrajšajo razvojni čas.
- **Ločitev in tehnološko prenovo dražbenega dela sistema (v nadaljevanju AP)**, in sicer:
    - Dražbeni del vključno z vsemi obstoječimi funkcionalnostmi se kompletno loči od preostalega BS sistema, komunikacija med njima pa predvideva samo predajo rezultatov avkcije iz AP sistema v BS sistem.
    - Tudi za AP sistem se izvede tehnološka podpora kot je to navedeno za tehnološko prenovo BS sistema;
- **Pilotni projekt za delovanje obeh sistemov v kontejnerskem načinu delovanja**

3.1. Tehnološka prenova BS sistema

Tehnološka prenova BS sistema je ključna zahteva naročnika. Zasnovana je tako, da v največji možni meri sledi standardom primernim za delo v sektorju energetike kot kritične infrastrukture. Kot že omenjeno uvodoma, in tudi v nadaljevanju, poleg ključnih varnostnih izboljšav in prednosti z nadgradnjo .NET, se je pri prenovi sistema v največji možni meri upošteval tudi varnostni vidik z vidika poslovnih procesov, ki jih sistem izvaja. V sladu s tem naj bo nov Balancing sistem brez vključenega modula za avkcije (ki je z zunanjimi deležniki bistveno bolj izpostavljen) ter se osredotoča na robustnost za ključne procese uravnavanja odstopanj trga preko lokalnih (ali mednarodnih) povezav. Več o tem v nadaljevanju.

3.1.1. Komunikacijski protokol HTTP/2

HTTP/2 je je izboljšava popularnega HTTP protokola, na katerem temelji ves svetovni splet. Standard je bil objavljen leta 2015 in ga danes podpirajo vsi brskalniki, vse bolj popularen je tudi za izmenjavo podatkov med aplikacijami (Googlova knjižnica gRPC). Med drugim omogoča bolj učinkovito uporabo TCP povezav, kompresijo headerjev in krajše odzivne čase. Deluje nad transportnima protokoloma IPv4 in IPv6.



To bo po našem mnenju odpravlja številne komunikacijske težave z Balancing sistemom prijavljene v obdobju zadnji 1-2 let.

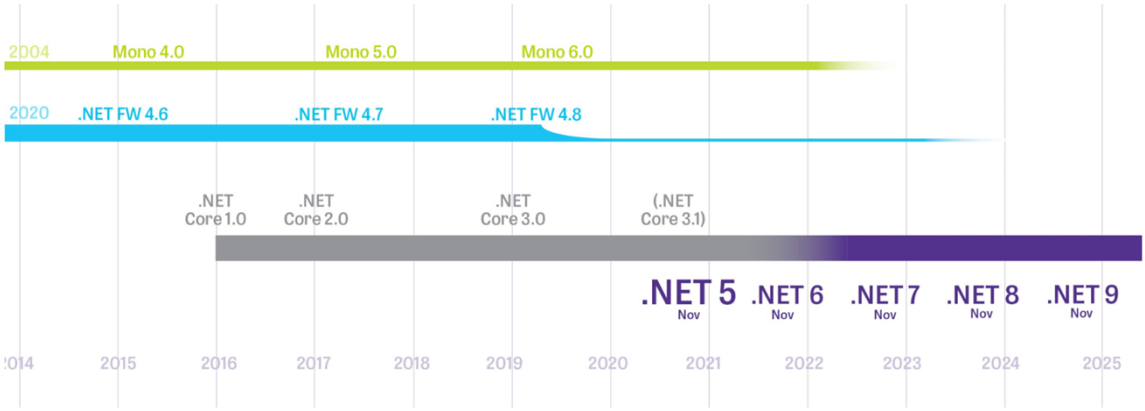
Vsi trije načini avtentikacije uporabnikov morajo ostajati še naprej podprti (certifikat, geslo, Windows domena).

3.1.2. Microsoft .NET 8/9

Microsoftovo razvojno okolje .NET in jezik C# sta ob svojem izidu leta 2002 pomenila prelomnico v razvoju Windows aplikacij. Razvoj je bil hitrejši in enostavnejši kot v starejših programskih jezikih, zato smo v tem okolju razvili tudi BS aplikacijo.

Do leta 2019 je Microsoft redno izdajal nove verzije .NET Framework, ki so bile kompatibilne s predhodnimi, vse pa so zahtevale Windows operacijski sistem. Odvisnost od Windowsov je

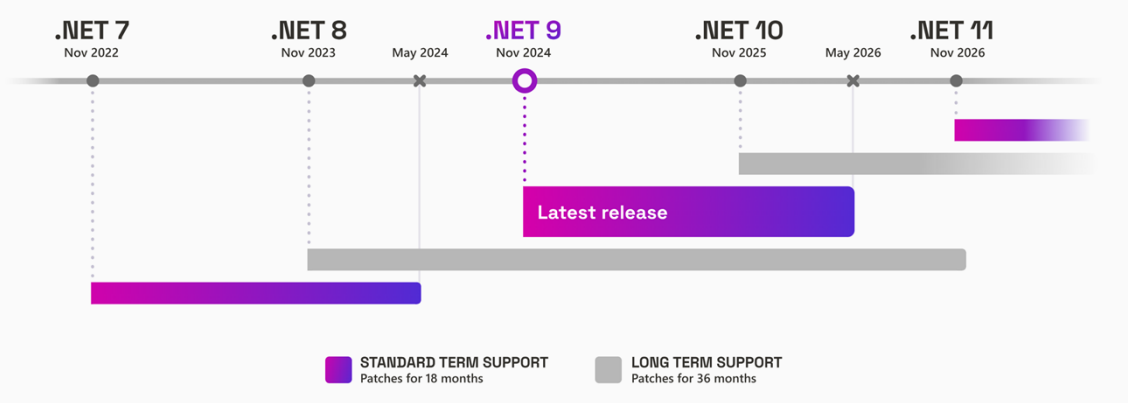
postajala vedno večja ovira, zato je Microsoft leta 2019 končal z razvojem klasičnega .NET Framework okolja in ves svoj razvoj usmeril v povsem novo .NET Core okolje, ki je voljo na Windows, Linux, macOS, iOS in Android. Zadnji klasični **.NET Framework 4.8** je izšel **aprila** 2019 in ga uporablja **trenutni Balancing sistem v ELES okoljih**.



Žal je .NET 8 samo delno kompatibilen z predhodnimi verzijami. Nekatere funkcionalnosti, ki bi jih težko prenesli na druge operacijske sisteme, so enostavno ukinili hkrati pa so ukinili standardno podporo za ta produkt in ter z novimi .NET (>7, 8) vpeljali:

- izboljšave v varnostnih funkcijah (boljša podpora za HTTPS: .NET 7 in 8 sta izboljšala privzeto uporabo šifriranja (npr. TLS 1.3), kar zmanjšuje možnosti napadov, kot so Man-in-the-Middle (MITM));
- napredne funkcije za preverjanje identitete: Vgrajene knjižnice za overjanje in avtorizacijo (kot je ASP.NET Core Identity) so bile posodobljene, da podpirajo najnovejše protokole, kot sta OpenID Connect in OAuth2;
- boljše ravnanje z občutljivimi podatki: Šifriranje, varno shranjevanje gesel in varnejše ravnanje s sejnimi žetoni (tokens);
- zmanjševanje ranljivosti – AOT (Ahead-of-Time) kompajliranje: .NET 7 in 8 omogočata izboljšano AOT, kar otežuje reverzno inženirstvo aplikacij;
- zmanjševanje ranljivosti – bolje zaščiteni runtime: Runtime (.NET Core runtime) vključuje zaščite pred pogostimi ranljivostmi, kot so buffer overflow napadi;
- redne varnostne popravke – t. i. Security Development Lifecycle (SDL), ki vključuje: redno pregledovanje kode zaradi varnostnih lukenj, hitra izdaja popravkov za znane ranljivosti in pa močnejša orodja za pregledovanje kode, kot je Static Analysis Security Testing (SAST).

Varnost, je ključen dejavnik novih Microsoft razvojnih okolij .NET. Časovne omejitve za podporo so ključne za varnost, saj razvijalce silijo k prehodu na novejše, bolj varne različice, s tem pa se zmanjša število **zastarelih sistemov, ki so pogosto tarča napadov**. Glavni cilj teh politik je torej izboljšanje zanesljivosti, varnosti in zmogljivosti aplikacij.



3.1.3. MSDTC

Microsoft je ukinil Microsoft Distributed Transaction Coordinator (MSDTC) v .NET 5 kot del prizadevanj za poenostavitev in racionalizacijo platforme. MSDTC je bil zgodovinsko uporabljen za obvladovanje porazdeljenih transakcij v različnih virih podatkov. Vendar pa porazdeljene transakcije pogosto prinašajo kompleksnost in režijske stroške.

Uporaba Microsoft Distributed Transaction Coordinator (MSDTC) lahko prinese več težav, še posebej v kontekstu občasnih motenj na mreži, kar je pogosto pri porazdeljenih sistemih. Porazdeljene transakcije preko MSDTC zahtevajo sinhronizacijo več podatkovnih virov, kar lahko postane zelo zapleteno v (občasno) nestabilnem omrežnem okolju zelo zapleteno in močno povečuje verjetnost za zaplete pri sledenju in odpravljanju napak.

Pri predhodnih verzijah sistemov smo na ELES-u že naleteli na podobne težave zaradi uporabe MSDTC v povezavi z mrežno infrastrukturo. Občasne motnje na mreži lahko povzročajo zaplete pri upravljanju porazdeljenih transakcij. Zaradi omrežnih napak so bile nekatere transakcije nezaključene, kar je povzročilo nedoslednost podatkov. Napake v omrežju so povzročile povečanje časovnega zamika pri transakcijah, kar je zmanjšalo učinkovitost in odzivnost aplikacij. Ko so se pojavile motnje, smo naleteli na težave pri sledenju in odpravljanju napak, kar je zahtevalo dodaten čas in vire za obnovitev normalnega delovanja sistemov.

Z uvedbo novega pristopa brez uporabe MSDTC za delo z dvema ločenima bazama, kjer ena hrani poslovno logiko in druga log zahtevkov, predvidevamo uvedbo nekoliko drugačne arhitekture in procesa za obvladovanje transakcij:

1. **Ločene transakcije:** Vsaka baza naj upravlja svoje transakcije ločeno, kar pomeni, da transakcije v posamezni bazi ne obsegajo več kot enega podatkovnega vira. To pomaga zmanjšati kompleksnost in odpraviti potrebo po koordinaciji transakcij med več viri.

2. **Poudarek na končni skladnosti:** Namesto poskusa zagotavljanja takojšnje skladnosti, naj bo sistem zasnovan za dosego končne skladnosti. Medtem ko se dejanja odvijajo ločeno v vsakem sistemu, naj se uvedejo procesni koraki, da na koncu dosežemo skladnost med obema bazama, tudi če pride do kratkotrajnih neusklajenosti.

3.2. LOČITEV IN TEHNOLOŠKA PRENOVA AP SISTEMA

Naročnik s tehnološko prenovo AP sistema zahteva tehnološko nadgradnjo aplikacije kot je to zahteva za BS sistem. Pri tem naročnik zahteva torej prehod iz .NET Framework 4.8 na .NET 8/.NET 9, migracijo iz WCF na gRPC in prehod iz uporabe SQL Stored Procedures na EF.

Poleg tehnološke prenove naročnik naroča še komplet ločitev BS in AP sistem. Ločitev zajema:

- Predelava uporabniškega vmesnika, da ta v AP sistem vsebuje samo procese, ki so vezani na izvedbo afrr/mfrr/fcr zakup kapacitet;
- Predelava sistema, da uporablja ločeno bazo podatkov;
- Predelava sistema, da ima ločen nabor šifrantov in uporabniških pravic;
- Predelava sistema, da podpre CCS FCR kooperacijski način dela
- Predelava sistema za podporo komunikacijskim kanalom ECP/EDX AMQP in EME Web Service

3.2.1. Dražbe splošno

Nov sistem kot celota mora podpirati celovit poslovni proces, po vsebinskih sklopih:

- Dražbe
    - Predloge dražb s parametri dražb.
    - Načrtovanje in upravljanje primerov dražb.
    - Sprejemanje in preverjanje ponudb.
    - Postopek dodelitve in algoritmi.
    - Objavljanje rezultatov.

- Registri:
    - Upravljanje udeležencev.
    - Shranjevanje kvalifikacij za sodelovanje v določenih vrstah dražb (za specifične kategorije izdelkov).
    - Upravljanje registracij za določene primere dražb.
    - Obvestila o sporočilih.
    - Upravljanje uporabnikov in prijav.
    - Drugi podporni registri.

- API (ECP/EDX AMQP, kot backup email, Web Service):
    - Oblike dokumentov za izmenjavo podatkov z udeleženci (ENTSO-E).
    - API vmesniki za povezovanje in prejemanje podatkov iz obstoječih sistemov.

  - Oblike dokumentov za izmenjavo podatkov na osnovi datotek z obstoječimi sistemi.

- Poročila:
  - Seznam vseh poročil kot v starem sistemu predelanih na osnovo produkta.
  - Podpora za predloge vseh poročil (Excel itd.).

- Uporabniški vmesnik:
  - Osnovna zasnova vseh oken in pogledov s seznami vključenih podatkov.
  - Dostop za branje podatkov za interne uporabnike podjetja – Pivot podpora.

Dražbe morajo biti v novi verziji opredeljene z naborom parametrov, vključno s smerjo transakcije, količino energije/moči in dolžino (trajanja) obdobja dobave. Sistem mora vključevati tudi začetne (privzete) vrednosti za dražbe.

Ko je dražba kreirana (z izbranim produktom in znanimi natančnimi datumi obdobja dobave), morajo biti vrednosti parametrov dražbe enake privzetim vrednostim produkta, vendar jih uporabnik lahko prilagoditi glede na potrebe pred odprtjem dražbe za javnost. Osnovne vrednosti ki se lahko spremenijo pred povezovanjem produkta z dražbo:

- Koda izdelka – uporablja se za interno sklicevanje in prikaz.
- Opis
- Vrsta kategorije izdelka:
  - Dolgoročni nakup
  - Dolgoročna prodaja
  - Kratkoročni nakup
  - Kratkoročna prodaja

- Vrsta pogodbe – trajanje dostave izdelka:
  - Ura – ena ura
  - Dnevna – en koledarski dan, 00:00–24:00
  - Tedenska – od ponedeljka do nedelje
  - Mesečna – en koledarski mesec
  - Letna – eno koledarsko leto
  - Po meri/urna – poljubno število zaporednih ur v istem dnevu dostave
  - Po meri/dnevna – poljubno število zaporednih dni.

- Resolucija – koliko neodvisnih cen in količin (Q, P) je možnih v obdobju dostave:
  - 15 minut
  - 30 minut
  - 60 minut

  - 1 dan
  - 1 teden (lahko se uporablja samo v kombinaciji z vrsto tedenske pogodbe)
  - 1 mesec (lahko se uporablja v kombinaciji bodisi z mesečno bodisi z letno vrsto pogodbe)
  - 1 leto (lahko se uporablja samo v kombinaciji z letno vrsto pogodbe)
- Podprte ločljivosti po vrstah pogodb

Res. / CT	urni (+ po meri)	dnevni (+ po meri)	tedenski	mesečni	četrtletni	letni
15 minut	**x**	x				
30 minut		x				
60 minut	x	x	x			
4 ure		x				
1 dan		x	x	x	x	x
1 teden			x			
1 mesec				x	x	x
1 leto						x

Produkt dražbe je lahko opredeljen kot preprost osnovni izdelek, kjer se moč zagotavlja enakomerno skozi celotno obdobje dostave. Lahko je tudi opredeljen s specifičnimi časovnimi intervali, kjer je kapaciteta na voljo le v določenih obdobjih dobave.

To omogoča definiranje »base, peak, off-peak, EU peak, EU off-peak« in drugih po meri oblikovanih produktov. Datum dobave izdelka je lahko kombinacija več časovnih intervalov, ki jih lahko definira uporabnik in so opredeljeni z naslednjimi lastnostmi:

- Čas od in do (00 – 24)
- Dnevne oznake (Ponedeljek, Torek, ..., Nedelja)
- Ne-delovni dnevi (kot so opredeljeni v registru specifičnih koledarskih datumov): lahko se uporabijo za določitev, ali je izdelek (ne)dostopen ob praznikih, poleg vikenda (Sobota, Nedelja).

Oblikovanje mora biti na voljo samo za izdelke z dnevno ali daljšo razločljivostjo.

Lastnosti produkta rabijo biti torej osnova (parametri) za dražbo. Ti naj vključujejo podrobnosti, kot so podatki o povečanju ponudb, trajanje dražbe in posebni pogoji ali določila. Kot predloga naj jih bo mogoče večkrat uporabiti za podobne storitve. Lastnosti produkta naj se obravnavajo kot začetne nastavitve ali vrednosti, ki se običajno nastavijo, pri ustvarjenju dražbe. Privzeti parametri niso »hardcoded« ampak se definirajo na podlagi predpostavk in preferenc operaterja dražbenega procesa. Kopija parametrov na dražbo mora biti zasnovana tako, da kasnejše spremembe lastnosti produkta ne vplivajo na obstoječe dražbe.

Produkt dražbe kot predloga dražbe, s parametri naj obsega vsaj:

- Kodo izdelka

- Kodo vrste izdelka
- Smer izdelka
- Vrsto pogodbe
- Vrsto resolucije
- Način dodelitve (algoritem ponudb):
    - Razvrstitev po ceni z proporcionalno dodelitvijo
    - Razvrstitev po ceni z načelom "prvi pride, prvi služi"
    - Načelo "prvi pride, prvi služi"
    - Proporcionalno

- Vrste oddajanja ponudb dražbe:
    - Bidding: Udeleženci tekmujejo med seboj za boljšo ceno v določenem časovnem obdobju. Platforma natančno razvršča ponudbe med dražbo.
    - Minimalni strošek: Udeleženci oddajo ponudbe v določenem časovnem obdobju. Platforma razvrsti ponudbe ob koncu dražbe.

- Korak cene (uporablja se kot natančnost cene in korak izboljšanja cene)

- Korak količine (uporablja se kot natančnost količine in korak izboljšanja količine)

- Minimalna količina ponudbe (moč)

- Maksimalna količina ponudbe (moč)

- Maksimalno število ponudb: Ta možnost naj bo nastavljena na 1 za dražbo vrste "bidding" in naj bo nedoločena (kar pomeni neomejeno) za druge vrste.

- Maksimalna in minimalna cena ponudbe: Udeleženci lahko oddajo ponudbe le s boljšimi ali enakimi cenami, sicer se ponudbe takoj zavrnejo.

- Maksimalna in minimalna zmagovalna cena (uporablja se, če se razlikuje od maksimalne cene ponudbe in ne sme biti razkrita ponudnikom): Udeleženci lahko oddajo ponudbe z slabšimi cenami, te ponudbe ne bodo zavrnjene, vendar ne bodo upoštevane pri izbiri zmagovalnih ponudb. Te dve parametri se lahko obravnavata kot rezervna cena dražbe (odvisno od smeri transakcije nakupa/prodaje).

- Sprejmi izključno deljive ponudbe (= dovoljena je delna dodelitev) in maksimalna količina, ki se lahko določi kot nedeljiva.

- Ali mora količina, ki se ponudi, biti enaka povpraševanju dražbe (= "polna ponudba"). Če ta možnost ni izbrana, so lahko sprejete ponudbe z delno količino (ob upoštevanju vseh drugih pogojev).

- Sprejmi samo izboljšane ponudbe (uporablja se, če je izbrana akcija ponudbe "Bidding"): Če je ta možnost omogočena, lahko udeleženec odda ponudbe, kjer je količina ali cena izboljšana.

- Trajanje dražbe – če se podatek vnese ročno, se konec trajanja dražbe izračuna tako, da se k času odpiranja dražbe doda trajanje. Konec prejemanja ponudb (dražbe) mora biti možno spremeniti – bodisi z zmanjšanjem bodisi z podaljšanjem trajanja.

- Objavljanje dražbe:
    - Relativni dnevi vnaprej, pred dnevom ponudbe + čas dneva
    - Predloga oznake dražbe
    - Predloga besedila dražbe

- Objavljanje rezultatov – Podprto rabi biti ročno objavljanje rezultatov ali samodejno, v kolikor je operater izbral to možnost.

- Informacije na uporabniškem vmesniku – sistem mora omogočat nastavitve, da so lahko nekatere občutljive informacije skrite ali prikazane:
    - Trenutna najboljša količina ponudbe
    - Trenutna najboljša cena ponudbe
    - Rezervna cena je dosežena. Glede na smer transakcije nakupa/prodaje to pomeni bodisi maksimalno bodisi minimalno zmagovalno ceno.
    - Seznam udeležencev
    - Udeleženci, razvrščeni po svojih ponudbah
    - Število udeležencev
    - Število ponudb na udeleženca
    - Skupno število ponudb
    - Za vsak zgoraj navedeno opcijo naj bo mogoče posamezno izbrati eno od nadaljnjih opcij:
        - Privzeto: Vidno interno, vendar šele po zaključku ponudbenega postopka.
        - Vedno vidno interno.
        - Vedno vidno interno in eksterno.

3.2.2. FCR dražbe

Definiranje vlog

Nadgrajena AP naj ima naslednje Party vloge:

- Auction Office
- Participant
- CCS (uporablja se v FCR procesu)

Meje in območja

Sistem potrebuje in podpira področja, ki se naprej klasificirajo v:

- Domain (= splošno)
- Bidding zone

Sistem za podprte funkcionalnosti ne potrebuje mej, niti UCTE »piramide«.

Brisanje območij

AP naj uporabniku preko GUI v tabelaričnem pogledu omogoča izbris izbranega območja, na katerega ni vezano nobeno drugo območje/dokument/vsebina.

Pred izbrisom naj AP uporabnika opozori o posledicah izbrisa v posebnem pop-up potrditvenem oknu.

Nastavitve območij

AP naj uporabniku omogoča nastavitev območja izravnave (dodajanje, urejanje, brisanje).

Dodajanje novih območij

AP naj uporabniku preko GUI omogoča dodajanje novih posameznih območij. Parametri za posamezno območje naj bodo razdeljeni v naslednje sklope:

Splošno:

- EIC koda: uporabnik naj bo omogočen vnos EIC kode območja (dovoljeni alfanumerični znaki); AP naj omogoča preverjanje podvajanja EIC kod in ostalih vnesenih obočij;
- Kratka šifra: ShortCode = 5 chars;
- prikazno ime: uporabniku naj bo omogočen vnos poljubnega prikazanega imena (dovoljeni alfanumerični znaki). DisplayName = 50 chars.

Pregled vseh območij

AP naj uporabniku preko GUI (v tabelarični obliki) omogoča pregled vseh dodanih območij glede na parametre:

- EIC koda;
- Short code;
- Izpisano ime (Display name).

Razvrščanje območij

AP naj uporabniku v GUI omogoča razvrščanje vseh zapisov padajoče/naraščajoče glede na prikazane parametre.

Anonimizacija dodanega PSI

AP naj omogoča anonimizacijo za potrebe sodelovanje v FCR kooperaciji. Uporabnik naj v polje vnese poljuben niz (alfanumerični znaki), ki zagotavlja anonimnost ponudnika. Pri posredovanju FCR ponudb lokalnih PSI na CCS se posreduje tudi podatek o PSI, vendar mora biti ta anonimen, »hash« koda ali podobno.

Visokonivojski opis procesa FCR dražb

Podpora zunanjim dražbam in sodelovanje na FCR kooperaciji

AP naj omogoča podporo zunanjim dražbam in sodelovanje na FCR kooperaciji. Visokonivojski opis procesa sodelovanja na FCR kooperaciji (standardni potek):

AS = avkcijski sistem (AP)

CCS = Central Clearing System (FCR platforma)

- definicija dražbe;
- kreiranje dražbe;
- pošiljanje parametrov dražb (pošiljanje PAD) (1);
- pošiljanje sprejetih ponudb izravnalne moči RVF (pošiljanje ARD) (1);
- izračun rezultatov dražbe v CCS;
- prejem nepotrjenih rezultatov CCS (PAD/ARD) (2,3);
- pošiljanje potrditve nepotrjenih rezultatov v CSS (iCNF);
- dokončna potrditev rezultatov (4) (fCNF);
- objava rezultatov;

Visokonivojski opis procesa sodelovanja na FCR kooperaciji (potek z motnjami):

- manjkajoče lokalne ponudbe za izravnalno moč RVF;
- sodelujoči OPS zamudi s pošiljanjem parametrov ali ponudb;
- CCS ne pošlje rezultatov;
- CCS me prejme vseh iCNF;
- razdružitev (Decoupling);
- popravljanje dokumentov.

*Opomba: Če pride v primeru motenj v procesu do zaključevanja avkcije v AP, naj ima uporabnik funkcionalnost preklica/zaključka dražbe.

Posredovanje ponudb za izravnalno moč na FCR kooperacijo

AP naj omogoča posredovanje zbranih (lokalnih) ponudb za izravnalno moč RVF na FCR kooperacijo ob GCT lokalne dražbe.

FCR nadomestni algoritem za izvajanje dražb RVF

AP naj omogoča izvajanje FCR nadomestnega algoritma za izbiro ponudb izravnalne moči RVF (lokalna dražba), v primeru nepovezanosti z FCR platformo (lokalno obratovanje).

AP naj sproži zagon FCR nadomestnega algoritma za izbor najugodnejših ponudb izravnalne moči RVF zgolj po koncu časa za oddajanje ponudb.

Upoštevanje rezultatov nadomestnega FCR algoritma sproži uporabnik (s klikom na Publish local results). Rezultate nadomestnega FCR algoritma za lokalni zaključek dražbe pa naj AP upošteva takrat, ko iz različnih razlogov (moteno delovanje, težave z komunikacijo, zahteva za razdruženo delovanje s strani OPS v času trajanja dražbe) CCS ne pošlje rezultatov. Vse omenjeno torej izvede izključno operator po lastni presoji s klikom na Publish local results.

Tehnično gledano "Publish local" najprej zaključi lokalno avkcijo (enako kot sicer vse, torej "native" rezultati), nato rezultate lokalne avkcije prepiše v RARD v obliki, v katero se sicer prepisujejo CCS rezultati.

Pravila izbire najugodnejših ponudb izravnalne moči RVF

FCR nadomestni algoritem naj izvaja izbiro najustreznejših ponudb izravnalne moči RVF skladno s pravili, časovnimi omejitvami in primeri, kot je to veljalo v BS sistemu.

Spremljanje izvajanja dražb za izravnalno moč FCR kooperacija

AP naj uporabniku preko GUI omogoča spremljanje poteka izvajanja dnevnih dražb za RVF, ki se izvajajo na FCR kooperaciji.

Prikaz izbrane dražbe in statusa v izbranem časovnem oknu

AP naj uporabniku preko GUI omogoča izbiro in prikaz izbrane dražbe ter statusov izmenjanih dokumentov za izbrano dražbo v časovnem oknu (datum mm/dd/llll).

Zahtevani statusi:

- Not available
- Auction created
- Auction announced
- Auction open for bidding
- Auction closed for bidding
- Auction results published
- Auction canceled
- Local documents submitted

- Local documents partially submitted
- Results received
- Results partially received
- Positive intermediate confirmation submitted
- Negative intermediate confirmation submitted
- Positive final confirmation received
- Negative final confirmation received
- Anomaly report received
- Published local results
- Published received results

Pregled statusov dražb na FCR kooperaciji

AP naj uporabniku v GUI omogoča pregled statusov dražb na FCR kooperaciji glede na naslednje statuse procesa in izmenjanih dokumentov:

1. Posredovane ponudbe na FCR kooperacijo (Offers and Parameters to CCS):
   - ponudbe (Offers): oddane ponudbe na FCR platformo (ARD);
   - parametri (Parameters): parametri dražbe (zahtevana količina za ELES, itd.) (PAD);
2. Rezultati FCR kooperacije (CCS Results):
   - rezultati (Results): prejeti rezultati dražbe (po partnerjih: ponujene ponudbe in dobljene ponudbe na dražbi);
   - parametri (Parameters): parametri dražbe po 4-urnih produktih za vse partnerje;
3. Potrditev prejema nepotrjenih rezultatov na FCR kooperacijo (Intermidiate Confirmation to CSS):
   - Potrditev nepotrjenih rezultatov (iCNF): prikažejo naj se informacije iz iCNF dokumenta.
4. Končna potrditev rezultatov (Final Confirmation from CCS):
5. Končna potrditev rezultatov (fCNF): prikažejo naj se informacije iz fCNF dokumenta.
6. Težave (Anomaly Report):
   - dokument v primeru težav (ANO): prikažejo naj se informacije iz ANO dokumenta.

Za zgornje statuse procesa (1. - 5.) naj se v osnovnem oknu Primary: CCS RL prikažejo naslednji parametri:

- Naslov (Title): izpis korakov v procesu;
- Obdelano (Processed): zapis datuma dd/mm/llll in ure hh:mm:ss obdelave oddanega/prejetega dokumenta;
- Status (Status): status dokumenta (iz ResourceScheduleConfirmationReport dokument);
- Potrditev:

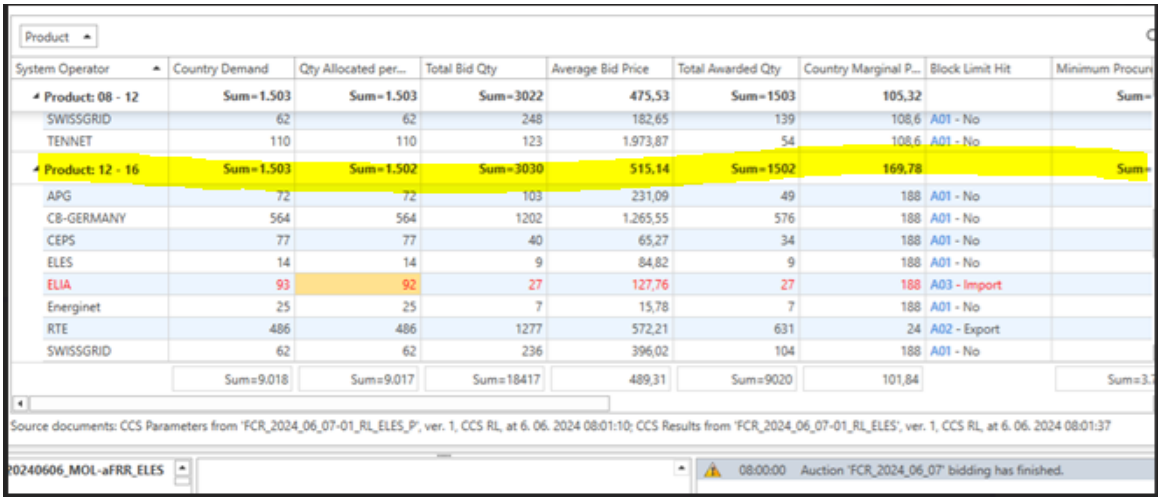
  [Status] Prikaže status za ACK (Accepted/Rejected)

[Processed]: datum in dd/mm/llll in ure hh:mm:ss obdelave; potrditev (ACK);

- Ročna akcija (Action): uporabnik naj ima možnost ponovnega ročnega pošiljanja dokumenta (gumb Resend).

Za vsak od 6 4h blok produktov naj bo prikazana tudi "krovna" vrstica (rumeno označena na spodnji sliki), v kateri so "statistični" povzetki za posamezen blok produkt na nivoju Kooperacije (torej neodvisno od nastavitev v filtru "System Operator" za koliko sistemskih operaterjev naj bodo vidni statistični podatki za posamezen blok produkt) – odvisno od tipa prikazanega podatka (pri količinah v MW običajno vsote, pri cenah pa povprečja).

Prikaz tudi v teh vrsticah obarva parametre, ki odstopajo (po definicijah, ki veljajo za obarvanje po posamezni državi). V spodnjem primeru (glej ELIA na sliki) naj se na enak način obarva (rdeče) ta vrstica (pri barvanju naj se uporabijo enaki kriteriji in pravila kot za vrstico pri ELIA).



Zaželen je privzeti pogled:

- za vsak blok produkt "krovna vrstica, ki prikazuje sumirane/povprečne podatke za celotno FCR kooperacijo in dodatno že prikazani podatki za ELES".
- optimalno bi bilo, da lahko filter za prednastavitve ureja uporabnik (AP si nastavitev zapomni tudi po restartu).
- po potrebi pa se razpre v obliko pogleda, ki ga poznamo v današnjem BS (primer je kar obstoječi pogled, prikazan na zgornji sliki).

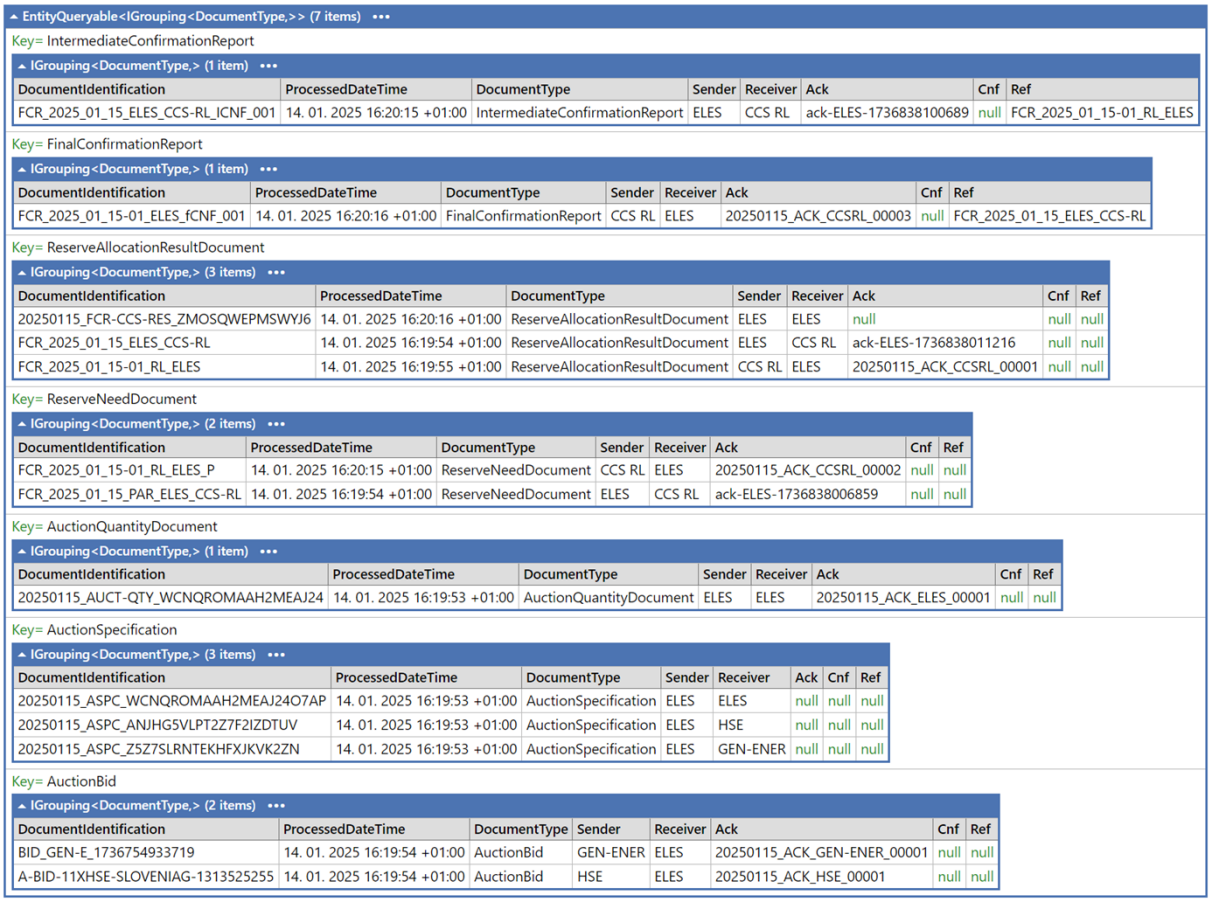
Vsaka dražba naj vsebuje povezave do dokumentov povezanih z njo (kot je npr. pri FCR-ju).

To bi zajemalo:

- vse poslane dražbene specifikacije,
- sprejete ponudbe (tehnično ustrezni bid dokumenti, sprejeti na avkcijo, neglede na rezultat),

- pri vsakem od prej naštetih dokumentov se doda tudi povezava na vezan ACK (če ta obstaja).
- rezultati dražb.

Lahko bi se recimo dodalo zavihek pod vsako dražbo z dokumenti, lahko se naredi povezave znotraj obstoječih pogledov. Rešitev je lahko predlagana tudi s strani izvajalca. Trenutno iskanje je namreč zelo zamudno in v kritičnih primerih neustrezno.



EntityQueryable<IGrouping<DocumentType,>> (7 items)

Key= IntermediateConfirmationReport

IGrouping<DocumentType,> (1 item)

DocumentIdentification	ProcessedDateTime	DocumentType	Sender	Receiver	Ack	Cnf	Ref
FCR_2025_01_15_ELES_CCS-RL_ICNF_001	14. 01. 2025 16:20:15 +01:00	IntermediateConfirmationReport	ELES	CCS RL	ack-ELES-1736838100689	null	FCR_2025_01_15-01_RL_ELES

Key= FinalConfirmationReport

IGrouping<DocumentType,> (1 item)

DocumentIdentification	ProcessedDateTime	DocumentType	Sender	Receiver	Ack	Cnf	Ref
FCR_2025_01_15-01_ELES_fCNF_001	14. 01. 2025 16:20:16 +01:00	FinalConfirmationReport	CCS RL	ELES	20250115_ACK_CCSRL_00003	null	FCR_2025_01_15_ELES_CCS-RL

Key= ReserveAllocationResultDocument

IGrouping<DocumentType,> (3 items)

DocumentIdentification	ProcessedDateTime	DocumentType	Sender	Receiver	Ack	Cnf	Ref
20250115_FCR-CCS-RES_ZMOSQWEPMSWYJ6	14. 01. 2025 16:20:16 +01:00	ReserveAllocationResultDocument	ELES	ELES	null	null	null
FCR_2025_01_15_ELES_CCS-RL	14. 01. 2025 16:19:54 +01:00	ReserveAllocationResultDocument	ELES	CCS RL	ack-ELES-1736838011216	null	null
FCR_2025_01_15-01_RL_ELES	14. 01. 2025 16:19:55 +01:00	ReserveAllocationResultDocument	CCS RL	ELES	20250115_ACK_CCSRL_00001	null	null

Key= ReserveNeedDocument

IGrouping<DocumentType,> (2 items)

DocumentIdentification	ProcessedDateTime	DocumentType	Sender	Receiver	Ack	Cnf	Ref
FCR_2025_01_15-01_RL_ELES_P	14. 01. 2025 16:20:15 +01:00	ReserveNeedDocument	CCS RL	ELES	20250115_ACK_CCSRL_00002	null	null
FCR_2025_01_15_PAR_ELES_CCS-RL	14. 01. 2025 16:19:54 +01:00	ReserveNeedDocument	ELES	CCS RL	ack-ELES-1736838006859	null	null

Key= AuctionQuantityDocument

IGrouping<DocumentType,> (1 item)

DocumentIdentification	ProcessedDateTime	DocumentType	Sender	Receiver	Ack	Cnf	Ref
20250115_AUCT-QTY_WCNQROMAAH2MEAJ24	14. 01. 2025 16:19:53 +01:00	AuctionQuantityDocument	ELES	ELES	20250115_ACK_ELES_00001	null	null

Key= AuctionSpecification

IGrouping<DocumentType,> (3 items)

DocumentIdentification	ProcessedDateTime	DocumentType	Sender	Receiver	Ack	Cnf	Ref
20250115_ASPC_WCNQROMAAH2MEAJ24O7AP	14. 01. 2025 16:19:53 +01:00	AuctionSpecification	ELES	ELES	null	null	null
20250115_ASPC_ANJHG5VLPT2Z7F2IZDTUV	14. 01. 2025 16:19:53 +01:00	AuctionSpecification	ELES	HSE	null	null	null
20250115_ASPC_Z5Z7SLRNTEKHFXJKVK2ZN	14. 01. 2025 16:19:53 +01:00	AuctionSpecification	ELES	GEN-ENER	null	null	null

Key= AuctionBid

IGrouping<DocumentType,> (2 items)

DocumentIdentification	ProcessedDateTime	DocumentType	Sender	Receiver	Ack	Cnf	Ref
BID_GEN-E_1736754933719	14. 01. 2025 16:19:54 +01:00	AuctionBid	GEN-ENER	ELES	20250115_ACK_GEN-ENER_00001	null	null
A-BID-11XHSE-SLOVENIAG-1313525255	14. 01. 2025 16:19:54 +01:00	AuctionBid	HSE	ELES	20250115_ACK_HSE_00001	null	null

Za FCR avkcije, kjer je sprememba statusa vezana na prejet ali poslan dokument, se lahko doda pregled (log) kateri dokument (in kdaj) je povzročil spremembo statusa avkcije.

Primer:

EntityQueryable<> (13 items)

Action	Status	ActionTime	UserName	DocumentIdentification
Created	Created	14. 01. 2025 17:15:51 +01:00	Admin	null
Create auction quantities	Created	14. 01. 2025 17:15:52 +01:00	Admin	null
Update auction quantities	Created	14. 01. 2025 17:15:52 +01:00	null	null
Open bidding	OpenForBidding	14. 01. 2025 17:15:53 +01:00	Admin	null
Publish announcement	OpenForBidding	14. 01. 2025 17:15:53 +01:00	Admin	null
Close bidding	ClosedForBidding	14. 01. 2025 17:15:54 +01:00	Admin	null
Submit local parameters	LocalDocumentsPartiallySubmitted	14. 01. 2025 17:15:54 +01:00	Admin	FCR_2025_01_15_PAR_ELES_CCS-RL
Submit local bids	LocalDocumentsSubmitted	14. 01. 2025 17:15:55 +01:00	Admin	FCR_2025_01_15_ELES_CCS-RL
Received results	ResultsPartiallyReceived	14. 01. 2025 17:15:56 +01:00	null	FCR_2025_01_15-01_RL_ELES
Received parameters	ResultsReceived	14. 01. 2025 17:16:15 +01:00	null	FCR_2025_01_15-01_RL_ELES_P
Submit intermediate confirmation report	PositiveIntermediateConfirmationSubmitted	14. 01. 2025 17:16:16 +01:00	null	FCR_2025_01_15_ELES_CCS-RL_ICNF_001
Received final confirmation report	PositiveFinalConfirmationReceived	14. 01. 2025 17:16:17 +01:00	null	FCR_2025_01_15-01_ELES_fCNF_001
Publish received results	PublishedReceivedResults	14. 01. 2025 17:16:17 +01:00	null	20250115_FCR-CCS-RES_ZMOSQWEPMSWYJ6

Podroben vpogled v dokumente dražb

AP naj uporabniku preko GUI omogoča podroben vpogled v dokumente.

Uporabnik naj ima možnost ogleda vsebine parametrov po naslednjih sklopih:

- vsebina (Content);
- informacije iz dokumenta (Document information).

Uporabnik naj ima možnost odprtja posameznega dokumenta (v največ 3 klikih) kjer si lahko pogleda podrobnosti.

Pregled oddanih ponudb PSI (zavihek Local Bids)

AP naj uporabniku preko GUI omogoča pregled vseh oddanih ponudb (dokumentov) PSI, ki sodelujejo na dražbi glede na naslednje parametre:

- pošiljatelj (Sender): ime PSI in njegova pripadajoča EIC koda;
- identifikacija dokumenta (Document Identification): identifikacijsko ime prejetega dokumenta;
- obdelava (Processed): časovna značka, ko je bil dokument obdelan (mm/dd/llll);

Uporabnik naj ima možnost podrobnega ogleda vsebine dokumenta.

Rezultati dražb FCR kooperacije

AP naj uporabniku preko GUI omogoča dinamičen tabelarični pregled rezultatov dražb za OPS doseženih:

- na FCR kooperaciji ali/in
- lokalno.

Na obstoječem prikazu Auctions/Instances se vključi 2 dodatna stolpca:

1) **Auction State Status** – prikazuje status posameznega stanja dražbe (Auction State) glede na spodnje pogoje (Auction State Status prikazuje v enem stolpcu le obarvana polja, brez vsebine):

Auction State		Auction State Status	
aFRR, mFRR	**FCR**	**Zelena**	**Rdeča**
Created		(čas T < ASP Time) && Created=1	(čas T < ASP Time) && Created=0
Announced		(ASP Time < čas T < GOT) && Announced=1	(ASP Time < čas T < GOT) && Announced=0

<table>
<tr><td colspan="2">BiddingStarted</td><td>(GOT < čas T < GCT) && BiddingStarted=1</td><td>(GOT < čas T < GCT) && BiddingStarted=0</td></tr>
<tr><td rowspan="2">BiddingClosed</td><td rowspan="2">BidsSentToCcsWaitingForResults</td><td rowspan="2">(GCT < čas T < Published Time) && BiddingClosed=1</td><td rowspan="2">(GCT < čas T < Published Time) && BiddingClosed=0</td></tr>
<tr></tr>
<tr><td rowspan="3">Published</td><td>PositiveFinalConfirmationReceivedCcsResultsPublished</td><td rowspan="2">(Published Time < čas T) && Published=1</td><td>(Published Time < čas T) && Published=0</td></tr>
<tr><td>LocalResultsPublished</td><td>(Published Time < čas T) && Published=0</td></tr>
<tr><td colspan="2">Canceled</td><td colspan="2">Canceled=1</td></tr>
</table>

Pri čemer oznake pomenijo: Created (1 – dražba kreirana, 0 – dražba ni kreirana), Announced (1 – ASP poslan, 0 – ASP ni poslan), BiddingStarted (1 – bidding možen, 0 – bidding ni možen), BiddingClosed (1 – bidding ni možen, 0 – bidding še možen), Published (1 – rezultati objavljeni, 0 – rezultati niso objavljeni), Canceled (1 – dražba je preklicana);

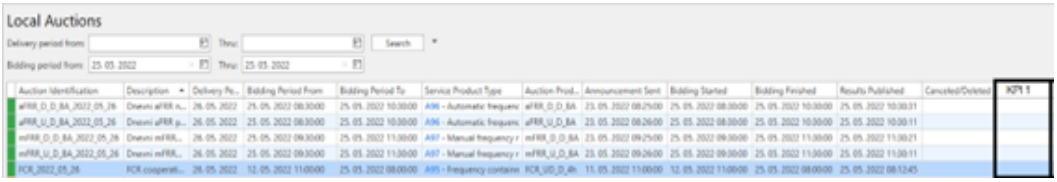
KPI – prikazuje razmerje Offered capacity / Asked Volume. KPI za posamezen produkt prikazuje stanje trenutno oddanih količin (t.j. količino vseh trenutno oddanih ponudb, katerih ponujena cena je ≤ zamejeni ceni) glede na povpraševane količine. Vrednosti in stanje kazalnika naj se ob prejetju vsake ponudbe sproti posodabljajo vse do zaključka dražbe (GCT), AP pa pri tem ločuje med naslednjimi stanji KPI:

- KPI zelen (ponujena količina ≥ povpraševane količine)
- KPI oranžen (ponujena količina < povpraševane količine)
- KPI rdeč (ponujena količina = 0)

Spodnja tabela prikazuje idealen primer spreminjanja vrednosti in stanja kazalnika tekom izvajanja aFRR/mFRR dražbe (barva je vezana na KPI – tudi tekom dražbe je lahko zelena ali ob zaključku rdeča):

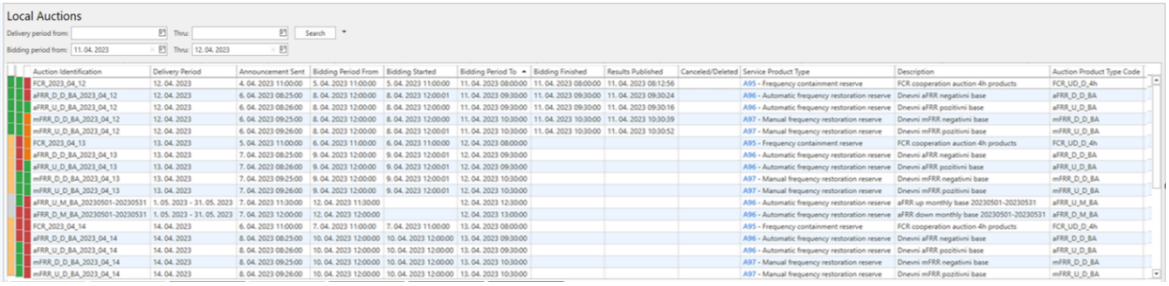
Čas	KPI (Offered capacity / Asked Volume)
Začetek dražbe	0 / 24
Tekom dražbe	18 / 24
Konec dražbe	24 / 24

Potek izvajanja posameznih dražb aRPF/rRPF (za FCR tega kazalnika ni, razen če bi lahko gledali končno dodeljeno količino s strani FCR kooperacije – to bi lahko bilo šele po zaključku dražbe) lahko izvajalec dražb sproti spremlja na zavihku Auctions Instances (glej sliko spodaj), kjer se v ločenem stolpcu prikazujejo vrednosti in stanje kazalnika. Stolpec KPI bi prikazali kot 3 z leve pred Auction Identification.



Na ta kazalnik naj bo vezano tudi obvestilo za izvajalce dražb v primerih, ko npr. 15 min pred koncem dražbe ne bi bilo zadostne ponujene količine za pokritje povpraševanja in bi izvajalec dražb lahko pozval ponudnike k posodobitvi ponudb.

Vrstni red stolpcev na pregledu Auction Instances naj bo sledeč:



AP naj uporabniku preko GUI omogoča dinamičen tabelarični pregled rezultatov dražb za OPS doseženih na FCR kooperaciji oziroma v sklopu lokalnega zaključka glede na:

- posamezni produkt (Product):
    - čas trajanja produkta (Time): 0:00–4:00;4:00–8:00...20:00–24:00;
    - zahtevana količina [MW] (Demand Qty): količina, ki jo na dražbi zahteva OPS;
    - udeleženci (Participants): število različnih udeležencev (PSI) na dražbi;
    - število oddanih ponudb (Bids Count): število oddanih ponudb na PSI;
    - celotna količina ponudb (Bids Total Qty): vsota količin ponudb za čas trajanja produkta.

- V spodnje oknu pa se glede na izbran posamezni produkt pokaže
    - PSI (Resource Provider): ime PSI, ki sodeluje na dražbi;
    - ponujeno (Tendered) [Qty] [Price] [Divisible];
    - dobljeno (Awarded) [Qty] [Price];
    - časovna značka (Time Stamp); mm/dd/llll;
    - identifikacija ponudbe (Bid Identification).

AP naj uporabniku preko GUI omogoča tabelarični pregled rezultatov dražb doseženih na FCR kooperaciji za vse sodelujoče OPS v FCR kooperaciji, glede na posamezen 4-urni produkt znotraj dneva po parametrih:

- OPS (System Operator);
- zahtevana količina za območje (Country Demand);

- minimalni zakup v bloku (Minimum Procured in Block);
- maksimalni izvoz v bloku (Maximum Exported in Block;
- marginalna cena za območje (Country Marginal Price);
- skupna količina (Total Bid Qty);
- povprečna cena ponudbe (Average Bid Price);
- skupno dobljena količina (Total Awarded Qty);
- marginalna cena FCR kooperacije (Cooperation Marginal Price);
- prekoračitev meje bloka (Block Limit Hit);
- določitev količine na blok (Qty Allocated per Block);
- določitev količine za območje (Qty Awarded in Country);
- delež območja (County Core Share);
- zahteva bloka (Block Demand Qty);
- dodeljena količina glede na območje (Qty Allocated per Country);
- pokrita zahteva (Covered Demand);
- pokritje deleža območja (Core Share Covered);
- pokritje limite izvoza (Covered Export Limit);
- prekoračitev meje območja (Country Limit Hit);
- notranja omejitev uvoza v bloku (Block Internal Import Limit);
- notranja omejitev izvoza v bloku (Block Internal Export Limit);

Za vsak od 6 4h blok produktov je prikazana tudi “krovna” vrstica, v kateri so “statistični” povzetki za posamezen blok produkt – odvisno od tipa prikazanega podatka (pri količinah v MW običajno vsote, pri cenah pa povprečja).

AP naj omogoča pregled objav rezultatov (dokumentov) dnevnih dražb RVF (Published Results), ki jih AP posreduje po uspešnem prejetju fCNF s strani FCR platforme sodelujočim PSI na dražbi glede na naslednje parametre:

- naslovnik (Receiver): ime naslovnika in njegova pripadajoča EIC koda;
- identifikacija dokumenta (Document Identification): identifikacijsko ime prejetega dokumenta;
- obdelava (Processed): časovna značka, ko je bil dokument obdelan (mm/dd/llll);

V primeru lokalnega zaključka dražbe (ob npr. decouplingu SI od Kooperacije) so tu vidni ustrezni dokumenti z rezultati dražb zaključenih po lokalnem/nadomestnem algoritmu. Uporabnik naj ima možnost podrobnega ogleda vsebine dokumenta po vzoru#1951

Produkt

V AP sistemu so FCR produkti definirani kot dnevni produkti, pri čemer obstaja 6 takih produktov, vsak s svojim 4-urnim profilom. Da bi delali na enak način kot platforma, je treba vnesti dnevni produkt in mu definirati 4 urni pas v katerem je aktiven. S tem še bolje sledimo

platformi, kjer je vsak 4 urni pas ločen produkt. V BS nimamo produktov, zato tam to ni bilo možno.

Simetričnost v novem sistemu ne manjka, saj nima smisla. V starem se je simetric uporabljal, ker ni bilo produktov, da si izbral up&down, potem je pa razbilo serije v ločene Up+Down, sicer je pa ostala ena serija Up&Down. Zdaj vneseš ali en Up&Down produkt, ali dva produkta, ločeno Up, Down in ločen Demand, bidi itd.

3.2.3. Dražbe

Sprememba pravil v procesu dražb

AP naj uporabniku preko GUI omogoča spremljanje izvajanja posameznih dražb glede na statuse, čas izvajanja, itd.

- Do začetka izvajanja dražb (Bidding Open) se lahko spreminja vse: Produkt, Dražba, dodatni parametri.
- Po začetku izvajanja dražbe (Bidding Open) in do konca dražb (Bidding Close) se lahko spreminja le še:
  - Podaljšanje trajanja dražb (Scheduled to)
  - Suspendiranje strank za dostop do dražb
  - Preklic dražbe
  - Možnost spremembe zamejene cene (maximum winning price)
  - Obstaja možnost ročnega proženja pošiljanja ASP. Poleg tega se ASP ob spremembi dovoljenih elementov akcije (npr. Reschedule) pošlje avtomatsko.
- Po koncu dražbe (Bidding Close) pa se lahko ročno spreminja le še:
  - Suspendiranje strank za dostop do akvcije
  - Preklic dražbe
  - Možnost spremembe zamejene cene (maximum winning price)
  - Možnost objave rezultatov
- Po objavi rezultatov dražbe (Publish) se lahko ročno prekliče dražbo.

Set dražb

V BS imajo možnost ustvarjanja seta dražb glede na ustrezne parametre.

Načini plačila

ELES ima tri načine plačila:

- pay as bid (že deluje);
- pay as cleared;
- pay as first rejected;

Preprečevanje uhajanja informacij pri izvajanju dražb

AP naj omogoča zakrivanje občutljivih informacij (skladno z REMIT) med izvajanjem dražb za izravnalno moč RVF, aRPF in rRPF, skladno s pravili.

AP naj podpira naslednja pravila s področja preprečevanja uhajanja informacij, ki so skladna z REMIT Uredbo:

1. Med trajanjem dražbe uporabnik nima pregleda nad cenami prejetih ponudb. Uporabnik naj vidi količine, ki niso urejene po ceni. Cen se ne vidi tudi v izmenjavi dokumentov, torej preko ECP/EDX ipd.

2. Ko je dražbe konec (bidding gate closed oz. konec prejemanja ponudb PSI), se naj rezultati prikažejo uporabnikom. Uporabnik naj preveri preliminarne rezultate in če ni nepravilnosti, jih uporabnik objavi, sicer lahko dražbo prekliče.

3. Po objavi rezultatov naj tudi PSI vidijo rezultate svojih ponudb (dobijo dokument z rezultati).

Tehnična sposobnost in kvalificirane količine PSI

PSI imajo poleg Kvalificirane količine za dražbe tudi Tehnično sposobnost nudenja storitve. V ta namen se bi si po vzoru Kvalificiranih količin želeli dodatno možnost vnosov podatkov o Tehnični sposobnosti – ti podatki bi potem služili kot osnova za prikaz v novem poročilu (pod formatted reports), ki je narejen po vzoru (tudi novega) poročila iz točke Pregled kvalificiranih količin (primeri reportov v nadaljevanju).

AP naj uporabniku preko GUI omogoča **vnos nove tehnične sposobnosti** za zagotavljanje izravnalne moči za posameznega PSI z naslednjimi parametri:

- vrsta ponudbe (Offer Type): uporabnik naj izbere ustrezno vrsto ponudbe;
- vrsta storitve (Business Type): uporabnik naj izbere vrsto storitve za katero želi uveljaviti omejitev (RVF, aRPF, rRPF);
- smer izravnave (Direction);
- maksimalna tehnična sposobnost [MW];
- veljavnost omejitve od (Valid From) [dd.mm.llll];
- veljavnost omejitve do (Valid Thru) [dd.mm.llll].

AP naj uporabniku preko GUI (kot je to bilo v BS za kvalificirane količine) omogoča **pregled vseh vnesenih nastavitev** tehničnih sposobnosti za zagotavljanje izravnalne moči, glede na spodnje parametre:

- vrsta ponudbe (Offer Type): uporabnik ima pregled vrste ponudbe;
- vrsta storitve (Business Type): uporabnik ima pregled vrste storitve za katero želi uveljaviti omejitev (RVF, aRPF, rRPF);
- smer izravnave (Direction);
- maksimalna tehnična sposobnost [MW];.

- veljavnost omejitve od (Valid From) [dd.mm.llll];
- veljavnost omejitve do (Valid Thru) [dd.mm.llll].

Implementira se paralelni vnosi, tabele, UI, kot za kvalifikacije, saj kriteriji niso enaki.

Avtomatizem dražb

Splošna podpora za avtomatične dražbe

1.1 AP naj uporabniku omogoča sprotno (avtomatično) izdelavo dražb.

1.1.1. Vzporedno z obstoječimi »auction instance«, se izdela nov šifrant, npr. »auction instance template«. Po vsebini je podoben sedanjim instancam, le da se pri vnosu definira koliko dni pred začetkom dobave se instanca materializira. Ostali parametri (announcement,. gate open, close…) so že del definicij produktov. Če ima avkcija več produktov, se vzamejo podatki s prvega. Kreiranje instance je vedno relativno na prvi dan dobave (delivery from) in se nikoli ne prilagaja drugim parametrom (npr. announcement, bidding). Uporabnik mora sam poskrbeti, da bo nastavil smiselno kombinacijo. Ostali parametri (npr. announcement, bidding) se izračunajo šele, ko se instanca sploh kreira.

1.1.2. Ob vsakokratnem poskusu avtomatične izdelave dražbe se izoblikuje sistemsko sporočilo (v desnem spodnjem okvirčku), ki poleg ID dražbe tudi javi, ali je bila izdelava uspešna ali ne. V primeru, da izdelava ni bila uspešna, je v sporočilu naveden tudi razlog. V primeru neuspešne izdelave je to sporočilo lahko ali interna napaka (internal error), ali pa da je validacija instance vrnila napako (ta del je sicer enak kot ob ročnem vnosu, ista preverjanja).

1.1.3. Vsaka kreirana avtomatska dražba mora vsebovati vse parametre, ki jih vsebuje trenutno ročno kreirana dražba. To pomeni da v kolikor je odkljukana avtomatična izdelava je potreben še parameter, koliko dni pred D (Delivery day oziroma delivery start day v primeru dražb za obdobje daljše od enega dneva) se dražba kreira (dejansko se mora kreirati pred pošiljanjem specifikacij zato bi vse vezali na D (Delivery day). Prav tako je potreben pri Bidding start time določiti koliko dni pred D se začne (trenutno tega ni oziroma se določi pri kreaciji dražbe). Bidding time je prav tako vezan na parametre..

1.1.4. Dodatne kontrole pri vnosu:

1.1.4.1. Announcement date days before delivery start >= bidding gate open date > delivery from.

1.1.4.2. Create instance days before delivery start > announcement date days before delivery start.

1.1.4.3. Šele ob kreiranju instance se točno določi bidding date, glede na dodatne parametre (npr. dan v tednu, dan meseca – kot v BS). Uporabniška napaka je, če se vpiše taka kombinacija, da se instanca kreira po tem ko je izračunan bidding date (ali nanj vezan announcement date) že mimo. Tega sistem ne bo preverjal, ker mora najprej sploh izdelati instanco.

1.1.5. Ob s strani uporabnika določeni uri naj sistem preveri:

1.1.5.1. katere instance dražb je potrebno kreirati danes in jih doda v bazi. Preverjanje deluje na način, da ima dražba dodaten parameter, koliko dni pred dobavo se instanca izdela.

1.1.6. Instanca je normalna entiteta, kot če bi bila ločeno ročno vnesena. To pomeni, da jo uporabnik lahko modificira v skladu z obstoječo implementacijo, vključno z brisanjem (torej enak obseg editiranja kot je že narejen).

Ponujena količina za dnevne dražbe

Ponujena količina za dnevne dražbe:

- AP naj omogoča uporabniku vpis potrebovane količine na nivoju posamezne storitve in smeri neodvisno od ročnosti dražbe (torej vsota večletnih, letnih, mesečnih in dnevnih) za določeno obobje (npr. mFRR+ 250 MW od 1.1.2020 – 28.2.2022; 190 MW od 1.3.2022 – 1.5.2022).
- Na podlagi vpisane količine in rezultatov predhodnih dražb naj aplikacija sama preračuna, koliko je potrebno zakupiti na dnevni dražbi, da bo zakupljena celotna potrebna količina.
- V primeru, da izračun ni mogoč (recimo, ker je uporabnik pozabil vpisati skupno potrebno količino), naj se pojavi ustrezno sistemsko obvestilo, na podlagi katerega se proži alarm.

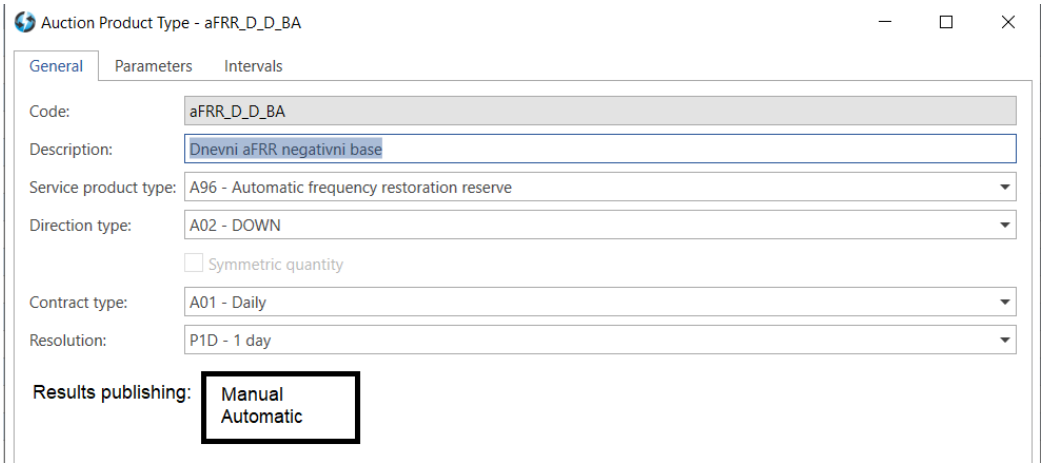
Avtomatično zaključevanje dražb

Zaradi uvedbe dnevnih dražb se pojavlja potreba po avtomatičnem zaključevanju dražb. Trenutno mora uporabnik po vsaki zaključeni dražbi ročno sprožiti objavo rezultatov. Avtomatično zaključevanje dražb naj bi tako zajemalo predvsem avtomatično objavo rezultatov po nekem pretečenem obdobju, in sicer:

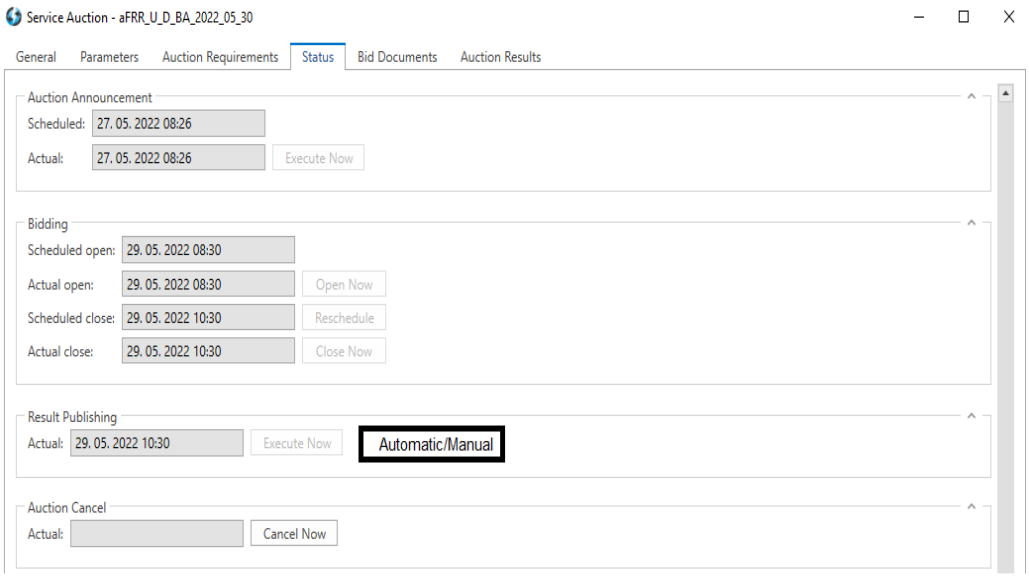
- Na trenuten prikaz Status se doda še predvidena ura objave rezultatov (podobno kot je za Auction announcement in bidding).
- Vrednost se prepiše iz Auction Product.
- Operater ima možnost prestaviti čas objave rezultatov s klikom na gumb reschedule (podobno kot pri bidding scheduled close), kjer ima tudi možnost izbrisa časa objave (s tem se dejansko prekliče avtomatična objava rezultatov).
- Če je bila zakupljena celotna povpraševana količina, se gre v avtomatsko objavo po pretečenem vpisanem roku.
- Če ni bila zakupljena celotna povpraševana količina, se avtomatska objava rezultatov ne izvede.
- Avtomatska objava rezultatov se izvede ob zaključku posamezne dražbe (GCT) le v primeru, če je čas avtomatične objave nastavljen in da je izpolnjen pogoj sprejeta količina = povpraševani količini.
- Avtomatika pri FCR dražbah se ne spreminja.

Nastavitev načina objave:

- AP naj v okviru kreiranja produkta omogoča, da se za dražbe aRPF/rRPF nastavi možnost avtomatske/ročne objave rezultatov (glej sliko spodaj). Katera opcija (Manual ali Automatic) je izbrana mora biti vidno.



- Preklop med avtomatsko in ročno objavo rezultatov je možen izvesti tudi na nivoju same dražbe.



3.2.4. XML transformacija:

XML transformacijska matrika (TM) (matrika, ki omogoča prilagoditev in preoblikovanje vsebine dokumenta izhodnih in vhodnih XML dokumentov):

- uporabnik ima možnost priprave transformacijske matrike (TM) v XML obliki, ki omogoča:

- uporabo ene TM za več pravil za pošiljanje/prejemanje dokumentov; Isto transformacijsko matriko se lahko vpiše k več pravilom.
- uporabo več TM (razširjena TM) za posamezno pravilo za pošiljanje dokumentov.

V novem AP sistemu je vloga SO = AO, ki se veže na izključno eno lokalno območje (bidding zone).

1.1.1. Dokumenti

Dodajanje in urejanje dokumentov

AP naj uporabniku v GUI omogoča dodajanje, razvrščanje, urejanje in brisanje XML dokumentov.

Potrebno je dodati povezavo do ACK dokumenta in obratno (zavihek Related document).

Pregled in iskanje dokumentov

V tabelo je potrebno dodati stolpec za potrditev prejema Acknowledgement dokumenta (datetime polje).

Med filtre za iskanje dokumentov je potrebno dodati Auction Identification.

1.1.2. Sistem za obveščanje

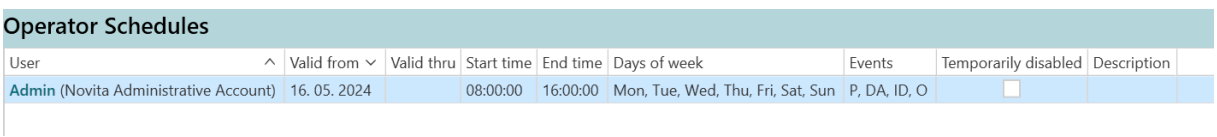
Operator Schedules (urniki)

Prvi korak je da se implementira Operator Schedules pregled (t.i. urniki). Urnik se implementira kot tabela, ki vsebuje naslednje izčiščene podatke:

- Uporabnik
- Veljavnost od-do (dd. MM. yyyy)
- Aktivne ure od-do (HH:mm)
- Aktivni dnevi (pon, tor, sre, … ned)
- Tip procesa (day-ahead, intraday, post, ostalo)
- Urnik (začasno) izklopljen (da/ne)

Za vsakega uporabnika lahko vnesemo poljubno število urnikov. Sporočilo se pošlje, če so pogoji izpolnjeni vsaj po enem zapisu. Če je izpolnjenih več, se sporočilo vseeno pošlje samo enkrat.

Pregled:



Pregled ima filter, default je valid from (ponedeljek tekočega tedna + 4 tedne) + user. Podatki se prikažejo ob pritisku na Search. Na »tabelarni pregled« (list view) GUI se v primeru, da ima uporabnik označene vse dneve, izpiše All namesto posameznih dni in enako za Events.

Podatki na maski za vnos sledijo prikazu zgoraj. Operator schedules ima tudi »Clone« možnost, ki predstavlja podvojitev že obstoječega vpisa. Možnost uporabniku odpre pred-izpolnjeno masko za vnos urnika, kar omogoča hitrejši vnos, običajno samo s spremembo datumov dežurstva.

Načini obveščanja (alarmi) in filtriranje dogodkov

Sistem bo omogočal/podpiral naslednje štiri tipe obveščanja (več o posameznem v nadaljevanju):

- Sistemska obvestila (System notifications)
- Obdelava dokumenta (Document Processing)
- Izidi (status) obdelave zahtevkov (Request log event)
- Spremembe statusa obdelave datoteke (Exchanged message event)

Naslovniki obvestil so lahko:

- Posamezni uporabniki po urniku;
- Vsi uporabniki v določeni vlogi (implementira se, da lahko uporabnik v sistemu nastavil več prejemnikov za posamezen alarm in sicer preko izbire User Role Types (npr: OPERATOR, ADMINISTRATOR, itd).);
- Neposredno vpisani e-mail naslovi.

Ne glede na pot, po kateri pridemo do uporabnika, ki naj prejme obvestilo, se v naslednjem koraku preveri še, ali ima aktiven urnik. Če je dogodek opremljen s podatkom o statusu dneva, mora biti ta označen tudi v urniku. Če za dogodek status dneva ni znan, mora imeti urnik označeno »ostalo«.

To pomeni, da se urnik uporabnika preverja v obeh primerih:

- Če je uporabnik neposredno vpisan kot prejemnik obvestila
- Če je vpisana vloga, ki je dodeljena uporabniku

Za obveščanje na samostojno vpisan e-mail naslov (enega ali več) pogoji urnika in izbranih opcij ne veljajo.

Uporabniku oziroma vsem v izbrani vlogi lahko pošljemo obvestilo na naslednje načine:

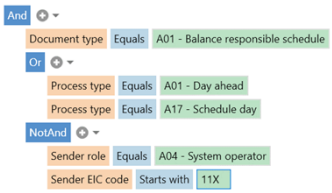
- V aplikacijo, kot sicer običajno obvestilo. Uporabnik ga prejme, če je prijavljen. Tako konfigurirano obvestilo je vedno dodatno k sporočilom, ki jih sistem proži brez posebne konfiguracije.
- Kot Windows Toast obvestilo. Tudi za ta način mora biti uporabnik prijavljen v aplikaciji.
- Na vpisan e-mail naslov.
- Na SMS številko.

Implementira se tudi informacija, da uporabnik zve, katero pravilo je alarm sprožilo.

Celoten obstoječ način proženja in sestavljanja sporočil se izvede tako, da je za proženje možno vnesti **filtrirni pogoj**, za sporočilo pa se sestavi poljuben tekst s parametri.

Filtrirni pogoji imajo različno število parametrov, ki se razlikujejo tudi od izbranega sprožilca in so podrobneje opisani v nadaljevanju te specifikacije. Vnos pogoja se izvede z vnosom formule, v kateri se lahko uporabi tudi različne logične funkcije (and, or, not…). Tak način je podoben vnosu formul za razne omejitve, le da mora formula na osnovi vsebovanih parametrov vrniti logični rezultat da / ne (boolean). V formuli se lahko uporabi le del ponujenih parametrov; posamezni parameter se lahko uporabi tudi večkrat (npr. v različnih členih formule).

Filtrirni pogoj se določi z vnosom logične operacije s parametri, ki so vpisani so odvisni od posameznega tipa dogodka. Možen je vnos sestavljenih pogojev z več gnezdenimi operatorji. Posamezen atribut filtriranega dogodka se lahko tudi večkrat ponovi, kar poenostavi in hkrati zmanjša število potrebnih vnosov konfiguracij. Omogoča tudi vnos negativnega pogoja:



Uporabniki si lahko nastavijo tudi lasten Notification Level, podprti so: Critigal Warning, Warning, Error, Success in Information. Uvede se tudi Notification Level Sucess. Nov level velja izključno za konfiguracijo, ki jo vnese uporabnik in ne zajema morebitnega spreminjanja s katerim level se pošlje nek notification iz kode (hardcoded notifications).

Sestavljanje izhodnih tekstov

Posamezno obvestilo o dogodku ima 4 polja za tekste, ki se uporabljajo:

- Za prikaz v aplikaciji
- Za naslov e-mail sporočila
- Za vsebino e-mail sporočila
- Za vsebino SMS sporočila

V vsako polje lahko neodvisno vpišemo želen tekst. Če pustimo polje prazno, se uporabi vnaprej fiksno definirano sporočilo (kot se izpisuje v aplikaciji v logu). Tudi ti parametri so odvisni od izbranega sprožilca. Parametri se v času pošiljanja nadomestijo s takratnimi vrednostmi.

V posamezno polje lahko namesto preprostega teksta vpišemo tudi XSL predlogo, ki se uporabi za naprednejše formatiranje sporočila. XSL način procesiranja se sproži, kadar se tekst začne z:

<xsl:template

Parametri se lahko pojavijo v obliki {ImeParametra:Format}, pri čemer je ime parametra eno od definiranih za posamezen način proženja, format pa je prevzet iz .NET specifikacije. Primeri so:

- dd. MM. yyyy – formatiran datum
- N2 – število z dvema decimalkam
- …

Format je neobvezen element. Če ga izpustimo, je potrebno izpustiti tudi dvopičje. Pri uporabi parametrov znotraj XSLT, se vpisana predloga najprej obravnava kot običajen tekst, v katerem se napolnijo parametri, nato pa se izvede transformacija. Parametri filtra so vedno vključeni v parametre za tekst sporočil. Pri parametrih za tekste so nekateri podvojeni in se končajo z »Code« ali »Name«. V tem primeru gre za parameter iz nabora konstant (npr. DocumenType) in ga ponudimo prvič kot šifro (Axx) in drugič kot ime (»Balance Responsible Schedule«).

Opis tipov obveščanja

Sistemska obvestila – v aplikaciji: SystemNotification.

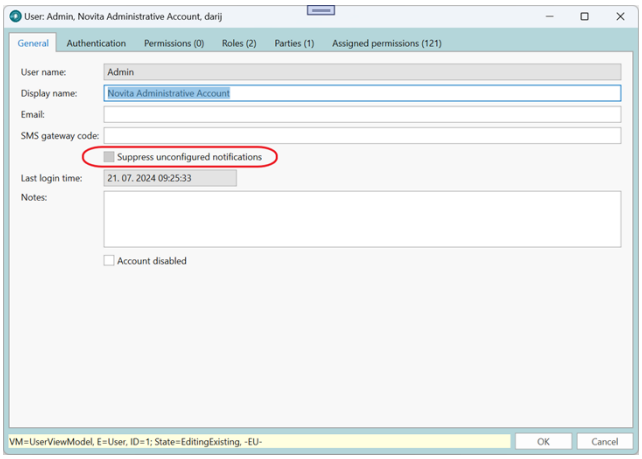
Parametri za filter:

- SystemNotification – kratka koda sporočila
- ScheduleDayStatus – status dneva, če je sporočilo vezano na tak dogodek
- DomainEicCode – šifra meje ali lokalnega področja, če je sporočilo vezano na tak dogodek

Dodatni parametri za tekst:

- EventTime – čas dogodka
- NotificationText – vnaprej definiran tekst sporočila
- ScheduleDayStatusCode
- ScheduleDayStatusName
- System Name

Implementira se podpora, s katero se lahko na uporabniku definira, ali ta prejema samo nastavljena sistemska obvestila in ne vsa hardcoded, kot zdaj.



Obdelani dokumenti – v aplikaciji: DocumentNotification. Proži se za dokumente, ki (oziroma ko) se uspešno obdelajo in zapišejo v bazo, ne glede na smer izmenjave (vhodni, izhodni, interno izdelani). Ta tip obveščanja ne deluje neposredno za zavrnjene dokumente. Deluje pa za negativni ACK ki se obdela ob zavrnitvi.

Parametri za filter:

- DocumentType
- SenderEicCode
- SenderRole
- ReceiverEicCode
- ReceiverRole
- ProcessType
- DomainEicCodes – nabor domen (mej) ki se pojavljajo v glavi ali serijah dokumenta
- BusinessType – nabor vseh kod v serijah dokumenta
- ScheduleDayStatus
- ReportType
- Referenced Document Type

Dodatni parametri za tekst:

- EventTime
- DocumentTypeCode
- DocumentTypeName
- SenderRoleCode
- SenderRoleName
- ReceiverRoleCode
- ReceiverRoleName
- ProcessTypeCode
- ProcessTypeName
- ScheduleDate
- DocumentIdentification
- DocumentVersion
- SenderDisplayName

- ReceiverDisplayName
- ScheduleDayStatusCode
- ScheduleDayStatusName
- ReportTypeCode
- ReportTypeName
- ReferencedDocumentType
- ReferencedDocumentTypeCode
- ReferencedDocumentTypeName
- System Name

Obdelani zahtevki (Request Statuses Notifications) – v aplikaciji: RequestNotification. Gre za dodatni vir obvestil, ki se sproži po koncu obdelave posameznega zahtevka. Pogoj za proženje je, da se je status obdelave uspešno zapisal v podatkovno bazo (vključno s primerom, ko se zapiše status »internal error«).

Parametri za filter:

- RequestType – kratka šifra tipa zahtevka
- RequestStatus
- UserName – uporabniško ime, le kadar je znan oziroma ne gre za interni zahtevek
- ProcessingDuration – trajanje obdelave

Okvirni podprti parametri:

- EventTime
- RequestStatusCode
- RequestStatusName
- RequestDescription – opis zahtevka, kot se sicer prikazuje v logu aplikacije.
- BeginProcessing
- EndProcessing
- System Name

Izmenjava datotek (Document Exchange Payload Statuses Notifications) – v aplikaciji: DocumentExchangeNotification. Dodatni vir obvestil, ki se proži ob vsakokratni spremembi statusa obdelave datoteke. Pogoj za proženje je, da se je status obdelave uspešno zapisal v podatkovno bazo (vključno s primerom, ko se zapiše status »internal error«). Med parametri je tudi nekaj takih, ki so vezani na vsebino XML dokumenta. Ti bodo napolnjeni šele po branju XML vsebine. Glede na pravilnost dokumenta je možno, da nekateri parametri ne dobijo vrednosti. Če se filtrirni pogoj sklicuje na specifično vrednost, se ne bo sprožil, če parameter (še) nima prirejene vrednosti.

Parametri za filter (na Exchanged Messages nimamo na voljo vseh podatkov, kot jih imamo za dokument po procesiranju, zato so filtri lahko sledeči):

- PayloadStatus
- FileName
- FileLength

- DocumentType
- SenderEicCode
- ReceiverEicCode
- TimeIntervalFrom
- TimeIntervalTo
- DocumentIdentification
- DocumentVersion
- DocumentDateTime
- RootElementName

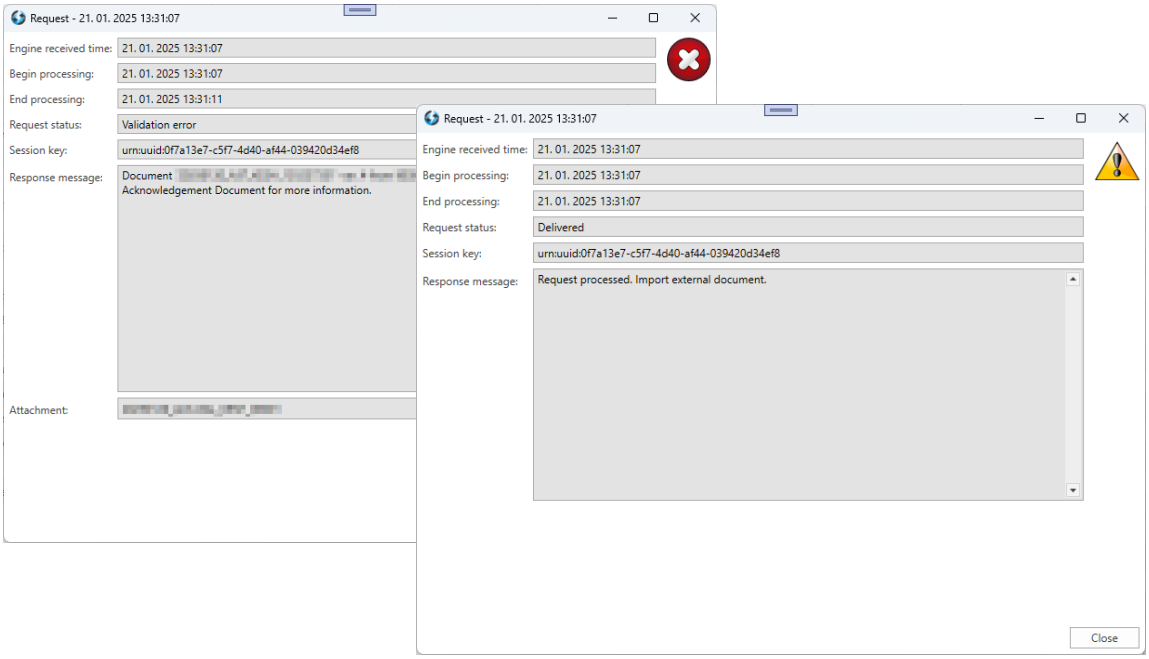
Dodatni parametri za tekst:

- EventTime
- PayloadStatusCode
- PayloadStatusName
- DocumentTypeCode
- DocumentTypeName
- System Name

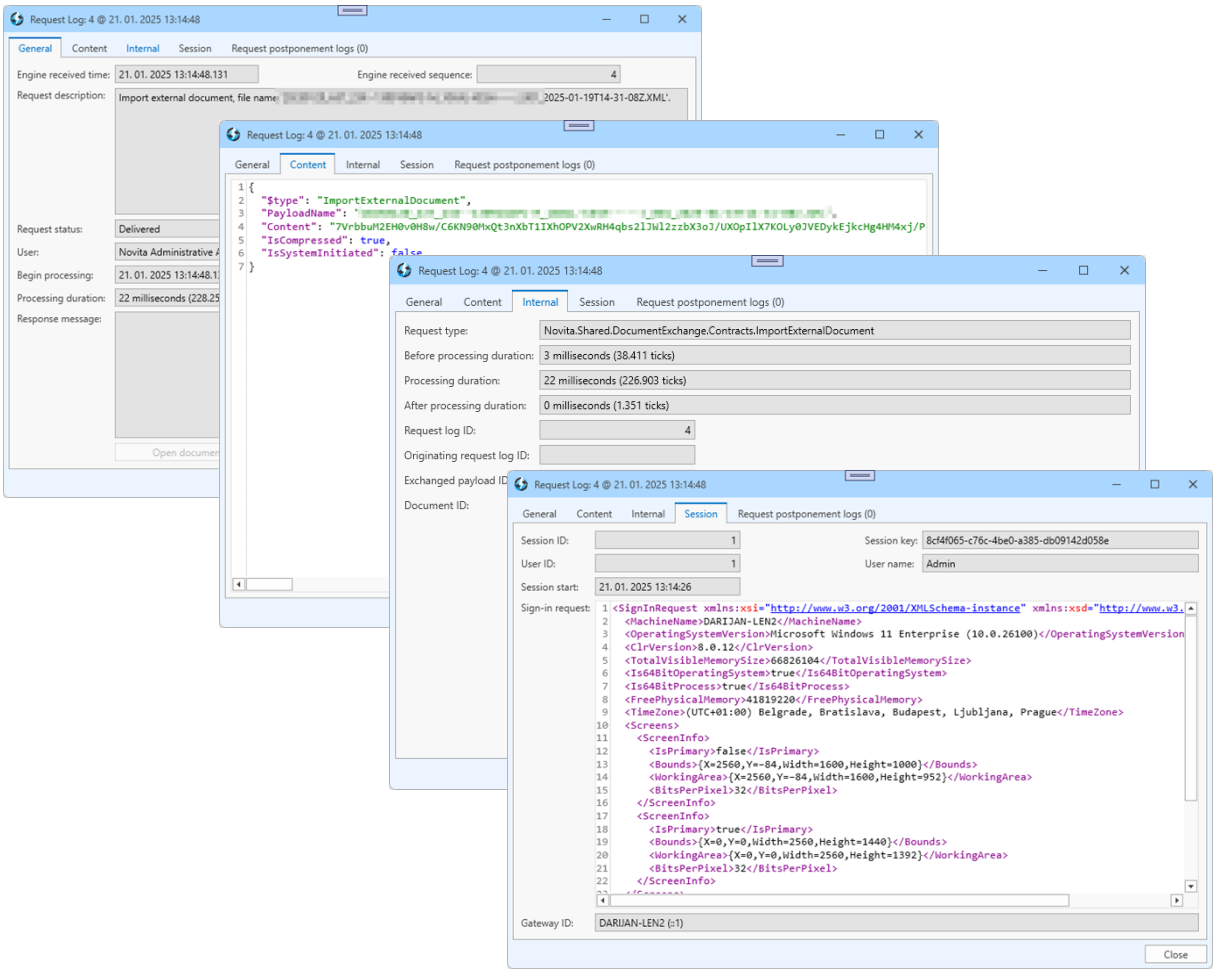
Nov izgled okna sistemskih notifikacij in odgovorov

S predelavo sistemskih notifikacij se nekoliko spremenijo tudi grafični pregledi (vizualni izgled) odgovorov, notifikacij sistema. Vsebina se razširi z dodatnimi podatki, ki pomagajo pri raziskovanju morebitnih težav in razumevanju delovanja sistema.

V starem sistemu smo imeli:



V novem sistemu se poenostavljena pregleda zamenja z novimi z več informacijami



V AP se dodajo novi system notifications, ki jih lahko uporabniki integrirajo v alarme kot definirano v osnovni rešitvi. Ti system notifications so:

- **Notification**: Result optimization warning treshold miliseconds exceeded.
- **Notification**: X minut pred zaključkom dražbe (GCT) ni zakupljena celotna povpraševana količina (*pri čemer je X nastavljiv s strani uporabnika, zakupljena celotna količina pa pomeni kot pri KPI (ponujena količina ≥ povpraševane količine), X minut pred GCT je del notification teksta, da operator vidi npr. 5 mon pred zaključkom dražbe on 13:00 ni zakupljena celotna količina*);
- **Notification**: Ni alocirana vsa količina, triger GCT (*KPI (alocirana količina ≥ povpraševane količine)*;
- **Notification**: Sprememba statusa dražbe (Send ASP, Start Bidding, End Bidding, Resoults published …).

1.1.3. Konfiguracija časovnih serij (Time Series Configuration)

Time Series (TS) Configuration omogoča uporabnikom, da določijo in upravljajo različne nastavitve ter parametre, ki nadzorujejo, kako se podatki TS obdelujejo in usklajujejo v sistemu. TS konfiguracija je ključna za zagotavljanje natančne obdelave in upravljanja podatkov TS v skladu s specifičnimi poslovnimi pravili in zahtevami. V stari verziji sistema so bila ta pravila »hardcoded«, težka za vzdrževanje in implementacijo (vsaka sprememba je zahtevala nadgradnjo verzije v produkciji), z navedeno implementacijo pa se konfiguracija prenaša v administrativni del aplikacije, kjer jo lahko preko grafičnega vmesnika urejajo administratorji sistema. Uporabniki lahko tako na ravni dokumenta urejajo:

- DocumentSchemaType: Določa vrsto sheme dokumenta.
- DocumentLabel: Oznaka za identifikacijo dokumenta.
- DocumentType: Vrsta dokumenta.
- ReportType: Vrsta poročila, povezana z dokumentom.
- SenderRole: Vloga pošiljatelja.
- ReceiverRole: Vloga prejemnika.
- Sender: Pošiljatelj dokumenta.
- Receiver: Prejemnik dokumenta.
- ProcessType: Vrsta procesa, povezanega z dokumentom.
- ClassificationType: Vrsta klasifikacije dokumenta.
- Domain: Področje dokumenta.

Poleg tega imajo uporabniki na voljo sledeče parametre na ravni TS:

- BusinessType: Vrsta posla/podatka
- InArea_ConnectingArea_Area: Vhodno območje, povezovalno območje ali entiteta območja.
- OutArea_AcquiringArea_SubjectArea: Izhodno območje, pridobljeno območje ali entiteta predmetnega območja.
- ContractType: Vrsta pogodbe, povezana s podatki v TS.

Vsak zapis ima parameter veljavnosti:

- ValidFrom: Datum začetka veljavnosti konfiguracije.
- ValidTo: Datum konca veljavnosti konfiguracije.

Uporabniki lahko tudi sami določijo prioriteto parametra razvrščanja (Priority) in druge ključne nastavitve:

- Resolution: Določitev resolucije podatkov.

- QuantityDecimals: Število decimalnih mest za količine.
- AmountDecimals: Število decimalnih mest za zneske in cene.

Koraki določanja uporabe TS Configuration parametrov:

1. Identifikacija vhodnih podatkov – prvi korak je identifikacija vhodnih podatkov, ki običajno vključujejo: Podrobnosti dokumenta, kot so vrsta sheme, vloge pošiljatelja in prejemnika, časovni interval idr.
2. Kreiranje ključa za ujemanje – na podlagi identificiranih vhodnih podatkov se ustvari ključ za ujemanje. Ta ključ služi kot edinstveni identifikator za iskanje ustrezne konfiguracije in vključuje vse relevantne podrobnosti iz ravni dokumenta in ravni TS.
3. Preverjanje predpomnjenih konfiguracij – sistem nadalje preveri, ali obstaja predpomnjena konfiguracija, ki ustreza ustvarjenemu ključu. Predpomnjenje izboljša zmogljivost, saj preprečuje ponavljajoče reševanje enakih konfiguracij.
4. Če predpomnjena konfiguracija ne obstaja, sistem pridobi vse razpoložljive konfiguracije. Te konfiguracije se nato filtrirajo glede na to, ali ustrezajo podanim vhodnim podatkom.
5. Filtriranje in razvrščanje konfiguracij – pridobljene konfiguracije se filtrirajo, da vključujejo samo tiste, ki ustrezajo vhodnim podatkom. Nato se razvrstijo glede na njihovo prioriteto in specifičnost. Prioriteta določa vrstni red uporabe konfiguracij, medtem ko specifičnost zagotavlja.
6. Izbira najboljše konfiguracije – iz razvrščenega seznama se izbere najboljša konfiguracija. To je konfiguracija, ki najbolje ustreza vhodnim podatkom in ima najvišjo prednost ter specifičnost.
7. Predpomnjenje rešene konfiguracije – rešena konfiguracija se shrani v predpomnilnik za prihodnjo uporabo. To izboljša zmogljivost, saj zmanjša potrebo po ponovnem reševanju enakih konfiguracij.
8. Uporaba konfiguracije – izbrana konfiguracija se uporabi za podatke v TS. To vključuje nastavitev različnih parametrov, kot so ločljivost, število decimalnih mest za količine, število decimalnih mest za zneske in roki za nominacije.

Ko sistem uskladi konfiguracije časovnih vrst, ustvari rezultat, ki vsebuje specifične nastavitve in parametre. Ta rezultat se uporabi za natančno obdelavo podatkov časovnih vrst v skladu z usklajenimi konfiguracijami, in sicer:

1. Zbiranje usklajenih konfiguracij – sistem najprej zbere vse konfiguracije, ki ustrezajo podanim vhodnim podatkom. Te konfiguracije se filtrirajo in razvrstijo glede na njihovo prednost in specifičnost.
2. Določitev najboljše konfiguracije – iz razvrščenega seznama usklajenih konfiguracij sistem izbere najboljšo konfiguracijo, običajno tisto z najvišjo prednostjo in specifičnostjo.

3. Izvleček parametrov – sistem nato iz izbrane konfiguracije izvleče relevantne parametre, kot so ločljivost, število decimalnih mest za količine in zneske ter roki za nominacije.
4. Ustvarjanje rezultata – z uporabo izvlečenih parametrov sistem ustvari objekt rezultata. Ta objekt vsebuje vse potrebne nastavitve in parametre za obdelavo podatkov časovnih vrst.

Poleg opisanega po vzoru Pary\Outgoing Document Rules vsebuje pregled še polje Description. Opisno polje, kjer is operater zabeleži namen definirane konfiguracije ali kakšno koli poljubno zabeležko. Ne vpliva na samo delovanje konfiguracije časovnih serij.

1.1.4. Poročila

AP naj uporabniku preko GUI omogoča pregled tabelaričnih (pivot) poročil glede na izbrano časovno obdobje v obliki [dd/mm/llll] in glede na naslednje parametre:

- stranka (Party): uporabnik naj izbere stranko za katero želi izdelati poročilo;
- ločljivost (Resolution):
  - 60 min;
  - 15 min.
- kategorija produkta (aFRR, mFRR, FCR)
- smer izravnave (Direction);
- identifikacija dražbe (Auction Identification);

Podatki (center pivota):

- BidQty
- BidPrice
- Qty
- Price
- Demand Qty

AP naj uporabniku omogoča izvoz izdelanega pivot poročila v obliki (.xlsx). Tiskanje in postavitev podatkov pred tiskanjem se ureja izven sistema, v Excelu.

Novi Formatted Reports - Pregled tehničnih sposobnosti PSI

Za večjo preglednost vseh tehničnih sposobnosti za vse PSI za vse storitve za posamezen dan dobave se v AP dodela nov prikaz. Uporabnik lahko poljubno izbere datum dobave pregleda. Pregled se izdela s klikom na gumb SEARCH.

Skica (filter zgoraj, spodaj vsebina):

Date: DD.MM.YYYY	SEARCH

PSI	STORITEV	TEHNIČNA SPOSOBONOST	VELJAVNOST OD	VELJAVNOST DO
PSI 1	FCR +-	0	DD.MM.YYYY	DD.MM.YYYY
PSI 1	AFRR +	10	DD.MM.YYYY	DD.MM.YYYY
PSI 1	AFRR -	15	DD.MM.YYYY	DD.MM.YYYY
PSI 1	MFRR +	20	DD.MM.YYYY	DD.MM.YYYY
PSI 1	MFRR -	30	DD.MM.YYYY	DD.MM.YYYY
PSI 2	FCR +-	10	DD.MM.YYYY	DD.MM.YYYY
PSI 2	AFRR +	20	DD.MM.YYYY	DD.MM.YYYY
PSI 2	AFRR -	155	DD.MM.YYYY	DD.MM.YYYY
PSI 2	MFRR +	203	DD.MM.YYYY	DD.MM.YYYY
PSI 2	MFRR -	307	DD.MM.YYYY	DD.MM.YYYY
PSI 3	FCR +-	17	DD.MM.YYYY	DD.MM.YYYY
PSI 3	AFRR +	18	DD.MM.YYYY	DD.MM.YYYY
PSI 3	AFRR -	19	DD.MM.YYYY	DD.MM.YYYY
PSI 3	MFRR +	20	DD.MM.YYYY	DD.MM.YYYY
PSI 3	MFRR -	30	DD.MM.YYYY	DD.MM.YYYY

Novi Formatted Reports - Pregled kvalificiranih količin

Za večjo preglednost vseh kvalificiranih količin za vse PSI za vse storitve za posamezen dan dobave se v AP dodela nov prikaz. Uporabnik lahko poljubno izbere datum dobave pregleda. Pregled se izdela s klikom na gumb SEARCH.

Skica (filter zgoraj, spodaj vsebina):

Date: DD.MM.YYYY	SEARCH

PSI	STORITEV	KVALIFICIRANE KOLIČINE	VELJAVNOST OD	VELJAVNOST DO
PSI 1	FCR +-	0	DD.MM.YYYY	DD.MM.YYYY
PSI 1	AFRR +	10	DD.MM.YYYY	DD.MM.YYYY
PSI 1	AFRR -	15	DD.MM.YYYY	DD.MM.YYYY
PSI 1	MFRR +	20	DD.MM.YYYY	DD.MM.YYYY
PSI 1	MFRR -	30	DD.MM.YYYY	DD.MM.YYYY
PSI 2	FCR +-	10	DD.MM.YYYY	DD.MM.YYYY
PSI 2	AFRR +	20	DD.MM.YYYY	DD.MM.YYYY
PSI 2	AFRR -	155	DD.MM.YYYY	DD.MM.YYYY
PSI 2	MFRR +	203	DD.MM.YYYY	DD.MM.YYYY

PSI 2	MFRR -	307	DD.MM.YYYY	DD.MM.YYYY
PSI 3	FCR +-	17	DD.MM.YYYY	DD.MM.YYYY
PSI 3	AFRR +	18	DD.MM.YYYY	DD.MM.YYYY
PSI 3	AFRR -	19	DD.MM.YYYY	DD.MM.YYYY
PSI 3	MFRR +	20	DD.MM.YYYY	DD.MM.YYYY
PSI 3	MFRR -	30	DD.MM.YYYY	DD.MM.YYYY

Novi Formatted Reports - Pregled tehničnih sposobnosti in kvalificiranih količin

S tem Formatted report uporabniku za dan dobave prikažemo ali je tehnična sposobnost posameznega PSI za posamezno storitev nižja od kvalificirane **za posamezen dan dobave**. Kvalificirana količina je lahko enaka ali manjša od tehnične sposobnosti. Na seznamu so samo taki PSI oz. PSI po storitvah. Če je za posamezno storitev PSI kvalificirana količina enaka ali manjša od tehnična sposobnosti, zapisa ni na reportu.

Pregled se izdela s klikom na gumb SEARCH.

Skica (filter zgoraj, spodaj vsebina):

Date from: DD.MM.YYYY	Date thru: DD.MM.YYYY	SEARCH

Datum dobeve: dd.mm.yyyy

Datum	PSI	STORITEV	TEHNIČNA SPOSOBONOST	KVALIFICIRANE KOLIČINE
1.1.2024	PSI 1	FCR +-	0	0
	PSI 1	AFRR +	10	5
3.1.2024	PSI 1	AFRR -	15	10
	PSI 1	MFRR +	10	20
	PSI 1	MFRR -	30	30
8.1.2024	PSI 2	FCR +-	10	10

Prenos rezultatov dražb iz AP v BS

Za potrebe uparjevanja rezultatov dražb za moč in oddanih energijskih ponudb mora AP izdelati in poslati ustrezen dokument z rezultati dražb v BS. Narejena naj bo izdelava RARD na publish avkcije v fiksni 15min resoluciji (ki upošteva profil produkta).

Iz AP se torej v BS pošilja:

- Dnevni presek rezultatov vseh avkcij in ročno vnesenih pogodb za moč v kumulativni obliki po tipu produkta,. Format: RARD v6r4 fiksno.

Iz AP se proti ponudbnikom pošilja (poleg dokumentov ki so del objave, zaključevanja avkcije) še:

- Dnevni presek kapacitet, kumulativno, po tipu produkta (aFRR, mFRR, FCR). Format: RARD v6r4 fiksno.

Nastavitev pošiljanja poročila

AP naj omogoča pošiljanje in odlaganje dokumentov zunanjim sistemom na osnovi predhodno določenih pravil preko različnih komunikacijskih kanalov za potrebe:

- ostalim zalednim poslovnim in tehničnim sistemom OPS;
- sodelovanja v mednarodnih kooperacijah in platformah;
- objavljanja podatkov na ETP;
- itd.

1.1.5. Pomoč uporabniku – Integrirana tehnična dokumentacija

Uporabniku naj bo v AP na razpolago (vgrajena) zadnja razpoložljiva dokumentacija/ uporabniški priročnik.

AP naj uporabniku preko GUI omogoča ogled in izvoz tehnične dokumentacije za pomoč pri rokovanju z aplikacijo AP, procesi in različnimi XML dokumenti in shemami in sicer v obsegu, in v obliki kot je bil v BS, v delih ki je še (koliko je) relevanten za AP.

AP naj uporabniku preko GUI omogoča ogled uporabniške/tehnične dokumentacije. Pomoč naj bo dosegljiva iz menija "Pomoč".

1.2. PILOTSKI PROJEKT – kontejnerske tehnologije

Namen pilotskega projekta je preveriti, ali je obstoječe .NET 8 aplikacije mogoče uspešno prenesti v Docker Windows okolje. S tem naročnik želi preveriti izvedljivost prenosa, prepoznati morebitne omejitve ali tveganja ter pridobiti vpogled v korake, ki bi bili potrebni

za dokončno implementacijo obstoječih aplikacij v kontejnerskem okolju. Rezultati pilotskega projekta naj vključujejo analizo skladnosti delovanja aplikacije v kontejnerskem okolju, identifikacijo morebitnih neskladnosti ter pripravo priporočil za nadaljnje korake v procesu implementacije.

Izvajalec bo moral zagotoviti pristop, ki bo temeljil na naslednjih ključnih fazah:

- analiza obstoječega stanja in priprava strategije prenosa aplikacij,
- priprava kontejnerske infrastrukture in implementacija tehničnih rešitev,
- testiranje delovanja aplikacij znotraj Docker Windows okolja,
- analiza in odprava težav ter priprava ocene izvedljivosti,
- priprava dokumentacije z rezultati, priporočili in podlago za odločitev o naslednjih korakih implementacije.

1.2.1. Pričakovanja in smernice uporabe kontejnerskih tehnologij

Poglavje predstavlja možno varianto namestitve rešitve. Zaenkrat je ta oblika namestitve predvidena za razvojno okolje.

Okolje bo postavljeno v obliki Docker kontejnerizacije, ki je nameščena na gostiteljskem razvojnem okolju. Povezano bo na testne zunanje sisteme in simulirano podatkovno skladišče, ki bo prav tako kontejnerizirano.

Pri uvedbi Docker tehnologije pričakujemo naslednje koristi, s katerimi želimo izboljšati procese razvoja, implementacije in upravljanja aplikacij:

1. Poenostavitev razvoja in implementacije:
   Verjamemo, da Docker omogoča hitro in enostavno pripravo okolij z vsemi potrebnimi odvisnostmi, kar bo zmanjšalo čas in kompleksnost nameščanja aplikacij.
2. Prenosljivost:
   Pričakujemo dosledno delovanje aplikacij v različnih okoljih (razvoj, test, produkcija) brez težav zaradi razlik v konfiguracijah.
3. Izolacija:
   Aplikacije bodo učinkovito ločene s kontejnerji, kar bo preprečilo konflikte med odvisnostmi in povečalo zanesljivost sistema.

4. Skalabilnost:
   Docker vidimo kot orodje, ki omogoča hitro prilagajanje zmogljivosti na podlagi spreminjajočih se potreb.
5. Standardizacija in avtomatizacija:
   Uporaba Dockerja bo poenotila procese razvoja in podprla avtomatizacijo (CI/CD).
6. Učinkovita raba virov:
   Z Dockerjem pričakujemo optimizacijo strežniških virov in zmanjšanje operativnih stroškov.
7. Minimalne motnje ob uvedbi:
   Želimo, da prehod na Docker prinese hitre koristi, ne da bi negativno vplival na obstoječe procese.
8. Skladnost z modernimi praksami:
   Verjamemo, da Docker odraža smernice sodobnega tehnološkega razvoja in podpira dolgoročno učinkovite prakse.

Pričakujemo, da bo sodelovanje z izvajalcem pripomoglo k jasni implementaciji ter praksi, ki bo prilagojena našim specifičnim potrebam.

Docker

Docker je odprtokodna platforma za razvoj, dostavo in izvajanje aplikacij v prenosljivih kontejnerjih. Omogoča sestavo okolja aplikacij skupaj z vsemi njihovimi odvisnostmi v izolirane enote, kar zagotavlja dosledno delovanje v različnih okoljih. Temelji na principih virtualizacije, vendar je učinkovitejši od klasičnih virtualnih naprav, saj uporablja skupno jedro operacijskega sistema. S tem omogoča ločevanje aplikacij od infrastrukture, prenosljivost, skalabilnost ter upravljanje aplikacij, kar je še posebej uporabno v sodobnih razvojnih in produkcijskih okoljih.

Razlika med Docker in virtualno napravo (VM)

Docker ne uporablja hipervizorja, kar pomeni da uporablja gostiteljski OS in njegovo jedro. Docker kontejnerji uporabljajo enak gostiteljski OS, vsebujejo pa le aplikacijo oz. njene odvisnosti. Izolacija se izvede na nivoju procesov, kar omogoča večjo učinkovitost, porabo virov, hiter zagon in veliko prenosljivost. S tem se doseže tudi konsistentno in predvidljivo delovanje ne glede na gostitelja izvajanja.

Glavne prednosti:

- **Prenosljivost** – Aplikacije pakirane v kontejner, delujejo enako na različnih sistemih, ne glede na to, ali so nameščene lokalno, v oblaku ali na strežniku.
- **Izolacija** - Vsak kontejnerj ima svoje okolje, kar preprečuje konflikte med aplikacijami in odvisnostmi.
- **Hitrost** - Kontejnerizacijski pristop omogoča hiter zagon in ustavitev aplikacij, saj ni potrebe po zaganjanju celotnega operacijskega sistema kot pri virtualnih strojih.
- **Učinkovita poraba virov** - Ker kontejnerji delijo isti operacijski sistem, porabijo manj pomnilnika in procesorske moči v primerjavi s klasičnimi virtualnimi stroji.

Kreiranje kontejnerja

Kontejnerizacija je proces ustvarjanja in obvladovanja kontejnerjev. Namesto, da aplikacijo namestimo na posamezni strežnik, jo spakiramo skupaj z vsemi potrebnimi odvisnostmi v kontejner. Tako se zagotovi, da bo aplikacija delovala enako ne glede na okolje, v katerem bo nameščena.

Za kreiranje kontejnerja je najprej potrebno kreirati Docker sliko (image). To lahko storimo s pomočjo Dockerfile, kjer v tekstovni obliki z ukazi opišemo osnovni sistem, odvisnosti in nastavitve aplikacije. Preko gradnje (build) se nato kreira slika, ki jo lahko uporabimo na drugih sistemih. Določeni sistemi (npr. RabbitMQ, MS SQL) imajo uradne javno objavljene slike za zagon kontejnerja, ki jih lahko uporabimo.

YML datoteka

Datoteka YML (tudi YAML) je format za shranjevanje podatkov v strukturirani obliki. Uporablja se za konfiguracijske datoteke v različnih sistemih kot so Docker compose, Kubernetes itd.

Docker-compose

Docker je orodje za ustvarjanje, izvajanje in upravljanje kontejnerjev. Docker-compose je orodje za definicijo in upravljanje večih kontejnerjev, ki delujejo skupaj.

docker-compose.yml je konfiguracijska datoteka, ki se uporablja za definiranje in upravljanje več povezanih Docker kontejnerjev, ki delujejo skupaj v enem okolju. V konfiguraciji se definira uporabljene storitve, prostorske sisteme, omrežja in druge nastavitve za delovanje aplikacij znotraj docker okolja.

Registri kontejnerjev

Registri kontejnerjev so centralizirani repozitoriji za shranjevanje in upravljanje slik, ki so ključne za razvoj in uvajanje aplikacij na osnovi kontejnerjev. Omogočajo shranjevanje, nalaganje in prenos slik, ki vsebujejo vse potrebne datoteke in komponente za zagon aplikacije. Obstajajo različne vrste registrov, vključno z javnimi (npr. Docker Hub), zasebnimi in odprtokodnimi (npr. Harbor). Registri se brezhibno integrirajo v delovne tokove neprekinjene integracije (CI) in neprekinjene dostave (CD), kar olajša upravljanje in uvajanje slik. Pogosto vključujejo funkcije za varnost slik, nadzor dostopa in upravljanje različic.

Shema dockeriziranega okolja



Slika 2: Shema dokeriziranega razvojnega okolja

Servisi/kontejnerji

Interni:

1. Keycloak
    a. Javni register Docker Hub image (keycloak/keycloak:26.1.0)
2. keycloak MS SQL
    a. Javni register Docker Hub image (mcr.microsoft.com/mssql/server:2022-latest)
3. RabbitMQ1
    a. Javni register Docker Hub image (rabbitmq:4-management)
4. NGINX
    a. Javni register Docker Hub
5. Modul Analitika
    a. Zasebni register Harbor

6. XBID primarni agent
    a. Zasebni register Harbor
7. XBID sekundarni agent
    a. Zasebni register Harbor
8. MARI primarni
    a. Zasebni register Harbor
9. MARI sekundarni agent
    a. Zasebni register Harbor
10. PICASSO agent
    a. Zasebni register Harbor

Podroben opis posameznega servisa v prilogi docker-compose.yml

Razvojni:

1. SQL Server DB
    a. Javni register Docker Hub (mcr.microsoft.com/mssql/server:2022-latest)
2. Mailhog
    a. Javni register Docker Hub (mailhog/mailhog)

Podroben opis posameznega servisa v prilogi docker-compose.dev.yml

Inicialna namestitev

Preko ukaza: docker-compose -f docker-compose.dev.yml up -d

Nadgradnja sistema

Posodobitev docker-compose.yml z definicijo drugega image name za posamezni servis. Ukaz enak kot za inicialno namestitev.

Priloge

Datoteka »docker-compose.yaml«

```
1. version: '3.9'
2.
3. x-common-variables: &syslog # Syslog server sistemske spremenljivke
4.   SYSLOG_HOST: ${SYSLOG_HOST} # Host za syslog server (npr. 192.168.1.221)
5.   SYSLOG_PORT: ${SYSLOG_PORT} # Vrata za syslog server (npr. 514)
6.
7. services:
8.   keycloak: # Keycloak -> AVTENTIKACIJA
9.     image: keycloak/keycloak:26.1.0 # Uradni keycloak image verzije 26.1.0 (https://hub.docker.com/layers/keycloak/keycloak/26.1.0/images/sha256-
7e3a06e8e2d1ac0a1461202f8a288e8fa0e15f40fd908927ac0694d3643460d0)
10.     container_name: keycloak # Ime kontejnerja
11.     ports:
12.       - "45000:8080" # Dostop do keycloak preko 45000
13.       - "9990:9990" # Odprta vrata za administrativni dostop
```

Datoteka .env

```
# Syslog server
SYSLOG_HOST=
SYSLOG_PORT=

# Keycloak
KEYCLOAK_DB_PASSWORD=YourStrong!Passw0rd
KEYCLOAK_USER=admin
KEYCLOAK_PASS=admin
KEYCLOAK_DB_PASSWORD=YourStrong!Passw0rd

# RabbitMQ
RABBITMQ_ERLANG_COOKIE=
RABBITMQ_USER=guest
```

Datoteka nginx.conf

```
server {
    listen 80;
    server_name localhost;

    return 301 https://$host$request_uri;
}

server {
    listen 443 ssl;
    server_name localhost;

    ssl_certificate /etc/ssl/certs/nginx.crt;
    ssl_certificate_key /etc/ssl/private/nginx.key;

    location / {
        proxy_pass http://analytics:80;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
    }
}
```

Datoteka rabbitmq.conf

```
cluster_formation.peer_discovery_backend = classic_config
cluster_formation.classic_config.nodes.1 = rabbit@rabbitmq1
```

```
3. cluster_formation.classic_config.nodes.2 = rabbit@rabbitmq2

4. cluster_formation.classic_config.nodes.3 = rabbit@rabbitmq3
```

1.2.2. Priprava in analiza obstoječe aplikacije

Izvajalec bo najprej izvedel podrobno analizo obstoječe .NET 8 aplikacije in njenih odvisnosti. Cilj analize je, da se določi nujne pogoje za uspešno delovanje aplikacije v Docker okolju ter pripravi seznam vseh komponent, ki lahko vplivajo na ta postopek. Analiza mora obsegati naslednje vidike:

- arhitekturo aplikacije (uporaba API klicev, podatkovne baze, komunikacijski protokoli, …),
- delovanje zunanje programske opreme in knjižnic (Odvisnost knjižnic do operacijskega sistema, dostopa do datotečnega sistema, konfiguracijskih datotek ali dodatnih okvirov),
- potencialne omejitve Docker Windows okolja, ki bi lahko vplivale na funkcionalnost aplikacije.

Na podlagi analize mora izvajalec pripraviti osnovno strategijo za delovanje aplikacije v kontejnerskem okolju in identificirati potrebne korake za implementacijo. To vključuje uporabo podatkovnih virov, omrežnih konfiguracij in certifikatov. Posebno pozornost bo potrebno nameniti integraciji aplikacije z IIS (Internet Information Services), pri čemer mora analiza zajeti:

- vse ključne IIS nastavitve (npr. konfiguracijo gostovanja aplikacije, SSL certifikate in pravila za preusmeritev URL-jev),
- trenutno stanje in uporabo certifikatov (npr. za HTTPS povezave ali šifriranje komunikacije),
- sodelovanje IIS z ostalimi infrastrukturnimi komponentami, kot so požarni zid, omrežni prehodi ter DNS zapisi

Priprava Docker infrastrukture

Naslednji korak bo izdelava osnovne kontejnerske slike za aplikacijo ter implementacija konfiguracij, ki bodo omogočale, da aplikacija deluje v Docker okolju, v kolikor je to mogoče. Izvajalec mora izbrati Docker sliko, ki je primerna za .NET 8 aplikacijo. Predvidevamo uporabo uradnih Microsoftovih slik, kot so mcr.microsoft.com/dotnet/aspnet:8.0 za runtime okolje ali mcr.microsoft.com/dotnet/sdk:8.0 za razvojna okolja. Na osnovi te slike je potrebno ustvariti

konfiguracijsko datoteko Dockerfile, kjer se definirajo koraki za ustvarjanje kontejnerske slike, vključno z:

- kopiranjem datotek aplikacije v kontejner,
- nastavitvijo zaganjalne točke aplikacije,
- nastavitvijo ustreznih omrežnih vrat,
- prilagoditvijo konfiguracijskih nastavitev za pravilno delovanje aplikacije.

Izvajalec v Dockerfile vključi korake, ki so potrebni za podporo IIS ali podobnega orodja, kot so:

- konfiguracija HTTPS komunikacije in nalaganje certifikatov,
- prilagoditve za dostop do IIS dnevnikov ter spreminjanje nastavitev HTTPS.

Testiranje delovanja aplikacije v kontejnerskem okolju

Izvajalec mora vzpostaviti testno okolje za preverjanje skladnosti med delovanjem obstoječe aplikacije in kontejnerskim okoljem. Testiranje mora vključevati:

- preverjanje delovanja osnovnih funkcionalnosti aplikacije, kot so:
    - zagon aplikacije,
    - spremljanje upoštevanja konfiguracij,
    - komunikacija s podatkovno bazo in API storitvami,
    - dostop do uporabniškega vmesnika in izvajanje funkcij aplikacije,
- preverjanje komunikacije prek HTTP in HTTPS, vključno z uporabo certifikatov,
- merjenje porabe strojnih virov aplikacije v kontejnerju (CPU, RAM),
- simulacijo ponovnih zagonov kontejnerjev in aplikacije,
- testiranje delovanja aplikacije z zunanjo infrastrukturo (omrežni viri, zunanje API storitve).

Identifikacija težav in ocena izvedljivosti

Izvajalec bo analiziral rezultate testiranj in pripravil oceno izvedljivosti prenosa aplikacije v Docker okolje. Naročnik zahteva, da analiza odgovori na naslednja vprašanja:

V tej fazi bomo analizirali rezultate testiranja in ocenili, ali je migracija aplikacije v Docker okolje izvedljiva. Če se med testiranjem pojavijo težave, bomo identificirali njihove vzroke in v okviru projekta predlagali rešitve, če je to možno. Posebno pozornost bomo namenili:

- oceni delovanja aplikacije v primerljivih pogojih kot na trenutnem okolju,
- stabilnosti aplikacije pri ključnih funkcijah,
- uporabi virov in zahtevam za gostovanje aplikacije znotraj kontejnerja.

Poudarek v fazi analize izvedljivosti bo na preučevanju naslednjih ključnih vprašanj:

- Ali Docker okolje deluje z obstoječo .NET aplikacijo in ali omogoča ustrezno uporabo vseh funkcionalnosti aplikacije.
- Ali obstoječo .NET aplikacijo lahko uspešno zgeneriramo in zaženemo v Docker okolju, pri čemer bomo ovrednotili postopek gradnje in optimizacije za kontejnersko okolje.
- Ali aplikacija uporablja odvisnosti, knjižnice ali dele, ki v Docker Windows terminalnem okolju niso združljivi (npr. dostop do datotečnega sistema, uporaba specifičnih Windows funkcionalnosti znotraj IIS ali drugih storitev).
- Ali IIS pravilno deluje znotraj Docker Windows okolja in omogoča izvajanje vseh zahtev aplikacije (npr. preusmeritve, obdelava HTTPS, in hitrost obdelave zahtev).
- Kako je mogoče dostopati do ključnih Windows storitev znotraj kontejnerja, kot so certifikati, dostop do dnevnikov (logov), upravljanje konfiguracijskih datotek itd.
- Kakšne bodo spremembe pri operativni podpori aplikacije v Docker okolju, vključno z:
    - dostopom do dnevnikov v primeru težav,
    - postopkom ponovnega zagona aplikacije ali kontejnerja,
    - posodabljanjem aplikacije v kontejnerskem okolju,
    - implementacijo avtomatskih varnostnih kopij,
    - dostopom do konfiguracijskih datotek znotraj kontejnerja.

Poudarek naj bo na iskanju razlogov, zakaj nekatere funkcionalnosti morda ne delujejo, ter oceni, kako kompleksno bi bilo odpraviti te težave za končno implementacijo.

Spremljanje sprememb pri podpori aplikacij

Migracija aplikacije v Docker okolje bo prinesla nekatere pomembne spremembe pri podpori aplikacije. Te spremembe naj bodo vključene v končno poročilo, na podlagi katerih mora izvajalec oceniti vpliv kontejnerskega okolja na vsakodnevno podporo aplikaciji.

Ocena mora vključevati:

- Dostop do logov aplikacije v primeru težav in kako učinkovito je mogoče izvajati diagnozo napak znotraj kontejnerskega okolja.
- Možnost ponovnega zagona aplikacije in zagotavljanje stabilnosti po ponovnih zagonih.

- Izvajanje varnostnih kopij podatkov in konfiguracij.
- Posodabljanje aplikacije.
- Vpliv kontejnerskega okolja na BuildServer, kjer se generira aplikacija.
- Upravljanje z občutljivimi podatki (certifikati, konfiguracijske datoteke ipd.).

Priprava dokumentacije in priporočil

Glavni namen dokumentacije je zagotoviti jasna priporočila za nadaljnje korake implementacije dockerizirane aplikacije ter prepoznati morebitne izzive, ki bi jih bilo potrebno rešiti pred dokončno implementacijo. S poudarkom na ugotavljanju izvedljivosti projekt ne zagotavlja končne rešitve, ampak zagotavlja vpogled v tehnične in operativne zahteve za uspešno implementacijo, kar bo naročniku omogočilo informirano odločanje o prihodnjih korakih.

Izvajalec bo na podlagi analiz, testiranj in izvedenih aktivnosti pripravil celovito dokumentacijo, ki bo naročniku omogočala jasen vpogled v izvedljivost dockerizacije obstoječih .NET 8 aplikacij. Dokumentacija bo zajemala:

- povzetek vseh izvedenih aktivnosti in uporabljene testne postopke,
- podrobne ugotovitve glede delovanja aplikacije v Docker Windows okolju,
- seznam morebitnih neskladnosti ali zahteve za prilagoditve aplikacije,
- oceno tveganj in priporočila za nadaljnje tehnične rešitve,
- opis vseh zaznanih omejitev, ki lahko vplivajo na stabilnost ali izvedbo prenosa aplikacije.

Rezultat projekta

Rezultat pilotskega projekta bo obsežna in podrobna analiza izvedljivosti prehoda obstoječih .NET 8 aplikacij v Docker Windows okolje. Cilj analize je oceniti tehnično ustreznost in zmožnost aplikacije za delovanje v kontejnerskem okolju, pri čemer se identificirajo tako priložnosti kot možne omejitve. V primeru uspešnega testiranja bo rezultat projekta tudi pripravljena osnovna, delujoča kontejnerska slika aplikacije, ki bo lahko služila kot temelj za nadaljnje korake implementacije.

2. TESTIRANJE

Pri izvajalcu

Tehnološka prenova zahteva, da se kompletna simulacija balancing procesov in sicer tako, da se izvede in preveri delovanje vseh možnih tipov izmenjav dokumentov v vseh okoliščinah. Simulacija se izvede lokalno; odgovore platforme in/ali vhodne dokumente tržnih udeležencev se simulira lokalno. Ob simulaciji se poleg rezultatov preveri tudi ustrezno delovanje grafičnega vmesnika (ustrezen prikaz rezultatov).

Tehnološka prenova zahteva da se testirajo tudi komunikacije, hitrost in odzivnost na novo zastavljenega sistema na osnovi HTTP/2 komunikacij in .NET 8. Posebno pozornost se nameni situaciji, ko sistem več ne bo šel v restart in hkrati da bo ustrezno preračunal definiral urnike reportov, gatov in drugih procesov, ki so bili do sedaj vezani na restart sistema.

Pri naročniku

Poleg testiranja celotnega procesa z aspekta naročnika, je poudarek pri testiranju naročnika še na testiranju s testnimi platformami, kar pri izvajalcu ni možno. Velja tudi za druge morebitne integracije (SCADA ipd.), kar pri izvajalcu ni možno testirati.

Pri naročniku se tudi podrobno preveri, da se rezultat iz Avkcijske platforme (AP) ustrezno prenese v balancing sistem in s tem da se ustrezno ohranijo povezani procesi, ki temeljijo na rezultatu avkcij (prej izvedenih v balancing sistemu).