

# **Riigi infosüsteemi haldussüsteem RIHA**

## **Arhitektuuridokument**

**Riigi Infosüsteemide  
Arenduskeskus**

**Version: 0.4**

## Sisukord

Sisukord .....	2
1. Üldine arhitektuur .....	3
2. Tarkvaraline vaade .....	3
2.1. Alternatiivid .....	4
3. Sisuline vaade .....	4
3.1. Esitluskiht .....	4
3.2. Teenuskiht .....	4
3.3. Andmekiht .....	5
4. Funktsionaalne arhitektuur .....	5
4.1. Kasutajate autentimine ja autoriseerimine .....	5
4.1.1. Kasutajate autentimine .....	5
4.1.2. Kasutajate autoriseerimine .....	5
5. Andmearhitektuur .....	5
6. Asukoha vaade .....	5
7. Integratsioon teiste süsteemidega .....	6
8. Lisa 1 - Nõuded RIHA realisatsioonil kasutavale arendusraamistikule .....	7

## 1. Üldine arhitektuur

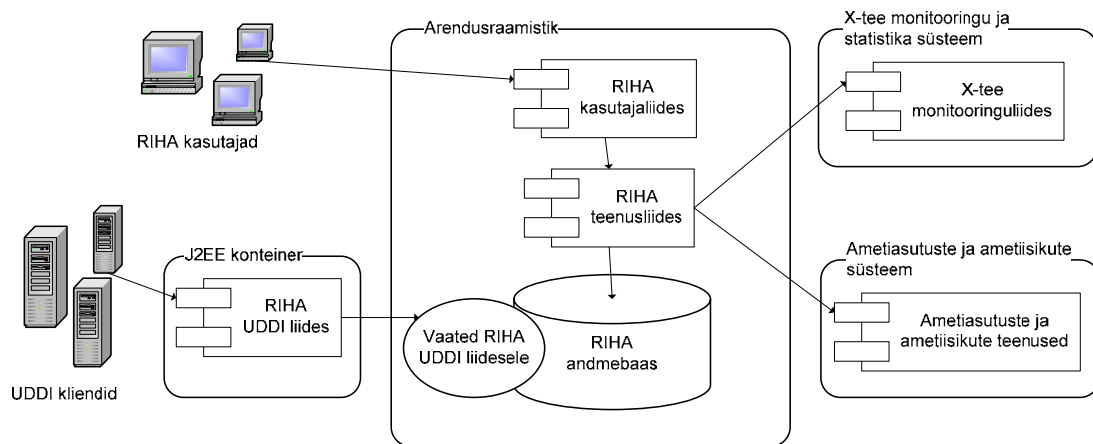
RIHA arhitektuur on loodud arvestades süsteemi otstarvet, kasutuse iseloomu ning organisatsioonilist keskkonda, kus see tööle hakkab. Valitud arhitektuur tagab RIHA võimalikult suure töökindluse, lihtsa edasiarendatavuse ning standardse ühilduvuse olemasolevate süsteemidega. RIHA põhiolemuseks on veebipõhine rakendus, mis seob endaga mitmeid väliseid teenuseid ning asub arendusraamistiku keskkonnas.

RIHA arhitektuuri määrab suuresti fikseeritav arendusraamistik.

Arhitektuur peab olema kooskõlas RIA IT-profiiliga.

RIHA peab olema kas realiseeritud riigiportaali arendusraamistikul või integreeritav riigiportaaliga, kasutades WSRP standardit.

## 2. Tarkvaraline vaade



Joonis 1 RIHA tarkvaraline vaade

Kasutajad pöörduvad RIHA poole peamiselt veebilehitseja abil. RIHA kasutajaliides peab olema teostatud mugava, atraktiivse ja efektiivsena, et süsteemile oleks tagatud laialdane kasutatavus.

ARR süsteemi pärandina on jätkuvalt toetatud *Universal Description, Discovery and Integration* (UDDI) liides, mis võimaldab standardsel moel sooritada RIHA poolt hallatud teenuste otsingut. Kuna aga UDDI ei ole mõeldud kasutamiseks rangete teenuste juurdepääsuõiguste piirangute tingimustes, siis ei ole võimalik UDDI liidest täiel määral kasutada ega ainult selle abil RIHA eesmärgiks seatud teenuseid pakkuda.

RIHA komponendid asuvad arendusraamistiku keskkonnas ning alluvad selle poolt kehtestatud reeglitele. Kasutatava arendusraamistiku funktsionaalsed ja mittefunktsionaalsed nõuded ja omadused on esitatud Lisas 1. Eraldi võib esile tuua RIHA kolmekihilise arhitektuuri osad.

- **Esitluskiht:** RIHA kasutajaliides, mis koosneb kasutajaliidese komponentidest ning vormide struktuuri komponentidest.
- **Teenuskiht:** RIHA teenusliides, mis koosneb lihtsatest RIHA toimeleogikat realiseerivatest teenustest. Teenused võivad pöörduda RIHA andmebaasi või üle X-tee välise süsteemide teenuste poole.
- **Andmekiht:** RIHA andmekiht koosneb tabelitest, kus hoitakse RIHA püsiandmeid ning vaadetest, mille abil neid andmeid vajadusel sobivale kujule teisendatakse.

RIHA UDDI liides on juba realiseeritud töötava ARR süsteemi raames Java rakendusena. Kuna UDDI laialdane kasutamine RIHA puhul ei ole otstarbekas, siis ei nähta ette olemasoleva liidese täiendamist. Erinevalt ARR süsteemist on RIHA UDDI liides kasutusel vaid andmete lugemiseks, mitte kirjutamiseks. Andmete kirjutamist UDDI abil ei saa lubada, sest kõik teenuste lisamise ja muutmise tegevused nõuavad vastavaid õigusi omava isiku otsustust ning kontrollmehhanisme, mida UDDI protokoll ei võimalda.

Väliste rakendustega suhtlemiseks kasutatakse X-tee teenuseid. Esimeses järgus saavad olema liidestatud X-tee monitooringu ja statistika ning Ametiasutuste ja ametiisikute süsteemid.

## **2.1. Alternatiivid**

Eelistatud on RIHA arhitektuur, mis põhineb Lisas 1 esitatud nõudeid toetaval riigiportaali arendusraamistikul.

RIHA arhitektuuri aluseks on kaalutud ka standardse J2EE arhitektuuri valimist, kuid olukorras kus RIA on oma süsteemid juba edukalt riigiportaali raamistikul välja arendanud, ei ole see põhjendatud. Olemasolevatest rakendustest erineva arhitektuuri kasutamine teeks RIA süsteemide haldamise põhjendamatult keerukamaks, andmata seejuures piisavalt lisaväärtust. Kõik süsteemile esitatavad mittefunktsionaalsed nõuded on täidetavad ka olemasoleva riigiportaali raamistiku baasil.

Töökindluse, hallatavuse ja arendatavuse tagamiseks eeldab J2EE arhitektuuri valik vähemalt Lisa 1 toodud võimalustega J2EE arendusraamistiku kasutuselevõttu. J2EE põhise arendusraamistiku kasutamisel migreeritakse olemasolev UDDI liides arendusraamistiku keskkonda. J2EE põhise arendusraamistiku kasutamisel peab RIHA rakendus olema kasutatav WSRP rakendusena (J2EE raamistik peab olema võimeline toimima WSRP-*produceri*'na).

## **3. Sisuline vaade**

Vaatleme süsteemi sisulist arhitektuuri rakenduse kihtide kaupa.

### **3.1. Esitluskiht**

Süsteemi kasutajaliides peab jätma kasutajale mulje ühtsest ja intuiitselt mõistetavast tervikust. Seda vaatamata asjaolule, et süsteem tegelikult seob endas mitmeid erinevaid süsteeme.

Süsteemi esitluskiht jaguneb vormide struktuuri komponentideks vastavalt analüüsidokumendis esitatud kasutajaliidese ekraanivormidele.

Lisaks vormide struktuuri komponentidele kasutatakse läbivalt arendusraamistiku kasutajaliidese komponente.

### **3.2. Teenuskiht**

Teenuskiht koosneb hulgast teenustest, mis realiseerivad RIHA loogikat. Iga teenus teostab konkreetsel kasutajaliidese vormil välja kutsutud tegevuse. Teenuste sisuks olev loogika on kirjeldatud vastavas kasutusloo kirjelduses. Teenused täidavad oma ülesandeid andmekihi ja väliste teenustega andmeid vahetades.

Teenuse olemuse paremaks mõistmiseks olgu antud järgmine teenuse definitsioon. Eraldi teenusena on defineeritud loogika, mis käivitub kasutajaliidese ekraanivormil algatatud sündmuse korral (nupulevajutus, vormivahetus, ...), mille tagajärjel on vaja lugeda või kirjutada andmeid andmekihti või välistesse süsteemidesse. Üks kasutajaliidese sündmus võib algatada ka mitu järjestikust või paralleelset teenuse väljakutset.

Teenustena ei ole realiseeritud kasutajaliidese sündmusi, mis ei vaja andmete vahetust andmebaasi ega väliste süsteemidega.

Näiteks vormi täitmisel igast sisestusväljast lahkumise sündmus enamasti ei põhjusta teenuse väljakutset. Vormi „Salvesta” nupu vajutus aga enamasti põhjustab teenuse väljakutse. Konkreetselt sõltub teenuse kasutamine või mitte kasutamine vastava kasutusloo poolt esitatud nõuetest.

### **3.3. Andmekiht**

Andmekihis asuvad RIHA püsiandmed vastavalt süsteemi andmemudelile.

## **4. Funktsionaalne arhitektuur**

Süsteem ei ole kasutuslugude kaupa jagatud alamsüsteemideks vaid vormideks ja teenusteks, mis on seotud andmemudeliga. Iga vormi jaoks on spetsifikatsioon disainidokumendis, mis koosneb ekraanivormi eskiisist, üldisest eesmärgi kirjeldusest, väljade seostest andmemudeliga ja vormil toimuda võivate sündmuste kirjeldustest.

### **4.1. Kasutajate autentimine ja autoriseerimine**

#### **4.1.1. Kasutajate autentimine**

Kasutajaid autentitakse RIHAsse sisenemisel arendusraamistiku kasutajate autentimise vahenditega. Samuti toimub kasutaja sessioonihaldus RIHA-s arendusraamistiku vastavate tehniliste lahenduste abil.

#### **4.1.2. Kasutajate autoriseerimine**

RIHA peaadministraator on määratud arendusraamistiku õigustesüsteemi abil. RIHA peaadministraatori ülesandeks on asutuste ja infosüsteemide loomisel nende esmaste kontaktisikute määramine. Samuti määrab RIHA peaadministraator statistikaameti, rahvusarhiivi ning andmekaitse inspeksiooni esindajad. RIHA kasutajad saavad asutuse või infosüsteemi piires halduri õigused, kui neid määratakse vastavateks kontaktisikuteks.

## **5. Andmearhitektuur**

RIHA andmeid hoitakse relatsioonilise andmebaasina. Välistele süsteemidest pärit andmeid ei kopeerita RIHA andmebaasi vaid kasutatakse otse välistest süsteemist.

## **6. Asukoha vaade**

RIHA töötab RIA baasinfrastruktuuri serveritel. RIHA kasutab arendusraamistikus olemasolevaid autentimise, autoriseerimise, X-tee kaudu suhtlemise ja kasutajaliidese

valmiskomponentide teenuseid. Riigiportaali arendusraamistiku kasutamisel asub ainsa erandina asub väljaspool arendusraamistiku keskkonda RIHA UDDI liides.

## 7. Integratsioon teiste süsteemidega

RIHA on väliste süsteemidega peamiselt seotud X-tee abil. Samas võib tulevikus tekkida uusi seoseid RIHA ja väliste süsteemidega, mis tingimata ei pea olema teostatud X-tee abil. Integratsioon väliste süsteemidega, nagu intsidendihalduse süsteem (infosüsteemi/teenuse *help tickets*) või ISKE tööriista süsteem (infosüsteemi/teenuse turvaklassid ja turbeastmed) võib olla teostatud ka mõne muu liidese kui X-tee abil, kui see osutub tehniliselt otstarbekaks. Tähtis on et RIHA säilitaks oma tervikliku ja lihtsa mulje kasutajale, kes RIHA kaudu teostab oma tegevusi ning kasutab selleks teiste süsteemide abi.

## 8. Lisa 1 - Nõuded RIHA realisatsioonil kasutavale arendusraamistikule

RIHA – realisatsioonil kasutatav arendusraamistik peab olema funktsionaalselt ja mittefunktsionaalselt omadustelt samaväärne riigiportaali arendusraamistikuga (<https://www.teine.eesti.ee/gfx/fw.doc>) ja omama alljärgnevat või samaväärseid võimalusi;

### **Kasutajaliidest kirjeldavate metaandmete interpreteerimise meetodil töötav deklaratiivne dünaamiline kasutajaliideste generaator (programmikoodi genereerimiseta)**

1. OLAP ja mitte-OLAP, statistiliste aruannete ja detailandmete väljavõtteid (sh risttabelitena) võimaldav aruannete generaator;
2. andmete otsimise, lisamise, muutmise ja kustutamise kasutajaliides:
  - a. otsing kõikide view või tabeli atribuutide järgi;
  - b. väljade valik otsingutulemusse;
  - c. otsingutulemusest kirje valik muutmiseks;
  - d. otsingu, muutmise, lisamise ja kustutamise reeglite kirjeldamine;
  - e. dünaamiliselt genereeritavad navigeerimisliipikud;
  - f. toetus erinevatele väljatüüpidele (sh failid);
  - g. muutmine kirje detailandmetel ja kirjete loetelus;
3. parameetrite salvestamise võimalus
4. veebiliidesed kõigi üldkasutatavate komponentide andmehalduseks;

### **Üldkasutatavad komponendid**

1. keskne staatiliste ja dünaamiliste klassifikaatorite haldus;
2. andmetabelite struktuurist sõltumatu keskne logimise funktsionaalsus ja veebiliidesed;
3. süsteemsete parameetrite monitooring;
4. integreeritud diagrammide moodul;
5. veebipõhiste administreerimisliidestega asünkroonsete (ühekordsed ja perioodilised) protsesside haldur;
6. autentimine (ID-kaart, Mobiil-ID, internetipangad, MISP legacy päring, BASIC-AUTH), OCSP-päring autentimisel;
7. autoriseerimine (veebipõhine kasutajaliides, ametikohapõhine autoriseerimine, delegeeritav kasutajate ja õiguste haldus);
8. tarkvara versioonihaldus (automaatne tarkvaramuudatuste monitooring);
9. funktsioonide ligipääsuõiguste administreerimine;
10. integratsioon Oracle andmebaasidega;
11. sessioonihaldus, veebiliides: aktiivsed sessioonid, kasutaja sessiooni ülevõtmine, mitmed turvalisuse tagamise meetmed lisaks tavapärastele, sessiooni veebipäringute logimine;
12. X-tee päringute ja adapterite loomise abifunktsioonid;
13. round-robin andmebaaside toetus;
14. keskkondade töörežiimid:
  - a. arenduskeskkond vs toodangukeskkond;
  - b. testkeskkond vs mitte-testkeskkond

- c. tavaline (brauser -> Apache (pordid 80, 443)) vs puhverdatud-skaleeritud (brauser -> 443:pound (SSL terminaator) -> 80:squid (veebikiirendi) -> 80:apache) töörežiim.
15. taimer töökiiruse optimeerimiseks (süsteemi ja veebileidete tasemel);
16. sisuhaldusvahend;
17. WSRP standardi toetus.

#### **Üldkasutatavad teenused (ilma süsteemivälise integreerimise vajaduseta)**

1. e-kirjade saatmine (sh S/MIME);
2. teavitusteenused (sh mobiilne teavitust);
3. digiallkirjastamist toetavad funktsioonid;
4. DVK andmevahetust toetav funktsionaalsus;
5. e-vormide põhine menetlusteenus.

#### **Kõigi rakenduste visuaalset ja käitumismudeli ühtsust tagavad komponendid**

1. accessibility: mitmekeelsuse tugi, printvaade, teksti suurus;
2. ühine dünaamiline metaandmetel põhinev menüüsüsteem (vasak- ja parempoolne menüü, Ajax tehnoloogial menüüpunktide avamine ja sulgemine); reeglite kirjeldamine ligipääsuks menüüdele;
3. üldkasutatavad veebielementide esitamise funktsioonid;
4. enamkasutatavad kasutajaliidete komponendid: tabelid, hierarhiad, lipikmenüüd.

#### **Protsessid ja neid toetav tarkvara**

1. 4 keskkonda (raamistiku arenduskeskkond -> arenduskeskkond -> testkeskkond -> toodangukeskkond); tarkvara automaatseks keskkondadevaheliseks tarkvara ja andmete erinevuse tuvastamiseks ja erinevusi kõrvaldavate skriptide loomiseks ja käivitamiseks;
2. automatiseeritud andmete ja tarkvara varundamine;
3. protseduurid suuremahuliste muudatuste teostamiseks ja süsteemse tarkvara vahetamiseks;
4. avaliku keskkonna hot-standby ja skaleeritud rakenduskiht.