

Tehniline kirjeldus

1. Mõisted ja lühendid

Mõiste / Lühend	Kirjeldus
ADR	Arhitektuuriline otsus (Architecture Decision Record)
Consul	HashiCorp Consul – teenuste registri ja teenuse avastamise tarkvara; käesolevas töös hinnatav alternatiiv Istiole
DoD	Valmisoleku kriteeriumid (ingl k <i>Definition of Done</i>) – Tellija ja Täitja vahel kokku lepitud tingimuste kogum, mille täitmisel loetakse töö edukalt lõpetatuks ja vastuvõetuks
EBS	Majandustarkvara Oracle E-Business Suite
Edasimineku peatamise otsus või	Ingl k <i>Go/No-Go decision</i> – projekti vaheetapi lõpus tehtav otsus, kas projekt on järgmise etapi käivitamiseks valmis (edasi minna) või vajab parandusi enne edasiliikumist (peatada)
GAP-analüüs	Lahknevusanalüüs, milles võrreldakse olemasolevat SKA infosüsteemi arhitektuuri tulevikuvisioniga (TO-BE), et selgitada välja ja prioriseerida lahknevuste (GAP) likvideerimiseks vajalikud tööd
Istio	Avatud lähtekoodiga (Apache License 2.0) teenuse võrgu ja teenuse avastamise platvorm; käesolevas töös eelistatud tehnoloogiavalik
Koormuse jaotamine (load balancing)	Päringute jaotamine teenuse mitme eksemplari vahel
Kubernetes (K8s)	Konteineritöökoormuste orkestreerimise platvorm
mTLS	Vastastikune TLS (<i>mutual TLS</i>) – kahepoolne krüptograafiline autentimine teenuste vahel, mis annab igale teenusele tagatud identiteedi
Pilootkasutuselevõtt	Valitud teenuste piiratud ulatusega kasutuselevõtt, mis valideerib pakutud lähenemist praktikas enne täismahulist kasutuselevõttu
SKA	Sotsiaalkindlustusamet
SKAIS1	Sotsiaalkaitse infosüsteem (vananenud platvormil)
SKAIS2	Sotsiaalkaitse infosüsteem uus rakendus (hõlmab Ametnikurakendust ja Iseteenindust)
Spring Cloud	Spring-põhiste rakenduste hajussüsteemi mustreid (sh teenuse avastamine ja konfiguratsioon) toetav teek
Teenuse avastamine (Service Discovery)	Muster ja mehhanism, mille abil teenused leiavad üksteise otspunktid keskse registri kaudu, ilma et otspunkte tuleks igasse sõltuvasse teenusesse käsitsi konfigureerida
Teenuse võrk (Service Mesh)	Teenustevahelise suhtluse haldamise infrastruktuurikiht (turvalisus, marsruutimine, töökindlus ja jälgitavus), mis on teenuse avastamise loomulik edasiarendus

2. Ülevaade

Sotsiaalkindlustusamet (SKA) kasutab ülesannete täitmiseks sotsiaalkaitse infosüsteemi, mis koosneb SKAIS1, EBS ja SKAIS2 rakendustest. Tegemist on infosüsteemiga, mida on arendatud hulk aastaid nii, et uute

funktsionaalsuste loomise kõrval ei ole jätkunud ressursi tehnoloogilise ajakohasuse tagamiseks. Süsteemis tekkinud oluliste kitsaskohtade leevendamiseks käivitati 2025. aastal SKA infosüsteemi arhitektuuri tulevikuvisioni (TO-BE lahenduse) väljatöötamise projekt, mille käigus loodi arhitektuuri reeglistik, millele peavad vastama teostatavad arendused ning migreerimisplaan. Ühtlasi kaardistati probleemid ja eesmärgid, teostati GAP-analüüs, st võrreldi olemasolevat lahendust TO-BE arhitektuuriga, et selgitada välja ja prioriseerida GAP likvideerimiseks vajalikud tööd. Lisaks oli juba varasemast teada kitsaskohti (nt seotud teenustevahelise suhtlusega), mis vajavad lahendamist.

Käesolev leping koondab SKAIS2 teenustevahelise suhtluse arhitektuuriliste tööde eeltööd, et tagada süsteemi operatiivne töökindlus ja skaleeritavus teenuste arvu kasvades. Selle lepingu tööde tulem võimaldab edasi minna laiaulatuslikuma teenuse avastamise kasutuselevõtu ja teenuse võrgu töödega. Käesoleva lepingu skoobis on teenuse avastamise arhitektuuri ja tehnoloogiavaliku kirjeldus (ADR-ide kujul) ning valitud teenustega pilootkasutuselevõtt, mis valideerib pakutud lähenemist praktikas (nii Kubernetese kui virtuaalmasinate keskkonnas) ja kalibreerib edasiste etappide töömahu hinnangud.

2.1. GAP 2.2. Teenuse avastamine (Service Discovery)

SKAIS on välja arenenud monoliitsest infosüsteemist, kus teenustevahelist suhtlust praktiliselt ei olnud. Eraldiseisvate teenuste arvu kasvades hakati igale teenusele käsitsi sõltuvate teenuste otspunkte konfigureerima ja tänaseks on dubleeritud otspunktide konfiguratsioon kasvanud suureks. GAP 2.2 töö eesmärk on võtta kasutusele keskne teenuste register, mis võimaldab teenustel üksteist leida ilma otspunkte igasse teenusesse käsitsi konfigureerimata ja toimib nii Kuberneteses kui virtuaalmasinates, säilitades samal ajal selge arengutee kõikehõlmava teenuse võrgu suunas.

GAP-analüüsi raames on koostatud teenuse avastamise arhitektuuriettepanek (vt lisainfo allpool), mis annab tervikpildi probleemist, peamistest riskidest, soovituslikust tehnoloogiavalikust ja edasisest arengusuunast. Ettepaneku peamised järeldused:

1. Dubleeritud ja käsitsi hooldatav otspunktide konfiguratsioon muudab süsteemi operatiivselt hapraks: see soodustab süsteemi tööd häirivaid inimlikke vigu, ei võimalda teenuseid skaleerida nii, et lisatud eksemplarid muutuksid sõltuvatele teenustele kättesaadavaks, ega paku kesket koormuse jaotamise võimalust.
2. Lahenduseks on keskne teenuste register, mis peab olema piisavalt paindlik, et pakkuda teenuse avastamise võimekust nii virtuaalmasinates kui Kuberneteses käivitatavatele teenustele.
3. Tehnoloogiavaliku puhul on praegu parim valik Istio, võimaliku alternatiivina HashiCorp Consul. Istio kasuks räägivad avatud ja tingimusteta litsents (Apache License 2.0), de facto standardi seisund teenuse avastamise ja teenuse võrgu pakkujana, tugi Spring-põhiste rakenduste vähete koodimuudatustega, iga teenuse krüptograafiliselt tagatud identiteet (mTLS ja asukohast sõltumatu adresseerimine) ning selge arengutee kõikehõlmava teenuse võrgu suunas.
4. Peamised puudused on oma olemuselt ajutised ja seotud Kubernetese ja virtuaalmasinate keskkondade ühildamisega: esialgne ülesseadmise keerukus, suurem esialgne õppimisvajadus, hilisema tehnoloogiavahetuse kulukus ja vajadus eraldi lahenduse järele kohalikus arenduskeskkonnas. Mida suuremas ulatuses SKAIS Kubernetese platvormile üle läheb, seda väiksemat mõju need puudused avaldavad.
5. Teenuse avastamise kasutuselevõtul peab säilima võimalus käivitada rakendusi ka ilma teenuse avastamise funktsionaalsuseta (oluline nii kohaliku arenduskeskkonna kui tagasiühilduvuse seisukohast). Teenuse registreerimist saab juhtida Spring Cloud parameetritega

(spring.cloud.discovery.enabled) ja klientide puhul saab kasutada varulahendust, mis pöörub otspunkti URL-i poole teenuse registrinime asemel.

Käesoleva lepingu skoobist jäävad välja teenuse avastamise täismahuline kasutuselevõtt kõikidele SKAIS2 teenustele, täieliku teenuse võrgu funktsionaalsuse realiseerimine (täpsem marsruutimine, töökindlusmustrid ja kõikehõlmav teenustevaheliste päringute turvapoliitika) ning SKAIS2 üleminek Kubernetese platvormile. Need kuuluvad GAP 3.3 (Teenuse võrk) ja GAP 3.1 (Kubernetes) skoopi ning tellitakse eraldi lepingutega käesoleva lepingu väljundite alusel.

2.2. Sõltuvus paralleelsete GAP-töödega

GAP 2.2 on eeltingimuseks GAP 3.3 (Teenuse võrk / Service Mesh) tööde sisukokkuleppele: SKAIS Arhitektuur TO-BE lehel on GAP 3.3 sõnastatud nii, et see tugineb GAP 2.2 raames loodud teenuse avastamise võimekusele ja laiendab Istio kasutust kõikehõlmava teenuse võrgu mustriks. Käesoleva lepingu väljundid (kinnitatud ADR-id, pilootkasutuselevõtu tulemid ja edasimineku või peatamise otsus) on sisendiks GAP 3.3 ja teenuse avastamise täismahulise kasutuselevõtu kava koostamisele.

- SKAIS Arhitektuur TO-BE, GAP 2.2 Teenuse avastamine (Service Discovery) arhitektuuriettepanek:
 - SKAIS2 wiki (<https://wiki.sm.ee/spaces/SKAIS2/>) → SKAIS Arhitektuur TO-BE → GAP analüüsit tulenevad tööd → GAP projektid → 2.2. Teenuse avastamine (Service Discovery)
- Seotud GAP-projektid SKAIS Arhitektuur TO-BE lehel:
 - 3.3 Teenuse võrk (Service Mesh) (tugineb GAP 2.2 tulemitel)
 - 3.1 Kubernetes (Istio on Kubernetese-põhine tehnoloogia; teenuse avastamine peab toimima ka virtuaalmasinates)
 - 2.1 API lüüsi loomine (väline liiklus, erinevalt sisemisest teenustevahelisest avastamisest)
 - 3.2 Tarneprotsessi moderniseerimine (paigaldus ja konfiguratsioonihaldus)
 - 2.3 Jälgitavus (teenustevaheliste päringute seire)

3. Töö eesmärk

Töö eesmärk on luua SKAIS2 teenuse avastamise kasutamise põhimõtted ja neid toetavad arhitektuurilised otsused (ADR-id), sh kinnitada keske teenuste registri tehnoloogiavalik. Lähtudes GAP-analüüsi raames koostatud teenuse avastamise arhitektuuriettepanekust, valideeritakse loodud põhimõtteid valitud teenuste pilootkasutuselevõtu kaudu nii Kubernetese kui virtuaalmasinate keskkonnas. Töö tulemina esitatakse järgmiste tööde (GAP 3.3 ja teenuse avastamise täismahuline kasutuselevõtt) skoop ja sellest tulenev mahuhinnang.

Pilootkasutuselevõtu lähtekood ja konfiguratsioon loovutatakse Tellijale tootmiskvaliteediga kujul. Tellija (TEHIK või Tellija valitud kolmanda osapoole partner) paigaldab need toodangusse omal valitud ajal ja viisil.

4. Tellitavad tööd

Lepinguga tellitakse SKAIS2 teenuse avastamise kasutuselevõtu põhimõtted ja nende valideerimine valitud mikroteenustega vähemalt järgmises ulatuses:

- Teenuse avastamise arhitektuuri ja tehnoloogiavaliku otsuste lukustamine ADR-idenäiteks:

- keskse teenuste registri tehnoloogiavalik (Istio eelistatud, HashiCorp Consul alternatiivina) koos valiku põhjenduse ja hindamiskriteeriumidega;
 - teenuse avastamise toimimine hübriidkeskkonnas (Kubernetes ja virtuaalmasinad) ning sellega seotud halduslahendus;
 - teenuste krüptograafilise identiteedi (mTLS) ning teenusepõhise, asukohast sõltumatu adresseerimise mudel;
 - klientide adresseerimismudel ja varulahendus (teenuse registrinimi või otspunkti URL);
 - kohaliku arenduskeskkonna ja tagasiühilduvuse mudel (teenuse avastamise väljalülitatavus, `spring.cloud.discovery.enabled`);
 - koormuse jaotamise ning teenuse mitme eksemplari adresseerimise põhimõtted.
- Keskse teenuste registri pilootkeskkonna ülesseadmine ja valitud SKAIS2 mikroteenuste liidestamine teenuse avastamisega. Teenused valitakse Tellija ja arhitektuuripartnerite ühisarutelu tulemusena selliselt, et kaetud oleks nii Kuberneteses kui virtuaalmasinas käivitatav teenus ja vähemalt üks tegelik teenustevaheline päring (kliendi ja teenuse vahel).
 - Olemasolevate teenuste väheste koodimuudatustega liidestamise valideerimine ning dubleeritud otspunktide konfiguratsiooni asendamine teenuse avastamisega piloodi ulatuses.
 - Kohaliku arenduskeskkonna lahendus, mis võimaldab rakendusi käivitada ka ilma teenuse avastamise funktsionaalsuseta (varulahendus), ning tagasipööramise strateegia teenuse avastamiselt senisele konfiguratsioonipõhisele adresseerimisele.
 - Taaskasutatav teenuse avastamisega liidestamise muster ja juhend (sh konfiguratsioon ja Helm-chartide muudatused), mis on aluseks järgnevatele teenuste liidestamise lainetele ja GAP 3.3 (Teenuse võrk) tööle.
 - Testimisstrateegia ja valmisoleku kriteeriumid (DoD) iga pilootteenuse jaoks, sh vastutuste jaotus testimisel, nõutavate automaatsete minimaalne kogum ja kvaliteedikriteeriumid piloodi vastuvõtuks.
 - Pilootkasutuselevõtu tulemuste tagastusring koos töömahu kalibreerimisega ning edasimineku või peatamise otsus järgmise faasi (täismahuline kasutuselevõtt ja GAP 3.3) käivitamiseks.
 - Muud arendusvajadused, mis selguvad analüüsi käigus ja on olulised realiseerida käesoleva lepingu skoobis.
 - Tööde loetelu tekib eelnevalt kokkulepitud backlogi alusel ja hallatakse Jiras.

5. Tööprotsess ja tulemid

Järgnevalt täpsustatakse Täitjaga sõlmitud lepingus kirjeldatud tingimusi tööprotsessi, dokumenteerimise, töökorralduse, testimise ja mittefunktsionaalsete nõuete osas.

5.1. Tööprotsess

- Tööraamistik peab tuginema etapiviisilisele planeerimisele, võimaldades arendusmeeskonna prioriteete jooksvalt muuta;
- Tööraamistik peab võimaldama süsteemset ülevaadet iga kokkulepitud töö saavutustest;
- Tööraamistik peab võimaldama arendusmeeskonna tulemuslikkuse (kasvu) jälgimist ajas;

- Tööraamistik peab tagama, et arendusmeeskond tegeleb omal initsiatiivil enda tulemuslikkuse parandamisega ja parendusettepanekute esitamisega.
- Täpsem protsess lepitakse kokku tööde käigus.

5.2. Dokumentatsioon

Vastavalt tehnilise kirjelduse punktis 4 (tellitavad tööd) tuleb töö käigus luua ja täiendada SKAIS2 dokumentatsiooni, milleks on minimaalselt:

- Funktsionaalsuse ja äriprotsesside kirjeldused kokkulepitud kujul (nt BPMN standard, diagrammid ja kirjeldused Confluence'is) koos ärireeglitega, mida peab olema hiljem võimalik täiendada järgnevates etappides lisanduva funktsionaalsuse alusel ning siduda tervikliku protsessi- ja ärireeglite kirjeldusega.
- Tehniline dokumentatsioon (nt paigaldusjuhendid, Confluence keskkonnas).
- Andmemudel.
- Kasutusjuhendid.
- Dokumentatsiooni täpsed liigid ja sisu formaat lepitakse kokku tööde käigus, kuid see peab vastama minimaalselt järgnevatele nõuetele: [Nõuded arendusprojektide dokumentatsioonile SKAISi näitel - SKAIS2 - Sotsiaalministeeriumi wiki](#)

5.3. Töökorraldus

Projekti töökorraldus toetub raamlepingu kodukorrale (lisa 5) ja töökorralduslikele kokkulepetele. Täitja peab tagama piisava kaasamise ja ülevaadete andmise Tellijale.

5.4. Tööde üleandmine

Arendustööde lähtekoodi tarne toimub pideva integreerimise (continuous integration) teel. Tööde järjekord, üleandmise aeg ja UAT tagasiside andmise aeg lepitakse kokku tööde käigus taasesitamist võimaldavas vormis (tellija ja täitja poolt).

Töö üleandmisel kontrollib tellija üle töö nõuetekohasuse ning annab vajadusel tähtaja töö parandamiseks.

Täitja teab, et tellija võib igat üle antud funktsionaalsust paigaldada toodangusse ning täitja peab tagama, et lõplikult valmimata funktsiooni protsessid on võimalik toodangust välja lülitada (feature flags). Funktsioonide sisse-välja lülitamise võimekus lepitakse tellija ja täitja vahel kokku tööde käigus.

5.5. Nõuded automaattestimisele

Automaattestimise nõuete täitmisel tuleb lähtuda nõuetest, mis on leitavad siit: [Automaattestide nõuded](#).

Testidega kaetakse serveripoolne ja kasutajaliidese kood. Kaetavuse kontrolliks ja raporteerimiseks seatakse tellija koodihoidla keskkonnas (Gitlab) üles pideva integratsiooni töövoog, mis seadistatakse koostoimima SonarQube koodi ja testide kaetavuse analüsaatoriga.

Automaattestid peavad üle antavas lähtekoodis continuous-integration'i töövoos edukalt läbima. Samuti peab täitja hoolt kandma, et varasemalt loodud testid jäävad töö lõpetamisel edukalt läbitavaks. Nõuete muutumisel peab täitja viima testid vastavusse uute nõuetega või neid eemaldama, kui selliseid nõudeid enam ei eksisteeri.

Teenuse avastamisega liidestatud pilootteenus peab edukalt läbima ka olemasolevad süsteemiülesed regressiooni- ja suitsutestid, tõendamaks, et üleminek teenuse avastamisele ei ole rikkunud teenuse väliseid API-lepinguid ega teenuste vahelist suhtlust.

5.6. Koormustestimised

Iseteenindus peab toetama paralleelset minimaalselt 200 päringut (siinkohal on oodatav arendajapoolne hinnang, kas see on piisav päringute maht, vajadusel tõsta arvu).

API teenustele ning keskmine päringu teenindamise aeg ei tohi olla suurem kui 2 sekundit.

Nõude täitmise kontrolliks tuleb luua koormustestid. Koormustestide tegemine lepitakse tellija ja täitja poolt kokku tööde käigus.

Nõudest eraldi tuleb vaadelda staatiliste failide allalaadimist – need tuleb http serveris puhverdada.

5.7. Mittefunktsionaalsed nõuded

Mittefunktsionaalsete nõuete täitmisel tuleb lähtuda järgnevast:

- Uue funktsionaalsuse loomisel tuleb lähtuda raamlepingu (nr 3-9/3443-1) ja selle lisades kirjeldatud nõuetest
- Täpsemad mittefunktsionaalsed nõuded, millest lähtuda on leitavad siit: [Mittefunktsionaalsed nõuded](#)

5.8. Tulemid

Valminud ja üleantavate tööde tulemid peavad sisaldama:

- Teenuse avastamise arhitektuuri ja tehnoloogiaavaliku ADR-e (sh Istio ja Consul'i vahelise valiku põhjendus);
- Pilootkasutuselevõtu käigus tuvastatud täiendavate otsustuskohtade kaardistust;
- Keskse teenuste registri pilootkeskkonna konfiguratsiooni ning pilootteenuste teenuse avastamisega liidestatud tootmiskvaliteediga lähtekoodi ja konfiguratsiooni (sh Helm-chartide muudatused);
- Taaskasutatavat teenuse avastamisega liidestamise mustrit ja juhendit (sh kohalik arenduskeskkond ja varulahendus);
- Töömahu kalibratsioonidokumenti ja tagastusringi protokollid koos edasimineku või peatamise otsusega;
- Viiteid dokumentatsioonile Confluence'is;
- Viiteid tehtud töödele (Jira piletitena);
- Viiteid koodile GitLabis;

5.9. Tööde teostamise tähtaeg

Tööde üleandmise (so Jira piletid "Ready for AT" ja toodangu keskkonda paigaldatav kvaliteetne lähtekood) tähtaeg on **30.09.2026**. Tööde üleandmisele järgneb tellija poolne tööde testimine mõistliku aja jooksul ja

vajadusel täitja poolne paranduste või täienduste tegemine, kui tööd ei ole tähtajaks teostatud nõuetekohaselt või tellijal tekib vajadus väiksemamahuliste täienduste koheseks realiseerimiseks.

Tööd loetakse teostatuks tööde üleandmise-vastuvõtmise akti tellija poolt allkirjastamise hetkest.

Tellijal on õigus pikendada tööde üleandmise tähtaega, kui ilmneb tööde ajakavas täitjast sõltumatuid olulisi viivitusi.