

Asia: VN/5386/2020-VM-152

Lausuntopyyntö ehdotuksesta julkisen hallinnon API-periaatteiksi

1.1 Tarjoa ja hyödynnä tietoja pääsääntöisesti ohjelmointirajapintojen kautta

Onko periaate realistinen ja käyttöönotettavissa organisaatiossasi? Miltä tukimateriaali ja esimerkit vaikuttavat? Periaatteet, tukimateriaali ja esimerkit on kuvattu liitteissä tarkemmin.

-

1.2 Määritä ohjelmointirajapintojen tarjoamiselle ja hyödyntämiselle tavoitteet ja hanki riittävät resurssit

Onko periaate realistinen ja käyttöönotettavissa organisaatiossasi? Miltä tukimateriaali ja esimerkit vaikuttavat? Periaatteet, tukimateriaali ja esimerkit on kuvattu liitteissä tarkemmin.

-

1.3 Varmista hankinnoissa yhteentoimivuus muiden tietojärjestelmien kanssa

Onko periaate realistinen ja käyttöönotettavissa organisaatiossasi? Miltä tukimateriaali ja esimerkit vaikuttavat? Periaatteet, tukimateriaali ja esimerkit on kuvattu liitteissä tarkemmin.

-

1.4 Edistä sisäistä ja ulkoista yhteistyötä

Onko periaate realistinen ja käyttöönotettavissa organisaatiossasi? Miltä tukimateriaali ja esimerkit vaikuttavat? Periaatteet, tukimateriaali ja esimerkit on kuvattu liitteissä tarkemmin.

-

2.1 Kehitä ohjelmointirajapintoja tarvelähtöisesti

Onko periaate realistinen ja käyttöönotettavissa organisaatiossasi? Miltä tukimateriaali ja esimerkit vaikuttavat? Periaatteet, tukimateriaali ja esimerkit on kuvattu liitteissä tarkemmin.

-

2.2 Määritä ohjelmointirajapintojen tarjoamiseen ja hyödyntämiseen liittyvät roolit, tehtävät, vastuut ja toimintamallit.

Onko periaate realistinen ja käyttöönotettavissa organisaatiossasi? Miltä tukimateriaali ja esimerkit vaikuttavat? Periaatteet, tukimateriaali ja esimerkit on kuvattu liitteissä tarkemmin.

-

2.3 Kuvaa ohjelmointirajapintojen muodostama kokonaisuus

Onko periaate realistinen ja käyttöönotettavissa organisaatiossasi? Miltä tukimateriaali ja esimerkit vaikuttavat? Periaatteet, tukimateriaali ja esimerkit on kuvattu liitteissä tarkemmin.

-

2.4 Tunnista ja hallitse ohjelmointirajapintoihin liittyvät riskit

Onko periaate realistinen ja käyttöönotettavissa organisaatiossasi? Miltä tukimateriaali ja esimerkit vaikuttavat? Periaatteet, tukimateriaali ja esimerkit on kuvattu liitteissä tarkemmin.

-

3.1 Kehitä ohjelmointirajapinnat avoimilla ja teknologiariippumattomilla standardeilla ja protokollilla

Onko periaate realistinen ja käyttöönotettavissa organisaatiossasi? Miltä tukimateriaali ja esimerkit vaikuttavat? Periaatteet, tukimateriaali ja esimerkit on kuvattu liitteissä tarkemmin.

-

3.2 Kuvaa ohjelmointirajapintojen käsittelemät tiedot yhteisten ja yleisten tietomallien mukaisesti

Onko periaate realistinen ja käyttöönotettavissa organisaatiossasi? Miltä tukimateriaali ja esimerkit vaikuttavat? Periaatteet, tukimateriaali ja esimerkit on kuvattu liitteissä tarkemmin.

-

3.3 Testaa, versio, dokumentoi ja julkaise ohjelmointirajapinnat

Onko periaate realistinen ja käyttöönotettavissa organisaatiossasi? Miltä tukimateriaali ja esimerkit vaikuttavat? Periaatteet, tukimateriaali ja esimerkit on kuvattu liitteissä tarkemmin.

-

3.4 Seuraa ohjelmointirajapinnoille asetettuja mittareita ja muita ominaisuuksia

Onko periaate realistinen ja käyttöönotettavissa organisaatiossasi? Miltä tukimateriaali ja esimerkit vaikuttavat? Periaatteet, tukimateriaali ja esimerkit on kuvattu liitteissä tarkemmin.

-

Lausunnonantajan lausunto

Voitte kirjoittaa lausuntonne alla olevaan tekstikenttään

Terveyden ja hyvinvoinnin laitos (THL) kiittää TiHA-hanketta tehdystä työstä ja mahdollisuudesta lausua julkisen hallinnon API-periaatteista. Yleisesti ottaen periaatteet sekä niihin liittyvä tukimateriaali ovat hyvin laadittuja ja edistävät niille asetettuja tavoitteita.

Periaatteiden jako (kohta 3) strategiseen, taktiseen ja operatiiviseen tasoon on toimiva. Samoin kohdassa 2.3 tehty API:en tyypitys sisäisiin ja ulkoisiin (kumppani ja julkinen) on looginen.

Ohjelmointirajapinnan eli API:n määritelmä (kohta 2.1) on laava, mikä on perusteltua periaatteiden laajemman sovellettavuuden kannalta. Perustelua tukee myös määritelmän linkitys tiedonhallintalain ja EU:n julkaisun termeihin. Termiin ”API” yhdistetään kuitenkin usein suppeampi määritelmä: reaaliaikaisesti kutsuttava, avoimia internet-protokollia hyödyntävä web-pohjainen rajapinta. API-termin määrittely näin voi siis aiheuttaa väärinkäsityksiä. Ohjelmointirajapinnoille tyypillistä on toistuva käyttö, joka voisi myös näkyä määritelmässä.

Usein samaa tietoa käsitellään tiedon elinkaaren aikana useilla erilaisilla ja erimuotoisilla rajapinnoilla. Esimerkiksi tietoturvan määritelmän sitominen ”ohjelmointirajapinnan elinkaareen” on harhaanjohtavaa, kun tietoturvallisuus käytännössä ja myös tietoturvallisuuden yleisten periaatteiden mukaisesti on huomioitava tiedon koko elinkaareessa, riippumatta siitä millä rajapinnoilla tietoja siirretään tai käsitellään. Elinkaariaasia on sinällään hyvin kuvattu luvussa 2.4, mutta määritelmien sitominen tulisi tehdä tiedon eikä rajapinnan elinkaareen.

Kohdan 1.3 viimeinen rajausta on tärkeä: toimialakohtaisia ohjeita tai määrittelyjä voidaan toimialoittain kehittää periaatteiden päälle. Tämä on jopa suositeltavaa, jotta voidaan hyödyntää kunkin toimialan parhaat käytännöt ja esimerkiksi toimialakohtaiset semanttista yhteentoimivuutta edesauttavat standardit.

Periaate 1.1 Tarjoa ja hyödynnä tietoja pääsääntöisesti ohjelmointirajapintojen kautta on hieman epäselvä. Varmaankin tarkoitus on, että suurin osa tarjottavista ja hyödynnettävistä tiedoista liikkuisi ohjelmointirajapintojen kautta. Kohdan voi kuitenkin tulkita myös niin, että ohjelmointirajapintojen pitäisi olla ensisijainen kanava. Useimmilla toimijoilla ensisijainen tietojen tarjoaminen ja hyödyntäminen tapahtuu kuitenkin muulla tavoin kuin ohjelmointirajapintojen kautta. Esimerkiksi tietojen tarjoamisen pääsääntöisiä kanavia voivat olla julkaistut raportit, web-sivustot, tietojen loppukäyttäjille tarjottavat näkymät tai sovellukset, jolloin ohjelmointirajapinnat eivät ole välttämättä pääsääntöinen kanava.

Periaate 1.3 Varmista hankinnoissa yhteentoimivuus muiden tietojärjestelmien kanssa on tärkeä, sitä kannattaa ja pitäisi entisestään korostaa. Alakohtaksi olisi syytä lisätä ”Hyödynnä valmiita malleja ja standardeja”. Tämä sopii strategiselle tasolle, koska auttaa varmistamaan, että resurssien käyttö on tehokasta.

Periaate 1.4 Edistä sisäistä ja ulkoista yhteistyötä on tavallaan itsestäänselvyys ja kuuluu jokaiseen hyvin toimivaan organisaatioon. Tämän voisi jättää pois muiden strategisten periaatteiden korostamiseksi. Jos asia halutaan pitää mukana, sen voisi sisällyttää lyhennettynä periaatteeseen 1.1.

Periaate 2.3 Kuvaa ohjelmointirajapintojen muodostama kokonaisuus voisi laajentaa ja nimetä uudelleen, esimerkiksi ”Suunnittele tietoarkkitehtuurin tavoitetila siten, että se perustuu ohjelmointirajapintojen käyttämiseen”. Ohjelmointirajapintoja ei kannata ottaa käyttöön sellaisinaan nykyarkkitehtuurin lisäosina vaan kokonaisuus tulisi suunnitella ensin. Lisäksi on hyvä miettiä, mitä lisäarvoa API:t tuovat liiketoiminnalle. Niitä voisi hyödyntää esimerkiksi tiedolla johtamisessa ja tiedon keräämisessä BI:n kautta. Ohjelmointirajapinnoista kuvattavien asioiden listaan tulisi sisällyttää: yksityiskohtainen rajapintamäärittely tai linkki siihen.

Periaate 2.4 Tunnista ja hallitse ohjelmointirajapintoihin liittyvät riskit on tärkeä osio, jonka tukimateriaali sisältää hyvin linkkejä hyödyllisiin materiaaleihin. Kohdassa olisi hyvä mainita, että API:t lisäävät väistämättä tietoturvariskiä ja monimutkaistavat ympäristöä, jolloin lisäresursointia API:en tietoturvan takaamiseksi olisi vastaavasti hyvä harkita.

Periaatteen 3.1 Kehitä ohjelmointirajapinnat avoimilla ja teknologiariippumattomilla standardeilla ja protokollilla ydinviestistä THL on täysin samaa mieltä. Myös toimialakohtaisten standardien hyödyntäminen näkyy periaatteessa kuten on syytäkin. Periaatteen tukimateriaalissa tehdään tärkeä suositus: ”Tiedot ja toiminnot on suositeltavaa tarjota web-pohjaisten rajapintojen kautta, jos mahdollista ja käyttötarkoituksen mukaista”. Tätä suositusta voisi tuoda esille enemmänkin.

Periaatteessa 3.1 rajapintakuvausten logiikka ei ole täysin systemaattista separation of concerns -periaatteen näkökulmasta. Rajapintakuvauksissa usein erotetaan toisistaan ainakin tiedonsiirtoprotokolla, toiminnallinen ja sisällöllinen rajapintakuvaus (operaatioiden/tietosisältöjen kuvaukset, syntaktinen ja semanttinen taso molemmissa) ja tietoturvakäytännöt. Nämä voivat jopa jossain määrin olla usein riippumattomia toisistaan. Esimerkiksi saman XML-rakenteen mukainen tiedosto voi siirtyä yhtä hyvin HTTP:llä, FTP:llä, Bluetoothilla, tai muistitikulla. Tietoturva- tai autentikointiparametrien sisällyttäminen toiminnallisiin rajapintakuvauksiin ei ole suositeltavaa. Sama rajapinta (tiedonsiirto ja sisällöllinen rajapintakuvaus) voi toimia ilman autentikointia ”sisäinen API” käytössä mutta autentikoinnin kanssa ”kumppani API” käytössä.

Periaate 3.2 Kuvaa ohjelmointirajapintojen käsittelemät tiedot yhteisten ja yleisten tietomallien mukaisesti on semanttisen yhteentoimivuuden näkökulmasta suosituksen tärkein kohta. Tässä kohdassa tulee korostaa toimialaspesifien standardien, termistöjen ja tietomallien hyödyntämistä. Esimerkiksi näin: ”Huomioi rajapintojen määrittelyssä tai hankinnassa erityisesti toimialakohtaiset säädökset ja standardit, jotka voivat määrittää jollain tietyllä toimialalla tai toimialan joihinkin käyttötarkoituksiin kansallisesti tai kansainvälisesti yhteisesti käytössä olevat tietorakenteet, koodistot, termistöt tai luokitukset.” Tukimateriaalin esimerkkeihin – jos ei yleisiin niin ”Esimerkkejä toimialakohtaisista koodistoista ja tietomalleista” – THL ehdottaa lisättäväksi: Sosiaali- ja terveydenhuollon kansallisen koodistopalvelun sisällöt (<https://koodistopalvelu.kanta.fi/>) tai Sosiaalihuollon tietokomponentit ja asiakirjarakenteet (<https://sosmeta.thl.fi/>).

Periaatteessa 3.3 Testaa, versio, dokumentoi ja julkaise ohjelmointirajapinnat on hyvin huomioitu versiointi, dokumentaatio ja julkaisu. Testauksen yhteydessä mainitaan suorituskyky, joten vastaavasti testauksen vaiheessa olisi hyvä huomioida kuormitustestaus.

Periaatteen 3.4 Seuraa ohjelmointirajapinnoille asetettuja mittareita ja muita ominaisuuksia Mittareiden kohdalla voisi korostaa asiakaslähtöisyyttä: millainen on rajapinnan hyödyntäjien kokemus (päälläolosta, vasteajoista jne.). Esimerkkinä annettu "CPU-resurssien riittävyys" on toissijainen mittari, joka voi vaikuttaa rajapinnan toimivuuteen ja sitä kautta rajapinnan hyödyntäjien kokemukseen, tai sitten ei.

Nousjärvi Joni
Terveiden ja hyvinvoinnin laitos THL