1. ÚTGÁFA | SEPTEMBER 2024

Þróunarhandbók

Leiðarvísir fyrir hugbúnaðarþróun fyrir Embætti landlæknis

Þessi þróunarhandbók hefur að geyma leiðbeiningar, almennt verklag og leiðarvísa sem snúa að þróun, gæðaeftirlits og útgáfu hugbúnaðar fyrir embætti landlæknis Íslands.

Athugið að efni þessa skjals ber að líta á sem **leiðarvísi** til að hjálpa embættinu að ná sínum langtíma markmiðum í hugbúnaðargerð og eru allir sem lesa þennan texta **eindregið hvattir** til að koma með **tillögur að úrbótum** og benda á **villur**, **veikleika** eða **vankanta,** á mrh@landlaeknir.is

Efnisyfirlit

Áskorun embættisins

Hvað er verið að reyna að leysa

Embættið miðar við að hugbúnaðarlausnir sem smíðaðar eru fyrir það verði í rekstri og viðhaldi í allt að 30 ár. Langtímasýn er nauðsynleg því að á þessum 30+ ára rekstrar og viðhaldstíma er gert ráð fyrir að mörg hugbúnaðarteymi komi að verkefnunum og geri á þeim viðbætur og lagfæringar. Yfir 30 ára tímabil verða einnig miklar tækniframfarir og margar sveiflur í straumum og stefnum. Því leggur embættið áherslu á að við hugbúnaðargerð séu ákvarðanir teknar sem miða að því að arkitektúr og tæknistafli standist til lengri tíma litið. .

Það sem embættið leggur sérstaklega áherslu á er

- Framtíðartryggðar tækniákvarðanir
- Viðhaldsþolinn hugbúnaðararkitektúr
- Skýr og einföld sjálfvirknivæðing
- Kjarnyrt skjölun

Framtíðartryggðar tækniákvarðanir

Það er ómögulegt að vita hvað framtíðin ber í skauti sér. Það er samt mögulegt og á sama tíma mikilvægt að reyna eftir bestu getu að lágmarka kostnað og “spól í sömu förum” tengdum breytingum á tækniumhverfi yfir lengri tíma.

Eftirfarandi spurningar þarf að hafa í huga þegar tekin er ákvörðun um notkun á tækni í bæði ný og eldri verkefni

1. **Bakhjarlinn** Hverskonar bakhjarl hefur tæknin og hversu líklegt er að bakhjarlinn fjárfesti í henni til langtíma?
    a. Er tæknin ný vara og líkleg til að taka miklum breytingum? Er henni viðhaldið af hópi áhugafólks eða af fámennum hópi innan nýs fyrirtækis?

   b. Hvert er orðspor bakhjarlsins, fjárfestir hann almennt til langtíma í tækni eða eru teknar "skyndiákvarðanir" að hætta með verkefni?

2. **Tæknin** Er forritasafnið (e. library) eða flutningsramminn (e. framework) sem verið er að velja í lausnina þroskað? Hversu líklegt er það til að vera enn þá til staðar og notað eftir 5 ár? En eftir 10 ár?

   a. Samræmist tæknin öðrum verkefnum hjá embættinu? Er þetta gjörólíkt því sem þegar er til staðar og þekking er á?

   b. Krefst tæknin þess að nota þarf nýtt forritunarmál?

   c. Er tæknin hönnuð fyrir þær kröfur sem embættið hefur til síns hugbúnaðar? Er tækninni ætlað að leysa vandamál sem einkenna stærri markaði/hópa en eiga lítið eða ekkert við kröfur smærri markaðs eins og Íslands?

   d. Fellur tæknin inn í sjálfvirknivæðingu og sjálfvirkt gæðaeftirlit stofnunarinnar?

   e. Hversu oft koma út nýjar stórútgáfur (e. major version) af tækninni? Hversu langan líftíma hefur hver stórútgáfa? Er langtíma stuðningur fyrir útgáfur?

   f. Hversu vel fellur tæknin að forritunarumhverfum sem til eru í dag? Er tæknin studd í kóðaritlum og tólum? Hvernig er upplifun hugbúnaðarforritara?

3. **Mannafli** Hversu auðvelt verður að sækja nýja forritara til að viðhalda hugbúnaðinum í tæknistaflanum eftir 5 ár? 10 ár?

   a. Hversu auðvelt er að sækja forritara í dag? Eru margir á Íslandi með þekkingu á að reka og viðhalda lausnum í þessari tækni?

   b. Er tæknin aðgengileg óreyndari forriturum?

   c. Er auðvelt að sækja verktaka erlendis frá til að vinna að verkefninu?

   d. Hversu mikið hefur verið skrifað og er þekkt um lausnina? Eru upplýsingar á StackOverflow t.d.? Eru fáir sérfræðingar til? Er tæknin vel skjöluð?

Viðhaldsþolinn hugbúnaðararkitektúr

Embættið leggur áherslu á að skipulag og uppbygging forritunareininga innan kerfisins styðji vel við það að upphaflegu hönnunarmynstri kerfisins sé viðhaldið rétt í lengri tíma og án aðkomu upphaflegu

hugbúnaðarsmiðanna. Nauðsynlegt er að það sé auðvelt fyrir nýjan forritara, án sérstakrar handleiðslu annarra, að vinna í eldri virkni eða að útfæra nýja virkni á skjön við heildar högun lausnarinnar.

Arkitektúrinn þarf að aðgreina skýrt eftirfarandi þætti

1. Móttöku og afhendingu gagna (e. request/response).
2. Stýringar (e. controllers).
3. Flæðistjórnun innan kerfisins.
4. Gagnasókn og gagnaskrif.
5. Ytri og innri gagnahögun (e. domains/models).

Til að tryggja þetta vill embættið að allur hugbúnaður sem er skrifaður fylgi eftirfarandi högun og hugmyndafræði.
Það er okkar álit að þessi aðferðarfræði geti stutt best við þá langtímaáskoranir sem hugbúnaðurinn okkar mun standa frammi fyrir

1. SOLID principles + SoC (Separation of concerns)
2. ONION architecture + Principle of least knowledge / Multilayered architecture
3. CLEAN code
4. KISS (keep it simple silly)
5. DRY (don’t repeat yourself)
6. YAGNI (You aren’t going to need it)

Nauðsynlegt er að skipuleggja kóða á þann hátt að hægt sé að sannreyna á sjálfvirkan hátt (t.d. með einingaprófunum) að arkitektúr högun sé framfylgt rétt í lausninni. Embættið mun einnig fjárfesta í viðhaldi á sjálfvirkni til að reyna að tryggja að lausnir sem afhentar eru fylgi þessum högunum til lengri tíma.

Skýr og einföld sjálfvirknivæðing

Mikil áhersla er lögð á að nýta sjálfvirknivæðingu sem mest í hugbúnaðarumsýslu embættisins. Snertir þetta alla hluta ferlisins og sérstaklega afurðaframleiðsluna og eftirlit.

Mögulegt er að draga mikið úr skjölun og villuhættu með því að útfæra prófanir í hugbúnaðarkóða sem framfylgja og styðja við eftirfarandi ferla.

Embættið gerir kröfu um að hugbúnaðurinn sem smíðaður er fyrir það uppfylli í það minnsta eftirfarandi sjálfvirkni

- Útgáfuferla: sjálfvirkar CI/CD pípur sem byggja og gefa út hugbúnaðinn, kubernetes samvirkni.
- Samþættingar: OpenAPI skilgreiningar.
- Prófanir: Sjálfvirkar eininga, viðmóts og samþættingaprófanir.
- Gæðaeftirlit: Sannreyningar á gæðum hugbúnaðar, skjölun sem styðja vottunarþarfir embættisins.
- Rekstur og eftirlit: Hugbúnaðarvirkni sem styður við sjálfvirka vöktun og bilanaleit í hugbúnaðinum.

Kjarnyrt skjölun

Markmiðið með skjölun er að þær upplýsingar sem þarf til að viðhalda langtímagæðum og högun lausna fylgi hugbúnaðarútgáfum fyrir embættið hverju sinni.

Góð og auðskilin skjölun eykur líftíma lausna til muna og felur í sér mikla langtíma hagræðingu fyrir nýja forritara og hönnuði sem koma að lausninni.

Markmið embættisins er ekki framleiðsla á þykkum doðröntum eða umritanir á almennri vitneskju, heldur kjarnyrt skráning á mannamáli sem lýsir högun og virkni viðkomandi lausnar, tengslum hennar við umheiminn og skráning á forsendum þeirra ákvarðana sem teknar voru.

Kóðageymslur

> 🤗 **Leiðarljós**
>
> Gæði og sjálfvirknivæðing er mikilvæg fyrir embættið, hins vegar vill embættið ekki setja upp óþarfa þröskulda sem hindra geta vinnu verktaka meðan verkið er að vinnast. Verklaginu sem lýst er hérna er ætlað að reyna að ná fram sem mestum sveigjanleika og sjálfstæði allra aðila í samvinnu þegar unnið er að verkefnum fyrir embættið en á sama tíma tryggja endanleg gæði þess sem skilað er í miðlægar kóðageymslur embættisins.

Skipulag kóðageymsla (e. code repository)

Hugbúnaðarkóði fyrir aðgreindar þjónustur skal ekki vera geymdur saman í einni kóðageymslu (e. mono-repo fyrirkomulag). Ein kóðageymsla skal notuð fyrir hvern hluta lausnarinnar sem er gefinn út sjálfstætt.

Allar kóðageymslur embættisins eru læstar og er ekki hægt að gera breytingar á innihaldi þeirra nema að fara í gegnum kóðarýni og samþykktarferli.

Einungis ein langlíf kóðagrein skal vera til í kóðageymslum landlæknis og skal hún nefnd main.

Sérfræðingar embættisins stofna allar nýjar kóðageymslur í GitHub umhverfi embættisins við upphaf verkefna.

Nafnareglur á kóðageymslum

Kóðageymslur skulu fylgja eftirfarandi nafnareglum

1. Nöfn skulu ekki innihalda hástafi og nota bandstrik í stað bila eða undirstrika
    a. `audit-ui-web`
2. Allar kóðageymslur sem tilheyra sömu lausn skulu bera sama forskeyti
    a. `audit-ui-web`
    b. `audit-api`
    c. `audit-service`

   d. audit-database
   e. audit-ui-app-android
   f. audit-ui-app-ios
   g. audit-app-backend

3. Viðskeytingar skulu reyna eftir bestu getu að vera eitt af eftirfarandi (eða samræmast einu af eftirfarandi)
   a. -ui-web
   b. -ui-admin-web
   c. -api
   d. -service
   e. -database
   f. -jobs
   g. -docs
   h. -app-ui-android
   i. -app-ui-ios
   j. -app-backend
   k. -app-docs-android
   l. -app-docs-ios
4. Ekki skal nota tímabundin og bráðabirgða orð í nöfnum, líkt og -new, -old, -temp osfrv. Ekki skal heldur nota útgáfunúmer í nöfnum (t.d. -v1)

Skilastjórar

Verktaki skipar einn eða fleiri aðila (skilastjóra) sem fá aðgang sem utanaðkomandi notandi (e. outside collaborator) og geta þá sótt afrit (e. fork) af kóða úr kóðageymslu embættisins. Skilastjórar eru þeir einu sem geta skilað útgáfum af kóða inn í rýni til embættisins.

Aðgangur skilastjóra sem ekki hafa skilað inn í kóðageymslur s.l. 3 mánuði verða sjálfvirkt fjarlægðir af kóðageymslum. Tilkynna ber til embættisins ef verktaki skiptir um skilastjóra.

Hvernig ytri aðilar vinna með kóðageymslurnar

Embætti landlæknis geymir allan kóða sinn í **GitHub**, kóðageymslur nota **fork-and-branch** högun.

Embættið gerir kröfu um notkun á staðlaðri aðferðarfræði sem kallast **Fork-and-Branch**.
https://docs.github.com/en/pull-requests/collaborating-with-pull-requests/working-with-forks/fork-a-repo

Ferlinu er ætlað að gefa verktökum kost á að sinna vinnu við hugbúnaðinn í sínu umhverfi og samkvæmt sínum vinnureglum og öryggiskröfum. Embættið hefur svo einungis yfirsjón með endanlegum skilum á hugbúnaðarafurðum til útgáfu eða prófana. Verktakar og aðrir aðilar utan vébanda embættisins skulu ávallt vinna samkvæmt **Fork-and-Branch** ferlinu.

Skylda er að eyða **öllum** afritum (e. forks) úr tölvukerfum, afritum, skýjum og útstöðvum verktaka eftir að verki lýkur.

Fork-and-Branch ferlið virkar í stórum dráttum á eftirfarandi hátt:

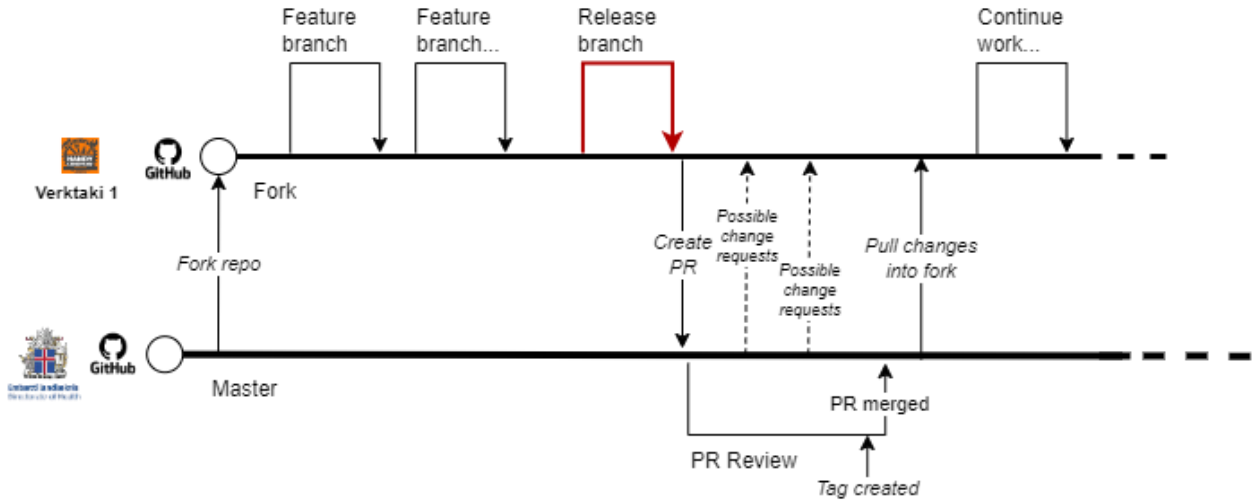
1. Verktaki býr til `fork` af repoinu í GitHub Landlæknis og vistar yfir á GitHub svæði verktaka.
    a. Ath, hvort búa þarf til fork af `main` eða ákveðnu útgáfu tagi
2. Búa til `git clone` af forkinum þínum á local vélina þína
3. Bæta við nýjum remote sem heitir `upstream` fyrir þitt klón sem bendir á GitHub repo landlæknis
4. Byrja að vinna að verkinu og committa breytingar í þinn fork
    a. Mælt er með að notist við feature branches.
5. Þegar kemur að gefa út útgáfu af lausninni, þá búa til útgáfu branch byggða á forkinum þínum
    a. Athugið nafnareglur á útgáfu branches.
6. `commit` þær breytingar sem þarf fyrir útgáfu (uppfæra pakka, númer etc) í útgáfu branchið
7. `push` útgáfu branchinu upp í forkinn í þínu GitHub
8. Opna PR frá útgáfubranchinu yfir í Repoið hjá Landlækni
    a. Athugið að öll commit skulu vera `squashed` og einungis eitt merge commit er leyft til landlæknis

9. PR verður code reviewað og leiðréttinga gæti verið þörf, leiðréttingar gerast í útgáfu branchið.
    a. Eftir að PR er samþykkt þá verður búið til tag í GitHub landlæknis fyrir útgáfunúmerið
10. Eftir að PR hefur verið samþykkt, tögguð og sameinuð við Repoið hjá Landlækni, þá þarftu að uppfæra forkinn í þínu GitHub

```
a. git pull upstream main
b. git push origin main
```

Meðfylgjandi er nánari yfirferð yfir helstu þætti þessarar aðferðarfræði fyrir byrjendur https://blog.scottlowe.org/2015/01/27/using-fork-branch-git-workflow/

Ferlið í myndum



Nafnareglur á kóðagreinum

Kóðageymslur Embættisins taka einungis við breytingarbeiðnum (Pull requests, eða PR) sem fylgja eftirfarandi nafnareglum.

Titlar/nöfn á breytingabeiðnum þurfa að vera nákvæmlega eins og nafn á viðkomandi kóðagrein.

Kóðagrein (e. branch name)	**Lýsing**
release/2024.08.05 release/2024.08.05-hotfix1	Loka útgáfur af fyrirfram ákveðnum útgáfum eða neyðarútgáfum af hugbúnaði sem ætlaður er til útgáfu á pre-production og production

	umhverfi embættisins. Athugið einungis útgáfur merktar á þennan hátt munu vera leyfðar á production umhverfi.
qa/2024.08.05-rc1	Útgáfur eingöngu ætlaðar til sjálfvirkrar útgáfu á pre-production umhverfi embættisins. Ath: -rc breyta verður að fylgja
test/2024.08.05-dev	Prófunarútgáfur af hugbúnaði sem ætlaður er til sjálfvirkrar útgáfu á test umhverfi embættisins.

Gæða og útgáfuferlar

Sjálfvirknivæðingarmarkmið

> 🌞 **Leiðarljós**
>
> Embættið leggur mikla áherslu á réttleika og gæði hugbúnaðarins sem smíðaður er í nafnið þess. Til að tryggja einsleita og hlutlæga nálgun á gæðaeftirliti þá innihalda kóðageymslur landlæknis sjálfvirka ferla til að framfylgja eftirlitinu. Skilvirkni sjálfvirknivæðingarinnar byggir á að þeir sem vinna í kóðageymslunum tileinki sér ákveðnar nafnahefðir og vinnureglur.

Skipulag útgáfunúmera

Hugbúnaður landlæknis fylgir dagsetningar útgáfum (e. Calendar Versioning, http://Calver.org). Dagsetningarútgáfur gefa breiðari hópi notanda betri yfirsýn yfir innihald útgáfa, samvirkni milli útgáfa og hversu nýlegur hugbúnaðurinn er. Semantic versioning (þ.e. 1.0.0) er ekki notuð.

Númerin skulu fylgja eftirfarandi formi

YYYY.**0M**.**0D**-*breyta*

- YYYY - Fjögurra stafa ár - 2024, 2026, 2106…
- 0M - Tveggja stafa mánaðarnúmer, núll fremst ef þarf - 01, 02 ... 11, 12
- 0D - Tveggja stafa mánaðardagur, núll fremst ef þarf - 01, 02 ... 30, 31
- breyta - Valkvæður viðbótar texti fyrir séraðstæður, t.d. "dev", "alpha", "beta", "rc1", “hotfix”, o.s.frv.. Ef gefa þarf út margar breytu útgáfur innan sama dags skal nota “-breytaN” þar sem N - er tala 1…99)

Undantekning frá þessari reglu er hugbúnaður sem gefin eru út í app-verslunum á borð við Google store og App store, þar sem gerð er krafa um að nota Semantic versioning fyrir ákveðin útgáfunúmer.

Allur hugbúnaður hjá embættinu hlýtur útgáfunúmer á sjálfvirkan hátt eftir að gæðastýringarferlum er lokið. Útgáfunúmer ákvarðast af nafni á þeirri kóðagrein sem er rýnd hverju sinni.

Sjálfvirkir gæðastýringarferlar

Allir sjálfvirkir ferlar eru keyrðir af GitHub Actions keyrsluumhverfinu

Fyrir hver kóðaskil, PR (e. pull request), sem eru búin til í kóðasöfnum embættisins fer eftirfarandi ferill í gang

1. Kóðinn er prófaður m.t.t. uppbyggingar lausnar
2. Kóðinn fer í gegnum build ferli
3. Kóðinn fer í gegnum einingaprófunarferli
4. Kóðinn fer í gegnum sjálfvirkt kóðarýni og kóðagæðaeftirlit
5. Keyrslustýringar (docker) fer í gegnum öryggisprófun
6. Allur gagnagrunnskóði fer í gegnum gæðaeftirlit

Skili eitthvert af þessum sjálfvirku skrefum villu er PR sjálfkrafa hafnað og höfundi ber að gera viðeigandi breytingar til að tryggja að gæðakröfur embættisins séu uppfylltar.

Hálf-sjálfvirkir gæðastýringarferlar

Ef að sjálfvirkir ferlar tilkynna engar villur þá fer í gang kóðarýni á skilunum. Þessi kóðarýni eru gerð af sérfræðingum embættisins.

Áhersla er lögð á eftirfarandi atriði í kóðarýnum

1. Réttleiki hugbúnaðar og gagnagrunnskóða
2. Langtíma heilsa og viðhalds eiginleikar lausnar
3. Arkitektúr lausnarinnar
4. Gæði skjölunar
5. Uppbygging og skipulag skráa og lausnar
6. Gæði og viðhalds eiginleikar sjálfvirkra prófana
7. Öryggis og áhættumat á ytri tengslum (e. dependency) við aðra hugbúnaðarpakka og lausnir.

Ef kóðarýnir (e. reviewer) metur að eitthvað af áhersluatriðum sé ábótavant er PR hafnað og þess óskað að höfundur geri breytingar til að tryggja að gæðakröfur embættisins séu uppfylltar.

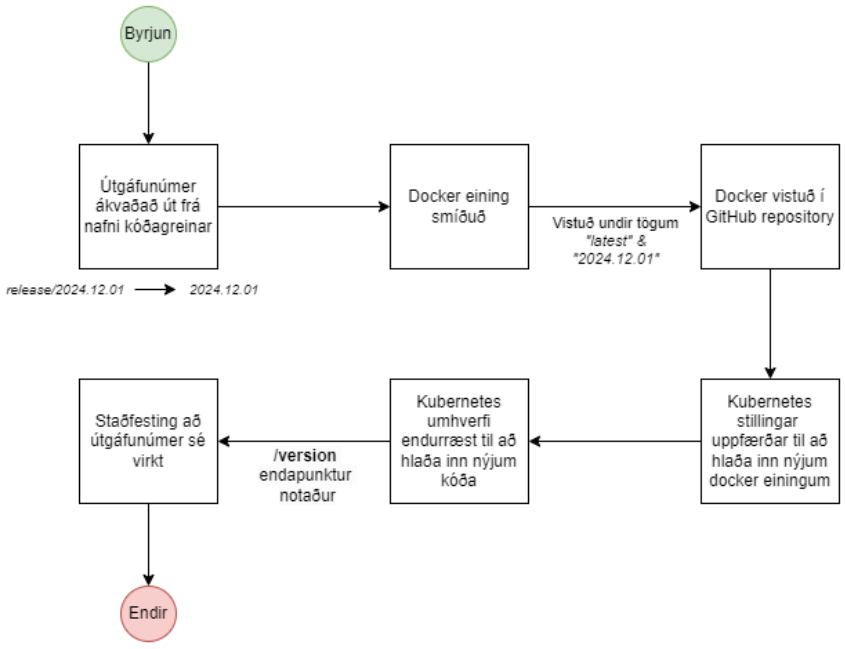
Sjálfvirkir útgáfuferlar

Allur hugbúnaður sem ætlaður er til útgáfu á rekstrarumhverfi embættisins er útbúinn til útgáfu með sjálfvirkum ferlum. Þetta á við um öll umhverfi, test, pre-production og production. Þessi sjálfvirknivæðing er til að tryggja gæði útgáfuferla og til þess að draga úr líkum á mannlegum mistökum við útgáfu hugbúnaðar.

Sjálfvirk útgáfuferli embættisins fylgja eftirfarandi skrefum

1. Kóði fær sjálfvirkt útgáfunúmer (e. tag) í Git kerfinu
2. Smíði Docker einingar (e. container)
3. Docker einingu er gefið sama útgáfunúmer og vistuð í miðlæga geymslu (e. container registry) í GitHub embættisins
4. Kubernetes stillingar eru uppfærðar með nýjum útgáfunúmerum og stillingum
5. *Einungis á Pre-prod og Test*: Nýjar docker einingar eru virkjaðar á Kubernetes umhverfi og lausnin er gefin út
6. *Einungis á Production*: Kubernetes stillingar þurfa að vera handvirkt uppfærðar á umhverfi af sérfræðingi embættisins

Útgáfuferillinn í mynd



Á hvaða umhverfi eru útgáfur leyfðar á?

Sjálfvirkir útgáfuferlar skulu bjóða upp á að gefa út útgáfur á eftirfarandi umhverfi eftir því hvernig kóðagrein var verið að senda inn.

“Útgáfa” á hugbúnaði á einungis að fela í sér breytingar í Kubernetes á því útgáfunúmeri docker einingarinnar sem á að nota.

Umhverfi/Kóðagrein	Sjálfgefin útgáfutög	release/YYYY.MM.DD	qa/YYYY.MM.DD	test/YYYY.MM.DD
Production	*latest*	**Handvirkt**	**Ekki leyft**	**Ekki leyft**
Pre-Production	*latest-qa*	**Sjálfvirkt**	**Sjálfvirkt**	**Ekki leyft**
Test	*latest-test*	**Handvirkt**	**Handvirkt**	**Sjálfvirkt**

Sjálfvirkar gæða og öryggisprófanir

Athugið að prófanir og gæðaeftirlit eru ekki keyrðar fyrir test/xxxx.yy.zz útgáfur.

Gæðaprófanir

Allar PR sem eru búnar til í kóðageymslum embættisins fara í gegnum sjálfvirkar gæða og öryggisprófanir, þessar prófanir þurfa að standast kröfur til þess að kóðinn sé samþykktur og að hægt sé að gefa hann út á umhverfum embættisins.

Stillingarnar á gæða og öryggisprófunum er í höndum embættisins og er óleyfilegt að breyta þeim án samþykkis verkefnastjóra embættisins.

Embættið notar meðal annars SonarCloud til að framkvæma öryggis og gæðaprófanir á kóða.

Athugið, SonarCloud prófanir eru stilltar þannig að viðvaranir eru meðhöndlaðar sem villur (e. treat warnings as errors). Lagt er til að forritarar stilli þróunarumhverfi sín á sama hátt.

Docker öryggisprófanir

Allar lausnir þurfa að innihalda docker skipanir og þurfa þær skrár að standast öryggisprófanir (e. container scanning). PR er hafnað ef athugasemdir finnast með hæsta viðvörunarstig (e. “Critical”).

Eftirfarandi tegundir prófana eru alltaf gerðar á docker einingum (viðbótar prófanir gætu verið keyrðar á ákveðnum tegundum af docker einingum).

1. Greining á grunn docker einingum sem notaðir eru (base image scanning)
2. Greining á stýrikerfispökkum sem grunneiningar stóla á (third-party dependencies)

Skil á hugbúnaðarafurðum

Embættið hvetur eindregið til þess að vinnuhópar vinni samkvæmt Minimum-Viable-Product högun og að skilastjórar skili reglulega hugbúnaði og hugbúnaðarafurðum inn til rýni hjá embættinu (að lágmarki einu sinni á 3 mánaða fresti). Með því er hægt að leysa fyrr úr mögulegum málum og gera stefnuleiðréttingar ef þörf krefur.

Mælst er til að reynt sé að skila reglulega inn útgáfum af þeirri virkni sem kláruð er til rýni og samþættingaprófana.

Hverju á að skila?

Vinnuhópar eru hvattir til að skila reglulega inn útgáfu af öllu því sem þeir hafa hugsað sér að skila í lok verkefnisins til að hægt sé að rýna verkið eins og það vinnst. Skil ná til allra þátta verkefnisins, þ.m.t. kóða, prófana og skjölunar.

Hver skilar?

Skilastjóri hvers verkefnis skilar vörum inn til embættisins.

Hvenær er æskilegt að skila inn?

Skil	1x í viku	1x í mánuði	<1x í mánuði	Í lok verkefnis
Á test umhverfi	**Æskilegt**	**Í lagi**	**Óæskilegt**	**Óæskilegt**
Á pre-production umhverfi	**Í lagi**	**Æskilegt**	**Æskilegt**	**Óæskilegt**
Á production umhverfi	**Óæskilegt**	**Óæskilegt**	**Í lagi**	**Æskilegt**

Forritunarkóði

Uppbygging og hönnun

> 🌞 **Leiðarljós**
>
> Embættið gerir ráð 30+ ára rekstrar- og viðhaldstímabils fyrir hugbúnaðarlausnir. Af þessum sökum er mikilvægt fyrir embættið að velja bæði arkitektúr, högun, forritunarmál og forritunaraðferðir sem styðja vel við slík langtíma verkefni. Sérstaklega þarf að hafa í huga að margir og mismunandi hugbúnaðarsmiðir og hönnuðir geta komið að verkefnum í styttri eða lengri tíma. Tæknival og forritunarstíll þarf því að vera óháð tímabundnum vinsældum í straumum og stefnum. Tækni sem valin er þarf að vera líkleg til að laða að sér nýtt hugbúnaðarfólk til lengri tíma.

Forritunarmál og framework

Bakendakerfi / API

Bakendakerfi, s.s. APIs og þjónustur, skulu skrifuð í C# og notast við .NET umhverfið. Ávalt skal notast við nýjustu LTS (e. long term support) útgáfuna af .NET umhverfinu sem verður í gildi við áætlaðan útgáfudag hugbúnaðarins. Bakendalausnir skulu leitast til við að nýta til hins ýtrasta innbyggða virkni í þau forritunar framework sem þau eru hönnuð fyrir, sbr ASP.NET.

Framendar / Web

Forritunarlausnir fyrir framenda skulu leitast við að nýta á sem bestan máta innbyggða staðla til vefsíðubirtinga s.s. HTML5 og CSS. Notast skal við þau framenda framework sem eru þegar ráðandi í lausnum hjá embættinu. Ekki skal smíða lausn í forritunarmáli eða umhverfi sem ekki hefur verið notað áður hjá embættinu nema með leyfi sérfræðings frá embættinu.

Allir framendar sem eru smíðaðir, verða að fylgja útgefnum hönnunarstöðlum og útlitsstílsniði embættisins

- Design system – Desktop UI Library
- @storybook/cli - Storybook

Til útfærslu á gagnvirkni skal reynt að nýta hugbúnaðartækni sem útbýr og þjónustar vefsíðuefni af netþjóni (e. server-side rendering, SSR) fram yfir virkni sem keyrir eingöngu í vafra notanda. Dæmi um slíka tækni er Blazor.NET og ASP.NET. Upplýsingakerfi sem stóla á aðgengi leitarvéla að vefslóðum (e. URLs), kerfi sem krefjast ekki sérstakrar gagnvirkni í viðmóti, eða þurfa ekki að styðja slitrót samband við alnetið skal útfæra alfarið sem SSR kerfi. Ekki skal notast við SSR virkni í SPA forritunarsöfnum (líkt og Angular, React, Svelte, etc).

Ef SSR nálgun er ekki fýsileg og þörf er á að útfæra SPA vefsíðu sem keyrir í vafra notenda þá skal stuðst við Javascript forritunarmálið á þann hátt að hægt sé að gæta að kóðagæðum (e. type checking og static analysis) í forritunarkóða til að lágmarka hættu á forritunarmistökum. Ef Javascript útgáfan styður ekki slíka virkni skal notast við nýjustu LTS útgáfu af TypeScript málinu. Einungis skal notast við React við gerð SPA veflausna.

Ekki skal undir neinum kringumstæðum útfæra veflausnir fyrir embættið í vefumsjónarkerfum án skriflegs samþykkis frá tæknilegum sérfræðingum og verkefnastjórn embættisins.

Snjallforrit / Mobile

Almenn regla er að snjallforrit ætluð fyrir snjalltæki (Android eða iOS) skulu í lengstu lög vera hönnuð og útfærð sem snjalltækjavænar vefsíður frekar en með annari tækni. Þumalputtareglan er að útfæra skuli hefðbundna veflausn ef forritið krefst ekki mjög sértækrar tækjavirkni (sem erfitt er að gera í veflausn).

Í þeim undantekningartilvikum þar sem ekki er hægt að útfæra veflausnir þá skal leitað samþykkis sérfræðinga embættisins áður en ákveðið forritunarumhverfi eða framework er valið.

Öll viðmót fyrir snjallforrit skulu vera hönnuð og virkni þeirra aðgengileg á ásættanlegan máta á a.m.k. 75% af þeim snjalltækjum og útgáfum þeirra sem eru í notkun í markhóp á útgáfudegi.

Hönnun og arkitektúr

Allur arkitektúr og hönnun þarf að vera unnin í samráði við sérfræðinga embættisins. Hugbúnaðarsmíði skal ekki hefjast fyrr en samþykki á hönnun og arkitektúr liggur fyrir frá embættinu. Hönnun á högun og venslum gagna skal einnig liggja fyrir áður en hugbúnaðarsmíði hefst.

Viðmót og vefir

Leitast skal við að hönnun og forritun á vef og app framendum fylgi micro-frontend hönnunarmynstrum og aðgreini skýrt mismunandi aðgerðarsvið innan framendalausnarinnar.

Þjónustur

Embættið tekur ekki við hugbúnaði til rekstrar sem samsettur er úr smáþjónustum (e. micro/nano services). Bakendakóði skal vera útfærður samkvæmt þjónustuhögun (e. service orientated architecture) en með áherslu á stærri þjónustueiningar.

Allar þjónustur skulu hafa OpenAPI skilgreiningar sem innihalda skjölun á endapunktum, breytum og gildum sem tekið er við og skilað er. Einnig skal OpenAPI skilgreiningin innihalda dæmi um innsend og skilagögn (e. request and response). Skilgreiningin skal líka skjala villukóða og skilagildi þeirra.

Skýrt skal vera hvaða auðkenningarmáti er notaður í forritunarskilum þjónustunnar og skal vera hægt að prófa þjónustuna á sjálfvirkan máta óháð ytri auðkenningarþjónustum.

Aðgreina skal með viðskeyti á nafni á milli þjónusta sem eiga að vera aðgengilegar á internetinu og þeirra sem sitja bak við eldvegg. Nota skal -api viðskeyti fyrir þjónustur sem eru aðgengilegar internetinu, viðskeytið -service skal nota fyrir þær sem sitja fyrir innan eldvegg. T.d. auditing-service og auditing-api

Forritunarkóði

Allur kóði skal fara í gegnum samræmingar og stílaðlögunar forrit (e. linting). Embættið notast við SonarCloud og SonarLint til að samræma kóðastíl.

- https://www.sonarsource.com/products/sonarlint/ide-login/
- https://www.sonarsource.com/products/sonarcloud/

Til viðbótar þá eru eftirfarandi sér forrit notuð til að tryggja samræmi og öryggi í kóða.

- C#: Roslyn Analyzers (StyleCop)

Öll samræmingar og stílaðlögunarforrit skulu vera stillt á þann hátt að viðvaranir séu meðhöndlaðar sem villur (e. treat warnings as errors).

Mælst er til að verktakar sinni reglulegu gæðaeftirlit, t.a.m. séu með reglulegt kóðarýni, á meðan að á verkinu stendur til þess að unnið sé jafnt og þétt að því að tryggja gæði, öryggi og stöðuleika afurðarinnar.

Forritarar skulu gæta að forrita í sama kóðastíl og viðkomandi lausn hefur nú þegar, eða í þeim stíl sem sambærilegar lausnir hjá embættinu bera. Ekki skal gera breytingar á kóðastíl lausnar án fyrra samtals og samþykki frá sérfræðingi embættisins.

PR skulu ekki innihalda breytingar sem eru ótengdar þeirri vinnu sem var verið að framkvæma, eða innihalda breytingar í ótengdum hlutum kóðasafnsins.

Pakkar og namespaces skulu byrja á “Directorate.Health.PROJECTNAME” eða “Directorate/Health/PROJECTNAME” eftir því sem við á. Þar sem *PROJECTNAME* er nafnið á verkefninu.

Prófanir

- Einingaprófanir (Unit tests) skulu fylgja öllum forritunarkóða
    - Einingaprófanir skulu aldrei þekja minna en 65% af kóða þjónustunnar. Allar meginlínur þjónustufalla skulu vera prófaðar af a.m.k. einu einingaprófi sem staðfestir virkni aðgerðar og öðru einingaprófi sem staðfestir villumeðhöndlun aðgerðar
    - Einingaprófanir skulu vera í sér geymdar í sér projecti (C#), sér rótar pökkum (Java) eða í sér rótar möppum/namespace (önnur mál)
    - Uppbygging einingaprófanna skulu endurspegla möppuskipulag lausnarinnar, en hafa rótarmöppurnar /Tests/UnitTests
- Prófunargögn fyrir einingarprófanir skulu fylgja forritunarkóða og vera hluti af einingaprófunar projecti. Prófunargögn skulu sitja undir rótarmöppu verkefnisins í /TestData og endurspegla möppuskipulag lausnarinnar
- Öll forritunarskil (APIs) skulu innihalda viðeigandi samþættingaprófanir fyrir alla endapunkta sem eru aðgengilegir
- Það skal vera hægt að prófa forritunarskil þjónusta á sjálfvirkan máta óháð ytri auðkenningarþjónustum
- Einingapróf skulu vera skrifuð í viðurkenndu einingarprófunar frameworki, s.s. xUnit, jUnit eða sambærilegu. Ekki er leyfilegt að nýta heimasmíðað eða óreynt framework til einingaprófana sem hluta af hugbúnaðarlausninni né nota eitthvað sem var smíðað af verktaka og ætlað fyrir önnur verkefni
- Skjölun í forritunarkóða (e. comments)
- Comment, forritunarkóði og leiðbeiningar fyrir forritara skulu skrifuð á ensku

- Vanda skal málfar og gæta þess að persónulegar skoðanir eða formælingar eiga ekki heima í forritunarkóða
- Að lágmarki skulu klasar, föll, fastar, enum, osfrv, sem eru merkt “public” vera skjöluð með upplýsingum sem lýsa tilgangi viðkomandi einingar, auk sambærilegrar skjölunar á svæðum skilagildum og inntaksgildum og villuskilyrðum
- Enum og constant kóði skal hafa skjölun fyrir hvert gildi ásamt lýsingum á virkni þegar gildi er valið.
- Leitast skal við að comment lýsi afhverju kóðinn er gerður á þann hátt sem hann er gerður en ekki hvað hann er að gera.
    - Flott: `//Confirm that the user has read privileges to the data`
    - Slæmt: `//Call privilege service`

Hugbúnaðarprófanir

> 🤗 **Leiðarljós**
>
> Gæði og hagkvæmni prófana á hugbúnaði er jafn mikilvægar embættinu og lausnarkóðinn sjálfur. Ómögulegt er að byggja upp öryggi í framtíðauppfærslum og breytingum án fjárfestingu í góðum gæða prófunum frá byrjun verkefnis. Það er stefna embættisins að hugbúnaðarlausnir okkar verði í rekstri og viðhaldi í fjölda ára. Með þetta að leiðarljósi er mikilvægt að allar prófanir sem fylgja hugbúnaðarlausnum embættisins séu bæði skrifaðar og skjalaðar á skýran hátt til þess að standast þessa kröfu um

Almennar uppsetningakröfur

Prófunarumhverfi

- Embætti landlæknis veitir eftir þörfum aðgang að þeim gögnum sem þarf til þess að verktaki geti komið upp prófunarumhverfi sínu

Staðsetning prófana

- GDPR reglugerðin krefst þess að öll gögn landlæknis þurfi að vinna með innan EEA og því er sú krafa gerð að prófanir fari fram innan EEA

Umsýslukerfi fyrir prófanir (test management system)

- Verktaki þarf að vera með umsýslukerfi fyrir prófanir
    - Kerfið þarf að vera aðgangsstýrt
    - Innskráningar, breytingar og keyrslur þurfa að vera skráðar

Utanumhald með villum í þróun

- Sýna þarf fram á að villur séu skráðar, leystar og prófaðar á ný
    - Þetta kerfi má vera aðskilið eiginlega umsýslukerfi prófana

Samhengi krafa og prófanna

- Nauðsynlegt er að það komi skírt fram í prófunargögnum hvaða krafa fylgir hverju prófi
- Hver krafa þarf að hafa að lágmarki 1 skráð próf

Prófanir

Sjálfvirkar prófanir

Öll sjálfvirk próf skal geyma í þar til gerðri möppu sem hlýtur sömu nafnareglum og er talað um í kaflanum um forritunarkóða. Embættið setur sig ekki á móti því að sjálfvirk próf séu keyrð en það er á ábyrgð verktaka að skrifa þær á þann hátt að þær fylgi fyrir framtíðar uppfærslur.

- Einingaprófanir (unit tests)
    - Lágmarks yfirgrip prófanna er 65% með viðmiðið að allur kóði uppfylli >75% viðmið.
        - Prósentan má ekki lækka milli útgáfa
    - Ef kröfum um einingarprófanir er ekki uppfyllt er PR hafnað sjálfvirkt
- Samþættingarprófanir (integration tests)
    - Öll föll í ytri þjónustuskilum skulu vera prófuð af a.m.k. einu samþættingaprófi sem staðfestir virkni aðgerðar (e. success criteria) og öðru samþættingaprófi sem staðfestir villumeðhöndlun aðgerðar (e. failure criteria)
- Sjálfvirkar notendaviðmótsprófanir (UI)
    - Við uppsetningu á sjálfvirkum notendaviðmótsprófanir

Handvirkar prófanir

- Notendaviðmóts próf (UI test)
    - Notendaviðmóts prófanir þurfa að uppfylla þær notendaviðmóts kröfur sem hafa verið settar fram af notendaviðmóts hönnuði við upphaf verkefnisins
        - Í því tilfelli þar sem þessar prófanir eru gerðar handvirkar á slembikenndan hátt (exploratory testing) þarf að gera prófunarskýrslu þess efnis
- Viðhald handvirkra prófana
    - Verktaki þarf að viðhalda öllum handvirkum prófum við uppfærslur
    - Ekki má eyða prófum sem hafa verið skráð sem kröfupróf
        - Verktaki getur merkt þau óvirk

Samþykktarprófanir

- Verksmiðju samþykktarprófanir (FAT: factory acceptance test)
    - Almennt séð fara fram samþykktarprófanir fyrir afhendingu á vörunni
        - Starfsmaður embættis landlæknis er viðstaddur þessa prófana keyrslu
        - Verkefnastjóri verktaka er viðstaddur þessar prófanir
        - Forritari og/eða tæknimaður verktaka er á staðnum ef þess er óskað
    - Þessar samþykktarprófanir eru bókaðar með 2 vikna fyrirvara

Gögn sem þarf að afhenta með kóðanum

- Prófunargögn
    - Við afhendingu er gert krafa um að embættið fái afhent öll prófunargögn
        - Prófunarsvítu (test suite) verkefnis
            - Skráðar keyrslur með tímasetningu
        - Möppur með sjálfvirkum prófunum sem fylgja kóðanum
- Prófunarskýrslur
    - Prófunarskýrsla þarf að berast í beinu framhaldi af samþykktarprófunum eða í hið minnsta 2 sólahringum fyrir útgáfu ef ákveðið hefur verið að sleppa samþykktarprófunum
        - Mikilvægt er að allar skjámyndir og staðfesting á keyrslu séu inní prófunarskýrslunni
        - Prófunarskýrsla er undirrituð af prófara og verkefnastjóra verkefnis eða gæðastjóra prófunar

Hugbúnaðarskjölun

„Enginn les doðranta“

> 🌞 **Leiðarljós**
>
> Hugbúnaður er aldrei betri en skjölunin sem fylgir honum. Markmiðið með skjölun er að búa til það magn af upplýsingum sem fylgir hugbúnaðarútgáfum þannig að hann tryggir sem best að hægt sé að viðhalda langtíma gæðum og högun lausnanna. Markmiðið er ekki mikill fjöldi blaðsíða eða umritanir á almennri vitneskju, heldur högun viðkomandi lausnar og ákvarðanir teknar.

Hvaða skjölun er skilað við afhendingu

Skjölun skal skilað inn í PDF eða Word skjölum og skal samræma útgáfunúmer og hugbúnaðarlausnin sem er skilað.

Athugið að allar tæknilegar myndir sem notaðar eru við skjölunina þarf að skila einnig sér sem .png, eða .jpg skrám.

- Architecture og system design sem varpa skýrri yfirlitsmynd af kerfinu
    - Myndir sem lýsa helstu ytri tengingum við kerfið og samskiptum
    - Myndir sem lýsa tæknistack lausnarinnar
    - Myndir sem lýsa virkni og samtenginga helstu innviða
    - Myndir sem lýsa gagnaflæði í gegnum hugbúnaðinn
- Forsendur og helstu hönnunarákvarðanir (e. architecture design records. ADR). Mælt er með einfaldri nálgun, sbr. https://medium.com/olzzio/y-statements-10eb07b5a177. Hægt er að skrá ADR með því að klára málsgrein sem inniheldur eftirfarandi lykilorð:
    - “Í tengslum við”: Virkni (saga eða notkunartilvik), hluti kerfis, kerfishögun o.s.frv.
    - “standandi frami fyrir”: lýsing á eiginleika sem æskilegur var eða upplifun sem óskað var eftir (non-functional)
    - “ákváðum við”: ákvörðunin sem tekin var

  - “hafandi skoðað”: aðrir valkostir sem skoðaðir voru en hentuðu ekki
  - “til að ná fram”: hagurinn af valinu, hvernig valið uppfyllti skilyrðin sem lögð voru til grundvallar
  - “vitandi það að”: lýsing á veikleikum eða takmörkunum sem eru þekktar tengdar ákvörðununni, langtíma/skammtíma, áhrif á aðra virkni eða kerfi
- Sequence rit fyrir þá kjarna virkni sem útfærð er ásamt lýsingum og athugasemdum. Sérstaklega þarf að gera sequence rit fyrir öll samskipti sem kerfið hefur við ytri þjónustur, og önnur flæði sem forritarar telja að slík rit hjálpi öðrum hugbúnaðaraðilum til þess að skilja betur flæði aðgerða
- Rekstrarhandbók/leiðbeiningar, miðuð við daglegan rekstur lausnarinnar, t.a.m. uppsetning vélbúnaðar (ef við á), innviðaþarfir, keyrsluumhverfi, kubernetes configuration, logga, metrics, monitoring, regluleg maintenance ferli sem þarf að keyra á lausn og gagnagrunni osfrv.

Hvaða skjölun fer í GitHub

Eftirfarandi skjölun lifir með forritunarkóða í kóðageymslum embættisins.

- Eitt **Readme.md** í rót reposins sem lýsir hvað er að finna í því og hvernig lausnirnar hanga saman
- **Readme.md** skjal fyrir hverja sjálfstæða lausn (þ.e.a.s. lausn sem ætlað er að keyra í production eða er gefin út sem library eða pakki sem aðrar lausnir nota)
  - Stutt lýsing á því um hvaða hugbúnað er að ræða, tilgang hans og hvaða hlutverki í stærri verkefnum hann tilheyrir
  - Tæknileg lýsing og högun, hvaða forritunarmál, útgáfur af frameworkum hann er byggður á (t.d. .NET 8 fyrir .net verkefni, spring boot v5.0 fyrir Java, þannig háttar)
  - Yfirlitsmynd sem sýnir hvernig helstu hlutar hugbúnaðarins vinna saman og tengjast
  - Dependency lýsing, hvaða ytri tengingar, gagnagrunn eða ytri hugbúnað stólar þessi hugbúnaður á. Upplýsingar um hvaða secrets er þörf og VPN tenginga ef við á
- **deployment.md** skrá í rót hverjar service eða library projects/solutionar
  - Deployment description, hvernig lausn er gefin út, lýsing á docker fyrirkomulagi, buildscriptum, ásamt skrefum sem þarf að taka til að búa til production útgáfu af hugbúnaðinum

    - Deployment leiðbeiningar og þarfir til keyrsluumhverfa, skjölun á kröfum sem docker scriptur hafa (port, env breytur, etc), gagnagrunnskröfum, eldveggja eða infrastructure þörfum etc

- **developers.md** skrá í rót hverjar service eða library projects/solutionar
    - Leiðbeiningar fyrir nýja forritara hvernig á að koma lausninni upp á forritunarvél og keyra upp til að debugga, þannig að forritari sem kemur lausninni getur keyrt lausnina og debuggað á vélinni sinni
    - Development leiðbeiningar, compatibility matrixur fyrir external library, user secrets, env breytur, aðgangur að kerfum, appsettings etc, property files, env files etc, upplýsingar um hvernig setja skuli upp tengdar þjónustur ef slíkt á við), hvernig eigi að nálgast logga sem þjónusturnar skrifa
- **continuous-integration.md** skrá í rót verkefnis eða repo eins og það á við
    - Continuous integration/deployment, lýsingu á hvernig github actions eru settar upp, branching strategy fyrir repoið (dev work, og svo útgáfu strategia), container repo fyrirkomulag
    - Útgáfunúmerastýring, útgáfu fyrirkomulag úr github,
- **database.md** skrá í rót hverjar service eða í repo rótinni ef einn gagnagrunnur fyrir alla lausnina
    - Gagnagrunns leiðbeiningar, uppsetning á gagnagrunni, vísun í hvar sql scriptur eru að finna oþh.
    - Helstu gagnagrunnsnotendur sem lausnin þarfnast og schema aðgang sem viðkomandi notandi þarf
    - Öll scheduled database jobs sem eru hluti af lausninni og keyrslu schedule þeirra.
    - Leiðbeiningar um uppsetningar og viðhald á gagnagrunnum. Hvernig á að hreinsa gömul gögn, periodic cleanup etc.

Viðkvæm gögn

Öllum viðkvæmum gögnum skal miðlað á öruggan hátt til ábyrgðaraðila verkefnisins innan embættisins.

- Dulritunarlyklar, öryggisskilríki,

- Upplýsingar um allar aðgangstillingar fyrir tengdar þjónustur (s.s. username, passwords, client_id, api_keys etc)

Gagnagrunnar

Geymsla og miðlun gagna

> 🌞 **Leiðarljós**
>
> Gagnaþörf embættisins er mikil og eru öll gögn sem embættið vinnur með í eðli sínu mjög viðkvæm gögn, bæði persónulega og þjóðfélagslega. Áhersla skal vera á að tryggja bæði gæði gagna og kóða, sem og öryggi gagnanna sjálfra, til að auðvelda langtíma viðhald og úrvinnslu þeirra gagna sem verða til í heilbrigðiskerfinu. Einfaldleiki er oftast besta leiðin til að tryggja langtíma gæði.

Hlutverk gagnagrunna skal vera að geyma og miðla gögnum. Gagnaúrvinnsla er ekki æskileg í gagnagrunnum sem liggja undir kerfislausnum (aðrar reglur gilda um gagnavöruhús). Embættið tekur ekki við gagnagrunnskerfum þar sem mikilvæg eða kjarnavirkni kerfisins er skrifuð í stefjum, sýnum, töflu virkni eða tímasettum keyrslum (e. procedures, views, triggers, scheduled database jobs).

Velja skal nýlega útgáfu af Microsoft SQL Server (í samráði við sérfræðinga embættisins). Ef krafa er að velja aðra tegund gagnagrunns þarf að fá leyfi fyrir því frá sérfræðingum og gæðastjóra embættisins.

Forritunarpakkar (e. ORM)

Hugbúnaðarteymi sem eiga samskipti við gagnagrunna skulu velja létta gagnagrunns forritunarpakka (e. ORM, object-relational mapping).

Forðast skal ORM pakka sem virka á þeim grundvelli að gagnagrunnsnotandinn sé með hærri réttindi en SELECT og EXECUTE á það schema þar sem gögnin eru geymd (sbr. Entity Framework, xHibernate). Af öryggissjónarmiðum á gagnagrunnsnotandi sem bakendaþjónustur nýta sér ekki að hafa CREATE, ALTER eða sambærileg aðgangsréttindi sem geta breytt uppbyggingu gagnaschemans í grunninum.

Gæta skal að velja ORM ásamt uppfærslu (e. migration) stefnu sem myndar ekki óhóflegt magn af viðbótar gagnagrunnshlutum, sem gera erfitt að vinna með gagnagrunnskemað í gagnagrunninum (t.d. töflunöfn innihaldi GUID gildi og þannig háttar).

Gagnagrunnshlutir sem ekki skal nota

Embættið vill forðast það að fjárfesta í forritunar- og úrvinnslukóða skrifuðum ofan í gagnagrunnana sjálfa. Þessar kröfur miðast því við að tryggja það að stöðugleiki gagna og heilindi gagnagrunna sé tryggður með því að lágmarka fjárfestingu í sértækri gagnagrunnsvirkni.

Því skal ekki notast við *Stored Procedures*, *Functions* eða *Triggers* til að útfæra virkni kerfisins ofan í gagnagrunninum sem þjónustar keyrsluhluta (e. online part) kerfisins.

Ekki skal nota sértæka SQL virkni sem ekki er hluti af stöðluðum gagnagrunnsútfærslum.

Ef nauðsyn krefst þess að eitthvað af þessari virkni sé notuð skal ræða það á stöðufundi með tæknilegum sérfræðingum embættisins. Ef leyfi fæst skal skýrt skjalað í skjölun kerfisins allar *Stored Procedures*, *Functions* eða *Triggers* sem notaðar eru, hvar þær eru notaðar, og tilgang þeirra.

Scheduled Jobs / *Database Jobs* skal ekki útfæra til að sinna nauðsynlegri kerfisvirkni á keyrsluhluta (e. online part) kerfisins. Ef gagnagrunnskerfið krefst þess að keyrð sé regluleg virkni (t.d. til eyðingar á gögnum) þarf að skjala slíkar keyrslur og tíðni þeirra.

Surrogate/Technical keys

Mælst er til að gagnagrunnskerfi notist við surrogate lykla (Surrogate key) til að einkvæmt einkenna gögn í kerfum og til að halda stöðugleika í lyklum milli gagnagrunnskerfa og uppfærsla.

Dulkóðun gagna

Dulkóðun skal vera útfærð á gagnagrunns stigi eða stýrikerfis stigi. Almennt skal ekki notast við dulkóðun á einstaka gildum í dálkum og töflum í gagnagrunni.

Gögn sem þarf einungis til samanburðar við önnur, sbr lykilorð, á ekki að dulkóða heldur geyma sem hash+salt. Undantekningar á þessu er ef kerfi þarf að viðhalda notendanöfnum eða lykilorðum fyrir ytri kerfi sem ekki er hægt að dulrita á annan hátt og geyma.

Útgáfustjórnun

Geyma skal og útgáfustýra skal öllum gagnagrunnskóða í GitHub repoi embættisins. Ef hægt er skal geyma gagnagrunnskóðann sem SQL scriptur í sér repo (sjá ***Hvernig ytri aðilar vinna með kóðageymslurnar***), ef slíkt er ekki hægt eða óskynsamlegt skal geyma SQL scripturnar með þjónustukóða bakendaþjónusta.

Keyrslustýringar

Docker og Kubernetes

Leiðarljós

Íslendingar eru fámennur hópur (tæp 380þúsund) og almennt séð er álagsþungi í samtengdum notendum eða gagnamagni á heilbrigðishugbúnað ekki mikill. Öryggi, aðgangsmál og uppitími er á hinn bóginn mikilvægur fyrir kerfi sem þjónusta íslenska heilbrigðisgeiranum. Það er því mikilvægt fyrir embættið að keyrsluinnviði kerfa séu sem einsleitust, og það að tryggja öryggi náist fram með einsleitni og einfaldleika. Mikilvægt er að vinnuferlar keyrsluumhverfa séu hannaðir með það í huga að tryggja öryggi og stöðugleika þeirra, og bjóði upp á skýra og skilgreinda aðgreiningu fyrir þróun, prófanir og raunkeyrslur á hugbúnaðarafurðum.

Docker

- **Stærð Image-a**
  - Leitast skal við að hafa image eins litlum og hægt er fyrir hraðari build ferla
  - Alpine er góð byrjun til að halda imageum litlum og hrað virkum
- **Build Process**
  - Skal vera eins einfalt og mögulegt er
  - Samræmt milli umhverfa (Dev, Pre-Production, Production)
  - Á **EKKI** að byggja forritunarkóða (e. compile process)
  - Á **EKKI** að keyra prófanir
  - Á ekki að keyra upp hugbúnað sem **root** notanda heldur skal stofna non-root notanda í ferlinu og keyra hugbúnað á honum
  - Notast skal við tags í samræmi við útgáfustjórnun embættis landlæknis Sjálfvirkir gæða- og útgáfuferlar Landlæknis | Skipulag útgáfunúmera

  - Docker skráin skal innihalda athugasemdir og útskýringar á ensku

- **Uppbygging skráa**
  - Eru einhverjar möppur sem samnýtast milli imagea?
  - Hvar eru loggar geymdir?
- **Netkerfi**
  - Skilgreina skal hvaða port þurfa að vera opin og hvaða þjónustur treysti á þau
  - Almennt skal reyna að halda utan um samskipti innan containera og eins fá port opnuð eins og hægt er
  - Fylgja skal Topology teikning af netkerfi ef notast er við docker networking
- **Startup**
  - Í Readme skrá skal tekið fram hvað þarf að gera til að keyra upp kerfið
  - Hvernig á að byggja upp image
  - Dæmi um command line keyrslu
  - Notast skal við Docker compose yaml skrár ef það á við.
- **Hvað má ekki**
  - Ekki byggja forritunarkóða
  - Ekki keyra prófanir
- **Secrets**
  - Hvernig appsettings.json og slíkar skrár eru stilltar
  - Hvernig github secrets eru notuð í buildpípum til að populatea config skrár
  - Lykilorð eiga ekki að sjást í clear text í loggum eða með “env” commandi ef tengst er inn á container

Kubernetes

- Hýsingarumhverfi embættis landlæknis notast við Argo

- Allar hugbúnaðarlausnir embættis landlæknis þurfa að geta keyrt í kubernetes (undanskilið þessu eru gagnagrunnar)
- Þjónustur skulu innihalda viðeigandi health-checks til þess að hægt sé að nýta sjálfvirknivæðingu í Kubernetes

Rekstur og eftirlit

Mælikvarðar og sjálfvirknivæðing

> 🌞 **Leiðarljós**
>
> Mikilvægt er að eftirlit með stöðu og afköstum hugbúnaðarlausna embættisins sé gagnsætt og staðlað. Slíkt stuðlar bæði að rekstrarhagræðingum og sjálfvirknisvæðingu. Af þessum sökum er mikilvægt að lausnir safni saman og skili af sér á staðlaðan hátt mæligögnum, keyrslustöðu og villuskrám sem geri bilanaleit og rekstrarvöktun bæði skilvirka og einsleita. Einnig er mikilvægt að kerfisstjórar og rekstraraðilar hafi góðan skilning á innviða og tengiþörfum hugbúnaðarins.

Mælingar

Öll kerfi skulu halda lifandi skrá um núverandi keyrsluafköst þess. Staðallinn sem notast skal við er Prometheus (prometheus.io).

Þjónustur þurfa að útfæra eftirfarandi endapunkt til að birta mælingarniðurstöður

- /metrics
    - Birtir núverandi stöðu á Prometheus mælingum þannig að hægt sé að skrapa þá.

Mælingarendapunktar ættu að vera aðgengilegir á öðru porti (9102) en aðrir almennir endapunktar til að auðveldara sé að loka á utanaðkomandi aðgang að þeim í eldveggjum.

Keyrslustaða

Þjónustur þurfa að útfæra eftirfarandi endapunkta til að tryggja að viðeigandi sjálfvirknivæðing sé möguleg innan Kubernetes umhverfisins

- /health/alive
    - Er þjónustan í gangi og hefur ekki hrunið eða er í lás (e. Deadlock). Ef þetta tékk virkar ekki þá mun Kubernetes endurræsa þjónustuna sjálfkrafa.

- /health/ready
    - Er þjónustan tilbúin til að taka við beiðnum, þ.e.a.s. nær þjónustan sambandi við aðrar þjónustur og hefur allar upplýsingar sem þarf til að byrja að fá ytri traffík til meðhöndlunar. Ef þetta tékk virkar ekki þá mun Kubernetes ekki senda traffík á þjónustuna (en þjónustan er skilin eftir í gangi).
- /health/startup
    - Er hýsingareiningin tilbúin, fortékk fyrir alive tékkið og er oft útfært fyrir utan þjónusturnar sjálfar þar þetta próf er meira á eininga leveli (e. Container level) heldur en application leveli.

Útgáfuupplýsingar

Þjónustur þurfa að útfæra eftirfarandi endapunkt sem notaður verður í sjálfvirka útgáfuferla og sannreyningarferla.

- /version
    - Notaður til að sannreyna hvaða útgáfa er á hvaða umhverfi til að styðja við sjálfvirkni í útgáfu á þjónustum
    - Birtir útgáfu upplýsingar og upplýsingar um þann feril sem notaður var til að gefa út þessa útgáfu á umhverfinu.

      t.d.

```
{
  "commit_sha": "c9a911a85e0b29c4f08bdf547e37376d800981c1",
  "commit_workflow_id": "304328",
  "version": "2024.08.10"
}
```

Logging (upplýsinga og villuskráning)

Notast skal við structured logging á JSON formi. Logging skal fylgja Elastic Common Schema (https://www.elastic.co/guide/en/ecs/current/ecs-reference.html) í grunninn og bæta við þeim viðbótum sem skilgreind eru af embættinu.

Leitast skal við að kerfi ætlað er að keyri í kubernetes og eða azure, skrái logga út í console glugga en skrifi ekki logga út í skrár.

Viðbótar svæði

ECS Schema	Lýsing