



Þróunarhandbók

Leiðarvísir fyrir þróun
stafrænna lausna fyrir
íslenskar
heilbrigðisstofnanir

Útgefandi:

Heilbrigðisráðuneytið | Stafræn Heilsa

Þróunarhandbók

Skrifuð af Sverri Sigmundarsyni, Einari Loga Einarssyni og

Ólafi Kristjáni Ragnarsyni hjá Stafrænni Heilsu.

4. ÚTGÁFA | JÚLÍ 2026

sh@hrn.is

<https://stafrænheilsa.is>

Umbrot og textavinnsla:

Heilbrigðisráðuneytið

©2026 Heilbrigðisráðuneytið

ISBN 978-9935-515-94-0

Þessi þróunarhandbók skilgreinir kröfur, verklag og leiðbeiningar sem snúa að þróun, gæðaeftirliti og útgáfu hugbúnaðar fyrir verkkaupa.

Handbókin er hluti af heildstæðu handbókakerfi og skal lesa hana í samhengi við:

- Tæknistefnu verkkaupa
- Tækniröfur verkkaupa
- Rekstrarhandbók verkkaupa
- Hönnunarhandbók verkkaupa

Ef árekstur verður milli skjala gilda tæknistefna og tækniröfur um tæknilegt val og tæknilegar afstöður. Um þróunarferla, gæðakröfur og skil gildir þróunarhandbók þessi.

Allir sem lesa þennan texta eru hvattir til að hugsa gagnrýnið um efni skjalsins og koma með tillögur að úrbótum til verkefnastjóra.

Efnisyfirlit

1 Fyrirvari	13
2 Breytingar frá fyrri útgáfu	13
3 Áskorunin okkar	14
3.1 Framtíðartryggðar tækniákvarðanir	14
3.2 Meginreglur handbókarinnar	15
3.3 Lögum og fyrirmælum um skráningu í sjúkraskrá framfylgt	15
3.4 Hugbúnaður sem lækningatæki	17
3.4.1 Flokkunarferli	17
3.4.2 Áhrif ef hugbúnaður telst lækningatæki	17
4 Kóðageymslur	19
4.1 Allur kóði er geymdur í git kóðageymslum	19
4.2 Skipulag kóðageymsla	19
4.2.1 Rökstuðningur	19
4.3 Nafnareglur á kóðageymslum	20
4.4 Viðskeyti á kóðageymslur	20
4.5 Hvernig ytri aðilar vinna með kóðageymslurnar	21
4.6 Nafnareglur á kóðagreinum	21
5 Viðtöku- og útgáfufarlar	22
5.1 Viðtökuferill verkkaupa	22
5.1.1 Sjálfvirkt gæðamat	22
5.1.2 Kóðarýni og gæðamat	22
5.1.3 Ferli viðtöku og samskipti	22

5.2 Útgáfuferill.....	23
6 Gæðaferlar.....	23
6.1 Ábyrgðaskipting.....	24
6.2 Sjálfvirkir gæðastýringarferlar.....	24
6.3 Hvað er prófað sjálfvirkt.....	24
6.4 Gæðahlið (Quality Gates).....	24
6.5 Gervigreind í gæðaeftirliti.....	24
6.6 Ábyrgð verksala á innri gæðum.....	25
7 Skil á hugbúnaðarafurðum.....	25
7.1 Skilastefna.....	25
7.2 Hverju á að skila?.....	25
7.3 Hver skilar?.....	26
7.4 Hvenær er æskilegt að skila inn?.....	26
7.5 Rekjanleikakröfur í skilum.....	26
7.6 Undantekningar frá skilareglum.....	27
8 Forritunarmál og tæknival.....	27
8.1 Bakendakerfi og API.....	28
8.1.1 Undanþágur.....	28
8.2 Framendar / Vef.....	28
8.2.1 Leið A, ASP.NET Server-Side Rendering.....	28
8.2.2 Leið B, Gervigreindarsmiðaður React framendi.....	29
8.2.3 Óheimilar framendatæknilausnir.....	33
8.2.4 JavaScript / TypeScript, sameiginlegar reglur.....	34
8.3 Hönnunarstaðlar.....	34
8.4 Snjallforrit / Mobile.....	34
8.4.1 Rökstuðningur.....	34
8.4.2 Cross-platform lausnir (React Native, Flutter, .NET MAUI o.p.h.).....	35
8.4.3 Hvenær native snjallforrit eru heimil.....	35
8.4.4 Aðgengi og útbreiðsla.....	36
8.4.5 Afmörkun gagnvart SPA.....	36
9 Hönnun og arkitektúr.....	36
9.1 Meginreglur.....	36

9.2 Þjónustuhögun.....	37
9.2.1 Rökstuðningur	37
9.2.2 Arkitektúrhögun þjónusta.....	37
9.2.3 Bakgrunnsvinnslur.....	37
9.2.4 Caching.....	38
9.3 Samskipti við ytri þjónustur og samvirkni.....	38
9.3.1 OpenAPI og forritunarskil.....	38
9.3.2 Gagnasnið.....	38
9.3.3 Heilbrigðisstaðlar	38
9.3.4 Nafnareglur þjónusta.....	38
9.4 Þolgæði.....	39
9.5 Lagskipting og skipulag lausna.....	39
9.5.1 Af hverju aðskilin verkefni (.csproj) og möppuskipan, ekki nafnabil (namespaces)?	39
9.5.2 Af hverju möppuskipan fremur en nákvæm tæknileg skjölun eingöngu?.....	39
9.5.3 Af hverju samræmd möppunöfn milli kóðageymsla?	39
9.5.4 Af hverju þriggja laga aðskilnaður?.....	40
9.5.5 Leið B, framendi.....	40
9.5.6 Meginreglur	41
10 Forritunarkóði og vinnureglur	41
10.1 Kóðastíll og sjálfvirk samræming.....	41
10.1.1 Umfang breytinga í PR	41
10.1.2 Nafnarými og pakkar	41
10.2 Virknisflögg (Feature flags).....	41
10.2.1 Kröfur um útfærslu.....	41
10.2.2 Nafnareglur.....	42
10.2.3 Takmarkanir	42
10.2.4 Hreinsun.....	42
10.3 Þýðingar	42
10.3.1 Umfang.....	42
10.3.2 Staðfærsla.....	42
10.3.3 Þýðingarlyklar.....	42
10.3.4 Texti í myndum.....	42

10.4 Skjölun í forritunarkóða (comments).....	42
10.4.1 Tungumál og málfar.....	42
10.4.2 Hvað skal skjalfesta.....	42
10.4.3 Umfang.....	43
10.5 Endurnýting kóða / forritunarsöfn.....	43
10.5.1 Notkun forritunarsafna verksala.....	43
10.5.2 Ytri forritunarsöfn eða gagnapakkar.....	43
11 Gervigreind í hugbúnaðarþróun	43
11.1 Gervigreindarstudd hugbúnaðarþróun.....	44
11.1.1 Væntingar verkkaupa	44
11.1.2 Skipulögð notkun og leiðbeiningakerfi (.agent/ mappa).....	44
11.1.3 Kröfur til verksala.....	44
11.1.4 .agent/ og Leið B framendaverkefni.....	45
11.1.5 Eftirlit með gervigreindarkóða.....	45
11.1.6 Ábyrgð og mannleg yfirumsjón	45
11.1.7 Gagnaöryggi og trúnaður	45
11.1.8 Gagnsæi og upplýsingaskylda.....	46
11.1.9 Gervigreindardröfn rýnisniðmát verkkaupa.....	46
11.1.10 Æskilegt gervigreindarþróunarferli	46
11.2 Gervigreindarlausnir sem hluti af afurð	47
11.2.1 Grunnreglur	47
11.2.2 Gildissvið.....	47
11.2.3 Samþykkt forritunarsafn	47
11.2.4 Kröfur í stuttu máli	48
11.2.5 Takmarkanir.....	48
12 Öryggi og auðkenningar	48
12.1 Almennar öryggiskröfur	48
12.2 Auðkenningar	49
12.3 Aðgangsheimildir (Authorization).....	49
12.3.1 Meginreglur	49
12.3.2 Rökstuðningur.....	49
12.3.3 Nafnasnið heimilda.....	50

12.3.4 Óheimil högun	50
12.4 Öryggiskröfur framendar	50
12.5 Öryggiskröfur bakendaþjónustur	51
12.6 Öryggiskröfur ytri forritunarsöfn	51
12.7 Öryggiskröfur gagnagrunnar	51
13 Hugbúnaðarprófanir	51
13.1 Ábyrgð og umhverfi	51
13.1.1 Staðsetning prófana	52
13.1.2 Samhengi kröfu og prófana	52
13.2 Sjálfvirkar prófanir	52
13.2.1 Einingaprófanir (Unit tests)	52
13.2.2 Rökstuðningur þekjumarka einingaprófana	52
13.2.3 Samþættingarprófanir (Integration tests)	53
13.2.4 Sjálfvirkar viðmótsprófanir (UI tests)	53
13.2.5 Aðgengisprófanir	53
13.3 Handvirkar prófanir	53
13.4 Samþykktarprófanir	53
13.4.1 Prófunarskýrsla	53
13.5 Gögn sem fylgja afhendingu	54
14 Aðgengismál	54
14.1 WCAG staðallinn	54
14.1.1 Lágmarkskröfur, allar lausnir	54
14.1.2 Viðbótarkröfur, valin AAA skilyrði	54
14.2 WAI-ARIA staðallinn	55
14.2.1 Lausnir ætlaðar almenningi	55
14.2.2 Lausnir ætlaðar heilbrigðisstarfsfólki	55
14.3 Prófanir á aðgengi	55
15 Hugbúnaðarskjölun	55
15.1 Almennar kröfur	55
15.1.1 Snið og geymsla	56
15.2 Skjölun sem skilað er við afhendingu	56
15.2.1 Arkitektúr og kerfishönnun	56

15.2.2 Hönnunarákvarðanir (Architecture Decision Records, ADR).....	56
15.2.3 Flæðislýsingar (Sequence diagrams).....	56
15.2.4 Rekstrarskjölun.....	56
15.3 Skjölun sem verður að vera í kóðageymslum.....	57
15.3.1 README.md í rót kóðageymslu.....	57
15.3.2 Uppsetningarleiðbeiningar (í README eða sérstakri skrá).....	57
15.4 Viðkvæm gögn	57
15.5 Ábyrgð	57
15.6 Skjölun í forritunarkóða.....	57
16 Geymsla og miðlun gagna	58
16.1 Meðhöndlun heilbrigðisgagna	58
16.2 Ábyrgð verksala.....	58
16.3 Miðlun og deiling gagna.....	58
16.4 Dulkóðun gagna	58
17 Gagnagrunnar	59
17.1 Hlutverk gagnagrunna	59
17.2 Tegund gagnagrunna	60
17.3 Forritunarpakkar (ORM).....	60
17.4 Aðgangsréttindi gagnagrunnsnotenda.....	60
17.5 Bannaðir gagnagrunnshlutir	60
17.6 Rekjanleiki eyðingar	60
17.7 Leyfilegir gagnagrunnshlutir	61
17.8 Undanþágur á gagnagrunnsvirkni	61
17.9 Surrogate / Technical keys.....	61
17.10 Útgáfustjórnun gagnagrunna	61
17.11 Migration-ferlar:	61
17.12 Skipulag gagnagrunnskóða.....	61
18 Útgáfufurli og keyrslustýring	62
18.1 Ábyrgðarskipting.....	62
18.2 Almennar kröfur til hugbúnaðar	62
18.3 Docker (einingaumhverfi).....	63
18.4 Kubernetes	63

18.5 Geymsla leyndarmála.....	64
19 Rekstur og eftirlit.....	64
19.1 Mælingar (Metrics).....	64
19.1.1 Lágmarks mæliflokkur.....	64
19.2 Keyrslustaða (Health endpoints)	64
19.3 Útgáfuupplýsingar (Version endpoint).....	65
19.3.1 Svæði í /version svari	65
19.4 Logging (atvikaskráning).....	65
19.5 Rekjanleiki milli þjónusta.....	65
19.6 Stöðluð viðbótarsvæði.....	65
19.7 Bann og takmarkanir.....	66
20 Viðauki: Kröfulistí þróunarhandbókar.....	66
20.1 Kóðageymslur	66
20.2 Viðtöku- og útgáfuférlar.....	66
20.3 Gæðaférlar	66
20.4 Skil á hugbúnaðarafurðum	67
20.5 Forritunarmál og tæknival	67
20.5.1 Bakendi og API.....	67
20.5.2 Framendi, Leið A: ASP.NET Server-Side Rendering.....	67
20.5.3 Framendi, Leið B: Gervigreindarsmiðaður React framendi.....	67
20.5.4 Óheimilar framendatæknilausnir (gildir fyrir báðar leiðir A & B).....	68
20.5.5 Aðrar kröfur.....	68
20.6 Hönnun og arkitektúr.....	68
20.6.1 Leið A og bakendi (allar leiðir).....	68
20.6.2 Leið B framendi	69
20.7 Forritunarkóði og vinnureglur.....	69
20.8 Gervigreind í hugbúnaðarþróun	69
20.9 Öryggi og auðkenningar	70
20.10 Hugbúnaðarþrófanir.....	70
20.11 Aðgengismál	70
20.12 Hugbúnaðarskjölun.....	71
20.13 Geymsla og miðlun gagna	71

20.14 Útgáfuferli og keyrslustýring.....	71
20.15 Rekstur og eftirlit.....	72
20.16 Lækningatæki.....	72
21 Viðauki: Undanþáguferrill frá stefnu verkkaupa	72
21.1 Hvenær þarf undanþágu?.....	73
21.2 Kröfur til undanþágubeiðni.....	73
21.3 Samþykktarferli	74
21.4 Ábyrgð	74
21.5 Endurskoðun og endurmat.....	74
22 Viðauki: Yfirferðarskjal kóðarýni	74
22.1 Snið yfirferðarskjals.....	74
22.1.1 Auðkenning og flokkun.....	74
22.1.2 Athugasemdir og tillögur verkkaupa.....	75
22.1.3 Svörun og samskipti.....	75
22.1.4 Tímamat og lagfæringartillögur	75
22.2 Flokkunarkerfi.....	75
22.3 Alvarleikastig.....	75
22.4 Vinnulag við yfirferð.....	76
22.5 Gervigreindarstuðningur í kóðarýni.....	76
22.6 Dæmi úr yfirferðarskjali	76
22.6.1 Dæmi 1: Org-Policy / Critical.....	77
22.6.2 Dæmi 2: Security-Standard / Critical.....	77
22.6.3 Dæmi 3: Best-Practice / High.....	77
22.7 Varðveisla og rekjanleiki.....	77
23 Viðauki: Gátlistar við verkefnastýringu	78
23.1 Gátlisti 1: Áður en forritun hefst	78
23.1.1 Lög, heimildir og forsendur.....	78
23.1.2 Hönnun og arkitektúr.....	78
23.1.3 Gagnagrunnar og gögn.....	78
23.1.4 GitHub og skipulag.....	78
23.1.5 Öryggi.....	78
23.1.6 Gervigreindarstudd þróun.....	78

23.2 Gátlisti 2: Meðan á forritun stendur	79
23.2.1 Kóðauppbygging.....	79
23.2.2 Gæðastýring	79
23.2.3 Prófanir.....	79
23.2.4 Sjálfvirknivæðing.....	79
23.2.5 Öryggi meðan á þróun stendur	79
23.3 Gátlisti 3: Fyrir skil og QA.....	79
23.3.1 Pull Requests.....	79
23.3.2 Kóðarýni	79
23.3.3 Skjölun við skil.....	79
23.4 Gátlisti 4: Fyrir útgáfufarla.....	80
23.4.1 Almennt	80
23.4.2 Docker	80
23.4.3 Kubernetes.....	80
23.4.4 Gagnagrunnsbreytingar	80
23.5 Gátlisti 5: Fyrir keyrslu, mælingar og logging.....	80
23.5.1 Mælingar	80
23.5.2 Keyrslustaða.....	80
23.5.3 Útgáfuupplýsingar	80
23.5.4 Logging.....	80
23.6 Gátlisti 6: Við verklok (tæknileg atriði).....	81
23.6.1 Skjölun	81
23.6.2 Afhending.....	81
23.6.3 Staðfesting.....	81
24 Viðauki: Þjónustur og verkfæri verkkaupa.....	81
24.1 Staða og bindandi gildi	81
24.1.1 Kóðageymslur og útgáfa	81
24.1.2 Gæðaeftirlit og kóðagreining.....	81
24.1.3 Gervigreind í gæðaeftirliti.....	82
24.1.4 Gervigreindarforritunarsöfn fyrir afurðapróun.....	82
24.1.5 Keyrsluumhverfi	82
24.1.6 Vöktun og rekstrareftirlit.....	82

24.1.7 Hönnun og verkefnastýring.....	82
24.2 Athugasemdir.....	82

Myndaskrá

Mynd 1 Heildaryfirlit þróunar- og útgáfufurils.....	18
Mynd 2 Afhendingarferill í smáatriðum, sem sýnir hvernig gervigreindardrifið rýniferli og samskipti verkefnastjóra beggja aðila virka í framkvæmd.....	23
Mynd 3 Útgáfufurill í smáatriðum, sem sýnir flæði milli QA og PROD umhverfa og handvirkt samþykktarskref.....	23
Mynd 4,„Rekjanleiki skila frá QA til PROD" og sjá viðauka „Rekjanleiki milli kóðagreina, breytingabeiðna og útgáfa" fyrir nánari lýsingu.....	27

1 Fyrirvari

Þróunarhandbók þessi er eign verkkaupa og er frekari birting óheimil nema með samþykki.

Kröfur sem settar eru fram í þessari handbók eru bindandi í útboðum, samningum og verkefnum þar sem handbókin er hluti af samningsgögnum. Frávik frá bindandi kröfum eru einungis heimil í samræmi við skilgreindan undanþágufेरil (*sjá viðauka Undanþágufेरill frá stefnu verkkaupa*).

Lausnir í lausnabanka verkkaupa eru birtar „eins og þær koma fyrir“ (as-is), án ábyrgðar á réttlæika eða hæfi til notkunar í tilteknu verkefni. Ekki er ætlast til að kóði úr lausnabanka sé notaður óbreyttur eða órýndur í lausnir verksala, sama gildir um fyrir fram tilbúna forritunarlausnir sem verksali leggur til verksins. Allur kóði sem er hluti af skilum fellur undir sömu kröfur og annar hugbúnaðarkóði, óháð uppruna.

Hvers kyns notkun á lausnum eða tillögum úr þessari handbók eða lausnabanka, eins og þær koma fyrir, er á ábyrgð verksala og verkkaupa að skaðlausu

2 Breytingar frá fyrri útgáfu

Nýtt efni

- Skjalinu skipt í tvö sjálfstæð skjöl: meginskjal (þróunarhandbók) og viðaukar (tæknilegt viðhengi).
- Nýr viðauki: Útboðskröfur þróunarhandbókar (SKAL/GETUR), samantekt allra bindandi krafna og ráðlegginga í útboðshæfu sniði.
- Nýr viðauki: Undanþágufेरill frá stefnu verkkaupa, formlegur ferill fyrir frávik frá tæknistefnu.
- Nýr viðauki: Yfirferðarskjal kóðarýni, snið, flokkunarkerfi og dæmi fyrir formlega kóðarýni.
- Nýr viðauki: Gátlistar við verkefnastýringu, sex gátlistar fyrir mismunandi stig verkefnis.
- Nýr viðauki: Þjónustur og verkfæri verkkaupa, listi yfir samþykktar þjónustur, aðskilinn frá meginmáli.
- Nýr kafli (11): Gervigreind í hugbúnaðarþróun, A) gervigreindarstudd þróun og B) gervigreindarlausnir sem hluti af afurð. Kafli endurskrifaður frá grunni.
- Nýr undirkafli: .agent/ mappa og Agent OS v3.0 staðall sem hluti af kóðageymslum.
- Nýr undirkafli: Gervigreindardrífín rýnisniðmát verkkaupa.
- Nýr undirkafli: Viðmiðunararkitektúr og þjónustuhögun (kafli 9).
- Nýr undirkafli: Bakgrunnsvinnslur sem Kubernetes CronJobs (kafli 9).
- Nýr undirkafli: Gæðahlið (Quality Gates) í gæðafærlum (kafli 6).
- Nýr undirkafli: Viðskeyti á kóðageymslum (kafli 4).
- Ábyrgðartafla fyrir lagafylgni (Lögum og fyrirmælum um skráningu í sjúkraskrá) bætt við.
- Meginreglur handbókarinnar settar fram sem töflu.
- Matsþáttar tækniþvalsins (bakhjarl, þroski, mannaflí) settir fram sem töflu í Áskorunin okkar.
- Heildaryfirlit þróunar- og útgáfufेरils (Mynd 1) bætt við.
- Ný röð í lagafylgnitöflu: Lög nr. 132/2020 (lækningatæki) og reglugerð (ESB) 2017/745.
- Nýjar raðir í ábyrgðarskiptingartöflu vegna lækningatækjamats.
- Nýr undirkafli: Hugbúnaður sem lækningatæki, flokkun og ferli.
- Nýr hluti í Gátlista 1: Atriði tengd lækningatækjaflokkun.
- Nýr hluti í Viðauka útboðskrafna: P16 (Lækningatæki).
- Nýr viðauki: Hugbúnaður sem lækningatæki (SaMD/MDSW), viðmiðunarskjal um regluverk, staðla, flokkun og samræmismatsleið.
- Ný röð í lagafylgnitöflu §A3: Reglugerð (ESB) 2024/1689 (EU AI Act), tilvísun í tæknistefnu.

Breytingar

- Handbókin notar „verkkaupa“ / „verksala“ í stað tilvísana í sérstakar stofnanir (t.d. embætti landlæknis, MRH).
- Handbókakerfi: Skýr stigveldi skjala kynnt, tæknistefna, rekstrarhandbók, hönnunarhandbók og þróunarhandbók, ásamt árekstrarreglu.
- Fyrirvari endurskrifaður: Bindandi gildi krafna skýrt, aðskilið frá as-is fyrirvara lausnabanka. Ný regla um kóða úr lausnabanka og fyrirfram tilbúna forritunarlausnir verksala.
- Forritunarmál og tæknival (kafli 8) nú sjálfstæður kafli; C#/ .NET/ASP.NET krafa gerð afdráttarlaus. SPA bann, JS/TS sem burðarás óheimilt, WebForms óheimilt.
- Framendi: ASP.NET MVC / Razor Pages + Server-Side Rendering sett sem eina samþykkt framendanálgunin. HTMX heimilt sem hjálpartæki en ekki burðarás.
- Hönnun og arkitektúr (kafli 9): Smáþjónustur (micro/nano-services) gerðar óheimilar. Hangfire og in-process job schedulers gerðir óheimilir. REPR-högun og vertical slice bætt við sem ráðlögð nálgun.
- Viðtöku- og útgáfufەرlar (kafli 5) aðskilnir frá gæðafærlum sem sjálfstæður kafli.
- Gæðafەرlar (kafli 6): Gervigreind í gæðaeftirliti bætt við. Ábyrgðarskipting skýrð.
- Skil á hugbúnaðarafurðum (kafli 7): Rekjanleikakröfur í skilum rýmkaðar (trace frá kóðagrein til útgáfu).
- Öryggi (kafli 12): MFA krafa, JWT kröfur, claims-based authorization (ekki role-based), SRI, throttlng kröfur og bann við ytri söfnum sem bæta sjálfvirk við endapunktum bætt við.
- Prófanir (kafli 13): ≥80% þekja á nýjum kóða og ≥65% heildarþekja gerðar bindandi. Aðgengisprófanir sem hluti af CI bætt við.

- Aðgengismál (kaflí 14): 15 valin WCAG AAA skilyrði bætt við umfram AA. Skipting milli lausna fyrir almenning og heilbrigðisstarfsfólk innleidd.
- Hugbúnaðarskjölun (kaflí 15): /Docs möppuskipulag og README.md kröfur skýrðar. ADR og sequence rit gerð að skilakröfu.
- Geymsla og miðlun gagna (kaflí 16): Kafla skipt í A (meðhöndlun og vernd) og B (gagnagrunnar). Dapper.NET staðfest sem eina heimila ORM; Entity Framework, Hibernate o.þ.h. gerð óheimil. Stored procedures með rökfræði, triggers og scheduled jobs gerðir óheimilir.
- Útgáfufurli (kaflí 18): Docker build krafa skýrð (engin þýðing/bygging inni í Docker). GitOps og Argo CD samhæfni gerð bindandi.
- Rekstur og eftirlit (kaflí 19): ECS-samhæfð viðbótarsvæði bætt við. Stöðluð logunarforritunarsöfn verkkaupa gerð að bindandi kröfu.
- Meginmál stytur: Ítarleg tækniskjölun, kóðadæmi og stillingarleiðbeiningar fluttar í viðaukaskjal.
- Hanbók var sett í sniðmát HRN
- Tilvísanir í kafla í texta voru lagaðar eftir færslu yfir í HRN sniðmát.

Fjarlægt efni

- Hálfsjálfvirkir gæðastýringarferlar fjarlægðir (nú fullsjálfvirkt).
- Sjálfvirkir útgáfufurli og umhverfislýsingar fluttar í viðaukar.
- Skipulag útgáfunúmera flutt í viðauka.
- Ítarleg kóðadæmi og sniðmát flutt í viðauka.
- Tilvísanir í sérstakar stofnanir fjarlægðar (verkkaupa abstraction).

3 Áskorunin okkar

Grunnforsenda handbókarinnar er langtímahugsun. Hugbúnaðarlausnir sem þróaðar eru fyrir verkkaupa eru ætlaðar til reksturs og viðhalds í tuttugu til þrjátíu ár eða lengur. Á þeim tíma er gert ráð fyrir að margir og mismunandi aðilar komi að viðhaldi og viðbótum og að verulegar tæknibreytingar eigi sér stað í umhverfi lausnanna.

Því leggur handbókin áherslu á:

- framtíðartryggðar tækniávarðanir,
- viðhaldsþólinn hugbúnaðararkitektúr,
- skýra og einfalda sjálfvirknivæðingu,
- kjarnyrta skjölun,
- fylgni við lög og fyrirmæli um skráningu heilbrigðisupplýsinga.

Hvert þessara atriða er útfært nánar í viðeigandi köflum handbókarinnar. Í þessum kafla er lagt fram það mat og þau sjónarmið sem liggja til grundvallar tæknivali og ákvörðunum.

3.1 Framtíðartryggðar tækniávarðanir

Ómögulegt er að spá fyrir um tæknilega þróun til langs tíma. Hins vegar er bæði mögulegt og nauðsynlegt að taka upplýstar, íhaldssamar og vel rökstuddar tækniávarðanir sem lágmarka líkur á ótímabærri úreldingu, endurteknum endursmíðum og ófyrirsjáanlegum viðhaldskostnaði.

Þessu til viðbótar þarf að taka tillit til fjármögnunarraunveruleika opinberra verkefna. Hugbúnaðarverkefni verkkaupa eru að jafnaði fjármögnuð í upphafi með afmörkuðu fjármagni sem nær til hönnunar, smíði og innleiðingar. Reglulegt viðhaldsfjármagn eftir innleiðingu er oft takmarkað eða ekki tryggt. Tæknival skal því lágmarka viðhald sem tengist ekki beint viðskiptaþörfum, sérstaklega tæknilega endurnýjun sem knúin er af breytingum á undirliggjandi forritunarsöfnum (e. framework churn) frekar en viðskiptalegum kröfum.

Við val á tækni skal meta eftirfarandi þrjá þætti. Matið skal vera hluti af hönnunargögnum og skjalfest í arkitektúrákvörðunum (ADR).

Matsþáttur	Lykilspurningar	Áhættuvísar
Bakhjarl, Hver stendur að baki tækninni og hversu traustur er sá bakhjarl til langs tíma?	Er tæknin studd af rötgrónum bakhjarli? Hefur bakhjarlinn sögu um langtímastuðning (LTS)? Er tæknin kjarnaafurð eða jaðarverkefni?	Fámennt áhugasamfélag, sprotafyrirtæki án rekstrarsögu, óstöðug útgáfustefna

Þroski , Er tæknin nægilega þroskuð og hönnuð með stöðugleika að leiðarljósi?	Hefur tæknin verið í virkri notkun í mörg ár? Er hún líkleg til að vera nothæf eftir 5, 10 og 20 ár? Samræmist hún núverandi tækniumhverfi verkkaupa? Fellur hún að sjálfvirkum gæðaferlum?	Tíðar stórar útgáfur án LTS, takmarkað stuðningsumhverfi, léleg samhæfni við tól og greiningartæki
Mannafli , Er hægt að finna hæft fólk til að viðhalda lausninni til langs tíma?	Er þekking á tækninni útbreidd á Íslandi? Er tæknin aðgengileg óreyndari forriturum? Er auðvelt að fá verktaka, bæði innanlands og erlendis? Er til mikið af opinberri skjölun og dæmum?	Mjög sérhæfð tækni, fáir sérfræðingar, lítil opinber þekking
Viðhaldsbyrði , Hversu mikils viðhalds krefst tæknin sem er ótengd viðskiptaþörfum?	Hversu oft krefst tæknin uppfærslu á grunnkóða? Er langtímastuðningur (LTS) í boði? Krefst uppfærsluferill reglulegrar endursmíði? Er afturvirk samhæfni tryggð milli útgáfa?	Tíðar stórar útgáfur sem brjóta samhæfni, engin LTS stefna, stutt stuðningstímabil, sögulegur ferill um framework churn

Tæknival sem standast ekki þetta mat eru almennt óheppileg fyrir lausnir verkkaupa og verða ekki samþykkt nema rökstudd undanþága liggi fyrir.

3.2 Meginreglur handbókarinnar

Eftirfarandi meginreglur liggja til grundvallar öllum kröfum í þessari handbók:

Meginregla	Lýsing	Nánar í kafla
Viðhaldspolinn arkitektúr	Arkitektúr lausna skal vera skýr, lagskiptur og auðveldur í viðhaldi af nýjum aðilum. Lausnir skulu fylgja viðurkenndum högunarmynstrum (SOLID, ONION, CLEAN, KISS, DRY, YAGNI).	Forritunarkóði
Einföld sjálfvirknivæðing	Bygging, prófanir, útgáfa og eftirlit skulu vera sjálfvirkt og endurtekningarhæft. Sjálfvirknivæðing dregur úr mannlegum mistökum og tryggir rekjanleika.	Gæðaferlar, Útgáfuferli
Kjarnyrt skjölun	Skjölun skal lýsa högun, ákvörðunum og forsendum, ekki endursegja almennt efni. Markmið er að nýr forritari geti hafið vinnu án óþarfa tafar.	Hugbúnaðarskjölun
Lagafylgni	Lausnir skulu uppfylla lög um sjúkraskrár, reglugerðir og fyrirmæli um skráningu heilbrigðisupplýsinga. Engin undanþága er heimil á grundvelli tæknilegrar einföldunar.	Geymsla og miðlun gagna

3.3 Lögum og fyrirmælum um skráningu í sjúkraskrár framfylgt

Allar hugbúnaðarlausnir sem vinna með heilbrigðisgögn skulu vera hannaðar og útfærðar í fullu samræmi við lög um sjúkraskrár, reglugerðir og fyrirmæli sem gilda um lágmarksskráningu, varðveislu og meðhöndlun heilbrigðisupplýsinga á Íslandi.

Engin undanþága frá lögum og fyrirmælum um skráningu í sjúkraskrár er heimil á grundvelli tæknilegrar einföldunar eða tímapressu.

Eftirfarandi lög og reglugerðir gilda sérstaklega um hugbúnaðarlausnir verkkaupa og skulu vera höfð til hliðsjónar við hönnun og útfærslu:

Lög / Reglugerð	Áhrif á hugbúnaðarþróun
Lög nr. 90/2018 (persónuvernd, GDPR)	Persónuupplýsingar skulu flokkaðar, vinnsla skráð, gagnavinnslusamningur til staðar þegar þriðji aðili vinnur gögn
Lög nr. 77/2014 (opinber skjalasöfn)	Eyðing gagna óheimil nema heimild liggi fyrir. Lausnir skulu hannaðar þannig að gögn séu merkt sem óvirk frekar en þeim eytt (sjá nánar hér að neðan).
Lög nr. 78/2019 (öryggi mikilvægra innviða)	Lausnir sem teljast hluti af mikilvægum innviðum skulu uppfylla sérstakar öryggiskröfur

Reglugerð nr. 959/2012 (trúnaðarupplýsingar)	Meðferð trúnaðarupplýsinga skal vera í samræmi við reglugerð og tengd lög
Upplýsingalög nr. 140/2012	Gagnaflokkun skal styðja gagnsæi, aðgengi almennings og traust á opinberri stjórnsýslu
Lög um sjúkraskrár	Lágmarksskráning, varðveisla og meðhöndlun heilbrigðisupplýsinga samkvæmt gildandi fyrirmælum
Lög nr. 132/2020 (lækningatæki) og reglugerð (ESB) 2017/745 (MDR)	Hugbúnaður sem hefur skilgreindan læknisfræðilegan tilgang getur fallið undir lög um lækningatæki og þarf þá að uppfylla sértækar kröfur um gæðakerfi, hugbúnaðarlíftíma, áhættustýringu og CE-merkingu. Mat á flokkun (sbr. MDCG 2019-11) skal fara fram við verkupphaf og vera skjalfest. Lyfjastofnun er lögbært yfirvald. Sjá undirkafla hér á eftir.
Reglugerð (ESB) 2024/1689 (gervigreindarreglugerð, EU AI Act)	Gervigreindarkerfi í heilbrigðisþjónustu geta fallið undir hááhættuflokkun. Mat á flokkun skal fara fram við verkupphaf þar sem gervigreindarvirkni er hluti af afurð. Sjá tæknistefnu verkkaupa (kaflí 4.4 og 10.3).

Eftirfarandi ábyrgðarskipting gildir um lagafylgni í öllum verkefnum:

Aðili	Ábyrgð
Verksali	Kynna sér gildandi lög, reglugerðir og fyrirmæli um skráningu í sjúkraskrár áður en þróun hefst.
Verksali	Hanna og útfæra hugbúnað þannig að tæknileg útfærsla styðji við lögbundna skráningu, rekjanleika og ábyrgð.
Verksali	Tryggja að skráning gagna, breytinga og aðgerða sé rekjanleg og ekki sé hægt að komast hjá lögbundinni skráningu með tæknilegum leiðum.
Verksali	Skjalfesta hvernig lausnin uppfyllir kröfur laga og fyrirmæla, m.a. í hönnunargögnum, prófunum og gæðamati.
Verksali	Gera verkkaupa tafarlaust viðvart ef lagakröfur eru óskýrar, stangast á við fyrirhugaða útfærslu eða kalla á sérstakar tæknilegar ákvarðanir.
Verkkaupi	Veita upplýsingar um gildandi lög, reglugerðir og fagleg fyrirmæli sem gilda um viðkomandi lausn.
Verkkaupi	Skýra væntingar um lágmarksskráningu, rekjanleika og aðgangsstýringu þar sem við á.
Verkkaupi	Meta og samþykkja tæknilegar lausnir með tilliti til laga- og reglufylgni.
Verkkaupi	Meta við verkupphaf hvort hugbúnaður gæti talist lækningatæki samkvæmt lögum nr. 132/2020 og MDCG 2019-11. Matið skal vera skjalfest.
Verkkaupi	Ef hugbúnaður telst lækningatæki: tryggja að viðeigandi gæðakerfi (ISO 13485), hugbúnaðarlíftímaferlar (IEC 62304) og áhættustýring (ISO 14971) séu til staðar, og að CE-merking sé fengin áður en hugbúnaður er tekinn í notkun.
Verkkaupi	Tilkynna Lyfjastofnun um frávík, galla eða bilanir sem gætu valdið eða hafa valdið heilsutjóni, í samræmi við 35. gr. laga nr. 132/2020.
Verksali	Upplýsa verkkaupa tafarlaust ef verksali telur að hugbúnaður eða hluti hans gæti fallið undir skilgreiningu lækningatækis.
Verksali	Ef hugbúnaður telst lækningatæki: fara eftir þeim viðbótarkröfum sem gilda um hugbúnaðarlíftíma (IEC 62304), þar á meðal skjölun, rekjanleika og breytingastýringu sem ganga lengra en almennar kröfur þessarar handbókar.

3.4 Hugbúnaður sem lækningatæki

Hugbúnaður sem er ætlaður til læknisfræðilegs tilgangs, svo sem sjúkdómsgreiningar, vöktunar, meðferðarákvörðunar eða spár um sjúkdómsgang, getur talist lækningatæki samkvæmt lögum nr. 132/2020 og reglugerð (ESB) 2017/745 (MDR). Flokkun hugbúnaðar sem lækningatæki hefur veruleg áhrif á þróunarferli, skjölun, prófanir og rekstur og skal því metin **við upphaf verkefnis**.

3.4.1 Flokkunarferli

Við upphaf hvers verkefnis skal verkefnastjóri (eða tilnefndur ábyrgðaraðili) framkvæma mat á flokkun hugbúnaðar samkvæmt leiðbeiningum MDCG 2019-11 Rev.1:

Skref	Spurning	Ef já
1	Er afurðin hugbúnaður (vinnur úr inntaksgögnum og framleiðir úttaksgögn)?	Halda áfram í skref 2
2	Framkvæmir hugbúnaðurinn aðgerðir á gögnum umfram geymslu, samskipti, einfalda leit eða tapslausla þjöppun?	Halda áfram í skref 3
3	Eru aðgerðirnar til hagsbóta fyrir einstaka sjúklinga (ekki eingöngu rannsóknir á hópgrunni eða stjórnun)?	Halda áfram í skref 4
4	Uppfyllir hugbúnaðurinn skilgreiningu á lækningatæki skv. 4. gr. laga nr. 132/2020 (greining, forvörn, vöktun, spá, meðferð)?	Hugbúnaður telst lækningatæki

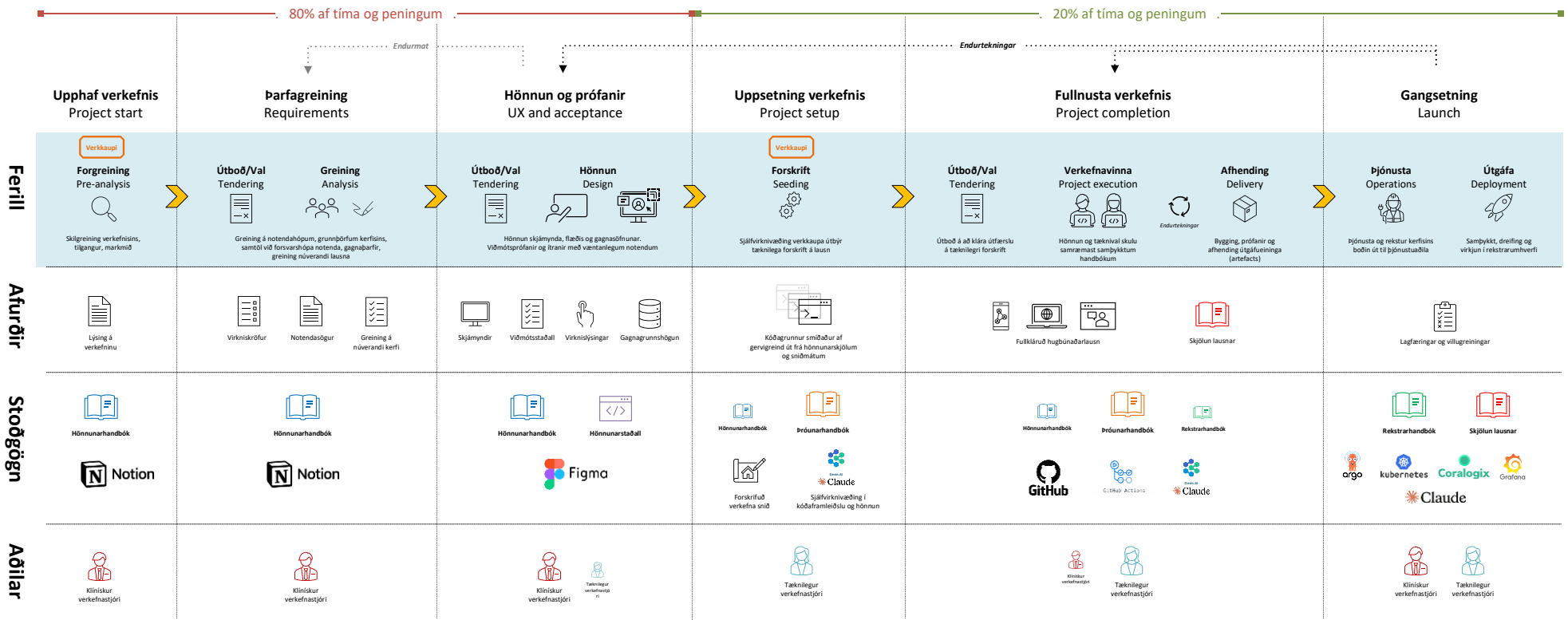
Niðurstaða matsins skal skjalfest og geymd sem hluti af hönnunargögnum verkefnisins, óháð niðurstöðu. Matið skal endurtekið ef tilgangslýsing hugbúnaðar breytist á líftíma verkefnis.

3.4.2 Áhrif ef hugbúnaður telst lækningatæki

Ef hugbúnaður er flokkaður sem lækningatæki gilda viðbótarkröfur sem ganga lengra en almennar kröfur þessarar handbókar. Helstu viðbótarkröfurnar eru:

- **Gæðakerfi** framleiðanda samkvæmt ISO 13485
- **Hugbúnaðarlíftímaferlar** samkvæmt IEC 62304, þ.m.t. öryggisflokkun hugbúnaðar og formlegur rekjanleiki frá kröfum til prófana
- **Áhættustýring** samkvæmt ISO 14971 alla líftíð tækisins
- **CE-merking** og samræmismat áður en hugbúnaður er tekinn í notkun; flokkur IIa og hærri krefst aðkomu tilkynnts aðila (Notified Body)
- **Eftirlit eftir markaðssetningu** (post-market surveillance) og reglubundnar öryggisuppfærsluskýrslur
- **Tilkynningarskylda** til Lyfjastofnunar vegna atvika (sjá hér að neðan)

Nánari lýsing á stöðlum, flokkunarreglum, dæmum og tilvísunum er í Viðauka, Hugbúnaður sem lækningatæki (SaMD/MDSW).



Mynd 1 Heildaryfirlit þróunar- og útgáfufेरils

Myndin sýnir hvernig handbækur, ferlar og þjónustur verkaupa tengjast hverju skrefi ferilsins. Athugið að þjónustur listaðar eru einungis til hliðsjónar, nákvæmt val á þjónustum verkaupa geta tekið breytingum milli ára. Nánari lýsing á afhendingarferli (Delivery) er í kafla 7. Nánari lýsing á útgáfuferli (Deployment) er í kafla 18.

4 Kóðageymslur

👉 Leiðarljós

Markmið verklagsins er að tryggja gæði, rekjanleika og öryggi í öllum kóðaskilum án þess að takmarka sjálfstæði verktaka að óþörfu.

Verktakar vinna í eigin umhverfi samkvæmt sínum verkferlum, en allar endanlegar breytingar sem fara til útgáfu skulu fara í gegnum skilgreint samþykktar- og kóðarýniferli í kóðageymslum verkkaupa.

4.1 Allur kóði er geymdur í git kóðageymslum

Allur hugbúnaðarkóði sem þróaður er fyrir verkkaupa skal geymdur í git-umhverfi verkkaupa. Verkkaupi ákvarðar hvaða git-vettvangur er notaður og hvernig aðgangi er háttað.

Verktakar sinna þróunarvinnu í eigin umhverfi og samkvæmt eigin innri vinnureglum og öryggiskröfum. Verkkaupi hefur ávallt yfirsýn og endanlegt vald yfir þeim afurðum sem samþykktar eru til prófunar eða útgáfu.

Að verkefni loknu, eða við lok samningssambands, er verktaka skylt að eyða öllum afritum kóðageymslu verkkaupa úr öllum tölvukerfum, afritum, skýjaumhverfum og útstöðvum starfsmanna.

Atriði	Krafa
Megin geymsla kóða	Git-umhverfi verkkaupa
Þróunarumhverfi	Í umsjón verktaka
Endanlegt vald yfir kóða	Verkkaupi
Afrit eftir verklok	Verktaka skylt að eyða öllum afritum

4.2 Skipulag kóðageymsla

Hugbúnaðarkóði fyrir aðgreindar þjónustur eða einingar skal ekki vera geymdur saman í einni kóðageymslu. Mono-repo fyrirkomulag er ekki leyfilegt.

4.2.1 Rökstuðningur

Mono-repo (ein kóðageymsla fyrir margar sjálfstæðar þjónustur eða einingar) er viðurkennt skipulagsmynstur sem nýtist stórum stofnunum þar sem eitt teymi ber ábyrgð á öllu. Í útboðslíkani verkkaupa, þar sem mismunandi verksalar vinna samtímis að mismunandi kerfum, skapar mono-repo vandamál sem vega þyngra en kostirnir:

Áskoran	Lýsing
Aðgangsstýring milli verksala	Í mono-repo fá allir verksalar lestraraðgang að allri kóðageymslunni, ekki er hægt að afmarka aðgang að einstökum hlutum kóðagrunnsins á áreiðanlegan hátt. Þetta brýtur gegn meginreglu um lágmarksaðgang og gerir kleift að einn verksali sjái kóða annars verksala. Aðskildar kóðageymslur gera verkkaupa kleift að veita hverjum verksala aðgang eingöngu að þeim geymslum sem hann vinnur í.
Sjálfstæðar útgáfur og útgáfufarlar	Hver þjónusta þarf sjálfstætt útgáfunúmer, sjálfstæðan CI/CD-feril og sjálfstætt samþykktarferli. Í mono-repo verða þessir ferlar flóknir og viðkvæmir, breyting á einni þjónustu getur valdið endurbyggingu og endurprófun allra þjónusta. Aðskildar kóðageymslur tryggja að útgáfufarill hvernar þjónustu sé einangraður og sjálfstæður.
Gæðahlið per þjónustu	Gæðahlið verkkaupa (kaflí 6.4) er stillt eftir kóðageymslu. Í mono-repo gilda sömu gæðaviðmið á allt, ekki er hægt að aðlagja þekjukurfur, kóðagreiningarreglur eða öryggisprófanir eftir eðli hvernar þjónustu án sérsniðinna og viðkvæmra CI/CD-uppsetninga.
Skipting og yfirtaka verkefna	Verkkaupi þarf getu til að færa einstaka kerfi milli verksala, t.d. ef verksali lýkur samningi eða ef verkkaupi óskar eftir að annar aðili taki yfir viðhald tiltekins kerfis. Aðskildar kóðageymslur gera þetta einfalt: verkkaupi afturkallar aðgang og veitir nýjum verksala aðgang að viðkomandi geymslu. Í mono-repo er slík skipting ekki

	möguleg án þess að kljúfa kóðagrunninn, ferli sem er flókið, tímafrekt og áhættusamt.
Fork-vinnulag	Samstarfsferill verkkaupa (kafla 4.5) byggir á því að verksalar vinni í afritum (forks) af kóðageymslum verkkaupa. Fork-vinnulag virkar vel þegar ein kóðageymsla táknar eina sjálfstæða einingu. Fork af mono-repo myndi afrita allan kóðagrunn allra kerfa til verksala, bæði kóða sem hann vinnur í og kóða sem hann á ekkert erindi við.

Ein kóðageymsla skal notuð fyrir hvern hluta lausnar sem er gefinn út, útgáfustýrður og rekinn sjálfstætt.

Allar kóðageymslur í umsjón verkkaupa eru læstar. Breytingar á innihaldi þeirra eru einungis heimilar í gegnum breytingabeðni (Pull Requests) sem fara í gegnum skilgreint kóðarýni- og samþykktarferli.

Einungis ein langlíf kóðagrein skal vera til staðar í hverri kóðageymslu og skal hún bera heitið main.

Sérfræðingar verkkaupa stofna allar nýjar kóðageymslur við upphaf verkefna.

Atriði	Regla
Mono-repo	Ekki leyfilegt
Fjöldi kóðageymsla	Ein per sjálfstætt útgefin eining
Langlíf grein	main
Beinar breytingar á main	Ekki leyfilegt
Breytingarferli	Pull Request + kóðarýni
Stofnun kóðageymslu	Framkvæmd af verkkaupa

4.3 Nafnareglur á kóðageymslum

Kóðageymslur skulu fylgja eftirfarandi nafnareglum:

Regla	Krafa
Stafsetning	Lágstafir og bandstrik (-) í stað bila eða undirstrika
Forskeyti	Sama forskeyti fyrir allar kóðageymslur sömu lausnar
Tímabundin heiti	Ekki leyfileg (t.d. -new, -old, -temp)
Útgáfunúmer í nafni	Ekki leyfileg (t.d. -v1)

Dæmi: Forritari vinnur að bókunarkerfi fyrir heilsuveru og þarf kóðageymslur fyrir vefviðmót, vef-API og bakendaþjónustu:

- heilsuvera-appointments-ui-web
- heilsuvera-appointments-api
- heilsuvera-appointments-service

4.4 Viðskeyti á kóðageymslum

Viðskeyti á kóðageymslur skulu vera í samræmi við eftirfarandi töflu. Sé viðskeyti ekki skilgreint í töflunni skal velja heiti sem samræmist mynstri hennar og lýsir hlutverki kóðageymslunnar á skýran og ótvíræðan hátt.

Samsetning viðskeyta er heimil þegar þess er þörf. Þá skal samsetningin vera frá almennu að sértæku (t.d. -database-fhir en ekki -fhir-database).

Viðskeyti	Lýsing
-service	Bakendaþjónustur sem eru ekki aðgengilegar utan eldvegs
-api	Bakendaþjónustur sem eru aðgengilegar af internetinu (utan eldvegs)
-ui-web	Viðmótslausnir fyrir vef
-ai-service	Gervigreindarþjónusta sem situr bak við eldvegg
-ai-api	Gervigreindarþjónusta sem er aðgengileg utan eldvegs (sjaldgæft)
-database	Gagnagrunnskóði
-database-APEX	Gagnagrunnskóði útfærður fyrir lausnir á borð við Oracle APEX
-database-pipeline	Gagnagrunnskóði og annar kóði tengdur gagnamedhöndlunarpípum
-infra	Innviðakóði (Kubernetes, Argo CD eða álíka)
-package-nuget	Pakkar fyrir .NET NuGet kerfið

-package-npm	Pakkar fyrir JavaScript NPM kerfið
-package-maven	Pakkar fyrir Java Maven kerfið
-jobs	Forrit sem keyrð eru reglulega eða sem bakgrunnsvinnslur
-docs	Skjölun
-ui-android	Native viðmótsforrit smíðað fyrir Android
-ui-ios	Native viðmótsforrit smíðað fyrir Apple iOS
-cli	Console forrit (skipanalínuviðmót)
-docs-android	Skjölun fyrir viðmótsforrit smíðað fyrir Android
-docs-ios	Skjölun fyrir viðmótsforrit smíðað fyrir iOS
-automation-actions	Sjálfvirknikóði skrifaður fyrir CI/CD pípulínur
-automation-pa	Sjálfvirknikóði skrifaður fyrir Microsoft Power Automate
-dashboard-powerbi	Mælaborð skrifað í Microsoft Power BI
-dashboard-web	Mælaborð skrifað sem vef sérsmiði
-config	Kóðageymsla sem inniheldur einungis stillingarskjöl; enginn ótengdur forritunarkóði
-fhir	Kóði sem tengist FHIR stöðlun og útfærslum

Sjá viðauka „Dæmi um nafnagjöf kóðageymsla“ fyrir frekari dæmi um nafnagjöf.

4.5 Hvernig ytri aðilar vinna með kóðageymslurnar

Verkkaupi stofnar og heldur utan um allar kóðageymslur. Við upphaf verkefnis útbýr verkkaupi grunnuppsetningu kóðageymslu, þar á meðal sjálfvirkniferla, aðgangsstýringar og möppustrúktúr.

Verktakar vinna eingöngu í afritum (forks) af kóðageymslum verkkaupa og hafa ekki beinan skrifaðgang að main-grein.

Allar endanlegar breytingar sem fara til prófunar eða útgáfu skulu berast til verkkaupa í gegnum breytingabeirni (Pull Requests) og sæta samþykkt samkvæmt skilgreindu ferli.

Skref	Lýsing	Ábyrgð
Stofnun kóðageymslu	Repo búið til og grunnuppsetning gerð	Verkkaupi
Þróun	Unnin í fork verktaka	Verktaki
Skil	Pull Request send til verkkaupa	Verktaki
Yfirferð	Kóðarýni, gæðamat og sjálfvirkar prófanir	Verkkaupi
Endanleg samþykkt	Merge í main	Verkkaupi
Útgáfa	Byggð af main	Verkkaupi

Verksala er skylt að eyða öllum afritum (forks) úr tölvukerfum, afritum, skýjum og útstöðvum eftir að verki lýkur.

Sjá viðauka „Samstarfsferli verktaka og verkkaupa“ fyrir nánari lýsingu á fork-and-branch ferlinu.

4.6 Nafnareglur á kóðagreinum

Kóðageymslur verkkaupa taka einungis við breytingabeirnum (Pull Requests) sem fylgja skilgreindum nafnareglum á kóðagreinum.

Heiti breytingabeirni (PR titill) skal vera nákvæmlega það sama og heiti viðkomandi kóðagreinar. Þetta tryggir skýran rekjanleika milli kóðagreina, breytingabeirna, prófunarskýrsla og samþykkt.

Tegund kóðagreinar	Nafnasnið	Tilgangur	Leyfilegt umhverfi
Feature	feature/*	Ný virkni	Ekki gefið út; notað til móttöku og rýni kóða
Hotfix	hotfix/YYYY.MM.DD-LÝSING	Bráðabirgðalagfæring á kóða í keyrsluumhverfi	Pre-Production, Production
QA	qa/YYYY.MM.DD-rcNN	Prófunarútgáfa, eingöngu ætluð Pre-Production	Pre-Production

Release	release/YYYY.MM.DD eða release/YYYY.MM.DD-hotfixN	Rekstrarútgáfa	Pre-Production, Production
---------	--	----------------	-------------------------------

Aðeins útgáfur sem eru merktar samkvæmt þessum reglum eru heimilar til uppsetningar í rekstrarumhverfi.

Útgáfunúmer í -rc sniði eru aldrei leyfileg í Production umhverfi.

Sjá viðauka „Rekjanleiki milli kóðagreina, breytingabeiðna og útgáfa“ og „Útgáfunúmer og merking“ fyrir nánari dæmi.

5 Viðtöku- og útgáfuferlar

Þessi kafli lýsir því hvernig verkkaupi tekur við hugbúnaðarafurðum frá verksala og hvernig útgáfur eru stýrðar frá móttöku til rekstrarumhverfis.

5.1 Viðtökuferill verkkaupa

Allar breytingabeiðnir (Pull Requests) sem berast kóðageymslum verkkaupa fara í gegnum formlegan viðtökuferil sem samanstendur af sjálfvirkum og handvirkum þáttum.

5.1.1 Sjálfvirkt gæðamat

Við móttöku breytingabeiðni eru sjálfvirkt keyrðir gæða- og öryggisferlar (bygging, prófanir, kóðagreining). Ef sjálfvirkir ferlar bregðast er breytingabeiðni sjálfkrafa hafnað og verksala ber að lagfæra og endursenda.

5.1.2 Kóðarýni og gæðamat

Að lokinni sjálfvirkri yfirferð framkvæmir verkkaupi formlega kóðarýni með aðstoð sérfræðinga og/eða gervigreindarlausna í umsjón verkkaupa. Efni þessarar þróunarhandbókar og tengdra stoðskjala liggur til grundvallar.

Niðurstöður eru skráðar í staðlað yfirferðarskjal sem er afhent verksala. Skjalið inniheldur allar athugasemdir sem komu fram í yfirferð, flokkaðar eftir uppruna (Org-Policy, Security-Standard, Best-Practice) og metnar eftir alvarleikastigi (Critical, High, Medium, Low).

Athugasemdir sem eru flokkaðar sem **Critical** koma í veg fyrir samþykkt útgáfu þar til þeim hefur verið sinnt. Verksali ber ábyrgð á að svara öllum athugasemdum í yfirferðarskjalinu, óháð alvarleikastigi.

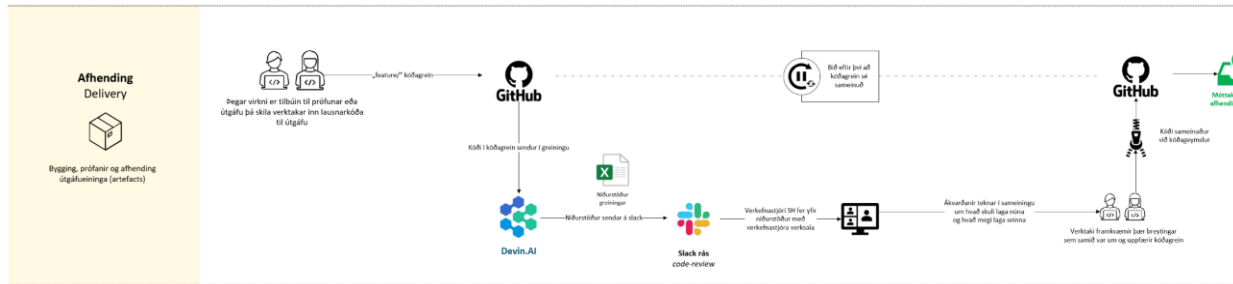
Sjá viðauka „Yfirferðarskjal kóðarýni: snið, flokkunarkerfi og dæmi “ fyrir fulla lýsingu á sniði skjalsins, flokkunarkerfi, alvarleikastigum, vinnulagi og dæmum.

5.1.3 Ferli viðtöku og samskipti

Eftir að yfirferðarskjal hefur verið afhent fer eftirfarandi ferli í gang:

Skref	Lýsing	Ábyrgð
Sjálfvirkt gæðamat	Sjálfvirkir ferlar keyrðir á breytingabeiðni	Verkkaupi (innviðir)
Kóðarýni	Sérfræðingar og/eða gervigreindar lausnir verkkaupa fara yfir kóða	Verkkaupi
Yfirferðarskjal	Athugasemdir skráðar í staðlað skjal og afhent verksala	Verkkaupi
Endurgjöf og samráð	Fundur haldinn til að fara yfir athugasemdir og ákveða lagfæringar	Báðir
Lagfæringar	Verksali gerir nauðsynlegar breytingar og uppfærir breytingabeiðni	Verksali
Svörun í yfirferðarskjal	Verksali fyllir út „Svar frá verksala“ og „Tímamat frá verksala“ fyrir hverja athugasemd	Verksali
Samþykkt eða höfnun	Verkkaupi metur svörun og lagfæringar; samþykkir eða hafnar	Verkkaupi

Verksali ber ábyrgð á að svara öllum athugasemdum í yfirferðarskjali. Ef verksali er ósammála athugasemd skal hann rökstyðja afstöðu sína skriflega í svæðinu „Svar frá verksala.“ Verkkaupi tekur lokaákvörðun um hvern lið.



Mynd 2 Afhendingarferill í smáatriðum, sem sýnir hvernig gervigreindardrifið rýniferli og samskipti verkefnastjóra beggja aðila virka í framkvæmd.

5.2 Útgáfufेरill

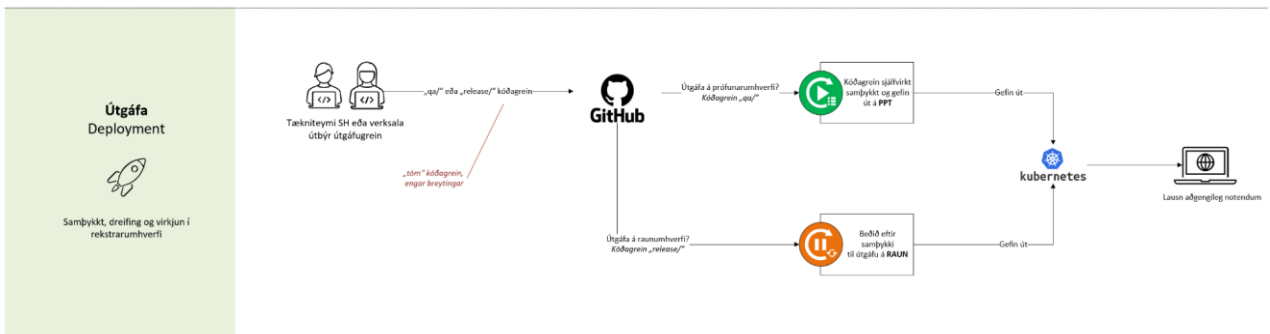
Útgáfur til prófunar- og rekstrarumhverfa skulu ávallt byggðar á main-grein. Ekki er heimilt að gefa út hugbúnað beint af öðrum greinum.

Allar útgáfur sæta skilgreindu samþykktarferli áður en þær eru settar í viðkomandi umhverfi. Samþykktarferli ræðst af tegund útgáfu:

Umhverfi	Tegund útgáfu	Byggt á	Samþykkt
Pre-Production (QA)	qa/YYYY.MM.DD-rcNN	main	Sjálfvirk
Production (PROD)	release/YYYY.MM.DD eða release/YYYY.MM.DD-hotfixN	main	Handvirk (verkkaupi)

Verktökum er óheimilt að gefa út hugbúnað beint eða óbeint í prófunar- eða rekstrarumhverfi án samþykkis verkkaupa.

Óheimilt er að setja QA-útgáfur, prófunarútgáfur eða útgáfur merktar -rc í rekstrarumhverfi (PROD).



Mynd 3 Útgáfufærill í smáatriðum, sem sýnir flæði milli QA og PROD umhverfa og handvirktt samþykktarskref.

Sjá viðauka „Útgáfufærilar (QA og PROD)“ og „Útgáfunúmer og merking“ fyrir nánari lýsingu á ferlum og merkingum.

6 Gæðafærilar

Leiddarljós

Réttleiki, öryggi og gæði hugbúnaðar skulu ávallt vera í forgangi. Allur hugbúnaður sem skilað er til verkkaupa fer í gegnum sjálfvirka gæðastýringarferla áður en samþykki fæst fyrir prófun eða útgáfu. Gæðastýring skal vera einsleit, hlutlæg og sjálfvirk eftir því sem unnt er.

Þessi kafli lýsir þeirri gæðainnviðagrind sem verkkaupi rekur og sem verksali verður að uppfylla við skil. Viðtöku- og útgáfufærilið sjálf er lýst í kafla Viðtöku- og útgáfufærilar.

6.1 Ábyrgðaskipting

Ábyrgðarsvið	Verkkaupi	Verksali
Hýsing og rekstur gæðainnviða (CI/CD, gæðahlið, greiningartól)	✓	
Skilgreining og viðhald gæðaviðmiða og stillinga	✓	
Þróunarumhverfi, hermipjónustur og innri sjálfvirkni		✓
Að hugbúnaður standist gæðakröfur við skil		✓
Að dagleg þróunarvinna samræmist gæðakröfum		✓

Vegna flækjustigs innviða verkkaupa eru tímabundin (ephemeral) umhverfi ekki í boði.

6.2 Sjálfvirkir gæðastýringarferlar

Allar breytingabæiðir (Pull Requests) sem berast kóðageymslum verkkaupa fara í gegnum sjálfvirka gæðastýringarferla. Þessir ferlar eru skilgreindir, reknir og viðhaldnir af verkkaupa.

Ef eitthvert sjálfvirkt gæðaskref bregst er breytingabæiðni sjálfkrafa hafnað. Verksali ber ábyrgð á að lagfæra og endursenda.

6.3 Hvað er prófað sjálfvirkt

Eftirfarandi atriði eru prófuð sjálfvirkt við hverja breytingabæiðni:

Prófunarþáttur	Lýsing	Afleiðing ef bregst
Uppbygging lausnar	Kóðinn er prófaður m.t.t. möppuskipulags og skipulags	PR hafnað
Bygging (build)	Kóðinn er þýddur og byggingarferli keyrt	PR hafnað
Einingaprófanir	Sjálfvirkar einingaprófanir keyrðar; þekjukröfur metnar	PR hafnað ef þekja undir viðmiðum
Kóðagæði og kóðarýni	Sjálfvirk kóðagreining og gæðamat keyrð	PR hafnað ef gæðahlið bregst
Öryggisprófanir á keyrslueiningum	Docker einingar fara í gegnum öryggisprófun	PR hafnað ef alvarlegar athugasemdir finnast
Gagnagrunnskóði	Allur SQL kóði fer í gegnum gæðaeftirlit	PR hafnað ef villur finnast

6.4 Gæðahlið (Quality Gates)

Gæðahlið eru sjálfvirk viðmið sem kóði verður að standast til að breytingabæiðni fái samþykkt. Stillingar á gæðahliðum eru í höndum verkkaupa og er óleyfilegt að breyta þeim án formlegs samþykkis.

Gæðahlið notar clean-as-you-code aðferðafræði: kóði í breytingabæiðni er borinn saman við núverandi stöðu kóðagrunnsins. Þetta tryggir að gæði kóðans batni jafnt og þétt án þess að leggja á verksala fyrirbyggjandi vandamál í ótengdum eldri kóðahlutum.

Athugasemdir á Critical stigi koma í veg fyrir samþykkt. Einingaprófanir verða að ná skilgreindum þekjumörkum (sjá kafla 13, Hugbúnaðarprófanir) og viðvaranir í kóðagreiningu eru meðhöndlaðar sem villur.

Ástæðan er sú að viðvaranir sem ekki eru meðhöndlaðar safnast upp yfir tíma og verða að tækniskuld sem erfitt er að greiða niður eftir á. Með því að meðhöndla viðvaranir sem villur frá upphafi er tryggt að kóðagrunnur haldist hreinn og að vandamál séu leyst þegar þau koma upp, ekki mánuðum eða árum síðar þegar samhengið er glatað.

Sjá viðauka „Kóðastíll og gæðatól, verkfæri, stillingar og reglusett“ fyrir nákvæm gæðaviðmið, þýðistillingar, reglusett og kröfur um Docker öryggisprófanir.

6.5 Gervigreind í gæðaeftirliti

Verkkaupi nýtir gervigreindarlausnir sem hluta af gæðaeftirliti við kóðaskil. Gervigreindin aðstoðar við sjálfvirka greiningu á kóðagæðum, öryggisatriðum og samræmi við handbækur og stefnur verkkaupa.

Niðurstöður gervigreindaryfirferðar eru hluti af yfirferðarskjali sem lýst er í kafla 5 *Viðtöku- og útgáfufarlar*.

Sjá viðauka „Yfirferðarskjal kóðarýni“ fyrir snið, flokkunarkerfi og dæmi um niðurstöður gervigreindarrýni.

Sjá viðauka „Þjónustur og verkfæri verkkaupa“ fyrir lista yfir þau tól og þjónustur sem verkkaupi nýtir í gæðaeftirliti.

6.6 Ábyrgð verksala á innri gæðum

Verkkaupi ber ábyrgð á gæðainnviðum við skil. Verksali ber hins vegar ábyrgð á að innri þróunarvinna sé skipulögð þannig að skil standist gæðakröfur.

Verksala er ráðlagt að:

- stilla þróunarumhverfi sín þannig að viðvaranir séu meðhöndlaðar sem villur,
- keyra sambærilega kóðagreiningu og gæðaeftirlit innanhúss áður en skil fara fram,
- gera sér grein fyrir gæðakröfum verkkaupa snemma í þróunarferlinu til að forðast óþarfa endurvinnu við skil.

Verkkaupi útvegar eftir þörfum aðgang að gæðatólum og stillingum sem gera verksala kleift að herma sjálfvirka gæðaferla áður en skil fara fram.

7 Skil á hugbúnaðarafurðum

7.1 Skilastefna

Verksali skal skila hugbúnaðarafurðum reglulega til rýni og samþykktar hjá verkkaupa. Skil skulu fara fram í afmörkuðum og virkum einingum þannig að unnt sé að meta virkni, gæði og framvindu verkefnisins jafnóðum.

Að lágmarki skal skila hugbúnaði til rýni á þriggja mánaða fresti, nema annað sé sérstaklega samið í verkefnissamningi. Verkkaupi áskilur sér rétt til að krefjast tíðari skila ef tilefni er til.

Notkun Minimum Viable Product (MVP) nálgunar er æskileg leið til að uppfylla þessa kröfu, en leysir ekki undan skyldu til reglulegra skila.

7.2 Hverju á að skila?

Skil skulu ná til allra þátta lausnarinnar sem áformað er að afhenda við lok verkefnis. Eftirfarandi þættir skulu, eftir því sem við á, fylgja hverri afhendingu:

Þáttur	Lýsing
Hugbúnaðarkóði	Allur kóði sem tilheyrir verkefninu, þ.m.t. bakendi, framendi og innviðakóði
Gagnagrunnskóði	SQL scriptur og migration skrár sem tilheyra útgáfunni
Prófanir	Eininga-, samþættingar- og aðrar sjálfvirkar prófanir ásamt prófunargögnum
Prófunarniðurstöður	Niðurstöður prófana, keyrsluskrár og gæðaskýrslur
Skjölun	README, ADR, rekstrarskjölun og önnur skjölun sem tilheyrir útgáfunni
Docker stillingar	Dockerfile og tengdar stillingar
Kubernetes stillingar	.kube/ mappa með Argo CD innviðakóða
Gagnaútflutningsleiðbeiningar	Skjölun sem lýsir hvernig öll gögn lausnarinnar eru flutt út á stöðluðu sniði, þar á meðal gagnasnið, útflutningsferli og tengsl milli gagnatafla. Leiðbeiningar skulu gera verkkaupa kleift að flytja gögn án aðkomu verksala.
Úttektaráætlun (exit plan)	Skjal sem lýsir skrefum, forsendum, tíma og kostnaði við yfirtöku lausnar af öðrum aðila. Skal afhent við fyrstu formlegu skil og uppfært samhliða verulegum breytingum.

Skil skulu almennt vera QA-útgáfur sem endurspeгла raunverulega stöðu lausnarinnar á hverjum tíma. Skil sem innihalda ófullkláraða, óprófaða eða ósamsetta virkni skulu merktar sem slíkar.

7.3 Hver skilar?

Verksali ber ábyrgð á að skila afurðum verkefnisins inn til verkkaupa. Verksali tilnefnir ábyrgðaraðila skila sem hefur heimild og yfirlit yfir þær afurðir sem skilað er.

Heimilt er að skilgreina aðra skilastjóra með samþykki verkkaupa.

7.4 Hvenær er æskilegt að skila inn?

Eftirfarandi tafla skilgreinir æskilega og heimila tíðni skila eftir tegund:

Tegund skila	Vikulega	Mánaðarlega	Sjaldnar en 1x í mánuði	Í lok verkefnis
Feature branch skil	Leyfilegt	Æskilegt	Óæskilegt	Óheimilt
Á pre-production umhverfi	Leyfilegt	Leyfilegt	Æskilegt	Krafist
Á production umhverfi	Óheimilt	Óæskilegt	Leyfilegt	Krafist

Taflan setur fram viðmið um regluleg skil. Verkkaupi áskilur sér rétt til að krefjast tíðari skila ef tilefni er til.

7.5 Rekjanleikakröfur í skilum

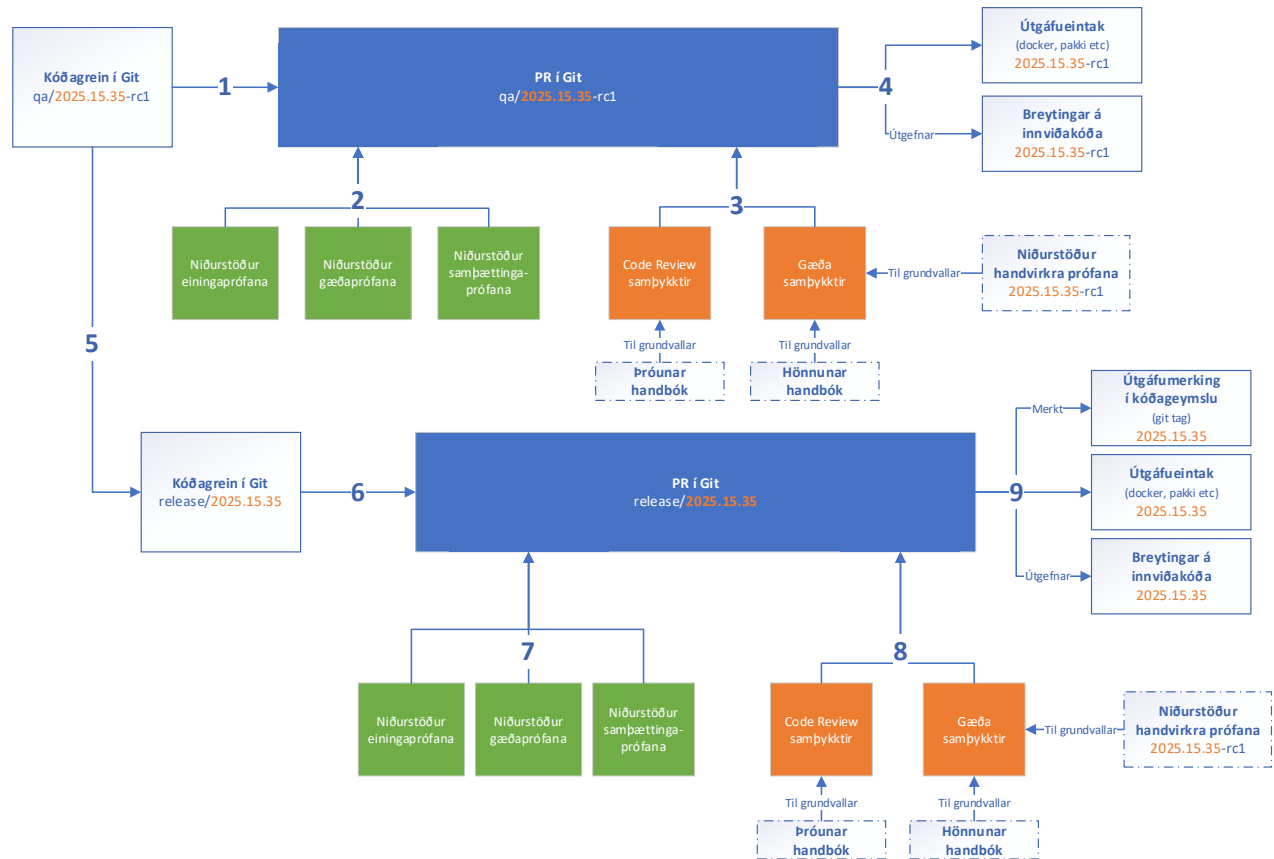
Rekjanleiki skal vera tryggður á einfaldan, skýran og endurtekanlegan hátt milli:

- kóðagreina,
- breytingabeiðna (Pull Requests),
- prófunarniðurstæðna,
- samþykktar,
- útgáfa í QA og rekstrarumhverfi.

Rekjanleiki er náð fram með því að:

Aðferð	Lýsing
Samræmd nöfn	Heiti kóðagreina, breytingabeiðna og handvirkra prófana eru samræmd
Prófunarniðurstöður í PR	Breytingabeiðnir innihalda niðurstöður gæða-, eininga- og samþættingaprófana
Samþykktir í PR	Samþykktir á kóðaskilum og gæðamati eru skráðar með breytingabeiðnum
Byggingarsaga	Breytingabeiðnir innihalda yfirlit yfir hvernig hugbúnaður var byggður og prófaður
QA/Release samræmi	Samræmi er milli nafna QA- og Release-kóðagreina

Rekjanleikaskýringun er hluti af samþykktarferli skila.



Mynd 4 „Rekjanleiki skila frá QA til PROD“ og sjá viðauka „Rekjanleiki milli kóðagreina, breytingabeidna og útgáfa“ fyrir nánari lýsingu.

7.6 Undantekningar frá skilareglum

Heimilt er að víkja frá skilgreindum skilareglum í neyðartilvikum vegna alvarlegra rekstrartruflana í rekstrarumhverfi. Slíkar undantekningar eru einungis heimilar:

með formlegu samþykki tæknilegs ábyrgðaraðila verkkaupa, og að undangengnu mati á áhættu, áhrifum og nauðsyn aðgerðar. Allar undantekningar skulu skjalfestar og raktar í kjölfar atviks.

8 Forritunarmál og tæknival

Leiðarljós

Hugbúnaðarlausnir sem þróaðar eru fyrir verkkaupa skulu vera hannaðar til reksturs og viðhalds í að lágmarki 30 ár. Tæknival skal taka mið af langtíma stöðugleika, fyrirsjáanleika og viðhaldshæfni, en ekki tímabundnum tæknivinsældum.

Tækni sem valin er skal vera rótgróin, vel studd til langs tíma og líkleg til að laða að hæft hugbúnaðarfólk áratugum saman.

Þessi kafli skilgreinir hvaða forritunarmál, framework og tæknieiningar eru samþykktar fyrir lausnir verkkaupa. Allir þættir sem ekki eru sérstaklega nefndir hér eru háðir formlegu samþykki samkvæmt undanþágufærlu.

8.1 Bakendakerfi og API

Allar bakendaþjónustur, forritunarskil (API) og aðrar keyrslueiningar sem eru hluti af rekstrarumhverfi verkkaupa skulu vera skrifaðar í **C#** og keyra á **.NET** umhverfi.

Ávallt skal notast við nýjstu **Long Term Support (LTS)** útgáfu af **.NET** sem er í gildi við áætlaðan útgáfudag lausnar.

Bakendalausnir skulu nýta innbyggða virkni og staðla ASP.NET og forðast óþarfa innleiðingu á ytri forritunarsöfnum þar sem sambærileg virkni er þegar til staðar.

Krafa	Gildi
Forritunarmál	C#
Keyrsluumhverfi	.NET (nýjasta LTS)
Framework	ASP.NET
Gagnasnið	JSON
Gagnavinnsla	SQL + Dapper.NET
Keyrsla	Docker
Samvirkni	HL7 / FHIR þar sem við á

8.1.1 Undanþágur

Undanþágur frá notkun C# og .NET eru einungis heimilar í algerum undantekningartilvikum þar sem:

- sannanlegar og rökstuddar tæknilegar ástæður liggja fyrir, og
- fyrirhuguð lausn uppfyllir sambærilegar kröfur um langtímastuðning, öryggi, rekjanleika og sjálfvirka gæðastýringarferla.

Allar undanþágur skulu fylgja skilgreindu undanþágufærli og hljóta formlegt skriflegt samþykki áður en þróun hefst.

Sjá viðauka „Undanþágufærill frá stefnu verkkaupa“.

8.2 Framendar / Vef

Framendalausnir sem þróaðar eru fyrir verkkaupa skulu fylgja annarri af tveimur samþykktum leiðum:

Leið	Lýsing	Staða
Leið A , ASP.NET Server-Side Rendering	ASP.NET MVC eða Razor Pages með server-side rendering (SSR) sem meginaðferð. Meginhluti domain flæðis, stöðustjórnunar, gagnavinnslu og öryggis útfærður á netþjóni.	Meginleið (sjálfgefið val)
Leið B , Gervigreindarsmíðaður React framendi	Léttur React framendi í TypeScript, smíðaður eingöngu með gervigreindarþróunarverkfærum, sem á samskipti við .NET bakenda-API. Enginn Node.js í rekstrarumhverfi.	Viðbótarleið (heimilt ef öll skilyrði eru uppfyllt)

Val á leið skal vera skjalfest í arkitektúrákvörðunum (ADR) og samþykkt af tæknilegum sérfræðingum verkkaupa áður en þróun hefst. Leið A er sjálfgefið val og krefst ekki sérstakrar rökstuðnings. Ef verksali óskar eftir Leið B skal rökstyðja valið í arkitektúrákvörðun.

Báðar leiðir falla undir sömu kröfur um aðgengi (kafla 14), hönnunarstaðla (kafla 8.3) og öryggi (kafla 9).

8.2.1 Leið A, ASP.NET Server-Side Rendering

Framendalausnir í Leið A skulu vera byggðar á .NET og ASP.NET með server-side rendering (SSR) sem meginaðferð. Meginhluti domain flæðis, stöðustjórnunar, gagnavinnslu og öryggis skal ávallt vera útfærður á netþjóni, ekki í vafra notanda.

8.2.1.1 Samþykktur framendatæknistafi, Leið A

Þáttur	Krafa
Forritunarmál	C#
Umhverfi	ASP.NET MVC eða Razor Pages
Keyrsla	Server-Side Rendering
Framenda gagnvirkni	HTMX (takmörkuð notkun; sjá skilyrði í kafla 8.2.1.2)

JavaScript / TypeScript	Aðeins sem hjálpartæki; aldrei burðarás
Stílar	HTML5 og CSS
Web Components	Heimilt þar sem við á
Aðgengi	WCAG 2.2 AA (sjá kafla 14 Aðgengismál)

8.2.1.2 HTMX og gagnvirkni (Leið A)

HTMX (og sambærileg tækni) er heimil sem viðbótartæki til að bæta notendaupplifun í server-side lausnum.

Heimil notkun	Óheimil notkun
Hlutaendurnýjun á síðum	Flutningur domain flæðis í vafra
Einfaldar notendaaðgerðir	Flókin stöðustjórnun (state) á client
Aukin svörun í viðmóti	Eftirmynd SPA-hegðunar

Af hverju er notkun HTMX takmörkuð?

HTMX er öflugt verkfæri sem gerir þróunaraðilum kleift að bæta gagnvirkni við server-side lausnir án hefðbundinnar JavaScript forritunar. Takmarkanir verkkaupa lúta ekki að HTMX sem tækni heldur að þeim áhrifum sem umfangsmikil notkun þess hefur á rekjanleika, rýnihæfni og yfirtökuhæfni lausnar.

Áskorun	Lýsing
Samhliða leiðarkerfi	Þegar HTMX er notað sem arkitektúrburðarás (hx-get, hx-post, hx-target, hx-swap á meirihluta síðna) verður samskiptaflæði lausnarinnar tvískipt: hluti hegðunar er í ASP.NET stýringum (controllers/pages) og hluti í HTML eigindum (attributes). Forritari sem les C# kóðann sér ekki alla hegðun kerfisins og öfugt. Þetta brýtur gegn meginreglu um skýra aðgreiningu ábyrgðar (SoC) og læsilegan kóða (kafla 9.1).
Gæðaferlar og kóðarýni	Gæðahlið verkkaupa (SonarCloud, gervigreindardrífni kóðarýni) greina C# kóðaslóðir. Ef meginhluti samskiptaflæðis er skilgreindur í HTMX eigindum í Razor sniðmátum nær kóðagreining ekki yfir þann hluta hegðunar. Þetta skapar blinda bletti í gæðaeftirliti sem grefur undan gæðahliðum (kafla 6).
Prófanaforsendur	HTMX-drifin gagnvirkni er einungis prófanleg á vafrastigi (Playwright, Cypress). Umfangsmikil HTMX-notkun færir þungann frá einingaprófunum og samþættingarprófunum í C# (hröð, ódýr, stöðug) yfir í viðmótsprófanir í vafra (hæg, dýr, viðkvæm). Þetta stangast á við markmið kafla 13 um að einingaprófanir séu grunnforsenda gæðatryggingar.
Falinn SPA-eiginleiki	Full HTMX-lausn getur hegðað sér eins og SPA, efni skipt út gagnvirkt, vafrasaga meðhöndluð, síður aldrei endurhlaðnar, þótt hún sé tæknilega server-rendered. Þetta ógnar þeim mörkum sem framendastefna verkkaupa dregur á milli server-side rendered lausna (Leið A) og client-side lausna (Leið B / undanþágur).
Yfirtaka og endursmíði	Nýr verksali sem yfirtekur lausn þarf að skilja bæði ASP.NET leiðarkerfi og HTMX eigindakerfi til að rekja flæði kerfisins. Þetta eykur lærdómskúrfu og yfirtökuáhættu (kafla 7.2) umfram það sem hefðbundin ASP.NET MVC/Razor Pages lausn krefst.

Þegar HTMX er notað í takmörkuðu hlutverki (hlutaendurnýjun síðna, einfaldar notendaaðgerðir, aukin svörun) er meginhluti samskiptaflæðis áfram í C# stýringum og ASP.NET leiðarkerfi. Ofangreindar áhyggjur eiga þá ekki við.

1.1.1.1 JavaScript / TypeScript í Leið A

JavaScript og TypeScript eru einungis heimil sem hjálpartæki fyrir afmarkaða virkni í birtingarlagi. Þau mega aldrei vera burðarásar framendalausna í Leið A, hvorki hvað varðar arkitektúr, domain flæði né stöðustjórnun.

8.2.2 Leið B, Gervigreindarsmiðaður React framendi

Leið B heimilar þróun léttvægs React framenda í TypeScript, að uppfylltum öllum skilyrðum sem tilgreind eru hér að neðan. Tilgangur Leiðar B er að opna fyrir gervigreindardrífna framendapróun þar sem aðskilinn, léttvægur framendi þjónar notendaupplifun betur en server-side rendered lausn.

Leið B er viðbótarleið, ekki staðgengill fyrir Leið A. Hún breytir engu um kröfur til bakenda, allur bakendi skal áfram vera í C# á .NET samkvæmt kafla 81.

8.2.2.1 Forsendur og gildissvið

Leið B á við þegar:

- framendi er aðskilinn frá bakenda og á samskipti í gegnum skilgreint .NET API lag,
- framendinn er smíðaður eingöngu með gervigreindarþróunarverkfærum,
- framendinn er nógu einfaldur til þess að hann sé auðveldlega endursmíðaður með gervigreindarverkfærum í framtíðinni.

Leið B á ekki við þegar:

- framendi krefst flókinnar stöðustjórnunar, sjálfstæðs rökfræðilags eða viðamikils domain flæðis í vafra,
- framendi gegnir hlutverki bakenda (BFF, Backend-for-Frontend),
- verkefnið krefst Node.js keyrslu í rekstrarumhverfi.

8.2.2.2 Af hverju er krafist gervigreindarsmíði?

Krafa Leiðar B um að framendakóði sé smíðaður eingöngu með gervigreindarverkfærum er ekki handahófskennd tæknistefna heldur bein afleiðing af þeirri hönnunar- og skipulagsaðferð sem handbókakerfi verkkaupa byggir á.

Verkkaupi leggur áherslu á að meginhluti fjárfestingar í hugbúnaðarverkefni fari í skilgreiningu, greiningu, hönnun og notendaprófanir áður en forritun hefst. Hönnunarhandbók verkkaupa skilgreinir fjögurra þrepa ferli (Skilgreining → Greining → Hugmyndir → Hönnun og þróun) þar sem hvert þrep hefur formleg afhendingargögn og gæðatékklista. Lýsingar á notendasögum skulu samþykktar af ábyrgðaraðila og lykilnotendum áður en forritun hefst (hönnunarhandbók §8.1). Fullhönnuð viðmótsútfærsla samkvæmt hönnunarkerfi verkkaupa (Design system), þar á meðal lokahönnun allra skjámynda, skal liggja fyrir sem hönnunarskjöl (Figma eða sambærilegt) við upphaf þróunar (hönnunarhandbók §4.6.4). Þróunarhandbók verkkaupa (kaflí 6) krefst þess jafnframt að hugbúnaðarsmíði megi ekki hefjast fyrr en formlegt samþykki liggur fyrir á heildararkitektúr lausnar, högun og samskiptum eininga, og gagnalíkani.

Þegar þessu ferli er fylgt liggja eftirfarandi gögn fyrir áður en framendakóði Leiðar B er skrifaður:

Inntak	Uppruni
Samþykktar skjámyndir og viðmótsflæði	Hönnunarskjöl (Figma eða sambærilegt)
Hönnunarkerfi (Design system) og stílar	Hönnunarhandbók og Storybook verkkaupa
Notendasögur með viðtökuskilyrðum	Hönnunarhandbók §8.4
OpenAPI-skilgreiningar bakenda	.NET bakendi (kaflí 8.1)
Arkitektúrákvörðun (ADR)	Þróunarhandbók kaflí 9
.agent/ leiðbeiningar	Kaflí 11.1

Leið B framendi er þar af leiðandi ekki sjálfstæð hönnunarúrlausn heldur birtingarlag, útfærsla á ákvörðunum sem mannlegir hönnuðir, notendur og sérfræðingar hafa þegar tekið og samþykkt. Þetta er nákvæmlega sú tegund verkefna sem gervigreindarþróunarverkfæri eru best til fallin í: vel afmarkað, forskriftarbundið og stutt vinnulag þar sem útkoman er skilgreind fyrirfram.

Minni háttar hönnunaraðlögun á smíðastigi, sem er eðlilegur hluti allrar þróunar, kemur ekki í veg fyrir gervigreindarsmíði. Slíkar aðlaganir (t.d. villumeðhöndlun í samtengingum, jaðartilvik í flæði, smávægilegar stílbreytingar) eru innan getu nútímagervigreindarverkfæra þegar samþykkt hönnun, notendasögur og .agent/ leiðbeiningar liggja fyrir.

Ef framendi krefst umfangsmikillar hönnunarúrlausnar á forritunarstigi bendir það til þess að undirbúningsferlið hafi ekki gengið sem skyldi eða að framendinn sé of flókinn fyrir Leið B og að hann eigi að útfæra í Leið A.

Verksali sem telur sig þurfa mannlega forritara til að skrifa framendakóða ber sönnunarbyrði á annarri af tveimur fullyrðingum: annaðhvort var hönnunarferlið ófullnægjandi (sem er ferlisgalli, ekki rökstuðningur gegn gervigreindarsmíði) eða að framendinn sé of flókinn (sem þýðir að Leið A eigi við). Í hvorugu tilvikinu er réttlætning til að víkja frá kröfunni um gervigreindarsmíði án formlegrar undanþágu.

8.2.2.3 Skilyrði fyrir notkun Leiðar B

Öll eftirfarandi skilyrði eru bindandi (SKAL) ef Leið B er valin:

Skilyrði	Krafa
----------	-------

Smíðaaðferð	Framendakóði skal vera smíðaður eingöngu með gervigreindarþróunarverkfærum (t.d. Claude Code, Devin.AI eða sambærilegu). Verksali skal staðfesta þetta skriflega (yfirlýsingarskylda) og verkkaupi áskilur sér rétt til stikkprófana.
Handvirkar breytingar	Einungis heimilt í stillingaskrá og staðfæringaskrá. Sjá nákvæma skilgreiningu í undirkafla „Handvirkar breytingar“ hér að neðan.
Forritunarmál	TypeScript með strict mode (sjá tsconfig kröfur í viðauka).
Framework	React (með Vite eða sambærilegu smíðaverkfæri sem framleiðir statískar skrár).
Bakendi	Framendinn skal eiga öll gagnaskipti við .NET bakenda-API. Engin bakendavirkni, API, millilög (middleware), BFF eða þjónustulag, í Node.js eða TypeScript.
Node.js í keyrslu	Engin Node.js keyrandi sem hluti af lausn í rekstrarumhverfi (production). Node.js er heimilt í þróunar- og CI/CD umhverfi.
.agent/ mappa	Framendaverkefni skal hafa .agent/ möppu samkvæmt kafla 11.1. Gervigreindarverkfæri skulu uppfæra .agent/ þekkingarsafn stöðugt á meðan á smíði stendur. Gæði og innihald .agent/ möppu er metið með auknum þunga við viðtöku á Leið B verkefnum.
API-samskipti	Samskipti við bakenda byggð á OpenAPI-skilgreiningum. Sjálfvirk smíði á TypeScript-týpum úr OpenAPI er æskileg.
Aðgengi	WCAG 2.2 AA (sömu kröfur og Leið A; sjá kafla 14).
Hönnunarstaðlar	Skulu fylgja hönnunarstöðlum og útlitsstílsniði verkkaupa (sömu kröfur og Leið A; sjá kafla 8.3).
Þróunarferli	Vinnulag samkvæmt viðauka „Æskilegt gervigreindarþróunarferli“ skal fylgt í heild sinni. Áætlunar- og framkvæmdaáætlun skulu vistaðar í -docs kóðageymslu og samþykktar af tæknilegum ábyrgðaraðila SH áður en kóðasmíði hefst.

8.2.2.4 Samþykktur tæknistafla, Leið B

Þáttur	Krafa
Forritunarmál	TypeScript (strict mode)
Framework	React
Smíðaverkfæri	Vite eða sambærilegt (statísk skjöl, enginn Node.js þjónn)
Stöðustjórnun	Innbyggð React stöðustjórnun (useState, useReducer, useContext). Ytri stöðustjórnunarsöfn eru óheimil.
API-samskipti	fetch eða innbyggðar aðferðir; eða léttvæg OpenAPI-mynduð þjónustulög.
Stílar	CSS Modules, Tailwind CSS eða innbyggð CSS.
Prófanir	Vitest eða Jest með React Testing Library (keyra í CI/CD; ekki í rekstrarumhverfi).
Dreifing	Statískar skrár (HTML/JS/CSS), annaðhvort frá .NET bakenda eða úr docker gámi með statískum skráarþjóni (t.d. nginx).
Gæðatöl	ESLint og TypeScript compiler (sjá viðauka).

8.2.2.5 Óheimilar aðferðir og tæknilausnir innan Leiðar B

Tækni eða nálgun	Staða	Ástæða
Next.js, Remix, Gatsby, Astro	Óheimilt	Meta-framework sem krefst Node.js keyrslu eða BFF-högunar
Angular, Vue, Svelte og álíka	Óheimilt	Ekki hluti af samþykktum tæknistafla Leiðar B
Node.js bakendi (NestJS, Express, Fastify)	Óheimilt	API-þróun eingöngu í .NET/C#
Ytri stöðustjórnunarsöfn (Redux, Zustand, MobX, Recoil)	Óheimilt	Eykur flækjustig og hindrar einfalda endursmíði
Umfangsmikil þriðja aðila UI-söfn (Material UI, Ant Design, Chakra UI)	Óheimilt	Eykur hæði áhættu (dependencies) og úreldingaráhættu
Umfangsmikil gagnasamskiptasöfn (TanStack Query, Apollo, SWR)	Óheimilt nema rökstutt	Eykur flækjustig; notkun krefst skriflegs rökstuðnings og samþykkis
GraphQL biðlarasöfn (Apollo, urql)	Óheimilt	REST/OpenAPI er samþykkt samskiptaleið
Server-Side Rendering í React	Óheimilt	Krefst Node.js keyrslu
Handskrifaður framendakóði	Óheimilt	Framendi skal vera AI-smíðaður; sjá smíðaaðferð

8.2.2.6 Skilgreining á einfaldleika (flækja við endursmíði)

Einfaldleiki framenda í Leið B er ekki skilgreindur sem hámark á línnum kóða eða fjölda íhluta (components), heldur sem mælikvarði á flækju á endursmíði. Framendinn skal vera útfærður þannig að eftirfarandi viðmið séu uppfyllt:

Viðmið	Lýsing
Lágmarka flækju við endursmíði	Gervigreindarþróunarverkfæri skulu geta endursmíðað framendann í heild sinni á viðráðanlegum tíma, án sérhæfðrar mannlegrar þekkingar á upprunalegu útfærslu.
Uppfæranlegur	Framendinn skal vera nógu einfaldur til þess að gervigreindarverkfæri geti flutt hann milli meiriháttar útgáfa af React (t.d. frá React 18 til React 22) í einni endursmíðalotu.
Lágmarka hæði (dependency)	Framendinn skal hafa sem færstar ytri ósjálfgæfnar háðir (third-party dependencies). Hver ósjálfgæfin háð skal vera rökstudd og skjalfest. Framendinn skal reiða sig á innbyggða React virkni og staðlaða vefstaðla (HTML, CSS) fremur en þriðja aðila söfn.
Enginn falin flækjustig	Framendinn skal ekki innihalda sjálfstætt rökfræðilag, flókna stöðuvél (state machine), eigin högunarmynstur eða aðra arkitektúr sem gerir endursmíði erfiðari.

Verkkaupi metur flækju við endursmíði sem hluta af viðtökuferli. Ef framendi uppfyllir ekki þessi viðmið getur verkkaupi krafist endursmíði eða einföldunar áður en lausn er samþykkt.

8.2.2.7 Handvirkar breytingar, skilgreining og takmarkanir

Handvirkar breytingar eru breytingar sem gerðar eru beint í kóðagrunni framendans án aðstoðar gervigreindarverkfæra. Slíkar breytingar eru einungis heimilar á:

Tegund skráar	Dæmi	Heimil breyting
Stillingaskrár	.env, vite.config.ts, tsconfig.json	Umhverfisbreytur, smíðastillingar, þýðandastillingar
Staðfæringaskrár (i18n/l10n)	locales/is.json, messages.is.ts	Íslenskur viðmótstexti, villuboð, staðfestingarboð (validation messages)

Handvirkar breytingar mega **aldrei** ná til:

- React-íhluta (components), síðna (pages/views) eða leiðar (routing),
- domain rökfræði, gagnameðhöndlunar eða API-samskiptalags,
- stílsniða (CSS) umfram stakar leiðréttingar á texta,
- prófunarkóða.

Ef breyting fellur utan framangreindra marka skal hún fara í gegnum gervigreindarverkfæri.

8.2.2.8 Smíði og staðfesting (yfirlýsingarskylda)

Verksali skal undirrita skriflega yfirlýsingu þar sem staðfest er að framendakóði hafi verið smíðaður í samræmi við skilyrði Leiðar B. Yfirlýsingin skal ná til eftirfarandi atriða:

Atriði	Yfirlýsing
Smíðaaðferð	Framendakóði var smíðaður eingöngu með gervigreindarþróunarverkfærum.
Handvirkar breytingar	Handvirkar breytingar voru takmarkaðar við stillingaskrár og staðfæringaskrár samkvæmt skilgreiningu.
.agent/ mappa	.agent/ mappa var viðhaldið stöðugt á meðan á smíði stóð og endurspeglar þekkingarsöfnun gervigreindarverkfæranna.
Flækja við endursmíði	Framendinn uppfyllir viðmið um flækju við endursmíði samkvæmt skilgreiningu.

Verkkaupi áskilur sér rétt til stikkprófana á commit-sögu, .agent/ innihaldi og öðrum gögnum til að sannreyna yfirlýsinguna. Röng eða villandi yfirlýsing telst brot á samningi.

8.2.2.9 Keyrsla og dreifing

Framendinn skal vera smíðaður (built) í statískar skrár (static assets: HTML, JavaScript, CSS) sem ekki krefjast Node.js keyrslu.

Tveir dreifingarmöguleikar eru heimilir:

Dreifingaleið	Lýsing
---------------	--------

Frá .NET bakenda	Statískar skrár afhentar úr wwwroot eða sambærilegri möppu ASP.NET þjónustu.
Sjálfstæður docker gámur	Statískar skrár afhentar úr léttum skráarþjóni (t.d. nginx, caddy). Gámurinn skal ekki innihalda Node.js runtime.

Val á dreifingaleið skal vera skjalfest í arkitektúrákvörðunum (ADR).

Node.js er heimilt í þróunar- og CI/CD umhverfi (t.d. til smíði, prófana og kóðagreiningar) en má aldrei vera hluti af keyrsluafurð í rekstrarumhverfi.

Ef framendinn er drifinn sem sjálfstæður gámur skal hann uppfylla sömu rekstrarkröfur og aðrir gámar verkkaupa hvað varðar heilsuendapunkta, útgáfuendapunkt (/version) og dreifingarstefnu (sjá kafla 18 og 19), að svo miklu leyti sem þær eiga við statískan skráarþjón.

8.2.3 Óheimilar framendatæknilausnir

Eftirfarandi eru óheimilar í öllum lausnum sem afhentar eru verkkaupa til reksturs, óháð því hvort Leið A eða Leið B er valin:

Tækni eða nálgun	Staða	Ástæða
Angular, Vue, Svelte og álíka	Óheimilt	Ekki hluti af samþykktum tæknistafla
Next.js, Remix, Gatsby, Astro	Óheimilt	Meta-framework sem krefst Node.js keyrslu eða BFF-högunar
Node.js bakendi eða millilög (NestJS, Express, Fastify)	Óheimilt	API-þróun eingöngu í .NET/C#
Framendar sem keyra að mestu í vafra með flókna stöðustjórnun	Óheimilt	Samræmist ekki langtímarekstri
ASP.NET WebForms	Óheimilt	Úreitt tækni
Vefumsjónarkerfi (CMS)	Óheimilt nema skrifleg undanþága	Krefst sérstaks samþykkis bæði tæknilegra sérfræðinga og verkefnastjórnar

8.2.3.1 Af hverju eru vefumsjónarkerfi (CMS) óheimil?

Vefumsjónarkerfi (t.d. Umbraco, Optimizely, Sitecore, WordPress, Drupal) eru algengar lausnir fyrir efnisstjórnun og vefbirtingu. Þau samræmast hins vegar ekki forsendum tæknistefnu verkkaupa af eftirfarandi ástæðum:

Áskorun	Lýsing
Samhliða tæknistafla	CMS-kerfi innleiða eigin tæknistafla (gagnagrunnshögun, framendahögun, viðbætur (plugins), sniðmátamál og dreifingarlíkan) samhliða samþykktum .NET/C# stafla verkkaupa. Þetta tvöfaldar tæknilega flækju og krefst sérhæfðrar þekkingar sem er aðskilin frá kjarna tæknistafla verkkaupa.
Gagnagrunnshögun utan stjórnar	CMS-kerfi stýra eigin gagnagrunnsskema. Verkkaupi getur ekki beitt sömu gagnagrunnsreglum (handgerðar SQL-breytingar, lágmarksréttindi, bann við sjálfvirkri skemabreytingu kafla 17) á gagnagrunn sem CMS-kerfið stjórnar sjálft.
Viðbóta- og uppfærsluháð	CMS-kerfi reiða sig á viðbætur (plugins/modules) þriðja aðila sem krefjast eigin uppfærsluferla, öryggisúttekta og samhæfingarprófana. Þetta er ósamrýmanlegt kröfu um að allur kóði og söfn í keyrsluvöru fylgi í kóðageymslum verkkaupa (kafla 10 og 12).
Langtíma birgjalás	CMS-framleiðandi getur breytt leyfiskjörum, hætt stuðningi eða úreitt tæknistafla sinn. Yfir 20–30 ára líftíma kerfis skapar þetta þvingaða flutningsáhættu (forced migration) sem fjármögnunarlíkan verkkaupa gerir ekki ráð fyrir.
Yfirtökuáhætta	Ef skipt er um verksala þarf nýr aðili að búa yfir CMS-sérhæfingu til viðbótar við .NET/C# þekkingu. Þetta þrengir hóp mögulegra verksala og eykur kostnað við yfirtöku (kafla 7.2).
Óþarft fyrir tilgang verkkaupa	Efnisstjórnunarþarfir verkkaupa er hægt að mæta með sérsniðnu stjórnviðmóti í núverandi .NET stafla án þess að innleiða samhliða tæknistafla. Slíkt stjórnviðmót fellur undir sömu gæða-, öryggis- og rekstrarkröfur og önnur virkni.

Ef verkefni krefst efnisstjórnunarvirkni skal hún útfærð sem hluti af bakendalausn í samþykktum tæknistafla (C#/.NET) og fylgja sömu hönnunarreglum og önnur virkni. Ef sýnt þykir að sérstakt CMS-kerfi sé nauðsynlegt skal fylgja undanþágufærli (sjá viðauka „Undanþágufærill frá stefnu verkkaupa“).

8.2.4 JavaScript / TypeScript, sameiginlegar reglur

Eftirfarandi reglur gilda um notkun JavaScript og TypeScript í báðum leiðum:

Regla	Leið A	Leið B
Hlutverk	Hjálpartæki fyrir afmarkaða virkni í birtingarlagi. Aldrei burðarás.	TypeScript er burðarás framenda. Domain rökfræði og stöðustjórnun skulu þó vera í algjöru lágmarki.
Domain rökfræði í vafra	Óheimilt	Óheimilt, domain rökfræði skal vera í .NET bakenda.
TypeScript strict mode	Krafa í eldri lausnum og undanþágum (sjá viðauka)	Krafa (sjá viðauka).
ESLint	Krafa í eldri lausnum og undanþágum (sjá viðauka)	Krafa (sjá viðauka).
API-þróun í TypeScript/JavaScript	Óheimilt	Óheimilt, API-þróun eingöngu í .NET/C#.

8.3 Hönnunarstaðlar

Allar framendalausnir skulu fylgja útgefnum hönnunarstöðlum og útlitsstílsniði verkkaupa, þar á meðal:

- Hönnunarhandbók verkkaupa
- Hönnunarkerfi verkkaupa (Design system)
- Storybook leiðbeiningar verkkaupa

Frávik frá samþykktum hönnunarstöðlum eru einungis heimil með formlegu samþykki.

Sjá Hönnunarhandbók verkkaupa og viðauka „Þjónustur og verkfæri verkkaupa“ fyrir tengla og aðgang.

8.4 Snjallforrit / Mobile

Meginregla verkkaupa er að virkni sem ætluð er snjalltækjum skuli vera útfærð sem snjalltækjavæn veflausn fremur en sem sérstakt snjallforrit. Slíkar veflausnir skulu fylgja sömu tæknistefnu og aðrar framendalausnir.

8.4.1 Rökstuðningur

Native snjallforrit (Android og iOS) krefjast aðskilinna þróunarferla, aðskilinna kóðagrunna og aðskilinna útgáfufarla sem margfalda kostnað og flækjustig. Í útboðslíkani verkkaupa vega eftirfarandi þættir þungt:

Áskorun	Lýsing
Tvöfaldur þróunar- og viðhaldskostnaður	Native snjallforrit krefjast tveggja aðskilinna kóðagrunna (iOS og Android) með mismunandi forritunarmálum, verkfærum og prófunarumhverfum. Þetta tvöfaldar viðhaldskostnað sem verkkaupi ber í mörg ár, kostnað sem hægt er að forðast með einni snjalltækjavænni veflausn.
Óstýranleg útgáfudreifing	Uppfærslur á native snjallforritum krefjast samþykkis Apple og Google (App Store Review), sem getur tekið daga eða vikur. Notendur þurfa einnig að uppfæra sjálfir, sem þýðir að margir keyra gamlar útgáfur samtímis (fragmentation). Veflausnir eru uppfærðar tafarlaust og allir notendur sjá sömu útgáfu. Þetta skiptir sérstaklega máli þegar öryggisuppfærslur eða lagfæringar á heilbrigðisgögnum þurfa að taka gildi strax.
Ósamræmi við samþykktan tæknistafla	Native snjallforrit krefjast Swift/Objective-C (iOS) og Kotlin/Java (Android), forritunarmál sem falla utan samþykks tæknistafla verkkaupa (C#/.NET). Þetta skapar sérhæfða þekkingarsíló sem er erfitt að manna á Íslandi og torveldar samræmda kóðarýni og gæðafarla.
Flóknari yfirtaka	Ef verksali lýkur samningi þarf nýr verksali að yfirtaka tvo viðbótarkóðagrunna á forritunarmálum sem eru ólík öðrum kerfum verkkaupa. Þetta eykur áhættu og kostnað við yfirtöku (sjá kafla 7.2 um úttektaráætlun).

Þriðja aðila dreifingaháð	Native snjallforrit eru háð App Store og Google Play, stöðlum, reglum og stefnum sem verkkaupi hefur enga stjórn á. Breytingar á dreifingarstefnu þessara vettvanga geta haft áhrif á aðgengi og uppfærsluferli lausna verkkaupa.
Vettvangsdrifið viðhald (platform churn)	Apple og Google breyta reglulega kröfum til snjallforrita, t.d. lágmarks API-stig, nýjar persónuverndarkröfur, skylduuppfærslur á SDK og úrelding API-viðmóta. Slíkar breytingar krefjast kóðabreytinga, prófana og nýrrar útgáfu sem er ótengd viðskiptaþörfum verkkaupa. Þetta gengur beint gegn meginreglu handbókarinnar um að lágmarka viðhald sem er knúið af ytri tæknivettvangi frekar en viðskiptalegum kröfum (kaflí A1).

Snjalltækjavænar veflausnir hafa ekkert af þessum vandamálum og uppfylla þarfir flestra notendahópa verkkaupa. Undanþágur eru einungis heimilar þegar tæknilegar takmarkanir veflausna eru raunverulegar og rökstuddar, sjá hér að neðan.

8.4.2 Cross-platform lausnir (React Native, Flutter, .NET MAUI o.þ.h.)

Cross-platform frameworks eru hugbúnaðarumhverfi sem gera kleift að skrifa einn kóðagrunn sem þýðist yfir í native snjallforrit fyrir bæði iOS og Android. Þekktustu lausnirnar eru React Native (Meta), Flutter (Google) og .NET MAUI (Microsoft).

Cross-platform lausnir eru óheimilar á sama grundvelli og native snjallforrit, enda framleiða þær native útgáfueiningu sem dreift er í gegnum App Store og Google Play. Þær leysa vandamálið um tvöfaldan kóðagrunn en leysa ekki þau vandamál sem skipta mestu máli fyrir verkkaupa:

Vandamál	Leyst?	Lýsing
Tvöfaldur kóðagrunnur	Að hluta	Einn kóðagrunnur, en vettvangssértæk útfærsla (native modules, platform channels) er oft nauðsynleg og krefst þekkingar á iOS og Android innviðum.
Óstýranleg útgáfudreifing	Nei	Útgáfur fara áfram í gegnum App Store og Google Play. Notendur þurfa að uppfæra. Öryggislagfæringar taka jafn langan tíma og í hreinum native forritum.
Þriðja aðila dreifingaháð	Nei	Lausnin er áfram háð Apple og Google reglum og samþykktarferlum.
Vettvangsdrifið viðhald (platform churn)	Nei	Apple og Google breyta reglulega kröfum til snjallforrita, t.d. lágmarks API-stig, nýjar persónuverndarkröfur, skylduuppfærslur á SDK og úrelding API-viðmóta. Slíkar breytingar krefjast kóðabreytinga, prófana og nýrrar útgáfu sem er ótengd viðskiptaþörfum verkkaupa. Þetta gengur beint gegn meginreglu handbókarinnar um að lágmarka viðhald sem er knúið af ytri tæknivettvangi frekar en viðskiptalegum kröfum (kaflí A1).
Flóknari yfirtaka	Að hluta	Einn kóðagrunnur auðveldar yfirtöku, en þekking á viðkomandi krossvettvangstækni (React Native, Dart, MAUI) er enn sérhæfð og aðskilin frá meginþekkingu verkkaupa (.NET/C# bakendar og ASP.NET framendar).
Ósamræmi við tæknistafla	Mismunandi	React Native krefst JavaScript/TypeScript sem burðarás og Node.js verkfæra, óheimilt samkvæmt kafla 8.1 og 8.2. Flutter krefst Dart, forritunarmáls sem er utan samþykks tæknistafla og uppfyllir ekki matsþætti kafla A1 (mannafli, bakhjarl, þroski) á íslenskum markaði. .NET MAUI er næst samþykktum stafla (C#/.NET) en er tiltölulega ungt umhverfi með takmarkaða reynslu í stórum heilbrigðiskerfum, og framleiðir engu að síður native dreifingareiningu sem háð er App Store ferlum.

Ef undanþága frá veflausn er veitt (sjá undanþáguskilyrði hér að neðan) og native snjallforrit er samþykkt, skal val á tækni, hvort sem um er að ræða hreint native eða krossvettvangslaun, vera hluti af formlegri arkitektúrákvörðun (ADR) og háð samþykki tæknilegra sérfræðinga verkkaupa.

Þróun native snjallforrita (Android eða iOS) er skilgreind sem undanþága og fellur undir formlegan undanþáguferil.

8.4.3 Hvenær native snjallforrit eru heimil

Undanþága getur einungis komið til álita ef eitt eða fleiri af eftirfarandi skilyrðum eru uppfyllt:

- ekki er tæknilega mögulegt að útfæra nauðsynlega virkni með veflausn,
- lausnin krefst sértækrar tækjavirkni (t.d. lífkennagreining, viðvarandi bakgrunnskeyrsla, sértækir tækjaskynjarar, viðvarandi offline vinnslu),
- rökstutt er með formlegum hætti að veflausn uppfylli ekki kröfur verkefnis.

8.4.4 Aðgengi og útbreiðsla

Öll viðmót snjallforrita skulu vera hönnuð þannig að virkni og notendaupplifun sé ásættanleg á að lágmarki 75% þeirra snjalltækja og stýrikerfisútgáfa sem eru í notkun hjá skilgreindum markhópi á útgáfudegi.

8.4.5 Afmörkun gagnvart SPA

Native snjallforrit skulu ekki vera notuð til að komast hjá samþykktari framendastefnu. Lausnir sem fela í sér SPA-hegðun, umfangsmikla client-side stöðustjórnun eða JavaScript/TypeScript sem burðarás eru óheimilar, óháð afhendingarformi.

9 Hönnun og arkitektúr

Allur arkitektúr og hönnun hugbúnaðarlausna skal vera unnin í nánu samráði við tæknilega sérfræðinga verkkaupa. Hugbúnaðarsmíði má ekki hefjast fyrr en formlegt samþykki liggur fyrir á:

- heildararkitektúr lausnar,
- högun og samskiptum eininga,
- gagnalíkani og venslum gagna.

Allar verulegar breytingar á samþykktari hönnun eða arkitektúr skulu lagðar fyrir verkkaupa til samþykktar áður en þær eru innleiddar.

9.1 Meginreglur

Allur hugbúnaður sem smíðaður er fyrir verkkaupa skal fylgja eftirfarandi meginreglum nema formleg undanþága hafi verið veitt:

Meginregla	Lýsing
SOLID	Single Responsibility, Open/Closed, Liskov Substitution, Interface Segregation, Dependency Inversion
SoC	Separation of Concerns, skýr aðgreining ábyrgðar milli eininga
ONION / Multilayered	Lagskiptur arkitektúr þar sem innri lög eru óháð ytri lögum
CLEAN code	Læsilegur, viðhaldshæfur og fyrirsjáanlegur kóði
KISS	Keep It Simple, forðast óþarfa flækjustig
DRY	Don't Repeat Yourself, forðast tvíverknað
YAGNI	You Aren't Going to Need It, útfæra ekki virkni sem ekki er krafist

Arkitektúr skal aðgreina skýrt eftirfarandi þætti:

- móttöku og afhendingu gagna (request/response),
- stýringar (controllers / endpoints),
- flæðistjórnun innan kerfisins,
- gagnasókn og gagnaskrif,
- ytri og innri gagnahögun (domains / models).
- Viðmiðunararkitektúr
- Lausnir skulu byggja á rótæklega einföldum tækniarkitektúr sem lágmarkar fjölda tæknieininga, keyrsluumhverfa og sundurliðunar.
- Einföldun er meðvituð hönnunarákvörðun sem stuðlar að:
- styttri lærdómskúrfu nýs tæknifólks,
- minni rekstraráhættu og bilanatíðni,
- einfaldari uppfærslum og villuleit,
- minni háðu við sérþekkingu einstaklinga.

Lausnir sem fela í sér óþarfa flækju, fjölda smáeininga, eða mikla dreifingu ábyrgðar milli tæknilaga eru óæskilegar og verða ekki samþykktar nema rökstudd undanþága liggi fyrir.

Sjá viðauka „Viðmiðunararkitektúr, þjónustur og dæmi um lausnahögun“ fyrir skýringarmyndir og samanburð á æskilegu og óæskilegu flækjustigi.

9.2 Þjónustuhögun

Verkkaupi tekur ekki við hugbúnaði til reksturs sem er samsettur úr smáþjónustum (micro- eða nano-services).

9.2.1 Rökstuðningur

Smáþjónustuarkitektúr (microservices) er viðurkennd lausn á vandamálum sem stórar stofnanir með tugi eða hundruð þróunarteymi glíma við, sérstaklega þegar einstök teymi þurfa sjálfstæðar útgáfur og teymisskilin falla saman við þjónustuskilin. Þessi vandamál eiga ekki við um verkkaupa:

Forsenda smáþjónusta	Raunveruleiki verkkaupa
Stór skipulagsheildir með tugum teymi sem hindra hvert annað í sameiginlegum kóðagrunni	Lítill teymi (2–5 manns), oft eitt teymi per verkefni. Ekkert skipulagslegt vandamál sem krefst þjónustuskipta.
Sérhæfð rekstrarteymi (DevOps/SRE) sem viðhalda dreifðum innviðum	Sameiginlegir Kubernetes innviðir verkkaupa sem þjónusta öll kerfi. Hver viðbótarþjónusta eykur rekstrarálags á innviðateymi verkkaupa.
Stöðugt fjármagn til reksturs og viðhalds dreifðra kerfa	Opinber verkefni fjármögnuð með afmörkuðu stofnfjármagni. Reglulegt viðhaldsfjármagn takmarkað. Hver viðbótarþjónusta margfaldar rekstrarkostnað sem verkkaupi hefur ekki fjármagn til að mæta til langs tíma.
Stór kóðagrunnar þar sem breytingar hafa ófyrirsjáanleg áhrif	Vel lagskiptur kóðagrunnur samkvæmt kafla 9.6 þar sem Onion-arkitektúr og .csproj verkefnaaðskilnaður tryggja afmörkun áhrifa innan eins kerfis, án rekstrarkostnaðar dreifðs kerfis.

Hver smáþjónusta sem bætt er við krefst sjálfstæðs: CI/CD ferils, Docker stillinga, Kubernetes uppsetningar, heilsuendapunkta, mælinga, loggunar, rekstrarskjölunar, gagnagrunns og útgáfustýringar. Þetta margfaldar rekstrarkostnað línulega við fjölda þjónusta, kostnað sem verkkaupi ber sjálfur og verður að viðhalda í 20–30 ár, löngu eftir að upprunalegt þróunarteymi er hætt störfum.

Þar að auki eykur smáþjónustuarkitektúr birgjalás: eftir því sem þjónustum fjölgar eykst þekkingin sem einungis upprunalegt teymi býr yfir, sem gerir yfirtöku nýs aðila flóknari og kostnaðarsamari. Þetta gengur beint gegn markmiðum verkkaupa um auðvelda yfirtöku og óháð verksala (sjá kafla 7.1 og tæknistefnu verkkaupa um úttektaráætlun).

Bakendalausnir skulu fylgja þjónustuhögun (service-oriented architecture) með áherslu á stærri, sjálfbærar þjónustueiningar.

Krafa	Lýsing
Ábyrgðarsvið	Hver þjónusta skal hafa skýrt skilgreint ábyrgðarsvið
Sjálfbærni	Þjónusta skal innihalda alla þá virkni sem henni er ætlað að sinna
Dreifing	Óheimilt er að sundurliða lausnir í margar sjálfstæðar þjónustur nema rökstutt samþykki verkkaupa liggi fyrir

9.2.2 Arkitektúrhögun þjónusta

Þar sem við á skal arkitektúr bakendaþjónusta miða við:

Högun	Lýsing
Vertical Slice Architecture	Löðréttar sneiðar fremur en hefðbundin lagskipting
REPR (Request–Endpoint–Response)	Fyrir WebAPI lausnir; fremur en MVC högun
Skýr aðgreining	Ysta beiðnalag aðskilið frá innra domain flæði

9.2.3 Bakgrunnsvinnslur

Bakgrunnsvinnslur (scheduled jobs, recurring tasks) skulu vera útfærðar sem sjálfstæð forrit (console applications) sem keyra sem Kubernetes CronJobs.

Óheimilt er að nota:

Tækni	Staða
Hangfire, Quartz.NET eða álíka in-process job schedulers	Óheimilt
Database scheduled jobs	Óheimilt (sjá kafla <i>Gagnagrunnar</i>)
Bakgrunnsþráðir sem halda stöðu í minni	Óheimilt

Rökstuðningur

In-process job schedulers (Hangfire, Quartz.NET o.þ.h.) halda stöðu í minni eða gagnagrunni, sem brýtur gegn kröfu um stöðuleysi (stateless) og gerir lárétta skölun ómögulega. Þeir búa til falda bakgrunnsþráði sem torvelða villuleit og rekstrareftirlit, og bæta sjálfvirkt við rekstrarviðmótum og endapunktum (t.d. Hangfire Dashboard) sem brjóta gegn öryggiskröfu um fulla stjórn verkkaupa á öllum endapunktum (sjá kafla 12). Kubernetes CronJobs eru innviðanálgun sem er hluti af stöðluðu keyrsluumhverfi verkkaupa, bakgrunnsvinnslur eru sýnilegar, rekjanlegar og viðhaldshæfar með sömu verkfærum og öll önnur keyrslueiningu.

Bakgrunnsvinnslur skulu vera:

- stöðulausar (stateless),
- endurtekningarhæfar (idempotent),
- rekjanlegar í loggum og mælingum,
- keyrsluhæfar í Docker/Kubernetes umhverfi.

9.2.4 Caching

Þjónustur skulu nýta tímabundnar gagnageymslur (caching) þar sem það á við, til að bæta afköst og draga úr álagi á gagnagrunna.

Caching skal ávallt vera útfært á netþjóni og vera hluti af samþykktari arkitektúrhögun. Ekki skal færa caching-rökfræði í framenda.

9.3 Samskipti við ytri þjónustur og samvirkni

9.3.1 OpenAPI og forritunarskil

Allar þjónustur skulu hafa OpenAPI-skilgreiningar sem lýsa forritunarskilum þeirra með fullnægjandi hætti til að styðja sjálfvirka prófun, kóðagerð og skjölun.

Skilgreiningin skal meðal annars lýsa endapunktum, inntaks- og skilagögnum, villukóðum, auðkenningarmáta og aðgangsheimildum per endapunktur.

Þjónustur skulu vera prófanlegar á sjálfvirkan hátt án háðs við raunverulegar ytri auðkenningarþjónustur.

Sjá viðauka „Kröfur til OpenAPI skilgreininga“ fyrir fulla kröfulýsingu.

9.3.2 Gagnasnið

Gögn sem eru send og móttækin af þjónustum skulu vera á JSON sniði. Innri þjónustur geta nýtt sér gRPC/Protobuf ef við á.

Ekki skal gera greinamun á stórum og litlum stöfum í svæðaheitum í JSON skeytum (t.d. „data“, „Data“, „DATA“ eru allt sama svæðið).

9.3.3 Heilbrigðisstaðlar

Þar sem við á skulu kerfi:

- lesa, meðhöndla og skrifa gögn samkvæmt HL7 staðli eins og hann er skilgreindur á landsvísu,
- nýta miðlægar FHIR innviðalausnir verkkaupa,
- forðast sértækar eða staðbundnar gagnaskilgreiningar nema sérstaklega samþykkt.

9.3.4 Nafnareglur þjónusta

Aðgreina skal þjónustur sem eru aðgengilegar á internetinu frá þeim sem sitja bak við eldvegg:

Viðskeyti	Aðgengi	Regla
-----------	---------	-------

-api	Aðgengilegt utan eldveggis (internetinu)	—
-service	Einungis aðgengilegt innan innra nets	Skal aldrei vera aðgengilegt af internetinu

9.4 Þolgæði

Allar lausnir skulu vera hannaðar með þolgæði (fault tolerance) að leiðarljósi og geta tekist á við tímabundnar og viðvarandi truflanir í samskiptum við ytri þjónustur, gagnagrunna og innviði.

Þolgæði skal útfært með viðurkenndum mynstrum og forritunarsöfnum, ekki sérsníðnum lausnum.

Meginregla	Lýsing
Stýrðar endurtekingar (retry)	Með takmörkunum, stigvaxandi biðtímum og slembun
Rofavarnir (circuit breakers)	Til að koma í veg fyrir keðjuverkandi bilanir
Aðgreining villna	Skýr aðgreining á varanlegum og tímabundnum villum
Viðurkennd söfn	Óheimilt er að innleiða sérsníðnaðan þolgæðakóða

Þolgæðisrökfræði, stöðustjórnun og villumeðhöndlun skulu vera á netþjóni. Ekki skal færa þessa rökfræði í framenda.

Sjá viðauka „Þolgæði, dæmi og útfærslur“ fyrir viðurkennd forritunarsöfn og útfærsluleiðbeiningar.

9.5 Lagskipting og skipulag lausna

Allar lausnir skulu vera skipulagðar með skýrri lagskiptingu og fyrirsjáanlegri möppuskipting. Skipulag skal vera einfalt, samræmt og auðvelt í yfirlestri fyrir nýtt tæknifólk.

9.5.1 Af hverju aðskilin verkefni (.csproj) og möppuskipting, ekki nafnabil (namespaces)?

Í .NET er hægt að útfæra lagskiptingu á tvo vegu: annars vegar með namespaces innan eins verkefnis (project), hins vegar með aðskildum möppum og verkefnum (.csproj) sem hvert um sig tákna eitt lag arkitektúrsins. Verkkaupi krefst þess síðarnefnda.

Ástæðan fyrir því er þegar hvert lag er aðskilt verkefni (.csproj) verður þýðandinn (compiler) sjálfur að framfylgjanda arkitektúrreglna. Ef Managers/ verkefnið vísar ekki í Endpoints/ verkefnið í ProjectReference þá er ómögulegt fyrir kóða í Managers/ að kalla á Endpoints/, þýðing mistekst. Arkitektúr er vélrænt framfylgt í hverri einustu þýðingu.

Namespace-byggð eftirfylgni (namespace based enforcement) krefst sérhæfðra greiningartóla til að uppgötva brot á lagskiptingarreglum. Þessi töl greina einungis þýddan kóða (compiled CIL bytecode), krefjast .NET SDK, og villur eru ósýnilegir nema prófanir séu keyrðar. Þar að auki er namespace textayfirlýsing sem forritari getur breytt án þess að færa skrá, namespace MyApp.Domain.Entities getur orðið að namespace MyApp.Application.Services með einni línu, án þess að skráin hreyfist úr stað. Þetta gerir namespace-skiptingu veikari en möppu-skiptingu þar sem skrá sem er í Domain-verkefni er ávallt hluti af Domain-assembly, óháð nafnabilayfirlýsingunni.

Verkkaupi leggur þó áherslu á að namespace endurspegli möppuskipting. Roslyn-greiningartólið IDE0130 (*Namespace does not match folder structure*) skal vera virkt á villustig (error severity) í öllum lausnum verkkaupa.

9.5.2 Af hverju möppuskipting fremur en nákvæm tæknileg skjölun eingöngu?

Möppuskipting er sú mynd af arkitektúrnum sem er sýnileg öllum, án sérhæfðra tóla. Verkkaupa, nýju teymi, gervigreindarþróunarverkfærum og ytri úttektaraðilum, öllum nægir að skoða skráarkerfið til að skilja arkitektúrinn. Sannprófun á möppuskipting krefst engra sérhæfðra .NET tóla: einfalt bash script getur staðfest tilvist réttra möppa og greint óheimilar ProjectReference tilvísanir í .csproj skráum á nokkrum sekúndum, án þess að þýða kóðann. Þetta er sérstaklega mikilvægt í innkaupasamhengi þar sem verkkaupi þarf að geta sannreynt arkitektúrsamræmi óháð verkfærum verksala.

9.5.3 Af hverju samræmd möppunöfn milli kóðageymsla?

Samræmd nöfn á rótarmöppum (Endpoints/, Managers/, Repositories/, Core/ o.s.frv.) tryggja að allir sem koma að verkefnum verkkaupa, hvort sem það er nýr forritari, nýtt teymi eða gervigreindarþróunarverkfæri, finni strax fyrir skipulagi kóðans. Gervigreindarþróunarverkfæri kortleggja möppuskipting sem fyrsta skref þegar þau opna nýja kóðageymslu, skráarslóðir eru aðalviðmið um samhengi, ekki namespace í kóðaskráum. Þegar öll kóðageymsla verkkaupa fylgja sama möppuskipting þarf ekki að læra skipulag hvers

verkefni sérstaklega og gervigreindarverkfæri geta nýtt eina skilgreiningu um hvar kóði á heima. Sérsmíðuð eða frjáls skipulagsmynstur eru óheimil.

9.5.4 Af hverju þriggja laga aðskilnaður?

Aðskilnaður ysta beiðnalags (Endpoints/ eða Pages/) frá flæðistjórnunarlagi (Managers/) og gagnasóknarlagi (Repositories/) er lykilforsenda langtíma kóðagæða:

- **Ysta lag (Endpoints/, Pages/)** tekur á móti beiðnum, sannreynir inntak og skilar svari. Það veit ekki hvernig gögn eru sótt eða unnin.
- **Flæðistjórnunarlag (Managers/)** heldur utan um domain flæði og reglur. Það getur kallað á gagnasóknarlög og þjónustur en er óháð ysta lagi. Ef skipta þarf um framenda (t.d. úr API í bakgrunnsvinnslu) breytist Manager-kóði ekki.
- **Gagnasóknarlag (Repositories/)** sækir og skrifar gögn, hvort sem er úr gagnagrunni (DataAccess), ytri þjónustu (ApiAccess), biðröð (QueueAccess) eða skráarkerfi (FileAccess). Það vinnur með gagnamódel, ekki domain módel, og er auðvelt að skipta út án þess að snerta flæðistjórnun.

Þessi aðskilnaður gerir það að verkum að breytingar á einum hluta (t.d. nýr API, nýr framendi) hafa afmörkuð áhrif og krefjast ekki endursmíði á ótengdum lögum. Hann tryggir jafnframt að nýir aðilar geti tekið yfir einstaka hluta kerfisins án þess að þurfa djúpan skilning á heildinni.

9.5.5 Leið B, framendi

Leið B er önnur framfylgdarleið en lýtur sömu meginreglum. Ofangreind lagskipting með aðskildum .csproj verkefnum á við Leið A (.NET bakendi og framendi) og allan bakendakóða, óháð framendaleið. Leið B framendi (React í TypeScript, sjá kafla 8.2.2) fylgir sömu meginreglum um samræmda möppuskipan og skýra aðgreiningu, en framfylgdarleiðin er önnur vegna eðlis tæknistafla og tilgangs framendans.

Leið B framendi er eitt verkefni (ekki mörg .csproj) og er viljandi haldið einföldu og skipulag flatt, hámark þrjú möppustig. Í stað þýðanda eftirlits (compiler) byggir Leið B á þremur stoðum:

- **Einfaldleiki.** Leið B framendi er hannaður til tíðrar endursmíða, gervigreindarverkfæri skulu geta endursmíðað hann í heild sinni (sjá kafla 8.2.2). Flókið skipulag myndi grafa undan þessari forsendu. Framendi sem er nógu einfaldur til fullrar endursmíði þarf ekki djúpa lagskiptingu til að viðhalda sér yfir tíma, vegna þess að hann er ætlaður til endurnýjunar, ekki viðhalds.
- **ESLint reglur.** ESLint regluset (eslint-plugin-import, no-restricted-paths) skulu framfylgja include og skipulagi: src/pages/ getur flutt inn kóða úr src/components/ og src/services/ en ekki öfugt. Þetta kemur ekki í stað þýðanda eftirlits .csproj en tryggir sjálfvirka sannprófun á réttleika í CI/CD ferlum.
- **Samræmd möppuskipan.** Rótarmöppunöfn Leiðar B (src/components/, src/pages/, src/services/, src/types/, Tests/) eru stöðluð fyrir allar Leið B kóðageymslur verkkaupa. Sami rökstuðningur gildir og fyrir Leið A: samræmi milli kóðageymsla, engir sérsmíðaðir möppunafnastílar, og gervigreindarverkfæri þurfa aðeins eina skilgreiningu á hvar kóði á heima. Í Leið B er þetta enn mikilvægara en í Leið A vegna þess að gervigreindarverkfæri eru smiðurinn.
- Aðgreiningu í skipulagi í Leið B framenda má lýsa svona:
- **src/pages/ eða src/views/**, skjámyndir og leiðir (routing). Samsvarar ysta lagi (Endpoints/) í Leið A.
- **src/components/**, endurnýtanlegir React-íhlutir. Viðmótseiningar sem vita ekki hvaðan gögn koma.
- **src/services/ eða src/api/**, gagnasókn og API-samskipti við .NET bakenda. Samsvarar gagnasóknarlagi (Repositories/) í Leið A.
- **src/types/**, TypeScript-týpur, oft myndaðar sjálfvirkir úr OpenAPI skilgreiningum bakenda. Samsvarar Core/ í Leið A.

Öll domain rökfræði og flæðistjórnun (Managers/) er í .NET bakenda, aldrei í Leið B framenda. Ef framendi krefst sjálfstæðrar rökfræði eða flæðistjórnunar er það vísbending um að Leið B eigi ekki við og verksali skuli nota Leið A (sjá kafla 8.2.2).

Nánari skilgreiningar á möppuskipan Leiðar B eru í viðauka „Lagskipting og möppuskipan lausna.“

9.5.6 Meginreglur

- Hvert lag skal vera aðskilt verkefni (.csproj) og hvert verkefni skal hafa afmarkað hlutverk.
- Tengingar milli verkefna (ProjectReference) skulu einungis vísa í þá átt sem lagskiptingarreglur leyfa. Þýðandinn framfylgir þessu sjálfvirk.
- Namespace skulu endurspegla möppuskipan. IDE0130 skal vera á villustig í öllum lausnum.
- Forðast skal hringháð (circular dependencies).
- Domain flæði og kjarna-rökfræði (Managers/, Core/) skulu aldrei vera háð ysta lagi (Endpoints/, Clients/) eða gagnalagi (Repositories/).
- Verulegar breytingar á rótarmöppuskipulagi skulu ekki gerðar án samþykkis verkkaupa.
- Einingaprófanir skulu sannreyna lagskiptingarreglur, þar á meðal að kóði í tilteknu lagi vísi ekki í lög sem hann á ekki/má ekki eiga samskipti við.

Sjá viðauka „Lagskipting og möppuskipan lausna“ fyrir ítarlega rótarmöppuskipan, lagskiptingarreglur, tengingar heimildir milli laga, skipulag prófunarkóða og lista yfir óheimil arkitektúrhugtök.

10 Forritunarkóði og vinnureglur

Þessi kafl skilgreinir reglur og leiðbeiningar um daglega kóðun, stílsamræmi, virknisflögg, þýðingar, skjölun í kóða og endurnýtingu forritunarsafna.

10.1 Kóðastíll og sjálfvirk samræming

Allur forritunarkóði sem skilað er til verkkaupa skal fara í gegnum sjálfvirka samræmingu og gæðamat (linting, static analysis).

Gæðahlið verkkaupa krefjast þess að viðvaranir séu meðhöndlaðar sem villur (treat warnings as errors). Þetta gildir um þýðanda, kóðagreiningu og stílreglur. Sjá kafla 6 (Gæðafærslur) og viðauka „Kóðastíll og gæðatöl, verkfæri, stillingar og reglusettt“ fyrir nákvæm viðmið og stillingar.

Forritarar skulu halda sig við þann kóðastíl, uppbyggingu og arkitektúr sem fyrir er í lausninni eða er skilgreindur af verkkaupa. Víðtækar breytingar á kóðastíl eða arkitektúr lausnar eru óheimilar án fyrirfram samþykkis.

10.1.1 Umfang breytinga í PR

Breytingabeirðir (Pull Requests) skulu einungis innihalda breytingar sem tengjast skilgreindu verki. Óheimilt er að bæta við óskyldum endurbótum, hreinsunum eða endurskipulagningu í ótengdum hlutum kóðagrunns nema sérstaklega samþykkt.

Tilgangur þessarar kröfu er að tryggja rekjanleika, einfalda kóðarýni og koma í veg fyrir óvart innleiddar villur í ótengdum hlutum lausnar.

10.1.2 Nafnarými og pakkar

Nafnarými (namespaces) og samsvarandi pakkaskipan skulu vera samræmd og fyrirsjáanleg.

Nöfn skulu byrja á einu af eftirfarandi, eftir því sem við á:

- [Digital,Ministry,Directorate].Health.PROJECTNAME
- [Digital,Ministry,Directorate]/Health/PROJECTNAME

þar sem PROJECTNAME er heiti eða stöðluð stytting verkefnis.

10.2 Virknisflögg (Feature flags)

Heimilt er að nota virknisflögg til að stýra aðgengi að virkni, draga úr áhættu við útgáfur og styðja stígvaxandi innleiðingu.

10.2.1 Kröfur um útfærslu

Virknisflögg skulu vera útfærð með:

- innbyggðri virkni í samþykktum forritunarramma (t.d. FeatureFilter í .NET), eða
- build-stýrðum aðferðum sem tryggja að óvirkur kóði sé fjarlægður í PreProd og Production.

10.2.2 Nafnareglur

Nöfn virknisflagga skulu fylgja sniðinu: <scope>.<feature>.<state>

Hluti	Lýsing	Dæmi
scope	Umfang eða lausn	appointments, auth, reporting
feature	Lýsing á virkni	newBookingFlow, auditExport
state	Staða eða tilgangur	enabled, rollout, temporary

10.2.3 Takmarkanir

Virknisflögg má ekki nota til að:

- fela ófullbúna eða óprófaða virkni í rekstrarumhverfi,
- viðhalda varanlegum greinaskilum í kóðagrunni,
- komast hjá samþykktu útgáfu- og gæðastýringarferli,
- flytja rökfræði yfir í framenda.

10.2.4 Hreinsun

Þegar virkni hefur verið innleidd að fullu skal virknisflaggið fjarlægt úr kóðagrunni innan hæfilegs tíma. Öll virknisflögg skulu vera skráð, tengd skilgreindu verki og hafa skilgreindan eiganda.

10.3 Þýðingar

Allar framendalausnir skulu vera hannaðar þannig að hægt sé að birta notendaviðmót á fleiri en einu tungumáli. Að lágmarki skal styðja íslensku og ensku.

10.3.1 Umfang

Þýðingarkrafa nær til alls texta sem birtist notendum:

- texta á stjórnækjum og hnöppum,
- leiðbeininga, hjálpartexta, titla og fyrirsagna,
- villu- og stöðutilkynninga.

10.3.2 Staðfærsla

Þegar tungumáli er breytt skulu allar framsetningar gagna fylgja viðmiðum viðkomandi tungumáls (dagsetningar, tímasnið, tölur, upphæðir).

10.3.3 Þýðingarlyklar

Þýðingar skulu byggðar á skýrum og stöðugum þýðingarlyklum sem endurspeglar virkni eða viðmótseiningu. Þumalputtaregla er að nota forskeyti sem endurspeglar síðu eða einingu, t.d. viewPatient.close.

10.3.4 Texti í myndum

Forðast skal notkun táknmynda eða myndefnis sem inniheldur texta sem þarfnast þýðinga.

10.4 Skjölun í forritunarkóða (comments)

Allur forritunarkóði skal vera skjalfestur með það að markmiði að vera skiljanlegur og viðhaldshæfur fyrir ólíka forritara yfir langt tímabil.

10.4.1 Tungumál og málfar

Allar athugasemdir og leiðbeiningar í kóða skulu vera á ensku. Málfar skal vera hlutlægt og faglegt, án persónulegra skoðana eða óformlegs orðalags.

10.4.2 Hvað skal skjalfesta

Að lágmarki skulu eftirfarandi einingar vera skjalfestar þegar þær eru aðgengilegar utan viðkomandi einingar (public):

- klasar og helstu einingar,

- föll og aðferðir,
- föst gildi (constants) og enum ásamt lýsingum á hverjum gildum,
- inntaksgildi, skilagildi og villuskilyrði.

Skjölun skal svara spurningunni af hverju lausn er útfærð á ákveðinn hátt, ekki einungis hvað kóðinn gerir.

Gæði	Dæmi
Gott	// Confirm that the user has read privileges to the data
Ófullnægjandi	// Call privilege service

10.4.3 Umfang

Of mikil eða ómarkviss skjölun sem endurtekur augljósa virkni kóðans er óæskileg. Skjölun skal vera hnitmiðuð, uppfærð samhliða kóða og samræmd innan lausnar.

10.5 Endurnýting kóða / forritunarsöfn

Verksali er skyldugur til að nýta fyrirliggjandi endurnýtanleg forritunarsöfn verkkaupa þar sem þau eiga við.

Tilgangur þessa er að:

- tryggja samræmi milli lausna,
- draga úr tvíverknaði,
- einfalda viðhald og rekstur,
- styrkja öryggi og rekjanleika.

Óheimilt er að hunsa eða komast hjá notkun staðlaðra forritunarsafna nema rökstutt samþykki tæknilegra sérfræðinga verkkaupa liggja fyrir.

Sjá viðauka „Stöðluð forritunarsöfn verkkaupa“ fyrir lista yfir tiltæk forritunarsöfn.

10.5.1 Notkun forritunarsafna verksala

Ef verksali nýtir eigin forritunarsöfn gilda eftirfarandi reglur:

Tegund safns	Krafa
Opinberlega útgefið (t.d. á nuget.org)	Tilgreina nákvæma útgáfu
Einka- eða óútgefið safn	Kóði viðkomandi útgáfu skal fylgja skilum og vera vistaður í kóðageymslum verkkaupa

10.5.2 Ytri forritunarsöfn eða gagnapakkar

Allur kóði og öll forritunarsöfn sem eru hluti af útgefni keyrsluvöru skulu fylgja lausninni í kóðageymslum verkkaupa. Þetta á m.a. við um:

- þriðja aðila JavaScript pakka, TypeScript skilgreiningar,
- leturgerðir, myndir, stílsnið (CSS),
- og aðrar tengdar skrár.

Lausnir skulu vera sjálfbærar þannig að þær sé hægt að byggja og reka án háðs við ytri netþjónustur sem ekki eru hluti af samþyktri rekstrarhögun.

Öryggiskröfur um ytri forritunarsöfn, þar á meðal bann við forritunarsöfnum sem bæta sjálfvirkt við endapunktum í lausn, eru skilgreindar í kafla 12 (Öryggi og auðkenningar).

11 Gervigreind í hugbúnaðarþróun

 Leiðarljós

Gervigreind er orðin óaðskiljanlegur hluti af nútíma hugbúnaðarþróun. Verkkaupi gerir ráð fyrir, og hvetur eindregið til, þess að verksalar nýti gervigreindarverkfæri á skipulegan og ábyrgan hátt í þróunarferli hugbúnaðar.

Þessi kafli skiptist í tvo hluta:

Hluti	Efni
A. Gervigreindarstudd hugbúnaðarþróun	Reglur, væntingar og viðmið um notkun gervigreindar sem þróunarverkfæri, hvernig verksali notar gervigreind við forritun, skjölun, prófanir og rýni
B. Gervigreindarlausnir sem hluti af afurð	Reglur um hönnun, þróun og rekstur gervigreindareininga (AI agents, MCP) sem verða hluti af keyrsluafurð verkkaupa

11.1 Gervigreindarstudd hugbúnaðarþróun

11.1.1 Væntingar verkkaupa

Verkkaupi gerir sterka væntingu um að verksalar nýti sér gervigreindarverkfæri í hugbúnaðarþróun. Skipulögð og ábyrg notkun gervigreindar stuðlar að hærri gæðum, betri skjölun og skilvirkari þróun.

Gervigreind er þó verkfæri, en þó ekki staðgengill fagþekkingar. Mannlegir forritarar bera fulla ábyrgð á öllum kóða sem skilað er til verkkaupa, hvort sem hann er skrifaður af manni eða myndaður af gervigreind. Sama gildir um réttmæti, öryggi, gæði og prófanir.

Óskipulögð notkun gervigreindar ,handahófskennd notkun á ChatGPT, Claude ,Copilot eða öðrum gervigreindartólum án skilgreindra leiðbeininga eða samhengis, telst ekki þroskaður vinnuferill og samræmist ekki væntingum verkkaupa.

11.1.2 Skipulögð notkun og leiðbeiningakerfi (.agent/ mappa)

Verkkaupi birtir skipulagðar leiðbeiningar og samhengi fyrir gervigreindarverkfæri í sérstakri .agent/ möppu sem fylgir hverri kóðageymslu sem verkkaupi stofnar. Leiðbeiningarnar byggja á opna staðlinum Agent OS v3.0 (<https://buildermethods.com/agent-os>) og innihalda afurðaskjölun, kóðastaðla og leiðbeiningar sem gervigreindarverkfæri lesa beint.

Leiðbeiningarnar byggja á:

- þessari þróunarhandbók,
- tæknistefnu verkkaupa,
- rekstrarhandbók verkkaupa,
- hönnunarhandbók verkkaupa,
- gervigreindardrífnum rýnisniðmátum (review prompts) verkkaupa.

Leiðbeiningarnar skilgreina almenn viðmið (guardrails) sem gervigreindarverkfæri eiga að fylgja þegar þau aðstoða forritara við þróun á kóða fyrir verkkaupa.

11.1.3 Kröfur til verksala

Krafa	Lýsing
Virðing fyrir leiðbeiningum	Gervigreindarverkfæri verksala skulu vera stillt þannig að þau taki mið af leiðbeiningum í .agent/ möppu
Viðhald og auðgun	Verksali skal viðhalda og bæta við í .agent/ möppuna á meðan á þróun stendur, ekki einungis við skil
Þekkingarsöfnun	Gervigreindarverkfæri skulu nýtt til að skrá hönnunarákvarðanir, útfærsluþekkingu og önnur mikilvæg atriði í .agent/ möppu jafnóðum
Hluti af rýni	Gæði og innihald .agent/ möppu er hluti af gæðamati við skil og verður metið í kóðarýni
Aukin krafa fyrir Leið B	Í Leið B verkefnum (kafli 8.2.2) gegnir .agent/ mappan auknu hlutverki sem óbeint sönnunargagn þess að AI-smíðaðferli hafi verið fylgt. Gæði, dýpt og stöðug uppfærsla .agent/ efnis er metin með auknum þunga í viðtökuferli Leið B verkefna.

Sjá viðauka „Skipulag .agent/ möppu“ fyrir skilgreiningu á möppuskipulagi, staðlasskrám og innihaldi.

11.1.4 .agent/ og Leið B framendaverkefni

Leið B framendaverkefni (kafla 8.2.2) skulu hafa .agent/ möppu sem uppfyllir sömu kröfur og aðrar lausnir, auk eftirfarandi viðbótarkrafna:

Krafa	Lýsing
Stöðug uppfærsla	.agent/ mappa skal endurspeglar þekkingu sem gervigreindarverkfæri hafa safnað á meðan á smíði stóð. Möppan skal vera uppfærð stöðugt á meðan á þróun stendur, ekki einungis í lokin.
Flækja við endursmíð	.agent/ mappa skal innihalda nægjanlegar leiðbeiningar til þess að gervigreindarverkfæri geti endursmíðað framendann án sérhæfðrar mannlegrar þekkingar. Þetta er hluti af mati á flækju við endursmíð verkkaupa (kafla 8.2.2).
Tæknistafla og takmarkanir	.agent/product/techstack.md skal tilgreina samþykktan tæknistafla Leiðar B, þar á meðal bann við ytri stöðustjórnunarsöfnum, Node.js í keyrslu og handskrifuðum kóða.
Gæðamat	Gæði og innihald .agent/ möppu í Leið B verkefnum er metið sem hluti af viðtökufæri, með auknum þunga miðað við önnur verkefni. Ónóg .agent/ mappa getur leitt til synjunar á afhendingu.

Að auki gildir vinnulagið sem skilgreint er í viðauka „Æskilegt gervigreindarþróunarferli“ sem bindandi (SKAL) fyrir Leið B verkefni. Áætlunar- og framkvæmdaáætlun skulu liggja fyrir og samþykktar af tæknilegum ábyrgðaraðila SH áður en kóðasmíði hefst.

11.1.5 Eftirlit með gervigreindarkóða

Verkkaupi fylgist með hlutfalli gervigreindarkóða í afhendingum í gegnum sjálfvirka gæðaferla og gervigreindardrífna kóðarýni (sjá kafla 6, Gæðaferlar og viðauka „Yfirferðarskjal kóðarýni“).

Bæði öfgatilkviku eru viðvörðunarkerki:

Tilvik	Viðbrögð
Enginn eða mjög lítill gervigreindarkóði	Bendir til þess að teymi nýti sér ekki þau verkfæri sem tiltæk eru. Verkkaupi mun hefja samtal við teymið um ástæður og hvort stuðnings sé þörf.
Of mikill eða óskoðaður gervigreindarkóði	Bendir til þess að mannleg rýni og yfirferð sé ónóg. Verkkaupi mun hefja samtal við teymið um gæðaferla og ábyrgð.

Markmiðið er ekki tiltekið hlutfall, heldur jafnvægi þar sem gervigreind eykur skilvirkni án þess að draga úr ábyrgð og gæðum mannlegrar yfirferðar.

11.1.6 Ábyrgð og mannleg yfirumsjón

Meginregla	Lýsing
Ábyrgð verksala	Verksali ber fulla ábyrgð á öllum kóða sem skilað er, óháð uppruna. Gervigreindarmyndaður kóði er ekki undanþegin gæðakröfum.
Mannleg rýni	Allur kóði sem myndaður er af gervigreind skal fara í gegnum mannlega rýni áður en hann er hluti af breytingabeidni
Prófanir	Gervigreindarmyndaður kóði fellur undir sömu prófunarkröfum og annar kóði (sjá kafla 13)
Sömu gæðakröfur	Engin sérstök gæðaviðmið gilda um gervigreindarkóða, hann fer í gegnum sömu gæðahlið og allur annar kóði

11.1.7 Gagnaöryggi og trúnaður

Verksali ber ábyrgð á að gervigreindarverkfæri sem notuð eru í þróunarferlinu uppfylli eftirfarandi kröfur:

Krafa	Lýsing
Engin heilbrigðisgögn í gervigreindarþjónustur	Óheimilt er að senda raunveruleg heilbrigðisgögn, persónugreinanleg gögn eða viðkvæm rekstrargögn í gervigreindarþjónustur
Trúnaður um kóðagrunn verkkaupa	Óheimilt er að hlaða upp kóða verkkaupa eða sértækri arkitektúr í opinberar gervigreindarþjónustur (t.d. vefviðmót ChatGPT eða sambærilegt) nema öruggar aðgangsflyssamningar (enterprise agreements) séu til staðar
Leyfismál	Verksali ber fulla ábyrgð á hugverkis- og leyfismálum kóða sem gervigreind myndar. Ef kóði sem myndaður er af gervigreind byggir á copyleft-leyfðum þjálfunargögnum ber verksali ábyrgð á leyfisfyllni.

11.1.8 Gagnsæi og upplýsingaskylda

Verkkaupi gerir kröfu um gagnsæi í notkun gervigreindar í þróunarferlinu.

Krafa	Lýsing
Upplýsingaskylda um verkfæri	Verksali skal upplýsa verkkaupa um hvaða gervigreindarverkfæri eru notuð í þróunarferlinu (t.d. Claude Code, GitHub Copilot, Cursor, ChatGPT, Codeium, DevinAI eða önnur)
Aðferðafræði	Verksali skal lýsa hvernig gervigreindarverkfæri eru samþætt við þróunarferil teymisins, ekki einungis nefna verkfærin
Breytingar á verkfærum	Ef verksali skiptir um gervigreindarverkfæri eða breytir aðferðafræði á meðan á verkefni stendur skal tilkynna verkkaupa skriflega
Engin sérstök vernd	Upplýsingar um notkun gervigreindar eru ekki trúnaðarmál, verkkaupi áskilur sér rétt til að spyrja um og ræða vinnuferla hvenær sem er

11.1.9 Gervigreindardrifin rýnisniðmát verkkaupa

Verkkaupi birtir þau rýnisniðmát (review prompts) sem notuð eru í gervigreindardrifinni kóðarýni í lausnarbanka verkkaupa (GitHub). Rýnisniðmát ná til allra tæknilegra þátta sem handbókin skilgreinir, þar á meðal .NET kóðagæða, öryggis, arkitektúrs, skjölunar, innviða og Kubernetes samhæfni.

Tilgangur þessa er tvíþættur:

1.

Gagnsæi, verksali veit nákvæmlega hvaða viðmið gervigreindarrýnin leggur til grundvallar.
2.

Sjálfsstjórn, verksali getur keyrt sömu rýnisniðmát sjálfur sem hluta af innri gæðaferli áður en kóði er skilað.

Rýnisniðmátin fela m.a. í sér:

Sniðmát	Umfang
.NET kóðarýni	Kóðagæði, arkitektúr (ONION, SOLID), öryggi (OWASP, CWE), prófanir, skjölun, innviðir (Docker, Kubernetes), kóðastíll og samræmi
Innviðarýni	Kubernetes stillingar, Helm gæði, ArgoCD samhæfni, öryggi innviða, samkvæmni milli umhverfa
Framendarýni (eldri lausnir og Leið B)	TypeScript/React gæði, öryggisatriði, samræmi við ESLint/tsconfig reglur, á við eldri lausnir með formlega undanþágu og Leið B verkefni (kafli 8.2.2). Fyrir Leið B verkefni nær rýnin einnig til flækju við endursmíð, AI-smíðakröfu og gæða .agent/ möppu.

Rýnin nær einnig til .agent/ möppunnar, gervigreindarrýnin notar .agent/ innihald sem samhengi við yfirferð, sem tryggir að staðlar séu ekki einungis skjalfestir heldur einnig framfylgt.

Mælt er með að verksali nýti rýnisniðmát verkkaupa til sjálfsmats á afhendingum áður en breytingabeiðnir eru sendar.

Sjá viðauka „Þjónustur og verkfæri verkkaupa“ fyrir aðgang að lausnarbanka.

11.1.10 Æskilegt gervigreindarþróunarferli

Verkkaupi mælir með skipulögðu gervigreindarþróunarferli í fimm áföngum sem byggir á þeirri grunnreglu að *áætlanagerð og framkvæmdaáætlun séu sjálfstæðir verkáfangar sem ljúka skal, og rýna skal, áður en gervigreindarstudd kóðasmíði hefst.*

#	Áfangi	Niðurstaða
1	Áætlanagerð	Númeruð útfærsluáætlunarskjöl í -docs kóðageymslu
2	Framkvæmdaáætlun	Session-skipt áætlun með afritunarhæfum kvaðningum
3	Skuldbinding og rýni	Samþykkt framkvæmdaáætlun eftir rýni tæknilegs ábyrgðaraðila SH
4	Útfærsla vinnulotu kvaðninga	Smíðuð virkni, sannreynd í lok hversrar lotu
5	Lokastaðfesting og prófanir	Tilbúin lausn fyrir formlega afhendingu

Áfangar 1 og 2 byggja á langtíma (long-horizon) ósamfelldri áætlanagerð. Til þess henta sérstök agent-verkfæri eins og **Devin.AI** sem geta unnið sjálfstætt í langan tíma og lesið heilt skjalasafn án mannlegrar íhlutunar í hverju skrefi.

Ekki má hefja keyrslu á kvaðningum fyrir vinnulotur fyrr en rýni tæknilegs ábyrgðaraðila SH er lokið og samþykki liggur fyrir.

Vinnulagið er **bindandi (SKAL)** fyrir Leið B framendaverkefni (kaflí 8.2.2) og **æskilegt (GETUR)** fyrir öll önnur verkefni þar sem gervigreindarverkfæri eru notuð sem aðalsmíðatæki.

Sjá viðauka „Æskilegt gervigreindarþróunarferli“ fyrir nánari lýsingu á hverjum áfanga, kröfur til verksala og dæmi um kvaðningar.

11.2 Gervigreindarlausnir sem hluti af afurð

Þessi hluti fjallar um gervigreindarlausnir sem verða hluti af keyrsluafurð verkkaupa, þ.e. AI agentar, MCP lausnir og skyld forritunarsöfn sem eru hluti af rekstrarumhverfi. Þessi hluti á aðeins við þegar verkefni fela í sér slíka virkni.

11.2.1 Grunnreglur

Allar gervigreindarlausnir sem þróaðar eru fyrir verkkaupa skulu teljast hluti af bakendakerfum og falla undir sömu kröfur og aðrar ASP.NET þjónustur hvað varðar arkitektúr, gæðastýringu, öryggi, útgáfufarla og rekstur.

Gervigreindarlausnir skulu ávallt vera:

- fyrirsjáanlegar í rekstri,
- rekjanlegar og endurtakanlegar,
- auðveldlega aftengjanlegar eða skipta megi út fyrir aðrar,
- tæknilega íhaldssamar og byggðar á þroskuðum lausnum.

Gervigreind er hér skilgreind sem hjálpartæki innan skýrt afmarkaðs domain flæðis, en ekki sjálfstæður ákvörðunaraðili nema það sé sérstaklega samþykkt.

11.2.2 Gildissvið

Atriði	Innifalið	Ekki innifalið / óheimilt
AI agentar	Agentar útfærðir í C# með samþykktu gervigreindarforritunarsafni Microsoft sem bakendaþjónustur	Agentar í framenda, skríptur/keyrslur utan staðlaðs rekstrarumhverfis. Python eða annað forritunarmál en C#
MCP lausnir	Skýrt skilgreind Model/Message/Control samskipti og samningar	Óformleg textasamskipti milli laga, óskilgreindar langtímastöður ("agent memory")
Samþættingar	Köll í innri kerfi í gegnum viðurkennda API og auðkenningu	Beinar gagnagrunnstengingar frá agent lagi nema sérstaklega samþykkt
Rekstur	CI/CD, logging, metrics, health endpoints, audit trail	Lausnir án rekstrarvöktunar eða rekjanleika

11.2.3 Samþykkt forritunarsafn

Gervigreindar agenta lausnir skulu byggðar á opinberu gervigreindarforritunarsafni Microsoft fyrir .NET. Verkkaupi skilgreinir hvaða útgáfa og forritunarsafn er samþykkt hverju sinni.

Forritunarsafnið skal uppfylla eftirfarandi skilyrði:

Skilyrði	Lýsing
Framleiðandi	Microsoft
Bein samþætting við .NET / ASP.NET	Forritunarsafnið skal vera innfætt (native) .NET safn, ekki hjúpun á safni í öðru forritunarmáli
Langtímastuðningur	Forritunarsafnið skal vera stutt af Microsoft til langs tíma (LTS eða sambærilegt)
Opinbert og opið	Forritunarsafnið skal vera opinberlega gefið út og aðgengilegt á NuGet
Rekstrarsamhæfni	Forritunarsafnið skal styðja enterprise-öryggi, rekjanleika og samþættingu við CI/CD ferla

Notkun annarra forritunarsafna, þriðja aðila agent-framework lausna eða sérsníððra agent-útfærslna er óheimil nema með formlegri undanþágu samkvæmt undanþáguferli verkkaupa.

Sjá viðauka „Þjónustur og verkfæri verkkaupa“ fyrir nafn og útgáfu þess forritunarsafns sem er samþykkt hverju sinni.

11.2.4 Kröfur í stuttu máli

Flokkur	Krafa
Forritunarsöfn	Samþykkt opinbert Microsoft gervigreindarforritunarsafn (sjá viðauka „Þjónustur og verkfæri“)
Högun	Sömu ASP.NET bakendahögun og aðrar þjónustur verkkaupa
Öryggi	Engin heilbrigðisgögn til LLM nema samþykkt, lágmörkun og rekjanleiki tryggður
Prófanir	Prófanlegt án raun-LLM (mocks + deterministic tests)
Rekstur	Metrics, health endpoints, structured logging, sömu kröfur og önnur bakendakerfi
Kvaðningar	Endurnýtanlegar og útgáfustýrðar
Fjöl-agent kerfi	Óheimilt nema með formlegri samþykkt verkkaupa
LLM háð	Lausn skal vera keyranleg í mock-mode án raun-LLM

11.2.5 Takmarkanir

Óheimilt er að:

- nota gervigreind beint í framenda,
- nota ósamþykkt eða heimasníðaðar agent forritunarsöfn,
- búa til fullsjálfvirka agenta án mannlegrar yfirlýsingar,
- fastsetja útfærslu við eina ákveðna gervigreindarlausn (t.d. binding við OpenAI eða Google er óheimil),
- byggja flókin multi-agent kerfi nema með samþykki verkkaupa.

Nánari kröfur um arkitektúr, lagskiptingu, MCP-högun, öryggi og prófanir eru skilgreindar í viðauka „Gervigreindarlausnir, arkitektúr og kröfur.“

12 Öryggi og auðkenningar

Þessi kafli skilgreinir grunnkröfur til öryggis, auðkenningar og aðgangsstýringar sem tengjast hugbúnaðarsmiðinni beint.

Kaflinn er ekki tæmandi lýsing á öllum öryggiskröfum. Heildarkröfur um öryggi, persónuvernd og rekstur koma fram í:

- tæknistefnu verkkaupa (kafli um öryggi, auðkenningu og aðgangsstýringu),
- rekstrarhandbók verkkaupa (kafli um aðgangsstýringu, auðkenni og leyndarmál),
- gæðahandbók verkkaupa, og
- útboðsgögnum viðkomandi verkefnis, eftir því sem við á.

Ef árekstur verður milli þessara skjala gilda strangari kröfur.

12.1 Almennar öryggiskröfur

Allar lausnir skulu vera hannaðar með öryggi að leiðarljósi frá upphafi (security by design) og uppfylla eftirfarandi lágmarkskröfur:

Krafa	Lýsing
Auðkenningarupplýsingar ekki í kóða	Lykilorð, lykilar og önnur viðkvæm gögn skulu aldrei vera geymd í kóða eða stillingarskrám
Leyndarmál afhent við keyrslu	Leyndarmál skulu vera afhent kerfi við ræingu í gegnum öruggt leyndarmálakerfi (sjá kafla <i>Geymsla leyndarmála</i>)
Dulkóðuð samskipti	Öll samskipti skulu vera dulkóðuð með viðurkenndum stöðlum (TLS, HTTPS, VPN)
Engar sérsníððar öryggislausnir	Óheimilt er að útfæra eigin dulkóðunargrím, auðkenningarkerfi eða aðrar öryggislausnir

Stöðugir öryggisþakkar	Kóði og forritunarsöfn sem sinna auðkenningu skulu vera í stöðugum (stable/release) útgáfum. Alpha eða beta útgáfur eru óheimilar.
------------------------	--

12.2 Auðkenningar

Lausnir skulu nota auðkenningar- og heimildaveitingarlausnir sem verkkaupi útvegar og hýsir, þar sem slíkt er mögulegt.

Krafa	Lýsing
Margbáttaaauðkenning (MFA)	Allar lausnir með notendaviðmóti skulu styðja MFA nema lýsing verkefnisins kveði á um annað
JWT	Þar sem við á skal notast við JWT með viðurkenndum forritunarsöfnum (sjá jwt.io/libraries)
Endurnýting tóka óheimil	Auðkenningartókar frá ytri kerfum skulu aldrei endurnotaðir óbreyttir
Umbreyting tóka	Ytri tókar skulu sannreynðir og umbreyttir í innri auðkenningartóka lausnar með aðstoð auðkenningarþjónustu verkkaupa
Ógildingarmöguleiki	Lausnir skulu bjóða upp á að ógilda auðkenningartóka í neyðartilvikum
Active Directory	Lausnir sem keyra á útstöðvum notenda skulu notast við AD heimildaveitingu

12.3 Aðgangsheimildir (Authorization)

Lausnir skulu nýta miðlæga aðgangsheimildapjónustu verkkaupa þar sem mögulegt er.

12.3.1 Meginreglur

Meginregla	Lýsing
Aðgerðamiðuð heimildaveiting	Heimildir skulu vera claims-based, ekki role-based
Stöðluð og fyrirsjáanleg	Heimildir skulu fylgja samræmdu nafnasniði
Óháð hlutverkum	Heimildir skulu ekki vera bundnar við starfsheiti eða hlutverk

12.3.2 Rökstuðningur

Í hefðbundnu hlutverkamiðuðu aðgangsstýringarkerfi (RBAC) eru heimildir bundnar við hlutverk, t.d. „Læknir“, „Hjúkrunarfræðingur“, „Stjórnandi“. Þetta virkar vel þegar hlutverk eru fá, stöðug og vel afmörkuð. Í íslenskri heilbrigðisþjónustu er raunveruleikinn annar:

Áskorun	Lýsing
Mörg hlutverk	Einstaklingur gegnir oft fleiri en einu hlutverki samtímis, sérfræðilæknir sem er einnig deildarstjóri, hjúkrunarfræðingur sem sinnir bæði legudeild og göngudeild, rannsóknaraðili sem er einnig klínískur starfsmaður. Hlutverkamiðuð kerfi þurfa að búa til samsett hlutverk (t.d. „SérfræðilæknirOgDeildarstjóriMeðRannsóknarheimild“) sem margfaldast hratt.
Hlutverka verðbólga (role explosion)	Eftir því sem kerfum fjölgar og starfssvið skarast geta hlutverkin orðið hundruð eða þúsundir. Þetta er vel þekkt vandamál í heilbrigðisgeiranum, rannsóknir lýsa þessu sem „the most costly and complex process limitation“ í RBAC-kerfum sjúkrahúsa.
Skipulagsbreytingar	Hlutverk eru bundin við skipurit stofnana. Þegar deildir sameinast, skipta sér eða breyta nafni þarf að endurskipuleggja hlutverkakerfi kerfisins. Aðgerðir (t.d. „PatientSummary-Read“) eru óháðar skipuriti og breytast ekki þegar stofnun endurskipuleggur sig.
Harðkóðun í kóða	Hlutverkamiðuð kerfi leiða oft til þess að hlutverkanöfn eru harðkóðuð í forritunarkóða (t.d. <code>if (user.Role == "Doctor")</code>). Þetta gerir breytingar á heimildaskipulagi háðar kóðabreytingum, sem krefst nýrrar útgáfu, prófana og samþykkis, í stað þess að vera stjórnunarleg ákvörðun.
Langtíma viðhaldshæfni	Lausnir verkkaupa eru ætlaðar til reksturs í 20–30 ár. Skipurit heilbrigðisstofnana mun breytast margsinnis á þeim tíma. Heimildakerfi sem er byggt á aðgerðum (claims) þolir slíkar breytingar án kóðabreytinga; heimildakerfi byggt á hlutverkum þolir þær ekki.
Fjöl-birgjasamræmi	Í útboðslíkani verkkaupa vinna margir verksalar að mismunandi kerfum. Ef verksalar útfæra eigin hlutverkakerfi verða heimildamódel mismunandi milli

	kerfa. Samræmt nafnasnið heimilda (Scope-Operation) sem skilgreint er hér að neðan tryggir samræmi óháð verksala.
--	---

Aðgerðamiðuð heimildaveiting (claims-based) leysir þessi vandamál á einfaldan hátt: heimildir eru bundnar við aðgerðir (t.d. „MedicationDispense-Write“) en ekki við hlutverk. Hver einstaklingur fær þær heimildir sem hann þarf, óháð starfsheiti, deild eða skipuriti. Heimildir eru stjórnað í miðlægri aðgangshimildaþjónustu verkkaupa og breytast án kóðabreytinga.

Þetta þýðir ekki að hugtakið „hlutverk“ hverfi úr stjórnun, verkkaupi getur skilgreint hópa (t.d. „hjúkrunarfræðingar á bráðamóttöku“) sem fá sjálfkrafa ákveðnar heimildir. En heimildarákvarðanir í kóða skulu ávallt miðast við aðgerðir (claims), aldrei við hlutverkanöfn.

12.3.3 Nafnasnið heimilda

Heimildir skulu fylgja eftirfarandi mynstri:

<Scope>-<Operation>

Hluti	Lýsing	Leyfileg gildi
Scope	Virknissvið eða gagnasvið innan lausnar	T.d. PatientSummary, MedicationDispense, TreatmentRelations
Operation	Aðgerð	Read eða Write

Dæmi:

Heimild	Lýsing
PatientSummary-Read	Lesaðgangur að sjúklingayfirliti
MedicationDispense-Read	Lesaðgangur að lyfjagjöf
MedicationDispense-Write	Skrifaðgangur að lyfjagjöf
TreatmentRelations-Read	Lesaðgangur að meðferðartengslum

Ef frekara niðurbrots er þörf skal nota forskeyti á forminu <Scope>-<SubScope>-<Operation>:

Heimild	Lýsing
TreatmentRelations-Add-Write	Heimild til að bæta við meðferðartengslum
TreatmentRelations-Remove-Write	Heimild til að fjarlægja meðferðartengsl

Forðast skal að hafa annað en Read og Write sem aðgerðir nema brýn nauðsyn krefji.

12.3.4 Óheimil högun

Óheimilt	Ástæða
Role-based heimildir (t.d. Administrator, Doctor, Staff)	Bindir heimildir við hlutverk frekar en aðgerðir; skalast ekki vel
Harðkóðuð hlutverk í kóða	Torveldar breytingar á heimildaskipulagi
Heimildaákvæðanir í framenda	Öryggisstýring skal vera á netþjóni

12.4 Öryggiskröfur framendar

Krafa	Lýsing
Rafræn skilríki	Skulu notuð til auðkenningar þar sem mögulegt er
MFA/2FA	Ef rafræn skilríki eru ekki möguleg skal nota viðurkenndar MFA lausnir
Httponly session cookie	Auðkenningarupplýsingar (JWT eða session-id) skulu geymdar í httponly session cookie
JWT skipting í kökur	Heimilt er að skipta JWT í tvær kökur (payload og signature); bakendi setur saman
Ekkert í localStorage	Óheimilt er að geyma auðkenningarupplýsingar eða viðkvæm heilbrigðisgögn í localStorage
JavaScript aðgangur takmarkaður	Takmarka skal aðgang JavaScript kóða að auðkenningarupplýsingum
Subresource Integrity (SRI)	Ef kóði er sóttur af öðrum lénnum skal nota SRI integrity attribute til að tryggja heilindi

Athugasemd um SRI: Almenna reglan er að allur JavaScript kóði skuli vera hluti af lausninni og geymdur í kóðageymslum verkkaupa (sjá kafla *Ytri forritunarsöfn*). SRI á því fyrst og fremst við þegar ytri auðlindir eru notaðar með formlegu samþykki.

12.5 Öryggiskröfur bakendaþjónustur

Krafa	Lýsing
API-lyklar ef ekki JWT	Ef JWT er ekki notað skal nota API-lykla eða sambærilegt fyrir sambættingu milli kerfa
IP-læsing á API-lykla	API-lyklar skulu bjóða upp á læsingu á ákveðnar IP-númera raðir; annars VPN
Throttling	Útfæra skal throttling á ógildum auðkenningarfyrirspurnum per auðkenningu
Engin geymsla ógildra fyrirspurna	Til að hindra DDOS skal ekki vista ógildar auðkenningarfyrirspurnir í gagnagrunni
Stigvaxandi töf	Beita stigvaxandi töf við endurtekna ógildar auðkenningu (0s → 2s → 4s → 8s → 16s...)
Engar varanlegar læsingar	Forðast varanlegar læsingar notenda vegna rangra auðkenninga
Takmarkanir á endurteknum aðgerðum	Setja takmarkanir á fjölda sambærilegra aðgerða per tímaeiningu per endapunktur
Stöðluð forritunarsöfn	Notast skal við innbyggða .NET virkni eða stöðluð forritunarsöfn við útfærslu öryggistakmarkana

12.6 Öryggiskröfur ytri forritunarsöfn

Krafa	Lýsing
Allur kóði í kóðageymslum	Allur kóði og öll forritunarsöfn sem eru hluti af keyrsluvöru skulu fylgja lausninni í kóðageymslum verkkaupa
Sjálfbærni	Lausnir skulu vera byggðar og keyrðar án háðs við ytri netþjónustur sem ekki eru hluti af samþykktu rekstrarhögun
Bann við sjálfvirkum endapunktum	Ytri forritunarsöfn sem bæta sjálfvirk við endapunktum í lausn eru óheimil (t.d. Hangfire dashboard, Elmah endpoints)

Ástæða banns við sjálfvirkum endapunktum er að verkkaupi þarf fulla stjórn á öllum endapunktum sem lausn birtir, bæði af öryggis- og rekstrarsjónarmiðum.

12.7 Öryggiskröfur gagnagrunnar

Krafa	Lýsing
Dulkóðuð geymsla	Auðkenningarupplýsingar (password hashes, API keys) skulu geymdar dulkóðaðar í aðgreindu töfluskema
Disk dulkóðun	Allir gagnagrunnar skulu nota innbyggða dulkóðun á disk

Sjá kafla „Geymsla og miðlun gagna“ og „Dulkóðun gagna“ fyrir frekari kröfur um gagnamedhöndlun og dulkóðun.

13 Hugbúnaðarprófanir

🧐 Leiðarljós

Gæði og hagkvæmni hugbúnaðarprófana eru jafnmikilvæg og lausnarkóðinn sjálfur. Án markvissra og vel skilgreindra prófana er ekki unnt að tryggja öryggi, stöðugleika og fyrirsjáanleika breytinga yfir líftíma lausna sem spannar mörg ár.

Allur hugbúnaðarkóði sem þróaður er fyrir verkkaupa skal vera studdur sjálfvirkum prófunum sem tryggja réttleika, stöðugleika og viðhaldshæfni.

Prófanir eru óaðskiljanlegur hluti af afhendingu hugbúnaðarafurða og forsenda samþykktar skila og útgáfu.

13.1 Ábyrgð og umhverfi

Ábyrgðarsvið	Ábyrgð
Skilgreining, framkvæmd og viðhald prófana	Verksali
Umsýslukerfi fyrir prófanir (aðgangsstýrt, rekjanlegt)	Verksali
Skráning og eftirfylgni villna	Verksali

Aðgangur að prófunargögnum eftir þörfum	Verkkaupi
Gæðahlið og sjálfvirk þekjumat við skil	Verkkaupi

13.1.1 Staðsetning prófana

Prófanir sem meðhöndla raunveruleg eða afleidd Íslensk heilbrigðisgögn skulu fara fram innan Evrópska efnahagssvæðisins (EEA) í samræmi við GDPR.

13.1.2 Samhengi kröfu og prófana

Samhengi milli kröfu og prófana skal vera skýrt og rekjanlegt. Hver skilgreind krafa skal eiga að lágmarki eitt skráð próf.

13.2 Sjálfvirkar prófanir

Sjálfvirkar prófanir skulu vera hluti af lausnarkóða og keyrðar reglulega í samþykktum CI-ferlum. Prófanir skulu vera skrifaðar með það að markmiði að styðja framtíðaruppfærslur og vera viðhaldshæfar til langs tíma.

Prófanakóði skal vera skrifaður í viðurkenndum prófunarrömmum sem eru rótgrónir í samþykktu tæknihverfi. Óheimilt er að nota heimasmiðuð eða óreynd prófunarframework.

13.2.1 Einingaprófanir (Unit tests)

Einingaprófanir eru grunnforsenda gæðatryggingar og forsenda samþykktar skila.

Krafa	Gildi
Þekja á nýjum kóða	≥ 80%
Heildarþekja lausnar	≥ 65%; má ekki lækka milli útgáfa
Meginlínur þjónustufalla	Prófaðar með a.m.k. einu prófi fyrir eðlilega virkni og einu fyrir villumeðhöndlun
Ef kröfur ekki uppfylltar	PR sjálfkrafa hafnað

13.2.2 Rökstuðningur þekjumarka einingaprófana

Þekjumörk verkkaupa (≥80% á nýjum kóða, ≥65% heildarþekja) eru ekki handahófskennd viðmið heldur byggja á jafnvægi milli gæðaávinninga og kostnaðar, aðlagð að raunveruleika verkkaupa.

Viðmið	Rökstuðningur
≥80% á nýjum kóða	Við þetta þekjustig eru meginlínur viðskiptarökfræði, villumeðhöndlun og jaðartilvik prófaðar á áreiðanlegan hátt. Prófanir ná yfir þann kóða sem skiptir máli fyrir réttleika og stöðugleika. Hærra þekjustig (t.d. ≥90%) skilar minnkandi ávinningi, það krefst prófana á léttvægum kóða (DI-skráning, stillinga-klösum, einföldum gagnategundum) sem veita litla villuvörn en auka viðhaldskostnað prófana.
≥65% heildarþekja	Lausnir verkkaupa innihalda í sumum tilvikum eldri kóða sem var yfirtekinn með lægri þekju. 65% viðurkennir þá stöðu en setur gólf sem tryggir að þekja minnki aldrei milli útgáfa. Saman með ≥80% kröfunni á nýjum kóða þýðir þetta að heildarþekja hækkar stöðugt eftir því sem nýr kóði bætist við.
Sjálfvirk framfylgd	Þekjumörkum er framfylgt af SonarCloud gæðahliðum sem eru í höndum verkkaupa (kaflí 6). Breytingabeidni sem uppfyllir ekki þekjumörk er sjálfkrafa hafnað, engin handvirk túlkun eða samningaviðræður eiga sér stað. Þetta tryggir hlutlægt og fyrirsjáanlegt mat sem gildir jafnt fyrir alla verksala.
Undanskilningarferli	Kóði sem er rökstutt undanskilinn frá einingaprófunum (sjá hér að neðan) telst ekki með í þekjumati. Þetta tryggir að þekjumörk mæli raunverulega prófaðan kóða, ekki tilbúna tölfærð. Ekki er heimilt að undanskilja kóða til að ná þekjumörkum slíkt er brot á gæðaviðmiðum og verður meðhöndlað í kóðarýni.

Heimilt er að undanskilja afmarkaðan kóða frá einingaprófunum, enda liggi fyrir rökstutt og samþykkt mat tæknilegra sérfræðinga verkkaupa. Þegar kóði er undanskilinn skal rökstuðningur fylgja, t.d.:

[ExcludeFromCodeCoverage(Justification="Class inherited from external system, code not modified to maintain parity with original implementation")]

13.2.3 Samþættingarprófanir (Integration tests)

Öll ytri forritunarskil (API) skulu vera studd samþættingarprófunum.

Krafa	Lýsing
Þekja	Allir aðgengilegir endapunktur skulu hafa a.m.k. eitt próf fyrir eðlilega virkni og eitt fyrir villumeðhöndlun
Sjálfvirk	Prófanir skulu vera keyranlegar sjálfvirk í CI-ferlum
Óháðar ytri kerfum	Prófanir skulu ekki krefjast handvirkra uppsetningar gagnagrunna eða gagna
Samhliða keyrsla	Prófanir skulu vera keyranlegar samhliða án hliðarverkana
Óháðar auðkenningu	Hægt skal vera að prófa þjónustur óháð raunverulegum ytri auðkenningarþjónustum

13.2.4 Sjálfvirkar viðmótsprófanir (UI tests)

Viðmótsprófanir skulu vera í lágmarki til að draga úr langtíma viðhaldskostnaði.

Krafa	Lýsing
Umfang	Halda í lágmarki; einblína á lykilvirkni
Gögn	Skulu keyra á fölsuðum (mock) bakendum og gögnum
Flækjustig	Í algjöru lágmarki
Verkfæri	Cypress eða Playwright nema eðli verkefnis krefjist annars

13.2.5 Aðgengisprófanir

Framendalausnir skulu fara í gegnum sjálfvirkar aðgengisprófanir til að tryggja að viðmót uppfylli lágmarkskröfur.

Sjálfvirkar prófanir koma ekki í stað handvirkra prófana en eru viðbót til stöðugs eftirlits.

Sjá kafla „Aðgengismál“ fyrir fulla lýsingu á WCAG og WAI-ARIA kröfum.

13.3 Handvirkar prófanir

Þar sem sjálfvirkar prófanir ná ekki yfir alla þætti skal framkvæma handvirkar prófanir.

Krafa	Lýsing
Skjölun	Handvirkar prófanir skulu vera skjalfestar
Viðhald	Uppfærðar samhliða breytingum á lausn
Kröfupróf	Prófanir sem tengjast skilgreindum kröfum má ekki eyða; heimilt er að merkja þær óvirkar með samþykki verkkaupa
Viðmótsprófanir	Skulu uppfylla kröfur notendaviðmótshönnuðar
Aðgengi	Viðmót skulu einnig prófuð með skjálesara (sjá kafla Aðgengismál)
Slembíprófanir (exploratory)	Ef notaðar skulu þær skjalfestar í prófunarskýrslu

13.4 Samþykktarprófanir

Samþykktarprófanir skulu fara fram fyrir afhendingu hugbúnaðarafurða nema annað hafi verið sérstaklega samþykkt.

Krafa	Lýsing
Fyrirvari	Skipulagðar með að lágmarki 2 vikna fyrirvara
Þátttaka verkkaupa	Starfsmaður verkkaupa viðstaddur
Þátttaka verksala	Verkefnastjóri verksala viðstaddur; tækniaðili til taks ef óskað er
Niðurstöður	Prófunarskýrsla skal berast í beinu framhaldi, eða í síðasta lagi 2 sólahringum fyrir áætlaða útgáfu

13.4.1 Prófunarskýrsla

Prófunarskýrsla samþykktarprófana skal innihalda:

- niðurstöður allra prófana með stöðumerkingu (staðist / fallið),

- skjámyndir og staðfestingu á keyrslu,
- lista yfir óleystar athugasemdir eða villur, ef einhverjar.

Prófunarskýrsla skal undirrituð af ábyrgum prófara og gæðastjóra eða verkefnastjóra verksala.

13.5 Gögn sem fylgja afhendingu

Eftirfarandi prófunargögn skulu fylgja hverri afhendingu hugbúnaðar:

Tegund	Lýsing
Prófunarsvitur	Öll skilgreind próf (test suites) ásamt keyrsluskráum og tímasetningum
Sjálfvirkar prófanir	Allar sjálfvirkar prófanir sem hluti af kóðanum
Prófunargögn	Gögn sem nauðsynleg eru til keyrslu sjálfvirkra prófana
Leiðbeiningar fyrir prófara	Skýringarmyndir og leiðbeiningar, ef við á
Prófunarskýrslur	Niðurstöður prófana, undirritaðar af ábyrgum aðilum

Sjá einnig kafla „Skil á hugbúnaðarafurðum“ fyrir heildarlista yfir þá þætti sem skulu fylgja hverri afhendingu.

14 Aðgengismál

Leiðarljós

Eitt af meginmarkmiðum verkkaupa er að tryggja gott og jafnt aðgengi allra að þjónustu og upplýsingum í hugbúnaðarkerfum sem hann þróar, rekur og viðheldur. Hugbúnaður skal taka mið af þörfum fatlaðs fólks og einstaklinga sem lifa með algengar skerðingar, svo sem sjónskerðingu, heyrnarskerðingu, hreyfihömlun eða litblindu.

Aðgengi er óaðskiljanlegur hluti af gæðum hugbúnaðar og forsenda samþykktar lausna til reksturs.

Sjá einnig Hönnunarhandbók verkkaupa, sérstaklega kafla um hönnun fyrir aðgengi (inclusive design) og almenna viðmótshönnunarstaðla.

14.1 WCAG staðallinn

Allar notendamiðaðar lausnir skulu uppfylla kröfur nýjustu útgáfu WCAG staðalsins sem er í gildi á útgáfudegi hugbúnaðar.

14.1.1 Lágmarkskröfur, allar lausnir

Eftirfarandi kröfur gilda um **allar** lausnir, bæði þær sem eru ætlaðar almenningi og þær sem eru ætlaðar heilbrigðisstarfsfólki:

Kröfustig	Krafa
WCAG A	Öll skilyrði skulu uppfyllt
WCAG AA	Öll skilyrði skulu uppfyllt

14.1.2 Viðbótarkröfur, valin AAA skilyrði

Eftirfarandi valin AAA skilyrði skulu uppfyllt í öllum lausnum:

WCAG skilyrði	Lýsing
1.3.6	Identify Purpose, tilgangur inntakssvæða auðþekkjanlegur
1.4.7	Low or No Background Audio, lágmarks bakgrunnshljóð
1.4.8	Visual Presentation, sjónræn framsetning texta
1.4.9	Images of Text (No Exception), enginn texti í myndum
2.1.3	Keyboard (No Exception), öll virkni aðgengileg með lyklaborði
2.2.3	No Timing, engin tímamörk á aðgerðir
2.2.5	Re-authenticating, gögn varðveitt við endurauðkenningu
2.2.6	Timeouts, notendur varaðir við tímamörkum
2.3.2	Three Flashes, ekkert blikkar meira en þrisvar á sekúndu
2.3.3	Animation from Interactions, hægt að slökkva á hreyfimyndum
2.4.8	Location, staðsetning notanda í kerfi sýnileg

2.4.10	Section Headings, innihald skipulagt með fyrirsögnum
2.4.12	Focus Not Obscured (Enhanced), fókus ekki hulinn
3.3.6	Error Prevention (All), villuvarnir á öllum innsláttaraðgerðum
3.3.9	Accessible Authentication (Enhanced), aðgengileg auðkenning

Frávik frá þessum kröfum eru einungis heimil með formlegu, rökstuddu samþykki verkkaupa.

14.2 WAI-ARIA staðallinn

14.2.1 Lausnir ætlaðar almenningi

Öll notendaviðmót ætluð almenningi skulu vera hönnuð og útfærð með viðeigandi WAI-ARIA merkingum, þannig að staðlaðar hjálpartækjalausnir geti túlkað og miðlað innihaldi og virkni viðmóta á réttan hátt.

ARIA merkingar skulu vera:

- rétt notaðar og í samræmi við HTML uppbyggingu,
- ekki notaðar í stað rétttrar HTML eiginda notkunar þegar slík er til staðar.

14.2.2 Lausnir ætlaðar heilbrigðisstarfsfólki

Lausnir sem eru eingöngu ætlaðar heilbrigðisstarfsfólki skulu leitast við að uppfylla WAI-ARIA kröfur eftir því sem kostur er. Verkkaupi metur í hverju verkefni hvort full WAI-ARIA krafa sé gerð, með hliðsjón af notendahópi og eðli lausnar.

Óháð WAI-ARIA kröfu skulu allar lausnir uppfylla WCAG A og AA skilyrði að fullu, þar sem WCAG skilyrðin tryggja grunnframkvæmd aðgengis óháð ARIA.

14.3 Prófanir á aðgengi

Aðgengi skal sannreynt með reglulegum prófunum fyrir hverja útgáfu hugbúnaðar.

Tegund prófunar	Krafa
Sjálfvirkar aðgengisprófanir	Skulu notaðar þar sem mögulegt er sem hluti af CI-ferlum
Skjálesari	Lausnir skulu prófaðar með skjálesara fyrir hverja útgáfu
Íslenskur raddgervill	Notaður þar sem við á
Ráðgjöf frá hagsmunaaðilum	Við val á hjálpartækjum skal taka mið af ráðleggingum Blindrafélagsins eða sambærilegra aðila

Sjálfvirkar aðgengisprófanir koma ekki í stað handvirkra prófana, heldur eru þær viðbót til stöðugs eftirlits yfir líftíma lausna.

15 Hugbúnaðarskjölun

 **Leiðarljós**

Hugbúnaður er aldrei betri en skjölunin sem fylgir honum. Markmiðið er ekki mikill fjöldi blaðsíðna eða umritanir á almennri vitneskju, heldur kjarnyrt skráning sem lýsir högun lausnar, tengslum hennar við umheiminn og forsendum þeirra ákvarðana sem teknar voru.

Skjölun er óaðskiljanlegur hluti af afhendingu hugbúnaðarafurða og fellur undir sömu gæðakröfur og forritunarkóði. Skortur á fullnægjandi skjölun telst galli á afhendingu.

15.1 Almennar kröfur

Krafa	Lýsing
Uppfærð	Skjölun skal vera í samræmi við þá útgáfu hugbúnaðar sem er afhent
Uppfærð við hverja afhendingu	Ef breytingabeidni breytir virkni eða arkitektúr skal skjölun uppfærð samhliða. Skjölun sem er ekki uppfærð telst galli í afhendingu.
Raunveruleg	Skal endurspeгла raunverulega virkni, arkitektúr og frávik frá stöðlum
Nægjanleg	Reyndur forritari getur hafið vinnu án óþarfa tafar

Skiljanleg	Verkefnastjórar og rekstrarteymi skilja lausnina án þess að lesa kóðann
Kjarnyrnt	Einblínir á högun, ákvarðanir og forsendur, ekki almennt efni

15.1.1 Snið og geymsla

Atriði	Krafa
Afhendingsskjöl (arkitektúr, rekstrarhandbók o.þ.h.)	PDF eða Word; útgáfunúmer samræmt lausninni
Tæknileg skjölun ætluð forriturum	Markdown í kóðageymslum, undir /Docs möppu
Yfirlitsskjal	Docs/INDEX.md skal vera til staðar og vísa í öll undirskjöl
Myndir	Geymdar í Docs/img/ á PNG eða JPG sniði
Upprunaskjöl	Geymdar með lausninni í /Docs möppu
README tengill	README.md í rót skal innihalda tengil í Docs/INDEX.md

Sjá viðauka „Skipulag og snið hugbúnaðarskjölunar“ fyrir fulla lýsingu á möppuskipulagi, README sniði, og gæðamatsviðmiðum sem notuð eru við kóðarýni.

15.2 Skjölun sem skilað er við afhendingu

Eftirfarandi skjölun skal liggja fyrir við afhendingu hugbúnaðar og vera samræmd útgáfunúmerum lausnarinnar.

15.2.1 Arkitektúr og kerfishönnun

Þáttur	Lýsing
Yfirlitsmynd	High-level architecture, heildaryfirlit af kerfinu
Ytri tengingar	Myndir sem sýna ytri tengingar, samskiptastaðla og samskiptaleiðir
Tæknistakkur	Mynd sem lýsir tæknistakk lausnarinnar
Innviðir	Myndir sem sýna helstu innviði og tengsl þeirra
Gagnaflæði	Mynd sem sýnir gagnaflæði í gegnum lausnina
Úttektaráætlun (exit plan)	Lýsing á skrefum og forsendum til yfirtöku eða flutnings lausnar til annars aðila. Skal innihalda: gagnaútflutningsferli, þekkingaryfirfærslu, rekstrarsamfellu og áætlaðan tíma.

15.2.2 Hönnunarákvarðanir (Architecture Decision Records, ADR)

Allar veigamiklar tæknilegar og arkitektúrlegar ákvarðanir skulu vera skjalfestar sem ADR.

ADR skal vera stutt og hnitmiðuð, ein málsgrein getur dugað, og svara eftirfarandi:

Lykilorð	Spurning
„Í tengslum við“	Hvaða virkni, kerfishluti eða högun varðar ákvörðunin?
„standandi frammi fyrir“	Hvaða eiginleiki eða upplifun var æskileg?
„ákváðum við“	Hvað var ákveðið?
„hafandi skoðað“	Hvaða aðrir valkostir voru skoðaðir og hvers vegna hentuðu þeir ekki?
„til að ná fram“	Hvaða ávinning skilar valið?
„vitandi það að“	Hvaða veikleikar eða takmarkanir eru þekktar?

15.2.3 Flæðislýsingar (Sequence diagrams)

Sequence-rit skulu fylgja öllum kjarnaflæðum lausnar, sérstaklega:

- samskiptum við ytri þjónustur,
- flóknum domain flæðum,
- öryggis- og auðkenningarflæðum.

15.2.4 Rekstrarskjölun

Rekstrarskjölun skal miða við daglegan rekstur og innihalda að lágmarki:

Þáttur	Lýsing
Keyrsluumhverfi	Kröfur til keyrsluumhverfis og innviða
Stillingar	Kubernetes stillingar og tengdar kröfur
Útgáfufurli	Hvernig lausn er gefin út

Logging og metrics	Hvar og hvernig loggar og mælingar eru aðgengilegar
Viðhaldsverk	Regluleg viðhaldsverk sem þarf að keyra (t.d. gagnahreinsun)

Sjá einnig Rekstrarhandbók verkkaupa fyrir kröfur til rekstrarskjölunar frá sjónarhóli rekstraraðila.

15.3 Skjölun sem verður að vera í kóðageymslum

Eftirfarandi skjölun skal lifa samhliða forritunarkóða í kóðageymslum verkkaupa. Frjálst er að sameina eða skipta skjölum eftir þörfum, en allt efnið skal vera til staðar.

15.3.1 README.md í rót kóðageymslu

Eitt README.md skjal skal vera í rót hversrar kóðageymslu og innihalda að lágmarki:

Þáttur	Lýsing
Lýsing á lausn	Hvað lausnin gerir, tilgangur hennar og hlutverk í stærra samhengi
Tæknilýsing	Forritunarmál, framework útgáfur og helstu tæknival
Yfirlitsmynd	Hvernig helstu hlutar lausnarinnar vinna saman
Ytri tengsl	Hvaða ytri tengingar, gagnagrunnar eða þjónustur lausnin stólar á; upplýsingar um secrets og VPN ef við á
Frávik	Öll helstu frávik frá þróunarhandbók og viðurkenndum best practices, skjalfest og rökstudd
Uppsetning fyrir forritara	Hvernig nýr forritari kemur lausninni upp á sinni vél og keyrir hana til að debugga.
Gagnagrunnur	Uppsetning á gagnagrunni, vísun í SQL scriptur, helstu notendur og schema, scheduled jobs og viðhaldsferlar

15.3.2 Uppsetningarleiðbeiningar (í README eða sérstakri skrá)

Uppsetningarleiðbeiningar skulu vera nægilega ítarlegar til að forritari sem hefur aldrei séð lausnina geti komið henni í gang án utanaðkomandi aðstoðar.

Leiðbeiningar skulu ná til:

- þróunarumhverfis og kröfur til þess,
- ytri þjónustu sem þarf að herma (mock) eða tengjast,
- stillingaskráa, env breyta, secrets og aðgangs,
- hvernig á að nálgast logga í þróunarumhverfi.

15.4 Viðkvæm gögn

Öllum viðkvæmum gögnum sem stofnuð eru í framvindu verkefnis skal miðlað á öruggan hátt til ábyrgðaraðila verkkaupa sem hluta af skilum. Viðkvæmum gögnum skal **aldrei** miðlað í skjölun eða kóða.

Dæmi um viðkvæm gögn:

Tegund	Dæmi
Dulritunarlyklar	TLS skilríki, signing keys
Öryggisskilríki	Certificates, keypairs
Aðgangsstillingar	Notendanöfn, lykilorð, client_id, API-lyklar
Netstillingar	IP whitelisting, VPN uppsetningar

15.5 Ábyrgð

Aðili	Ábyrgð
Verksali	Öll skjölun til staðar, rétt og uppfærð. Endurspeglar raunverulega útfærslu. Frávik skjalfest og rökstudd.
Verkkaupi	Skilgreinir lágmarkskröfur til skjölunar. Ákvarðar hvað er forsenda samþykktar og reksturs.

15.6 Skjölun í forritunarkóða

Skjölun í sjálfum forritunarkóðanum (athugasemdir, XML summary o.þ.h.) er skilgreind í kafla 10 Forritunarkóði og vinnureglur. Kröfur um inline skjölun eru hluti af gæðamati við skil og metnar bæði sjálfvirkt og handvirkt.

Sjá viðauka „Skipulag og snið hugbúnaðarskjölunar“ fyrir nákvæmar kröfur um XML summary, gæðamat á skjölun og dæmi.

16 Geymsla og miðlun gagna

🧑🏻 Leiðarljós

Öll gögn sem unnið er með í hugbúnaðarlausnum verkkaupa eru í eðli sínu mjög viðkvæm, bæði persónulega og þjóðfélagslega. Áhersla skal vera á að tryggja gæði gagna og kóða, sem og öryggi gagnanna sjálfra, til að auðvelda langtíma viðhald og úrvinnslu. Einfaldleiki er oftast besta leiðin til að tryggja langtíma gæði.

Sjá einnig tæknistefnu verkkaupa (kaflí um gagnageymslu, gagnalíkan og stöðlun) og rekstrarhandbók verkkaupa (kaflí um gagnagrunna og afritun) fyrir kröfur sem falla utan þróunarhandbókar.

16.1 Meðhöndlun heilbrigðisgagna

Öll heilbrigðisgögn teljast mjög viðkvæm persónugögn og skulu meðhöndluð í samræmi við gildandi lög, reglugerðir og öryggisstefnur.

Meðhöndlun skal ávallt taka mið af lágmörkun gagna, aðgangsstýringu, rekjanleika og vernd gegn óheimilli dreifingu.

16.2 Ábyrgð verksala

Verksali ber fulla ábyrgð á að:

- starfsfólk hans þekki og fylgi reglum um meðhöndlun heilbrigðisgagna,
- viðeigandi verklag, fræðsla og innra eftirlit sé til staðar,
- öll frávik eða atvik séu tilkynnt tafarlaust til verkkaupa.

16.3 Miðlun og deiling gagna

Heilbrigðisgögnum má eingöngu miðla í gegnum viðurkenndar, öruggar þjónustur sem verkkaupi samþykkir.

Aðgerð	Staða
Deiling í gegnum öruggar þjónustur verkkaupa	Heimilt
Deiling eða birting í fjarfundum, myndsímtölum eða samskiptatólum	Óheimilt
Sending í tölvupósti eða óvottuðum samskiptaleiðum	Óheimilt
Geymsla á útstöðvum forritara eða fartölvum án dulkóðaðs harðadisks	Óheimilt
Vistun í kóðageymslum	Óheimilt
Notkun raunverulegra gagna í þróunar- eða prófunarumhverfi	Óheimilt nema skriflegt leyfi liggi fyrir

Þar sem prófunargögn eru nauðsynleg skulu þau vera nafnlaus eða dulrituð, ekki rekjanleg til einstaklinga, og útbúin í samræmi við samþykki verkkaupa.

16.4 Dulkóðun gagna

Dulkóðun er skyldubundin þar sem við á og skal ávallt byggja á viðurkenndum, staðlaðum aðferðum. Sérsmíðaðar eða óstaðlaðar dulkóðunarlausnir eru óheimilar.

Tilvik	Krafa	Dæmi um staðla
Í flutningi	Öll samskipti milli kerfa, þjónusta og viðmóta dulkóðuð	TLS, HTTPS, VPN
Í geymslu	Dulkóðun á gagnagrunns- eða stýrikerfis stigi	Database-level encryption, disk encryption

Í úrvinnslu	þar sem vinna þarf með viðkvæm gögn án fullrar afkóðunar	Dulkóðunarlyklar verkkaupa
-------------	--	----------------------------

Dulkóðunarstaðlar:

- Að lágmarki AES-256 eða sambærilegur styrkur þar sem samhverf dulkóðun á við.
- Almennt skal ekki dulkóða einstök dálkagildi nema persónuverndarlög geri það nauðsynlegt.

Hökkun og auðkenningargögn:

- Gögn sem einungis eru notuð til samanburðar (t.d. lykilorð) skulu geymd sem hash + salt með viðurkenndum algrímum.
- Slík gögn skulu aldrei vera geymd í lesanlegu formi.

17 Gagnagrunnar

17.1 Hlutverk gagnagrunna

Hlutverk gagnagrunna er eingöngu að geyma og miðla gögnum á öruggan, skilgreindan og fyrirsjáanlegan hátt. Gagnagrunnar skulu ekki innihalda viðskiptareglur, domain þekkingu, flókið forritunarflæði eða hegðun sem hefur áhrif á virkni kerfa.

Gagnaúrvinnsla, reiknirit, ákvörðunartaka og flæðistýring skulu vera útfærð í forritunarkóða bakendaþjónusta.

Rökstuðningur

Stored procedures, triggers og önnur rökfræði í gagnagrunni er viðurkennt mynstur í mörgum fyrirtækjaumhverfum. Í útboðslíkani verkkaupa, þar sem lausnir eru viðhaldnar af mismunandi verksölum í 20–30 ár, skapar þetta mynstur vandamál sem vega þyngra en kostirnir:

Áskorun	Lýsing
Ósýnileg gæðaferlum	Rökfræði í stored procedures og triggers fer ekki í gegnum sjálfvirka gæðaferla verkkaupa (kaflí 6). Hún birtist ekki í breytingabeidnum, fer ekki í gegnum kóðarýni, er ekki mæld af gæðahliðum og fellur utan gervigreindarrýni. Þetta skapar annan kóðagrunn sem fylgir engum af þeim vinnureglum sem handbókin skilgreinir.
Ekki prófanleg með einingaprófunum	Rökfræði í gagnagrunni er óþægilega erfið eða ómöguleg að prófa með sjálfvirkum einingaprófunum. Hún uppfyllir ekki þekjkröfur (kaflí 13.2) og torveldar clean-as-you-code aðferðarfræðina sem gæðahlið verkkaupa byggja á.
Falin hegðun og villuleit	Triggers, procedures og flókin views búa til falda hliðarverkanir sem eru ekki sýnilegar í forritunarkóða. Þetta gerir villuleit erfiða, eykur tíma í úrvinnslu athugasemda úr kóðarýni og skapar áhættu í rekstrarumhverfi. Forritari sem les bakendakóða sér ekki alla hegðun kerfisins nema hann lesi einnig gagnagrunnskóðann, sem brýtur gegn meginreglunni um læsilegan og fyrirsjáanlegan kóða (kaflí 9.1).
Háð sérþekkingu gagnagrunns	Rökfræði í stored procedures krefst sérhæfðrar SQL/T-SQL þekkingar sem er aðskilin frá .NET/C# þekkingu þróunarteyma. Þetta skapar flöskuháls þar sem fáir sérfræðingar geta breytt mikilvægri virkni, og eykur áhættu við yfirtöku þegar skipt er um verksala (sjá kafla 7.2 um úttektaráætlun).
Gagnagrunnslás (vendor lock-in)	Rökfræði í gagnagrunni er bundin við viðkomandi gagnagrunnstækni. Ef verkkaupi þarf á einhverjum tímapunkti að skipta um gagnagrunnskerfi eða samhæfa við FHIR-innviði þarf að endurskrifa alla rökfræði. Rökfræði í C# er óháð gagnagrunnum og flyst á milli kerfa án endurskriftar.
Útgáfustýring og rekjanleiki	Þrátt fyrir að gagnagrunnskóði sé geymdur í /SQL möppu og fari í gegnum breytingabeidnir og gæðaferla (kaflí 17.12 og 6.3), skapar skipting rökfræði á tvo staði, C# kóðagrunn og SQL scriptur, veikari rekjanleika en þegar öll rökfræði er á einum stað.

Aðskilnaðurinn er skýr: gagnagrunnurinn geymir, bakendinn hugsar. Sjá kafla 17.5 fyrir lista yfir bannaða gagnagrunnshluti og kafla 17.8 fyrir undanþágufærli ef nauðsyn krefst.

Sjá einnig tæknistefnu verkkaupa (kafla 6.1) fyrir rökstuðning á gagnagrunnstæknivali (Dapper.NET vs. full-ORM), sem byggir á sömu meginreglu: gagnsæi gagnagrunnsaðgerða og sjálfstæði gagnagrunnsumsjónar frá forritunarkóða.

17.2 Tegund gagnagrunna

Krafa	Lýsing
Sjálfgefinn gagnagrunnur	Microsoft SQL Server í nýlegri, studdri útgáfu
Val á öðru	Óheimilt nema formlegt samþykki tæknilegra sérfræðinga og gæðastjóra liggi fyrir
Hýsing (PreProd, Production)	Vélasalir verkkaupa (self-hosted)

17.3 Forritunarpakkar (ORM)

Samskipti við gagnagrunna skulu byggjast á beinum SQL skipunum.

ORM / Pakki	Staða	Ástæða
Dapper.NET	Heimilt	Létt hjúpun utan um beinar SQL skipanir; breytir ekki skemum; krefst ekki aukinna réttinda
Entity Framework / EF Core	Óheimilt	Sjálfvirk skemabreyting, falin hegðun, háar réttindakröfur
Hibernate / NHibernate	Óheimilt	Sama og að ofan
SQLAlchemy	Óheimilt	Sama og að ofan
Önnur ORM sem mynda eða breyta skemum sjálfvirk	Óheimilt	Draga úr gagnsæi, auka flækju og viðhaldskostnað

17.4 Aðgangsréttindi gagnagrunnsnotenda

Gagnagrunnsnotendur sem bakendapjónustur nýta skulu vera takmarkaðir við lágmarksréttindi:

Heimilt	Óheimilt
SELECT, INSERT, UPDATE, DELETE	CREATE, ALTER, DROP
EXECUTE (á samþykkt procedures ef undanþága hefur verið veitt)	DDL skipanir í keyrsluumhverfi

17.5 Bannaðir gagnagrunnshlutir

Eftirfarandi gagnagrunnshlutir og útfærslur eru óheimilar í gagnagrunnum sem þjónusta keyrsluhluta (online part) kerfa:

Gagnagrunnshluti	Staða	Ástæða
Stored Procedures sem útfæra domain rökræði	Óheimilt	Falin hegðun, erfið villuleit
Functions (scalar eða table-valued) með flæði	Óheimilt	Sama og að ofan
Triggers	Óheimilt	Falin hegðun, erfitt útgáfufurli
Views með flókinni rökræði	Óheimilt	Fela domain reglur
Scheduled Database Jobs	Óheimilt	Kjarnavirkni skal vera í forritunarkóða
Sérnsniðin / óstaðluð SQL-viðbótavirkni	Óheimilt	Bindur lausnir við tiltekna gagnagrunnsútgáfu
Eyðing gagna	Óheimilt	Eyðing gagna (hard delete) úr rekstrargagnagrunnum er óheimil nema lagaheimild liggi sérstaklega fyrir, í samræmi við lög nr. 77/2014 um opinber skjalasöfn. Lausnir skulu hannaðar þannig að gögn séu merkt sem óvirk (soft delete) frekar en þeim eytt.

17.6 Rekjanleiki eyðingar

Ef lagaheimild fyrir eyðingu liggur fyrir skal eyðing vera rekjanleg, skráð skal hver eyddi, hvenær og á hvaða grundvelli.

17.7 Leyfilegir gagnagrunnshlutir

Gagnagrunnshluti	Staða	Skilyrði
Views sem einfalda gagnasýn	Heimilt	Engin flókin rökfræði eða domain reglur
Indexes	Heimilt	Fyrir frammistöðu
Constraints og foreign keys	Heimilt	Fyrir gagnheilindi

17.8 Undanþágur á gagnagrunnsvirkni

Ef nauðsyn krefst þess að notuð séu stored procedures, functions eða önnur virkni sem er almennt bönnuð, skal:

- fá formlegt samþykki tæknilegra sérfræðinga verkkaupa,
- skjalfesta alla slíka virkni í skjölun kerfisins (hvar hún er notuð og tilgangur),
- tryggja að slík virkni sé hluti af útgáfustýrðum SQL scriptum.

17.9 Surrogate / Technical keys

Gagnagrunnskerfi skulu nota surrogate lykla til að einkvæmt auðkenna færslur.

Krafa	Lýsing
Primary keys	Tæknilegir, einfaldir, óbreytanlegir yfir líftíma færslu
Business-gildi (t.d. kennitölur)	Geymd í sér dálkum; ekki notuð sem primary keys
Foreign keys	Vísa á surrogate lykla, ekki business-gildi

17.10 Útgáfustjórnun gagnagrunna

Allar breytingar á gagnagrunnsskemum skulu vera fyrirfram skilgreindar, útgáfustýrðar og rekjanlegar.

Breytingar á skema (backward compatibility):

Eftirfarandi breytingar eru óheimilar í keyrsluumhverfum:

Óheimil breyting	Ástæða
Endurnefning dálka, taflna eða views	Brýtur eldri útgáfur
Breyting á gagnatýpum dálka	Brýtur eldri útgáfur
Fjarlæging dálka eða taflna	Brýtur eldri útgáfur
Breyting á primary eða foreign keys	Getur valdið gagnaósamræmi

Slíkar breytingar eru einungis heimilar í gegnum skilgreint og samþykkt umbreytingarferli.

17.11 Migration-ferlar:

Krafa	Lýsing
Handgerðar SQL scriptur	Allar breytingar skulu vera handskrifaðar SQL skipanir
ORM-byggð migration	Óheimilt
Útgáfustýring	Migration-skrár skulu vera nefndar eftir útgáfunúmeri og geymdar í /SQL/migrations
Afrit	Afrit skal tekið af gagnagrunni áður en breytingar eru framkvæmdar
Ábyrgð	Verksali skal ekki framkvæma beinar gagnagrunnsuppfærslur í PreProd eða Production nema í samráði við verkkaupa

17.12 Skipulag gagnagrunnskóða

Allur gagnagrunnskóði skal vera geymdur sem handgerðar SQL skipanir í kóðageymslum verkkaupa, undir /SQL möppu.

Krafa	Lýsing
Staðsetning	/SQL mappa í sama repo og bakendaþjónusta; eða sérstakt repo ef margar þjónustur deila grunni
Uppskipting	Aðgreindar eftir tilgangi og númeraðar eftir keyrsluröð

Byggjanlegt frá grunni	Hægt skal vera að byggja gagnagrunn frá grunni eingöngu með SQL scriptunum
Rekjanlegt	Skýrt hvaða scriptur tilheyrja hvaða útgáfu

Dæmi um skipulag:

```
/SQL
├── 01-create-database.sql
├── 02-tables.sql
├── 03-views.sql
├── 04-indexes.sql
├── 05-users.sql
├── 06-roles.sql
├── 07-grants.sql
├── 08-static-data.sql
├── migrations/
│   ├── 2025.05.12.sql
│   └── 2025.06.01.sql
```

18 Útgáfuferli og keyrslustýring

 **Leiðarljós**

Þessi kafli skilgreinir þær tæknilegu forsendur sem hugbúnaður þarf að uppfylla til að vera útgáfuhæfur, keyrsluhæfur og rekstrarhæfur í innviðum verkkaupa.

Kaflinn fjallar ekki um daglegan rekstur, samþykktir, neyðarferla eða rekstrarábyrgð. Slík atriði eru skilgreind í *Rekstrarhandbók verkkaupa*.

Nauðsynlegt er að verksali eigi samtál við tækniteymi verkkaupa **snemma í þróunarferlinu**, helst áður en forritun hefst, til að tryggja samræmi við útgáfu- og keyrslustefnu.

18.1 Ábyrgðarskipting

Ábyrgðarsvið	Verkkaupi	Verksali
Hýsing og rekstur keyrsluumhverfa (PreProd, Production)	✓	
Kubernetes innviðir, orkestrun og dreifing	✓	
Leyndarmálageymsla (Vault) og rekstur hennar	✓	
Sjálfvirkir útgáfuferlar (CI/CD)	✓	
Samþykktir og neyðarferlar	✓	
Að hugbúnaður sé útgáfuhæfur og keyrsluhæfur í innviðum verkkaupa		✓
Þróunarumhverfi, hermipjónustur og innri sjálfvirkni		✓
Docker stillingar og Kubernetes samhæfni		✓

18.2 Almennar kröfur til hugbúnaðar

Allur hugbúnaður sem afhentur er verkkaupa til útgáfu skal uppfylla eftirfarandi:

Krafa	Lýsing
Stöðuleysi (stateless)	Lausnir mega ekki viðhalda innri stöðu milli beiðna. Lárétt skölun skal vera möguleg.
Endurræsingapól	Lausnir skulu þola endurræingu tilvika án gagnataps eða ósamræmis
Engin handvirk skref	Lausnir mega ekki krefjast handvirkra aðgerða við útgáfu eða ræingu
Logging í stdout/stderr	Allar logfærslur skulu prentaðar í stdout/stderr; skrif í skrár á disk er óheimilt
Structured logging	Logfærslur á JSON sniði samkvæmt ECS staðli (sjá kafla <i>Rekstur og eftirlit</i>)
Metrics aðgengilegar	Tæknilegar mælingar skulu vera aðgengilegar á stöðluðu sniði
Stillingar úr keyrsluumhverfi	Lausnir skulu lesa allar stillingar úr umhverfisbreytum eða keyrslustýrðum stillingaveitum

Engar harðkóðaðar stillingar	Óheimilt að treysta á harðkóðaðar stillingar eða umhverfissértækar skrár
Samræmi við CI/CD	Lausnir mega ekki krefjast sérstakra undantekninga í útgáfuflerum

18.3 Docker (einingaumhverfi)

Allar lausnir skulu vera pökkunarhæfar og keyrsluhæfar sem Docker einingar.

Krafa	Lýsing
Dockerfile í rót	Dockerfile skal vera í rót verkefnis
Ein eining per keyrsluhluta	Lausn skal framleiða eina Docker einingu per sjálfstæða keyrslueiningu
Engin bygging í Docker	Build (compile) skal ekki fara fram í Docker einingu; einungis pökkun forsmíðaðs kóða
Engar prófanir í Docker	Prófanir skulu keyrðar utan Docker eininga
Ekki root	Hugbúnaður skal ekki keyra sem root notandi
Litlar einingar	Leitast skal við að hafa einingar eins litlar og mögulegt er
Engin leyndarmál	Leyndarmál mega aldrei vera hluti af Docker einingu, Dockerfile eða build ferli
Logging í stdout	Loggar í skrár á disk er óheimilt; allt í stdout/stderr
Docker-in-Docker	Óheimilt

Docker er notað hjá verkkaupa í sínu upprunalega tilgangi: að pakka forsmíðuðum hugbúnaði og skilgreina keyrsluumhverfið sem hann keyrir í. Docker er ekki smíðaverkfæri. Þýðing, prófanir og gæðamat eiga heima í CI-ferlinu sem á undan kemur.

Krafa um að bygging (compile) fari ekki fram í Docker einingu byggir á tveimur forsendum:

Gæðahlið krefjast aðgangs að milliafurðum. Gæðahlið verkkaupa (kafli 6.4), einingaprófanir, kódagreining, þekjumat, keyra utan Docker og þurfa aðgang að þýddum kóða áður en hann er pakkaður. Ef þýðing fer fram inni í Docker eru þessar milliafurðir óaðgengilegar og gæðahliðin verða virkjalaus.

Flækjustig skal vera í lágmarki. Þegar Docker er notað til þýðingar verður Docker-ferlið sjálft að flóknu smíðaumhverfi sem krefst sérþekkingar umfram það sem pökkun krefst. Þetta leiðir af sér vandamál eins og Docker-in-Docker háð, margþrepa Dockerfile sem erfitt er að viðhalda, og ósamræmi milli smíðaumhverfis inni í Docker og CI-umhverfis utan þess. Með því að halda Docker-ferlinu sem einföldri pökkunaraðgerð á forsmíðuðum kóða er flækjustigið lágmarkað, Docker einingar eru minni og dreifing hraðari.

Sjá viðauka „Docker kröfur og bestu venjur“ fyrir fulla kröfulýsingu um build ferli, port, startup, networking og stillingar.

18.4 Kubernetes

Kubernetes er keyrsluumhverfi verkkaupa. Allar lausnir (að undanskildum gagnagrunnum) skulu geta keyrt í Kubernetes.

Krafa	Lýsing
Kubernetes-hæfni	Allar lausnir skulu vera keyrsluhæfar í Kubernetes
GitOps samhæfni	Lausnir skulu vera samhæfðar GitOps útgáfumódeli; innviðakóði í .kube/möppu
Liveness probe	Lausn skal bjóða upp á liveness probe (t.d. /system/alive)
Readiness probe	Lausn skal bjóða upp á readiness probe (t.d. /system/ready)
Logging og metrics	stdout/stderr logging og metrics aðgengilegar; söfnun á ábyrgð innviða verkkaupa
Stillingar og secrets	Öll keyrsluleyndarmál í Vault; sprautuð inn við keyrslu
Stöðuleysi	Stateless; engin háð við staðbundið ástand
Engin disk-skrif	Skrif á disk í runtime óheimilt nema tímabundið (t.d. /tmp)
Engar beinar breytingar	Verksali má ekki gera beinar breytingar í Kubernetes cluster

Sjá viðauka „Kubernetes kröfur og samhæfni“ fyrir kröfur um Argo CD samhæfni, probe útfærslu, og bann og takmarkanir.

Sjá Rekstrarhandbók verkkaupa fyrir rekstur Kubernetes, Argo CD stillingar, vöktun, afritun og neyðarferla.

18.5 Geymsla leyndarmála

Öll leyndarmál (lykilorð, API-lyklar, auðkenningarlyklar, tengistrengi) skulu vera geymd í miðlægri leyndarmálageymslu verkkaupa.

Krafa	Lýsing
Geymsla	Í leyndarmálageymslu verkkaupa (Vault); aldrei í kóða, Docker einingum eða stillingarskrám
Innsprautun	Leyndarmál sprautuð inn við keyrslu í Kubernetes
Birting í loggum	Óheimilt, leyndarmál mega aldrei birtast í loggum eða debug úttaki
Meðhöndlun í þróunarumhverfi	Ábyrgð verksala; skal endurspegla keyrslulíkan verkkaupa eftir fremsta megni
Flutningur milli umhverfa	Leyndarmál úr þróunarumhverfum skulu aldrei flytjast í rekstrarumhverfi

Sjá viðauka „Geymsla leyndarmála“ fyrir útfærsluatriði (.NET appsettings, environment breytur, Vault slóðir og Kubernetes keyrslu).

19 Rekstur og eftirlit

Þessi kafli skilgreinir þær tæknilegu kröfur sem hugbúnaður þarf að uppfylla til að styðja sjálfvirkan rekstur, vöktun og bilanagreiningu í innviðum verkkaupa.

Kaflinn fjallar um hvað verksali þarf að byggja inn í hugbúnað. Hvernig verkkaupi nýtir þessi gögn til vöktunar, viðbragða og reksturs er skilgreint í Rekstrarhandbók verkkaupa.

Verksölum er skylt að nýta staðlaðar forritunareiningar verkkaupa til að tryggja samræmi í mælingum, keyrslustöðu, útgáfupplýsingum og loggum. Frávik eru einungis heimil með formlegri undanþágu.

Sjá viðauka „Stöðluð forritunarstöfn verkkaupa“ fyrir lista yfir viðeigandi forritunarsöfn.

19.1 Mælingar (Metrics)

Allar þjónustur skulu safna og birta staðlaðar afkastamælingar til að styðja sjálfvirktt eftirlit og rekstrargreiningu.

Krafa	Lýsing
Staðall	Prometheus snið
OpenTelemetry	Heimilt innan lausnar, en mæligögn skulu vera samhæf Prometheus
Endapunktur	/system/metrics, birtir mæligögn á Prometheus sniði
Aðgengi	/system/metrics skal ekki vera aðgengilegur af internetinu; einungis innan innra nets
Staðlaður pakki	Verksala er skylt að nota stöðluðan mælingarpakka verkkaupa

19.1.1 Lágmarks mæliflokkur

Eftirfarandi flokkar mæligagna skulu vera til staðar þar sem við á:

Flokkur	Dæmi
Afköst þjónustu	Fjöldi beiðna, svartími, villuhlutfall
Ytri auðlindir	Gagnagrunnstengingar, skyndiminni, ytri API köll
Kerfisálag	CPU, minni, þræðir, tengingar
Domain mælingar	Sérhæfðar mælingar ef skilgreindar í verkefni

19.2 Keyrslustaða (Health endpoints)

Kröfur um heilsuathuganir (liveness og readiness probes) eru skilgreindar í kafla 18 Útgáfufærli og keyrslustýring og viðauka Kubernetes kröfur og samhæfni.

Þessi kafli bætir við eftirfarandi tæknilegum kröfum:

Krafa	Lýsing
Staðlaður pakki	Verksala er skylt að nota stöðluðan heilsuathugunar pakka verkkaupa

Readiness efni	/system/ready skal prófa tengingar við nauðsynlegar ytri einingar (gagnagrunn, skyndiminni, innri þjónustur)
Annað port	Heimilt er að birta /system/* endapunkta á porti 9102
Ytri aðgengi	Allir /system/* endapunktur skulu vera lokaðir fyrir ytri umferð (internetinu)

19.3 Útgáfuupplýsingar (Version endpoint)

Allar þjónustur og vefkerfi skulu birta staðlaðar útgáfuupplýsingar til að styðja sjálfvirka útgáfuflerla, sannreyningu og rekjanleika.

Krafa	Lýsing
Endapunktur	/version
Aðgengi	Skal vera aðgengilegur af internetinu
DDOS vörn	Endapunktur þarf að uppfylla öryggiskröfur vegna ytri aðgengis
Staðlaður pakki	Verksala er skylt að nota stöðluðan útgáfuupplýsinga pakka verkkaupa

19.3.1 Svæði í /version svari

Svæði	Lýsing
commitSha	SHA-gildi Git commit sem notað var til að smíða útgáfuna. Tryggir nákvæman rekjanleika í kóðagrunn.
workflowRunId	Auðkenni CI keyrslu sem byggði og gaf út útgáfuna. Gefur kost á að rekja prófunartilvik og gæðaeftirlit.
version	Útgáfunúmer samkvæmt CalVer reglum verkkaupa.

Gildin eru fastsett sjálfvirk í forritunarkóða þegar hann er byggður sem hluti af CI/CD ferlum verkkaupa. Handvirk skilgreining eða yfirskrift útgáfuupplýsinga í kóða er **óheimil**.

Dæmi um svar:

```
{
  "commitSha": "c9a911a85e0b29c4f08bdf547e37376d800981c1",
  "workflowRunId": "304328",
  "version": "2025.08.10"
}
```

19.4 Logging (atvikaskráning)

Allar lausnir skulu skrá atvik (logs) á staðlaðan, strúktúraðan og rekjanlegan hátt.

Krafa	Lýsing
Snið	Structured logging á JSON formi
Staðall	Elastic Common Schema (ECS)
Útprentun	stdout/stderr; skrif í skrár á disk óheimilt
Staðlaður pakki	Verksala er skylt að nota stöðluðan logging pakka verkkaupa
Production logging-stig	WARNING og hærra; lausnir skulu skila nothæfum loggum til bilanagreiningar á þessu stigi

19.5 Rekjanleiki milli þjónusta

Allar logfærslur **verða** að innihalda lágmarks rekjanleika:

Svæði	Krafa
trace.id	Skylda
span.id	Skylda
x-correlation-id	Skylda ef til staðar í innkomandi beiðni
x-forwarded-*	Skylda ef til staðar í innkomandi beiðni

Þjónustur verða að áframsenda rekjanleika-hausar (headers) óbreytta þegar kallað er á aðrar þjónustur. Proxy-þjónar, eldveggir og millistig mega ekki fjarlægja þessar upplýsingar.

19.6 Stöðluð viðbótarsvæði

Eftirfarandi ECS-samhæfð viðbótarsvæði skulu notuð þar sem við á:

ECS slóð	Lýsing
labels.application	Einkvæmt nafn lausnar. Mælt er með að nota nafn kóðageymslu.
labels.x-correlation-id	Rakningarauðkenni úr x-correlation-id HTTP header
labels.x-road-client	X-Road auðkenni úr x-road-client HTTP header (ef við á)

19.7 Bann og takmarkanir

Atriði	Staða
Viðkvæm gögn í loggum (kennitölur, heilsufarsupplýsingar, lykilorð, API-lyklar)	Óheimilt
Secrets eða aðgangsupplýsingar í loggum, jafnvel í debug ham	Óheimilt
Logskrár á disk í Kubernetes umhverfi	Óheimilt

20 Viðauki: Kröfulisti þróunarhandbókar

Viðauki þessi dregur saman bindandi kröfur og ráðleggingar úr þróunarhandbók verkkaupa í hreinar, mælanlegar útboðskröfur. Kröfur eru flokkaðar í **SKAL** (ófrávíkjanlegt) og **GETUR** (heimilt / valkvætt / ráðlagt).

Viðaukinn er ætlaður til stuðnings og notkunar í útboðsgögnum, samningum og gæðamati. Ef árekstur verður milli kröfuskjala þá gilda sértæk útboðsgögn framar þessari handbók. Við árekstra innan handbókar milli viðauka og meginmáls kafla gildir meginmál kaflans.

Tilvísanir í kafla vísa í meginmál þróunarhandbókar. Kröfur sem einnig koma fram í tæknistefnu verkkaupa eru merktar með (T) í flokkadálki.

20.1 Kóðageymslur

Flokkur	Krafa	Tegund
Geymsla	Allur hugbúnaðarkóði skal geymdur í git-umhverfi verkkaupa	SKAL
Skipulag	Ein kóðageymsla per sjálfstætt útgefna einingu; mono-repo óheimilt	SKAL
Greinar	Ein langlíf kóðagrein (main); beinar breytingar á main óheimilar	SKAL
Ferli	Allar breytingar í gegnum breytingabeirðir (Pull Requests) og kóðarýni	SKAL
Nafnareglur	Kóðageymslur skulu fylgja skilgreindum nafnareglum (lágstafir, bandstrik)	SKAL
Nafnareglur	Tímabundin heiti og útgáfunúmer í nafni óheimil	SKAL
Viðskeyti	Viðskeyti á kóðageymslur samkvæmt skilgreindri töflu	SKAL
Kóðagreinar	Breytingabeirðir skulu fylgja skilgreindum nafnareglum á kóðagreinum (feature/, qa/, release/, hotfix/)	SKAL
Rekjanleiki	Heiti breytingabeirðni skal vera nákvæmlega sama og heiti kóðagreinar	SKAL
Eyðing	Verksala skylt að eyða öllum afritum kóðageymsla við verklok	SKAL

20.2 Viðtöku- og útgáfuferlar

Flokkur	Krafa	Tegund
Sjálfvirkt mat	Allar breytingabeirðir fara í gegnum sjálfvirka gæðaferla; ef bregst er PR hafnað	SKAL
Kóðarýni	Verkkaupi framkvæmir formlega kóðarýni; niðurstöður skráðar í staðlað yfirferðarskjal	SKAL
Svörun	Verksali ber ábyrgð á að svara öllum athugasemdum í yfirferðarskjali	SKAL
Critical	Athugasemdir á Critical stigi koma í veg fyrir samþykkt útgáfu	SKAL
Útgáfur	Útgáfur byggðar á main; óheimilt að gefa út af öðrum greinum	SKAL
Production	QA-útgáfur og -rc merktar útgáfur óheimilar í Production	SKAL
Samþykkt	Útgáfa í Production krefst handvirkrar samþykktar verkkaupa	SKAL
Óheimilt	Verktökum óheimilt að gefa út hugbúnað í prófunar- eða rekstrarumhverfi án samþykkis	SKAL

20.3 Gæðaferlar

Flokkur	Krafa	Tegund
---------	-------	--------

Gæðahlið	Stillingar á gæðahliðum í höndum verkkaupa; óleyfilegt að breyta án samþykkis	SKAL
Viðvaranir	Viðvaranir meðhöndlaðar sem villur (treat warnings as errors)	SKAL
Clean-as-you-code	Gæðahlið bera kóða í breytingabeidni saman við núverandi stöðu	SKAL
Gervigreind	Verkkaupi nýtir gervigreindarlausnir sem hluta af gæðaeftirliti	SKAL
Innri gæðir	Verksala ráðlagt að keyra sambærilega kóðagreiningu innanhúss	GETUR

20.4 Skil á hugbúnaðarafurðum

Flokkur	Krafa	Tegund
Tíðni	Skil að lágmarki á þriggja mánaða fresti nema annað sé samið	SKAL
Innihald	Skil skulu ná til kóða, gagnagrunns, prófana, skjölunar, Docker og Kubernetes stillinga	SKAL
Ábyrgð	Verksali tilnefni ábyrgðaraðila skila	SKAL
Rekjanleiki	Rekjanleiki tryggður milli kóðagreina, breytingabeidna, prófana, samþykta og útgáfa	SKAL
MVP	Notkun MVP nálgunar æskileg en leysir ekki undan skyldu til reglulegra skila	GETUR
Neyðarfrávik	Frávik frá skilareglum einungis heimil í neyðartilvikum með formlegu samþykki	SKAL
Úttektaráætlun	Exit plan afhent við fyrstu formlegu skil og uppfært samhliða verulegum breytingum á lausn	SKAL
Gagnaútflutningur	Gagnaútflutningsleiðbeiningar skulu fylgja skilum; lýsa sniði, ferli og tengslum milli gagnatafla	SKAL

20.5 Forritunarmál og tæknival

20.5.1 Bakendi og API

Flokkur	Krafa	Tegund
Bakendi	Allar bakendaþjónustur í C# á .NET (nýjasta LTS) með ASP.NET	SKAL
Gagnasnið	JSON sem gagnasnið	SKAL
Gagnavinnsla	SQL + Dapper.NET	SKAL
Keyrsla	Docker	SKAL
Undanþágur	Frávik frá C#/.NET einungis heimil með formlegri undanþágu	SKAL

20.5.2 Framendi, Leið A: ASP.NET Server-Side Rendering

Flokkur	Krafa	Tegund
Framendi (Leið A)	ASP.NET MVC eða Razor Pages með Server-Side Rendering	SKAL
HTMX	Heimilt sem hjálpartæki í Leið A; óheimilt sem burðarás	GETUR
JS/TS í Leið A	Einungis sem hjálpartæki; aldrei burðarás framendalausna	SKAL

20.5.3 Framendi, Leið B: Gervigreindarsmiðaður React framendi

Flokkur	Krafa	Tegund
Leið B, val	Leið B er heimil sem viðbótarleið ef öll skilyrði kafla 8.2.2 eru uppfyllt. Val á leið skal skjalfest í ADR og samþykkt áður en þróun hefst.	GETUR
Al-smíði	Framendakóði í Leið B skal vera smíðaður eingöngu með gervigreindarþróunarverkfærum. Verksali undirritar yfirlýsingu; verkkaupi áskilur sér rétt til stikkprófana.	SKAL
Handvirkar breytingar	Einungis heimilt í stillingaskrá og staðfæringaskrá (i18n). Aldrei á React-ihlutum, leiðum, domain rökfræði eða prófunarkóða.	SKAL
Forritunarmál (Leið B)	TypeScript með strict mode	SKAL
Framework (Leið B)	React (með Vite eða sambærilegu). Next.js, Remix, Gatsby, Astro og önnur meta-framework óheimil.	SKAL
Bakendi (Leið B)	Öll gagnaskipti við .NET bakenda-API. Engin bakendavirkni í Node.js/TypeScript.	SKAL

Node.js í keyrslu	Engin Node.js keyrsla í rekstrarumhverfi. Node.js heimilt í þróunar- og CI/CD umhverfi.	SKAL
Stöðustjórnun (Leið B)	Innbyggð React stöðustjórnun eingöngu (useState, useReducer, useContext). Ytri stöðustjórnunarsöfn óheimil.	SKAL
Þriðja aðila UI-söfn	Umfangsmikil þriðja aðila UI-söfn (Material UI, Ant Design, Chakra UI) óheimil	SKAL
Flækja við endursmíði	Framendinn skal uppfylla viðmið um flækju við endursmíði samkvæmt kafla 8.2.2	SKAL
.agent/ mappa (Leið B)	Framendaverkefni skal hafa .agent/ möppu skv. kafla 11.1; metið með auknum þunga við viðtöku	SKAL
API-samskipti (Leið B)	Samskipti við bakenda byggð á OpenAPI-skilgreiningum	SKAL
Dreifing (Leið B)	Statískar skrár, frá .NET bakenda eða gámi með statískum skráarþjóni (nginx/caddy). Enginn Node.js runtime í gámi.	SKAL

20.5.4 Óheimilar framendatæknilausnir (gildir fyrir báðar leiðir A & B)

Flokkur	Krafa	Tegund
Angular, Vue, Svelte	Angular, Vue, Svelte og álíka óheimil	SKAL EKKI
Next.js o.þ.h.	Next.js, Remix, Gatsby, Astro og önnur meta-framework óheimil	SKAL EKKI
Node.js bakendi	NestJS, Express, Fastify og álíka óheimil; API-þróun eingöngu í .NET/C#	SKAL EKKI
WebForms	ASP.NET WebForms óheimilt	SKAL EKKI
CMS	Vefumsjónarkerfi óheimil nema skrifleg undanþága	SKAL

20.5.5 Aðrar kröfur

Flokkur	Krafa	Tegund
Hönnunarstaðlar	Framendalausnir (báðar leiðir) skulu fylgja hönnunarstöðlum og útlitsstílsniði verkkaupa	SKAL
Snjallforrit	Native snjallforrit eru undanþága; veflausnir eru meginregla	SKAL
Aðgengi snjallf.	Virgni ásættanleg á ≥75% snjalltækja markhóps á útgáfudegi	SKAL

20.6 Hönnun og arkitektúr

Flokkur	Krafa	Tegund
Samþykki	Hugbúnaðarsmíði má ekki hefjast fyrr en formlegt samþykki á arkitektúr	SKAL
Meginreglur	SOLID, SoC, ONION, CLEAN, KISS, DRY, YAGNI	SKAL
Einföldun	Rótæklega einfaldur tækniarkitektúr; óþarfa flækju hafnað	SKAL
Microservices	Smáþjónustur (micro/nano-services) óheimilar	SKAL
Vertical slice	Lóðréttar sneiðar og REPR-högun æskileg fyrir WebAPI	GETUR
Bakgrunnsvinnslur	Sjálfstæð console applications sem Kubernetes CronJobs	SKAL
Hangfire o.þ.h.	In-process job schedulers (Hangfire, Quartz.NET) óheimilar	SKAL
Caching	Á netþjóni; ekki í framenda	SKAL
OpenAPI	Allar þjónustur skulu hafa OpenAPI-skilgreiningar	SKAL
Þolgæði	Viðurkennd forritunarsöfn (Polly.NET); sérsníðaður þolgæðakóði óheimill	SKAL
Samþætting	Samskipti við ytri kerfi í gegnum skilgreind API; beinar gagnagrunnstengingar milli þjónusta óheimilar	SKAL

20.6.1 Leið A og bakendi (allar leiðir)

Lagskipting, aðskilnaður	Hvert arkitektúrlag skal vera aðskilt .csproj verkefni. Þýðandinn framfylgir háðastefnu í gegnum ProjectReference.	SKAL
Lagskipting, namespace	IDE0130 (Namespace does not match folder structure) skal vera virkt á villustig (error severity) í öllum lausnum.	SKAL
Lagskipting, samræmi	Öll kóðageymsla verkkaupa skulu nota sömu rótarmöppunöfn samkvæmt skilgreiningu í viðauka. Sérsníðuð skipulagsmynstur eru óheimil.	SKAL
Lagskipting, arkitektúrprófanir	Einingaprófanir skulu sannreyna lagskiptingarreglur, þar á meðal að kóði í tilteknu lagi vísi ekki í lög sem hann á ekki samskipti við.	SKAL

Lagskipting, breytingar	Verulegar breytingar á rótarmöppuskipulagi skulu ekki gerðar án samþykkis verkkaupa.	SKAL
-------------------------	--	------

20.6.2 Leið B framendi

Flokkur	Krafa	Tegund
Lagskipting, samræmi (Leið B)	Leið B kóðageymsla skulu nota sömu rótarmöppunöfn (src/components/, src/pages/, src/services/, src/types/, Tests/) samkvæmt skilgreiningu í viðauka.	SKAL
Lagskipting, möppudýpt (Leið B)	Hámark þrjú möppustig undir src/.	SKAL
Lagskipting, ESLint (Leið B)	ESLint innflutningsreglur (eslint-plugin-import, no-restricted-paths) skulu framfylgja innflutningsstefnu og vera keyrðar í CI/CD.	SKAL
Lagskipting, engin rökfræði (Leið B)	Leið B framendi skal ekki innihalda sjálfstætt rökfræðilag eða flæðistjórnun. Öll viðskiptarökfræði er í .NET bakenda.	SKAL
Lagskipting, breytingar (Leið B)	Verulegar breytingar á rótarmöppuskipulagi skulu ekki gerðar án samþykkis verkkaupa.	SKAL

20.7 Forritunarkóði og vinnureglur

Flokkur	Krafa	Tegund
Linting	Allur kóði í gegnum sjálfvirka samræmingu og gæðamat	SKAL
PR umfang	Breytingabeiðnir einungis með breytingum tengdum skilgreindu verki	SKAL
Nafnarými	Byrja á [Digital,Ministry,Directorate].Health.PROJECTNAME	SKAL
Virknisflögg	Heimilt með innbyggðri virkni (.NET FeatureFilter) eða build-stýrt	GETUR
Virknisflögg bann	Má ekki fela ófullbúna virkni í rekstrarumhverfi né komast hjá gæðaferli	SKAL
býðingar	Styðja íslensku og ensku að lágmarki	SKAL
Skjölun í kóða	Allir public klasar, föll, enums skulu hafa XML summary	SKAL
Athugasemdir	Á ensku; hlutlægt og faglegt málfar	SKAL
Endurnýting	Verksali skyldugur til að nýta staðluð forritunarsöfn verkkaupa	SKAL
Ytri söfn	Allur kóði og söfn í keyrsluvöru skulu fylgja í kóðageymslum verkkaupa	SKAL
Sjálfbærni	Lausnir byggjanlegar og keyranlegar án háðs við ytri netþjónustur	SKAL

20.8 Gervigreind í hugbúnaðarþróun

Flokkur	Krafa	Tegund
AI í þróun	Sterk væntingu um skipulagða notkun gervigreindar í þróunarferli	GETUR
.agent/ mappa	Virða og viðhalda .agent/ möppu; hluti af gæðamati við skil	SKAL
Ábyrgð	Mannlegir forritarar bera fulla ábyrgð á öllum kóða, óháð uppruna	SKAL
Gagnaöryggi	Óheimilt að senda heilbrigðisgögn eða kóða verkkaupa í opinberar AI þjónustur	SKAL
Gagnsæi	Verksali skal upplýsa um AI verkfæri, aðferðafræði og breytingar	SKAL
AI agentar í afurð	Skulu vera ASP.NET backendþjónustur í C# með samþykktu Microsoft forritunarsafni	SKAL
Sérsmíðaðir agentar	Heimasmíðuð eða ósamþykkt agent-framework óheimil	SKAL
LLM háð	Lausn skal vera keyranleg í mock-mode án raun-LLM	SKAL
Multi-agent	Fjöl-agent kerfi óheimil nema formleg samþykkt	SKAL
Þróunarferli	Skipulagður fimm-áfanga gervigreindarþróunarferill (áætlanagerð, framkvæmdaáætlun, gap-rýni, útfærsla, lokastaðfesting) skv. viðauka „Æskilegt gervigreindarþróunarferli“	GETUR
Þróunarferli	Vinnulagið er bindandi fyrir Leið B framendaverkefni	SKAL
Áætlanagerð	Áætlunar- og framkvæmdaskrár í -docs kóðageymslu, númeraðar frá 01, í markdown sniði	SKAL ¹
Framkvæmdaáætlun	Framkvæmdaáætlun skal innihalda afritunarhæfar vinnulotu kvaðningar og lýsingu á keyrslufyrirkomulagi (dependency graph)	SKAL ¹
Rýni	Rýni og samþykki tæknilegs ábyrgðaraðila SH áður en útfærsla vinnulotu kvaðninga hefst	SKAL ¹
Sannvottun	Mannleg sannvottun á útfærslu hverrar vinnulotu áður en næsta hefst	SKAL ¹

¹ Krafa á við þegar verksali nýtir vinnulagið, annaðhvort vegna þess að um Leið B er að ræða eða vegna þess að verksali velur að fylgja vinnulaginu. Ef verksali fylgir ekki vinnulaginu gilda almenn ákvæði kafla 11.1.

20.9 Öryggi og auðkenningar

Flokkur	Krafa	Tegund
Security by design	Öryggi að leiðarljósi frá upphafi	SKAL
Leyndarmál	Aldrei í kóða eða stillingarskrám; afhent við keyrslu í gegnum öruggt leyndarmálakerfi	SKAL
Dulkóðun í flutningi	Öll samskipti dulkóðuð með viðurkenndum stöðlum (TLS, HTTPS, VPN)	SKAL
Dulkóðun í geymslu	Database-level eða disk encryption; AES-256 eða sambærilegur styrkur að lágmarki	SKAL
Sérsmíðaðar lausnir	Eigin dulkóðun, auðkenning eða öryggislausnir óheimilar	SKAL
Stöðugir pakkar	Öryggispakkar í stable/release útgáfum; alpha/beta óheimil	SKAL
Öryggisstaðlar	Verksali skal sýna fram á þekkingu á OWASP og CWE; lausnir skulu verjast þekktum veikleikum	SKAL
MFA	Allar lausnir með notendaviðmóti skulu styðja fjölpátta auðkenningu	SKAL
JWT	Viðurkennd forritunarsöfn; endurnýting ytri tóka óheimil	SKAL
Claims-based	Aðgerðamiðuð heimildaveiting (claims-based); ekki eingöngu role-based	SKAL
Lágmarksaðgangur	Aðgangsheimildir byggðar á meginreglu um lágmarksréttindi (least privilege)	SKAL
Nafnasnið	Heimildir á forminu Scope-Operation (Read/Write)	SKAL
Frontend-heimildaveiting	Aðgangsáskvarðanir teknar í bakenda; framendi birtir en framfylgir ekki heimildum	SKAL
Framendi cookies	Httponly session cookie; ekkert í localStorage	SKAL
SRI	Subresource Integrity ef kóði sóttur af öðrum lénunum	SKAL
Throttling	Stigvaxandi töf við ógildar auðkenningar; engar varanlegar læsingar	SKAL
Rekjanleiki	Allar aðgerðir sem snerta viðkvæm gögn skulu vera rekjanlegar	SKAL
Endir endapunktur	Ytri söfn sem bæta sjálfvirk við endapunktum óheimil	SKAL
Gagnavinnslusamningur	Gagnavinnslusamningur skal vera til staðar þegar persónuupplýsingar eru unnar af þriðja aðila (lög nr. 90/2018)	SKAL

20.10 Hugbúnaðarprófanir

Flokkur	Krafa	Tegund
Einingaprófanir	≥80% þekja á nýjum kóða	SKAL
Heildarþekja	≥65%; má ekki lækka milli útgáfa	SKAL
Meginlínur	Prófaðar með prófi fyrir eðlilega virkni og villumeðhöndlun	SKAL
Samþættingarpr.	Allir ytri endapunktur með prófi fyrir virkni og villu	SKAL
Óháð auðkenningu	Þjónustur prófanlegar án raunverulegra ytri auðkenningarþjónusta	SKAL
Viðmótsprófanir	Í lágmarki; Cypress eða Playwright	GETUR
Aðgengisprófanir	Sjálfvirkar aðgengisprófanir sem hluti af CI	SKAL
Handvirkar pr.	Skjalfestar og uppfærðar samhliða breytingum	SKAL
Samþykktarpr.	Skipulagðar með ≥2 vikna fyrirvara; starfsmaður verkkaupa viðstaddur	SKAL
GDPR	Prófanir með Íslensk heilbrigðisgögn innan EEA	SKAL
Prófunarrámmar	Viðurkennd framework (xUnit, junit); heimasmiðuð óheimil	SKAL
Prófunargögn	Skulu fylgja afhendingu	SKAL

20.11 Aðgengismál

Flokkur	Krafa	Tegund
WCAG A	Öll skilyrði uppfyllt í öllum lausnum	SKAL
WCAG AA	Öll skilyrði uppfyllt í öllum lausnum	SKAL
WCAG AAA (valin)	15 tilgreind AAA skilyrði skulu uppfyllt	SKAL
WAI-ARIA (almenningur)	Viðeigandi ARIA merkingar í lausnum ætluðum almenningi	SKAL

WAI-ARIA (heilbr.st.)	Leitast við að uppfylla WAI-ARIA; verkkaupi metur í hverju verkefni	GETUR
Skjálesari	Lausnir prófaðar með skjálesara fyrir hverja útgáfu	SKAL
Sjálfvirkir aðgengi	Sjálfvirkir aðgengisprófanir sem hluti af CI-ferlum	SKAL

20.12 Hugbúnaðarskjölun

Flokkur	Krafa	Tegund
Uppfærð við skil	Skjölun uppfærð samhliða breytingabeiðni; annars galli	SKAL
Docs/ mappa	Tæknileg skjölun í /Docs möppu með INDEX.md	SKAL
README.md	Í rót hverrar kóðageymslu; lýsing, tæknival, ytri tengsl, frávik, uppsetning	SKAL
ADR	Veigamiklar arkitektúrakvarðanir skjalfestar	SKAL
Sequence rit	Kjarnaflæði, ytri þjónustur, öryggisflæði	SKAL
Rekstrarskjölun	Keyrsluumhverfi, stillingar, útgáfufurli, logging, viðhaldsverk	SKAL
Viðkvæm gögn	Aldrei í skjölun eða kóða; miðlað á öruggan hátt	SKAL
Snið afhendingsskjala	PDF eða Word; útgáfunúmer samræmt	SKAL
Úttektaráætlun	Exit plan, skref, forsendur, tími og kostnaður við yfirtöku lausnar til annars aðila	SKAL
CronJob skjölun	Bakgrunnsvinnslur skjalfestar með tíðni, inntaki, úttaki og villumeðhöndlun	SKAL

20.13 Geymsla og miðlun gagna

Flokkur	Krafa	Tegund
Heilbrigðisgögn	Mjög viðkvæm; lágmörkun, aðgangsstýring, rekjanleiki	SKAL
Miðlun	Einungis í gegnum öruggar þjónustur verkkaupa; tölvupóstur/fjarfundir/kóðageymslur óheimilt	SKAL
Raunveruleg gögn	Óheimil í þróunar-/prófunarumhverfi nema skriflegt leyfi	SKAL
Dulkóðun í flutn.	TLS, HTTPS, VPN	SKAL
Dulkóðun í geymslu	Database-level eða disk encryption	SKAL
Dulkóðunarstaðall	AES-256 eða sambærilegur styrkur að lágmarki	SKAL
Gagnagrunnur	MS SQL Server; annað krefst formlegs samþykkis	SKAL
ORM	Dapper.NET heimilt; Entity Framework, Hibernate o.þ.h. óheimil	SKAL
Réttindi	Lágmarksréttindi (SELECT, INSERT, UPDATE, DELETE); DDL óheimilt	SKAL
Bannaðir hlutir	Stored Procedures með rökfræði, Triggers, Scheduled Jobs, sérsmíðuð SQL-viðbótavirkni	SKAL
Surrogate keys	Tæknilegir primary keys; business-gildi ekki sem lyklar	SKAL
Backward compat.	Endurnefning, tegundabreyting, fjarlæging dálka/taflna óheimil í keyrslu	SKAL
Migration	Handgerðar SQL scriptur; ORM-byggð migration óheimil	SKAL
SQL skipulag	/SQL mappa, númerað eftir keyrsluröð, byggjanlegt frá grunni	SKAL
Eyðingarbann	Hard delete óheimil nema lagaheimild; soft delete er meginregla (lög nr. 77/2014)	SKAL
Gagnaeignarhald	Verkkaupi á öll gögn; eignarhald færast aldrei til verksala eða þjónustuaðila	SKAL
Gagnaflýttjanleiki	Gögn á stöðluðu sniði, auðflýttjanleg án sérstaks samstarfs við verksala; útflutningur án aukakostnaðar innifalinn í samningsverði	SKAL
Eyðingarbann	Hard delete óheimil nema lagaheimild (lög nr. 77/2014); soft delete er meginregla; eyðing skal vera rekjanleg	SKAL

20.14 Útgáfufurli og keyrslustýring

Flokkur	Krafa	Tegund
Stöðuleysi (T)	Lausnir mega ekki viðhalda innri stöðu milli beiðna	SKAL
Endurræsingapól	Þola endurræingu tilvika án gagnataps	SKAL
Engin handvirk skref	Engar handvirkar aðgerðir við útgáfu eða ræingu	SKAL
Logging	stdout/stderr; skrif í skrár óheimilt; JSON/ECS	SKAL
Metrics	Tæknilegar mælingar á stöðluðu sniði	SKAL

Stillingar	Úr keyrsluumhverfi; engar harðkóðaðar stillingar	SKAL
Docker (T)	Dockerfile í rót; ein eining per keyrsluhluta; ekki root; engin leyndarmál	SKAL
Docker build	Build/compile ekki í Docker; einungis pökkun forsmíðaðs kóða	SKAL
Kubernetes (T)	Allar lausnir keyrsluhæfar í Kubernetes	SKAL
GitOps	Samhæft GitOps útgáfumódeli; innviðakóði í .kube/ möppu	SKAL
Liveness probe	/system/alive	SKAL
Readiness probe	/system/ready	SKAL
Leyndarmál	Í Vault; aldrei í kóða, Docker einingum eða stillingarskrám	SKAL
Disk-skrif	Óheimilt í runtime nema tímabundið (/tmp)	SKAL
Beinar breytingar	Verksali má ekki gera beinar breytingar í Kubernetes cluster	SKAL

20.15 Rekstur og eftirlit

Flokkur	Krafa	Tegund
Staðlaðir pakkar	Verksali skyldugur að nota staðlaðar forritunareiningar verkkaupa	SKAL
Prometheus	Mæligögn á Prometheus sniði á /system/metrics	SKAL
Metrics aðgengi	/system/metrics ekki aðgengilegt af internetinu	SKAL
/version	Endapunktur aðgengilegur af internetinu; commitSha, workflowRunId, version	SKAL
Útgáfuuppl.	Fastsett sjálfvirk í CI/CD; handvirk yfirskrift óheimil	SKAL
Logging staðall	Structured logging á JSON/ECS; staðlaður pakki verkkaupa	SKAL
Production stig	WARNING og hærra; nothæfir loggar til bilanagreiningar	SKAL
Rekjanleiki	trace.id, span.id, x-correlation-id skylda	SKAL
Áframsending	Rekjanleikahausar áfram sendir milli þjónusta	SKAL
Bann í loggum	Viðkvæm gögn (kennitölur, lykilorð, API-lyklar) aldrei í loggum	SKAL
Logskrár á disk	Óheimilt í Kubernetes umhverfi	SKAL
ECS viðbótarsvæði	Stöðluð ECS-samhæfð viðbótarsvæði (labels.application, x-correlation-id, x-road-client) þar sem við á	SKAL

20.16 Lækningatæki

Flokkur	Krafa	Tegund
Tilgangsmat	Mat á hvort hugbúnaður telst lækningatæki (sbr. MDCG 2019-11) skal fara fram við verkupphaf og vera skjalfest í hönnunargögnum	SKAL
Endurmat	Endurmat á flokkun skal fara fram ef tilgangslýsing hugbúnaðar breytist	SKAL
Skjölun mats	Niðurstaða flokkunarmats skal geymd sem hluti af verkefnisgögnum óháð niðurstöðu	SKAL
Gæðakerfi	Ef hugbúnaður telst lækningatæki: framleiðandi skal starfa eftir ISO 13485 gæðastjórnunarkerfi	SKAL
Hugbúnaðarlíftími	Ef hugbúnaður telst lækningatæki: þróun og viðhald skv. IEC 62304	SKAL
Áhættustýring	Ef hugbúnaður telst lækningatæki: formleg áhættustýring skv. ISO 14971	SKAL
CE-merking	Lækningatæki skulu vera CE-merkt áður en þau eru tekin í notkun	SKAL
Tilkynningarskylda	Tilkynning til Lyfjastofnunar um atvik skv. 35. gr. laga nr. 132/2020	SKAL
Upplýsingaskylda	Verksali skal upplýsa verkkaupa ef hann telur hugbúnað gæti fallið undir skilgreiningu lækningatækis	GETUR
IEC 62304 samræmi	Ef lækningatæki: verksali skal sýna fram á samræmi við IEC 62304 hvað varðar skjölun, breytingastýringu og rekjanleika	SKAL
SBOM	Ef lækningatæki: Software Bill of Materials (SBOM) skal afhent sem hluti af tækniskjölun	SKAL

21 Viðauki: Undanþáguferill frá stefnu verkkaupa

Tilgangur undanþáguferils er að:

- tryggja samræmi og fyrirsjáanleika í tæknivali,

- koma í veg fyrir ósamræmi milli lausna,
- halda stjórn á langtímaviðhaldi og rekstrarkostnaði,
- leyfa frávik einungis þegar þau eru nauðsynleg og rökstudd.

Undanþágur eru undantekning, ekki valkostur. Samþykki á undanþágu í einu verkefni skapar ekkert fordæmi fyrir önnur verkefni.

21.1 Hvenær þarf undanþágu?

Undanþáguferil skal beita ef óskað er eftir frávikum frá þeim aðferðum sem lýst er í tæknistefnu verkkaupa eða þróunarhandbók verkkaupa. Dæmi:

Frávik	Kaflí sem frávikið snertir
Notkun SPA eða JavaScript/TypeScript-miðaðan framenda sem uppfyllir ekki skilyrði Leiðar B	Tæknistefna §5, þróunarhandbók kaflí 8
Víkja frá ASP.NET + SSR stefnu (Leið A) eða frá skilyrðum Leiðar B	Tæknistefna §5.1, þróunarhandbók kaflí 8
Nota annað forritunarmál en C# í bakenda	Tæknistefna §5.2, þróunarhandbók kaflí 8
Nota annan gagnagrunn en MS SQL Server	Tæknistefna §6.1, þróunarhandbók kaflí 17
Nota Entity Framework eða annað full-ORM	Tæknistefna §6.1, þróunarhandbók kaflí 17
Þróa native snjallforrit	Þróunarhandbók kaflí 8
Nota CMS kerfi	Þróunarhandbók kaflí 8
Nota þriðja aðila AI agent framework	Tæknistefna §10.3, þróunarhandbók kaflí 11
Innleiða tæknistafla sem ekki er skilgreindur sem samþykktur	Tæknistefna §5.2, þróunarhandbók kaflí 8
Nota Angular, Vue, Svelte eða annað framework en React í framenda	Þróunarhandbók kaflí 8
Nota Next.js, Remix, Gatsby, Astro eða annað meta-framework	Þróunarhandbók kaflí 8
Nota ytri stöðustjórnunarsöfn (Redux, Zustand o.fl.) í Leið B verkefni	Þróunarhandbók kaflí 8
Nota Node.js í rekstrarumhverfi	Tæknistefna §5, þróunarhandbók kaflí 8
Handskrifa framendakóða í Leið B verkefni	Þróunarhandbók kaflí 8

Athugasemd: Leið B (gervigreindarsmiðaður React framendi samkvæmt kafla 8.2.2) krefst ekki undanþágu ef öll skilyrði Leiðar B eru uppfyllt. Hins vegar krefst hvers konar frávik frá skilyrðum Leiðar B formlegrar undanþágu samkvæmt þessum ferli, þar á meðal frávik frá AI-smíðakröfu, frá banni við ytri stöðustjórnunarsöfnum, eða frá banni við Node.js í keyrslu.

21.2 Kröfur til undanþágubeiðni

Undanþágubeiðni skal vera skrifleg og innihalda að lágmarki:

Krafa	Innihald
Lýsing á fráviki	Hvaða tækni er óskað eftir að nota, hvaða kafla tæknistefnu eða þróunarhandbókar frávikið snertir
Rökstuðningur	Hvers vegna samþykktur tæknistaflí er ekki nægjanlegur, hvaða virkni er ómöguleg eða óraunhæf með samþykktum aðferðum. Almenn ósk um aðra tækni er ekki nægjanlegur rökstuðningur.
Langtímatímat	Áhrif á viðhald næstu 10–20 ár, aðgengi að sérhæfðu tæknifólki, áhætta vegna úreltingar tækninnar
Heildarkostnaður (TCO)	Mat á heildarkostnaði frávíks yfir áætlaðan líftíma lausnar, þar á meðal viðhaldskostnaður, leyfiskostnaður og kostnaður við þekkingaröflun
Áhrifamat	Mat á áhrifum frávíks á rekstur, öryggi, samvirkni og aðgengi
Gæðatrygging	Hvernig aðgengi, öryggi, prófanir og skjölun verða tryggð, hvernig lausnin fellur að CI/CD og rekstri verkkaupa

21.3 Samþykktarferli

Krafa	Lýsing
Samþykktaraðili	Undanþága telst einungis gild ef hún er samþykkt skriflega af tæknilegum sérfræðingum verkkaupa og verkefnastjórn verkkaupa
Tímasetning	Samþykki þarf að liggja fyrir áður en þróun hefst. Munnleg samþykki, óformleg umræða eða samþykki verktaka telst ekki gilt.
Tímabinding	Allar undanþágur skulu vera tímabundnar þar sem mögulegt er. Gildistími skal vera skráður í samþykkt.
Umfang	Undanþága gildir einungis fyrir það verkefni og þá virkni sem tilgreind er í beiðni. Samþykki skapar ekkert fordæmi fyrir önnur verkefni eða framtíðarbeiðnir.
Skjalfesting	Samþykkt undanþága skal skráð í miðlæga skrá yfir virkar undanþágur sem verkkaupi heldur utan um

21.4 Ábyrgð

Krafa	Lýsing
Kostnaðarábyrgð	Ef frávik er samþykkt ber verksali fulla ábyrgð á auknum viðhaldskostnaði, þekkingaröflun, skjölun og yfirtöku
Sömu gæðakröfur	Lausn sem byggir á undanþágu fellur undir sömu gæðakröfur og önnur lausn, undanþága frá tæknistafla er ekki undanþága frá gæðum, öryggi eða prófunum
Rekjanleiki	Verksali ber ábyrgð á að skjalfesta frávik í arkitektúrakvörðunum (ADR) og README

21.5 Endurskoðun og endurmat

Krafa	Lýsing
Reglubundin endurskoðun	Verkkaupi endurskoðar allar virkar undanþágur að lágmarki einu sinni á ári
Endurmat	Verkkaupi áskilur sér rétt til að endurmeta undanþágu hvenær sem er og gera kröfu um að lausn verði færð aftur að samþykkttri stefnu við meiriháttar endurvinnslu
Útrunnin undanþága	Þegar gildistími undanþágu rennur út skal verksali leggja fram nýja beiðni ef frávik er enn nauðsynlegt. Útrunnin undanþága veitir enga heimild til áframhaldandi fráviks.
Tilkynning	Verkkaupi skal tilkynna verksala skriflega ef undanþága er afturkölluð eða ekki endurnýjuð, með hæfilegum fyrirvara til að áætla aðlögun

22 Viðauki: Yfirferðarskjál kóðarýni

Þessi viðauki lýsir sniði, flokkunarkerfi og vinnulagi yfirferðarskjals sem verkkaupi notar við formlega kóðarýni á breytingabeiðnum (Pull Requests).

Yfirferðarskjalið er aðalsamskiptatæki milli verkkaupa og verksala í viðtökufurli hugbúnaðarafurða. Það er búið til af verkkaupa (með aðstoð gervigreindarlausna og/eða sérfræðinga) og afhent verksala til svörunar og lagfæringa.

Verksali ber ábyrgð á að svara öllum athugasemdum í skjalinu, hvort sem um er að ræða lagfæringu, rökstuðning eða beiðni um nánari útskýringu.

22.1 Snið yfirferðarskjals

Yfirferðarskjalið er afhent sem töflureikniskjal (t.d. Excel) með einum lið per röð. Hver liður samsvarar einni athugasemd sem kom fram við kóðarýni.

Skjalið skiptist í fjóra meginhluta:

22.1.1 Auðkenning og flokkun

Dálkur	Lýsing	Útfyllt af
Nr	Raðnúmer athugasemdar	Verkkaupi

Staður	Nákvæm staðsetning í kóðagrunni: skráarheiti, línunúmer og/eða mappa	Verkkaupi
Flokkun	Tegund athugasemdar eftir uppruna og eðli (sjá flokkunarkerfi)	Verkkaupi
Tegund	Alvarleikastig (sjá alvarleikastig)	Verkkaupi
Þarf fyrir útgáfu?	Hvort lagfæring sé forsenda þess að útgáfa fái samþykkt	Verkkaupi
Flækjustig	Mat á umfangi lagfæringar: Small, Medium eða Large	Verkkaupi

22.1.2 Athugasemdir og tillögur verkkaupa

Dálkur	Lýsing	Útfyllt af
Athugasemd frá verkkaupa	Lýsing á atriði sem fundist hefur, með tilvísun í staðsetningu og áhrif	Verkkaupi
Aðgerðir verkkaupa	Aðgerðir sem verkkaupi hefur þegar framkvæmt tengdar athugasemdinni, ef við á	Verkkaupi
Tillögur frá verkkaupa	Tillaga að lagfæringu eða lausn, í sumum tilvikum útbúin af gervigreindarlausn	Verkkaupi

22.1.3 Svörun og samskipti

Dálkur	Lýsing	Útfyllt af
Svar frá verksala	Svörun verksala við athugasemd. Getur verið staðfesting á lagfæringu, rökstuðningur fyrir afstöðu, eða beiðni um nánari útskýringu	Verksali
Athugasemdir	Frekari umræða eða samskipti milli aðila ef þörf krefur	Báðir
Ákvarðanir	Lokaákvörðun um hvern lið, hvort athugasemd sé lagfærð, frestað, hafnað eða samþykkt með fyrirvara	Verkkaupi

22.1.4 Tímamat og lagfæringartillögur

Dálkur	Lýsing	Útfyllt af
Áætlun á tímamati (AI)	Sjálfvirk mat gervigreindar á fjölda klukkustunda sem lagfæring krefst. Matið er leiðbeinandi, ekki bindandi.	Verkkaupi (sjálfvirk)
Tímamat frá verksala	Mat verksala á tíma sem lagfæring krefst, í klukkustundum	Verksali
Lagfæringartillaga (AI)	Tengill á tillögu að lagfæringu (t.d. Pull Request búin til af gervigreindarlausn) eða lýsing á fyrirhugaðri aðgerð	Verkkaupi (sjálfvirk)

22.2 Flokkunarkerfi

Athugasemdir eru flokkaðar eftir uppruna og eðli í þrjá meginflokka:

Flokkun	Lýsing	Dæmi
Org-Policy	Frávik frá bindandi stefnu, stöðlum eða kröfum verkkaupa. Nær til þróunarhandbókar, tæknistefnu, nafnareglna, skipulags kóðageymsla og annarra skjalfestra reglna.	Vantar skylduendapunkt (/system/alive); nafnareglur ekki uppfylltar; skylduverflæði vantar í CI
Security-Standard	Öryggistengd atriði sem varða meðhöndlun gagna, auðkenninga, dulkóðunar, aðgangsstýringar eða aðra öryggisþætti.	Viðkvæm gögn í lögum (logs); harðkóðaðar keyrslubreytur; leyndarmál í stillingarskrám
Best-Practice	Atriði sem tengjast viðurkenndum starfsvenjum, kóðagæðum, viðhaldshæfni eða langtíma heilsu lausnar. Þessi flokkur nær til atriða sem eru ekki brot á bindandi reglum en geta haft neikvæð áhrif á gæði og viðhald lausnar til langs tíma.	Notkun úreltra forritunarsafna; óþarfa flækjustig; brot á type safety; ósamræmd nafnagjöf

22.3 Alvarleikastig

Alvarleikastig ákvarðar hversu brýnt er að bregðast við athugasemd og hvaða áhrif hún hefur á samþykkt útgáfu:

Alvarleiki	Lýsing	Áhrif á samþykkt
------------	--------	------------------

Critical	Frávik sem kemur í veg fyrir útgáfu eða brýtur bindandi kröfu verkkaupa. Getur einnig átt við um alvarlega öryggisáhættu.	Breytingabeiðni hafnað þar til lagfært. Útgáfa ekki möguleg.
High	Verulegur galli eða frávik sem þarf að lagfæra. Getur varðað gæði, öryggi eða samræmi við staðla.	Lagfæring krafist. Tímasetning og forgangur samið um milli aðila.
Medium	Atriði sem æskilegt er að lagfæra en kemur ekki í veg fyrir útgáfu. Getur varðað langtíma viðhald, læsileika eða samræmi.	Lagfæring mælt með. Ákvörðun tekin í samráði.
Low	Minniháttar athugasemd eða tillaga að úrbót. Getur verið ábending um betra vinnulag eða minni lagfæringar.	Til skoðunar hjá verksala. Ekki forsenda samþykktar.

22.4 Vinnulag við yfirferð

Eftirfarandi skref lýsa vinnulagi milli verkkaupa og verksala þegar yfirferðarskjal er notað:

1. **Verkkaupi framkvæmir kóðarýni** og skráir allar athugasemdir í yfirferðarskjalið. Rýni er framkvæmd af sérfræðingum og/eða gervigreindarlausnum verkkaupa.
2. **Verkkaupi sendir yfirferðarskjal til verksala** ásamt samantekt á helstu atriðum og upplýsingum um tímasetningu endurgjafar.
3. **Verksali fer yfir allar athugasemdir** og fyllir út:
 - **Svar frá verksala**, lagfæring framkvæmd, rökstuðningur fyrir óbreyttri útfærslu, eða beiðni um nánari skýringu.
 - **Tímamat frá verksala**, mat verksala á tíma sem lagfæring krefst.
4. **Fundur er haldinn milli aðila** þar sem farið er yfir athugasemdir, sérstaklega þar sem ágreiningur er um lagfæringar, alvarleikastig eða tímasetningu.
5. **Verkkaupi skráir ákvarðanir** í dálkinn „Ákvarðanir“ fyrir hvern lið.
6. **Verksali framkvæmir umsamdar lagfæringar** og uppfærir breytingabeiðni.
7. **Verkkaupi staðfestir lagfæringar** og samþykkir eða hafnar breytingabeiðni.

Verksali ber ábyrgð á að svara öllum athugasemdum, óháð alvarleikastigi. Athugasemdir sem verksali telur rangar eða óviðeigandi skulu rökstuddar skriflega í svæðinu „Svar frá verksala.“ Verkkaupi tekur lokaákvörðun.

22.5 Gervigreindarstuðningur í kóðarýni

Verkkaupi nýtir gervigreindarlausnir sem hluta af kóðarýni. Gervigreindin aðstoðar við:

- sjálfvirka greiningu á kóðagæðum, öryggisatriðum og samræmi við handbækur,
- sjálfvirkt tímamat á lagfæringum (dálkurinn „Áætlun á tímamati (AI)“),
- tillögur að lagfæringum sem geta verið afhentar sem Pull Request eða lýsing (dálkurinn „Lagfæringartillaga (AI)“).

Athugið:

- AI-tímamat er **leiðbeinandi**, ekki bindandi. Mat verksala (í dálkinum „Tímamat frá verksala“) er notað við áætlanagerð.
- AI-lagfæringartillögur eru **tillögur**, ekki samþykktar lausnir. Verksali ber ábyrgð á að meta og staðfesta allar lagfæringar áður en þær eru innleiddar.
- Verkkaupi áskilur sér rétt til að uppfæra og betrumbæta gervigreindarverkfæri og kvaðningar (prompts) á meðan á verkefni stendur.

22.6 Dæmi úr yfirferðarskjali

Eftirfarandi dæmi sýna þrjár tegundir athugasemda sem gætu komið fram í yfirferðarskjali. Dæmin eru sýnd til skýringar á sniði og innihaldi; þau endurspeгла ekki raunverulegt verkefni.

22.6.1 Dæmi 1: Org-Policy / Critical

Svæði	Gildi
Nr	1
Staður	.github/workflows/
Flokkun	Org-Policy
Tegund	Critical
þarf fyrir útgáfu?	Já
Flækjustig	Small
Athugasemd frá verkkaupa	Skylduverkflæðisskrá vantar í kóðageymslu. Allar kóðageymslur verkkaupa skulu innihalda þetta sjálfvirka verkflæði fyrir kóðarýni. Sniðmát er aðgengilegt í lausnarbanka.
Tillögur frá verkkaupa	Bæta við verkflæðisskrá samkvæmt sniðmáti úr lausnarbanka.
Áætlun á tímamati (AI)	0,5 klst

22.6.2 Dæmi 2: Security-Standard / Critical

Svæði	Gildi
Nr	2
Staður	Dockerfile:36
Flokkun	Security-Standard
Tegund	Critical
þarf fyrir útgáfu?	Já
Flækjustig	Small
Athugasemd frá verkkaupa	Keyrslubreyta sem stýrir hegðun þróunarumhverfis er harðkóðuð í Docker keyrslueiningu. Þetta veldur því að þróunarviðmót birtist í rekstrarumhverfi.
Tillögur frá verkkaupa	Fjarlægja harðkóðaða keyrslubreytu úr keyrslueiningu. Nota byggingartímabreytu (ARG) eða láta leyndarmálageymslu/CI sjá um gildið. Staðfesta að byggingatímagildi sé yfirskrifanlegt.
Áætlun á tímamati (AI)	1 klst

22.6.3 Dæmi 3: Best-Practice / High

Svæði	Gildi
Nr	3
Staður	Víða í kóðagrunni (30+ staðir)
Flokkun	Best-Practice
Tegund	High
þarf fyrir útgáfu?	Nei
Flækjustig	Medium
Athugasemd frá verkkaupa	Viðtæk notkun á as any í villumeðhöndlun sem skringilegar forritunarmáls öryggiskerfi (type safety). Fundist í fjölda eininga. Dregur úr gæðum og viðhaldshæfni lausnar til langs tíma.
Tillögur frá verkkaupa	Útfæra hjálparfall með réttri týpu til að vinna úr villuskilaboðum. Skipta út öllum as any tilvikum fyrir þetta hjálparfall.
Áætlun á tímamati (AI)	4 klst

22.7 Varðveisla og rekjanleiki

Öll yfirferðarskjöl skulu varðveitt sem hluti af verkefnisgögnum og tengd viðkomandi breytingabeidnum til að tryggja rekjanleika.

Yfirferðarskjöl eru hluti af gæðasögu verkefnis og skulu vera aðgengileg til endurskoðunar ef ágreiningur rís um afhendingu eða gæði hugbúnaðar.

23 Viðauki: Gátlistar við verkefnastýringu

23.1 Gátlisti 1: Áður en forritun hefst

23.1.1 Lög, heimildir og forsendur

Atriði	Ábyrgð
Tilvist og tilgangur hugbúnaðar er studdur lögum, reglum eða fyrirmælum og skjalfestur	Verkkaupi
Lög og reglur sem hafa áhrif á lausnina eru auðkennd og skjalfestar í README eða /Docs (þ.m.t. lög nr. 90/2018, 77/2014, 78/2019 og lög um sjúkraskrár eftir því sem við á)	Báðir
Gagnavinnslusamningur til staðar ef þriðji aðili vinnur persónuupplýsingar	Verkkaupi
Undanþágubeiðnir sendar og samþykktar ef frávik frá tæknistefnu eða þróunarhandbók	Verkkaupi
Mat á flokkun hugbúnaðar sem lækningatæki (sbr. MDCG 2019-11) framkvæmt og skjalfest	Verkkaupi
Ef hugbúnaður telst lækningatæki: viðeigandi gæðakerfi (ISO 13485), hugbúnaðarlífímaferlar (IEC 62304) og áhættustýring (ISO 14971) til staðar eða á áætlun	Verkkaupi
Ef hugbúnaður telst lækningatæki: samræmismatsleið (CE-merking) og tengiliður við tilkynntan aðila (Notified Body) skilgreind ef flokkur ≥ IIa	Verkkaupi

23.1.2 Hönnun og arkitektúr

Atriði	Ábyrgð
Yfirlitsmynd af arkitektúr lausnar liggur fyrir og er samþykkt	Verkkaupi
Högun og meginákvarðanir (ADR) eru skjalfestar	Verksali
Forritunarmál og framework eru samþykkt (C#, .NET, ASP.NET)	Verkkaupi
Framendaáhersla er server-side rendering; engar SPA lausnir	Verkkaupi
Auðkenningarleið valin og samþykkt	Verkkaupi
Úttektaráætlun (exit plan) liggur fyrir í drögum	Verksali

23.1.3 Gagnagrunnar og gögn

Atriði	Ábyrgð
Ákveðið hvaða gagnagrunnur er notaður (MS SQL nema undanþága)	Verkkaupi
Gagnagrunnsskipanir eru skilgreindar í SQL (ekki ORM)	Verksali
HL7/FHIR notkun skilgreind ef við á	Báðir
Eyðingarstefna skilgreind (soft delete sem meginregla, lög nr. 77/2014)	Báðir
Gagnaeignarhald staðfest, verkkaupi á öll gögn	Verkkaupi

23.1.4 GitHub og skipulag

Atriði	Ábyrgð
Fjöldi og nöfn kóðageymsla skilgreind samkvæmt nafnareglum	Verkkaupi
Kóðageymslur stofnaðar af verkkaupa og grunnuppsetning klár	Verkkaupi
.agent/ mappa sett upp í kóðageymslum	Verkkaupi
Skilatiðni og fyrirkomulag skilgreint og aðilar með aðgang	Báðir

23.1.5 Öryggi

Atriði	Ábyrgð
Leyndarmálastjórnun skilgreind (Vault/leyndarmálageymsla)	Verkkaupi
OWASP Top 10 og CWE viðmið kynnt teymi	Verksali
Kröfur um dulkóðun í flutningi og geymslu staðfestar	Báðir

23.1.6 Gervigreindarstudd þróun

Atriði	Ábyrgð
Áætlunarskjöl (Áfangi 1) liggja fyrir í -docs kóðageymslu	Verksali

Framkvæmdaáætlun (Áfangi 2) með afritunarhæfum vinnulotu kvaðningum liggur fyrir	Verksali
Framkvæmdarröð (dependency graph) og MVP-merkingar eru í framkvæmdaáætlun	Verksali
Rýni tæknilegs ábyrgðaraðila SH er lokið og samþykki skráð í /docs/execution/00-approval-log.md	Verkkaupi
.agent/ mappa er til staðar og uppfærð	Verksali

23.2 Gátlisti 2: Meðan á forritun stendur

23.2.1 Kóðauppbygging

Atriði	Ábyrgð
Möppuskipulag fylgir þróunarhandbók	Verksali
Engar óheimilar tæknilausnir (SPA, JS burðarás, ORM o.fl.)	Verksali
.agent/ mappa viðhaldið og auðguð jafnóðum	Verksali

23.2.2 Gæðastýring

Atriði	Ábyrgð
Sjálfvirk kóðagreining virk í þróunarumhverfum	Verksali
Viðvaranir meðhöndlaðar sem villur (treat warnings as errors)	Verksali
Undanþágur frá reglum skjalfestar og samþykktar	Verkkaupi

23.2.3 Prófanir

Atriði	Ábyrgð
Einingapróf ≥80% fyrir nýjan kóða, ≥65% heild	Verksali
Samþættingarpróf fyrir alla ytri endapunkta	Verksali
Aðgengisprófanir virkar í CI	Verksali

23.2.4 Sjálfvirknivæðing

Atriði	Ábyrgð
CI ferlar skilgreindir í .github/	Báðir
Gæðahlið tengd og virk	Verkkaupi

23.2.5 Öryggi meðan á þróun stendur

Atriði	Ábyrgð
Engin heilbrigðisgögn eða kóði verkkaupa sendur í opinberar AI þjónustur	Verksali
Engin leyndarmál í kóða, stillingarskrám eða Docker einingum	Verksali
Engin raunveruleg heilbrigðisgögn í þróunar- eða prófunarumhverfi nema skriflegt leyfi	Verksali

23.3 Gátlisti 3: Fyrir skil og QA

23.3.1 Pull Requests

Atriði	Ábyrgð
PR heitir sama nafni og kóðagrein	Verksali
PR inniheldur prófunarniðurstöður og samþykktir	Verksali
Allir sjálfvirkir ferlar standast	Verksali
PR inniheldur einungis breytingar tengdar skilgreindu verki	Verksali

23.3.2 Kóðarýni

Atriði	Ábyrgð
Kóðarýni farið fram af verkkaupa	Verkkaupi
Formlegt yfirferðarskjal sent til verksala	Verkkaupi
Verksali hefur svarað öllum athugasemdum í yfirferðarskjali	Verksali
Engar Critical athugasemdir óleystar	Báðir

23.3.3 Skjölun við skil

Atriði	Ábyrgð
--------	--------

README.md uppfært	Verksali
/Docs möppu innihald uppfært	Verksali
.agent/ mappa uppfærð	Verksali
Úttektaráætlun (exit plan) uppfærð ef verulegar breytingar	Verksali
Gagnaútflutningsleiðbeiningar uppfærðar ef gagnagrunnur breyttist	Verksali

23.4 Gátlisti 4: Fyrir útgáfufarla

23.4.1 Almennt

Atriði	Ábyrgð
Útgáfur aðeins af main með qa/release tags	Verkkaupi
CalVer útgáfunúmer notuð	Verkkaupi
Útgáfa fylgir útgáfudagatali (engin frysting virk)	Verkkaupi

23.4.2 Docker

Atriði	Ábyrgð
Dockerfile í rót kóðageymslu	Verksali
Docker image byggir á forsmíðuðum artefaktum (ekki build í image)	Verksali
Engin leyndarmál í Docker einingu	Verksali
Docker image keyrir ekki sem root	Verksali

23.4.3 Kubernetes

Atriði	Ábyrgð
.kube/ mappa til staðar og samræmd Argo CD	Verksali
/system/alive og /system/ready útfærð	Verksali
Heilsuathuganir (liveness/readiness probes) tengdar í Kubernetes	Verksali

23.4.4 Gagnagrunnsbreytingar

Atriði	Ábyrgð
SQL-skipanir afhentar og samþykktar	Verksali
Backward-compatibility staðfest (engar brotabreytingar á skema)	Verksali
Afrit tekið af gagnagrunni áður en breytingar eru keyrðar	Verkkaupi

23.5 Gátlisti 5: Fyrir keyrslu, mælingar og logging

23.5.1 Mælingar

Atriði	Ábyrgð
/system/metrics útfærður	Verksali
Prometheus samhæfð mæligögn	Verksali
/system/metrics ekki aðgengilegt af internetinu	Verkkaupi

23.5.2 Keyrslustaða

Atriði	Ábyrgð
/system/alive og /system/ready endapunktur útfærðir	Verksali
Tengdir sjálfvirkri endurræsingu í Kubernetes	Verkkaupi

23.5.3 Útgáfuupplýsingar

Atriði	Ábyrgð
/version endapunktur skilar commitSha, workflowRunId og version	Verksali
Útgáfuupplýsingar fastasettar sjálfvirk í CI/CD (handvirk yfirskrift óheimil)	Verksali

23.5.4 Logging

Atriði	Ábyrgð
Structured logging á JSON/ECS sniði	Verksali
Rekjanleiki (trace.id, span.id, x-correlation-id) tryggður	Verksali
Notaðar staðlaðar logging pakkningar verkkaupa	Verksali

ECS viðbótarsvæði (labels.application, x-correlation-id, x-road-client)	Verksali
Engin viðkvæm gögn í loggum (kennitölur, lykilorð, API-lyklar)	Verksali
Engar logskrár á disk í Kubernetes umhverfi	Verksali

23.6 Gátlisti 6: Við verklok (tæknileg atriði)

Athugið: Þessi gátlisti vísar einungis í tækniatriði. Formlegan gátlista fyrir verklok er að finna í útboðsgögnum og hjá verkefnastjórn.

23.6.1 Skjölun

Atriði	Ábyrgð
README í rót og /Docs uppfært	Verksali
ADR, arkitektúrmyndir og sequence rit fylgja	Verksali
CronJob skjölun uppfærð (ef við á)	Verksali
Úttektaráætlun (exit plan) uppfærð og endanleg	Verksali
Gagnaútflutningsleiðbeiningar endanlegar	Verksali

23.6.2 Afhending

Atriði	Ábyrgð
Allur kóði í kóðageymslum verkkaupa	Verksali
Öllum afritum kóða verkkaupa eytt hjá verksala	Verksali
.agent/ mappa endanleg og uppfærð	Verksali
Öll leyndarmál og aðgangsupplýsingar verksala afturkallaðar	Verkkaupi

23.6.3 Staðfesting

Atriði	Ábyrgð
Gagnaeignarhald staðfest, verkkaupi hefur fullan aðgang að öllum gögnum	Verkkaupi
Lausn keyranleg og reksturshæf án aðkomu verksala	Verkkaupi
Engar virkar undanþágur sem krefjast sérstakrar meðhöndlunar eftir verklok	Báðir

24 Viðauki: Þjónustur og verkfæri verkkaupa

Þessi viðauki listar þær þjónustur og verkfæri sem verkkaupi nýtir við gæðaeftirlit, útgáfu, rekstur og eftirlit hugbúnaðarlausna. Viðaukinn er viðhaldið sjálfstætt og uppfærður eftir þörfum án þess að breyta meginmáli handbókarinnar.

Meginmál þróunarhandbókar vísar til þessara þjónusta með almennum hugtökum (t.d. „sjálfvirk kóðagreining“, „miðlæg kóðageymsla“, „leyndarmálageymsla“). Þessi viðauki tengir þau hugtök við raunverulegar þjónustur og verkfæri.

24.1 Staða og bindandi gildi

Verksali er skyldugur til að tryggja að afhent lausn virki innan þeirra kerfa sem verkkaupi rekur og styðji samþættingu við þau, samkvæmt skilgreindum sniðum, ferlum og aðgangi sem verkkaupi útvegar.

Verksala er ekki skylt að nota sömu þjónustur í eigin þróunarumhverfi meðan á vinnu stendur, nema annað sé tekið fram.

24.1.1 Kóðageymslur og útgáfa

Þjónusta	Hlutverk	Krafa við afhendingu
GitHub	Miðlæg kóðageymsla verkkaupa	SKAL skila kóða í kóðageymslu verkkaupa
GitHub Actions	CI/CD og gæðahlið	SKAL styðja og keyra sjálfvirka ferla verkkaupa

24.1.2 Gæðaeftirlit og kóðagreining

Þjónusta	Hlutverk	Krafa við afhendingu
----------	----------	----------------------

SonarCloud	Sjálfvirkar gæðaprófanir og kóðagreining; inniheldur gæðahlið sem stöðvar ferli ef kröfur eru ekki uppfylltar	SKAL standast skilgreind gæðaviðmið
StyleCop / Roslyn Analyzers	Kóðastílsgreining og samræming fyrir C#	SKAL vera hluti af lausn

24.1.3 Gervigreind í gæðaeftirliti

Þjónusta	Hlutverk	Krafa við afhendingu
Gervigreindarlausnir verkkaupa (t.d. AI kóðarýni)	AI-studd kóðarýni og gæðamat á breytingabeiðnum	GETUR, niðurstöður notaðar af verkkaupa sem hluti af viðtökuferli

Verkkaupi áskilur sér rétt til að uppfæra og betrumbæta gervigreindarverkfæri og kvaðningar (prompts) á meðan á verkefni stendur.

24.1.4 Gervigreindarforritunarsöfn fyrir afurðaðróun

Eftirfarandi forritunarsöfn eru samþykkt til þróunar á gervigreindarlausnum sem verða hluti af keyrsluafurð verkkaupa.

Þjónusta / Safn	Hlutverk	Krafa við afhendingu	Staða
Microsoft Agent Framework	Forritunarsafn til þróunar gervigreindar agenta í .NET/C#	SKAL nota ef gervigreindarvirkni er hluti af afurð	GA (útgáfa 1.0, Apríl 2026)
Microsoft.Extensions.AI	Grunnur fyrir gervigreindarsamskipti í .NET	SKAL nota sem grunnlag fyrir öll LLM-samskipti	GA (útgáfa 10.4.0)

Athugasemd: Gervigreindarforritunarsöfn Microsoft eru á hraðri þróunarbraut. Verkkaupi mun uppfæra þessa töflu eftir þörfum þegar Microsoft breytir heitum eða gefur út nýjar útgáfur. Nafn og útgáfunúmer forritunarsafns sem tilgreint er hér gildir á útgáfudegi þessarar handbókar. Ef vafi leikur á skal hafa samband við tæknilega sérfræðinga verkkaupa.

Meginreglan er óbreytt: Einungis er heimilt að nota opinber Microsoft forritunarsöfn fyrir .NET sem uppfylla skilyrði (bein .NET samþætting, langtímastuðningur, opinbert á NuGet, rekstrarsamhæfni). Þriðja aðila agent-framework lausnir eða sérsníðaðar útfærslur eru óheimilar.

24.1.5 Keyrsluumhverfi

Þjónusta	Hlutverk	Krafa við afhendingu
Kubernetes	Keyrslu- og rekstrarumhverfi	SKAL fylgja uppgefnum Kubernetes sniðum verkkaupa
Argo CD	Sjálfvirk dreifing í Kubernetes	SKAL vera samhæft
Hashicorp Vault	Leyndarmálageymsla í keyrsluumhverfi	SKAL nota fyrir öll keyrsluleyndarmál

24.1.6 Vöktun og rekstrareftirlit

Þjónusta	Hlutverk	Krafa við afhendingu
Coralogix	Miðlæg atvikaskráning (logging) og leit	SKAL senda logga í samþykkt snið
Prometheus	Mælingar og afkastamælikerfi	SKAL birta mæligögn á stöðluðu sniði
Grafana	Mælaborð og rekstraryfirsýn	Notað af verkkaupa; engin bein krafa á verksala

24.1.7 Hönnun og verkefnastýring

Þjónusta	Hlutverk	Krafa við afhendingu
Figma	Hönnun og frumgerðir	GETUR, valfrjálst
Jira	Verkefnastýring	GETUR, valfrjálst
Notion	Skjölun	GETUR, valfrjálst

24.2 Athugasemdir

- Þjónustur merktar **SKAL** eru bindandi kröfur sem afhent lausn verður að uppfylla.
- Þjónustur merktar **GETUR** eru valfrjálstar; verkkaupi nýtir þær sjálfur en gerir ekki kröfu um að verksali nýti þær.
- Þessi viðauki getur breyst yfir líftíma verkefnis ef verkkaupi skiptir um þjónustuveitendur eða verkfæri. Breytingar á SKAL-þjónustum skulu tilkynntar verksala skriflega.

