



MEÐHÖNDLUN VIÐVARANA FRÁ SÍMSTÖÐVUM

Lokaskýrsla

Vor 2011

Davíð Halldór Lúðvíksson

Gunnlaugur Sigurðsson

BSc. í tölvunarfræði

Leiðbeinandi: Birgir Kaldal Kristmannsson
Prófdómari: Hlynur Sigurþórsson

T-404-LOKA
Tölvunarfræðideild

Efnisyfirlit

Inngangur	4
1.1 Lýsing verkefnis	5
2 Skipulag.....	7
2.1 Aðferðafræði	7
2.2 Hlutverk.....	8
2.3 Ítranir	8
2.4 Skipulagsþóker.....	8
2.5 Afkastageta	9
2.6 Dagbók.....	9
2.7 Aðstaða.....	9
2.8 Framtíðarsýn.....	9
2.9 Próunarumhverfi og aðferðir	9
2.10 Samstæðustjórnun.....	10
2.11 Útgáfustjórnun	10
3 Greining.....	11
3.1 Þarfagreining	11
3.2 Áhættugreining	12
4 Hönnun.....	13
5 Forritun.....	15
5.1 Móttakari.....	15
5.2 Notendaviðmót.....	17
6 Prófanir.....	18
6.1 Sjálfvirkar prófanir	18
6.2 Kerfisprófanir	18
6.3 Notendaprófanir	18
7 Framvinda	20
8 Lokaorð	21
8.1 Hvað gekk vel?	21
8.2 Hvað gekk illa?.....	21
8.3 Hvað lærðum við?	22
9 Framtíðin.....	24
10 Ábendingar.....	25
11 Umsagnir.....	26
12 Heimildaskrá	27

Inngangur

Þetta verkefni er lokaverkefni í tölvunarfræði við Háskóla Reykjavíkur og er unnið í samvinnu við Símann. Verkefnið er unnið af Davíð Halldóri Lúðvíkssyni og Gunnlaugur Sigurðssyni.

Verkefnið fólst í að smíða hugbúnað fyrir starfsmenn Símans sem sjá um daglegan rekstur og þá sem starfa við stjórnborð til að vakta viðvaranir og hjartslátt frá símstöðvum.

Hugbúnaðurinn er tvíþættur. Móttakari sem er notaður til að skrá viðvaranir og hjartslátt í gagnagrunn og svo annars vegar vefviðmót sem starfsmenn nota til að fylgjast með viðvörum og hjartslátt. Verkefnið fékk nafnið AOS (e. Alarm Operation System).

Markmið AOS var að leysa af hólmi eldra kerfi sem hefur sama hlutverk og hið nýja kerfi, en er ekki hægt að nota mikið lengur vegna tæknilegra annmarka í samskiptum við símstöðvar.

Viðhald og uppsetning á eldra viðmóti er tímafrekt og flókið, enda um biðlara (e. client side) forrit að ræða. Við uppfærslur þurfti að setja upp nýja útgáfu hjá öllum notendum kerfisins. Vinnuflæðið í eldra viðmótinu er flókið og óþjált. Mikið er af aðgerðum og eiginleikum sem eru óvirkir og ónotaðir.

1.1 Lýsing verkefnis

Síminn hefur yfir að ráða stærsta símstöðvaneti á Íslandi. Undanfarin ár hefur Síminn verið að uppfæra símstöðvar sínar, en því hefur fylgt ný kynslóð símstöðva sem nota aðeins IP (e. Internet Protocol) til að eiga samskipti við önnur kerfi Símans.

Símstöðvarnar senda frá sér ógrynni af gögnum og má þar nefna hjartslátt, viðvaranir og losanir á þeim, upplýsingar um endurræsingar og margt fleira.

Hluti símstöðvanna getur átt samskipt við bæði viðvaranakerfin, meðan elsta símstöðin getur aðeins átt samskipti við núverandi kerfi. Núverandi viðvaranakerfi er 18 ára gamalt og ræður ekki við IP-samskipti. Nýja kerfið hefur ekki samskipti við elstu símstöðina sem notar eingöngu X-25 samskiptastaðal (e. network protocol). Núverandi kerfi mun vakta elstu símstöðina í þann tíma sem hún er í rekstri. Þangað til mun nýtt kerfi fylgjast sérstaklega með viðvörum frá elstu símstöðinni og áframsenda yfir í gagnagrunn nýja kerfisins. Það er ekki hluti af þessu verkefni að útbúa sérstakt forrit til að vakta elstu símstöðina.

Verkefnið varðaði miðlægan móttakara viðvarana og hjartslátt sem hlustar eftir gögnum frá símstöðvum og sér um að vinna úr þeim og skrá í gagnagrunn þar sem þeim er haldið til haga. Að auki var ráðist í að skrifa nýtt notendaviðmót þar sem hægt væri að sjá virkar og óvirkar viðvaranir í símstöðvunum og sýna myndrænt samband við símstöð út frá hjartslætti. Hjartsláttur á að berast á mínútu fresti, en ef það dregst má gera ráð fyrir því að lífsmarki símstöðvarinnar hafi hrakað. Þegar talað eru um hjartslátt símstöðvar er það lífsmark hennar og er það orð notað í nýja viðmótinu.

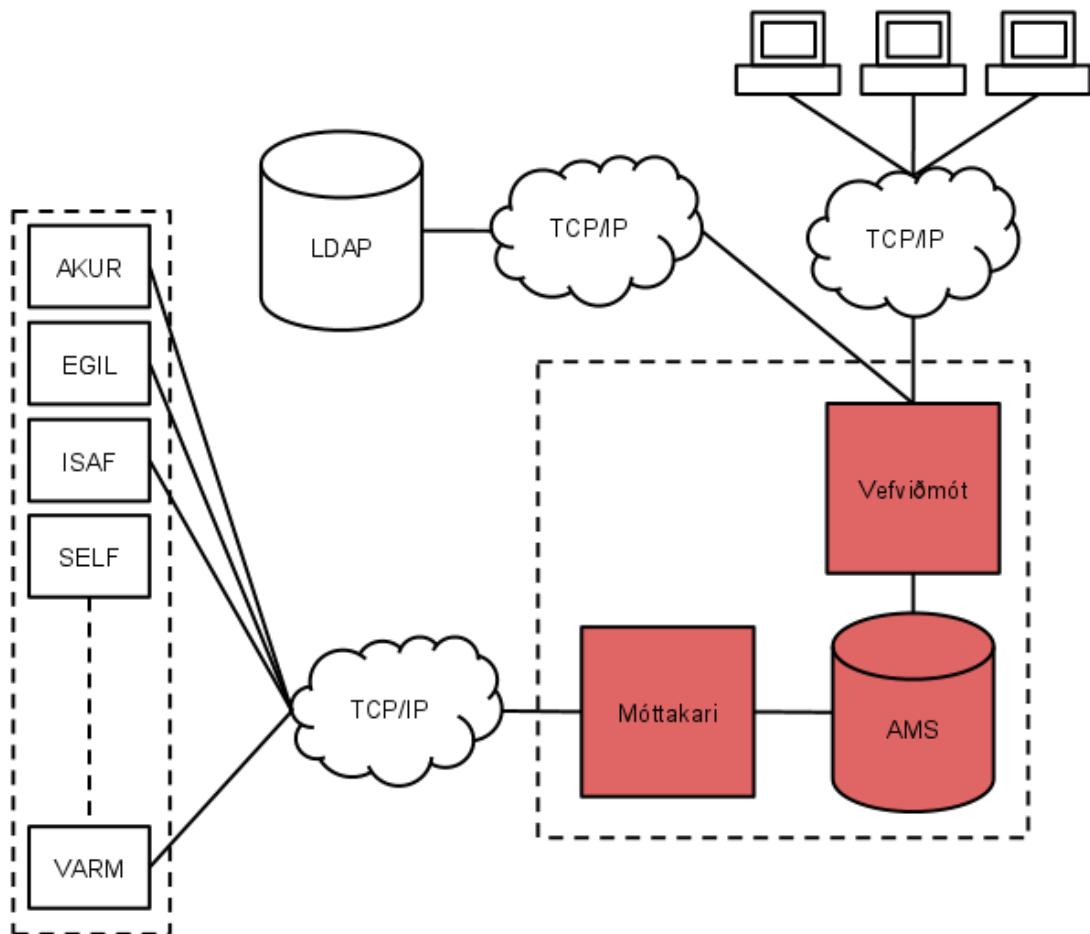
Símstöðvar þarf að stilla sérstaklega svo þær sendi hjartslátt og viðvaranir í nýja kerfið. Viðvörum má skipta í nokkra flokka og klasa sem ákværðar tegund og alvarleika. Sniðið (e. format) á viðvörum er eign Ericsson og er sérhæft fyrir Ericsson-símstöðvar. Sniðið á viðvörum er ekki hægt að birta vegna höfundarréttar Ericsson. Sniðið er mismunandi eftir því hvort um sé að ræða nýja (Ericsson APG-40) eða eldri kynslóð símstöðva (Ericsson IOG-20).

Notendur kerfisins eru starfsmenn stjórnstöðvar, en þeir sjá um vöktun á ýmsum búnaði Símans auk starfsmanna talsímareksturs sem sjá um daglegan rekstur stöðvanna.

Í vefviðmótinu er boðið upp á eftirfarandi virkni:

1. Sjá virkar viðvaranir sem berast sjálfvirkt á forsíðu.
2. Stilla síu sem síar út virkar viðvaranir sem ekki er óskað eftir að birtist á forsíðu.
3. Nákvæm leit að virkum og óvirkum viðvörunum.
4. Að fylgjast með lífsmarki símstöðva.

Á mynd 1 gefur að líta yfirlitsmynd af AOS-kerfinu og hvernig símstöðvar og notendur tengjast því.



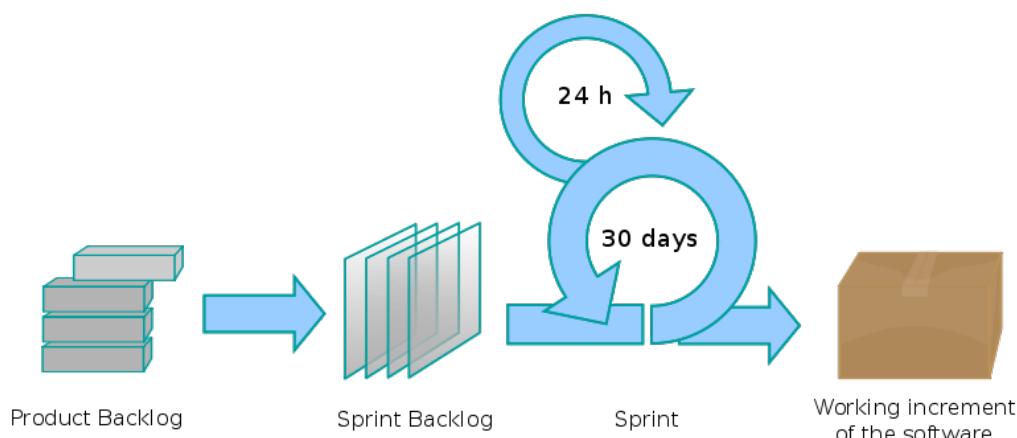
Mynd 1: AOS yfirlitsmynd

2 Skipulag

Þriðjudaginn 18. janúar 2011 var upphafsfundur teymisins. Á honum voru tekna ákvarðanir varðandi helstu atriði verkefnisins.

2.1 Aðferðafræði

Síminn notar Scrum-aðferðafræðina við hugbúnaðarvinnu og því var ákveðið að fylgja þeirri aðferðafræði eins og sést á mynd 2.



Mynd 2: Scrum-hugbúnaðarferli (Scrum Process, 2009)

SCRUM byggir á Agile-hugmyndafræði og er notast við lista (e. product backlog) sem hefur að geyma notendasögur (e. user stories). Sögunum er forgangsraðað og þær metnar. Verkefninu er svo skipt upp í ítranir (e. iteration), oft kallaðar sprettir (e. sprint). Fyrir hvern sprett eru teknar inn sögur sem unnar verða og er haldið utan um þær í verkefnalista (e. sprint backlog) sprettsins. Sögunum er skipt upp í verkliði (e. task) og þeim gefin tímaáætlun áður en hin eiginlega vinna sprettsins hefst. Lengd spretta er ákvörðuð við upphaf verkefnis. Daglegr stöðufundir (e. daily stand-up meeting) eru haldnir þar sem starfsmenn gera grein fyrir því sem gert var daginn áður, hvað stendur til að gera þann dag og hvort einhverjar hindranir séu til staðar sem koma í veg fyrir að hægt sé að halda áfram með verkefnið. Í lok spretts er haldin opin kynning (e. demo) þar sem afurðin (e. working increment of the software) er sýnd og farið yfir hana ásamt því að gera grein fyrir því sem til stendur að gera í næsta spretti. Í lokin er svo komið saman á sérstökum fundi (e. retrospective) þar sem farið er yfir liðinn sprett og rætt um það sem fór vel, hvað fór illa og hvernig sé hægt að bæta úr því.

2.2 Hlutverk

Á áðurnefndum upphafsfundi voru hlutverk einstaklinga ákveðin. Hlutverkin eru sem hér segir:

Eigendur verkefnis: Ágúst Þór Guðmundsson, Jóhannes Hilmisson, Kristmundur Halldórsson, Tómas Árni Jónsson, Porkell Lillie Magnússon og Haukur Þór Bragason.

SCRUM-meistari: Gunnlaugur Sigurðsson.

Teymið: Davíð Halldór Lúðvíksson og Gunnlaugur Sigurðsson.

2.3 Ítranir

Í upphafi var ákveðið að hafa hvern sprett sjö daga og næsti dagur á eftir yrði útgáfu- og skipulagsdagur. Þannig var fengin útgáfuáætlun sem hljóðaði upp á þrettán spretti. Fyrsti spretturinn hófst 20. janúar. Útgáfuáætlunin tók svo breytingum vegna tæknilegra hindrana sem höfðu áhrif á eina þrjá spretti og þ.a.l. þótti ekki rétt að vera með útgáfu og sýningu á afurðinni. Síðasti spretturinn kláraðist svo 16. maí, en ákveðið var að hafa hann þrettán daga.

2.4 Skipulagsþóker

Skipulagsþóker er aðferð sem notuð er til áætlunargerðar í liprum eða snörpum (e. agile) forritunarverkefnum. Á fundi (e. planning meeting) les eigandi (e. product owner) verks eða viðskiptavinur upp notendasögu (e. user story) eða lýsir þeim eiginleika sem á að meta fyrir meðlimum teymisins sem svo sjá um matið. Hver meðlimur hefur spilastokk með spilum sem hafa gildin 0, 1, 2, 3, 5, 8, 13, 20, 40, og 100 sem er aðlöguð fibonacci-röð. Gildin tákna sögupunkta (e. story points) sem er huglæg mælieining sem tákna flækjustig, tíma eða aðra mælieiningu sem ákveðið er að nota.

Eiginleikinn er ræddur og eigandi eða viðskiptavinur spurður spurninga eftir þörfum. Þegar eiginleikinn er fullræddur dregur hver og einn spil úr stoknum, þó án þess að aðrir sjái, sem gefur til kynna mat viðkomandi. Svo afhjúpa allir spilin sín á sama tíma. Ef öll spilin hafa sama gildi er það matið fyrir söguna. Ef ekki, þá er sagan rædd betur og er þá sérstaklega horft til þeirra sem hafa verið á öndverðum meiði hvað matið varðar. Eftir umræðurnar er svo nýtt spil valið og öll spilin svo aftur afhjúpuð á sama tíma. Þetta er endurtekið eins oft og þurfa þykir. (Planning Poker for Estimating on Agile Projects)

2.5 Afkastageta

Áætlunin gerði ráð fyrir að afkastageten yrði 14 sögupunktar á sprett. Eftir tvo spretti var meðaltalið hins vegar 19 punktar. Að fimm sprettum liðnum var það komið í 16 punkta og var svo komið í 14.4 punkta við lok verkefnis. Viðvera var frá klukkan 13.00 á virkum dögum og eftir þörfum og samkomulagi um helgar og lagt upp með a.m.k. fimm tíma viðveru þá daga sem unnið var að spretti. Hver vinnudagur var 6 klukkustundir og fundum við út að hver 7 daga sprettur hefði 67 klukkustundir í heildina með 20% slaka (e. buffer) (sjá AOS-sprettir.xls).

2.6 Dagbók

Haldið var utan um viðveruskráningu í þartilgerðu skjali (sjá AOS-Tímaskráningar.xls). Auk þess voru allar notendasögur og verkliðir skráð í Jira-kerfi Símans. Verkefnið hafði sitt eigið svæði í samvinnukerfi (e. wiki) Símans sem kallast Confluence og er frá Atlassian. Í Confluence voru öll skjöl geymd og einnig fór fram umræða í formi athugasemda um atriði tengd verkefninu.

2.7 Aðstaða

Vinna fór fram í húsakynnum Símans við Suðurlandsbraut 28. Sögur og verkliðir voru hengdar upp á scrum-vegg sem var til staðar og þar fóru daglegir stöðufundir fram.

2.8 Framtíðarsýn

Ákvörðun var tekin um að semja lyfturæðu (e. elevator statement) sem lýsir kerfinu í mjög stuttu máli:

„Fyrir rekstaraðila símstöðva sem þurfa að fylgjast með lífsmarki og viðvörunum þeirra. AOS er vefkerfi sem veitir notendum aðgang að viðvörunum símstöðva á einfaldan og fljótegan hátt.“

2.9 Þróunarumhverfi og aðferðir

Við forritunina voru tvö Java-þróunarumhverfi notuð ásamt verkfærum til að gera fyrirspurnir í gagnagrunn. Eclipse, Helios-útgáfa, var notað við þróun móttakara og SpringSource Tool Suite fyrir notendaviðmót.

Móttakarinn var skrifaður samkvæmt biðlara-miðlara (e. client-server) mynstri og er móttakarinn í hlutverki miðlara og símstöðvarnar biðlara.

Notendaviðmótið fylgir MVC-mynstri, en því er skipt niður í 3 einingar, módel (e. model), sýn (e. view) og stjórnklasa (e. controller).

2.10 Samstæðustjórnun

Við samstæðustjórnun var notast við sjálfvirkja útgáfusmíði (e. build automation) á sjálfvirkum útgáfujóni (e. continuous integration server) sem heitir Bamboo frá fyrirtækinu Atlassian og allur kóði var geymdur í Subversion sem er útgáfuumsýslukerfi (e. code versioning system). Bamboo sótti kóðann skjálfkrafa úr Subversion þegar kóðinn var skráður inn, smíðaði kerfin og einingaprófaði þau.

Skjöl voru ýmist unnin í Word eða Excel og geymd í Confluence eða með Google Docs og er þau að finna á meðfylgjandi geisladisk.

2.11 Útgáfustjórnun

Notast var við ákveðið kerfi við að merkja afurð að spretti loknum. Þetta kerfi fólst í því að nota útgáfunúmer í nafn spretts sem táknaði útgáfunúmer úr Bamboo. Útgáfunúmer er á sniðinu a.b.c.

A	B	C
Fullnaðarútgáfa	Númer spretts	Útgáfunúmer í Bamboo
o	1	3

Tafla 1: Útgáfunúmer

Eftir fyrsta sprett yrði útgáfunúmerið t.d. 0.1.3. Fyrsta talan gefur til kynna hversu oft kerfið hefur verið sett í rekstur. Seinni talan segir til um númer spretts (ítrun) sem kerfið var þróað fyrir. Síðasta talan segir til um Bamboo-útgáfunúmerin á kerfunum til samans. Sjá mynd 3 sem sýnir Bamboo-útgáfunúmer inni í rauðu kössunum.

☒ AOS

☒ [AOS-IP > AOS-DEF](#) 202

☒ [AOS-WEB > AOS-AOSWEB](#) 177

Mynd 3. Útgáfunúmer í Bamboo

3 Greining

Góð greining verkefnis er einn mikilvægasti þáttur þess svo niðurstaðan verði eins nálægt því sem lagt er af stað með í upphafi og mögulegt er. Því var lagst í yfirgripsmikla þarfa- og áhættugreiningu á verkefninu.

3.1 Parfagreining

Fundað var með eigendum verkefnisins og þarfirnar greindar og skráðar á kröfulista (sjá AOS-Kröfulisti.xlsx). Kröfurnar voru flokkaðar eftir A, B, C skilgreiningum eftir forgangi í samráði við eigendurna. Flokkunin var eftifarandi: A-kröfur eru nauðsynlegar og í mesta forgangi, B-kröfur eru nauðsynlegar en ekki í forgangi og C-kröfur eru hvorki nauðsynlegar né í forgangi.

Kröfulistinn var síðar notaður til hliðsjónar við gerð á notendasögum. Notendasögurnar eru meira lýsandi en kröfur í kröfulista. Notendasögurnar lýsa ákveðnu markmiði sem notandi vill ná fram með tilgreindu verkflæði. Framsetning á notendasögu eru: Sem [hlutverk] vil ég framkvæma [markmið] svo að [ávinningur]. Til er styttri útgáfa en þá er [ávinningi] sleppt úr sögunni. Orðunum inni í hornklofunum er skipt út fyrir raunveruleg gildi, hlutverk gæti orðið starfsmaður sem dæmi.

Dæmi: Sem [starfsmaður] vil ég [auðkenna mig] svo að [ég geti unnið í kerfinu]. (User story)

Notendasögu var gefinn sögupunktur og raðað í forgangsröð, efsta sagan hafði mesta forgang en neðsta sagan minnsta forgang (sjá AOS-Product_Backlog.xlsx). Fyrir hvern sprett voru sögurnar svo greindar niður í verkliði frá einni upp í átta klukkustundir. Ef verkliður tók meira en átta klukkustundir var hann brotinn niður í fleiri verkliði.

3.2 Áhættugreining

Helstu áhættuþættir í verkefninu voru greindir með því að taka saman atriði sem gætu komið upp í framvindu verkefnis. Stuðst var við almenna áhættugreiningu í verkefnastýringu og notast var við áhættustuðul sem samanstendur af líkindum og alvarleika.

Líkur	Lýsing	Alvarleiki	Lýsing
1	Litlar líkur (0-20%)	1	Minni háttar alvarlegt
2	Frekar ólíklegt (20-40%)	2	Lítið alvarlegt
3	Líklegt (40-60%)	3	Meðal alvarlegt
4	Frekar líklegt (60-80%)	4	Frekar alvarlegt
5	Mjög líklegt (80-100%)	5	Mjög alvarlegt

Tafla 2: Áhættustuðlar

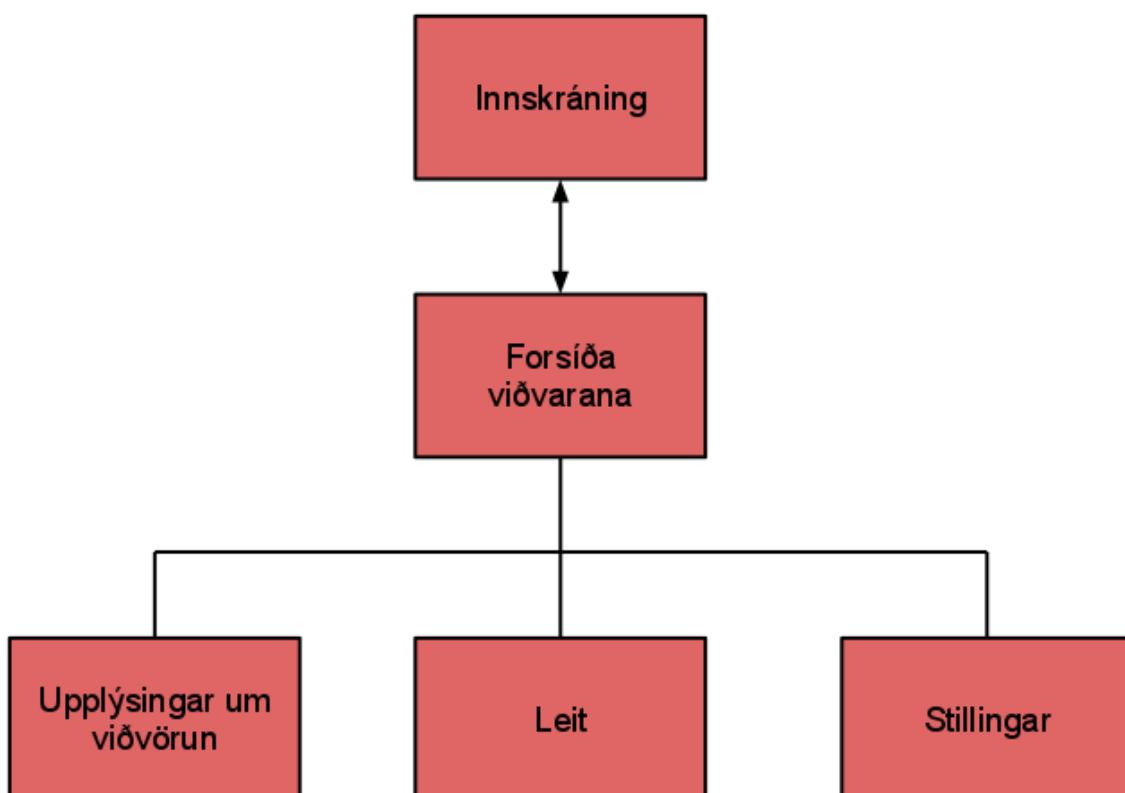
Áhættustuðullinn var reiknaður með því að margfalda líkur við alvarleika. Atriði sem höfðu mesta áhættu gátu fengið í mesta lagi áhættustuðulinn 25 en minnst 1. Áhættugreining af þessu tagi hefur þann galla að tvö atriði geta fengið sama áhættustuðul en haft þveröfugar líkur og alvarleika. Sem dæmi má nefna atriði sem hefur litlar líkur (1) en er mjög alvarlegt (5), hefur sama áhættustuðul og atriði sem er mjög líklegt (5) en er minni háttar alvarlegt (1). Bæði atriðin hafa áhættustuðul upp á 5 en atriðin gætu verið mjög ólík og erfitt að greina hvort þeirra hefði meiri áhrif á verkefnið. (Risk management)

Nákvæma útlistun áhættuþátta má finna á meðfylgjandi geisladisk (sjá AOS-Áhættugreining.xls).

4 Hönnun

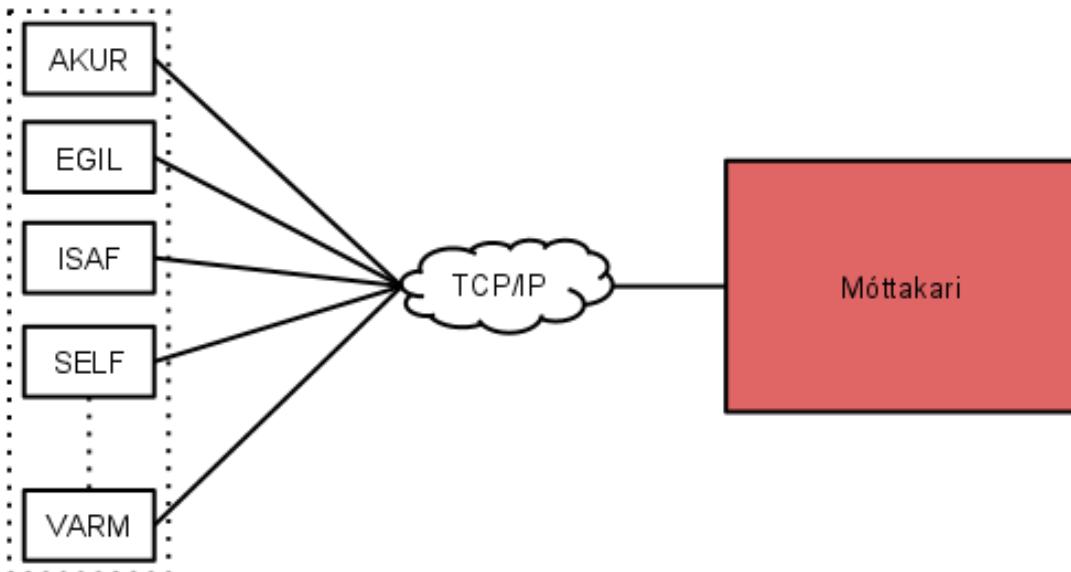
Við hönnunina voru nokkrir notendur kerfisins fengnir til að sýna og lýsa notkun sinni á gamla AlarmMonitor-notandaviðmótinu. Viðmótið er komið til ára sinna þó að það hafi þjónað tilgangi sínum vel og því verk að búa til nýtt notendaviðmót sem notendur yrðu sáttir við. Gamla kerfið er flókið með mörgum eiginleikum sem margir hverjir eru ekki notaðir. Hjartsláttur símstöðvar er ekki aðgengilegur. Öflug leit er í gamla kerfinu en flókin. Við hönnun nýja viðmótsins var lagt upp með að einfalda siglingaleiðarritið svo hægt væri að framkvæma allar aðgerðir með eins lítilli fyrirhöfn og mögulegt væri. Últitið var samræmt við aðra innri vefi Símans og til þess var notað sameiginlegt CSS (e. cascading style sheet) sem allir nýir vefir Símans nota.

Flæðið í nýja viðmótinu er einfalt. Fyrsta síðan er innskráning, þaðan fer notandinn á forsíðu sem innheldur virkar viðvaranir og lífsmark símstöðva. Þaðan kemst notandinn á síðurnar upplýsingar um viðvörun, leita og stillingar. Mynd 4 sýnir siglingaleiðarrit viðmótsins.



Mynd 4: Siglingaleiðarrit notendaviðmóts

Móttakarinn í nýja kerfinu hefur samskipti við símstöðvar með TCP/IP (e. transmission control protocol/internet protocol) samskiptastaðlinum. Hlutverk móttakarans er að flokka skeytin sem koma frá símstöðvunum og vista í gagnagrunni. Guli kassinn á mynd 5 er sá hluti sem var útfærður í nýja kerfinu.



Mynd 5: Einföld yfirlitsmynd af móttakara

Gagnagrunnsskemað þurfti að hanna upp á nýtt. Uppbygging gamla gagnagrunnsins er flókin og ekki til þess gerð að hægt sé að einfalda kerfið ásamt því að verið er að fasa út Informix-gagnagrunnum fyrir Oracle. Við hönnun nýja gagnagrunnsins var leitast við að hafa uppsetninguna einfalda og því fengu margar ónotaðar og lítið notaðar töflur úr gamla gagnagrunninum að falla út, án þess að það hefði áhrif á virkni nýja kerfisins.

5 Forritun

Móttakarann mátti forrita í C, C++ eða Java. Þar sem mikið af forritun hjá Símanum er í Java var ákveðið að notast við það. Notendaviðmótið átti að skrifa í Java og stóð valið á milli Oracle AFT eða Spring Roo sem ramma (e. framework) við þróunina. Spring Roo varð fyrir valinu þar sem kostnaðurinn við Oracle AFT þótti of mikill til að hægt væri að réttlæta kaup á leyfum fyrir það.

5.1 Móttakari

Móttakarinn hefur það hlutverk að taka við skeytum frá símstöðvunum, vinna úr þeim og vista ofan í gagnagrunn. Hann er útfærður með miðlari/biðlari-mynstri og sem slíkur er hann tenglamiðlari (e. socket server). Samkvæmt skilgreiningu er “socket” annar endapunkturinn í tvíhliða samskiptum tveggja forrita á neti. “Socket” er bundið við ákveðið port svo tryggt sé að TCP-samskiptatalagið geti sent forritinu þau gögn sem því er ætlað. (Oracle)

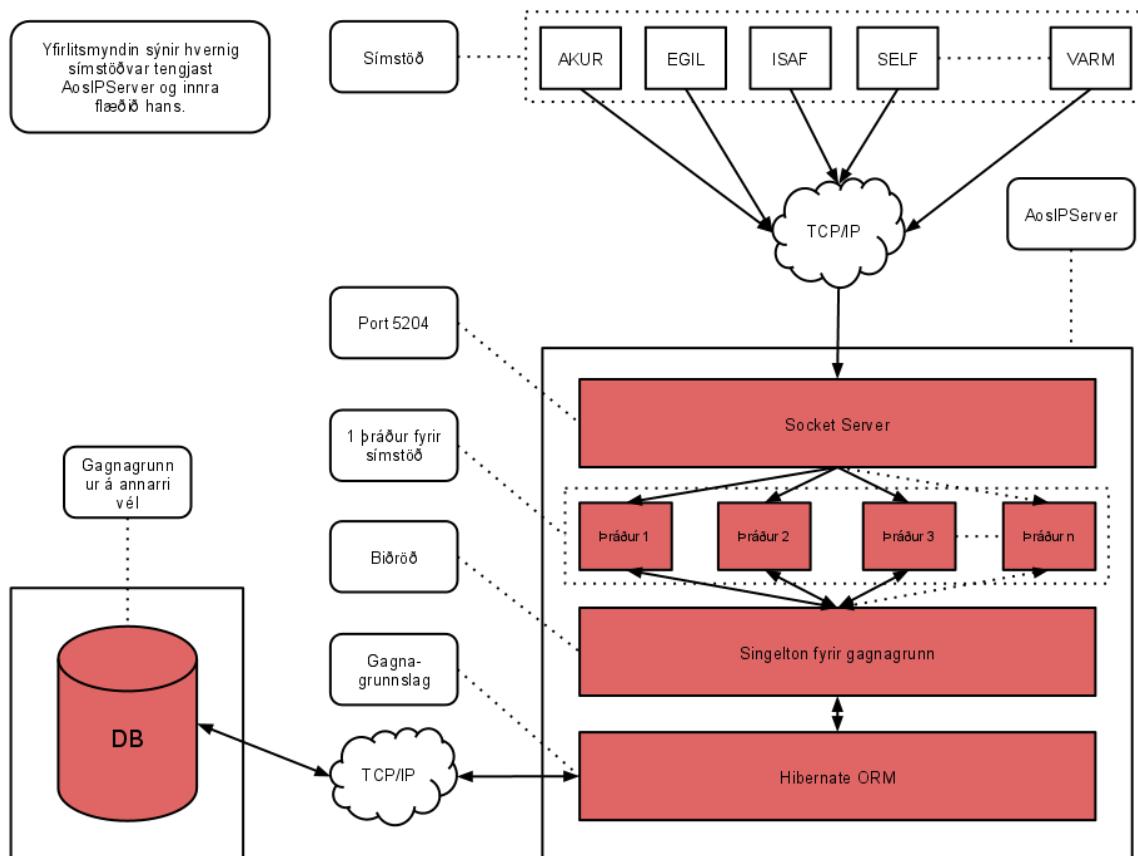
Tengimiðlarinn var útfærður með þráðum (e. threads) sem gerir það að verkum að hægt er að vera með fjölverkavinnslu (e. multitasking) þar sem tvær eða fleiri verkeiningar keyra samhliða eða fléttast á einhvern hátt.

Við hverja tengifyrirspurn er búinn til nýr þráður sem er eins konar verkeining sem fær það ábyrgðahlutverk að sjá um tenginguna þar til að einingin lýkur vinnslu, en þá er tengingunni lokað.

Notast var við “singleton”-mynstur til þess að tryggja gagnaheilindi við skrif og uppfærslur í gagnagrunni. Þegar þráður hefur lokið við að flokka skeytin úr símstöðinni býr hann til gagnagrunnshlut sem er færður yfir í “thread-safe”-biðröð (e. queue). Í “singleton”-klasanum er sérstakur þráður sem hefur það hlutverk að fylgjast með biðröðinni og ef eitthvað er í henni er gagnagrunnshlutarinn vistaður ofan í gagnagrunninn.

Hibernate er notað til þess að kortleggja (e. mapping) gagnagrunnskemað (e. database schema) yfir í hlutbundna (e. object oriented) nálgun. Gagnagrunnurinn er af tegundinni Oracle 1og sem Síminn kýs að nota.

Eins og áður segir var móttakarinn skrifaður samkvæmt miðlari/biðlari (e. client-server) mynstri þar sem símstöðvar eru í hlutverki biðlara. Á mynd 6 má sjá flæðið á milli símstöðvanna og miðlara ásamt því að sýna hvernig innra flæði miðlarans er háttáð.



Mynd 6: AosIPServer-yfirlitsmynd

5.2 Notendaviðmót

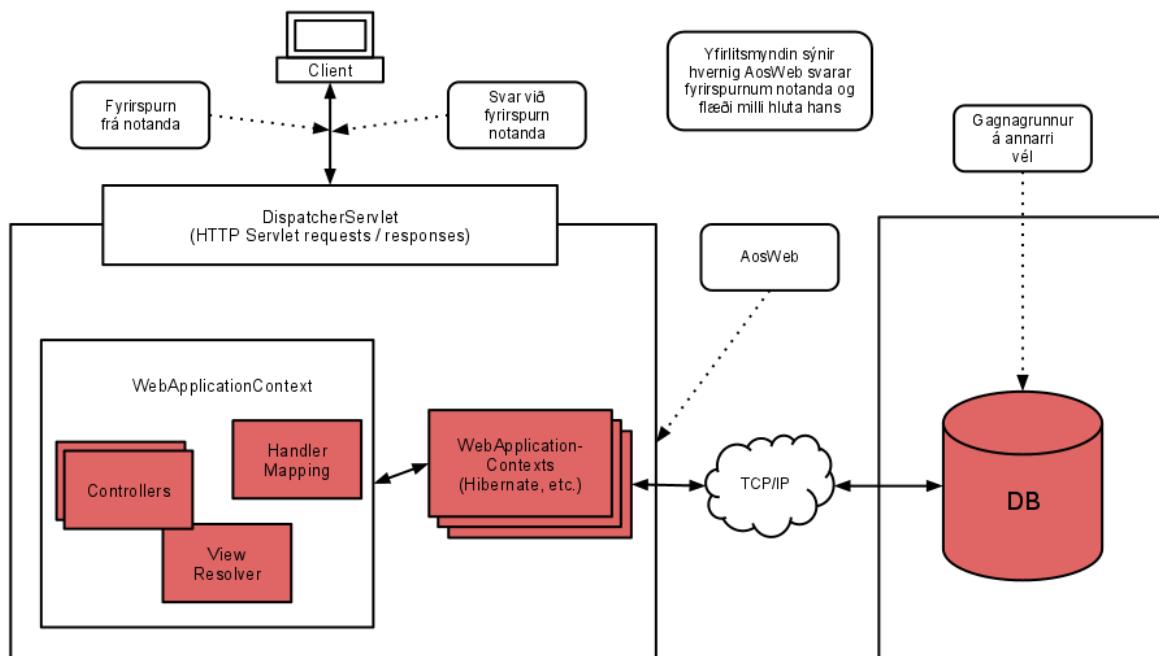
Notendaviðmótið var útfært sem Java-vefviðmót sem notar Spring MVC- og Spring Roo-ramma (e. framework). Vefviðmótið notar MVC-högun þar sem hver hluti er vel aðskildur frá öðrum sem gerir kerfið einfaldara í þróun. Stýriklasarnir (e. controller) nota REST (e. representational state transfer) högun fyrir URL (e. uniform resource locator) vefföng. Þegar fyrirspurn kemur inn sér stýriklasinn um að svara fyrirspurn (e. request), sækja gögn upp úr gagnagrunninum og setja í módel (e. model), breytu og skilgreina sýn (e. view) sem móttetur svar (e. response) sem birtist notanda á JSP (e. java server pages) síðu.

Spring Roo-ramminn einfaldaði mjög þróun vegna þess hversu þægilegt var að nota skipanir til þess að framkvæma hinum og þessar aðgerðir sem tengdust kortlagningu gagnagrunnsins.

Spring Roo notar AOP (e. aspect oriented programming), einnig kallað (e. separation of concerns) högun til þess að skipta hlutum niður í sjálfstæðar einingar (e. modules).

Gagnagrunnseigindi (e. entity) voru t.d. brotin niður í mismunandi skrár þar sem skilgreindir voru liðir (e. join-point) sem tengdu hlutina saman. Gagnaliðir í eigindunum voru settir í sér skrár og gagnagrunnsaðgerðir um lestur, uppfærslu og skrif á gögnum voru í annarri skrá.

Vefviðmótið notar Hibernate líkt og móttakarinn til þess að kortleggja gagnagrunninn enda er um sama gagnagrunnskemað að ræða. Gagnagrunnskemað í viðmótinu fylgir (e. active record) mynstri en móttakarinn notar (e. service layer) mynstur. Mynd 7 sýnir hvernig högun vefviðmótsins er háttáð.



Mynd 7: AosWeb-yfirlitsmynd

6 Prófanir

Þrenns konar prófanir voru notaðar við vinnslu afurðarinnar. Einingaprófanir á kóða, kerfisprófanir til þess að ganga úr skugga um að hverri sögu væri lokið og notendaprófanir sem voru framkvæmdar að forrituninni lokinni.

6.1 Sjálfvirkar prófanir

Við sjálfvirku prófanirnar var notast við Log4j-rammann frá Apache sem er opinn hugbúnaður notaður af allflestum Java-forriturum til prófana.

Í miðlaranum voru smíðuð sjálfvirk próf sem höfðu það hlutverk að skrá í gagnagrunninn nýjar viðvaranir, losa viðvaranir og hjartslætti sjálfvirk. Að auki voru prófin notuð til að setja nauðsynleg gögn eins og símstöðvar, viðvörunarflokka, viðvörunarklasa og útstöðvar í gagnagrunninn.

6.2 Kerfisprófanir

Hver saga innhélt verklið sem voru prófanir á því að útfærslunni á sögunni væri lokið. Þessar prófanir voru framkvæmdar af teyminu sjálfu. Flestar villur sem fundust við kerfisprófanir voru leystar jafnóðum, ekki var haldið sérstaklega utan um villur nema þær sem voru tímafrekar og voru þær skráðar í Jira.

6.3 Notendaprófanir

Leitað var til tveggja væntanlegra notenda kerfisins ásamt atvinnuprófara til að framkvæma notendaprófanirnar. Þeir fengu í hendurnar skjal (sjá AOS-Notendaprófanir.doc) með lista af 26 fyrirframákveðnum prófunartilvikum sem þeir voru beðnir um að framkvæma og skrá niðurstöður í skjalið. Niðurstaða prófana leiddi í ljós þrjár villur og voru þær allar lagaðar.

Nafn	Starf	Athugasemdir	Mynd
Erlingur Tómasson	Hópstjóri Stjórnstöð	„Stjórnstöð Símans er með eftirlit alla daga allan sólahringinn með öllum þeim vél- og hugbúnaði Símans sem nauðsynlegt er talið að vakta. AOS-forritið á að auðvelda okkur vinnuna og höfum við til þess góðar væntingar miðað við það sem ég hef séð. Við leggjum mikið upp úr einfaldleika og skýrleika (aflestur) því mikilvægt er að nauðsynlegar upplýsingar fáist með sem minnstri fyrirhöfn og sýnist mér að það takist. Áhyggjuefni er helst að veflausnir í gaumkerfum geta verið óstöðugar og svartími breytilegur, en það er ef til vill utan þessa forrits.“	
Jóhannes Hilmisson	Hópstjóri Talsímarekstur	„Kröfurnar til AOS-forritsins byggja á reynslu af AlarmMonitor-forritinu sem hefur verið í notkun hjá Símanum seinustu 10 – 12 ár. Með AOS er virkni eldra forritsins betrumbætt og gefur væntingar um að framtíðarþróun verði auðveldari en í AlarmMonitor-forritinu.“	
Ægir Gauti Þorvaldsson	Prófari UT-rekstur og þróun	„AOS-kerfið er þægileg og einföld lausn til að fylgjast með lörmum í sjálfvirkum símstöðvum. Notendavænt og sýnir mér það sem ég þarf að sjá til að fylgjast með stöðu á hverjum tíma.“	

Tafla 3: Notendur sem prófuðu kerfið.

7 Framvinda

Eins og fram hefur komið gerði áætlun verkefnisins ráð fyrir 13 sprettum. Sú áætlun breyttist þegar á verkefnið leið og kom það til vegna hindrana í verkefninu og var tekin ákvörðun um að lengja nokkra spretti. Í töflu 4 má sjá upphafs- og lokadaga þeirra 11 spretta sem verkefnið endaði í. Nákvæmari útlistun á framvindunni er að finna á meðfylgjandi geisladisk (sjá AOS-Framvinda.pdf).

Sprettur	Upphafsdagur	Lokadagur	Fjöldi daga
0.1.50	20.01.2011	27.01.2011	7
0.2.103	29.01.2011	07.02.2011	10
0.3.145	08.02.2011	15.02.2011	7
0.4.189	17.02.2011	24.02.2011	7
0.5.256	26.02.2011	05.03.2011	7
0.6.282	08.03.2011	17.03.2011	9
0.7.291	18.03.2011	23.03.2011	5
0.8.293	25.03.2011	31.03.2011	6
0.9.321	05.04.2011	19.04.2011	15
0.10.362	21.04.2011	30.04.2011	9
1.11.0	04.05.2011	16.05.2011	13

Tafla 4. Yfirlit spretta.

8 Lokaorð

8.1 Hvað gekk vel?

Í heildina má með sanni segja að verkefnið hafi gengið vel þótt hindranir hafi verið í veginum. Þau atriði sem standa upp úr verða tíunduð hér.

Teyminu gekk vel að ná utan um þær kröfur sem gerðar voru til nýja kerfisins. Til að ná utan um kröfurnar var gamla viðmótið skoðað með þeim sem það nota og þeir beðnir að sýna það og segja hvað væri notað og hvað ekki. Vel gekk að koma upp stöðluðu útliti innri vefja Símans í viðmótinu.

Samstæðustjórnunin kom vel út og reyndist teyminu vel við útgáfustjórnun og til að tryggja að forritin brotnuðu ekki við að kóðinn væri skráður inn í Subversion.

Mikil þægindi fylgdu því að nota Hibernate við að kortleggja gagnagrunninn. Gott er að geta eytt honum út og sett aftur upp með einni skipun.

Auðveldlega gekk að leysa hluti sem komu upp um leið og búið var að átta sig á því hvernig það skyldi gera.

Samvinna gekk vel. Auðvelt var að komast að niðurstöðu um útfærslur á hinum ýmsu hlutum. Ef mismunandi hugmyndir komu upp hvað útfærslu eða önnur mál varðaði voru þau atriði rædd þar til að ásættanleg niðurstaða fékkst sem allir gátu sætt sig við.

Brennsla sögupunkta gekk í heildina vel. Af 187 punktum voru 10 punktar eða tvær sögur sem fóru inn í næstu útgáfu en þær voru í rauninni ekki hluti af þessu verkefni. Ef frá eru dregnir 29 punktar sem voru rangt greindir og aðeins litið á þá punkta sem voru kláraðir og þá sem fóru í næstu útgáfu kláruðust 93.7% af verkefninu.

8.2 Hvað gekk illa?

Í upphafi stóð til að nota Google Web Toolkit í forritun viðmótsins en það átti eftir að reynast erfitt því ekki var hægt að nota GWT Designer sem er tól til þess að hanna viðmót. Spring Roo ramminn styður ekki GWT Designer. Spring Roo býr sjálfvirkt til grunniðmót sem notar útlitsþema sem hefði verið mjög tímafrekt að aðlaga stöðluðu útliti Símans. Því var ákveðið að smíða nýja stýriklasa (e. controllers) og sýn (e. view) með JSP-síðuðum í stað þess að nota það sem Spring Roo getur búið til sjálfvirkt og nýta sameiginlegt CSS (e. cascading style sheets) innri vefja Símans.

Til þess að tryggja gagnaheilindi við móttöku og vinnslu á gögnum frá símstöðvum ákváðum við að nota þræði til þess að vinna gögnin á meðan móttakarinn hélt áfram að hlusta eftir gögnum. Þessi aðferð reyndist ekki vel þegar xinetd (e. the extended Internet services daemon) tók á móti gögnum og ræsti upp nýjan móttakara fyrir hverja gagnasendingu frá símstöðvunum. Þetta hafði þær afleiðingar að þróunarvélin kláraði minnið og fór niður því móttakarinn var forritaður þannig að hann væri stöðugt lifandi. Það kom svo í ljós að sumar símstöðvar halda uppi tengingunni á meðan aðrar gera það ekki. Því var ekki hentugt að nota xinetd-forritð. Þá hefði þurft að hafa tvær útgáfur af móttakara, einn sem hættir keyrslu þegar símstöðin er búin að senda gögnin og annan sem er stöðugt í gangi. Sú lausn hefði aldrei gengið því þá hefði xinetd þurft að vita hvaða símstöðvar slíta tengingu og hverjar ekki. Því var ákveðið að skrifa socket server í stað þess að eyða meiri tíma í xinetd.

Eftir að gögnin frá símstöðvunum höfðu verið móttokin fór þráðavinnsla af stað við að flokka gögnin og skrifa í gagnagrunninn. Í þessum samtíma vinnslum töpuðust uppfærslur í þeim tilfellum þegar þurfti að lesa úr gagnagrunninum og uppfæra. Til þess að leysa þetta vandamál var búin til “thread-safe”-biðröð þar sem aðeins einn þráður gat sett inn í biðröðina og einungis ein gagnaþærsla skrifuð í gagnagrunninn hverju sinni.

Til þess að auðkenna notanda þurfum við að nota Active Directory (e. directory service) Símans. Það tók langan tíma að stilla Spring Security svo hægt væri að lesa hópa sem notendur tilheyrðu. Flestallar leiðbeiningar og bækur sem lesnar voru gerðu ráð fyrir að Spring Security væri tengt við gagnagrunn eða LDAP en ekki Active Directory.

8.3 Hvað lærðum við?

Vinnan við verkefnið var afskaplega lærðómsrík og munu meðlimir teymisins búa vel að þeirri reynslu. Að fá að koma að skipulagningu verkefnis, sjá um alla verkstjórnun, áætlanir og annað sem fylgir nýjum verkefnum hefur veitt meðlimunum tilfinningu fyrir skipulagningu verkefna frá upphafi til enda. Í eins stóru fyrirtæki og Símanum er verkaskipting að öllu jöfnu meiri og þar af leiðandi ekki sjálfgefið og í raun afar ólíklegt að einhver einn aðili komi og vinni í öllum þáttum sem að verkefninu snúa.

Læra þurfti á ný tæki, tól og aðferðir við forritun og þróun hugbúnaðar.

Lítill reynsla var af notkun Hibernate og því reyndist sú ákvörðun að nota það ákaflaga lærðómsrík. Spring Roo-rammann (e. framework) höfðu meðlimirnir aldrei prófað og var því um nýja áskorun að ræða og þurfti að nota SpringSource Tool Suite-þróunarumhverfið fyrir það. Spring Roo notar AspectJ til að aðskilja viðskiptareglur (e. business logic) frá almennum kóða til að ná fram högun sem kallast “separation of concerns” og mun sú reynsla án efa nýtast meðlimum í framtíðarverkefnum. Ágætisreynsla á MVC-mynstrinu var til staðar þegar

verkefnið hófst en sú reynsla og þekking jókst til muna við vinnslu verkefnisins. Þræðir og socket-forritun er eitthvað sem lítil reynsla var af áður en verkefnið hófst en nú er hún tölvert meiri.

9 Framtíðin

Næsta útgáfa af kerfinu mun að öllum líkindum innihalda tvær sögur sem voru ekki hluti af grunnkröfum en væri gott að hafa í kerfinu. Þessar sögur eru: Sem AOS get ég sent tölvpóst með viðvörunum og Sem IP fæ ég sendi skeyti frá AMS fyrir MIC.

Á sýningu kerfisins kom fram ósk um breytingar sem munu verða útfærðar í næstu útgáfu þess. Fyrri óskin er sú að notendur geti opnað nýjan lista með virkum viðvörunum á forsíðu sem myndi aðeins endurhlaða lista af virkum viðvörunum frá þeim tíma sem notandi kallaði á aðgerðina. Seinni tillagan fólst í því að vinna enn frekar úr innihaldi viðvarana svo að hægt væri að birta þau gögn á forsíðu.

10 Ábendingar

Fylgiskjöl sem vísað er til í skýrslunni er að finna á geisladisk sem fylgdi skýrslunni við skil.

Listi yfir fylgiskjöl sem eru á geisladisk:

1. AOS-Áhættugreining.xls
2. AOS-Framvinda.pdf
3. AOS-Kerfishandbók.pdf
4. AOS-Kröfulisti.xlsx
5. AOS-Notendaprófanir.doc
6. AOS-Notendaprófanir_erling.htm
7. AOS-Notendaprófanir_niðurstaða_JH.htm
8. AOS-Notendaprófanir_Ægir.doc
9. AOS-Product_Backlog.xlsx
10. AOS-Rekstrarhandbók.pdf
11. AOS-Sögur_með_verkliðum.pdf
12. AOS-Tímaskráningar.xls
13. AOS-sprettir.xls
14. DATA_INSERT.sql
15. schema.sql

Á geisladiskinum sem leiðbeinandinn fékk í hendurnar er að finna frumkóða fyrir heildarlausnina.

Við skil verkefnisins er kerfið keyrandi á þróunarvélum. Viðmótið er að finna á lokuðu neti Símans á vél með slóðina <http://aos-devel.siminn.is:8080/AOS/> og móttakarinn er staðsettur á ishka.siminn.is og hlustar á port 5204. Í framtíðinni mun viðmótið verða uppsett á aos.siminn.is ásamt því að móttakarinn verði keyrandi á vél sem staðsett er í raunumhverfi.

Til að notandi geti skráð sig inn í notendaviðmótið þarf hann að tilheyra “aos-users”-hópunum í Active Directory-grunni Símans.

11 Umsagnir

„Gunnlaugur Sigurðsson og Davíð Halldór Lúðvíksson hafa á undanförnum mánuðum unnið að lokaverkefni hjá Símanum. Þeir hafa unnið að endurbótum á eldra kerfi sem sér um móttöku og birtingu á viðvörunum frá símstöðvum. Endurbæturnar hafa aðallega snúið að því að geta móttekið viðvaranir yfir IP. Nýrri gerðir símstöðva hafa einungis þann möguleika til að senda viðvaranir frá sér. Verkefnið var því orðið mjög brýnt fyrir Símann. Einnig sneri verkefnið að því að gera vefviðmót til að birta og sýsla með viðvaranir. Notast var við scrum-aðferðafræðina til að stýra verkinu og hefur það gengið mjög vel. Skjölunin fyrir verkefnið er geymd í Wiki hjá Símanum. Skjölun verkefnisins er alveg til fyrirmynnar. Allir þeir virknipættir sem gert var ráð fyrir að yrðu inni í kerfinu er nú virkir. Hugbúnaðarumhverfið sem þeir hafa sett upp virkar mjög vel og ættu viðbætur sem gerðar verða á kerfinu framtíðinni að verða þægilegar. Notendur hafa að miklu leyti stýrt því hvernig kerfið virkar. Samstarf við notendur kerfisins hefur verið mjög gott. Séð frá Símanum hefur verkefnið gengið mjög vel og verður kerfið örugglega í rekstri í mörg ár.“

Tengiliður Símans

Tómas Árni Jónsson

Deildarstjóri Grunnkerfi

12 Heimildaskrá

Mountain Goat Software. (án dags.). *Planning Poker for Estimating on Agile Projects.*

(Mountain Goat Software) Sótt 14. maí 2011 frá Mountain Goat Software Web site:

<http://www.mountaingoatsoftware.com/topics/planning-poker>

Oracle. (án dags.). *What Is a Socket?* (Oracle) Sótt 15. maí 2011 frá Oracle Java Tutorials Web site: <http://download.oracle.com/javase/tutorial/networking/sockets/definition.html>

Wikipedia. (9. janúar 2009). *File:Scrum_process.svg.* (Wikipedia) Sótt 14. maí 2011 frá Wikipedia: http://en.wikipedia.org/wiki/File:Scrum_process.svg

Wikipedia. (án dags.). *Risk management.* Sótt 14. maí 2011 frá Wikipedia: http://en.wikipedia.org/wiki/Risk_management

Wikipedia. (án dags.). *User story.* (Wikipedia) Sótt 14. maí 2011 frá Wikipedia: http://en.wikipedia.org/wiki/User_story

