

## IWatch Home Solution



Jon Westerlund wesjon-5@student.ltu.se  
Robin Olofsson, roblof-8@student.ltu.se  
Philip Eriksson phieri-5@student.ltu.se  
Philip Neuman phineu-8@student.ltu.se

Luleå University Of Technology  
971 87 Luleå, Sweden



29 mars 2021

# Innehåll

<b>1</b>	<b>Inledning</b>	<b>3</b>
1.1	Bakgrund . . . . .	3
1.2	Problembeskrivning . . . . .	3
1.3	Uppgift och mål . . . . .	3
<b>2</b>	<b>Tekniker i det smarta hemmet</b>	<b>3</b>
<b>3</b>	<b>Systemdesign</b>	<b>4</b>
3.1	Statisk design . . . . .	4
3.2	Dynamisk design . . . . .	6
<b>4</b>	<b>Resultat</b>	<b>9</b>
4.1	Leverans . . . . .	9
4.2	Testning . . . . .	9
4.2.1	Regressionrisker . . . . .	9
4.2.2	Strategi för regressiontester . . . . .	9
4.2.3	Utvecklingsbeskrivning . . . . .	10
4.2.4	Teststrategi . . . . .	11
<b>5</b>	<b>Etisk bedömning</b>	<b>11</b>
<b>6</b>	<b>Slutsats och framtida arbete</b>	<b>12</b>
6.1	Fortsatt utökad skalbarhet . . . . .	12

# 1 Inledning

## 1.1 Bakgrund

Med dagens smart teknik finns det många nya möjligheter att förbättra sitt hem. Man kan göra det mer effektivt genom att automatiskt slå på/av många apparater eller system i hemmet för att spara ström. Det kan även vara för att förbättra bekvämligheten i hemmet, genom att styra lampor, mikrovågsugn, vitvaror och eluttag mm. Hemmet kan även individanpassas, öppna garageporten då det är dags att åka till jobbet, eller sätta på kaffebyggaren för morgonkaffet och stänga av eneheter när du lämnar huset.

Nu när även armbandsuret är tillräckligt "smart", kan även denna sköta/styra alla delar i hushållet samt få uppdateringar från alla centrala/vitala delar i huset.

De smarta produkterna kan kontrolleras av många olika system på marknaden, exempelvis Fibaro Home System eller Philips HUE. Det finns även olika system för positionering i hemmet, Widefind är ett sådant system som är på uppgång.

## 1.2 Problembeskrivning

Det övergripande problemet vi står framför är, hur får vi ett fullt fungerande smart hem med en Apple watch som central kontrollenhet. Apple watchen ska även hämta annan data så som pulsmätning, syremättnad i blodet och blodtryck.

Syftet med projektet är att på ett enkelt sätt styra och få uppdateringar från flertalet sensorer och enheter i en användares smarta hem som använder sig av olika kommunikationsprotokoll. Vilket leder till att fysiska uppgifter lätt kan genomföras med hjälp av ett knapptryck.

## 1.3 Uppgift och mål

Målet med denna tjänst är att en användare ska få kontroll över det smarta hemmet genom att styra olika enheter från sin Apple watch samt få en överblick om sin hälsa och aktiva sensorer i hemmiljön.

Styrningen av dessa enheter ska kunna ske manuellt via interaktion med klockan samt automatiskt efter rörelser rapporterade från widefinds positionerings system. Utöver styrning av diverse enheter ska klockan övervaka användarens hälsovärden och visa dessa värden till användaren i realtid, när syremättnad eller puls visar farliga värden skall användaren notifieras.

Konfiguration av de enheter och sensorer i det smarta hemmet kommer inte att implementeras via applikationen utan det görs via respektive kontroll enhet (Z-wave enheter via Fibaro, ZigBee via philip hue bridge).

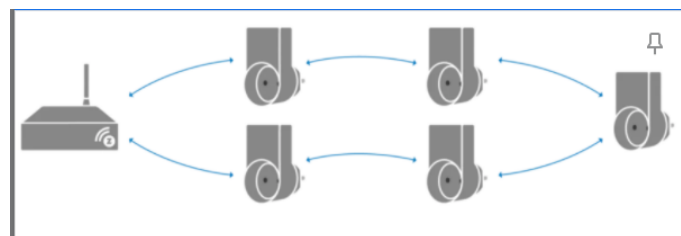
# 2 Tekniker i det smarta hemmet

### Widefind

Widefinds WISPR system används för UWB positionering inom gruvinindustrin, systemet ger en exakt position av den bärbara taggen inom sitt koordinatsystem. Taggens avstånd till WISPR ankarna utplacerade i övervaknings området skickas via en gateway till en WISPR server som sammanställer datan och beräknar taggens position. Taggens position hämtas därefter enkelt via REST och streaming APIs i form av 'REPORT' meddelanden. [1]

### Fibaro

Fibaro Home center är en kontrollenhet för att styra och hålla reda på smarta produkter i ett lokalt nätverk. Fibaro använder sig utav Z-wave teknologin för att kommunicera med de olika sensorerna och enheterna i nätverket. Z-wave tekniken tillåter noderna att kommunicera med varandra, vilket leder till snabbare och flexiblare kommunikation. [2]



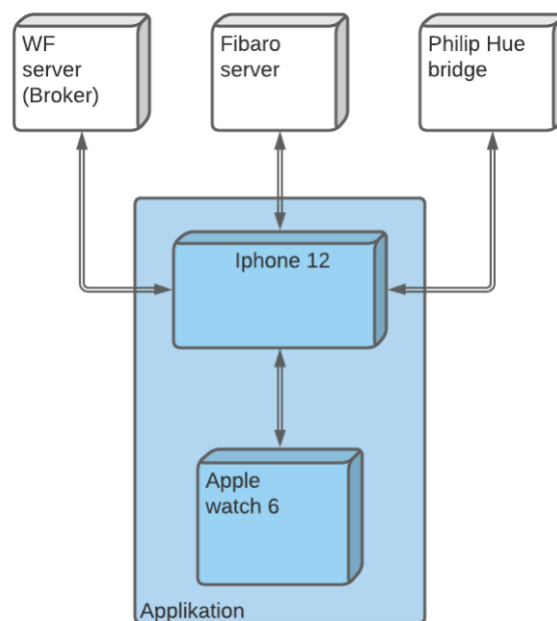
Figur 1: FHC kommunicerar med noderna

### Philips Hue

Philips har en mängd olika produkter som kan integreras i ett smart hem. Under det här projektet har ett gäng olika lampor satts in i det smarta hemmet och de kan styras genom Philips egna kontroll enhet Hue bridge. Hue bridge kommunicerar med de olika lamporna via ZigBee kommunikations protokollet. [3]

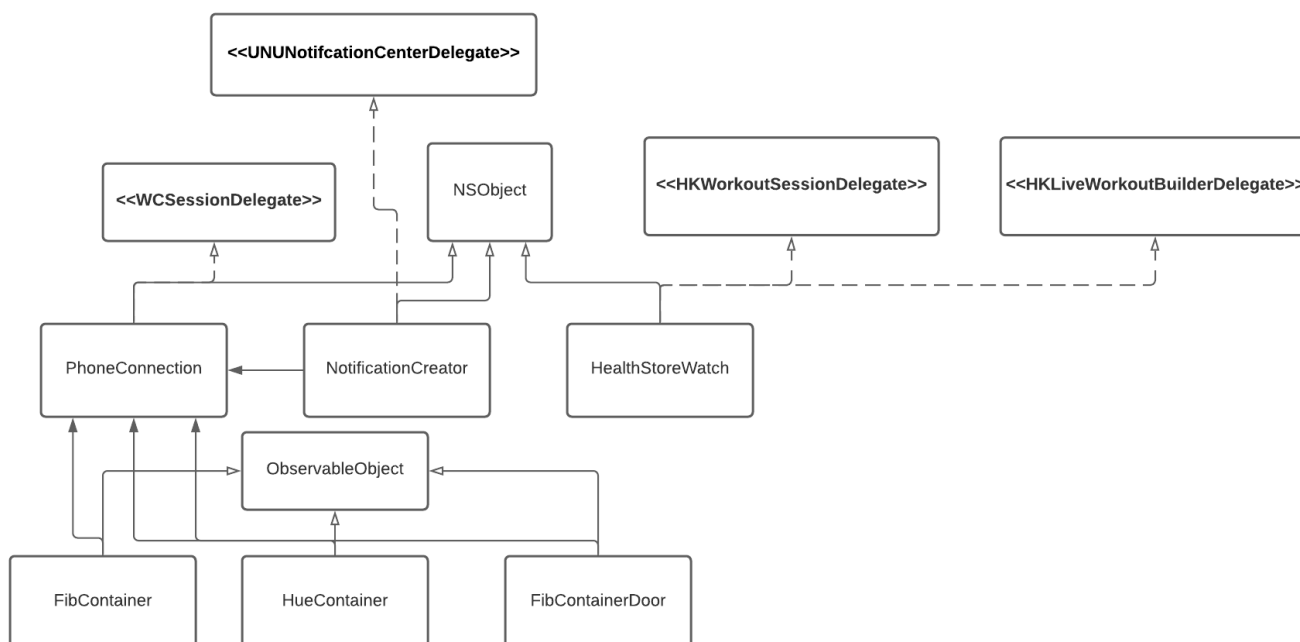
## 3 Systemdesign

### 3.1 Statisk design



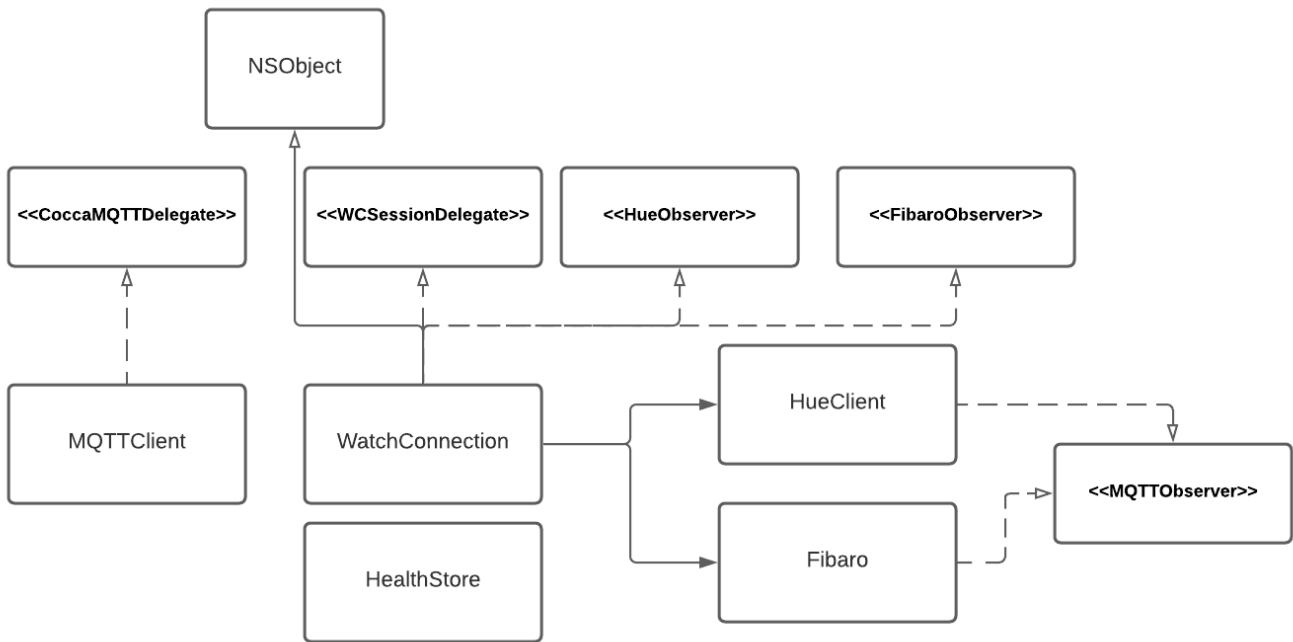
Figur 2: Högnivå system arkitektur diagram. Användaren styr de olika sensorerna via *Apple watch 6*.

Figur 2 visar en översikt av systemet. Systemet är uppdelat i två olika moduler. Applikationen på klockans sida opererar via *Model view controll* mönstret och är den delen av systemet som användaren interagerar med. Applikationens primära bearbetning av data och kommunikation med de olika sensorerna sker via mobilen. Applikationen på mobilsidan är designad för att främst operera via *Observer* mönstret.



Figur 3: Klassdiagram av applikationen på klockan

All kommunikation till mobilen från klockan sker via *PhoneConnection* som är en wrapper till *WCSessionDelegate* (*WCSession* är Apples egna API för kommunikation mellan iOS och watchOS enheter). Sensordatan som hämtas via *PhoneConnection* är sparade lokalt och uppdaterad via mobilen då användaren uppdaterar sin vy eller gör anrop för att ändra status för en sensor i de smarta hemmet. Alla olika sensortyper som visas i klockan (Z-wave binära switchar, Philips hue lampor, Z-wave dörröversvakning sensorer) är uppdelade i egna containers för att säkerställa korrekt uppdatering av vyn vid hämtning av dessa sensorers och enheters status.

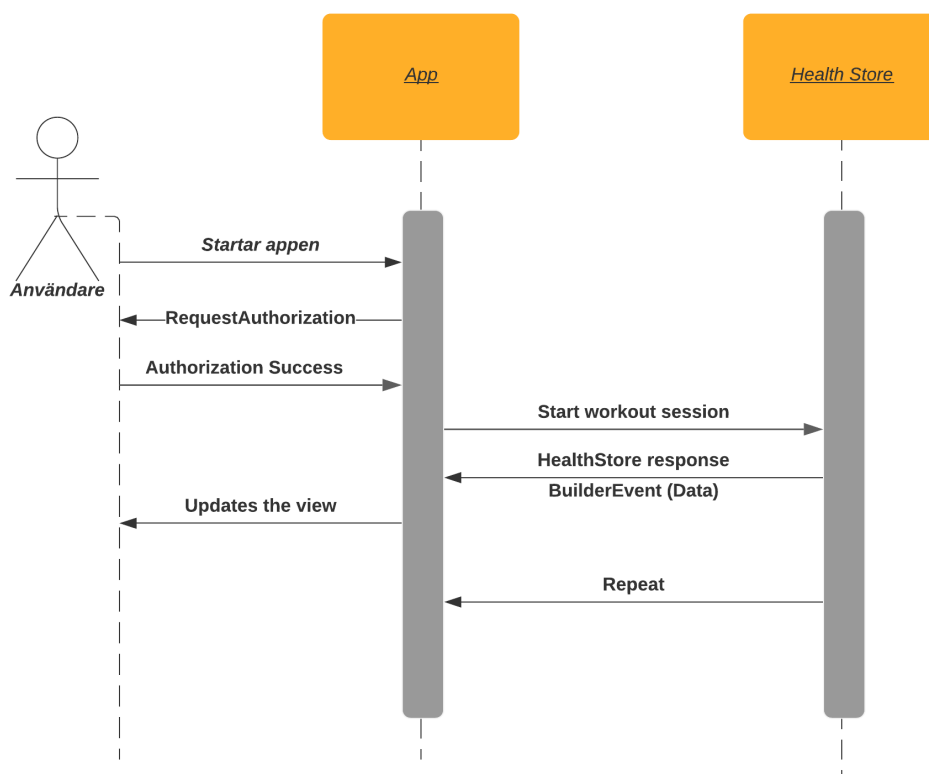


Figur 4: Klassdiagram av applikationen på mobilen

*WatchConnection* hanterar all kommunikation mellan klockan och telefonen via ett *WCSession* objekt. Beroende på meddelandedet från klockan så kommunicerar *WatchConnection* det vidare till *Fibaro* eller *PhilipHue* klienterna, som sen beroende på vad meddelandet säger ska antingen uppdatera datan eller ändra tillståndet på enheterna. *MQTT* klienten används som en kommunikations länk mellan telefonen och widefind brokern som är placerad i de smarta hemmet. Beroende på vart widefind taggen befinner sig i de smarta hemmet uppdateras *MQTT* klienten angående taggens position. Om taggen förflyttar sig från ett rum till ett annat anropar klienten alla observerare av objektet av de specifika eventent som skett. *Fibaro* och *HueClient* klienterna skötter anrop till respektive sensorer och enheter. Mer detaljerade klassdiagram går att hitta i bilagan.

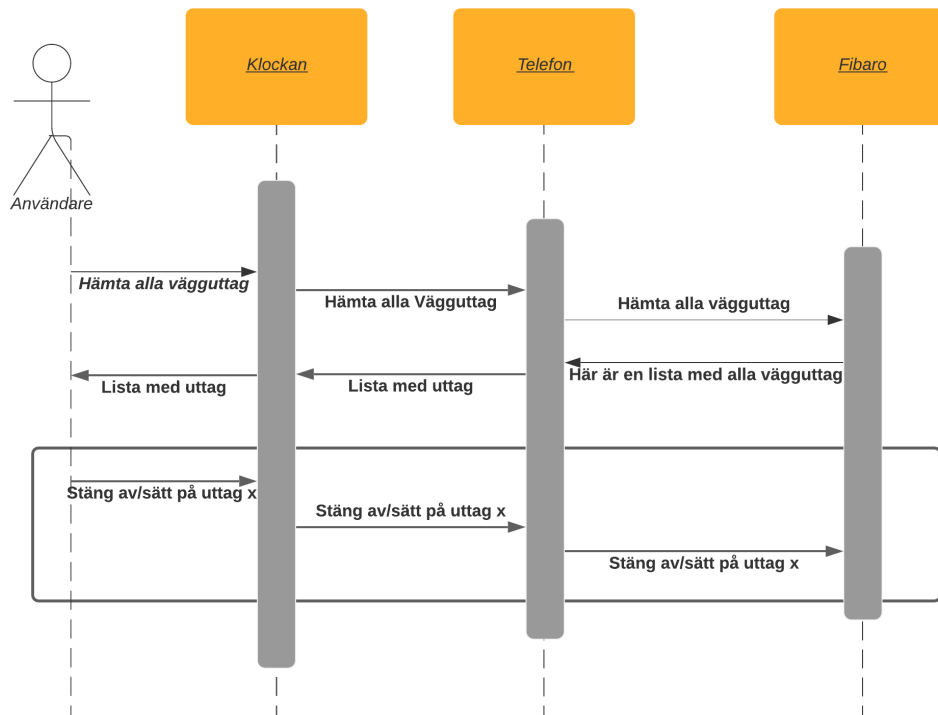
## 3.2 Dynamisk design

I den här sektionen kommer vi att kika på fyra olika sekvensdiagram som visar flödet för några av de funktioner som har implementerats i tjänsten.



Figur 5: Hämta och håll reda på vitalvärden

Vid app start så kommunicerar appen med klockan och begär rättigheter för att få använda sensorer som t.ex hjärtfrekvensen. Detta visas för användaren som sen får välja ifall vår applikation får använda dessa tjänster. När väl användaren har godkänt så skapas ett så kallad workout session (HKWorkoutSession). Under workout sessionet så kommunicerar healthstore med appen och skickar session och builder events till appen. Dessa event skapas då healthstore får in ny data då workouten är aktiverad. Datan tas in i builder event funktioner och sparas sedan i health Store klassen och skickas därefter till vyn för att visas till användaren.



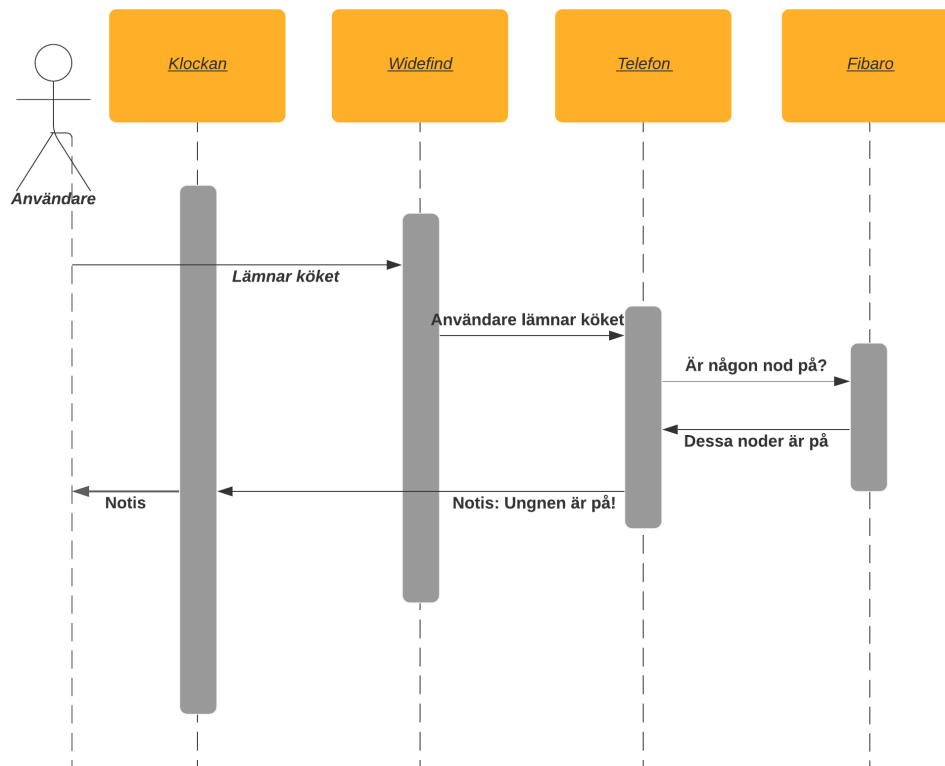
Figur 6: Hämta en lista med eluttag och sätt på/av

Diagrammet ovan visar ett exempel på hur användaren styr Z-wave sensorer i hemmet.

Först väljer användaren att se alla uttag som finns i det lokala nätverket. Anropet går via telefonen som frågar Fibaro enheten om alla binära switchar som är registrerade som eluttag. Samtidigt som alla noder hämtas kollar nätverket även vilka av dessa som är av och vilka som är på. Informationen skickas sedan tillbaka till användaren via klockan med hjälp av en vy som visar en lista med av/på knappar. Nu kan användaren välja att stänga av eller sätta på uttagen.

Så som systemet är uppbyggt för tillfället, så finns det inget som indikerar att en tredje part har ändrat ett värde på en nod medans användaren är i vyn. Användaren måste gå ur vyn och in igen för att få den uppdaterad. Då alla beräkningar och anrop görs i varje enskild användares mobiltelefon, kan inte ett övergripande observeranrop göras. I kapitel 6.1 beskrivs ett framtida system där en lokal server används istället för telefonen. I den nya modellen går all trafik genom den lokala servern istället. När servern får ett tillbakaaanrop från Fibaro kan den notifiera alla klockor att uppdatera sin vy istället för bara en.

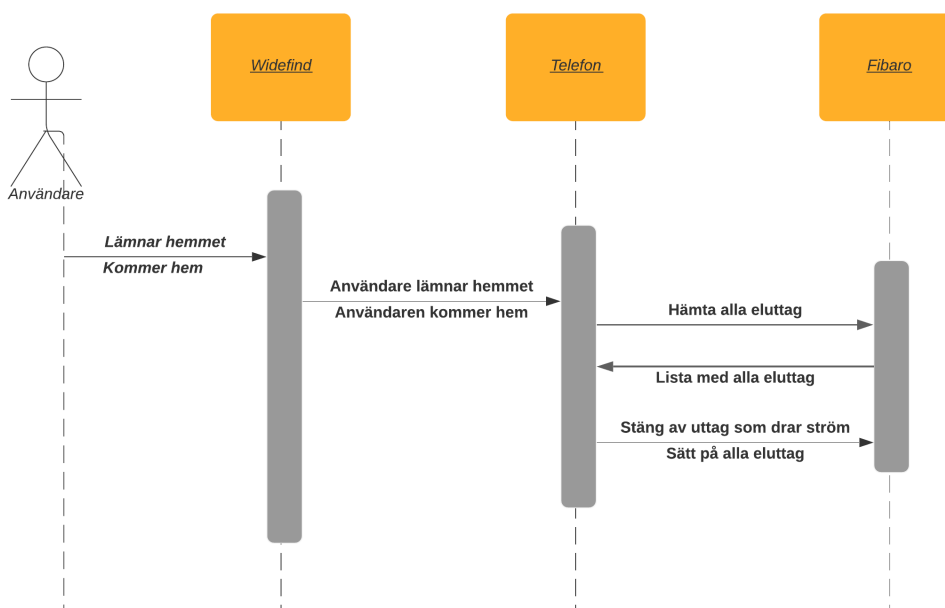
Detta är en övergripande design för hur noder ska kunna vissas i klockan. Designen används för alla interaktiva funktioner som klockan ska kunna göra med Fibaro och Philips HUE.



Figur 7: Skicka notiser till användaren

Notiser är en av de största styrkorna med en smart watch och tjänsten utnyttjar denna styrka genom att varna användaren när, t.ex ugnen är på när användaren lämnar köket.

Innan systemet körs igång måste en administratör markera upp olika delar av hemmet med hjälp av Widefinds koordinatsystem, t.ex kök, sovrum eller toalett. Widefind håller reda på vart användare befinner sig i hemmet med hjälp av Widefind taggen som användaren bär på. Om taggen lämnar köket informerar Widefind telefonen att ett eventet har skett. Ett anrop till FHC3 görs för att hämta nodinformation i rummet som lämnats. Telefonen processar responsen från FHC3 och utvärderar om en enhet som kan utgöra fara är igång. Om t.ex ugnen är på, kommer telefonen skicka till klockan att notifiera användaren om ugnens status.



Figur 8: Stäng av alla eluttag som drar ström när användaren lämnar hemmet/rummet.



Ett effektivt sätt att spara både pengar och bli mer grön i vardagen är att stänga av alla apparater när ingen befinner sig i hemmet eller rummet. Därför har en funktion som kollar om användaren lämnar hemmet implementerats. Precis som i förra diagrammet så märker Widefind klienten av att användaren lämnar hemmet. Detta skickas till telefonen som frågar FHC3 om en lista med alla binära switchar. Telefonen processar svarsmeddelandet från FHC3 och kollar om det är någon av de binära switcharna som drar ström, om de gör det meddelas FHC3 att stänga av dessa switcharna. De avstängda binära switcharna sparas lokalt i telefonen och återaktiveras då användaren återanländer till sitt smarta hem.

Om man vill vidareutvecklar denna funktion och man implementerar den nya systemmodellen i kapitel 6.1, skulle ett exempel på förbättring vara att klassificera noderna i den lokala servern. I klassificeringen kan man lägga till ytterligare information om noden som kanske inte finns tillgänglig i Fibaro eller Philips HUE. Ett exempel skulle kunna vara att sätta en riskfaktor för att avgöra om det är säkert att slå på noden. Om en nod har fått en hög riskfaktor måste användare godkänna att den startas vid hemkomst.

## 4 Resultat

### 4.1 Leverans

Det som har levererats är en tjänst med en kompanjonsapplikation med ett användargränssnitt på en Apple Watch Series 6 med tillhörande iPhone 12. För att avlasta klockan från all databeräkning så hanterar mobiltelefonen all kommunikation mellan applikationen och de olika sensorerna utplacerade i det smarta hemmet. För att hantera kopplingen mellan mobiltelefonen och klockan används Apples inbyggda API, WCsession.

I användargränssnittet kan noderna i hushållet styras så som på- och avslagning, hämta nodernas status, på så sätt se om lampor eller väggkontakter är aktiva eller inaktiva. All nodrapportering och styrning sker via Fibaro Homesystem samt Philip Hue bridge.

Applikationen hämtar hälsovärden på användaren via "HeathKit store" vilket är Apples inbyggda API för hämtning av sensordata så som puls, stegräkning vilken sedan sparas lokalt på klockan.

Tjänsten erbjuder platsspecifik styrning samt påminnelser från noder om användaren ex. lämnar rummet eller lägenheten. Platsdatan fås genom positioneringssystemet Widefind.

### 4.2 Testning

#### 4.2.1 Regressionrisker

Då systemet konstant kommer få input från diverse sensorer kan race conditions bli en regressions faktor - både lokalt och avlägset. Mobilen som är den centrala beräkningsenheten måste ta emot och processa alla inputs från klockan samt widefind brokern. Om då inputsen inte hinner processas och skickas till klockan för att visa event och förändringar, eller uppdatera tillstånd för de olika sensor typerna uppkopplad i de smarta hemmet uppstår problem. Att klockan visar korrekt och uppdaterad information och att aktioner via klockan reflekteras i de utplacerade sensorerna inom rimlig tid är grunden i vår applikation; därav blir race conditions en stor risk faktor.

Fibaros kontroll enhet (FHC) som kommer att sköta kommunikationen mellan z-wave sensorerna och den mobila enheten kan även få regressions problem i form utav race conditions. Men då vi primärt fokuserar på att utveckla en enhet inom labbmiljö ser vi ingen större risk för race conditions mellan FHC och de utplacerade sensorerna. För ut skalning av systemet där fler användare befinner sig i laborationsmiljön är kommunikationen mellan FHC och de olika sensorerna en större regressions faktor då fler anrop sker till FHC kontrollern.

I ett stort utvecklingsteam med många enheter som kommunicerar finns stor risk för avmaskerade regressions problem. Då vi är ett relativt litet utvecklingsteam (4 medlemmar) och kommer därav inte hinna producera världens mest invecklade system så ser vi inget större problem i att undvika avmaskerade regressions problem.

#### 4.2.2 Strategi för regressionstester

Som benämns vid regressionriskerna så kan race condition vara problem när flera enheter kommunicera med FHC. Ifall flera enheter försöker göra en förändring samtidigt, kan det leda till serier av events som inte är som de brukar. Vilket kan göra att system hamnar i ett konstigt state, till exempel man försöker sätta på en lampa men den stängs av istället, eller i värsta fall krashar. Strategin för att både konstatera att detta inte händer är att följa flödet av systemet (Traceability) samt kolla hur det uppstår (Change analysis). Genom att logga beteendet av varje state av systemet så kan vi upptäcka vart det gick snett någonstans samt även jämföra med tidigare serier av event:s som anses som bra states och göra en jämförelse, för att se vart det gick fel. Då vi har tillgång till data strukturen kan vi även se vilka andra delar av systemet som kan bli berörda av förändringen av systemets beteende (remote regression).

Eftersom systemet som syns i diagram 2, där mycket av all data och logik sker på andra ställen så är säkerhetsåtgärder viktiga att vidta. Vid systemuppdateringar så är det viktigt att hitta buggarna tidigt så att man snabbt kan agera innan konsekvenserna blir för stora. Genom att sätta upp testing på logiken för att se så att all funktionalitet gör som den ska kommer att underlätta detta.

### 4.2.3 Utvecklingsbeskrivning

Ett krav på produkten är att användaren ska kunna styra sitt smarta hem med hjälp av sin smarta klocka. Detta är en teknologi som har funnits några år redan och har ett flertal olika system. Miljön som produkten ska användas i kommer att använda sig av systemet Fibaro.

Fibaro använder sig av ett REST API system som returnerar ett JSON format. Med hjälp av det här systemet kommer vi att bygga ett GUI som kan lista upp alla smarta produkter i hemmet. Vi kommer att kalla dessa produkter för noder. Beroende på vad för slags nod det är, så kommer de att kunna göra olika funktioner. T.ex. en lampa ska gå att dimmra, ändra färg eller bara stänga av/ sätta på. Andra exempel kan vara ett larm, en spis eller dörrklocka.

För att få en bra överblick av vad som måste göras för att sätta upp detta system i den Iwatch:en har följande exempel på User story tagits fram. Storyn inkluderar olika task för att sätta upp systemet, tidsuppskattning, beroenden och prioritet.

Story ID	Story	Task ID	Task Risk: 1 = low, 10 = High	Tidsuppskattning (Story)	Använd tid (story)	Tidsuppskattning (task)	Använd tid (task)	Beroenden	Risk	Prioritet
1	<b>Hantering av smarta produkter</b> <b>Användning:</b> Användaren ska kunna styra olika smarta produkter i hemmet. <b>Önskade egenskaper:</b> Användaren ska kunna få en lista på de smarta produkter som är kopplade till det lokala nätverket. Dessa produkter beroende på vad det är har olika funktionalitet som varierar. T.ex. en lampa ska gå att stänga av och sätta på, men även gå att dimmra. <b>Testfall:</b> Testa om det går att kommunicera med alla noder på det sätt som är tänkt. Stresstesta för att se om systemet hakar upp sig när man gör flera funktioner i snabb följd.	1.1	<b>Att göra:</b> Sätt upp och lära sig Fibaro <b>Risk: 1</b>	40 tim		10 tim		N/A	1 + 1 + 1 + 1 / 4 = 1	Medium
		1.2	<b>Att göra:</b> Hämta noder i det lokala nätverket <b>Risk: 1</b>			10 tim		N/A		
		1.3	<b>Att göra:</b> Sätta på / Stänga av enheter <b>Risk: 1</b>			10 tim		1 2		
		1.4	<b>Att göra:</b> Dimmer funktion <b>Risk: 1</b>			10 tim		1 2		

Figur 9: User Story: Fibaro

Apple watch:en kommer automatiskt skicka sensordata till en databas i mobiltelefonen. För redundans kan en centraliserad server upprättas om mobiltelefon inte kan nås. Datan kan sedan lagras för att möjliggöra till exempel en hälsöhistorik. Smart watchen har även möjligheten att upptäcka om personen faller, så kan z-led accelerometern, känna accelerationen och sedan skicka ut en varning till de som berörs.

Klockan har även en sensor med grönt ledljus kopplad tillsammans med en ljuskänslig fotodiod för att mäta mängden blod som flödar genom handleden. I Apples egna programmeringsprogram och språk, finns färdiga funktioner som kan kalla på sensorerna för att sedan lagras.

Nedan är en user story på processen för att få en överblick på sensordatan.

Prioritet	Risk	Beroenden	Använd tid (task)	Tidsuppskattning (task)	Använd tid (story)	Tidsuppskattning (story)	Task	Task ID	Story	Story ID
Kritisk	3	N/A	TBD	AA=8	TBD	32h	<b>Att göra:</b> Få sensorn att skicka de olika datan <b>Risk:</b> Hög / 3	2.1	<b>Titel:</b> Hämta data från sensorer i Smart Watch <b>Avsedd användning:</b> Få vitalparametrar för historik samt förändring i z-led vid fall <b>Testfall:</b> Skicka datan till en databas och skriva ut värdena och jämföra de med verklig mätning	2
		N/A	TBD	BB=8			<b>Att göra:</b> Få accelerometer att skicka förändringar i z-led <b>Risk:</b> Hög / 3	2.2		
		2.1 2.2	TBD	CC=8			<b>Att göra:</b> Skapa och lagra datan i en databas <b>Risk:</b> Medium / 3	2.3		

Figur 10: User Story: Senordata

#### 4.2.4 Teststrategi

Kommunikationen mellan applikationens klienter och de olika kontroll enheterna i det smarta hemmet har skett via black box testning där HTTP request meddelanden analyserats med hjälp av *wireshark*.

Whitebox testningen har till största dels bestått av utskrifter som håller reda på systemet nuvarande state, och de är så vi har jobbat under projektets gång. Med hjälp av Xcode output window så kan vi följa exekverings flödet internt, detta hjälper oss att veta vart någonstans i system felet uppstår. Eftersom systemet inte är så stort har denna teknik fungerat bra och varit lätt att utföra. Med hjälp av simulatorn så kan vi även se ifall det antagande resultatet blivit uppfyllt. Vi har fokuserat på att bygga ihop delarna en i taget och testat dem genom att isolera funktionalitet.

En förbättring hade varit att skapa enhetstestning till systemet. Med enhetstestning kan vi förlita oss på att systemet kan testas efter flera ändringar har implementerats. Detta ger en större säkerhet och garanti att systemet fungerar som det är tänkt. Vilket i sin tur ger en bättre överblick och kontroll av flödet i systemet.[9]

## 5 Etisk bedömning

Då tjänsten i detta projekt kan hämta vitalparametrar av användaren och övervakning kan ske, skulle det kunna ses som ett etiskt dilemma.

- Skulle en organisation kunna använda sig av detta data för att på nått sätt få inflytande över användaren?
- Skulle de kunna använda sig av datan för sin egen vinning?
- Om dessa problem skulle komma upp, vem ska då stå som ansvarig för detta och vad behöver göras för att motverka problemet?

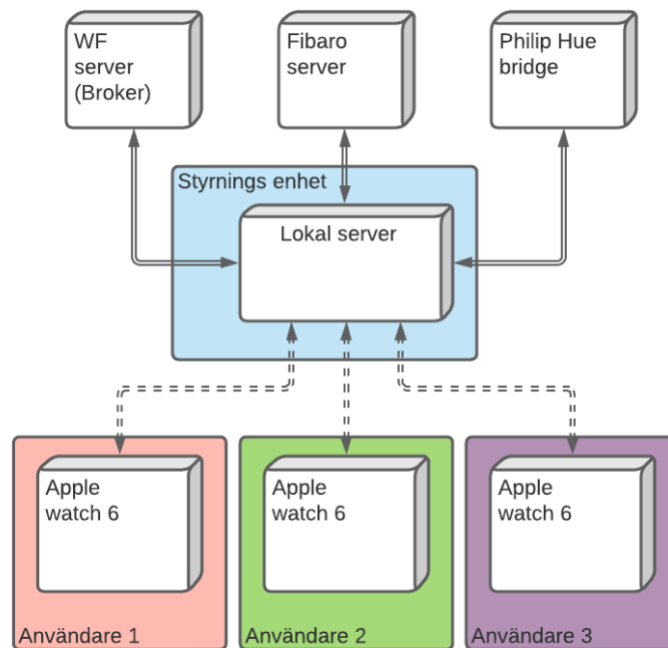
För tillfället så sparar tjänsten all data som hämtas från sensorerna i klockan på användarens iCloud konto. Detta medför att användaren måste godkänna Apples GDPR regler och kan använda sig av deras skydd. Då denna tjänst är tänkt att användas personer med funktionsnedsättning så skulle ett krav kunna vara att en vårdgivare eller en anhörig måste godkänner användandet av tjänsten och alla sensorer som behöver sättas in i hemmet. Om inte vårdtagaren kan göra den bedömningen själv vill säga. För tillfället så måste man vara inom ett stängt nätverk och vara registrerad för att kunna styra systemen i det smarta hemmet, vilket bidrar till att systemet blir säkrare.

## 6 Slutsats och framtida arbete

Den utvecklade applikationen tillåter användaren att fjärrstyra *Z-wave* och *Zigbee* enheter samt automatiskt kontrollera dessa enheter via olika event med hjälp av *widefinds* positionering system. Användaren får även med hjälp utav applikationen hälsodata om sitt eget välmående. Valet av att använda oss utav *observer* design mönstret i kontrollern till de olika enheterna gör det enkelt för andra mjukvara utvecklare att integrera fler enheter till applikationen som kan styras via automatiska event från widefind klienten eller input från klockans interface. Från uppdragsgivarens beställning har alla sensorer integrerats med Apple watch enheten förutom *Double robotics* roboten.

### 6.1 Fortsatt utökad skalbarhet

Implementationen av system arkitekturen från figur (2) komplicerar skalbarheten av systemet. När ytterligare en användare introduceras till systemet kommer den behöva göra exakt lika många anrop till de olika sensor kontrollerna för att få åtkomst till de aktuella tillståndet från de olika sensorerna i de smarta hemmet. Detta i sin tur skapar en högre belastning på de lokala nätverket. En lösning till detta skulle vara att flytta all logik och beräkning från mobilen till en central server i lägenheten som håller reda på tillståndet av de olika sensorerna.



Figur 11: Ny design för ett effektivare, säkrare och mer skalbart system.

Med designen av systemet från figur (11) behöver inte datan hämtas från sensorernas kontrollenhet två gånger för att notifiera två användare om nuvarande status för sensorerna, endast ett anrop krävs från den centrala servern till de olika sensorkontrollerna för att notifiera de två användarna om sensorernas nuvarande tillstånd.

En annan fördel för användaren genom att flytta arkitekturen till en lokal server inriktad design är att användarna behöver inte konfigurera anslutningen till varje enskild sensors kontrollsystem, utan de görs via servern vilket skapar en mer *plug and play* upplevelse för användaren som nu endast måste autentisera sig mot den lokala servern. Med att användaren nu med den nya designen endast måste autentisera sig mot den lokala servern skapar det även ett säkrare system där en administratör har möjligheten att granska anropen via den lokala servern och filtrera anrop mot de olika sensorerna som kan vara av skadande natur. Ytterligare en fördel med att välja den nya skalbara design är att mobilen i systemet blir överflödig - systemet kan styras via klockan utan att behöva interagera med mobilen då klockan är av modell *series 6* som tillåter applikationer att köras utan koppling till tillhörande mobil [4].

## Referenser

- [1] Widefind, 25-02-2021  
<https://www.widefind.se/>
- [2] Fibaro, 25-02-2021  
<https://www.fibaro.com/se/>
- [3] Philips HUE, 01-03-2021  
<https://www.philips-hue.com/>
- [4] Apple inc: Creating Independent watchOS Apps, 03-03-2021  
[https://developer.apple.com/documentation/watchkit/creating\\_independent\\_watchos\\_apps](https://developer.apple.com/documentation/watchkit/creating_independent_watchos_apps)
- [5] Bill Wilson: eLearning root cause analys, 22-01-2021,  
<https://www.bill-wilson.net/root-cause-analysis/rca-tools/change-analysis>
- [6] Science direct: Overview of tracability, 23-01-2021  
<https://www.sciencedirect.com/topics/agricultural-and-biological-sciences/traceability>
- [7] Science direct: Software testing, 22-01-2021,  
<https://www.softwaretestinghelp.com/risk-management-during-test-planning-risk-based-testing/>
- [8] Somphy: Smarthem, 17-03-2021,  
[https://www.mynewsdesk.com/se/somfy-sverige/blog\\_posts/daerfoer-aer-ett-smart-hem-bra-foer-miljoen-och](https://www.mynewsdesk.com/se/somfy-sverige/blog_posts/daerfoer-aer-ett-smart-hem-bra-foer-miljoen-och)
- [9] KTH: Software testing, 17-03-2021,  
<https://www.kth.se/social/files/57e28091f27654494845cc18/Enhetstestning.pdf>

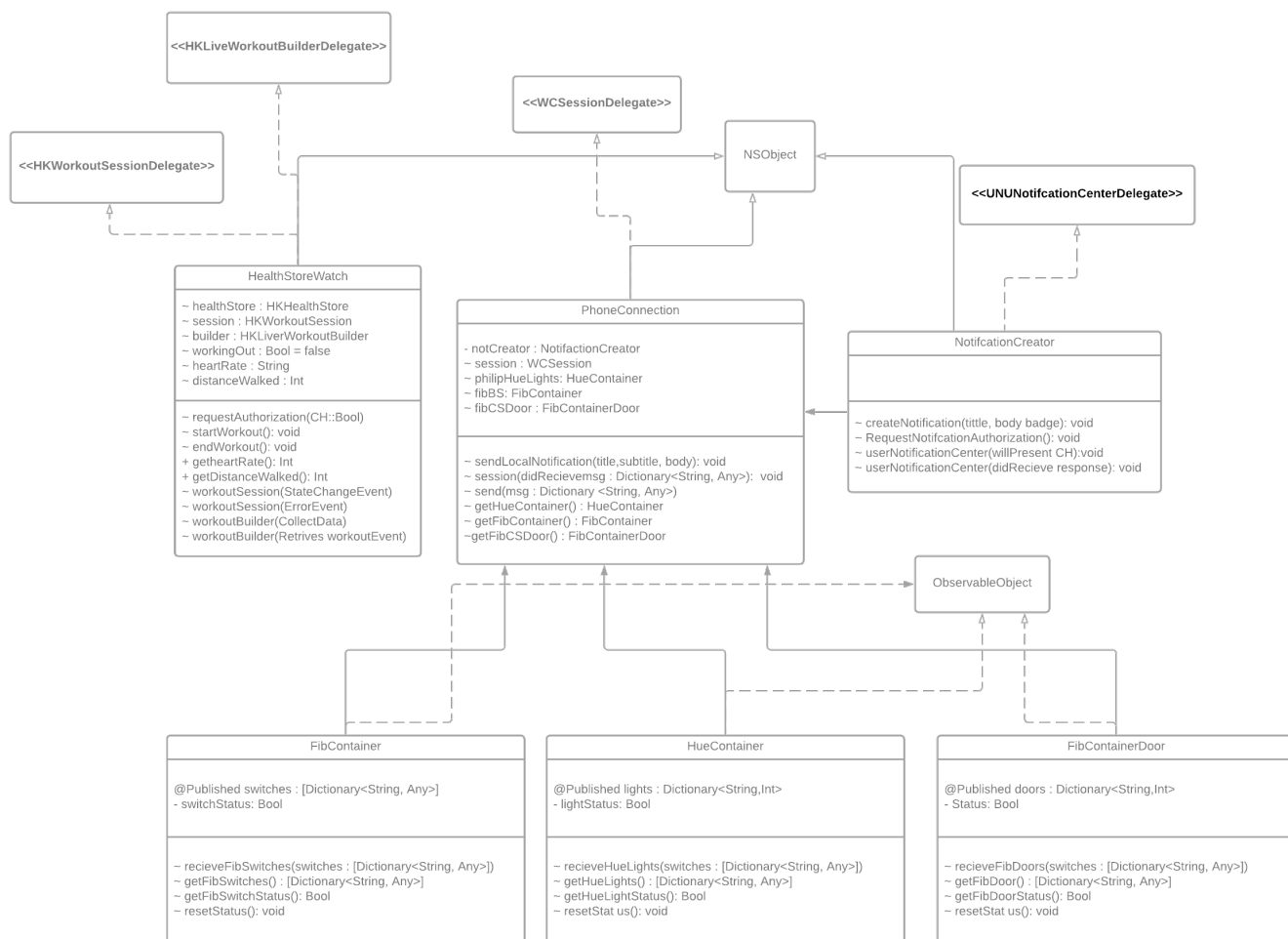
# Bilaga

## Länkar

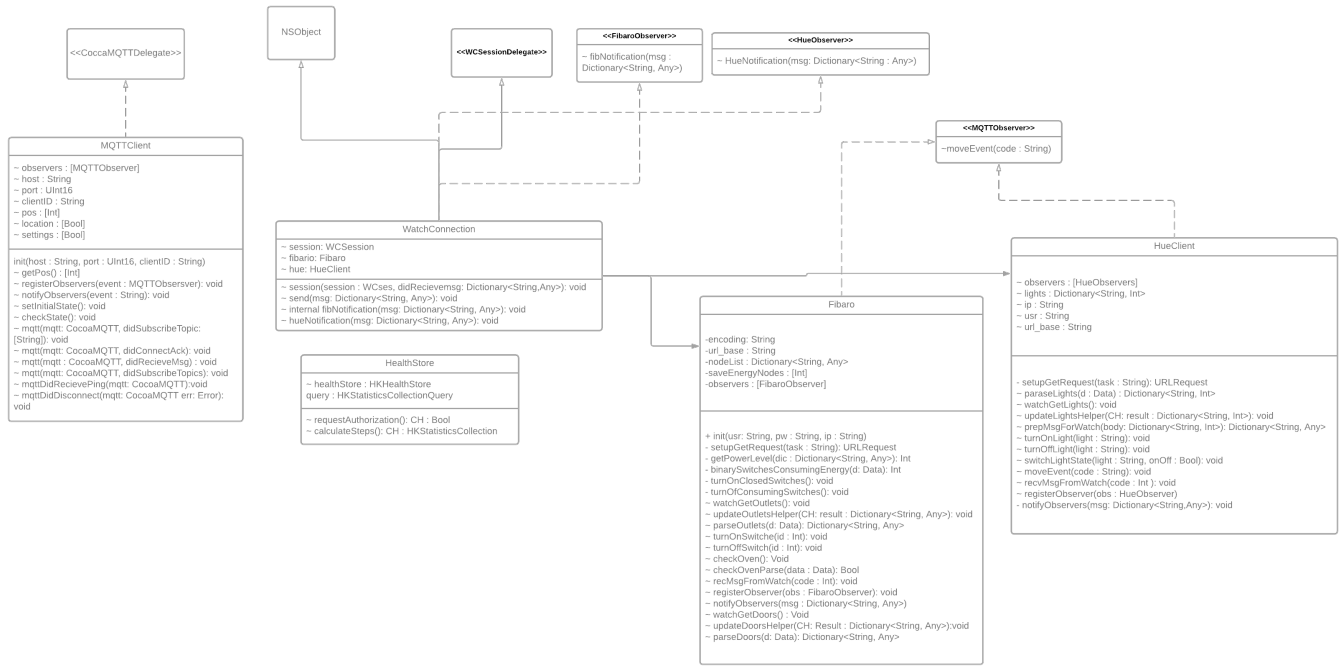
Länk till GitHub repository:

<https://github.com/pheriksson/IwatchHomeSolution>

## Klassdiagram



Figur 12: Fullstændigt klock modul klassdiagram



Figur 13: Fullständigst telefon modul klassdiagram