KVALITETSSÄKRING AV INFORMATION I STÖRRE ORGANISATIONER

Johan Holmström, Stefan Lindén, Anders Svensson

2019:14

Datum för godkännande: 23.05.2019
Handledare: Agneta Eriksson-Granskog

EXAMENSARBETE

Högskolan på Åland

Utbildningsprogram:	IT-programmet
Författare:	Johan Holmström, Stefan Lindén, Anders Svensson
Arbetets namn:	Kvalitetssäkring av information i större organisationer
Handledare:	Agneta Eriksson-Granskog
Uppdragsgivare:	Ålands hälso- och sjukvård

Abstrakt:

I vårt examensarbete skriver vi om hur vi utvecklar en lösning för att öka kvaliteten vid digital hantering av personuppgifter i en större organisation.

I en större organisation med många system som hanterar personuppgifter är det lätt att det blir diskrepans mellan informationen i de olika systemen om förändringar av information sköts manuellt. Syftet med arbetet har varit att ta fram och implementera en lösning för att automatisera synkronisering av utvald information mellan system som hanterar personuppgifter.

För att uppnå syftet har vi tagit fram en lösning med ett centralt datalager, utifrån vilket förändringar av personuppgifter synkroniseras med kopplade system.

I uppsatsen beskriver vi hur vi implementerat datalagret med tillhörande gränssnitt och integrerat det primära verksamhetssystemet som hanterar störst mängd personuppgifter.

Nyckelord (sökord):

Datalager, masterdata

Högskolans serienummer:	ISSN:	Språk:	Sidantal:
2019:14	1458-1531	Svenska	42

Inlämningsdatum:	Presentationsdatum:	Datum för godkännande:
23.05.2019	15.05.2019	23.05.2019

DEGREE THESIS
Åland University of Applied Sciences

Study program:	IT – Information Technology
Author:	Johan Holmström, Stefan Lindén, Anders Svensson
Title:	Quality assurance of information in larger organizations
Academic Supervisor:	Agneta Eriksson-Granskog
Technical Supervisor:	Ålands hälso- och sjukvård

Abstract:

In our thesis, we write about how we develop a solution to increase the quality of digital processing of personal data in a larger organization.

In a larger organization with many systems that handle personal data, it is easy that there is discrepancy between the information in the different systems if changes in information are handled manually. The purpose of the work has been to develop and implement a solution to automate synchronization of selected information between systems that handle personal data.

To achieve this goal, we have developed a solution with a central data layer, from which changes in personal data synchronizes with linked systems.

In this thesis, we will describe how we have implemented the data layer with its associated interface and integrated the primary business system that handles the largest amount of personal data.

Key words:

Data layer, master data

Serial number:	ISSN:	Language:	Number of pages:
2019:14	1458-1531	Swedish	42

Handed in:	Date of presentation:	Approved on:
23.05.2019	15.05.2019	23.05.2019

INNEHÅLLSFÖRTECKNING

1. INLEDNING

Inom sjukvården har man idag många olika system som hanterar patienters personuppgifter. Exempel på sådana system är olika journal- och ekonomisystem. De flesta av dessa system har olika leverantörer och egna databaser. En del av den information som lagras i de olika systemen är unik för det enskilda systemet, men vissa uppgifter om patienter är lika i alla systemen. Exempel på sådana uppgifter är namn, personbeteckning, adress etc.

Ett problem med att samma uppgifter om en patient förekommer i flera helt åtskilda databaser är att det lätt blir problem vid förändring av den informationen. I praktiken sköts de olika systemen av organisatoriskt skilda delar av sjukvården, vilket innebär att det lätt blir diskrepans mellan de olika systemens information över tid om förändringar sköts manuellt av olika personer i olika delar av en organisation.

Nedan följer ett möjligt scenario som kan uppstå när personuppgifterna i olika system inte stämmer överens:

En patient informerar om ändrad adress vid ett besök till sjukvården. Denna information tas emot av en del av organisationen som hanterar det primära journalsystemet. En kallelse går ut för ett besök och kommer fram till rätt adress, men när fakturan för besöket skickas adresseras den utifrån information som finns i ekonomisystemet som har den gamla adressen och kommer således fel.

Förutom olägenheten för patienten i fråga att fakturan kommer rätt först då den gått till inkasso, så kan fakturan i sig avslöja hälsoinformation av känslig karaktär, exempelvis att den avser ett besök till psykiatriska kliniken.

1.1 Syfte

Syftet med detta examensarbete är att ta fram och implementera en lösning för att åtgärda problem med låg kvalitet på data i system som hanterar patienters personuppgifter inom

sjukvården. Detta primärt ur ett tekniskt perspektiv men även ur ett administrativt perspektiv, eftersom det krävs för att lösningen ska fungera i praktiken.

1.2 Metod

Som metod har vi använt oss av ett agilt arbetssätt där mindre delar specificerats för att sedan realiseras under en begränsad tidsperiod. Utgående från resultatet av den då implementerade delen har nästa del valts för design och implementation. Till hjälp i projektet har vi haft en underleverantör som skött den huvudsakliga programmeringen utifrån vår kravställning och den gemensamt framdiskuterade designen.

1.2.1 Huvudsakliga designprinciper

Lösningen ska använda sig av principer för ”masterdata”; data ska ändras endast på ett ställe och den förändringen av data ska propagera till andra delar i systemet av system där samma data används.

Administrationen av den centrala databasen ska ske genom ett så naturligt gränssnitt som möjligt för de som använder sig av systemen. Det innebär i praktiken att om man normalt använder sig av det primära journalsystemets gränssnitt för att hantera patienters personuppgifter ska man även i fortsättningen använda samma gränssnitt för hantering av data som ägs av den centrala personuppgiftsdatabasen. Genom studier i informationshantering och arbetserfarenhet från olika integrationsprojekt har vi skaffat oss kunskap för detta arbete

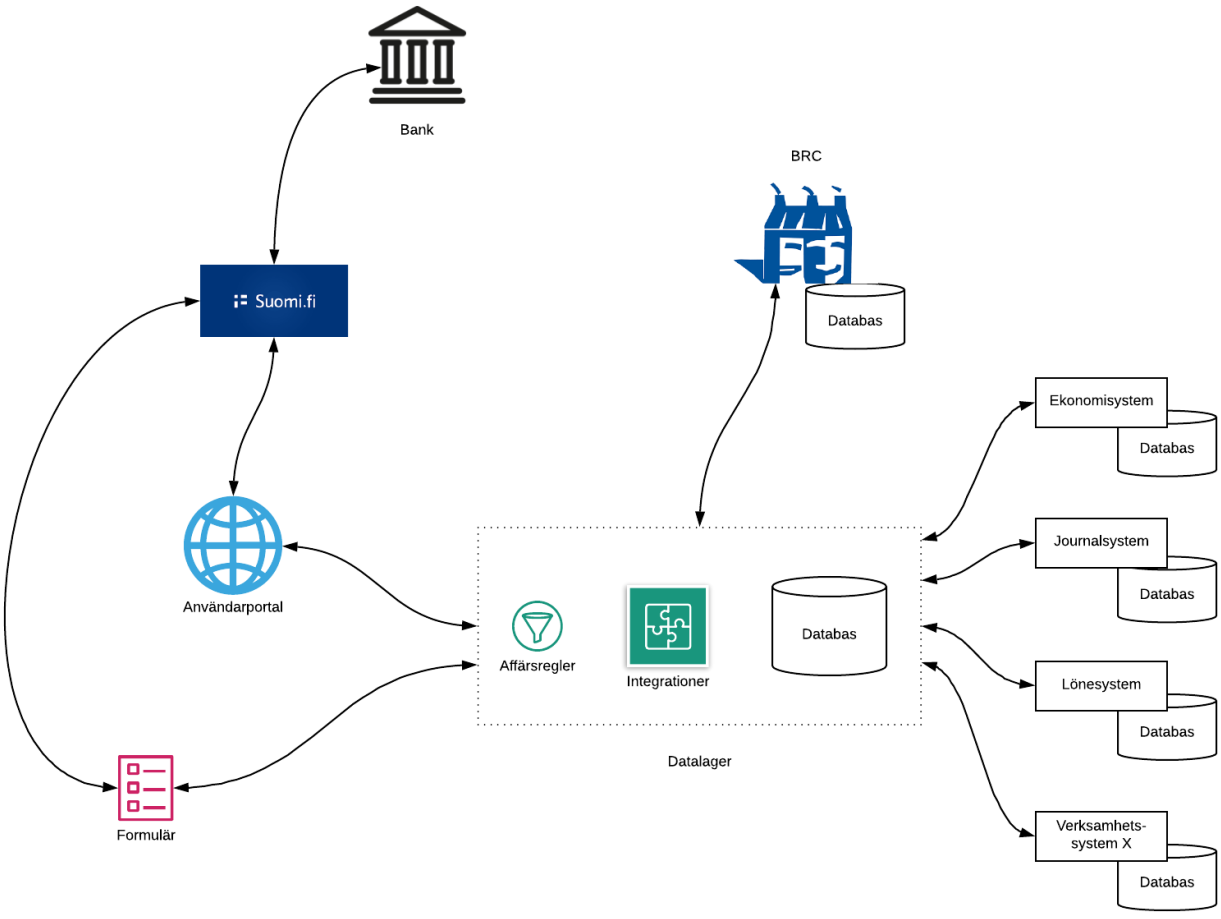
1.3 Avgränsningar

Examensarbetet behandlar den centrala databasen (datalagret), dess gränssnitt och kopplingar till några kringliggande system. Att möjliggöra gränssnitt och implementera kopplingar till alla system som hanterar personuppgifter inom sjukvården är ett arbete som skulle ta flera år, vilket därför faller utanför avgränsningen av detta projekt. Då vissa saker är av känslig natur kommer vi inte att beskriva dessa närmare t.ex. vissa detaljer i API:er.

2. DATALAGRET

Hjärtat i lösningen med centralt hanterade data är den databas som innehåller patienternas personuppgifter, det så kallade datalagret. Till datalagret kopplas alla de system inom organisationen som hanterar personuppgifter. I de fall där två eller fler system hanterar samma information om en person ska den informationen ägas av datalagret som sedan synkroniserar informationen till berörda system. Detta sätt att hantera uppdateringar och förändringar säkerställer att samma information är lika i alla kopplade system.

En översiktlig systemkarta med datalagret och integrerade system och tjänster finns i Figur 1.

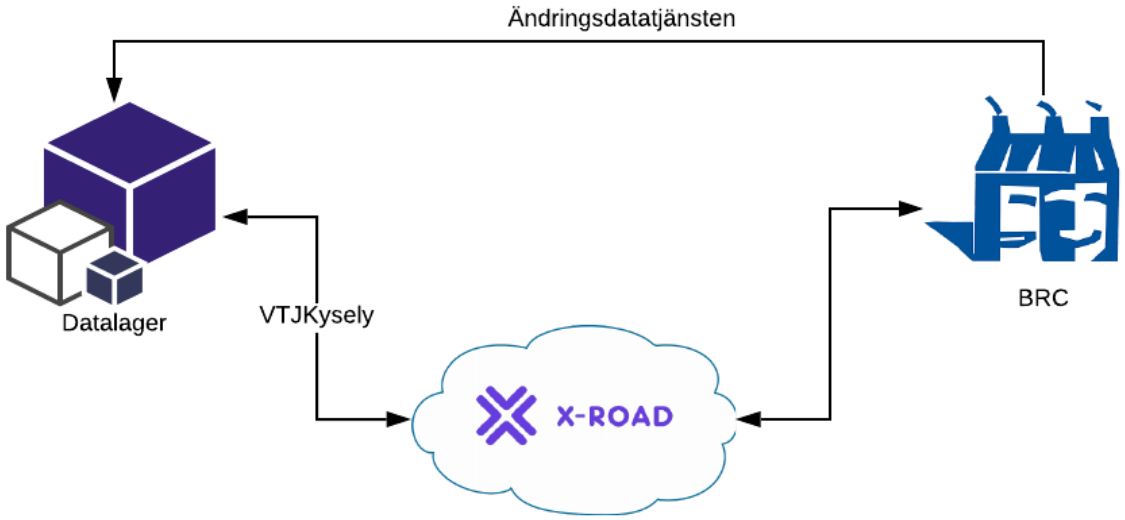


Figur 1 Datalager med integrerade system och tjänster.

2.1 Kvalitet

För att säkerställa att informationen i datalagret är så korrekt och uppdaterad som möjligt är datalagret kopplat till det nationella registret för personuppgifter som hanteras av Befolkningsregistercentralen (BRC).

Se Figur 2 nedan för BRC-kopplingen.



Figur 2 Koppling av datalagret till BRC

Det lokala datalagret är master internt för organisationen men relaterat till det nationella registret hos BRC är det senare master för de data som finns där. Exempelvis adress och relation mellan barn och förälder finns registrerat nationellt. Exempel på information som inte finns i det nationella registret är mobiltelefonnummer och vem eller vilka en enskild patient önskar ha som kontaktpersoner vid sjukdomsfall eller olycka. Denna information hanteras då av det lokala datalagret och synkroniseras således inte med det nationella.

Endast uppgifter om personer som är folkbokförda på Åland läses in från det nationella datalagret. Vid en första körning lästes all information i det urval av uppgifter som kan fås från

det nationella datalagret in. Uppdateringar sker därefter veckovis och då läses endast förändringar av data in.

Uppsättningen har även, vid behov, möjlighet att genom en direkt fråga till det nationella datalagret läsa in information om en person i realtid. Detta kan exempelvis vara aktuellt i de fall en patient inte behandlats hos ÅHS tidigare och inte är folkbokförd på Åland men som har en finsk personbeteckning.

En viktig aspekt vad det gäller säkringen av god kvalitet på informationen är att det ska vara otvetydigt var en korrigering av felaktig information ska göras. Detta säkerställs genom ovanstående sätt att hantera uppdateringar/korrigeringar. Relaterat till detta är även spårbarheten i hur den felaktiga informationen uppstod. Detta är viktigt att veta så det går att åtgärda källan till felet och detta så snabbt som möjligt. Hur denna spårbarhet möjliggörs i lösningen beskrivs i 2.3.10 Spårbarhet.

2.2 Informationssäkerhet

Lösningen vi tagit fram behandlar primärt information bestående av personuppgifter. Kraven på hur sådana uppgifter hanteras har länge varit strängare än kraven på behandling av övrig information. Dessa krav har även skärpts i och med EU:s Dataskyddsförordning som trädde i kraft 2018. Med detta i beaktande har vi designat och implementerat lösningen utgående från de generella principerna för god informationssäkerhet som återfinns i standarden ISO 27001 (Svenska institutet för standarder, 2017).

De huvudsakliga kvalitéerna på information som behöver beaktas för att uppnå en god informationssäkerhet är:

Riktighet

- Informationen ska vara skyddad mot förvanskning eller otillåten förändring.

Tillgänglighet

- Behörig användare ska få tillgång till den information som behövs för att utföra sina arbetsuppgifter.

Konfidentialitet

- Informationen ska vara skyddad mot obehörig åtkomst.

Spårbarhet

- Hantering av information ska kunna följas upp och inte gå att förneka i efterhand

Hur dessa kvaliteter på information uppnås tekniskt i lösningen beskrivs i 2.3 Teknik.

2.3 Teknik

Datalagret innehåller funktionalitet inom fyra områden:

- Integration mot befolkningsdatasystemet
- Integration mot andra system
- Funktionalitet för administratörer
- Batchfunktionalitet

2.3.1 Integration mot befolkningsdatasystemet

Befolkningsregistercentralen (BRC) upprätthåller och underhåller det nationella datalagret befolkningsdatasystemet samt tillhandahåller informationstjänster som baserar sig på befolkningsdatasystemet för myndigheter och företag.

Det är mot detta nationella datalager som ÅHS datalager integreras och erhåller sina masterdata för personuppgifterna.

Några av de tjänster som datalagret integrerar mot är:

- Ändringsdatatjänsten Muutostietopalvelu

Prenumeration på personuppgifter för personer bosatta inom valt område (Andringsdatatjanst, 2019)

- Befolkningsdatasystemets tillämpning

Direktförfrågan om personuppgifter baserat på en finsk personbeteckning, ej bundet till ett område (Vtjkysely, 2019)

- Suomi.fi-identifikation

Bankinloggning eller autentisering genom Mobil-ID eller elektroniskt ID på chip-kort (certifikatkort) som returnerar namn och personbeteckning för den inloggade (beroende på syfte och myndighetens rättigheter kan även adressuppgifter och uppgifter om spärrmarkering/hemlig adress hämtas vi denna tjänst) (Suomi-fi identifikation, 2019)

- Suomi.fi-fullmakter

Information om förmyndarskap och fullmaktstillstånd för personer och företag (Suomi-fi fullmakter, 2019)

- Suomi.fi-informationsled

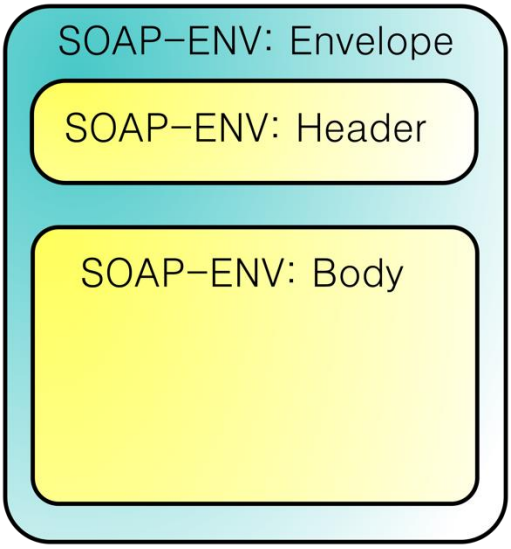
Standardiserat kommunikationsgränssnitt för data mellan organisationer (Informationsled, 2019)

Dessa tjänster utgör tillsammans en helhet för att ÅHS skall erhålla komplett information om sina patienter.

De tekniker som används vid integrationer är följande:

2.3.2 SOAP-webtjänst

En SOAP-webtjänst är ett XML-baserat gränssnitt som består av tre delar – SOAP Envelope, SOAP Header och SOAP Body. Se Figur 3 nedan.



Figur 3 SOAP-webtjänst

SOAP Envelope är det element som identifierar XML-dokumentet som en SOAP-förfrågan och som kapslar in hela webtjänstförfrågan.

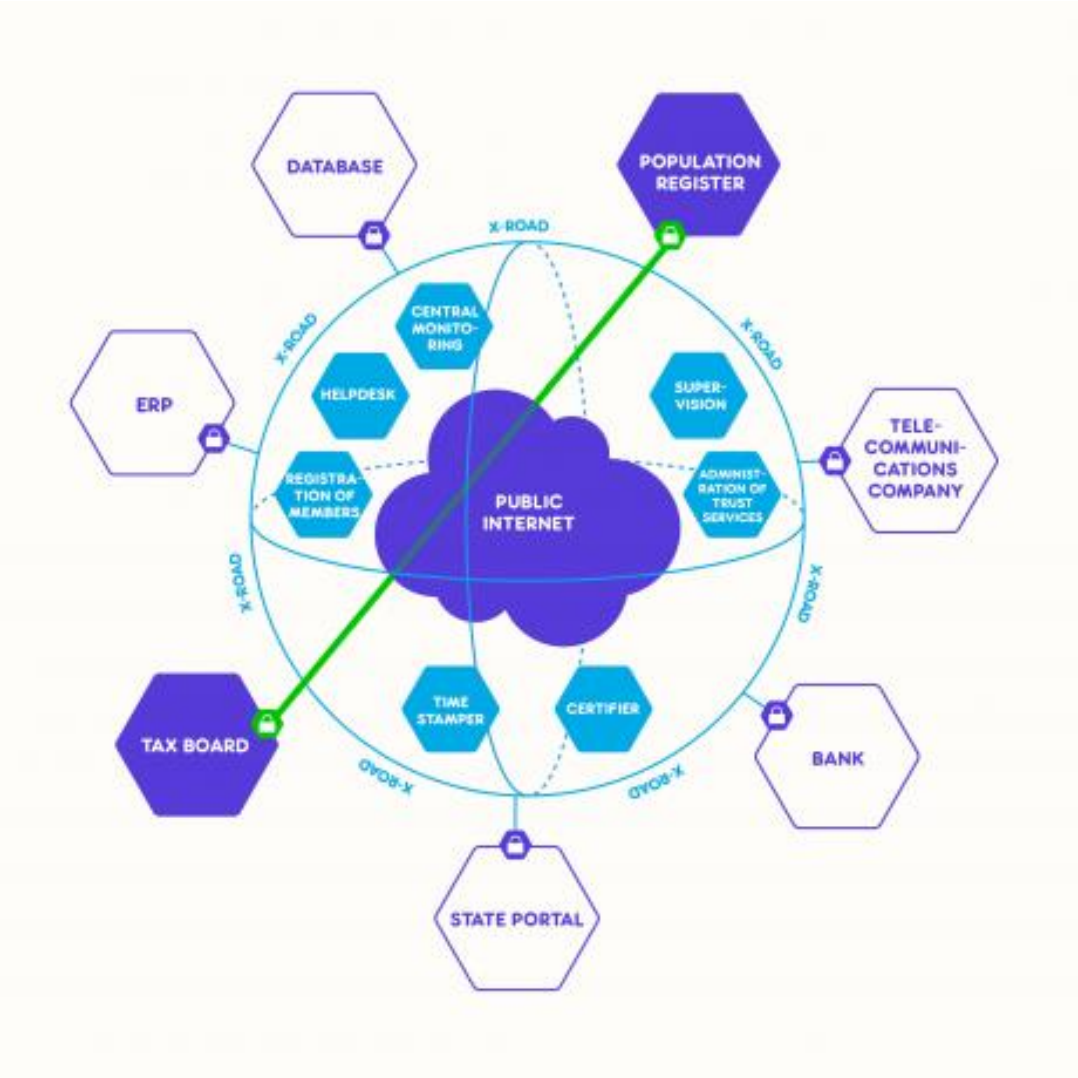
SOAP Header är den del av XML-dokumentet som innehåller autentisering och identifiering av klienten, I det här fallet datalagret.

SOAP Body är den del av XML-dokumentet som innehåller dom frågeparametrar som datalagret skicka för att begära information från webtjänsten.

Svaret från en SOAP-webtjänst kommer sedan i sin tur som ett nytt XML-dokument som datalagret tolkar för att kunna lagra informationen som returneras.

Suomi.fi-informationsleden (X-Road)

Suomi.fi-informationsleden erbjuder organisationer ett standardiserat sätt att överföra data både mellan offentliga och privata organisationers datasystem. Se Figur 4 för en schematisk bild av X-ROAD.



Figur 4 X-Road Kopplingsschema (X-Road, 2019)

Med Suomi.fi-informationsleden kan en organisation bygga upp säkra tjänstehelheter för medborgare, företag och myndigheter. Anslutningskatalogen som erbjuds som en del av Suomi.fi-informationsleden är uppdaterad katalog över anslutningarna till den nationella informationsleden. Syftet med anslutningskatalogen är att hjälpa den som producerar och genomför tjänster att utveckla effektivare elektroniska tjänster och stödja återanvändning av uppgifterna. I katalogen beskrivs elektroniska tjänster som tillåter att uppgifter som behandlas i

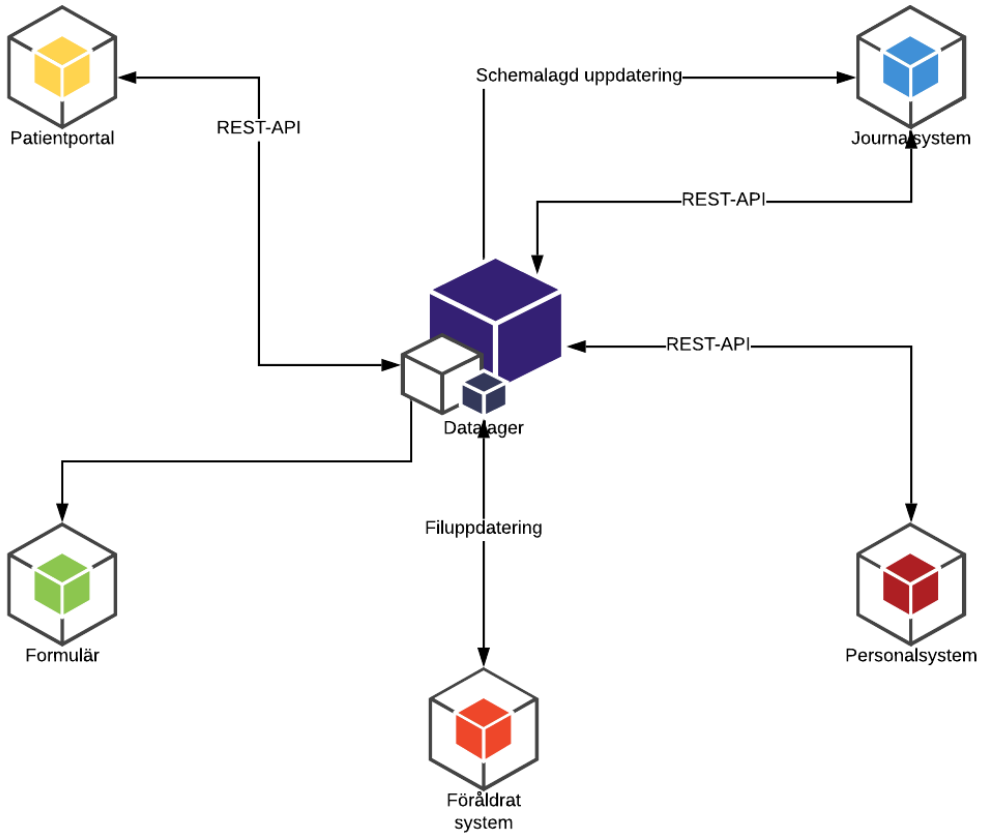
dessa tjänster också kan används i andra datasystem. I datalagret används Suomi.fi-informationsleden för att göra förfrågningar mot tillämpningen BDS-förfrågan (VTJkysely).

2.3.3 Filimport

Vissa äldre BRC-tjänster fungerar endast genom prenumeration av datafiler som hämtas via krypterad FTP-överföring och som sedan importeras i datalagret. Detta är dock något som på sikt troligen kommer att ersättas med X-Road eller SOAP-webtjänst.

2.3.4 Integration med andra system

Vid integration mot andra system behöver dessa ges en möjlighet att kommunicera mot datalagret och detta sker med hjälp av ett kommunikationsprotokoll som kallas för REST-API (Architectural Styles and the Design of Network-based Software Architectures, 2019). Även andra former av kommunikation kan användas vid integrationen, till exempel filutbyten. Se Figur 5 för en schematisk bild av exempel på integrationer.



Figur 5 Datalager - integrationer

REST-API är en förkortning av Representational State Transfer Application Programming Interface. Ett API är en specifikation av hur olika system och applikationer skall kommunicera med varandra och specifikationen definierar vilka funktioner som dessa system kan använda för att läsa och skriva data från eller till API:et.

REST-delen av begreppet REST-API beskriver hur funktionerna i API-delen kan nås via webbteknologi. Grunderna för REST-API är att man har ett antal unika adresser enligt Uniform Resource Identifier (URI) som refererar till en funktion i API:et.

Exempel på en sådan kan vara:

https://datalager.at.ahs/person/filter

Vidare använder man sig av några olika metoder när man adresserar en REST-API-URI; dessa är bland annat POST, GET, PUT och DELETE.

POST skapar en post, GET hämtar en post, PUT uppdaterar (ersätter) en post och DELETE raderar en post. Observera att dessa metoder kan ha andra uppgifter, detta kan man styra i API-metoderna som står för logiken.

Nedanstående är exempel på REST-API-metoder i datalagret och som kan användas vid en integration mot datalagret.

Exempel på REST-API-metoder

Nr	**Metod**	**URL**	**Beskrivning**
1	GET	API_URL/person/filter	Hämtar en eller flera personer enligt ett filter
2	POST	API_URL/person	Lägger till en person
3	POST	API_URL/person/ice	Lägger till en persons anhörigkontaktuppgifter
4	POST	API_URL/person/get	Lägger till en persons anhörigkontaktuppgifter
5	POST	API_URL/person/get	Hämtar en person med personbeteckning som nyckel
6	POST	API_URL/person/history	Hämtar all historik för en person med personbeteckning som nyckel
7	PUT	API_URL/person	Uppdaterar en person
8	PUT	API_URL/person/addresses	Uppdaterar adresser för en person
9	PUT	API_URL/person/ice	Uppdaterar en persons anhörigkontaktuppgifter

Vi kan sammanfatta datalagrets metoder för integration genom att definiera att det skall ge anslutande system möjlighet att hämta och uppdatera en eller flera personers uppgifter med

REST-API-metoder samt ges möjlighet att schemalägga uppdateringskörningar för att hålla sina respektive databaser uppdaterade mot datalagret.

2.3.5 Funktionalitet för administratörer

Datalagret innehåller ett administrativt gränssnitt som har följande funktioner:

- Sökning av data
- Möjlighet att redigera metadata (data som inte finns hos BRC men som lagras om personer)
- Läsa användarloggar
- Hantera API-nycklar för de system som skall kommunicera med datalagret
- Hantera användare och roller för de personer som skall komma åt det administrativa gränssnittet
- Hantera övrig konfiguration, BRC-certifikat, inställningar för schemaläggning etc.

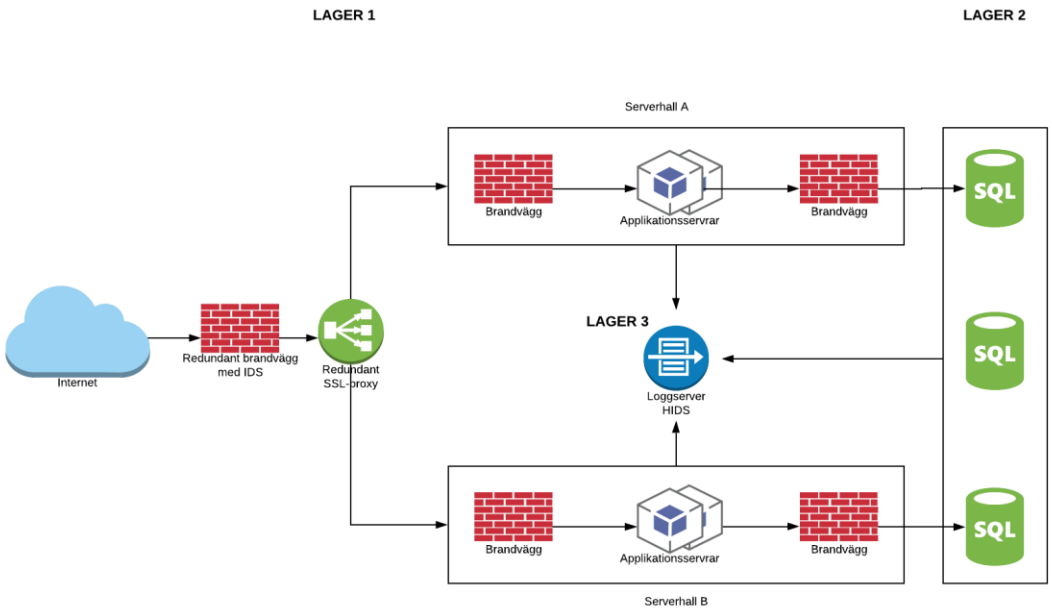
2.3.6 Funktioner för schemaläggning

Ett av huvudsyftena med datalagret är att de integrerade systemen skall kunna prenumerera på information från datalagret. Detta innebär en prenumerationslista (tabell) där datalagret har fullständig kontroll på vilka system som skall få uppdateringar på vilka personer samt exakt vilken information dessa system skall få tillgång till och när detta skall schemaläggas. Denna funktionalitet ger flera fördelar, dels så behöver de integrerade systemen inte hålla reda på om en persons uppgifter har uppdaterats eller inte, systemet kan helt enkelt fråga datalagret med jämna mellanrum efter en lista på de personer vars uppgifter har ändrats sedan systemet senast frågade efter uppdateringar. Ett annat sätt att hålla ett integrerat system uppdaterat är att datalagret signalerar det integrerade systemet att det finns uppdateringar att hämta och den tredje formen av integration är att datalagret ges tillstånd att uppdatera det integrerade systemets databas direkt vid en uppdatering av personinformation.

En annan fördel med prenumerationsfunktionaliteten är att dataskyddsförordningen (GDPR, General Data Protection Regulation) blir lättare att följa (The purposes and scope of the General Data Protection Regulation, 2019). Organisationen får på ett smidigt sätt en automatisk kartläggning om i vilka system en viss persons uppgifter finns lagrade och detta hjälper då organisationen att vid en förfrågan från en patient ta fram all information som lagrats om den patienten.

2.3.7 Systemuppbyggnad

Datalagret är uppbyggt i flera av varandra oberoende lager. Se Figur 6 nedan



Figur 6 Datalager - uppbyggnad

I det första lagret finns applikations- och proxyservrarna som har i uppgift att kommunicera mot de integrerade systemen samt sköta all systemlogik.

I det första lagret finns även administrationsservern som tillhandahåller alla de administrativa funktionerna som administratörerna använder för att underhålla datalagret.

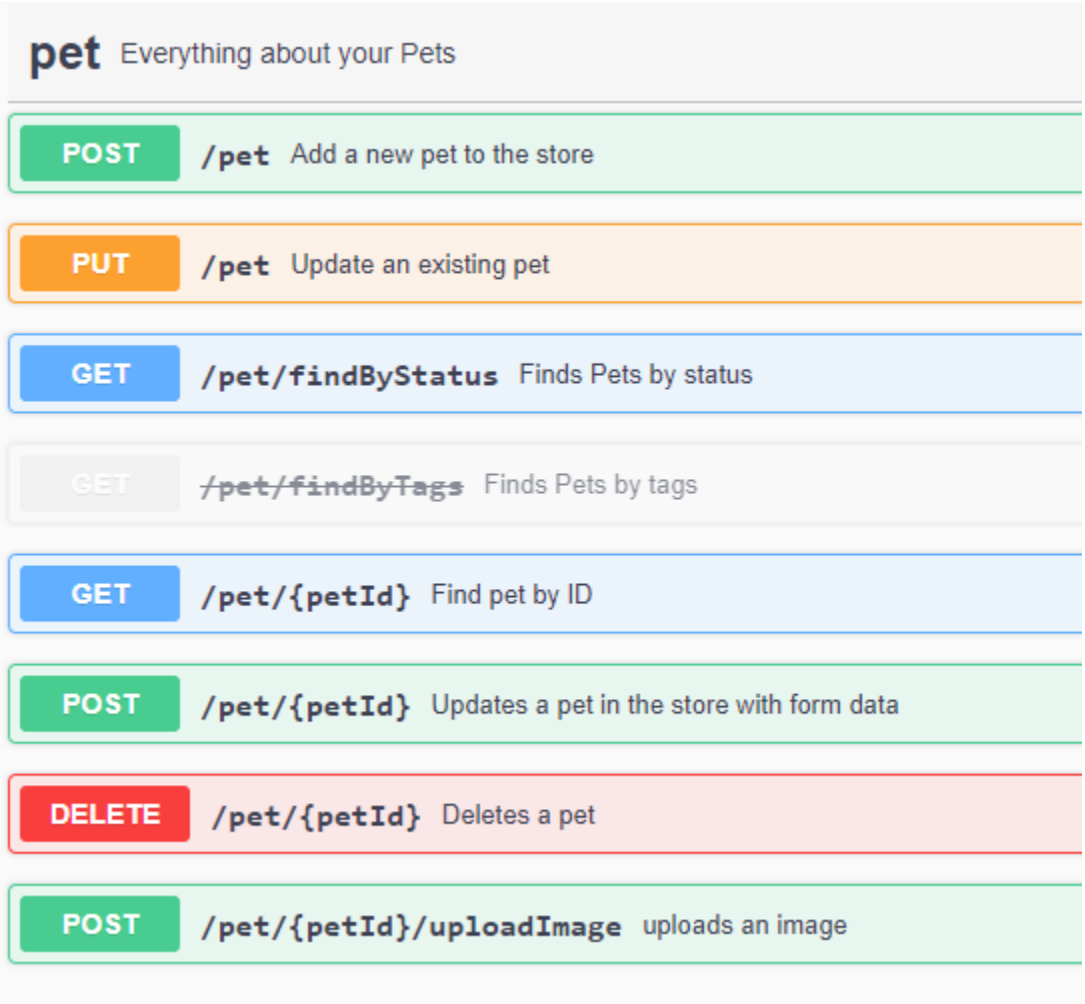
I det andra lagret finns ett databaskluster som består av tre av varandra oberoende databasnoder, detta för att maximera tillgängligheten.

I det tredje lagret finns loggservern vars uppgift är att lagra samtliga loggar som systemet producerar.

2.3.8 Dokumentation av REST-API

För att minska leverantörsberoendet är datalagrets REST-API:er beskrivna med hjälp av definitionen OpenAPI. OpenAPI innehåller information om vilka funktioner och åtgärder som finns tillgängliga i API:et och hur både förfrågningar och data skall vara strukturerade samt hur vilka svar som dessa funktioner och åtgärder returnerar.

OpenAPI ger också utvecklarna en API-klient online som gör att man kan testa API:erna i en vanlig webbläsare. Se Figur 7 nedan för ett exempel på hur klienten ser ut.



Figur 7 Exempel på API-klient

2.3.9 Säkerhet

Då syftet med ett lokalt datalager är att lagra uppgifter om personer ger det vid handen att säkerhetskravet kring ett sådant datalager är högt. Säkerheten kan delas upp i tre områden.

Intern säkerhet

Den interna säkerheten avser hur datalagret kan hanteras internt samt hur tillgången till datalagret styrs. Med internt avser vi hanteringen internt hos ÅHS där användarna som kommer i kontakt med datalagret är personal hos ÅHS.

En av funktionerna som endast nås internt är administrationen av datalagret. Administrationen avser följande funktionalitet:

- Användarhantering
  Uppläggning av användare med olika roller och behörigheter

- API-nycklar
  Möjlighet att administrera vilka system som skall tillåtas få kontakt mot datalagrets API

- Enklare sökverktyg
  Super-administratörer kan göra direkta sökningar för att undersöka ev. problem med personuppgifter

- Loggverktyg
  Administratörerna kan läsa samt exportera loggdata som beskriver förändringar i innehållet i datalagret i kombination med loggning av samtliga händelser i datalagret. Med samtliga händelser avses även icke-destruktiva händelser som läsningar

Tillgången till datalagret internt sker genom autentisering via den lokala katalogtjänsten Microsoft Active Directory (AD). Användare som ingår i de fördefinierade grupperna ges tillgång till datalagret och deras roller och behörigheter styrs genom deras medlemskap i AD-grupperna.

Nätverksmässigt är det endast utvalda nätverk som når administrationsgränssnittet och detta sker genom brandväggskonfigurationer och accesslistor både via ÅHS brandväggar samt interna brandväggar på de servrar som datalagret hostas i.

Extern säkerhet

Den externa säkerheten avser hur datalagret kan nås via integrerade system.

Ett integrerat system definieras som ett datasystem som kommunicerar med datalagret genom att läsa från och/eller skriva till datalagret via datalagrets fördefinierade API-metoder och/eller genom schemalagda exporter och importer via filer.

Säkerhet via API-metoder

En API-metod kan användas av ett integrerat system genom att systemets IP adress tillåts kommunicera med datalagret samt att systemet använder en tillåten API-nyckel för att nå datalagret. API-nyckeln är unik för varje system och datalagret styr även access till vilka API-metoder ett integrerat system får nyttja.

All nätverkstrafik mot API-metoderna sker krypterat med SSL-certifikat med stark kryptering.

Säkerhet via schemalagda fil-importer/-exporter

För de integrerade system som inte klarar av att hantera REST-API:er medger datalagret kommunikation även via fördefinierade filer. En sådan fil kan till exempel vara en prenumerationsfil på personuppgifter som ett integrerat system hämtar från datalagret och importerar till sin egen databas.

Dessa filer hanteras via en server med hjälp av protokollet SFTP (SSH File Transfer Protocol). Med SFTP överförs filerna på ett säkert sätt via kryptering, ingen information skickas som ren text. Servern är avgränsad både från datalagret och från det integrerade systemet. Denna server har unika konton för varje integrerat system samt för datalagret. Detta medger att ett obegränsat antal integrerade system kan läsa och/eller ladda upp filer från/till denna server.

Lagringssäkerhet

All information som datalagret hanterar lagras i ett krypterat databaskluster. Den metod av kryptering som används kallas för Data-at-rest Encryption, metoden är även känd under benämningen Transparent Data Encryption (TDE).

TDE innebär att endast den information som lagras är krypterad, det vill säga information som används till exempel under läsning eller skrivning är okrypterad under dessa processer. Detta underlättar hanteringen av information som skall skickas/tas emot till/från de integrerade systemen.

I kort kan man benämna informationsflödet i tre steg:

- Data at rest
  Inaktiv information som lagras fysiskt i digitalt format
- Data in process
  Aktiv information som är befinner sig i status som skrivning, läsning eller borttagning
- Data in transit
  Information som befinner sig i status transport mellan enheter

Se exempel på dessa i Figur 8 nedan.

Figur 8 TDE Informationsflöde (Sealpath, 2019)

Data at rest skyddas med hjälp av det krypterade databasklustret (Encryption key management, 2019). Data in process och data in transit skyddas med hjälp av krypterad datatrafik.

Lagringssäkerheten avser kryptering av samtlig information i databaserna. Krypteringen baserar sig på nyckelhantering samt en krypteringsplugin för databasen. Detta beskrivs mer detaljerat nedan.

Nyckelhantering

Kort beskrivet används krypteringsnycklar för att göra information obegriplig/oläsbar för obehöriga personer och system. Nycklarna används både för att kryptera information men även för att dekryptera information (göra läsbart).

Krypteringsnycklar finns i en rad varianter och standarder. Databaskrypteringen i datalagret använder sig av Advanced Encryption Standard (AES) som är en symmetrisk kryptering. Med symmetrisk kryptering avser man att både kryptering och dekryptering använder samma hemliga nyckel.

Nyckelhanteringen baserar sig på Amazon Web Services (AWS) Key Management Service (KMS) för att generera och lagra AES-nycklar lokalt, i krypterad form. Dessa nycklar krypteras i sin tur med Customer Master Key (CMK) som lagras i AWS KMS (AWS Key Management Service (KMS), 2019). CMK-nyckeln är den nyckeln som en Amazonkund själv skapar och är att betrakta som en huvudnyckel för all kryptering för den specifika kunden.

När databasen startas dekrypterar ett insticksprogram för kryptering de krypterade nycklarna med hjälp av AWS KMS Decrypt-API. Databasen kommer sedan att få sin information krypterad och dekrypterad med hjälp av AES-nyckeln. Denna lösning stöder flera krypteringsnycklar och samt även så kallad nyckelrotation.

Kryptering av informationen i databasen

Krypteringen sker varje gång databasen skriver till hårddisken. Databasen skriver även temporära filer till hårddisken och även dessa krypteras tillsammans med de binära loggar som skapas av databasen. Samtlig kryptering sker med hjälp av AES-nycklarna som beskrivs ovan.

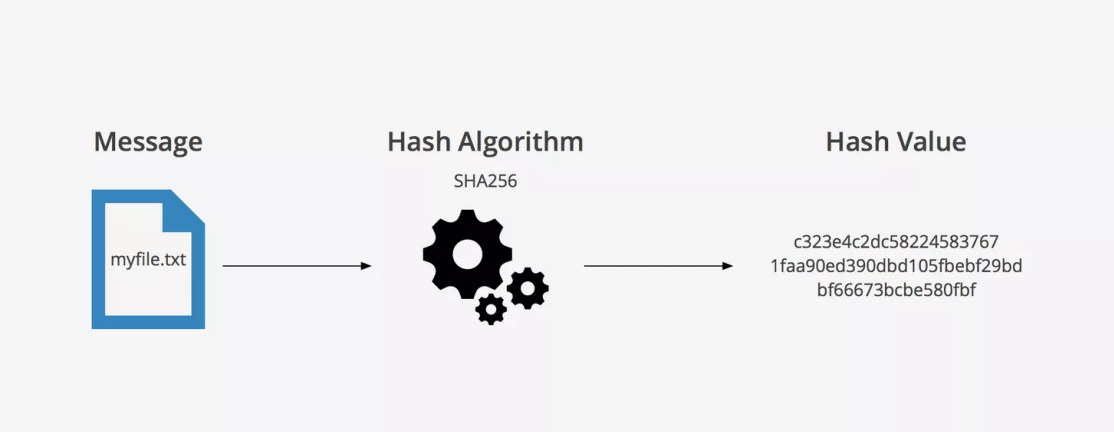
Backuper

Samtliga backuper av både datalagrets system och dess innehåll i form av databaser och loggar säkerhetskopieras regelbundet. Dessa backuper lagras krypterade i en backupmiljö där backuplagringen sparas på tre olika media (två hårddisklagringar och en backupbandslagring) och dessa media existerar på två geografiskt åtskilda platser. Vidare tas även en årlig kopia av samtliga säkerhetskopior ut på backupband och dessa lagras i ett bankfack. Backuperna på backupbanden är även dessa krypterade.

2.3.10 Logghantering

Utöver den interna logghantering av allt innehåll i datalagret samt av användningen av datalagret så skapas även externa loggfiler som beskriver händelser i datalagret. Dessa loggfiler skapas och roteras varje timme.

Datalagret skapar en hash för varje loggfil. En hash eller hashfunktion är en funktion som konverterar en textsträng, i detta fall innehållet i en loggfil, till ett heltal. Det heltal som produceras är unikt och ändrar om man redigerar innehållet i loggfilen och skapar en ny hash. Se figur 9 nedan för ett exempel.



Figur 9 Hashning av loggfiler (Keycdn, 2019)

Varje timme genererar och levererar datalagret en fil innehållande den senaste timmens loggfiler samt varje loggfils-hash (digestfil). Varje sådan hash är en så kallad fil-digest, dvs. en digital summering av innehållet i en annan fil.

Vid varje loggrotation så skickas både loggfiler och digestfilerna till en separat loggserver.

Digestfilerna sparas separat från loggfilerna. Denna separation av digestfilerna och loggfilerna möjliggör ett äkthetsbevis på att loggarna är orörda medan man fortsättningsvis kan nyttja loggarnas innehåll i ett sökbart gränssnitt.

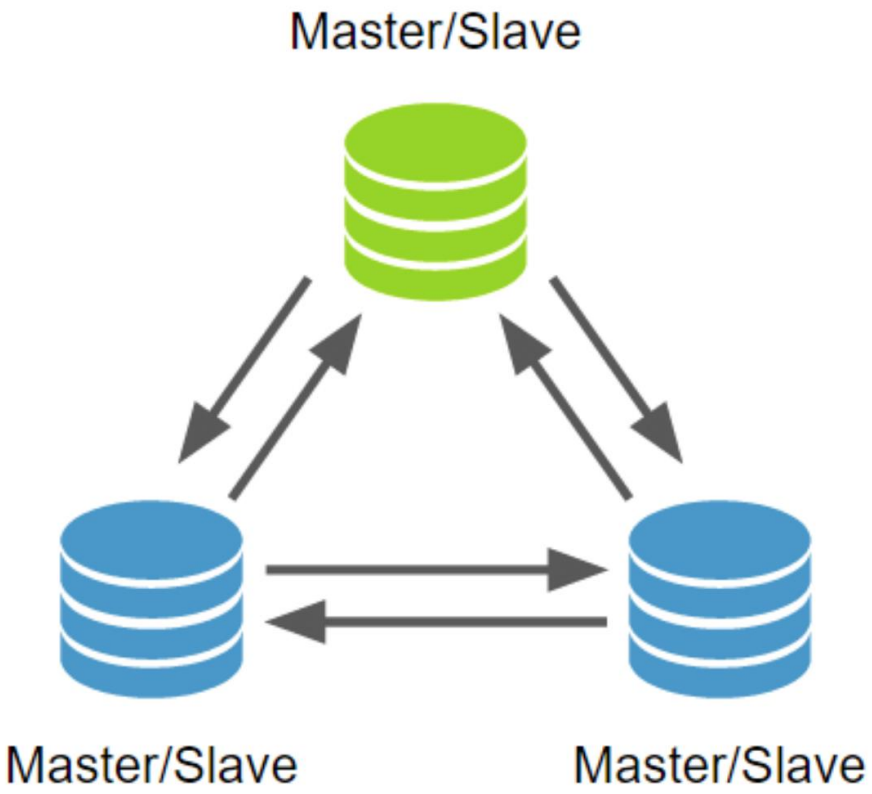
Vid misstanke om loggmanipulation kan man jämföra originalloggarna samt deras matchande digestfil för att verifiera loggfilernas riktighet och att ingen manipulation har utförts av innehållet.

Samtliga loggar krypteras även för utökad säkerhet.

2.3.11 Tillgänglighet

Plattformen är installerad i två, geografiskt åtskilda (+5km) serverhallar med fiberanslutning mellan hallarna och 1Gbps bandbredd mot Internet. Strömförsörjningen är kompletterad med avbrottsfri kraftförsörjning samt dieselgenerator för att trygga tillgängligheten. Samtliga ingående servrar är dubblerade vilket innebär att tjänsten kan uppdateras online utan att störa funktionaliteten hos de integrerade systemen. Detta gäller även både brandväggar och nätverksutrustning.

Databasklustret består av tre av varandra oberoende noder (servrar). Se Figur 10 nedan för en schematisk bild över ett databaskluster med tre noder.



Figur 10 Databaskluster

Övervakning

Samtliga ingående servrar och valda funktioner i systemmiljön övervakas och jourhavande personal får larm om ev. funktionsstörningar så att åtgärder kan vidtas omedelbart.

2.3.12 Spårbarhet

En av hörnstenarna i systemuppbyggnaden av datalagret är att all information i datalagret loggas. Det vill säga att alla förändringar i datalagrets innehåll spåras och lagras i interna loggar samt även i loggar som skrivs till hårddisk. Utöver detta loggas även all användning av datalagret. Om någon administratör söker upp en person och granskar någon uppgift så loggas detta. Även aktiviteter som till exempel en visning av aktivitetsloggen loggas. Syftet med detta är att uppnå fullständig spårbarhet kring information i datalagret och anledning är att man vill veta vem som gjort vad gällande personlig information.

Det som loggas gällande ändringar i datalagret är:

- Vilket system härstammar ändringen ifrån? Detta kan vara BRC, datalagret i sig eller något integrerat system
- När har ändringen gjorts? (datum och klockslag)
- Hur såg informationen ut innan ändringen
- Hur såg informationen ut efter ändringen

Det som loggas gällande användningen av datalagret är:

- Aktivitet. Vad har gjorts?
- Person och/eller system. Vem eller vilket system har gjort vad?
- När? Vilket datum och klockslag utfördes aktiviteten?

Tillsammans skapar dessa loggningar en fullständig spårbarhet som även hjälper administratörer att veta var data om personer existerar, det vill säga vilka system som har tillgång till datalagret och vilka av dessa som hanterat uppgifter om en viss person.

3. INTEGRATIONER

3.1 Patientjournal

Patientjournalen är ur patientsynvinkel ett av de mest kritiska systemen hos ÅHS. Detta beaktat är det naturligtvis av stor vikt att alla patientdata är uppdaterade och aktuella.

Den primära information som behöver hållas aktuellt avseende patienter i journalsystemet är i huvudsak:

- Namn
- Adress
- Kontaktuppgifter
- Anhörig-kontakter

Journalsystemet innehåller samtliga bosatta på Åland samt även personer som besöker ÅHS under t.ex. en semesterresa eller på annat sätt inte har finskt personnummer.

3.1.1 Integrationen mot journalsystemet

Integrationen mot journalsystemet är en helautomatisk integration där datalagret äger uppgiften att hålla journalsystemet uppdaterad. Principen är att datalagret är master, det vill säga vid en eventuell differens är det datalagrets uppgifter som gäller.

Prenumeration

Journalsystemet prenumererar på samtliga personer i datalagret, det vill säga varje gång en ny person dyker upp i datalagret skapas därmed ett kund-kort i journalsystemet. Detta gäller oavsett om personen kommer som tillagd från BRC eller från till exempel lönesystemet eller ekonomisystemet.

I datalagret har varje system en unik id som identifierar systemet. Denna id lagras tillsammans med personbeteckning i en prenumerationslista och på det sättet identifierar datalagret vilka

personer som skall synkronisera mot ett system. Varje synkronisering loggas med datum och klockslag vilket gör att journalsystemet när som helst kan be om en uppdatering av de personer som ändrats sedan senaste uppdatering. Av prestandahänseende är det viktigt att endast synkronisera data som har ändrat då det annars skulle bli onödigt hög belastning på de ingående systemen.

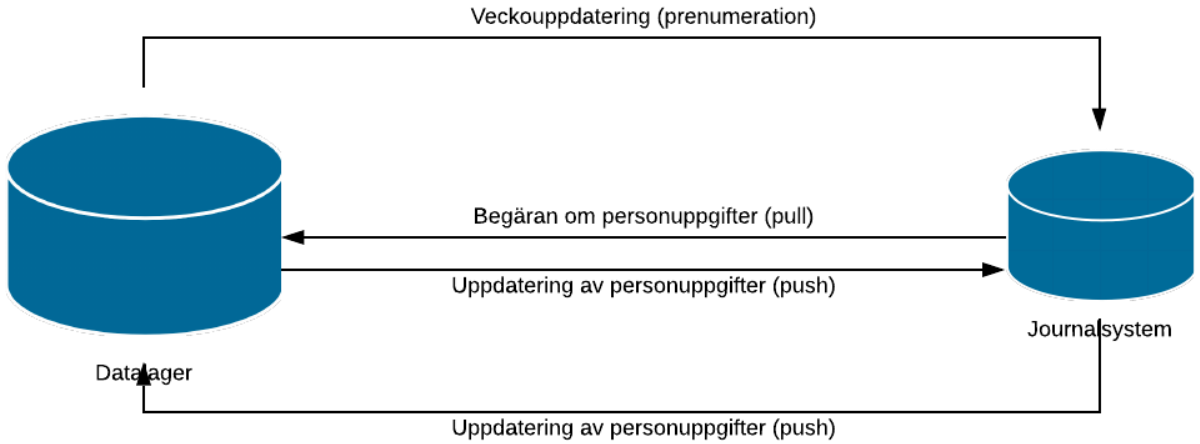
Push

Varje gång en uppdatering eller en ny post skapas gällande patientuppgifter skickar journalsystemet över denna information till datalagret som sedan uppdaterar innehållet i sin databas. Varje ändring loggas med information om varifrån ändringen härstammar samt vid vilket datum och klockslag ändringen gjordes. Vidare loggas även hur informationen såg ut före ändring och hur den ser ut efter ändringen. Alla sådana ändringsposter lagrar sin ändringsinformation i filformatet JSON för att man enkelt skall kunna implementera funktioner för att reversera ändringar vid behov.

Pull

Så fort en anställd läkare eller sjukskötare öppnar ett kund-kort i journalsystemet skickas en pull-förfrågan till datalagret där journalsystemet ber datalagret att uppdatera kunden i journalsystemet givet att något har ändrat på personen. Detta göras för att undvika att man missar ändringar som kan ha skett mellan två synkroniseringar.

Se Figur 11 nedan för en schematisk bild.



Figur 11 Pull-förfrågan

3.1.2 Modulbaserad lösning

Denna integration skiljer sig från övriga integrationer genom att datalagret har mandat att uppdatera journalsystemets databas direkt. Sådana moduler kan löpande skapas i datalagret och målet är att skapa integrationer för samtliga system som handhar personuppgifter.

I denna specifika integration konsumerar datalagret REST-API:er i journalsystemet för att utföra sina operationer.

3.2 Digitala formulär - ett praktiskt exempel på en integration mot datalagret

3.2.1 Digitala formulär

Digitala formulär är en intern SaaS-lösning för hantering av elektroniska formulär både internt inom organisationen samt externt för patienter och övriga personer.

Tjänsten innehåller funktionalitet för att grafiskt producera formulär som uppfyller kraven på tillgänglighet och läsbarhet (WCGA 2.0 nivå AA) och har inbyggt stöd för upprätthållande av GDPR, tillgänglighet samt behörighet.

De digitala formulären har inbyggt stöd för integration mot Suomi.fi-identifiering, Suomi.fi-identifiering, X-Road-kommunikation etc.

De digitala formulären har möjlighet att integrera både inlämnad information via formulär samt även integrera formulärens beståndsdelar mot bakomliggande system. Det kan t.ex. röra sig om information som skall visas i en lista i ett formulär.

De digitala formulären integreras mot datalagret vilket ger en möjlighet att erbjuda en smidig service mot patienterna då dessa får stora delar av formulären ifyllda genom information som automatiskt hämtas från datalagret.

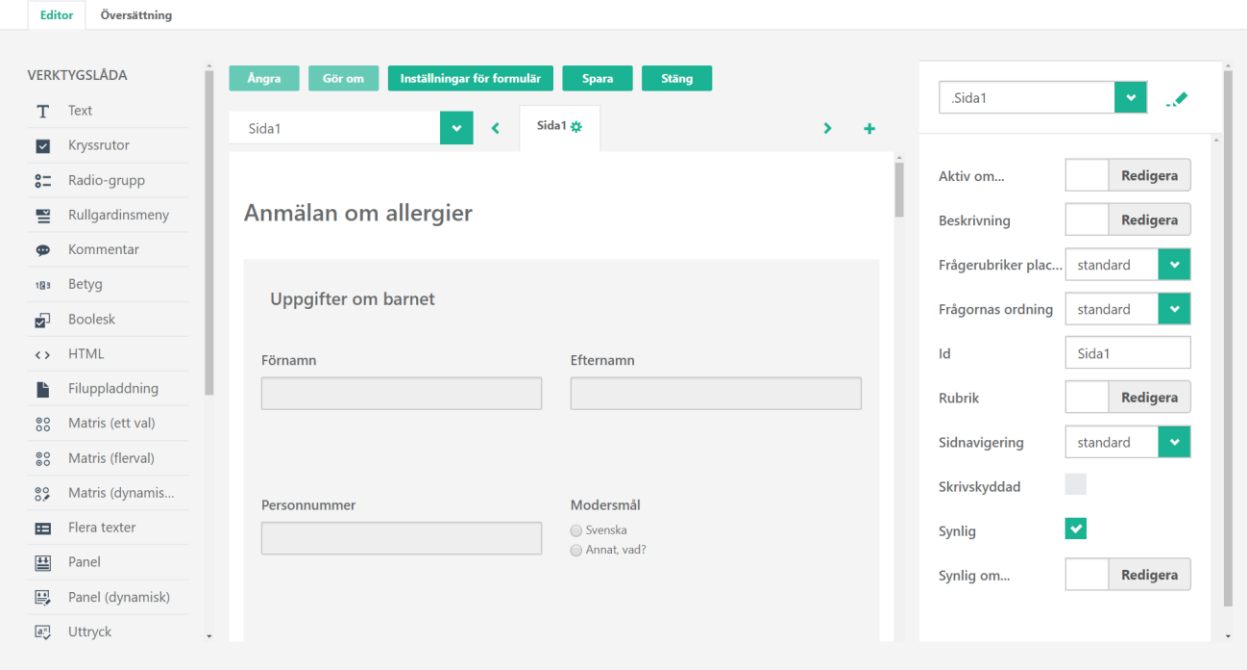
Syftet med denna integration är att eliminera felaktig inlämning av information som t.ex. namn- och adressuppgifter, personnummer etc. Vidare erhåller patienterna en möjlighet att automatiskt välja ev. övriga personer som skall ingå vid ifyllandet av ett formulär, detta kan t.ex. röra sig om hantering av patientens barn.

3.2.2 Integration av formulärets objekt

Personalen arbetar med en formuläreditor som har inbyggda formulärfält som är integrerade mot datalagret. Ett exempel kan ses i bilden nedan, här är dessa fält namngivna KaPa-xxxxxx.

(KaPa=Kansallinen Palveluarkitektuuri/Nationell Servicearkitektur, från den finska lagen med samma namn vilken reglerar myndigheters skyldigheter och rättigheter att integrera servicen).

Fälten fungerar genom att patienten är inloggad i en patientportal och således har tjänsten kunskap om personbeteckningen för den inloggade patienten. Genom att använda de API-metoder som ingår i datalagret kan informationen som är kopplade till dessa formulärfält automatiskt hämtas och populeras i fälten (Figur 12).



Figur 12 Exempel på integration till datalagret

Effekten av denna integration kan ses i Figur 13

Figur 13 Exempel på integration till datalagret

Fälten förnamn, efternamn samt personbeteckning är automatiskt ifyllda och kan inte redigeras av patienten.

3.2.3 Teknisk beskrivning av integrationen

De digitala formulären integrerar sina formulärdata från datalagret genom att anropa datalagret med ett REST API-anrop.

Exempel:
https://url-till-datalagret/person/get

Metod: POST

Inparametrar:
SSN: Patientens personbeteckning

Exempel på retursvar:

LANG:	Patientens språk
SEX	Patientens kön
LASTNAME	Patientens efternamn
FIRSTNAMES	Patientens samtliga förnamn
FIRSTNAME_IN_USE	Patientens tilltalsnamn
PHONE	Patientens telefonnummer
EMAIL	Patientens e-postadress
MUNICIPALITY	Patientens kommun

Autentisering:
API-anropet kräver att det anropande systemet använder en godkänd API-nyckel i kombination med en autentisering som göras med hjälp av en JSON Web Token (JWT).

JWT är baserat på en öppen standard för att skapa en så kallad access token vilket är information som är en sträng bestående av en hash. Denna sträng är sedan den nyckel som systemet använder för att få tillgång till Datalagrets API.
En JWT består av tre olika delar, en header som innehåller typ och hash algoritm, payload, den information man vill skicka med och en signatur som använder header och payload tillsammans med en hemlig sträng från servern för att generera en hash.

Integrationen använder payload-delen för att lagra patientinformationen från datalagret och som sedan används i de digitala formulären för att automatiskt fylla i formulärobjekt som man vill koppla mot datalagret.

4. GRÄNSSNITT

Då man talar om gränssnitt inom informationsteknologi (IT) menar man kommunikationen mellan mjukvara och hårdvara eller mellan människa och maskin. I detta examensarbete kommer vi att fokusera på den senare benämningen, det vill säga, gränssnittet för en användare i förhållande till olika system på arbetsplatsen. (Wikipedia, 2019)

De system som kopplas till datalagret är många gånger verksamhetssystem vars användare har det systemet som sitt primära systemstöd i vardagen. Exempelvis så dokumenterar personalen inom vården det som rör patienter i ett journalsystem och personal inom ekonomi använder sig av ett ekonomisystem till stöd för majoriteten av sina arbetsuppgifter.

Ett sätt att hantera förändringar av masterdata är att endast göra dessa förändringar på ett ställe som alla kopplade delar/system sedan läser och uppdaterar från. Ett exempel på detta är fallet med relationen mellan det lokala och det nationella datalagret. Där kan förändringar av data endast läsas in från det nationella datalagret men det lokala kan inte skriva tillbaka förändringar till det nationella. Det gäller även att den data som ägs av det nationella datalagret inte kan ändras i det lokala annat än när en förändring läses in från det nationella.

Det ovan beskrivna sättet att hantera förändringar av data som hanteras i de olika verksamhetssystemen, men ägs av det lokala datalagret, skulle medföra att de anställda blir tvungna att växla mellan flera gränssnitt beroende på vilken information de för in/förändrar.

För att undvika detta scenario integreras verksamhetssystemen med det lokala datalagret med logik och regelverk som möjliggör att förändringar kan göras från verksamhetssystemen även om den informationen ägs av datalagret. Dessa förändringar propagerar sedan till övriga verksamhetssystem som använder samma information. Detta sätt att hantera förändringar medför att slutanvändarna av respektive verksamhetssystem inte behöver använda sig av flera gränssnitt/system eller hålla reda på vilken data som ägs av respektive databas.

4.1 Patientportalen

För att möjliggöra att patienter själva ska kunna se vilka uppgifter som finns registrerade om dem samt ge möjlighet för dem att även kunna ändra dessa behövs ett gränssnitt med tillhörande användar- och behörighetshantering. Den lösning vi tagit fram för detta är en portal där patienten själv kan logga in och administrera sina uppgifter.

Patienter loggar in med till exempel sina bankuppgifter, som autentiseras mot den bank man valt. Portalen använder sig av e-tjänsterna för identifikation och fullmakter som erbjuds av e-Suomi.fi. Dessa två tjänster är färdiga lösningar som kopplas på till befintliga system. De beskrivs mer allmänt under rubriken E-Suomi.

Genom att koppla på färdiga tjänster till portalen medför detta att ÅHS inte behöver administrera vem som skall få tillgång till portalen. Det gör att hanteringen av användare blir betydligt enklare. Har man finsk personbeteckning och tillgång till bankinloggning, eller annan godkänd autentiseringsmetod, kan man således logga in – tjänsten finns redan hos Suomi.fi. Bankinloggning som metod för identifikation medför hög säkerhet om att personen i fråga faktiskt är den hon/han utger sig för att vara.

Kopplingen mellan portalen och e-Suomitjänsten för fullmakter medför att användare av portalen får rätt sorts rättigheter.

4.1.1 Suomi.fi

Suomi.fi-tjänsten samlar tjänster och information om den offentliga förvaltningens tjänster för medborgare och företag på en och samma adress. E-Suomi är en tjänst som erbjuds av Suomi.fi. Tjänsten erbjuder hjälp till organisationer inom den offentliga och privata sektorn att skapa en gemensam plattform för digitala tjänster. Tanken är att samla alla digitala tjänster ansedda för befolkningen under en och samma plattform, med en enda identitet och en enda identifikationskod.

Dessa uppgifter kan användas av en vanlig medborgare eller då vi representerar en organisation inom den privata eller offentliga sektorn. Det är även möjligt att bevilja fullmakter och att

genomföra transaktioner på en annan persons eller organisations vägnar. Tanken är att kommunikationen med offentliga myndigheter skall bli enklare. Nättjänsten samlar ihop information för medborgare, företag och myndigheter.

Tjänsten Identifikation erbjuder en datasäker elektronisk identifiering. Finländare eller EU-medborgare kan med hjälp av bankoder eller mobilcertifikat identifiera sig på ett säkert sätt på nätet. Det ger ett smidigt sätt att förflytta sig mellan tjänster utan att behöva logga in flera gånger. EU-medborgare från andra länder än Finland kan använda sig av de finska tjänsterna om de tar i bruk Suomi.fi-identifikation. Tjänsten möjliggör t.ex. elektronisk underskrift, förhandsifyllda blanketter m.m. (Suomi-fi identifikation, 2019), (Suomi.fi-tjänster, 2019)

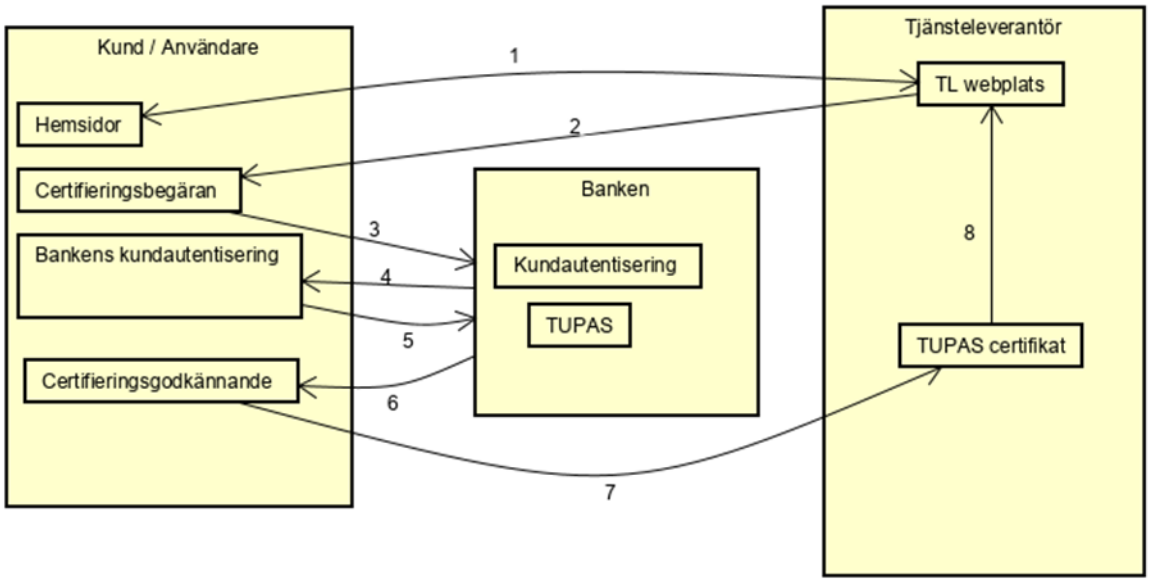
TUPAS/bankinloggning

När man talar om digital autentisering menar man den kontroll av uppgiven identitet som sker vid t.ex. inloggning. TUPAS är en metod utvecklad av Finans Finland (FA), tidigare Bankföreningen i Finland, vars syfte är att möjliggöra ett säkert sätt att identifiera sig på nätet. TUPAS är den officiella metoden för digital identifiering i Finland och används av alla stora banker - t.ex. Aktia, Nordea, S-Banken och även Ålandsbanken. Metoden är baserad på den finska lagen gällande stark elektronisk identifiering. (Wikipedia, 2019), (Finanssiala, 2019)

Lagen kräver att två av tre följande identifieringsmetoder uppfylls:

- Lösenord eller liknande som endast personen vet om.
- Kort med chip eller liknande som personen besitter.
- Fingeravtryck eller liknande som är unikt för personen.

Det vanligaste sättet som identifieringen sker på är med hjälp av lösenord samt en tabell med engångskoder eller en enhet som genererar koder. Nedan följer en beskrivning av hur TUPAS fungerar, konceptet illustreras i Figur 14 och varje steg beskrivs var för sig.



Figur 14 TUPAS (Developer Signicat, 2019)

1. Användaren går in på tjänsteleverantörens hemsida. Specifika tjänster kommer att kräva autentisering från användaren.
2. Tjänsteleverantören producerar en autentiseringsförfrågan - användaren får välja mellan olika banker från en lista.
3. Autentiseringsförfrågan skickas tillsammans med användaren till banken - banken verifierar förfrågan.
4. Om förfrågan lyckas ombeds användaren att autentisera sig.
5. Användaren autentiserar sig.
6. Banken skapar ett TUPAS-certifikat och användaren ombeds acceptera detta.
7. Det godkända certifikatet skickas till tjänsteleverantören - där det måste valideras före användaren får tillgång till tjänsten. (Developer Signicat, 2019), (Finanssiala, 2019)

Om TUPAS-certifikatet är godkänt får användaren tillgång till tjänsten.

Fullmakter

Tjänsten Fullmakter gör det möjligt att kontrollera en persons lagliga rätt till att uträtta ärenden på någon annans vägnar. Det kan handla om en person eller en organisation. Om du har en namnteckningsrätt kan du i egenskap av företrädare uträtta ärenden på ett företags vägnar. I detta fall syftar vi på att uträtta vårdärenden åt sina barn eller någon annan nära stående person. Hur

tjänsten fungerar illustreras i Figur 15 nedan. (Suomi-fi fullmakter, 2019)

Figur 15 e-suomitjänsten Fullmakter (Suomi-fi fullmakter, 2019)

5. SLUTSATSER

Syftet med lösningen var att åtgärda problem med låg kvalitet på data/information i system som hanterar patienters personuppgifter inom vården. Den realiserade lösningen är nu endast kopplad till det primära journalsystemet vilket ger en viss effekt vad det gäller att öka kvaliteten på informationen i just det systemet, främst genom datalagrets kopplingar till det nationella personuppgiftsregistret. Den stora effekten på höjning av kvaliteten beräknas fås först när fler verksamhetssystem kopplas till lösningen, då man får en synkronisering av personuppgifterna i alla kopplade system.

Arbetet har gjorts enligt principer för agila arbetssätt, där mindre delar specificerats och realiserats i sprinter. Tidsplan och ordningsföljd på realisering av de olika delarna har många gånger fått anpassas och justeras efter de ledtider som uppstått i samband med de många ansökningar som behövts göras vid koppling mot de olika nationella tjänsterna som lösningen nyttjar.

I denna text har vi beskrivit de mest centrala delarna i lösningen vi arbetat med, enligt den beskrivna avgränsningen. De kopplingar och integrationer som beskrivs har bedömts vara relevanta för att läsaren ska få en bättre förståelse för lösningen i ett mer konkret sammanhang.

Ju fler verksamhetssystem som hanterar personuppgifter inom organisationen som kopplas till datalagret desto större blir den kvalitetshöjande effekten på den samlade informationsmängden. Detta arbete med att integrera fler system kommer att fortgå i flera år framöver. Den realiserade lösningen med ett centralt datalager med tydliga gränssnitt möjliggör även för andra lösningar än endast den med syfte att höja kvaliteten på personuppgiftshanteringen i organisationen. Exempelvis den beskrivna delen, där patienter ska kunna administrera sina egna personuppgifter, kan utökas med funktionalitet för olika typer av e-tjänster i framtiden.

REFERENSER

Andringsdatatjanst. (den 10 05 2019). Hämtat från Vrk: https://vrk.fi/sv/andringsdatatjanst

Architectural Styles and the Design of Network-based Software Architectures. (den 10 05 2019). Hämtat från University of California: https://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm

AWS Key Management Service (KMS). (den 10 05 2019). Hämtat från Amazon: https://aws.amazon.com/kms/

Developer Signicat. (den 13 05 2019). *Finnish TUPAS*. Hämtat från Developer Signicat: https://developer.signicat.com/id-methods/finnish-tupas/

Encryption key management. (den 10 05 2019). Hämtat från Mariadb: https://mariadb.com/kb/en/library/encryption-key-management/

Finanssiala. (den 13 05 2019). *About us*. Hämtat från Finanssiala: https://www.finanssiala.fi/en/about-us/Pages/svenska.aspx

Finanssiala. (den 13 05 2019). *TUPAS identification principles*. Hämtat från Finanssiala: http://www.finanssiala.fi/maksujenvalitys/dokumentit/TUPAS_identification_principles_v20c.pdf

Informationsled. (den 10 05 2019). Hämtat från esuomi: https://esuomi.fi/suomi-fi-tjanster/suomi-fi-informationsled/?lang=sv

Keycdn. (den 23 05 2019). *SHA1 vs SHA256*. Hämtat från Keycdn: https://www.keycdn.com/support/sha1-vs-sha256

Sealpath. (den 23 05 2019). *Protecting the three states of data*. Hämtat från Sealpath: http://sealpath.com/protecting-the-three-states-of-data/

Suomi.fi-tjänster. (den 03 05 2019). Hämtat från esuomi: https://esuomi.fi/?lang=sv

Suomi-fi fullmakter. (den 03 05 2019). Hämtat från esuomi: https://esuomi.fi/suomi-fi-tjanster/suomi-fi-fullmakter/?lang=sv

Suomi-fi identifikation. (den 03 05 2019). Hämtat från esuomi: https://esuomi.fi/suomi-fi-tjanster/suomi-fi-identifikation/?lang=sv

Svenska institutet för standarder. (2017). Ledningssystem för informationssäkerhet 27001. Svenska institutet för standarder.

The purposes and scope of the General Data Protection Regulation. (den 10 05 2019). Hämtat från Datainspektionen: https://www.datainspektionen.se/other-lang/in-english/the-general-data-protection-regulation-gdpr/the-purposes-and-scope-of-the-general-data-protection-regulation/

Wikipedia. (den 02 05 2019). *Gränssnitt*. Hämtat från Wikipedia: https://sv.wikipedia.org/wiki/Gr%C3%A4nssnitt

Wikipedia. (den 13 05 2019). *TUPAS*. Hämtat från Wikipedia: https://en.wikipedia.org/wiki/TUPAS

Vtjkysely. (den 10 05 2019). Hämtat från Vrk: https://vrk.fi/sv/vtjkyselysv

X-Road. (den 23 05 2019). *Data Exchange Layer X-Road*. Hämtat från X-Road: https://www.niis.org/data-exchange-layer-x-road