Marie Senumstad Sagedal

What are the Inconviniences of Storing Data Centrally?

Information on Central Geospatial Databases in Norway

Trondheim, December 2017

PROJECT THESIS: TBA4560

Supervisor: Terje Midtbø

External supervisor: Lars Eggan, Norconsult Informasjonssystemer

Faculty of Engineering Science and Technology

Department of Civil and Transport Engineering

Norwegian University of Science and Technology (NTNU)

NTNU – Trondheim
Norwegian University of
Science and Technology

Preface

This project assignment is written for the division of Geomatics at the Norwegian University of Science and Technology (NTNU). It is part of the MSc in Engineering and ICT, and was written in the autumn of 2017.

I would like to thank my advisers Lars Eggan (Norconsult Informasjonssystemer) and Terje Midtbø (NTNU) for all the help I have been given.

Trondheim, 2013-12-20

Marie Senumstad Sagedal

Abstract

This paper presents a state-of-the-art description of the main geospatial data stores in Norway (the cadastre and the basic geospatial data), and techniques of updating and synchronising these. There is a new implementation for storing the basic map data; instead of only changing the municipalities' distributed copies of the geospatial data (and uploading these changes once or twice a year into the national database), the changes are now directly updated into a central data store. Different transaction techniques for both the cadastre system and the basic map system are presented, as well as generic web services such as OGC's (Open Geospatial Consortium)'s WFS (Web Feature Service). A litterature review shows that the exchange of data on standardised interfaces seems to strengthen the flexibility and adaptability of projects. The are some disadvantages of updating the geospatial data centrally, such as latency in transactions, but solutions for preventing this are present. The benefits of storing centrally seems to be greater than the disadvantage, reassuring the users of the new managing system of the basic map data that it is sustainable and advantageous.

Contents

Chapter 1

Introduction

In this ever-changing world we are more dependant on up-to-date geospatial data, and without frequent updating of map data the service databases can fast become out-of-date. Traditionally the management system of the basic map data (*FKB*) in Norway was a local production at the municipalities, where the updated data was sent to the central data store once or twice a year. When occasions for updating are rare this may cause great data quality problems (Lehto et al. 2015, Peng 2005).

For about ten years the Norwegian cadastre, *Matrikkelen*, has had a well functioning centralised production system (Falkanger 2017) - will this work for other geospatial systems in Norway as well? Today the new implementation of the the management system of the FKB map data in Norway follows a system where the municipalities directly update the map data to the data store managed by the Norwegian Mapping Authority, *Kartverket*. Are there only advantages to this change, or are there some challenges by updating geospatial data directly in to a central data base? In this paper we will try to answer this question, and the main objectives are:

1. To give a state of the art description of the management systems of the cadastre and FKB data.
2. To give an overview of techniques for transferring geospatial information, and see whether they are platform independent or not.
3. To discuss the pros and cons of having a centralised management system of geosaptial data.

The paper is structured as follows: Initially there is a theoretical chapter presenting the two main registers of geospatial information in Norway. The next chapter is presenting techniques for updating and synchronising the main registers, as well as having a brief discussion on generic versus platform dependent APIs. The following chapter discuss the benefits and disadvantages of updating data into a central data store. We close the paper with a conclusion on whether or not there are disadvantages to centralising the updating and management of the FKB data.

For an explanation of the central terms of this text, the reader may consult with the Appendix A.

Chapter 2

The Main Geospatial Data Systems in Norway

Two of the main registers of geospatial data in Norway are the *Felles Kartdatabase* (FKB), the primary map data, and the second one the cadastre, *Matrikkelen*. The production, maintenance and updating of both are mainly done by *Kartverket*, the Norwegian Mapping Authority. The FKB is also done in collaboration with the members of the *Geovekst* project. Geovekst is a geodata collaboration between different Norwegian public agencies, such as Kartverket, the municipalities and Statens vegvesen (Kartverket 2017*c*). By carrying out joint mapping projects, Geovekst establishes and maintains a common set of map data in Norway.

2.1 *Sentral Felles Kartdatabase* - The Central Map Data Store

To enhance the efficiency of integration, distribution and transport of the FKB-data, the implementation of *Sentral felles kartdatabase* (SFKB), the centralised map data store, started in October 2016. This is a system that centralises the management of the primary map data (FKB) in Norway (Kartverket 2017*d*). The FKB data is the most detailed map data, ranging on a scale from 1:500 to 1:30000. They are all on a vectorised form and are used in for instance production of technical maps and geographical analysis. Some examples of the FKB data are buildings, roads and water.

Through the SFKB system, public agencies, such as the municipalities, are able to directly update the FKB-map data into a central map data store. This ensures the users of the FKB-map data access to fresh and quality assured data at all times. Earlier the updated map data was stored locally at the municipalities, and sent to Kartverket only once or twice times a year. As of December 2017, five municipalities (Bergen, Bærum, Oslo, Stavanger and Trondheim) are not members of the Geovekst collaboration. While all five municipalities have the opportunity to update their FKB-data directly to the central registry, only Bærum has chosen to do so (Kartverket 2017*d*). As a pilot, Trondheim municipality, using geosynchronisation (Will be further explained in chapter 3.1.2), works as a provider of the FKB by offering map data to subscribers (Sæther 2016, Sandal 2016). The remaining three municipalities have chosen to keep and update local copies, and send them to Kartverket regularly, as was the earlier standard.

2.2 *Matrikkelen* - The Norwegian Cadastre

The objective of the cadastre, *Matrikkelen*, is to serve as a registry of cadastral units in Norway, i.e buildings, parcels and addresses. There are several usages of the cadastre. For the municipalities and the local administrations, the cadastre is, among others, an important tool for land use planning and for collection of fees. For the government it is a tool for deriving statistic,while for the private sector, it can provide valuable information about the real estate marked(Mjøs 2002). The server of the cadastre runs centrally at Kartverket.

2.3 Data Flow of the Two Systems

The management systems of the *SFKB* and the cadastre have both the same dataflow as is illustrated in figure 2.1. *The municipalities* update the central data stores through an application programming interface (API). *The central management* (Kartverket) - responsible for the content of the database - controls the data stores and does periodical updates on them. With a synchronisation API (further explained in chapter 3), local copies at the municipalities and distributional copies for all end-users of the mapdata will be kept up-to-date (Kartverket 2015).

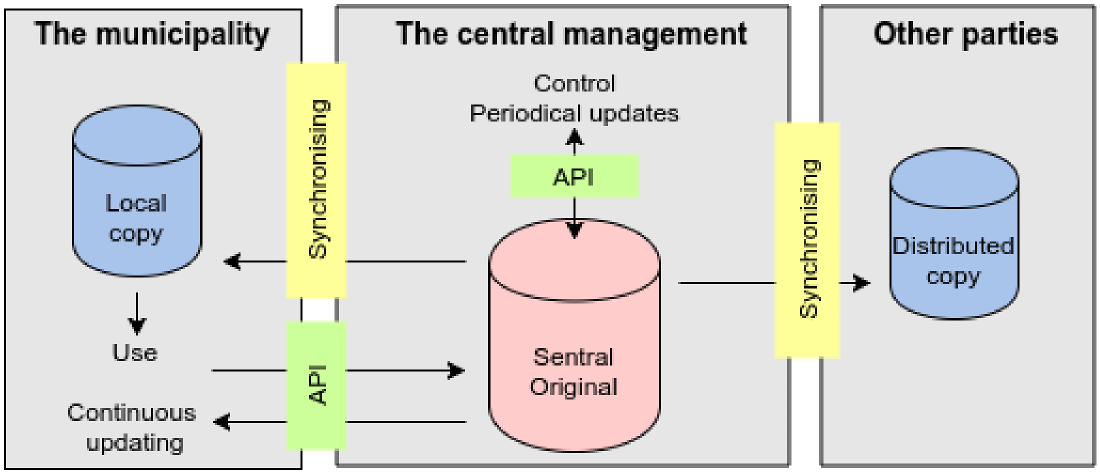

Figure 2.1: Illustration of the concept of data distribution in the systems of both SFKB and the cadastre. The figure is adapted from Kartverket (2015)

Chapter 3

Techniques for Updating and Synchronising Geodata

When submission of new data and modifications on existing data is done for the cadastre and basic map data in Norway, one of the criteria is that all the databases with the same content are updated regardless of where the update is done. What is needed is a geosynchronization service. The synchronization of FKB data is done with *geosynchronisation*, the updating is done through the *NGIS-API*, while the cadastre data is done through the *MatrikkelAPIs*. This chapter deals with these three techniques for getting access to and synchronizing geospatial data, in addition to web services.

3.1 Quadri Map Server (QMS) and Its Synchronising Services

The *Quadri Map Server* (QMS) is the server of the SFKB system, and has a client-server architecture. That is, the servers distribute the data while the clients are able to both fetch and perform services on them (Norkart AS 2010). The structure of the QMS system is depicted in Figure 3.1. The *archives* (data stores) are used for storing the FKB-data, the *portals* define the clients and authorize users for different tasks, where a task is defined as access to an archive. The data stored in, and uploaded into, QMS is defined by the *object catalogue*, and the service of translating logical, humanly meaningful, names of the distributing servers is the *name server*. The *NGIS-API* is the application programming interface for storing spatial data into QMS. All the feature instances in the archives are given unique identifiers. The attributes of the identifiers are LocalId, VersionId and Namespace. Namespace is identifying the data source of the instance, the VersionId identifies the version of the object and the LocalId, based on a Universally unique identifier (UUID), ensures that each feature instance is unique[1].

.

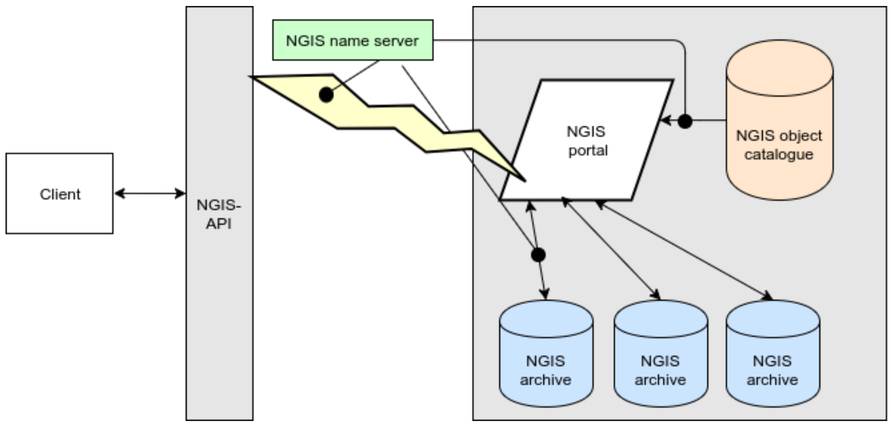

Figure 3.1: The structure of the Quadri Map Server (QMS) system. For updating the data in QMS the NGIS-API is used. The main components of the system are portals, object catalogue and archives. It is possible to use geosynchronisation (chapter 3.1.2) to manage the data in QMS, in addition to the NGIS-API. Figure adapted from (Kartverket 2017*a*)

3.1.1 NGIS-API

When editing an existing object, say a building or the land use of an area, in QMS, the object gets locked in the archive, and the user edits the object locally. When the updating is done, the user has to send the object back to its archive, along with objects that have been edited, added or deleted. When objects are locked in the archives, other users cannot edit them, but they may look at them. This process is called *long transactions* (Kartverket 2017*a*).

3.1.2 *GeoSynkronisering* - the Geosynchronisation Standard

Geosynchronising (*GeoSynkronisering*) is a Norwegian standard for synchronising geospatial information across computer systems, and is a provider-subscriber system as illustrated in figure 3.2. The *provider* updates the database with new data, while the *subscribers* are allowed to update their databases with the changes done by the provider. The SFKB works as a provider: if data is changed or added in QMS, the features will automaticly be geosynchronised to all subscribers. Common subscribers in the SFKB-system are the municipalities and GeoNorge.

[1]The probability to get the same number is small because there are som many possibilities. E.g., the number of random version 4 UUIDs which need to be generated in order to have a 50% probability of at least one collision, is 2.71 quintillion. This is computed as follows: $n \approx \frac{1}{2} + \sqrt{\frac{1}{4} + 2 \times \ln 2 \times 2^{122}} \approx 2.71 \times 10^{18}$ (Eggan & Pedersen 2017)

GeoNorge is the web page where public agencies and institutions, as well as private actors, can get map data and other geospatial information for Norway.

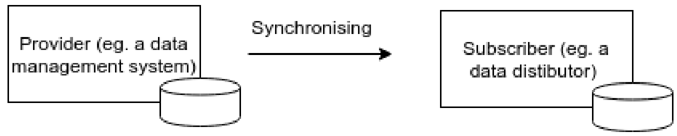

Figure 3.2: The geosynchronisation concept, adopted from (Kartverket 2013, p. 16).

The project of making a standard interface for geosynchronisation services in Norway was started in 2012,used in production systems in 2017 and was a collaboration between Kartverket, the Norwegian Institute of Bioeconomy Research (NIBIO) and system providers in Norway. Geosynkronisering is a open and freely available project on GitHub, and this Norwegian standard is based on the concepts and methods from international standards, i.e. ISO 19100 Geographic Information/Geomatics and Open Geospatial Consortium (OGC) (Eggan & Pedersen 2017, Kartverket 2013).

The geosynchornisation standard is model driven, using models as key artefacts in the development. UML is used on the conceptual level to describe the data, and the data format with GML, using XSD as an application schema showing the meta data. An example of a UML is given below.

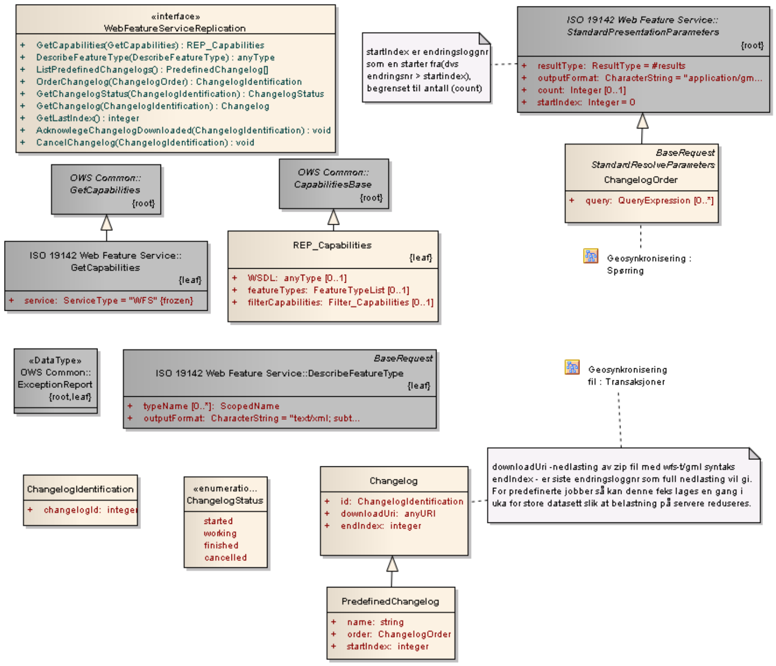

Figure 3.3: Figure from Kartverket (2012*a*) showing an UML model of the services of the geosynchronisation.

The data transfer with geosynchronising (figure 3.4) is a set of transactions of changed features on the GML-format, and their respective changelogs are specified by the WFS-T. GML (Geography Markup Language) is an open interchange format for geospatial transactions and a modelling language for geospatial systems (OGC 2017). WFS-T (Transactional Web Feature Service) allows creation, deletion, and updating of geospatial features on the web (OGC Network 2011). This will be further described in chapter 3.3.1.

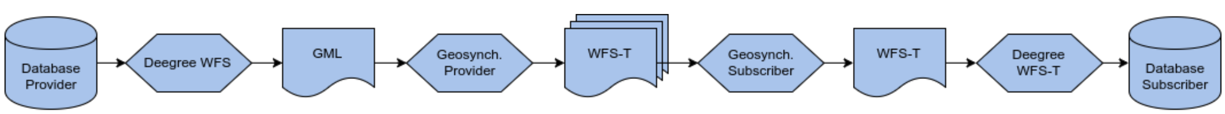

Figure 3.4: A detailed description of the Norwegian geosynchronisation process, adapted from Eggan & Pedersen (2017). The data stored in the database TODO . Deegree is an implementation of the Web Feature Service

Geosynchronising is a service that permits databases with geospatial content to synchronize across different platforms and system solutions (Kartverket 2013), as opposed to the NGIS-API that is a platform dependent API due to its criterion of using c++ programming language and that the developer using it needs to provide the compiler Microsoft Visual Studio (Kartverket 2017*a*, Norkart 2011).

3.2 The Central Cadastral Server and its APIs

There are several services provided by the cadastre; the *cadastral server* serves all clients, both editing and viewing (figure 3.5). The logic of the cadastre server includes management and assembling of cadastre information, as well as control and validation of the business rules of it. The only way to get access to the data on the *cadastre database* is through the *MatrikkelAPI* (chapter 3.2.1) that is on the cadastral central server (Matrikkelavdelingen 2017, p. 338). Submissions and modifications on the cadastral data are done to the centralised data store directly, as for the new SFKB system. When wanting to keep a local cadastre copy, the Matrikkel system provides a *Changelog API* for synchronizing changes on the central data store (See figure 3.5).

The physical architecture of the central cadastral server is clustered, meaning there are several physical servers, but for the clients they will appear as one.

3.2.1 The MatrikkelAPIs

The MatrikkelAPIs is application programming interfaces for extracting data from the cadastre, in both small and large scale.

There are three MatrikkelAPIs. The first one is for the clients with access to updating the cadastre data - the *Editing API*. The second one, the *Viewer API*, includes services for inspecting and reading the data and last the *Changelog API*, as stated earlier, provides services for fetching data changes - primarily for the local copies at the municipalities, but also for external registers. The Viewing API support other services as well: web services as WMS that deliver georeferenced raster maps as well as WFS delivering vectorised map and properties for addresses, buildings and land parcels.

There are several types of clients to the cadastre system: the *updating*-, *viewing*- and *retrieval clients* as well as *other systems*. The updating clients support inspection and updating for all information on the cadastre. Viewing clients supports inspection of the data, as well as viewing the cadastral data joint with data from external systems. Lastly, the retrieval clients supports updating external municipality registers. Other systems using the cadastre system are mainly systems that uses cadastral information for other software solutions, e.g. a municipal specialized system (Matrikkelavdelingen 2017, p. 337-338).

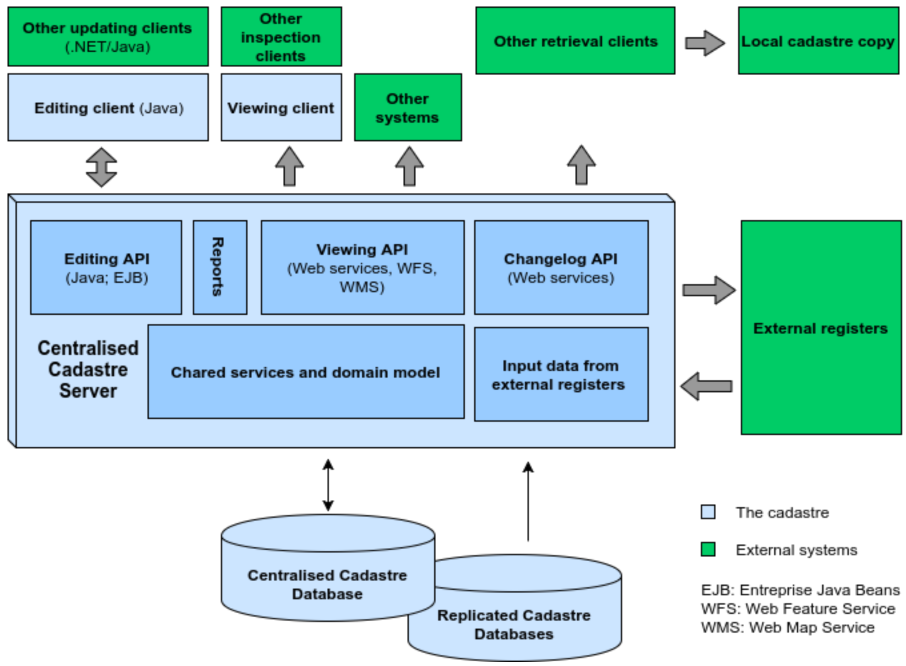

Figure 3.5: The cadastre system, *Matrikkelen.* Solutions for databases through API, clients and relations to other information systems. Adapted from (Matrikkelavdelingen 2017, p. 337)

3.3 Other Generic Services

There are several web services, ways to access web-based geospatial data. These technologies have been created to facilitate the exchange of geospatial information, to find new ways to communicate geodata and to redefine the work flow of the geospatial analysis. In the following subchapters some of these are presented.

3.3.1 Standard Spesifications from the Open Geospatial Consortium)

To get a higher interoperability in the Geographical Information System (GIS) community, the Open Geospatial Consortium (OGC) has created standard specifications for data sharing, processing, retrieval, visualisation, content and cataloguing (Giuliani et al. 2013).

The Web Feature Service (WFS)

The Web Feature Service (WFS) is a standard geodata extracting service for describing data manipulation on a feature level (Peng 2005, {Norge digitalt} 2014). The specification defines interfaces required to support transactions and query operations on geospatial features over the internet.

The GML-format is the defacto standard transferring format, but WFS supports other formats as well (Eggan & Pedersen 2017). By using GML for the exhange of geospatial data, interoperability between the heterogeneous system is provided (Yao & Zou 2008).

Whereas WFS allows querying and retrieval of features, transactional Web Feature Service (WFS-T) permits the user to create, delete and update features (OGC Network 2011),

The Web Mapping Service (WMS)

The Web Mapping Service (WMS) is another standad from OGS for exchanging geograpichal information over the web. When using the Web Mapping Service(WMS), the user is given an image of a map that cannot be edited or spatially analysed.

3.3.2 Simple Object Access Protocol (SOAP)

Simple Object Access Protocol (SOAP) is a protocol that defines how XML and HTTP are used to get access to objects, services and servers from a web based service, thus aiding in interoperability. SOAP defines an envelope for Web Service messages, where the envelope consists of a header with meta data (e.g. how the recipient of the SOAP message should interpret the message) and the body with the actual message. SOAP is often referred to as Web Service, as it is the original web service (Kartverket 2013), and may serve as a standard for all systems because it is platform independent (Sipes 2004).

The Cadastre Viewing and Changelog APIs and the geosynchonisation are examples of SOAP web services.

3.3.3 Representational State Transfer (REST)

There are many definitions of Representational State Transfer (REST) (Fielding et al. 2017, Richardson & Amundsen 2013), this paper uses the description by Fielding & Taylor (1995): *REST is a coordinated set of architectural constraints that attempts to minimize latency and network communication while at the same time maximizing the independence and scalability of component implementations.*

REST is commonly based on the HTTP and HTML standards, and provide simplified calls to services through HTTP. It is only data structures and the transferring of their state that is dealt with in a RESTful service (Battle & Benson 2008). The state transferring calls are POST, GET, PUT and DELETE, which receptively creates, reads, updates and deletes resources in the RESTful service. A resource is identified and resolved with a URL.

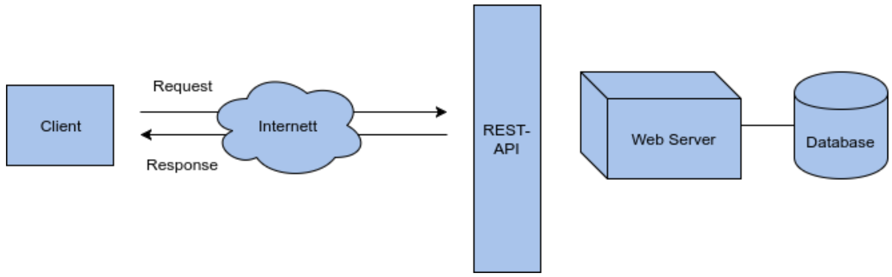

Figure 3.6: REST-API: Clients use HTTP (or other protocols') queries to create, read, update and delete data on the server-side applications. Adapted from (Ceeb 2013)

3.4 Platform Independent Services

This section will discuss some advantages and disadvantages of generic APIs and framework depending APIs. A generic API (or service) is a platform independent service, a software able to run on any hardware or software platform.

An overview of the different services and APIs discussed in this chapter is presented in table 3.1 below, giving an indication whether they are platform independent or not.

Table 3.1: An overview of the services presented in chapter 3. Showing which ones that are platform independent and not.

Technique / Service	Platform independent?	Comment
NGIS-API	No	
Geosynchronisation	Yes	
Editing API (Cadastre)	No	Is Java dependent, but with the help of a Java/-NET Bridging Service component it can be used with .NET. Will be available as a SOAP Service Q1 in 2018, and thus platform independent.
Viewing API (Cadastre)	Yes	
Changelog API (Cadastre)	Yes	
WFS	Yes	
WFS-T	Yes	
WMS	Yes	
REST	Yes	
SOAP	Yes	

3.4.1 Previous Work - Generic and Proprietary APIs

In their paper Giuliani et al. (2013) wanted to benchmark and evaluate the quality of the web services WFS and WCS (Web Coverage Service). They conclude, amongst other things, that the OGC specifications (WFS, WCS) are providing interoperable access to data in an efficient and timely manner, but the specifications are not convenient for transferring large volumes of data. The experience from the Norwegian geosynchronisation projects shows the opposite. By zipping the transaction files (WFS-T with GML), the volume is decreased, resulting in a minor latency for synchronising of the data. An example is provided: a large initial dataset with 220'000 features from the Planning dataset was synchronised from the provider to the Geosynchronising subscriber in only 2.5 minutes. The objects were sent as a initial dataset, meaning they were preprocessed (see figure 3.4), but it still proves that the WFS standard can transfer voluminous data in a fairly short time (ref. conversation with Lars Eggan, Norconsult Informasjonssystemer).

The comparison between a project using a high degree of geospatial standards was compared to another project using few, by Allen Hamilton (2005),shows that platform independent projects have a lower cost of operations and maintenance of data exchange, than proprietary platform dependent projects. Giuliani et al. (2013) states that over time, exchange of data and information performed on standardised interfaces strengthens the flexibility and adaptability of projects.

Chapter 4

Discussion on Centralised versus Distributed Updates

To keep service databases in a multi-provider situation up-to-date, data consistency is important. Data consistency means that if a change to the map data in the data base has occurred, the update is available for all subscribers of the system. A well-architected system does this right after the map-data is updated (Breslow 2004). As was stated in chapter 3.4.1, the transaction time for updating the subscribers of the SFKB system with geosynchronizing was within a reasonable time, thus providing data consistency. Another reason why updating of the FKB data in the SFKB does not have huge latencies is because of (in addition to zippng of the GML files) *short transactions* (ref. conversation with Eggan, L.). Short transactions are transaction without extreme amounts of data, due to limited session time. When updating a feature instance it gets locked (ref. chapter 3.1.1) TODO: skrive om short transactions.

The benefits of the new SFKB sytem are numerous, as compared to local storage. The changes will carry through to *all* end-users, as opposed to local storage, which only applies to the users of that specific municipality (Dontigney 2017). Another benefit is a more efficient and effective validation process, as the requirements are defined and validated one place, and every change to the central store is done by the agency where the change actually occurred (Kartverket 2017*b*).

There are several authors that agree on the benefits (Reichardt 2012, Zhao et al. 2012). The Canadian geosynchronisation project was according to Reichardt (2012) providing the most current and reliable data, consequently avoiding unnecessary versioning and keeping the duplication of data at a minimized level. As the Norwegian geosynchronisation standard is a further development of the Canadian, it gives an indication that the Norwegian geosynchronising process are providing these advantages as well (ref. conversation with Lars Eggan from Norconsult Informasjonssysemer).

Spatial Data Infrastuctures

According to Peng (2005), sharing data on file level is causing latency in update of the data, and it is difficult to deduce a generalised standardisation for all geodata. Reaching consensus may therefore be difficult. Spatial data infrastructures (SDI) seek to facilitate the access and integration of geospatial data coming from various sources. To achieve this objective, systems must be interoperable (Giuliani et al. 2013).

As each country normally operates with separate file formats for the geospatial data (Frenvik 2017), interoperability across borders can be a challenge. The Infrastructure for Spatial Information in the European Community (INSPIRE) is the European collaboration for a common standard for describing and sharing spatial data (INSPIRE 2017). With this directive at hand geospatial data exchange across country borders, and system solutions, becomes easier.

The disadvantage of updating directly to a data store (having validating difficulties when a generalised data format is not obtained) will not be a problem for the SFKB system. The Norwegian map data has been standardised for 30 years, with the standard for exchanging geospatial data in Norway, SOSI (Samordnet Opplegg for Stedfestet Informasjon). The SOSI-standard follows the INSPIRE directive for sharing data, and for identifying spatial objects amongst other things (LocalId was introduced chapter 3.1) (Kartverket 2012*b*, Hokstad 2016). By having the SOSI standard for exchanging geospatial data, validation problems will mainly be because of humanly errors. This last assumptions can be further investigated in later works.

GeoIntegrasjon, common standards for collaboration across systems(e.g. GIS and archive systems) for the public sector in Norway, is another topic that can be further introduced in later works.

Chapter 5

Conclusion

The process of storing map data locally and later send those data to *Kartverket* was perhaps unnecessarily time-consuming and inefficient. But, as this paper has presented there are a few drawbacks of updating directly to a central data store as well. The time it takes to transfer a change to a central system may have latencies. Solutions of this issue for the SFKB system are zipping of files for downscaling the load, and short transactions providing short sessions, thus preventing latency. On the other hand, the benefits seem to be greater than the disadvantage - the SFKB-system seems to have a bright future ahead. It will reduce operational cost by lessening the import, export, copying and control costs of the data, as Kartverket (2017*b*) sums it up. This is in addition to improving service delivery for the public, providing it with fundamental and reliable map data, freshly updated at all times.

Appendix A

Glossary

API Application Programming Interface

FKB Felles kartdatabase (The basic map data)

GML

INSPIRE Infrastructure for Spatial Information in the European Community (An European collaboration for a common geographical infrastructure)

UUID Universally Unique Identifier (A number used to identify objects, all feature instances in SFKB have a LocalId based on an UUID)

Matrikkelen The Cadastre

OGC Open Geospatial Consortium

NGIS API for updating features saved in QMS

QMS Quadri Map Server (The server for the SFKB system)

REST Representational State Transfer

SDI Spatial Data Infrastuctures

SOAP Simple Object Access Protocol

SOSI Samordnet Opplegg for Stedfestet Informasjon

SFKB Sentral Felles kartdatabase (The Central Map Data Store)

UML Unified Modeling Language

WCS Web Coverage Service

WFS Web Feature Service

WFS-T Web Feature Service Transactional

WMS Web Map Service

Appendix B

Oppgaveteksten

TBA4560, Geomatikk, fordypningsprosjekt (15 stp)

Prosjektoppgåve for Marie Senumstad Sagedal

Tittel:

Begrensninger ved sentral lagring av geografisk informasjon

Bakgrunn:

I dag utvikles det et nytt sentralt kartdataforvaltningssystem kalt Sentral felles kartdatabase (SFKB). Ved hjelp av dette systemet skal de som forvalter kartdataene, f.eks. kommunene, kunne oppdatere dataene direkte i en sentral kartbase. På denne måten sikres brukerne av FKB-data tilgang til ferske og kvalitetssikre data til enhver tid. Tidligere ble de oppdaterte kartdataene kun sendt inn til Kartverket en til to ganger i året. Fra før av har vi i Norge en sentral matrikkeltjener og matrikkeldatabase, dette er et system som har fungert bra, og Sentral lagring vil ta i bruk tilsvarende teknikk [1].

Det denne prosjektoppgaven skal fokusere på fordeler og begrensninger ved å lagre data (FKB og matrikkeldata) sentralt, kontra lokal forvalting.

Studenten skal gi:

- Informasjon om sentrale lagringssystemer av kart- og eiendomsdata i Norge (SFKB og Ma-

[1] Kartverket *Innføring av Sentral FKB I kommunene* http://www.kartverket.no/Prosjekter/Sentral-felleskartdatabase/sentral-lagring-av-fkb-data--innforing-i-kommunene/

trikkelen)

- Beskrivelse av teknikker for å synkronisere data (eksempelvis NGIS-API, Matrikkel-API og GeoSynkronisering)
- Bruk og argumenter for/mot bruk av generiske tjenester (WMS, WFS, WPS), kontra egne spesialiserte API'er (?). Plattformuavhengige API'er (WebService, REST) vs. Plattformavhengige API'er.
- Fordeler og ulemper ved sentral lagring.

Ekstern Vegleiar:

Lars Eggan, Norconsult Informasjonssystemer AS

NTNU Vegleiar:

Terje Midtbø

Bibliography

Allen Hamilton, B. (2005), 'Geospatial Interoperability Return on Investment Study', *National Aeronautics and Space Administration Geospatial Interoperability Office* .
URL: *http://lasp.colorado.edu/media/projects/egy/files/ROI_Study.pdf*

Battle, R. & Benson, E. (2008), 'Bridging the semantic Web and Web 2.0 with Representational State Transfer (REST)', *Web Semantics: Science, Services and Agents on the World Wide Web* **6**(1), 61–69.
URL: *https://www.sciencedirect.com/science/article/pii/S1570826807000510*

Breslow, N. (2004), 'Wide Area File Sharing Across the WAN', *Computer Technology Review* **24**(8), 33–34.
URL: *https://search.proquest.com/docview/220625417/fulltext/F62DDAFAD0C2496EPQ/1?accountid=12870*

Ceeb (2013), 'How to build a REST Web API on a Raspberry PI in JavaScript - The Floppy Disk'.
URL: *http://www.robert-drummond.com/2013/05/08/how-to-build-a-restful-web-api-on-a-raspberry-pi-in-javascript-2/*

Dontigney, E. (2017), 'The Advantages of Centralized Database Storage | Bizfluent'.
URL: *https://bizfluent.com/info-8623828-advantages-centralized-database-storage.html*

Eggan, L. & Pedersen, J. (2017), WFS for transaksjoner WFS-T.
URL: *https://www.geonorge.no/globalassets/geonorge2/teknologiforum/2017/eggan_wfs-t_nois_nov2017_tf.pdf*

Falkanger, T. (2017), 'Matrikkel'.
URL: *https://snl.no/matrikkel*

Fielding, R. T. & Taylor, R. N. (1995), 'Principled Design of the Modern Web Architecture'.
URL: *https://www.ics.uci.edu/ fielding/pubs/webarch_icse2000.pdf*

Fielding, R. T., Taylor, R. N., Erenkrantz, J. R., Gorlick, M. M., Whitehead, J., Khare, R. & Oreizy, P. (2017), 'Reflections on the REST Architectural Style and " Principled Design of the Modern Web Architecture " (Impact Paper Award)', *Proceedings of 2017 11th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering* .
URL: *https://static.googleusercontent.com/media/research.google.com/no//pubs/archive/46310.pdf*

Frenvik, A. (2017), 'Det åpne geodata-økosystemet', *Kart og plan* **77**(5003), 108–120.
URL: *http://docs.atlefren.net/ogde.pdf*

Giuliani, G., Dubois, A. & Lacroix, P. (2013), 'Testing OGC Web Feature and Coverage Service performance: Towards efficient delivery of geospatial data', *Journal of Spatial Information Science* **2013**(7), 1–23.
URL: *http://www.grid.unep.ch/products/3_Reports/Testing_OGC_WFS_WCS_performance.pdf*

Hokstad, I. (2016), 'Kartlegger stikkrenner for åforbygge flom', *Posisjon Magasin for kart, oppmåling og geografisk informasjon* (1), 8.
URL: *https://wpstatic.idium.no/www.geoforum.no/2014/09/Posisjon_1_2016-web.pdf*

INSPIRE (2017), 'About INSPIRE'.
URL: *https://inspire.ec.europa.eu/about-inspire/563*

Kartverket (2012*a*), 'Geosynkronisering'.
URL: *http://www.geosynkronisering.no/files/umlmodell/0.3/*

Kartverket (2012*b*), SOSI standard -versjon 4.5 2 Del 1: Realisering i SOSI-format og GML.
URL: *https://www.kartverket.no/globalassets/standard/sosi-standarden-del-1-og-2/sosi-standarden/del1_2_realiseringsosigml_45_20120608.pdf*

Kartverket (2013), Standarder Geografisk Informasjon - Geosynkronisering Versjon 1.0.
URL: *https://www.kartverket.no/globalassets/standard/bransjestandarder-utover-sosi/geosynkronisering-standard-versjon-1.pdf*

Kartverket (2015), Sentral forvaltningsløsning for primaerdata.
URL: *https://kartverket.no/globalassets/kart/geodatasamarbeid/sentral-forv-for-primardata/konseptskisse-sentralforvaltningprimardata.pdf*

Kartverket (2017*a*), 'Informasjon og dokumentasjon om NGIS-API | Kartverket'.
URL: *https://www.kartverket.no/prosjekter/Sentral-felles-kartdatabase/Oppdaterings-API/*

Kartverket (2017*b*), 'Ofte stilte spørsmål'.
URL: *https://kartverket.no/Prosjekter/Sentral-felles-kartdatabase/Ofte-stilte-sporsmal/*

Kartverket (2017*c*), 'Om Geovekst-samarbeidet'.
URL: *https://www.kartverket.no/geodataarbeid/Geovekst/Om-Geovekst-samarbeidet/*

Kartverket (2017*d*), Sentral Felles Kartdatabase - Systembeskrivelse.
URL: *https://www.kartverket.no/globalassets/arkiv/kart/sentral-forvlosning-for-primardata/sentralfelleskartdatabase_systembeskri*

Lehto, L., Hietanen, E. & Latvala, P. (2015), 'Using Geosynchronization for Incremental Update of INSPIRE Service Databases', *GEOProcessing 2015 : The Seventh International Conference on Advanced Geographic Information*

Systems, Applications, and Services .
URL: *https://www.maanmittauslaitos.fi/sites/maanmittauslaitos.fi/files/attachments/2017/09/geoprocessing_2015_1_20_30079.pdf*

Matrikkelavdelingen (2017), Samla systemspesifikasjon Matrikkelens innhold og funksjonalitet Klient Dokumentnummer: MA–nnnn.
URL: *https://www.test.matrikkel.no/matrikkel/docs/SamlaSystemspesifikasjonVer.3.12.pdf*

Mjøs, L. B. (2002), 'New Cadastre in Norway', *TS7.12Regional Experience in the Cadastre – Europe II* .
URL: *https://www.fig.net/resources/proceedings/fig_proceedings/fig_2002/Ts7-12/TS7_12_mjos.pdf*

{Norge digitalt} (2014), Veileder for Web Feature Service (WFS).
URL: *https://www.geonorge.no/globalassets/geonorge2/veiledere/veileder-for-web-feature-service_0_54.pdf*

Norkart (2011), Quadri Map Server - Utviklerhåndbok.
URL: *http://sehavnivaa.no/globalassets/kart/geodatasamarbeid/sentral-forv-for-primardata/quadri-map-server—utviklerhandbok.pdf*

Norkart AS (2010), Quadri Map Server - Overordnet beskrivelse.
URL: *https://www.kartverket.no/globalassets/kart/geodatasamarbeid/sentral-forv-for-primardata/quadri-map-server—-overordnet-beskrivelse.pdf*

OGC (2017), 'Geography Markup Language'.
URL: *http://www.opengeospatial.org/standards/gml*

OGC Network (2011), 'WFS-Transaction (WFS-T) tutorial'.
URL: *http://www.ogcnetwork.net/node/1629*

Peng, Z. (2005), 'A proposed framework for feature-level geospatial data sharing: a case study for transportation network data', *International Journal of Geographical Information Science* **19**(4), 459–481.
URL: *http://www.tandfonline.com/doi/abs/10.1080/13658810512331319127*

Reichardt, M. (2012), 'New OGC standard supports "GeoSynchronization"', *GeoInformatics; Emmeloord* **15**(7), 1.
URL: *https://search.proquest.com/docview/1115586129/citation/4047B7870DC04126PQ/1?accountid=12870*

Richardson, L. & Amundsen, M. (2013), *RESTful Web APIs*.
URL: *https://books.google.no/books?id=ZXDGAAAAQBAJ&printsec=frontcover&hl=no&source=gbs_ge_summary_r&cad=0#v=onepa*

Sæther, M. H. (2016), Status geosynkronisering i Trondheim kommune.
URL: *https://www.geonorge.no/globalassets/geonorge2/plan/moter-i-plandataforum/20160519/geosynkronisering_status_trondheim kommune.pdf*

Sandal, K. (2016), Sentral lagring av FKB med ISY WinMap Kjell Sandal Norconsult Informasjonssystemer AS.
URL: *https://wpstatic.idium.no/www.geoforum.no/2016/10/NGIS-Sentral-lagring-NOIS.pdf*

Sipes, J. (2004), 'A Standard Approach', *CADalyst* **21**(12), 58–61.
URL: *https://search.proquest.com/docview/216094181?OpenUrlRefId=info:xri/sid:primo&accountid=12870*

Yao, X. & Zou, L. (2008), 'Interoperable internet mapping–an open source approach.(Technical report)', *Cartography and Geographic Information Science* **35**(4).

Zhao, P., Foerster, T. & Yue, P. (2012), 'The Geoprocessing Web', *Computers & Geosciences* **47**, 3–12.
URL: *https://www.sciencedirect.com/science/article/pii/S0098300412001446*