FFI Forsvarets forskningsinstitutt

22/00793

FFI-RAPPORT

Datainnsamling for analyser av sosiale medier i en totalforsvarssammenheng

– metoder og implikasjoner

Gard-Inge Rosvold
Arild Bergh

Datainnsamling for analyser av sosiale medier i en totalforsvarssammenheng

– metoder og implikasjoner

Gard-Inge Rosvold
Arild Bergh

Forsvarets forskningsinstitutt (FFI) 4. juli 2022

Emneord
Databaser
Datainnsamling
Datamaskinprogrammer
Datastruktur
Hybridkrigføring
Sosiale medier

FFI-rapport
22/00793

Prosjektnummer
1582

Elektronisk ISBN
978-82-464-3380-6

Engelsk tittel
Collecting data for social media analysis regarding national security – methods and implications

Godkjennere
Stig Rune Sellevåg, *forskningsleder*
Janet Martha Blatny, *forskningsdirektør*

Dokumentet er elektronisk godkjent og har derfor ikke håndskreven signatur.

Sammendrag

Nasjonale trusselvurderinger i 2022 viser til en økning av statlige aktørers bruk av sosiale medier for å spre desinformasjon og utøve påvirkning for å skade demokratiske land. Det er også en markant økning i ikke-statlige aktørers bruk av sosiale medier til å spre feilinformasjon i forbindelse med kriser som for eksempel covid-19-pandemien.

Sammenstilling av informasjon fra sosiale medier som en del av det sammensatte trusselbildet er relevant for ansvarlige etater og myndigheter for Norges sikkerhet. Forsvarets forskningsinstitutts (FFIs) undersøkelser av problemstillinger relatert til sosiale medier, påvirkning og desinformasjon har fremhevet nødvendigheten av fleksible analyser som kan dekke dette behovet. Samtidig er det nødvendig å ha kunnskap om og forske videre på sosiale medier og sammensatte trusler. Både operative analyser og forskning krever tilgang til relevante sosiale mediedata som kan studeres.

Denne rapporten beskriver hvordan data kan samles inn fra sosiale medier. Rapporten har to målgrupper: i) de som har behov for dataanalyser (her kalt *bestillere*), og ii) de som gjennom utvikling eller administrasjon av databaser er ansvarlige for tekniske aspekter ved datainnsamling (her kalt *utførere*). Rapporten kan også være av interesse for andre som jobber med sosiale medier og sammensatte trusler. Fokuset i rapporten er på de tekniske og praktiske sider ved datainnsamling fra sosiale medier. Det er derfor utenfor denne rapportens rammer å diskutere spesifikke aktører og tilnærminger.

For *bestillere* vil rapporten belyse praktiske problemstillinger. Spørsmålene man ønsker å få svar på ved å analysere data fra sosiale medier, vil påvirke mengden av data som skal samles inn. Her må man balansere kostnader, i form av tid og penger, med behovet for detaljnivået på analysene. Sosiale mediers relasjonelle natur er spesielt viktig å forstå. Det sentrale konseptet med venners venner og deres innlegg som kilde for informasjon fører til en mye større vekst i datamengden enn man kan anta, en såkalt eksponentiell økning. Det er utfordrende å gi et enkelt svar på hvordan man skal foreta avveininger, så rapporten fokuserer på å synliggjøre problemstillingen og illustrere hvilke faktorer som må tas i betraktning.

For *utførere* gir rapporten en detaljert beskrivelse av hvordan man kan utvikle programvare for datainnsamling fra sosiale medier. Funnene diskutert her er basert på en prototypedatainnsamler for YouTube som ble utviklet for å utforske relevante problemstillinger. Rapporten utforsker YouTube sitt programmeringsgrensesnitt (API) og hva den relasjonelle naturen betyr for datainnsamling. Sosiale medier-relasjoner diskuteres fra et utviklerperspektiv, spesielt med tanke på sosiale mediers kvoter for nedlasting av data, og hvordan det påvirker innsamlingsstrategier. Avslutningsvis drøfter rapporten mulighetene for å overføre tilnærmingene fra YouTube-prototypen til andre sosiale medier, og kommer med forslag til sanntids datainnsamlingsstrategier.

Summary

Recent national threat assessments have highlighted the increase in state actors' use of social media to disseminate disinformation and undertake influence operations to damage democratic countries. There is also a significant increase in non-governmental actors' use of social media to spread misinformation in connection with crises such as the Covid-19-pandemic.

Authorities responsible for national security will therefore need to analyse data from social media to create a situational awareness as part of a larger threat picture. The Norwegian Defence Research Establishment (FFI) investigations on issues related to social media-based influence operations and disinformation have highlighted the demand for flexible analyses to meet this need. At the same time, it is necessary to undertake further research on social media as a part of hybrid threats. Both operational analyses and research require access to relevant data for study from social media.

This report describes how to collect data from social media. The target groups are: i) Those who require data analyses (here called *the customer*), and ii) those who, through development or administration of databases, are responsible for the technical aspects of data collection (here called *the supplier*). The report may also be of interest to others who work with social media and hybrid threats. The focus here is on the technical and practical aspects of data collection from social media. It is beyond the scope of this report to discuss specific disinformation and influence operations issues such as actors or approaches.

For the *customer* target group, practical considerations are explored. The questions one wants answered by analysing data from social media will affect the amount of data to be collected. Here one will have to balance costs, in terms of time and money, with the level of details an analysis requires. It is particularly important to understand the relational nature of social media. The key concept of friends' friends and their social media posts as a source for information leads to a steeper growth in the data size than one may assume, a so-called exponential increase. It is not possible to give a simple answer as to what trade-offs to make; instead, the report focuses on highlighting the issues and illustrates some factors to consider.

For the *supplier* target group the report provides a detailed description of how data collection software can be developed. The findings discussed here are based on a prototype data collector for YouTube that was designed to explore issues relevant to social media data collection. YouTube's programming interface (API) is explored and the relational character of social media and the implications it has for data collection are discussed from a developer's perspective. The effect of social media quotas on data downloads and overall collection strategies is considered. Finally, the report examines the possibilities of transferring the YouTube prototype approaches to other social media, as well as providing suggestions for real-time data collection strategies.

Innhold

1 Innledning

1.1 Bakgrunn

I det siste tiåret har det vært en sterk økning i statlige aktørers bruk av sosiale medier for å spre desinformasjon eller forsøke å influere andre lands borgere gjennom koordinerte påvirkningsoperasjoner [1]. De norske sikkerhetstjenestene har også fremhevet dette som en trussel mot Norge i de senere år [2, 3]. Siden utbruddet av covid-19-pandemien i 2020 er det også blitt tydelig at des- og misinformasjon som spres gjennom sosiale medier av ikke-statlige aktører utgjør et stadig økende samfunnsproblem [4]. Disse aktørene kan ha finansielle eller ideologiske motiver, eller et ønske om å rekruttere støttespillere til ytterliggående bevegelser.

Disse aktivitetene utgjør en del av det sammensatte trusselbildet som aktører innenfor totalforsvaret må forberede seg på å møte i fremtiden [5, 6]. Totalforsvarsaktører har derfor behov for et oppdatert situasjonsbilde av hva som skjer på sosiale medier. Dette situasjonsbildet må være relevant til oppgavene aktøren er pålagt, det vil derfor være behov for forskjellige analyser av sosiale medier. For eksempel, helsemyndigheter ville under covid-19 pandemien hatt behov for informasjon om anti-vaksine desinformasjon i sosiale medier. Samtidig er taktikkene og metodene som benyttes av desinformasjons-aktørene i stadig endring, dette for å unngå å bli stoppet av de forskjellige sosiale medie-selskapene som forsøker å fjerne feilinformasjon fra sine plattformer.

Gitt situasjonen beskrevet her er det et klart behov for både operative analyser og generelle forskningsbaserte undersøkelser på dette området. FFI har jobbet med problemstillinger relatert til påvirkning i det digitale rom siden 2016 [7, 8, 9] og FFI-prosjektet *Cyber-social Propaganda and Influence (C-SPI)* har påpekt behovet for oppdaterte analyser basert på data fra sosiale medier fra både et forskningsperspektiv og brukere [10, 11]. Dette kan dreie seg om øyeblikksbilder som ser på feilinformasjon om vaksiner, eller dypere utforskning av narrativ, metoder og taktikker som forskjellige desinformasjons-aktører benytter.

Denne typen analyser er avhengig av et relevant datagrunnlag som består av informasjon samlet inn fra sosiale medier. Dette dreier seg typisk om en **tidsbegrenset** innsamling av **pseudonymiserte** data fra sosiale medier for å skaffe tilstrekkelig og relevant data for analysen(e). For å fremskaffe slik data har man flere valg. For det første kan man benytte eksisterende datasett, disse kan være fritt tilgjengelig fra forskere[1] eller det kan kjøpes fra såkalte *data brokers*. Slike datasett gir umiddelbar tilgang for analyser. Ulempen er at datasettene vil ha underliggende begrensninger som kan gjøre det vanskelig å få et relevant situasjonsbilde [12]. For eksempel vil et datasett som kun har samlet inn engelske tweets ha en del data fra norske brukere, men vil mangle mye relevant data for norske aktører.

[1] Se for eksempel https://www.kaggle.com/search?q=disinformation+datasetFileTypes%3Acsv+datasetFileTypes%3Atxt.

En annen mulighet er kommersielle løsninger hvor man selv spesifiserer hvilke data som skal samles inn – men disse har også sine begrensninger. For eksempel tilbyr mange tjenester tilgang til data fra Twitter. Men i Norge er Twitter brukt daglig av mindre enn 10 prosent av befolkning, og er derfor av liten interesse for å utvikle et relevant situasjonsbilde.

Slike begrensninger med hensyn til tredjeparters datainnsamling gjør at det ofte vil være behov for å samle inn egne data som analyseres for å oppnå et relevant situasjonsbilde eller for å utforske aktuelle forskningsspørsmål. Per i dag er det få eller ingen totalforsvarsaktører som foretar egne analyser. Det vil derfor være nyttig å dele FFIs erfaringer fra utviklingen av en prototype på programvare for datainnsamling fra YouTube. Denne rapporten beskriver hvordan man kan utvikle programvare for automatisk innsamling av online data og hvilke avveininger man må foreta med hensyn til omfang, metoder og tilgjengelige ressurser. Slik kunnskap kan bidra til bedre evne til å skreddersy datainnsamling fra sosiale medier. Det vil også gi relevant personell bedre kunnskap om hvordan man kan evaluere datainnsamlingsmetoder og hvilke ressurser en slik innsamling kan kreve.

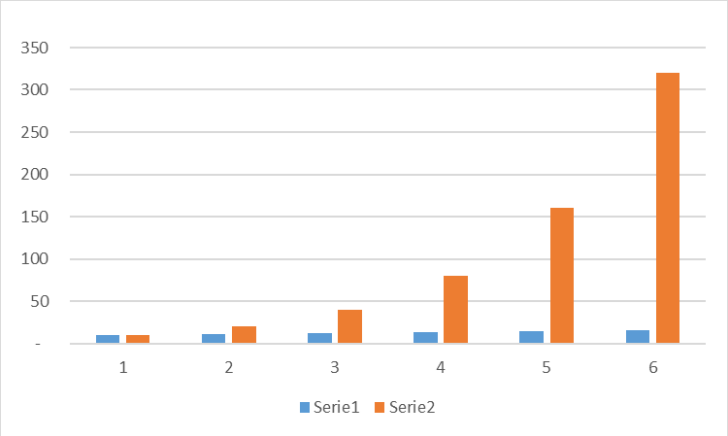
FFI har i forbindelse med sitt arbeid utviklet detaljerte rutiner for ivaretagelse av personvernet og etterlevelse av EUs General Data Protection Rules (GDPR) og relevante norske lover og regler [13]. Det er utenfor omfanget av denne rapporten å gå i detaljer på personvern. Dette må hver enkelt organisasjon som skal samle inn data selv håndtere.

1.2 Definisjoner

Desinformasjon er opprettelsen og/eller spredning av informasjon som bevisst er forfalsket eller fordreid. **Misinformasjon** kan ha identisk innhold som desinformasjon (og er ofte basert på innhold skapt av desinformasjonsaktører), men den som oppretter eller deler informasjonen tror selv den er sann. Begge former for feilinformasjon kan være problematisk for totalforsvarsaktører, i denne rapporten brukes derfor disse to uttrykkene om hverandre.

Datainnsamling betegner en prosess som består i å søke etter, laste ned og lagre informasjon. I inneværende rapport dreier dette seg om informasjon fra sosiale medier.

Profil: Brukes om entiteter som legger ut informasjon i sosiale medier. Det kan være en organisasjon, for eksempel Forsvaret, eller det kan være en enkeltperson. I påvirknings-operasjoner kan profiler kontrolleres av såkalte bots, automatisert programvare, eller de er falske profiler som utgir seg for å være en annen person.



Figur 1.1 Prosentvis kontra eksponentiell vekst

Eksponentiell: Denne rapporten diskuterer implikasjonene av sosiale mediers «eksponentielle natur». En eksponentiell vekst er en økning i et antall som er proporsjonal med antallet selv, dette er en mye sterkere vekst enn en fast prosentvis økning. Dette er illustrert i figur 1.1. Et typisk eksempel på dette er hvordan en person kan ha ti kontakter på et sosialt medium, som hver har ti kontakter og hvor disse igjen har ti kontakter. På det tiende nivået blir dette 10 milliarder kontakter.

Pseudonymisering er behandling av personopplysninger på en slik måte at personopplysningene ikke lenger kan knyttes til en bestemt registrert bruker uten bruk av tilleggsopplysninger, forutsatt at nevnte tilleggsopplysninger lagres atskilt og omfattes av tekniske og organisatoriske tiltak som sikrer at personopplysningene ikke kan knyttes til en identifisert eller identifiserbar fysisk person [14].

Et applikasjonsgrensesnitt, **API** (application programming interface), er en standardisert liste over kommandoer som utviklere kan bruke for å sende eller forespørre data fra en tjeneste. Hvis man for eksempel vil ha en liste over de 25 mest populære videoene på YouTube i norsk region – sendes kommandoen *chart=mostPopular&regionCode=NO&maxResults=25* til YouTube. YouTube sender da tilbake en liste over disse videoene.

1.3 Rapportens innhold og målgrupper

Denne rapporten har to hoveddeler. Den første delen, kapittel 2, er ment for de som vil ta beslutninger om hva som skal undersøkes på sosiale medier, her kalt en bestiller (basert på bestiller/utfører-prinsippet som ofte benyttes i offentlig forvaltning). Det kan være analytikere, forskere, beslutningstagere eller ledere. Den andre delen, kapittel 3, er for utførere. Typisk programmerere og database-arkitekter som skal implementere en teknisk løsning for å samle inn data. Formålet med denne rapporten er å gi disse to gruppene en oversikt over de praktiske utfordringene forbundet med datainnsamling med hensyn til tidsbruk og kostnader. Dette er elementer som må veies opp mot analysebehovet. Er for eksempel behovet for en kjapp oversikt større enn behovet for en detaljert analyse som vil bruke lenger tid på å samle inn data?

For **bestillere, de som forespør data,** er det viktig å forstå hvordan sosiale medier skaper relasjoner mellom brukerne og informasjonen brukerne legger ut. Det er disse relasjonene som gjør manipulering av sosiale medier nyttig for de som ønsker å spre des- og misinformasjon. Implikasjonen av disse relasjonene er at data fra sosiale medier har en eksponentiell karakter. Det vil si datamengden, i form av innlegg, øker mer enn en fast prosentandel for hver relasjon som utforskes. Hentes data fra en video med én kommentar, resulterer dette i ca. 0,1 megabyte data, mens 50 kommentarer resulterer i nesten 12 megabyte data (og ikke 5 megabyte som man kunne tro). De som har behov for data, bør av den grunn vurdere nøye behovet for detaljer i analysen opp mot kostnader i form av tid, penger og prosesserings- og datamaskinkapasitet. Eksempelvis kan det være nyttig å analysere alle innlegg som følgere av en profil poster for å evaluere om de videreformidler narrativ fra påvirkningsoperasjoner. Et slikt detaljnivå øker kostnadene sammenlignet med en analyse som kun ser på kommentarer som de samme følgerne har lagt inn hos profilen de følger.

For **utviklere og databasearkitekter/administratorer** gir rapporten en detaljert innføring i mulighetene for automatisert datainnsamling fra sosiale medier. Rapporten diskuterer de praktiske aspektene ved datainnsamling basert på en proof-of-concept (PoC) datainnsamler som FFI utviklet og testet på YouTube-data. Via en gjennomgang av dette verktøyet redegjør kapittelet for hvordan man kan benytte sosiale mediers egne applikasjonsgrensesnitt, kalkulere forventet datamengde, og gjøre bruk av sosiale mediers egne datastrukturer. Med dette kan det bygges opp en lokal database for analyse med informasjon som gjenspeiler relasjoner og interaksjoner som har funnet sted online. Flere detaljerte beregninger er gjengitt i vedleggene C til E. YouTube er brukt som eksempel i denne rapporten for å illustrere problemstillinger og diskusjoner. Teknikkene som er brukt med YouTube er i varierende grad overførbare til tilsvarende arbeid på andre sosiale medier.

Avslutningsvis følger en kort oppsummering av rapporten.

1.4 Avgrensninger, problemstillinger og metode

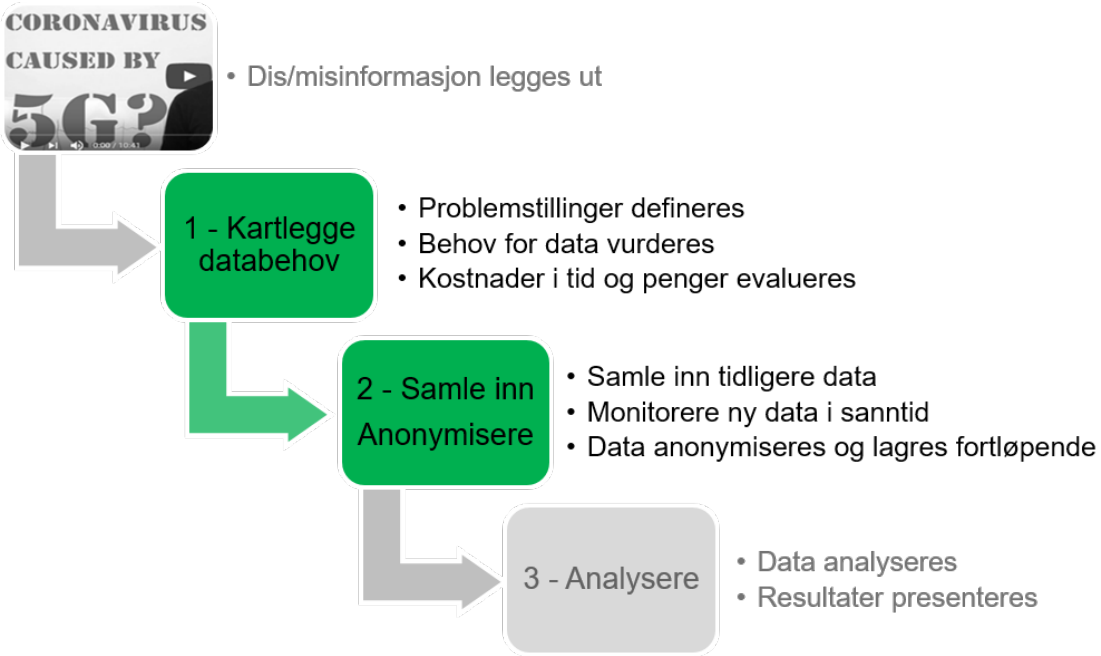
Hovedavgrensning for denne rapporten er at den kun omhandler datainnsamling fra sosiale medier for analyseformål. Diskusjoner om hvilke aktører som benytter seg av påvirkning eller desinformasjon og eventuelle teknikker, metoder eller mål er utenfor rammene for denne rapporten. På det praktiske nivået (utviklingen av programvare) benyttes YouTube som et case for diskusjon. Andre sosiale medier diskuteres kun med hensyn til overførbarhet av den YouTube spesifikke programvaren. Det er også utenfor rammen å diskutere analyser, personvern og datasikkerhet.

Forskningsspørsmålene som FFI adresserer i denne rapporten gjennom bruk av en *proof of concept* tilnærming er:

1. Hvordan kan man samle inn data som støtter Totalforsvarets behov (i motsetning til å benytte en kommersiell, predefinert løsning)?
2. Hvor lang tid tar det å utvikle en slik løsning?

3. Hva er kostnadene av et slikt innsamlingsarbeid med hensyn til initiell utvikling samt
   a. pågående drifting på egne dataservere?
   b. pågående drifting i en offentlig nettsky
4. Hvor forskjellige er sosiale medier når det gjelder datainnsamling?

Disse problemstillingene ble utforsket gjennom en anvendt tilnærming: programvare ble utviklet og brukt for å bevise at konseptet virket gjennom en begrenset, pseudonymisert datainnsamling. All data ble slettet uten analyse av innholdet. Formålet var kun å teste innsamlingsmetoden og få et statistisk beregningsgrunnlag.



Figur 1.2 Eksempel på datainnsamlingsrutine. Steg 1 diskuteres i kapittel 2, og steg 2 diskuteres i kapittel 3.

Utviklingen av datainnsamleren har blitt utført i tre faser. Først ble en demonstrator utviklet for å vise potensialet i analyser av sosiale medier til eventuelle brukere i totalforsvaret. En demonstrator i denne konteksten er et forenklet, interaktiv brukergrensesnitt som viser hvordan en ferdig løsning vil fungere. Denne ble så utvidet med kode som bevis på konseptet (proof of concept, PoC) som ble testet på faktisk datainnsamling. Til slutt ble dette videreutviklet til en enkel prototype hvor datainnsamlingskoden ble benyttet i interaksjon med annen kode som lagret og hentet data fra en database og benyttet disse dataene i statistiske beregninger.

I løpet av denne prosessen ble praktiske beslutninger med hensyn til datavalg og innsamlingskostnader dokumentert og matematiske beregninger ble utført, testet og dokumentert. Denne informasjonen danner grunnlaget for den inneværende rapporten.

2 For bestillere – sosiale mediers oppbygging og kost implikasjoner

Dette kapittelet er beregnet på bestillere av analyser fra sosiale medier som gjennom sine behov definerer dataene som må samles inn. Først beskrives et eksempel på en analyse. Denne tenkte analysen brukes for diskusjoner om kostnader ved datainnsamling, sosiale mediers relasjonelle oppbygging generelt, og hvordan dette kan brukes i analyse og som mulig fremgangsmåte for å samle inn data. Til slutt følger noen betraktninger om kostnader kontra nytte ved å samle inn forskjellige mengder data fra sosiale medier.

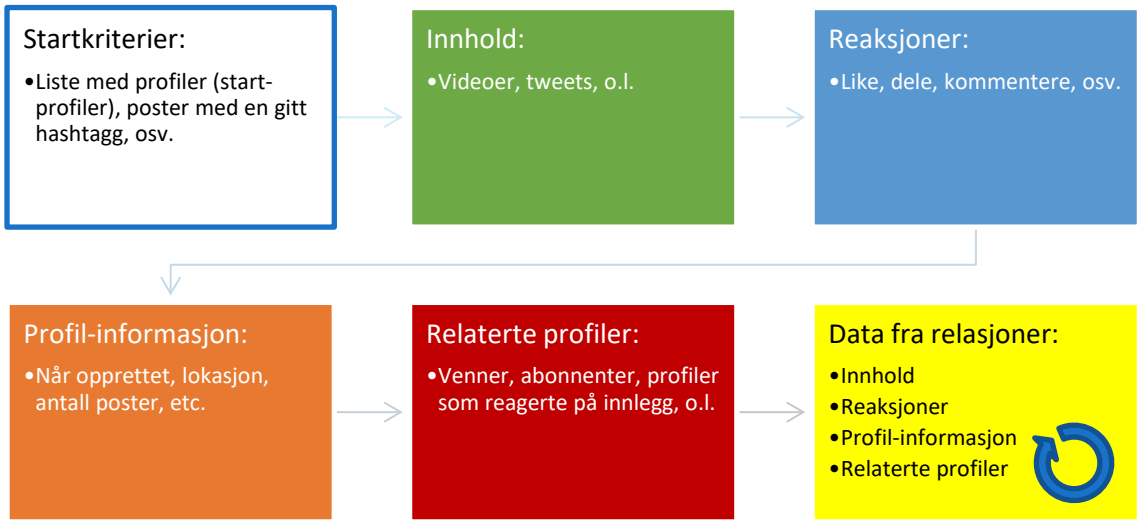
Eksempel på analyse
For å analysere sosiale medier må man først definere hva man ønsker å få ut av analysen, dette vil diktere fremgangsmåten. Et eksempel kan være å finne ut om Nato-land utsettes for koordinerte påvirkningsforsøk rettet mot deres offentlige kommunikasjonskanaler på sosiale medier. En mulig fremgangsmåte for YouTube kan være som følger:

1. oppsett av liste over forskjellige Nato-lands offisielle kanaler på YouTube
2. samle inn kommentarer på videoer lagt ut i disse kanalene for en gitt periode
3. analysere sentiment i disse kommentarene - er de positive eller negative
4. kommer negative kommentarer kun fra relativt nye profiler
5. hvilke andre kommentarer har disse profilene lagt ut andre steder på YouTube

Da sosiale medie-plattformer kontinuerlig jobber med å identifisere og fjerne desinformasjon fra sine tjenester er profilene som benyttes i påvirkningsforsøk ofte nyopprettede. Hvis mange slike nye profiler legger ut negative kommentarer kan det være mulig at det stammer fra en påvirkningsoperasjon. En slik analyse vil kun gi indikasjoner, og må støttes opp med andre analyser.

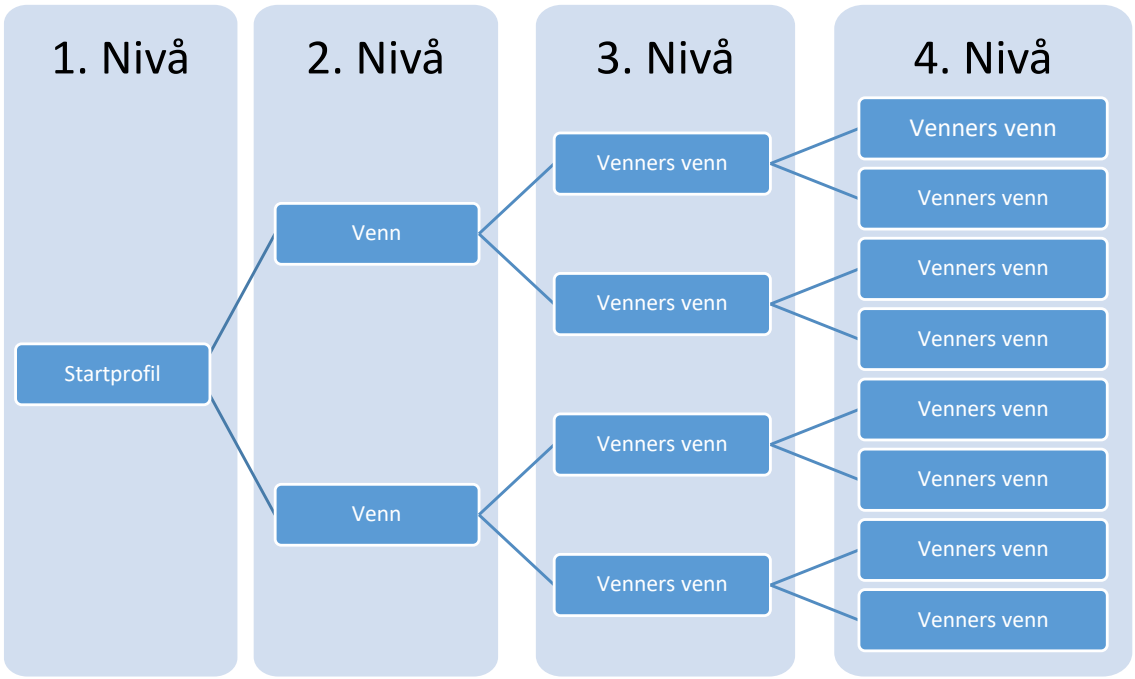
2.1 Relasjoner og interaksjoner i sosiale medier

Den viktigste kostnadsdriveren i datainnsamling fra sosiale medier er medienes relasjonelle natur, som fører til en eksponentiell økning i data som må samles inn ettersom man følger flere ledd i relasjonene. Alle sosiale medier har sine måter å bygge opp relasjoner mellom brukere av mediet på, men felles for alle er at brukere har minst en konto de logger inn med. Dette representerer deres *online identitet*, som i denne rapporten refereres til som en profil. Profilen benyttes til å danne relasjoner til andre, og/eller interagere med andre.



Figur 2.1 En generisk fremstilling av relasjoner i sosiale medier. Det siste punktet, hvor relasjoner har relasjoner, gir sosiale mediers sin eksponentielle natur.

Facebook, for eksempel, har primært "vennskap" til andre som relasjon. Interaksjon kan skje ved at en profil først deler noe, som bilde, tekst, status eller lenke. Deretter kan andre profiler kommentere, eller reagere (like, osv.) på det som ble delt, eller til og med dele videre. En generell visualisering av relasjoner og interaksjoner vises på figur 2.1. Begrensninger på hvem som kan interagere er som regel definert ut ifra relasjonen til den som først delte noe, kjent som eieren eller forfatteren. På Twitter derimot, opprettes relasjoner ved at profiler følger hverandre, det vil si at de mottar oppdateringer når den de følger legger ut nytt innhold. I motsetning til Facebook er ikke denne relasjonen toveis. Det vil si at et vennskap på Facebook er en relasjon som begge profiler må akseptere før den opprettes. På Twitter kan en profil følge en annen profil, uten at den andre profilen må godkjenne dette eller følge den første profilen. I tillegg er ikke interaksjoner begrenset av relasjonen mellom profiler, for alle kan se alle interaksjoner. Selve interaksjonene på Twitter er likevel i stor grad de samme som på Facebook, med annen terminologi. Det man deler er "tweets", og man kan svare på (kommentere) andres tweets, reagere med hjerte (like) på tweets, eller retweete (dele videre) en tweet.



Figur 2.2 Illustrasjon av profil-nivåer i sosiale medier (illustrer også en eksponentiell vekst).

I analyse-eksempelet som benyttes i dette kapittelet beskrives det å finne profiler som har interaksjoner med hva vi kaller *startprofilene*. Startprofilene er en liste med profiler valgt via definerte startkriterier. I eksempelet er det en liste over YouTube-profiler til forsvarsgrener i utvalgte Nato-land. På bakgrunn av et slikt utvalg kan det analyseres hvilke andre profiler som har interagert med startprofilene via en eller flere av de tilgjengelige relasjonene eller interaksjonene i det valgte sosiale mediet, for eksempel ved å kommentere på en video eller abonnere på en kanal. Disse profilene kan kalles 1. nivå-profiler. Nivå referer her til antall ledd man må gå fra en profil til en annen i den relasjonelle modellen. En venn på Facebook er 1. nivå, venners venner er 2. nivå, og så videre (illustrert i figur 2.2). I figur 2.1 illustreres det siste elementet i sosiale mediers sirkulære natur, det at profiler leder til profiler som leder til profiler, og så videre. Avhengig av hvilket nivå en analyse krever kan man stoppe ved disse profilene, eller fortsette for å finne profiler som interagerer med 1. nivå-profilene, og/eller den andre veien, det vil si hvilke profiler (foruten startprofilene) 1. nivå profiler interagerer med. Det er forventet at profiler interagerer med andre brukere enn kun startprofiler, så hvert nivå øker datamengden eksponentielt. Et naivt estimat er at datamengden dobles for hvert nivå – sannsynligvis vil datamengden øke mye mer.

Fremgangsmåten over for å nå 1. nivå-profiler på det sosiale mediet YouTube, og hvilken informasjon som er tilgjengelig for disse profilene er beskrevet i kapittel 3.

2.2 Kostnader ved datainnsamling fra sosiale medier

Sosiale medier er enestående i størrelsesskala. Eksempelvis har det vært registrert 1,73 milliarder daglige brukere av Facebook [15], og 2 milliarder månedlige brukere av YouTube [16]. Dette genererer store mengder data som akkumuleres over tid, det vil si datamengen er alltid økende. Det er derfor ikke mulig å samle inn et komplett datasett for å foreta analyser av sosiale medier. Enhver analyse må dermed være gjenstand for en individuell kostnad/nytte-analyse. En slik analyse må veie opp behovet for hvor grundig en analyse skal være kontra tidsrammer og tilgjengelige ressursene.

Denne rapporten er, som nevnt tidligere, ment for aktører som skal håndtere trusler mot Norge gjennom manipulasjon av informasjon i sosiale medier. Ofte vil dette innebære at analyser utføres i situasjoner hvor bestillere av analyser jobber under tidspress. Da vil den viktigste kostnaden være tiden det vil ta å samle inn selve dataen. Tidskostnaden påvirkes selvfølgelig av mengden data som samles inn, men enhver tidsbruk forsterkes gjennom sosiale medie-tjenesters begrensning i form av kvoter for å samle inn data. Alle datainnsamlere som benytter et API (se kapittel 3.2) vil ha begrensninger for hvor mye data de kan laste ned. Twitter for eksempel lar en vanlig bruker samle inn 18 000 tweets per 15 minutter. Å samle inn ca. 350 000 tweets (tilsvarende **ett minutts** aktivitet på Twitter) ville med den begrensningen tatt nesten **fem timer**.[2] Hvis man har tilgang til data som allerede er samlet inn, eller man jobber med langtidsanalyser, faller dette bort som en kostnadsfaktor.

En annen kostnad i tid er tiden det vil ta å utvikle programvaren som benyttes for å samle inn data. I kapittel 3 beskrives FFIs tilnærming for PoC-programvaren. FFI utviklet kun en enkel prototype av programvaren. For mer kompliserte analyser trengs det mer robuste og fleksible datasamlere, som vil kreve mer utviklingsarbeid. Tilgjengelige utvikler-ressurser er derfor en annen viktig faktor. Dersom man kan gjenbruke eksisterende programvare er ikke denne kostnadsfaktoren relevant for kostnad/nytte-analysen man foretar.

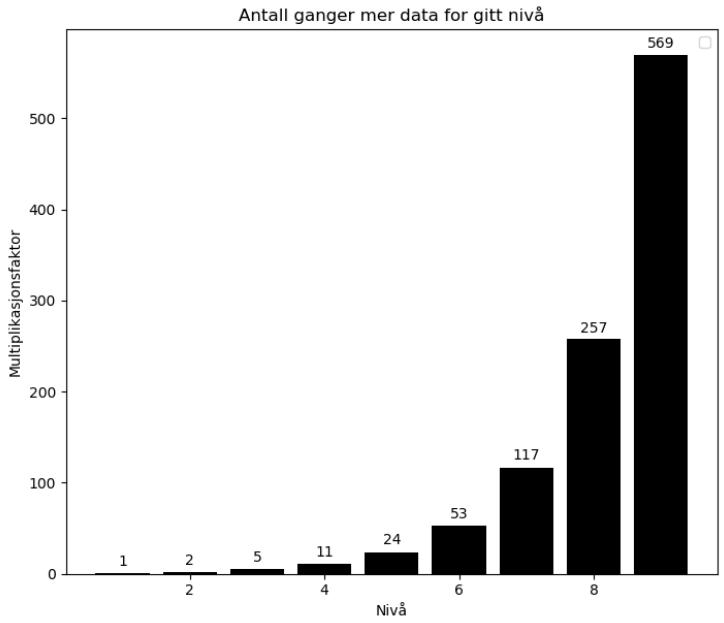
De finansielle kostnadene, i tillegg til kostnadene forbundet med utvikling, inkluderer også infrastruktur, herunder antall datamaskiner som trengs for å laste ned og prosessere data, og lagringskostnadene over tid. Bruk av offentlige nettskyer[3], der det lar seg gjøre, kan bidra til en mulig kostnadsreduksjon av infrastruktur. Disse faktorene vil aldri falle helt bort, men gjenbruk av eksisterende data og analyser vil redusere behovet.

2.3 Implikasjoner av sosiale mediers natur på kostnader

Sosiale mediers eksponentielle natur er viktig å ta i betraktning når data samles inn. Ved å bruke informasjonsmengden hentet i PoC kan vi illustrere hvordan datamengden øker ved hvert nivå. For PoC holdt vi oss til 1. nivå-profiler og informasjon det var mulig å hente om dem. Figur 2.3

[2] Basert på tall fra 2014: https://blog.twitter.com/official/en_us/a/2014/the-2014-yearontwitter.html
[3] Internett-baserte tjenester for dataprosessering hvor man betaler for tiden tjenesten brukes og ikke for maskin- eller programvare.

viser den potensielle økningen i mengde data for hvert nivå med utgangspunkt i den daglige økningen på nye profiler fra PoC. Merk at kun informasjon om video ble samlet, som for eksempel lengde, tittel, beskrivelse og lignende, men ikke selve innholdet. Nedlastning, bearbeiding og lagring av video medfører noen ytterligere problemstillinger. Videoer er vesentlig større enn den tekstlige informasjonen om dem, som krever tilsvarende større lagringsplass. I tillegg gjør størrelsen det ønskelig med større båndbredde for kortere nedlastingstid. Det må også gjøres et arbeid for å definere hva man er interessert i å hente eller finne av informasjon i videoene, og implementasjon for å ekstrahere dette fra videoen. På bakgrunn av dette ble ikke videofilene lastet ned. I en utvidelse eller ny implementasjon av PoC kan bruk av maskinlæring for å analysere innholdet i videoer være interessant, dette vil føre til store endringer i kostnad/nytte-analysen for datainnsamlingen.



Figur 2.3 Dataforholdet hentet i proof of concept programvaren (PoC) gir et eksempel på den eksponentielle naturen til sosiale medier, ved at datamengden litt mer enn dobles for hvert nivå det skal hentes informasjon om fra en bruker, og at det er behov for 569 ganger mer lagringsplass for nivå 9 sammenligned med nivå 1.

På grunn av denne eksponentielle datamengden må det avveies hva som er viktig. Samler man for eksempel inn data til det høyeste nivået vist i figur 2.3 (nivå 9) vil en hel dags standard YouTube API kvote bli brukt opp på kun en video og dataene linket til denne videoen. Kontinuerlig sanntids datainnsamling kan være viktig i krisesituasjoner for å gi oppdaterte analyser som kan bidra til et situasjonsbilde over sammensatte trusler mot norske interesser. Men dersom et tema er svært omdiskutert på sosiale medier (som covid-19-pandemien) kan det fort utgjøre enorme datamengder som skal samles inn hvis man ikke setter begrensninger. Dette vil øke kostnadene (se kapittel 2.2).

Mer problematisk vil det være at analyser sakker akterut i forhold til mengden data som skal analyseres, ettersom man stadig må samle inn sterkt økende (eksponentielle) datamengder. Lignende problemstillinger oppstår også når man skal samle inn eldre data om et nytt emne i en krisesituasjon. Man har med andre ord et etterslep av eksisterende data av ukjent mengde som man ønsker å samle inn i tandem med eventuelle sanntidsanalyser.

Denne rapporten har ingen fasit på disse problemstillingene, men ønsker å påpeke viktigheten av å ta i betraktning disse begrensningene når datainnsamling planlegges. Som en illustrasjon på denne problematikken kan man ta situasjonen i Norge da det i mars 2020 ble besluttet å «stenge ned» landet for å håndtere korona-pandemien [17]. Verdens helseorganisasjon hadde allerede uttrykt bekymringer over covid-19 des- og misinformasjon på sosiale medier [4]. Hvis norske aktører ønsket å se om desinformasjon ble spredt om pandemien i Norge kunne man fått et øyeblikksbilde ved å samle inn tweets med en gitt «hashtag». Hvis dette tenkte eksempelet hadde inkludert så mange tweets at det ville tatt en time å samle det inn, kunne en analyse skje allerede samme dag. Hvis man derimot ønsket å samle inn informasjon om alle som hadde reagert på disse tweetene ved å like eller videresende de kunne datamengden fort bli så stor at det ville tatt flere dager å samle inn alt. Da ville man altså hatt valget mellom et umiddelbart svar på om desinformasjon ble spredt i det hele tatt, eller mer detaljert informasjon flere dager senere.

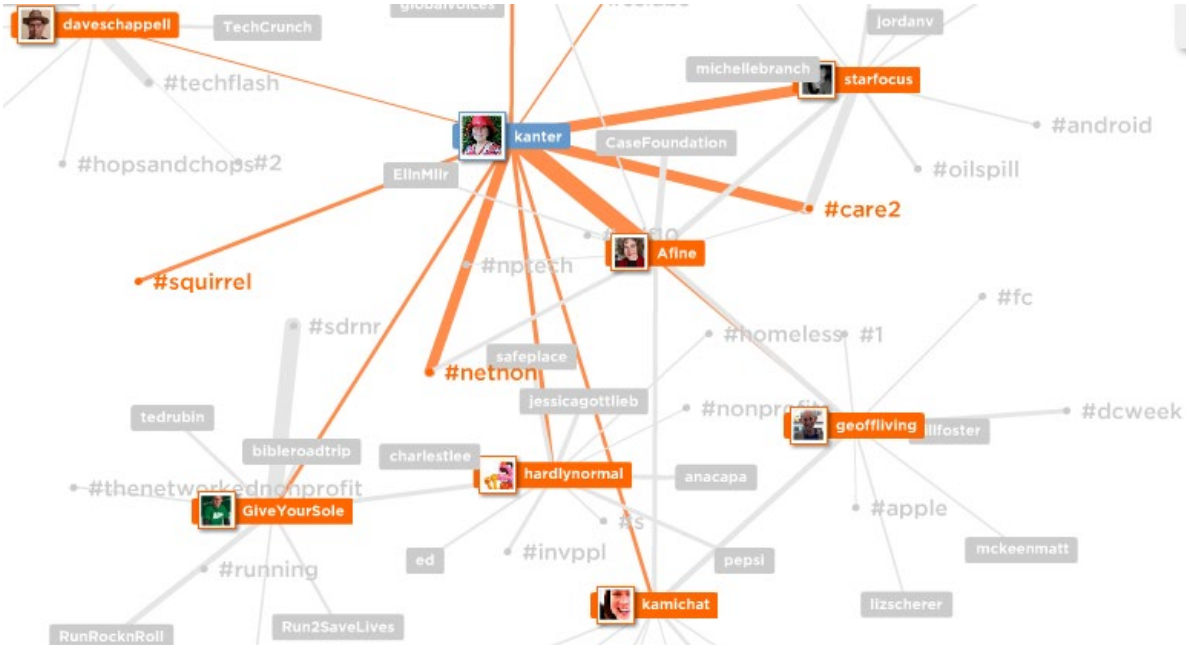
3 For utførere – eksempel og overførbarhet

Denne delen er beregnet for de som ønsker mer tekniske og implementasjonsmessige detaljer om datainnsamling, der YouTube brukes som et forklarende eksempel. YouTube ble valgt fordi det har et såkalt applikasjonsgrensesnitt som gjør det lettere å samle inn data, samt at det har en åpen «kringkastingsfunksjon» som Twitter. Det er altså ikke spesielt personfokusert, ulikt Facebook, som har strengere regler for datainnsamling. Kapittelet starter med en detaljert beskrivelse av YouTube sin arkitektur, etterfulgt av hvordan det ble implementert som *proof-of-concept* (PoC). Videre gjøres det noen beregninger og utregninger på estimert tid og kostnad en lignende implementasjonen kan kreve ved utvidelse. Til slutt foreslås det hva som bør vurderes i en eventuell utvidelse eller en ny implementasjon av funksjonaliteten i PoC.

3.1 YouTube-spesifikke relasjoner og interaksjoner

En type interaksjon som er spesifikk for YouTube (og lignende videobaserte sosiale medier) er å lagre videoer i spillelister. Det kan være spillelisten *Favoritter* som hver kanal har, eller i andre spillelister som opprettes av brukerne selv. Et viktig poeng er at *en* video tilhører *en* kanal - kanalen som lastet opp videoen. Det vil si at en video kan være i flere spillelister, men er bare lastet opp av en kanal. Dette gjør det mulig å finne kanaler som har lastet opp videoene lagret i spillelister man kommer over.

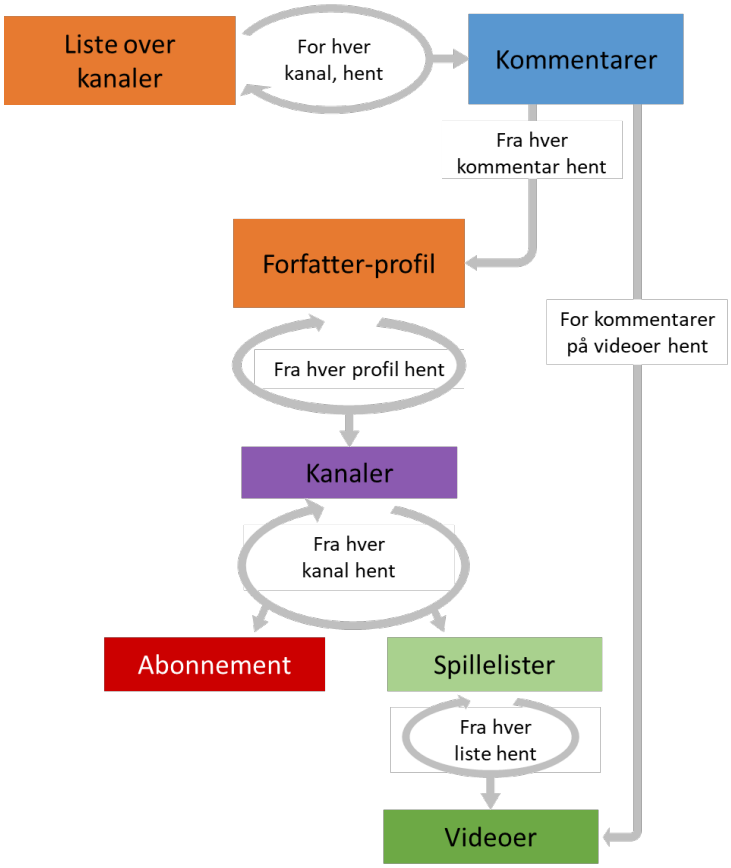
Analysen i eksempelet fra kapittel 2 kan bli forbedret med en oversikt over de som har lagt til en video i spillelisten sin, det ville gi en pekepinn over brukere som liker videoen. Dette fordi det tar ekstra tid å legge inn en video i en spilleliste, og brukere vil antakeligvis ikke lagre referanser til videoer de ikke liker.



Figur 3.1 Utsnitt av et nettverkskart som viser hvem man har kommunisert mest med (oransje linjer, tykkere er mer kommunikasjon) og hva man har snakket om (grå linjer). © Cambodia4kids.org Beth Kanter, flickr.com/photos/cambodia4kidsorg/4714104649

Det er også mulig å kommentere - primært på videoer, men i noen tilfeller også på kanaler[4]. En kommentar kan også være et svar på en annen kommentar. Kommentarer kan analyseres for sentiment, og positive/negative kommentarer er svært nyttige for å se hvordan forskjellige typer innlegg eller videoer blir mottatt. Det er også mulig å abonnere på kanaler. Dette er for å få vite når kanalen det abonneres på laster opp ny videoer eller foretar seg andre handlinger på YouTube. Siden 2016 har også kanaler kunnet legge ut innlegg på en såkalt Community-side hvis kanalen har nok abonnementer.[5] Dette innholdet er ikke tatt til vurdering grunnet vanskeligheter med å hente det ned. Abonnementsinformasjon (som tilsvarer «følgere» på Twitter) vil imidlertid kunne gi nyttig informasjon om hvem som er interessert i hva. Dette kunne for eksempel benyttes til å lage et nettverkskart (som vist i figur 3.1) for å se om de samme brukerne følger mer enn en forsvarsrelatert kanal, dette kan gi videre innsikt for analysen som ble diskutert i eksempelet i kapittel 2.

[4] Kommentering på kanal er vanligvis deaktivert, men kan aktiveres om kanalen selv tar ansvar for moderering av kommentarene.
[5] I skrivende stund er grensen 1000 abonnementer.



Figur 3.2 Dataflyt i PoC som finner 1. nivå-profiler og deres informasjon om abonnementer, spillelister og videoer.

Med de ovenstående beskrivelsene kan vi finne 1. nivå-profiler ved å begynne med et antall kanaler som tilhører startprofilene og finne alle (andre) profiler som har gjort eller gjør minst en av følgende:

1. kommentert på en video som kanalen har lastet opp
2. kommentert på kanalen, hvis kanalen har aktivert denne muligheten
3. svart (kommentert) på en kommentar fra de forrige punktene

Arkitekturen til YouTube gjør det nødvendig å få tak i en kanal for å hente mer informasjon om en profil. Derfor må det først og fremst undersøkes om profilen (kontoen) til en kommentar har en tilhørende kanal. Tilhørende kanaler har noen faste egenskaper relatert til videoer, slik som lister over opplastede videoer og favorittvideoer, samt abonnementer til andre kanaler. Annen informasjon knyttet til online interaksjoner er kommentarer på kanalen, kommentarer på kanalens opplastede videoer, og andre spillelister enn de to nevnte faste spillelistene. Informasjon som kunne vært ønskelig, men ikke er mulig å skaffe, er alle kommentarene lagt ut av en profil(/kanal) og alle profiler som abonnerer på en gitt kanal. Eier av kanal/konto kan hente denne informasjonen, men den er altså ikke tilgjengelig for andre. Dette kan i noen tilfeller føre til et

behov for å «skrape» informasjon direkte fra nettsidene til YouTube (se kapittel 3.4.2 for forklaring på skraping), avhengig av hvor viktig slik informasjon er for analysen som skal utføres. Et eksempel på en video med kommentarer, samt fargekoder for noen av relasjonene og interaksjonene for YouTube som beskrevet over vises på figur 3.3.

Figur 3.3 *Eksempel på video [18] med kommentarer farget etter relasjon – startprofilens kanal, videorelatert, 1. nivå brukere, kommentar på video, kommentar på kommentar (på video).*

3.2 Detaljert YouTube-arkitektur

En god forståelse av den underliggende dataarkitekturen er nødvendig for å lage verktøy som skal finne data for spesifikke analyser. En utvikler må bygge en bro mellom analytikeres ønskede resultater og det sosiale mediets interne data. YouTube har gode kilder for å forstå datamodell, konsepter og applikasjonsgrensesnittet (APIet) [19, 20]. Resten av denne seksjonen består derfor av ytterligere beskrivelse for hver datatype på YouTube. Det er laget en grafisk fremstilling av de viktigste datamodellene (datatypene) med relasjon og interaksjon for YouTube i vedlegg B, ved figur B.1.

Ressurs (Resource) er kun en intern fellesbeskrivelse av alle entitetene fra YouTube, og har dermed ikke en egen type fra YouTube APIet. Det betyr at der det nevnes ressurs i programkode eller dokumentasjon kan hvilken som helst av datatypene under benyttes.

Konto (Account, også referert til som en profil) representerer en "bruker" av tjenesten, og har heller ikke en egen type i YouTube APIet. Det fins to typer kontoer som kan administrere kanaler på YouTube; (personlig) Google-konto eller merkevarekonto. En Google-konto tilhører bare én bestemt person og er knyttet til ett navn og én identitet som brukes for alle tjenester fra Google, inkludert for eventuelle tilknyttede YouTube-kanaler [21]. En merkevarekonto er en konto som administreres av en eller flere Google-kontoer. Hvis en YouTube-kanal er knyttet til en merkevarekonto, kan dermed flere kontoer ha tilgang til kanalen (via merkevarekontoen). Det er også mulig at én Google-konto kan administrere flere merkevarekontoer som er tilknyttet ulike YouTube-kanaler. En konto kan ikke ha flere kanaler tilknyttet seg, men kan altså administrere flere merkevarekontoer som hver kan ha en egen kanal. Det er mulig å kommentere på en kanal eller video uten å ha en tilknyttet kanal (eller merkevarekonto med kanal). Alle kommentarer er dermed koblet til en konto, men det er ikke mulig å vite hva slags type konto som ble brukt for å legge ut kommentaren. Det er mulig å finne kanalen til en konto via en kommentar skrevet av kontoen, eller via navnet på kontoen *hvis* kontoen har en kanal tilknyttet seg.

Kanal (Channel) linker alle videoene, abonnentene og spillelistene for en profil, gruppe eller organisasjon [22], på nettsiden til YouTube er denne informasjonen vist på en side. Alle kanaler er administrert av en konto, men ikke alle kontoer har en kanal. Det er mulig å finne en kanal fra en konto *hvis* kontoen har en tilhørende kanal, men det er ikke mulig å finne en konto fra en kanal. Dette kan føre til et behov for å «skrape» informasjon direkte fra YouTubes nettsider.

Kommentar (Comment) vil være tilknyttet en kanal, enten direkte som en kommentar på kanalen eller indirekte som kommentar på en video kanalen har lastet opp [23]. Kommentaren vil da være starten på en kommentartråd (CommentThread [24]). Kommentarer kan være svar (reply) på en annen kommentar(tråd). Foreløpig har YouTube en begrensning som kun gjør det mulig å svare på en kommentar på toppnivå. Det vil si at det ikke kan svares på en kommentar som allerede er et svar til en annen kommentar. Ifølge dokumentasjonen utelukkes det ikke at det kan bli endret i fremtiden [25]. Et svar vil ha den første kommentaren i en kommentartråd som sin forelder (parent). En kommentar er knyttet til kontoen som skrev kommentaren. Hvis kontoen som

la inn en kommentar har en tilhørende kanal, vil kommentarressursen også ha informasjon om den kanalen.

Spilleliste (Playlist) er en liste med videoer [26]. Spillelister blir opprettet hos en kanal, og kan inneholde både egne videoer og videoer fra andre kanaler. Eneste unntak er en kanals spilleliste over *opplastede* videoer, dette vil kun være kanalens egne videoer.

SpillelisteElement (PlaylistItem) er koblingen mellom en spilleliste og en video, med noe ekstra informasjon [27].

Abonnement (Subscription) er en enveiskobling mellom to kanaler, der den ene kanalen følger den andre [28]. At abonnement er enveis betyr altså at de to kanalene ikke nødvendigvis abonnerer på hverandre.

Video (Video) er *opplastet* av en kanal og representerer en YouTube video [29].

Kanaler med mer enn 1 000 abonnenter har en såkalt «Community»-fane. Det er ingen datatype for innlegg på denne fanen da APIet til YouTube ikke har funksjonalitet for å hente denne typen aktiviteter fra en kanal.

3.3 Implementasjon

For å illustrere og teste hvordan det kan samles inn data fra sosiale medier er det gjennomført et bevis på konseptet (PoC) ved å implementere kode som hentet data fra YouTube via APIet og lagret det midlertidig i en database for statistisk analyse. Hvordan denne fremgangsmåten ble implementert er gjengitt her. En viktig faktor i utviklingen og tilnærmingene som ble valgt var at det skulle være mulig og enkelt å utvide implementasjonen til innhenting av data fra andre sosiale medier, samt kontinuerlig hente "live" data fra de sosiale mediene. Videre blir en overordnet beskrivelse av den implementerte prosessen forklart og vist.

3.3.1 Dataflyt

Et flytdiagram som illustrerte innhentingen av informasjon om 1. nivå-profiler ble presentert i kapittel 2.1. Kort oppsummert var flyten:

1. Det begynte med en startprofil (som for YouTube er en kanal, fordi kommentarer legges til kanaler eller videoer, og videoer er knyttet til kanaler) og tilhørende kommentarer. APIet gir mulighet for å hente alle kommentarer (inkludert kommentarer på videoer) til en kanal, men eventuelle kommentarer til kommentarer må hentes separat.
2. Hver kommentar som legges ut er knyttet til en konto, og denne kontoen kan ha en kanal. Hvis kontoen til kommentaren har en dedikert kanal er denne tilgjengelig i informasjonen som følger med kommentaren.
3. Hvis kontoen til en kommentar har en dedikert kanal kan det hentes informasjon om profilen ved å hente informasjonen om kanalen. I PoC ble det hentet informasjon om

profilens (kanalens) spillelister, abonnementer til andre kanaler, i tillegg til metadata om videoer i spillelistene.

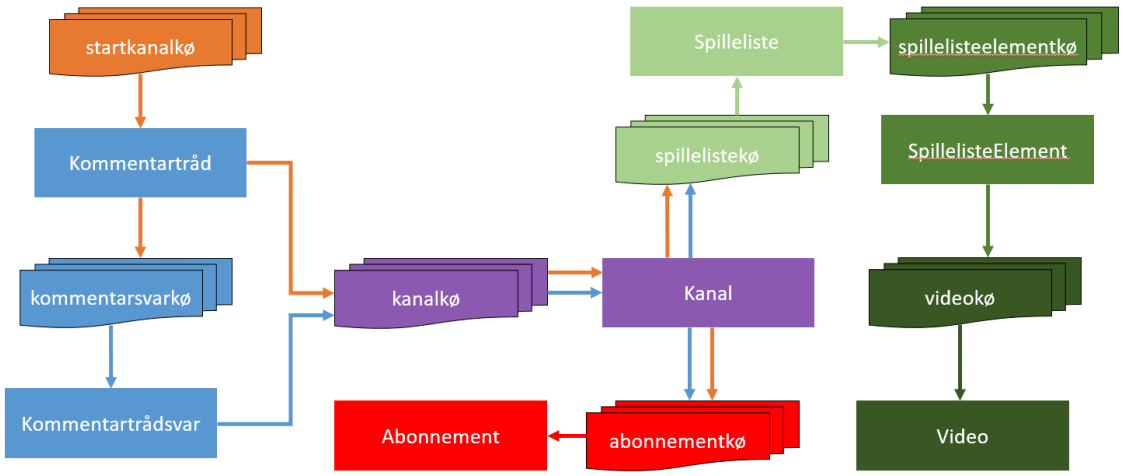
Når dette er gjort for alle startprofilene og profilene som har interagert med disse vil 1. nivå-profiler og deres informasjon være samlet. De utvalgte startprofilene og deres kanaler er gjengitt i tabell A.1.

3.3.2 Implementasjonsdetaljer

PoC ble implementert i Python som en terminalapplikasjon. Utviklingen tok ca. 10 uker. Et av formålene med implementasjonen var å legge til rette for enkel utvidelse til andre sosiale medier, samt enkelt endring av startprofilene og lignende konfigurerbar informasjon. Det ble derfor lest inn en konfigurasjonsfil med bl.a. de utvalgte startprofilenes kanaler på YouTube, informasjon om databasen det skulle lagres til, samt nødvendig påloggings- og autentiseringsinformasjon. Metoden som ble brukt for å hente data fra YouTube var API-kallet *list* på en datatype. Videre i rapporten blir derfor uttrykket "listing" brukt for å beskrive prosessessen for å hente data fra YouTube. For å parallelisere henting av data, samt tilrettelegge for et bredere søk, ble arbeidet fordelt på tråder (parallellprogrammering). Hver arbeidstråd jobbet primært med en bestemt datatype, med en egen tilhørende kø for oppgaver. Unntaket her er arbeidstrådene som hentet kommentarer. Disse arbeidstrådene hentet ikke bare informasjon om kommentarer, men også informasjon om de tilhørende kanalene til kommentarene, når det var mulig. Utover dette førte alle arbeidsoppgavene, foruten video og abonnement, til arbeidsoppgaver for andre arbeidstråder. Det vil si at arbeidstråden som hentet foreldrekommentarer fyller opp køen til arbeidstråden som hentet svar på kommentarer. Begge arbeidstrådene for kommentarer ga arbeidsoppgaver til arbeidstrådene for abonnement og spilleliste. Arbeidstråden for spillelister fylte opp køen for spillelisteelement-oppgaver, som igjen fylte køen til arbeidstråden for videoer. Oversikt over arbeidstråder og køer er illustrert i figur 3.4, der samme farge på pilene representerer samme arbeidstråd. De opprettede køene for å hente data fra YouTube var altså:

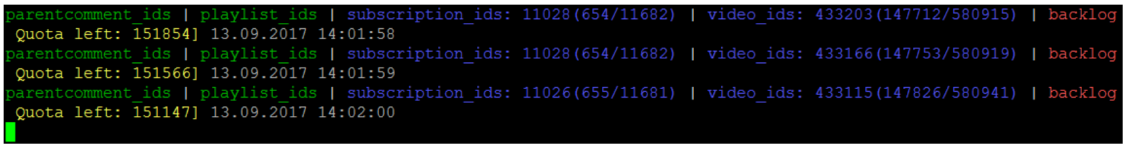
Fra startprofilenes kanaler:
- foreldrekommentarer
- svar på foreldrekommentarer

Fra 1. nivå-profilene:
- abonnementer
- spillelister
- spillelisteelement
- video



Figur 3.4 Dataflyt i PoC med fokus på køene og pilfarge for arbeidstråder

Om programmet ble avsluttet før alle køene var tømt, ble oppgavene som ikke var ferdigbehandlet lagret til en egen fil. Om programmet startet igjen ble denne filen automatisk lastet inn og køene fylt opp med oppgavene fra forrige kjøring. Under kjøring viste terminalapplikasjonen status ved hjelp av fargekoder, som vist i figur 3.5. På data som ble hentet ble det gjennomført enveiskryptering på utvalgte felter med personopplysninger, som for eksempel profilnavn. Dette gjorde det mulig å beholde unikheten til en profil, men samtidig unngå at man enkelt finner tilbake til den ekte profilen fra innholdet. Data var dermed pseudonymisert når de ble lagret. Data ble kun lagret for å gi et beregningsgrunnlag, og for å realisere relasjonsmodellen. Alle data har i ettertid blitt slettet.



Figur 3.5 Et utsnitt av terminalen under kjøring. Hver kø representeres av '(..)ids', der fargekoden beskriver aktiviteten. Blå er aktiv, rød er inaktiv, grønn er tom kø, gul er informasjon og hvit er tidsstempelet.

3.4 Tekniske og utviklingsrelaterte utfordringer ved videreutvikling

Det ble kun implementert henting av informasjon fra YouTube i PoC. Ved videreutvikling anbefales følgende to punkter:

- Robust, kontinuerlig og automatisk innsamling
- Videreutvikle PoC for andre sosiale medier (diskutert i kapittel 3.4.2)

Det gjøres et estimat på kostnader i vedlegg C for det første punktet. Siste punktet, videreutvikling for andre sosiale medier, vil bygge på grunnlaget fra YouTube-implementasjonen når det

første punktet er forbedret. På bakgrunn av dette er utvidelse av innsamlingsverktøy for andre sosiale medier ikke forventet å kreve like lang tid eller kostnad som det tar, eller vil ta, å implementere eller utvide henting fra YouTube.

3.4.1 Kontinuerlig og automatisk innsamling (på YouTube)

I kontinuerlig og automatisk innsamling legges det her vekt på at innsamlingen skal foregå hele døgnet (kontinuerlig), og at applikasjonen styrer hvordan og hva som samles inn (automatisk) på bakgrunn av en gitt konfigurasjon. I følgende avsnitt redegjøres det vurderinger som må gjøres for en slik konfigurasjon, samt beskrivelse av kostnad i form av kvote for YouTube. Til slutt drøftes noen innsamlingsstrategier.

For automatisk innsamling må det først bestemmes hvor *ofte* det skal sjekkes for nye responser til allerede innsamlede videoer og kanaler, samt hvordan og hvor ofte ny data skal letes etter. Det siste leddet er avhengig av analysen som skal finne sted, og mediets oppbygging. I tillegg må det defineres handlinger som skal trigges hvis informasjon i lagret data er annerledes fra det som hentes. Fremgangsmåten for hvordan ny data blir funnet i PoC er beskrevet tidligere, og det ble ikke implementert hverken handlinger eller triggere til handlinger i PoC. I PoC ble eksisterende data oppdatert med ny informasjon etter hvert som det ble samlet inn.

For å estimere kostnader er det gjort grove utregninger på to enkle scenarier, basert på hyppighet og på hvor mye som skal sjekkes. Mer komplekse scenarier må sannsynligvis evalueres ved en eventuell utvidelse. Utregningene tar utgangspunkt i maskinvarekrav og kvotekrav. Kvoter i denne sammenhengen er, som diskutert i kapittel 2.2, begrensninger satt av sosiale medie-selskaper for hvor mye data man kan laste ned innenfor en viss tidsramme. Hos YouTube er standardkvoten i skrivende stund på 10 000 kvoteenheter (ψ) i døgnet [30]. Det er mulig å søke YouTube om større kvote gjennom en søknad med bl.a. utfyllende begrunnelse for *hvorfor* det er nødvendig med større kvote, og hva man bruker APIet til. Det ser ikke ut til å være direkte kostnader, utover det å ta kontakt med YouTube, skulle et forsøk på å øke kvoten gjennomføres. Under implementasjon av PoC var det tilgang på en konto med 50 000 000 ψ i døgnet. Denne kvoten brukes i videre beregninger, da det kun er kvote på denne størrelsen som er nær nok det som er nødvendig uavhengig av sjekkhyppighet.

3.4.1.1 Sanntid sjekkhyppighet

I sanntidsscenariet skal all data som tidligere er hentet sjekkes for endringer opp mot YouTube hvert sekund. Det er ikke forventet at dette vil bli aktuelt, da det vil kreve for stor YouTube-kvote. I tillegg er det sjeldent data endrer seg så hyppig at det er nødvendig med sjekk i sanntid. Det er likevel gjort en estimering, da det danner grunnlaget for utregninger i de andre scenariene.

Vedlegg C.1 har noen overslag vedrørende maskinvarekrav. Med bakgrunn i overslagene, og størrelsen på mengde data som ble hentet i PoC, kan vi si noe om hva som kreves for å sjekke YouTube innlegg i sanntid. På sikt vil et cluster for lagring, flere kjerner/prosessorer, og større mengde minne for innsamling, være gode og muligens nødvendige investeringer. Det er ikke

gjort vurdering av nettverkskrav på grunn av utfordringen med kvote. Grunnet den store datamangden, med påfølgende daglige økninger, er det meget kvotekrevende å gjennomføre sjekk i sanntid. Det er beregnet å koste omtrent 65 596 000 ψ av kvoten *i sekundet* å sjekke databasen i sanntid, som er 6 560 ganger mer enn standardkvoten, eller 1,3 ganger mer enn den største (daglige) kvoten tilgjengelig. Den største kvoten ville gjort det *nesten* mulig å sjekke alle entiteter daglig, men langt ifra hvert sekund. Dermed må enten kvoten økes drastisk, eller sjekkhyppighet begrenses.

3.4.1.2 Begrenset sjekkhyppighet

Ved begrenset sjekkhyppighet er det estimert for to enkle varianter; likestilt viktighet og foretrukket viktighet.

Med likestilt viktighet vil alle entiteter sjekkes like ofte. Dette intervallet er satt til *hver andre dag* på bakgrunn av kvotekostnaden i sanntidsscenariet og den største kvotestørrelsen.

I foretrukket viktighet er noen entiteter, som for eksempel en ny kommentar, angitt større interesse, og sjekkes derfor hyppigere enn andre. Standard sjekkrate er daglig, hver tredje dag, eller hver femte dag. Samtidig som det i perioder kan foregå hyppigere sjekk som hvert minutt eller hver time. Et forslag til sjekkrater er listet i vedlegg C.2.

Grunnet den lave mengden data sammenlignet med sanntid er det ikke forventet større krav til maskinvare i disse scenariene. Det er også betydelig mindre krav til kvote når sjekken er per dag og ikke per sekund. Likevel er det fortsatt for mye innhold til å sjekke alt minst en gang om dagen – selv med den største daglige kvoten som har vært tilgjengelig. En økning i kvoten bør utredes hvis det er ønskelig å sjekke entiteter ofte.

3.4.2 Overførbarhet av proof of concept

De fleste sosiale medier følger et lignende mønster som YouTube når det gjelder relasjoner (se kapittel 2.1 og 3.1), men noe som skiller de fra hverandre er fokus for mediet. Facebook vektlegger vennskap og mulighet til å følge med på hva venner holder på med, siden vennskap går begge veier og det meste kan deles. For Twitter er fokus små budskap - siden følgere i hovedsak er enveis, og en tweet som deles er en avgrenset tekstblokk, eventuelt med bilde. Andre media, som Instagram, har fokus på å dele bilder. Selv om relasjon- og innholdsfokus kan være forskjellig, har alle medier til felles å legge til rette for interaksjoner via relasjoner. Det betyr at arbeidet rapporten har gjort gjennom vurderinger, fremgangsmåter, og eksempler på YouTube, kan for det meste overføres til andre sosiale medier.

YouTube har et såkalt offentlig API. Et API er som tidligere beskrevet en standardisert liste over kommandoer som utviklere kan bruke for å sende eller forespørre data fra en tjeneste. For å få en liste over de 25 mest populære videoene på YouTube kan kommandoen *chart=mostPopular&regionCode=NO&maxResults=25* sendes til YouTube. YouTube sender da tilbake en liste med disse videoene.

Sosialt medium	Kjerneformat	Relasjon	Datainnsamling	FFIs PoC relevans
Instagram	Bilder	Følgere	Begrenset API, Skraping	Noe
Twitter	Kort tekst	Følgere	API	Stor
Facebook	Tekst / Alle	Venner	Skraping	Liten
YouTube	Video	Seere, abonnement	API	Stor
TikTok	Video	Seere, følgere	API	Stor
Pinterest	Bilder	Følgere, boards	Begrenset API, Skraping	Noe
LinkedIn	Tekst	Kontakter	Begrenset API, Skraping	Noe
Reddit	Tekst	Delta i diskusjon	API	Stor

Tabell 3.1 Enkel oversikt over populære sosiale medier, og hvor relevant FFIs PoC kan være for å hente informasjon fra dem.

Når et sosialt medium har et slikt API er det relativt lett å samle inn data. Om et sosialt medium derimot ikke gir tilgang på denne måten må det *skrapes* data. Å skrape data innebærer å benytte en annen programvare som laster ned nettsider akkurat som i en nettleser. Deretter brukes visse metoder for å ekstrahere informasjon fra nettsiden, som for eksempel kommentarer eller bilder. Denne fremgangsmåten krever mer utviklingstid, er tregere i bruk enn et API, og er mindre presist når det gjelder data man får ut. Tabell 3.1 viser om et utvalg populære sosiale medier har API-tilgang eller ikke, og derav hvor relevant PoC kan være for å hente informasjon fra mediet.

Den største forskjellen mellom disse to datainnsamlingsmetodene er overførbarhet av tilnærmingen beskrevet i denne rapporten til et annet sosialt medium. Uansett metode er datainnsamlingen relasjonell, det vil si at venner/følgere/kontakter er koblet sammen. Ved bruk av skraping må man selv kartlegge koblinger mellom forskjellige relasjoner og laste ned relatert data, typisk ved å følge lenker på en nettside. Det er utenfor rammen for denne rapporten å beskrive bruk av andre APIer enn YouTubes, eller forklare teknikker som benyttes for å skrape data. Her henvises det til fagbøker som for eksempel *Mining the Social Web: Data Mining Facebook, Twitter, LinkedIn, Instagram, GitHub, and More* [31].

4 Konklusjon

Denne rapporten har diskutert problemstillinger knyttet til innsamling av data fra sosiale medier for å få en situasjonsforståelse av mulig påvirkning og desinformasjon i sosiale medier rettet mot norske interesser. Dette drøftes fra to innfallsvinkler. Den første er fra bestillers synspunkt. Bestiller er her den som har behov for analyser av sosiale medier, for eksempel for å avdekke påvirkningsforsøk. Det er derfor bestiller som vil sette rammene for hvilke data man trenger å samle inn, dette ble diskutert i kapittel 2. I kapittel 3 ble det diskutert hvordan datainnsamling

kan gjøres rent praktisk, sett fra en utførers ståsted. Utfører er vanligvis en programmerer eller databaseekspert. Problemstillinger som angår utførere har vært belyst gjennom en diskusjon av FFIs konsepttest (proof of concept) som samlet inn testdata fra YouTube.

For å hente inn relevant informasjon for analyser av sosiale medier må innsamling spisses inn mot analysen som ønskes utført. Det vil si at det er nødvendig for bestiller å vurdere balansen i forholdet mellom hvor oppdatert og detaljert datagrunnlaget for en analyse må være, opp mot mulighetsrommet som defineres av tilgjengelige ressurser. For konsepttesten var ressursbegrensning først og fremst kvoten hos det aktuelle sosiale mediet man ønsker å jobbe med, men maskinvare kan være en faktor i andre situasjoner.

I tillegg er det viktig å vurdere hvordan innsamlingen skal finne relevant informasjon, altså hvilke utvalgskriterier som for eksempel plattformer, tidspunkt og søkeord innsamlingen skal starte fra, og hvilke relasjoner det er relevant å utforske. I prinsippet vil det være ønskelig å utforske alle relasjonene som et sosialt medium legger til rette for, men dette vil som regel føre til for store datamengder, noe som vil være upraktisk fordi det tar for lang tid å samle inn eller analysere dataene. Linket til dette er beslutninger om hvor mange nivåer ut i relasjonene innsamling skal foregå. Analysen definerer om det er viktigst å få mest mulig informasjon nært utgangspunktet (ikke dyp), eller dypest mulig informasjon gjennom relasjoner, samtidig må analysen ta høyde for hva som er mulig å gjennomføre praktisk sett. Relasjoner og koblinger er essensen av sosiale medier, og er derfor en viktig del av enhver analyse. Men hvert nivå som analyseres vil øke datamengden eksponentielt. Rapporten diskuterte et eksempel fra datainnsamlingen hvor man fant at når ett innlegg resulterer i 0,1 megabyte data ga ikke 50 innlegg 5 megabyte med data, men nærmere 12 megabyte.

Denne vurderingen av hva som skal analyseres, i hvilken periode, og på hvilken måte, vil bidra til en fornuftig og riktig balanse i innsamlingsløsningen. Dette gjøres best i dialog mellom bestiller og utfører, som sammen bør foreta en avveining av hva det er mulig å oppnå innenfor de føringer som tid, kostnader og utviklings- og databehandlingsressurser gir.

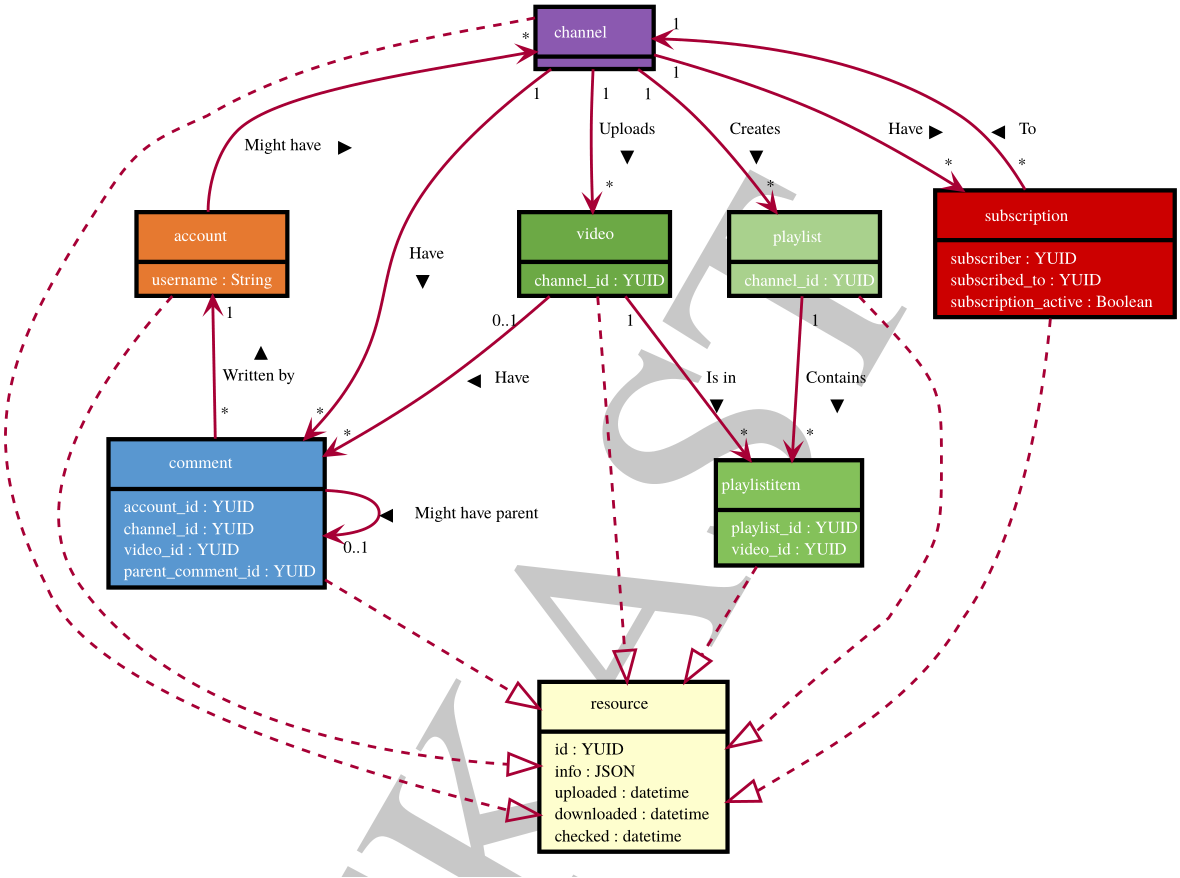
Vedlegg

A Analyseutvalg

Land	Avdelinger på sosiale medier	Forklaring
Tyskland	alle i en	Større Europeisk NATO land, men relativt lite aktivt utenfor landet. Relativt lite bruk av sosiale medier
Nederland	alle i en	Lite land i NATO (som FFI også samarbeider med i andre sammenhenger), ikke involvert i egne operasjoner. En del bruk av sosiale medier.
Storbritannia	land, luft, vann, byråkrati	Involvert i flere egne operasjoner, NATO land med historiske knytninger til mange land. En god del bruk av sosiale medier.
USA	land, luft, vann, byråkrati	Største NATO-land, involvert i mange egne operasjoner utenfor landet. Utstrakt bruk av sosiale medier.
Sverige	alle i en	Sammenlignbart med Norge, men utenfor NATO. En god del bruk av sosiale medier.
Danmark	alle i en	Sammenlignbart med Norge, veldig lite forsvar, innenfor NATO. Lite bruk av sosiale medier.

Tabell A.1 Oversikt over landene valgt som startprofiler på YouTube, med tilhørende begrunnelse. USA og Storbritannia er delt i 4 avdelinger hver på sosiale medier, mens resterende 4 land har kun en konto for forsvaret i sin helhet.

B Datarelasjoner i YouTube



*Figur B.1 Oversikt over relasjonene i datamodellen for YouTube brukt i PoC. YUID står for YUID står for **Y**outube**U**nique**ID** og er en intern unik ID hos YouTube. Hele linjer er avhengigheter, og stiplede linjer indikerer arv.*

C Beregninger for innsamling av data fra YouTube

I vedlegg C gjøres en overordnet beregning av forventet tid og ressurskostnad basert på informasjon hentet i et avgrenset tidsrom. Det er en beregning for uthenting av data i sanntid og to forskjellige beregninger for uthenting av data over tid. De forskjellige utregningene er gjort for å få en oversikt over hva som kan være forventet behov for regnekraft og kvoter på sosiale medier, avhengig av datamengde og oppdateringsfrekvens. Kvotekostnadene det er beregnet for var gjeldende i forrige versjon av APIet (v2).

C.1 Databasestørrelse og forventet vekst

For å ha et felles utgangspunkt å beregne forskjellige tid og kvotekostnader på, benyttes datamengden som ble samlet i tidsspennet for PoC. Denne datamengden ble hentet gjennom dataflyten i figur 3.2 fra startprofilene, og tar derfor utgangspunkt i antall kommentarer i perioden. Ved å ta utgangspunkt i kommentarer kan man da bygge opp en forventet gjennomsnittlig økning i datamengde, basert på perioden fra første kanal ble opprettet frem til siste henting av data fra YouTube. Merk at all aktuell data ikke nødvendigvis ble hentet ned, men datamengden gir likevel et tilsynelatende godt nok statistisk grunnlag.

Type	Antall unike ID'er	Daglig vekst
Abonnement (Subscription)	2 877 323	649
Kanal (Channel)	45 259	10
Kommentar (Comment)	100 399	23
Konto (Account)	57 266	13
Spilleliste (Playlist)	61 908	14
SpillelisteItem (PlaylistItem)	3 742 474	845
Video (Video)	2 013 101	454
Totalt	8 897 730	2 008

Tabell C.1 Oversikt over data som eksisterte i databasen pr. 20.09.2017, og beregnet daglig vekst i tidsrommet mellom 04.08.2005 og 20.09.2017

Tidsspennet på innsamlet data fra første kanal ble opprettet til siste data ble hentet ut var fra 04.08.2005 til 20.09.2017, og mengden data som ble hentet vises i tabell C.1. Det må presiseres at det fortsatt var mer data å hente blant abonnementer, spilleliste-elementer, og videoer, når datahenting ble avsluttet. Resterende datatyper, som er kanaler, kommentarer og spillelister, er forventet å være tilnærmet riktig. Det betyr at mengden for de uferdige datatypene er et pessimistisk anslag, men benyttes videre for å ha et statistisk grunnlag. Fra start til slutt var tidsspennet 4 430 dager, mens det totale antallet kommentarer i perioden var på 100 399. Dette gir omtrent 23 kommentarer daglig. Det var 57 266 profiler, som gir en økning på rundt 13 profiler daglig. Det unike antallet kanaler det var mulig å få en ID på var 45 259. Antallet kanaler hver dag blir da ca. 10. For profiler med tilhørende kanal ble mer informasjon hentet. Denne informasjonen

er kanalens abonnementer og de to spillelistene; opplastede-, og favoriserte videoer (og spillelisteelementene som kobler en video til en spilleliste). Ikke alle kanaler laster opp videoer, eller oppretter en favorittspilleliste, så antallet varierer. Derfor er snittet av 61 908 spillelister totalt omtrent 14 spillelister pr. dag. I tillegg inneholder hver spilleliste et varierende antall videoer (og deres spillelisteelementer). Et viktig punkt å huske på er at forskjellige spillelisteelementer kan representere samme video, men i forskjellig spillelister. Derfor er det flere spillelisteelementer enn videoer. Med 3 742 474 spillelisteelementer blir den daglige økningen i spillelisteelementer 845. For 2 013 101 videoer blir det daglig 454 nye videoer. Avslutningsvis hentes også alle abonnementer til hver nye kanal. Fra perioden ble det hentet 2 877 323 abonnementer totalt, som blir 649 abonnementer daglig.

Den totale daglige veksten i entiteter estimeres derfor til 2 008. Totalen og hver enkelt datatypes økning er også synlig i tabell C.1.

Krav til maskinvare og YouTube-kvote ved sanntid

Sanntid vil si at en endring skal oppdages tilnærmet "med en gang", som her er definert til innen ett sekund. De to kvantitative kostnadene ved å gjøre sjekk på YouTube entiteter er datakraft (maskinvare) og YouTube-kvote (nedlastningsgrense).

Maskinvare er kanskje det enkleste å gjøre noe med - da det kun trengs å oppgradere ved å bytte ut enkeltkomponenter eller å gå over til serverpark. De typiske komponentene det stilles krav til er prosessor, minne, og lagringsstørrelse. Dessverre er det ikke gjort en profilering, så for å utrede krav til maskinvare for prosessor gjøres det et estimat på antall operasjoner prosessoren kan måtte gjøre i løpet av 1 sekund for å sjekke og oppdatere hele databasen. Gitt tallene fra perioden er det i sanntid behov for å sjekke 8 897 730 entiteter i sekundet. Da dette arbeidet krever forbindelse til server og oppdatering av database er det nødvendig å benytte CPU og ikke GPU til denne jobben. Uten profilering tas det utgangspunkt i et kernell-kall på 1 500 cycler, eller instruksjoner, som estimat på en moderne CPU å foreta sjekk, nedlastning, og oppdatering [32]. Da er det behov for $8\,897\,730 \times 1\,500 = 13\,346$ millioner instruksjoner per sekund. En vanlig stasjonær prosessor er typisk mellom 2,3–3,8 GHz, med 2 eller 4 kjerner. Med de tallene har en vanlig stasjonær maskin i teorien mellom 5 200–15 200 GHz, eller millioner instruksjoner pr. sekund.

Det vil si at en maskin uten prosessor i toppsjiktet i teorien ikke vil klare å gjøre de nødvendige instruksjonene på mindre enn et sekund. Samtidig krever det lite utvidelse i hardware for å oppnå bedre ytelse i både kort-, og langtidsperspektiv. Likevel er det ikke opplevd at det går så få instruksjoner som 1 500, så dette utgangspunktet kan være satt for lavt. Det kan også være på grunn av nettverksforbindelsen (til YouTube), som gjør at det brukes instruksjoner på ingenting mens maskinen venter på svar fra nettverket. Uten en dypere analyse av det ovennevnte er det ikke mulig å fastslå årsaken. Vanlig minne på stasjonære maskiner er vanligvis fra 8 GB til 32 GB, men kan både være lavere eller høyere. Hvis alle entitetene skal være lagret i minne til enhver tid må det utredes hvor mye plass en entitet tar. Uten profilering må det gjøres noen estimater, og størrelsen på JSON strukturen til en tilfeldig kanal er på 1,7kB. Hvis det videre antas at hver av entitetene trenger samme størrelse og skal oppbevares i minnet trengs det minst

$15\,123\,141\,000B$, eller $15GB$ minne. Det betyr at en stasjonær minne med minst $16GB$, helst mer, trengs for å gjøre oppdatering i sanntid. Siste vanlige komponent er lagringsplass, men det er ikke gjort noen utredning på krav eller kostnad for dette. Grunnen til det er at hvis det er mulig å ha all dataen i minnet, er det verken kostbart eller et problem å ordne samme mengde eller mer persistent lagringsplass. Avslutningsvis er det ikke forventet at det skal gjennomføres sjekk i sanntid for så mange entiteter på bakgrunn av kravene til kvotekostnader i neste avsnitt. Da er det ikke nødvendig med maskinvare i toppsjiktet, og maskinvarekravene kan nedjusteres.

Listeressurs	listing	deler(kvotekostnad ψ)	total ψ
channel	kanal fra id	id(1), snippet(2), statistics(2), status(2), contentDetails(2)	9
channel	kanal id fra username	id(1)	1
comment	en kommentar fra kommentar id	id(1), snippet(1)	2
comment	alle svar på en kommentartråd id	id(1), snippet(1)	2
commentthread	alle kommentartråder for en kanal	id(1), snippet(2)	3
commentthread	alle kommentartråder fra en video	id(1), snippet(2)	3
playlist	playlist fra id	id(1) snippet(2)	3
playlistitem	en playlistitem fra playlistitem id	id(1), snippet(2), contentDetails(2), status(2)	7
playlistitem	alle playlistitems fra en playlist id	id(1), snippet(2), contentDetails(2), status(2)	7
video	video fra id	id(1), snippet(2), contentDetails(2), topicDetails(2), localizations(2)	9
subscription	subscription fra id	id(1), snippet(2), contentDetails(2), subscriberSnippet(2)	7

Tabell C.2 Oversikt over kvotekostnadene for å gjør en listing fra YouTube.

YouTube-kvote er det andre kravet, og er blant annet en fast daglige kvote fra YouTube (ψ). I skrivende stund får man en kvote på rundt $10\,000\ \psi$ ved opprettelse av en ny YouTube konto. I tidligere prosjekt har det vært tilgang til kontoer med 10 og 50 millioner i kvote. Det trekkes av kvoten når det hentes data fra YouTube, som kalles å gjøre en *listing*. Hver eneste listing som blir gjort koster mellom $1–9\psi$ fra kvoten, og er gjengitt i tabell C.2 for direkte listing av en entitet. Noen av listingene trengs imidlertid kun å gjennomføres ved sjekk i ettertid, fordi første gangen kommer informasjonen "gratis" via en annen listing. Eksempel på dette er kanalen som kommer med kommentarer når man spør om alle kommentartrådene til en kanal/video. En kompleksitet på kvotekostnaden er varierende maks-begrensning ved listing av flere entiteter. Eksempel på dette er kommentartråder fra kanal/video: ved listing får man maks 100 kommentartråder pr. listing. Hvis det er 300 kommentartråder må det da listes tre ganger og betales en

kvotekostnad på totalt 9ψ, altså 3ψ for hver listing. Basert på den daglige økningen ser ikke dette ut til å skje ofte. Det kan derfor antas at det ikke er nødvendig med flere listinger for å hente ny data i løpet av en dag. Oversikten over kvotekostnadene er for å sjekke *en* entitet direkte. Det som derfor vil koste stadig mer er å sjekke *all* informasjonen i databasen via sjekk på en-og-en entitet. Å gjøre sjekk på alt som det ble hentet fra YouTube i PoC med denne teknikken vil koste $65\,598\,000\ \psi$, og kan sees i tabell C.3. Skal dette gjøres i *sanntid* slår en ny grense fra YouTube til: maks kvotebruk på $180\,000\ \psi$ per 100 sekund. Dette betyr at det for sjekk i sanntid er behov for flere kontoer for å unngå å sprenge en eller begge grensene fra YouTube.

På bakgrunn av det høye kravet til kvote i dette eksempelet er det ikke ventet at sjekk i sanntid vil være ønskelig, og kanskje ikke mulig.

Type	Antall	ψ per listing	total ψ
Abonnement (Subscription)	2 877 323	7	20 141 261
Kanal (Channel)	45 259	9	407 331
Kommentar (Comment)	100 399	2	200 798
Spilleliste (Playlist)	61 908	7	433 356
SpillelisteItem (PlaylistItem)	3 742 474	7	26 197 318
Video (Video)	2 013 101	9	18 117 909
Total	8 840 464		65 497 973

Tabell C.3 De totale kvotekostnadene for å sjekke alle unike entiteter fra YouTube. Tilsvarende hva det vil koste av kvoten hvert sekund med sjekk i sanntid.

C.2 Forventet behov med begrenset sjekkhyppighet

Som nevnt tidligere vil det være vanskelig å gjøre sjekk i sanntid. Heldigvis er det sannsynligvis ikke nødvendig med sjekk i sanntid uansett da de fleste entiteter endres sjeldent til aldri. I tillegg spiller andre faktorer spiller også inn. Et eksempel kan være ny kommentar. Det kan forventes større sannsynlighet for at noen vil svare på den nye kommentaren i nær fremtid enn at en eldre kommentar får et nytt svar. Samme gjelder forøvrig sannsynligheten for nye kommentarer på en nylig opplastet video enn en gammel video. Ved å gjøre fornuftige antakelser kan det kuttes betydelig i nødvendige ressursbehov.

For å regne ut hvor ofte en entitet skal sjekkes, bør det gjøres vurderinger på:

- hvor ofte det er ønskelig å sjekke
- hvor mange entiteter er det, og forventes det å komme
- hvor mye det totalt vil koste å sjekke

Type	Antall (k)	Kost (ψ)	Total kost ($k\psi$)	Antall daglig (k)	Daglig kost($k\psi$)
Abonnement	2 877	7	20 141	1 439	10 072
Kanal	45	9	407	23	204
Kommentar	100	2	201	50	100
Spilleliste	62	7	433	31	217
SpillelisteItem	3 742	7	26 197	1 871	13 099
Video	2 013	9	18 118	1 007	9 059
Total	8 840		65 498	4 420	32 750

Tabell C.4 Kvotekostnaden ved å sjekke hver entitet i løpet av 2 dager, og hva det koster av kvoten daglig.

Kvotekostnaden for å sjekke alle entiteter er estimert tidligere i tabell C.3. Estimatet var i sanntid, men total kvotekostnad vil være den samme selv om den er spredt utover en periode. Den daglige kvoten på $50\,000\,000\psi$ er brukt i resten av beregningene.

Eksempel på likestilt sjekk

Ved likestilt sjekk vil alle entiteter bli sjekket like ofte, i eksempelet er dette annenhver dag. Det vil koste rundt 66 % av den daglige kvoten å sjekke halve databasen daglig, gjengitt i tabell C.4. Med overskuddet på 34 % er det mulig å gjør hyppigere prefererte sjekker av eksempelvis startprofilenes kanaler, og legge til eventuelle nye forandringer som beskrevet over. Det gir også mulighet til å hente og oppdatere andre deler av databasen når en sjekk på en entitet viser seg å være forandret. Overskuddet gir også rom for den kontinuerlige økningen over tid.

Eksempel på preferansebasert sjekk

Med preferansebasert sjekk kan entiteter som anses viktigere eller med antatt hyppigere endringer sjekkes oftere. På alle ressurser indikerer feltet '`publishedAt`' når entiteten "ble til" hos YouTube. På kommentarer fins det også et '`editedAt`' felt som indikerer når siste endring fant sted. Siden den kun indikerer *siste endring* trengs flere sjekker av kommentaren i kort tidsrom for å fange opp hyppige endringer. Ressurser uten '`editedAt`' feltet må sjekkes både før og etter en endring/sletting/avslutning har funnet sted for å bli fanget opp. De to entitetene det er viktigst i PoC å finne endringer på er startprofilenes kanaler og tilhørende kommentarer. Grunnen til dette er at det er ønskelig å finne nye profiler, og startprofilenes kanaler kan bli kommentert på av nye profiler, og kommentarer fordi det kan være nye profiler som har skrevet den nye kommentaren.

Under følger et forslag til hyppighet for preferert sjekk regime. Dette er utarbeidet på bakgrunn av beskrivelsen over, størrelsen på databasen, og den største daglige kvoten. Forslaget er å sjekke:

- alle kanaler hver dag.
- alle kommentarer hver dag, og ved endring/svar sjekk:
    - etter ny endring av samme kommentar hvert minutt den neste timen.
    - etter svar på kommentaren hvert femte minutt det neste døgnet.
- alle spillelister hver dag.
- hvert spillelisteelement hver femte dag.

- en kanals abonnementer hver dag.
- etter at en av startprofilene laster opp en ny video, sjekk:
    - etter nye kommentarer på kanalen/videoen omtrent hvert femte minutt det neste døgnet.
    - etter nye kommentarer på kanalen/videoen omtrent hver time etter det første døgnet, i en uke før normal sjekk gjentas.
- ved nye kommentarer blir tidene over restartet, og sjekk den nye kommentaren etter:
    - endring hvert minutt den neste timen.
    - endring hver time det neste døgnet etter den første timen.
    - svar hvert minutt den neste timen.
    - svar hvert kvarter det neste døgnet etter den første timen.
- hver video omtrent hver tredje dag.

Avrundet estimerte kostnader finnes i tabell C.5. Merk at kvotekostnad ved ytterligere sjekk på nye entiteter ikke er beregnet, men forventet å være mindre enn gjenværende daglig kvote.

Type	Antall (k)	kost (ψ)	intervall (dager)	total ($k\psi$)	daglig kost ($k\psi$)
Abonnement	2 877	7	1	20 141	20 141
Kanal	45	9	1	407	407
Kommentar	101	2	1	201	203
Spilleliste	62	7	1	433	433
SpillelisteItem	3 742	7	5	26 197	5 239
Video	2 013	9	3	18 118	6 039
Total	8 840			65 498	32 462

Tabell C.5 Forenklet oversikt over kostnad ved å sjekke med forskjellig hyppighet. Kun de daglige sjekkene er inkludert, ytterligere sjekk ved nye entiteter antas å kreve mindre kvote enn gjenstående daglig kvote.

C.3 Kvotekostnader for noen gitte antall kanaler og kvoter på YouTube

For å samle og oppdatere informasjon fra et sett med kanaler, må man vite hvor mye det er forventet at hver nye kanal har av innhold og hvor mye de vil vokse. Estimatet for å regne ut dette baseres på perioden for innsamlede data i PoC, som videre har dannet grunnlaget for å beregne forholdet mellom antall kanaler og all annen innsamlet data. Den daglige økningen av antall kanaler kombinert med disse forholdene gir en indikasjon på forventet økning i total mengde lagret data. Dette forholdet er tilgjengelig i tabell C.6. Merk at det da forventes at de nye kanalene også vil bli lagret og få informasjonen oppdatert etterhvert. Dette medfører en eksponentiell økning i både kostnader som trengs for å hente informasjonen etterhvert som nye kanaler blir lagret og oppdatert for informasjon.

Type	Forhold 1:x	Daglig Økning
Channel	1.0	10.2
Account	1.3	12.3
Comment	2.2	22.6
Playlist	1.4	13.9
PlaylistItem	82.7	844.5
Subscription	63.6	649.6
Video	44.5	454.3

Tabell C.6 Oversikt over observert forhold for datamengde fra en kanal. Daglig økning i datamengde foruten kanaler kommer da av at det øker med 10.2 kanaler, og resten øker da gitt forholdet multiplisert med 10.2.

Antall kanaler det i utgangspunktet regnes ut fra er 1 000, 10 000 og 50 000. Forventet tid det vil ta å oppdatere alle kombinasjoner vises i tabell C.7. Hvis det er ønskelig å gjøre daglige oppdateringer, men man ikke har tilgang på større kvoter enn grunnkvoten på 10 000 ψ, vil det være behov for omtrent like mange kontoer som det tar dager å oppdatere. For 10 000 kanaler vil det da si behov for rundt 145 kontoer i utgangspunktet. Det er ikke undersøkt om dette er noe YouTube ser på som akseptabelt, ei heller tatt stilling til internt.

Kvote\Kanaler	1,000	10,000	50,000
10 000ψ	5 mnd (144 dager)	4 år (1440 dager)	20 år (7210 dager)
1 000 000ψ	1.4 dager	2 uker	2 mnd
50 000 000ψ	2 min	17 min	1,5 timer

Tabell C.7 Oversikt over tiden det vil ta å sjekke alle tilhørende ressurser for gitt antall kanaler og kvote

Hvis det brukes begrenset sjekkhyppighet som beskrevet tidligere vil den totale kostnaden omtrent halveres i dette tilfellet. Det vil si at det er behov for omtrent halvparten så mange kontoer uten en utvidet kvote. Dette er gjengitt i tabell C.8.

Kvote\Kanaler	1,000	10,000	50,000
10 000ψ	2 mnd (71 dager)	2 år (712 dager)	10 år (3560 dager)
1 000 000ψ	17 timer	1 uke	1 mnd (35 dager)
50 000 000ψ	1 min	10 min	42 min

Tabell C.8 Oversikt over tiden det vil ta å sjekke alle tilhørende ressurser for gitt antall kanaler og kvote med begrenset preferert sjekk.

D Ressursbehov

De to største ressursbehovene for datainnsamling er utviklertid og maskinvare. Å gi et estimat for utviklertid er utenfor rammen av denne rapporten, det må gjøres når man vet hva som skal samles inn av data, fra hvilke kilder, og på hvilken måte. For datainnsamling som er begrenset i omfang med hensyn til tid og nivåer man går ned i vil utviklertid være den største kostnaden. Men med større datafangst vil lagringskostnadene fort bli det største problemet. For eksempel, på en dag lastes det opp ca 720 000 timer video som tar opptil 385 terabyte med diskplass [33, 34]. Lagringsplassen alene vil i dette tilfellet per i dag kostet ca. 135 000 kroner. Her vil vi her i første omgang dokumentere kostnadene for å laste ned data, ikke for langtidslagring av større datamengder.

D.1 Kort om maskinvareoppgraderingsbehov ved bruk

Implementasjon og test i rapporten ble gjennomført på en enkelt virtuell maskin, så det er mulig å starte opp slik. Etterhvert som mengden data øker er det derimot forventet behov for å oppgradere, eller bytte over til å bruke flere maskiner.

Hvis det ses bort fra kvotebegrensningen er det utredet grovt forventet maskinvarebehov for sanntidseksempelet demonstrert tidligere i rapporten. Det viktigste fortrinnet ved å bruke flere maskiner er bedre arbeidsfordeling av oppgavene, som innsamling og lagring på separate maskiner, samt mulighet for skalering etter arbeidsmengde. Det kan også være nødvendig med større mengder lagringsplass og minne for å håndtere dataen under innsamling. I tillegg er det viktig å sikre dataene mot ødelagt hardware. Alt dette er enklere å håndtere med flere maskiner - spesielt for store mengder data. Fortrinnet til enkelmaskin er at kostnadene primært er engangsutgift ved oppstart, og at det kan være billigere å få opp en fungerende implementasjon. Likevel er det som nevnt sannsynligvis nødvendig å ta i bruk flere maskiner når datamengden øker nok. En overgang fra enkelmaskin til cluster i ettertid kan føre til en ikke estimerbar kostnad som er arbeidstimer for utviklere som må endre implementasjonen fra enkelmaskin til cluster.

Estimeringene i tabell D.1 tar derfor utgangspunkt i at det enten er en enkelmaskin, eller et cluster av maskiner fra starten av, som skal anskaffes(/oppgraderes). Merk at det for enkelmaskin er en engangskostnad, mens det utredes månedskostnader for cluster av maskiner fra en kommersiell leverandør. Det er også viktig å bemerke at implementasjon har mye å si for effektiviteten til maskinvaren, både for enkelmaskin og cluster, og at det finnes andre og muligens bedre valg av maskiner/maskinvare som ikke er utredet her.

Sjekktype	Singel maskin	Cluster	
Sanntid	Minst 30,000 kr Ytterligere komponenenter kan bli nødvendig	Maskintype (VM info/id)	månedspris
		Database (Single General Purpose Gen5 6vCore 32GB)	10,000 kr
		Innsamlingsmaskin (1x F8s: 8 vCPUs 16 GB RAM)	3,000 kr
		Oppdatering (2x E8-2s v3: 2 vCPUs 64 GB RAM)	12,700 kr
		Totalt:	25,800 kr
Begrenset likestilt	Fra 5,000 til 30,000 Avhengig av ønsket/forventet datamengde og sjekkhyppighet	Maskintype (VM info/id)	månedspris
		Database (Single General Purpose Gen5 2vCore 32GB)	3,000 kr
		Innsamlingsmaskin (1x F4: 4 vCPUs 8 GB RAM)	1,500 kr
		Oppdatering (1x B8MS: 8 vCPUs 32 GB RAM)	2,500 kr
		Totalt:	7,300 kr

Tabell D.1 Et estimat på mulige priser for singel maskin, eller cluster av maskiner fra kommersielle leverandører [35, 36]. Merk singel maskin er engangskostnad, mens cluster er månedspriser hvor maskinene kjører 24/7.

Referanser

[1] S. Bradshaw, H. Bailey og P. Howard, «Industrialized Disinformation: 2020 Global Inventory of Organised Social Media Manipulation. Working Paper 2021.1. Oxford, UK: Project on Computational Propaganda.,» Oxford, 2021.

[2] Etterretningstjenesten, «Fokus 2021 - Etterretningstjenestens vurdering av aktuelle sikkerhetsutfordringer,» Oslo, 2021.

[3] P. Sikkerhetstjeneste, «Trusselvurdering 2021,» Oslo, 2021.

[4] M. Richtel, «W.H.O. Fights a Pandemic Besides Coronavirus: An 'Infodemic'.,» The New York Times, 06 Februar 2020. [Internett]. Available: https://www.nytimes.com/2020/02/06/health/coronavirus-misinformation-social-media.html. [Funnet 09 Desember 2021].

[5] EEAS, «EEAS SPECIAL REPORT: Disinformation on the coronavirus – short assessment of the information environment,» East StratCom Task Force, Brussel, 2020.

[6] A. C. H. N. T. Stein Malerud, «Situasjonsforståelse ved sammensatte trusler - et konseptgrunnlag,» Forsvarets Forskningsinstitutt, Kjeller, 2021.

[7] A. L. Bjørnstad, «Understanding influence in a defense context: A review of relevant research from the field of psychology,» FFI, Kjeller, Norge, 2019.

[8] A. Bergh, «Understanding Influence Operations in Social Media: A Cyber Kill Chain Approach,» FFI, Kjeller, Norway, 2020.

[9] V. Alme, «Falske nyheter som sjanger,» FFI, Kjeller, Norge, 2019.

[10] E. G. Sivertsen, L. K. P. Bjørgul, H. Lundberg, I. Endestad, T. Bornakke, J. B. Kristensen, N. M. Christensen and T. Albrechtsen, "Uønsket utenlandsk påvirkning? – kartlegging og analyse av stortingsvalget 2021," Kjeller.

[11] A. Bergh, «Påvirkningsoperasjoner i sosiale medier - oversikt og utfordringer,» FFI, Kjeller, Norge, 2020.

[12] A. Bergh, «Are you seeing what I am seeing? Ensuring data relevance for online information environment assessments,» i *COVID 19 Disinformation: A Multi-National, Whole of Society Perspective*, Springer, 2022.

[13] A. Bergh og S. K. Larsen, «Innsamling og analyse av data fra sosiale medier - Oppsummering,» Forsvarets Forskningsinstitutt, Kjeller, 2021.

[14] Stortinget, «Lovdata,» 27 04 2016. [Internett]. Available: https://lovdata.no/lov/2018-06-15-38/gdpr/a4.

[15] J. Constine, «Tech Crunch,» 29 01 2020. [Internett]. Available: https://techcrunch.com/2020/01/29/facebook-earnings-q4-2019/.

[16] G. Weiss, «tubefilter,» 05 02 2019. [Internett]. Available: https://www.tubefilter.com/2019/02/05/youtube-2-billion-monthly-users/.

[17] M. S. F. T. W. K. H. Runa Fjellanger, «Slik var dramaet da Norge stengte ned,» VG, 24 Juli 2020. [Internett]. Available: https://www.vg.no/nyheter/innenriks/i/lA6Bpe/slik-var-dramaet-da-norge-stengte-ned . [Funnet 09 Desember 2021].

[18] U. S. A. Force, «YouTube,» 08 09 2014. [Internett]. Available: https://www.youtube.com/watch?v=4rj9-fmxbUg.

[19] YouTube, «YouTube API Reference,» 14 07 2020. [Internett]. Available: https://developers.google.com/youtube/v3/docs/.

[20] YouTube, «YouTube Data API Overview,» 14 07 2020. [Internett]. Available: https://developers.google.com/youtube/v3/getting-started.

[21] Google, «YouTube,» 16 10 2020. [Internett]. Available: https://support.google.com/youtube/answer/4642409?hl=no&ref_topic=9267586.

[22] YouTube, «YouTube Channel Resource,» 14 07 2020. [Internett]. Available: https://developers.google.com/youtube/v3/docs/channels.

[23] YouTube, «YouTube Comment Resource,» 14 07 2020. [Internett]. Available: https://developers.google.com/youtube/v3/docs/comments.

[24] YouTube, «YouTube CommentThreads Resource,» 14 07 2020. [Internett]. Available: https://developers.google.com/youtube/v3/docs/commentThreads.

[25] YouTube, «YouTube Data API Comments:List,» 14 07 2020. [Internett]. Available: https://developers.google.com/youtube/v3/docs/comments/list.

[26] YouTube, «YouTube Playlist Resource,» 14 07 2020. [Internett]. Available: https://developers.google.com/youtube/v3/docs/playlists.

[27] YouTube, «YouTube PlaylistItems Resource,» 14 07 2020. [Internett]. Available: https://developers.google.com/youtube/v3/docs/playlistItems.

[28] YouTube, «YouTube Subscription Resource,» 14 07 2020. [Internett]. Available: https://developers.google.com/youtube/v3/docs/subscriptions.

[29] YouTube, «YouTube Video Resource,» 14 07 2020. [Internett]. Available: https://developers.google.com/youtube/v3/docs/videos.

[30] YouTube, «YouTube Data API Quota Usage,» 14 07 2020. [Internett]. Available: https://developers.google.com/youtube/v3/getting-started\#quota.

[31] M. K. Matthew A. Russel, «Amazon,» 04 Desember 2018. [Internett]. Available: https://www.amazon.com/Mining-Social-Web-Facebook-Instagram/dp/1491985046/ref=sr_1_fkmr0_1?keywords=social+media+data+scraping+apis&qid=1638998939&s=books&sr=1-1-fkmr0. [Funnet 09 Desember 2021].

[32] o. Sergey Ignatchenko, «IT Hare,» 12 09 2016. [Internett]. Available: http://ithare.com/infographics-operation-costs-in-cpu-clock-cycles/.

[33] M. Mohsin, «10 YOUTUBE STATS EVERY MARKETER SHOULD KNOW IN 2021,» 21 Januar 2021. [Internett]. Available: https://www.oberlo.com/blog/youtube-statistics. [Funnet 09 Desember 2021].

[34] D. Price, «How Much Data Does Streaming Video Use?,» 13 Desember 2019. [Internett]. Available: https://www.makeuseof.com/tag/how-much-data-does-streaming-video-use/. [Funnet 09 Desember 2021].

[35] Komplett, «Komplett.no,» 07 10 2020. [Internett]. Available: https://www.komplett.no/category/11095/pc-nettbrett/pc-stasjonaer?nlevel=10723%C2%A711095&cnet-A00040-queryfacet=%5B3500000000%20TO%203990000000%5D&sort=PriceAsc%3AASCENDING.

[36] Microsoft, «Microsoft Azure,» 14 07 2020. [Internett]. Available: https://azure.microsoft.com/en-us/pricing/details/sql-database/.

Om FFI

Forsvarets forskningsinstitutt ble etablert 11. april 1946. Instituttet er organisert som et forvaltningsorgan, med særskilte fullmakter underlagt Forsvarsdepartementet.

FFIs formål

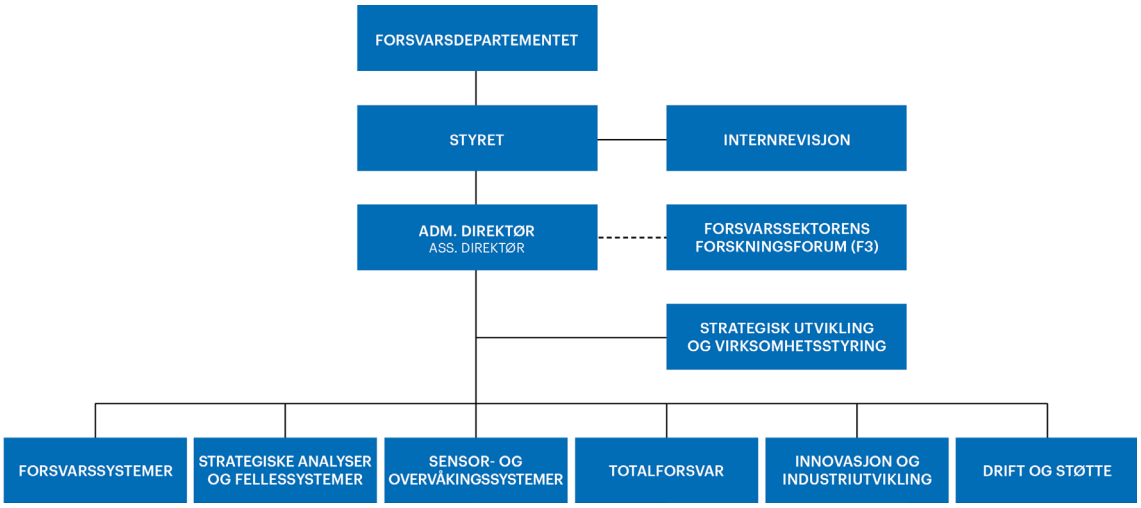
Forsvarets forskningsinstitutt er Forsvarets sentrale forskningsinstitusjon og har som formål å drive forskning og utvikling for Forsvarets behov. Videre er FFI rådgiver overfor Forsvarets strategiske ledelse. Spesielt skal instituttet følge opp trekk ved vitenskapelig og militærteknisk utvikling som kan påvirke forutsetningene for sikkerhetspolitikken eller forsvarsplanleggingen.

FFIs visjon

FFI gjør kunnskap og ideer til et effektivt forsvar.

FFIs verdier

Skapende, drivende, vidsynt og ansvarlig.



Forsvarets forskningsinstitutt
Postboks 25
2027 Kjeller

Besøksadresse:
Instituttveien 20
2007 Kjeller

Telefon: 63 80 70 00
Telefaks: 63 80 71 15
Epost: post@ffi.no

Norwegian Defence Research Establishment (FFI)
P.O. Box 25
NO-2027 Kjeller

Office address:
Instituttveien 20
N-2007 Kjeller

Telephone: +47 63 80 70 00
Telefax: +47 63 80 71 15
Email: post@ffi.no