ECUTBILDNING

En teknisk utvärdering av Matrix-protokollet som fundament för europeisk kommunikationssuveränitet: Arkitektoniska förutsättningar samt utmaningar vid praktisk implementering av en autonom infrastruktur.

Namn	**Victor Bohman Lind**
Utbildning	**IT-säkerhetsutvecklare**
Handledare/examinator	**Ludwig Simonsson Israelsson**
Datum	**2026-06-12**

Sammanfattning

I en alltmer geopolitiskt osäker omvärld utgör Europas omfattande beroende av utomeuropeiska, proprietära kommunikationsplattformar en kritisk sårbarhet. Molntjänster som ägs och administreras av externa aktörer medför inneboende risker rörande digital suveränitet, dataintegritet och leverantörsinlåsning. Samtidigt hamnar dessa plattformar allt oftare i direkt konflikt med europeisk lagstiftning, såsom Dataskyddsförordningen (GDPR) och de skärpta säkerhetskraven i NIS2-direktivet. För att bemöta dessa utmaningar utvärderar allt fler verksamheter öppna, decentraliserade standarder. Syftet med detta examensarbete är att, genom en praktisk implementation, utvärdera det alltmer populära Matrix-protokollets arkitektoniska och strategiska lämplighet som en robust grund för europeisk kommunikationssuveränitet.

Arbetets genomförande baserades på en kvalitativ och praktisk metodik där en autonom, containeriserad kommunikationsinfrastruktur byggdes upp från grunden. Genom att etablera servermiljön hos en nationell molnleverantör säkerställdes fullständig datalokalisering, varpå en organisatorisk testmiljö konfigurerades. Undersökningen omfattade systematiska tester av nätverksfederation mot externa servrar, administrativ identitetshantering, samt en utvärdering av de kryptografiska mekanismerna bakom protokollets enhetsbaserade totalsträckskryptering (E2EE).

Resultaten från utvärderingen visar att Matrix är ett arkitektoniskt lovande alternativ som effektivt kan eliminera beroende av proprietära leverantörer. Genom sin federerade struktur tillåter protokollet verksamheter att behålla kontroll över sin egen data och infrastruktur, samtidigt som man behåller sömlös kommunikation. Utvärderingen visar dock att denna extremt höga grad av suveränitet och säkerhet för med sig ett antal utmaningar. Den strikta krypteringsmodellen introducerar en märkbar friktion för slutanvändaren. För att balansera säkerhet med daglig produktivitet dras slutsatsen att organisationer inte bör tvinga fram kryptering överallt, utan istället tillämpa en strategiskt diversifierad och segmenterad rumsstruktur.

Ytterligare en central slutsats är att protokollet i sin initiala grundinstallation ofta prioriterar tillgänglighet framför säkerhet. För att uppnå den önskade suveräniteten krävs det att administratörer aktivt och metodiskt härdar plattformen, t.e.x genom att begränsa inkommande nätverkstrafik och förhindra läckage av metadata. Arbetet fastslår även att en framgångsrik och skalbar implementation i större organisationer är starkt beroende av att systemet integreras med existerande, centrala identitetshanteringssystem snarare än att förlita sig på manuell administration.

Slutligen konstateras att Matrix utgör ett överlägset och fullgott strategiskt val för resursstarka organisationer och myndigheter som söker reellt oberoende. För en bredare massadoption bland privatpersoner hämmas utvecklingen dock alltjämt av ett omoget marknadsekosystem. Innan branschen mognar och erbjuder fler tillförlitliga och lättillgängliga drifttjänster förblir den tekniska tröskeln för den genomsnittliga användaren hög.

ECUTBILDNING

Innehåll

1. Inledning

Detta inledande kapitel ger en introduktion till rapportens ämnesområde och sätter arbetet i en aktuell omvärldskontext. Kapitlet inleds med en bakgrundsbeskrivning och en precisering av arbetets syfte. Därefter bryts syftet ner i konkreta frågeställningar under problemformuleringen. Avslutningsvis redogörs för studiens avgränsningar samt den valda kvalitativa och praktiska metodiken för arbetets genomförande.

1.1 Bakgrund

I en alltmer geopolitiskt osäker omvärld har frågan om digital suveränitet blivit kritiskt aktuell i Europa. Det geopolitiska skiftet under de allra senaste åren har tydliggjort orsakerna till problemet med det omfattande europeiska beroendet av utomeuropeisk, proprietär digital infrastruktur. Idag hanteras en övervägande majoritet av den interpersonliga och organisatoriska kommunikationen inom Europa via stängda, kommersiella plattformar som ägs och kontrolleras av aktörer utanför unionens jurisdiktion. [URL1]
Denna centralisering innebär inte enbart en risk för leverantörsinlåsning, utan medför även känsliga sårbarheter rörande dataintegritet, spionage och en risk för att kritisk kommunikationsinfrastruktur kan användas som påtryckningsmedel vid internationella konflikter.

En möjlig väg för att motverka dessa risker och uppnå suveränitet i kommunikation är en övergång till öppna, decentraliserade standarder där kontrollen över datamängder och underliggande system vilar hos organisationen eller individen själv. Matrix-protokollet har vuxit fram som ett av de mest lovande tekniska alternativen för att utmana de etablerade, proprietära lösningarna. Protokollet förvaltas av en oberoende, icke-vinstdrivande stiftelse, *The Matrix.org Foundation*, vars struktur syftar till att standarden ska utvecklas utifrån medlemsstyrda beslut framför kommersiella vinstintressen. [URL2]
Matrix arkitektoniska uppbyggnad medger skapandet av helt autonoma och federerade kommunikationsmiljöer, och dess strikta kryptografiska fundament har lett till att det i ökande grad implementeras inom både statliga myndigheter, regeringar och militära organisationer runtom i Europa. [URL3]

I detta projekt ska en helt autonom kommunikationsmiljö baserad på Matrix-protokollet distribueras, konfigureras och utvärderas i praktiken. Syftet med att göra projektet är att systematiskt undersöka och dokumentera de arkitektoniska förutsättningarna, fördelarna och de praktiska utmaningarna som uppstår när man helt fasar ut proprietära system till förmån för en suverän, öppen standard.

Denna fördjupning har ett stort värde för yrkeskåren inom IT-säkerhet, då branschen står inför en omfattande strukturell omställning driven av skärpta lagkrav (såsom NIS2) och ökade krav på datalokalisering. [URL4]
Att praktiskt kunna utvärdera och härda decentraliserade arkitekturer är en kritisk spetskompetens. För den egna yrkesrollen som IT-säkerhetsutvecklare innebär projektet en direkt kompetenshöjning i att bygga robusta, oberoende system från grunden, vilket är direkt applicerbart och eftertraktat på den framtida arbetsmarknaden där förmågan att designa säkra, suveräna infrastrukturer blivit en nödvändighet.

1.2 Syfte

Syftet med arbetet är att genom en praktisk undersökning av Matrix-protokollets tekniska egenskaper, fördelar och problem utvärdera dess lämplighet och implementerbarhet som en öppen standard för att ersätta proprietära kommunikationssystem och därmed främja europeisk digital suveränitet.

1.3 Problemformulering

Det huvudsakliga problemet som detta arbete adresserar är bristen på praktiska, tekniska utvärderingar rörande övergången från centraliserade kommunikationsplattformar till helt autonoma och suveräna alternativ. Till skillnad från vissa andra krypterade kommunikationsverktyg, såsom Signal, vilka integrerar med befintliga adresseringssystem som telefonnummer, introducerar Matrix en helt ny, oberoende standard som arkitektoniskt liknar e-post. Denna decentraliserade struktur innebär att identiteter är direkt knutna till specifika, fristående servrar.

Detta paradigmskifte medför komplexa tekniska utmaningar, och det krävs djupgående kunskap inom protokollet för att avgöra hur infrastrukturen bäst ska utformas. Samtidigt saknas det ofta en balanserad kartläggning av hur dessa unika egenskaper påverkar den praktiska användbarheten och storskaliga spridningen. Utan en sådan systematisk granskning är det svårt för organisationer och myndigheter att bedöma hur de strategiskt bör förhålla sig till protokollet för att uppnå en optimal implementation.

Med utgångspunkt i arbetets syfte har undersökningen strukturerats och delats upp i följande fyra konkreta frågeställningar:

- **1:** Vilka specifika tekniska egenskaper och arkitektoniska fördelar gör Matrix-protokollet, i egenskap av en oberoende och decentraliserad standard, till ett lockande alternativ för digital suveränitet?
- **2:** Vilka negativa aspekter, begränsningar eller tekniska utmaningar existerar inom Matrix-ekosystemet, som kan försvåra eller hindra en massadoption på samhällsnivå?
- **3:** Hur presterar Matrix-protokollets centrala säkerhetsmekanismer i en verklig implementation?
- **4:** Hur bör europeiska organisationer och myndigheter strategiskt samt tekniskt förhålla sig till protokollet för att hantera dess komplexitet och uppnå den mest praktiska implementationen?

1.4 Avgränsningar och fokus

Följande områden har aktivt **valts bort** och kommer inte att studeras i arbetet:

- **Avancerad tilläggsfunktionalitet och integrationer:** Den praktiska serverimplementationen kommer att hållas på en grundläggande nivå för att säkerställa en ren utvärdering av kärnprotokollet. Detta innebär att kringliggande funktionalitet såsom bryggor (bridges) till externa kommunikationsprotokoll, automation, botar samt avancerade moderationsverktyg exkluderas. Däremot kommer end-to-end-kryptering

(E2EE) att aktiveras och tillämpas maximalt inom ramen för denna basimplementation.

- **Utvärdering av individuella Matrix-klienter:** Undersökningen avser inte att utvärdera, testa eller jämföra specifika användarapplikationer (klienter) i ekosystemet. Istället fokuserar studien på bakomliggande tekniska aspekter, protokollmekanismer och egenskaper som delas av ekosystemet i sin helhet.
- **Bred komparativ analys av decentraliserade standarder:** Studien avser inte att göra en djupgående teknisk jämförelse mellan Matrix och andra decentraliserade protokoll, såsom XMPP eller ActivityPub. Matrix har valts ut som det exklusiva studieobjektet baserat på dess nuvarande relevans inom den europeisk försvars- och myndighetssektor.
- **Djupgående prestandautvärdering och uttömmande konfigurationsanalys:** Studien avser inte att genomföra storskaliga stresstester, tunga prestandamätningar under extrem belastning eller en uttömmande kartläggning av samtliga existerande konfigurationsflaggor. Det tekniska utförandet begränsas till vad som krävs för att dra slutsatser på en strategisk och strukturell nivå.

Studiens **fokus** har kalibrerats och motiverats utifrån följande inriktningar:

- **Europeisk molninfrastruktur:** Den praktiska distributionen sker på en virtuell privat server (VPS) hos en leverantör under EU-jurisdiktion. Detta val är motiverat av att det direkt anammar studiens syfte rörande digital suveränitet.
- **Säkerhetshärdning och konfiguration på strategisk nivå:** Fokus läggs på de mest kritiska inställningarna för federation och end-to-end-kryptering (E2EE). Detta val är motiverat då dessa områden är direkt avgörande för organisationer som hanterar skyddsvärd information, och analyseras utifrån ett strategiskt implementeringsperspektiv.
- **Teknisk användarupplevelse (UX) och adresseringsmodell:** Undersökningen fokuserar på hur Matrix e-postliknande adresseringsmodell och strukturella uppbyggnad påverkar användarupplevelsen på ett tekniskt plan (exv. hantering av autentisering och identiteter). Detta fokus är motiverat då dessa bakomliggande tekniska faktorer har en direkt inverkan på möjligheterna till storskalig spridning och massadoption på samhällsnivå.

1.5 Metod/Arbetssätt

Undersökningen genomförs som ett praktiskt arbete baserat på en definierad metodik för systemimplementation och teknisk utvärdering.

Undersökningens avsikt är att:

- **Förstå:** Söka djupgående kunskap om de tekniska och säkerhetsmässiga förhållandena kring Matrix decentraliserade arkitektur.
- **Beskriva:** Systematiskt redogöra för det praktiska installationsförloppet och konfigurationsstegen för att utifrån det resulterande systemet kunna dra slutsatser om hur organisationer och myndigheter strategiskt bör förhålla sig till Matrix.

Undersökningen använder en kvalitativ metod där det är de tekniska egenskaperna, säkerhetsaspekterna och adoptionshindren som är avgörande för att besvara frågeställningarna, inte antalet svar eller statistiska mätdata (kvantitativ metod).

Det praktiska arbetets genomförande är uppdelat i fyra sekventiella arbetsmoment som beskriver processen för hur undersökningen exekveras:

- **1: Driftsättning av infrastruktur:** Processen inleds med att en virtuell privat server (VPS) hyrs och konfigureras hos en molnleverantör lokaliserad inom Europa. På denna server installeras bassystemet samt serverprogramvaran Synapse.
- **2: Systemkonfiguration och härdning:** I detta steg upprättas en stängd testmiljö i form av en digital community som struktureras för att simulera en verklig organisation med specifika användarkonton och kommunikationskanaler. Inom denna isolerade miljö aktiveras och tvingas end-to-end-kryptering (E2EE) för samtliga kanaler, vilket lägger grunden för att kunna analysera reella användarfall (use cases) utifrån både praktiska och säkerhetstekniska aspekter.
- **3: Dokumentation och valanalys:** Under hela installations- och konfigurationsförloppet förs en löpande logg över de tekniska stegen. Varje specifik inställning dokumenteras systematiskt och analyseras utifrån sina tekniska och administrativa för- och nackdelar.
- **4: Strategisk utvärdering:** Slutligen utvärderas den driftsatta miljön. Processen fokuserar på att analysera hur den tekniska hanteringen av exv. autentisering, kryptonycklar och den e-postliknande adresseringsmodellen påverkar administrationen, samt vilka strukturella utmaningar detta medför vid en storskalig adoption.

För att tydligt beskriva särskilt intressanta delar av det praktiska arbetet (såsom den federerade nätverkstopologin och centrala konfigurationsmoment) används **figurer och bilder**. I enlighet med givna instruktioner placeras dessa inte löpande i brödtexten, utan bifogas fristående under **bilagor** i rapportens sista del med tydliga hänvisningar i texten.

2. Teori

Detta kapitel etablerar den teoretiska och regulatoriska grund som utgör fundamentet för arbetets praktiska genomförande och efterföljande utvärdering. Syftet är att förse läsaren med de tekniska koncept och juridiska ramverk som krävs för att kunna tolka och analysera studiens resultat. Kapitlet inleds med en genomgång av begreppet digital suveränitet samt de skärpta kraven i NIS2-direktivet. Därefter redogörs för olika nätverksarkitekturer för kommunikation, följt av en djupdykning i Matrix-protokollets tekniska uppbyggnad. Avslutningsvis beskrivs de kryptografiska fundamenten för end-to-end-kryptering (E2EE) i distribuerade miljöer.

2.1 Digital suveränitet & regulatoriska ramverk

Begreppet digital suveränitet har under de senaste åren skiftat från att vara en teoretisk diskussion till att bli ett konkret tekniskt behov för europeiska organisationer. Europeiska kommissionen definierar EU:s digitala suveränitet som unionens och dess medlemsstaters förmåga att agera självständigt i den digitala domänen och utöva kontroll över sina egna

tekniska lösningar, utan ett ohållbart beroende av utomstående aktörer [URL5]. Inom ramen för modern informationssäkerhet delas digital suveränitet ofta upp i tre bärande pelare: datasuveränitet (kontroll över var data lagras och behandlas), infrastrukturell suveränitet (kontroll över fysisk hårdvara och nätverk) samt mjukvarusuveränitet (insyn i och kontroll över källkoden) [URL5].

Detta ökade fokus drivs primärt av ett förändrat säkerhetspolitiskt landskap. Ökade geopolitiska konflikter, t.e.x den eskalerande stormaktsrivaliteten och det politiska maktskiftet i USA, har blottlagt Europas infrastrukturella sårbarhet [URL6]. Skiftande amerikansk utrikespolitik har tydliggjort att ett ensidigt europeiskt beroende av utomeuropeisk digital infrastruktur utgör en kritisk risk för både säkerhet och diplomatisk handlingsfrihet [URL6].
När plattformar för kritisk kommunikation ägs och driftas av amerikanska teknikjättar uppstår ett geopolitiskt sårbarhetsfönster som direkt påverkar europeiska organisationers förmåga att garantera kontinuitet och oberoende.

Denna sårbarhet fördjupas ytterligare av en direkt juridisk konflikt mellan amerikansk och europeisk lagstiftning. Genom den amerikanska lagstiftningen *U.S. CLOUD Act* kan amerikanska myndigheter tvinga USA-baserade tjänsteleverantörer att lämna ut användardata, oavsett var i världen servrarna rent fysiskt är placerade [URL7].
Detta står i direkt strid med Dataskyddsförordningen (GDPR), specifikt artikel 48, som förbjuder överföring av personuppgifter till tredjeland baserat enbart på utländska domstolsbeslut utan formella internationella avtal [URL7].
För en europeisk organisation innebär detta att användningen av amerikanska molntjänster medför en inbyggd risk för att konfidentiella data röjs, oberoende av om datacentret är lokaliserat i Europa.

För att möta dessa sårbarheter och tvinga fram en höjd cybersäkerhetsnivå har EU implementerat Europaparlamentets och rådets direktiv (EU) 2022/2555, mer känt som NIS2-direktivet [URL4].
Direktivet ställer strikta rättsliga krav på organisationer inom samhällsviktiga sektorer och fokuserar särskilt på säkra leveranskedjor (*Supply Chain Security*) och riskhantering [URL4].
Om de primära kommunikationskanalerna som ska användas vid incidenthantering ligger hos en extern part med utomeuropeisk koppling, uppstår en oacceptabel sårbarhet enligt ramverkets krav på kontinuitet och incidentrapportering [URL4].

När dessa geopolitiska risker och regulatoriska krav ställs mot det rådande marknadsläget blir det tydligt att traditionella, centraliserade kommunikationsverktyg inte fullt ut kan garantera den autonomi som krävs. Detta har lett till att aktörer med extremt höga säkerhetskrav, exv. inom den europeiska försvars- och myndighetssektorn, i allt större utsträckning har börjat migrera mot öppna och suveräna arkitekturer [URL3].
För att uppnå faktisk kommunikationsmässig suveränitet krävs system som medger fullständig lokal kontroll över kryptonycklar, användardata och serverfederationer, egenskaper som motiverar en djupare teknisk utvärdering av decentraliserade protokoll som Matrix.

2.2 Nätverksarkitekturer för kommunikation

Inom nätverkskommunikation och systemdesign kategoriseras informationsutbytet traditionellt i tre övergripande arkitekturer: centraliserade, peer-to-peer (P2P) och federerade (decentraliserade) system (Kurose & Ross, 2021).

Centraliserade arkitekturer: I en centraliserad arkitektur ("Client-Server") dirigeras all nätverkstrafik, autentisering och datalagring genom en central server eller ett serverkluster som ägs och administreras av en enskild aktör (Kurose & Ross, 2021).
Slutanvändarna (klienterna) är helt beroende av denna centrala nod för att kunna kommunicera. Exempel på detta är kommersiella plattformar som Microsoft Teams, Slack och WhatsApp. Även om en centraliserad arkitektur bidrar till effektiv systemhantering och hög prestanda ur leverantörens perspektiv, skapar den en kritisk sårbarhet (*Single Point of Failure*) (Kurose & Ross, 2021). Ur ett suveränitetsperspektiv innebär en centraliserad modell att organisationen tvingas överlämna kontrollen över exv. nätverks, metadata och datalagring till en extern part – oavsett hur pålitlig denna part är.

Peer-to-Peer (P2P): I ett renodlat P2P-nätverk elimineras den centrala servern helt. Istället etablerar klienterna (noderna) direkta anslutningar till varandra och fungerar som både klient och server i nätverket (Kurose & Ross, 2021).
Exempel på detta inom meddelandehantering är Briar eller Jami. P2P maximerar teoretiskt sett suveräniteten och eliminerar leverantörsberoendet. Arkitekturen lider dock av fundamentala begränsningar vid asynkron kommunikation, eftersom båda parter måste vara uppkopplade samtidigt för att meddelanden ska kunna överföras direkt, såvida inte distribuerade hashtabeller (DHT) eller externa proxy-noder tillämpas (Kurose & Ross, 2021). På grund av detta är P2P generellt sett svårt att implementera och administrera i professionella och organisatoriska miljöer där central identitetshantering, krypterad revision och beständig historik är verksamhetskritiska krav.

Federerade och decentraliserade arkitekturer: En federerad arkitektur kombinerar den asynkrona pålitligheten från klient-server-modellen med oberoendet från P2P. I denna modell existerar ingen central, global auktoritet. Istället drivs det övergripande nätverket av oberoende servrar som kommunicerar med varandra via ett standardiserat protokoll [URL8].
Det mest välkända exemplet är e-post (SMTP), men inom modern realtidskommunikation återfinns protokoll som XMPP och Matrix [URL9].
Varje användare är knuten till en specifik lokal hemserver (*Home Server*), och när användare på olika hemservrar interagerar, synkroniseras nätverkstrafiken och rumstillstånden mellan dessa oberoende noder [URL9].

Denna typ av arkitektur är den huvudsakliga inriktningen som möjliggör skalbar digital suveränitet. Den tillåter en europeisk organisation att drifta sin egen hemserver på egen eller upphandlad nationell infrastruktur. Detta garanterar full kontroll över t.e.x krypteringsnycklar, användaridentiteter och accessloggar, samtidigt som förmågan att kommunicera sömlöst med externa parter och andra myndigheter över det federerade nätverket bibehålls [URL9].
Protokoll som är byggda på denna struktur förhindrar därmed i själva designen att en enskild aktör ensidigt kan stänga ner nätverket, blockera åtkomst eller förändra villkoren för informationsutbytet.

2.3 Matrix-protokollets tekniska uppbyggnad

Matrix är i grunden en öppen standard utformad för distribuerad, federerad realtidskommunikation. Till skillnad från traditionella meddelandesystem som primärt bygger på att dirigera meddelanden från en sändare till en mottagare, är Matrix arkitektoniskt uppbyggt kring konceptet att synkronisera distribuerade datastrukturer över ett nätverk av oberoende noder [URL9].
Denna design säkerställer att det inte finns någon central auktoritet eller felkälla (*Single Point of Failure*). Protokollets uppbyggnad kan kategoriseras utifrån tre huvudsakliga komponenter: klientskiktet, serverfederationen och mekanismen för tillståndssynkronisering.

Hemservrar och Klientskikt (Client-Server API): Den fundamentala noden i Matrix-nätverket kallas för en hemserver (*Home Server*). Varje användare eller organisation är knuten till en specifik hemserver som lagrar kontoinformation, krypteringsnycklar och användarens kommunikationshistorik [URL9].
Slutanvändaren interagerar inte direkt med det federerade nätverket, utan använder en helt fristående klient (exv. Element/FluffyChat/Cinny) som kommunicerar exklusivt med sin egen hemserver via ett standardiserat RESTful HTTP-API, ofta kallat *Client-Server API*. All data som utbyts via detta API formateras som JSON-objekt [URL9].
Detta skiktade tillvägagångssätt innebär att organisationer kan utveckla eller anpassa sina egna klienter för specifika säkerhetsbehov, utan att behöva modifiera underliggande nätverksprotokoll.

Federation (Server-Server API): När en användare på en hemserver vill kommunicera med en användare på en annan organisations hemserver, sker detta via Matrix *Server-Server API*. Identitets- och adresseringsmodellen liknar e-post, där en användare adresseras enligt formatet **@användarnamn:hemserver.se** [URL9].
Om flera användare från olika hemservrar befinner sig i samma digitala rum, etablerar deras respektive hemservrar en federation. Det innebär att servrarna automatiskt börjar dela och synkronisera alla händelser (*events*) som inträffar i rummet [URL9].

Tillståndssynkronisering och DAG (Directed Acyclic Graph): Den mest kritiska tekniska skillnaden mellan Matrix och andra protokoll är hur kommunikationsrum hanteras. I Matrix existerar inte "rum" på en enskild central server, likt många centraliserade motsvarigheter. Istället är federerade chattrum i en Matrix-klient en distribuerad datastruktur där alla deltagande hemservrar lagrar en exakt, oberoende kopia av rummets händelsehistorik [URL9].

Eftersom nätverket är decentraliserat och asynkront kan det uppstå situationer där händelser (exv. meddelanden eller statusuppdateringar) anländer i olika ordning till olika servrar, särskilt vid tillfälliga nätverksavbrott. För att hantera detta utan en central tidsserver använder Matrix en datastruktur känd som en riktad acyklisk graf, eller *Directed Acyclic Graph* (DAG) (Jacob et al., 2021).
Varje nytt meddelande innehåller en kryptografisk referens till det eller de omedelbart föregående meddelandena som hemservern kände till vid skapandet.

Genom att använda komplexa algoritmer för tillståndsupplösning (*State Resolution Algorithms*) kan samtliga deltagande hemservrar matematiskt räkna ut den korrekta, kronologiska ordningen för alla händelser i nätverket när de återfår kontakt (Jacob et al., 2021).

Denna arkitektur garanterar att en organisation alltid har en komplett, oförvanskad kopia av sin egen data, även om andra parter i nätverket stängs ner, vilket är en absolut förutsättning för teknisk suveränitet.

2.4 Kryptografiska fundament

För att garantera konfidentialitet och dataintegritet i en distribuerad nätverksmiljö tillämpar Matrix totalsträckskryptering, *End-to-End Encryption* (E2EE), som standard. Den kryptografiska arkitekturen skiljer sig fundamentalt från traditionella centraliserade system genom att vara utformad kring enhetsbaserad (*device-centric*) kommunikation, snarare än att vara knuten enbart till användarkontot (Ginesin & Nita-Rotaru, 2024). Varje enskild enhet som en användare loggar in från genererar och hanterar sina egna asymmetriska nyckelpar lokalt. För att etablera och underhålla de krypterade sessionerna mellan dessa oberoende enheter över det federerade nätverket förlitar sig Matrix på två samverkande kryptografiska protokoll: **Olm** och **Megolm** [URL9].

Olm-protokollet (P2P-kryptering)
För all direktkommunikation mellan två specifika enheter används Olm-protokollet. Olm är en specialanpassad implementation av den välkända kryptografiska metoden *Double Ratchet Algorithm*, vilken bygger på en kombination av asymmetriska Diffie-Hellman-nyckelutbyten och symmetriska hashratchets (Ginesin & Nita-Rotaru, 2024).
Denna design garanterar två kritiska säkerhetsegenskaper för P2P-kommunikationen: *Forward Secrecy* (PFS), vilket innebär att en kompromitterad krypteringsnyckel aldrig kan användas för att dekryptera tidigare avlyssnade meddelanden i historiken, samt *Post-Compromise Security* (PCS). PCS innebär att om en anfallare temporärt lyckas extrahera en nyckel, kommer angriparen att förlora åtkomsten igen så snart den legitima parten skickar ett nytt meddelande och ratcheten roterar framåt (Ginesin & Nita-Rotaru, 2024).
På grund av sin beräkningsintensitet används Olm i Matrix exklusivt för att upprätta en-till-en-kanaler mellan enheter, i syfte att säkert distribuera sessionsnycklar för de större grupprummen [URL9].

Megolm-protokollet (Gruppkryptering)
Ett välkänt kryptografiskt problem med renodlade Double Ratchet-implementationer är skalbarhet. I ett federerat Matrix-rum med tusentals användare, som i sin tur använder flera enheter vardera, skulle en avsändare behöva kryptera och skicka samma meddelande asymmetriskt tusentals gånger. Detta skulle överbelasta både nätverkets bandbredd och klientens processorkraft. För att lösa denna flaskhals tillämpas Megolm-protokollet vid all rumskommunikation [URL9].

Megolm använder istället en envägs symmetrisk ratchet-struktur. När en enhet (avsändaren) vill börja skriva i ett rum, genererar den en ny, unik Megolm-sessionsnyckel. Denna sessionsnyckel distribueras sedan asymmetriskt till alla andra verifierade enheter i rummet via de säkra Olm-kanalerna (Ginesin & Nita-Rotaru, 2024).
När distributionen är klar används Megolm-nyckeln för att snabbt kryptera de faktiska chattmeddelandena symmetriskt (exv. via AES-256) innan de skickas till hemservern för synkronisering [URL9].

För varje nytt meddelande som avsändaren skickar i rummet, roterar Megolm-ratcheten kryptografiskt framåt ett steg. Eftersom denna funktion är matematisk enkelriktad bibehålls egenskapen *Forward Secrecy* - en angripare med en aktuell nyckel kan inte räkna algoritmen baklänges för att läsa gamla meddelanden (Ginesin & Nita-Rotaru, 2024).
Däremot saknar Megolm inbyggd *Post-Compromise Security*, eftersom samma startnyckel delas av många mottagare. För att hantera denna risk och minska sårbarhetsfönstret är Matrix-klienter tvingade att periodiskt rotera och kassera Megolm-sessionen helt (oftast efter 100 meddelanden eller 7 dagar), varpå en helt ny sessionsnyckel skapas och distribueras via Olm [URL9].
Det är denna dualitet och växelverkan mellan protokollen som gör den storskaliga kommunikationssuveräniteten säker i praktiken.

3. Resultat

I detta kapitel redovisas de konkreta och objektiva resultaten från det praktiska arbetets genomförande. Kapitlet beskriver det faktiska utfallet av undersökningen utan subjektiva värderingar eller efterföljande analyser. Innehållet är strukturerat kronologiskt och omfattar driftsättningen av den underliggande server- och nätverksinfrastrukturen, installationen av Matrix-hemservern Synapse, den administrativa konfigurationen av användarkonton och rumsstrukturer samt utfallet av genomförda tekniska tester.

3.1 Driftsättning av infrastruktur

En virtuell privat server (VPS) anskaffades hos molnleverantören Bahnhof med en hårdvaruspecifikation omfattande 4 GB RAM, 50 GB lagring samt 2 CPU-kärnor *(se Figur 3.1 i Bilaga A)*. Som operativsystem för servern driftsattes Linuxdistributionen Ubuntu 24.04 LTS. Den initiala fjärranslutningen till servern upprättades via protokollet SSH från en lokal klient med en tillhandahållen privat kryptonyckel. Omedelbart efter anslutningen genomfördes en fullständig uppdatering av samtliga systempaket i operativsystemet *(se Figur 3.2 i Bilaga A)*.

För hantering och isolering av applikationsmiljön installerades containerplattformen Docker på servern. Installationsprocessen verkställdes genom att lägga till Dockers officiella GPG-nyckel och tillhörande konfigurationsförråd (repository) till operativsystemets pakethanterare, följt av driftsättning av paketen docker-ce, docker-ce-cli och docker-buildx-plugin *(se Figur 3.3 i Bilaga A)*.

I syfte att etablera en unik ingångspunkt för kommunikationstjänsten utan att påverka huvuddomänens ordinarie webbplats skapades en dedikerad underdomän med namnet matrix.generalis.se via domänleverantören STRATOS gränssnitt. För denna underdomän konfigurerades en DNS-post av typen A (IPv4) som pekade direkt mot serverns publika IP-adress *(se Figur 3.4 i Bilaga A)*. För att möjliggöra server-till-server-kommunikation (federation) etablerades en DNS-delegering genom att skapa en SRV-post (_matrix._tcp) i DNS-inställningarna för huvuddomänen generalis.se, med en pekare inställd mot matrix.generalis.se på port 443.

Webbservern Nginx installerades på värdsystemet för att fungera som en omvänd proxy (reverse proxy). Operativsystemets inbyggda brandvägg (UFW) konfigurerades restriktivt till

att exklusivt tillåta inkommande trafik för OpenSSH samt profilerna för HTTP och HTTPS via Nginx (Nginx Full). En virtuell värdkonfiguration (Virtual Host) upprättades i Nginx för att avlyssna inkommande trafik till matrix.generalis.se och vidarebefordra denna internt till port 8008 *(se Figur 3.5 i Bilaga A)*. För att tillgodose protokollets krav på krypterad kommunikation implementerades slutligen automatiserad certifikathantering via verktyget Certbot, varvid ett giltigt TLS-certifikat hämtades från Let's Encrypt och applicerades på webbservern för att tvinga all trafik över HTTPS.

3.2 Installation av Synapse

För orkestreringen av Matrix-servern upprättades en dedikerad katalogstruktur (~/matrix/) vari en konfigurationsfil för Docker Compose (docker-compose.yml) definierades. Konfigurationen strukturerades för att gemensamt driftsätta två samverkande tjänster: en PostgreSQL-databas för datalagring samt själva hemservermjukvaran Synapse *(se Figur 3.6 i Bilagan)*.

Initialt genererades serverns grundkonfiguration och kryptografiska nycklar genom exekvering av kommandot docker compose run --rm synapse generate. Vid det efterföljande försöket att starta systemet och provisionera en administratörsanvändare uppstod ett krasch-loop-fel, identifierat som ett behörighetsfel ([Errno 13] Permission denied) relaterat till läsningen av serverns kryptografiska signeringsnyckel (signing.key).

Felsökning via containerns interna loggverktyg (docker compose logs) påvisade en diskrepans gällande filsystemets äganderätt. Av säkerhetsskäl exekveras processerna i den officiella Synapse-containern under ett icke-privilegierat användar-ID (UID 991), medan den monterade datavolymen på värdsystemet var tilldelad standardanvändaren (UID 1000). Problemet åtgärdades genom en justering av katalogens och filernas äganderätt till UID 991 *(se Figur 3.7 i Bilagan)*, varpå containern startades om. Systemets driftstatus verifierades därefter med kommandot docker compose ps, vilket bekräftade att samtliga tjänster var framgångsrikt driftsatta och aktiva *(se Figur 3.8 i Bilagan)*.

3.3 Administration

Efter driftsättningen av infrastrukturen genomfördes administrativ provisionering av användarkonton. Utöver det initiala administratörskontot skapades ytterligare två lokala testkonton (medarbetare1 och medarbetare2) manuellt via serverns kommandotolk (CLI) genom exekvering av kommandot register_new_matrix_user inuti Synapse-containern.

För att simulera en organisatorisk miljö etablerades en rumsstruktur via Matrix-klienten Element. Funktionen "Spaces" tillämpades för att skapa en övergripande samlingskatalog ("TestCorp HQ") för den fiktiva organisationen. Inom denna katalog upprättades två separata kommunikationsrum med skilda säkerhetskonfigurationer *(se Figur 3.9 i Bilagan)*.

Det första rummet (#prod) konfigurerades utan end-to-end-kryptering (E2EE) för att hantera allmän kommunikation. Vid test av åtkomst gavs inbjudna användarkonton omedelbar läsbehörighet till all tidigare meddelandehistorik som existerade i rummet innan de anslöt.

Det andra rummet (#secret) konfigurerades med tvingande end-to-end-kryptering. Utfallet i detta rum påvisade att inbjudna användare tekniskt saknade åtkomst till meddelandehistorik som utbytts före deras anslutning, eftersom avkrypteringen hanteras lokalt hos användarna och servern inte betraktas som en betrodd part. Administrationen av detta rum, inklusive rättigheter att bjuda in användare och ändra rumsinställningar, visade sig hanteras via protokollets numeriska behörighetssystem benämnt "Power Levels".

3.4 Tekniska tester

För att utvärdera protokollets kryptografiska säkerhetsmekanismer genomfördes ett test av enhetsverifiering (Cross-signing). Ett befintligt användarkonto (*medarbetare1*) loggades in på en andra enhet, varpå Matrix-klienten omedelbart krävde att den initiala enheten godkände den nya sessionen för att bekräfta identiteten. Testet visade att om användaren avbröt eller ignorerade denna verifieringsprocess, genererade klienten onormala inloggningsvarningar till övriga deltagare i det E2EE-krypterade rummet för att indikera närvaron av oidentifierade enheter.

Vidare testades inställningarna i konfigurationsfilen homeserver.yaml gällande säkerhetshärdning. Flaggan enable_registration bekräftades vara avaktiverad för att förhindra oönskat publikt kontoskapande, och hanteringen av "presence"-data stängdes av, vilket resulterade i att servern slutade sända information om användarnas online-status till nätverket. Även funktionen för federationsbegränsning (federation_domain_whitelist) utvärderades, vilket demonstrerade serverns förmåga att avvisa inkommande federationstrafik från samtliga domäner utom de som uttryckligen var förgodkända i konfigurationen.

Slutligen genomfördes ett praktiskt test av server-till-server-kommunikationen (federationen) för att verifiera interoperabiliteten med andra nätverksnoder. Testet utfördes genom att initiera en kommunikationssession mellan det lokala administratörskontot (@generalis:generalis.se) och ett externt konto (@thegeneralis:matrix.org) på den publika hemservern matrix.org. Resultatet av testet bekräftade att både direktmeddelanden och inbjudningar till lokala E2EE-krypterade rum synkroniserades felfritt över servergränserna utan fördröjning, och att klienterna framgångsrikt kunde hantera utbytet av kryptografiska nycklar.

4. Diskussion

Detta kapitel analyserar och utvärderar resultaten från det praktiska arbetet i relation till studiens syfte och de inledande frågeställningarna. Arbetet har framgångsrikt resulterat i en fungerande, autonom Matrix-infrastruktur där både end-to-end-kryptering (E2EE) och serverfederation har implementerats och utvärderats i en praktisk miljö.

Frågeställning 1

Den första frågeställningen berörde vilka specifika egenskaper som gör Matrix till ett attraktivt alternativ för digital suveränitet. Resultatet av den egna driftsättningen bekräftar att protokollets decentraliserade arkitektur effektivt eliminerar beroendet av en central aktör. Möjligheten att placera all serverinfrastruktur hos en svensk leverantör innebär en omedelbar och konkret lösning på problematiken kring datalokalisering.

Den mest framträdande styrkan visade sig dock ligga i integrationsmöjligheterna. Ett proprietärt system kräver ofta att organisationen anpassar sig efter tjänstens begränsningar, medan en autonom Matrix-instans tillåter direkt kontroll över användardata. Utifrån insikterna under administrationstesterna framstod det som tydligt att Matrix administrativa potential maximeras när det kopplas till organisationens befintliga identitetshanteringssystem (exv. Active Directory via SSO), snarare än genom manuell kontohantering.

Frågeställning 2
Gällande den andra frågeställningen, som fokuserade på hinder för massadoption, identifierades flera signifikanta utmaningar. För privatpersoner bedöms tröskeln för närvarande vara orimligt hög. Matrix nuvarande ekosystem påminner visserligen strukturellt om e-post, men saknar dess etablerade mångfald av professionella tjänsteleverantörer. Följden blir att den oproportionerligt stora belastningen ofta hamnar på den publika servern matrix.org, vilket underminerar upplevelsen av nätverket som decentraliserat och tillförlitligt. Även ur ett strikt suveränitetsperspektiv identifierades vissa kompromisser under installationen. Infrastrukturen tvingades förlita sig på amerikanskt ägda teknologier, såsom Docker Inc. för containerisering och Let's Encrypt för TLS-certifikat. Även om själva datalagringen är suverän, existerar därmed fortfarande ett underliggande tekniskt beroende i systemets yttre lager.

Frågeställning 3
Den tredje frågeställningen syftade till att utvärdera protokollets kryptografiska fundament. Tillståndssynkroniseringen och E2EE-mekanismerna (Olm/Megolm) presterade exemplariskt över servergränserna, men tester påvisade en tydlig konflikt mellan maximal säkerhet och användarvänlighet (UX).

Matrix arkitektur innebär att en inbjuden användare i ett E2EE-krypterat rum tekniskt saknar möjlighet att avkryptera historik från tiden före inbjudan. Vidare visade testet av enhetsverifiering (Cross-signing) att systemet direkt varnar övriga deltagare om en användare underlåter att verifiera en ny session. Denna strikta säkerhetsdesign är imponerande ur ett tekniskt perspektiv, men den introducerar en friktion som riskerar att leda till "warning fatigue" om organisationens medarbetare saknar adekvat utbildning i varför varningarna uppstår. Dessutom omöjliggör E2EE ofta värdefulla produktivitetsintegrationer. Lösningen på detta är inte att överge krypteringen, utan att organisationer måste lära sig att pragmatiskt anpassa sina rumsstrukturer.

Frågeställning 4
Den fjärde frågeställningen behandlade hur organisationer strategiskt bör förhålla sig till protokollet. Ett kritiskt moment under arbetet var insikten om att Matrix standardkonfiguration prioriterar öppenhet framför säkerhet. För att uppnå den avsedda suveräniteten var det absolut nödvändigt att strama åt filen *homeserver.yaml*, exv. genom att avaktivera "presence"-data och begränsa federationen ("*allow-listing*"). Detta bekräftar vikten av specifik teknisk kompetens vid implementeringen.

Under genomförandet uppstod även ett behörighetsfel (en krasch-loop) orsakat av inkompatibilitet mellan standardanvändarens rättigheter (UID 1000) och containerns striktare säkerhetspolicy (UID 991). Detta fel tjänar som ett praktiskt exempel på den typ av driftutmaningar som kräver teknisk förståelse och som organisationer måste vara beredda att

hantera. Felet korrigerades systematiskt och påverkade inte det slutgiltiga utfallet, utan stärkte snarare förståelsen för systemets underliggande säkerhetsarkitektur.

Sammanfattningsvis bekräftar arbetet att det tekniska antagandet var korrekt: Matrix kan fungera som ett kraftfullt fundament för europeisk kommunikationssuveränitet. Framgång förutsätter dock inte bara korrekt teknisk konfiguration, utan även strategisk planering kring administration, användarutbildning och rumsstruktur.

5. Slutsatser

Utifrån den genomförda tekniska utvärderingen och den efterföljande analysen kan följande slutsatser dras rörande Matrix-protokollets lämplighet som fundament för europeisk kommunikationssuveränitet:

- **Matrix är arkitektoniskt bärkraftigt för digital suveränitet:** Den decentraliserade och federerade strukturen gör det fullt möjligt för europeiska organisationer att drifta autonoma kommunikationsmiljöer. Genom självhostning (exv. via en nationell VPS-leverantör) elimineras beroendet av utomeuropeiska aktörer för datalagring och nätverkskontroll, vilket direkt möter kraven på datalokalisering.
- **Storskalig adoption kräver integrationsförmåga:** För att Matrix ska fungera effektivt i en organisatorisk kontext räcker det inte med manuell administration. Protokollets sanna potential och hanterbarhet uppnås först när det integreras med organisationens existerande infrastruktur, såsom Active Directory och Single Sign-On (SSO) för centraliserad kontohantering.
- **Säkerhet kontra användbarhet (UX) är en kritisk balansgång:** Protokollets totalsträckskryptering (E2EE) erbjuder en exceptionell säkerhetsnivå, men medför oundviklig friktion för slutanvändaren, i synnerhet rörande tillgång till historik och enhetsverifiering. Denna friktion måste hanteras genom diversifierade rumsstrukturer (blandning av låsta, krypterade rum och öppna, okrypterade rum) samt genom omfattande intern användarutbildning.
- **Privat adoption hämmas av ekosystemets mognadsgrad:** Medan Matrix är ett starkt alternativ för organisationer med egna resurser, är tröskeln för massadoption bland privatpersoner för närvarande för hög. Beroendet av överbelastade publika servrar som matrix.org och bristen på tillförlitliga kommersiella leverantörer (likt e-postens ekosystem) förhindrar en bredare spridning på samhällsnivå.

Detta arbete bevisar, trots nedsidor, att Matrix-protokollet, med rätt teknisk kompetens och strategisk implementering, är ett **fullgott** alternativ för organisationer som prioriterar digital suveränitet och säkerhet framför proprietära standardlösningar.

5.1 Rekommendationer

Baserat på resultaten av detta examensarbete är min rekommendation tydlig: Matrix är ett lovande, kraftfullt och i många avseenden överlägset alternativ för den som vill uppnå bred digital kommunikationssuveränitet. Protokollet lider av vissa “barnsjukdomar” gällande ekosystemet, men de grundläggande fördelarna överväger nackdelarna.

Beroende på vilken målgrupp du tillhör, rekommenderar jag Matrix utifrån följande principer och insikter:

- **Till myndigheter och större organisationer:** Jag rekommenderar starkt en övergång till Matrix för verksamheter som vill röra sig bort från kommersiella, proprietära plattformar som Teams, WhatsApp eller Slack. Matrix ger er möjligheten att äga er egen infrastruktur fullt ut, vilket gör plattformen nästan oslagbar ur ett suveränitetsperspektiv. Mitt råd är dock att bygga smart: tvinga inte in hela verksamheten i end-to-end-krypterade (E2EE) rum. Använd E2EE där skyddsvärdet kräver det, men våga behålla öppna rum för den dagliga produktiviteten och snabbheten. Med rätt intern kompetens för integration och administration finns det egentligen inga fundamentala hinder kvar för att byta.
- **Till suveränitetssugna privatpersoner: Ha tålamod med ett omoget ekosystem.** För privatpersoner som vill ta tillbaka kontrollen över sin data från Big Tech rekommenderar jag starkt Matrix, men du måste gå in med rätt förväntningar. Målet är eventuellt att du ska kunna välja en Matrix-leverantör lika enkelt som du idag väljer en säker e-posttjänst (som Proton eller Tuta). Problemet är att detta professionella tjänsteekosystem ännu inte har etablerats på bred front. Idag är miljön ofta kaotisk, och valet står mellan informella servrar drivna av privatpersoner och den överbelastade offentliga "gateway-servern" matrix.org. Mitt råd är att använda matrix.org för att lära känna protokollet, men att ha tålamod och vara beredd att flytta ditt konto till en mer permanent och professionell leverantör så snart marknaden för dessa tjänster mognar. Om du å andra sidan är tillräckligt teknikintresserad är det faktiskt lättare än man tror att installera en egen server, eller att anlita en av de existerande aktörerna för Managed Hosting där du i beställningsrutan enkelt kan klicka hem en förkonfigurerad instans.
- **Till IT-branschen: Investera i tjänsteekosystemet.** Den största flaskhalsen för massadoption på samhällsnivå är idag att det krävs för mycket teknisk kompetens för att snabbt sätta upp en egen server. Min rekommendation till branschen är därför att skala upp utbudet av "Managed Hosting". När mindre företag, föreningar och privatpersoner smidigt kan klicka hem en förkonfigurerad, säker Matrix-instans utan att behöva hantera Linux och Docker själva, är standarden redo att utmana kommersiella jättar på allvar.
- **Till alla användare: Omfamna säkerhetens friktion.** Jag rekommenderar Matrix till stor del baserat på dess oöverträffade, kryptografiska design, men det är viktigt att förstå att äkta säkerhet alltid har en prislapp i form av friktion. Matrix kommer att be dig verifiera dina enheter och varna dig när okända parter dyker upp i hemliga rum. Min uppmaning är att inte se detta som ett UX-problem. Genomför utbildningar i varför systemet fungerar som det gör, så att medarbetare och användare lär sig uppskatta säkerheten istället för att ignorera legitima funktioner av ren utmattning. Matrix skyddar dig, men förutsätter att du förstår hur systemet är tänkt att användas.

6. Referenslista

Litteraturförteckning

Ginesin, J., & Nita-Rotaru, C. (2024). *A Formal, Symbolic Analysis of the Matrix Cryptographic Protocol Suite*. Northeastern University / arXiv.

Jacob, F., Beer, C., & Hartenstein, H. (2021). *Matrix state resolution: Exploring the decentralized synchronization of distributed data*. International Symposium on Dependable, Autonomic and Secure Computing (DASC). IEEE.

Kurose, J. F., & Ross, K. W. (2021). *Computer Networking: A Top-Down Approach* (8:e uppl.). Pearson.

Webbkällor

URL1: European Parliamentary Research Service (EPRS). (2020). Digital sovereignty for Europe. https://www.europarl.europa.eu/RegData/etudes/BRIE/2020/651992/EPRS_BRI(2020)651992_EN.pdf [Hämtad 2026-06-06].

URL2: The Matrix.org Foundation. (u.å.). *The Matrix.org Foundation*. https://matrix.org/foundation/ [Hämtad 2026-06-06].

URL3: Computer Weekly. (2025). European governments opt for open source alternatives to Big Tech encrypted communications. https://www.computerweekly.com/news/366633894/European-governments-opt-for-open-source-alternatives-to-Big-Tech-encrypted-communications [Hämtad 2026-06-11].

URL4: Europeiska unionens publikationsbyrå. (2022). *Europaparlamentets och rådets direktiv (EU) 2022/2555 (NIS2-direktivet)*. https://eur-lex.europa.eu/eli/dir/2022/2555 [Hämtad 2026-06-09].

URL5: Europeiska kommissionen (JRC). (2024). Open but Not Powerless: Towards a Common Understanding of EU Digital Sovereignty. https://publications.jrc.ec.europa.eu/repository/bitstream/JRC144908/JRC144908_01.pdf [Hämtad 2026-06-11].

URL6: European Council on Foreign Relations (ECFR). (2020). Europe's digital sovereignty: From rulemaker to superpower in the age of US-China rivalry. https://ecfr.eu/publication/europe_digital_sovereignty_rulemaker_superpower_age_us_china_rivalry/ [Hämtad 2026-06-11].

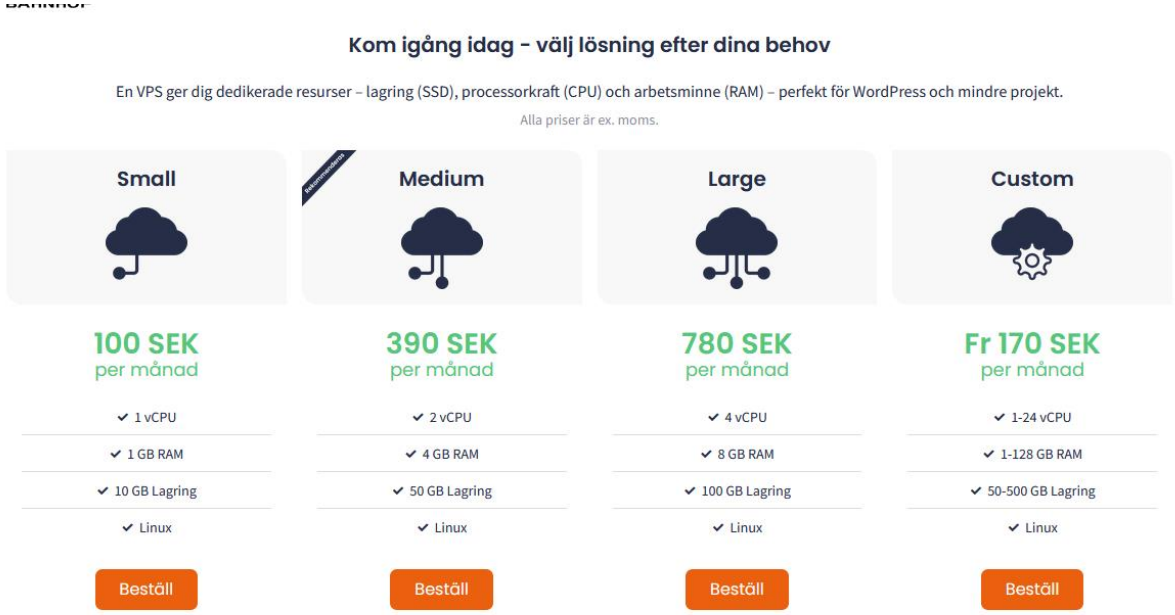
URL7: activeMind.legal. (2020). U.S. CLOUD Act vs. GDPR. https://www.activemind.legal/guides/us-cloud-act/ [Hämtad 2026-06-11].

URL8: Saint-Andre, P. (2011). Extensible Messaging and Presence Protocol (XMPP): Core. RFC 6120. Internet Engineering Task Force (IETF). https://datatracker.ietf.org/doc/html/rfc6120 [Hämtad 2026-06-11].

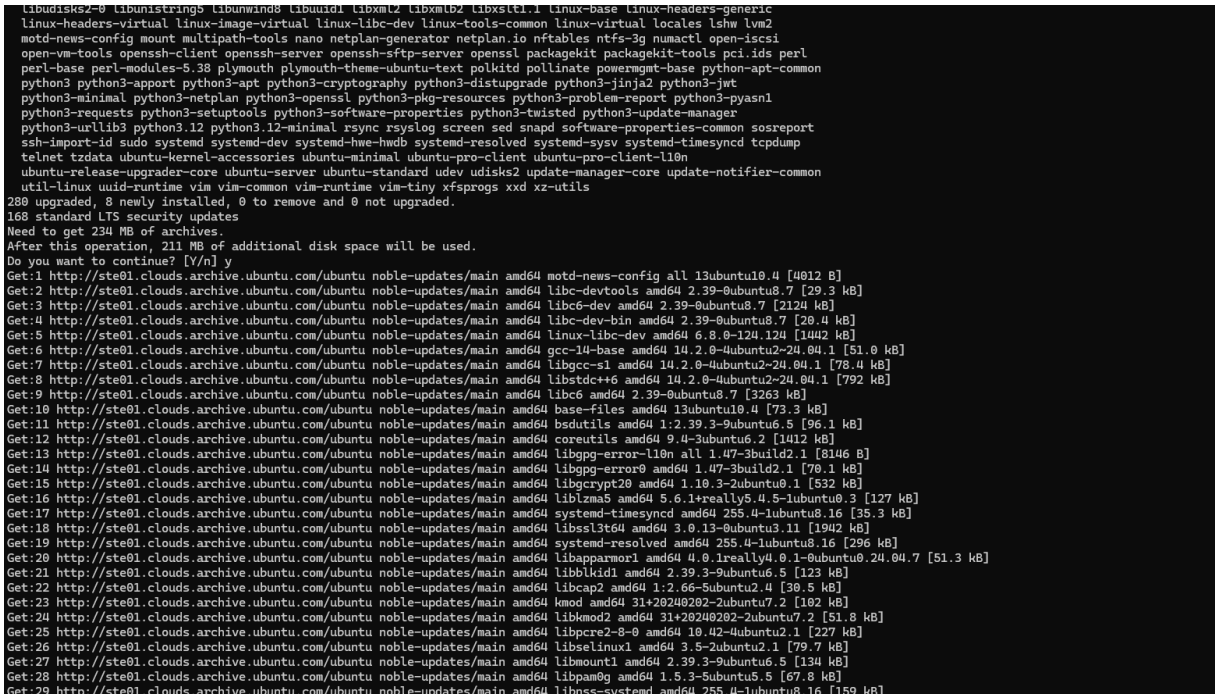
URL9: The Matrix.org Foundation. (2024). Matrix Specification. https://spec.matrix.org/ [Hämtad 2026-06-11].

Bilagor

Figur 3.1: Specifikation och resursallokering för den valda virtuella privata servern (VPS) hos leverantören Bahnhof.



Figur 3.2: Genomförande av initial paketuppdatering i operativsystemet Ubuntu 24.04 LTS via SSH.

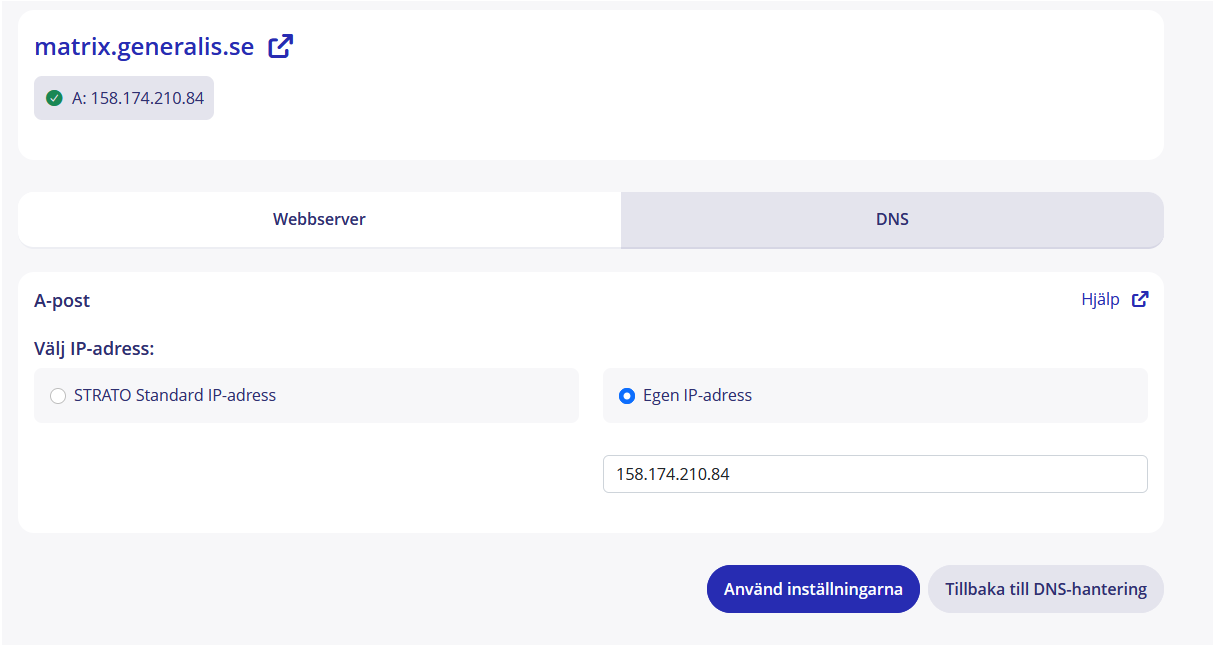


Figur 3.3: Kommandosekvens för installation av Docker-plattformen och tillhörande

säkerhetsnycklar.

```
ubuntu@cloud-vm-s4810-c2512:~$ sudo install -m 0755 -d /etc/apt/keyrings
ubuntu@cloud-vm-s4810-c2512:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg
ubuntu@cloud-vm-s4810-c2512:~$ sudo chmod a+r /etc/apt/keyrings/docket.gpg
chmod: cannot access '/etc/apt/keyrings/docket.gpg': No such file or directory
ubuntu@cloud-vm-s4810-c2512:~$ sudo chmod a+r /etc/apt/keyrings/docker.gpg
ubuntu@cloud-vm-s4810-c2512:~$ echo \
  "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu \
  $(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
ubuntu@cloud-vm-s4810-c2512:~$ sudo apt update
sudo apt install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin -y
Get:1 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:2 https://download.docker.com/linux/ubuntu noble InRelease [48.5 kB]
Hit:3 http://ste01.clouds.archive.ubuntu.com/ubuntu noble InRelease
Hit:4 http://ste01.clouds.archive.ubuntu.com/ubuntu noble-updates InRelease
Get:5 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [1197 kB]
Hit:6 http://ste01.clouds.archive.ubuntu.com/ubuntu noble-backports InRelease
Get:7 https://download.docker.com/linux/ubuntu noble/stable amd64 Packages [57.1 kB]
Get:8 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [232 kB]
Fetched 1661 kB in 1s (3253 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libfwupd2 libgusb2
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  docker-ce-rootless-extras pigz
Suggested packages:
  cgroupfs-mount | cgroup-lite docker-model-plugin
The following NEW packages will be installed:
  containerd.io docker-buildx-plugin docker-ce docker-ce-cli docker-ce-rootless-extras docker-compose-plugin pigz
```

Figur 3.4: Konfiguration av DNS A-post i STRATOS gränssnitt för underdomänen matrix.generalis.se.



Figur 3.5: Nginx virtuella värdkonfiguration (Virtual Host) för reverse proxy av inkommande

trafik till Docker-containern.

```
 GNU nano 7.2                                                                /e
server {
    listen 80;
    server_name matrix.generalis.se;

    location / {
        # Skickar trafiken vidare till Docker-containern
        proxy_pass http://127.0.0.1:8008;

        # Säkerställer att rätt IP-adresser och protokoll loggas
        proxy_set_header X-Forwarded-For $remote_addr;
        proxy_set_header X-Forwarded-Proto $scheme;
        proxy_set_header Host $host;

        # Tillåter rimligt stora filuppladdningar i chattarna (standard är för lågt)
        client_max_body_size 50M;
    }
}
```

Figur 3.6: Konfigurationsfilen `docker-compose.yml` för orkestrering av PostgreSQL och Synapse.

```
ubuntu@cloud-vm-s4810-c25
 GNU nano 7.2                                                  docker-compose.yml *
services:
  db:
    image: postgres:15-alpine
    restart: always
    environment:
      POSTGRES_USER: synapse
      POSTGRES_DB: synapse
      POSTGRES_PASSWORD: "password" # låtsas som att det är ett slumpgenererat lösenord här, håller det simpelt för projektets skull
    volumes:
      - ./data/postgres:/var/lib/postgresql/data

  synapse:
    image: matrixdotorg/synapse:latest
    restart: always
    environment:
      SYNAPSE_SERVER_NAME: "generalis.se"
      SYNAPSE_REPORT_STATS: "no"
    volumes:
      - ./data:/data
    ports:
      - 8008:8008
    depends_on:
      - db
```

Figur 3.7: Generering av serverkonfiguration samt korrigering av filrättigheter (UID 991) för

Synapse-containern.

```
db744e23eac5 Pull complete 0B
f3b83f2d9173 Extracting 1B
f3b83f2d9173 Extracting 1B
f3b83f2d9173 Extracting 1B
f3b83f2d9173 Extracting 1B
f3b83f2d9173 Extracting 1B
e4f4b0eb5b0d Pull complete 0B
3e49ec65d3ca Pull complete 0B
f3b83f2d9173 Pull complete 0B
b816cf103d53 Extracting 1B
b816cf103d53 Extracting 1B
b816cf103d53 Extracting 1B
b816cf103d53 Extracting 1B
b816cf103d53 Extracting 1B
b816cf103d53 Extracting 1B
5fdee0cd6b79 Pull complete 0B
b816cf103d53 Pull complete 0B
e6b8208284e4 Extracting 1B
e6b8208284e4 Extracting 1B
e6b8208284e4 Extracting 1B
e6b8208284e4 Extracting 1B
e6b8208284e4 Extracting 1B
e6b8208284e4 Extracting 1B
e6b8208284e4 Extracting 1B
e6b8208284e4 Extracting 1B
e6b8208284e4 Extracting 1B
e6b8208284e4 Extracting 1B
e6b8208284e4 Extracting 1B
e6b8208284e4 Extracting 2B
e6b8208284e4 Extracting 2B
e6b8208284e4 Extracting 2B
e6b8208284e4 Extracting 2B
e6b8208284e4 Extracting 2B
e6b8208284e4 Extracting 2B
e6b8208284e4 Extracting 2B
e6b8208284e4 Extracting 2B
e6b8208284e4 Extracting 2B
b50944496ba89 Pull complete 0B
e589de1b9f91 Pull complete 0B
e6b8208284e4 Pull complete 0B
Image matrixdotorg/synapse:latest Pulled
Container matrix-synapse-run-1ceb3843de01 Creating
Container matrix-synapse-run-1ceb3843de01 Created
Creating log config /data/generalis.se.log.config
Setting ownership on /data to 991:991
Generating config file /data/homeserver.yaml
Generating signing key file /data/generalis.se.signing.key
A config file has been generated in '/data/homeserver.yaml' for server name 'generalis.se'. Please review this file and customise it to your needs.
ubuntu@cloud-vm-s4810-c2512:~/matrix$
```

Figur 3.8: Verifiering av systemstatus via `docker compose ps`, bekräftande att tjänsterna är i drift.

```
ubuntu@cloud-vm-s4810-c2512:~/matrix$ sudo docker compose ps
NAME              IMAGE                          COMMAND                  SERVICE   CREATED          STATUS                         PORTS
matrix-db-1       postgres:15-alpine             "docker-entrypoint.s…"   db        12 minutes ago   Up 12 minutes                  5432/tcp
matrix-synapse-1  matrixdotorg/synapse:latest    "/start.py"              synapse   35 seconds ago   Up 4 seconds (health: starting)   8448/tcp, 0.0.0.0:8008->8008/tcp, [::]:8008->8008/tcp, 19090/tcp
ubuntu@cloud-vm-s4810-c2512:~/matrix$
```

Figur 3.9: Etablerad rumsstruktur i klienten Element, uppdelad i okrypterade och E2EE-krypterade kanaler.