

# Integrations- och API- strategi

Örebro kommun 2025-2028

**PROGRAM**

Uttrycker värdegrund och önskvärd utveckling av verksamheten.

**POLICY**

Uttrycker ett värdegrundsbaserat förhållningssätt och principer för vägledning.

**STRATEGI**

Konkretiserar ett program eller en policy och utgör en grund för prioritering.

**HANDLINGSPLAN**

Beskriver konkreta mål och åtgärder.

**RIKTLINJER**

Säkerställer ett riktigt agerande och en god kvalitet vid handläggning och utförande.

## Inledning och syfte

Örebro kommun äger och hanterar väldigt stora mängder data. Denna data är en enorm tillgång och kan tillföra ett stort värde om den hanteras på rätt sätt. För att kunna dra nytta av denna tillgång krävs att data kan utbytas mellan olika system och processer. De tekniska lösningarna som möjliggör sådant datautbyte benämns ofta som systemintegrationer. I ett modernt IT-landskap är systemintegration med hjälp av API:er (av engelskans Application Programming Interface), som ger möjlighet till datautbyte i realtid, en nyckelkomponent.

Integrations- och API-strategin syftar till att peka ut en riktning och ge vägledning kring hur Örebro kommun ska utforma sina systemintegrationer. Den tar sin utgångspunkt i Digitaliseringsstrategi för Örebro kommun 2020-2025 [1] och arkitekturprinciper [2] och bidrar till att skapa en sammanhållen, säker och hållbar digital infrastruktur där data och system effektivt kan samverka. Strategin ska möjliggöra en mer datadriven, automatiserad och tillgänglig kommunal verksamhet.

Genom att ha en tydlig strategi för integrationer och API:er som möjliggör effektiv kommunikation mellan system säkerställs att kommunen rör sig mot en modulär och skalbar arkitektur som undviker låsning till specifika leverantörer och tekniska silos, vilket är avgörande för att leva upp till de principer som styr kommunens arkitektur.

## Målgrupp

Beslutsfattare och ledning vid anskaffning, etablering, utveckling och förvaltning av digitala stöd och tjänster, samarbetspartners, IT-specialister med säkerhetskompetens samt IT-personal.

## Innehåll

<b>Inledning och syfte</b> .....	<b>3</b>
Målgrupp.....	3
Målbild.....	5
<i>Övergripande målbild</i> .....	5
<i>Effekt mål</i> .....	5
<i>Uppföljning och utvärdering</i> .....	5
Styrning och ledning .....	5
<i>Ägarskap</i> .....	5
<i>Informationskontrakt</i> .....	6
<i>Integrationskatalog</i> .....	6
Kategorier av integrationer .....	6
Val av teknik för systemintegrationer .....	7
<i>API som förstahandsval</i> .....	7
Dataformat .....	8
Återanvändning.....	8
Teknisk infrastruktur .....	8
<i>Integrationsplattform och ingående funktioner</i> .....	9
<i>API hantering- och integration</i> .....	9
<i>API-format</i> .....	9
<i>Underliggande teknisk IT-infrastruktur</i> .....	10
Kompetens .....	10
<i>Kompetensutveckling</i> .....	10
<i>Resurser och kompetenser</i> .....	10
<b>Referenser</b> .....	<b>11</b>
<b>Bilagor</b> .....	<b>11</b>
Bilaga - Teknisk infrastruktur.....	11

## Målbild

### Övergripande målbild

Örebro kommuns integrations- och API-strategi möjliggör en säker, standardbaserad och återanvändbar informationsdelning som främjar automatisering, innovation och ökad tillgänglighet för både interna och externa aktörer. Genom att prioritera API:er, öppna standarder och interoperabilitet skapas en flexibel och framtidssäker integrationsarkitektur som effektiviserar verksamhetsprocesser och möjliggör bättre service för medborgare, organisationer, medarbetare och samarbetspartners.

### Effektmål

#### Ökad automatisering

Genom integration mellan e-tjänster och verksamhetssystem möjliggörs automatisering vilket ger kortare handläggningstider och förbättrad service. Minst 60% av inkommande e-tjänsteärenden ska hanteras utan manuell handpåläggning senast 2028. Detta bygger vidare på målsättningar om plattform för avancerad automatisering i handlingsplan utifrån digital agenda.

#### Mer öppna data för innovation och nya tjänster

Minst 10 öppna API:er ska publiceras för att möjliggöra extern innovation och utveckling av nya tjänster senast 2028. Detta bygger vidare på målsättningar kring öppna data i handlingsplan utifrån digital agenda.

#### Ökad återanvändbarhet av integrationer

Minst 50% av nya integrationer ska vara återanvändbara och tillgängliga via en central API-plattform senast 2026 för att närma oss digitaliseringsstrategins delmål, ”Digitala tjänster och information ska kunna återanvändas”

#### Ökad säkerhet och styrning av API:er

För 100% av kommunens API:er ska det finnas en fastställd informationsklassning av informationen som hanteras samt en adekvat säkerhetslösning och hantering som motsvarar denna klassning senast 2028.

#### Uppföljning och utvärdering

Strategin och måluppfyllnad av effektmålen följs upp halvårsvis av enheten för Strategi, arkitektur och transformation på informationsförsörjnings- och digitaliseringsavdelningen. Status rapporteras till avdelningens ledningsgrupp.

## Styrning och ledning

### Ägarskap

Varje systemintegration ska ha utpekad en ägare som ansvarar för att informationskontrakt upprättas och upprätthålls. Som huvudregel bör ägarskapet för en integration sammanfalla med ägandeskapet för den information som utbyts. I Örebro kommuns Styr- och samverkansmodell för hantering av digitala stöd (pm3) innebär detta att det är objektet som stöttar den nämnd som äger data som är ägare för integrationen, inte eventuellt IKT-objekt som står för den tekniska lösningen. Ägaren (objektet) ansvarar för att berörda parter informeras vid förändringar i ett gränssnitt.

### Informationskontrakt

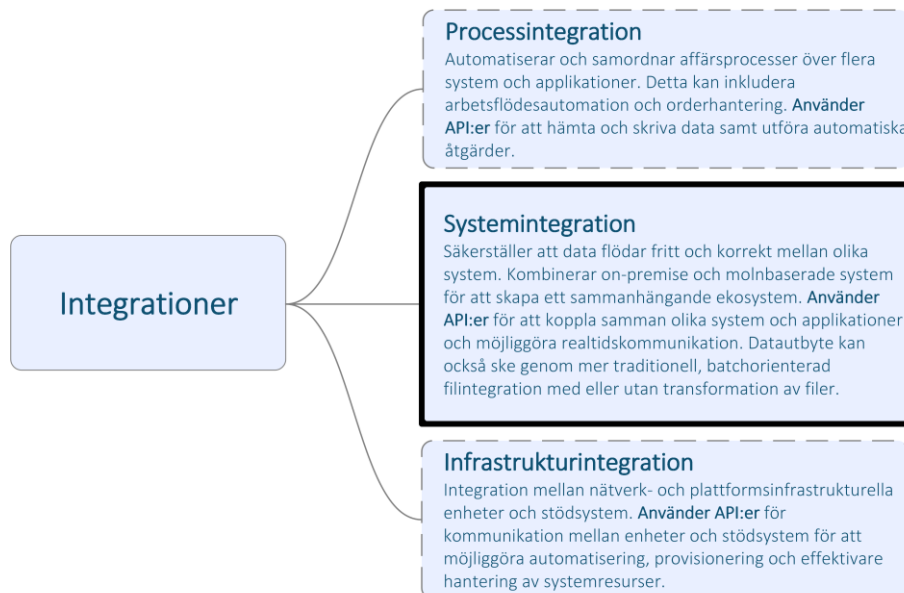
Informationskontrakt ska upprättas för alla systemintegrationer. Dessa ska specificera vilken information som utbyts och på vilket format (kan ske genom hänvisning till definitionsfiler) samt tillgänglighet. Behörigheter till ett API eller annan integrationspunkt ska inte tilldelas någon utan att ett informationskontrakt finns på plats.

### Integrationskatalog

För att ge integrationsägaren en övergripande bild för att kunna hålla koll på sina integrationer krävs någon form av digitalt stöd. Detta kan t.ex. ske genom en kommungemensam API-katalog där alla integrationer och aktuella datamängder synliggörs och/eller en behörighetskatalog där regelbundna kontroller sker.

## Kategorier av integrationer

Det finns olika kategorier av integrationer, som illustreras i Figur 1. Denna strategi behandlar i första hand systemintegration, det vill säga kommunikation mellan olika system och applikationer. Integrationer inom IoT-området omfattas inte heller av denna strategi.



Figur 1 Kategorisering av integrationer

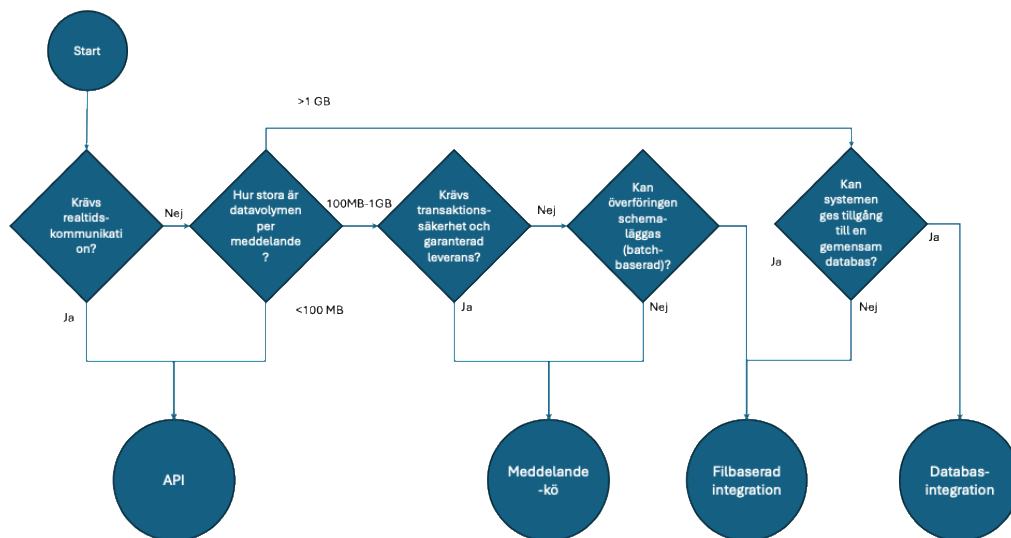
## Val av teknik för systemintegrationer

Vilken teknik som är lämplig för en viss systemintegration beror på många olika faktorer men förenklat kan det sägas avgöras av följande egenskaper:

- Krav på realtidskommunikation och synkron/asynkron kommunikation
- Datavolymer
- Behov av transaktionssäkerhet och garanterad leverans
- Möjlighet till schemaläggning / batchhantering

I Figur 2 finns ett enkelt beslutsträd som ger en fingervisning om vilken teknik som kan vara lämplig att använda i ett specifikt fall. Volymerna som anges är inte avsedda som exakta gränsvärden utan som grova tumregler för vilka val som kan vara lämpliga i olika fall. Specifikt kring API:er kan det förtydligas att även 100 MB sannolikt är för stort för ett enskilt anrop, men med hjälp av s.k. paginering kan ett meddelande delas upp i flera mindre paket av lämplig storlek.

Den uppmärksamme läsaren inser att det finns fall som inte passar in i beslutsträdet, till exempel om meddelandestorleken överstiger 1 GB men batchhantering inte är möjlig. I dessa fall behöver man hitta alternativa lösningar som att till exempel stycka upp meddelanden i mindre delar eller jobba med microbatchar som körs med hög frekvens.



Figur 2 Beslutsträd för vala av teknik för systemintegration

Beslutsträdet i Figur 2 förutsätter fritt val av integrationsteknik. I fall där integrationer ska byggas mot befintliga system kan vi dock oftast inte fritt välja integrationsteknik utan är hänvisade till de möjligheter som det aktuella systemet erbjuder. Vid anskaffning och utveckling av nya system ska kravställningen på integrationsmöjligheter göras utifrån denna strategi för att säkerställa att kommunen över tid rör sig i önskad riktning.

### API som förstahandsval

API:er är förstahandsvalet för integrationer inom Örebro kommun. Andra tekniker används då det finns särskilda skäl såsom framgår i Figur 2 med tillhörande text i föregående avsnitt.

Det finns många skäl till att välja API:er varav några viktiga listas nedan.

- Med API:er kan system kommunicera i realtid, vilket möjliggör snabbare svar, uppdateringar och bättre användarupplevelse vilket stöttar principen om ”Digitalt först” i Örebro kommuns digitaliseringsstrategi.
- API:er definierar klara gränser mellan system och klargör vilken information som är tillgänglig och hur den får användas. Det leder till bättre modularitet och lös koppling, vilket ligger i linje med Örebro kommuns arkitekturprincip 7, ”Bygg modulärt med lösa kopplingar”
- API:er kan enkelt återanvändas av flera system och klienter vilket ligger i linje med Örebro kommuns arkitekturprincip 6 ”Återanvänd från andra” och 8 ”Öppna upp och återanvänd information”
- API:er möjliggör god kontroll och säkerhet, i synnerhet om de hanteras via en API-gateway. Stöd finns för autentisering, åtkomstkontroll, loggning och versionering, vilket ger god kontroll över vilka som får använda datan och hur. Detta stöttar Örebro kommuns arkitekturprincip 10 ”Informations- och it-säkerhet ska vara en integrerad del av all utveckling”

## Dataformat

I enlighet med Örebro kommuns arkitekturprincip 5 ska integrationer använda öppna standarder i första hand. Om lämpliga öppna standarder saknas ska etablerade branschstandarder användas.

API:er bör utformas enligt etablerade principer och strukturer för att underlätta förståelse och återanvändning. Det är viktigt att undvika godtyckliga eller inkonsekventa lösningar och i stället sträva efter enhetlighet i design, dokumentation och hantering – exempelvis genom att följa vedertagna standarder som OpenAPI eller OData. Valet av standard bör dock utgå från användningsområdet och vilken domän informationen berör, exempelvis tillhandahåller OGC [3] standarder för utbyte av geodata.

## Återanvändning

Örebro kommuns digitaliseringsstrategi säger att ”Digitala tjänster och information ska kunna återanvändas”. Därför ska Örebro kommun röra sig bort från punkt-till-punkt-integrationer framtagna för att stödja ett specifikt användningsfall och i stället skapa återanvändbara gränssnitt via en central integrationsplattform. På så sätt ökar värdet som kommunen får ut av sin information och de investeringar som läggs på att bygga integrationer.

När ett behov av att hämta en viss informationsmängd uppstår ska befintliga gränssnitt användas i första hand. Att bygga ett nytt gränssnitt trots att behovet skulle kunna uppfyllas av ett befintligt ska betraktas som ett undantag och detta ska dokumenteras tillsammans med en tydlig motivering.

## Teknisk infrastruktur

En integrationsplattform består av flera tekniska komponenter som kan existera enskilt eller tillsammans beroende på val av leverantör och lösning. Komponenterna skapar tillsammans en funktionell infrastruktur för att hantera integrationer mellan olika system och applikationer.

Örebro kommun ska i första hand anskaffa komponenterna i sin integrationsplattform genom att anskaffa välbeprövade produkter som har god förvaltningsbarhet. För att underlätta långsiktig förvaltning bör plattformen i sig och integrationerna som den

hanterar kunna sättas upp, konfigureras och underhållas utan djup kunskap i specifika programmeringsspråk. Mer avancerade integrationer och/eller anpassningar av plattformen kan dock i vissa fall komma att kräva programmeringskompetens.

Nedan en summering och sammanfattning av vanliga funktioner som ingår i en integrationsplattform. Mer orienterande informations finns i bilagan ”Teknisk infrastruktur”.

### Integrationsplattform och ingående funktioner

- **API Gateway:** Hanterar dataflöden mellan olika system och applikationer.
- **Meddelandehanteringssystem** som hanterar asynkron kommunikation mellan olika system och applikationer.
- **Databas(er)** för att lagra och hantera data som flödar mellan olika system och applikationer.
- **Säkerhetsfunktioner** för att skydda data och system samt skydda mot obehörig åtkomst och attacker.
- **Övervakning och loggning** för att säkerställa att systemet fungerar korrekt samt att snabbt kunna identifiera och åtgärda problem.

### API hantering- och integration

API-hantering är kritisk funktion i en integrationsplattform och möjliggör direkt kommunikation mellan olika system och applikationer. Att använda API:er är en av de mest flexibla och kraftfulla metoderna för integration mellan system och applikationer.

Införandet av en API-gateway kan hjälpa till att abstrahera befintliga API:er som finns i verksamhetssystemen. Detta bidrar till att minska beroendet till externa leverantörer eftersom de interna verksamhetssystemen endast pratar med API-gatewayen. Integrationerna blir även mer flexibla och modulära eftersom förändringar i en leverantörs API:er hanteras direkt i API-gatewayen utan att övriga verksamhetssystem behöver anpassas.

För att säkerställa god säkerhet över tid bör en automatiserad och kontinuerlig sårbarhetsskanning ske av API-gateway, detta underlättar analys, identifiering av sårbarheter samt säkerhetshöjande aktiviteter. Individuella API:er ska också säkerhetsgranskas regelbundet enligt Riktlinjer för informationssäkerhet [4].

### API-format

Enligt Örebro kommuns arkitekturprinciper ska öppna standarder användas i första hand. Detta för att sänka kostnader och bidra till ökad återanvändning och en öppen marknad.

- **REST** är förstahandsvalet för standardiserade och öppna API:er inom Örebro kommun i enlighet med rekommendationer på [dataportal.se](http://dataportal.se) [5] och [digg.se](http://digg.se) [6].
- **GraphQL** kan vara ett alternativ för användning internt inom organisationen där flexibilitet och möjlighet till anpassade anrop väger tyngre än prestanda och enkelhet.
- **SOAP** ska inte användas vid utveckling av nya API:er inom Örebro kommun. Däremot behöver kompetens finnas för att konsumera API:er som använder SOAP eftersom det är ett vanligt format i befintliga API:er i framför allt lite äldre system.

### **Underliggande teknisk IT-infrastruktur**

- Nätverksutrustning som routrar, switchar för kommunikationen både internt och externt.
- Brandväggar för att kontrollera och filtrera nätverkstrafik baserat på säkerhetsregler, vilket skyddar integrationsplattformen från obehörig åtkomst och attacker.
- Servrar för att hantera och köra integrationstjänster.

## **Kompetens**

### **Kompetensutveckling**

För att kunna förvalta och vidareutveckla en modern integrationsplattform behöver Örebro kommun:

- Utbilda relevant personal i API-utveckling och användning för att säkerställa att de har den kompetens som krävs.
- Skapa en kultur av kontinuerligt lärande och förbättring.

### **Resurser och kompetenser**

Följande roller/kompetenser behövs för att förvalta en integrationsplattform:

- IT-specialister med kompetens inom nätverk, serverdrift, lastbalansering och loggning. Dessa kompetenser krävs för att förvalta och drifva komponenten API Gateway.
- Integrationsutvecklare med kompetens inom REST, GraphQL och SOAP, Access Control och grundläggande programmeringsspråk. Dessa kompetenser krävs för att hantera API Manager och tillhörande API:er.
- IT-specialister med kompetens inom serverdrift, applikationsdrift och övergripande koll på integrationer. Denna kompetens krävs för att drifva utvecklarportalen (API-katalogen).
- Utvecklarportalen (API-katalogen) kommer även ställa krav på utvecklingskompetens, men denna ses mer som en extern kompetens, alltså den som vill bygga en anslutning till ett API. Kan både vara interna och externa aktörer.
- Samordnare inom informationssäkerhet för att säkerställa att API:er följer gällande lagar och regler, såsom GDPR.
- Samordnare inom informationshantering för att säkerställa att vi har koll på den information som hanteras och att lösningen uppfyller de krav som ställs utifrån de klassning och riskanalyser som gjorts.
- IT-Specialist med IT-säkerhetskunskap som kan genomföra regelbundna säkerhetskontroller, säkerhetsgranskningar och penetrationstester.

## Referenser

- [1] Örebro kommun, ”Digitaliseringsstrategi”, Ks 994/2020.  
<https://intranat.orebro.se/download/18.56fc80391773f590b4632f3f/1613385033432/Digitaliseringsstrategi%20f%C3%B6r%20%C3%96rebro%20kommun%202020-2025.pdf>
- [2] Örebro kommun, ”Arkitekturprinciper”, Ks 749/2022.  
<https://intranat.orebro.se/download/18.1fea80c91872c95e966137a/1680268252564/Arkitekturprinciper.pdf>
- [3] Open Geospatial Consortium, <https://ogcapi.ogc.org/>.
- [4] Örebro kommun, ”Informationssäkerhet, riktlinje”, Ks 267/2019.  
<https://intranat.orebro.se/download/18.4184571315b947ed07a174e/1572518679410/Informationss%C3%A4kerhet - riktlinje.pdf>.
- [5] Digg, ”Sveriges dataportal,”.  
<https://www.dataportal.se/api-playbook/en-introduktion-till-rest>.
- [6] Digg, ”Ena - Sveriges digitala infrastruktur”.  
<https://www.digg.se/styrning-och-samordning/ena---sveriges-digitala-infrastruktur/byggblock/api-hantering>.

## Bilagor

### Bilaga - Teknisk infrastruktur

Ger en mer detaljerad teknisk beskrivning av de delar som beskrivs i kapitlet Teknisk infrastruktur.

# Bilaga - Teknisk Infrastruktur

## Integrationsplattform

Kopplar samman olika system och applikationer, hanterar dataflöden och säkerställer att information delas korrekt. En integrationsplattform fungerar som en central hub vilken samordnar och hanterar dataflöden mellan olika system samt gör det möjligt att minimera informationssilos och förbättra processer.

### Integrationsplattform huvudsakliga funktioner

- **API Gateway:** Hanterar dataflöden mellan olika system och applikationer. En API-gateway hanterar API-anrop, autentisering, auktorisering, trafikstyrning och övervakning.
- **Meddelandehanteringssystem** som hanterar asynkron kommunikation mellan olika system och applikationer ex. är Message- och Servicebus.
- **Automatisering av arbetsflöden:** Integrationsplattformen kan automatisera repetitiva och manuella uppgifter, vilket minskar risken för mänskliga fel och ökar produktiviteten
- **Databas(er)** har en central roll i en integrationsplattform genom att lagra och hantera data som flödar mellan olika system och applikationer.
- **Säkerhetsfunktioner** är avgörande för att skydda data och system i en integrationsplattform, dessa säkerställer att data och system är skyddade mot obehörig åtkomst och attacker. För att upprätthålla API:ers integritet måste lösningen innehålla en robust säkerhet, inklusive autentisering, auktorisering och kryptering för att skydda känslig information.
- **Övervakning och loggning** är kritiska komponenter i en integrationsplattform för att säkerställa att systemet fungerar korrekt och för att snabbt kunna identifiera och åtgärda problem.

## API-format

### REST (Representational State Transfer)

- **Arkitektur:** REST är en arkitekturstil som använder HTTP-protokollet för att kommunicera. Det är baserat på resurser och använder standard HTTP-metoder som GET, POST, PUT och DELETE.
- **Format:** Vanligtvis JSON eller XML.
- **Fördelar:** Enkelhet, skalbarhet, och lätt att använda. REST är statslöst, vilket innebär att varje anrop är oberoende.
- **Användningsområden:** Passar bäst för webbtjänster och applikationer där enkelhet och skalbarhet är viktiga.

### GraphQL

- **Arkitektur:** GraphQL är ett frågespråk för API:er som låter klienter begära exakt den data de behöver.
- **Format:** Vanligtvis JSON.

- **Fördelar:** Flexibilitet, effektivitet, och minskar överföring av onödiga data. Klienten kan specificera exakt vilka fält som behövs.
- **Användningsområden:** Passar bäst för applikationer där dataflexibilitet och effektivitet är viktiga.

### SOAP (Simple Object Access Protocol)

- **Arkitektur:** SOAP är ett protokoll som använder XML för att definiera meddelandestrukturer och regler för att utbyta information. Det är mer komplex än REST och erbjuder avancerade funktioner som säkerhet och transaktioner.
- **Format:** Endast XML.
- **Fördelar:** Standardiserat, plattformsoberoende, och stöder WS-Security för säkerhet.
- **Användningsområden:** Passar bäst för företagsmiljöer där säkerhet och tillförlitlighet är kritiska.

## API-gateway

Med en API-gateway hanteras dataflöden mellan olika system och applikationer. En API-gateway hanterar API-anrop, autentisering, auktorisering, trafikstyrning och övervakning.

### Huvudsakliga funktioner

- **Request Routing:** API-gatewayen tar emot API-anrop från klienter och vidarebefordrar dem till rätt backend-tjänst. Detta gör det möjligt att ha en enda ingångspunkt för alla API-anrop.
- **Autentisering och Auktorisering:** API-gatewayen verifierar API-nycklar, JWT-tokens och andra autentiseringsuppgifter för att säkerställa att endast auktoriserade användare kan komma åt tjänsterna. Rekommendation: Använd **OAuth**: En standard för auktorisering som tillåter applikationer att få begränsad åtkomst till användares resurser utan att avslöja deras lösenord
- **Rate Limiting:** Begränsar antalet API-anrop som en klient kan göra inom en viss tidsperiod för att förhindra överbelastning och missbruk
- **Request och Response Transformation:** API-gatewayen kan transformera inkommande begäranden och utgående svar för att matcha olika format och krav. Detta kan inkludera att ändra dataformat från XML till JSON eller lägga till/ta bort fält i svaren.
- **Caching:** API-gatewayen kan cachelagra svar för att förbättra prestanda och minska belastningen på backend-tjänster
- **Övervakning och Loggning:** API-gatewayen samlar in och loggar data om API-anrop för att övervaka prestanda, identifiera problem och analysera användningsmönster

## Meddelandehanteringssystem

En meddelandeplattform är en viktig komponent i en integrationsplattform som hanterar asynkron kommunikation mellan olika system och applikationer.

### Huvudsakliga funktioner

- **Meddelandehantering:** Meddelandeplattformen hanterar och vidarebefordrar meddelanden mellan olika system och applikationer. Detta gör det möjligt för system att kommunicera utan att vara direkt anslutna till varandra
- **Asynkron kommunikation:** Meddelandeplattformen möjliggör asynkron kommunikation, vilket innebär att system kan skicka och ta emot meddelanden oberoende av varandra. Detta förbättrar prestanda och skalbarhet
- **Köhantering:** Meddelandeplattformen använder köer för att lagra meddelanden tills de kan behandlas av mottagande system. Detta säkerställer att meddelanden inte går förlorade och hanteras i rätt ordning
- **Pub/Sub-modell:** Meddelandeplattformen kan använda en publicerings/prenumerationsmodell där meddelanden publiceras till en kanal och prenumeranter får dessa meddelanden. Detta är särskilt användbart för att distribuera meddelanden till flera mottagare
- **Felhantering:** Meddelandeplattformen hanterar fel och undantag som kan uppstå under meddelandeöverföring, vilket säkerställer att system kan återhämta sig från problem och fortsätta fungera korrekt.

Två viktiga egenskaper hos ett meddelandehanteringssystem är transaktionssäkerhet och garanterad leverans.

- **Transaktionssäkerhet** innebär att en integration följer principerna för ACID (Atomicity, Consistency, Isolation, Durability), vilket säkerställer att dataöverföringar är korrekta, fullständiga och inte leder till inkonsistens – även vid systemfel eller krascher.
- **Garanterad leverans** i en meddelandekö betyder att ett meddelande alltid kommer att nå sin destination, även om mottagarsystemet är nere då meddelandet skickas. När en meddelandehanteringssystemet bekräftat att det tagit emot ett meddelande tar det över ansvaret från det skickande systemet, lagrar meddelandet persistent och gör vid behov upprepade försök att leverera till mottagande system tills att överföringen lyckas.

## Databaser

Databaser har en central roll i en integrationsplattform genom att lagra och hantera data som flödar mellan olika system och applikationer.

### Huvudsakliga funktioner

- **Datahantering:** Databaser lagrar data som samlas in från olika system och applikationer. Detta inkluderar både strukturerade data (t.ex. tabeller) och ostrukturerade data (t.ex. dokument och filer)
- **Dataintegration:** Databaser möjliggör integration av data från olika källor genom att kombinera och harmonisera data till ett enhetligt format. Detta säkerställer att alla system har tillgång till konsekvent och aktuell information
- **Dataöverföring:** Databaser fungerar som mellanstationer för dataöverföring mellan system. Data kan extraheras från en källa, transformeras och laddas in i en annan databas eller system (ETL-processen: Extract, Transform, Load)

- **Dataanalys och rapportering:** Integrationsplattformar använder databaser för att lagra data som kan analyseras och rapporteras. Detta hjälper företag att få insikter och fatta välgrundade beslut baserat på integrerade data.
- **Säkerhet och åtkomstkontroll:** Databaser i en integrationsplattform implementerar säkerhetsåtgärder för att skydda data och säkerställa att endast auktoriserade användare har åtkomst till känslig information

## Säkerhetskomponenter

Säkerhetskomponenter är avgörande för att skydda data och system i en integrationsplattform, dessa säkerställer att data och system är skyddade mot obehörig åtkomst och attacker. För att upprätthålla API:ers integritet måste lösningen innehålla en robust säkerhet, inklusive autentisering, auktorisering och kryptering för att skydda känslig information.

### Ingående komponenter

- **Autentisering och auktorisering:** Autentisering verifierar identiteten hos användare och system som försöker få åtkomst till plattformen. Vanliga metoder inkluderar användarnamn och lösenord, API-nycklar, OAuth och JWT (JSON Web Tokens). Auktorisering: Bestämmer vilka resurser och tjänster en autentiserad användare eller system har rätt att komma åt. Detta kan hanteras genom rollbaserad åtkomstkontroll (RBAC) eller attributbaserad åtkomstkontroll (ABAC)
- **Kryptering:** Data i vila: Kryptering av data som lagras i databaser och filsystem för att skydda den mot obehörig åtkomst
- **Data i transit:** Kryptering av data som överförs mellan system och applikationer, vanligtvis med hjälp av TLS/SSL-protokoll
- **Övervakningsverktyg** används för att kontinuerligt övervaka systemets prestanda och säkerhet, och för att snabbt identifiera och reagera på säkerhetsincidenter
- **Säkerhetskopiering och återställning:** Regelbundna säkerhetskopieringar av data och systemkonfigurationer för att säkerställa att data kan återställas i händelse av en incident

## Övervakning och loggning

Övervakning och loggning är kritiska komponenter i en integrationsplattform för att säkerställa att systemet fungerar korrekt och för att snabbt kunna identifiera och åtgärda problem.

### Övervakning

- **Prestandaövervakning:** Övervakar systemets prestanda, inklusive svarstider, CPU-användning, minnesanvändning och nätverkstrafik. Detta hjälper till att identifiera flaskhalsar och optimera systemets prestanda.
- **Hälsokontroller:** Regelbundna kontroller av systemets komponenter för att säkerställa att de fungerar korrekt. Detta inkluderar kontroll av API:er, databaser och meddelandehanteringssystem.

- **Incidenthantering:** Identifierar och hanterar incidenter som kan påverka systemets funktionalitet. Detta inkluderar automatiska varningar och notifieringar när problem upptäcks.
- **Säkerhetsövervakning:** Övervakar säkerhetsrelaterade händelser, såsom obehöriga åtkomstförsök och potentiella attacker. Detta hjälper till att skydda systemet mot säkerhetshot.

### Loggning

- **Aktivitetsloggar:** Loggar alla aktiviteter och händelser inom integrationsplattformen, inklusive API-anrop, dataöverföringar och systemändringar. Detta ger en detaljerad historik över vad som har hänt i systemet.
- **Felhantering:** Loggar fel och undantag som uppstår under systemets drift. Detta hjälper till att felsöka problem och förbättra systemets stabilitet.
- **Auditloggar:** Loggar säkerhetsrelaterade händelser, såsom autentisering och auktorisering. Detta hjälper till att spåra och analysera säkerhetsincidenter.
- **Analys och rapporter:** Använder loggdata för att generera analyser och rapporter som ger insikter om systemets prestanda och säkerhet. Detta hjälper till att fatta välgrundade beslut och förbättra systemets effektivitet.