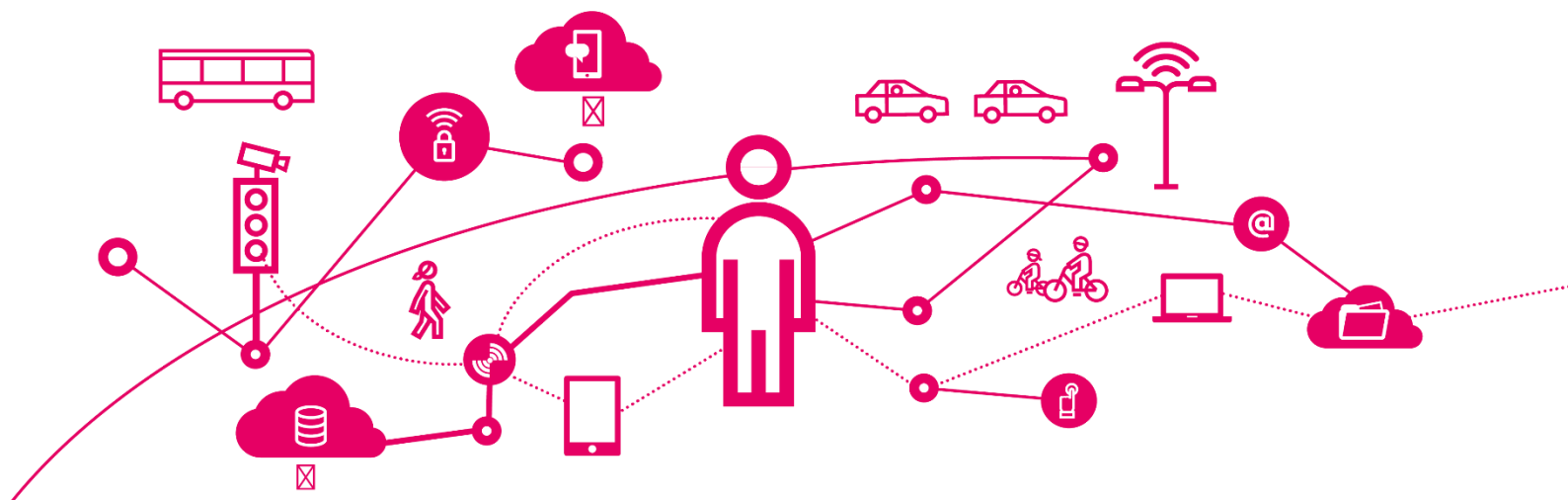


Vägledande dokument

Målkitektur för IoT och dataplattform



Sammanfattning

Denna arkitektur beskriver, i mera detalj, en av de fyra konceptuella delar av en smart stad som kräver omarbetade eller utökade it-förmågor för att kunna påbörja realiseringen av Stockholm som en smart och uppkopplad stad.

De fyra konceptuella delarna som stadens förmågor kan kompletteras inom är:

- En *IoT och data arkitektur* för interaktion med tingen i staden eller andra datakällor med strömmande data
- En arkitektur för *delning av information* från staden både inom och utanför stadens verksamheter
- En *tjänsteorienterad arkitektur* för att hantera förfrågningar från medborgare, företag eller mellan förvaltningar i staden och andra samhällsfunktioner för påverkan av tingen inom staden
- *Platser för interaktion mellan människor och den Smarta staden*, kan vara portaler, appar eller likande platser i framtiden för åtkomst till ovanstående förmågor på ett samlat sätt

Detaljerna för hur den IoT baserade arkitekturen och datahanteringen ska realiseras beskrivs i denna arkitektur. Arkitekturen beskriver koncept för hur Stockholm stad generellt ska använda sig av IoT förmågor i verksamhetsimplementationer samt hur samverkan mellan dessa förmågor och mera verksamhetsnära funktioner ska gå till.

Begrepp

Begreppet *förmåga* förekommer i dokumentet och beskriver VAD som en funktion ska göra. Förmågor kan ses som de byggstenarna i en arkitektur där varje förmåga är unik, oberoende och representerar flera stabila funktioner som behöver utföras för att arkitekturen kan skapa och leverera värde. I målarkitekturen beskrivs översiktligt HUR varje förmåga kan realiseras och samverka med andra förmågor samt om eventuella nya eller befintliga tjänster ska användas för att realisera förmågorna.

Innehåll

Sammanfattning	2
1 Summering	6
1.1 Bakgrund och omfattning	6
1.2 Mål och avgränsningar	7
2 Inledning	9
2.1 Varför behövs en målarkitektur?	9
2.2 Strukturen innehåller principer istället för detaljer	11
2.3 Syfte med arkitekturen	11
2.4 Målgrupp	11
3 Arkitektur – logisk indelning.....	13
3.1 Den smarta stadens arkitektur	13
3.2 Komplexitet i den smarta staden.....	16
3.3 Styrande principer	16
3.4 Övergripande förmågor	18
3.5 Uppdelning	20
4 Uppdelning av arkitekturen och begrepp	21
4.1 Konceptuella dataflöde	21
4.2 Begreppsmodell för arkitekturen	22
5 Edge	24
5.1 Enheter.....	24
5.2 Infrastruktur och kommunikation.....	26
5.3 Säkerhet i Edge	28
5.4 Enhetshantering	30
5.5 Funktion i Edge och Edge computing	31
6 Centrala förmågor för IoT och datahantering	32
6.1 Händelsestyrd arkitektur för datahantering	34
6.2 Förmågor och funktioner	34
6.3 Datainsamling och enhetskommunikation	35
6.4 Hantera data och meddelande.....	38
6.5 Lagring	40
6.6 Dataarkitektur.....	41
6.7 Enhetshantering	43

6.8	Förhållande till övriga plattformar i staden	46
-----	--	----

1 Summering

1.1 Bakgrund och omfattning

Stockholms stads vision för 2040 är att Stockholm ska bli världens smartaste stad. För att kunna uppnå detta har en strategi för Stockholm som smart och uppkopplad stad tagits fram. Med smart och uppkopplad stad menas en stad som utnyttjar digitalisering och ny teknik för att förbättra tillvaron för invånare, företagare och besökare.

För att ta tillvara på möjligheterna som digitaliseringen ger och starta förflyttningen mot en smart stad krävs tekniska förutsättningar i form av arbetssätt, plattformar, riktlinjer och standarder. Dessa tekniska förutsättningar ska säkerställa en långsiktig utveckling mot en smart stad som är kostnadseffektiv samt där tekniska lösningar och arbetssätt kan utnyttjas gemensamt mellan samtliga verksamheter och förvaltningar inom Stockholm Stad samt externa parter som t.ex. andra regionala organisationer, kommuner och myndigheter.

En del av de tekniska förutsättningarna är en målarkitektur som beskriver vilka förmågor som behöver finnas i gemensamma plattformar och vilka styrande principer, standarder och arbetssätt som ska användas vid framtagande av lösningar som använder förmågorna. Detta dokument beskriver övergripande vad som krävs för realisering av en Internet of Things (IoT) och dataarkitektur som ligger till grund för en del av realiseringen av strategin för Stockholm som smart och uppkopplad stad.

Dokumentet beskriver en övergripande nivå av en arkitektur som kan användas för att detaljera respektive område i ytterligare lösningsarkitekturbeskrivningar. Arkitekturen kan fungera som underlag för uppdaterade anvisningar, arbetssätt, standarder samt lösningsarkitekturer för specifika verksamhetsförändringar som har behov av förmågor beskrivna i arkitekturen.

Arkitekturen har använt de prioriterade verksamhetsprojekten i programmet Smart och uppkopplad Stad för att exemplifiera och förtydliga vad och hur beskrivningarna i arkitekturen kan realiseras samt hur förmågorna samverkar och påverkar verksamhetslösningarna både med befintlig arkitektur och framtida nya möjligheter.

Arkitekturen kan alltså komma att uppdateras när detaljeringar och ytterligare kravinsamling inkommer som gör att förutsättningarna ändras. Arkitekturen har alltså skapats för att kunna vara utbyt- och uppdateringsbar i respektive förmåga och funktion.

1.2 Mål och avgränsningar

Frågeställningar som arkitekturen besvarar

Den övergripande vyn som dokumentet beskriver är hur en gemensam IoT- och dataarkitektur för Stockholm stad ska fungera i samverkan med befintlig och ny it-infrastruktur och arkitektur samt de nya behov som uppstår med införandet av fler IoT-projekt på fem till tio års sikt.

Förstudien *Tekniska förutsättningar för genomförandet av strategin för Stockholm som smart och uppkopplad stad* fokuserade på fem områden¹, där denna arkitektur koncentrerar sig på att detaljera områdena markerade med lila färg nedan:



Figur 1 - delfrågor som tidigare utredningar fokuserat på

I detta dokument beskrivs vilka nödvändiga förutsättningar som krävs av en gemensam arkitektur² för insamling av olika dataströmmar i realtid samt möjlighet att kunna påverka och styra de uppkopplade tingen i omvärlden. Arkitekturen beskriver också hur hanteringen och övervakning av enheter, Enhetshantering, samt hur delning av ting och dess dataströmmar ska fungera. Delarna skapar en arkitektur som stödjer den behovsbild i form av digitala förmågor som krävs för att starta genomförandet av *Strategin för Stockholm som smart och uppkopplad stad*.

Arkitekturen fungerar framförallt för att kunna skapa nödvändiga tekniska förutsättningar som ska fungera som grundläggande kravställning för nya digitala lösningar alternativt förändringar i befintliga system som kan komma att krävas för att uppnå målen i strategin.

¹ Stockholm Stad, ”Tekniska förutsättningar för genomförandet av strategin för Stockholm som smart och uppkopplad stad”, sid 7, 2016

² Stockholm Stad, ”Strategi för Stockholm som smart och uppkopplad stad”, sid 4, 2017

Avgränsningar

Anvisningar och vilka tekniska standarder som ska användas beskrivs inte i detalj i detta dokument utan endast vägledningar i hur dessa kan formuleras och beskrivas i verksamhetslösningar.

Denna arkitektur förhåller sig till andra stadsövergripande arkitekturer som är nödvändiga för all digitalisering. IoT-arkitekturen är en del av den smarta stadens arkitektur varför den har tydliga punkter hur de kan samverka. Därför är denna arkitektur avgränsad från att beskriva andra dessa andra stadsgemensamma arkitekturer som it-säkerhetsarkitektur inklusive funktioner som SOC och SIEM, Molnarkitektur och Informationsarkitektur.

2 Inledning

2.1 Varför behövs en målarkitektur?

Syftet med målarkitekturen är att beskriva en riktning, ett långsiktigt mål och väsentliga val för att kunna skapa en uppkopplad stad. Den beskriver en struktur som motsvarar strategin och planen för den smarta och uppkopplade staden. Strukturen omfattar både verksamhet och it och utgör det spelrum inom vilket verksamheten och it-stödet behöver utvecklas de närmaste åren med en omfattning på 5 till 10 år.

Målarkitekturen ger stöd i hur lösningar ska utformas både vid verksamhetsutveckling och förvaltningsaktiviteter. Den ska vara tillräckligt konkret för att kunna ge arkitekturstöd för planering, styrning och uppföljning av insatser som leder till målet utan att samtliga detaljer har beskrivits i förväg. Den är alltså inte ”komplett” utan ska förfinas och revideras löpande för att fylla det syfte som krävs för aktuella och kommande frågeställningar.

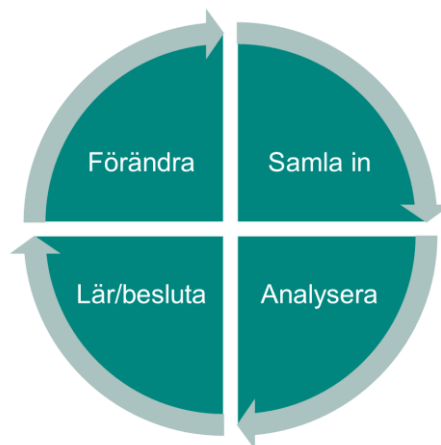
Arkitekturen har utarbetats för att nya behov identifierats bland annat i och med att utökad IoT-funktionalitet införs på en bredare front och för att kunna uppnå strategin mot att bli världens smartaste och uppkopplade stad 2040. Allt detta medför att vissa förmågor saknas eller inte är fullt utvecklade för att stödja förflyttningen, t.ex. kan dessa förmågor vara att hantera en stor mängd data som strömmas in till centrala plattformar från flera olika källor och verksamheter för t.ex. analys och uppföljning. För att också kunna öka nyttan och användandet av data som samlas in från olika källor behövs nya förmågor att lagra, dela och bearbeta data på liknande sätt från olika verksamhetsområden. Delandet av data ska ske både internt mellan stadens verksamheter och externt som öppna data till allmänheten, näringslivet och forskning för att skapa ny möjlighet för nytta till Stockholm som plats.

IoT-arkitektur och teknologier innebär alltid maskin- till maskininteraktion. För att denna information ska ge nytta måste interaktion med människor skapas. Denna interaktion kräver integration med verksamhetssystem, ting, enheter, visualisering av information och digitala platser för tillgängliggörande av information och insikter t.ex. portaler, appar. Det är därför viktigt att principer och riktlinjer sätts för att befintliga eller nya ting kan samverka effektivt för att skapa den nytta som både IoT och datainsamling innebär.

IoT-arkitektur är komplex och påverkar eller interagerar med många delar av befintlig it-arkitektur samt skapar behov av nya förmågor inom närliggande områden och arbetssätt. Det innebär nya sätt för integrations- och kommunikationslösningar, nya förmågor för analys samt påverkan på befintlig säkerhetsarkitektur. Denna arkitektur delas därför upp i områden för att kunna utvecklas och förändras i olika steg och hastighet till viss del oberoende av varandra. Den största utmaningen med en IoT-arkitektur är inte att sätta den på plats utan att kunna utveckla och förvalta den när den växer.

Denna version av arkitekturen kommer att användas främst för att stödja och styra arkitekturen i programmet ”Smart och uppkopplad stad”. Därför har dokumentet prioriterat det som var relevant för programmet men tar ansats i Strategin och den framtida smarta staden.

En grundläggande faktor i arkitekturen är möjligheten att samla in data från omvärlden för att få eller skapa nya insikter och information till underlag för beslut och styrning av den smarta staden. En avgörande förändring i arbetssätt är förkortningen av tiden mellan en förändring i staden och analysen av resultatet till nya insikter och möjligheten till påverkan i form av ytterligare förbättringar i anslutning till förändringen. Detta tillsammans med noggrannheten i informationen t.ex. genom att faktiskt mäta saker som tidigare inte mättes eller mättes sällan och med hjälp av mera data gör att beslutsprocesser kan bli effektivare. Eftersom möjligheter skapas att direkt efter en förändring samla in nytt data gör att tiden mellan analysen och insikten av förändringen kan gå från i värsta fall är år eller månader till veckor och dagar.



Figur 2 - Cykel för hur verksamheten kontinuerligt kan utvecklas med data

2.2 Strukturen innehåller principer istället för detaljer

Eftersom teknologin, verksamheternas behov och omvärlden förändras i allt snabbare takt, speciellt inom områden rörande datainsamling, informationshantering och Smart Stad beskriver arkitekturen övergripande principer och strukturer snarare än detaljer och riktlinjer. Detaljerade principer beskrivs i separata styrdokument, som t.ex. val av kommunikationsprotokoll, säkerhetsriktlinjer och enhetshantering för att dessa ska kunna uppdateras och förändras utan att målarkitekturen och dess principer behöver förändras.

Inom centrala delar beskrivs ett antal förmågor i mer detalj för att dels exemplifiera vad som avses med förmågan dels eftersom förmågorna anses särskilt viktiga att prioritera för införandet av arkitekturen.

2.3 Syfte med arkitekturen

Målarkitekturen har fler syften men i huvudsak:

- Underlag i diskussioner med olika beslutsfattare
- Referensarkitektur för framtida lösningsarkitekturer och detaljering av verksamhetslösningar
- Stödja framtagandet av viktiga principer, riktlinjer och standarder
- Motivering till val av principer och inriktningar
- Underlag för att kunna upphandla och beställa lösningar

Arkitekturen är på relativt hög abstraktionsnivå och i många stycken generaliserande. Delar av arkitekturen kan utgöra referensarkitektur för andra projekt och uppdrag inom samma områden i realiseringen av den smarta staden.

2.4 Målgrupp

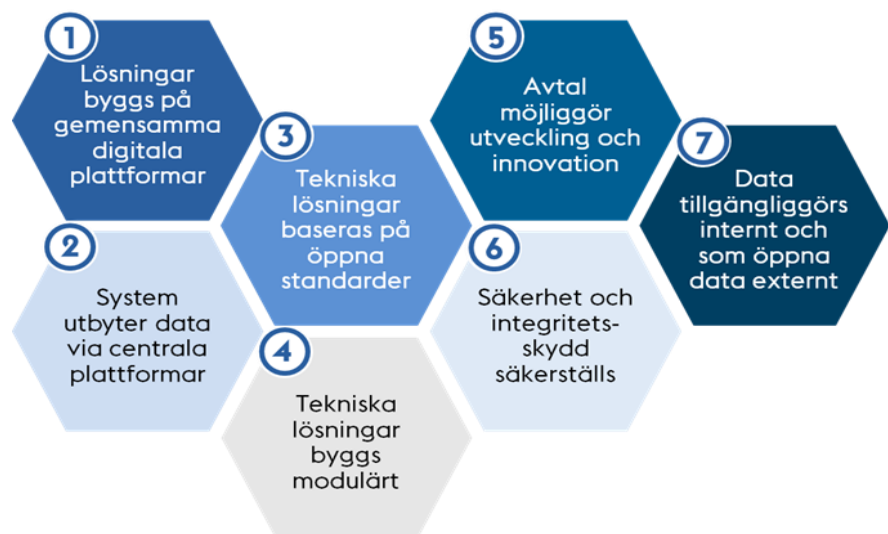
Målarkitekturen är framtagen som stöd och styrning för följande roller:

- Beslutsfattare, t.ex. verksamhetsansvarig chef och programägare
- Programledningen
- Projekten inom programmet
- Arkitekturgrupperingar inom Stadsledningskontoret

- Ansvariga för tekniska plattformar och utvecklingsplattformar
- Objektägare och förvaltningsansvariga

3 Arkitektur – logisk indelning

Arkitekturen utgår och förhåller sig till de möjliggörande strategiska principerna för realisering av den smarta staden³. Varje princip och förmåga ska kunna härledas till nedanstående sju strategiska möjliggörande principer.



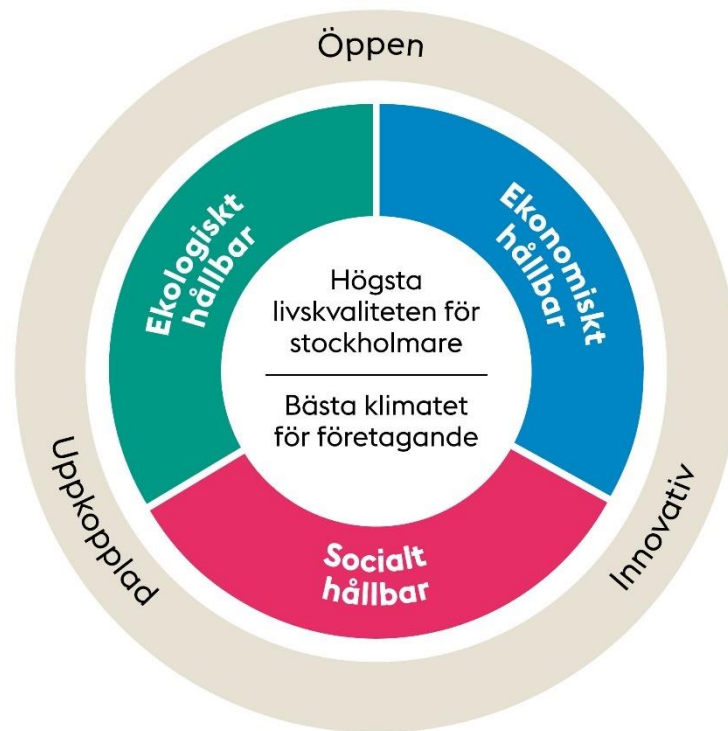
Figur 3 - Strategiska möjliggörande principer från strategin för Stockholm som smart och uppkopplad stad

Principerna i målarkitekturen kommer att hänvisa tillbaka till dessa strategiska principer löpande för att verifiera och säkerställa att dessa följs och kan implementeras.

3.1 Den smarta stadens arkitektur

För att förstå den uppkopplade stadens arkitektur måste den ses i sitt sammanhang i den smarta staden och hur delarna/förmågorna samverkar samt hur dessa tillgängliggörs för både invånarna och användarna. Det måste finnas tydliga visioner och mönster i hur människorna interagerar med förmågorna för att dessa ska kunna utnyttjas effektivt enligt strategins tre lager.

³ Stockholm Stad, ”Tekniska förutsättningar för genomförandet av strategin för Stockholm som smart och uppkopplad stad”, 2016

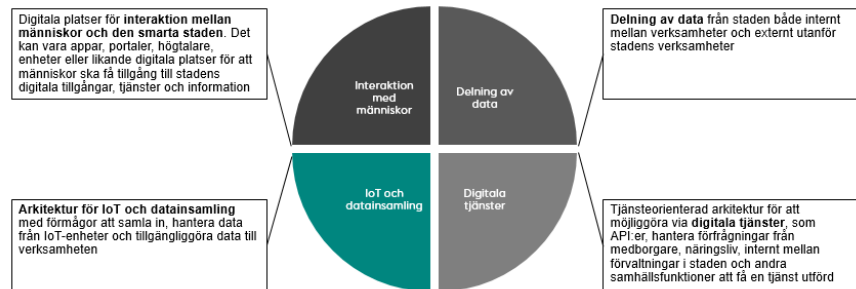


Figur 4 - Strategiska möjliggörande principer från strategin för Stockholm som smart och uppkopplad stad

En smart stad kan definieras i många lager och dimensioner och standardiseringar och likande pågår för att få en enhetlig bild för detta.

Målarkitekturen för den uppkopplade staden har i detta dokument delas upp i fyra konceptuella delar. Denna uppdelning ska möjliggöra för staden att hantera stora mängder strömmande data, automatisera och stödja efterfrågan av tjänster, engagera samhället i att både ta del av nya insikter samt att använda information för att skapa egna nya insikter:

- En *IoT-baserad dataarkitektur* för interaktion med saker i staden
- En *tjänsteorienterad arkitektur* för att hantera förfrågningar från medborgare, företag, mellan förvaltningar i staden och andra samhällsfunktioner
- En arkitektur för *delning av information* från staden både inom och utanför stadens verksamheter
- Platser för interaktion mellan *människor och den Smarta Staden*, kan vara portaler, appar, högtalare eller likande digitala platser i framtiden för åtkomst till stadens tjänster, information, insikter och data för invånare, besökare, entreprenörer eller anställda inom Stockholms Stad



Figur 5 - Delar i den smarta stadens arkitektur

IoT och datainsamling

Detta dokument fokuserar främst på den *IoT och datainsamlade arkitekturen* som skapar en grund för att stadens förmågor för informationshantering, i den uppkopplade staden, kan interagera med ting och enheter i staden.

Delning av information

All information som samlas in inom den smarta staden skall kunna vara tillgängligt både som öppna data till aktörer utanför staden samt delat data till stadens samtliga verksamheter. Förmågor för detta måste finnas både inom verksamhets- och it-arkitekturen för att målet ska kunna realiseras.

Digitala tjänster

Den tjänsteorienterade arkitekturen är en arkitektur för att effektivt och kontrollerat kunna digitalt utnyttja, stadens IoT-arkitektur samt övriga verksamhetsfunktioner i andra delar av it-landskapet. Dessa tjänster möjliggör för alla, både inom och utom staden, att påverka hur stadens enheter agerar, hur digitala processer ska genomföras och uppträda samt möjligheten att interagera med övriga digitala funktioner.

Interaktion med människor

För de som inte vill eller kan konsumera information som delas via Öppna data eller de digitala tjänsterna/API:erna ska staden och stadens verksamheter kunna visualisera informationen och tillgängliggöra dessa tjänster via digitala platser. Dessa platser ska kunna vara tillgängliga för medborgare, besökare, anställda och entreprenörer eller andra som kan behöva få tillgång till informationen.

Digitala platser kan t.ex. vara en portal, app, högtalare, informationstavla, trafiksignal eller ljus som tillgängliggör via visualisering av verksamhetsfunktionerna och -informationen till de användare som har behov av funktionen. På samma platser ska även

stadsövergripande insikter och funktioner kunna hanteras och presenteras.

3.2 Komplexitet i den smarta staden

Det är tre olika typer av system; fysiska (tingen), digitala (tjänster) och mänskliga som tillsammans måste kunna samverka för att den smarta staden ska bli just smart. Detta är en komplex uppgift att lösa för att det krävs förändringar i flera olika delar. För att effektivt hantera komplexiteten beskriver denna arkitektur tillräckligt mycket för att starta resan mot målet genom att följa mönstret i arkitekturen för kopplingen mellan de fysiska och digitala systemen samt ett förslag i hur det går att dela upp utmaningen i flera delar.

3.3 Styrande principer

För att förklara och förenkla tolkningen av arkitekturen har ett antal styrande principer definierats. Nedan beskrivs de övergripande principerna som är kopplade mot de strategiska principerna.

Gemensamma plattformar och data via centrala plattformar

Denna målarkitektur beskriver en sammanhållen arkitektur med gemensamma mönster för hur interaktion med IoT-enheter ska ske. De delar av arkitekturen som delas av flera verksamheter och kunna aggregera data till ny information via t.ex. insikter, AI och visualisering för beslutsunderlag för övergripande insikter för hela staden, ska hanteras av centrala plattformar.

Delat och öppna data behöver också hjälp av centrala insamlade plattformar för att kunna fungera på ett enhetligt sätt och följa strategin oberoende av typen av datakällor och dess innehåll.

Samverkan mellan lösningar och verksamheter, både inom och utom staden, underlättas av en gemensam uppsättning förmågor för att utbyte av information kan göras när nya lösningar utvecklas.

Unika verksamhetslösningar för styra IoT-enheter i staden kommer att krävas speciellt när styrningen och upprätthållande av enheterna är en del av verksamhetens utförande, t.ex. trafikstyrning. Där kan förutsättningarna för att detta kan ske i samma plattform som de gemensamma förmågorna kan vara begränsade i fråga om kostnader eller kunskap. Dock ska alltid dessa lösningar följa arkitekturen och de riktlinjer som har skapats genom denna.

System utbyter data via centrala plattformar

Arkitekturen beskriver förmågan att tillgängliggöra data och dataströmmar för att kunna delas och öppnas. Denna förmåga är det grundläggande sättet att dela data som samlas in centralt inom arkitekturen. Tillgänggörandet av data via API:er i den centrala plattformen sker till övriga delar av stadens arkitektur via Integrationsplattformens förmågor och vidare till verksamhetssystem, analysplattformar, öppna eller delade dataportaler.

Tekniska lösningar baseras på öppna standarder

För många delar inom IoT och Smarta städer pågår standardiseringsarbeten och förslag till standarder. Det är få områden som det finns färdiga De-Facto standarder som går att använda idag speciellt när det gäller hanteringen av enheterna (Edge).

Arkitekturen tar höjd för detta genom att vara så utbytbar och flexibel som det är möjligt med dessa förutsättningar genom att ta utgångspunkt från de mönster och trender som kan överleva i åtminstone ett medellångt perspektiv och välja bland de standarder där samarbeten finns i industrin. Dessa val av standarder ska användas som krav för nya verksamhetsrealiseringar inom arkitekturen håller över tid och baseras på öppna standarder.

Tekniska lösningar byggs modulärt

Uppdelningen av arkitekturen i olika förmågor och delar skapar möjligheten att hanteras modulärt och skapar möjlighet att implementera förmågorna separerat från varandra. Det modulära skapar också möjlighet att kunna byta ut eller komplettera förmågor med nya teknologier eller leverantörer utan att hela den centrala plattformen behöver bytas ut eller hur delarna i Edge kan samverka. Dessutom kan flera olika tekniker samverka då tydliga gränssnitt mellan modulerna upprättas.

För att uppnå detta krävs ytterligare en underprincip att system, komponenter och funktioner i staden behöver vara *öppna och väldefinierade*, vilket innebär att det data och de verksamhetsfunktioner som system och komponenter kan hantera måste vara möjliga att komma åt för att kunna integreras med varandra via API:er.

Säkerhet och integritetsskydd säkerställs

När det gäller IoT-teknologier är säkerhet en stor tekniska utmaning. It-säkerhet och informationssäkerhet är avgörande för hur

förmågorna kan användas och påverkar dessutom hur principer och teknologier får användas. För varje del i arkitekturen beskrivs översiktligt vad som krävs för att upprätthålla it-säkerheten. Inom varje del av arkitekturen från sensorn till de gemensamma plattformarna för analys och delning av data måste säkerheten alltid definieras och upprätthållas i respektive verksamhetslösning.

Återanvändning

Återanvändning betyder att gjorda investeringar ska, om lämplig, återanvändas och eventuellt uppgraderas. Dessutom kan nya gemensamma resurser återanvändas om dessa funktionellt fungerar lika, t.ex. befintlig processorkraft och algoritmer implementerade för machine learning eller lagringsresurser för stora datamängder.

Följden blir alltså att det inte alltid är nödvändigt att byta ut befintliga funktioner i dagens arkitektur för att stödja nya förmågor som beskrivs i detta dokument utan befintliga kan kanske vidareutvecklas eller uppgraderas.

3.4 Övergripande förmågor

Den översiktliga arkitekturen för IoT och datahantering är uppdelad funktionellt för att varje förmåga i arkitekturen ska vara oberoende av andra delar och vara möjlig att byta ut eller uppdatera individuellt samt att göra arkitekturen skalbar i alla delar.

Detta innebär att gränssnitten, API:erna, mellan de olika förmågorna måste vara väldefinierade som gör det möjligt att välja rätt teknologi och implementation för varje förmåga. Då kan också förmågorna kompletteras, utvecklas och realiseras i steg, oberoende av varandra, eftersom gränssnitten mellan dem är väldefinierade.

Arkitekturen måste kunna hantera en mängd olika enheter/ting som inte kan definieras på förhand varken till funktion eller antal. Det måste därför finnas flera olika scenarios och mönster som måste stödjas eller kunna byggas ut till en gemensam lösning för de olika lösningarna. Detta kräver en öppenhet mot standarder som kan komma att kräva beroenden till olika typer av tredjepartsprogramvaror som ska passa in i arkitekturen och komplettera befintliga förmågor.

Arkitekturen behöver genomgående vara händelse (event) baserad för att stödja en skalbar lösning i varje del, som beskrivs i detalj senare.

Säkerhet

För alla delar i arkitekturen måste grundläggande it-säkerhet alltid vara ett baskrav för val av teknologi och lösning samt att lösningsdesignen ska ske med ”Säkerhet först” perspektiv. Det gäller både för data under transport såväl som i vila (under lagring) varför stor vikt behöver läggas på att den grundläggande infrastrukturen är säker i Staden. Både enheter och användare måste kunna identifiera sig för att kunna ha tillgång till de data och dataflöden som skapas och produceras. De grundläggande säkerhetsförmågor som krävs för all datahantering i staden kan återanvändas i IoT-lösningarna och kompletterat till detta krävs det en sammanhållen, övergripande möjlighet för att kontrollera och övervaka säkerheten för samtliga lösningar och plattformar.

För it-säkerhet relaterad till IoT gäller samma it-säkerhetsarkitektur och riktlinjer som för all datahantering, konfidentialitet, integritet och tillgänglighet. Unika säkerhetsegenskaper för att säkra enheter finns beskrivet under avsnittet Edge samt under förmågan Enhetshantering. IoT handlar om dataströmmar och inte lika mycket om system som traditionell arkitektur varför informationssäkerheten och it-säkerheten måste säkerställas på informationsnivå.

Spårbarhet

Det är nödvändigt att genom hela arkitekturen kunna spåra var information och anrop härstammar för att kunna stödja felsökning och uppföljning i hur lösningar fungerar. Spårbarheten är också grundläggande för att de stadsövergripande it-säkerhetsförmågorna ska kunna användas i IoT-lösningarna.

Spårbarheten är också nödvändigt för att kunna spåra användningen av data och göra uppföljning på t.ex. datakvalitet och eventuella fel och brister. Gemensam loggfunktion för teknisk spårbarhet är nödvändigt för att stödja detta.

Tillgänglighet

Tillgängligheten måste bedömas av användningsområdet för varje enhet och datakälla. Eftersom förmågorna måste ha stöd för många olika mönster av tillgänglighet och dessutom skalbart på bredden för att kunna hantera t.ex. en snabb ökning av enheter/ting måste tillgängligheten på dessa centrala delar alltid vara mycket hög.

IoT tillför ytterligare en dimension till tillgängliggörandet av data mot traditionellt data, där hastigheten och åldern på informationen är av betydelse.

3.5 Uppdelning

Den övergripande uppdelningen beskrivs i nedanstående figur 4; färgsättningen i bilden används i resten av beskrivningen för arkitekturen:

Grön

Den gröna färgen beskriver de delar av arkitekturen nedan kallad *Edge* som beskriver tingen/enheterna och de närliggande förmågorna ute i stadsmiljön, som kan kommunicera med de gemensamma förmågorna eller IoT-verksamhetssystem.

Blå

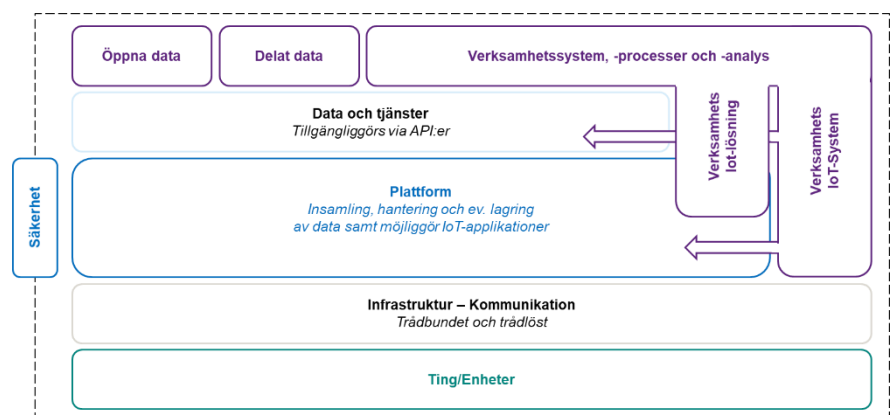
Blå beskriver de gemensamma förmågorna som IoT och dataarkitektur behöver för att kunna ta emot data eller skickas styrning till enheterna, samt hanteringen av dessa förmågor in och ut från resten av it-arkitekturen.

Lila

Lila är de verksamhetssystem som styr och utnyttjar tingen i den verkliga världen för att utföra verksamhetsfunktioner, t.ex. styra belysning i ett kvarter, skapa en digital kopia av en fysisk plats, tilldela en digital nyckel; tillgång till ett utrymme, eller mäta vattentryck samt andra verksamhetssystem som kan dra nytta av informationen eller tjänsterna som förmågorna i IoT-arkitekturen skapar.

Ljusblå

Ljusblå används där arkitekturen använder/återanvänder befintliga delar i stadens centrala plattformar som befintligt Integrationsplattform via dess API-gateway.



Figur 6 - Övergripande målbild för den smarta staden arkitekturen med den centrala IoT-plattformen positionerad

4 Uppdelning av arkitekturen och begrepp

Arkitekturen har delats upp i tre områden för att kunna hantera och beskriva principerna för respektive område. Beroende på var i arkitekturen en princip hör hemma behöver den uttryckas och implementeras olika för att uppfylla de styrande principerna. Det finns en samverkan mellan varje del som beskrivs i respektive del hur den kan ske.



Figur 7 - Logisk uppdelning av målarkitekturen

4.1 Konceptuella dataflöde

Det konceptuella flödet och ansvaret mellan de olika delarna beskrivs enligt bilden nedan.



Skapandet av data och information sker i omvärlden. I detta schematiska flöde skapas informationen i Edge, som förutom enheter eller IoT-verksamhetssystem även kan vara externa datakällor som den Smarta Staden är i behov av intresserad. Dessa ”dataskapare” har en förmåga att löpande skicka information till de gemensamma förmågorna. Det är alltså frågan om att för staden kunna ta emot data löpande, när data produceras, snarare än att hämta data. Om skapandet av data i Edge av Stadens enheter beskrivs detta närmare i Edge-delen nedan. Där beskrivs generella mönster som möjliggör för gemensamma förmågor att prenumerera på dataflöden genom API:er med hjälp av den infrastruktur som finns tillgänglig i Edge.

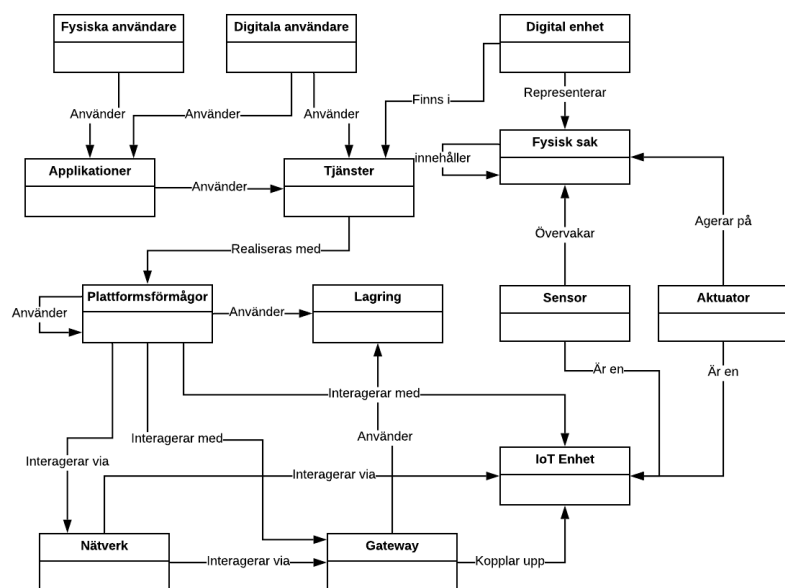
Prenumeration är ett enkelt sätt för att stödja en händelsestyrd arkitektur, dvs. när skaparen av data vill dela med sig kan mottagaren lyssna. Detta bidrar till en bra förmåga att kunna skala ut och upp förmågorna efter behoven i flödeskedjan.

Mottagningsdelen i Plattformen har förmåga att lyssna och ta emot data på flera kanaler samtidigt, samla in och hantera dessa dataströmmar vidare genom arkitekturen till de lyssnare som är intresserade. Hanteringen av data skickar alltså vidare, efter nödvändig bearbetning, dataströmmarna till lagring, analys och/eller delning/integration mot verksamhetsdelen av arkitekturen. Analys av data och dataströmmar kan ske i Edge, i centrala plattformen eller i verksamhetssystem, beroenden på vilka egenskaper som krävs av analysen och det faktiska behovet/kravet.

Användningen av informationen sker i verksamhetssystemen genom att informationen integreras i de verksamhetsprocesser systemen används alternativt visualiseras, delas eller öppnas till olika portaler för användare. Det finns också möjlighet att användningen av data sker direkt i Edge för beslut i enheternas närhet eller i någon av de gemensamma centrala förmågorna och återkopplas tillbaka till Edge.

4.2 Begreppsmodell för arkitekturen

Nedan beskrivs en visuell modell över de begrepp som används i arkitekturen och hur de förhåller sig till varandra. Modellen är inspirerad av ISO 30141⁴.



Figur 8 - Begreppsmodell för IoT

⁴ ISO, <https://www.iso.org/news/ref2361.html>, “Architecting a connected future”, Jan 2019

Fokus för modellen är att visa vilken relation respektive begrepp har till varandra i begreppslistan. Några begrepp som kan behöva förklaras ytterligare är:

Nätverk

Nätverk är ofta refererad till infrastruktur i arkitekturen, men är den del av arkitekturen som möjliggör kommunikationen mellan enheter och övrig it-arkitektur. Nätverket kan vara både trådbundet och trådlöst.

Tjänster

Tjänster beskriver de digitala tjänster, gjorda för maskin till maskininteraktion, som möjliggör att skapa nya applikationer eller inkluderas i befintliga verksamhetsapplikationer för att få tillgång till enheter och information från enheter. Detta inkluderar både delat och öppet data, samt digitala tjänster och realiserar ofta som API:er.

Digital enhet

Kallas ofta digital tvilling, den digitala tvillingen består oftast av flera digitala enheter som sammankopplas. Kort är den digitala enheten en virtuell kopia av den fysiska saken.

5 Edge

I Edge finns framförallt de enheter/ting med sensorer och/eller aktuatorer som ska kopplas upp. Det kan även finnas mera avancerade ting i form av gateways eller smarta enheter som kan innehålla programvaror som agerar mellanhand mellan enheterna och de gemensamma förmågorna. Det som utmärker IoT är möjligheten att använda enheter som kan *identifiera sig, omvandla* analoga signaler till digitalt data samt *koppla upp sig* för att kommunicera med omvärlden.

Nedan beskrivs de grundläggande egenskaper som ska finnas i Edge-delen av arkitekturen för att skapa förutsättningarna för den uppkopplade staden. Edge-delen av arkitekturen är behovsstyrd eftersom enheterna och ibland även programvaran som styr och hanterar data från tingen i princip alltid köps in från tillverkare. Det ligger inte i stadens intresse att själva utveckla och tillverka enheter och skapa generiska plattformar för Edge utan snarare ha möjlighet att välja det som passar verksamhetsbehovet bäst. Denna sektion innehåller därför principer för hur dessa enheter och programvaror ska fungera för att följa resten av målarkitekturen.

En viktig egenskap med Edge-delen, som genomsyrar alla lösningar som följer arkitekturen, är att det ofta är svårt (dyrt och komplext) att ändra i Edge eftersom det oftast innebär utbyte av enheterna såväl som konfigurering av kommunikationen. Därför är det viktigt att kunna förstå och kravställa på vilka delar av enheterna som kan kontrolleras och förändras digitalt på avstånd.

5.1 Enheter

Enheter är uppdelad i tre funktionella delar som kan höra ihop i samma ting men kan vara separerade ifrån varandra:

Sensor och aktuator

Den del av enheten som interagerar med den fysiska världen och kan bestå av två delar:

- Sensor; mäter eller känner av omgivningen för var den är placerad t.ex. temperatur, passage eller ljud.
- Aktuator*; Agerar på omgivningen och påverkar direkt tinget, t.ex. som av/på knapp, ljusstyrka, ljud.

*Definition av Aktuator: "Ett ställdon (här kallad aktuator efter engelskans actuator) är en anordning som används för att styra en mekanism eller mekaniskt system. Ställdonet styrs av en

signal och omvandlar denna signal till en mekanisk rörelse eller en annan fysikalisk effekt. Signalen är ofta elektrisk, men kan också vara hydraulisk eller pneumatisk. Ställdon får normalt sin styrsignal från ett styr- och reglersystem som med hjälp av information från sensorer beräknar vad som ska utföras. Styr- och reglersystemet kan vara enkelt (ett fast mekaniskt eller elektriskt reglersystem), mjukvarubaserat (till exempel styrningen av en skrivare, styrsystem för en robot), eller manuellt. Ställdon är därmed den mekanism som styrsystemet använder för att reagera i en miljö.”

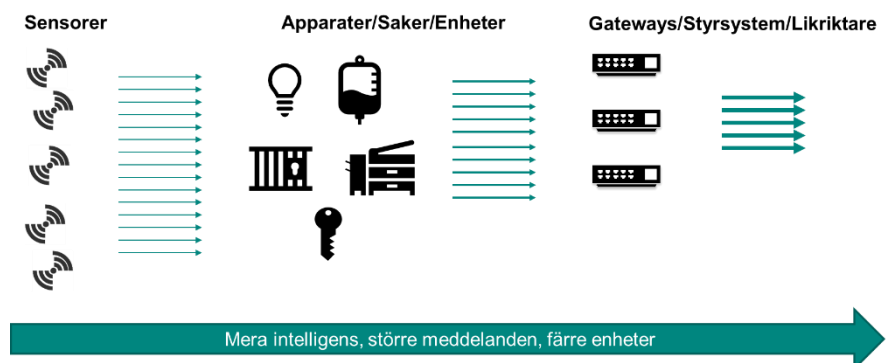
Hantering och kontroll

Denna del av enheten tar emot och hanterar data från sensorn/sensorerna och kan utföra styrning av aktuator om en sådan finns i samma enhet som sensorn är placerad i. Styrningen kan vara olika avancerad/intelligent i hur den kan kontrollera aktuatorn beroende på data från sensorn; från enkel styrningen som att sätta på en lampa om ljussensorn mäter ljusstyrka som understiger ett gränsvärde till att avgöra om data ska skickas vidare till kommunikationsfunktionen.

Kommunikation

Denna funktion ger enheten möjlighet att kommunicera med omgivningen genom att vara uppkopplad. Kommunikationen kan ske med flera delar som direkt andra ting i Edge eller mera centralt placerade förmågor. Sättet som denna kommunikation sker är via olika typer av infrastruktur och kommunikationsprotokoll beroende av behov och möjlighet i både platsen och enhetens förmåga.

Det finns många typer av enheter när det gäller teknisk uppbyggnad samt att det även varierar mellan verksamheter/industrier. Denna arkitektur pekar inte ut specifika teknologier i hur en enhet är/ska vara uppbyggd utan det hänvisas till respektive branschstandarder.



Från enheterna ska tvåvägskommunikation kunna ske både till och från *Edge och Verksamhetssystem* via den gemensamma IoT och Dataarkitekturens förmågor. I de fall där enheterna är många och/eller producerar en så stor mängd meddelanden att det inte är

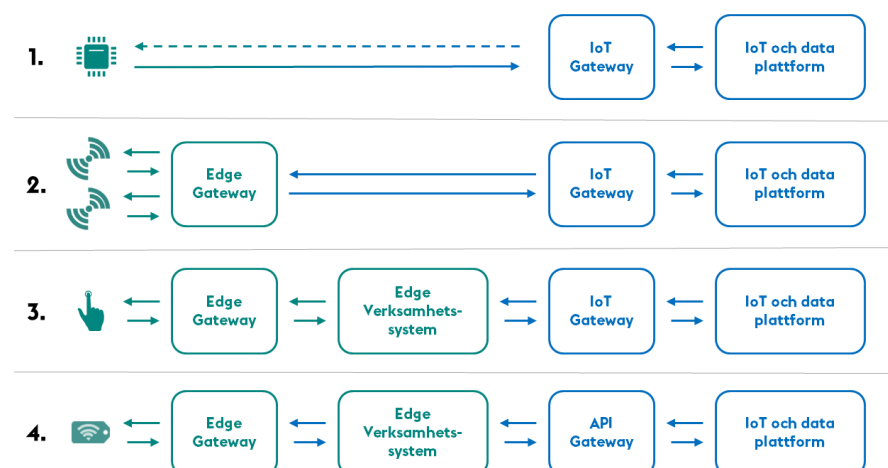
hanterbart med de gemensamma förmågorna kan ”verksamhetsspecifika” IoT-system och/eller Edge-gateways filtrera och aggregera data genom så kallad Edge computing.

Om verksamhetsfunktionerna kräver att det fattas snabba lokala beslut i Edge ska återkopplingen till tingen kunna sker direkt i Edge utan att passera en central plattform. Svarstider och latens (eng. latency) blir längre ju fler komponenter som måste passeras för att komma till hanteringen av data. Dessutom kan andra saker vara avgörande när man behöver använda sig av Edge computing, t.ex. för att skydda integritet där man kan behöva filtrera bort eventuella integritetskränkande information direkt i Edge innan data skickas vidare genom arkitekturen.

En ytterligare aspekt på att hantera information i Edge är kostnad och hållbarhet för att transportera, centralt hantera och lagra all den information som uppstår i enheterna. Det kan vara effektivare och mera hållbart att bara samla in den vitala informationen centralt och låta övrig information processas i Edge.

Ur en hållbarhetsynvinkel kan det vara bättre att hantera informationen i Edge för att inte använda onödigt många delar av infrastrukturen för att lösa en uppgift.

5.2 Infrastruktur och kommunikation



Figur 9 - Huvudmönster för hur enheter kan kommunicera med centrala förmågor

Kommunikationen från Edge till gemensamma förmågor kan ske på olika sätt beroende på behov och lösning för uppgiften som enheterna ska uppfylla. Kommunikationen kan ske direkt eller via speciella Edge-gateways innan den når de centrala funktionerna.

Fyra principer beskrivs i bilden ovan:

1. Enheter som kan koppla upp direkt mot den gemensamma IoT-gatewayen. Dessa enheter stöder de kommunikations- och dataprotokoll som den gemensamma IoT-gatewayen hanterar. Enhetsstatus och dataströmmar direkt in i den gemensamma IoT-gatewayen som rådata för varje datapunkt som enheten är inställd på att skicka eller som telemetridata om enheten kan producera detta. Kommunikationen behöver vara IP-baserad i detta fall för att fungera. I sektionen om enhetshantering kommer vissa krav på dessa enheter beskrivas beroende på hur hanteringen och kontrollmekanismen fungerar i dessa. Enheterna, i detta fall, skickar data och styrningen av dessa ting är till största delen hantering av enheten som t.ex. statushantering eller uppdatering av programvaror. Exempel på sådan är t.ex. vattenmätare, solceller eller luftkvalitetssensorer.
2. Flera enheter är uppkopplade till en gemensam Edge-gateway som kan analysera, aggregera och filtrera data från enheterna. Kommunikationen mellan enheterna och Edge-gatewayen kan dessutom vara på ett sätt som gemensamma IoT-gatewayen inte stödjer med den infrastruktur som finns i Staden. Edge-gatewayen kan då översätta enheternas protokoll till ett sätt som den centrala hanterar. Det betyder också att dessa enheter inte behöver vara direkt uppkopplade mot stadens nät för att kunna fungera utan denna uppkoppling sker i Edge-gateway. Gateway kan även ha andra uppgifter som att ta egna beslut över styrning av enheterna, buffra data om exempelvis kommunikationen inte fungerar eller sker batchvis till centrala förmågor. Exempel på detta kan vara trådlösa enheter som för att spara på batteri endast skickar data för datainsamling ett fåtal gånger per dygn. Edge-gateway samlar då sensordata och utifrån konfiguration en lagrar data, efter eventuell filtrering, och sedan skickar över allt insamlat data för ett tidsintervall i taget.
3. Enheterna kommunicerar i Edge via Edge-gateway med ett gemensamt *IoT-verksamhetssystem* för att bland annat kunna ta ”smarta lokala beslut” med hjälp av konfiguration som finns i Edge delen. Data samlas upp i Edge verksamhetssystemet och processas för att sedan transporteras vidare till centrala förmågor. Styrningen av enheterna sker via Edge verksamhetssystemet som har

kunskap om den lokala fördelningen av sensorer och hur logiken lokalt är uppbyggd.

Detta skulle t.ex. kunna vara en trafik korsning där sensorer för att mäta och analysera fordon styr korsningens enheter. Denna kommunikation, analys och styrning av korsningens enheter sker lokalt i Edge och Edge-gatewayen. Vid behov skickas data vidare till gemensamma förmågor för t.ex. mera avancerad analys som kan resultera i underlag av beslut som skickas tillbaka till Edge från centrala IoT-gatewayen.

Ytterligare exempel på detta kan vara en mobilapp som fungerar som Edge-gateway och översättare av data från enheten som kommunicerar med mobiltelefonen och via mobilappen kommunicerar med plattformen. Viss logik kan finnas i mobilappen för att ta beslut eller analys direkt i Edge, t.ex. som en personlig ”Fitnesstracker”.

4. Enheterna i Edge kommunicerar med en molntjänst som en leverantör hanterar för att leverera den verksamhetsfunktion som enheterna ska hantera. Molnplattformen fungerar i sin tur som Edge-gateway mot den centrala IoT-gatewayen och strömmar ner data via antingen IoT-gateway eller via API-gatewayen i Integrationsplattformen om den strukturen är mera lämplig. Exempel på detta är t.ex.

Fordonsövervakningen inom Grow Smarter programmet.

Det finns ytterligare ett mönster som kan vara möjligt som *inte* täcks in av detta, som när tingen kommunicerar med varandra direkt utan inblandning av annan infrastruktur eller gateways. Det kan finnas andra specifika standarder inom specifika områden som inte behöver täckas av denna arkitektur.

Infrastruktur som stödjer denna kommunikation beskrivs i separat anvisning för datakommunikation för IoT.

5.3 Säkerhet i Edge

IoT-enheter är en ny vektor för yttre påverkan genom infrastrukturen och kommer att kunna användas som en ingång till övriga förmågor inom Stadens it-system av de som vill göra dataintrång. Därför är det fortsatt viktigt att det ordinarie

säkerhetsarbetet⁵ pågår samtidigt som nya förmågor inom IoT säkras med ytterligare funktioner.

Enheter som är kritiska för staden och samhället, speciellt där styrning av enheter ska ske, t.ex. gatubelysning, trafikljus, vattenförsörjning eller där känslig eller verksamhetskritisk information samlas in måste uppfylla en rad krav på säkerhet⁶.

Lägsta nivån på egenskaper för enheterna i Edge som ska kommunicera med centrala förmågor är:

- Enheterna ska ha certifikatbaserad kommunikation (t.ex. PKI eller 802.x Certifikat) mot centrala plattformar och system
 - För att kunna verifiera att enheten är tillförlitlig
 - För att kunna ta tillbaka certifikat och förnya dessa vid behov
- Kommunikationen till/från enheterna bör kunna vara krypterad
 - För att data inte ska kunna manipuleras under transport
- Identifieringen av enheten måste finnas i hårdvaran (t.ex. via MAC ID eller enhets ID)
 - För att enheter inte ska kunna simuleras som riktiga enheter.
- Enheterna måste kunna uppdateras; firmware, operativsystem eller annan styrprogramvara separerat från varandra
 - Uppdatering av programvara i enheterna nödvändigt speciellt för att det kan uppstå säkerhetsrisker i dem
 - Uppdatering ska ske löpande på alla enheter
 - Uppdateringarna ska kunna fungera på krypterad kommunikation
- Om enheterna levereras med ett lösenord måste dessa kunna ändras
 - Default lösenord ska alltid ändras innan utsättning
 - Om det inte finns någon annan möjlighet än inloggning med lösenord måste det kunna ändras även utan fysisk tillgång till enheten
- Motstånd mot fysisk påverkan av enheter
 - Enheterna måste kunna stå emot fysisk påverkan för att säkert kunna leverera rätt kvalitet och data

⁵ Myndigheten för Samhällsskydd och Beredskap, ”[Säkrare IoT](#)”, 2018

⁶ Myndigheten för Samhällsskydd och Beredskap, ”[Så säkrar du ditt IoT](#)”, 2018

- Avaktivera fysiska interface på enheten som inte används
- Konfigurera enheterna så de bara kan koppla upp mot godkända trådlösa nätverk

Frågor som måste besvaras/dokumenteras för varje uppkopplad enhet är:

- Kan säkerhetsmekanismer uppdateras?
- Är enhetsprogramvara separerad från styrprogramvaran
- Går enheten att styras på något sett centralt, t.ex. att kunna stängas av, startas om.

Säker kommunikation

Endast tillåten trafik från enheterna ska släppas vidare in i stadens nät, varför det måste vara möjligt att konfigurera vilken trafik som släpps vidare in till datainsamlingen. Detta ska kunna ske både i den gemensamma infrastrukturen för Stadens nät och i den centrala förmågan för att koppla upp enheter beroende på lösningen och enheterna.

För andra enheter, t.ex. för vilka det endast samlas in information som inte ägs eller förvaltas av Stockholm Stad kan vissa av dessa krav åsidosättas så länge dessa inte exponeras i Stadens nät. Dock måste detta data vara markerat in till den centrala datahanteringen som osäkrare och hanteras utefter dessa förutsättningar.

5.4 Enhetshantering

Enhetshanteringen beskrivs i detalj senare som en del av de centrala förmågorna. Dock ställer denna förmåga krav på hur enheter och Edge funktioner är konstruerade. T.ex. måste enheterna eller andra Edge komponenter kunna rapportera in viss information till Enhetshanteringen för att den ska kunna fungera säkert och effektivt. Baskraven⁷ för detta är att enheterna kan:

Identifiering

- Identifiering; enheten måste ha en unik identifiering och kunna beskriva vad en utför
- Status; Kunna skicka statusuppdateringar hur enheten fungerar
- Positionering; Kunna skicka koordinater var enheten är placerad vid varje omstart/förflyttning.

⁷ Enligt Stockholm Stad, ”Strategi för Stockholm som smart och uppkopplad stad”, sid 24, 2017

- Batteri; Vid batteridrift kunna rapportera batterinivå
- Version av firmware, operativsystem och styrprogramvara
- Kommunikationssätt
- Sensortyp och sensorkategori

Hantering

- Uppdatering Edge programvara måste kunna uppdateras centralt, på samtliga enheter som berörs samtidigt.
- Styrning; Möjlighet att kunna stänga av och sätta på enheter centralt

Rapportering

- Automatisk felrapportering

Möjlighet att kunna hämta felloggar eller likande från enheterna

5.5 Funktion i Edge och Edge computing

Gateways och Edge-programvaror

Edge-gateways och Edge-programvaror kan ha flera olika funktioner i IoT-lösningar förutom den primära uppgiften att möjliggöra kommunikation både mellan enheter och mellan enheter och de centrala förmågorna och verksamhetens system.

För att åstadkomma hög tillgänglighet för verksamhetskritiska funktioner i Edge kan det krävas lösningar som är oberoende av uppkoppling eller av försörjning av data från centrala system för att kunna fungera. Detta innebär att Edge-programvaran behöver kunna t.ex. ta egna beslut på data som produceras i Edge, analysera dataströmmar, buffra data innan den skickas vidare till centrala förmågor och för att kunna kommunicera ut dessa beslut och data till andra enheter som är lokalt ihopkopplade. Detta brukar beskrivas som Edge computing, där enheter och gateways tar lokala beslut och lokal analys och bearbetning av data för att fatta beslut eller analysera data. Det kan t.ex. vara effektivare att analysera bilder och video i direkt anslutning till kameran och endast skicka vidare tolkningen/informationen i bilderna istället för bilderna själv till de centrala förmågorna.

Exempel på detta kan vara t.ex. en trafik Korsning, där konfigurationen för hur de ska agera tillsammans för att lösa trafiksituationen är nedladdad till korsningens programvara (Edge-programvara) och denna sköter hur enheterna som är ihopkopplade ska fungera och agera tillsammans både i hur dess sensorer uppfattar världen och hur aktuatorer kan påverkas. Om uppkoppling

finns till centrala förmågor och system, kan dessa skicka data som påverkar hur Edge-programvara ska styra enheterna, t.ex. information från kringliggande korsningar, trafiksituationer på andra platser (t.ex. olyckor), snöstatus eller vädersituation. Detta data gör att korsningen får mera insikter och kan göra intelligentare beslut. Men eftersom funktionen är kritiskt måste den också kunna fungera autonomt, dvs. om uppkoppling saknas och ingen styrning kommer från andra datakällor eller styrsystem ska korsningen kunna fungera med den information den producerar själv.

6 Centrala förmågor för IoT och datahantering

De centrala förmågorna för IoT och datahanterings roll⁸ i den smarta stadens arkitektur är att möjliggöra för övriga delar av Stadens arkitektur att kunna interagera med framförallt enheter i stadsmiljön, både genom att ta emot information och kunna direkt påverka dessa. Arkitekturen ska leverera ett gemensamt sätt att realisera de tekniska förmågor som krävs för ett effektivt utnyttjande av de IoT och datahanteringsbehov som finns i den smarta och uppkopplade stadens verksamheter och kunna effektivt utnyttja detta data.

Arkitekturen för de centrala delarna är uppdelad på ett antal förmågor vars huvudsakliga ansvar är att främst samla in och använda och dela data mellan verksamheter och funktioner inom samtliga delar av Stadens it-arkitektur och kunna kommunicera, direkt eller indirekt, med enheter i omvärlden.

Huvudsakliga ansvaret är:

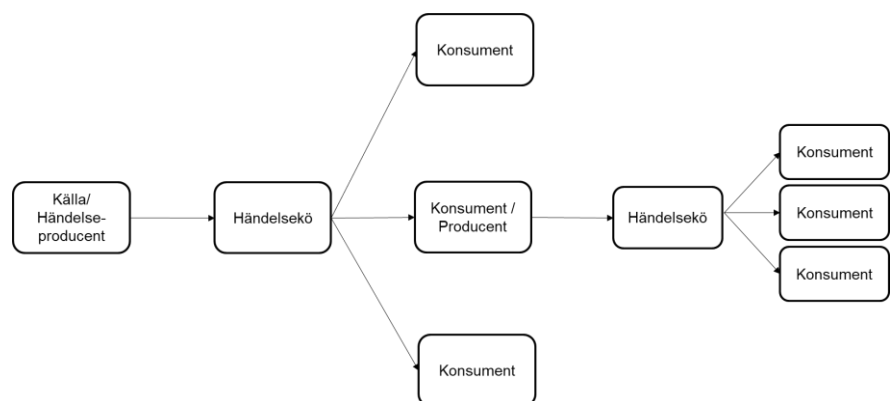
- **Kommunicera** mot dataströmmar till och från enheterna
- **Hantera** data genom transformering och normalisering av data från enheter för senare användning
- Möjlighet till **analys** av snabbt och rörligt data
- Effektiv **lagring** av IoT-data
- Integrera med andra datakällor som metadatakällor, tredjeparts data, och historiskt data
- **Tillgängliggöra** data för användning inom hela Staden
- **Hantera enheter** för att kunna förvalta alla uppkopplade enheter

⁸ Stockholm Stad, "Strategi för Stockholm som smart och uppkopplad stad", sid 26, 2017

Skalbarheten i arkitekturen måste ha förmåga att kunna fungera effektivt med små datamängder med fåtalet konsumenter och samtidigt ha möjligheten att skala upp effektivt när antalet datakällor och/eller konsumenter ökar utan omkonfigurering. Då det kan vara svårt att gissa, på förhand, vilka förmågor som kommer växa snabbast och kräva mest resurser varför en inbyggd elasticitet måste finnas i arkitekturen⁹.

En ny egenskap som IoT inför i Stadens arkitektur är att data kommer med hög hastighet och i stora volymer genom att antalet källor som producerar data är många och där källorna kan vara Stadens egna enheter samt andra befintliga system inom organisationen eller externa källor. Dessa datakällor kommer också producera information som tidigare inte har hanterats i befintlig datahantering och analys.

Huvuddragen för arkitekturen är att den är *händelsebaserad* för att åstadkomma förmågan att stödja denna elasticitet. Det innebär att när en händelse inträffar ”rapporterar” källan/händelseproducenten meddelanden/dataströmmen vidare i kedjan där prenumeranterna/händelsekonsumenterna kan lyssna. Prenumeranterna på detta data/kanal kan då lyssna och ta del av informationen. Händelserna sker alltid i realtid eller nära realtid för att konsumenterna har tillgång till händelserna när de inträffar.



Figur 10 - Visualisering av ett händelsestyrt flöde

Eftersom elasticiteten måste vara stor och leveranshastigheten på data till konsumenter är hög är det inte den som skapar informationens ansvar att se till att den levereras till lyssnarna eller i deras format utan informationen sänds till insamlingspunkten för att

⁹ Stockholm Stad, **Fel! Ingen text med angivet format i dokumentet.**, sid 38, 2017

sedan distribueras vidare. Alltså alla konsumenter är fristående från varandra och kan ta del av alla händelser.

6.1 Händelsestyrd arkitektur för datahantering

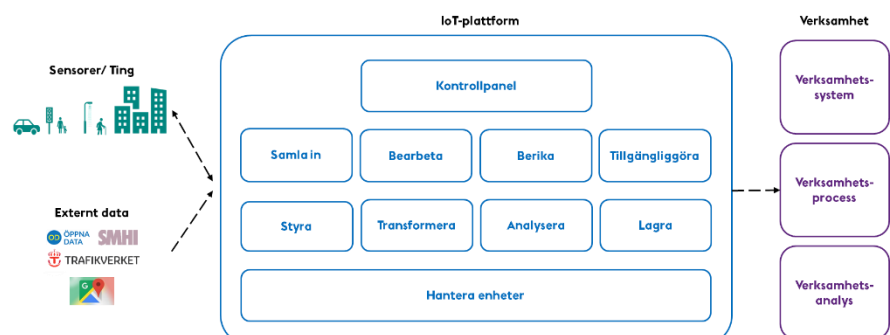
Data i den Smarta Staden skapas i realtid i stora mängder och kommer att användas samt analyseras i nära realtid i framtida lösningar, så kallad snabbt data, tillskillnad från traditionell analys som sker senare än själva händelsen. För att uppnå detta krävs det utökade funktioner och teknologier för all datahantering. De möjligheter som skapas genom att fånga upp realtids/strömmande data kräver nya funktioner och mönster i arkitekturen speciellt för hantering av meddelanden, databearbetning och lagring.

Denna arkitektur stödjer både strömmande data och mera traditionell batchorienterad datafångst och analys samtidigt. Mönstret för att stödja detta är att ha en Meddelandehanterare som tar emot meddelanden via centrala IoT-gateways från omvärlden både från Edge och andra externa datakällor. Meddelandehanterare är specialiserad på att granska mönster i de inkommande meddelanden för att avgöra hur dessa ska kunna skickas vidare genom arkitekturen. Denna del av arkitekturen ska alltså kunna stödja samtliga egenskaper som beskrivs i kapitel 6.6. (Dataarkitektur).

Arkitekturen kan använda antingen:

- **Publish/Subscribe:** När en producent skapat ett meddelande skickas händelsen till samtliga prenumeranter. När händelsen har tagits emot kan den inte återskapas.
- **Händelseström:** Alla händelser loggas och konsumenterna läser från loggen i vilken ordning och från vilken startpunkt som helst.

6.2 Förmågor och funktioner



Figur 11 - Den centrala IoT-plattformens förmågor

6.3 Datainsamling och enhetskommunikation

När det gäller datainsamling finns det många olika källor som arkitekturen behöver stödja, framförallt nya enheter som samlar in data och kommunicerar på ett effektivt och standardiserat sätt. Det finns idag också befintliga sensorer som producerar rådata vilkas data också behöver kunna fångas upp och behandlas, tillsammans med externa datakällor vars information kan vara lika viktig att komplettera dessa dataflöden med.

För att kommunikation ska kunna ske mellan enheter och Plattformen krävs infrastruktur i form av transport och kommunikationsprotokoll. Dessa delar tas upp i separat sektion för Infrastruktur nedan.

Samla in och Styra (Central IoT-gateway)

Enheter som producerar data i Edge behöver ha möjlighet att skicka detta data till gemensamma förmågor för datahantering utan att behöva lagra data. Uppkopplingen av de enheter i Edge som sensorer, Edge-gateways, externa datakällor etc. kommer följa några olika protokoll och kunna hantera stora mängder av enheter och dataströmmar samtidigt. En central IoT-gatewayen är en mjukvarubaserad integrationsfunktion specialiserad på kommunikation med IoT-enheter som kan hantera multipla dataprotokoll och stora mängder data samt möjlighet att skapa säkra/krypterade kanaler för kommunikationen.

En viktig egenskap med denna funktion är dataprotokollen som implementeras gemensamt för samtliga användningsfall. Enheter som används för verksamhetsimplementationer och behöver kommunikation och uppkoppling kan återanvända detta i både upphandling och implementation. Funktionen är också en viktig förutsättning för att Enhetshanteringen ska fungera på det sätt som arkitekturen beskriver.

Möjligheten att kunna skicka information till de uppkopplade enheterna genom konfigurationer eller kommandon hanteras av förmågan **styrning**. För styrning av enheter återanvänds de funktioner som samla in förmågan beskriver för att kunna kommunicera med enheterna. Samla in förmågan är mottagningspunkten för meddelanden från andra förmågor i plattformen som Analysera och Enhetshanteringen för att koppla upp och skicka meddelanden till enheterna. Viktiga egenskaper för förmågan är att kunna hantera många meddelanden och att kunna skicka dessa meddelanden till enstaka enheter men även kunna

”broadcasta” meddelanden dvs., kunna skicka ett/samma meddelande till flera enheter samtidigt (ett - till många).

Dataprotokoll

Vilka dataprotokoll som IoT-gatewayen behöver hantera kommer förändras med tiden beroende på hur standardisering¹⁰ runt enheter och datakommunikation utvecklas. Därför är det viktigt att funktionen är både tillräckligt flexibel för att kunna utökas med specifika dataprotokoll samtidigt som den innehåller och uppdateras med de standarder som är vanligast förekommande. Detta bidrar till en kostnadseffektiv lösning som stödjer större delen av de verksamhetsspecifika enheterna.

Datakommunikationen måste kunna vara dubbelriktad för att både kunna ta emot och skicka meddelanden till enheter samt stöd för enhetshantering. Idag är de dataprotokoll som används mest i generiska IoT-lösningar som ska finnas med i en central IoT-gateways är [MQTT](#) och REST. Önskvärt är att protokoll som [AMQP](#), [CoAP](#), [Websocket](#) och [XMPP](#) är möjliga att användas men skulle kunna implementeras i Edge-gateway istället.

Vissa verksamhetsspecifika tillämpningar inom IoT har egna protokoll som t.ex. energimarknaden och fastighetssystem. Om en sådan tillämpning inte har Edge kommunikation enligt mönster 2-4 i [Infrastruktur och kommunikation](#) sektionen bör det vara möjligt att utöka den centrala IoT-gatewayen med detta dataprotokoll. Förmågor för andra mera generiska dataprotokoll som REST, HPPT, SOAP, FTP bör hanteras via befintlig integrationsplattform som i sin tur kan kopplas in till andra förmågor i Plattformen.

Autentisering av enheterna ska ske vid anslutningen till IoT-gatewayen och hanteringen av den krypterade informationen ska implementeras här. Krypteringen bör i första hand ske via certifikat mellan enheterna och Edge-gateway men andra teknologier för kryptering kan vara möjliga om de bedöms som mera effektiva.

Kommunikationsprotokoll

För kommunikationen med enheterna behövs också en infrastruktur för kommunikationen med enheterna. Grundstommen är att enheterna är uppkopplade på Stadens nätverk och att förmågan från detta nätverk har åtkomst till enheterna. Arkitekturen beskriver här

¹⁰ Stockholm Stad, ”Strategi för Stockholm som smart och uppkopplad stad”, sid 21, 2017

endast plattformens förmågor där infrastrukturen för kommunikationen förutsätts vara inom Stadens nätverk.

Meddelandehantering

Hur förmågorna i Plattformen fungera tillsammans är via Meddelandehantering. Meddelandehantering är en viktig funktion för flera förmågor i plattformen och dess primära ansvar är att dela och förmedla data vidare inom arkitekturen oberoende av källa, format och användningsområde. Det innebär att ta emot meddelandeströmmarna från olika källor och IoT-gateways samt sända vidare meddelande till de rätta källorna inom plattformen.

I Plattformen skapas dataströmmar som konsumeras av de förmågor som ska hantera datameddelandet. En dataström är en data- och eller händelseström som innehåller en serie av meddelanden. Meddelandena är fördefinierade av IoT-gatewayen för att hamna på rätt dataström och att resten av arkitekturen kan konsumera detta data. Det måste t.ex. finnas en identifiering på varje meddelande för att de gemensamma förmågorna ska kunna avgöra vilket ursprungsmeddelandet var. I dataströmmarna finns det möjlighet att utnyttja transformering, bearbetning och analysförmågor i arkitekturen för att kunna agera på meddelanden i realtid. Detta kan utnyttjas t.ex. för att finna avvikande data som kan skapa larm eller andra händelser från enheten som behöver hanteras.

Generellt sorteras inkommande meddelanden på rätt kö och persisterar (lagrar) meddelanden i kön under en konfigurerad tid alternativt tills samtliga prenumeranter har läst/konsumerat meddelandet. Grundtanken bakom denna kö är att den är baserad på ett prenumerationsförfarande i en händelsestyrd arkitektur, vilket innebär att när meddelandet kommer in läggs det på rätt kö och detta meddelande tillgängliggörs sedan till prenumeranterna som är de andra förmågorna i Plattformen.

Arkitekturella mönster för hur detta realiseras kan vara Kafka eller Lambda referensarkitekturer lite beroende på hur övriga förmågor i arkitekturen delas upp. Kafka är en optimerad variant av Lambda som saknar ”batch” möjligheten och robustheten att kunna sända om data som inte kunde processas. Lambda är därmed något mera resurskrävande men stabilare i hanteringen av lagring/big data.

Viktiga egenskaper från båda mönstren är robustheten, feltoleransen och möjligheten att kunna skala ut snabbt. Kafka mönstret kan vara att föredra om/när machine learning finns etablerad då den bättre stödjer möjligheten att utnyttja detta i realtid.

6.4 Hantera data och meddelande

När informationen har passerat mottagningsdelen tar det funktionella lagret över för transformering, bearbetning, analys, beräkning.

Transformering

Transformera strömmande data i plattformen till förbestämda modeller för att möjliggöra och optimera användningen av data av övriga förmågor inom Plattformen. Att transformera insamlat data till en hanterbar struktur ökar värdet och möjligheten att använda data för plattformen. Exempel på transformering kan t.ex. vara att kunna ta emot samma information från flera olika typer av enheter i olika format och transformera dessa till samma modell för att effektivare kunna utnyttja analysmodeller, utvecklingsverktyg, testning och lagring.

Transformeringen ska ta bort en del av komplexiteten för bearbetning och analys. Det kommer att krävas flera olika modeller att både kunna transformera data från och till men plattformen ska i så stor utsträckning som möjligt hålla sig till standarder för de verksamheter sådana finns. Transformeringen kan också förbereda data genom normalisering och annan formatering för att kunna användas av Machine learning och direkt kunna delas via tillgängliggörande i en enhetlig modell.

Exempel på datamodeller kan vara FIWARE, TALQ, OSCP eller andra av Staden använda standardmodeller per domän

Bearbetning

Förmågan att bearbeta innefattar att kunna ta emot meddelanden och göra intelligenta beslut om var detta meddelande ska ta vägen närmast, t.ex. om något måste hanteras omgående eller om data endast ska lagras eller båda. I bearbetningen kan viss transformering av Dataströmmarna vara löst kopplade från varandra via kanaler för att kunna utnyttjas av samtliga förmågor oberoende av varandra. Det data som hanteras i plattformen behöver normaliseras, tvättas och kategoriseras. Detta sker genom olika typer av underförmågor som:

Aggregering: Möjlighet att kunna kombinera data från enheter till hanterbar information som t.ex. medelvärde över tid eller medelvärde för flera sensorer.

Kvalitetssäkring: genom att kunna använda t.ex. gränsvärden kan viss kvalitetssäkring göras genom att kunna ta bort brus i

dataströmmarna, dvs. värden från trasiga eller felaktiga sensorer inte används.

Skapa händelser: Genom bearbetning av data i dataströmmarna kan nytt data uppstå genom händelser. Detta kan vara t.ex. om felaktigt data skickas från en enhet under längre tid kan en händelse skapas för att larma/felanmäla enheten

Analys

En stor och viktig förmåga att kunna analysera det rörliga data. Plattformens analysförmåga ska komplettera de verksamhetsanalysförmågor som redan finns i Staden med fokus på realtidsanalyser.

Typiska frågor som analysen i IoT-plattformens ska kunna svara på är ”Vad kommer sannolikt att hända” eller ”Vad bör vi nu göra”, dvs. prediktiv analys är en viktig förmåga för att kunna.

För att kunna detta krävs:

- Förmågan att analysera sekvenser av strömmande data
- Förmåga att kunna dra nytta av flera datakällor i realtid

Det är viktigt att kunna hantera allt data även som man i förväg inte visste värdet i data vilket betyder att allt data ska inte vara aggregerat (förlust av detaljinformation).

Dra slutsatser: Via denna analys eller med hjälp av Machine Learning kunna dra slutsatser och skapa nya meddelanden antingen tillbaka till enheterna eller ut till resten av enterprise och verksamheter.

Kvalitetssäkring: Möjlighet via analysförmåga avgöra om data håller den kvalitet som förväntas för den enheten. Denna förmåga behöver i första hand ligga i Edge eftersom där finns ”bäst kunskap” om hur enheten fungerar men det finns tillfällen när denna funktion behövs även av centrala förmågor speciellt för möjligheten att kunna använda enheter som producerar mera rådata och inte har så hög kapacitet i Edge.

Förfining: Förbättra data med hjälp av metadata och referensdata för att ytterligare öka kvaliteten på informationen

Berika

Berika möjliggör att med hjälp av metadata och masterdata från andra källor eller från enhetshanteringen kunna skapa en kontext till sensordata.

För att effektivt kunna utnyttja Analysförmågan i IoT-plattformen krävs någon form av berikning för att beskriva i vilket sammanhang data har uppstått.

Typiska berikande funktioner är:

- Addera data om enhetens placering, temperatur eller status
- Geografiskt data som typ av omgivning (park, gata, sjö)
- Väderdata
- Enhetskonfiguration (mätnoggrannhet, enhet, periodicitet)

6.5 Lagring

IoT-lösningar genererar stora mängder av data som är beroende av mängden enheter som är uppkopplade och frekvensen på data som överförs och storleken på dessa meddelanden. I arkitekturen måste det kunna styras hur och var data ska lagras. All hantering av data i IoT-arkitekturen sker inte endast på realtidsdata utan har behov att använda sig av lagrat data som kan komma från lagringsförmågan eller från andra datalager för att kunna utföra analys eller bygga upp regler för förfining och kvalitetssäkring.

Data kommer att behöva lagras för att kunna hantera tidsserier och för att senare kunna visualisera och rapportera om data. Data delas här in i ”snabbt” och ”långsamt” data, där behoven för snabbt data är omedelbar åtkomst utan latency för realtidshantering som trend- och avvikelsehantering samt långsamt data för historik, data warehouse och för att t.ex. lära upp Machine Learning algoritmer.

Lagringen kan ske på olika sätt beroende på hur data är strukturerat och senare ska användas t.ex. tekniker som Kafka, NoSQL och Blob. Andra teknologier som t.ex. Hadoop kan bli nödvändigt för att effektivt kunna exportera stora datamängder ut ur plattformen beroende på användningsfallet.

Tillgängliggöra

I Plattformen skapas verksamhets API:er för de dataströmmar som ska tillgängliggöras för resten av enterprise som t.ex. verksamhetssystem, verksamhetsprocesser och verksamhetsanalys. Dataströmmarna tillgängliggörs via förmågan att skapa API:er i Plattformen som är den viktigaste förmågan för att kunna dra nytta av informationen som uppstår i plattformen.

API:erna delas Staden via Integrationsplattformens API-gateway för att kunna användas och integreras i resten av it-landskapet t.ex. via öppna och dela data portaler, analysplattformar eller integrationer direkt till verksamhetssystem.

6.6 Dataarkitektur

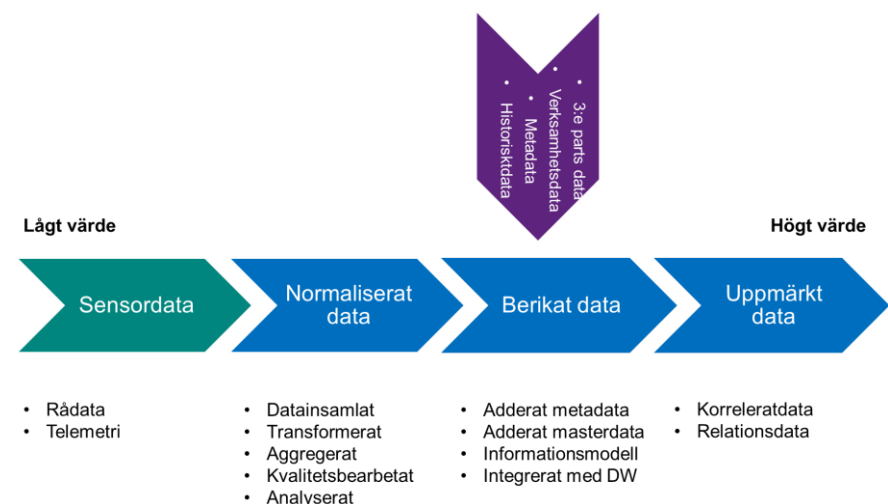
Principer för hur en dataarkitektur ska uppnås med hjälp av ovan beskrivna förmågor sammanfattas i detta kapitel. Avsikten är att förtydliga vilka principiella mål som ska kunna uppnås med arkitekturen och dess förmågor.

IoT-dataarkitekturen utgår från följande värdeprinciper som arkitekturen ska bidra till att uppfylla:

- Värdet på data är högre om kvaliteten är hög
- Värdet på data minskar med tiden
- Värdet på data är högre ju mer det används
- Värdet på data ökar om den integreras med annat data

Logiskt dataflöde

Dataflödet genom arkitekturen generellt möjliggör värdeadderade funktioner på data i flödet. Det logiska flödet ser ut som följande:



De egenskaper som data i detta flöde kan kategoriseras i är: *Struktur, storlek, hastighet* och *källa* och beskrivs i olika detalj beroende på vilken typ av dataflöde som beskrivs.

Strukturtyper

Rådata saknar modell eller inte kan organiseras på ett sådant sätt att det i förväg kan struktureras. Exempel på detta data är textbaserat innehåll, sociala medier, anteckningar.

Delstrukturerat data saknar en strikt datamodell för allt data i information. Vissa attribut är förbestämda, sedan kan ytterligare information vara inkluderat i datat. Exempel på detta kan vara foton som har meta-taggar som beskriver egenskaper för fotografiet medan fotografiet i sig är en mängd ostrukturerat data.

Strukturerat data är data som följer ett fördefinierat schema/modell som beskriver vilka olika typer av dataelement som kommer att vara grupperade tillsammans vid lagring. Fördelen med strukturerat data är att den är enkelt sökbar och analyserbar

Storlek

Mängden data avgör ofta hur den kan hanteras. Små mängder data kan med fördel hanteras i relationsdatabaser eller andra strukturerade datahanteringsförmågor. Big-data, stora datamängder kan kräva stora resurser i form av processorkraft och minne för att kunna hanteras och användas. Var gränsen går för Big-data är beroende av behov och teknologi.

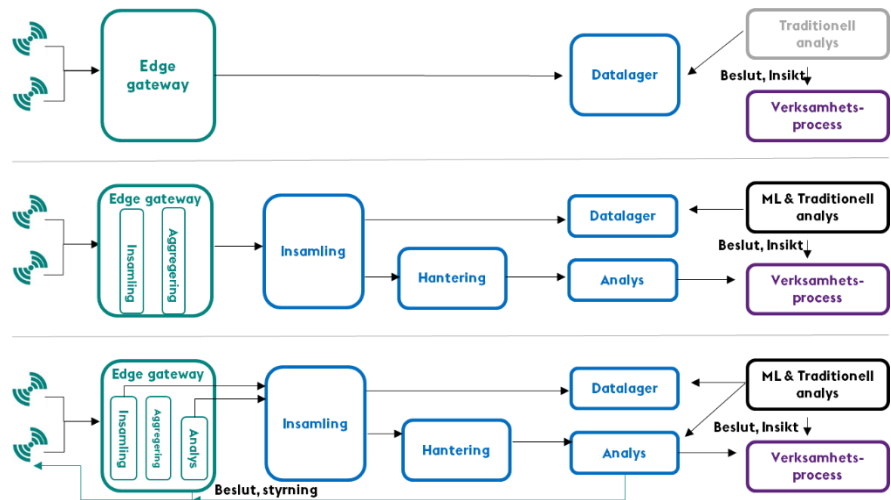
Hastighet

Traditionellt är data långsamt och i vila. Det betyder att data laddas in i databaser, efter att händelsen har inträffat, och struktureras där och hamna i vila. Detta data kan sedan analyseras och användas i efterhand.

För IoT kan mycket av data behöva användas snabbt och då data är i rörelse. Därför behöver både tidsserier analyseras, dvs. när värden ändras över tid i jämförelse med de närmaste händelserna eller om värden passerar brytpunkter skapas informationen. Detta data ”hinner” aldrig till vila/lagring förrän det behöver användas utan måste kunna styras hur resultaten av denna realtidsanalys ska kunna styras vidare inom arkitekturen.

Löst kopplade dataströmmar

Dataströmmarna till och genom Plattformen måste vara löst kopplade från varandra för att data ska kunna utnyttjas av samtliga förmågor inom plattformen. Insamling av data ska kunna vara händelsestyrd vilket innebär att förändringar i data skapar meddelanden som övriga förmågor kan agera och lyssna på. Den lösa kopplingen gör möjlighet att kunna skapa flera olika mönster för hur analys och återkoppling kan ske antingen tillbaka till enheterna eller till enterprise.



Figur 12 - Huvudmönster för hur och var analys och beräkning av IoT-data kan ske

I det första mönstret ovan används traditionell insamling av data för analys i efterhand. Resultatet av denna analys används som rapporter etc. för att förändra regelverk eller processer i verksamheten. I det andra mönstret används IoT-förmågorna mera där insamlingen sker dels för att stödja det första mönstret samt för att kunna användas realtidsdata (snabbt data) direkt in i verksamhetsprocesserna, kanske som händelser. Den traditionella analysen kopplar också tillbaka direkt på dataströmmarna i IoT-plattformen. I det tredje mönstret sker återkopplingarna och analysen i flera delar av dataflödet för att kunna ge effekter så tidigt som möjligt. Återkoppling och analys sker på flera delar i arkitekturen både i Edge, Plattform & Verksamhetssystemen. Samverkan för hur resultaten av insikterna kan användas är viktigt för att kunna skapa en dynamisk modell för hantering av realtidsdata.

6.7 Enhetshantering

Enhetshanteringen beskrivs här som en separat förmåga i arkitekturen då den i sig har flera delförmågor som beskrivs i mera detalj. Denna förmåga utnyttjar också flera av de förmågor som tidigare finns beskrivna i arkitekturen, då främst Insamlingsförmågan i IoT-gateway.

Enhetshantering har två huvuduppgifter:

- hålla rätt på samtliga enheter uppkopplade mot plattformen
- övervaka och hantera underhållet av enheter i Stadens kontroll

Till dessa huvuduppgifter hör att kunna få

- Avvikelsehantering om något inte fungerar som påverkar verksamhetsprocessen
- Hantera den digitala kopian av enheten

Det finns verksamheter vars uppgift är att ansvara för att de faktiska enheterna fungerar som de ska, t.ex. Trafikkontoret som ansvarar för att samtliga enheter i en trafik korsning fungerar. För dessa verksamheter blir enhetshanteringen en del av den faktiska verksamhetsprocessen då dessa enheter har aktuatorer som interagerar med omvärlden.

Det krävs egentligen inga ytterligare system förutom enhetshantering för att utföra den verksamhetsfunktionen att övervaka och hantera enheterna. Dock kan den behöva integreras med ytterligare system för att t.ex. kunna automatisera larmhantering och underhållsplanering. Önskvärt är att det data som produceras av enhetshanteringen kan användas för t.ex. prediktivt underhåll av enheterna och underlag för upphandlingar av likande enheter i framtiden. Standarder inom området finns för specifika områden, men även en mera generell standard som LightweightM2M (LWM2M) protokollet som allt fler leverantörer verkar ansluta sig till.

De underförmågor som Enhetshanteringen behöver kunna hantera är:

Autentisering och auktorisering

En enhet måste identifieras och verifieras av enhetshanteraren innan den kan anslutas för dataöverföring och enhetens data kan tas emot av plattformen. En säker identifiering är nödvändigt för att veta vem som anslutit enheten. Antigen sker identifieringen med certifikat (t.ex. X.509) utställda på Staden för att verifiera anslutningen. Det finns även sätt att i nätverksinfrastrukturen säkerställa att anslutna enheter kan verifieras.

Provisionering

Enhetens egenskaper registreras i ett centralt enhetsregister i enhetshanteraren när enheten har autentiserats. Önskvärt är att denna registrering kan kompletteras med manuellt data för att registrera egenskaper på enheten som inte klaras av att göras automatiskt. För denna registrering bör det finnas API:er för att genomföra den manuella kompletteringen där det är lämpligt. Om enheterna inte är direkt uppkopplade mot plattformen utan via en Edge plattform måste denna kunna provisionera och identifiera

varje enhet för sig in till enhetshanteraren. Denna identifiering måste sedan Edge-gatewayen behålla för att plattformen ska kunna kommunicera med enheten.

Konfiguration och kontroll

Enheter ska kunna automatiskt konfigureras vid uppstart när enheten är identifierad i Enhetshanteraren.

En konfiguration av en enhet kan vara t.ex. hur ofta ska data ska skickas, format på data, dataprotokoll, noggrannhet. Detta kan vara effektivare än att låta detta göras av tillverkaren av enheterna vid leveransen (pre-konfigurering), eftersom det kan tänkas att man vill ändra konfigurationen i framtiden för alla nya enheter av denna typ. Då får enheterna en ny konfiguration utan att de behöver hanteras fysiskt/manuellt för varje enhet.

Övervakning och monitorering via kontrollpanel (eng. dashboard)

Visualisering med filtreringsmöjligheter på enheternas egenskaper t.ex. typer av enheter, placering är nödvändigt. I kontrollpanel ska det även vara möjligt att kontrollera enhetens och kommunikationens status, ladda ner loggar från Edge, möjlighet att stänga av en enhet.

Mjukvaruuppdatering och hantering

Enheternas mjukvaror måste kunna uppdateras centralt från enhetshanteraren. En enhets mjukvaror bör bestå av flera separata delar som: firmware, operativsystem, kontrollprogramvara, säkerhetsprogramvara. Uppdateringar måste kunna ske i bulk och regelbaserat baserat på enheternas registrerade egenskaper (för att t.ex. kunna göra A/B tester).

Gemensamma enhetsegenskaper

En rad egenskaper attribut bör alltid kunna registreras för samtliga enheter. Dessa bör i detalj bestämmas när enhetshanteraren realiseras där grundläggande egenskaperna bör innehålla

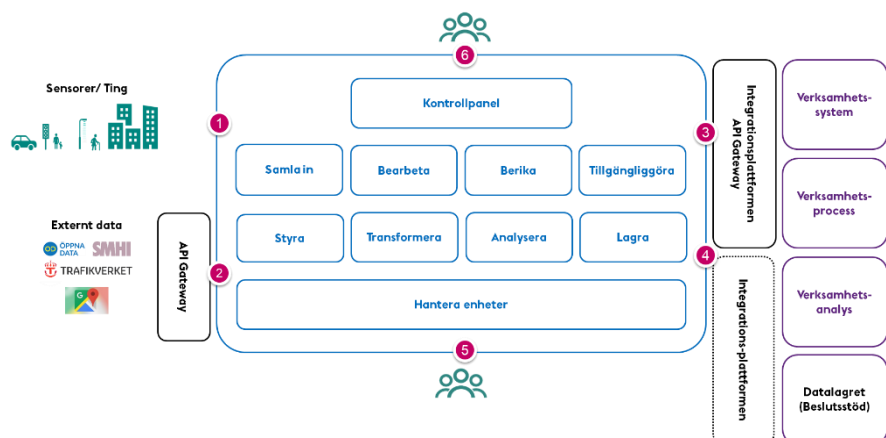
- **Identifiering**; enheten måste kunna ha en unik identifiering och kunna beskriva vad en utför
- **Rapportering** (provisionering)
 - Kunna skicka koordinater var enheten är placerad vid varje omstart/förflyttning.
 - Version av firmware, operativsystem och styrprogramvara
 - Kommunikationssätt

- Sensortyp och sensorkategori
- Modellbeteckning
- Tillverkare
- **Övervakning**
 - Status; Kunna skicka statusuppdateringar hur enheten fungerar, viken status den har
 - Batteri; Vid batteridrift kunna rapportera batterinivå
 - Automatisk felrapportering

6.8 Förhållande till övriga plattformar i staden

Stadens arkitektur består av flera olika digitala plattformar¹¹ som samverkar i digitaliseringen.

Varje plattform har sina egna unika sätt att implementera de förmågor som krävs för att uppfylla behoven från verksamheternas digitaliseringsbehov. Flera av förmågorna överlappar i syfte men när de samverkar uppstår nya möjligheter till insikt och åtgärder. Strategin har uttalat mål att återanvända och inte dubblera funktioner och nedanstående sammanfattning försöker belysa att funktionellt är det inte en dubbling av funktionerna även om det är samma förmågor som beskrivs som finns i andra plattformar i Staden.



Figur 13 - IoT-plattformens förhållande till övriga delar i IT-arkitekturen

Av Plattformens förmågor har endast två direkt interaktion med användare/människor; **Hantera enheter**, som kan ha användargränssnitt kopplade till sig för att övervaka och hantera enheterna och **Kontrollpanel** som är den förmåga att visualisera de dataströmmar och de analyser som plattformen utför i realtid. Här

¹¹ Stockholm Stad, ”Riktlinje för stadens it-infrastruktur”, Ett program för digital förnyelse dnr 031-760/2013

finns det användargränssnitt (färdiga eller utvecklingsbara) för förvaltare, verksamhetsutvecklare och programmerare att se och visualisera vad plattformen hanterar och bearbetar.

Säkerhet är exkluderad i denna bild då det är en förmåga som spänner över samtliga förmågor och är kopplad till den för stadens gemensamma säkerhetsarkitektur.

1 Hantera data från Edge

Inom IoT finns det speciella egenskaper för dataflöden som normalt Integrationsplattformen inte har stöd för. Dessa nya dataflöden möjliggörs via insamlingsförmågan i Plattformen. Bearbetning som sker i Plattformen kan återkopplas tillbaka till Edge. Integrationen sker då via Plattformen genom återanvändning av de protokoll för kommunikation med Edge som finns etablerade.

2 Inhämta data från externa datakällor

Om externa datakällor stödjer de protokoll och mekanismer som API-gateway i Integrationsplattformen stödjer tas detta data emot till Plattformen via API-gateway.

Nya verksamhetsscenariorna som det saknas stöd för idag i Integrationsplattformen:

- När externa datakällor kräver tekniker som inte stöds av Integrationsplattformen t.ex. Kafka eller Pub/Sub bör dessa kompletteras i Integrationsplattformen över tid alternativt återanvända motsvarande mekanismer i IoT-plattformen
- Vid integration inom en verksamhet i Staden kan integration direkt från datakällan till IoT-plattformen användas om det är lämpligt enligt arkitekturen för lösningen ur effektivitet, förvaltningsbarhet och kostnad

3 Tillgängliggöra data

Plattformen skapar verksamhets API:er för de dataströmmar som ska tillgängliggöras för att användas i verksamheten. API-gatewayen i Integrationsplattformen använder IoT-plattformen för att tillgängliggöra verksamhets API:er som skapats. Samma princip gäller för Plattformens som övrig API:er integration mellan system i Staden, att de funktioner och teknologier, som t.ex. säkerhet, biljetthantering och virtualisering ska skötas av API-gatewayen. Plattformens måste kunna publicera och tillhandahålla en API endpoint till API-gatewayen som kan tillgängliggöras där. De API:er som skapas och tillgängliggörs via Plattformen följer de standarder/metoder som erbjuds i Integrationsplattformen.

Detta görs enligt Integrationsplattformens princip ”*Vid extern kommunikation eller vid kommunikation mellan bolag ska stadens API-gateway användas*”.

API-gatewayen kompletterar Plattformens API:er med framförallt:

- Virtualisering och publicering
- Övervaka användningen
- Hantera behörigheter
- Tilldela resurser och åtkomst

Den teknologi som Integrationsplattformen har för detta idag är produkten Sentinel som stödjer förutom ovanstående funktioner, tekniska formaten REST och SOAP för API:er.

Om data behöver tillgängliggöras i stora mängder för t.ex. laddning till annan analys/bearbetning plattform eller lagring på annan plats än i Plattformen kan API-gatewayen vara begränsad eller dyr.

Alternativa teknologier till att tillgängliggöra data från IoT-plattformen kan vara

- Kafka
- Pub/Sub
- Filöverföring
- ETL, där framförallt Extrakt förmågan

Plattformen ska vara förberedd för att agera klient för sådana scenarion. Eventuellt behöver Integrationsplattformen kompletteras för att ansvara för Kafka och Pub/sub förfarande i framtiden.

5, 6 Användare direkt i Plattformen

Vissa användargrupper behöver ha direkt tillgång via någon sort av användargränssnitt till Plattformen utan att använda publicerade API:er. Denna funktion kallas här för Kontrollpanel och kan innehålla:

Visualisering

Förvaltare och informationsägare av informationsflödena genom plattformen har behov av att visualisera dataflödet för att få en bättre förståelse vad som sker. Användbara situationer är bland annat vid felsökning och vidareutveckling i hanteringen av informationsströmmen genom förmågorna.

Enhetshantering

Användare med ansvar för de uppkopplade enheterna behöver ha tillgång till enhetsinformation och

enhetsstatus, se *Enhetshantering i målarkitektur*. Detta sker genom tillgång till Enhetshanteringens användargränssnitt i Plattformen.

Begrepp och förkortningar

Begrepp SV	Begrepp ENG	Definition
AI	AI	Se Artificiell intelligens
Aggregator	Aggregator	Sammanställer data från flera olika datakällor genom grupperingar, filtreringar, summeringar eller likande metoder till en ny datakälla,
Aktuator	Actuator	Ställdon, även kallad aktuator, en fysisk sak som används för att styra en mekanism eller ett mekaniskt system
Analys	Analyze	Genom analys kan man omvandla data till insikter som har betydelse för att förstå verksamhetens användning av data
API (Application Programming Interface)	API	Application programming interface ("API") är en specifikation av hur applikationer och tjänster kan kommunicera med system eller plattformar. API:er är en viktig del i integrationsplattformar eftersom de används för att "översätta" kommunikationen mellan system och applikationer som annars inte hade kunnat kommunicera med varandra. Idag kan API:er definiera hela verksamhetstjänster, vilket gör att man kan se varje API som ett "eget system" som kan utföra en tydlig definierad men avgränsad verksamhetsfunktion.
API-gateway		En teknologi i Integrationsplattformen för att virtualisera API:er för att användas av andra centrala

		plattformar eller verksamhetssystem i Staden
Artificiell intelligens	Artificial Intelligence	Artificiell intelligens är intelligens hos maskiner och programvara som simulerar mänsklig intelligens. Det kan även beskrivas som system som är medvetna om sin omgivning och vidtar åtgärder för att anpassa sig efter den för att lyckas med sin uppgift.
Central förmåga		En it-förmåga som definierar vad som kan utföras utan koppling till teknologin som implementera den. Att den är central är att den uttrycks på ett generiskt sätt som gör den möjlig att användas av alla verksamheter centralt
Central plattform		
Certifikatbaserad kommunikation		Certifikatbaserad kommunikation baseras på olika former av PKI, Public Key Infrastructure. Certifikat fyller en funktion som gör det möjligt att två kommunicerade datorer kan lita på varandra under hela certifikatets livslängd. De behöver inte återautentisera varandra för varje meddelande utan har väsentligt längre giltighetstid.
Datainsamling		Se Fel! Hittar inte referensälla.
Datalagring	Data storage	En teknologi bestående av olika datakomponenter som används för att bevara digitalt data.
Dataström	Data stream	Dataström är data som genereras kontinuerligt av olika källor. Denna typ av data bör hanteras inkrementellt/direkt med hjälp av meddelande- eller

		strömhantering utan att behöva hantera det faktiska data i strömmen. Det kan alltså vara så att egenskaperna i data i strömmen kan förändras över tiden utan att dataströmmen i sig förändras.
Digital plats		Ett digitalt verktyg (applikation, webbplats, högtalare, skärm etc.) där människor kan interagera med data, tjänster och visualiseringar av data.
Digital tvilling	Digital Twin	Se 4.2, digital enhet
Edge		Ett engelskt uttryck för ytterkant, vilket här avser de platser i Staden (inom- och utomhus) där data kan insamlas och hanteras utanför stadens centrala plattformar eller verksamhetssystem.
Edge computing		Ett engelskt uttryck för den teknik som ger minne och datorkraft nära den fysiska plats där behovet finns.
Edge-gateway	Edge Gateway	Se 5.5
Edge-programvara	Edge Computing	Term för den teknik som ger minne och datorkraft närmare den fysiska plats där behovet finns
Enhet		En fysisk sak eller ett system som kommunicerar med IoT-plattformen. Exempelvis sensorer, aktuatorer och Edge-gateways.
Enhetshantering		Enhetshantering har två generiska funktioner inom IoT: dels att kunna hålla rätt på samtliga uppkopplade enheter samt övervaka och hantera underhållet av de uppkopplade enheterna

Förmågor		Den kapacitet som Beställaren efterfrågar, exempelvis funktioner och egenskaper.
Het data	Hot Data	Het data är data som måste hanteras omgående efter att data har skapats. Realtidslösningar kräver generellt het data.
Integration		
Integrationsplattform		
IoT (Internet of Things)		Ett engelskt uttryck för sakernas internet. Ett nätverk av fysiska saker som innehåller teknologi för att kommunicera, känna och interagera med omgivningen.
IoT-gateway		En teknologi som kan förbehandla information från och till enheter (sensorer och aktuatorer) innan dessa kommunicerar med centrala plattformar eller verksamhetssystem.
	Latency	Tiden mellan att en händelse har händelsens inträffande och konsekvensen av händelsen. Tiden kan orsakas av olika saker på vägen mellan stimuli och utföraren av konsekvensen.
IoT-plattform		Teknik, system, applikationer (teknologier) som sammantaget hanterar den data som IoT producerar och konsumerar.
IoT-lösningar		Tekniska lösningar som omfattas av teknik inom ramen för IoT, exempelvis sensorer, aktuatorer, enheter eller IoT-plattform.
Latens	Latency	Fördröjning, väntetid: den tid som det av tekniska skäl tar att få fram en signal eller att hämta data; i synnerhet den

		fördröjning som är oundviklig på grund av fysikens lagar.
Lösningsarkitektur		Arkitekturen för ett enskilt verksamhetskrav/lösning.
Maskininlärning	Machine Learning	Machine Learning (Maskininlärning) handlar om studier och konstruktioner av algoritmer som kan lära sig av och göra förutsägelser gällande data. Maskininlärning utnyttjar stora mängder historiskt data för att lära förutsäga eller hitta nya insikter som sedan kan applicera denna kunskap med hjälp av de algoritmer som utnyttjas.
Meddelandehanterare		Meddelandehanterare är en specialiserad förmåga att granska mönster i de inkommande meddelanden för att avgöra hur dessa ska kunna skickas vidare genom arkitekturen
Metadata		Data som beskriver, klassificerar och ger information om annat data som verksamhetsdata eller referensdata. Hjälper organisationen eller system att hitta och förstå sitt data.
Målarkitektur		En målarkitektur beskriver en ideal vision om det framtida systemlandskapet. En vision där funktioner och information har grupperats för att skapa gemensamma system med syfte att undvika dubbelfunktioner, silo-tänk samt att verksamhetensinformation har en naturlig placering.
Pub/Sub	Publish/Subscribe	Inom mjukvaruarkektur är publish–subscribe en meddelandehantering där

		sändaren av ett meddelande, kallad publishers, inte bestämmer specifikt vem som är mottagare, kallad subscribers, utan kategoriserar istället sitt meddelande utan att veta om mottagaren, om det finns några. På samma sätt så gör subscribers genom att registrera ett intresse i en kategori av meddelande och tar emot dessa utan att veta vem som är sändaren.
Realtidsdata		Se Het data
Referensarkitektur		En referensarkitektur kan ses som en spelplan för alla som vill samverka inom ett visst område. Spelplanen innehåller spelregler och principer och kan användas vid utveckling eller anskaffning av lokala it-system.
Sakernas Internet	Internet of Things (IoT)	Internet of Things (IoT) är ett samlingsbegrepp för fysiska objekt eller "saker" som kommunicerar digitalt med omvärlden för att skapa ett mervärde av något slag genom att utbyta data och information med exempelvis tillverkaren, operatören och/eller andra anslutna saker. I detta dokument används enhet för att beskriva både sensorer som samlar in data och aktuatorier som styr saker, exempelvis en lampa.
Sensor	Sensor	Ett samlingsbegrepp på en fysisk sak som insamlar, konverterar och distribuerar någon form av signal, stimuli eller data från omvärlden.
Smart stad	Smart City	En smart stad definieras här i enlighet med definitionen i strategin för Stockholm som smart och uppkopplad stad.

		På en övergripande nivå beskrivs det som ”En stad som utnyttjar digitalisering och ny teknik för att göra livet enklare och bättre för invånare, företagare och besökare. I den smarta staden skapas kontinuerligt nya smarta tjänster som verkar för en innovativ stad. Den smarta staden möjliggörs genom uppkoppling och öppna data, integrerade plattformar, sensorer och annan teknik.”
SOA-plattformen		
Stora data	Big data	Big data är ett begrepp som beskriver digitalt lagrad information av sådan storlek att det är svårt att bearbeta den med traditionell teknik och traditionella databasmetoder. Ingen svensk översättning av begreppet har blivit etablerad. I många fall skapas datamängderna kontinuerligt (i realtid) och analyseras även i realtid. I och med den tekniska utvecklingen har möjligheterna för denna teknologi förändrats och big data används nu inom många olika områden, dock ännu inte i någon nämnvärd omfattning i Stockholms stad.
Strömmande data		Se Varm data
Styrprogramvara		Se Edge Programvara
Styrsystem		Någon slags programmerbar dator som kan styra enheter och utföra automation på en eller flera enheter
Stödjande data		Data som i verksamhetssystemet kombineras med

		verksamhetsdata, till exempel för att koppla data till ett kodvärde, listor över tillåtna värden etc. Inom ramen för denna upphandling syftar begreppet till stödjande data som leverantören äger, förfogar över eller använder sig av.
Ting	Thing	Se Enhet
Tjänsteorienterad arkitektur		
Varm data	Warm Data	Data konsumeras relativt snabb och har intressant värde nära händelsen. Normalt innehåller inte varm data historiskt data.
Verksamhetsprocess		En följd av aktiviteter i en organisation som leder till en produkt eller tjänst för medborgare, användare eller kunder.
Verksamhetssystem		Ett it-stöd som utför det operativa stödet till verksamhetsprocesserna.
Öppna data	Open data	Digital information som är fritt tillgänglig utan inskränkningar. Vad som är öppna data regleras i PSI-direktivet och Inspire-direktivet från EU som också införts i svensk lagstiftning. Det innebär att de offentliga data som samlas in och lagras hos myndigheter i elektronisk form också ska tillgängliggöras till företag och privatpersoner för återanvändning.