

# Arkitektur & Design

## Team Med Fel

2009-10-06

### Klient - Server

Vi kommer att använda oss av en "server-klient"-lösning i systemet. Att använda en central server innebär lättare datahantering än (peer-to-peer, P2P), eftersom all data passerar genom samma nod. Eftersom servern är central i systemet är det viktigt att den är tillgänglig så mycket som möjligt. Vi vill maximera dess tillgänglighet och samtidigt hindra obehöriga att komma åt den.

### Peer-to-peer

I vissa delar av systemet kommer vi att använda oss av P2P. Detta gör vi för att avlasta servern från en del av trafikmängden i nätverket. Video- & röstsamtal kommer använda P2P; mer om detta nedan.

### Klienten

---

Programvaran på klienten kommer att vara uppbyggd av ett antal olika beståndsdelar. Dessa kommer att realiseras som olika processer som kommunicerar med varandra via D-Bus.

### Karta & uppdragsvy

Det program som är mest synligt för slutanvändaren är den grafiska kartan och uppdragsvyn. Det är detta gränssnitt som kommer att ha hand om interaktionen mellan systemet och användaren.

### Video- & röstsamtal

Det andra synliga programmet är video- & röstsamtalsfunktionen. Det kommer att upprätta en direktanslutning mellan de två samtalande enheterna.

### Kö

På lite lägre nivå bakom den grafiska kommer vi att implementera ett kösystem. Meddelanden som skickas mellan klient och server kommer att prioriteras och mellanlagras i en kö. Inkommande meddelanden kommer också att lagras i en kö, under tiden de väntar på att bli behandlade. Kön kommer att skapa en viss redundans i systemet om nätverksuppkopplingen är dålig. När man har dåligt batteri så kommer kön att prioriteras så att bara viktiga meddelanden kommer fram.

### Quality of Service Manager

En Quality of Service Manager (QoS-Manager) kommer vara en process som kan styra tjänstekvaliteten beroende på faktorer som exempelvis batterinivå och nätverkssignal.

### GPS

Enheten har en GPS-enhet som kan användas för att avgöra koordinaterna för var enheten befinner sig. Med hjälp av denna kan enheten rapportera till servern var den finns, och alla enheter får en gemensam lägesbild.

## Lokal databas

Klienten ska ha en lokal databas som innehåller de objekt som klienten för närvarande är intresserad av. Detta innebär att klienten inte hela tiden måste använda nätverket för att komma åt information.

## Server

---

På servern ligger databaser och patientjournaler. Servern ska också ha ett filter som gör att andra aktörer som försvarsmakten och brandkåren inte kan läsa allt som finns på servern.

## Larm & uppdrag

---

Larm kan skickas in från vilken enhet som helst till servern. Servern skickar vidare larmet till en ledningsenhet som tolkar detta och skapar ett uppdrag. Uppdraget kan sedan tilldelas en viss (eller flera) enheter. Ledningenheten skickar uppdraget till servern och servern vidarebefodrar uppdraget vidare till alla berörda enheter.

Ett larm ska innehålla koordinater, prioritet, id och vad som har hänt. Ett uppdrag ska innehålla vem som ska göra vad och på vilken plats det ska utföras. Vi har delat på uppdrag och larm för att vem som helst ska kunna skicka ett larm, men bara de med tillräcklig kompetens ska kunna dela ut uppdrag. Ett larm kan dessutom delas upp i flera uppdrag.

## FAQ & databas

---

När en ambulansförare har en fråga och ingen läkare befinner sig i närheten som han kan fråga ska ambulansföraren kunna slå upp sin fråga i en databas. Vi ska implementera en FAQ (Frequently Asked Questions) där det finns svar på vanliga frågor. FAQ:n kommer man åt genom servern. Behöver ambulansföraren information från patientjournaler måste han få godkännande från en läkaren. Ambulansföraren frågar servern och servern frågar läkaren. Om läkaren ger sitt godkännande så får ambulansföraren de uppgifter han begärde från patientjournalerna, via servern. Anledningen till att vi har en FAQ är att ambulansförarna inte ska behöva störa läkarna i onödan.

## Video- & röstsamtal

---

Är frågan mer avancerad ska ambulansföraren kunna ringa eller upprätta ett videosamtal med en läkare. När ett samtal (röst och/eller video) ska upprättas kommer enheten först att fråga servern om hur man kan kontakta enheten som man vill kontakta. Servern skickar de efterfrågade uppgifterna och sedan upprättas samtalet mellan de två enheterna direkt. Vi är tvungna att gå via servern först eftersom enheten inte kan hålla koll på var alla andra enheter finns. När enheten har hittat den andra enheten så går det mycket fortare om de kan skicka informationen mellan enheterna direkt.

