# Availability

En af de største fordele ved at anvende skyen er at data of services er tilgængelige overalt. Hvis man vælger en større udbyder med mange fysiske servere, vil det at anvende skyen ofte øge availability, da en fejl på en server ikke vil betyde noget, da den underliggende VM executer automatisk vil flytte eksekveringen uden at brugeren oplever nogen reduktion i availability. Det er dog vigtigt at man vælger den rigtige aftale med cloud udbyderen. Samtidig vil de fleste cloud udbydere have redundante forbindelse til internettet, som sikre høj forbindelses-availability.

Det er også vigtigt at undersøge den underliggende arkitektur i skyen. Anvender de en form for caching eller et specielt API, der gør at den givne sky opfører sig anderledes end man normalt ville forvente. Er der I/O caching som gør at data der lægges op i skyen først er tilgængelig efter et stykke tid, eller måske et ikke-intuitivt fil API som gør det svært at interagere med?

# Modifiability

Hvis der er tale om at man lægger hele applikationen op i skyen (IaaS) så har det mere betydning hvordan man laver sin architectur end om det er i Skyen eller ej, men hvis man anvender PaaS eller SaaS så er man begrænset af det bibliotek (PaaS) eller API (SaaS) som cloud udbyderen stiller til rådighed.

# Performance

Hvis der er tale om et system som TM12 der altid tilgås udefra vil performance i form af netværkskommunikationstid ikke blive påvirket af at lægge det op i skyen, faktisk er der en vis sandsynlighed for at det kan blive hurtigere, da en større cloud udbyder sandsynligvis vil have en meget kraftig forbindelse direkte til et internet back-bone, og dermed kan tilbyde en kommunikationshastighed og ping-tid som det vil være meget dyrt at skulle opbygge selv, specielt hvis man vælger en (eller flere) udbydere i nærheden af hvor data skal bruges. F.eks. hvis det skal bruges i Danmark vil man vælge en i Danmark eller som minimum i Europa.

Da cloud systemer mere eller mindre altid arbejder virtuelt skal dette naturligvis tages i betragtning, men moderne VM-Ware tilbyder performance der kan konkurrerer med eksekvering uden VM, så dette er ikke et stort problem mere, og der er mere et spørgsmål om at vælge den rigtige mængde ressourcer.

# Security

Her kan der være vise problemer med at anvende skyen, da data ikke mere fysisk ligger på en server man selv kontrollere. Kryptering kan afhjælpe meget af dette, men juridisk kan der være krav til godkendelse for at man må opbevare personfølsomme oplysninger. Der bør altid laves en risikovurdering før data flyttes op i skyen.

# Testability

Testability i Skyen er generelt sværere end på en lokal fysisk server, da man ikke har kontrol over det underliggende virtuelle eksekveringsmiljø. Hvis der er tale om SaaS eller PaaS, bliver dette problem større, da den underliggende applikation eller library ikke er tilgængelig for white-box test, og er der et problem i denne kode skal man vente på at cloud leverandøren retter fejlen. Fordelen er dog at da koden anvendes af mange andre er der en mindre risiko for at der er alvorlige fejl i koden.

# Usability

Hvis der er tale om SaaS er det naturligvis vigtigt at vælge en cloud leverandør der kan præcentere et logisk og let anvendeligt interface. Samtidig får man foræret en del end user test, da andre kunder har brugt interfacet før. Samtidig kan det være en fordel at man anvender et look and feel som er velkendt i markedet, da genkendelighed er utrolig vigtigt for usability. Vi har alle prøvet at skifte bank og skulle bruge tid på at sætte sig ind i et nyt interface, eller indenfor programmering er der ikke noget mere irriterende end når producenter afviger fra defakto-standarden, som f.eks. .NET Array.Copy, som anvender ”source, destination” når alle andre siden C anvender ”destination, source”.

I TM12 kan det f.eks. være et stort problem hvis læger og ikke mindst brugere er blevet vandt til det eksisterende interface og så skal til at lære et nyt. Samtidig, hvis hundredvis af klienter ude hos borgerne er blevet kodet op mod et interface, kan det være kompliceret at omkoden dem til et nyt.

Hvis der er tale om PaaS er problemet sandsynligvis mindre, da det hovedsageligt er applikationen der bestemmer usability, og ikke skyen.

# Anomalities

AMI = Amazon Machine Image

<http://blog.coredumped.org/2012/02/help-my-cpu-cycles-are-being-stolen.html>