Analyse af den arktekturelle fremgangsmåde (architecturel approach) er et vigtigt skridt, som gør brug af både scenarier, arkitekturdiagrammer, patterns og en stor del erfaring. En arkitekt med erfaring indenfor det område som analysen omhandler (f.eks. databaseinteraktion, IPC, webservices, …) vil kunne bruge denne til at vurdere arkitekturens anvendelighed i forhold til de kvalitetsattributter og businness drivers der er fremsat. Arktitekten kan også trækker på andre arkitekters erfaring og kikke på architecurel patterns. Architectual patterns minder om design patterns, bare på et højere plan. Forskelle patterns har forskellige fordele og ulempler. Her er det også vigtigt at nævne anti-patterns, som en erfaren arkitekt også vil kende og derfor kunne undgå.

For at kunne foretage analysen kan man f.eks. anvende den template som Bass et al foreslår.

Vi har valgt at analysere følgende QAS fra H2:

*#13 If a TM server crashes, system should be able to recreate data and clients should be still able to get their measurements processed.*

Dette scenarie indeholder to uafhængige detaljer: 1. vi skal sikre at der er en server som klienten kan anvende og 2: vi skal sikre at det data der er sendt til serveren ikke går tabt. Vi har også valgt at beskrive analysen på engelsk da scenariet oprindeligt er skrevet på engelsk (og det er templaten også).

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Scenario #: 13 | | | | Scenar*io: If a TM server crashes, system should be able to recreate data and clients should be still able to get their measurements processed.* | | | |
| Attribute(s) | Availability  (Performance) | | | | | | |
| Environment | Normal operations | | | | | | |
| Stimulus | Fatal server crash. | | | | | | |
| Response | Client continues normal operations with a maximum extra delay of 30 seconds. | | | | | | |
| Architectural decisions | | | Sensitivity | | Tradeoff | Risk | Nonrisk |
| Backup server | | | S1 | |  | R1 |  |
| RAID implementation | | |  | |  |  | N1 |
| Remote backup | | | S3 | |  |  | N2 |
| Client data retention | | | S4 | | T1 |  |  |
| Reasoning | | Back-up server is continuously ready and client switch-over if no response within 20 seconds.  Back-up server run same application and operating system as main server (risk-1)  A RAID implementation will protect against hardware failure in the hard-drive.  The remote backup will perform an incremental back-up of all data every 24 hours.  All clients keep data in a local cache until server response allows deletion. This ensures that in the event of a destructive server crash the data can be retrieved from the clients. The clients are instructed to erase local data when data has been backed up. | | | | | |
| Architecture diagram | |  | | | | | |

|  |  |
| --- | --- |
| S1 | We could keep a routing server in front of the main server, which could ping the main server every n seconds and switch server if non-responsive, yet this may lower performance due to extra routing and will also increase complexity and price, and impose an extra load on the server from the heartbeat. |
| S3 | By increasing the frequency of the remote back-up we can limit the required amount of client caching, yet at the cost of putting more load on the server (reducing performance). |
| R1 | By using the same application on the same OS there is a risk of application failure also existing on the backup-failure. F.eks. a failure to accept leap-year would exist on both servers. Secondly an OS update crash (the operating system vendor may send out an update that makes the system fail) is a risk when same OS is used. The alternative is a parallel implementation on a different architecture (very expensive). |
| S4 | By using a simple timer in the client data may be removed earlier. |
| T1 | Keeping the latest client data on the client as a safety feature (backup) lowers security, as the client is now vulnerable to theft. This is however a limited risk and only the clients own data is at risk. Encryption may be employed to ensure confidentiality. |