

### Harjoitus 3

1. Yrityksen tuottaman erään sähköisen palvelun käytöstä kerätään jatkuvasti lokitietoja yhteen tiedostoon. Lokiin kirjautuu tietoja palvelupyynnöiden saapumisajoista, järjestelmän antamien vastausten toimitusajoista, ja mahdollisista poikkeustilanteista (jos pyyntöä ei voitu jostain syystä täyttää). Pyyntöjä on useampaa eri tyyppiä. Kerran vuorokaudessa kaikki kirjaukset sisältävästä lokitiedostosta koostetaan tiivistetty tekstimuotoinen raportti, joka näyttää yhteenvedon pyyntötyypeittäin jaoteltuna ja tunneittain viimeksi kuluneen 24 tunnin ajalta:

- pyyntöjen lukumäärä tunneittain
- poikkeusten lukumäärää tunneittain
- Pyyntöjen keskimääräinen käsittelyaika (keskiarvo ja mediaani) tunneittain

Tiedot kirjataan lokitiedostoon yksittäisinä tapahtumina tietuemuodossa (riveinä), joka koostuu vastaanottavan palvelimen tuottamasta palvelupyynnön tunnisteesta (request ID), kirjauksen tyyppistä (pyyntötyypin tunniste-koodi, vastaus, poikkeus) ja aikaleimasta (tapahtuma-aika, ei kirjausaika). Palvelupyynnön tunniste kytkee yhteen palvelupyynnön ja sen vastauksen/poikkeuksen. Lokiin kertyy kirjauksia vuorokauden aikana n. 100 000. Huomaa, että kirjausten järjestyksestä lokissa *ei voi* suoraan päätellä tapahtumien järjestystä. Voit kuitenkin olettaa, että request ID on aina yksikäsitteinen tarkasteltavan 24 tunnin aikajaksolla (lokitiedosto nollautuu 24 tunnin välein automaattisesti). Esimerkki lokitiedostosta:

ID		TYPE	TIME
13334287	REQA	1379680320947	
13334287	RESP_OK	1379680321026	
13334288	REQB	1379680321251	
13334390	REQA	1379680321260	
13334390	RESP_OK	1379680321290	
13334288	ERR_110	1379680321266	

Hahmottele arkkitehtuuri yhteenvetoraportin koostavalle ohjelmistoratkaisulle. On tärkeää, että ratkaisuun on helppo lisätä uusia raporttimuotoja (esim diagrammien tuottaminen koostetusta datasta) sekä tuki uusille pyyntötyypeille. Mitä luennoilla käsiteltyjä *arkkitehtuurityylejä* voisit soveltaa tämän järjestelmän arkkitehtuurissa?

2. Tutustu *Oracle Tuxedo* väliohjelmistoon (middleware) hajautettuja sovelluksia varten: [http://en.wikipedia.org/wiki/Tuxedo\\_%28software%29](http://en.wikipedia.org/wiki/Tuxedo_%28software%29)

Mitä luennoilla läpikäytyjä tyylejä/patterneja tunnistat?

3. Viestinvälitykseen perustuvia hajautettujen järjestelmien toteuttamisen mahdollistavia ohjelmistoja kutsutaan usein nimillä *Message Queue* tai *Message Oriented Middleware*. Viestien välityksen semantiikkaan liittyy monia huomioon otettavia seikkoja: katso listaus Wikipedian *Message queue* –artikkelista otsikon *Usage* alta ([http://en.wikipedia.org/wiki/Message\\_queue](http://en.wikipedia.org/wiki/Message_queue)).

Ota selvää, miten nämä asiat on ratkaistu Amazonin *Simple Queue Service* (<http://aws.amazon.com/sqs/>) –viestinvälityspalvelussa (et välttämättä löydä vastauksia aivan kaikkiin kysymyksiin palvelun dokumentaatiosta).

4. Tutustu *BitTorrent* -tiedostonjakoprotokollaan ja vastaa kysymyksiin. Lähteitä:  
<http://www.bittorrent.com/help/guides/beginners-guide>  
<http://en.wikipedia.org/wiki/BitTorrent>  
<http://www.youtube.com/watch?v=urzOeD7ftbl> (YouTube - "Torrents Explained: How BitTorrent Works" by raindrop264)
  - a) Mitä tarkoittavat käsitteet *Torrent*, *Torrent file*, *Swarm*, *Seed*, *Peer*, *Seeder*, *Leech*?
  - b) Miten Torrent-tiedoston avaava Torrent Client saa tietää, mistä varsinaisen sisältötiedoston osat löytyvät? Eli miten Client saa tietää torrentia jakavien toisten solmujen osoitteet ja niiden jakamat osat?