

- 1a Leg uit wat met *middleware* bedoeld wordt. 5pt
- 1b Waarom is het normaal gesproken zo lastig om traditionele client/server applicaties te schalen? 5pt
- 1c Bedenk enkele voorwaarden waaronder *server-initiated client-side caching* in het Web goed zou werken. 5pt
- 1d Sommige gedistribueerde filesystemen reiken *leases* uit aan clients voor het cachen van objecten. Waarom zijn *leases* soms handig? 5pt
-
- 2a Wat is het verschil tussen een *hard link* en een *soft link* in een *naming system*? 5pt
- 2b Als je alleen maar pure namen hebt als object referenties, d.w.z. namen zonder enige informatie, hoe kun je dan een object op efficiënte wijze lokaliseren in een wereldwijd gedistribueerd systeem? 10pt
- 2c Leg uit wat een *closure mechanisme* is, en waarom deze inherent impliciet zijn. 5pt
-
- 3a Geef een exacte omschrijving van betrouwbare multicasting. 5pt
- 3b Waarom is het zo moeilijk om betrouwbare multicasting te schalen naar grote aantallen ontvangers? 10pt
- 3c In *virtual synchrony* zal process $P[i]$ altijd $N[i][j]$ meezenden in een multicast bericht naar de andere processen. $N[i][j] = N$ betekent dat $P[i]$ alle berichten $\#1, \dots, \#N$ van process $P[j]$ ontvangen heeft. Een process $P[i]$ zal bericht $\#N$ van $P[j]$ pas *afleveren* als het weet dat bericht $\#N$ door iedereen ontvangen is. Welke globale ordening van afleveren brengt dit met zich mee. Leg je antwoord duidelijk uit. 10pt
-
- 4a Onder welke *failure semantics* heb je $2k + 1$ leden nodig om een k -fout tolerante service te implementeren? 5pt
- 4b Geef een voorbeeld waarbij het verbergen van een *omission failure* leidt tot een *performance failure*. 5pt
- 4c Leg uit hoe *quorum-based replication* werkt. 10pt
- 4d Wat is de relatie tussen *quorum-based replication* en *group failure masking*? 5pt

Cijferbepaling: Het cijfer wordt bepaald door de punten die per onderdeel behaald zijn, bij elkaar op te tellen (totaal maximaal 90 punten), en vervolgens daar 10 extra punten bij te tellen. Er zijn dus totaal 100 punten te behalen.