CS7NS6 Distributed Systems

2024-2025

Exercise 2

Implementing a Globally-accessible Distributed Traffic Service

Imagine a world in which all road-vehicle drivers are required to prebook every journey that they wish to make, in particular, so that no driver is allowed to start a journey without having received a notification that the requested journey is acceptable. The exercise is to design, implement and demonstrate a system whose primary purpose is to allow drivers to book/cancel journeys and that is intended to be globally-accessible.

Note that the focus is not on how the system decides whether to not a journey is acceptable (but, of course, your system should recognize the fact that some requests may not be compatible) or how the requirement that journeys have been booked before starting is enforced (but should support such enforcement).

Your design should focus on providing appropriate levels of performance, scalability, availability and reliability, considering that your game may need to grow to be used by millions of users throughout the world.

Note that there is no 'right' design for such a system. In particular, you should begin by specifying a reasonable set of requirements for the game taking into account the level of dependability that might be expected by users of this service. Your design should then focus on how the system (and any individual subcomponents that you identify) might be designed to satisfy the requirements that you have specified. Be careful not to over engineer the system since providing higher levels of dependability is usually detrimental to the cost of operating/using a system.

The design will be assessed on the reasonableness of the requirements specified and the appropriateness of the corresponding design choices.

In designing the system you should consider whether and where paradigms/techniques such as transactions, process replication, checkpointing/message logging, caching, load balancing, replication, and/or partitioning as well as other appropriate techniques might be used. It is expected than any reasonable design will use some but not all of these techniques. Depending on the techniques being used you will need to consider, for example, the required degree of isolation, the degree of replication, and/or the level of replica consistency that are appropriate. Be explicit about any assumptions you make about the underlying failure model and the faults to be tolerated by the system.

You should also consider the expected usage and failure patterns of the system and how they might influence the design, for example, are users collocated or widely distributed? do more users read data or write data? is data shared between multiple users or not? are some data items more frequently used (read and/or updated) than others? are failures expected to be frequent or rare?

You may use any programming language, middleware, and/or development methodology that is agreed by all members of the group as you see fit. In demonstrating your solution you will be required to demonstrate its operation in a variety of failure scenarios, so you should build a deployment framework that allows such scenarios to be tested and eventually demonstrated.

Interim deadline

Please submit a short report outlining the technical architecture of your service and the technologies that you are planning to use to build it before Monday the 10th of March 2025. The report should be signed by all members of the group.

Final deadline

Each group is required to present and demonstrate its solution at a time to be advised, and to submit a report on the group's solution by 5.00pm on the 4th of April 2025 as a single PDF file by email to vinny.cahill@tcd.ie with the subject line "Group <number> CS7NS6 Exercise 2 Report". Further details will be notified on Blackboard.