

# Elottery - availability

## Abstract

In this section there is a presentation of e-lottery availability concept. At first a model is presented and tasks which are performed by each part of it. Next paragraph discusses important values about availability concept and proves it reaches satisfactory level. Various points of presented model are also expanded to present other parts of model and clarify a few assumptions that were made. Last section discusses about verification of described functionality.

## 1 E-lottery website

As a user who wants to take the part in the e-lottery he/she uses a website under known address. One domain for the website is used. From that the access to e-lottery and other its features is possible.

## 2 Availability structure

To make possible for lottery to work we need a set of servers that will work and accept requests. Servers are located in different parts of the world and they do their job. There is a specialized construct that has a task of being the boss/leader for the others. We will call it 'main' for now. Its task is to control other servers if they do their job or have issues. It can also oversee the work and help to manage it. The 'main' is 'divided' into two parts where one part has to take care of first half of servers and the second part of the second half (similar to the left and right hemisphere of human brain). At this moment it is good to establish that qualified members shall divide the servers (first and second half - mentioned earlier) into proper sets that will allow non-problematic work. 'Main' can do some limited amount of requests too due to the fact of other responsibilities. The servers have knowledge of the neighboring servers, and of course the 'main' one. Neighboring servers are defined as follows: For each server pick a given number of servers and connect them with each other, the number of servers picked is to be chosen by qualified members, each server can have different number of neighboring ones, 'main' is neighbour for all by default. Connection is defined as a procedure where servers can divide the work between themselves, send the data or cooperate. It is good to mention that servers can perform on their own and be a neighbour only to 'main'. For a given interval servers are checked if they are working and executing properly by 'main'. Servers also send an impulse to neighbouring ones and see if there is everything fine with them, while checking also if they are able to talk with 'main'. In case of emergency appropriate steps are taken to get rid of problems or unwanted situations. Moreover it is good to mention the fact that communication between two servers and a server with 'main' is different and properly secured to avoid even more problems.

### 3 Proving

Now let us consider what do we have acquired and prove that presented structure makes it possible to maintain fully available system. For a moment we will also discuss a little about communication between servers and expand some points mentioned in previous paragraph. ‘Main’ construct is the core of the system that controls and guides other devices. At the same moment devices can cooperate to avoid being overworked which may result in some not wanted situations as for example: some users might not be available to be served by the server due to amount of work it needs to do. ‘Main’ oversees if everything is going as planned and in some cases can guide to server to work properly. Cooperating devices can exchange the data between themselves by using protocols proven to be secure. It is good to mention that each set of neighboring servers has different keys for communication between each other. This solution makes it possible to avoid a danger situation in which someone unauthorized would acquire the key used for encryption/decryption and could read all communication by himself/herself and what’s even worse make attempts to tamper with it. The keys are hold in secret location known only to authorized and trusted members of the company. For a given time interval servers are sending an impulse to the neighbors and report if they received an answer to ‘main’. After that a report is generated to assess the condition of working servers. If result is positive then servers are working properly. If results are negative then appropriate procedures are taken to make the problem’s impact as small as it is possible. In some cases we can establish a new connection between two servers, which are not neighbors by generating/giving a new mutual key for them. At this moment we have taken care of managing the work, making it possible for everyone to be served and monitoring the status of our network. Such solution makes it possible to reach availability of an acceptable level.

### 4 Verifying

In this section we will discuss the verification aspect of availability of elottery. By verification we mean the condition that it is possible for servers to work and process data without creating any losses. To verify availability neighboring servers send impulse to each other and check if they received the response. After given time interval they create a report and send it to ‘main’. The reports have an established pattern. Later ‘main’ analyse the report and according to the results takes proper action. Mesages sent to ‘main’ have priority value that makes it possible to respond immediately to the event if it is necessary. To verify the availability tests are also conducted. From time to time servers are being tested if they are capable to work or they need conservation time. In such a way the condition of servers is controlled and verified. Let us consider one more situation when something happens to the ‘main’ server. In such a case we have a set of substistute servers to maintain the role. This is also why the ‘main’ is divided into more parts than one because in the other case we would lose not only amount of information but also entire communication could be interrupted and in result it would follow to chaos.