~~1. Online text editor~~

Design and implement an online text editor. Each user of the editor can see its current state and can add or erase text. The cursors symbolizing the currently modified text should be visible to all users.

~~2. Online spreadsheet~~

Design and implement an online spreadsheet. Each user of the spreadsheet can see its current state and can modify cells. The cursors symbolizing the currently modified cell should be visible to all users.

~~3. Game “Mensch ärgere Dich nicht”~~

Implement game (see: [en.wikipedia.org/wiki/Mensch\_%C3%A4rgere\_Dich\_nicht](https://en.wikipedia.org/wiki/Mensch_%C3%A4rgere_Dich_nicht) for more details). The game can be played by 2, 3, or 4 people at the same time. The game starts when everyone willing to play the game is ready. In a situation where one of the players loses connection with the rest, the pieces of that player should remain on the board. The disconnected user should be able to return to the game. When the user is disconnected, its movements are skipped, and the next connected players make their moves.

~~4. Message exchange system with encryption~~

Design and implement a system for text messages exchange between users. The system should enable the following operations by users:

-Connecting/disconnecting from the system,

-Checking the status of another user (available/unavailable),

-Sending a text message to the connected user,

-All the messages should be encrypted. With the key, the user can decrypt the message

~~5. Distributed file exchange system with a maximum size limit~~

Write an application that allows file sharing. The system should enable the following operations by users:

-Connecting/disconnecting from the system,

-Checking the status of another user (available/unavailable),

-Sending a file to the connected user,

-Each user can set a maximum size limit.

~~6. Sketch table with exclusive access~~

Design and implement a virtual drawing board. Each user of the table can see its current state and can draw or erase it. Any user can temporarily take over the table, then only the user has the right to change the state of the table.

~~7. Ride-sharing system.~~

Design and implement a simple ride-sharing system. Drivers and clients connect to different servers. Drivers share their location and availability. Clients create ride requests. The system should pair the client with the nearest available driver. After the ride, the position of the client and the driver changes. It can be assumed that the travel time is proportional to the Euclidean distance between the start and the end of the route.

~~11 Distributed sort~~

The nodes have numbers and contain numeric items. They should sort the items in reverse order (i.e. the node with the highest number must contain the smallest item) by exchanging items in pairs, if pairwise not sorted. It is a continuous sorting, because the items should be generated and canceled accidentally.

~~12 Write-invalidate~~

Implement distributed memory with read and write access, using write-invalidate protocol (see slides).

~~13 Write-update~~

Implement distributed memory with read and write access, using write-update protocol (see slides).

~~14 money transfers~~

Consider bank accounts that are modelled as distributed objects. Implement transfers of money from one account to another, which should happen concurrently. Implement the transfers transactionally.

~~15 Implement distributed voting algorithm, simulate errors~~

Members propose a voting subject, then the nodes apply to quorum, then they vote and a result is announced. Voting rounds run in parallel.

~~16. Implement distributed file system.~~

Every node contains the local file tree. They can mount and unmount local trees to the global structure. Nodes can create and destroy files in remote trees.

17. Implement queueing system.

Nodes contain queues, all nodes can send/receive messages from the queues.