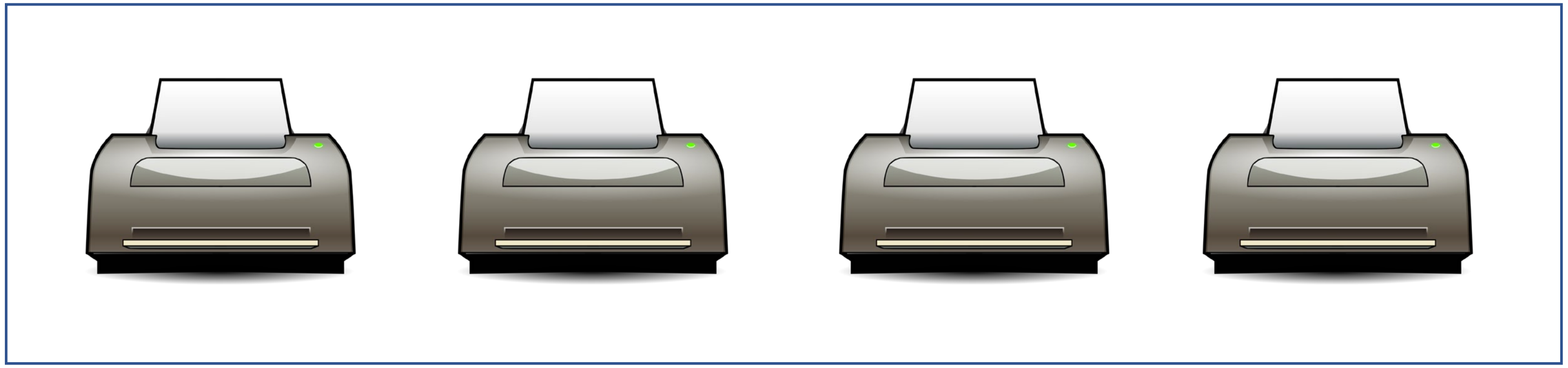
Exercise 1: Interfaces

- Consider a service for concurrently sharing a set of printers from a pool for several users (all printers are equal). Define an appropriate interface (for instance using Python-like oder C-like pseudocode) for such a service so that the openness requirements are satistified.
- Assume that the channel is secure.



- int Print(Document document, int PrinterID, Settings settings, int Token); => OK (0), ERROR (1), , NET_ERROR(2), ERR_DOCUMENT(3), NO_PAPER(4), ERR_BUSY(5), ERROR_AUTHFAILED(6)
- Bool IsPrinterAvailable(int PrinterID, int Token);
- Array<int> GetAllPrintersAvailable(int Token);
- int AuthorizeUser(User user); => Token
- Bool IsTokenValid(int Token);
- Int Logout(User name, int Token);
- String GetPrinterInfo(int PrinterID); => “name: Printer1, Location: office first floor … “
- Status GetPrinterStatus(int PrinterID, int Token); => returns printer status info
- List<Status> GetAllPrinterStates(int Token); => returns all current states of all printers

Exercise 2: Transparency

- Consider a system with the following requirements: a supercomputing cluster offers high-performance computing services to a scientific organization. This organization (and the cluster, too) is geographically distributed across different countries in several continents.
- The system must be capable of running batch computing jobs sent by its clients and reporting the results.
- List what types of transparencies are desirable and which not. Explain the reasons behind your choices.

Access	Location	Migration	Replication	Relocation	Concurrency	Fault
Yes => Hide resource (computing nodes) details	Yes => Hide where cluster nodes are located	Yes => Hide if a computing job is migrated to another node	Yes => Hide if there are replicas of the submitted jobs	Yes => Hide if a computing job/data is migrated to another node while in use.	Hide that more than one user is using the cluster	Yes => Hide that an error occurred
• Computers are all the same. • Difference if clients and cluster have different architectures • Differences in data representation are not relevant to the user.	• User does not care where the job is computed • If network latency is very high, then it might not be possible to hide location	• The user should not care. • If migration takes much time, then you cannot hide migration of resources	• If there are replicas (data) and these replicas are deleted afterwards, then it is OK to hide them	• Does not make sense if each job is being processed completeley on a given node. • If not, or if data and worker nodes are splitted, migrating data while in use would be desirable to hide faults.	• Yes, doesn’t need to know that other users are using the cluster	• Replicating data to get data from the nearest node • If one data node crashes, worker node does not notice