xTech @ bip.

Task 2
Technical Problem Solving

Context and task request

WorkSmart *is a company that offers its customers services to organize and manage their employees. The application grew from a few servers in the founder's garage to several hundred servers and appliances.*

However, the capacity of their infrastructure is now insufficient for the ***rapid growth of the application****.*

Due to this growth and the company's desire to innovate faster, WorkSmart asked the ***xTech team at BIP*** *to find a* ***solution*** *to its capacity problem.*

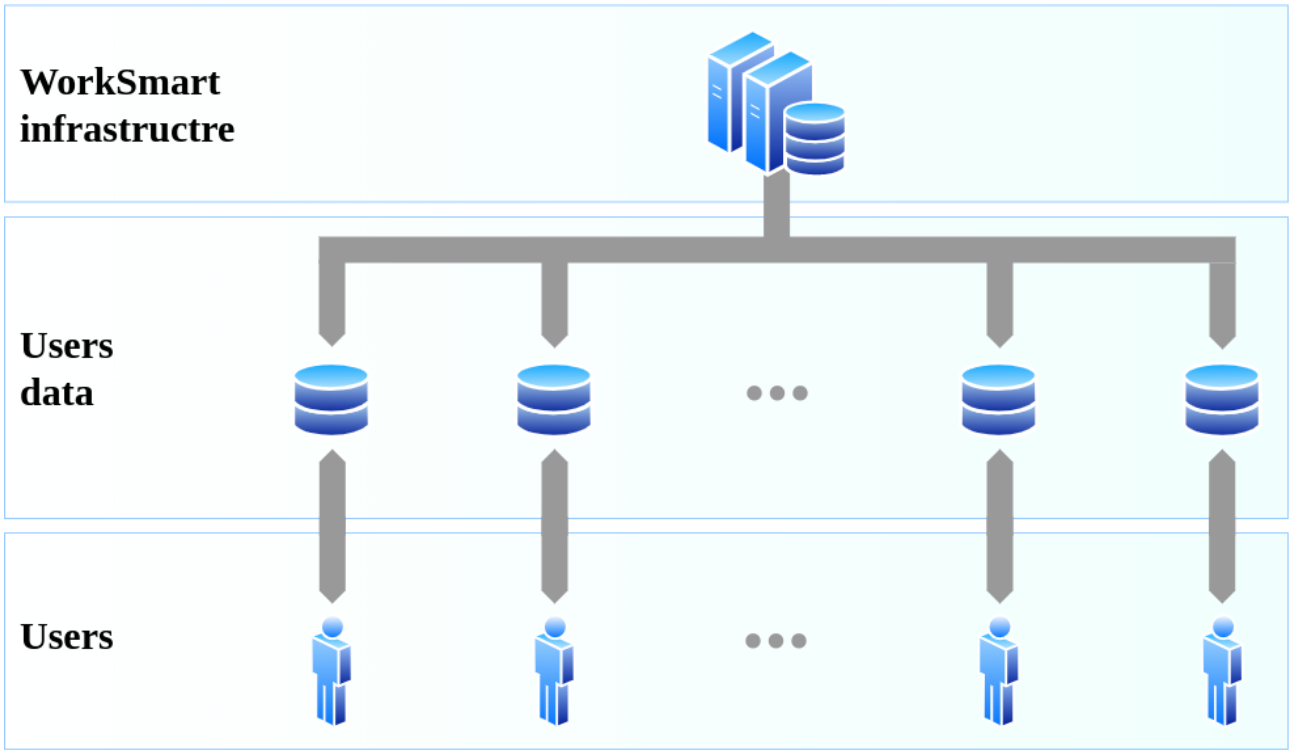
WorkSmart requires the ***design of a high-level architecture to load and transform large csv files****.*

The company is interested in evaluating both open source solutions and cloud services. Specifically your task is to:

- *design and explain a high-level architecture using only* ***open source frameworks and tools****;*
- *design and explain a high-level architecture using only* ***public cloud services*** *(AWS, AZURE, GCP).*

Problem analysis

On the basis of what we have been told, we can immediately identify two main problems to be tackled. The first is linked to the capacity of the company's infrastructure, which is now insufficient because of the increasing amount of work to be done, while the second concerns the need to have a large ". csv" file management system.
In light of what we know we can assume that we are in such a context:

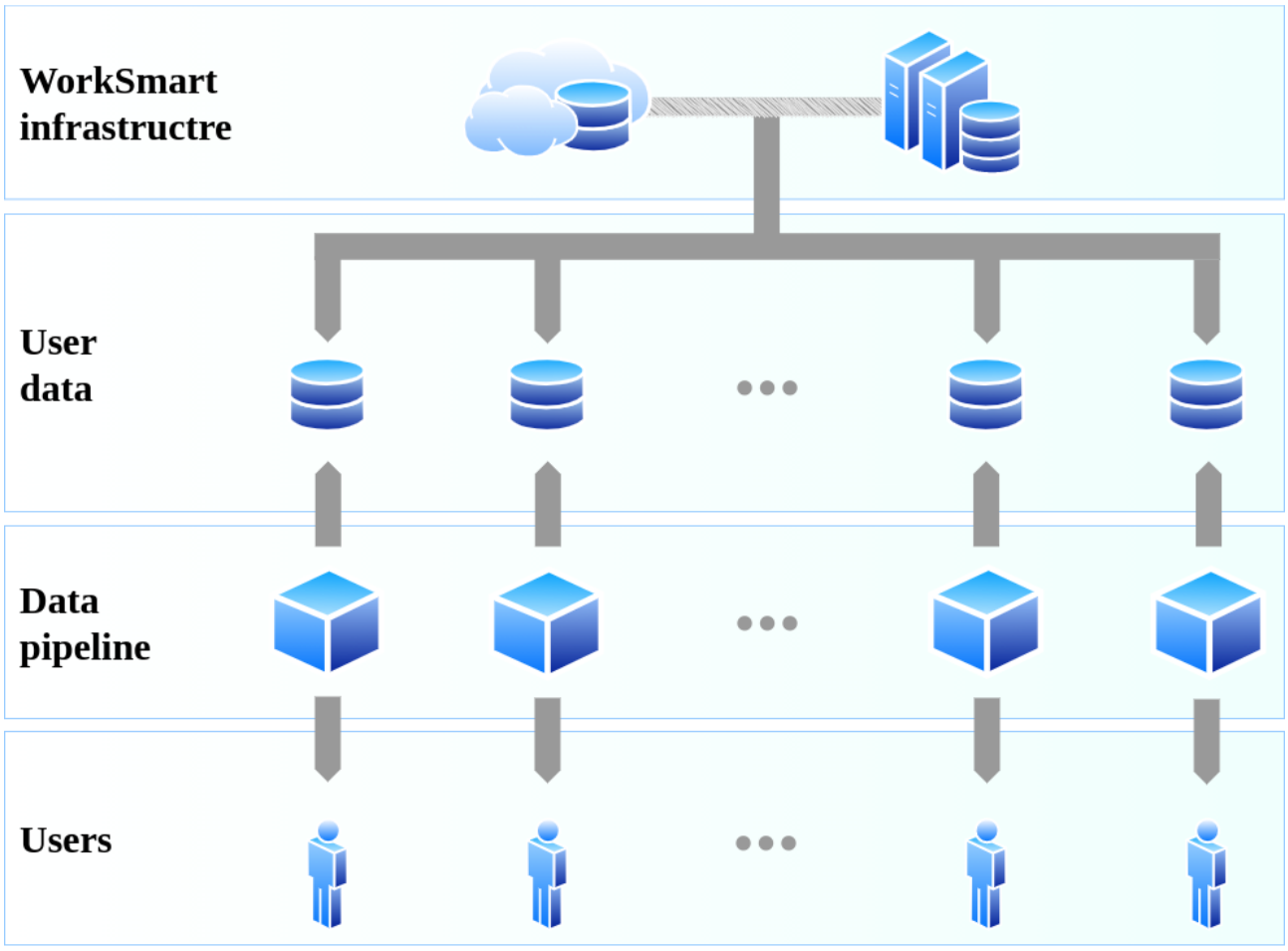


In this scenario we find three different levels: the first is *WorkSmart infrastructure* and is where the data of the various customers of the company are stored, in the second level there are the *data* of the various clients (e.g. documents, databases, files of various types, etc.) and finally there are the *customers* themselves who can access the data of their property.

NOTE: As this is outside of what is required, for simplicity the different levels of data access (reading, writing, special permissions, etc.) will not be considered, but we will assume that each user has full control.

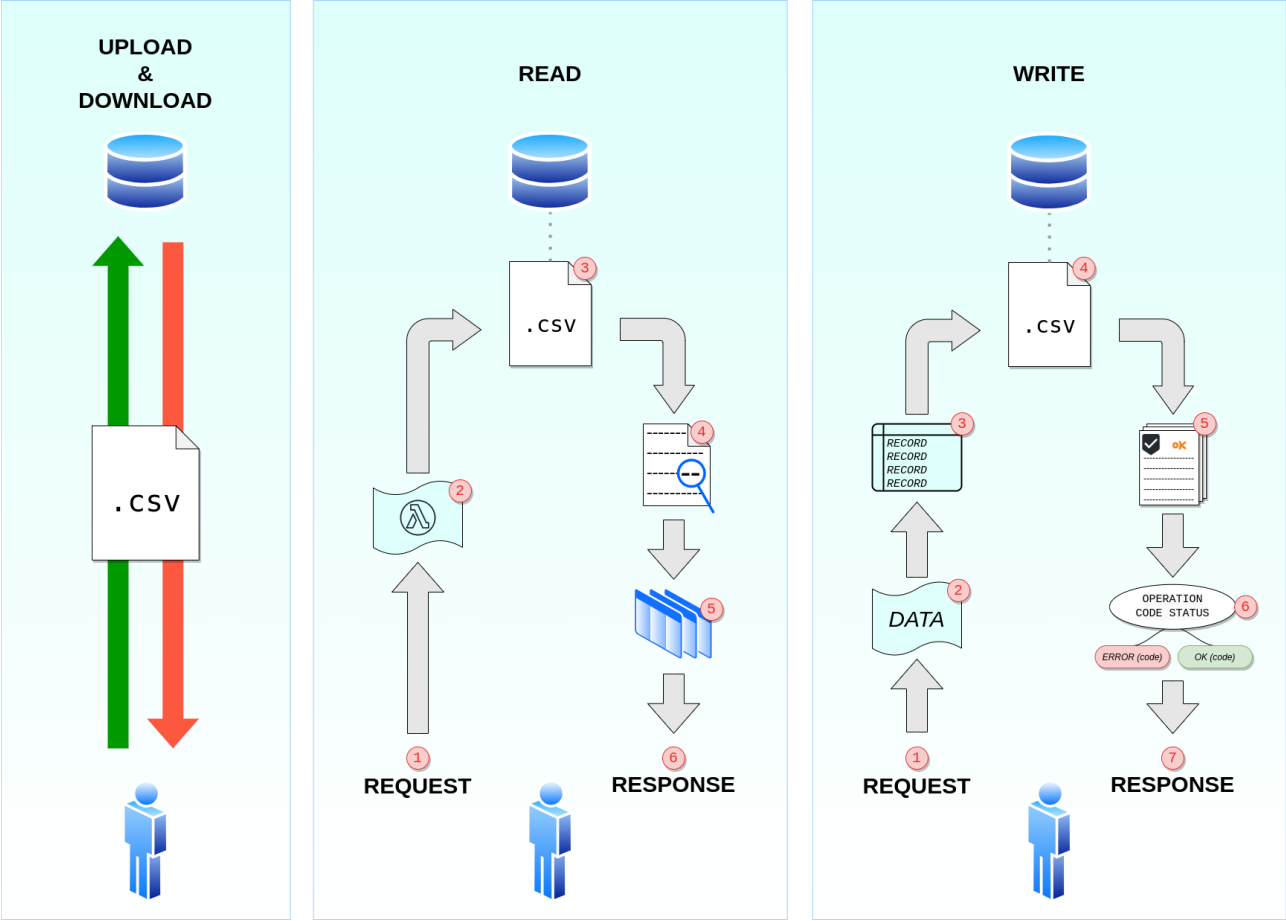
Based on this analysis and the problems we are called to solve, it's clear that before implementing any new feature in the system, the main issue to solve is capacity. To expand the existing infrastructure, it's possible to choose to add additional proprietary server, or rely on third-party cloud technologies. For generality purpouses, the diagram to be drawn up will take account of both possible solutions.

Regarding the problem of ".*csv*" file management , in order to access the data, any user's request passes through the *data pipeline*, within which the data are processed on the basis of the operation to be carried out. Thus, the previous context evolves as follows:


WorkSmart
infrastructre
User
data
Data
pipeline
Users

Task execution

In order to satisfy to the requests of the task, we will configure some "*basic operations*", whose implementation is the basis of the requests for formulated (upload and management of "*.csv*" files). These operations are file **uploads and downloads**, **read** and **write**.



The upload and download operations interface directly with the server (for simplicity it was deliberately omitted to consider the internal structure of the data archive of each user). The file is loaded or downloaded depending on the request of the user following the protocols of a common file system.

The reading operation is structured as follows:

1. The user sends a request for read to a certain file (or multiple files) to the server;
2. This request is processed in the format determined by the type of service that is intended provide users (see the Service developement section for more details);
3. The request is passed to the server that performs the operation to the file that concerns it;
4. The data shall be analysed and, where appropriate, processed on the basis of the request made. This phase allows to operate on large files (and possibly on portions of them) without necessarily having to download the entire document locally;
5. The extrapolated information so far is processed in the output format expected;
6. The result of the request is returned to the user in the desired format.

The writing operation is structured as follows:

1. The user sends a request for write to a certain file (or multiple files) to the server;
2. Data are extracted from the request;
3. The collected data are processed in a format compatible with that of the files that is required to change (since we are dealing with "*.csv*" files it is reasonable to think that the data will be reorganized into records);
4. The request is passed to the server that performs the operation to the file that concerns it;
5. The result of the operation shall be validated to verify the correctness and integrity of the data at the end of the operation;
6. On the basis of the validation process a code is generated, the task of which is to provide feedback to the user on the outcome of the operation;
7. The code indicating the result of the operation is returned to the user.

Service approach

As anticipated, the capacity problem is of fundamental importance for future business development.
The first possible approach is to expand existing infrastructure with additional servers. This would allow full control of the new infrastructure as it's entirely managed by the company itself. However it would certainly be very expensive in addition to being inconvenient in the long term (in addition to the purchase of new servers, costs related to installation and maintenance shall be considered), since, as the case has been presented to us, the workload to which the infrastructure is subjected is having a considerable growth.
Ithe second approach is the use of third-party cloud storage services. This solution offers many advantages: the reduction of costs related to the purchase and maintenance of the infrastructure, greater scalability, etc. By contrast, some "technical compromises" dependent on the service provider may be necessary.
As for the public cloud service that could be used to meet the needs of WorkSmart the choice was **AWS Simple Storage Service (S3)**, while among the open-source services that lend themselves to the requested purpose the choice fell on **MinIO**.
Both services offer a file storage system that addresses the problem of capacity (with the difference that Minio needs an infrastructure to run on). Moreover, the two systems offer similar performance in terms of file management allocations. Specifically, through the *Select*, function, Minio offers the possibility to read the contents of files through a special API. This functionality is based on the same function available for S3.
The problem of the ability can therefore be resolved, depending onthe technical necessities but also the approach that the company decides to follow (operate only with own infrastructures or to make use also of the collaboration of third party): AWS offers a wider range of "pre-packaged" services to operate files in the cloud (*AWS Athena* and *AWS Lambda* are good examples), In addition, the AWS services ecosystem would facilitate future expansion of WorkSmart services. On the other hand, since Minio is completely open-source, the possibilities of customizing the service and the potential offered to WorkSmart customers are unlimited, although this would require a greater effort in terms of the development of services.
In order to find a compromise between the two modes of expansion, a strategy to consider could be to rely on cloud services at first. This would allow to immediately solve the problem of the deficiency of infrastructures and could be used like "test bench" for new services deriving from the ecosystem AWS. All this in view of a future development of the corporate infrastructure, which in the meantime would have the opportunity to expand and improve systems through open-source tools such as MinIO.