Software Architecture Document

Amazon Simple Storage Service

Nathan Rose, Johnathan "JP" Prideaux

Version: 1.0

December, 2021

Revision History

Date	Version	Description	Author
11/16/2021	0.1	Created basic formatting	Nathan Rose
11/23/2021	0.3	Completed logical view section	JP
11/23/2021	0.4	Completed Module decomposition view	JP
11/23/2021	0.5	Completed Deployment view	JP
11/23/2021	0.6	Created Use Case View	Nathan Rose
11/29/2021	0.7	Created Process View	Nathan Rose
11/29/2021	0.8	Created Data View	Nathan Rose
11/29/2021	0.9	Issues and Concerns	JP
11/29/2021	0.10	Formatting Pass 1	Nathan Rose
12/01/2021	1.0	Final Formatting	Nathan Rose

Contents

1 Introduction

This document provides a high level overview and explains the architecture of Amazon Simple Storage Services(AS3) [1]. AS3 is a software service offered by Amazon through Amazon Web Service(AWS) whose goal is to provide organization and management of data in such a way that enables cost efficiences, security, and follows regulatary requirements.

1.1 Purpose

The Software Architecture Document(SAD) provides a comprehensive architectural overview of AS3. It presents a number of different architectural view to depict the various aspects of the system. It is intended to capture and convey the significant architectural decisions which have been made on the system.
In order to depict the software as accurately as possible the structure of this document is based on the "4 + 1" model view of architecture [2]. The "4+1" view model allows various stakeholders to find what they need in the software architecture.

1.2 Scope

The scope of the SAD is to depict the architecture of AS3 as described by Amazon in 2021.
This document describes the various aspects of AS3 that are considered to be archtecturally significant; that is, those elements and behavious that are fundabemtal for guiding the construction fo AS3 and for understanding this project as a whole. Stakeholders who require a technical understanding of AS3 are encouranges to start by reading this document. A video presentation is available as a suplimental resource to this at this link

1.3 Definitions, Acronyms, and Abbreviations

- AS3 - Amazon Simple Storage Service
- AWS - Amazon Web Service
- SAD - Software Architecture Document

1.4 References

[1] Amazon. "Amazon simple storage service." (2006), [Online]. Available: https://docs.aws.amazon.com/AmazonS3/latest/userguide/s3-userguide.pdf. (accessed: 11.17.2021).

[2] A. B.-T. 4. V. M. of Software Architecture. "Philippe kruchten." (1995), [Online]. Available: https://www.cs.ubc.ca/~gregor/teaching/papers/4+1view-architecture.pdf. (accessed: 11.16.2021).

[3] S. S. Council. "Data security standard." (2018), [Online]. Available: https://www.pcisecuritystandards.org/documents/PCI_DSS_v3-2-1.pdf?agreement=true&time=1637174273398. (accessed: 11.17.2021).

1.5 Overview

In order to fully document all aspects of the SAD contains the following subsections.

- section 2 describes the use of each view
- section 3 describes the architectural contraints of the system
- section 4 describees the most important use case realization
- section 5 describes the organization of the app
- section 6 Describes the designs concurency aspects
- section 7 Describes how the system is decomposed into modules

- section 8 Describes the data that is sent between modules
- section 9 Describes how the system will be deployed
- section 10 describes and performance issues and constraints
- section 11 describes and performance potential future issues

2 Architectural Representation

This document details the architecture using the views defines in the "4+1" model [2], but using the RUP naming convention. The views used to document the AS3 system are:

Use Case View

Audience: All the stakeholders of the system including end-users

Area: describes the set of scenarios and/or use cases that represent some significant, central functionality of the system. Describes the actors and use cases for the system. This view presents the needs of the user and is elaborated further at the design level to describe discrete flows and contraints in more detail. This domain vocabulary is independent of any processing models or representational syntex(ie XML)

Related Artifacts: Use-Case Model, Use-Case documents

Logical View

Audience: Designers

Area: Functional Requirements: describes the designs object model. Also describes the most import use case realizations and business requirements of the system

Related Artifacts: Design Model

Process View

Audience: Integrators

Area: Non functional requirements: describes the design's concurrency and synchronization aspects

Related Artifacts: (no specific artifact)

Module Decompoisiton View

Audience: Programmers

Area: Software components: describes the modules and subsystem of the application

Related Artifacts: Implimentation model, components

Data View

Audience: Data specialist, Database administrators

Area: Persistence: Desribes the architecturally significant persistent elements in the data model

Related Artifacts: Data model

Deployment View

Audience: Deployment managers

Area: Topology: describes the mapping of the software onto the hardware and shows the system's distributed aspects. Describes potential deployment structures, by including known and anticipated deployment scenarios in the archtiture we allow the implimenters to make certain assumptions on network performance, system interaction and so forth

Related Artifacts: Deployment Model

3 Architectural Goals and Constraints

The goals of S3 are varied as they ahve to encompess a wide variety of use cases. Ranging from small startups to large companies with with complex processes and large files. As a result of this the system has to be both highly reliable, accureate, and scalable.

3.1 Security

There are multiple security situations that have to be accounted for by AS3, not just security of the data from unauthorized users, but also compliance with regulations such as PCI DSS [3].

3.2 Persistence

This system has to persist the files and the metadata associated with files and users

3.3 Reliability/Availability

The system has to have high reliability and availablity in order to work with end users, however prefference is for reliability of information over availability.

4 Use-Case View

The use cases that are being analyzed for this project are:

1. upload files for sharing
2. Access Files
3. data processing of files
4. Ensure security for all files.
5. System scales up/down

These use cases are shown in Figure 1

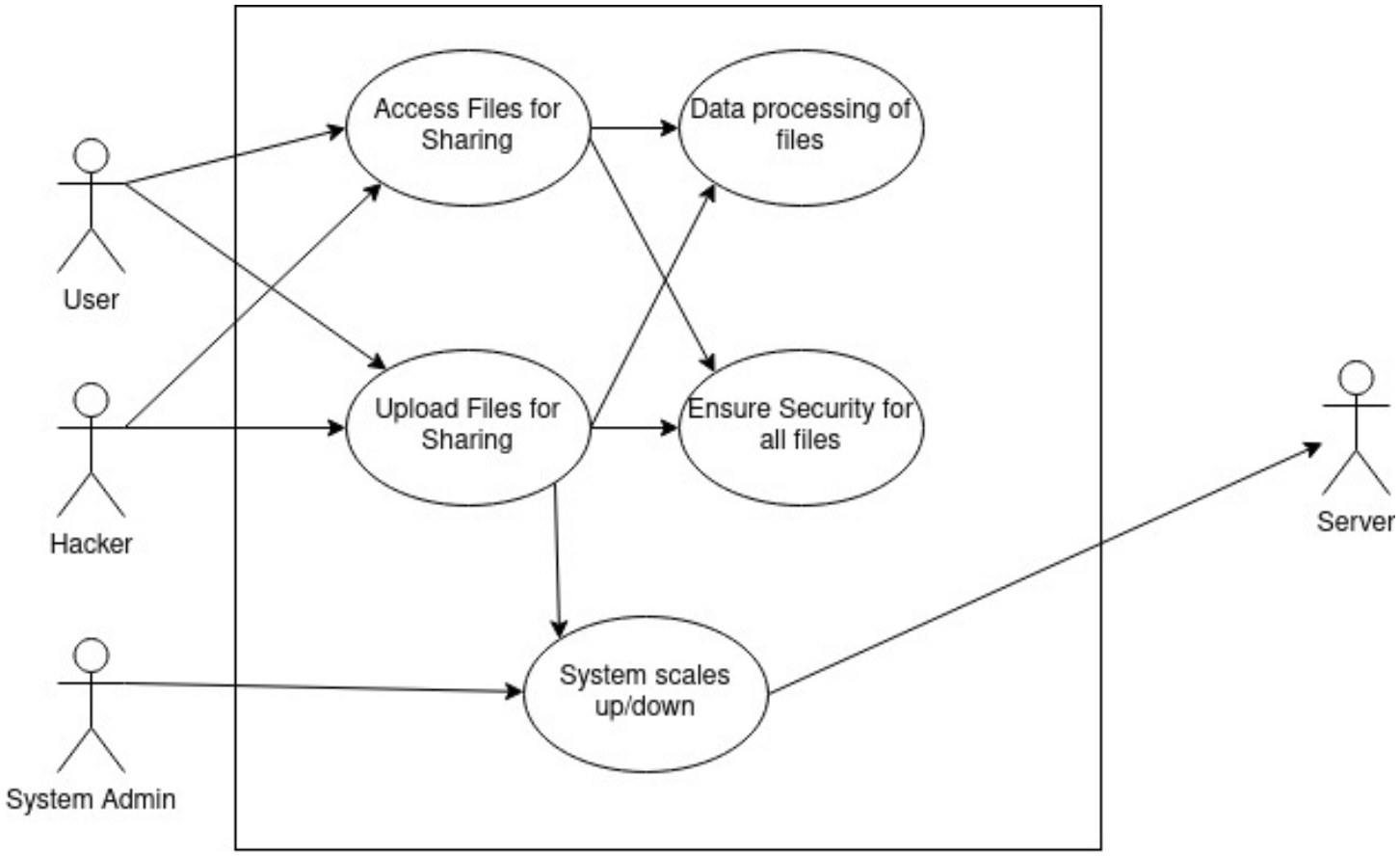

Figure 1: Use Case Diagram

4.1 Actors

The relevent actors for these use cases are:

1. User: User is the one who owns the files
2. System Admin: System admin is responsible for managing the servers and ensuring they are secure
3. Server: The physical computer that is responsible for running AS3
4. Hacker: An unauthorized user attempting to gain control of unauthorized information of a a user

4.2 Use-Case Realizations

I Upload Files for sharing
Scenario:

1 User selects a file for upload
2 System Verifies user has permissions
3 File is uploaded to Amazon S3
4 User selects Data processing of files
5 System performs post-upload processing

II Access files for sharing
Scenario:

1 User submits credentials

2 System verifies User

3 User selects desired file

4 System performs pre-download processing

5 System downloads file for user

III Data Processing of files
Scenario:

1 User specifies desired data processing at each step

2 System stores proccessing infromation

3 System performs processing when triggers are installed

IV Ensuring security of all files
Scenario:

1 User submits credentials

2 System verifies credentials

V System scales up/down
Scenario:

1 System requires more resources

2 System Admin ensures servers are available for it to scale into

5 Logical View

5.1 Overview

AS3's is made up of a buckets that hold objects which in this case are files. The system handles user input through the GUI and provides the appropriate security checks for when a user interacts with the system. The Data processing of the file then handles any changes made to the file if there happens to be a previous version. It also can handle new files and other changes.

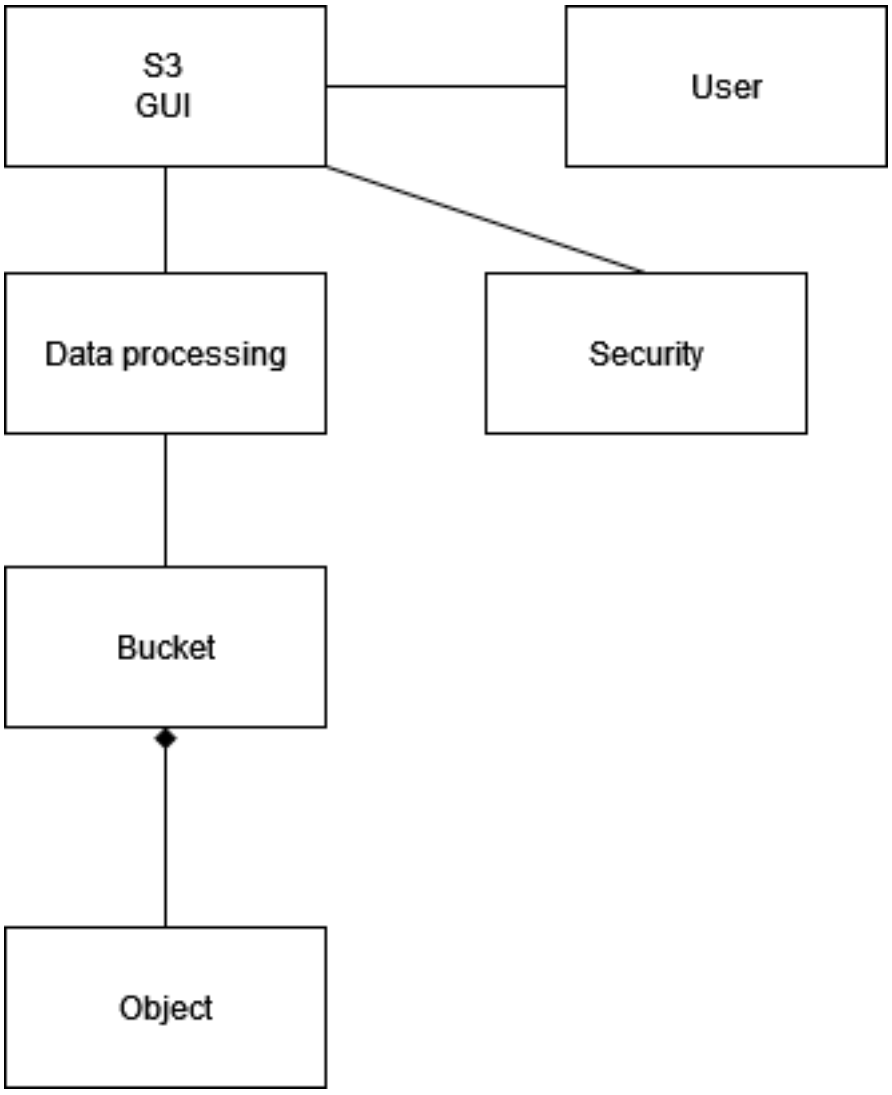

Figure 2: Logical View

6 Process View

This process view, shown in Figure 3 shows that most of the modules are run when another component or end product calls it however there are two exceptions to this. Those being the security module and the File Processing module. The security module is scanning for potential attacks and protection of user data. The file processing module is running that way certain process triggers can run at specific times.

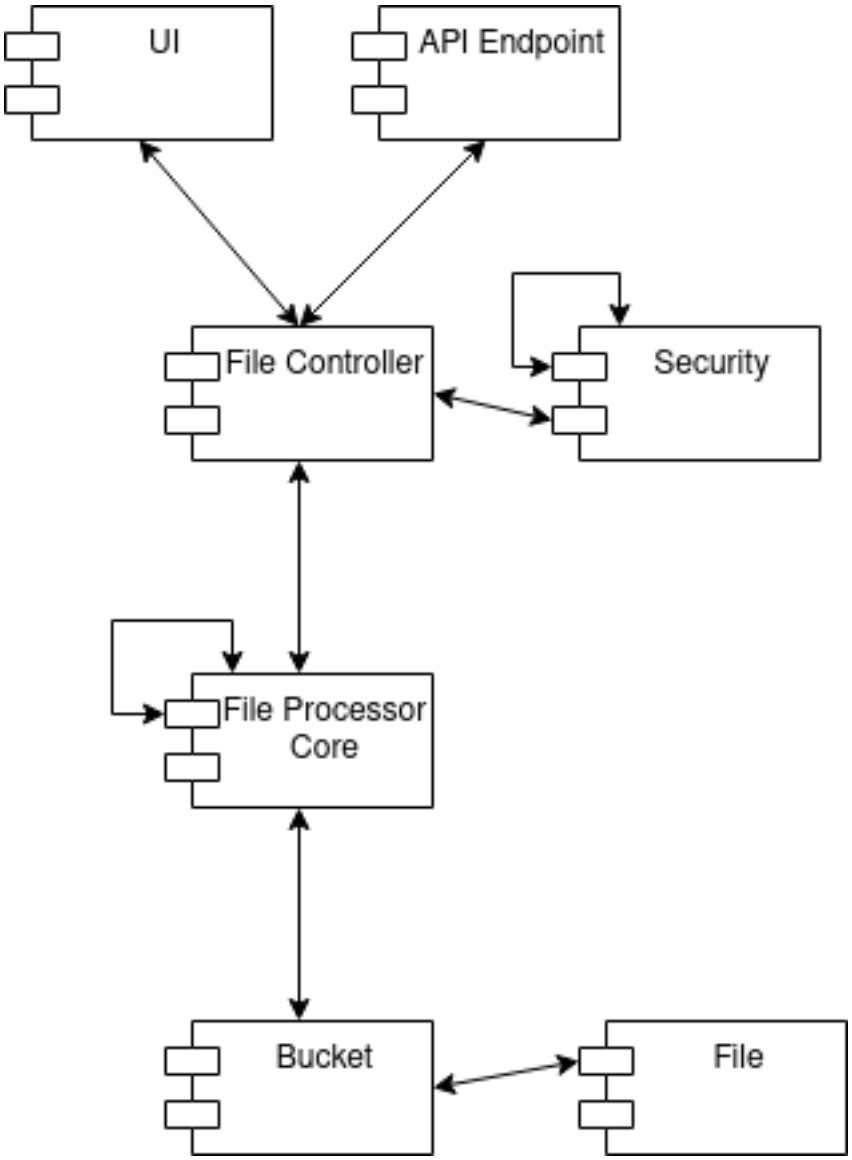

Figure 3: Process View

7 Module Decomposition View

The module decomposition view shows the three key components to AS3 and shows that redundancy in each component provides the reliability that is required for a system such as AS3.

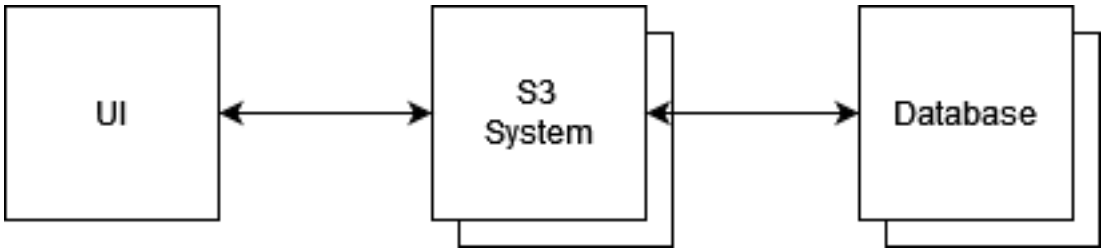

Figure 4: Module Decomposition View

8 Data View

The data view, shown in Figure 5, shows the data that AS3 has to store over time. This data is stored on the databases that is shown in Figure 4

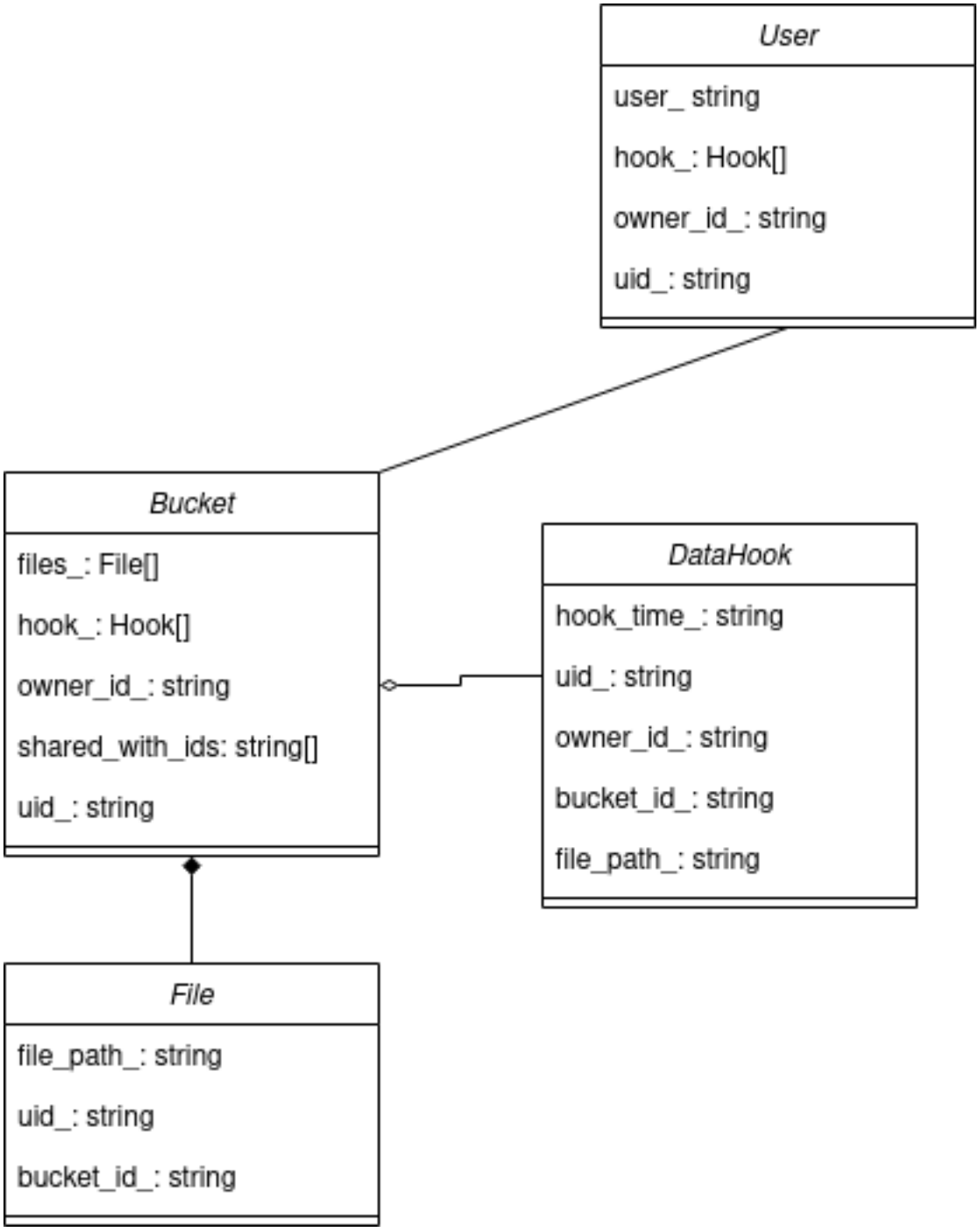

Figure 5: Data View

9 Deployment View

Deployment for AS3 relies on the following 4 layers. In order to expand the system to any scale only the bottom two layers will be impacted. The bottom layer will need to physically expand with more servers and hard drives and then the second layer will need to be aware of all the changes occurring to the bottom layer. Beyond that the

top two layers will not be impacted by change on the lower layers. This allows for flexible and easy scaling of AS3.

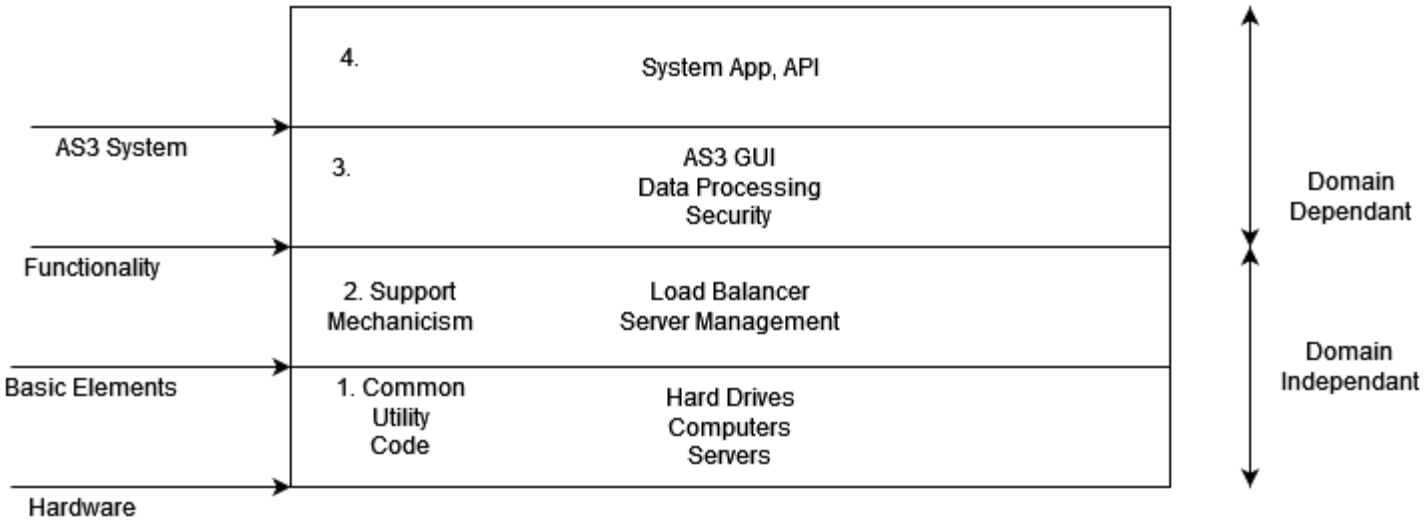

Figure 6: Deployment View

10 Size and Performance

With larger sizes there are multiple issues that need to be solved. The primary issue is that as buckets get larger and processing becomes larger, ensuring that all systems are in sync becomes more expensive. As the files and all file processing has to be in sync to ensure accuracy for the end users. In addition, to scale up, the system needs to distinguish what processing can be done in parallel and what has to be done in series. This change will likely make little to no difference for small-scale users, however for larger customers, this feature will likely be required in order to retain them.

11 Issues and Concerns

Some issues with the system as it stands is with scale. The larger the system the becomes the more the issues and concerns become obvious. The first one has to deal with size when a user wants to expand storage either the bucket max size has to grow or the content stored must be split to fit a fixed max bucket size. Both solutions have their own issues. If the buckets can change size you will not have the simplicity that comes with fixed bucket size. And if you have fixed bucket size splitting content just becomes another step in the process and another place where things can fail. With variable bucket size scaling is not an easy estimation or job to do but with fixed bucket size it is since you know how much a bucket can hold. Another concern with splitting up content is retrieval time. If a large piece of content is fragmented over several buckets this could impact retrieval time for clients.
Another issue with the system is once scaled up syncing across user will become more resource intensive as described in the previous section. A concern that relates to buckets that have multiple users accessing them is continuity with each users changes. On a small amount of changes with small amount of users this should not be a problem but with more and more users and bigger and bigger amounts of content constantly being changed this may impact the contents CIA attributes or the trade off may be slower performance of the system.