



Scienxt Journal of Computer Science & Information Technology
2023; Volume-1; Issue-1, pp. 1-11

Cloudroid

**Mrudul Borse¹, Akash Ranajkar², Parth Sakure^{3*},
Vaibhavi Sidam⁴, Prof. Nikita Kapadnis⁵**

*^{1,2,3,4,5}Department of Computer Engineering,
SCTR's Pune Institute of Computer Technology
Dhankawadi, Pune, India
Email: parthsakure01@gmail.com*

<https://zenodo.org/record/7965470>

*Corresponding author: Parth Sakure

Abstract:

The Babbage Difference Engine, created in the 1800s, is considered the first computer ever built. Despite its historical significance, the machine's memory storage capacity was quite limited, with just 675 bytes available. Fast forward to the present day, and technological advancements have progressed to the point where most of us carry smartphones in our pockets that boast storage capacities well beyond what the Babbage Difference Engine could ever have imagined. However, despite our smartphones being capable of storing more than 12 gigabytes of data, we still sometimes find ourselves running out of space. This is particularly true in a world where digital media has become an integral part of our daily lives, with high-resolution photos and videos, music, and various other files taking up valuable storage space.

As a result, the demand for extra storage has increased exponentially, and technology companies have responded by introducing various mediums for storing our data. External hard drives, cloud storage, and even physical storage devices like USB flash drives have become commonplace, providing us with additional storage space to store our important files. In today's society, with the vast amount of data being generated every day, the need for ample storage space is only expected to increase. As technology continues to evolve, we can expect to see even more innovative storage solutions being developed to meet the growing demands of users.

The storage that smartphones provide can be divided into 2 parts, one is used for storing application data, and the other is used for storing actual/personal data. While installing new applications into a smartphone there are times when the apt amount of storage is not available due to which we are asked to uninstall some pre-existing applications to accommodate the new application. There are times when uninstalling pre-existing applications is not possible as it will lead to loss of data.

In today's digital age, the amount of data generated every day is growing exponentially. This has led to the development of increasingly powerful mobile devices, which are capable of handling complex tasks and running a wide range of applications. However, these devices have limited storage capacity, which can make it challenging for users to store all of their files and applications on their devices.

To address this challenge, we aim to provide a platform that allows users to access virtual cloud-based storage, where they can install and use apps in real time without investing physical memory on their devices. This means that users can access a wide range of applications without worrying about running out of storage space on their devices.

By providing a virtual storage platform, we aim to reduce the reliance on physical memory storage and enable users to access the applications they need without any constraints. Our platform offers a scalable and cost-effective solution for users who require access to a large number of applications without having to worry about storage limitations on their devices.

Overall, we believe that our platform has the potential to revolutionize the way users access and use applications on their devices, making it easier and more convenient than ever before.

Cloudroid uses container-based technology to simulate Android OS on mobile/desktop devices so that users can access a virtual smartphone as and when required without any hardware restrictions provided they are connected to an active internet service.

Keywords:

Babbage Difference Engine, Cloud-based storage, Cloudroid, Android systems, IP Tunnelling, Remote Desk Protocol (RDP)