

dCloud : Making the power of distributed computing accessible to anyone

A Second Year Project Report
Submitted to the
Faculty of
Bennett University

By
[Anudit Nagar, Aachman Garg, Vinayak Tiwari]
[E18CSE024, E18CSE001, E18CSE211]



Department of Computer Science Engineering
November 2019

Greater Noida-201310, Uttar Pradesh, India

TABLE OF CONTENTS

LIST OF TABLES	3
LIST OF FIGURES	4
1. INTRODUCTION	5
1.1. Problem Statement	5
2. Background Research	6
2.1. Proposed System	6
2.2. Goals and Objectives	7
3. Project Planning	8
3.1. Project Setup	8
3.2. Stakeholders	9
3.3. Project Resources	9
3.4. Assumptions	10
4. SYSTEM ANALYSIS AND DESIGN	11
4.1. Overall Description	11
4.2. Users and Roles	11
4.3. User Stories (Requirements)	12
4.4. Design diagrams/ UML diagrams/ Flow Charts/ E-R diagrams	19
4.4.1. Use Case Diagrams	19
4.4.2. Class Diagram	20
4.4.3. Activity Diagrams	21

LIST OF TABLES

<u>Table</u>	<u>Page</u>
Table 1: Goal and Objectives	7
Table 2: Project Setup	8
Table 3: Stakeholders	9
Table 4: Project Resources	9
Table 5: Assumptions	10

LIST OF FIGURES

<u>Figure</u>	<u>Page Number</u>
Figure 1: Use-case diagram	17
Figure 2: Activity Diagram	18
Figure 3: Class Diagram	19
Figure 4: SequenceDiagram	20

1. INTRODUCTION

A Cloud that utilizes aims to be the decentralized alternative for AWS allowing users to run their applications and host their data on a distributed collection of devices ensure no central entity has access to the complete data ensuring privacy and security. dCloud is backed by the ethereum chain adding an additional layer of privacy. dCloud also allows users to provide computational resources to network according to which they are rewarded.

It utilizes IPFS, which attempts to address the deficiencies of the client-server model and HTTP web through a novel p2p file sharing system. The main advantages of dCloud are in decentralization, fault tolerance and scalability.

Nodes do not require central coordination, the system can function reliably even when nodes fail or leave the network, and dCloud can scale to accommodate millions of nodes. Together these features result in a cloud architecture that is generally more resilient than single client-server structures.

1.1 PROBLEM STATEMENT

Over the years, companies such as Facebook, Amazon, Google and Microsoft have come to dominate other companies that support the Internet. These centralized platforms attract a large percentage of Internet users' attention and hold their data, and other companies pay them to make their content discoverable on these platforms.

It has turned these centralized platforms into doorkeepers of information and made us believe in following them. dCloud aims to decentralize this architecture giving users back the control of their personal data and ensuring privacy and security is maintained right from the fundamental core of dCloud.

2. BACKGROUND RESEARCH

Storage and Computation

The storage layer of the network is handled by the Interplanetary File System (IPFS). IPFS is a peer-to-peer (p2p) file sharing system that fundamentally changes the way data is distributed across a distributed cluster of devices. IPFS consists of several innovations in communication protocols and distributed systems that have been combined to produce the next-gen file system.

Decentralized Coordination

Hash Table is a data structure that stores information as a key/value pair. In distributed hash tables (DHTs), the data is spread across a network of different computers and devices, this data is also coordinated to enable efficient access and lookup between nodes.

Here is the research paper we took help from to understand the functioning of these parts,

```
@misc{benet2014ipfs,
  title={IPFS - Content Addressed, Versioned, P2P File System},
  author={Juan Benet},
  year={2014},
  eprint={1407.3561},
  archivePrefix={arXiv},
  primaryClass={cs.NI}
}
```

Blockchain Technology

Blockchain technology allows us to gain instant access to a decentralized infrastructure where we have access to an ecosystem built from the ground up to allow decentralized computations and enable edge devices to communicate in a global fashion.

Here is the research paper we took help from to get insights into understanding the integration of blockchain and sharing data across devices securely,

```
@misc{nagar2019privacypreserving,
  title={Privacy-Preserving Blockchain Based Federated Learning with Differential Data Sharing},
  author={Anudit Nagar},
  year={2019},
  eprint={1912.04859},
  archivePrefix={arXiv},
  primaryClass={cs.CR}
}
```

2.1 PROPOSED SYSTEM

Utilizing Blockchain technology in conjunction with distributed computing technology to enable applications to be able to operate in a completely decentralized manner. Since the advent of the internet, Governments have attempted to control ideas and information, but with passing time the old social constructs they have created are crumbling. dCloud gives developers the platform to deploy their application in a scalable manner that cannot be censored, giving people access to the knowledge bank that the modern da internet is.

2.1 GOALS AND OBJECTIVES

Table 1: Goal and Objectives

SNo.	Goal or Objective
1	Our goal is to create a platform where users can deploy and run their containerized applications. Similar to Heroku.
2	This platform will be backed by a blockchain based decentralized coordinator that handles the deployment of the application across the nodes in the network
3	It enables people to contribute the spare resources available on their computer or mobile devices to contribute computational power towards the network by joining in as a node.
4	dCloud allows users to earn proportional to the resources they provide to the network in cryptocurrency allowing them to earn from the unused resources.
5	It provides developers with the flexibility to integrate with various version control and collaboration dashboards like GitHub and GitLab.
6	Gain experience while working in a Team on new and innovative technologies.

3. PROJECT PLANNING

The project was planned according to the previously discussed outline by the team. Submissions of documents like the Project Proposal, sprint and user-stories were taken as a base to plan out the details and implementation of the project so that the entire team has a clear idea of the project. Tasks were scheduled in a specified time-frame in order to reach our goal by the end of the semester.

Summary of the initial Project Setup Planning,

1. Setup AWS Servers to create a testbed where the app can be deployed and tested in real world conditions
2. Setup Continuous Integration and Continuous deployment pipeline in order to set up automated testing and deployment.
3. Utilize codecov to setup code testing coverage to ensure end to end testing of the application as it's developed and get reports on the same.
4. Setup language specific linter in the IDE to ensure coding standards are upheld across the codebase.
5. Setup language specific linter in the IDE to ensure coding standards are upheld across the codebase.

3.1 PROJECT SETUP

Table 2: Decision Description

SNo.	Decision Description
1	AWS v Azure v Google Cloud servers for development
2	Python, javascript and Solidity for development of the project.
3	Choose Linux as the Local development environment.
4	Decide the coding standards for the project

3.2 STAKEHOLDERS

Table 3: Stakeholders And Their Roles

Stakeholder.	Role
Anurag Goswami	Mentor
Rupak Chakraborty	Instructor
Anudit Nagar	Full Stack Developer.
Aachman Garg	Front End Developer
Vinayak Tiwari	Database Manager
Developers	Developing and using the platform to deploy applications.
Users	Using the applications deployed on top of the network.

3.3 PROJECT RESOURCES

Table 4: Project Resource Along With Description and Quantity

Resource	Resource Description	Quantity
AWS Servers	Servers used to deploy and maintain the application.	3
Database	GraphQL Instance	1
Capstone Team	Our Team of who will be the primary developers of the application.	3
Linux Workstations	These machines will be the primary devices our team will use for creating the project	3

Android Phone	An Android phone to be used as test hardware for the mobile version of the software.	1
---------------	--	---

3.4 ASSUMPTIONS

Table 5: Assumptions

#	Assumption
A1	The capstone team and mentors will be able to meet face to face once a week.
A2	Team will have sufficient time to complete a working model by the end-semester.
A3	AWS Free Tier would be for the team.
A4	Team members will be able to work with technologies like React Solidity and GraphQL.
A5	Team will have sufficient time to complete a working model by mid-semester.

4. SYSTEM ANALYSIS AND DESIGN

4.1 OVERALL DESCRIPTION

A distributed cloud means that the computation, storage, and networking are in a microcloud located outside the centralized cloud. The distributed cloud is closer to the end-user as a decentralized cloud system.

This cloud system helps us combat a variety of issues including privacy, data security, server downtimes and helps us to simultaneously incentivise the ecosystem letting users earn from extra storage and computational resources available on their devices, increasing the number of devices available in the network further improving fault resistance.

The application will be written in a combination of Python for backend, Javascript for Frontend, and Solidity for Blockchain. The application will enable seamless deployment of containerized applications made by the developers and let users use these decentralized applications in a similar fashion.

4.2 USERS AND ROLES

User	Roles
Development Team	The capstone team and mentors will be able to meet face to face once a week.
Admin	Team will have sufficient time to complete a working model by the end-semester.
Developers	AWS Free Tier would be for the team.
Users	Team members will be able to work with technologies like React Solidity and GraphQL.

4.3 USER STORIES (REQUIREMENTS)

ID	Feature Names	Story points
100	Be able to login/logout in the application	2
200	Be able to connect and import various code sources like GitHub and GitLab	5
300	Be able to create and host the application.	15
400	Be able to access the application from outside	8
#	Total Story Points	30

SPRINT 1

Estimated User Story Points: 2

Actual Completed User Story Points: 2

ID	Added	Description	Status	Story Points	Actual Equivalent Story Points	% Completed
100	onset	Be able to login/logout in the application	C	1	1	100
100	Onset	Be able to fetch the user data from the profile	C	1	1	100
Acceptance Criteria			Verification			
110	The user must have appropriate ID and password		The password is encrypted and set to the backend for verification			
111	ID should automatically logged out after 1 hr of idle time		The user session is expired if the site is left open for too long with no activity to ensure security			
Id	Tasks			Resources		
1	Fetch the correct API tokens for login			Anudit Nagar		
2	Create a login interface for the user to be able to Input Credentials			Aachman Garg		
3	Check and maintain user credentials			Vinayak		

SPRINT 2

Estimated User Story Points: 5

Actual Completed User Story Points: 5

ID	Added	Description	Status	Story Points	Actual Equivalent Story Points	% Completed
200	onset	As an user I must be able to connect and import various code sources like GitHub and GitLab	C	2	2	100
200	Onset	As an user, I should be able to access my connected repositories	C	3	3	100
Acceptance Criteria			Verification			
211	The user must have Github/GitLab account and be able to allow access to the code.		The user is able to connect their account and import code.			
212	The accessible code should be checked before deploying		The imported code is correctly compiled and checked before deploying.			
Id	Tasks			Resources		
1	Maintain Auth Tokens and			Anudit Nagar		
2	Maintain User's codebase internally			Anudit Nagar		
3	Create a front end for the user to be able to access and import code			Aachman Garg		
4	Maintain records of builds and build logs			Vinayak Tiwari		

SPRINT 3

Estimated User Story Points: 15

Actual Completed User Story Points: 15

ID	Added	Description	Status	Story Points	Actual Equivalent Story Points	% Completed
300	onset	As an user I must be able to create and host the application	C	7	7	100
311	onset	The application should be containerized and deployed	C	8	8	100
Acceptance Criteria			Verification			
311	The created application should be compiled and sent to the backend		The application is configured correctly, compiled and sent to the backend for propagation.			
312	The compiled application is then containerized and distributed to the network.		The application is packaged, containerized and sent to the required nodes in the network			
Id	Tasks			Resources		
1	Create a front end for the user to be able to discover/access the application			Aachman Garg		
2	Compile, Package and Containerize the Applications			Anudit Nagar		
3	Find a suitable node in the network and send the application to it to run			Anudit Nagar		
4	Maintain record of active nodes over DHT			Vinayak		

SPRINT 4

Estimated User Story Points: 8

Actual Completed User Story Points: 8

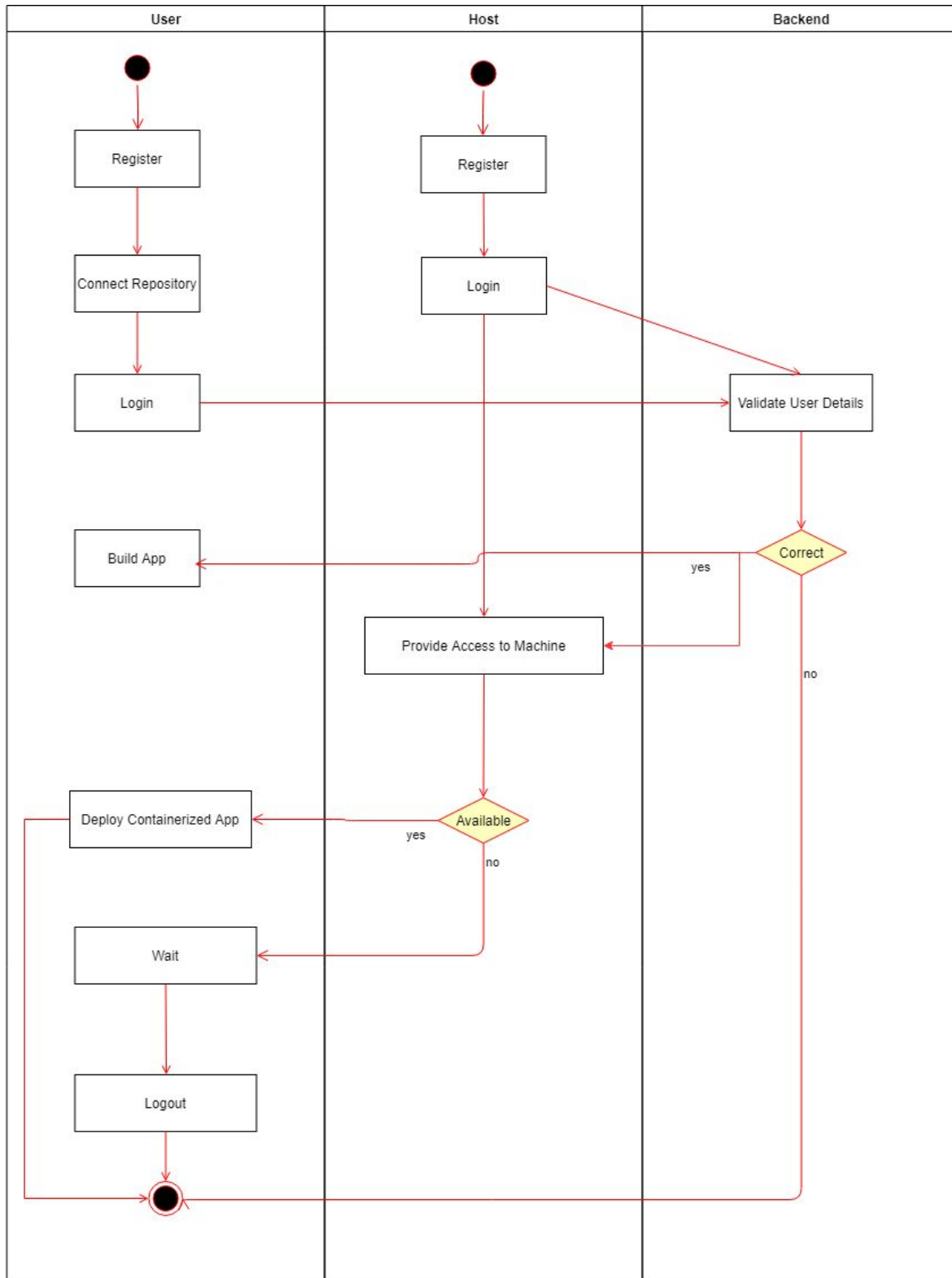
ID	Added	Description	Status	Story Points	Actual Equivalent Story Points	% Completed
400	Onset	As a User of platform, I want to be able to access the applications deployed on it on my phone.	C	4	4	100%
401	Onset	As a User of platform, The deployed applications should be accessible via a URL on my computer.	C	4	4	100%
Acceptance Criteria			Verification			
400	When the page is visited the application should load correctly.		The url link is open on a browser on Mobile and Computer to verify if the site loads correctly.			
401	The URL should point to the correct application.		When the application loads it should be the one that was intended to open.			
Id	Tasks			Resources		
1	Setup DNS over DHT and make the url point the application.			Anudit Nagar		
2	Create a front end for the user to be able to discover/access the application			Aachman Garg		
3	Maintain and update DHT over blockchain			Vinayak Tiwari		

4.4 Design diagrams/ UML diagrams/ Flow Charts/ E-R diagrams

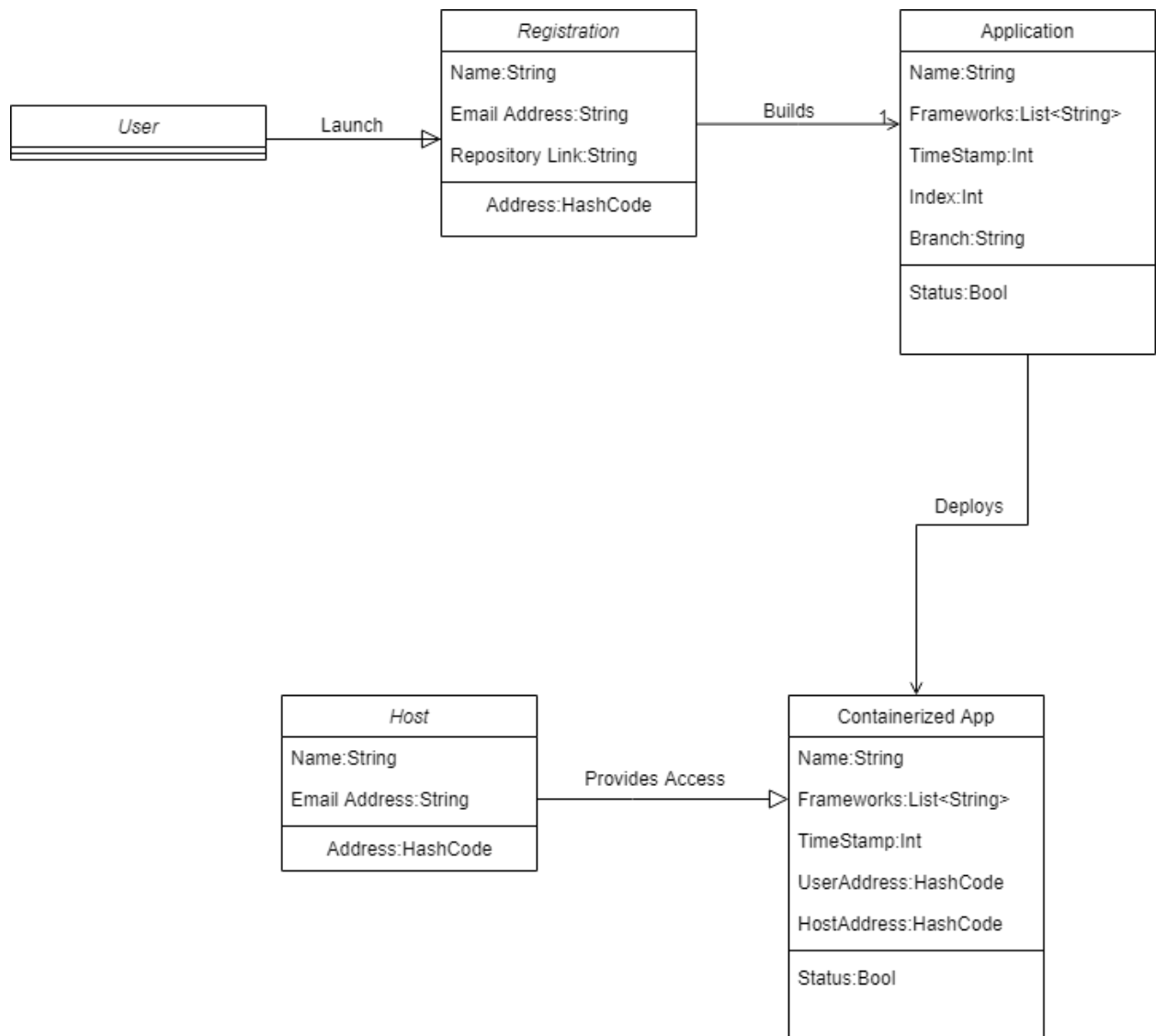
4.4.1 Use Case Diagram



4.4.2 Activity Diagram



4.4.3 Class Diagram



4.4.4 Sequence Diagram - Developer

