Distributed Cloud Storage – Technical Manual

0. Table of contents

1. Introduction

1.1 Overview

Distributed Cloud Storage – a set of programs that can turn your private servers into a cloud storage platform (think “Google Drive”, “iCloud”, or “Dropbox”). Our “node software” uses the Internet to connect your servers (“nodes”) into a cloud designed for storage. Use one of our “client programs” to connect to your network and upload/download files, all as if the network was a single cloud entity.

Distributed (de-centralised), secure, intelligent.

- Leverages the nodes’ underlying Operating Systems for persistent storage.
- Intelligent routing of files to the most optimal node in terms of storage load and network benchmarks.
- Reliability and privacy of storage at all times through redundancy and encrypted communications.
- Minimised single points of failure. Each node acts both as a client and as a server (a distributed system).

Portable (cross-platform), easily installable and configurable “node software” for technical/industry users requiring off-the-shelf private cloud storage solutions.

“Client programs” including a mobile friendly website client and graphical desktop client for the end-users of storage. Modern file explorer UI/UX to interact with the cloud storage platform.

1.2 Glossary

Cloud Node Go, Golang RPC Gob REST API JSON GCP Binary Fyne React.js Bootstrap Postgresql Shell Script Ansible Makefile

2. System Architecture

Go library. TCP.

Desktop GUI.

Desktop CLI.

Web app. Website. HTTPS

Secure communications.

3. High-Level Design

Class diagram.

Communications diagrams. TCP. HTTP (auth). web frontend <-> web backend <-> go library

4. Problems and Solutions

Data structure design (files, network).

Distribution algorithm (Calculating node benchmaks).

Frontend - bootstrap

Secure comms - HTTPS certs. Auth - JWT. Download. Auth middleware. DL. Login. Postgresql

5. Installation Guide

Obtain our binary. Or compile from source. Need Go. Go deps. create-react-app. Optional: Makefile, Ansible.

Any OS.

Need own servers to make a cloud.

6. Testing

Unit and integration tests.

System tests.

User testing.