

# Distributed Cloud Storage – Technical Manual

## 0. Table of contents

Distributed Cloud Storage – Technical Manual - 1. Introduction - 1.1. Overview - 1.2. Glossary - 2. System Architecture - 2.1. Operational Overview - 2.2. Class Diagram - 2.3. Communications Overview - 2.4. REST API Reference - 3. High Level Design - 3.1. Initial Design - 3.2. Current Design - 3.3. Major Design Considerations - 4. Problems and Solutions - 4.1. Network communications - 4.2. Cloud and Network data structures - 4.3. File Storage data structures - 4.4. Desktop Client - 4.5. Web Client - 4.5.1. Frontend - 4.5.2. Backend - 4.5.3. Secure Communications - 4.6. Automation Tools - 4.6.1. Deployment - 4.6.2. Scripting - 4.6.3. Dependency Management - 5. Installation Guide - 5.1. Cloud CLI - 5.2. Cloud desktop GUI - 5.3. Web client - 6. Testing - 6.1. GitLab CI - 6.2. Unit & Integration Tests - 6.3. System Tests - 6.4. User Tests - 6.5. Directory `/tests`

## 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 back-end <-> go library

## 4. Problems and Solutions

Data structure design (files, network).

Distribution algorithm (Calculating node benchmarks).

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.