An Exploration of Turing Pi Based Edge Cloud with Docker/Kubernetes

sdmay24-03

Hardware Team: Owen Perrin, Nick Bergan,

Software Team: Owen Henning, Cooper Caruso, Andrew Phelps, Kale Kester

Client & Advisor: Akhilesh Tyagi

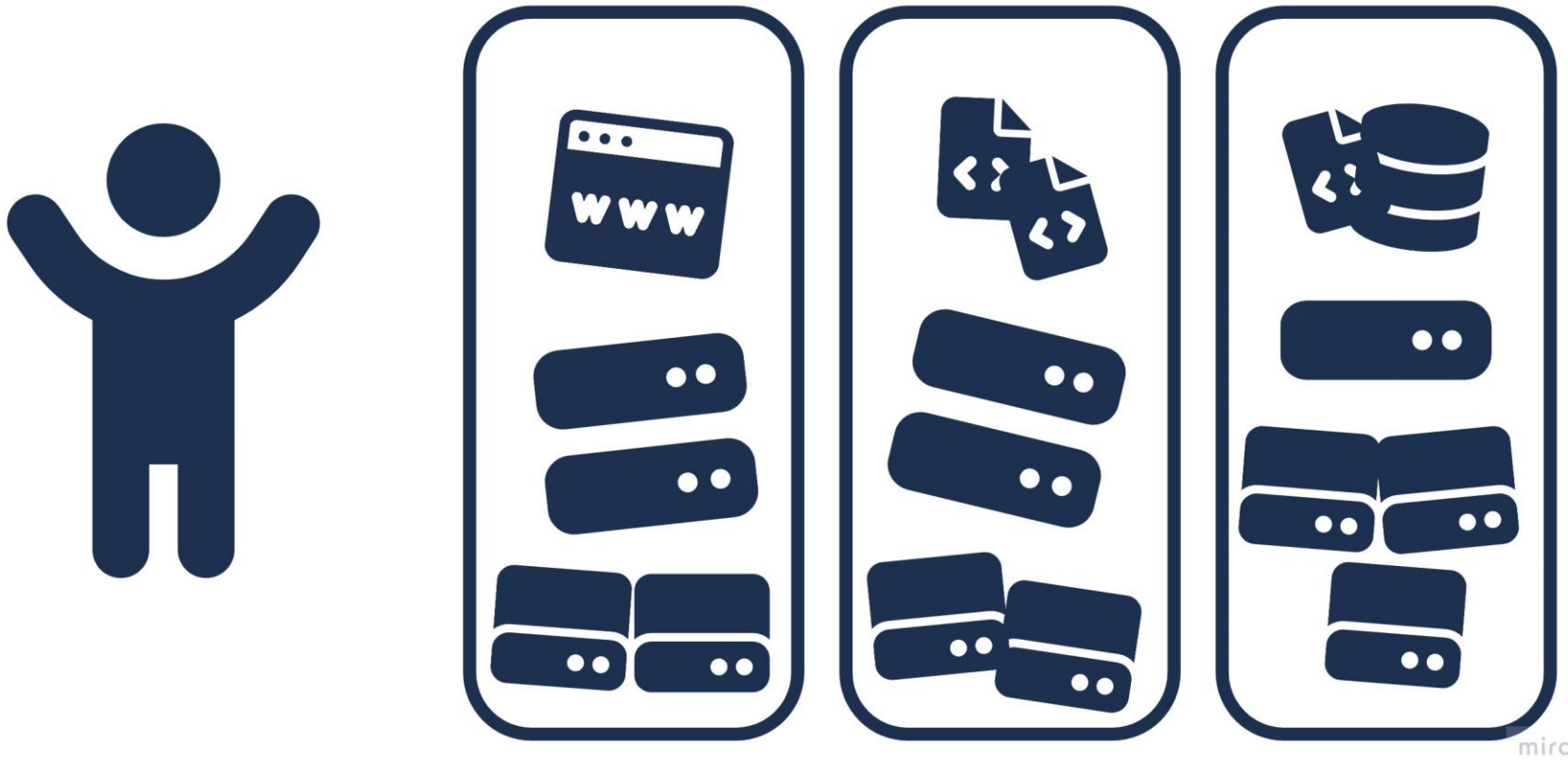
Project Vision

Proof of Concept:

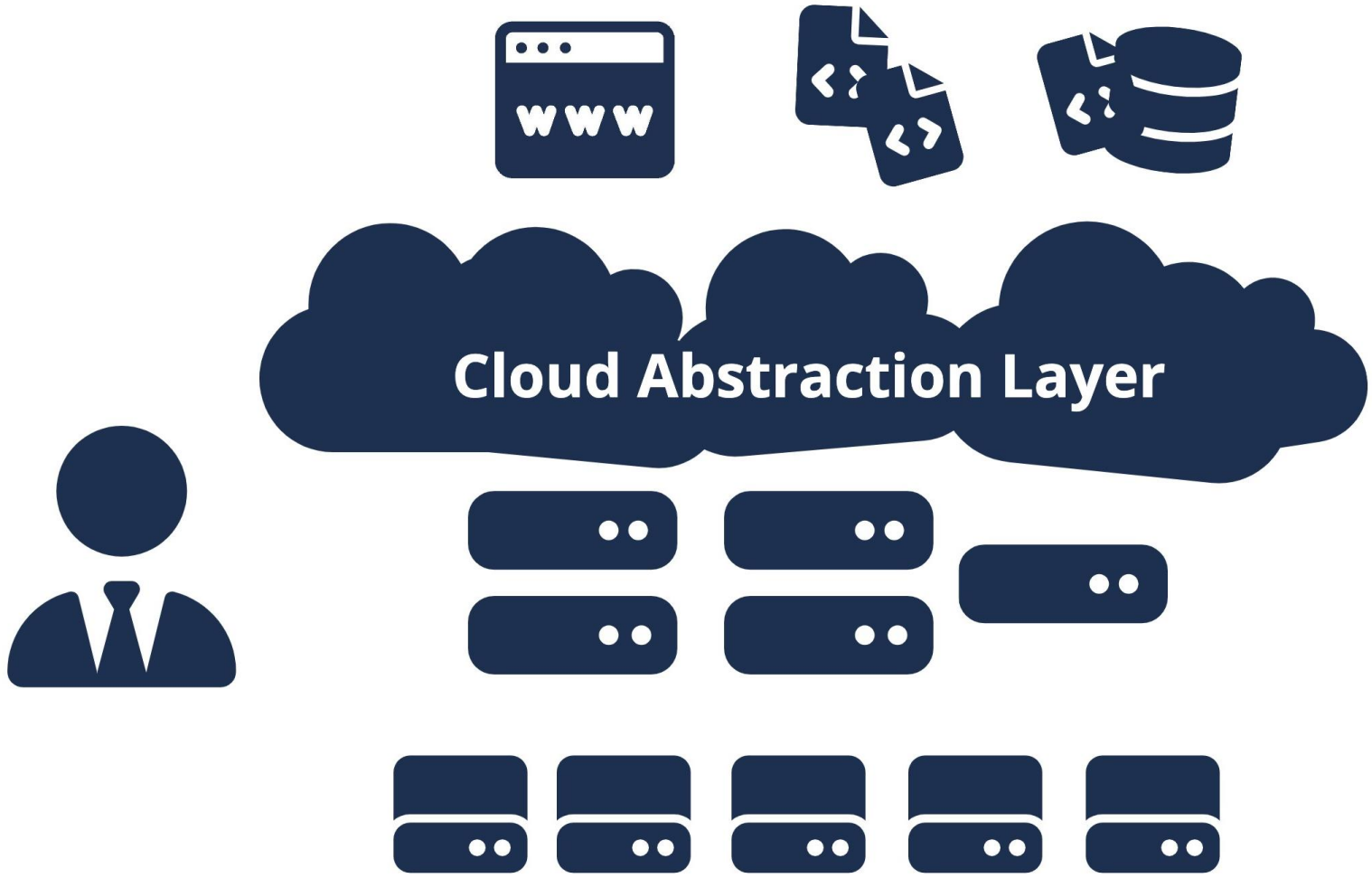
- Progression of cloud computing

- Abstraction, containerization, scalability

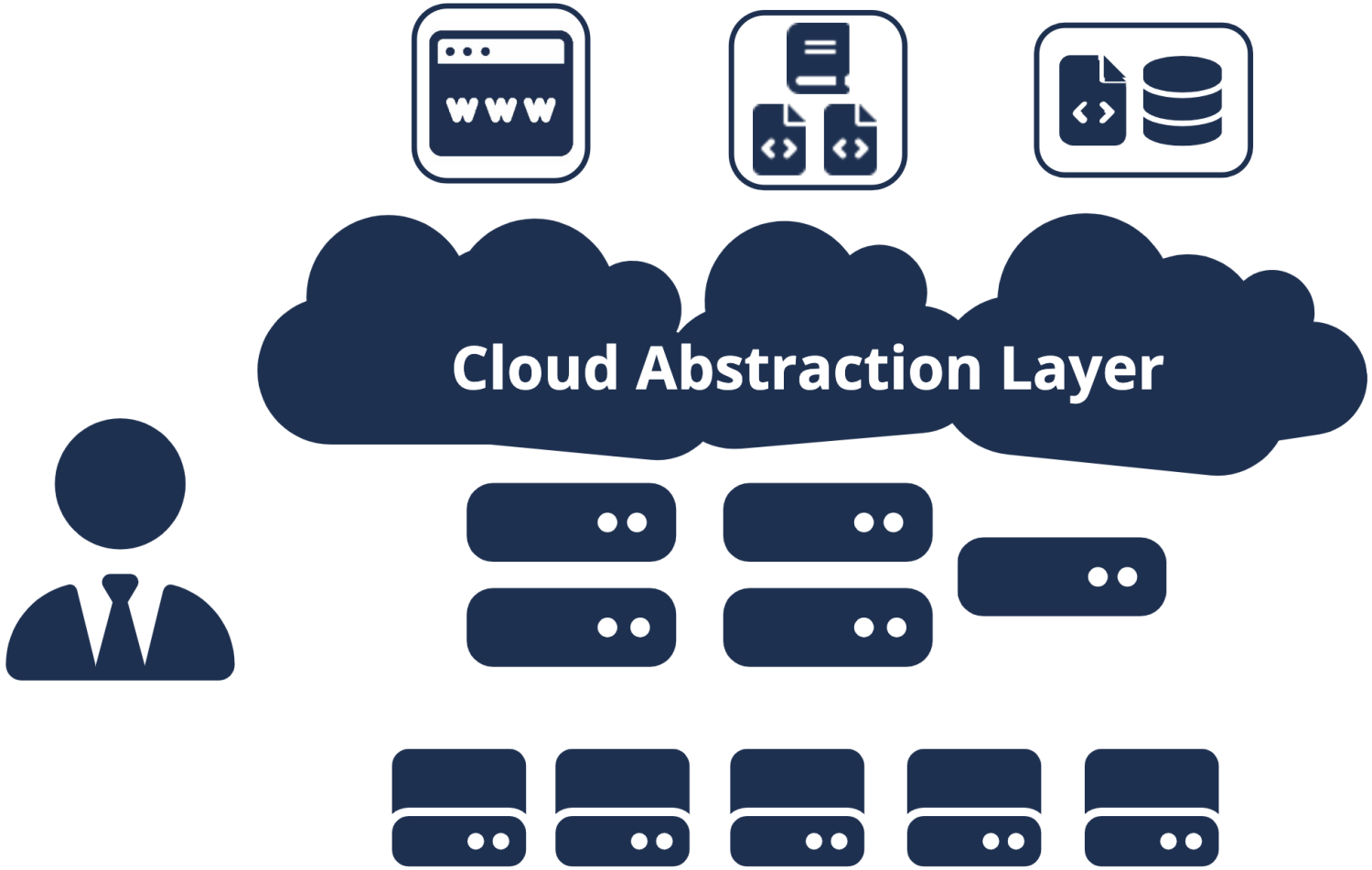
Conceptual/Visual Sketch



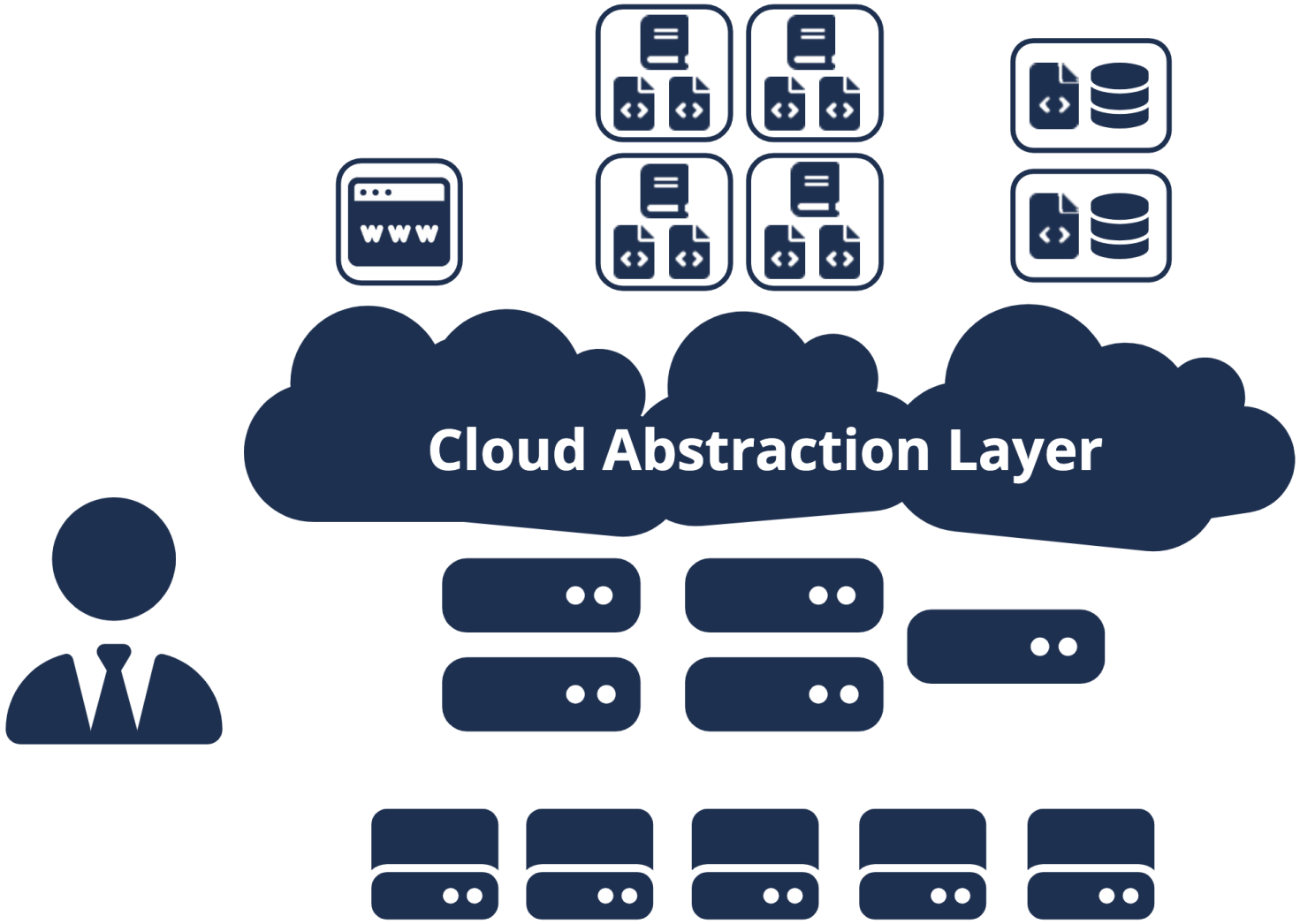
Conceptual/Visual Sketch – Abstraction



Conceptual/Visual Sketch – Containerization

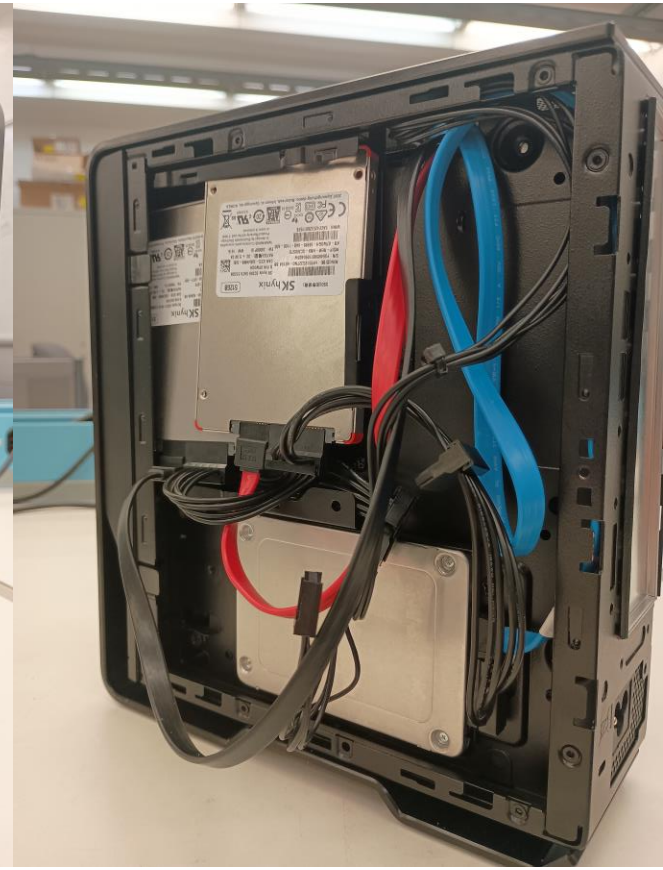


Conceptual/Visual Sketch – Scalability



Requirements - Hardware

- Private cloud deployed on a Turing Pi 2 mainboard
- Support 1-4 compute modules at any time
- Secondary storage for each compute module's Distributed file system



Requirements – Cloud and Containerization

- Support containerized applications via Docker
- Support container orchestration across all nodes via Kubernetes
- Be able to scale containerized applications across all clusters according to their resource needs
- Have a web API to deploy scalable containerized applications to the private cloud
- Expose API endpoints which support blob storage
- At least 90% of the API endpoints will be supported by a served website which allows users to perform all major actions (e.g. create, read, update, delete)

Requirements – Cloud and Containerization (Cont.)

- The website will be visually simple and aesthetically pleasing, using modern web components and UI principles
- The file storage supporting blob storage will be distributed across all compute nodes
- The system will have robust monitoring via its interface which reports functional status (e.g. nominal, process failures) and resource utilization
- The system will support a containerized video streaming application
- The system will show performance improvement for scalable containerized applications as more compute clusters are added

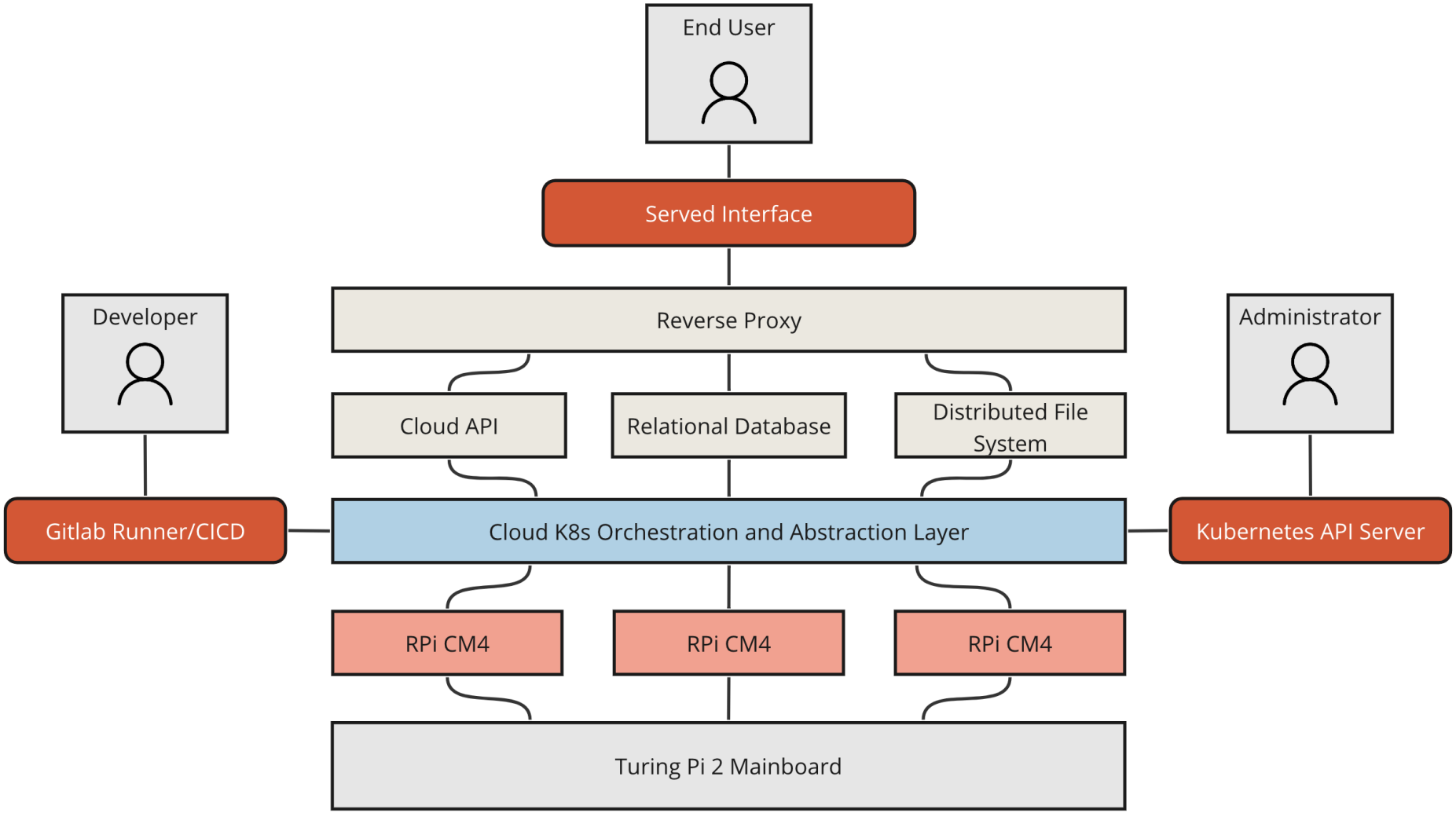
Requirements – Testing and Maintenance

- Support nonvolatile logging of both process-independent and process-dependent information
- Software and configuration will support updates via a CI/CD pipeline as deployed by GitLab
- All cloud-supporting software will have greater than 80% of lines covered by unit tests
- Each API endpoint will have a corresponding interface test guaranteeing basic functionality
- Have integration tests which validate interoperability of the DFS, cloud-supporting software, Docker, Kubernetes, and underlying hardware

Constraints

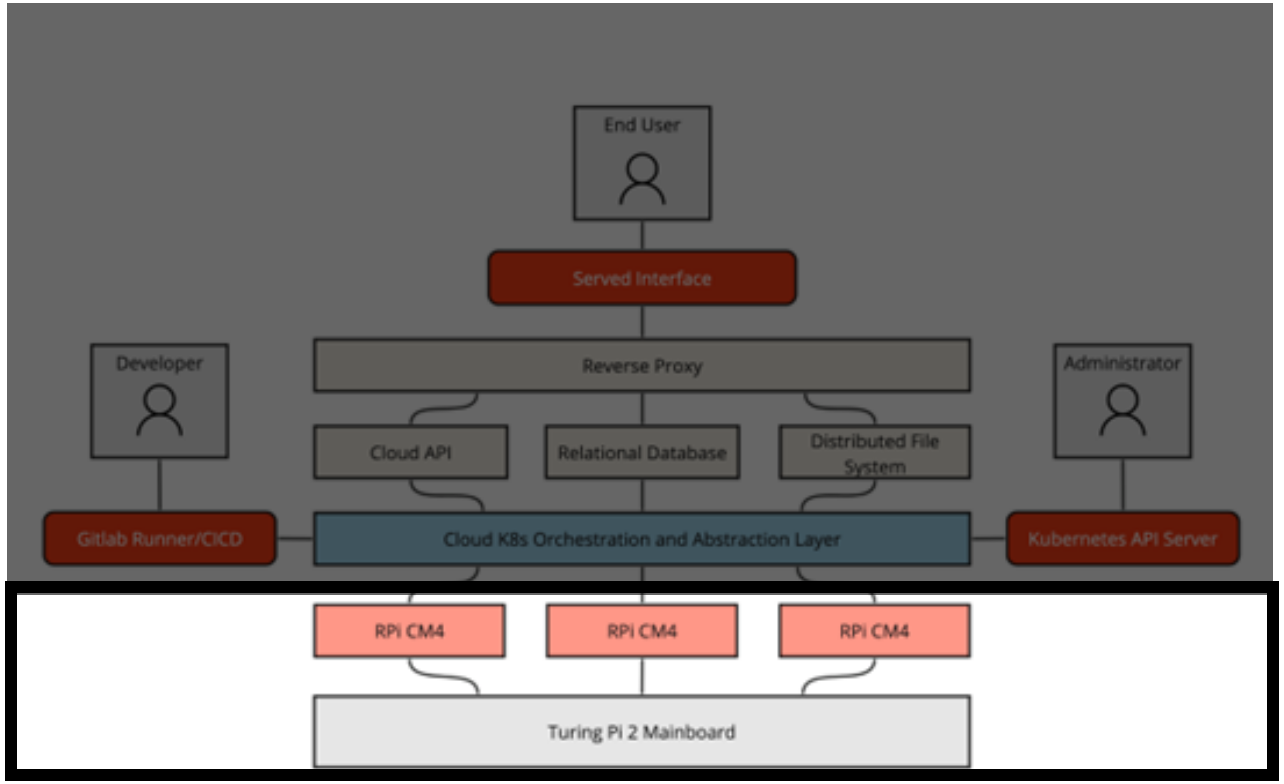
- The blob storage will support at least 10 users concurrently downloading and/or uploading
- The containerized video streaming application, when deployed, will support 3 simultaneous streams with a 3500 Kbps bitrate (1080p30)
- The blob storage will have an effective throughput of at least 15 Mbps upload and download
- Files of up to 32GB will be supported for upload and download by the blob storage

Conceptual Final Design Diagram



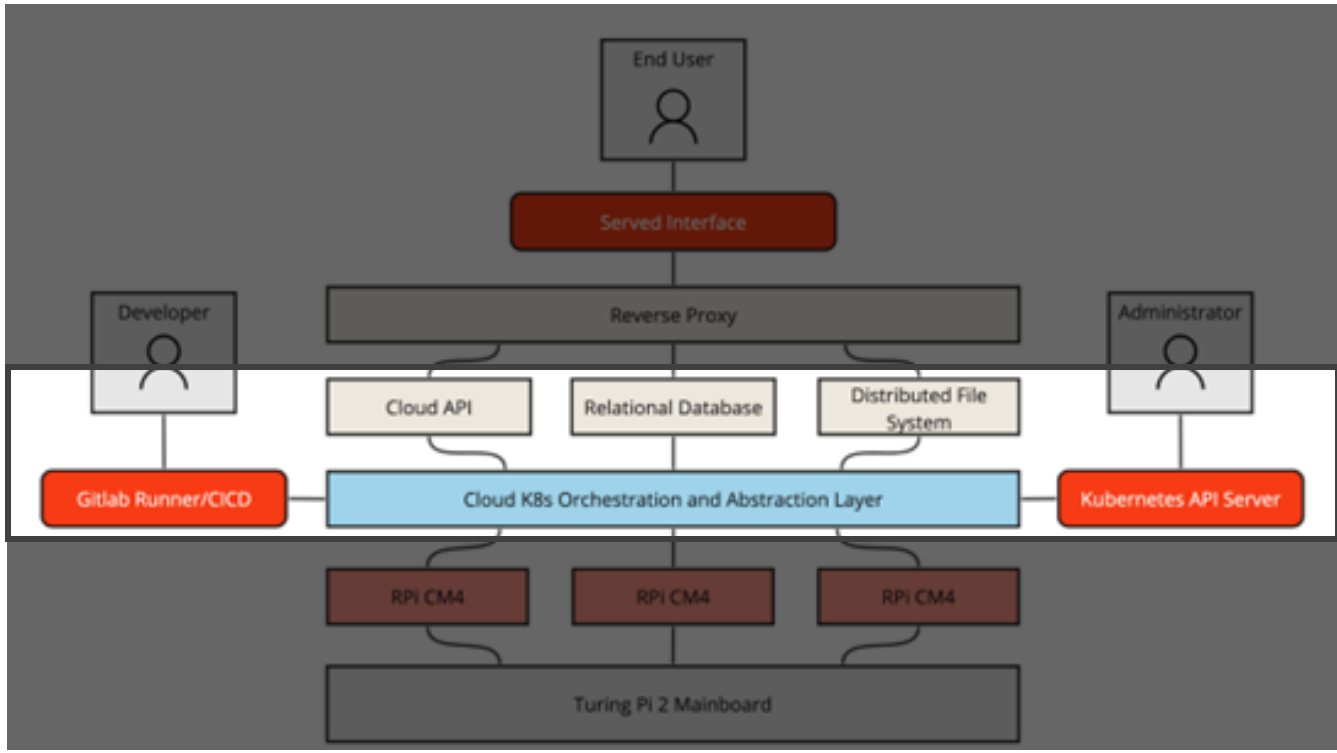
System Design (Hardware)

- Hardware (Turing Pi)
  - CM1 (RPi CM4)
    - HDMI, USB2.0, miniPCIe (converted to SATA)
  - CM2 (RPi CM4)
    - MiniPCIe (converted to SATA)
  - CM3 (RPi CM4)
    - SATA
  - CM4 (unpopulated)
    - USB3.0



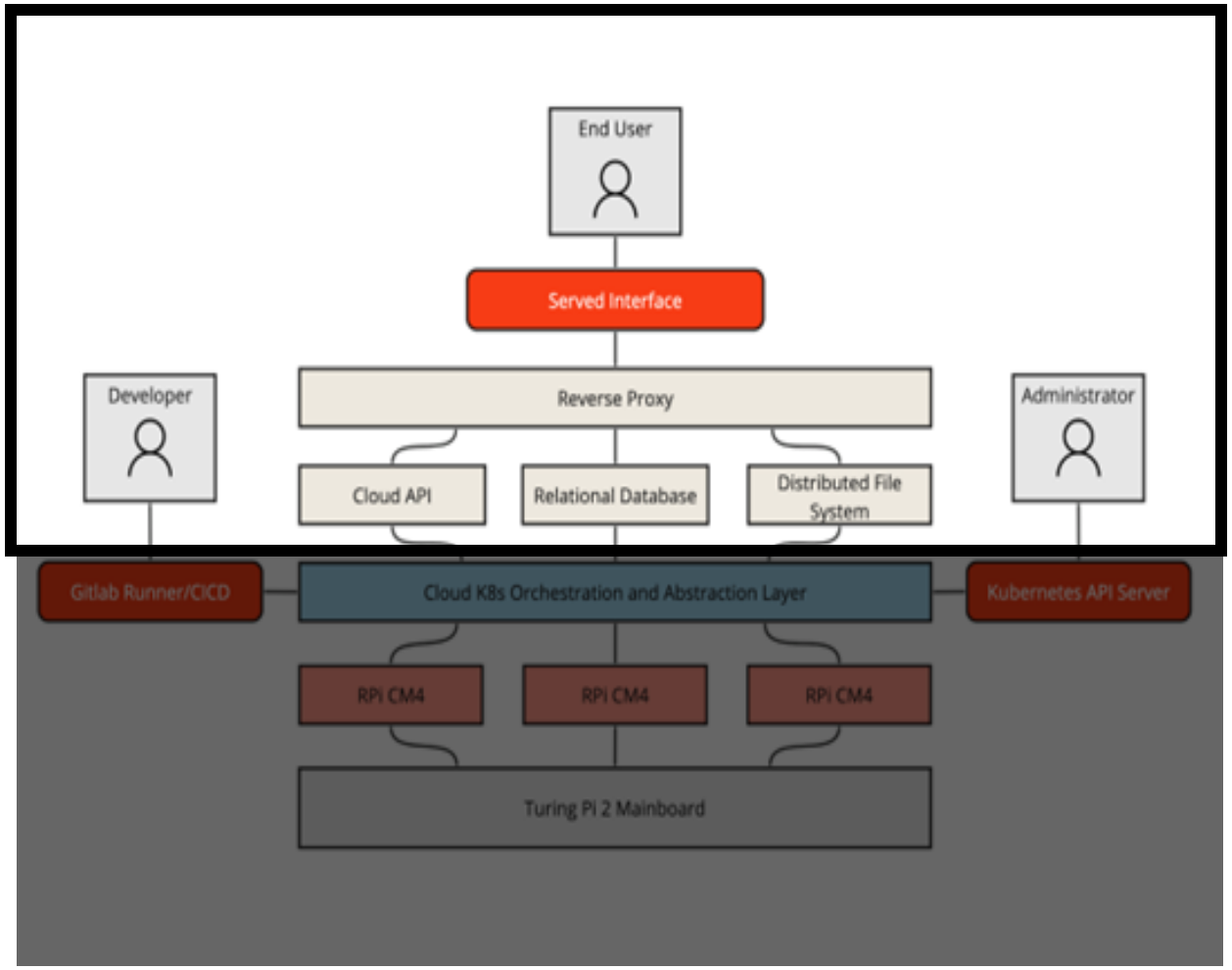
System Design (Hardware Abstraction)

- Backend Software
  - API for handling and dispatching requests (CRUD)
  - Kubegres for relational data storage
  - Rook-Ceph for distributed, hierarchical storage
  - Gitlab CI/CD and Kubernetes API for developer/admin interfacing



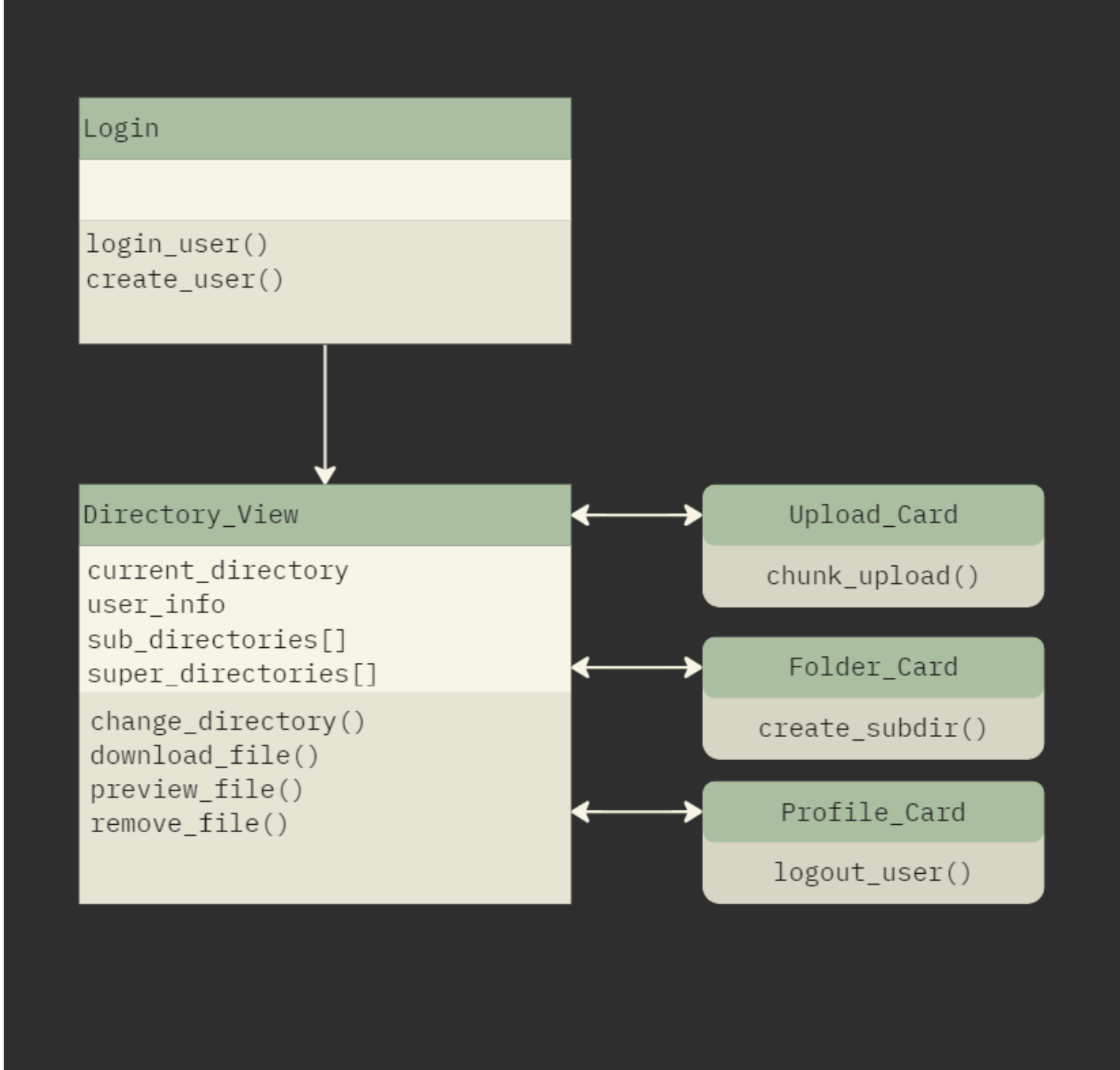
System Design (Software)

- Frontend-HAL integration
  - Ingress via Kubernetes Ingress
  - Reverse proxying with NGINX
  - Static file serving via NGINX
- Frontend
  - HTML+JS+CSS server-generated pages via React



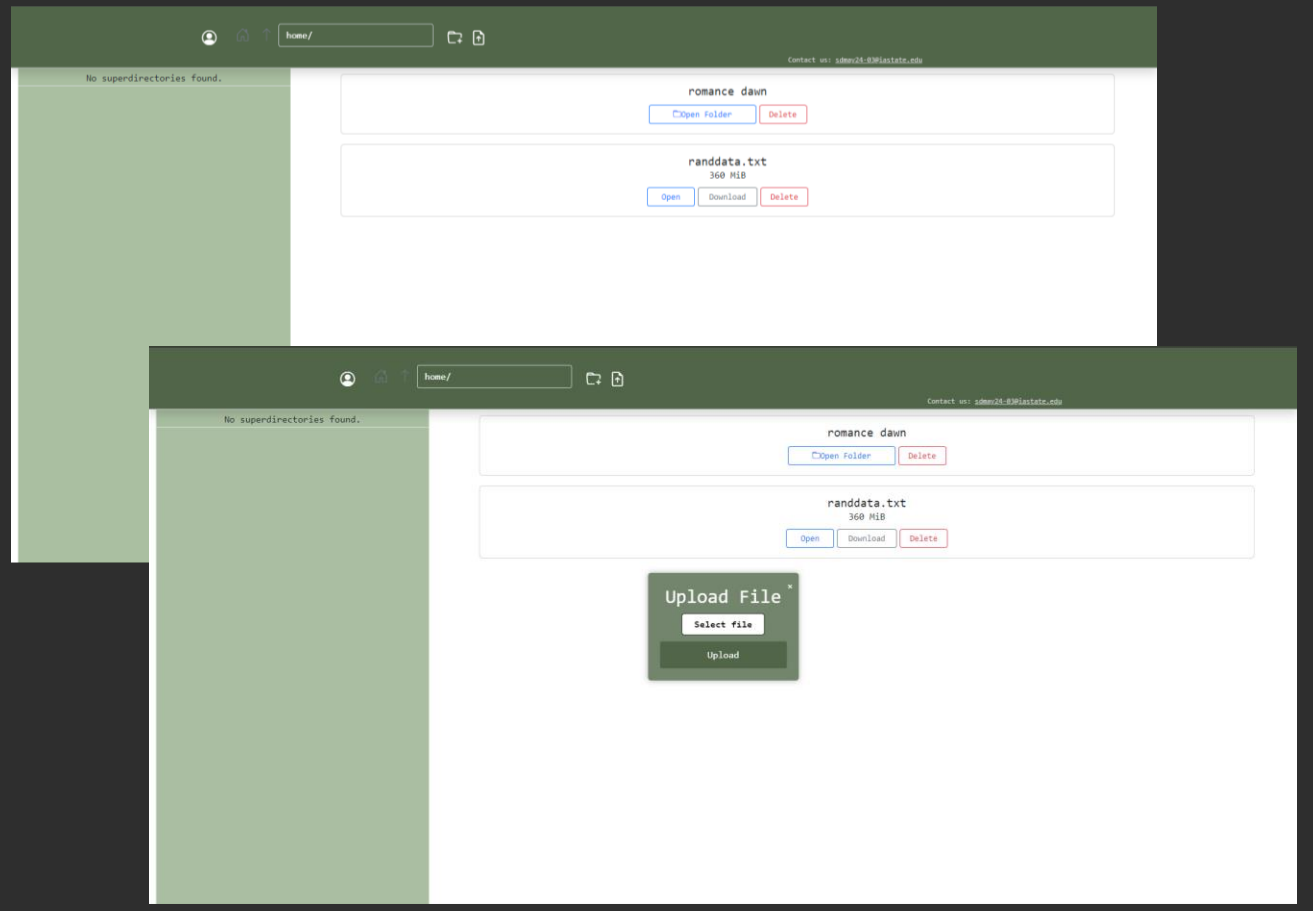
Webapp Screens Layout

- User friendly abstraction of Cloud Storage
  - Limited number of screens



GUI Design

- Clean interfaces to streamline usage
- Navigation familiar to users
- Pop-ups for supported functionality



Testing and Results

- Subsystems to test:
  - Private Cloud Stack
  - Implemented Software Application
  - DFS
  - Hardware connection to Internet
  - Overall management capabilities
- Testing Method:
  - Unit tests, and White Box Testing
  - Interface, Integration, and partial or whole system
- Maximum Tested Performance:
  - 328 Mbps transfer speed within the DFS storage
  - 28 Mbps between the DFS to local storage or to ethernet, from a single node
  - 32.8 Mbps from UI to DFS, through the entire system
  - 20 Watts average, 25-Watt peak under load

Demonstration