

Multifactor Authentication App

BY:

Tung Nguyen, duytung7@csu.fullerton.edu, 887001923
Gustavo Couto Vanin, gvanin@csu.fullerton.edu, 885517276
Saakshi Parikh, saakshi20@csu.fullerton.edu, 885147082
Andrew Kang, andrew.kang1209@csu.fullerton.edu, 886295328
Mike Thai, miket126@csu.fullerton.edu, 886590306
Daniel Le, PhuLe4108@csu.fullerton.edu, 887052900
Kiet Hoang, kiethoang1411@csu.fullerton.edu, 886579671

Responsibility

Kiet Hoang - Created server.py, client.py, endUserValidateOTP.py

Mike Thai - Created connection between 3 VMs, code testing

Daniel Le, Tung Nguyen, - Update code, check VMs working, Check slides

Saakshi Parikh, Andrew Kang - Slides Presenting

Gustavo Couto Vanin - created project proposal

Introduction

- In an era dominated by cloud computing, security is paramount. Traditional authentication methods no longer suffice in the face of sophisticated cyber threats. Our project introduces a Multi-Factor Authentication (MFA) App, designed to fortify security by requiring users to provide multiple forms of verification before accessing their accounts or data. Through this presentation, we will delve into the architecture and features of our MFA App, highlighting its significance in safeguarding digital systems against unauthorized access.

Problem statement

- In the age of digital transformation, the reliance on usernames and passwords for authentication has rendered systems vulnerable to increasingly sophisticated cyber threats. Single-factor authentication methods no longer provide adequate protection against unauthorized access to cloud-based services and sensitive data. As a result, there is a pressing need for a more robust security solution that can withstand evolving threats and safeguard the integrity of digital systems.

Solution

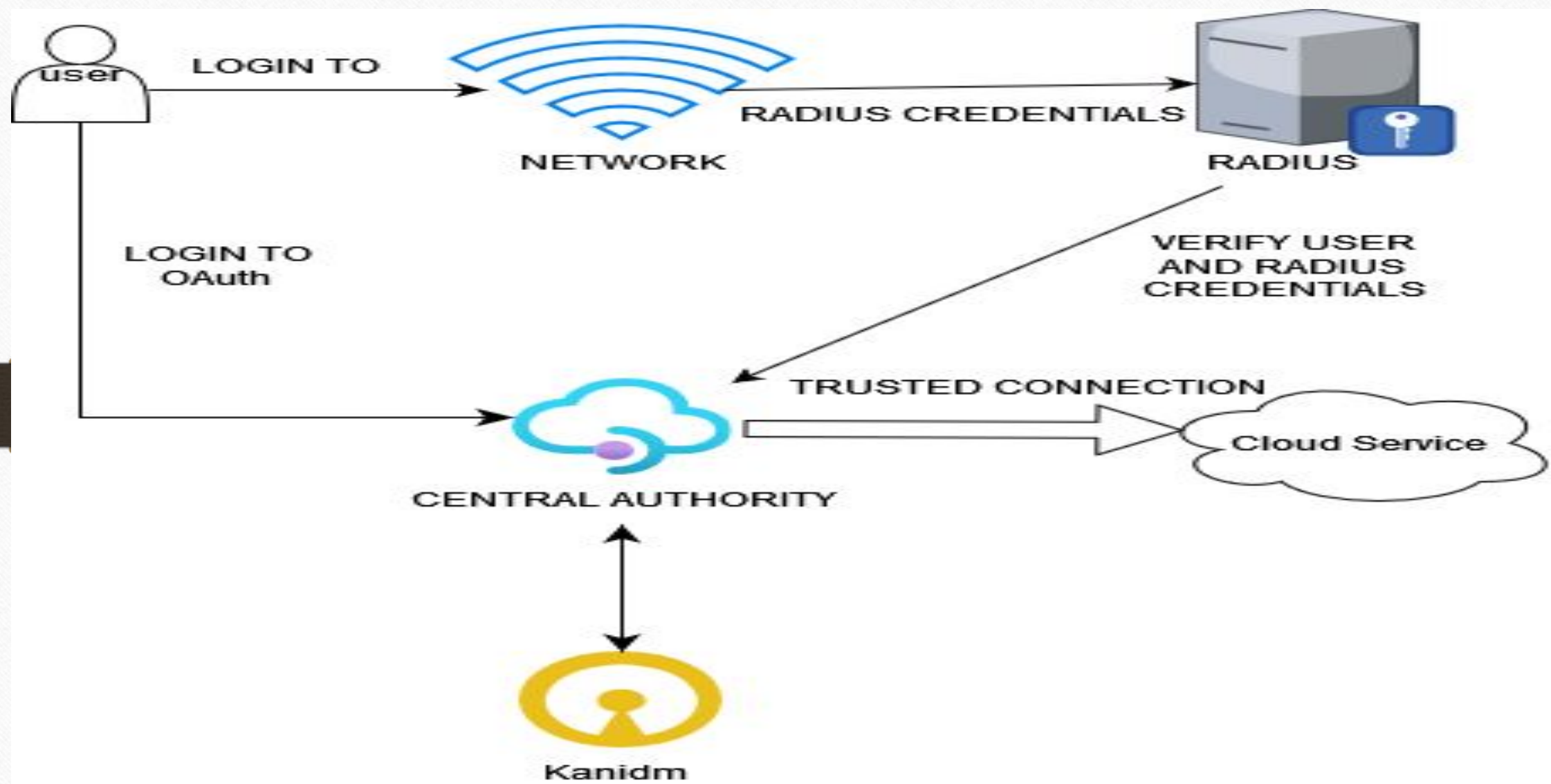
To solve this problem and combat the rise of cyber threats, a Multi-Factor Authentication application must be used. This application or tool requires the user to provide multiple forms of verifications before the user is granted the privilege to access their account or data. This ensures better security as it provides a stronger barricade or protection against unauthorized users, with the unauthorized users having to deal with many verification steps to be able to access your account. While a Single-Factor Authentication method brings a lot of risks as unauthorized users can easily access your account by stealing your username and password, a Multi-Factor Authentication method acts as a row of shields and gives cyber threats a difficult time trying to access your account and personal information.

Benefits of MFA

1. **Enhanced Security:** MFA adds extra layers of verification, reducing the risk of unauthorized access and data breaches.
2. **Protection Against Credential Theft:** By requiring multiple factors for authentication, MFA mitigates the risk of credential theft through methods like phishing.
3. **Compliance:** Helps organizations meet regulatory requirements by implementing strong authentication measures.
4. **Flexible Authentication Methods:** Supports various authentication factors, enhancing security without compromising user experience.
5. **Improved User Experience:** Offers convenient authentication options while maintaining security standards.
6. **Reduced Risk of Account Takeover:** Mitigates the risk of account takeover attacks by adding additional verification steps.
7. **Protection Against Insider Threats:** Helps prevent unauthorized access by insiders through additional authentication layers.
8. **Scalability and Adaptability:** Scalable and adaptable solutions suitable for organizations of all sizes and evolving security needs.

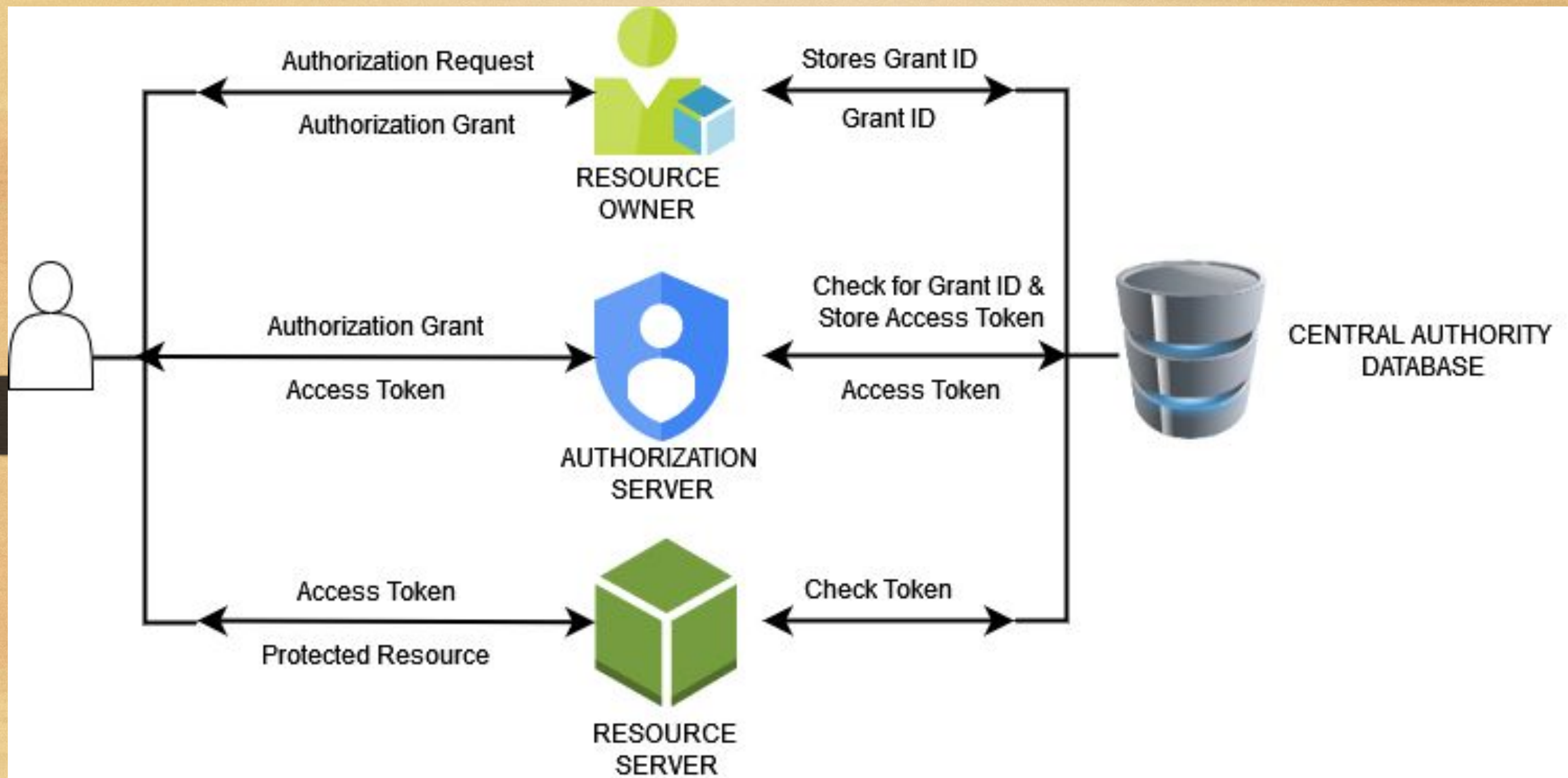
Our App

Our app utilizes a Multi-Factor Authentication method where the user connects to a central authority and the network at the same time. After the connection with the network, a radius checkpoint must be gone through to verify the user credentials and radius. After this verification, the radius checkpoint sends the verification to the central authority and the connection to the cloud service can be finalized. The central authority that is used is Kanidm, an open-source Multi-Factor Authentication algorithm.



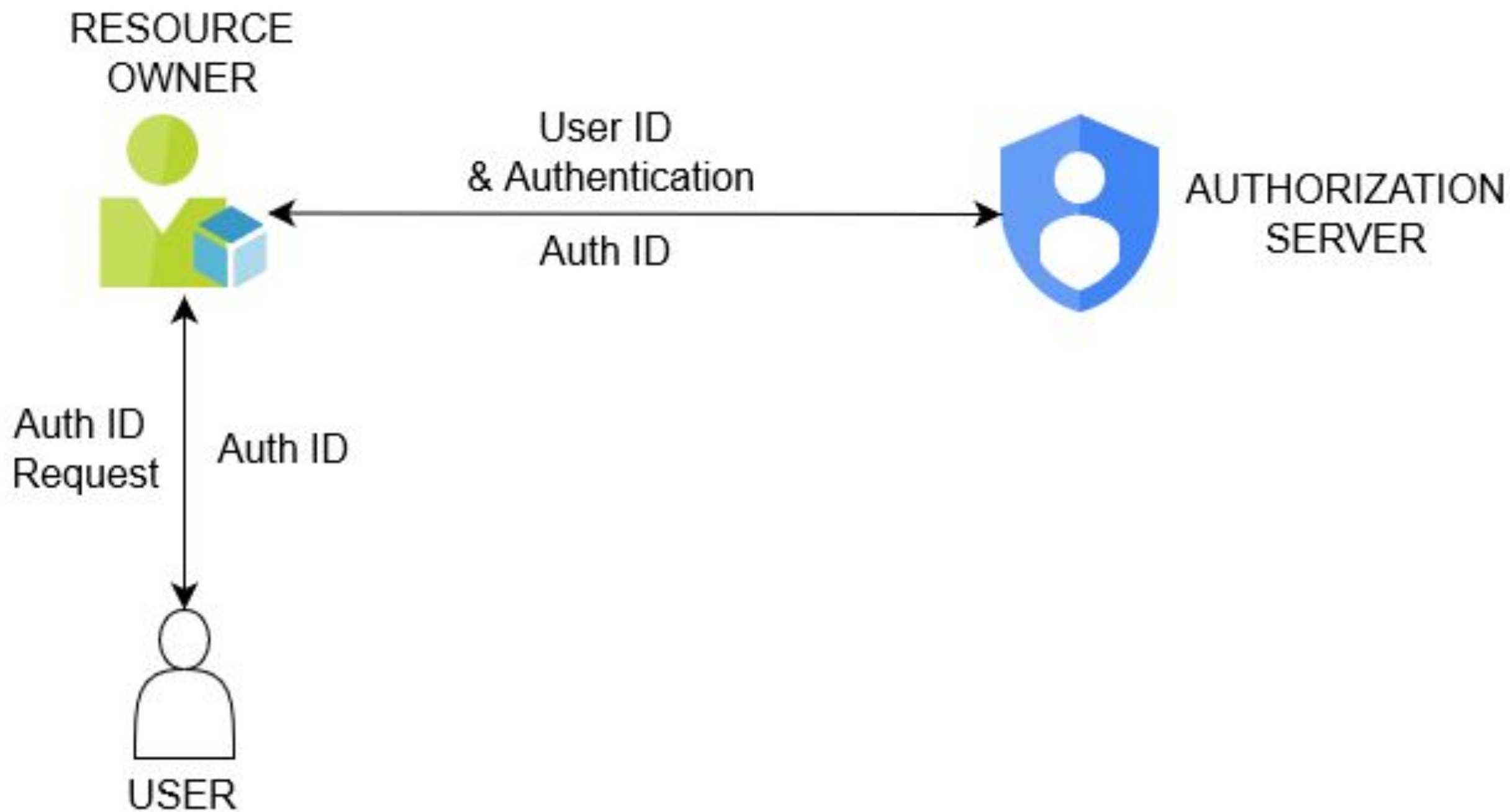
Central Authority Authentication

Central Authority Authentication is a pivotal security measure in cloud computing, ensuring robust user authentication before granting access to resources. When the user goes through an authentication process with the central authority, a three-step process is undertaken. These steps include connecting with the resource owner, connecting with the authorization server, and connecting with the resource server in order. As each step of the authentication is undergone, important items such as the Grant ID or the access token are all stored in the Central Authority Database.



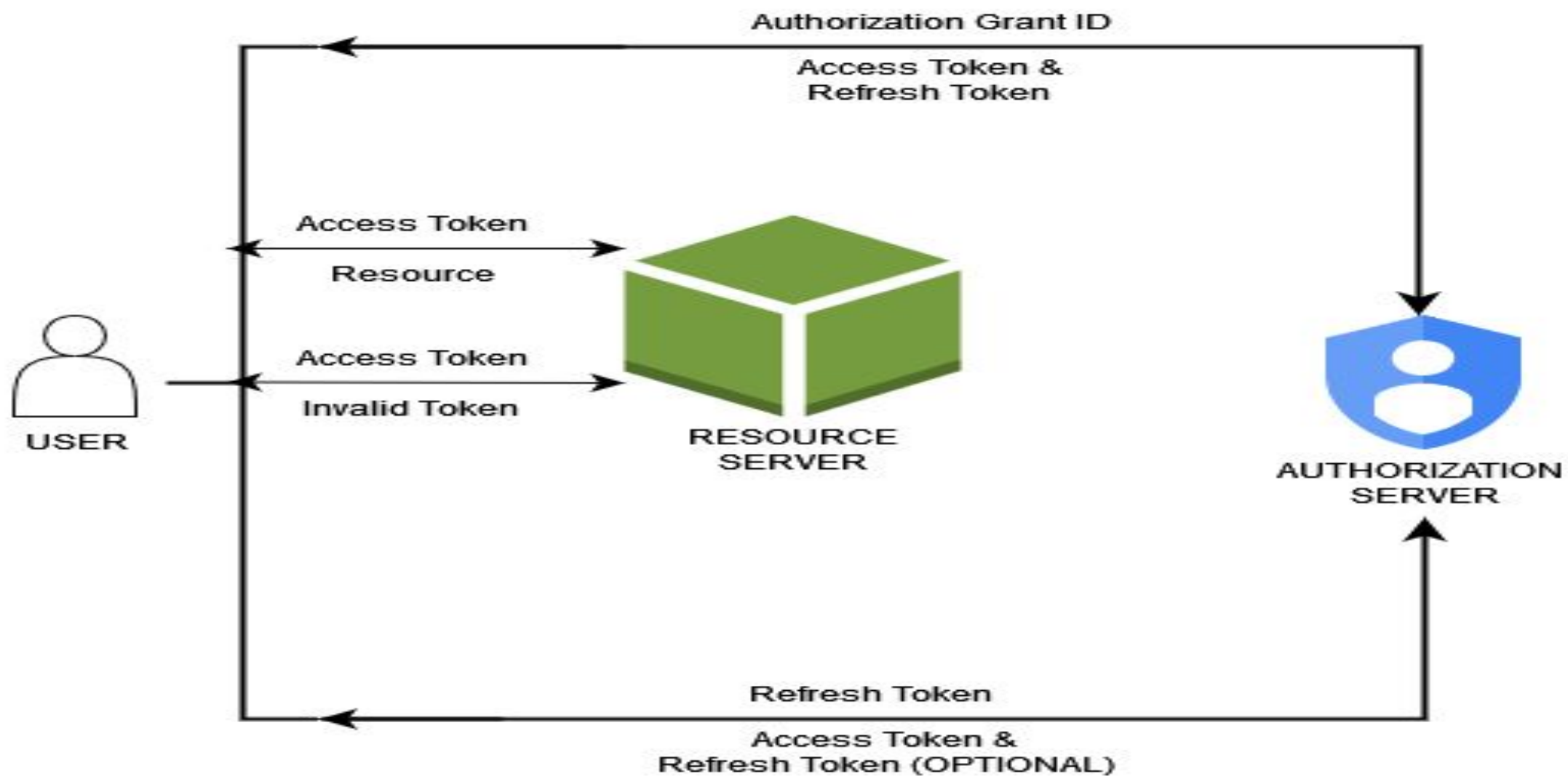
Resource Owner Authentication

1. **Initiation:** The Resource Owner begins by providing credentials (e.g., username/password) to the system.
2. **Verification:** The system verifies the provided credentials against stored records to authenticate the Resource Owner's identity. Optionally, multi-factor authentication (MFA) may be employed for added security.
3. **Authorization:** Once authenticated, the system proceeds to the Authorization Server for further validation.
4. **Validation:** The Authorization Server verifies the authenticity of the Resource Owner's request and provided credentials or tokens.
5. **Access Grant:** Upon successful validation, the Resource Owner is granted access to the requested resources, ensuring they have the necessary permissions.
6. **Token Management:** Temporary access tokens are issued to the Resource Owner, allowing them to access resources for a limited time. The system handles token expiration and renewal seamlessly.

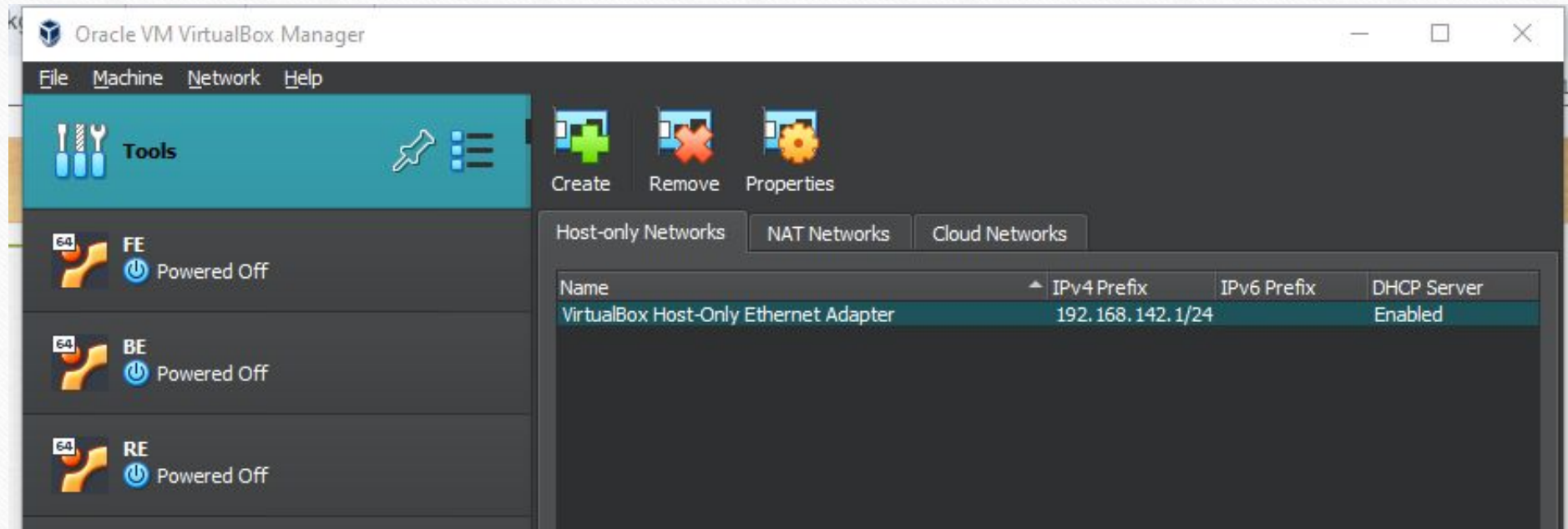


Authorization Server

The Authorization Server serves as the central gatekeeper in our cloud-based ecosystem, responsible for authenticating user requests and issuing temporary access tokens. It verifies user identities through rigorous checks, including proof of knowledge, location, and ownership. Once authenticated, it generates access tokens for resource access, which expire over time but can be seamlessly renewed using stored cookies. The Authorization Server plays a pivotal role in safeguarding our resources against unauthorized access while ensuring a streamlined user experience.



Virtual Network



Front-End

client.py

Adapter 1: Host-Guest communication

Adapter 2: Connect to Virtual Network



System

- Base Memory: 8192 MB
- Processors: 2
- Boot Order: Floppy, Optical, Hard Disk
- Acceleration: Nested Paging, KVM Paravirtualization

Display

- Video Memory: 16 MB
- Graphics Controller: VMSVGA
- Remote Desktop Server: Disabled
- Recording: Disabled

Storage

- Controller: IDE
- IDE Secondary Device 0: [Optical Drive] Empty
- Controller: SATA
- SATA Port 0: FE.vdi (Normal, 25.00 GB)

Audio

- Host Driver: Default
- Controller: ICH AC97

Network


- Adapter 1: Intel PRO/1000 MT Desktop (Bridged Adapter, Realtek PCIe 2.5GbE Family Controller)
- Adapter 2: Intel PRO/1000 MT Desktop (Host-only Adapter, 'VirtualBox Host-Only Ethernet Adapter')

USB


Back-End

server.py


Adapter 1: Connect to Virtual Network

**System**


Base Memory: 8192 MB
Processors: 2
Boot Order: Floppy, Optical, Hard Disk
Acceleration: Nested Paging, KVM Paravirtualization

**Display**


Video Memory: 16 MB
Graphics Controller: VMSVGA
Remote Desktop Server: Disabled
Recording: Disabled

**Storage**


Controller: IDE
IDE Secondary Device 0: [Optical Drive] Empty
Controller: SATA
SATA Port 0: BE.vdi (Normal, 25.00 GB)

**Audio**


Host Driver: Default
Controller: ICH AC97

**Network**

Adapter 1: Intel PRO/1000 MT Desktop (Host-only Adapter, 'VirtualBox Host-Only Ethernet Adapter')

**USB**

USB Controller: OHCI, EHCI



Resource Server

endUserValidateOTP.py

Adapter 1: Connect to the Internet

Adapter 2: Connect to Virtual Network



The screenshot displays the settings for a virtual machine, organized into several sections on the left and a preview window on the right.

- System**
 - Base Memory: 2048 MB
 - Boot Order: Floppy, Optical, Hard Disk
 - Acceleration: Nested Paging, KVM Paravirtualization
- Display**
 - Video Memory: 16 MB
 - Graphics Controller: VMSVGA
 - Remote Desktop Server: Disabled
 - Recording: Disabled
- Storage**
 - Controller: IDE
 - IDE Secondary Device 0: [Optical Drive] Empty
 - Controller: SATA
 - SATA Port 0: RE.vdi (Normal, 25.00 GB)
- Audio**
 - Host Driver: Default
 - Controller: ICH AC97
- Network**
 - Adapter 1: Intel PRO/1000 MT Desktop (NAT)
 - Adapter 2: Intel PRO/1000 MT Desktop (Host-only Adapter, 'VirtualBox Host-Only Ethernet Adapter')
- USB**
 - USB Controller: OHCI, EHCI
 - Device Filters: 0 (0 active)
- Shared folders**

On the right, a preview window shows a black screen with the text "RE" in white.

QUESTIONS??



Thank You