

Show code

⚙️ Initializing Phase 1 Multi-User Registration Infrastructure Environment...

Alias Name: [+ Register Handle Node](#)

- ✔ Success: Generated keys for '500100100'. Issued Public Handle: 0x510b999ecfa314f0ee8e4849dcfc53a066158c18
- ✔ Success: Generated keys for '500100200'. Issued Public Handle: 0x5823c36cc3a5732315b5dee06745d55abd27bca5
- ✔ Success: Generated keys for '500100300'. Issued Public Handle: 0x2ba05a3414fe5352166da483182757e8db92f0df
- ✔ Success: Generated keys for '500100400'. Issued Public Handle: 0x12c9e8e97198a2a25686b7c7954d61c1743733e6

Connected to
Python 3 Google Compute Engine backend
RAM: 1.00 GB/12.67 GB
Disk: 20.40 GB/107.72 GB

📁 Scalable Public Handle Identity Registry

Entity Alias	Public Cryptographic Address Handle (0x...)	Local Private Seed Key State
👤 500100100	0x510b999ecfa314f0ee8e4849dcfc53a066158c18	🔒 Secured in Client Node Memory
👤 500100200	0x5823c36cc3a5732315b5dee06745d55abd27bca5	🔒 Secured in Client Node Memory
👤 500100300	0x2ba05a3414fe5352166da483182757e8db92f0df	🔒 Secured in Client Node Memory
👤 500100400	0x12c9e8e97198a2a25686b7c7954d61c1743733e6	🔒 Secured in Client Node Memory

📊 Current Lineage Chain Block Depth: 4 entries | Latest Global Root: 5e903a3a600a5ff51ebe...

Show code

📄 **Outbound Compensation Requests Ledger (Awaiting Confirmation)**

Tx ID	Creditor (Alias)	Debtor (Alias)	Invoice Value	ATCUD Code	Current State
tx_64873944	0x12c9... (500100400)	0x510b... (500100100)	\$120,000	wweqewqew	
tx_0943da70	0x2ba0... (500100300)	0x12c9... (500100400)	\$120,000	fdsffdfhsfh	pending_debtor_confirmation
tx_5c9c6021	0x510b... (500100100)	0x5823... (500100200)	\$120,000	fgffd	pending_debtor_confirmation
tx_f0eb7602	0x5823... (500100200)	0x2ba0... (500100300)	\$120,000	sfdfdsfadfads	pending_debtor_confirmation
tx_32ef6cf4	0x12c9... (500100400)	0x510b... (500100100)	\$10,000	ffasdfadsfdas	pending_debtor_confirmation
tx_f888fccd	0x510b... (500100100)	0x2ba0... (500100300)	\$20,000	fadfadfdasfd	pending_debtor_confirmation
tx_5b4b32cc	0x12c9... (500100400)	0x2ba0... (500100300)	\$1	32r3r323r	pending_debtor_confirmation

Connected to
Python 3 Google Compute Engine backend
RAM: 1.00 GB/12.67 GB
Disk: 20.40 GB/107.72 GB

🔴 Dashboard: 500100400

Outbound Request Engine

Debtor Link:

Value (\$):

ATCUD ID:

Propose Lc...

🔔 Sign & Dispatch Claim

🔴 Dashboard: 500100300

Outbound Request Engine

Debtor Link:

Value (\$):

ATCUD ID:

Propose Lc...

🔔 Sign & Dispatch Claim

🔴 Dashboard: 500100100

Outbound Request Engine

Debtor Link:

Value (\$):

ATCUD ID:

Propose Lc...

🔔 Sign & Dispatch Claim

🔴 Dashboard: 500100200

Outbound Request Engine

Debtor Link:

Value (\$):

ATCUD ID:

Propose Lc...

🔔 Sign & Dispatch Claim

Show code

🔒 1. Execute DKG Ceremony (Generate Network Key)

🔒 [TEE Enclave] Initializing Secure Hardware Memory Boundary...

💛 Running Distributed Threshold Key Ceremony across active participant shards...

💛 **Network Threshold Public Key Online!**

Master Network Key Target Address: 0x_net_pub_f71e7d480b1185d6992426e9335f4dbf

Decryption keys are now split into shards. Debtors can now directly encrypt confirmations to this target.

Show code

✅ Success: Cryptographic match verified for tx_5c9c6021! Confirmation signed.

📄 Symmetrical Consensus Overview Grid

Tx ID	Creditor (Alias)	Debtor (Alias)	Invoice Value (Locked)	ATCUD Code	Proposed Line	Consensus Cap (L)	State
tx_64873944	0x12c9... (500100400)	0x510b... (500100100)	\$120,000	wweqewqew	\$10,000	\$300,000	pending_enclave_clearance
tx_0943da70	0x2ba0... (500100300)	0x12c9... (500100400)	\$120,000	fdsffdfhsfh	\$300,000	\$300,000	pending_enclave_clearance
tx_5c9c6021	0x510b... (500100100)	0x5823... (500100200)	\$120,000	fgffd	\$300,000	\$300,000	pending_enclave_clearance
tx_f0eb7602	0x5823... (500100200)	0x2ba0... (500100300)	\$120,000	sfdfsdfads	\$300,000	\$300,000	pending_enclave_clearance
tx_32ef6cf4	0x12c9... (500100400)	0x510b... (500100100)	\$10,000	ffasdfadsfdas	\$10,000	\$10,000	pending_enclave_clearance
tx_f888fccd	0x510b... (500100100)	0x2ba0... (500100300)	\$20,000	fadfadfdasfd	\$20,000	\$20,000	pending_enclave_clearance
tx_5b4b32cc	0x12c9... (500100400)	0x2ba0... (500100300)	\$1	32r3r323r	\$300,000	\$300,000	pending_enclave_clearance

Connected to
Python 3 Google Compute Engine backend
RAM: 1.00 GB/12.67 GB
Disk: 20.40 GB/107.72 GB

● Dashboard: 500100400

Inbound Verification Queue

No incoming bills awaiting confirmation.

● Dashboard: 500100300

Inbound Verification Queue

No incoming bills awaiting confirmation.

● Dashboard: 500100100

Inbound Verification Queue

No incoming bills awaiting confirmation.

● Dashboard: 500100200

Inbound Verification Queue

No incoming bills awaiting confirmation.

Show code

Show code

✅ CoinOR CBC Solver is installed and accessible at: /usr/bin/cbc

Show code

⚙️ Bootstrapping Secure FastAPI Payload Ingestion Endpoint...

📄 Ingest Payloads & Map Dynamic Gross Matrix

- 🔒 [TEE Enclave] FastAPI Payload Ingestion Gateway Active...
- 🔒 [TEE Enclave] Reassembling fractionalized threshold network keys...

🔑 --- STEP 4.1.A: ESTABLISHING CRYPTOGRAPHIC MATRIX HANDLES ---
Index [0] <==> Handle Address: 0x12c9e8e97198a2a25686b7... (500100400)
Index [1] <==> Handle Address: 0x2ba05a3414fe5352166da4... (500100300)
Index [2] <==> Handle Address: 0x510b999ecfa314f0ee8e48... (500100100)
Index [3] <==> Handle Address: 0x5823c36cc3a5732315b5de... (500100200)

🔒 --- STEP 4.1.B: DECRYPTING PAYLOADS & ASSIGNING GRID COORDINATES ---
-> Successfully Decrypted Tx #tx_64873944: Value \$120,000 | ATCUD Reference: wwqewqew
Mapping: Row [0] owes Column [2]
-> Successfully Decrypted Tx #tx_0943da70: Value \$120,000 | ATCUD Reference: fdsffdfhsfh
Mapping: Row [1] owes Column [0]
-> Successfully Decrypted Tx #tx_5c9c6021: Value \$120,000 | ATCUD Reference: fgffd
Mapping: Row [2] owes Column [3]
-> Successfully Decrypted Tx #tx_f0eb7602: Value \$120,000 | ATCUD Reference: sdfdsfadsfads
Mapping: Row [3] owes Column [1]
-> Successfully Decrypted Tx #tx_32ef6cf4: Value \$10,000 | ATCUD Reference: ffasdfadsfdas
Mapping: Row [0] owes Column [2]
-> Successfully Decrypted Tx #tx_f888fccd: Value \$20,000 | ATCUD Reference: fadfadfdasfd
Mapping: Row [2] owes Column [1]
-> Successfully Decrypted Tx #tx_5b4b32cc: Value \$1 | ATCUD Reference: 32r3r323r
Mapping: Row [0] owes Column [1]

📊 --- STEP 4.1.C: MEMORY RECONSTITUTED ENCLAVE MATRIX BOUNDARY ---

🔒 [GOD MODE] Enclave Protected Gross Exposure Matrix

This clear-text view represents total outstanding liabilities collected inside the hardware perimeter before optimization runs:

Creditor Handle \ Debtor Handle	500100400	500100300	500100100	500100200
500100400 (0x12c9...)	\$0.0	\$1.0	\$130,000.0	\$0.0
500100300 (0x2ba0...)	\$120,000.0	\$0.0	\$0.0	\$0.0
500100100 (0x510b...)	\$0.0	\$20,000.0	\$0.0	\$120,000.0
500100200 (0x5823...)	\$0.0	\$120,000.0	\$0.0	\$0.0

Total Gross Debt Volume parsed inside hardware perimeter: \$510,001.0

Connected to
Python 3 Google Compute Engine backend
RAM: 1.00 GB/12.67 GB
Disk: 20.40 GB/107.72 GB

> 📊 Model 5 Netting Engine & Global Proofs Verification

Show code

⚙️ Bootstrapping Scalable Pyomo Optimization Infrastructure...

🏠 Compute Model 5 & Generate Global Proofs

🔒 [TEE Enclave] Entering Protected Memory Optimization Subsystem...

Model 5 Optimization Complete. Extracting global verification invariants...

Connected to
Python 3 Google Compute Engine backend
RAM: 1.00 GB/12.67 GB
Disk: 20.40 GB/107.72 GB

⚙️ **Step 4.2 & 4.3: Model 5 Global Proof Verification Engine**

Verification Criterion Invariant Rule	Status Code
Verify original balances == new optimized balances	✓ TRUE (PASS)
Verify all new transactions are under consensus limits ($\text{netopt} \leq \text{lim}$)	✓ TRUE (PASS)
Verify all new individual CREDITS SUM are under previous credit sums	✓ TRUE (PASS)
Verify all new individual DEBTS SUM are under bilateral debt sums	✓ TRUE (PASS)

📢 Published Global Proof Root Hash (Appended to Chain):
0x_proof_root_cdc9e5cbf92a354165c5247a8d86b2903ea9eea11275930133b8c48945946ebc

📄 Publicly Viewable Network Cipher-Text Envelope:
🔒_net_enc_d8aaf10640df77259bc707fbe8e82907

> 👁 Enter God Mode: View Clear Matrix

Show code

👁️ Enter God Mode: View Clear Matrix

🟢 [GOD MODE] Optimized Netting Settlement Matrix Overview

This is the clear-text state calculated inside the secure hardware perimeter before egress row segregation occurs:

Connected to
Python 3 Google Compute Engine backend
RAM: 1.00 GB/12.67 GB
Disk: 20.40 GB/107.72 GB

Creditor Address Handle	Debtor [0] 500100400 0x12c9e8e9...	Debtor 500100300 0x2ba05a34...	500100100 0x510b999e...	Debtor [3] 500100200 0x5823c36c...
👤 500100400 (0x12c9e8...)	\$0.00	\$10,001.00	\$0.00	\$0.00
👤 500100300 (0x2ba05a...)	\$0.00	\$0.00	\$0.00	\$0.00
👤 500100100 (0x510b99...)	\$0.00	\$10,000.00	\$0.00	\$0.00
👤 500100200 (0x5823c3...)	\$0.00	\$0.00	\$0.00	\$0.00

📁 Concrete Settlement Payment Routes:

- ⚡ Handle 500100300 (0x2ba05a...) must pay \$10,001.00 to Handle 500100400 (0x12c9e8...).
- ⚡ Handle 500100300 (0x2ba05a...) must pay \$10,000.00 to Handle 500100100 (0x510b99...).

📊 Total Initial Gross Debt Claims:	\$510,001.00
📉 Total Optimized Net Settlement Flow:	\$20,001.00
🌟 Absolute System Liquidity Savings:	\$490,000.00

> 📁 Phase 4.4: Enclave Egress Distribution & Secure Client Retrieval Panels

Show code

🔒 1. Run Egress Row Segregation & Broadcast Ciphers

🖥️ Node Dashboard: 500100400 Egress Decryption Window Unwrap your secure clearing summary block using your isolated private credentials footprint. 🔑 Decrypt & Fetch Row	🖥️ Node Dashboard: 500100300 Egress Decryption Window Unwrap your secure clearing summary block using your isolated private credentials footprint. 🔑 Decrypt & Fetch Row	🖥️ Node Dashboard: 500100100 Egress Decryption Window Unwrap your secure clearing summary block using your isolated private credentials footprint. 🔑 Decrypt & Fetch Row	🖥️ Node Dashboard: 500100200 Egress Decryption Window Unwrap your secure clearing summary block using your isolated private credentials footprint. 🔑 Decrypt & Fetch Row
--	--	--	--

> Phase 4.4: Enclave Egress Distribution & Secure Client Retrieval Panels

Show code

```
⚙️ Bootstrapping Secure FastAPI Outbound Egress & Client Retrieval Systems...  
  🔒 1. Run Egress Row Segregation & Broadcast Ciphers  
  
🔒 [TEE Enclave] Initializing Secure Data Segregation and Outbound Packaging...  
  
⚙️ Processing 4 individual account matrix vectors...  
  - Vector row segregated & asymmetrically encrypted for: 500100400 (0x12c9e8e97198...)  
  - Vector row segregated & asymmetrically encrypted for: 500100300 (0x2ba05a3414fe...)  
  - Vector row segregated & asymmetrically encrypted for: 500100100 (0x510b999ecfa3...)  
  - Vector row segregated & asymmetrically encrypted for: 500100200 (0x5823c36cc3a5...)  
🟢 [TEE Egress Complete]: Outputs safely broadcasted to the network hash chain directory.  
🔒 [500100400 Window] Contacting network layer to retrieve block for handle: 0x12c9e8e97198a2a256...  
📡 Received encrypted cipher stream: 📦 cipher_aes_gcm_99adb27a49ab3c29a0051aeb0eb1ad9  
🔒 Injecting local private seed key [..1986f755cf..] to unlock unsegmented row...
```

Connected to
Python 3 Google Compute Engine backend
RAM: 1.00 GB/12.67 GB
Disk: 20.40 GB/107.72 GB

🛡️ Local Secure Vault: 500100400

Asymmetric Handle Key Verified: 0x12c9e8e97198a2a25686b7c7954d61c1743733e6

📊 Settled Internal Ledger Positions:

🟢 Your Cleared Credits Row (Amounts owed to you by others):

[0.0, 10001.0, 0.0, 0.0]

🔴 Your Cleared Debts Column (Amounts you must route out):

[0.0, 0.0, 0.0, 0.0]

🔗 Verified Parent Cryptographic Lineage Proof: 0x_proof_root_cdc9e5cbf92a354165c5247a8d86b2903ea9eea11275930133b8c48945946ebc

> Phase 4.4: Enclave Egress Distribution & Secure Client Retrieval Panels

Show code

⚙️ Bootstrapping Secure FastAPI Outbound Egress & Client Retrieval Systems...

🔒 1. Run Egress Row Segregation & Broadcast Ciphers

🔒 [TEE Enclave] Initializing Secure Data Segregation and Outbound Packaging...

⚙️ Processing 4 individual account matrix vectors...

- Vector row segregated & asymmetrically encrypted for: 500100400 (0x12c9e8e97198...)
- Vector row segregated & asymmetrically encrypted for: 500100300 (0x2ba05a3414fe...)
- Vector row segregated & asymmetrically encrypted for: 500100100 (0x510b999ecfa3...)
- Vector row segregated & asymmetrically encrypted for: 500100200 (0x5823c36cc3a5...)

🟢 [TEE Egress Complete]: Outputs safely broadcasted to the network hash chain directory.

🔒 [500100100 Window] Contacting network layer to retrieve block for handle: 0x510b999ecfa314f0ee...

👤 Received encrypted cipher stream: 📦 cipher_aes_gcm_cd6f3467f341cd97759d5e5c20bf2478

🔒 Injecting local private seed key [..57a8417b14..] to unlock unsegmented row...

Connected to
Python 3 Google Compute Engine backend
RAM: 1.00 GB/12.67 GB
Disk: 20.40 GB/107.72 GB

🛡️ **Local Secure Vault: 500100100**

Asymmetric Handle Key Verified: 0x510b999ecfa314f0ee8e4849dcfc53a066158c18

📊 **Settled Internal Ledger Positions:**

🟢 **Your Cleared Credits Row (Amounts owed to you by others):**

[0.0, 10000.0, 0.0, 0.0]

🔴 **Your Cleared Debts Column (Amounts you must route out):**

[0.0, 0.0, 0.0, 0.0]

🔐 Verified Parent Cryptographic Lineage Proof: 0x_proof_root_cdc9e5cbf92a354165c5247a8d86b2903ea9eea11275930133b8c48945946ebc

> 📁 Phase 4.4: Enclave Egress Distribution & Secure Client Retrieval Panels

[Show code](#)

⚙️ Bootstrapping Secure FastAPI Outbound Egress & Client Retrieval Systems...

🔒 1. Run Egress Row Segregation & Broadcast Ciphers

🔒 Injecting local private seed key [..e2902e237b..] to unlock unsegmented row...

Connected to
Python 3 Google Compute Engine backend
RAM: 1.00 GB/12.67 GB
Disk: 20.40 GB/107.72 GB

> 📁 Phase 4.4: Enclave Egress Distribution & Secure Client Retrieval Panels

Show code

⚙️ 📊 Settled Internal Ledger Positions

🟢 Your Cleared Credits Row (Amounts owed to you by others):

🔒 1. Run Egress Row Segregation & Broadcast Ciphers

[0.0, 0.0, 0.0, 0.0]

🔒 [TEE Enclave] Initializing Secure Data Segregation and Outbound Packaging...

🔴 Your Cleared Debts Column (Amounts you must route out):

⚙️ Processing 4 individual account matrix vectors...

[10001.0, 0.0, 0.0, 10000.0, 0.0]

Vector row segregated & asymmetrically encrypted for: 500100400 (0x12c9e8e97198...)

Vector row segregated & asymmetrically encrypted for: 500100300 (0x2ba05a3414fe...)

🔒 Vector row segregated & asymmetrically encrypted for: 500100100 (0x510b999ecfa31127f930133b8c48945946ebc)

Vector row segregated & asymmetrically encrypted for: 500100200 (0x5823c36cc3a5...)

🟢 [TEE Egress Complete]: Outputs safely broadcasted to the network hash chain directory.

🔒 [500100200 Window] Contacting network layer to retrieve block for handle: 0x5823c36cc3a5732315...

📡 Received encrypted cipher stream: 📄 cipher_aes_gcm_f2660c8ef9d96b2ecf27a4dca4cc8169

🔒 Injecting local private seed key [..a124b8cde0..] to unlock unsegmented row...

🛡️ Local Secure Vault: 500100200

Asymmetric Handle Key Verified: 0x5823c36cc3a5732315b5dee06745d55abd27bca5

📊 Settled Internal Ledger Positions:

🟢 Your Cleared Credits Row (Amounts owed to you by others):

[0.0, 0.0, 0.0, 0.0]

🔴 Your Cleared Debts Column (Amounts you must route out):

[0.0, 0.0, 0.0, 0.0]

🔒 Verified Parent Cryptographic Lineage Proof: 0x_proof_root_cdc9e5cbf92a354165c5247a8d86b2903ea9eea11275930133b8c48945946ebc

> 🛡️ Pre-Flight Security Guardrail Invariant Circuit

Show code

⚙️ Bootstrapping Symmetrical Invariant Perimeter Circuits...

🛡️ Run Pre-Flight Security Guardrails