

# Beyond Bitcoin & FORTH (BFF001)

## Hilbert Hotel with Infinite Cabinets

This is a series to recap our work on FORTH and the next generation Internet hardware, software & networking technologies, starting with David Hilbert's theory of infinity, as illustrated by the Grand Hotel with (a twist that is) rooms containing infinite cabinets, representing ownerships of digital assets (in the cabinets) using hash numbers. The Hash Filesystem replaces database servers, simplifying server node implementations based on FORTH architecture.

Liang Ng, January 2026

[omnixtar.github.io/svfig](https://omnixtar.github.io/svfig)

<https://www.linkedin.com/in/liang-nq>

- *alternative solution to Bitcoin and blockchain technologies for managing ownerships of digital assets*

**“Who owns digital assets?”**

**“Buy, sell, transfer, royalties?”**

- *Omnihash & DJSON*  
*Decentralised JSON*



**Hidden Services Manager**

Hidden Services Manager

These are the local services provided by your router. By default, most services are only accessible from "Clients".

**Status Messages**

Starting tunnel A01 Omni\*web HTTPS...  
Error listening for connections on /127.0.0.1 port 4444  
Stopping client HTTP Proxy on 127.0.0.1:4444  
Client error for 127.0.0.1:4444, check logs  
BLOCK timestamp

**Pay-to-PubKey-Hash (Pay-to-Public-Key-Hash, P2PKH)** is the basic form of making a transaction on the [Bitcoin network](#). Transactions that pay P2PKH scripts that are resolved by sending the public key and a digital signature from the private key.

The ScriptPub

[https://bitcoinwiki.org/wiki/  
pay-to-pubkey-hash](https://bitcoinwiki.org/wiki/pay-to-pubkey-hash)

#### Table of C

1. Pay-to-Pubkey Hash
  - 1.1. Pay-to-PubKey-Hash Review
  - 1.2. Pay-to-PublicKey Hash Example
  - 1.3. See also
  - 1.4. References

## FORTH / (Phoscript)

- *virtual machine in extreme environments*
- *abstraction of source code for monetisation*

publishing the public key only when the coins are spent (and assuming that address is rendered ineffective.



```
hongwu@hongwu-Latitude-5480: ~          x  hongwu@hongwu-Latitude-5480: /var/www/d...  x  hongwu@hongwu-Latitude-5480: /var/www/d...  x  hongwu@hongwu-Latitude-5480: /var/www/d...
-rw-r--r-- 1 www-data www-data  597 Jan  22 15:12 'Hzn18Vs0lg=='
-rw-r--r-- 1 www-data www-data  597 Jan  22 15:10 'Hhj3wWn0OA=='
-rw-r--r-- 1 www-data www-data  618 Jan  22 15:09 'GUD2koxhKQ=='
hongwu@hongwu-Latitude-5480:/var/www/dmeta/o/auth/Graph/dmeta$ ls -lt BYRAtdTkFw\=\=\= | head
total 52
-rw-r--r-- 1 www-data www-data  618 Jan  22 17:32 Gg-6Vx8ANQ==
-rw-r--r-- 1 www-data www-data  597 Jan  22 15:51 FcSSP_buqA==
-rw-r--r-- 1 www-data www-data  597 Jan  22 15:21 E9wHrPZbIA==
-rw-r--r-- 1 www-data www-data 1150 Jan  22 15:16 CFkif5lnhg==
-rw-r--r-- 1 www-data www-data  597 Jan  22 15:16 HBJa8olJPw==
-rw-r--r-- 1 www-data www-data 1150 Jan  22 15:16 ASE2NFkzjA==
-rw-r--r-- 1 www-data www-data  597 Jan  22 15:16 EpJddpwiPA==
-rw-r--r-- 1 www-data www-data  618 Jan  22 15:16 Hk27YiUtuA==
-rw-r--r-- 1 www-data www-data 1150 Jan  22 15:15 CemDL02Hfw==
hongwu@hongwu-Latitude-5480:/var/www/dmeta/o/auth/Graph/dmeta$ cat BYRAtdTkFw\=\=\= /Gg-6Vx8ANQ\=\=\= ; echo
{"t": "2026-01-22T09:32:39.065Z", "m": "from_parent_20260110_button", "c": "<h4 id=\"-solution-for-truly-decentralised-internet\">\u2013 Solution for Truly Decentralised Internet<a class=\"anchorjs-link\" href=\"#-solution-for-truly-decentralised-internet\" aria-label=\"Anchor\" data-anchorjs-icon=\"\ue9cb\" style=\"font: 1em / 1 anchorjs-icons; padding-left: 0.375em;\"></a></h4>", "x": "like", "u": {"href": "https://omnixtar.github.io/I3P/", "origin": "https://omnixtar.github.io", "protocol": "https:", "host": "omnixtar.github.io", "hostname": "omnixtar.github.io", "port": "", "pathname": "/I3P/", "search": "", "hash": ""}}
```

hongwu@hongwu-Latitude-5480: /var/www/dmeta/o/auth/Graph/dmeta

```
total 12
-rw-r--r-- 1 www-data www-data 597 Jan 22 15:12 'Hzn18Vs0lg=='
-rw-r--r-- 1 www-data www-data 597 Jan 22 15:10 'Hhj3wWn00A=='
-rw-r--r-- 1 www-data www-data 618 Jan 22 15:09 'GUD2koxhKQ=='
hongwu@hongwu-Latitude-5480:/var/www/dmeta/o/auth/Graph/dmeta$ cat BYRAtdTkFw\=\\=/Hzn18Vs0lg\\=\\= ;echo
{"t": "2026-01-22T07:12:52.930Z", "m": "like", "c": "<h4 id=\"-solution-for-truly-decentralised-internet\">\u2013 Solution for Truly Decentralised Internet<a class=\"anchorjs-link\" href=\"#-solution-for-truly-decentralised-internet\" aria-label=\"Anchor\" data-anchorjs-icon=\"\ue9cb\" style=\"font: 1em / 1 anchorjs-icons; padding-left: 0.375em;\"></a></h4>", "x": "search", "u": {"href": "https://omnixtar.github.io/I3P/"}, "origin": "https://omnixtar.github.io", "protocol": "https:", "host": "omnixtar.github.io", "hostname": "omnixtar.github.io", "port": "", "pathname": "/I3P/", "search": "", "hash": ""}}
hongwu@hongwu-Latitude-5480:/var/www/dmeta/o/auth/Graph/dmeta$ ls -lt|head
```

Hash Filesystem (HFS): Directory name is hash of public key (User Identifier)

*Dual chained properties replace many (all?) database functionalities?*

- chained strings: hashcodes can be chained (concatenated) with strings
- chained functions (hash of a string is a string)

$$H(s) \rightarrow s, H(H(s)) \rightarrow s, \dots, H^n(s) \rightarrow s$$

-rw-r--r-- 1 www-data www-data 97 Jan 16 15:58 1omyUqd8

hongwu@hongwu-Latitude-5480:/var/www/dmeta/o/auth/Graph/dmeta\$

hongwu@hongwu-Latitude-5480: /var/www/dmeta/o/auth/Graph/dmeta

```
hongwu@hongwu-Latitude-5480: ~      x  hongwu@hongwu-Latitude-5480: /var/www/d...  x  hongwu@hongwu-Latitude-5480: /var/www/d...  x  hongwu@hongwu-Latitude-5480: /var/www/d...  x  hongwu@hongwu-Latitude-5480: /var/www/d...
total 12
-rw-r--r-- 1 www-data www-data 597 Jan 22 15:12 'Hzn18Vs0lg=='
-rw-r--r-- 1 www-data www-data 597 Jan 22 15:10 'Hhj3wWn00A=='
-rw-r--r-- 1 www-data www-data 618 Jan 22 15:09 'GUD2koxhKQ=='
hongwu@hongwu-Latitude-5480:/var/www/dmeta/o/auth/Graph/dmeta$ cat BYRAtdTkFw\=\=\=/Hzn18Vs0lg\=\=\= ; echo
{"t": "2026-01-22T07:12:52.930Z", "m": "like", "c": "<h4 id=\"-solution-for-truly-decentralised-internet\">\u2013 Solution for Truly Decentralised Internet<a class=\"anchorjs-link\" href=\"#-solution-for-truly-decentralised-internet\" aria-label=\"Anchor\" data-anchorjs-icon=\"\ue9cb\" style=\"font: 1em / 1 anchorjs-icons; padding-left: 0.375em;\"></a></h4> ", "x": "search", "u": {"href": "https://omnixtar.github.io/I3P/", "origin": "https://omnixtar.github.io", "protocol": "https:", "host": "omnixtar.github.io", "hostname": "omnixtar.github.io", "port": "", "pathname": "/I3P/", "search": "", "hash": ""}}
hongwu@hongwu-Latitude-5480:/var/www/dmeta/o/auth/Graph/dmeta$ ls -lt|head
```

Omnihash: hashCode representing ANY kind of digital assets (Omni-)  
Omni\*Web: a truly decentralised web ecosystem built using Omnihash  
Omni – widely used by big corps  
\* = everything in programming but not a valid character by law  
Omni\*Web: a legally invalid name that cannot be prosecuted (theoretically)

- [REDACTED] - [REDACTED] - [REDACTED] - 1 www-data www-data 97 Jan 16 15:58 1emvJUD8

hongwu@hongwu-Latitude-5480:~/var/www/dmeta/o/auth/Graph/dmeta\$

## Dual chained properties replace many (all?) database functionalities?

- chained strings: hashcodes can be chained (concatenated) with strings
- chained functions (hash of a string is a string)

$H(s) \rightarrow s, H(H(s)) \rightarrow s, \dots, H^n(s) \rightarrow s$

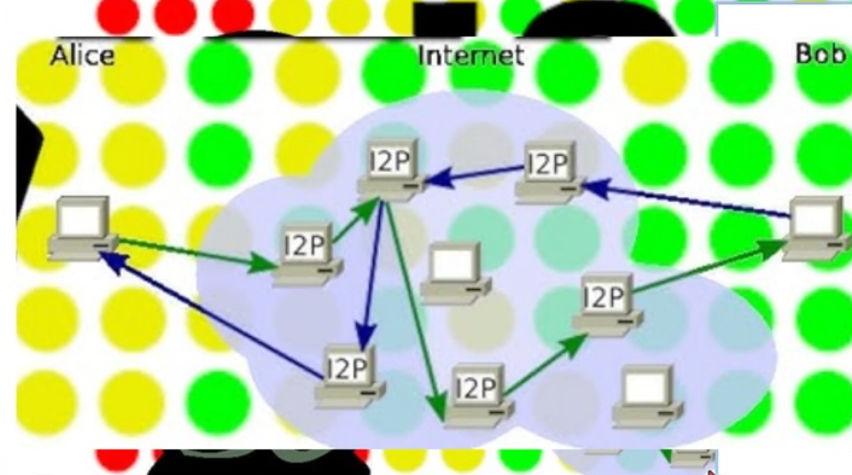
Example: `h_f0` in ajax response

Reason for lightweight “FORTH card computer” implementation →  
- so “database-less”

MDEtMjJUMDk6MzA6MDIuMzM5WiIsmhfZjAiOjIYkNUMF  
 BtZJRP0ifQ==", "b64d:", "jd:", "array:", "swap:", "ap:", "2dup:  
 ", "acdwa:", "over:", "HF0", "bv:", "a\_cmd", "gbv:", "HF0", "i:", "0",  
 "i:", "h\_f0", "usa:", "je:", "dup:", "hbp", "2", "pick:", "array:", "swap  
 :, "r:", "apk:", "isodt:", "t:", "apk:", "dup:", "je:", "dup:", "hbp", "2", "pick:",  
 "swap:", "h:", "apk:", "gbv:", "HF0", "i:", "swap:", "ap:", "je:",  
 "ON", "ECHO", "bv:", "dup:", "dup:", "ec:", "hbp", "9", "pick:", "sw  
 ap:", "2", "msss:", "w:", "j"], 1 => 'jd:', 2 => 'cdw:', 3 => 'am:', 4  
 => 'lcdw:', 5 => 'B\_AUTH1', ), 1 => 'Graph/dmeta/  
 BYRAtdTkFw==', 2 => array ( 0 => array ( 'a\_cmd' => array ( 0 => 's:', 1 => 't\_f0' => '2026-01-22T09:30:02.339Z',  
 'h\_f0' => 'HbCT0Pm\_2Q==', ), ), 3 => 'Graph/dmeta/  
 BYRAtdTkFw==', )  
 [{"a\_cmd":  
 ["s:", "t\_f0": "2026-01-22T09:30:02.339Z", "h\_f0": "HbCT0Pm\_2Q=="], "r": "Graph/dmeta/  
 BYRAtdTkFw==", "t": "2026-01-22T17:30:02.444+0000", "h": "BNQgJcsdnA=="}]  
  
 Don't allow 866196a35a9f.ngrok-free.app to prompt  
 you again  
  
 1602  
 fgl\_s  
 ["SE  
 4 =>  
 fgl\_s 345 < 2 > array ( 0 => array ( 0 => '{"B AUTH":  
 "v": "2026-01-22T17:30:02.444+0000", "h": "BNQgJcsdnA=="}')

OK

# INVISIBLE INTERNET PROJECT



ni\*Web



Omni\*Web

Training

Taichi-

[https://www.youtube.com/watch?v=-3CO5bG\\_q8&t=56s](https://www.youtube.com/watch?v=-3CO5bG_q8&t=56s)

Metaprogramming  
Fire

Infrastructure  
Earth

地

Digital Assets  
Water

水

<https://omnixtar.github.io/oxw>

Omni\*Web: Set Up YrOwn GLOBALLY ACCESSIBLE Web Server w/o Domain Name System in 4 Simple Steps (I2P)



MI4 Metaprogramming in Forth  
56 subscribers



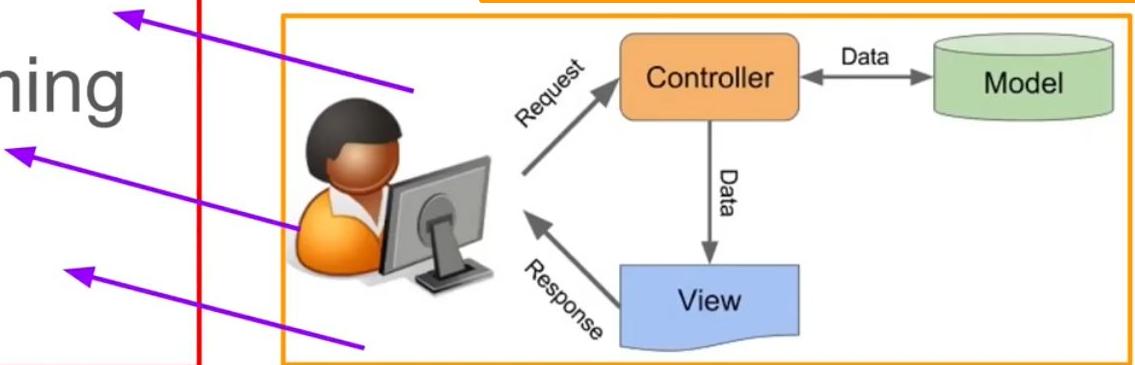
ONCE YOU UNWRAP,  
YOU CAN'T GO BACK



# Crypto- Meta- Programming (CMP)

Omni\*Web

[https://www.youtube.com/  
watch?v=P\\_M3PVn9J7I](https://www.youtube.com/watch?v=P_M3PVn9J7I)



Omni\*Web: Crypto-Metaprogramming (CMP) as alternative to Model-View-Controller; towards Metanarchy



MI4 Metaprogramming in Forth  
56 subscribers



0



Share



Download



Clip



All

Computer programming

Computer Science



Singapore's Tharman Destroys  
an Arrogant BBC Host





## Bitcoin: Technical Concepts

Bech32  
Blockchain Diagram

Bitcoin Encryption

Creating forks

Bitcoin mining

Blockchain

Bitcoin Improvement Proposals

Pay-to-Script Hash

Proof of Keys

UTXO

User Activated Soft Fork

OmniBOLT

Blockchain (database)

Segregated Witness

Lightning Network

Hashed Timelock Contracts

NSequence

Bitcoin Emission

Block timestamp

**Pay-to-PubKey-Hash (Pay-to-Public-Key-Hash, P2PKH)** is the basic form of making a [transaction](#) and is the most common form of transaction on the [Bitcoin network](#). Transactions that pay to a [Bitcoin address](#) contain P2PKH scripts that are resolved by sending the public key and a digital signature created by the corresponding [private key](#).

The ScriptPubKey and ScriptSig for a transaction is shown below:

### Table of Contents



1. Pay-to-Pubkey Hash
  - 1.1. Pay-to-PubKey-Hash Review
  - 1.2. Pay-to-PublicKey Hash Example
  - 1.3. See also
  - 1.4. References

<https://bitcoinwiki.org/wiki/pay-to-pubkey-hash>

## Pay-to-PubKey-Hash Review

Two types of payment are referred as P2PK (pay to public key) and P2PKH (pay to public key hash).

[Satoshi](#) later decided to use P2PKH instead of P2PK for two reasons:

Elliptic Curve [Cryptography](#) (the cryptography used by your public key and [private key](#)) is vulnerable to a modified Shor's [algorithm](#) for solving the discrete logarithm problem on elliptic curves. In plain English, it means that in the future a quantum computer might be able to retrieve a private key from a public key. By publishing the public key only when the coins are spent (and assuming that addresses are not reused), such attack is rendered ineffective.

With the hash being smaller (20 bytes) it is easier to print and easier to embed into small storage mediums like





- *alternative solution to Bitcoin and blockchain technologies for managing ownerships of digital assets*

**“Who owns digital assets?”**

**“Buy, sell, transfer, royalties?”**

- *Omnihash & DJSON  
Decentralised JSON*

**Pay-to-PubKey-Hash (Pay-to-Public-Key-Hash, P2PKH)** is the basic form of making a [transaction](#) and is the most common form of transaction on the [Bitcoin network](#). Transactions that pay to a [Bitcoin address](#) contain P2PKH scripts that are resolved by sending the public key and a digital signature created by the corresponding [private key](#).

The ScriptPubKey and ScriptSig for a transaction is shown below:

Table of Contents



W  
mple

[https://bitcoinwiki.org/wiki/  
pay-to-pubkey-hash](https://bitcoinwiki.org/wiki/pay-to-pubkey-hash)

ash Review

ed as P2PK (pay to public key) and P2PKH (pay to public key hash).

PKH instead of P2PK for two reasons:

cryptography used by your public key and [private key](#)) is vulnerable to a solving the discrete logarithm problem on elliptic curves. In plain English, it your computer might be able to retrieve a private key from a public key. By then the coins are spent (and assuming that addresses are not reused), such

bytes) it is easier to print and easier to embed into small storage mediums like



works  
Softfork  
Scalability  
Adaptive difficulty  
CVE-2012-4684-new  
CVE-2013-2293

Nanopayments

Block weight

BIP UNOFFICIAL DRAFT 0

Ideal Properties of Digital Commodities

Address reuse

Hashlock

Contingency plans

Offline transactions

Off-Chain Transactions

Funding network security

Bitcoin scalability problem

Segwit support

CVE-2012-3789

Proof of Ownership

Dump format

Test Cases

Hot wallet

Dominant Assurance Contracts

Bitcoin Binary Data Protocol

Coin analogy

CVE-2012-4683

BIP Draft – Instant Partial

A [Bitcoin address](#) is only a hash, so the sender can't provide a full public key in `scriptPubKey`. When redeeming coins that have been sent to a [Bitcoin address](#), the recipient provides both the signature and the public key. The `script` verifies that the provided public key does [hash](#) to the hash in `scriptPubKey`, and then it also checks the signature against the public key.

Checking process:

Stack	Script	Description
Empty.	<code>&lt;sig&gt; &lt;pubKey&gt; OP_DUP OP_HASH160 &lt;pubKeyHash&gt; OP_EQUALVERIFY OP_CHECKSIG</code>	scriptSig and scriptPubKey are combined.
<code>&lt;sig&gt; &lt;pubKey&gt;</code>	<code>OP_DUP OP_HASH160 &lt;pubKeyHash&gt; OP_EQUALVERIFY OP_CHECKSIG</code>	Constants are added to the stack.
<code>&lt;sig&gt; &lt;pubKey&gt; &lt;pubKey&gt;</code>	<code>OP_HASH160 &lt;pubKeyHash&gt; OP_EQUALVERIFY OP_CHECKSIG</code>	Top stack item is duplicated.
<code>&lt;sig&gt; &lt;pubKey&gt; &lt;pubHashA&gt;</code>	<code>&lt;pubKeyHash&gt; OP_EQUALVERIFY OP_CHECKSIG</code>	Top stack item is hashed.
<code>&lt;sig&gt; &lt;pubKey&gt; &lt;pubHashA&gt; &lt;pubKeyHash&gt;</code>	<code>OP_EQUALVERIFY OP_CHECKSIG</code>	Constant added.
<code>&lt;sig&gt; &lt;pubKey&gt;</code>	<code>OP_CHECKSIG</code>	Equality is checked between the top two stack items.
true	Empty.	Signature is checked for top two stack items.

works  
 Softfork  
 Scalability  
 Adaptive difficulty  
 CVE-2012-4684-new  
 CVE-2013-2293  
 Nanopayments  
 Block weight  
 BIP UNOFFICIAL DRAFT 0  
 Ideal Properties of Digital Commodities  
 Address reuse  
 Hashlock  
 Contingency plans

A [Bitcoin address](#) is only a hash, so the sender can't provide a full public key in `scriptPubKey`. When redeeming coins that have been sent to a [Bitcoin address](#), the recipient provides both the signature and the public key. The `script` verifies that the provided public key does [hash](#) to the hash in `scriptPubKey`, and then it also checks the signature against the public key.

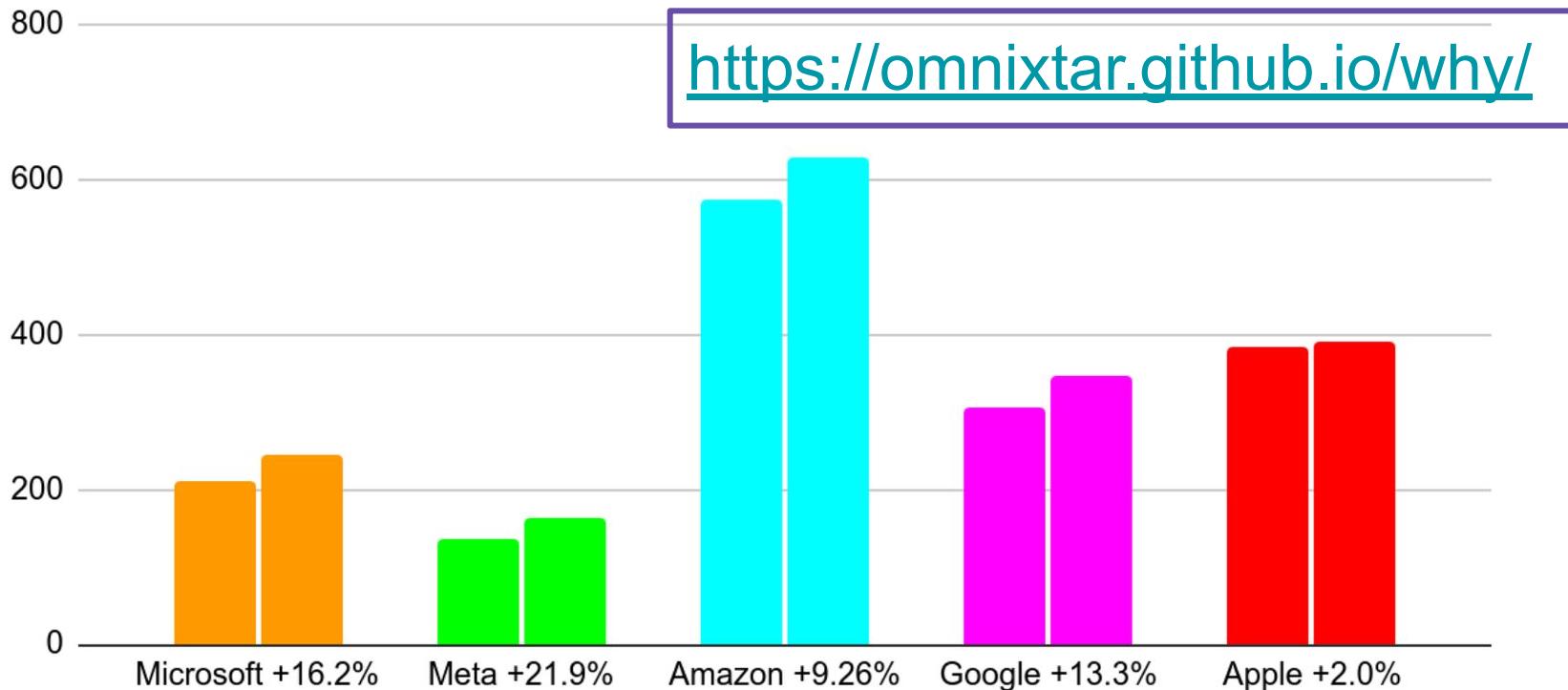
Checking process:

Stack	Script	Description
Empty.	<sig> <pubKey> OP_DUP OP_HASH160 <pubKeyHash> OP_EQUALVERIFY OP_CHECKSIG	scriptSig and scriptPubKey are combined.
<sig> <pubKey>	OP_DUP OP_HASH160 <pubKeyHash> OP_EQUALVERIFY OP_CHECKSIG	Constants are added to the stack.
	P_HASH160 <pubKeyHash> P_EQUALVERIFY OP_CHECKSIG	Top stack item is duplicated.
	<pubKeyHash> OP_EQUALVERIFY P_CHECKSIG	Top stack item is hashed.
	P_EQUALVERIFY OP_CHECKSIG	Constant added.
	P_CHECKSIG	Equality is checked between the top two stack items.
Empty.		Signature is checked for top two stack items.

- *FORTH used in both Bitcoin (Blockchains) & Omnihash/DJSON (Phoscript)*
- *“Managing ownerships of digital assets” vs. “who pays tariffs”*

# MMAGA Revenues 2023/24 (USD billions)

**Total 2024: USD 1.776 Trillion (+10.27%)**

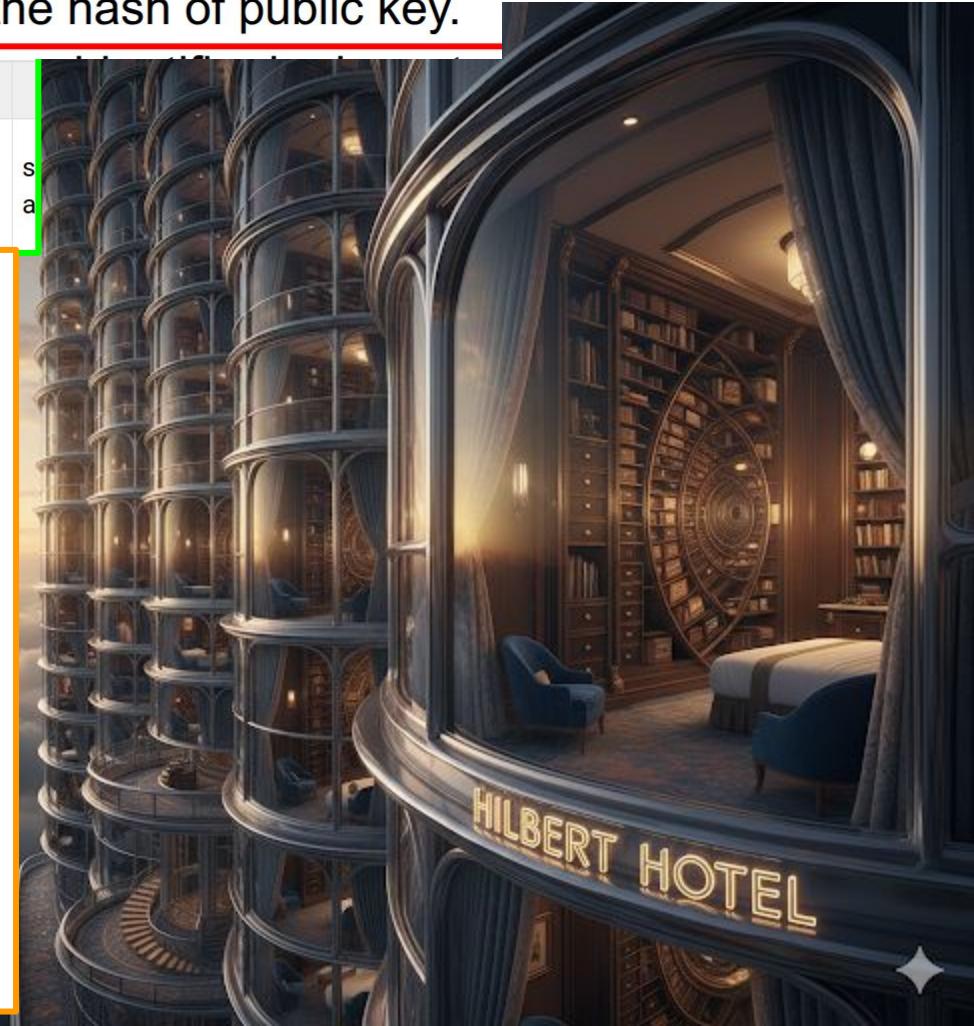


Bitcoin uses FORTH like syntax to verify the hash of public key.

Stack	Script
Empty.	<sig> <pubKey> OP_DUP OP_HASH160 <pubKeyHash> OP_EQUALVERIFY OP_CHECKSIG

Hilbert Hotel with Infinite Cabinets: Recursion added since 1924

Hilbert Metatron / Hilbertron  
virtual objects in Hilbert world:  
folders, documents, trains,  
ships, etc. labeled w/ hash  
numbers / hash code strings



Hilbert Hotel: has infinite number of rooms.

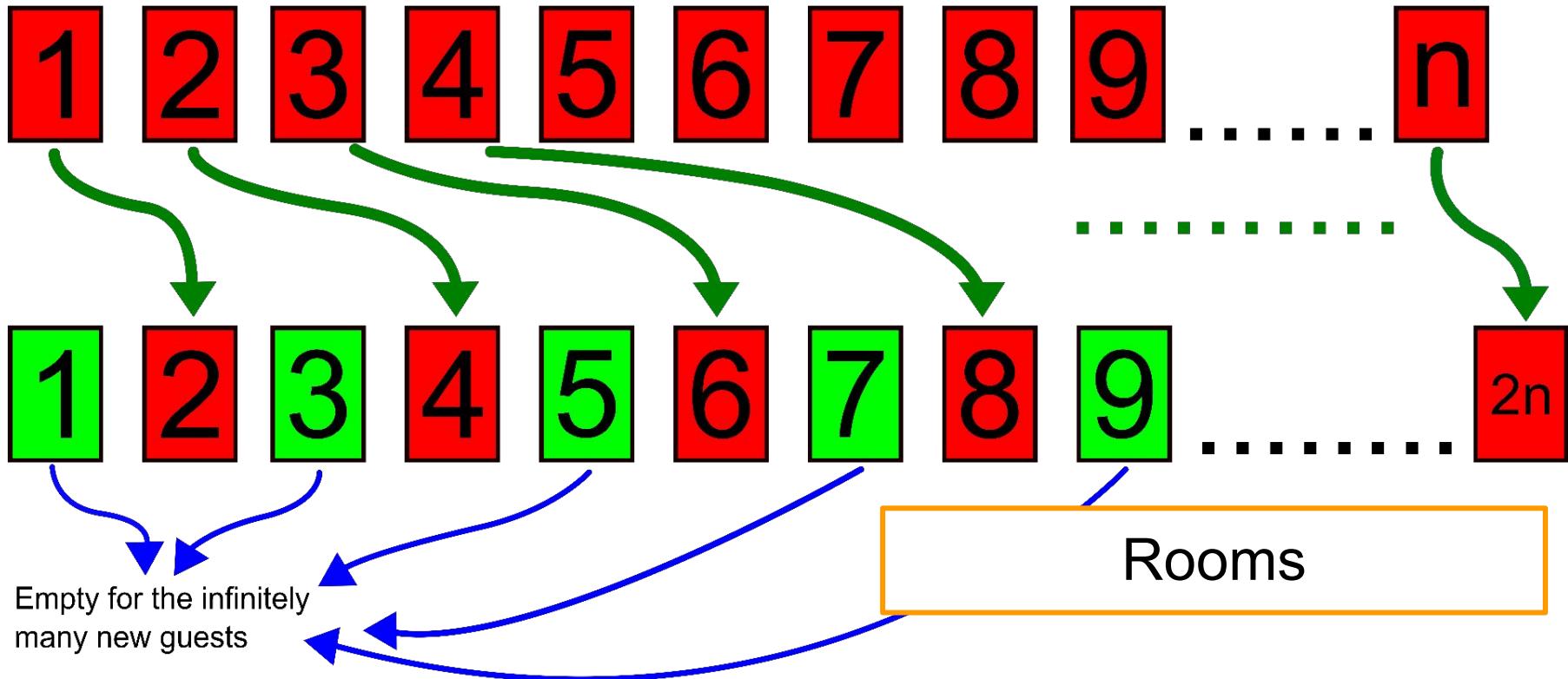
Room number is a hash number.

New: Each room has an Infinite Cabinet (Recursion)

Each drawer is labeled with hash number.



Guests



- Bitcoin uses FORTH like syntax to verify the hash of public key.
  1. Extend the use of hash of public key as user identifier in decentralised network and explore various properties of hash codes,
  2. Omnihash, literally meaning a hash code for representing all types of digital assets, and DJSON decentralized JSON, a special type of JSON string containing at least one Omnihash
  3. Omnihash as the basis of hash filesystem, which becomes the foundation of a database system without database server
  4. 3X architecture, namely database-less, domain-less and kernel-less
  5. Phoscript and Omnihash as used in Web and mobile applications, Phoscript metaprogramming shell: ported to any host programming language, including front end as well as back end environments  
Phase II of 3X architecture: Sandwich model of code substitution substitute user space services such as Web server and I2P routers with FORTH or Phoscript exclusive code base.

- Bitcoin uses FORTH like syntax to verify the hash of public key.
- 1. Extend the use of hash of public key as user identifier in decentralised network and explore various properties of hash codes,
- 2. Omnihash, literally meaning a hash code for representing all types of digital assets, and DJSON decentralized JSON, a special type of JSON string containing at least one Omnihash
- 3. Omnihash as the basis of hash filesystem, which becomes the foundation of a database system without database server.

#### 4. 3X architecture, namely database-less, domain-less and kernel-less

- 5. Phoscript and Omnihash as used in Phoscript metaprogramming shell including front end as well as back end Phase II of 3X architecture: Sandboxed substitute user space services such as FORTH or Phoscript exclusive code

How to relate “ownership of digital assets” to  
Ordinary Users

(who only listen to MMAGA)  
(Microsoft Meta Amazon Google Apple)

or FORTH Programmers

(who may not be familiar with JSON  
or database programming)

???

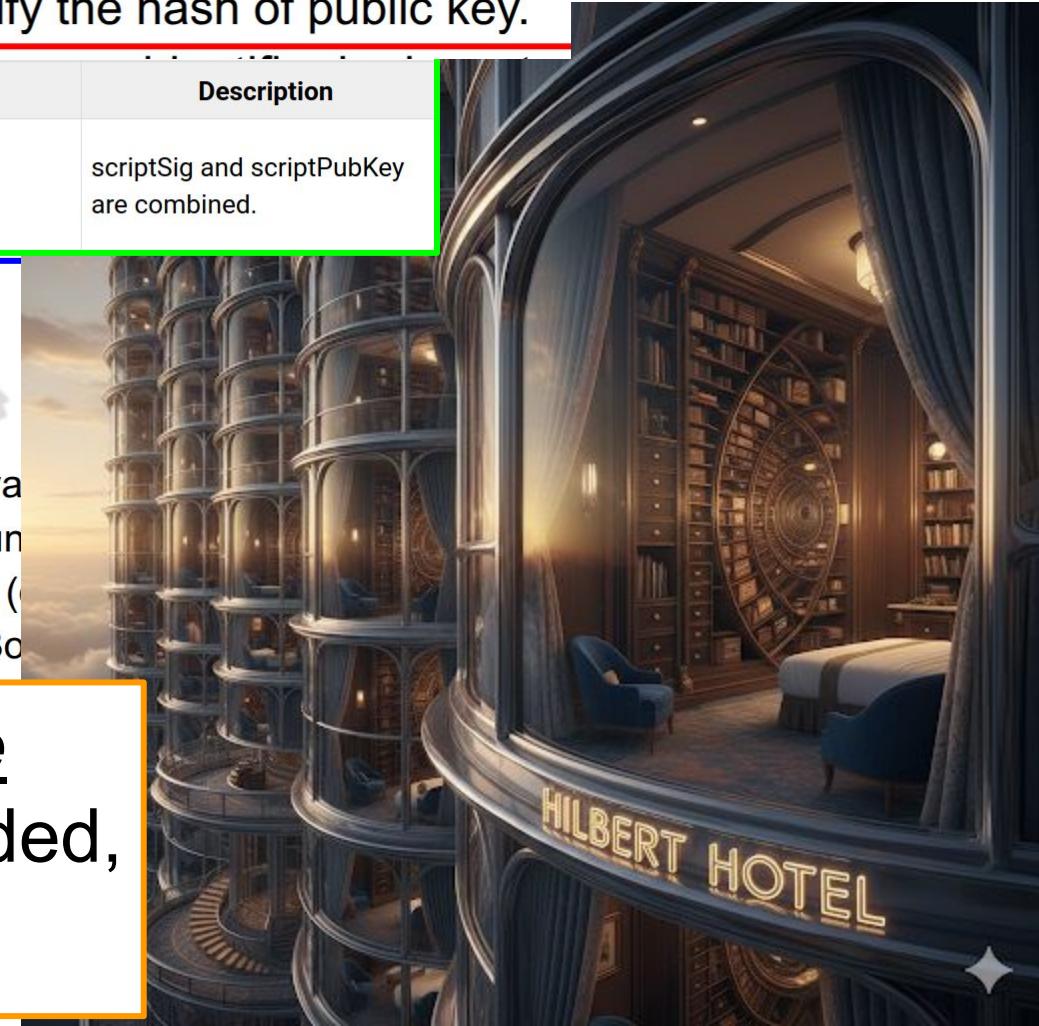
Bitcoin uses FORTH like syntax to verify the hash of public key.

Stack	Script	Description
Empty.	<sig> <pubKey> OP_DUP OP_HASH160 <pubKeyHash> OP_EQUALVERIFY OP_CHECKSIG	scriptSig and scriptPubKey are combined.

Phoscript-Linux/C-FORTH Sandwich Model

1. Minimal VM: Linux Java I2P Apache php
2. Sandwich Model (user space programs):
  - a. Top: P2C (Phos to C) P2J (Phos to Java)  
[ start: 0, end: replace host language fun]
  - b. Middle: C-lib, J-lib (Java), PHP-lib in .o (br)  
[ start: 100%, end: replaced by Top & Bottom]

*Hilbert Hotel with Infinite Cabinets*: Recursion added, since 1924.



Intro Recap (5 minutes)

Hilbert Hotel, Infinite Cabinets, Infinity Trains (5 minutes)

DJSON Decentralised JSON & Hash Filesystem (10 min)

Examples: [omnixtar.github.io/h](https://omnixtar.github.io/h) [omnixtar.github.io/contract](https://omnixtar.github.io/contract)

3X Architecture (5 min)

Conclusions, Future installment of this series (5 min)



Home



Shorts



Subscriptions



You

All

Shorts

Videos

Unwatched

Watched

Recently uploaded

Live

Subscriptions

Filters



## How An Infinite Hotel Ran Out Of Room

34m views • 4 years ago

Ve Veritasium

... to know if you're the manager of the Hilbert Hotel. ..... References: Ewald ...

0:12 Dies ist das Hilbert Hotel und du bist der Manager. Jetzt wirkt es vielleicht so, als ob du für jeden der...

4K Subtitles



## The Infinite Hotel Paradox - Jeff Dekofsky

25m views • 11 years ago

TED-Ed

Sign up for our newsletter and never miss an animation: <http://bit.ly/TEDEdNewsletter> View full lesson: ...

Subtitles



hilbert hotel digital asset ownership



AI Mode

All

Images

News

Videos

Shopping

Short videos

More ▾

Tools ▾

AI Overview

⋮

The term "Hilbert hotel digital asset ownership" refers to **Hilbert Group**, a Nasdaq-listed investment firm that specializes in quantitative trading strategies and asset management in digital asset markets, including Bitcoin and Ethereum. The name is a reference to Hilbert's paradox of the Grand Hotel, a mathematical thought experiment. 

Hilbert Group is a public company that offers traditional investors a regulated entry point to the digital asset space. Its ownership structure involves: 

- Public Shareholders:** The company is listed on the Nasdaq First North Growth Market (ticker HILB B).
- Key Personnel Ownership:** The Chief Investment Officer and a major shareholder, Russell Thompson, has expressed his intention to significantly increase his ownership in the company through open-market purchases.

About - Hilbert Group

Hilbert is a tech-enabled investment company operate...



 [www.hilbert.group](http://www.hilbert.group) 

Hilbert Group Acquires Renowned Digital Asset Manager ...

25 Nov 2024 — Headquartered in the British Virgin Islands, the firm focuses on digital...

 [www.hilbert.group](http://www.hilbert.group) 

Hilbert Group Acquires 100% of Fast-Growing...

1 Sept 2025 — Hilbert Group AP (Nasdaq: HILB.B) today...



# Interim Report Q3 2025 In Brief

## FINANCIAL EVENTS

### Third quarter

- Revenues amounted to KSEK 51,562.6 (KSEK 31,551.7)
- Operating profit/loss (EBIT) totalled KSEK -21,433.3 (KSEK -12,095.7)
- Results for the period were KSEK -54,330.9 (KSEK -18,820.5)
- Net Cash flow from operating activities amounted to KSEK -58,217.7 (KSEK -11,603.3)
- Earnings per share before and after dilution amounted to SEK -0.66 (SEK -0.29)

### Year to date January – September

- Revenues amounted to KSEK 142,764.5 (KSEK 70,786.4)
- Operating profit/loss (EBIT) totaled KSEK -51,354.3 (KSEK -33,830.2)
- Results for the period were KSEK -105,228.0 (KSEK -37,337.6)
- Cash flow from operating activities amounted to KSEK -69,077.6 (KSEK -26,100.8)
- Earnings per share before and after dilution amounted to SEK -1.43 (SEK -0.58)

## Financial overview

Nine months Nine months Full year

## Financial overview

KSEK	Q325	Q324	Nine months 2025	Nine months 2024	Full year 2024
Revenue	51,562.6	31,551.7	142,764.5	70,786.4	112,977.5
EBIT	-21,433.3	-12,095.7	-51,354.3	-33,830.2	-49,718.6
Profit/loss for the period	-54,330.9	-18,820.5	-105,228.0	-37,337.6	-43,329.3
Earnings per share	-0.66	-0.29	-1.43	-0.58	-0.60
Equity	-677.4	16,144.7	-677.4	16,144.7	23,101.0
Total assets	88,341.0	55,407.2	88,341.0	55,407.2	64,232.0
Equity/assets ratio, %	-0.01	0.29	-0.01	0.29	0.36

## SIGNIFICANT EVENTS

### During the third quarter and after the reporting period

- Hilbert Group Taps Four Leading Blockchain Figures to Syntetika Advisory Board – Max Rabinovitch, John Lilic, Vladimir Maslyakov, Chirdeep Chhabra
- Hilbert Group launches a Strategic Crypto Treasury Program; managed by a newly established Treasury Committee chaired by CIO Russell Thompson
- Hilbert Group executes on SEK200+ Million Financing to Support Crypto Treasury Strategy
- Lars Seir Christensen co-founder of Saxo Bank, appointed as an Advisor to the Board

- Deus X Capital commits SEK300million worth of Bitcoin to Hilbert Group's Crypto Treasury Strategy
- Liberty Road Capital's ownership increases; the Board of Directors resolved on a directed issue of 7,012,963 B shares based on the authorisation at the last AGM, as conditions were fulfilled for an additional equity payment
- Hilbert Group closes heavily oversubscribed seed round for Syntetika Tokenisation and Decentralised Trading Platform. The round was heavily oversubscribed, attracting over \$ 6 million in investor interest – six times the original \$1 million target
- Exercise of Hilbert Employee Stock Options between July 2025 & September 2025 by Mr Niclas Sandstrom and other participants; first time since inception
- Hilbert's Group shareholders proposes board changes – Steen Jackobsen, former CIO of Saxo Bank, is nominated to the Board. Niclas Sandstrom, Russell Thompson, David Butler, Stuart Connolly and Erik Nerpin proposed to the Board, Forde Foss Skiftesvik and Tim Grant to step down
- Hilbert Group Enters Flexible 3 Year SEK 150 Million Financing Arrangement with LDA Capital; Hilbert Group gains access not only to capital but also to LDA's global network of TradFi and Web 3.0 portfolio companies, crypto native advisors and strategic investors
- Hilbert Group's Syntetika Welcomes Asia-Based Investment Leader Juliet Tang. Former Investment Manager at Huobi, HTX Ventures & Ryan Horn, Ex-Binance Leader Behind Cristiano Ronaldo's Partnership to the Advisory Board
- Hilbert Group announces early repayment of the Eur2 million senior unsecured convertible bond, due in 2025
- Hilbert Group announced full regulatory approval from the Gibraltar Financial



## Bitcoin: Technical Concepts

Bech32  
Blockchain Diagram

Bitcoin Encryption  
Creating forks  
Bitcoin mining  
Blockchain

Bitcoin Improvement Proposals  
Pay-to-Script Hash

Proof of Keys

UTXO

User Activated Soft Fork

OmniBOLT

Blockchain (database)

Segregated Witness

Lightning Network

Hashed Timelock Contracts

NSequence

Bitcoin Emission

Block timestamp

**Pay-to-PubKey-Hash (Pay-to-Public-Key-Hash, P2PKH)** is the basic form of making a [transaction](#) and is the most common form of transaction on the [Bitcoin network](#). Transactions that pay to a [Bitcoin address](#) contain P2PKH scripts that are resolved by sending the public key and a digital signature created by the corresponding [private key](#).

The ScriptPubKey and ScriptSig for a transaction is shown below:

### Table of Contents



1. Pay-to-Pubkey Hash
  - 1.1. Pay-to-PubKey-Hash Review
  - 1.2. Pay-to-PublicKey Hash Example
  - 1.3. See also
  - 1.4. References

<https://bitcoinwiki.org/wiki/pay-to-pubkey-hash>

## Pay-to-PubKey-Hash Review

Two types of payment are referred as P2PK (pay to public key) and P2PKH (pay to public key hash).

[Satoshi](#) later decided to use P2PKH instead of P2PK for two reasons:

Elliptic Curve [Cryptography](#) (the cryptography used by your public key and [private key](#)) is vulnerable to a modified Shor's [algorithm](#) for solving the discrete logarithm problem on elliptic curves. In plain English, it means that in the future a quantum computer might be able to retrieve a private key from a public key. By publishing the public key only when the coins are spent (and assuming that addresses are not reused), such attack is rendered ineffective.

With the hash being smaller (20 bytes) it is easier to print and easier to embed into small storage mediums like



works  
Softfork  
Scalability  
Adaptive difficulty  
CVE-2012-4684-new  
CVE-2013-2293

Nanopayments

Block weight

BIP UNOFFICIAL DRAFT 0

Ideal Properties of Digital Commodities

Address reuse

Hashlock

Contingency plans

Offline transactions

Off-Chain Transactions

Funding network security

Bitcoin scalability problem

Segwit support

CVE-2012-3789

Proof of Ownership

Dump format

Test Cases

Hot wallet

Dominant Assurance Contracts

Bitcoin Binary Data Protocol

Coin analogy

CVE-2012-4683

BIP Draft – Instant Partial

A [Bitcoin address](#) is only a hash, so the sender can't provide a full public key in `scriptPubKey`. When redeeming coins that have been sent to a [Bitcoin address](#), the recipient provides both the signature and the public key. The `script` verifies that the provided public key does [hash](#) to the hash in `scriptPubKey`, and then it also checks the signature against the public key.

Checking process:

Stack	Script	Description
Empty.	<code>&lt;sig&gt; &lt;pubKey&gt; OP_DUP OP_HASH160 &lt;pubKeyHash&gt; OP_EQUALVERIFY OP_CHECKSIG</code>	scriptSig and scriptPubKey are combined.
<code>&lt;sig&gt; &lt;pubKey&gt;</code>	<code>OP_DUP OP_HASH160 &lt;pubKeyHash&gt; OP_EQUALVERIFY OP_CHECKSIG</code>	Constants are added to the stack.
<code>&lt;sig&gt; &lt;pubKey&gt; &lt;pubKey&gt;</code>	<code>OP_HASH160 &lt;pubKeyHash&gt; OP_EQUALVERIFY OP_CHECKSIG</code>	Top stack item is duplicated.
<code>&lt;sig&gt; &lt;pubKey&gt; &lt;pubHashA&gt;</code>	<code>&lt;pubKeyHash&gt; OP_EQUALVERIFY OP_CHECKSIG</code>	Top stack item is hashed.
<code>&lt;sig&gt; &lt;pubKey&gt; &lt;pubHashA&gt; &lt;pubKeyHash&gt;</code>	<code>OP_EQUALVERIFY OP_CHECKSIG</code>	Constant added.
<code>&lt;sig&gt; &lt;pubKey&gt;</code>	<code>OP_CHECKSIG</code>	Equality is checked between the top two stack items.
true	Empty.	Signature is checked for top two stack items.

Hilbert Hotel: has infinite number of rooms.

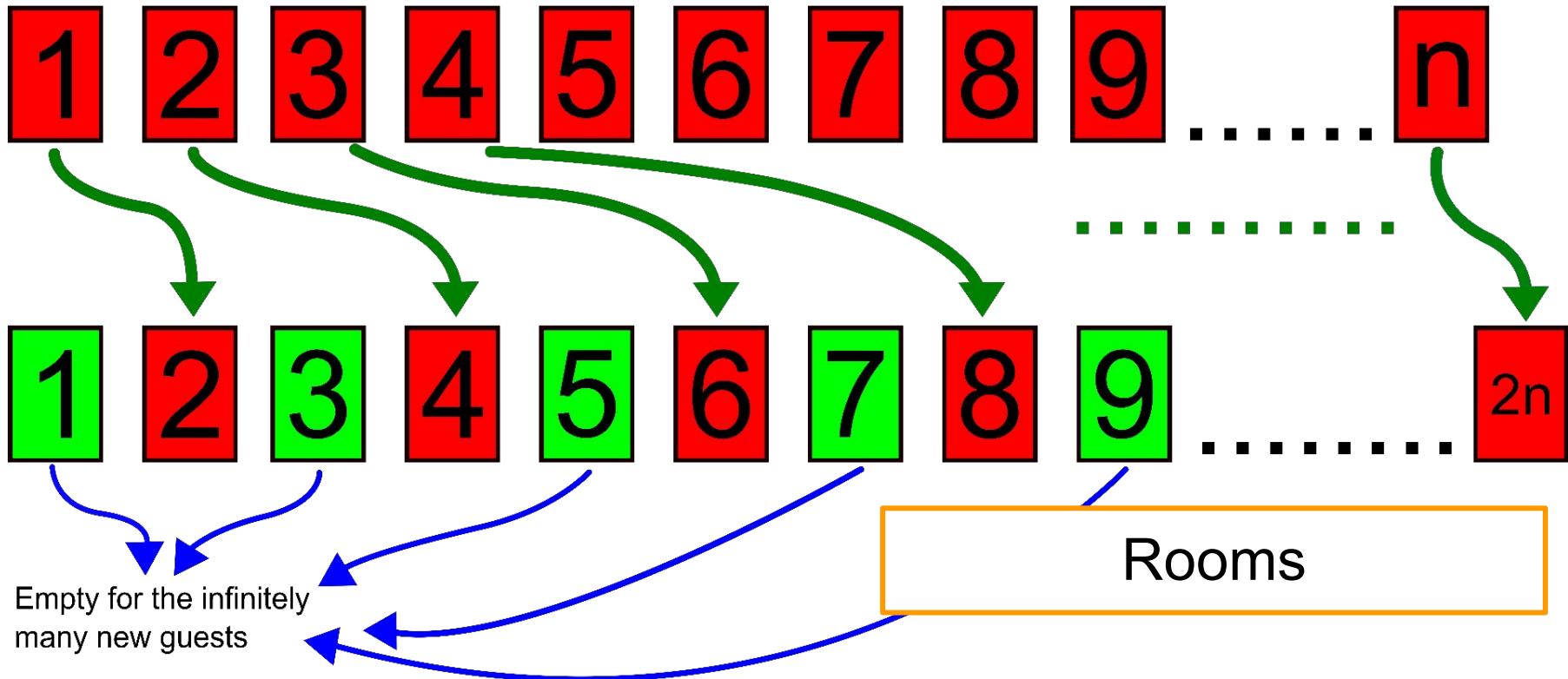
Room number is a hash number.

New: Each room has an Infinite Cabinet (Recursion)

Each drawer is labeled with hash number.



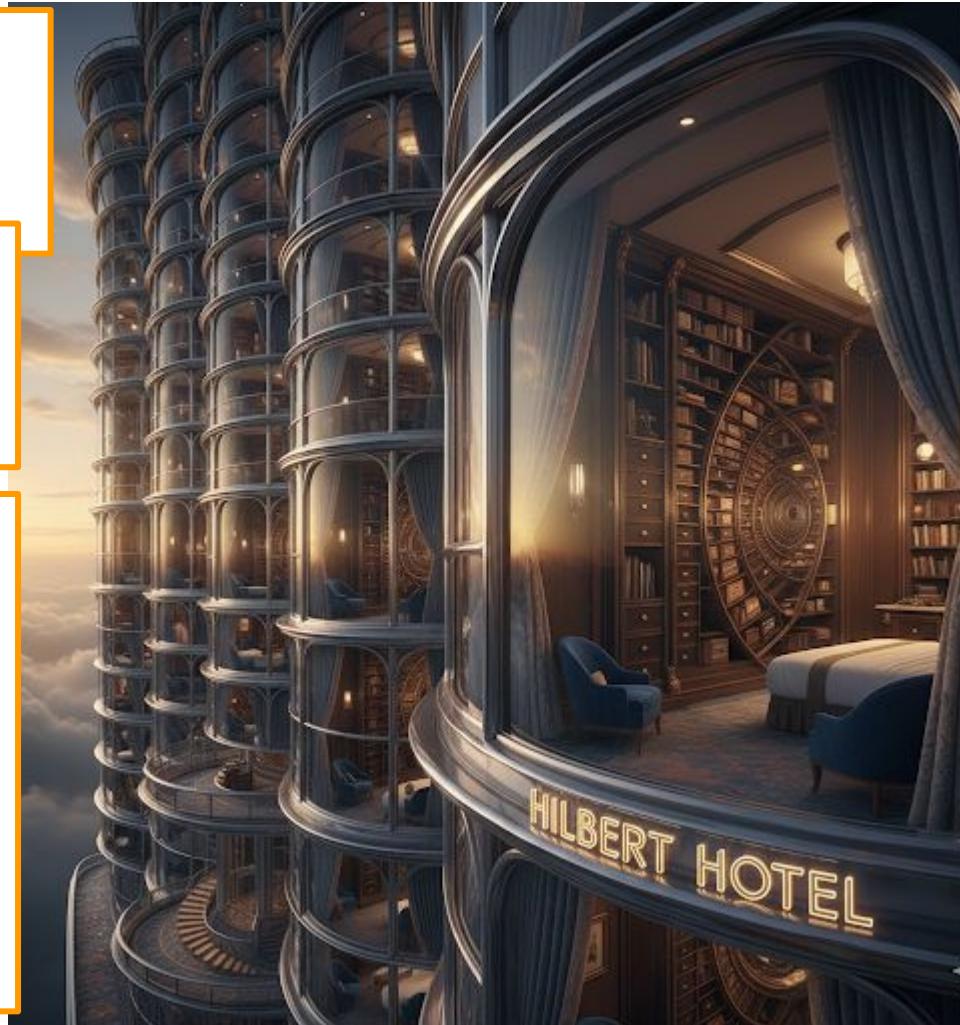
Guests



# Hilbert Hotel with Infinite Cabinets

What is the practical significance of this?

Imagine: Every comment, photo, video, document YOU posted on the Internet is like a piece of paper or file in Infinite Cabinet, each labeled with a hash number.



omnixtar.github.io/h/

omnixtar.github.io/h/

# Omni\* Omnihash

[Omnihash](#)[GenKeys](#)[Save](#)[Help](#)[Load Keypair](#) No file chosen

Your Hash is ECClo9IBig==

Your Parent Hash is root

You have 0 downlines.

```
{"x":"REG","p":"root","h":"ECClo9IBig==","t":2025-12-18T11:02:03.028Z"}
```

- You may copy & save this DJSON  
Decentralised JSON ...



*Practical use: manage SVFIG  
memberships or any organisations!!*

[Omnihash](#)[GenKeys](#)[Save](#)[Help](#)[Load Keypair](#)[Choose File](#) No file chosen

Your Hash is ECClo9IBig==

Your Parent Hash is root

You have 0 downlines.

```
{"x":"REG","p":"root","h":"ECClo9IBig==","t":2}
```

```
025-12-18T11:02:03.028Z"}
```

- You may copy & save this DJSON

Decentralised JSON ...



*Your User Identifier is the most valuable digital asset!!*

*Hash of public key!! (Remember & Repeat!!)*

All subsequent hashcode strings & Omnihash(es) contain parts of User ID (hash of public key)!!

omnixtar.github.io/h/?ECClo9lBig==

omnixtar.github.io/h/?ECClo9lBig==

New Chrome available

# Omni\* Omnihash

[Omnihash](#)[GenKeys](#)[Save](#)[Help](#)[Load Keypair](#) No file chosen

Your Hash is H-dYqaYMkQ==

Your Parent Hash is ECClo9lBig==

You have 0 downlines.



DevTools - omnixtar.github.io/h/?ECClo9lBig==

Console

6 Issues: 2 1 3 | 1 hidden

-----END PUBLIC KEY-----

ECClo9lBig== h/?ECClo9lBig==:225

A2RJY6IEjA== h/?ECClo9lBig==:234

⚠ The resource [https://corsmirror.com/\\_graphql?body=%7B%22persistedQueryNa...](https://corsmirror.com/_graphql?body=%7B%22persistedQueryNa...) was preloaded using link preload but not used within a few seconds from the window's load event. Please make sure it has an appropriate `as` value and it is preloaded intentionally.

Console What's new AI assistance

# Omni\* Omnihash

Omnihash | GenKeys | Save | Help

## Load Keypair

Choose File No file chosen

Your Hash is H-dYqaYMkQ==

Your Parent Hash is **ECClo9IBig==**

You have 0 downlines.



DevTools - omnixtar.github.io/h/?ECClo9lBig==

Elements Console Sources > ✘ 4 ⚠ 1 ✖ 2 ⌂ ⚙ ⋮

Default levels ⋮

6 Issues: ✖ 2 ⚠ 1 ✎ 3 ⋮ 1 hidden ⌂

was preloaded using link preload but not used within a few seconds from the window's load event. Please make sure it has an appropriate `as` value and it is preloaded intentionally.

> f('a h53:')

< undefined

> t()

< 7929297801672961

> s[s.length-1]

< 7929297801672961

> Math.pow(2,53)

< 9007199254740992

>

⋮ Console What's new ⌂ AI assistance ⌂ ⋮



14ce0dd ▾

omnixtar.github.io / h / Phos / librsa.js

↑ Top

Code Blame

Raw



```
193     }
194
195     const cyrb53 = function(str, seed = 0) {
196         let h1 = 0xdeadbeef ^ seed, h2 = 0x41c6ce57 ^ seed;
197         for (let i = 0, ch; i < str.length; i++) {
198             ch = str.charCodeAt(i);
199             h1 = Math.imul(h1 ^ ch, 2654435761);
200             h2 = Math.imul(h2 ^ ch, 1597334677);
201         }
202         h1 = Math.imul(h1 ^ (h1>>>16), 2246822507) ^ Math.imul(h2 ^ (h2>>>13), 3266489909);
203         h2 = Math.imul(h2 ^ (h2>>>16), 2246822507) ^ Math.imul(h1 ^ (h1>>>13), 3266489909);
204         return 4294967296 * (2097151 & h2) + (h1>>>0);
205     };
206
207     ...
208     function f_h53()
209     {
210         var S = M.S;
211         S.push(cyrb53(S.pop()));
212     }
213
```

[main](#) ▾[omnixtar.github.io / js / omni.js](#)

↑ Top

[Code](#)[Blame](#)[Raw](#)

```
35  var bnToB64=function bnToB64(bn){  
36      var hex = BigInt(bn).toString(16);  
37      if (hex.length % 2) { hex = '0' + hex; }  
38      var bin = [];  
39      var i = 0;  
40      var d;  
41      var b;  
42      while (i < hex.length) {  
43          d = parseInt(hex.slice(i, i + 2), 16);  
44          b = String.fromCharCode(d);  
45          bin.push(b);  
46          i += 2;  
47      }  
48      return btoa(bin.join(''));  
49  }  
50  var mkeval_l=function(FN){ // *_l means long form before minified  
51      var j1={n:FN}  
52      j1.s=eval(j1.n).toString()  
53      j1.t=(new Date()).toISOString()  
54      var j2={s:JSON.stringify(j1),h:bnToB64(cyrb53(JSON.stringify(j1)))} }
```

Activities Google Chrome Sab 20 Dis, 21:43

omnixtar.github.io/h/?ECClo9lBig==

# Omni\* Omnihash

Omnihash GenKeys Save Help

Load Keypair

Choose File No file chosen

Your Hash is H-dYqaYMkQ==

Your Parent Hash is ECClo9lBig==

You have 0 downlines.



DevTools - omnixtar.github.io/h/?ECClo9lBig==

Elements Console Sources >

2 Default levels

6 Issues: 2 1 3

```
> f('a h53: b64: ')
< undefined
> t()
< 'HCungs15AQ==' (cursor)
> f_b64=function(){s.push(bnToB64(s.pop()))}
< f (){s.push(bnToB64(s.pop()))}
>
```

Console What's new X AI assistance

[main](#) ▾[omnixtar.github.io / js / omni.js](#)[↑ Top](#)[Code](#)[Blame](#)[Raw](#)

```
9  function f_path() // path safe b64
10 {
11     // $a = array_pop($S);
12     var $a = s.pop()
13     // echo '  in path: '.$a." ".strpos($a, '/')."\n";
14     // if (strpos($a, '/')>=0) { $a = str_replace('/', '_', $a); } // echo $a; }
15     if ($a.indexOf('/')>=0) { $a = $a.replaceAll('/', '_'); }
16     // if (strpos($a, '+')>=0) $a = str_replace('+', '-', $a);
17     if ($a.indexOf('+')>=0) { $a = $a.replaceAll('+', '-'); }
18     s.push($a)
19 }
20
21 // mkeval_l("omnihash")
22 // DmII2G1lw==
23 // var omnihash=function(){
24 var cyrb53=function(str, seed = 0){
25     let h1 = 0xdeadbeef ^ seed, h2 = 0x41c6ce57 ^ seed;
26     for (let i = 0, ch; i < str.length; i++) {
27         ch = str.charCodeAt(i);
28         h1 = Math.imul(h1 ^ ch, 0xFFFFFFFF);
```



index.html X

h &gt; index.html &gt; html &gt; body &gt; script &gt; click() callback

```
1  <html>
23 <body>
123 <script type="text/javascript">
405 $('#omni').click(async function () {
411
412     // Get the input and crypted values.
413     var input = $('#pubkey2').val();
414     var crypted = $('#hpbkb').val();
415
416     console.log('bdmhs 2023 12 31 ', S0.K.s_pbk);
417
418     // var H = bnToB64( cyrb53( btoa(S0.K.s_pbk) ) )
419     var H = f_path( bnToB64( cyrb53( btoa(S0.K.s_pbk) ) ) )
420     dt = new Date();
421
422     s_up=JSON.stringify({c:$('#input2').val(),x:"REG",p:S_P,h:H,nn:nn,t:dt,
423     toISOString(),r:$('#hash4').val()});
424
425     $('#dmhs').val( JSON.stringify({c:$('#input2').val(),x:"REG",p:S_P,h:H,
426     t:dt.toISOString(),r:$('#hash4').val()}) + "\n\n- You may copy & save
```



omnixtar.github.io/h/?ECClo9lBig== x omnixtar.github.io/h/?ECClo9lBig== x +

omnixtar.github.io/h/?ECClo9lBig==

# Omni\* Omnihash

[Omnihash](#)[GenKeys](#)[Save](#)[Help](#)[Load Keypair](#) No file chosen

Your Hash is H-dYqaYMkQ==

Your Parent Hash is ECClo9lBig==

You have 0 downlines.



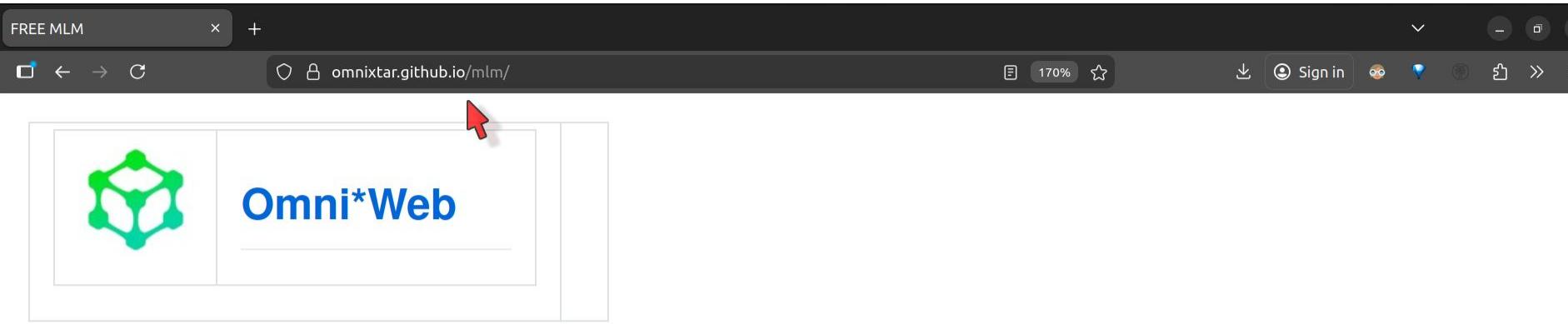
DevTools - omnixtar.github.io/h/?ECClo9lBig==

Elements Console Sources > 2 Default levels

6 Issues: 2 1 3 |

```
> f('S0.K.s_pbk ev: b2a: h53: b64: path: ')
< undefined
> t()
< 'H-dYqaYMkQ=='
> S0.K
< {s_pk: '-----BEGIN PRIVATE KEY-----\nMIJQgIBADANBgkqhkiG9w...134o2RsyS0p1XQmMUuT/A==\n-----END PRIVATE KEY-----', s_pbk: '-----BEGIN PUBLIC KEY-----\nMIICIjANBgkqhkiG9w0BAQE...D9e2VEnpiX5lBBcCAwEAAQ==\n-----END PUBLIC KEY-----', pair: {...}}
```

Console What's new X AI assistance



## Omnihash: THE One & Only FREE Multilevel Marketing

1. Start with no parent hash:

- <https://omnixtar.github.io/h/>
- [Demo video](#) 

2. Start with a parent hash:

- <https://omnixtar.github.io/h/?Du8iBzaMkw==>
- [Demo video](#) 

*Membership for SVFIG using  
Omnihash*

*– or any organisations  
– charity or commercial  
– membership is the most valuable  
digital asset!!  
– [omnixtar.github.io/mlm](https://omnixtar.github.io/mlm)*

Activities Firefox

Rab 2 Jul, 17:27

US zh

omnixtar.github.io/h/

170%

## Omni\* Omnihash

Omnihash

GenKeys

Save

Load Keypair

Browse... No file selected.

Your Hash is B3NfOsellg==

Your Parent Hash is undefined

You have 0 downlines.

```
{"x":"REG","h":"B3NfOsellg==","t":"2025-07-02T09:27:17.613Z"}
```

- You may copy & save this JSON  
Decentralised JSON ...



Get Omnihash with NO parent hash

0:16 / 0:32

II

⚙

✉

↗



MI4 Metaprogramming in Forth

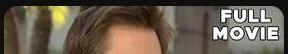
58 subscribers



All

For you

Recently uploaded



The Captains | FULL MOVIE |

William Shatner, Patrick

Activities Firefox

Rab 2 Jul, 17:28

US zh

omnixtar.github.io/h/?Du8iBzaMkw==

170%

## Omni\* Omnihash

Omnihash

GenKeys

Save

Load Key

Browse... No file selected.

Your Hash is BPcCpnRKrw==

Your Parent Hash is Du8iBzaMkw==

You have 0 downlines.

```
{"x":"REG","p":"Du8iBzaMkw==","h":"BPcCpnRKrw==","t":"2025-07-02T09:28:25.026Z"}
```

- You may copy & save this JSON  
Decentralised JSON ...



Get Omnihash WITH parent hash



MI4 Metaprogramming in Forth

58 subscribers



All

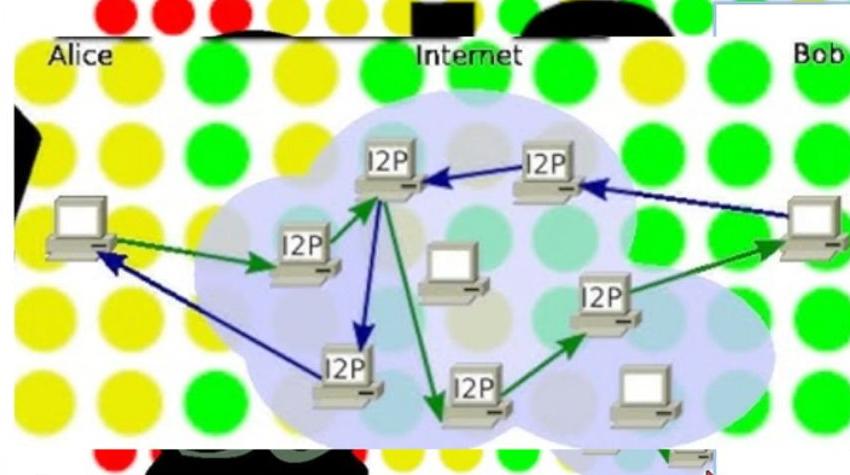
From MI4 Metaprogramming in Forth

Rela

&gt;

 Shorts

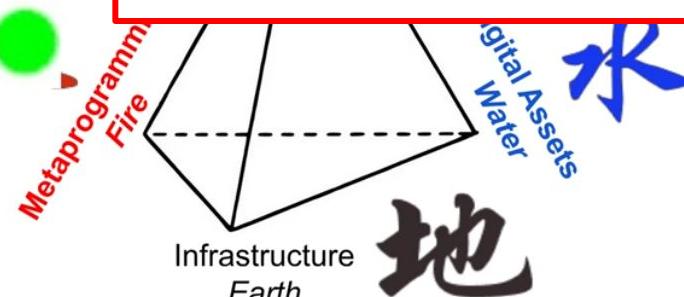
# INVISIBLE INTERNET PROJECT



[Bypass Domain Name System!!](#)

<https://omnixtar.github.io/oxw>

<https://www.youtube.com/watch?v=-3CO5bG -q8&t=56s>



Omni\*Web: Set Up YrOwn GLOBALLY ACCESSIBLE Web Server w/o Domain Name System in 4 Simple Steps (I2P)



MI4 Metaprogramming in Forth  
56 subscribers

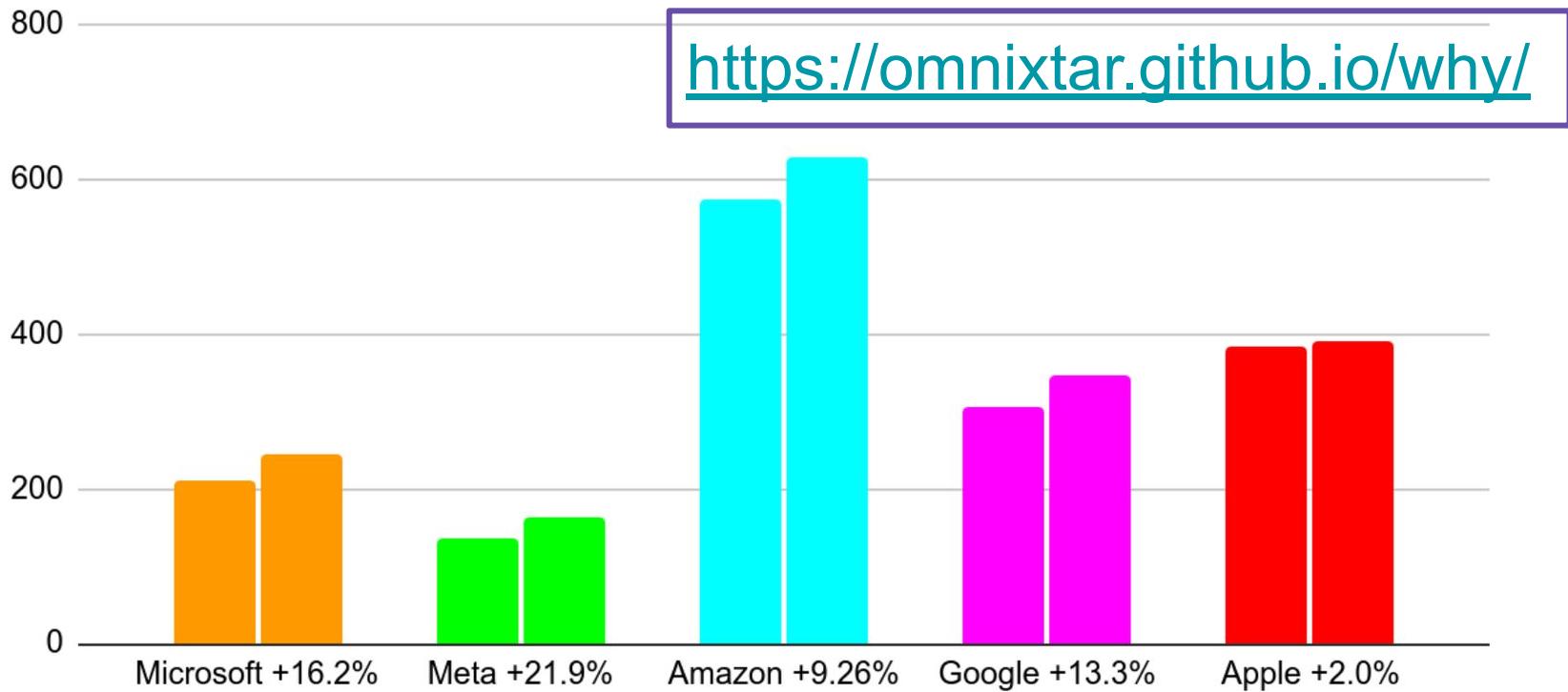


ONCE YOU UNWRAP,  
YOU CAN'T GO BACK



# MMAGA Revenues 2023/24 (USD billions)

**Total 2024: USD 1.776 Trillion (+10.27%)**





yo6sgmfq7pfvp2e4kcuhtfg X yo6sgmfq7pfvp2e4kcuhtfg X +

https://yo6sgmfq7pfvp2e4kcuhtfg7wfllt63igwcuuhbmuqm6lu3a3a.b32.i2p/oxm/oxm.php?nn=adam

170%



adam session\_id:adam-fer4f9qk69m9nj562u7kj5n2u9\_31  
session\_id:adam-fer4f9qk69m9nj562u7kj5n2u9\_31

## Omni\*Search Demo

Copy the following code and execute it in

f(): B\_F + JSON.stringify({w:'like', t:'recipi

1602 Phos 1611 S 1621 B\_F start omni:

fgl\_s 340 < 2 > array ( 0 => array ( 0 => '{  
'lcdw:', 5 => 'B\_AUTH', ), 1 => array ( 'SP'  
'LS0tLS1CRUdJTiBQVUJMSUMgS0VZLS0tLS0tKUJQ0lqQU5CZ2txaGt  
pRzl3MEJBUUVGQUFQ0FnOEFSNlDQ2dLQ0FnRUFyUlR1TWJCL3NS  
c2dXmNSeGhOUDlLWkNVRkJkdERS5VX1b3oZSC9XT05nYUlQWExG  
NHVwZFNUTHU0QXhWNutkUnFBTwlMlRzl2ZTzkaBqczS3MvQlVNd  
0p3VW1Jd1EyYmtFL0NsZFdsDCweExVvXNnQul20VNGOTRCGpmW  
EZMODVJL2Vaa1NSem5RWRHsZnQvYzJG4yMDFyelB3OGl0dEl1Z2R  
vNHB5anBLVzNus095S1AwZ2Nrck0dWJCS1HS1qlqOW5YeXZraSszUU  
V4b5t30JRWfhOR28yAh2UFR3akhTSVUvdhwbzKZYkpiVm8vTlZtU  
9vK2tStkl5Zw1manpyYTFEnzdyZDMrZjMveER4bGQraTg2K3dyRlkN  
WZlCQzOElpK0pZMkltWnZxd1hxaxVZ5eGirTU1ENUJpNDNmTnBis0Js  
ZKJHZQVNR0STZ  
ElsvVvWaSjVNVDhtMnBodGtM0V4WHaya0JtZm12ZmxPRXRrdmdaZ  
m1OQzJPbUcyXNYZ3pWl05qZk5kakd2ZwP0vU0YKZ6WFdZKzNRQ  
3o4UGxpBZUZeWtJZlloebVMNzQ5MW9tVmZjeUlFuZBwB1hqt1Vnbl  
NsdxPMFBxRk5oODdyTwpvzJYTyg4THpmQmh1UFJIN2pwVxd1Y0  
pvbG1GU3ZlYUfh1NjNW1NWnl1ZUZvMlJlV3UxWlhuK1pzRzhEekdM  
R3lCc19XYkxuNv01NktuZzdYvBzVMDRGTvIXNWE1VG95VVNDbi9ua1  
dYUkpb6UlhRcgFwQUtmgVNPeKFDUEvtak9GyNzEQWNkbGMrWVpBK  
0U1bXVBYVmTnNpCvQWn3pzNkt0Qk9rbWRTNVJMXVvOENBd0VB  
QVE9PQotLS0tLUVORCBQVUJMSUMgS0VZLS0tLS0=, 'NN' => 'adam',  
)

Don't allow  
 yo6sgmfq7pfvp2e4kcuhtfg7wfllt63igwcuuhbmuqm6lu3a3a.b32.i2p  
to prompt you again

OK

SEND

GET

DSC

cmd

123

Encrypted\_Message

Any\_other\_remarks

Misc

adam session\_id:adam-fer4f9qk69m9nj562u7kj5n2u9\_31  
session\_id:adam-fer4f9qk69m9nj562u7kj5n2u9\_31

## Omni\*Search Demo

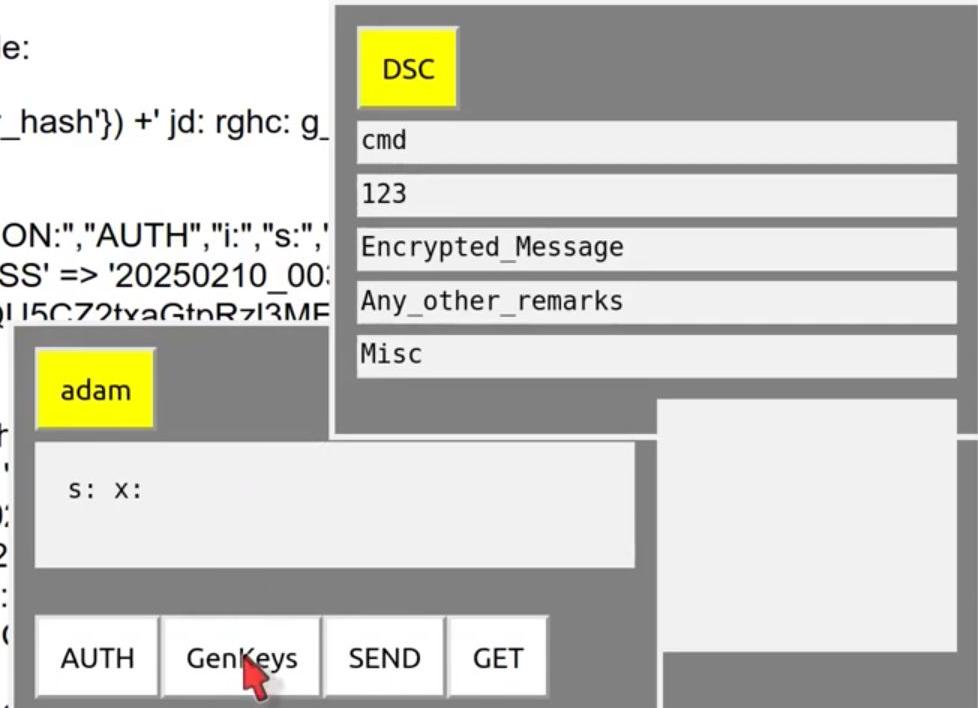
Copy the following code and execute it in the browser console:

```
f('B_F' + JSON.stringify({w:'like', t:'recipient_hash', f:'sender_hash'}) + 'jd:rghc:g')
```

1602 Phos 1611 S 1621 B F start omni:

```
fgl_s 340 < 2 > array ( 0 => array ( 0 => '{"B_AUTH": ["SESSION:", "AUTH", "i:", "s:", "lcdw:", 5 => 'B_AUTH', ], 1 => array ( 'SP' => '9fa64665', 'PASS' => '20250210_00: 'LS0tLS1CRUdJTiBQVUJMSUMgS0VZLS0tLS0KTUIJQ0lqQl5C72tvaGtnRzI3ME 'NN' => 'adam', ), )
```

```
fgl_s 340 < 4 > array ( 0 => array ( 0 => '{"B_AUTH1":  
["SESSION:","AUTH","i:","PBK","i:","o_hash.json","l_cdwjs","t  
1 => 'jd:', 2 => 'cdw:', 3 => 'am:', 4 => 'lcdw:', 5 => 'B_AUTH1'  
array ( 'a_cmd' => array ( 0 => 's:', 1 => ',', ), 't_f0' => '2025-0:  
'Graph/dmeta/CXAGcRKevA==', ) [{"a_cmd":["s:",",""], "t_f0":"2  
{"r":"Graph\dm\meta\CXAGcRKevA==", "t":"2025-02-10T00:38:  
fgl_s 340 < 2 > array ( 0 => array ( 0 => '{"B_AUTH":["SESSI  
'lcdw:', 5 => 'B_AUTH', ), 1 => NULL, )  
fgl_s 340 < 2 > array ( 0 => array ( 0 => '{"B_AUTH":["SESSI  
'lcdw:', 5 => 'B_AUTH' ), 1 => array ( 'SP' => 'e3235b7f', 'PAS
```



## Front End

PBK (Public Key)

Trillion Dollar Slide:  
defines user identifier!!

Ciphertext

+  
PVK (Private Key)

Secret Phrase (SP)

## Back End

PBK (Public Key)

+

Secret Phrase (SP)

Ciphertext

Foundation of Bitcoin  
user identifier:  
hash of public key

Secret Phrase (SP)

## Additional Options

Backend may send its PBK (B\_PBK) (prefix B) to frontend, so that

- front end may encrypt deciphered secret phrase (SP) with B\_PBK,
- then send to Backend, which then decipher it using Backend private key (B\_PVK).

```
"req_auth_sc":["b64d:","4","orpb:","hex:","dup2:","rsa:","2","pick:","lkey:","swap:","r
ecr:","b64e :","ON","ECHO","bv:","ec:","2","pick:","SP","ka:","AUTH","uss:","",""]
,"dts:","over:","over:","jsp:","rot:","rot:","swap:","_","jsd:","=chat","dc0:","_","jsd:","rot:",
"rot:","jsp:","jnl:","swap:","log","\/","jsd:","wa:",";"],"nick":["NN","ka:","AUTH","uss:",";"],"req
auth_sc":["b64d:","4","orpb:","hex:","dup2:","rsa:","2","pick:","lkey:","swap:","recr:","b64e:","ON
","ECHO","bv:","ec:","2","pick:","SP","ka:","AUTH","uss:",";"],"b_cdw":["=_\Backup\o_cdw.json\/",
"dw.json","3","mssx:","o_cdw.json","s:","copy:",";"],"s_ss":["SESSION:","s:",";"],"uu_ls":[".\Graph\
","swap:","2","mssx:","mkdir:",";"],"a_cache":["Web
","0","i:","\/","explode:","apop:","cache","apush
:","rgrep:","\/body\/","ig:","\/div\/","ig:","\/h
window\/","ig:","av:",";"],"a_keyword":["UUID","sw
ref\/","ig:","src\/","ig:","width\/","ig:","\"
553 function fgl_lkey()
554 {
555     global $S;
556     $keystr = array_pop($S);
557     $key = end($S);
558     $key->loadKey($keystr);
559 }
560 function fgl_recr()
561 {
562     global $S;
563     $str = array_pop($S);
564     $key = end($S);
565     $key->setHash('sha256');
566     $key->setMGFHash('sha256');
567     $S[] = $key->encrypt($str);
568 }
```

Open



Save



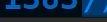
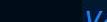
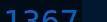
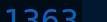
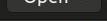
libfgl.php  
/var/www/dmeta/o/auth

```
1268 }  
1269 }  
1270 }  
1271 }  
1272  
1273 function FGL($a)  
1274 {  
1275     global $argv, $S, $SS, $xk, $xs, $SC, $SL, $CDW, $BV;  
1276     $a = preg_replace('/\s+/', ' ', $a);  
1277     $a = explode(' ', trim($a));  
1278     $SS[] = array(0, $a);  
1279     if (isset($D)) {  
1280         echo "\n\n" . __FUNCTION__ . " " . __LINE__ . " " . var_src($SS);  
1281     }  
1282     $xc = count($SS);  
1283     $xk = &$SS[$xc - 1][0];  
1284     $xs = &$SS[$xc - 1][1];  
1285     $xl = count($SS[$xc - 1][1]);  
1286     $vk = $xk;  
1287     $Z = $xl;  
1288     if (isset($D)) {  
1289         echo "\n" . __FUNCTION__ . " start " . __LINE__ . " " . $vk . " " . $Z . " " . $S . " " .  
var_src($SS) . "\n";  
1290         $S[] = $vk;  
1291     }  
1292 }
```

libfgl.php  
/var/www/dmeta/o/auth

```
var_src($$) . "\n";
    $S[] = $vk;
    $t = array_pop($S);
    $CC = $t[0];
}
do {
    // 20230609 Exception handling / errors etc.
    if (gettype(end($S))=="string") {
        if (substr(end($S),0,9)=="__ERROR__") { // exit; // break;
            echo "\n\n  FGL :post ".array_search(":post", $S[0])." ".var_src($BV)." ".var_src(
$S)."";
            // exit; // break;
            // array_pop($S);
            /*
            $xk=array_search(":post", $S[0])+1;
            $xc = count($S);
            // $xk =& $S[$xc - 1][0];
            $xs =& $S[$xc - 1][1];
            $xl = count($S[$xc - 1][1]);
            $vk = $xk;
            */
            echo "  ERROR xl ".$xl."  xk ".$xk."  xs ".var_src($xs)."  SS ".var_src($S);
        }
    }
}
```

Open



libfgl.php  
/var/www/dmeta/o/auth

Save

1361 } else {  
1362 if (\$v[\$l - 1] == ":") {  
1363 \$l = strlen(\$v);  
1364 \$fn = substr(\$v, 0, \$l - 1);  
1365 if (function\_exists("fgl\_" . \$fn)) {  
1366 if (isset(\$D)) {  
1367 echo " xs: line " . \_\_LINE\_\_ . " fgl\_" . \$fn . " ";  
1368 }  
1369  
1370  
1371 // if (gettype(end(\$S))=="string") if (substr(end(\$S),0,9)=="\_ERROR\_") { // exit;  
1372 /\* if (gettype(end(\$S))=="string") if (substr(end(\$S),0,6)=="==POST") { // exit;  
1373 echo "\n xs: " . \_\_LINE\_\_ . " ";  
1374 echo " v " . \$v . " length " . strlen(\$v) . " ord " . ord(\$v) . " vk " . \$vk . " " .  
1375 var\_src(\$xs);  
1376 exit;  
1377 }  
1378 \*/  
1379  
1380 call\_user\_func("fgl\_" . \$fn);  
1381  
1382  
1383 if (gettype(end(\$S))=="string") if (substr(end(\$S),0,9)=="\_ERROR\_") { // exit;

```
1272
1273 function FGL($a)
1274 {
1275     global $argv, $S, $SS, $xk, $xs, $SC, $SL, $CDW, $BV;
1276     $a = preg_replace('/\s+/', ' ', $a);
1277     $a = explode(' ', trim($a));
1278     $SS[] = array(0, $a);
1293 }
1294 do {
1295
1296     // 20230609 Exception handling / errors etc.
1297     if (gettype(end($S))=="string") {
1298         if (substr(end($S),0,9)=="__ERROR__") { // exit; // break;
1379
1380             call_user_func("fgl_" . $fn);
1381
1395
500             } else {
501                 if ($v == '====') {
502                     $S[] = array_pop($S) === array_pop($S);
503                 } else {
504                     $S[] = $v;
505                 }
506             }
507         }
508     }
509     $vk++;
510     if ($vk > $xl) break;
511
1538 } while ($vk < $xl); // LOL $xk++ but $vk is not?
1539
1540 array_pop($SS);
1541 }
```

```
>> t()  
< 7929297801672961   
>> f_("a h53:")
```

```
>> t()  
< 4438178491138429   
>> f_("A h53:")
```

Hilbert Hotel: Hash code on room, cabinet, files (physical paper file folders)  
- Ask Google Gemini to generate.

Example of hash numbers in browser console

Relate all these to Hilbert Hotel Cabinets

```
>> Math.pow(2,53)  
< 9007199254740992
```

```
>> t()  
< "D8R_9GoBfQ=="  
>> f_("A h53: b64: path:")
```

```
>> t()  
< "HCungsl5AQ=="  
>> f_("a h53: b64: path:")
```

<https://github.com/omnixtar/omnixtar.github.io/blob/main/js/omni.js>



Code

Issues 3

Pull requests

Actions

Projects

Wiki

Security 1

...



main ▾

omnixtar.github.io / js / omni.js



Go to file



omnixtar f\_path js/omni.js ✓

0a83e06 · 3 weeks ago



279 lines (250 loc) · 9.17 KB

Code

Blame

Raw



```
1  ↴  function include(url) {  
2      var s = document.createElement("script");  
3      s.setAttribute("type", "text/javascript");  
4      s.setAttribute("src", url);  
5      document.body.appendChild(s);  
6  }  
7  
8  // 2025-10-24
```

[main](#) ▾[omnixtar.github.io / js / omni.js](#)

↑ Top

[Code](#) [Blame](#)[Raw](#)

```
71
72  ↘  var phosinit=function(){
73  ↘  var Phos=function(){
74      // var S=[] // this is local, not accessible outside
75      var $ = this // macro; 20250804 use var to localise, $ is used by jquery
76      this.S=[]
77      var S=this.S // still need var S for local code access
78      S[0]={}
79      var S0=S[0]
80      var $CDW = {}
81      S[0].$CDW = $CDW;
82      S0.skip = 0;
83      S0.CDW = [];
84      S0.dlb = {};
85  ↘  var FGLA = function($WA) {
86      // arguments[0].split(' ').map(e=>{
87      var c_cdw=false; var i=0, ic, W=$WA;
88      $WA.map(e=>{ // WORD ARRAY
89          console.log(i, e);
90          var $v=e.$v[i].$1-$v.length;
```





main ▾

[omnixtar.github.io / js / omni.js](https://omnixtar.github.io/js/omni.js)

↑ Top

[main](#) ▾[omnixtar.github.io / js / omni.js](#)[↑ Top](#)[Code](#)[Blame](#)[Raw](#)

```
72  var phosinit=function(){
85      var FGLA = function($WA) {
112          else if (in_array($v, array_keys($CDW))) {
113              var $WA = $CDW[$v];
114              if (end($WA) == ';') array_pop($WA); // remove ; in definition before execution
115              S0.CDW.push([ $v, $vk, {} ]); // console.log(1176, 'before FGLA', JSON.stringify(S0.CDW));
116              FGLA($WA); S0.CDW.pop(); // console.log(1183, 'after FGLA', JSON.stringify(S0.CDW));
117              S0.cda = end(S0.CDW);
118          }
119          else if ($v[$l - 1] == ':') { // colon suffix word after symbol else : will fail
120              var $fn = $v.substr(0, $l-1);
121              if (typeof eval("f_"+$fn)!=="undefined") // console.log('is func', $v, typeof eval("f_"+$fn));
122                  eval("f_"+$fn+"()")
123              } else s.push(e)
124          }
125          i++
126      }
127  }
128 }
```

[main](#) ▾[omnixtar.github.io / js / omni.js](#)[↑ Top](#)[Code](#)[Blame](#)[Raw](#)

```
9  function f_path() // path safe b64
10 {
11     // $a = array_pop($S);
12     var $a = s.pop()
13     // echo '  in path: '.$a." ".strpos($a, '/')."\n";
14     // if (strpos($a, '/')>=0) { $a = str_replace('/', '_', $a); } // echo $a; }
15     if ($a.indexOf('/')>=0) { $a = $a.replaceAll('/', '_'); }
16     // if (strpos($a, '+')>=0) $a = str_replace('+', '-', $a);
17     if ($a.indexOf('+')>=0) { $a = $a.replaceAll('+', '-'); }
18     s.push($a)
19 }
20
21 // mkeval_l("omnihash")
22 // DmII2G1lw==
23 // var omnihash=function(){
24 var cyrb53=function(str, seed = 0){
25     let h1 = 0xdeadbeef ^ seed, h2 = 0x41c6ce57 ^ seed;
26     for (let i = 0, ch; i < str.length; i++) {
27         ch = str.charCodeAt(i);
28         h1 = Math.imul(h1 ^ ch, 0xFFFFFFFF);
```

[main](#) ▾[omnixtar.github.io / js / omni.js](#)

↑ Top

[Code](#)[Blame](#)[Raw](#)

```
24  var cyrb53=function(str, seed = 0){  
25      let h1 = 0xdeadbeef ^ seed, h2 = 0x41c6ce57 ^ seed;  
26      for (let i = 0, ch; i < str.length; i++) {  
27          ch = str.charCodeAt(i);  
28          h1 = Math.imul(h1 ^ ch, 2654435761);  
29          h2 = Math.imul(h2 ^ ch, 1597334677);  
30      }  
31      h1 = Math.imul(h1 ^ (h1>>>16), 2246822507) ^ Math.imul(h2 ^ (h2>>>13), 3266489909);  
32      h2 = Math.imul(h2 ^ (h2>>>16), 2246822507) ^ Math.imul(h1 ^ (h1>>>13), 3266489909);  
33      return 4294967296 * (2097151 & h2) + (h1>>>0);  
34  }  
35  var bnToB64=function bnToB64(bn){  
36      var hex = BigInt(bn).toString(16);  
37      if (hex.length % 2) { hex = '0' + hex; }  
38      var bin = [];  
39      var i = 0;  
40      var d;  
41      var b;  
42      while (i < hex.length) {
```



main ▾

omnixtar.github.io / js / omni.js

↑ Top

Code

Blame

Raw



```
35  var bnToB64=function bnToB64(bn){  
36      var hex = BigInt(bn).toString(16);  
37      if (hex.length % 2) { hex = '0' + hex; }  
38      var bin = [];  
39      var i = 0;  
40      var d;  
41      var b;  
42      while (i < hex.length) {  
43          d = parseInt(hex.slice(i, i + 2), 16);  
44          b = String.fromCharCode(d);  
45          bin.push(b);  
46          i += 2;  
47      }  
48      return btoa(bin.join(''));  
49  }  
50  var mkeval_l=function(FN){ // *_l means long form before minified  
51      var j1={n:FN}  
52      j1.s=eval(j1.n).toString()  
53      j1.t=(new Date()).toISOString()  
54      var j2={s:JSON.stringify(j1),h:bnToB64(cyrb53(JSON.stringify(j1)))} }
```



14ce0dd ▾

omnixtar.github.io / h / Phos / librsa.js

↑ Top

Code Blame

Raw



```
193     }
194
195     const cyrb53 = function(str, seed = 0) {
196         let h1 = 0xdeadbeef ^ seed, h2 = 0x41c6ce57 ^ seed;
197         for (let i = 0, ch; i < str.length; i++) {
198             ch = str.charCodeAt(i);
199             h1 = Math.imul(h1 ^ ch, 2654435761);
200             h2 = Math.imul(h2 ^ ch, 1597334677);
201         }
202         h1 = Math.imul(h1 ^ (h1>>>16), 2246822507) ^ Math.imul(h2 ^ (h2>>>13), 3266489909);
203         h2 = Math.imul(h2 ^ (h2>>>16), 2246822507) ^ Math.imul(h1 ^ (h1>>>13), 3266489909);
204         return 4294967296 * (2097151 & h2) + (h1>>>0);
205     };
206
207     ...
208     function f_h53()
209     {
210         var S = M.S;
211         S.push(cyrb53(S.pop()));
212     }
213
```



14ce0dd ▾

omnixtar.github.io / contract / js / omni.js

↑ Top

Code Blame

Raw



```
8     // 2025-10-24
9     function f_h53()
10    {
11        var s = s;
12        s.push(cyrb53(s.pop()));
13    }
14    var f_b64=function(){s.push(bnToB64(s.pop()))}
15    function f_path() // path safe b64
16    {
17        // $a = array_pop($s);
18        var $a = s.pop()
19        // echo ' in path: '.$a." ".strpos($a, '/')."\n";
20        // if (strpos($a, '/')>=0) { $a = str_replace('/', '_', $a); } // echo $a; }
21        if ($a.indexOf('/')>=0) { $a = $a.replaceAll('/', '_'); }
22        // if (strpos($a, '+')>=0) $a = str_replace('+', '-', $a);
23        if ($a.indexOf('+')>=0) { $a = $a.replaceAll('+', '-'); }
24        s.push($a)
25    }
26
27
```



omnixtar.github.io/h/?ECClo9IBig==

New Chrome available

# Omni\* Omnihash

[Omnihash](#)[GenKeys](#)[Save](#)[Help](#)

Load Keypair

 No file chosen

Your Hash is H-dYqaYMkQ==

Your Parent Hash is ECClo9IBig==

You have 0 downlines.

```
{"x":"REG","p":"root","h":"H-dYqaYMkQ==","t":"2025-12-20T11:42:04.749Z"}
```

- You may copy & save this DJSON



Infinity Trains: Coaches  
labeled with hash numbers



Coach contains Infinite  
Cabinet

# Hilbert World

Hilbert world

- Infinite ships

Universal Turing Machine

- Infinite tape (memory storage)

Hilbert Metatron / Hilbertron

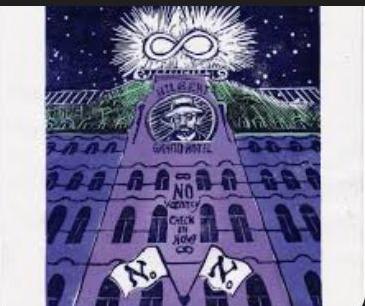
- Tron = thing
- Metatron = meta-thing
- (non-biblical)





Medium

Hilbert's Hotel: The Infinite Spaceship ...



Instagram

mathematician David Hilbe...



Chapman Newsroom - Chapman Univ...

Changing California" exhibition ope...



Pageant of the Masters

Hilbert Museum Celebrate C...



Eldar Zakirov

Hilbert Astoria. Co...



Daily Bulletin

Beach mural by Millard Sheets was...



Los Angeles Times

Looking back: Hilbert Museum offer...



Bandcamp

Hotel Hilbert | THE AFT...



Tripadvisor

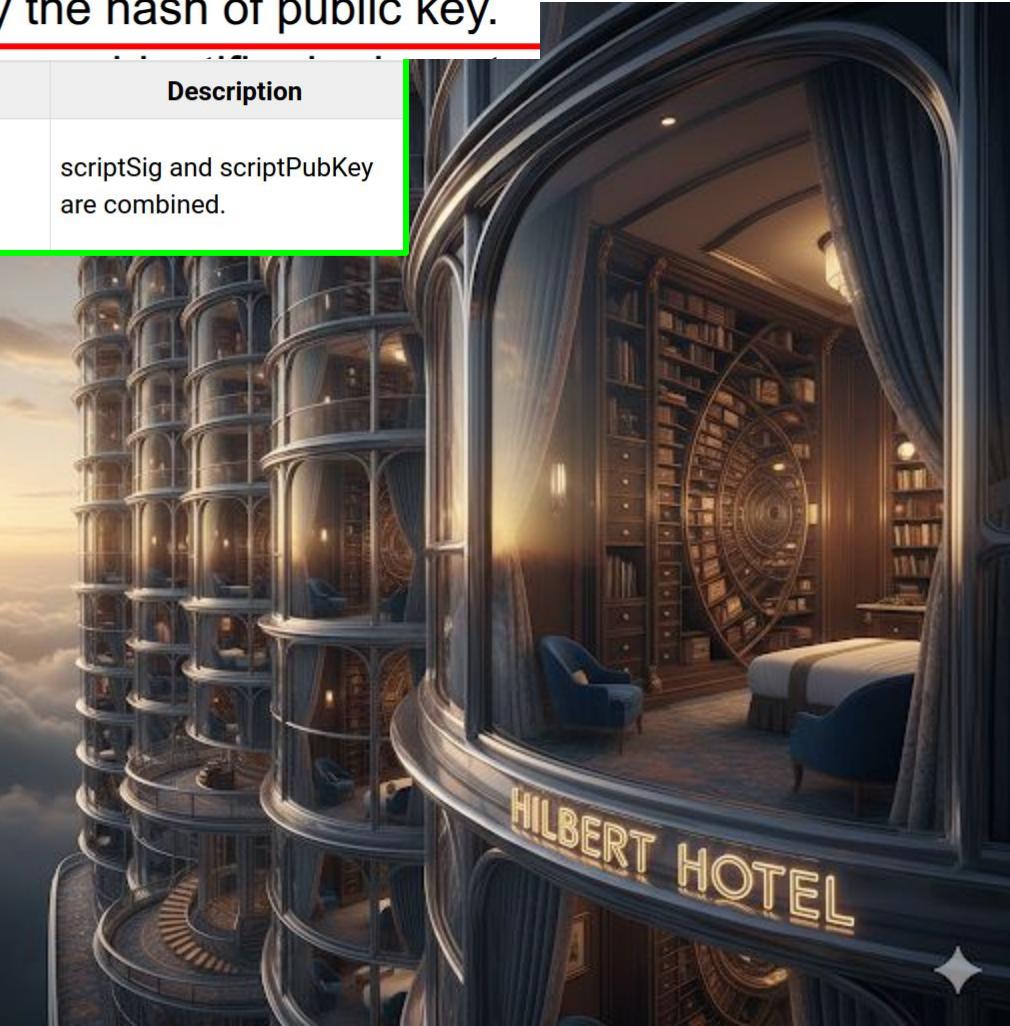
HILBERT MU...

Bitcoin uses FORTH like syntax to verify the hash of public key.

Stack	Script	Description
Empty.	<sig> <pubKey> OP_DUP OP_HASH160 <pubKeyHash> OP_EQUALVERIFY OP_CHECKSIG	scriptSig and scriptPubKey are combined.

### Phoscript-Linux/C-FORTH Sandwich Model

1. Minimal VM: Linux Java I2P Apache php 
2. Sandwich Model (user space programs):
  - a. Top: P2C (Phos to C) P2J (Phos to Java)  
[ start: 0, end: replace host language functions ]
  - b. Middle: C-lib, J-lib (Java), PHP-lib in .o (Linux)  
[ start: 100%, end: replaced by Top & Bottom ]
  - c. Bottom: need equivalent F-lib (FORTH)  
[ start: 0, end: replace Middle layer host ]
3. Replace Kernel? (Need experience from Phos)
  - a. Use P2C to replace C code with Phoscript
  - b. Write FORTH code to replace low level C code



# Infinity Trains: Coach labeled with hash numbers



Coach contains Infinite Cabinet

<https://omnixtar.github.io/h/> SVFIG memberships, memberships apps for all businesses, NGO etc. memberships data is big money, but also configurable by users, no fine print fraud by big data apps.

- Hash Filesystem examples for applications: /h /contract
- Get JSON from /h ; run hbp (h53: b64: path:) in PHP to save
- Users can add more information later

<https://omnixtar.github.io/contract/> Separation of Disclosure and Royalties

Infinity Trains = (network packets)

- Dijkstra's shunting yard algorithm: is the foundation of stack machines
- any mathematical relationships?

# Omni\* Omnihash

[Omnihash](#)[GenKeys](#)[Save](#)[Help](#)[Load Keypair](#) No file chosen

Your Hash is ECClo9IBig==

Your Parent Hash is root

You have 0 downlines.

<https://omnixtar.github.io/h/>

```
{"x":"REG","p":"root","h":"ECClo9IBig==","t":2025-12-18T11:02:03.028Z"}
```

- You may copy & save this DJSON  
Decentralised JSON ...





## Omni\*Contract: Ownership & Rights of Use of Digital Asset (Source Code)

Like

- On the Separation of Disclosure and Royalties of the Source Code July 21, 2024

1. You, a human agent of a company or government agency, may read the source code without making payments to the author or authors, but if you execute this program on behalf of your company or agency for commercial purposes, we reserve the rights to claim royalties from you or your company or agency.
2. Your copy of source code shall be attached with at least one Omni\* Hash Contract bearing the Omnihash of a Omni\* Agent and your own Omnihash, to authorise you the permissions to use or modify said source code, otherwise you shall pay maximum penalties allowed by a legal court of your jurisdiction, for the damages you have incurred for deploying the source code pertaining to clause (1).

Omnihash = Hash for any type of digital asset (photos, documents, videos, animations, simulations, PROGRAM SOURCE CODE etc.) = Hash of DJSON

DJSON Decentralised JSON = JSON string (object) containing at least one Omnihash (recursive definition)

Omni\*Web: \* = anything in Linux, cannot be copyrighted in English speaking countries? (NVIDIA, OpenAI, etc, all has "Omni" products.)

Crypto-Metaprogramming. Links.

<https://omnixtar.github.io/djson/>

**Omni\*Web: Crypto-Metaprogramming (CMP) as alternative to Model-View-Controller; towards Metanarchy**

[https://www.youtube.com/watch?v=P\\_M3PVn9J7I](https://www.youtube.com/watch?v=P_M3PVn9J7I)

DJSON Decentralised JSON is a JSON object or its encoded string where at least one of the fields is an Omnihash, representing the owner of this JSON object.

- `["2025-10-24T14:25:28.207+0000", "like", "CXAGcRKevA==", "CXAGcRKevA==", "HymWBzfj9A==", {"repo": "https://github.com/omnixtar/omnixtar.github.io/", "contract": "https://omnixtar.github.io/contract/", "ghh": "https://github.com/omnixtar/omnixtar.github.io/commit/19bb258190d57d6246840bf8ccc8957ae880e341", "datetime": "2025-10-24T04:41:21.000Z"}]`
- Omnihash: `DgV6_qnujw==`

Try this yourself:

1. Press F12 to bring up browser console.
2. Run the following code:

```
omnistart()
j0=["2025-10-24T14:25:28.207+0000", "like", "CXAGcRKevA==", "CXAGcRKevA==", "HymWBzfj9A==", {"repo": "http
s.push(JSON.stringify(j0))
f('h53: b64: path:')
s[s.length-1]
```

## Separation of Disclosure & Royalties

(C4A) Based on the dual chained property of hash codes, we arrived at one very important innovation on free software licenses, known as “separation of disclosure and royalties”, that is the author of the source code or the programmer, shall give permissions to third parties to view and analyse the source code in development and test environments, but shall reserve the rights to claim royalties if the source code and its binaries, if applicable, is used in production or commercial environments. We call this “separation of disclosure and royalties” as the conventional free software licenses mix them up. One might wonder, given hundreds of millions of very smart free software programmers out there, why is the idea of “separation of disclosure and royalties” almost unheard of, or are being restricted in isolated and small communities?