

Figure 1 shows the mechanisms of conventional social media platforms, taking in Users' data, which attract more users, and grow in revenues.

Figure 2 shows the differences between conventional social media platforms and homoiconic platforms, which take codes as User contributions, which modify the behaviours of the platform.

The Phoscript syntax shown in the following examples demonstrates homoiconism or homoinconicity, where code can be processed like data, being one of the cornerstone features of metaprogramming. This advantage is not just a technical or theoretical one, but also a practical and financial advantage, as illustrated in figures 2 and 3.

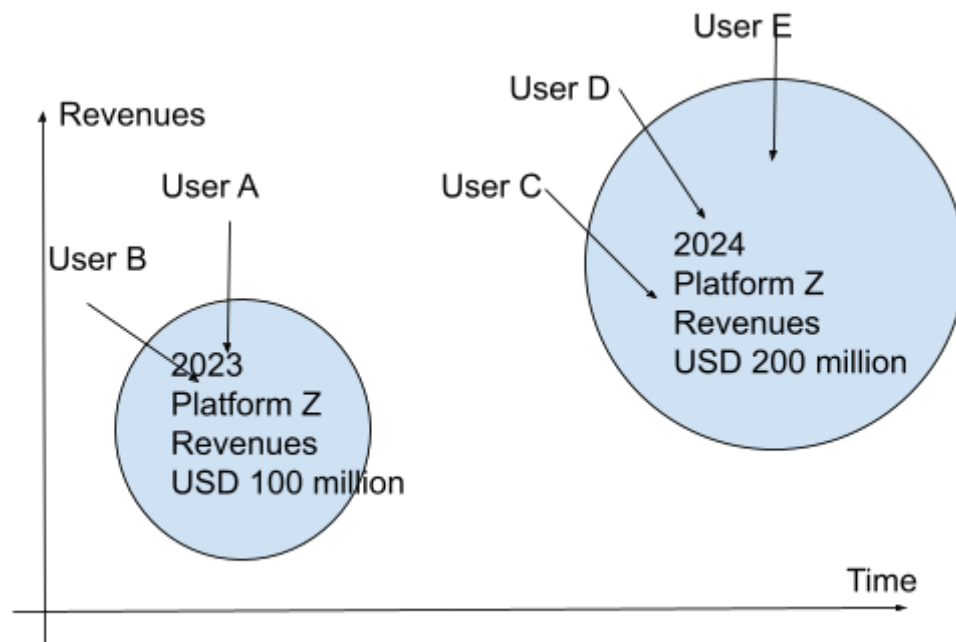


Figure 1: Conventional Social Media Platforms

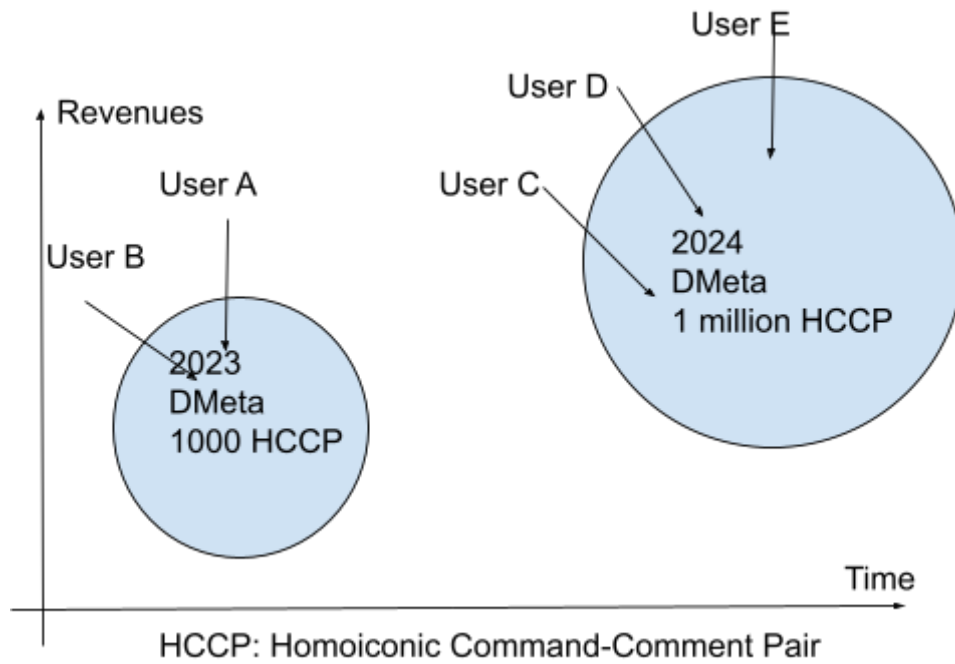


Figure 2

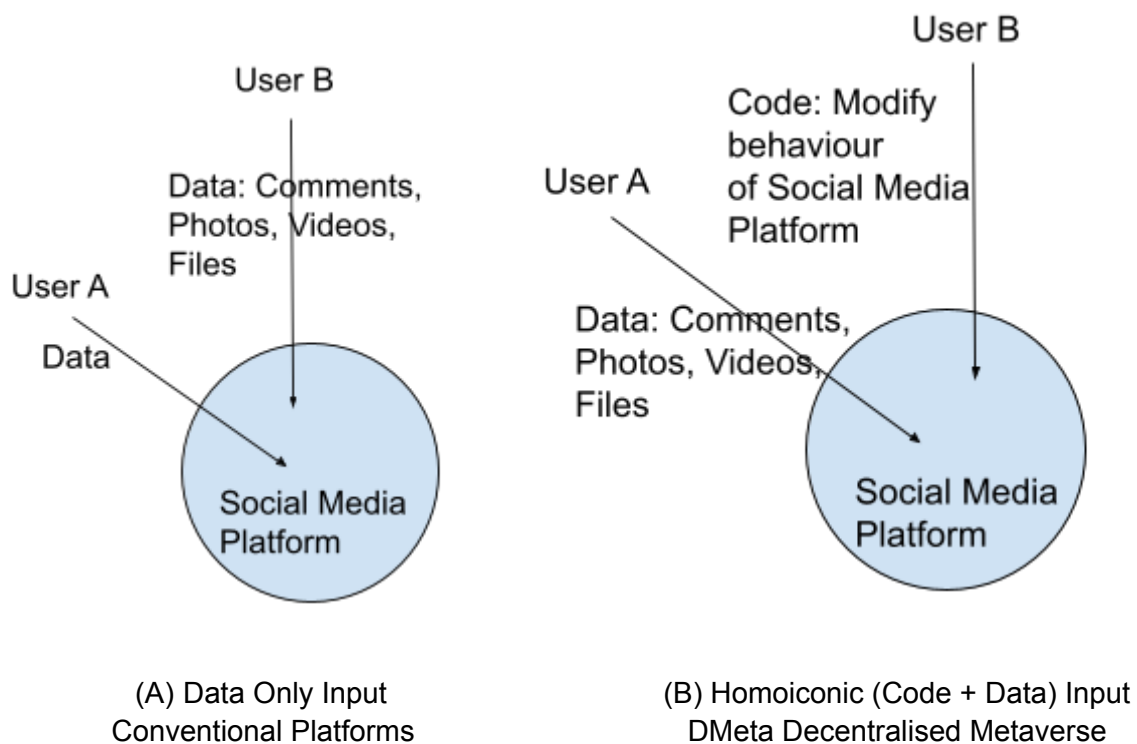


Figure 3: Conventional vs. Homoiconic Platforms

Figure 4B shows a browser console output screenshot of running DMetashell (figure 4A), hosted at a Virtualbox server, redirected via ngrok at:

- <https://godmeta.github.io/dmeta/>

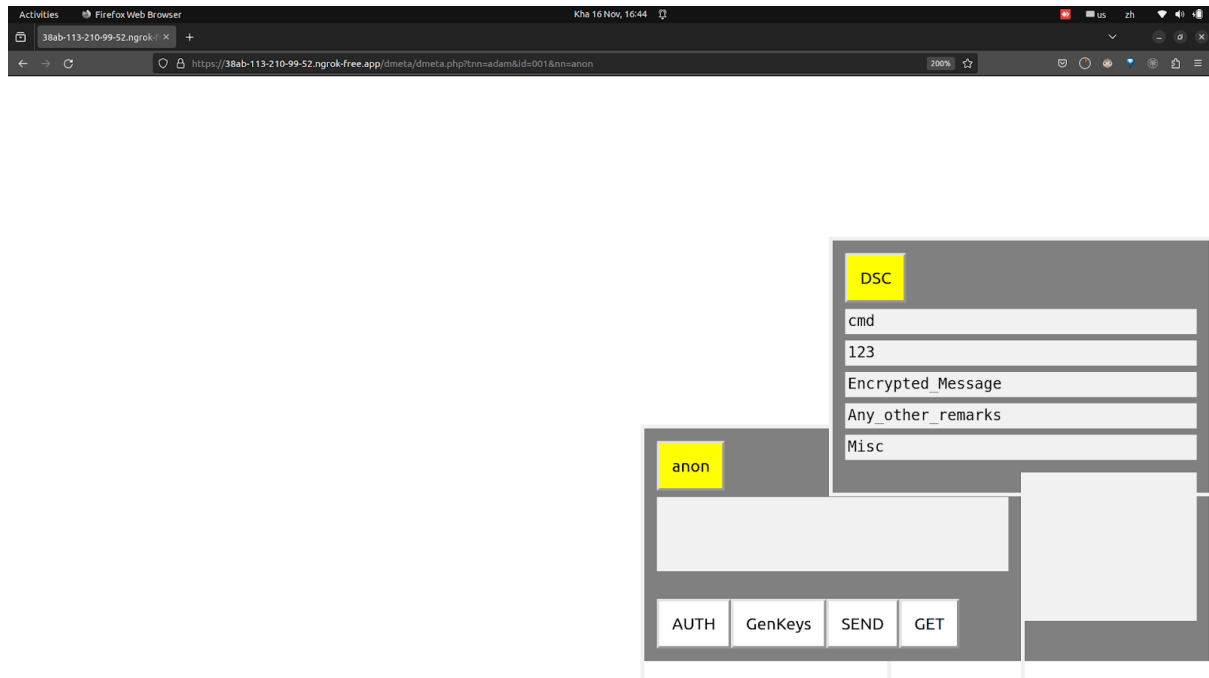


Figure 4A

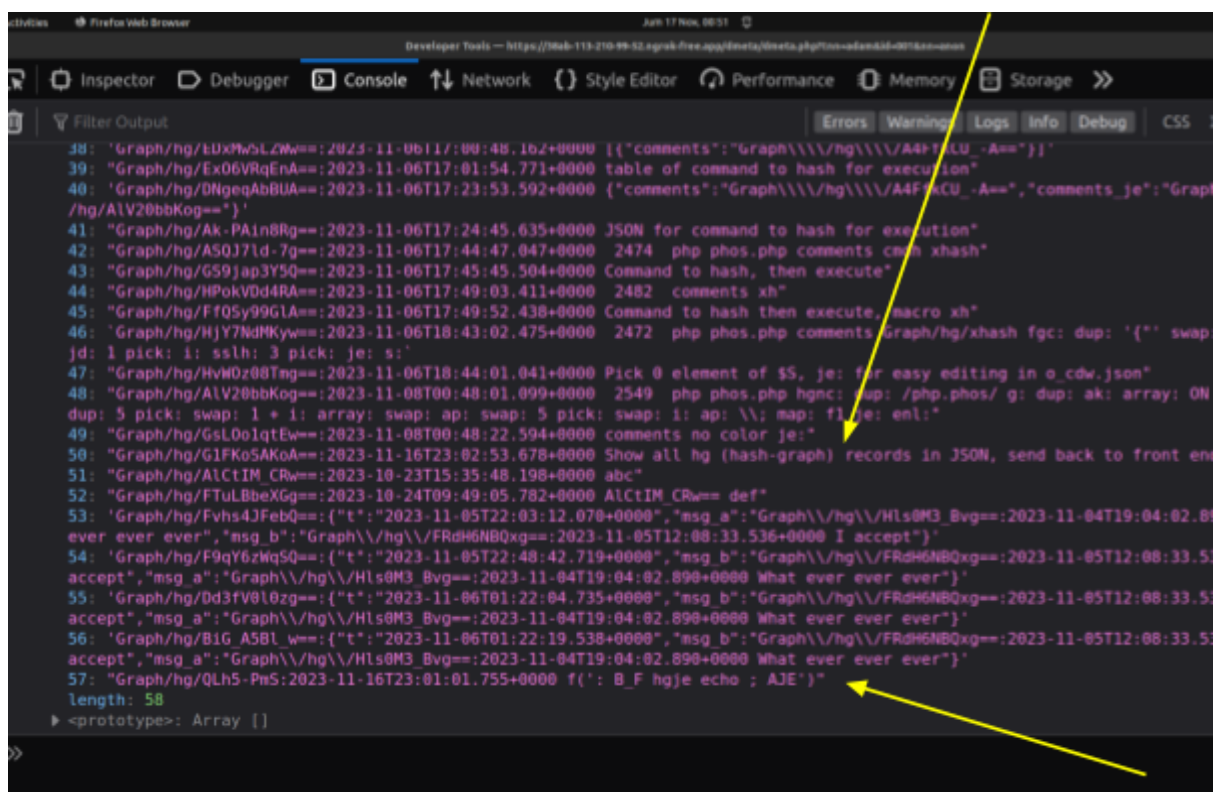


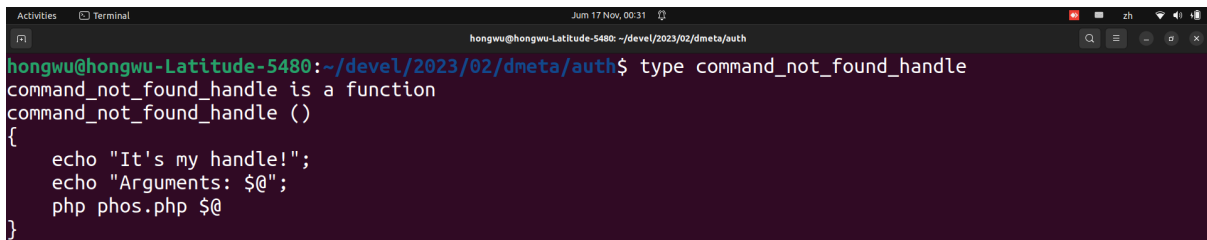
Figure 4B

1. command_not_found_handle

command_not_found_handle() is a special function in bash to which all exceptions (errors) are directed.

Its current definition can be obtained by:

```
$ type command_not_found_handle
```



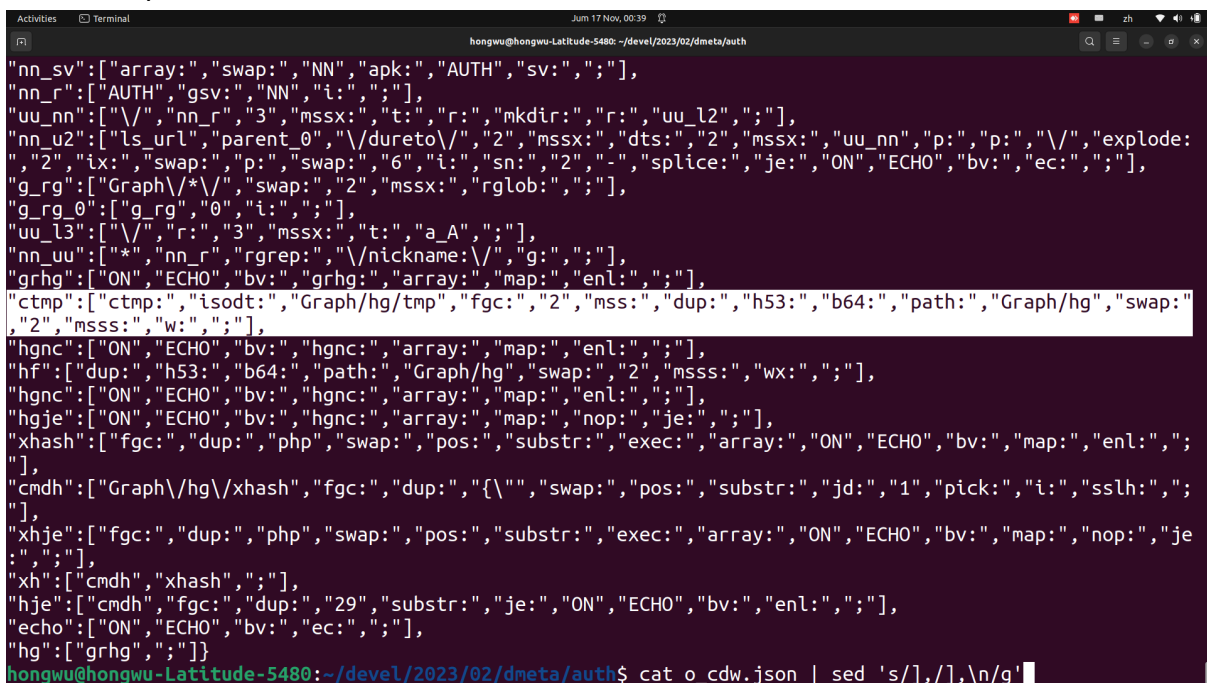
```
hongwu@hongwu-Latitude-5480: ~/devel/2023/02/dmeta/auth$ type command_not_found_handle
command_not_found_handle is a function
command_not_found_handle ()
{
    echo "It's my handle!";
    echo "Arguments: $@";
    php phos.php $@
}
```

Figure 5

For DMetashell, all arguments are redirected to “php phos.php” which then executes them in Phoscript:

```
command_not_found_handle ()
{
    echo "It's my handle!";
    echo "Arguments: $@";
    php phos.php $@
}
```

2. ctmp

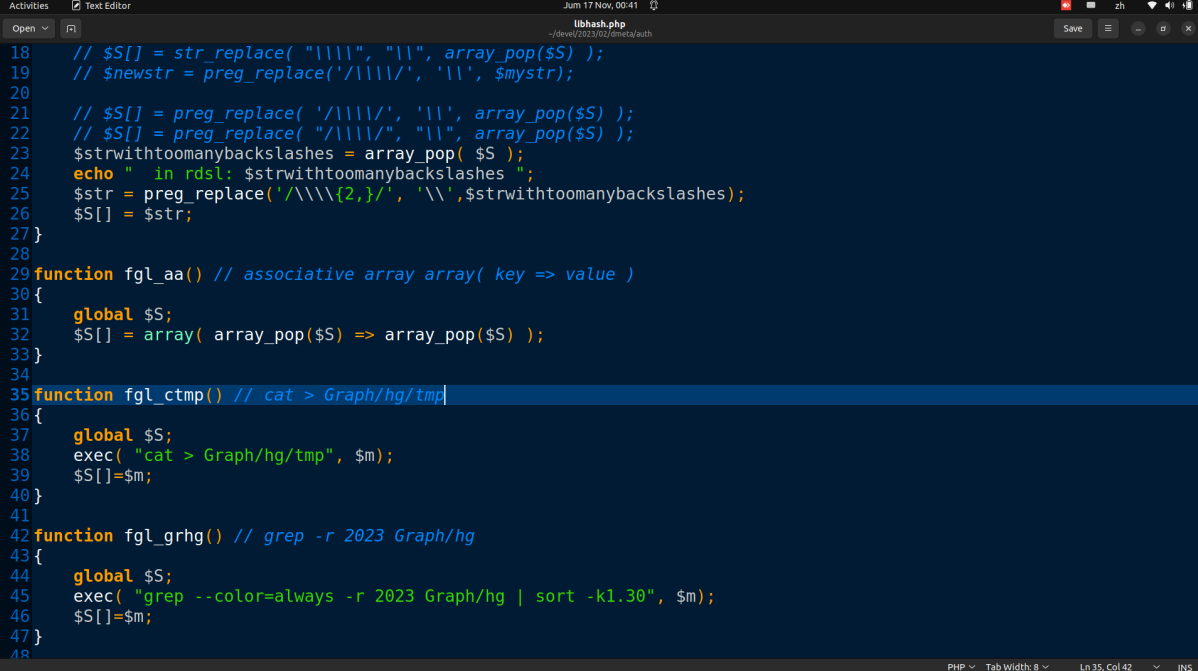


```
hongwu@hongwu-Latitude-5480: ~/devel/2023/02/dmeta/auth$ cat o_cdw.json | sed 's/[,./],\n/g'
"nn_sv":["array:", "swap:", "NN", "apk:", "AUTH", "sv:", ";"],
"nn_r":["AUTH", "gsv:", "NN", "i:", ";"],
"uu_nn":["\/", "nn_r", "3", "mssx:", "t:", "r:", "mkdir:", "r:", "uu_l2", ";"],
"nn_u2":["ls_url", "parent_0", "\dureto\/", "2", "mssx:", "dts:", "2", "mssx:", "uu_nn", "p:", "p:", "\\/", "explode:",
"2", "ix:", "swap:", "p:", "swap:", "6", "i:", "sn:", "2", "-", "splice:", "je:", "ON", "ECHO", "bv:", "ec:", ";"],
"grg":["Graph\/*\/", "swap:", "2", "mssx:", "rglob:", ";"],
"grg_0":["grg", "0", "i:", ";"],
"uu_l3":["\/", "r:", "3", "mssx:", "t:", "a_A", ";"],
"nn_uu":["*", "nn_r", "rgrep:", "\/nickname:\/", "g:", ";"],
"grhg":["ON", "ECHO", "bv:", "grhg:", "array:", "map:", "enl:", ";"],
"ctmp":["ctmp:", "isodt:", "Graph\/hg\/tmp", "fgc:", "2", "mss:", "dup:", "h53:", "b64:", "path:", "Graph\/hg", "swap:",
"2", "mss:", "w:", ";"],
"hgnc":["ON", "ECHO", "bv:", "hgnc:", "array:", "map:", "enl:", ";"],
"hf":["dup:", "h53:", "b64:", "path:", "Graph\/hg", "swap:", "2", "mss:", "wx:", ";"],
"hgnc":["ON", "ECHO", "bv:", "hgnc:", "array:", "map:", "enl:", ";"],
"hgje":["ON", "ECHO", "bv:", "hgnc:", "array:", "map:", "nop:", "je:", ";"],
"xhash":["fgc:", "dup:", "php", "swap:", "pos:", "substr:", "exec:", "array:", "ON", "ECHO", "bv:", "map:", "enl:", ";
"],
"cmdh":["Graph\/hg\/xhash", "fgc:", "dup:", "{", "swap:", "pos:", "substr:", "jd:", "1", "pick:", "i:", "sslh:", ";
"],
"jhje":["fgc:", "dup:", "php", "swap:", "pos:", "substr:", "exec:", "array:", "ON", "ECHO", "bv:", "map:", "nop:", "je:",
";", ";"],
"jh":["cmdh", "xhash", ";"],
"jhe":["cmdh", "fgc:", "dup:", "29", "substr:", "je:", "ON", "ECHO", "bv:", "enl:", ";"],
"echo":["ON", "ECHO", "bv:", "ec:", ";"],
"hg":["grhg", ";"]
hongwu@hongwu-Latitude-5480: ~/devel/2023/02/dmeta/auth$
```

Figure 6

ctmp is defined in o_cdw.json as shown in the highlighted line.

ctmp: is mapped to fgl_ctmp() as shown below:

A screenshot of a text editor window titled 'libhash.php' showing PHP code. The code defines several functions. The function 'fgl_ctmp()' is highlighted in blue. It takes a parameter '\$m' and executes the command 'cat > Graph/hg/tmp' with '\$m' as input. The code also includes other functions like 'fgl_aa()' and 'fgl_grhg()' and various string manipulation operations using 'preg_replace' and 'array_pop'.

```
18 // $S[] = str_replace( "||||", "||", array_pop($S) );
19 // $newstr = preg_replace('/||||/', '||', $mystr);
20
21 // $S[] = preg_replace( '/||||/', '||', array_pop($S) );
22 // $S[] = preg_replace( "/||||/", "||", array_pop($S) );
23 $strwithtoomanybackslashes = array_pop( $S );
24 echo " in rdsl: $strwithtoomanybackslashes ";
25 $str = preg_replace('/\\\\\\\\{2,}/', '\\\\', $strwithtoomanybackslashes);
26 $S[] = $str;
27 }
28
29 function fgl_aa() // associative array array( key => value )
30 {
31     global $S;
32     $S[] = array( array_pop($S) => array_pop($S) );
33 }
34
35 function fgl_ctmp() // cat > Graph/hg/tmp
36 {
37     global $S;
38     exec( "cat > Graph/hg/tmp", $m);
39     $S[]=$m;
40 }
41
42 function fgl_grhg() // grep -r 2023 Graph/hg
43 {
44     global $S;
45     exec( "grep --color=always -r 2023 Graph/hg | sort -k1.30", $m);
46     $S[]=$m;
47 }
48
```

Figure 7

It simply save the texts entered by user to Graph/hg/tmp. The subsequent commands then generate a base64 53 bit hash string from the texts, and save the texts using the hash string as the filename.

This is equivalent to creating a temporary variable in metaprogramming.

3. The following texts are saved using ctmp (figure 8):

f(: B_F hgje echo ; AJE')

Show all hg (hash-graph) records in JSON, send back to front end.

Referring to figures 4A and 4B, the output can be obtained by entering the command f(....) as shown above.

These texts form a *homoiconic command-comment pair* (HCCP) in Graph/hg (hash-graph) subdirectory, which can be considered as a graph database. The texts just entered can be viewed by entering the same command in the browser console, and results shown in figure 9 (yellow arrows).

(Monty Python Disclaimer) Fans of DMeta Metasophia series might have smelled something about the HCCP acronym LOL.

```
Activities Terminal Kha 16 Nov, 23:02 zh
[
  "msg_b": "Graph\\hg\\FRdH6NBQxg==:2023-11-05T12:08:33.536+0000 I accept"}
Graph/hg/FaIXjFqs7w==:{"msg_b": "Graph\\hg\\FRdH6NBQxg==:2023-11-05T12:08:33.536+0000 I accept"}
Graph/hg/BFqu3qwBJA==:{"msg_b": "Graph\\hg\\FRdH6NBQxg==:2023-11-05T12:08:33.536+0000 I accept", "msg_a": "Graph\\hg\\Hls0M3_Bvg==:2023-11-04T19:04:02.890+0000 What ever ever ever"}
Graph/hg/D70U8kYEDA==:{"msg_b": "Graph\\hg\\Hls0M3_Bvg==:2023-11-04T19:04:02.890+0000 What ever ever ever"}
Graph/hg/Fvhs4JFeb0==:{"t": "2023-11-05T22:03:12.070+0000", "msg_a": "Graph\\hg\\Hls0M3_Bvg==:2023-11-04T19:04:02.890+0000 What ever ever ever", "msg_b": "Graph\\hg\\FRdH6NBQxg==:2023-11-05T12:08:33.536+0000 I accept"}
Graph/hg/F9qY6zWqS0==:{"t": "2023-11-05T22:48:42.719+0000", "msg_b": "Graph\\hg\\FRdH6NBQxg==:2023-11-05T12:08:33.536+0000 I accept", "msg_a": "Graph\\hg\\Hls0M3_Bvg==:2023-11-04T19:04:02.890+0000 What ever ever ever"}
Graph/hg/Dd3fv0l0zg==:{"t": "2023-11-06T01:22:04.735+0000", "msg_b": "Graph\\hg\\FRdH6NBQxg==:2023-11-05T12:08:33.536+0000 I accept", "msg_a": "Graph\\hg\\Hls0M3_Bvg==:2023-11-04T19:04:02.890+0000 What ever ever ever"}
Graph/hg/BiG_A5Bl_w==:{"t": "2023-11-06T01:22:19.538+0000", "msg_b": "Graph\\hg\\FRdH6NBQxg==:2023-11-05T12:08:33.536+0000 I accept", "msg_a": "Graph\\hg\\Hls0M3_Bvg==:2023-11-04T19:04:02.890+0000 What ever ever ever"}
hongwu@hongwu-Latitude-5480:~/devel/2023/02/dmeta/auth$ ctmp
It's my handle!
Arguments: ctmp
f(' B_F hgje echo ; AJE')
hongwu@hongwu-Latitude-5480:~/devel/2023/02/dmeta/auth$ ctmp
It's my handle!
Arguments: ctmp
Show all hg (hash-graph) records in JSON, send back to front end.
hongwu@hongwu-Latitude-5480:~/devel/2023/02/dmeta/auth$
```

Figure 8

```
38: 'Graph/hg/LUXhW5LZW==:2023-11-06T17:00:48.16z+0000 [{"comments":"'Graph\\\\"hg\\\\"A4fxCU_-A=="}]'
```

```
39: 'Graph/hg/Ex06VRqEnA==:2023-11-06T17:01:54.771+0000 table of command to hash for execution'
```

```
40: 'Graph/hg/DNgeqAbBUA==:2023-11-06T17:23:53.592+0000 {"comments":"'Graph\\\\"hg\\\\"A4fxCU_-A==","comments_jc":"'Graph\\\\"hg\\\\"A1V20bbKog=="}'
```

```
41: 'Graph/hg/Ak-PAInBRg==:2023-11-06T17:24:45.635+0000 JSON for command to hash for execution'
```

```
42: 'Graph/hg/ASQJ7ld-7g==:2023-11-06T17:44:47.047+0000 2474 php phos.php comments cmn xhash'
```

```
43: 'Graph/hg/G59jap3Y5Q==:2023-11-06T17:45:45.504+0000 Command to hash, then execute'
```

```
44: 'Graph/hg/HKpQVddARA==:2023-11-06T17:49:03.411+0000 2482 comments xh'
```

```
45: 'Graph/hg/Ff05y99GIA==:2023-11-06T17:49:52.438+0000 Command to hash then execute, macro xh'
```

```
46: 'Graph/hg/HjY7NdMKYw==:2023-11-06T18:43:02.475+0000 2472 php phos.php comments Graph/hg/xhash fgc: dup: '(' swap  
jd: 1 pick: i: sslh: 3 pick: jc: s:'
```

```
47: 'Graph/hg/HvM0z08Tng==:2023-11-06T18:44:01.041+0000 Pick 0 element of $$, jc: for easy editing in o_cdw.json'
```

```
48: 'Graph/hg/AlV20bbKog==:2023-11-08T00:48:01.099+0000 2549 php phos.php hgnc: sup: /php/phos/ g: dup: ak: array: ON  
dup: 5 pick: swap: 1 + i: array: swap: ap: swap: 5 pick: swap: i: ap: \\\: map: file: enl:'
```

```
49: 'Graph/hg/GsL0oIqtEw==:2023-11-08T00:48:22.594+0000 comments no color jc:'
```

```
50: 'Graph/hg/G1FKp5AKoA==:2023-11-16T23:02:53.678+0000 Show all hg (hash-graph) records in JSON, send back to front end'
```

```
51: 'Graph/hg/AlCtIM_CRw==:2023-10-23T15:35:48.198+0000 abc'
```

```
52: 'Graph/hg/FTuLBbxGg==:2023-10-24T09:49:05.782+0000 AlCtIM_CRw== def'
```

```
53: 'Graph/hg/Fvhs4JFebQ==:{"t":"'2023-11-05T22:03:12.078+0000","msg_a":"'Graph\\\\"hg\\\\"Hls0M3_Bvg==:2023-11-04T19:04:02.890+0000 What ever ever ever","msg_b":"'Graph\\\\"hg\\\\"FRdH6NBQxg==:2023-11-05T12:08:33.536+0000 I accept"}'
```

```
54: 'Graph/hg/F9qY6WqSQ==:{"t":"'2023-11-05T22:48:42.719+0000","msg_b":"'Graph\\\\"hg\\\\"FRdH6NBQxg==:2023-11-05T12:08:33.536+0000 What ever ever ever","msg_a":"'Graph\\\\"hg\\\\"Hls0M3_Bvg==:2023-11-04T19:04:02.890+0000 What ever ever ever"}'
```

```
55: 'Graph/hg/Dd3fV8l0zg==:{"t":"'2023-11-06T01:22:04.735+0000","msg_b":"'Graph\\\\"hg\\\\"FRdH6NBQxg==:2023-11-05T12:08:33.536+0000 What ever ever ever","msg_a":"'Graph\\\\"hg\\\\"Hls0M3_Bvg==:2023-11-04T19:04:02.890+0000 What ever ever ever"}'
```

```
56: 'Graph/hg/BtG_A5Bl_w==:{"t":"'2023-11-06T01:22:19.538+0000","msg_b":"'Graph\\\\"hg\\\\"FRdH6NBQxg==:2023-11-05T12:08:33.536+0000 What ever ever ever","msg_a":"'Graph\\\\"hg\\\\"Hls0M3_Bvg==:2023-11-04T19:04:02.890+0000 What ever ever ever"}'
```

```
57: 'Graph/hg/QLh5-PmS:2023-11-16T23:01:01.755+0000 f(': B.F hgje echo : AJE)'  
length: 58
```

```
> <prototype>: Array [ ]
```

Figure 9

Here is where things get interesting. In figure 10A and 10B, the definitions for *hgje* and *hgje*: are given, which executes the following in line 56 (figure 10B):

```
grep -r 2023 Graph/hg | sort -k1.30
```

This command sorts the lines at column 30. However, the filename for the input command happens to be shorter than others:

```
Graph/hg/QLh5-PmS:2023-11-16T23:01:01.755+0000 f(' B_F hgje echo ; AJE')
```

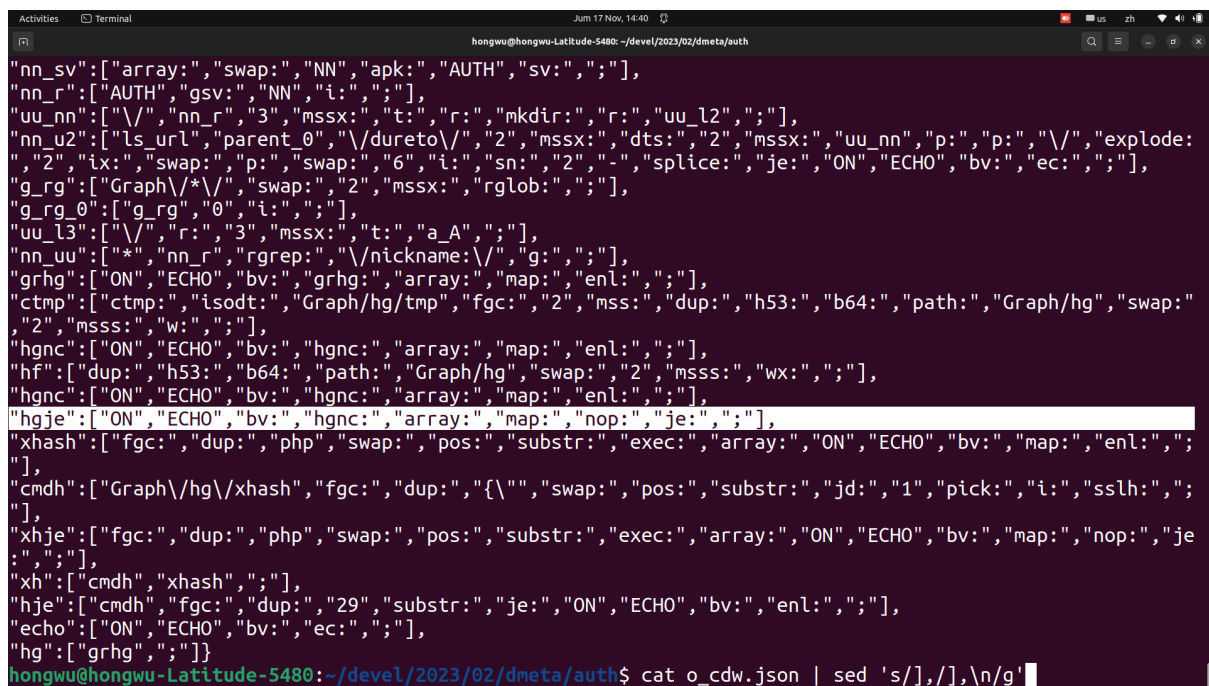
As a result, the output is wrong.

To rectify the mistakes, 2 new commands are added, whose outputs are shown in figures 11A and 11B. (See explanations on figures 4A and 4B for methods of entering the commands.)

```
f(' B_F hgst xh ; AJE')  
f(' B_F hgsd xh ; AJE')
```

The mappings to hash-graph file for *hgst* and *hgsd* are given in figure 11C, whose full definitions are given in figure 11A, lines 58 and 60.

hgst and *hgsd* first splits the output of “grep -r” on the pattern “:2023-”, hence making it independent of the length of filename, then sorts the results in ascending order (*hgst*) and descending order (*hgsd*).



```
nn_sv":["array:", "swap:", "NN", "apk:", "AUTH", "sv:", ";"],  
nn_r":["AUTH", "gsv:", "NN", "i:", ";"],  
uu_nn":["\/", "nn_r", "3", "mssx:", "t:", "r:", "mkdir:", "r:", "uu_l2", ";"],  
nn_u2":["ls_url", "parent_0", "\dureto\/", "2", "mssx:", "dts:", "2", "mssx:", "uu_nn", "p:", "p:", "\\/", "explode:",  
"2", "ix:", "swap:", "p:", "swap:", "6", "i:", "sn:", "2", "-", "splice:", "je:", "ON", "ECHO", "bv:", "ec:", ";"],  
g_rg":["Graph\/*\/", "swap:", "2", "mssx:", "rglob:", ";"],  
g_rg_0":["g_rg", "0", "i:", ";"],  
uu_l3":["\/", "r:", "3", "mssx:", "t:", "a_A", ";"],  
nn_uu":["*", "nn_r", "rgrep:", "\/nickname:\/", "g:", ";"],  
grhg":["ON", "ECHO", "bv:", "grhg:", "array:", "map:", "enl:", ";"],  
ctmp":["ctmp:", "isodt:", "Graph/hg/tmp", "fgc:", "2", "mss:", "dup:", "h53:", "b64:", "path:", "Graph/hg", "swap:",  
"2", "msss:", "w:", ";"],  
hgnc":["ON", "ECHO", "bv:", "hgnc:", "array:", "map:", "enl:", ";"],  
hf":["dup:", "h53:", "b64:", "path:", "Graph/hg", "swap:", "2", "msss:", "wx:", ";"],  
hgnc":["ON", "ECHO", "bv:", "hgnc:", "array:", "map:", "enl:", ";"],  
hgje":["ON", "ECHO", "bv:", "hgnc:", "array:", "map:", "nop:", "je:", ";"],  
xhash":["fgc:", "dup:", "php", "swap:", "pos:", "substr:", "exec:", "array:", "ON", "ECHO", "bv:", "map:", "enl:", ";  
"],  
cmdh":["Graph/hg\/xhash", "fgc:", "dup:", "{", "swap:", "pos:", "substr:", "jd:", "1", "pick:", "i:", "sslh:", ";  
"],  
xhje":["fgc:", "dup:", "php", "swap:", "pos:", "substr:", "exec:", "array:", "ON", "ECHO", "bv:", "map:", "nop:", "je:  
:", ";"],  
xh":["cmdh", "xhash", ";"],  
hje":["cmdh", "fgc:", "dup:", "29", "substr:", "je:", "ON", "ECHO", "bv:", "enl:", ";"],  
echo":["ON", "ECHO", "bv:", "ec:", ";"],  
hg":["grhg", ";"]  
hongwu@hongwu-Latitude-5480:~/devel/2023/02/dmeta/auth$ cat o_cdw.json | sed 's/./,/,n/g'
```

Figure 10A

```
Activities Text Editor Jun 17 Nov, 14:42 libhash.php
~/dev/2023/02/dmeta/auth Save
49 function fgl_grhg() // grep -r 2023 Graph/hg
50 {
51     global $S;
52     exec( "grep --color=always -r 2023 Graph/hg | sort -k1.30", $m);
53     $S[]=$m;
54 }
55
56 function fgl_hgnc() // grep -r 2023 Graph/hg no color
57 {
58     global $S;
59     exec( "grep -r 2023 Graph/hg | sort -k1.30", $m);
60     $S[]=$m;
61 }
62
63 function fgl_exec() // exec
64 {
65     global $S;
66     exec( array_pop($S), $m);
67     $S[]=$m;
68 }
69
70
71 function fgl_hist() // history
72 {
73     global $S;
74     exec( "history", $m);
75     $S[]=$m;
76 }
77
78 function fgl_man() // keyword man: manuals, from rg:
```

Figure 10B

```
Activities Firefox Web Browser Jun 17 Nov, 15:36
Developer Tools — https://38ab-113-210-99-52.ngrok-free-app/dmeta/dmeta.php?tnn=adam&ld=001&nn=anon
Inspector Debugger Console Network Style Editor Performance Memory Storage
Filter Output Errors Warnings Logs Info Debug CSS XHR Requests
▶ 41: Array [ "Graph/hg/HpWjjzMcg==", "11-06T16:19:17.339+0000 Unpack array or arrays of | comments commands | and echo" ]
▶ 42: Array [ "Graph/hg/GZBnxZre_Q==", "11-06T16:30:15.350+0000 2384 php phos.php Graph/hg/A4FfkCU_-A== fgc: dup: php swap: pos:
substr: exec: array: ON ECHO bv: map: enl:" ]
▶ 43: Array [ "Graph/hg/AnGEGbZ4IA==", "11-06T16:30:39.435+0000 Read phos commands by hash file, exec, echo." ]
▶ 44: Array [ "Graph/hg/EDxMwSLZWw==", "11-06T17:00:48.162+0000 [{"comments":"Graph\\\\\\hg\\\\\\A4FfkCU_-A=="}]' ]
▶ 45: Array [ "Graph/hg/Ex06VRqEnA==", "11-06T17:01:54.771+0000 table of command to hash for execution" ]
▶ 46: Array [ "Graph/hg/DNgeqAbBUA==", "11-06T17:23:53.592+0000 {"comments":"Graph\\\\\\hg\\\\\\A4FfkCU_-A==","comments_je":"Graph
/hg/ALV20bbKog==","htab33":"Graph/hg/Et3ygPECUQ==","hgst":"Graph/hg/CpoFL52tIg==","hgscd":"Graph/hg/HpsPg14WNQ=="}' ]
▶ 47: Array [ "Graph/hg/Ak-PAin8Rg==", "11-06T17:24:45.635+0000 JSON for command to hash for execution" ]
▶ 48: Array [ "Graph/hg/ASQJ7ld-7g==", "11-06T17:44:47.047+0000 2474 php phos.php comments cmdh xhash" ]
▶ 49: Array [ "Graph/hg/GS9jap3Y5Q==", "11-06T17:45:45.504+0000 Command to hash, then execute" ]
▶ 50: Array [ "Graph/hg/HPokVd4RA==", "11-06T17:49:03.411+0000 2482 comments xh" ]
▶ 51: Array [ "Graph/hg/FFQSy99GLA==", "11-06T17:49:52.438+0000 Command to hash then execute, macro xh" ]
▶ 52: Array [ "Graph/hg/HjY7NdMKyw==", "11-06T18:43:02.475+0000 2472 php phos.php comments Graph/hg/xhash fgc: dup: '{' swap: pos:
substr: jd: 1 pick: 1: sslh: 3 pick: je: s:' ]
▶ 53: Array [ "Graph/hg/HvW0z08Tmg==", "11-06T18:44:01.041+0000 Pick 0 element of $S, je: for easy editing in o_cdw.json" ]
▶ 54: Array [ "Graph/hg/ALV20bbKog==", "11-08T00:48:01.099+0000 2549 php phos.php hgnc: dup: /php.phos/ g: dup: ak: array: ON ECHO bv:
: f1 dup: 5 pick: swap: 1 + i: array: swap: ap: swap: 5 pick: swap: i: ap: \\; map: f1 je: enl:" ]
▶ 55: Array [ "Graph/hg/GsL0o1qtEw==", "11-08T00:48:22.594+0000 comments no color je:" ]
▶ 56: Array [ "Graph/hg/QLh5-PmS", "11-16T23:01:01.755+0000 f(': B_F hgje echo ; AJE')" ]
▶ 57: Array [ "Graph/hg/GLFKoSAKoA==", "11-16T23:02:53.678+0000 Show all hg (hash-graph) records in JSON, send back to front end." ]
▶ 58: Array [ "Graph/hg/CpoFL52tIg==", "11-17T10:55:45.936+0000 2178 php phos.php hgnc: ON ECHO bv: : f1 delim: explode: \\; array:
map: f1 dup: 0 col: swap: 1 col: swap: pop: msort: amc: array: : f2 je: enl: \\; map: nop: je: enl:" ]
▶ 59: Array [ "Graph/hg/FvnmYGJv_g==", "11-17T10:56:37.598+0000 Sort hash-graph by time, ascending." ]
▶ 60: Array [ "Graph/hg/HpsPg14WNQ==", "11-17T10:56:48.757+0000 2176 php phos.php hgnc: ON ECHO bv: : f1 delim: explode: \\; array:
map: f1 dup: 0 col: swap: 1 col: swap: pop: amsd: amc: array: : f2 je: enl: \\; map: nop: je: enl:" ]
▶ 61: Array [ "Graph/hg/Dn3ZFua9HQ==", "11-17T10:57:04.597+0000 Sort hash-graph by time, descending." ]
length: 62
▶ <prototype>: Array []
>> f(': B_F hgst xh ; AJE')
```

Figure 11A


```
Activities FirefoxWeb Browser Jun 17 Nov, 15:36
Developer Tools — https://38ab-113-210-99-52.ngrok-free-app/dmeta/dmeta.php?tnnadam&ld=001&nnanon
Inspector Debugger Console Network Style Editor Performance Memory Storage
Filter Output Errors Warnings Logs Info Debug CSS XHR Requests
▶ 41: Array [ "Graph/hg/D/0U8kYEUa==":{"msg_b":"Graph\\hg\\Hls0M3_Bvg==", "11-04T19:04:02.890+0000 What ever ever ever"} ]
▶ 42: Array [ "Graph/hg/Foyq2WD52w==":{"msg_a":"Graph\\hg\\Hls0M3_Bvg==", "11-04T19:04:02.890+0000 What ever ever ever", "msg_b":"Graph\\hg\\FRdH6NB0xg==" } ]
▶ 43: Array [ "Graph/hg/Fvhs4JFeb0==":{"t":"2023-11-05T22:03:12.070+0000", "msg_a":"Graph\\hg\\Hls0M3_Bvg==", "11-04T19:04:02.890+0000 What ever ever ever", "msg_b":"Graph\\hg\\FRdH6NB0xg==" } ]
▶ 44: Array [ "Graph/hg/Hls0M3_Bvg==", "11-04T19:04:02.890+0000 What ever ever ever" ]
▶ 45: Array [ "Graph/hg/GODK6YHGzA==", "11-04T18:42:23.654+0000 do re mi fa so 1842" ]
▶ 46: Array [ "Graph/hg/G809bqhdKg==", "11-04T18:37:36.382+0000 1837" ]
▶ 47: Array [ "Graph/hg/CiYc3Z_NeA==", "11-04T11:27:17.729+0000 Run grep -r Graph/hg in exec -- print results" ]
▶ 48: Array [ "Graph/hg/CCCBlyRPDA==", "11-04T11:24:59.720+0000 2132 php phos.php ON ECHO bv: Graph/hg/tmp fgc: exec: array: map: enl:" ]
▶ 49: Array [ "Graph/hg/AzRX8SRfDg==", "11-04T11:23:27.013+0000" ]
▶ 50: Array [ "Graph/hg/GkxSNYnZE0==", "11-04T01:16:27.248+0000 Write contents of tmp to hash file" ]
▶ 51: Array [ "Graph/hg/Bwh-gV0Ntw==", "11-04T01:15:48.864+0000 2092 php phos.php isodt: Graph/hg/tmp fgc: 2 mss: dup: h53: b64: path: Graph/hg swap: 2 msss: w: s:" ]
▶ 52: Array [ "Graph/hg/CpMxcGgqRA==", "11-04T01:14:07.302+0000 Enter comments after each line of commands into hash graph" ]
▶ 53: Array [ "Graph/hg/FuREqPKT-Q==", "11-04T01:09:58.414+0000 sort grep hg results by time preserve color" ]
▶ 54: Array [ "Graph/hg/EUg1fASd0g==", "11-04T01:08:43.689+0000 2078 grep --color=always -r 2023 Graph/hg | sort -k1.30" ]
▶ 55: Array [ "Graph/hg/DtR8pwBbyg==", "11-04T00:58:55.655+0000 get last 2 hashes" ]
▶ 56: Array [ "Graph/hg/EIrfMfxaAQ==", "11-04T00:58:18.435+0000 2063 php phos.php Graph/hg lsl: dup: 3 i: space: explode: dup: c: 1 - i: 1 pick: 1 i: space: explode: dup: c: 1 - i: 2 mss: dts: swap: 2 mss: dup: h53: b64: Graph/hg swap: 2 msss: s:" ]
▶ 57: Array [ "Graph/hg/A1oYgfJtzQ==", "11-03T22:57:56.115+0000 2046 php phos.php isodt: url 2 mss: dup: h53: b64: path: Graph/hg swap: 2 msss: s:" ]
▶ 58: Array [ "Graph/hg/Fhxv2rEtaA==", "11-03T22:57:40.796+0000 url" ]
▶ 59: Array [ "Graph/hg/C00LTeg4hw==", "11-03T16:30:26.006+0000 https://godmeta.github.io/doc/DMC1_Hash_Contract.pdf" ]
▶ 60: Array [ "Graph/hg/FTuLBbeXGg==", "10-24T09:49:05.782+0000 AlCtIM_CRw== def" ]
▶ 61: Array [ "Graph/hg/AlCtIM_CRw==", "10-23T15:35:48.198+0000 abc" ]
length: 62
▶ <prototype>: Array []
>> f(' B_F hgdsd xh ; AJE')
```

Figure 11B

```
Activities Terminal Jun 17 Nov, 15:41
hongwu@hongwu-Latitude-5480: ~/devel/2023/02/dmeta/auth$ sed 's/,/,\\n/g' Graph/hg/xhash
2023-11-06T17:23:53.592+0000 {"comments": "Graph\\hg\\A4FfkCU_-A==",
"comments_je": "Graph/hg/AlV20bbKog==",
"htab33": "Graph/hg/Et3ygPECUQ==",
"hgst": "Graph/hg/CpoF152tIg==",
"hgdsd": "Graph/hg/HpsPg14WNQ=="}
```

Figure 11C

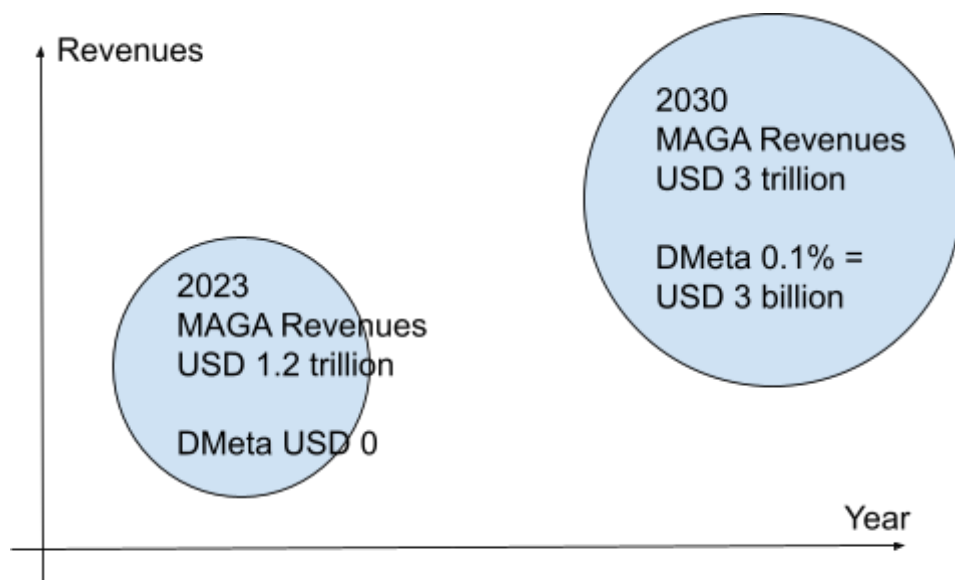


Figure 12: Projection of Revenues

Figure 12 shows how DMeta Decentralised Metaverse, a homoiconic social media platform, may grow into a billion dollar business by 2030, assuming the users and free software programmers on DMeta platforms work on cloning conventional social media functions, and make 0.1% of MAGA revenues.