ONEYE - A Cloud Operating System (DaaS)

Anjan Prasad Hari (1322978) Istvan Mate Horvath (1322910)
Ivan Serunkuma (1322525) Ranjith Kumar Rama (1322172)

Under the guidance: Prof. Dr. Christian Baun
Cloud Computing - WS2021
Frankfurt University of Applied Sciences
High Integrity Systems (MSc.)

February 12, 2021

Table of Content

- Desktop as a Service (DaaS)
- Oneye - Cloud Operating System
- Architecture
- Communication in Oneye
- Data Storage
- Users in Oneye
- Security issues and Challenges
- Oneye Improvement solutions
- Application and Demonstration

Desktop as a Service (DaaS)

Virtual desktops running in the cloud:

- Access with web-browser or portal app
- Low hardware requirements for clients
- Flexibility & mobility
- Business continuity via remote maintenance & backups
- Security & automation
- Form of Virtual Desktop Infrastructure (VDI)

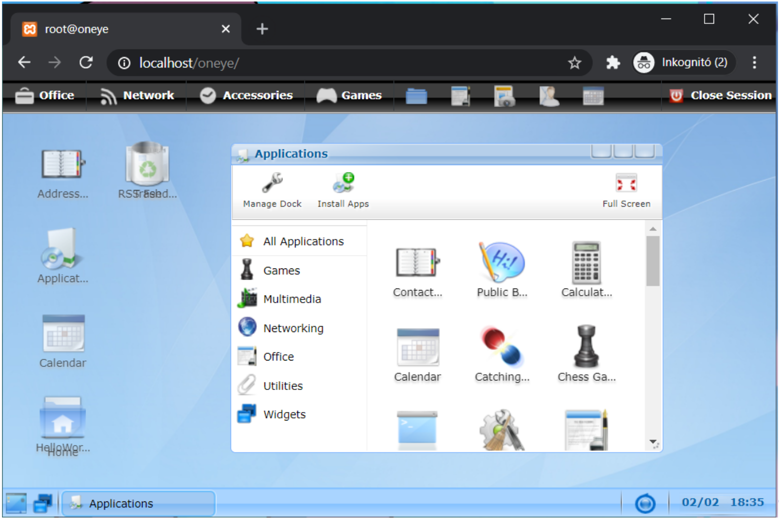
Oneye - Cloud Operating System

- DaaS solution
- Fork from eyeOS
- PHP-based with .js and .css
- Access with web-browser
- Main development 2012-2015

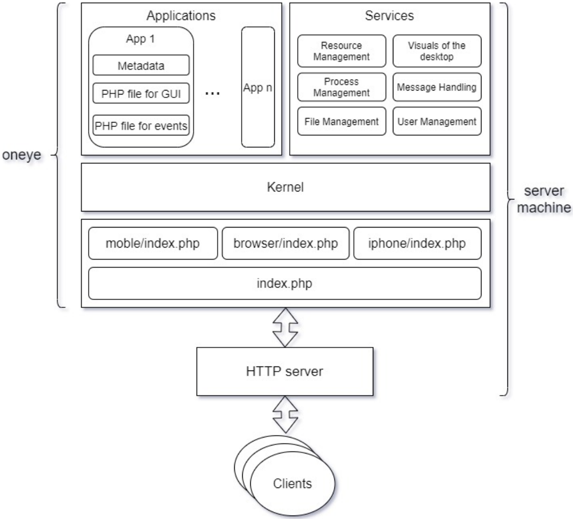
Installation

- HTTP server & PHP
- PHP 5.6 support (and newer versions)
- Copy .php files from GitHub

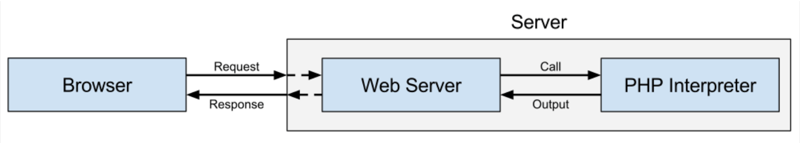
Oneye - Cloud Operating System

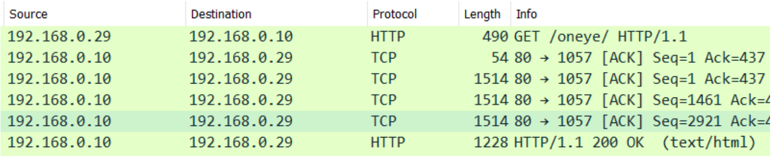


Oneye Architecture



Communication in Oneye





Source	Destination	Protocol	Length	Info
192.168.0.29	192.168.0.10	HTTP	490	GET /oneye/ HTTP/1.1
192.168.0.10	192.168.0.29	TCP	54	80 → 1057 [ACK] Seq=1 Ack=437
192.168.0.10	192.168.0.29	TCP	1514	80 → 1057 [ACK] Seq=1 Ack=437
192.168.0.10	192.168.0.29	TCP	1514	80 → 1057 [ACK] Seq=1461 Ack=4
192.168.0.10	192.168.0.29	TCP	1514	80 → 1057 [ACK] Seq=2921 Ack=4
192.168.0.10	192.168.0.29	HTTP	1228	HTTP/1.1 200 OK (text/html)

Communication in Oneye

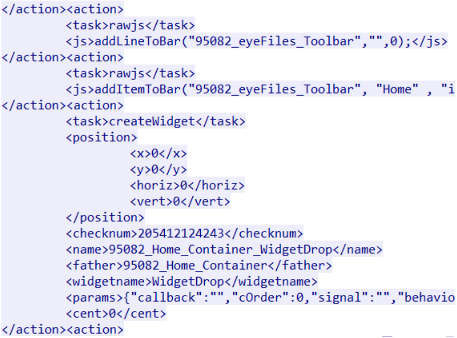
Login with root/root.
HTTP POST:

```
params=
<eyeLogin_Textbox_1_User>root</eyeLogin_Textbox_1_User>
<eyeLogin_Textbox_1_Password> cm9vdA== </eyeLogin_Textbox_1_Password>
<eyeLogin_Select_1_Language>[auto]</eyeLogin_Select_1_Language>
```

"root" with base64 is "cm9vdA==".

Communication in Oneye

```
POST /oneye/index.php?checknum=591345485083&msg=Command
HTTP/1.1
Host:  192.168.0.10
...  [HTTP header] ...
params=
   <type>top</type> <place>left</place>
   <id>element-1</id> <entry>entry-1</entry>
```

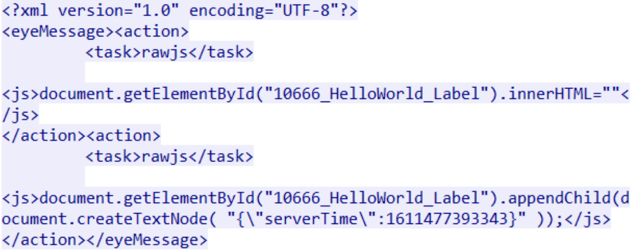


```
</action><action>
        <task>rawjs</task>
        <js>addLineToBar("95082_eyeFiles_Toolbar","",0);</js>
</action><action>
        <task>rawjs</task>
        <js>addItemToBar("95082_eyeFiles_Toolbar", "Home" , "i
</action><action>
        <task>createWidget</task>
        <position>
                <x>0</x>
                <y>0</y>
                <horiz>0</horiz>
                <vert>0</vert>
        </position>
        <checknum>205412124243</checknum>
        <name>95082_Home_Container_WidgetDrop</name>
        <father>95082_Home_Container</father>
        <widgetname>WidgetDrop</widgetname>
        <params>{"callback":"","cOrder":0,"signal":"","behavio
        <cent>0</cent>
</action><action>
```

Communication in Oneye

```
function HelloWorld_on_HelloWorld_Button($params = ’’) {
// API call
$response = file_get_contents(’https://api.binance.com/api/v3/time’);
// Update the label of the app.
$GLOBALS[’HelloWorld_Label’]->setText($response );

}
```



```
<?xml version="1.0" encoding="UTF-8"?>
<eyeMessage><action>
        <task>rawjs</task>

<js>document.getElementById("10666_HelloWorld_Label").innerHTML=""<
/js>
</action><action>
        <task>rawjs</task>

<js>document.getElementById("10666_HelloWorld_Label").appendChild(d
ocument.createTextNode( "{\"serverTime\":1611477393343}" ));</js>
</action></eyeMessage>
```

Data Storage

Virtual file system (VFS)

- Services for accessing files and directories
- PHP functions like **fopen()** and **unlink()** are not used
- VFS provides file access based on user type
  (VFS_INSUFFICIENT_PERMISSIONS)
- Ensures security to files

VFS - Background

- Virtual module uses two files when user creates a file
- Example,

  ```
  service('vfs','create',array('file.txt'));
  ```

- Two files namely, **file.txt_[RANDOM_STRING].eyeFile, and file.txt_[RANDOM_STRING].eyeInfo**
- Deletion operation - deletes both the files
- Two types of files - User's files(virtual) , System files(real)
- Every user has a home directory to work in
  - **conf**/ - the configuration files are stored here as real files with XML format.
  - **files**/ - the virtual files are stored here, and are visible to the user from the File Manager

Users in Oneye

- Multiple users support
- User Manager(UM) provides necessary methods

  `$currentUser; $path=service('um','getCurrentUserDir'));`
- Groups in Oneye
- **groups**/ - shared files between the group members
- For group managing - VFS and UM services

  `$um_getCurrentGroups; $vfs_real_getDirContent_group;`
- Disadvantage of having group in Oneye

User Session in Oneye

- Sessions are stored in an array structure
- **eyeSession** library has abstract method for accessing session
- Creates new session when it's not stored in array

```
if(eyeSessions('checkVar',array(TABLENAME))==false){
eyeSessions('makeArrayVar',array(TABLENAME));
}
```

Security issues and Challenges

- Client-Server communication is out of scope.
- User passwords can be easily decoded.
- Cannot provide anonymity to users as it's still in development stage
- Oneye browser cannot render most of the websites
- Restriction of files usage in Oneye group is limited
- No shared applications which increases productivity.

Oneye Improvement solutions

- Authentication of users based on hashing
- Providing anonymity to users outside DaaS
- Browser should support X-Frame-Options HTTP header
- Improving users permission for Oneye group
- Providing office function with more functionalities (integrating with open-source Groupware)
- Implementing session timeout and session lock

Application - Folder Structure

- opt/lampp/htdocs/
- oneye/system6f547cf659/apps/<application_name>
- Files - app.eyecode, events.eyecode, info.xml

Application - Initializing and Ending (app.eyecode)

- To run the application.
- Initialization and to end an application.
- Two functions: **ApplicationName_run** and **ApplicationName_end**
- Graphical application uses it to initialize the User Interface.
- Non-graphical and small applications concentrate all their code in this function.

Application - Events (events.eyecode)

- Applications are intractable.
- Information is shared with the server in the form of the message.
- Has various functions depending on the applications.

Demonstration

- PHP Version
- Sticky Notes

Thank You

Any Questions?