Reporting with MunkiReport

John Eberle (@tuxudo) and Rick Heil (@refreshingapathy)

2017 MACADMINS CONFERENCE

John: Mac Admin @ University of Pittsburgh

@tuxudo

@tuxudo

github.com/tuxudo

Rick: Senior IT Manager @ Myelin

@refrshingapathy

@refreshingapathy

github.com/rickheil

What is MunkiReport?

Myelin Mac Reporting | Dashboard | Reports | Listings | Admin | rickheil

Free Disk Space

2	14	187
< 10GB	< 25GB	25GB +

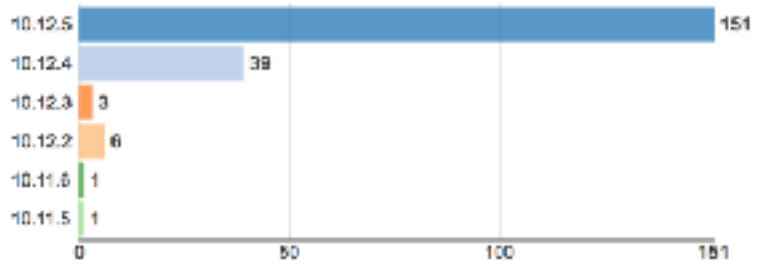
Uptime

53	97	51
7 Days +	< 7 Days	< 1 Day

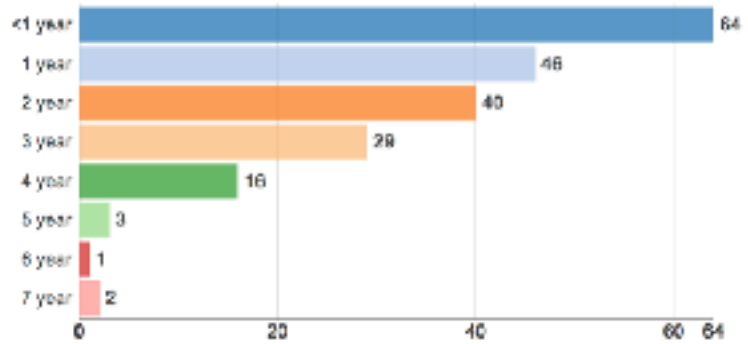
Munki

4	3
Errors	Warnings

OS Breakdown



Hardware Age



Events

[redacted]	3 minutes ago
[redacted] Free Disk Space < 25GB	11 minutes ago
[redacted] crashplan crashplan.backup_failed	17 minutes ago
[redacted] crashplan crashplan.backup_failed	22 minutes ago
[redacted] Free Disk Space < 10GB	32 minutes ago

Pending Installs

Dropbox 29.3.19	20
Google Chrome 59.0.3071.115	19
Cyberduck 6.1.0	3
[redacted]	2
Spotify 1.0.57.474.gce9c9538	2


Verizon LTE
14:12
80%
[redacted]
Myelin Mac Reporting
Free Disk Space
2
< 10GB
14
< 25GB
187
25GB +
Uptime
53
7 Days +
97
< 7 Days
51
< 1 Day
Munki
4
Errors
3
Warnings
OS Breakdown
10.12.5
151


You are without doubt the worst programming language i have ever heard of.
php
But you HAVE heard of me!

Originally designed as a reporting system for Munki

Boot ROM Version
Homebrew
MalwareBytes
Profiles
SSH Logins
Pending Installs
Temperatures
Local Admins
Caching Server
Thermal Load
Installed Fonts
User Sessions
Highest Supported OS
Server Metrics
FileVault Keys
Battery Health
SCCM Checkins
Application Usage
GPU Models
Time Machine Backups
Certificates
Disk Usage
Software Inventory
ARD Text
Location
SMART Information
Software Updates
WiFi Networks
GSX Lookups
SIP Status
Printers
USB Devices
AD/OD Binding
DeployStudio Image Data
Hardware Age
Xprotect Updates
Display Serial Numbers
Fan Speeds
CrashPlan Status
Network Share Path
Sleep Assertions
Extension Codesigning
Munki Keys

Username	Device Type	Battery Percent
[redacted]	Apple Wireless Mouse	7%
[redacted]	Apple Magic Mouse	50%
[redacted]	Magic Trackpad 2	32%
[redacted]	Apple Magic Mouse	35%
[redacted]	Magic Keyboard	94%

Similar to

Created and maintained by Arjen van Bochoven

Open Source

- Over 2,750 commits
- 36 Contributors
- More than 33,000 lines of code
- Always looking for new ideas
- Pull requests are always welcome! :D

Community Interaction

- Github issues - for help and feature requests
- Visit #munkireport channel on MacAdmins Slack
- Post to munkireport Google Group
- Send @bochoven baked goods

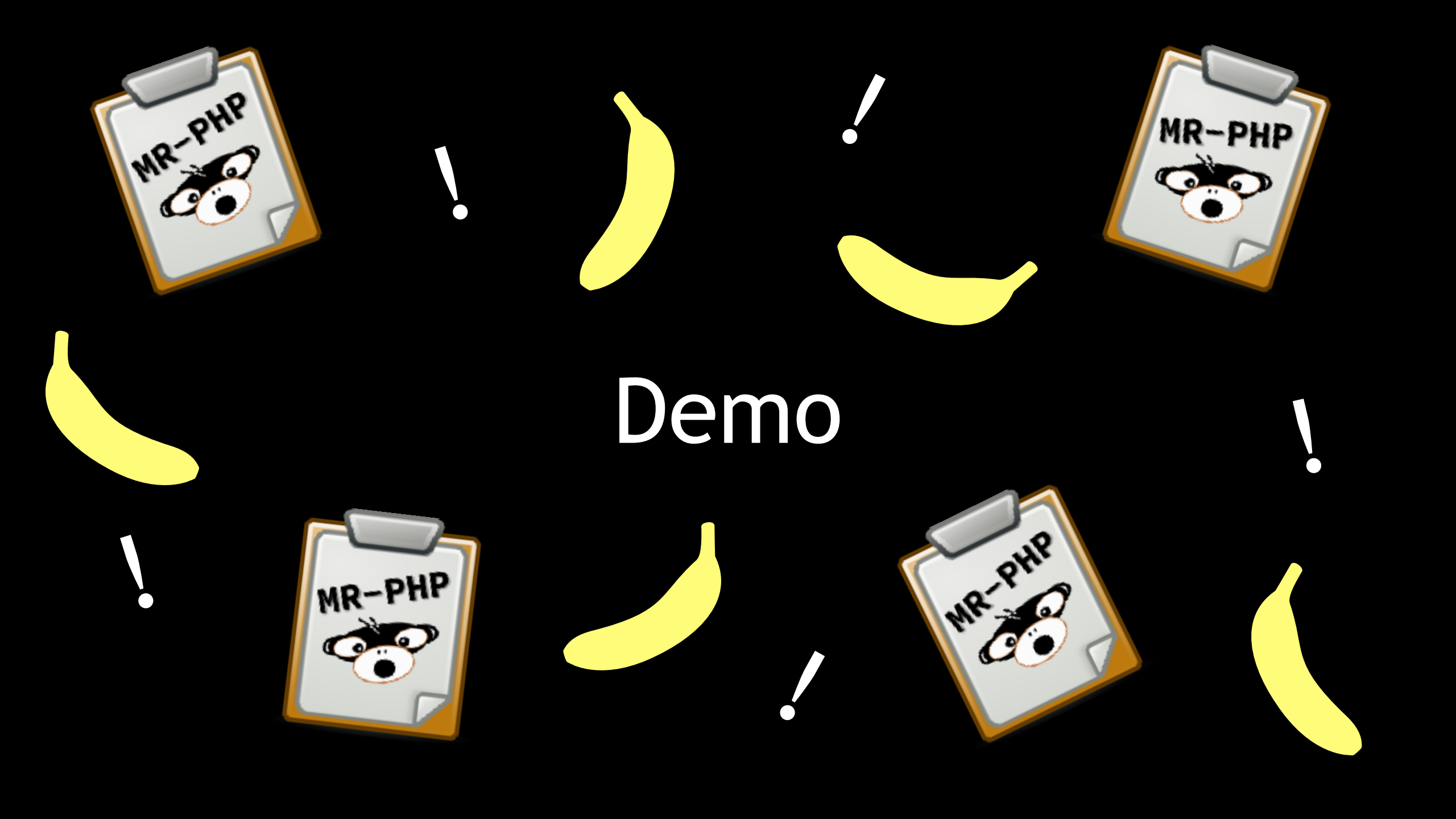

MR-PHP
MR-PHP
MR-PHP
MR-PHP
What does it do?

Main features

- Gather data from clients
- Store data in database
- Provide export of data to CSV
- Display data in the web interface

MunkiReport UI

- Dashboard
  - Main landing page and full of widgets
- Reports
  - Groupings of similar widgets
- Listings
  - Sortable, filterable tables containing data about machines
- Client tabs
  - Sections containing data for a particular machine



Demo

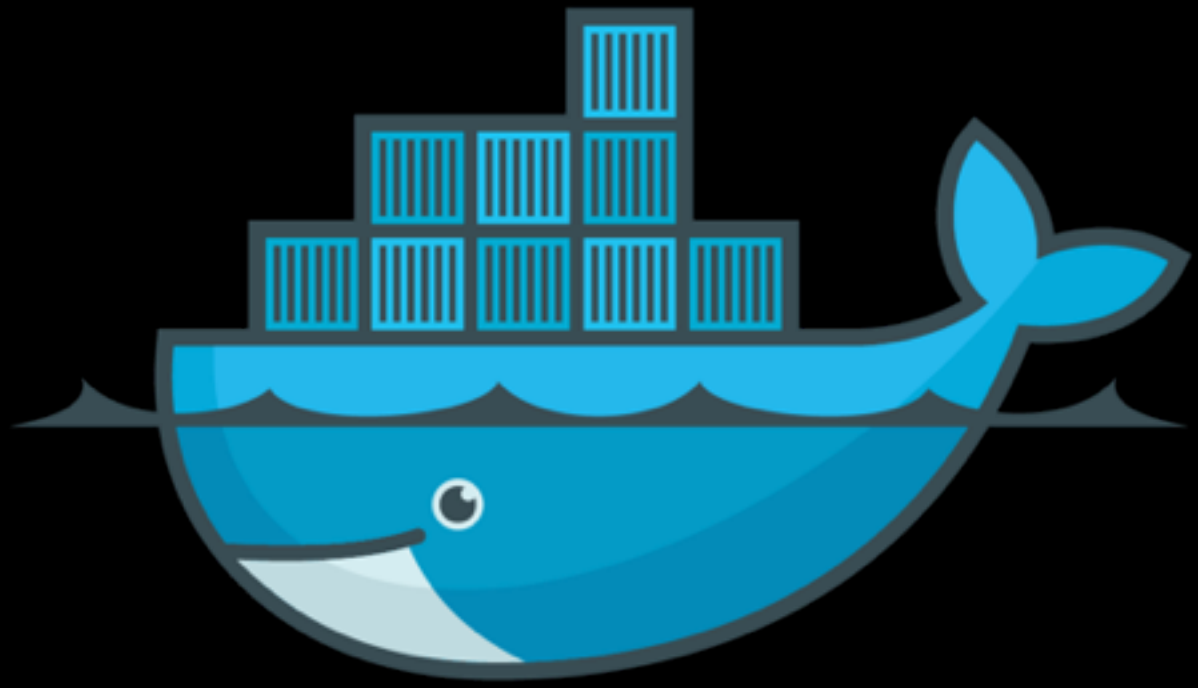
How can this be used in the real world?

- Assist in remote troubleshooting
- Software inventory
- Asset management
- Security / compliance checking

Server Installation

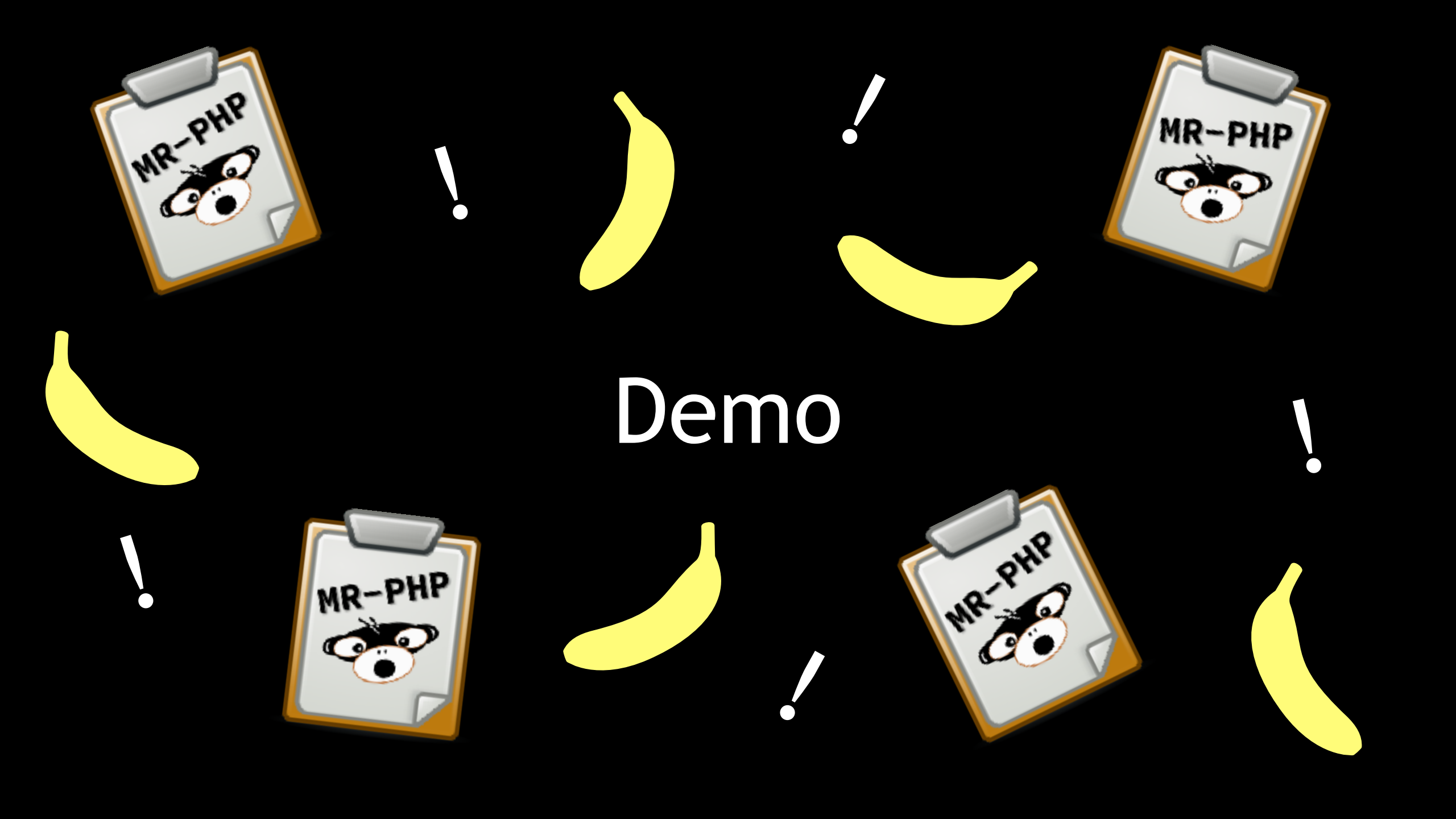
- Requirements
  - A web server
  - PHP 5.4 or higher with pdo-sqlite2 and libxml
- Download latest release from GitHub, extract, configure, and upload to webroot

Docker container is available!



Client Installation

- Curl install script into bash
- Make a client package and install via Munki
- Setup AutoPKG(r) to make client package and import into Munki



Demo

Instance Security

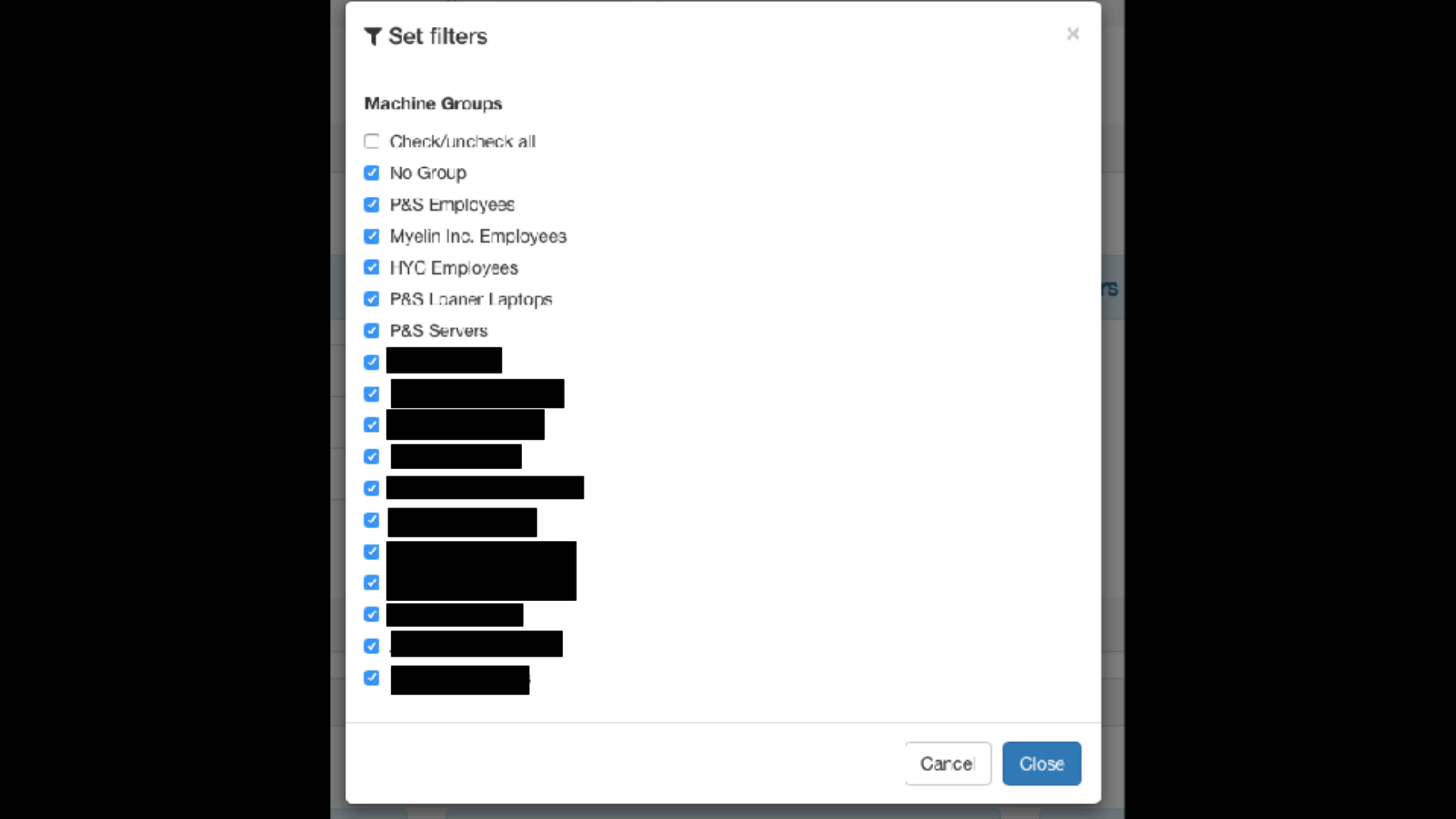
- Install with client package method
- If using SQLite, protect the file
- Should use HTTPS
- Set passphrase to limit clients
- Enable reCaptcha
- Use authentication

Integrations

- Supports many authentication methods and roles
  - No authentication, Local/Hashed, Active Directory, LDAP
- Database types
  - Local sqlite or MySQL
- External integrations
  - Google Maps, GSX, DeployStudio

Expandability

- JSON API
- Localizable
- Machine groups
- Business units


Set filters
Machine Groups
Check/uncheck all
No Group
P&S Employees
Myelin Inc. Employees
HYC Employees
P&S Loaner Laptops
P&S Servers
[redacted]
Cancel
Close

Business Units +

Partners & Simons

23 Drydock Ave, Suite 810W Boston MA 02210
More information

Machine Groups	Managers	Users
No Group 6	[redacted]	
P&S Employees 96	[redacted]	
P&S Loaner Laptops 4		
P&S Servers 7		

delete

Myelin Communications

23 Drydock Ave, Suite 810W Boston MA 02210
More information

Machine Groups	Managers	Users
Myelin Inc. Employees 1	[redacted]	
	[redacted]	
	[redacted]	

delete

HYConnect

142 East Ontario Street Suite 13 Chicago, IL 60611
More information

Machine Groups	Managers	Users
HYC Employees 22	[redacted]	
	[redacted]	
	[redacted]	

delete


Global Service Exchange
Wi Fi
Certificate
Modules

What's a module?

- Bundles of PHP and scripts
- Installed on clients or server side only
- Modular in installation and extendability
- Easy to make and update

Core Parts of a Module

- Model
- Controller
- Provides
- Locales

Model

- Model is the powerhouse of the module
- Creates database table
- Processes incoming client data
- Recalls and processes data for widgets

Model

```php
<?php
class Demo_model extends Model
{
    public function __construct($serial = '')
    {
        parent::__construct('id', 'demo'); //primary key, tablename
        $this->rs['id'] = 0;
        $this->rs['serial_number'] = $serial;$this->rt['serial_number'] = 'VARCHAR(255) UNIQUE';
        $this->rs['Text1'] = '';
        $this->rs['Text2'] = '';

        // Schema version, increment when creating a db migration
        $this->schema_version = 0;

        // Add indexes
        $this->idx[] = array('Text1');
        $this->idx[] = array('Text2');

        // Create table if it does not exist
        $this->create_table();

        if ($serial) {
            $this->retrieve_record($serial);
        }

        $this->serial = $serial;
    }

    public function process($data)
    {
        require_once(APP_PATH . 'lib/CFPropertyList/CFPropertyList.php');
        $parser = new CFPropertyList();
        $parser->parse($data);

        $plist = $parser->toArray();

        foreach (array('Text1', 'Text2') as $item) {
            if (isset($plist[$item])) {
                $this->$item = $plist[$item];
            } else {
                $this->$item = '';
            }
        }

        $this->save();
    }
}
```

Controller

- Main purpose is to control access to model’s functions and data stored within the module’s table
- Responsible for the module’s API hooks
- Passes data from model’s widget functions to widgets using JSON API calls

Controller

```php
<?php

class Demo_controller extends Module_controller
{
    public function __construct()
    {
        $this->module_path = dirname(__FILE__);
    }

    public function index()
    {
        echo "You've loaded the demo module!";
    }

    public function get_data($serial_number = '')
    {
        $obj = new View();

        if (! $this->authorized()) {
            $obj->view('json', array('msg' => 'Not authorized'));
            return;
        }

        $demo = new Demo_model($serial_number);
        $obj->view('json', array('msg' => $demo->rs));
    }
}
```

Provides

- Instructs MunkiReport to load listings, widgets, and client tabs
- Required if the module has any UI elements

Provides

```php
<?php

return array(
    'client_tabs' => array(
        'demo-tab' => array('view' => 'demo_tab', 'i18n' => 'demo.client_tab_title'),
    ),
    'listings' => array(
        'demo' => array('view' => 'demo_listing', 'i18n' => 'demo.listing_title'),
    ),
    'widgets' => array(
        'demo' => array('view' => 'demo_widget'),
    ),
    'reports' => array(
        'demo' -> array('view' -> 'demo', 'i18n' -> 'demo.report_title'),
    ),
);
```

Locales

- JSON files containing translations for UI elements
- Required if making a listing, widget, or client tab
- Only translate into your native language(s), contributors will translate your locale file

Locales

```
{
    "client_tab_title": "Demo Client Tab",
    "demo_widget": "Demo Widget",
    "listing_title": "Demo",
    "listing": {
        "Text1": "Text 1",
        "Text2": "Text 2"
    },
    "report_title": "Demo Report",
    "widget_tooltip": "The demo widget"
}
```

UI Parts of a Module

- Listing
- Client tab
- Widget

Listing

- Comprised of a static HTML table that is generated and filled by JavaScript
- Table contents pulled from controller via internal JSON call
- Not every module needs a listing
- Can have more than one per module

Listing

```
<?php $this->view('partials/head'); ?>

<?php
new Machine_model;
new Reportdata_model;
new Demo_model;
?>

<div class="container">
  <div class="row">
    <div class="col-lg-12">
          <h3><span data-i18n="demo.report_title"></span> <span id="total-count" class='label label-primary'>…</span></h3>
          <table class="table table-striped table-condensed table-bordered">
            <thead>
              <tr>
                <th data-i18n="listing.computername" data-colname='machine.computer_name'></th>
                <th data-i18n="serial" data-colname='reportdata.serial_number'></th>
                <th data-i18n="username" data-colname='reportdata.long_username'></th>
                <th data-i18n="ard.listing.text" data-i18n-options='demo.Text2' data-colname='demo.Text1'></th>
                <th data-i18n="ard.listing.text" data-i18n-options='demo.Text2' data-colname='demo.Text2'></th>
              </tr>
            </thead>
            <tbody>
                <tr>
                    <td data-i18n="listing.loading" colspan="5" class="dataTables_empty"></td>
                </tr>
            </tbody>
          </table>
    </div>
  </div>
</div>
```

Client Tab

- Shows data for a single machine in the client overview page
- Pull data from the model through the controller via API call
- Can be static HTML table filled with JavaScript or dynamically generated JavaScript tables
- Not every module needs a client tab
- Can have more than one per module

Client Tab

```
<div id="demo-tab"></div>
<h2 data-i18n="demo.client_tab_title"></h2>

<script>
$(document).on('appReady', function(){
    $.getJSON(appUrl + '/module/demo/get_client_tab_data/' + serialNumber, function(data){
        var skipThese = ['groupdate'];
        $.each(data, function(i,d){

            // Generate rows from data
            var rows = ''
            for (var prop in d){
                // Skip skipThese
                if(skipThese.indexOf(prop) == -1){
                    if(prop.indexOf('bytes') > -1){
                        rows = rows + '<tr><th>'+i18n.t('demo.'+prop)+'</th><td>'+fileSize(d[prop], 2)+
                    } else {
                        rows = rows + '<tr><th>'+i18n.t('demo.'+prop)+'</th><td>'+d[prop]+'</td></tr>';
                    }
                }
            }
            $('#demo-tab')
                .append($('<h4>')
                    .append($('<i>')
                        .addClass('fa fa-database'))
                    .append(' '+d.groupdate))
                .append($('<div style="max-width:370px;">')
                    .addClass('table-responsive')
                    .append($('<table>')
                        .addClass('table table-striped table-condensed')
                        .append($('<tbody>')
                            .append(rows))))
        })
    });
});
</script>
```

Widget

- Are displayed on the dashboard
- Processed data is pulled from model through controller via API call
- Mostly written in HTML and JavaScript
- Modules can have multiple or no widgets
- Do not have to pull data from MunkiReport; ie can show weather information or pull from external data source

Widget

```
<div class="col-lg-4 col-md-6">
    <div class="panel panel-default" id="caching-widget">
        <div class="panel-heading" data-container="body" data-i18n="[title]demo.widget_tooltip">
            <h3 class="panel-title"><i class="fa fa-database"></i>
                <span data-i18n="demo.widget_title"></span>
                <list-link data-url="/show/listing/demo/demo"></list-link>
            </h3>
        </div>
        <div class="panel-body text-center"></div>
    </div><!-- /panel -->
</div><!-- /col -->

<script>
$(document).on('appUpdate', function(e, lang) {

    $.getJSON( appUrl + '/module/demo/demo_widget', function( data ) {

        if(data.error){
            return;
        }

        var panel = $('#demo-widget div.panel-body'),
        baseUrl = appUrl + '/show/listing/demo/demo';
        panel.empty();

        // Set statuses
        if(data.redblock != "0"){
            panel.append(' <a href="'+baseUrl+'" class="btn btn-danger"><span class="bigger-150">'+
            <br>'+i18n.t('demo.redblock')+'</a>');
        }
        if(data.yellowblock){
            panel.append(' <a href="'+baseUrl+'" class="btn btn-warning"><span class="bigger-150">'
            <br>'+i18n.t('demo.yellowblock')+'</a>');
        }
        if(data.greenblock){
            panel.append(' <a href="'+baseUrl+'" class="btn btn-success"><span class="bigger-150">'
            <br>'+i18n.t('demo.greenblock')+'</a>');
        }

    });
});
</script>
```

Client Parts of a Module

- Install script
- Uninstall script
- Main data gathering script
- Cache file

Install Script

- Run by MunkiReport installer on client
- Downloads main script and sets its permissions
- Activates module on the client by setting the cache file in the MunkiReport.plist
- Required in all but advanced cases

Install Script

```
#!/bin/bash

MODULE_NAME="demo"
MODULESCRIPT="demo"
MODULE_CACHE_FILE="demo.plist"

CTL="${BASEURL}index.php?/module/${MODULE_NAME}/"

# Get the scripts in the proper directories
"${CURL[@]}" "${CTL}get_script/${MODULESCRIPT}" -o "${MUNKIPATH}preflight.d/${MODULESCRIPT}"

# Check exit status of curl
if [ $? = 0 ]; then
    # Make executable
    chmod a+x "${MUNKIPATH}preflight.d/${MODULESCRIPT}"
    touch "${MUNKIPATH}preflight.d/cache/${MODULE_CACHE_FILE}"

    # Set preference to include this file in the preflight check
    setreportpref $MODULE_NAME "${CACHEPATH}${MODULE_CACHE_FILE}"

else
    echo "Failed to download all required components!"
    rm -f "${MUNKIPATH}preflight.d/${MODULESCRIPT}"

    # Signal that we had an error
    ERR=1
fi
```

Uninstall Script

- Runs on client when the module is disabled and the MunkiReport installer is run
- Deletes the data gathering script and the cache file
- Required in all but advanced cases

Uninstall Script

```
1   #!/bin/bash
2
3   rm -f "${MUNKIPATH}preflight.d/demo.py"
4   rm -f "${CACHEPATH}demo.plist"
```

Main Data Gathering Script

- Is executed when preflight is run, either manually or by Munki
- Runs as root
- Can be any script or binary that macOS supports
- Must complete in less than 10 seconds or MunkiReport will kill it
- Required in all but advanced cases

Main Data Gathering Script

```
#!/usr/bin/python

import subprocess
import os
import plistlib
import sys

def main():
    """Main"""
    # Create cache dir if it does not exist
    cachedir = '%s/cache' % os.path.dirname(os.path.realpath(__file__))
    if not os.path.exists(cachedir):
        os.makedirs(cachedir)

    # Skip manual check
    if len(sys.argv) > 1:
        if sys.argv[1] == 'manualcheck':
            print 'Manual check: skipping'
            exit(0)

    # Get results
    result = dict()
    info = get_demo_info()
    result = flatten_demo_info(info)

    # Write results to cache
    output_plist = os.path.join(cachedir, 'demo.plist')
    plistlib.writePlist(result, output_plist)

if __name__ == "__main__":
    main()
```

Cache File

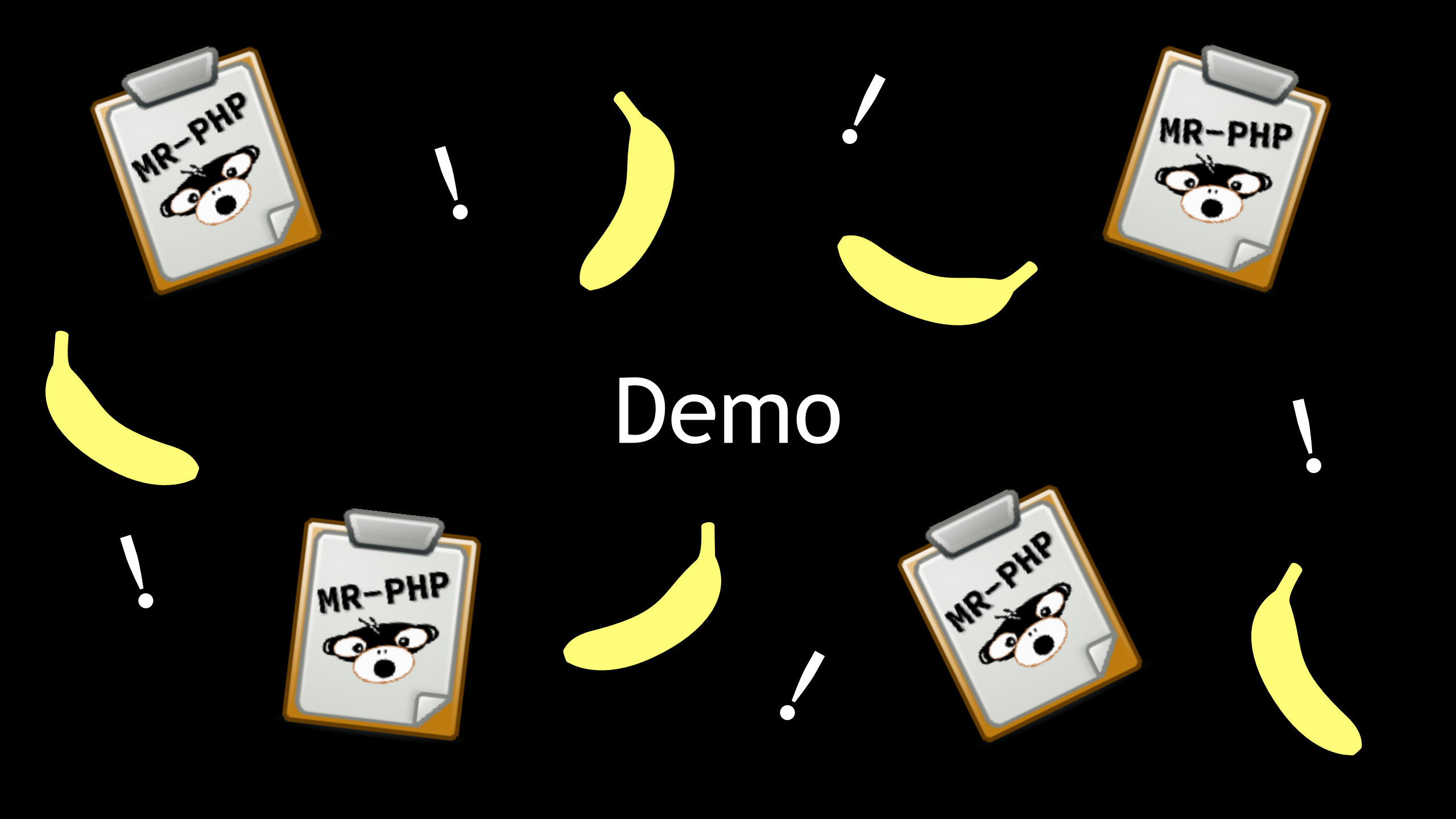
- File that is uploaded to MunkiReport server for processing by model
- Only one per module
- Does not have to be a file generated by the main script; ie can be any file on the client
- Should be kept small to limit network traffic and timeouts

Cache File

```
1   <?xml version="1.0" encoding="UTF-8"?>
2   <!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN"
3   <plist version="1.0">
4   <dict>
5       <key>Text1</key>
6       <string>This is text 1</string>
7       <key>Text2</key>
8       <string>This is text 2</string>
9   </dict>
10  </plist>
```

Mega Module Pack

Machine Orientation
App Store Apps
Thunderbolt Devices
Audio Amplification
User Feedback
FileMaker Assets
System Voltages
Launch Daemons
FireWire
Liquid Sensors
Bees?
Module App Store
Skype for Business
LOM Status
Mouse Tracking
Eye of Sauron
Xsan
Slack Integration
Email Notifications
User Lunch
Jamf Pro
Radio Interference
User Backgrounds
TouchBar
iOS Devices
Launch Agents
Local Weather
Trackpad Touchpoints



Demo

Questions?

Feedback URL:
https://bit.ly/psumac2017-160