

Centralized API Document

Table of Content

- [Centralized API Feature Categorization](#)
 - Login
 - System Tools
 - Client Services
 - Client Devices
 - Client IP Addresses
 - Client Support
 - Client Reports
 - Client Tools
 - Client Billing
 - Client Setting
 - Centralized API Admin
- [Permission/Roles](#)
- [Client Area](#)
 - [Caching/performance](#)
- [Service Cancellation Automation](#)
- [Service Order Automation](#)
 - Centralized API Admin
 - [Manage Client](#)
 - [Manage Services](#)
 - [Automation Flow](#)
 - [Selecting Device Logic](#)
 - [OS Install Automation](#)

*content in RED are api calls and variables

The features below need to be part of the centralized API where all request must be made.

For example -

- client area -----> centralized API -----> Ubersmith
- client area ----> centralized API ----> edge layer router
- client area ----> centralized API ----> IPMI authoring
- client area ---> centralized API ----> submit ticket to ubersmith
- API for clients ----> centralized API ---> Ubersmith
- WHMCS reseller module ---> centralized API ---> ubersmith
- dashboard ---> centralized API ---> check ubersmith tickets

You can read more about centralized api at [Centralized API](#)

Centralized API Features Categorization

1. Login (OpenID Implementation)
 - 1.1. Register
 - 1.2. Login
 - 1.3. Password Reset
 - 1.4. Login As Client
2. System Tools (GUI NOT REQUIRED)
 - 2.1. IP Tools
 - 2.1.1. Add IP
 - 2.1.2. Remove IP
 - 2.2. Router Level Null
 - 2.2.1. IP Null Status

- 2.2.1.1. Single IP
 - 2.2.1.2. Multiple IPs
 - 2.2.2. Null Single IP
 - 2.2.3. Null Multiple IPs
 - 2.2.4. Unnull Single IP
 - 2.3. NTA Null
 - 2.3.1. Add Null
 - 2.3.2. Remove Null
 - 2.4. cPanel License
 - 2.4.1. Add License
 - 2.4.2. Cancel License
 - 2.5. Shipping
 - 2.5.1. Create shipment
 - 2.5.2. View Shipment
 - 2.5.3. Edit Shipment
 - 2.5.4. Delete Shipment
 - 2.6. Abuse Automation
 - 2.6.1. Check Abuse Queue
 - 2.6.2. Notify Client
 - 2.6.2.1. Instant Null based on specific rules/tags
 - 2.6.3. Notify Original Abuse Submitter
 - 2.6.4. Null Abused IP's If No Reply
 - 2.7. VLAN
 - 2.7.1. VLAN Database
 - 2.7.2. Duplicate Devices
 - 2.7.3. Duplicate VLANs
3. Client Services (GUI NOT REQUIRED)
- 3.1. List Services
 - 3.2. View Service Details
 - 3.3. Service Automation ← HIGH PRIORITY
 - 3.3.1. [Automated Service Provisioning](#)
 - 3.3.2. Manual Service Provisioning
 - 3.4. Service Cancellation ← HIGH PRIORITY
 - 3.4.1. [Automated Service Cancellation](#)
 - 3.4.2. Manual Service Cancellation
 - 3.5. Reactivate Service
4. Client Devices (GUI NOT REQUIRED)
- 4.1. List Devices
 - 4.2. View Device Details
 - 4.2.1. Start IPMI
 - 4.2.2. Stop IPMI
5. Client IP Addresses (GUI NOT REQUIRED)
- 5.1. ~~View default IP Group~~

- 6. Client Support (GUI NOT REQUIRED)
 - 6.1. List Support Ticket
 - 6.2. View Support Ticket ID
 - 6.3. Update Support Ticket ID
 - 6.4. Open New Support Ticket
 - 6.5. Search Support Ticket
- 7. Client Reports (GUI NOT REQUIRED)
 - 7.1. Traffic Diversion Report
 - 7.1.1. List Variables
 - 7.1.2. Get Report
 - 7.2. DDoS Null/Unnull Report
 - 7.3. Attack Summary Report
 - 7.3.1. Plotting of charts
 - 7.3.2. Live View
- 8. Client Tools (GUI NOT REQUIRED)
 - 8.1. Manual IP Diversion
 - 8.1.1. Add IP Diversion
 - 8.1.2. Delete Diversion
 - 8.1.3. Enable/Disable Diversion
 - 8.2. Firewall
 - 8.2.1. Edge Layer ACL
 - 8.2.1.1. Add Edge Layer ACL Firewall Rule
 - 8.2.1.2. Delete Edge Layer ACL Firewall Rule
 - 8.2.1.3. Restore Edge layer ACL Rules
- 9. Client Billing (GUI NOT REQUIRED)
 - 9.1. My Invoices
 - 9.1.1. List Invoices
 - 9.1.2. View Invoices
 - 9.1.3. Pay Invoices
 - 9.2. Account Credits
 - 9.2.1. List credits
 - 9.2.2. view credit details
 - 9.3. Orders
 - 9.3.1. List Orders
 - 9.3.2. View order details
- 10. Client Settings (GUI NOT REQUIRED)
 - 10.1. My Details
 - 10.1.1. View Details
 - 10.1.2. Update Details
 - 10.2. My Contacts
 - 10.2.1. List Contacts
 - 10.2.2. View Contact Details
 - 10.2.3. Edit Contact Details

- 10.2.3.1. Edit Permissions - we need the sub contact to have permission for all sections including tools and reports sections.
 - 10.2.4. Add Contacts
 - 10.2.5. Delete Contact
- 10.3. Notification
 - 10.3.1. Diversion Notification
 - 10.3.1.1. Enable
 - 10.3.1.2. Disable
 - 10.3.1.3. Pause
 - 10.3.2. Null Route Notification
 - 10.3.2.1. Enable
 - 10.3.2.2. Disable
 - 10.3.2.3. Pause
 - 10.3.3. Login Notification - Notify main email when a new IP logs into client area
 - 10.3.3.1. Enable
 - 10.3.3.2. Disable
- 10.4. Payments
 - 10.4.1. Add credit card
 - 10.4.2. List credit card
- 10.5. My Inbox
 - 10.5.1. List Emails
 - 10.5.2. View Emails
- 10.6. Password
 - 10.6.1. Update password
- 11. Centralized API Admin (GUI REQUIRED)
 - 11.1. Manage Services
 - 11.1.1. List Services
 - 11.1.2. Edit Services
 - 11.2. Manage Racks
 - 11.2.1. List Racks
 - 11.2.2. Edit Racks
 - 11.3. Managed Devices
 - 11.4. Manage Clients
 - 11.5. Manage API Access
 - 11.6. [Settings](#)

Permission/Roles

All services in centralized API should have roles and permissions assigned to them. Some services are for internal use only and shouldn't have access from outside use. Some services will be accessed by client area only and will not have access by reseller API. Therefore, it is crucial the roles/permissions are set for each action in the centralized API.

Example:

Role: Admin

Action: Service Cancellation

Authorization: Allow

Role: Client

Action: Service Cancellation

Authorization: Disallow

Role: Support Admin

Action: Null IP

Authorization: Allow

Role: Reseller

Action: Null IP

Authorization: Disallow

Role: Reseller

Action: Order Service

Authorization: Allow

Role: Reseller

Action: Order Service ID XXX

Authorization: Disallow

Client Area

Our client area currently resides in <https://www.psychz.net/client> and we need to redo the code to allow account/sub account permissions for each individual section. We also need this done so we can manage API access control for accounts and sub accounts.

Client Area -----> Centralized API -----> Ubersmith

New client area will reside in <https://www.psychz.net/dashboard>

All features in client area will communicate to centralized API and centralized API will communicate with Ubersmith via API. All features such as devices, services, reports, invoices, support tickets will have permissions associated with them and centralized API will manage those permissions by using [client management](#) tool in centralized API.

All requests from the client area will be submitted to centralized API and utilizing queue, we will manage the tasks and execute them accordingly.

For example -

- A client will submit a request for edge layer acl in client area. First the script will verify if the client has permission to edge layer acl. If permission is there, it will accept the request and request will be submitted to centralized API in the queue. The task will execute once the queue runs and execute the code.
- Main account holder created a sub account and disabled permission to view support tickets. This permission will be updated on the client management. Sub account holder logs into <https://www.psychz.net/client> and submits a request to view support tickets to centralized API. Centralized API will check permission and give error since sub account has no permission to view support tickets.

Structure

- Client area will be built using Yii framework just like the centralized API to speed up the coding process.
- All current functions and structure will be kept same.
- Script that will communicate with ubersmith will reside in api.psychz.net and we will use API to communicate with ubersmith

- We will create API for each ubersmith function in api.psychz.net, so we can communicate from our client area.

Client Area Features (api calls and parameter are in red) -
Please make sure the uber calls are done from api.psychz.net →
portal.psychz.net. All communication from www.psychz.net
should be done to api.psychz.net only to ensure that every call
has permission to make that call.

Following urls need to be copied to <https://www.psychz.net/dashboard>

<https://www.psychz.net/client/>
<https://www.psychz.net/client/login>
<https://www.psychz.net/client/forgotpassword>
<https://www.psychz.net/client/register/>
<https://www.psychz.net/client/users/dashboard>
<https://www.psychz.net/client/users/services>
https://www.psychz.net/client/users/devices?mode=devices_list
<https://www.psychz.net/client/users/supporttickets>
<https://www.psychz.net/client/users/submitticket>
<https://www.psychz.net/client/users/submitticket?step=2&deptid=20>
<https://www.psychz.net/client/users/submitticket?step=2&deptid=1>
<https://www.psychz.net/client/users/submitticket?step=2&deptid=2>
<https://www.psychz.net/client/users/submitticket?step=2&deptid=3>
<https://www.psychz.net/client/users/billing?mode=invoices>
<https://www.psychz.net/client/users/review?mode=invoicepay&invid=xxxxxxxxxxxx>
<https://www.psychz.net/client/users/billing?mode=orders>
<https://www.psychz.net/client/users/billing?mode=credits>
https://www.psychz.net/client/users/account?mode=payment_method_list
<https://www.psychz.net/client/users/ddosreports>
<https://www.psychz.net/client/users/nullrouterreports>
<https://www.psychz.net/client/users/attackreports>
<https://www.psychz.net/client/users/manualipdiversion>
<https://www.psychz.net/client/users/shipments>
<https://www.psychz.net/client/users/edgelayeracl>
<https://www.psychz.net/client/users/account?mode=details>
<https://www.psychz.net/client/users/account?mode=contacts>
<https://www.psychz.net/client/users/account?mode=notification>
<https://www.psychz.net/client/users/account?mode=emailslis>
<https://www.psychz.net/client/users/account?mode=changepw>

1. Design

- a. Dashboard - all pages that require login will have a special theme that will have collapsible left column and smaller header with just main website links.
- b. Order Forms - order forms need to be part of main website theme.
- c. Knowledgebase - the url, content, structure needs to stay the same at <https://www.psychz.net/client/kb/en/>
- d. Blog - the url, content, structure needs to stay same at <https://www.psychz.net/client/blog/en/>
- e. Community - the url, content, structure needs to stay same at <https://www.psychz.net/client/questions>

2. Caching/Performance

- a. ~~We need to utilize cache and to avoid delay from dashboard → api.psychz.net → portal.psychz.net, we will need to cache content when the client is logging into the dashboard.~~
- b. ~~When the client is on client/web/site/login and clicks on "Login" button, after successful login api.psychz.net will pull all latest client information such as service, devices, invoices, support tickets and build a latest cache in api.psychz.net.~~
 - i. ~~This method will avoid running cache for all clients and also caching old data.~~

3. Login page (**uber.check_login**)

- a. Login Name (**login**)
- b. Password (**pass**)
- c. password reset (**uber.forgot_pass**)
 - i. Username or Client ID (**user_login**)
 - ii. Verification code
- d. remember me
- e. Notification
 - i. Client notification when a new IP logs into client area

4. Register page - (**client.add**)

- a. First name (**first**)
- b. Last name (**last**)
- c. Company name (**company**)
- d. Email address (**email**)
- e. password (**uber_pass**)
- f. confirm password (**uber_pass**)
- g. Address (**address**)

- h. City (**city**)
 - i. State/Region (**state**)
 - j. Zip code (**zip**)
 - k. Country (**country**)
 - l. Phone Number (**phone**)
 - m. Fax (**fax**)
 - n. How did you hear about us (**will require storage in api.psychz.net and will be added to client profile page**)
 - o. Terms of service agreement
 - p. Acceptable usage policy agreement
 - q. Privacy agreement
5. Order Forms
- a. Only services with “main website” or “special page” enabled can be ordered - <http://api.psychz.net/backend/web/manage/service>
 - b. Shopping cart needs to be removed. If client doesn't complete the order, the information will be lost once the client leaves the page.
 - i. Client can edit configuration from the checkout page and make changes and return to check out page.
 - c. Make sure the order forms have responsive design - <https://i.gyazo.com/dde2ee1e0074a84a49e6f07fafd163a8.png>
 - d. Order forms will follow the current design and structure of pulling plan ID information - <https://i.gyazo.com/bcb7ffa40bbe92ece39ce2c6201ec762.png>
 - i. Please remove “Chosen Addons” and “Included Items”
 - ii. For the right hand column, we want to show “Order Details” and they should update as client selects different options.

6. Dashboard

a. Design

Psychz Logo | **Compatible left navigation**

Header: Dedicated, Colocation, Private cloud, DDoS Protection, IP Transit

Left Sidebar:

- Dashboard
- Services/Devices
 - Active Services
 - Active Devices
 - Order Service
- Support
 - List tickets
 - Open New Ticket
- Billing
 - Unpaid Invoices
 - List Invoices
 - Credit
 - List orders
 - Payment Method
- Reports
 - Traffic Diversion
 - DDoS notification
 - Live DDoS Report
- Tools
 - DDoS Diversion
 - Edge Layer ACL
 - Shipping
- My Account
 - My Details
 - My Contacts
 - Notification
 - My Inbox
 - Change Password
- Logout

Main Content Area:

Account Overview

Client ID: [Name: John Doe] [Update]

Services: 4 [Order New]

Due Invoices: 1

Open Tickets: 0 [Create]

* This section will only list content that falls under unpaid invoice, open ticket, new shipment, active devices

Due Invoices

Invoice	Date Sent	Date Due	Amount	Amount Outstanding	Status

Shipments

Shipment ID	Request Date	Type	Hardware Label	Position	Status	Actions

My Devices

Device ID	Description	Status	Service	Type	Location	Label

b.

- c. Download - <https://www.dropbox.com/s/4jnrplrdj4wen6j/CC103082016.jpg?dl=0>

7. Services/Devices

- a. Active services (**client.service_list**)
 - i. Parameters

1. client_id
 2. pack_type_select=4
 3. devices=1
- ii. Table Data
 1. Service ID
 2. Product Description
 3. Device
 4. Price
 5. Billing Cycle
 6. Next Due Date
 7. Status
- b. Active Devices (device.list)
 - i. Parameters
 1. client_id
 - ii. Table Data
 1. Device ID
 2. Description
 3. Status
 4. Service
 5. Type
 6. Location
 7. Label
 - c. Order New Service - This page will list all services that have “main website” enabled - <http://api.psychz.net/backend/web/manage/service>
 - i. First we need to work on new order forms. Our order forms can be hosted in <https://www.psychz.net/dashboard/order/id139?coupon=couponcode>
 1. Order forms will have a simple header and footer design. For example - <https://i.gyazo.com/7183bd1e7666b3a74b89cbd41953eae4.png>
 2. Order forms will be accessible for both logged in and non logged in clients
 - a. If logged in, they will see their name and drop down tool on top corner
 - b. If not logged in, it will just look like a regular page
 3. Order form will be a one page design and it will not use the same design as dashboard. It will have the following components
 - a. Header
 - i. Logo
 - ii. Phone

- iii. Client Info (Name and drop down)
 - b. Body
 - i. Service Name
 - ii. Fixed/Floating Menu (Configure, Account, Payments, Submit)
 - iii. Order Form + Summary that will scroll
 - iv. Account (Login or signup)
 - v. Payment Options
 - vi. Place Order
- 8. Support
 - a. List tickets (`support.ticket_list`)
 - i. Parameters
 - 1. `contact_id`
 - 2. `type=open` (default will be open but filter option should be there to allow on hold, closed, ClientAll (All))
 - b. Open New Ticket (`support.ticket_submit`)
 - c. Knowledgebase
 - d. Question/Answer
- 9. Billing
 - a. Unpaid Invoices (`client.invoice_list`) - please make sure we offer the option to client to enter amount for specific line item on the invoice. If you want to test, please login as 7155 at <https://portal.psychz.net> and try to pay the invoice. We want to give client the ability to pay specific entries in Ubersmith.
 - i. Pay via Paypal - <http://paypal.github.io/PayPal-PHP-SDK/> - charge, verify and record payment
 - 1. Record payment using `client.invoice_post_gw_payment`
 - ii. Pay via Alipay
 - iii. Pay via credit card (`client.invoice_charge`)
 - b. List Invoices (`client.invoice_list`)
 - c. Credit (`client.credit_list`)
 - d. List Orders (`order.list`)
 - e. Payment Method (`client.payment_method_list`)
- 10. Reports (`You will need to create the api call from api.psychz.net`)
 - a. Traffic Diversion
 - b. DDoS Null/Unnull
 - c. Live DDoS Report
- 11. Tools (`You will need to create the api call from api.psychz.net`)
 - a. DDoS Diversion
 - b. Edge Layer ACL
 - c. Shipping
- 12. My Account

- a. My Details
- b. My Contacts (`client.contact_list`)
 - i. Parameter
 - 1. `client_id`
 - 2. `contact_active=1`
 - ii. Edit Contacts
 - 1. Security
 - a. Diagram -
<https://i.gyazo.com/59beee204bcb4b214b17115b01b35d0e.png>
 - b. Enable Two Factor Authentication
 - i. The temporary code will be active for 5 mins only and it will be tracked by `api.psychz.net`.
 - ii. Phone - For the SMS, we will utilize <https://www.plivo.com/pricing/#!/sms> and use the API to send a text code to the mobile phone.
 - 1. Enable
 - a. Verify: First the client will be required to enter their phone number and the script will send a code to the phone number. The client will verify that code.
 - b. Once verified, the script will send a code to the phone each time someone tries to login to that account and the client will be required to enter that temp code on the login page.
 - 2. Disable
 - iii. Email
 - 1. Enable
 - a. Verify: Send verification email with temp code to the email address on file for contact to make sure email is being delivered to their inbox. Contact will copy the temp code and verify on the page.
 - b. Once verified, the login will require pin to access the dashboard. The system will send a temporary PIN to the email address and the client will need to obtain that pin and enter it in

the dashboard to go to the main dashboard.

2. Disable

- c. Add A new Contact (**client.contact_add**)
 - i. Parameter
 - 1. **Client_id**
 - 2. Contact's Name (**real_name**)
 - 3. Contact's Phone (**phone**)
 - 4. Contact's Email Address (**email**)
 - 5. Username (**login**)
 - 6. Password (**password**)
 - d. Notification
 - e. My Inbox
 - f. Logs
 - i. Activity Logs
 - 1. Contact ID
 - 2. Username
 - 3. Name
 - 4. Email
 - 5. Date (This information will be captured during the login process and sent to api.psychz.net)
 - 6. Time (This information will be captured during the login process and sent to api.psychz.net)
 - 7. IP Address (This IP will be captured during the login process and sent to api.psychz.net)
 - 8. Action
 - a. Disable ("primary contact" user can't be disabled)
 - g. Change Password
- 13. Logout

Service Cancellation Automation

The call for service cancellation will be sent by API only. For now the feature will be built inside centralized API and key will be generated for dashboard.psychz.net that will allow admins from dashboard.psychz.net to cancel services by entering the service ID in the field and clicking on cancel button. Once this tool is properly tested, service cancellation will be enabled for www.psychz.net/client using API key where it will give client the ability to request service cancellation for dedicated servers.

*It is critical that service cancellation is only done for services that are categories as dedicated servers only. All other services, will require manual steps and will be managed via dashboard.

*You will probably need to add VLAN database just like dashboard because that database is used in the logic. Here is a link to vlan database in dashboard -

<http://dashboard.psychz.net/dashboard/public/vlans>

Auto Service Cancellation Process

For service cancellation, the script will perform the following tasks automatically -

1. Cancel service based on service id input via API
2. Cancellation reason is required for each service. Please provide the following reasons and make sure they are logged -
 - 2.1. No Longer Needed
 - 2.2. Support Issues
 - 2.3. Network Issues
 - 2.4. Hardware Issues
 - 2.5. High Price
 - 2.6. Other
3. Get service by calling `client.service_get`
4. Verify the `plan_id` from the response is associated with a category that allows automated cancellation. *please see section 4.3.3 under centralized API admin.
5. If the service has no device associated with it, the script should simply skip to service and invoice cancellation and process the task.
6. Remove any edgelayar acl rules for the device IPs
 - 6.1. Please make sure to use SSH for PHP - <http://php.net/manual/en/book.ssh2.php>
 - 6.2. Script needs to take into account any failed tasks in centralized API. We need to be able to pull these failed logs and show them on dashboard homepage.
7. Remove any manual diversion rules for the device IPs.
 - 7.1. Script needs to take into account any failed tasks in centralized API. We need to be able to pull these failed logs and show them on dashboard homepage.
8. Remove any whitelisting on NTA for the device IPs
 - 8.1. This step will just require the script to create a ticket. So, please create a ticket in support department to remove it for now
 - 8.2.

Subject: Remove Whitelisting for Device ID XXXX

Body:

Hi Staff,

Please remove whitelisting for device ID XXX from NTA. Below you will find the IP's that belong to the device:

Device url: <http://portal.psychz.net/admin/devicemgr/view.php?device=xxxx>

xx.xxx.xx.xx/24

xx.xxx.xx.xx/32

=====

Auto Cancellation Script

9. Remove nulls for any of the IP's that is assigned to the device

- 9.1. Unnull code can be copied from

<http://dashboard.psychz.net/dashboard/public/nulledip>

- 9.2. Please make sure to use SSH for PHP - <http://php.net/manual/en/book.ssh2.php>

10. ~~Remove any SMTP blocks for the IPs on the router's VLAN. You will use the same logic that we use when adding/removing ips from router. We are after the logic that tells the script which router and vlan to go after.~~

- 10.1. ~~Here are the commands that will be executed on the router. vlanxx will be replaced with the vlanid. Usually xx is the device id.:~~

- 10.1.1.

```
conf t
interface vlanxx
no ip access-group block-smtp-out
exit
```

- 10.1.2. ~~Please make sure to use SSH for PHP~~

~~<http://php.net/manual/en/book.ssh2.php>~~

~~Subject: Manually remove SMTP block from VLAN XXXX on Router
xx.xxx.xx.xxx~~

~~Body:~~

~~Hi,~~

~~Please manually remove the SMTP Block from the VLAN on router
xx.xxx.xx.xxx~~

~~Router: xx.xxx.xx.xx~~

~~VLAN ID: xxxx~~

~~Device Id: xxxx~~

~~Log: <http://api.psychz.net/backend/web/log/details?id=xxx>~~

~~=====~~
~~Auto Cancellation Script~~

11. Remove the IP's from the device and also on the router

11.1. ~~Remove IP code can be copied from~~

~~<http://dashboard.psychz.net/dashboard/public/removeip>~~

11.2. ~~Please make sure to use SSH for PHP — <http://php.net/manual/en/book.ssh2.php>~~

11.2.1. ~~If telnet is being used, we will need to create a ticket in support department requesting manual removal of the IP's from the router.~~

~~Subject: Manually Remove IP's from VLAN XXXX on Router
XX.XXX.XX.XXX~~

~~Hi,~~

~~Please manually remove the following IP's from the VLAN on router
XX.XXX.XX.XXX~~

~~Router: XX.XXX.XX.XXX~~

~~VLAN ID:~~

~~IP's:~~

~~Log: <http://api.psychz.net/backend/web/log/details?id=xxx>~~

~~=====~~
~~Auto Cancellation Script~~

11.3. When cancelling services, we will simply issue a command for the entire VLAN that will remove all ips, configs that are associated with the VLAN. These commands are only to be used when cancelling a service where the goal is to remove all IPs, configs, smtp blocks, acls etc. We will not be using this command when removing specific IP ranges from a device or vlan.

11.3.1.

```
no interface vlan XXXX
interface vlan XXXX
no shutdown
ip verify unicast reverse-path
```

12. Disable monitoring for the device

12.1. Use device.monitor_list to get list of device monitors

- 12.2. Use `mod_id` obtained from `device_monitor_list` and disable it by using `device.monitor_disable`
- 13. Add the "Pending Reclaim" tag in `ubersmith`
 - 13.1. `device.tag` API call can be used to set the status to reclaim
 - 13.2. `tag=Pending Reclaim`
- 14. Remove the "inuse", "Pending cancellation" tag in `ubersmith`
 - 14.1. `device.untag` API call can be used to remove the "inuse" and "pending cancellation" tag
 - 14.2. `tag=In Use`
 - 14.3. `tag=Pending Cancellation`
- 15. Disable the bandwidth port in `ubersmith`
 - 15.1. If the script fails to disable the bandwidth port, we need to send our email so the staff can manually disable it

Hi,

The auto cancellation process failed to disable the device bandwidth port.
Please manually disable the bandwidth port on
<http://portal.psychz.net/admin/devicemgr/view.php?device=xxxx>

Once you are done, please close this ticket.

====

Auto Cancellation Script

- 16. Turn off device if that option is available
- 17. Add any account credit for unpaid balance
 - 17.1. Use `client.service_get` to obtain the `unpaid_balance`
 - 17.2. Use this unpaid balance amount to create credit using `client.credit_add` and make note of the `credit_id`
 - 17.2.1. Parameters
 - 17.2.1.1. `client_id`
 - 17.2.1.2. `value`
 - 17.2.1.3. `payment_type=Service Deactivation`
 - 17.2.1.4. `auto_apply=0`
 - 17.2.1.5. `pack_ids=xxxx` (This is service ID)
- 18. Apply credit to invoice related to the service
 - 18.1. using `client.credit_apply`, we need to apply the credit created in previous step to the invoice id obtained from the `client_service_get`
- 19. Set the service to cancel
 - 19.1. Using `uber.service_plan_update`, the script will set the `status=canceled`
- 20. Make sure child services are also canceled and follow the same logic
 - 20.1. Services with \$0.00 should be canceled

- 20.2. For bandwidth overages, please use “service deactivation” and these services are not required to go through cancel steps that involve creating credit, applying credit etc. We simply want to use deactivate option for bandwidth overages and that should automatically take care of any invoices related to the bandwidth overage service.
21. Remove device association with service and client.
- 21.1. You will use **device.update** and remove the **client_id** and **service_id**
22. Log the entry in logs.
- 22.1. It is critical that we can search the logs and look for logs by cancellation reason.
- 22.2. All failed tasks must be logged properly and we want to be able to view them by calling the api remotely in dashboard.
23. We want to delay sending the HD format ticket. So once the cancellation task is finished running, we want to wait 24 hours before sending the ticket to “Reformat Drive” queue. The new queue is located at https://portal.psychz.net/admin/supportmgr/ticket_list.php?queue=20&type=16&assign=All. We will need to set up internal cron that will allow the script to wait 24 hours after cancellation.
- 23.1. We also need the ability to cancel the HD format 24 hour queue. So , please add the link all the way at the bottom of the specific cancellation log where admins can cancel the request. If the admin clicks on the cancel link, the system will not send out any HD format emails.

Subject: HD Format Required For Device XXXX

Body:

Hi Staff,

Please format the drives belonging to Device XXXX:

Device url: <http://portal.psychz.net/admin/devicemgr/view.php?device=xxxx>

Once you are done, please click on <https://api.psychz.net/v1/devices/drives/format/complete/id/xxxx> and close this ticket.

=====

Auto Cancellation Script

*purpose of the link in ticket is to ensure the hard drives are formatted before we can set the tag to “available”. By clicking on the link, the script will ensure that formatting is done and it will automatically update the tag to “available”. Once the device is “available”, it is available to be used for other clients.

*purpose of the delay is because some clients cancel the service and request their data after the service is cancelled. Therefore, we will wait 24 hours after the service is cancelled before sending the HD format email. This way if the client does request the data, we can step in and cancel the HD format email.

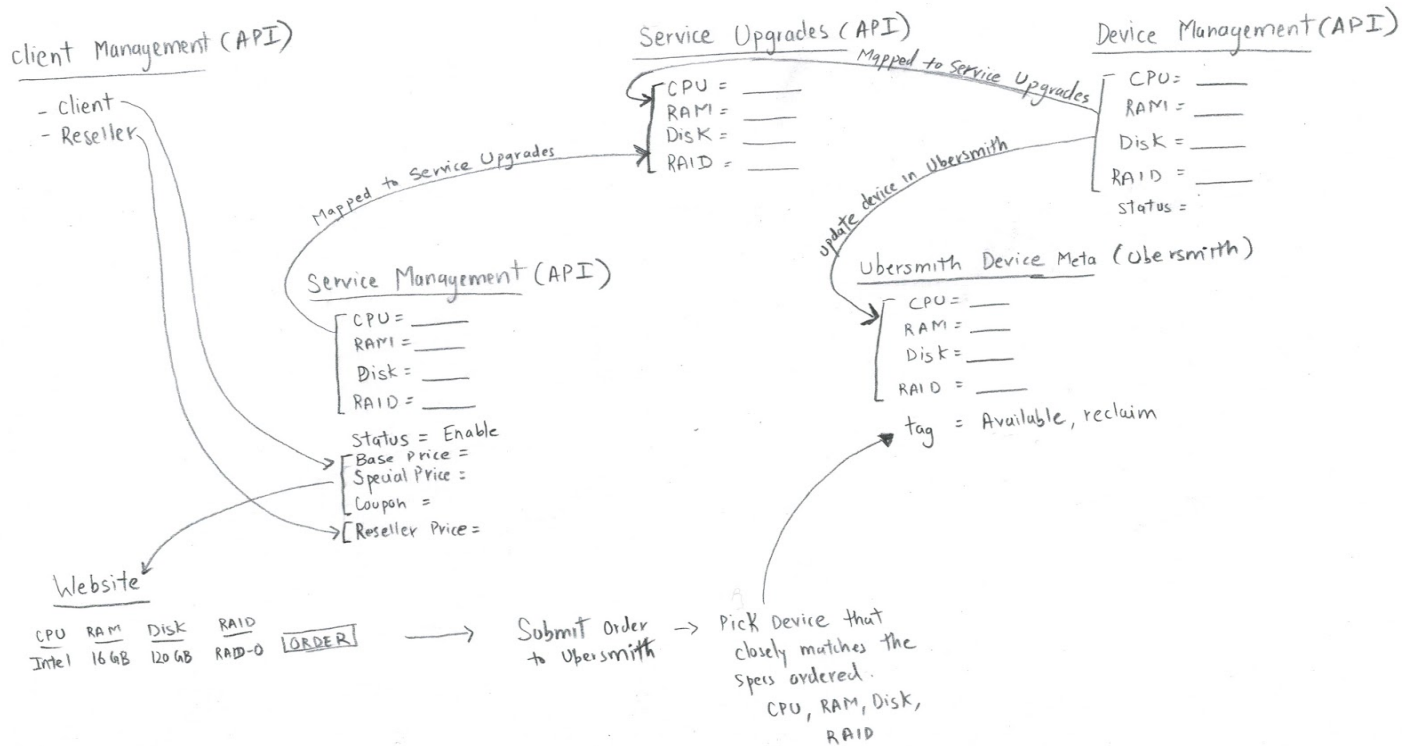
Service Order Automation

We need to automate the order processing in Psychz's Client Area. Client area will communicate directly with centralized API and centralized API will execute all task for client area, reseller api, dashboard, client area api etc.

*use the order manager for setup and API access.

Centralized API Admin

Backend Process For Automation



Download - <https://www.dropbox.com/s/lcmwsngsc2wddgx/CCI01282016.jpg?dl=0>

1. Manage Clients

1.1. List clients (**client.list**)

1.1.1. Quick search by client ID

1.2. Modify clients

1.2.1. Category

1.2.1.1. Client - default for all clients in ubersmith

1.2.1.2. Reseller

1.2.1.2.1. If reseller, select the reseller type - If reseller is selected, the system will automatically apply the reseller discount when displaying the prices. Reseller discount is only applicable to base prices of the service and % is provided on service level only. Reseller discount only applies to [dedicated category](#).

- 1.2.1.2.2. For reseller, we need the ability to add option that will allow admin to select and skip the auto promotion and demotion for the reseller.
- 1.2.1.2.3. Also, the script needs automatically track reseller level by looking at the active services that fall under [dedicated category](#).
 - 1.2.1.2.3.1. Level 1
 - 1.2.1.2.3.1.1. 3 Active services required and if the number of services fall below 3, the client ID should be automatically demoted to “Client” category. Once in client category, the client will be not auto promoted to reseller. The admin has to manually edit the client category. If the client reaches 10, the client should be auto promoted to level 2.
 - 1.2.1.2.3.2. Level 2
 - 1.2.1.2.3.2.1. 10 Active services required and if below, it will be demoted to level 1. If 30 active servers, the client will be auto promoted to level 3.
 - 1.2.1.2.3.3. Level 3
 - 1.2.1.2.3.3.1. 30 Active services required and if below, it will be demoted to level 2.
- 1.2.2. Permissions: By default all clients will get “client” permission which will not give any access to API or any of the tools via API.
- 1.2.3. Permission Management for contacts

Add New Contact

Profuse Solutions Internal Testing
Client ID: 7155

Contact Details

Login & Password

Permissions

Filter resources:

() shows inherited values

Resources	View	Create	Update	Delete
Client Portal	(None)	(None)	(None)	(None)
- Client Profile	Inherit	Inherit	Inherit	Inherit
Profile	(Inherit)	(Inherit)	(Inherit)	(Inherit)
Payment Methods	(Inherit)	(Inherit)	(Inherit)	(Inherit)
Change Password	(Inherit)	-	(Inherit)	-
Manage Contacts	(Inherit)	(Inherit)	(Inherit)	(Inherit)
Manage Contact Facility Access	(Inherit)	(Inherit)	(Inherit)	(Inherit)
- Billing & Services	Inherit	Inherit	Inherit	Inherit
Invoices	(Inherit)	(Inherit)	(Inherit)	(Inherit)
Services	(Inherit)	(Inherit)	(Inherit)	(Inherit)
Tax Exemptions	(Inherit)	(Inherit)	(Inherit)	(Inherit)
Credits	(Inherit)	(Inherit)	(Inherit)	(Inherit)
Orders	(Inherit)	(Inherit)	(Inherit)	(Inherit)
Paypal Subscriptions	(Inherit)	(Inherit)	(Inherit)	(Inherit)
Quotes	(Inherit)	(Inherit)	(Inherit)	(Inherit)
Billing Email	(Inherit)	(Inherit)	(Inherit)	(Inherit)
- Reports	Inherit	Inherit	Inherit	Inherit
Ledger Event Log	(Inherit)	(Inherit)	(Inherit)	(Inherit)
Mail Log	(Inherit)	(Inherit)	(Inherit)	(Inherit)
- Domain Management	Inherit	Inherit	Inherit	Inherit
List Domains	(Inherit)	(Inherit)	(Inherit)	(Inherit)
Lookup Domain	(Inherit)	(Inherit)	(Inherit)	(Inherit)
- Support	Inherit	Inherit	Inherit	Inherit
Tickets	(Inherit)	(Inherit)	(Inherit)	(Inherit)
Other's Tickets	(Inherit)	-	(Inherit)	(Inherit)
Ticket Search	(Inherit)	-	-	-
- Device Manager	Inherit	Inherit	Inherit	Inherit
Devices	(Inherit)	(Inherit)	(Inherit)	(Inherit)
Racks	(Inherit)	(Inherit)	(Inherit)	(Inherit)
Cages	(Inherit)	(Inherit)	(Inherit)	(Inherit)
Facilities	(Inherit)	(Inherit)	(Inherit)	(Inherit)
IP Addresses	(Inherit)	(Inherit)	(Inherit)	(Inherit)
DNS Zones	(Inherit)	(Inherit)	(Inherit)	(Inherit)
DNS Records	(Inherit)	(Inherit)	(Inherit)	(Inherit)

1.2.3.1.

1.2.3.2. Download -

<https://www.dropbox.com/s/068f7c0epzq5xhg/48aa80e4b13196998a51dcbe8b75e46d.png?dl=0>

1.2.3.3. List contacts (client.contact_list)

1.2.3.3.1. Edit Permission (This will be managed in api.psychz.net where contact access can be controlled for the following pages)

1.2.3.3.1.1. Dashboard

1.2.3.3.1.2. Service/Devices

1.2.3.3.1.2.1. active services

1.2.3.3.1.2.2. active devices

1.2.3.3.1.2.3. order service

1.2.3.3.1.3. Support

1.2.3.3.1.3.1. list tickets

1.2.3.3.1.3.2. open new ticket

1.2.3.3.1.4. Billing

1.2.3.3.1.4.1. unpaid invoices

1.2.3.3.1.4.2. list invoices

1.2.3.3.1.4.3. credit

- 1.2.3.3.1.4.4. list orders
- 1.2.3.3.1.4.5. payment methods
- 1.2.3.3.1.5. Reports
 - 1.2.3.3.1.5.1. traffic diversion
 - 1.2.3.3.1.5.2. DDos null/unnull
 - 1.2.3.3.1.5.3. Live DDos Report
- 1.2.3.3.1.6. Tools
 - 1.2.3.3.1.6.1. DDos Diversion
 - 1.2.3.3.1.6.2. Edge Layer ACL
 - 1.2.3.3.1.6.3. Shipping
- 1.2.3.3.1.7. My Account
 - 1.2.3.3.1.7.1. My Details
 - 1.2.3.3.1.7.2. My Contacts
 - 1.2.3.3.1.7.3. Notification
 - 1.2.3.3.1.7.4. My Inbox
 - 1.2.3.3.1.7.5. Change Password

1.2.4. Logs

2. Manage Services -

2.1. List services page will list out all services from ubersmith using

[uber.service_plan_list](#)

2.1.1. Edit option will allow modifying the fields listed under Manage Service Fields and will be managed inside centralized API database.

2.1.1.1. Name

2.1.1.2. Enable/Disable Main Website

2.1.1.2.1. This will enable and disable the service on main website
<https://www.psychz.net>

2.1.1.3. Enable/Disable Special Page

2.1.1.3.1. This will enable/disable the service on the special page
<https://www.psychz.net/special-psychz.html>

2.1.1.4. Category

2.1.1.4.1. Services will show based on the [service category](#) that is selected here.

2.1.1.4.2. After specific category is selected, it will auto load fields related to that category automatically without reloading the page. Edit Specs will list out all the features from the specific category and this information will be used to display the services via API for client area, reseller etc. Please refer to "[Edit Categories Section](#)"

2.1.1.5. Base Price

2.1.1.6. Special Page Sale Price -

2.1.1.7. Special Page Coupon

2.1.1.8. Reseller Level 1 Discount %

2.1.1.9. Reseller Level 2 Discount %

2.1.1.10. Reseller Level 3 Discount %

3. Manage Devices

3.1. List Devices using **device.list**

3.2. Modify Devices

3.2.1. Name

3.2.2. Location

3.2.3. Category

3.2.3.1. Fields will show based on the [device category](#) that is selected here.

3.2.3.2. After specific category is selected, it will auto load fields related to that category automatically without reloading the page. Edit Specs will list out all the features from the specific category.

3.2.3.3. The device information will be tracked in centralized API and it will be useful when selecting device for orders. The information in centralized API will need to be updated on the device page in ubersmith. Therefore, we will need to [map the fields to the ubersmith device parameters](#).

3.3. April 15, 2016 change --- Next step will involve updating the device page <http://api.psychz.net/backend/web/device/update?deviceId=xxxx>

3.3.1. For example -

<https://i.gyazo.com/fa027d01717c3b6859beff4e5ce6b40c.png>

3.3.2. We need to implement 3 content sections:

3.3.2.1. Device Info

3.3.2.1.1. Device (This information will be pulled from ubersmith and will not require any modification)

3.3.2.1.1.1. Device ID

3.3.2.1.1.2. Name

3.3.2.1.1.3. Facility

3.3.2.1.1.4. Device Category ID:

3.3.2.1.2. Client Info (client id and service ID will require option to be updated both by staff and via automation)

3.3.2.1.2.1. Client ID

3.3.2.1.2.2. Service ID

3.3.2.2. Tools

3.3.2.2.1. Assigned IPs (add ip / remove ip using the ip tools)

3.3.2.2.1.1. Add IP button will either show ajax expanded tool or pop up that will allow adding IPs to the device without leaving the page

- 3.3.2.2.1.2. Remove IP will submit the request to queue for IP removal without leaving the page
 - 3.3.2.2.2. Bandwidth Monitoring (enable/disable)
 - 3.3.2.2.3. Tags (add/remove)
 - 3.3.2.2.4. Welcome Email (send service welcome email if device has a service associated with it)
 - 3.3.2.3. Device Specs (Information will be updated). Most of the fields are already mapped to ubersmith, so i am providing the api parameter for ubersmith.
 - 3.3.2.3.1. Main
 - 3.3.2.3.1.1. CPU
 - 3.3.2.3.1.2. RAM
 - 3.3.2.3.1.3. Motherboard ([device_motherboard](#))
 - 3.3.2.3.1.4. RAID
 - 3.3.2.3.2. Hard Drives
 - 3.3.2.3.2.1. HD 1
 - 3.3.2.3.2.2. HD 2
 - 3.3.2.3.2.3. HD 3
 - 3.3.2.3.2.4. HD 4
 - 3.3.2.3.2.5. HD 4
 - 3.3.2.3.2.6. HD 5
 - 3.3.2.3.2.7. HD 6
 - 3.3.2.3.2.8. HD 7
 - 3.3.2.3.2.9. HD 8
 - 3.3.2.3.3. Ethernet
 - 3.3.2.3.3.1. Eth0 MAC Address ([eth0_mac_address](#))
 - 3.3.2.3.3.2. Eth1 MAC Address ([eth1_mac_address](#))
 - 3.3.2.3.4. IPMI
 - 3.3.2.3.4.1. IPMI IP
 - 3.3.2.3.4.2. Login ([ipmi_login](#))
 - 3.3.2.3.4.3. Password
 - 3.3.2.3.4.4. Admin Login ([ipmi_admin_login](#))
 - 3.3.2.3.4.5. Admin Password ([ipmi_admin_password](#))
 - 3.3.2.3.5. Control Panel
 - 3.3.2.3.5.1. URL: ([control_panel_url](#))
 - 3.3.2.3.5.2. Panel Login: ([control_panel_login](#))

- 3.3.2.3.5.3. Panel Password: (control_panel_password)
- 3.3.2.3.6. Server Login:
 - 3.3.2.3.6.1. Server IP:
 - 3.3.2.3.6.2. Server Login:
 - 3.3.2.3.6.3. Server Password:
- 3.3.2.3.7. Operating System
 - 3.3.2.3.7.1. OS:
 - 3.3.2.3.7.2. Arch Type:
- 3.3.2.3.8. DDoS Protection Level

4. **Manage Racks (May 2016)**

- 4.1. ~~List Racks~~ This page will list out all racks from ubersmith using
 - 4.1.1. ~~Racks can be filtered by Location~~
 - 4.1.1.1. Los Angeles
 - 4.1.1.2. Dallas
 - 4.1.1.3. Ashburn
- 4.2. ~~Modify Racks~~ This will be managed and tracked inside centralized API
 - 4.2.1. Category [More Info](#)
 - 4.2.2. Location
 - 4.2.3. Facility
 - 4.2.4. Space
 - 4.2.5. Circuit
 - 4.2.6. PDU 1
 - 4.2.7. PDU 2

5. **Mapping Tool**

- 5.1. Map Upgrades To Internal Services
 - 5.1.1. Call **uber.plan_upgrade_list** and Map DDoS Protection Level (pu_id=2151) To [IP Networks](#) category such as 2 Gbps, 10 Gbps etc. The goal is to help identify the order selection with the DDoS protection level category and allow the automation script to pick the right network and assign the IP from within that network. For example, if the client orders 20 Gbps protection, the script will go to network page and select IP networks in specific location that have 20 Gbps category. It will use those IP's to pull a specific range and assign it to the device.
 - 5.1.2. Call **uber.plan_upgrade_list** and Map IP upgrade (pu_id=2174) to specific IP ranges. For example - **/29 Private VLAN (5 Usable IPs)** has **po_id=7359** and we will map this to /29. If a client orders 20 Gbps DDoS protection, the script will first check and pick the right IP network that is assigned to 20 Gbps DDoS protection and afterwards, it will pick a /29 from those IP ranges and assign it to the device. For now, we want to offer /24, /25, /26, /27, /28, /29
- 5.2. ~~Map Upgrades To Device Categories~~

- 5.2.1. Call ~~uber.plan_upgrade_list~~ and map the entries with special device categories entries. For example—

Service Plan Upgrades	Device Specs
Processor:	Processor:
Atom D525 1.8GHz / 1MB Cache	Atom D525 1.8GHz / 1MB Cache
E6700 3.20GHz / 2MB Cache	E6700 3.20GHz / 2MB Cache
E5520 2.26GHz / 8MB Cache	E5520 2.26GHz / 8MB Cache

- 5.2.2. Location— Map to pu_id=2123 under service plan upgrades
- 5.2.3. CPU— Map to pu_id=2160 under service plan upgrades
- 5.2.4. RAM— Map to pu_id=2161 under service plan upgrades
- 5.2.5. HD 1— Map to pu_id=2162 under service plan upgrades
- 5.2.6. HD 2— Map to pu_id=2164 under service plan upgrades
- 5.2.7. HD 3— Map to pu_id=2165 under service plan upgrades
- 5.2.8. HD 4— Map to pu_id=2166 under service plan upgrades
- 5.2.9. HD 5— Map to pu_id=2168 under service plan upgrades
- 5.2.10. HD 6— Map to pu_id=2169 under service plan upgrades
- 5.2.11. HD 7— Map to pu_id=2170 under service plan upgrades
- 5.2.12. HD 8— Map to pu_id=2171 under service plan upgrades

5.3. Map Device Specs Header to Ubersmith Device Fields

- 5.3.1. We need to make sure we update the correct fields in Ubersmith device page, so we will need to map the device category headers with the API variable names in Ubersmith. When a staff look up the device under device management and update the fields, it will update the information in centralized API database along with fields on the ubersmith device page. The ubersmith device API parameters are provided in red below.

Central API Device Category	Ubersmith Device API Variable
CPU	device_cpu
RAM	device_ram
RAID Card	device_raid_card
Hard Drive 1	hard_drive_1
Hard Drive 2	hard_drive_2

Hard Drive 3	hard_drive_3
Hard Drive 4	hard_drive_4
Hard Drive 5	hard_drive_5
Hard Drive 6	hard_drive_6
Hard Drive 7	hard_drive_7
Hard Drive 8	hard_drive_8
IPMI IP	ipmi_ip
IPMI Password	ipmi_password
Server IP	server_ip
Server Login	server_login
Server Password	server_password
Operating System	operating_system
Arch Type	arch_type
Protection Level	ddos_protection_level

- 5.4. ~~Map Services to Devices~~—Tool will basically provide a search field that will search the device CPU field and list out devices that match the query.
- 5.4.1. ~~Search by Query and Location.~~ The search field will mainly look for device parameter field of ~~device_cpu~~
- 5.4.2. ~~Multiple Select boxes along with “Select All” option will allow updating of the device field~~ ~~service_plan_id~~
- 5.4.2.1. ~~Field “Service ID” will require admin to enter the numerical value and clicking on “Submit” button will update the devices with the specific plan ID.~~

6. Settings

- 6.1. System Config
- 6.1.1. Ubersmith Login Info
- 6.1.2. Locations
- 6.1.2.1. List Location
- 6.1.2.2. Add Location
- 6.1.2.3. Edit Location
- 6.1.3. Facilities
- 6.1.3.1. List Facilities
- 6.1.3.2. Map Facilities To Location

- 6.1.4. Routers
 - 6.1.4.1. List Routers
 - 6.1.4.2. Add Router
- 6.1.5. Edgelaye Router
 - 6.1.5.1. List Routers
 - 6.1.5.2. Add Router
 - 6.1.5.3. Edit Router
- 6.1.6. ADS
 - 6.1.6.1. List ADS
 - 6.1.6.2. Add ADS
 - 6.1.6.3. Edit ADS
- 6.1.7. NTA
 - 6.1.7.1. List NTA Server
 - 6.1.7.2. Add NTA Server
 - 6.1.7.3. Edit NTA Server
- 6.1.8. IP Networks
 - 6.1.8.1. List IP Networks - This page should have uber ID, IP Range, IP network type, location, ddos category (2 Gbps, 10 Gbps, 20 Gbps, 30 Gbps, 40 Gbps, 50 Gbps, 100 Gbps)
 - 6.1.8.2. Add IP Network
 - 6.1.8.3. Private Ip Ranges
- 6.2. Manage Rack Categories (May 2016)
 - 6.2.1. Dedicated Servers
 - 6.2.2. Colocation
- 6.3. Manage Service Categories
 - 6.3.1. List Categories
 - 6.3.2. Create Categories
 - 6.3.3. Edit Categories - This page will allow to the admins to create fields that are specific to the category and values that can be used for those fields.
 - 6.3.3.1.1. Dedicated Servers
 - 6.3.3.1.1.1. Location - Map to pu_id=2123 under service plan upgrades
 - 6.3.3.1.1.2. Number of CPU - Map to pu_id=2178 under service plan upgrades.
 - 6.3.3.1.1.3. Processor - Map to pu_id=2160 under service plan upgrades
 - 6.3.3.1.1.4. CPU Cores - Map to pu_id=2179
 - 6.3.3.1.1.5. Number of Threads - map to pu_id=2180
 - 6.3.3.1.1.6. Smart Cache - map to pu_id=2184
 - 6.3.3.1.1.7. Processor Base Frequency - map to pu_id=2181
 - 6.3.3.1.1.8. Hyper-Threading Technology - map to pu_id=2182
 - 6.3.3.1.1.9. Virtualization Technology (VT-x) - map to pu_id=2183

- 6.3.3.1.1.10. RAM - Map to pu_id=2161
- 6.3.3.1.1.11. HD 1 - Map to pu_id=2162
- 6.3.3.1.1.12. HD 2 - Map to pu_id=2164
- 6.3.3.1.1.13. HD 3 - Map to pu_id=2165
- 6.3.3.1.1.14. HD 4 - Map to pu_id=2166
- 6.3.3.1.1.15. HD 5 - Map to pu_id=2168
- 6.3.3.1.1.16. HD 6 - Map to pu_id=2169
- 6.3.3.1.1.17. HD 7 - Map to pu_id=2170
- 6.3.3.1.1.18. HD 8 - Map to pu_id=2171
- 6.3.3.1.1.19. Bandwidth - Map to pu_id=2172
- 6.3.3.1.1.20. Switch Port Speed - Map to pu_id=2173
- 6.3.3.1.1.21. IP Addresses - Map to pu_id=2174
- 6.3.3.1.1.22. DDoS Protection - Map to pu_id=2151
- 6.3.3.1.1.23. RAID - Map to pu_id=2167
- 6.3.3.1.1.24. ~~Allows automated cancellation (yes/no)~~
 - 6.3.3.1.1.24.1. Yes
 - 6.3.3.1.1.24.1.1. ~~Immediate - the system cancel the service using the auto cancellation tool right away~~
 - 6.3.3.1.1.24.1.2. ~~End of Billing Cycle - the system will auto-cancel at the end of billing cycle for the service~~

6.3.3.1.2. Colocation

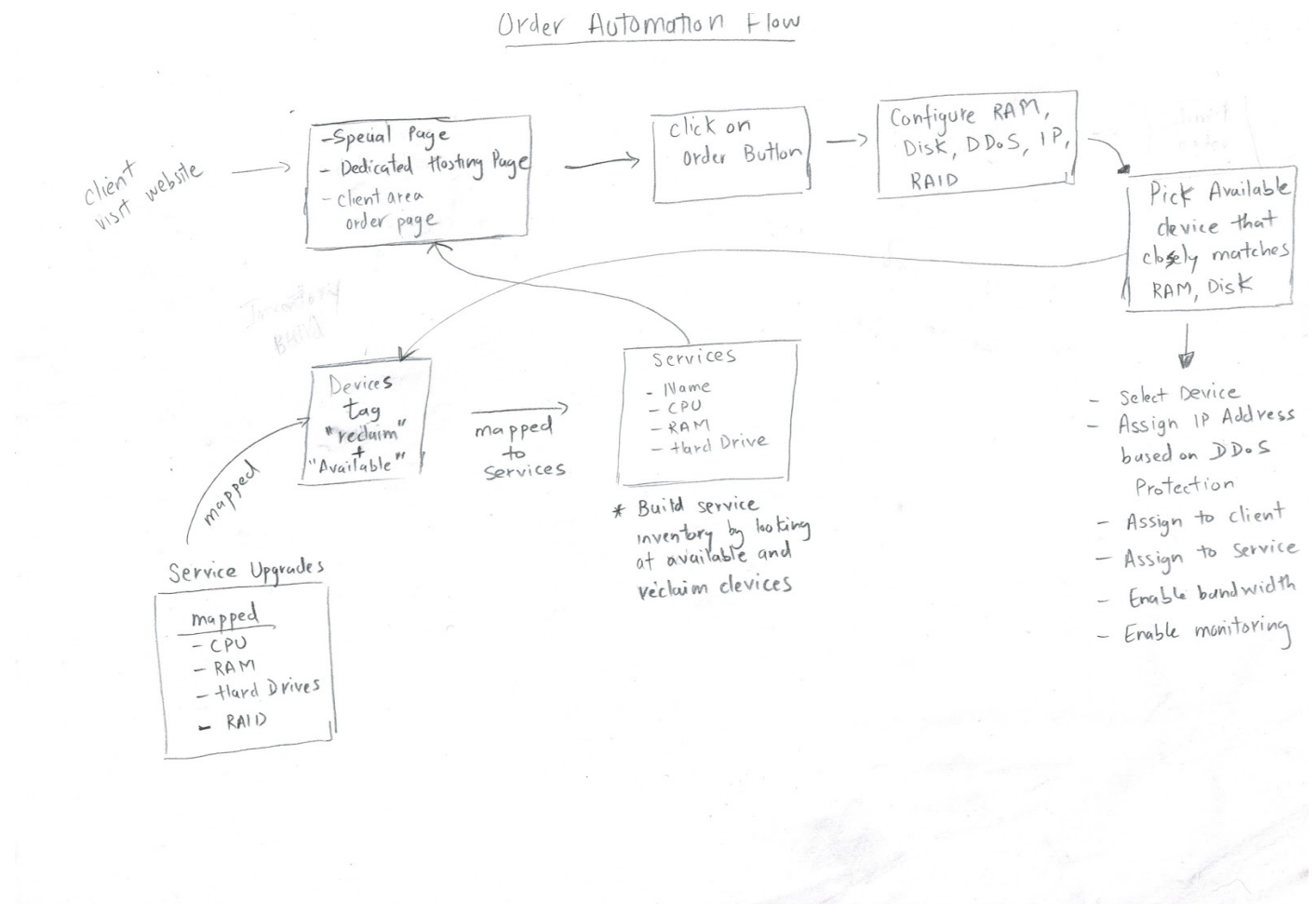
- 6.3.3.1.2.1. Rack Space
- 6.3.3.1.2.2. Power
- 6.3.3.1.2.3. Bandwidth
- 6.3.3.1.2.4. Included IPs
- 6.3.3.1.2.5. Switch Port Speed
- 6.3.3.1.2.6. Bandwidth Overage
- 6.3.3.1.2.7. Additional Bandwidth
- 6.3.3.1.2.8. DDoS Protection
- 6.3.3.1.2.9. 20A/120V Primary Power
- 6.3.3.1.2.10. 20A/208V Primary Power
- 6.3.3.1.2.11. 30A/208V Primary Power
- 6.3.3.1.2.12. 20A/120V Secondary Power
- 6.3.3.1.2.13. 20A/208V Secondary Power
- 6.3.3.1.2.14. ~~Allow automated cancellation (yes/no)~~

6.4. Manage Device Categories

- 6.4.1. List Categories
- 6.4.2. Edit Categories
- 6.4.3. Create Categories - This page will allow the admins to create fields that are specific to the category and value that can be used for those fields.
 - 6.4.3.1. Dedicated Servers

- 6.4.3.1.1. Location- Map to pu_id=2123 under service plan upgrades
- 6.4.3.1.2. CPU - Map to pu_id=2178 under service plan upgrades
- 6.4.3.1.3. RAM - Map to pu_id=2161 under service plan upgrades
- 6.4.3.1.4. HD 1 - Map to pu_id=2162 under service plan upgrades
- 6.4.3.1.5. HD 2 - Map to pu_id=2164 under service plan upgrades
- 6.4.3.1.6. HD 3 - Map to pu_id=2165 under service plan upgrades
- 6.4.3.1.7. HD 4 - Map to pu_id=2166 under service plan upgrades
- 6.4.3.1.8. HD 5 - Map to pu_id=2168 under service plan upgrades
- 6.4.3.1.9. HD 6 - Map to pu_id=2169 under service plan upgrades
- 6.4.3.1.10. HD 7 - Map to pu_id=2170 under service plan upgrades
- 6.4.3.1.11. HD 8 - Map to pu_id=2171 under service plan upgrades
- 6.4.3.1.12. IPMI Password - Just a basic text entry field is required
- 6.4.3.1.13. Server IP - mapped to server_ip parameter under device api
- 6.4.3.1.14. Server Login - mapped to server_login parameter under device api
- 6.4.3.1.15. Server Password - mapped to server_password parameter under device api
- 6.4.3.1.16. RAID Selection - Map to pu_id=2167 under service plan upgrades
- 6.4.3.1.17. Mitigation Method
- 6.4.3.1.18. DDoS - Map to pu_id=2151 under service plan upgrades

Service Order Automation Flow



Flow Download -

<https://www.dropbox.com/s/lvl3tz49g8nq7s1/order-automation-workflow.jpg?dl=0>

Client Side Order Automation

1. The website will list active services using the API call to api.psychz.net.

```

- Main
--Category
---Dedicated Servers
-----Los Angeles
-----1 CPU
-----2 CPU
-----Dallas
    
```

```

-----1 CPU
-----2 CPU
-----Ashburn
-----1 CPU
-----2 CPU

- Special
--Category
---Dedicated Servers
-----Los Angeles
-----1 CPU
-----2 CPU
-----Dallas
-----1 CPU
-----2 CPU
-----Ashburn
-----1 CPU
-----2 CPU

```

2. Client visits <https://www.psychz.net/dedicated-hosting.html> or <https://www.psychz.net/special-psychz.html>
 - a. <https://www.psychz.net/special-psychz.html> will list out all active services that have enable for “special” flag along with the coupon code and special price
 - b. <https://www.psychz.net/dedicated-hosting.html> will list out all active services that have “enable” status for “main” tag.
 - c. If the client is reseller and he/she is logged in, they will see the discount prices automatically and the order process will take that into account. The system should always show base price and show discount by crossing the base price and listing the discount price.
 - d. Service that have no availability will be listed as “Out of Stock” and will have a “Get Notified” button.
 - i. Clicking on “Get Notification” will simply make an entry in database and it will notify the client that server is in stock. Once notification is sent, it will discard the get notification request until the customer sign up again to be notified.
3. Customer will go through the order process on the website and place the order and the order tracking will take place on <https://www.psychz.net/client/order/complete> url.
 - a. At this stage the script will create the order in ubersmith and do few checks before moving to automated stage
 - i. verify the fraud score
 - ii. verify the invoice is paid
 - iii. Initiate order automation by logging the task in <http://api.psychz.net/backend/web/log/index>

Selecting Device Logic

When selecting devices, the script needs to look at the devices in the following order:

- Device tag
 - First look for devices with reclaim or available tag
- CPU
 - Look for devices that match the CPU
- Hardware RAID
 - If hardware RAID is ordered, only look for devices with specific RAID setup.
 - If none found, find device that matches that HD requirement for RAID.
 - If RAID 0, RAID 1, the device must have 2 equal size drives
 - If RAID 10, the device must have 4 equal size drives
 - If still no devices, find device that matches the RAM requirement and request hardware changes.
- RAM
 - Look for devices that match the RAM requirement. If none found, find the one that comes close to requirement
 - For example, if client orders 64 GB and script is only able to find servers with 16 GB, 32 GB and 128 GB, it will select 32 GB device because it will just require adding of 32 GB RAM to the device.
- HD
 - Final step involves checking for HD and try to find the device that matches the closest HD requirements.
 - If the client orders 2 TB and script only finds a server with 256 GB SSD and 2 TB drive, the script will select that server and request removal of 256 GB SSD by staff.

Example in Google sheets -

<https://docs.google.com/spreadsheets/d/1Z-xSRcYuGXvsCOJlozNF1ZcJPVFzUGNTZFm0bJAbTqQ/edit?usp=sharing>

Example -

Order	Priority 1 Device	Priority 2 Device	Priority 3 Device
E3-1230 v3	E3-1230 v3	E3-1230 v3	E3-1230 v3
32 GB	12 GB	32 GB RAM	16 GB RAM
Hardware RAID-10	Hardware RAID-10	None	None
1 TB	1 TB	1 TB	256 GB SSD
1 TB	1 TB	1 TB	
1 TB	1 TB	1 TB	
1 TB	1 TB	1 TB	
	Will just require RAM change	Will require RAID-10 Build	Will require addition of 16 GB RAM, removal of 256 GB SSD, addition of four 1 TB drives, addition of RAID card

Order	Priority 1 Device	Priority 2 Device	Priority 3 Device
E3-1230 v3	E3-1230 v3	E3-1230 v3	E3-1230 v3
16 GB	16 GB	16 GB	32 GB RAM
None	None	None	None
2 TB	1 TB	256 GB SSD	1 TB
		2 TB	1 TB
		1 TB	1 TB
		1 TB	1 TB
	Will require HD removal and addition	Will require HD removal	Will require RAM removal and HD removal and HD addition.

OS Install Automation

Below are some of the requirements for OS Install automation -

- The script will run in api.psychz.net
- SSH connection is recommended when connecting to a server
- We can capture the following information before starting the OS install:
 - IP Address
 - OS Name
 - Hostname
- Security
 - Change boot order via IPMI to pxe prior to install
 - Change the boot order to Hard disk after OS install is done
- Kickstarter templates will be required for each OS type.

Building inventory in Dashboard

1. We need to create inventory section in dashboard that will use the centralized API to get the list of available servers
 - a. List Available Servers
 - b. We will be building the inventory by looking at devices in Ubersmith and only count the one's with reclaim tag.
 - c. We will be using `device.list` API call and get all devices and filter them for reclaim tag first and afterwards facility.
 - i. reclaim tag parameter is `tag_id` where `id=2`
 - d. This database needs to be updated every few minutes to make sure we have the latest information when client's are ordering on the website
 - e. Packages will be listed with option of "Order Now" and "Customize"
 - i. Order now button will not give customization option when it comes to RAM, disk space, RAID
2. The Order Process
 - a. Client visits - <https://www.psychz.net/dedicated-hosting.html>
 - b. The page will display if the service is in stock / out of stock
 - c. Client clicks on the order now button for the service in stock

- d. Client will be required to login or sign up
 - i. If signing up, the script will check the information against fraud database.
For more info - <http://dev.maxmind.com/minfraud/>
 - 1. If a fail score is returned, the script will forward the client to fraud page where the client will need to provide documentation for verification.
- e. Client goes through the order process and pays for the service
- f. The script verifies the invoice is paid
- g. The script will show pending order in
<http://dashboard.psychz.net/dashboard/public/>
- h. The admin's responsible for sales/provisioning will check the order and if approved click on the "Reject" or "Approve" button
 - i. If rejected, the script will refund the invoice and cancel the service associated with the invoice. It will also notify the client that order wasn't processed
 - ii. If approved, the following things need to take place
 - 1. A device needs to be picked and associated with the service
 - 2. Make sure there are no IP's assigned to the device and on the router's vlan for the device
 - 3. Next it will assign an IP range to the device based on the order specs
 - 4. The script will check the null router to make sure none of the IP's are nulled
 - 5. It will configure the device with the OS choice from the order and using PXE it will configure the server to install the desired OS

Website Listing

- Get notified when server is back in stock
- Get notified when specials are launched

Dashboard

Provisioning

- Import replies from focus colo tickets to dashboard.psychz.net
- Show oldest provisioning tickets with a red warning

- Sort tickets from oldest to newest and sort them from Immediate status to low status
- Release server up to IPMI and send email that requests client if they wish for us to install the OS. If they click on the link, it will open a ticket to support dept requesting OS install.

Communicating with centralized API from client area

We need to show (in stock / out of stock) on our website for dedicated server packages at <https://www.psychz.net/dedicated-hosting.html>

Listing will be sorted/updated automatically by location (Los Angeles, Dallas, Ashburn) and also based on CPU quantities (Single vs Dual CPU)

- List All Available Servers (GET inventory/servers/all)
- List Available Single CPU Servers (GET inventory/servers/singlecpu)
- List Available Dual CPU Servers (GET inventory/servers/dualcpu)
- List Available Servers by location (GET inventory/servers/all/location/:id)
- List Available Single CPU Servers by location (GET inventory/servers/singlecpu/location/:id)
- List Available Dual CPU servers by location (GET inventory/servers/dualcpu/location/:id)