

## Personal

### Michael Wallace Verhulst

US Citizen

Email: michael@terminallabs.com

Phone: (512) 270-8665

## Summary

My name is Michael Verhulst, I am an experienced Python Developer and Consultant. I am very energetic, flexible, hard working, skilled and based in Austin, TX. I have worked on multiple successful Python and Linux developments projects. I have worked with clients of many different sizes from a diverse range of fields. I love new and interesting challenges. Along with my real world development experience, I have a strong background in mathematics, astrophysics, and aerospace and I like to use my skills in these areas in projects that I work on.

## SKILLS

### Computer Science/ Programming:

Full-stack Development, Python Development, Python Optimization, Linux Application Development, Database Development and Administration, Server Administration, Algorithms, Computational Complexity Theory, Data Compression, Encryption and Data Security.

### Physical Science/Mathematics:

Astrophysics, Cosmology, Spacecraft Propulsion, Condensed Matter Physics, Quantum Mechanics, Nanotechnology, Relativity, Cell Biology, Number Theory.

### Miscellaneous:

Networking, Finance, Economics Patent Law, Securities Law, Aerospace and Aviation, Rockclimbing, Car repair/maintenance.

### Computer Hardware:

- I have assembled a small 16 node "cluster" and created a proof-of-concept load balancer for it in python.
- I have performed remote pc/mac maintenance via TeamViewer and LogMeIn.

- I have personally assembled, repaired, and/or upgraded hundreds of computers. Mostly Dells among other vendors, and some DIY customs builds in many configurations.
- I built a dedicated server for kvm, VMware, ESX, ESXi, VSphere, and compiled Xen for a round of experiments to configure a hypervisor with VT-d based PCI passthrough.
- I built a dedicated GPU server for high performance vector machine processing of MRI data with Nvidia CUDA.
- I built a dedicated buildbot server for continuous integration and automated testing of software projects that require VT-x and VT-d instructions in the cpu.
- I have done some hobbyist experimentation with Raspberry Pi including:
  - Specialize SD card formatting for os reinstall.
  - Performance testing of computational tasks run in Wolfram Language (Mathematica)
  - Ran a demonstration web service (written in Python) on the device.
- I have installed network infrastructure and equipment for clients. This includes:
  - installing wireless Internet systems, configuring routers, running cabling, cable management, setting up multiple monitor workstations, building racks and rack-mountable servers, converting tower servers into rack-mountable servers, upgrading servers, installing servers into cabinets with or without rails, configuring multi-boot systems, installing printers, settings up networks.
- I am familiar with (and have made apps for) many Apple products. These products include:
  - iPhone 5s, iPhone 4s, iPod 4, iPad mini, iPad Air, MacBook, Mac mini.

### **System Administration / DevOps:**

Admin/DevOps tool I have worked with:

- SSH, Coreutils, OpenVPN, OpenSSL, EC2, S3, Cloudfront, Route53, VirtualBox, Vagrant, SaltStack, Chef, Fabric, Libcloud, OpenStack, nmap, Wireshark / Ethereal, Bind9, Gearman, Supervisor, Terraform, lxc, docker.

Cloud providers and datacenters I have worked with:

- AWS, DigitalOcean, Rackspace, OnRamp, m5hosting.

DevOps highlights:

- I have experimentally run OpenStack on Dell R220 servers, in a handful of different configurations.
- I have created hundreds of virtual machines on a variety of providers, locally and in the cloud.
- A have worked with a AWS based vm cluster with over one hundred virtual machines all running at the same time all managed by SaltStack.

**Programming Languages:**

• HTML/CSS	13 years
• JavaScript	11 years
• Python	9 years
• jQuery	8 years
• C/C++	5 years
• Fortran	3 years
• SQL	3 years
• Ruby	2 years
• Wolfram Language (Mathematica)	1 year
• Objective-C	1 year
• Swift	<1 year
• Go	<1 year

**WORK EXPERIENCE**

- Python Consultant ----- 2007 – present  
I have lead the implementation of some advanced technologies, worked with a range of clients, lead development projects, designed and installed server/networking gear, worked with marketing firms, interacted with UI/UX designers/design firms, given a few public presentations (some in front of 100-200 people), Developed projects from scratch, inherited projects midway through development, spent a large amount of time researching emerging technologies, and written a lot of code.

- Temporary Web Design Consultant ----- August 2007 – November 2007  
As a freelance web design consultant I designed and implemented some aesthetic and code changes to a high traffic area of the medical school's website. This project was entirely front-end development. After mocking-up the design in Photoshop, I created a new static webpage by hand-coding the relevant source-code in HTML/CSS/JS.
- Counselor at the U.S. Space & Rocket Center ----- January 2006 – May 2006  
I taught several hundred children (and a few adults) from around the world a little bit about the exciting history of manned space flight and the history of the US space program. My duties included: academic tutoring, teaching classes on space science/aerospace engineering, keeping the team energised and on schedule, running some fairly complex flight simulator software, lifeguarding, leading teams of 8-12 campers through numerous engineering/design competitions, giving advice about how best to prepare for a career in aerospace and or science, helping campers complete challenging physical obstacle courses.
- Rock Climbing Instructor and Lakeside Lifeguard at BSA Summer Camp - Summer 2005  
I taught around 200-300 people, on a 75' climbing tower, basic/intermediate rock climbing techniques, equipment usage and safety procedures. When I was not on the climbing tower, I was a lifeguard/swimming instructor and astronomy/spaceflight merit badge teacher. I covered Apollo, New Horizons, and Mars Exploration Rover among other topics. As my first job, it was not as technical as my professional work experience or university studies, yet, leading and challenging so many people to do something that they had never done before was a very valuable learning experience.



## PROFESSIONAL PROJECTS

### **PYTHON CONSULTATION PROJECT**

Cox Media Group - Platform ----- February 2016 – April 2016

**Cox Media Group**, a department within Atlanta-based Cox Enterprises - Cox Media Group. The department's operations include maintaining infrastructure and platform technologies needed to run the organization's complex content management system.

Technologies used:

Python, jQuery, Apple OS X, HTML5/CSS3/JS, Git, Hg, Virtualenv, Vagrant, Node.js, Adobe Creative Cloud, Office, Less, Open Graph Protocol, Django.

My involvement included:

- Worked with HTML5/CSS3/Less//JS to make several small improvements to the look and feel of customer facing webpages.
- Worked with the 3-party tool Livefyre in order to modify how their service presented shared links on social networking sites.
- Debugged our organization's implementation of janrain, a 3-party customer profile-management tool.
- Worked with Apple OS X, iPhone 5s, iPhone 4s, iPad Air, mobile chrome, and safari in order to debug and fix an intermittent problem that only appeared on mobile devices.
- Worked with the 3-party tool Janrain while debugging and fixing a race condition where certain JS files were not being loaded in the customer's web browser in the correct order.
- Worked with the Open Graph Protocol in order to debug and fix an issue with how shared videos were playing on Facebook's wall.
- Provided general code maintenance and refactoring.
- Worked with Virtualbox and Vagrant to setup multiple VM guests for testing/development and other various uses.
- Studied the 3-party online video platform, Brightcove, in order to debug a ticket, we needed to gain a better understating of how Brightcove served video data.
- Modified Django templates.
- Modified backend Python files containing Django template tags.



- Updated a Python unit test.
- Fixed bugs.

## PYTHON CONSULTATION PROJECT

Cox Media Group - Radio ----- August 2015 - December 2015

**Cox Media Group - Radio**, a department within Atlanta-based Cox Enterprises - Cox Media Group. The department's operations include maintaining 86 radio stations.

Technologies used:

Python, CentOS, jQuery, Linux Mint, Apple OS X, HTML5/CSS3/JS, Git, Hg, Virtualenv, Vagrant, Bootstrap, Node.js, Adobe Creative Cloud, Office, Jade Templates, SCSS, Less, Glyphicons, Flask, Dropbox, SQLite, Jinja2 Templates, Nginx, Grunt, MongoDB, File Compression, Flot, AJAX, SQLAlchemy.

My involvement included:

- Converted PNG file, Office file, and/or PDF file based webpage design mock-ups files into fully functional interactive Node.js based webpages and UI features.
- Provided general code maintenance and refactoring.
- Worked with HTML5/CSS3/Less/SCSS/JS to make many small improvements to the look and feel of customer facing webpages.
- Worked with Virtualbox and Vagrant to setup multiple VM guests for testing/development and other various uses.
- Fixed bugs.
- Created new Jade templates which were rendered into new webpages.
- Refactored HTML5/CSS3/JS code in order to be more cross browser compatible and responsive to work well on small screens.
- Prototyped a Python Flask based web portal designed for streamlining the workflow of an internal reporting auditing process.
- Utilised the python module Django-htmlmin to to compress the HTML generated by Flask for the web portal.
- Worked with the Python module Flask-security to implement: user authorization, form validation, and cross-site request forgery protection. For the web portal.
- Worked with Flot/HTML/JS to add and modify live data-driven charts that



were rendered in the client-side browser. The data for these charts was generated in the backend and passed to the browser via AJAX and JSON.

- Worked with MongoDB/SQLAlchemy/SQLite to create a backend datastore that worked well with Python and Flask and supported storing user data for the web portal.
- Utilised Python decorators in order to streamline the configuration of which pages rendered by Flask (in the web portal) could be accessed by standard users and which pages could only be accessed by admin users.
- Converted existing static HTML5/CSS3/JS based webpages into HTML5/CSS3/SCSS/JS Jinja2 Template based pages rendered via Node.js.
- Studied, Audited and began automating an internal process that had previously be done manually once each month.
- Wrote documentation in order to help formalise the exact steps that needed to be carried out by any staffer who was assigned the task of completing the monthly internal reporting auditing process.
- Completed the monthly internal reporting auditing process. For a few months in late 2015.
- Worked with colleagues to specify how to best utilize Bootstrap in order to allow our internal project to responsively reflow at 5 breakpoints at 5 device/screen sizes.

## PYTHON CONSULTATION PROJECT

Cox Media Group ----- June 2014 – August 2015

**Cox Media Group**, a subsidiary of Atlanta-based Cox Enterprises, is an integrated broadcasting, publishing and digital media company. The company operations include 15 broadcast television stations and one local cable channel, 86 radio stations, four metro newspapers, more than a dozen non-daily publications and more than 100 digital services. The company operates one of the largest productions deployments of Django in the world.

Technologies used:

Python, Django, Django-South, jQuery, Ubuntu, HTML5/CSS3/JS, Git, Virtualenv, Vagrant, Adobe Analytics, New Relic, Brightcove, Qualtrics, Outbrain, AngularJS, Lazythumbs, Less, Grunt, JIRA, Review Board, BeautifulSoup, CoffeeScript.

My involvement included:

- Converted PNG and/or PDF webpage design mock-ups files into fully functional interactive HTML5/CSS3/Less/JS based webpages and UI features.
- Provided general code maintenance and refactoring.
- Created new Django HTML templates which were rendered into new customer facing webpages viewed by millions of users across the country.
- Helping successfully achieve launch goals of JP2 in the Fall of 2014.
- Worked with CMG's automatic ad serving systems to rearrange how and where certain ads were included in customer facing webpages.
- Worked with HTML5/CSS3/Less/JS to make many small improvements to the look and feel of customer facing webpages.
- Reworked how image thumbnails are generated via Django and Lazythumbs.
- Audited the CMG's usage of browser cookies in order to decrease the amount of data that customer's browsers needed to download in order to render a page. Outlined a set of recommendations for CMG to edit and in some cases remove unneeded cookies.
- Worked with Brightcove and Qualtrics to reduce the size of the cookies that they included in webpages served by CMG.
- Worked with Virtualbox and Vagrant to setup multiple VM guests for testing/development and other various uses.
- Worked with Adobe Analytics tools to add statistics gathering instrumentation to CMG's online assets.
- Worked with the Python library BeautifulSoup
- Worked with tools from Outbrain to better associate CMG's online content with related content elsewhere on the internet.
- Modified which site URLs are rendered by Django's sitemap generating system in order to create better organized sitemap.xml files.
- Fixed many bugs. These fixes ranged from small visual problems to complex backend issues.
- Wrote documentation for internal code and systems as well as additional documentation and usage summaries for third party products.
- CMG needed to gain a better understanding of their customer's browser resource utilization and in-browser performance statistics. So I worked with New Relic's services to build out custom JS instrumentation for inclusion in their webpages. I worked with New Relic, Application Performance Monitor (APM), Browser, Insights, and New Relic Query Language (NRQL).



## PYTHON CONSULTATION PROJECT

modavanti.com ----- March 2013 – June 2014

**Modavanti** is a New York based E-commerce company that sells sustainable fashion from around the world. They bring an element of environmental and social awareness to the runway.

### Technologies used:

Python, Django, Django-South, jQuery, Rackspace, elasticsearch, Ubuntu, Chef, HTML5/CSS3/JS, Git, Mercurial, Virtualenv, Gearman.

### My involvement included:

- Modavanti had their own internal graphic designer who created mock-ups and designed the UI for the website. I worked with him to provide technical recommendations as to the development schedule/feasibility of proposed features.
- Converted Adobe Photoshop/Illustrator design mock-ups into fully functional interactive HTML5/CSS3/JS based webpages and UI features.
- Reworked how product images are downloaded from the Rackspace CDN.
- Worked with Elasticsearch to enhance how customers search for products.
- Refactored Javascript classes that handled the behavior of the infinite-scroll shop page.
- Worked with Rackspace's API to their CDN in order to batch upload image files.
- Worked with Chef, Ruby and some recipe files to spin up some background Gearman worker tasks (which in turn ran on a rackspace VM) for logging user click activity into the database.
- Worked with the Python Gearman client to create some worker scripts that were deployed to the background worker server.
- Revised the project documentation to better describe how to perform a full-stack restart/cloud infrastructure transfer.
- Research the Kissmetrics analytics system and it's API in order to learn how best to integrate its features into the core project.
- Studied Chef and Salt in consideration of switching to Salt.
- Worked with Virtualbox to setup multiple VM guests for testing/development and other various uses.
- Supported Modavanti's cloud infrastructure both the production VMs and



the staging VM.

- Maintained the Supervisor configuration that controlled the Gearman background worker processes.
- Modavanti needed to gain a better understanding of their customer's click-through paths, so I used Django-South to modify the database schema in order to store click event data. I also customized the Django Admin portal so that it would display the data.

## PYTHON CONSULTATION PROJECT

mobiusmed.com ----- June 2013 – December 2013

**Mobius Medical Systems** offers innovative software for modern radiation oncology. Specifically, they offer a products that helps Medical Physicists analyze radiation treatment plans for accuracy. Intensity modulated radiation therapy allows for radiation to more selectively target cancer cells. However, plan error tends to increase with modulation. To fully leverage modulation, plan error must be controlled as much as possible.

Technologies used:

Python, CouchDB, PyCUDA, NumPy, Flask, Genshi, Foundation, jQuery, Ctypes, Coffee-Script, couchdb-python, HTML5/CSS3/JS, Selenium, PhantomJS, Git, Virtualenv.

My involvement included:

- Worked with CouchDB to create new MapReduce views to distill data.
- The Mobiusmed wanted to be able to access Nvidia GPU cards from a virtual machine. So I setup VMware, ESX, ESXi, VSphere, and Xen for a round of experiments to try and configure VT-d based PCI pass-through.
- Ported an algorithm originally written in NumPy to C, to be called in Python via ctypes.
- Added in product features, by working with Python Flask views, Genshi templates and HTML5/CSS3/JS. Also used GIMP and Adobe Photoshop to edit some web site artwork/images.
- Worked with PhantomJS to reconfigure the DPI resolution of exported PDF files.
- Worked with the Nvidia CUDA compiler on Linux Ubuntu and PyCUDA.
- Made many aesthetic and UI interactivity changes to the project code.



- Worked with WebGL and the scientific visualization toolkit XTK.
- Worked with Selenium to create a Python module that could be used to run UI tests against the HTML5 interface.
- Refactored some aspects of how data is stored and retrieved to/from CouchDB.
- Moved/created python functions that forked off as child processes to run in the background.
- Created a forked process monitoring module that gathers performance statistics ( including memory usage, cpu-time, wall-time) of each child process and stores this data into CouchDB.
- Created some HTML email and interfaced with the Python module smtpplib to send them.
- Developed a UI feature that displayed complex data, distilled from multiple sources, in a simple tooltip table. Used the both Foundation 2.0 and the jQueryUI, ultimately chose jQueryUI.
- Wrote unit tests, using the unittest Python module and Selenium

## PYTHON CONSULTATION PROJECT

agyield.com ----- January 2012 – March 2013

**Agyield.com** offers a very useful tool for risk management for farmers that allows them to understand the costs and risks related to their crops in order to keep track of and project profit scenarios.

Technologies used:

Python, Django, Less, AWS, MySQL, jQuery, Ubuntu, HTML5/CSS3/JS, Mercurial.

My involvement included:

- Worked with the CEO and CTO of the company that offers the agyield.com service to understand and revise the development roadmap.
- Performed a broad ranging review of the Agyield's technical operations.
- Worked with AWS to help maintain the Agyield's productions servers.
- Studied commodities trading and crop insurance.
- Worked with ReportLab and wkhtmltopdf to programmatically create PDF files.
- Worked with the Agyield's ETL systems that regularly downloaded and



- processed data purchased from a third party.
- Implemented many aesthetic and code changes in HTML5/CSS3/JS.
  - Refactor HTML5/CSS3/JS code in order to be more cross browser compatible and work well on a small screen.
  - Modified how and when emails were pragmatically sent.
  - Optimized/Refactored crop insurance calculation functions in Python.

## PRO BONO PROJECT

Disaster Recovery for client ----- February 2013

**Client** was a small company that had unrecoverably lost it's source code, and only had a single copy on their production server running their webapp. They didn't know how to access this copy safely, or if it was a complete repository or just a single checkout of the code. I worked with the client to understand the nature of the problem and estimate its severity. I participated in the creation and execution of a plan of action that would fix the issue, I also helped explain to the client what had been done in order to remedy the situation.



## SIDE PROJECTS

### OPEN SOURCE PROJECT

terminallabs.com ----- 2011 - Present

**terminallabs.com** is our own company website. Through it we have taken the opportunity to use several technologies. Though we have a fairly simple website, the body of work that supports our website continually evolves. As much as our time allows, we habitually use this to experiment with new technologies and methodologies, while striving to maintain ease of maintainability. Currently, our website is using Lektor as a CMS, Framework, and Static Site Generator.

The work is open source and can be found here:

[https://bitbucket.org/terminal\\_labs/tl-lektor](https://bitbucket.org/terminal_labs/tl-lektor)

Technologies used:

Python, Django, Mezzanine, Flask, Lektor, AWS (EC2, S3, Cloudfront, Route53), DigitalOcean, Vagrant, Salt Stack, Jira, Bitbucket, Git, Mercurial, Postgres, Debian, Buildbot, Jinja, Less, jQuery, Locust, Selenium.

My involvement included:

- Iteratively developing the frontend codebase working with HTML5/JS/LessCSS and Django or Jinja templates
- Iteratively developing the Python server backend with Django / Mezzanine / Flask. This included creating models changes, views, and modifying the CMS for altering content in admin.
- Using Lektor as a web-framework and static site generator and hosting this static site over AWS S3 & Cloudfront with SSL.
- Maintaining multiple code repositories in Mercurial and Git.
- Creating a repository that heavily uses Vagrant and Salt Stack to create and manage both development and production environments on either AWS or DigitalOcean. With a simple `vagrant up` with a couple optional arguments you can spin up local or cloud-based dev instances or cloud-based production instances.
- Used locust with selenium to do load testing against our deployment.



## OPEN SOURCE PROJECT

Inflation ----- 2017 - Present

**Inflation** is a tool that allows you to simply orchestrate and provision a cluster. It is built on top of **Rambo** (listed below). With Inflation you can with a couple commands you can create a variable number of virtual machines. This cluster can either be local on a powerful enough computer or created with cloud based virtual machines such as Digital Ocean Droplets or AWS EC2 instances. The cluster is easily configured to have subsets of VMs created and provisioned differently for specialized purposes, so subsets may have varying hardware and configured with different software, all networked together for powerful parallel computing.

The work is open source and can be found here:  
<https://github.com/terminal-labs/inflation>

### Technologies used:

Python 2 & 3, Bash, DigitalOcean, AWS EC2, Vagrant, Rambo, VirtualBox, LXC, SaltStack, Hadoop, HDFS, Ambari, Spark, Dask, Pandas, Jupyter, Distributed, Ubuntu/Cent/RHEL/Debian/Fedora, Yarn, Terraform, Packer.

### My involvement included:

- Configured inflation to install Hadoop with Hortonworks Ambari.
- Configured inflation to install Hadoop HDFS.
- Wrote Jupyter notebooks that ran on a Hadoop Edge Node to distribute data storage and computation, using Anaconda Dask and Distributed, Pandas, and Hadoop HDFS.
- Used Jupyter's Magic to use HDFS to import data and in situ configuration of Edge Nodes.
- Imported ~100 GB of data from a Digital Ocean Volume onto HDFS and processed it with Python.
- Cycled clusters repeatedly.
- Created a 64 node Hadoop cluster, and a 120 node general cluster.
- Built an automation layer around Packer to build several customized Vagrant boxes. This work can be found at <https://github.com/terminal-labs/vagrant-boxes>. The boxes can be found at <https://app.vagrantup.com/terminal-labs>
- Automated setting of FQDNs, hostnames, and SSH keys.
- Create a minimal Flask server to allow an API for guest machines / minions to communicate with the host / master Salt Server.



## OPEN SOURCE PROJECT

Rambo ----- 2015 - Present

**Rambo** is a tool that allows you to simply provision new virtual machines on any provider and have them be nearly identical. To accomplish this Rambo makes heavy use of Vagrant and it's various plugins for different providers. As a design philosophy, we really want to have development environments that are as similar as possible to production environments. This streamlines development, catches bugs, and helps smooth production releases. So to achieve this, we wanted to automate the provisioning of these environments. For this we used SaltStack. Among other things, SaltStack is great for provisioning machines. With it, we can provision a local or remote instance pretty easily.

The work is open source and can be found here:

<https://github.com/terminal-labs/rambo>

Technologies used:

Python, Bash, Ruby, AWS EC2 and S3, DigitalOcean, Vagrant, VirtualBox, Packer, LXC, SaltStack, Jira, Bitbucket, Git, Mercurial, Postgres, Debian, Buildbot, Bitbucket Pipelines, Apple OS X.

My involvement included:

- Setup a buildbot server for continuous integration and automated testing of Rambo which requires VT-x and VT-d.
- Created virtual machines on AWS EC2, DigitalOcean, VirtualBox, & LXC
- Worked with numerous Vagrant plugins each from different developers and organization with different design philosophies.
- Worked with complex SSH key vendor and authentication systems.
- Maintaining multiple code repositories in Mercurial and Git.
- Smoothed out many inconsistencies between several different cloud providers.
- Created SaltStack configurations with over 50 salt states and custom grains.
- Wrote some custom Vagrant code in Ruby.
- Integrated this project into a cluster with over one hundred virtual machines all running at the same time all managed by SaltStack.



- Worked with LXC on Ubuntu 16.04 and Debian 8 in order to add LXC as a supported provider.
- Worked with Bitbucket Pipelines.
- Created custom Virtualbox and LXC images for use with Rambo.
- Worked with Docker in order to incorporate it as a supported provider.
- Wrote a simple deployment and initiation script in Bash.
- Used Packer to create a few dozen VirtualBox images, in several flavors. Both Debian based and Ubuntu based.
- Publicly hosted VirtualBox images on AWS S3
- Wrote documentation and tested against its instructions.

## **PYTHON DEMONSTRATION PROJECT**

PiCloud/iPhone pi distribution statistics app ----- April 2013 – March 2013

This nice iOS app is a simple demonstration of calling the PiCloud cluster via an iPhone app. From the app, the user can search for an inputted string in several billion digits of the decimal expansion of Pi.

Technologies used:

Ubuntu Server, Apache Cordova, PiCloud, Flot, Apple OS X, Xcode, Python Tornado, Nginx, HTML5/CSS3/JS, jQuery, Mercurial.

My involvement included:

- Used Adobe Photoshop/Illustrator and GIMP to design mock-ups of the app.
- Converted Adobe Photoshop/Illustrator design mock-ups into fully functional interactive HTML5/CSS3/JS and UI features.
- Developed a functional backend server in Python Tornado for receiving/processing AJAX data send from the iPhone app.
- Studied and worked with PiCloud and their API in order to become proficient with their environment.
- Developed an iOS app in Xcode using PhoneGap (aka Cordova) and HTML5/CSS3/JS
- Created a simple HTML5 based iPhone/PhoneGap “mockup viewer” in order to ease and accelerate development. Configured a Nginx server to support the “mockup viewer”.
- Assembled some open source JavaScript modules into a small utility library



that abstracts the underlying JavaScript methods into a more “python-esq” syntax, in order to increase readability.

- Studied multiple JavaScript plotting libraries and ultimately chose Flot, and integrated it into the PhoneGap App.
- For this project we needed to test on physical iOS devices, so I used multiple developer tools from the Apple ecosystem (iOS provisioning portal, app signing, physical device deployment)

## EDUCATION

- 2008  
Bachelor’s Degree (with Honors), Liberal and Integrative Studies Emphasizing Commercial Spaceflight and Astrophysics, University of Illinois at Springfield.
- 2005  
Studied Mechanical/Aerospace Engineering, at the University of Alabama in Huntsville (Honors Program).
- Associate's in Science Degree, Lincoln Land Community College (Honors Program).

## PATENTS ISSUED

- High Specific Impulse Superfluid and Nanotube Propulsion Device, System and Propulsion Method.
  - Patent 8,991,150. 31 Mar. 2015.
- Web Insulation System, Valve for a Web Insulation System, and a Storage Container Using the Web Insulation System.
  - Patent 8,991,636. 31 Mar. 2015.
- Apparatus and Method for Anonymously Presenting Targeted Advertisements and Desirable Media Content in Association with a Virtual Currency.
  - Patent 9,001,979. 7 Apr. 2015.
- Web insulation system, valve for a web insulation system, and a storage container using the web insulation system (continuation).
  - Patent 9,279,540. 8 Mar. 2016.

## AWARDS AND ACCOMPLISHMENTS

- Member of the American Institute of Aeronautics and Astronautics (AIAA) Nuclear & Future Flight Propulsion Technical Committee (NFFPTC).
- Earned the Congressional Award. I received the Bronze, Silver, and Gold

Medals.

- Received the Boy Scouts of America (BSA) Ranger Award, Gold Award, Silver Award, and Bronze Award.
- Completed the BSA's Lifeguard Training program and the BSA's Climbing Instructor training program.
- Studied in the University of Alabama in Huntsville's Honors Program.
- Volunteered at the University of Alabama in Huntsville's Propulsion Research Center.
- Received first place at the 2008 Science Research Symposium held at the University of Illinois at Springfield for "Implementation of a Multi-Platform General Purpose Distributed Computation Grid.
- I hold a Professional Association of Diving Instructors (PADI) Open Water scuba diving certification.

## **VOLUNTEER AND SOCIAL**

I have taught beginner to intermediate students aspects of a diverse range of topics, including:

python development, linux, database systems, virtual machine usage, AWS EC2/DigitalOcean usage, c/c++, algebra, wolfram language, astrophysics, cosmology, relativity, raspberry pi usage, black hole mechanics, finance, economics, computer science, environmental science, space science, spacecraft propulsion, thermal dynamics.

## **ORGANIZATIONS**

- National Space Society
- American Institute of Aeronautics and Astronautics
- Association for Computing Machinery
- Experimental Aircraft Association
- American Physical Society

## PROGRAMS & TOOLS THAT I HAVE EXPERIENCE WITH

### Software Programs:

- Adobe Creative Cloud (Photoshop, Illustrator, InDesign, Premiere)
- Anark Studio
- AutoDesk Product Design Suite (AutoCAD, Inventor, Fusion 360, Showcase)
- GIMP
- Mathcad
- Mathematica
- Solid Edge
- SpaceClaim
- Onshape

### Misc Tools and Libraries:

- 7-Zip
- Clonezilla
- Docker
- Flot
- Gearman
- Genshi templates
- Django templates
- Git
- GParted
- Grunt
- Jade templates
- Jinja templates
- KVM virtualization
- Lastpass
- TeamViewer
- LogMeIn
- Less
- Mercurial
- OpenVPN
- PhantomJS
- Rsync
- SaltStack
- Selenium
- SpinRite
- SSH
- Subversion

### Python Frameworks, Platforms and Modules:

- Lektor
- Locust
- Anaconda
- BeautifulSoup
- Ctypes
- Django
- Django-CMS
- Django-htmlmin
- Falcon
- Flask
- Flask-security
- Lazythumbs
- NumPy
- PiCloud
- PIL
- PyInstaller
- PySide
- Python Mezzanine
- Python Pyramid
- Python Tornado
- SQLAlchemy

### Operating Systems:

- Apple OS X
- CentOS
- Damn Small Linux
- Debian
- Fedora
- FreeBSD
- iOS
- Linux Mint
- Slackware
- Sun Solaris 9
- Ubuntu

### Platforms and Analytics:

- Adobe Analytics
- Adobe DTM
- Brightcove
- Kissmetrics
- New Relic

### Web Apps:

- Bitbucket
- GitHub
- JIRA
- Dropbox
- Google Docs
- Google Drive
- Pivotal Tracker
- Redmine
- Review Board
- VersionOne

### Frameworks:

- AngularJS
- Backbone
- Bootstrap
- Ember
- InK
- jQuery
- Node.js
- PhoneGap (aka Cordova)
- React
- Qt

### Cloud Providers And Datacenters:

- Amazon Web Services
- Rackspace
- DigitalOcean
- OnRamp
- m5hosting

### Amazon Web Services:

- S3
- EC2
- CLI Tools
- Route 53
- CloudFront
- Certificate Manager

### Database Systems:

- CouchDB



- Syncthing
- TrueCrypt
- UnixUtils
- Vagrant
- VirtualBox
- Packer
- Virtualenv
- VMWare
- Xen
- LXC
- Buildbot
- Terraform

- Omniture
- OpenStack
- Outbrain
- Qualtrics

- Elasticsearch
- Memcached
- MongoDB
- MySQL
- PostgreSQL
- SQLite

Servers:

- Nginx
- LAMP
- Lighttpd

Compilers:

- Dev C++
- F95
- g95
- GCC
- Shed Skin
- Tiny C Compiler
- XCode
- Go