



Hosted python functions for humans.

Nicholas Petosa

Motivation

Python is the fastest growing major programming language [1]. Its unprecedented growth has been attributed to python's massive repository of scientific and statistical libraries such as scipy, pandas, numpy, keras, and tensorflow [2]. Python's machine learning libraries in particular enjoy wide usage and active development. Yet in many spheres of development, specifically mobile development which concerns either Java or Swift, python and its wildly popular libraries are inaccessible. The current way of integrating with python cross-platform is to create a Flask or Django web server to host some piece of functionality from a library. This requires resources to invest in learning a web framework, hosting, and application security. **The solution is skython, a web service for easily building, hosting, sharing, and executing python functions.** Skython is a web application that allows developers to write, test, and upload python functions to a collection in the cloud called skybox. Users can then query functions in the skybox from popular utilities like AJAX, requests, and postman. With skython, developers can seamlessly tie python scripts into any code with an HTTP request to the script's endpoint, meaning it can provide the entire backend for mobile or IoT devices. skython's simplicity makes it ideal for rapid prototyping and development – there are no frameworks to learn, just straight python. This solution can scale from weekend hackathon projects to serverless computation for companies.

Related Work

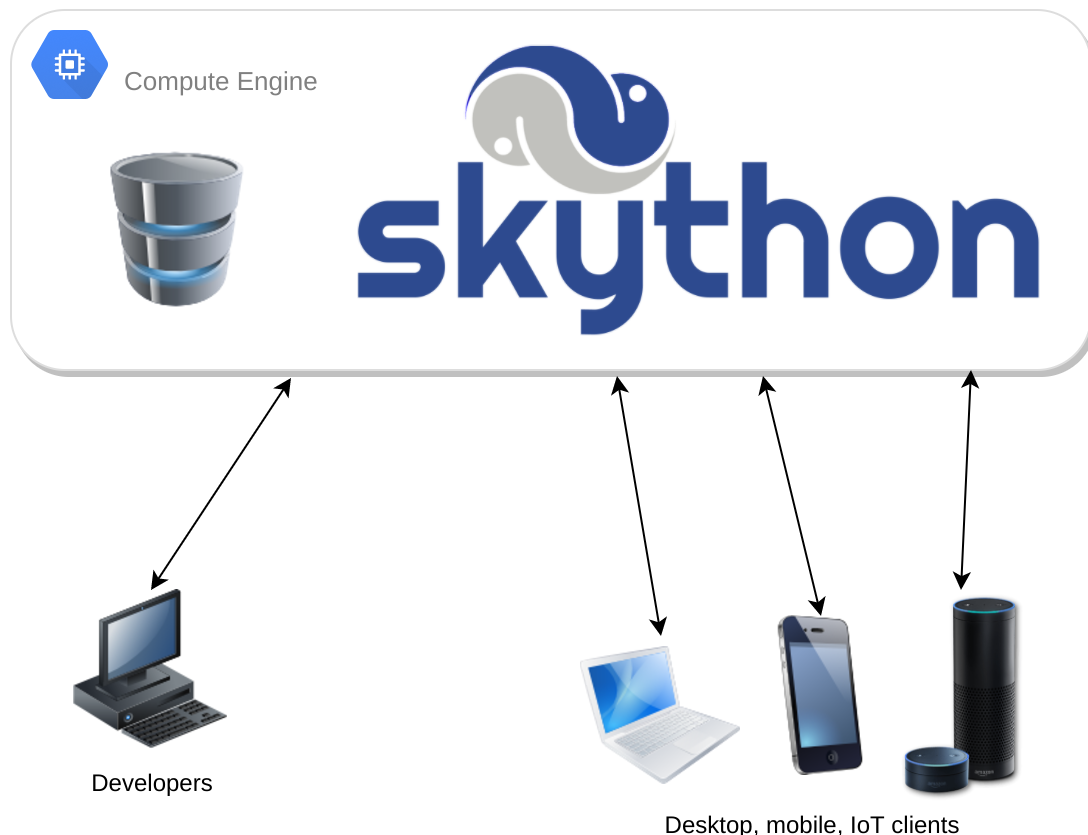
Serverless code/NoOps is not a new concept [3]. There are several popular services which can host anonymous functions with support for many programming languages. The biggest players are AWS Lambda, Google Cloud Functions, and Microsoft Azure Functions [4]. Many of these services require developers to adhere to a code template, forcing new and cumbersome programming models on developers [5]. Additionally, these services tend to be overly centralized and tied closely to their proprietary software environments; for example, AWS Lambda is designed to integrate easily only with Amazon's product ecosystem. Where skython differs is in its openness and simplicity. skython is open source, meaning a developer or company can run a skython server locally for their internal development teams – for free. Developers can share functions in their skybox over LAN, completely isolated from the internet, meaning code is secure. There are no complicated code templates to fill out, and no files to upload from your computer – the script as you write it is exactly what will be executed.

Another avenue in which skython stands out from competitors is its social aspect. The popular code hosting and version control site Github is by and large the leader in "social coding." [6] In skython, functions can be published to skybox, making them available to all users connecting to that skython instance. Developers using the web UI can browse the source code of these python functions as they would browse repositories on Github. Where skython stands out is that developers then have the ability to call these community scripts directly from their projects – effectively crowdsourcing the NoOps movement.

Features of Proposed Work

- **A web UI for creating and debugging python functions.** The development environment should have proper syntax highlighting and be familiar to developers. Each function will have the ability to request parameters (provided by user via POST body) and to return a value (provided to user via HTTP response). These functions will have access to a wide array of pre-installed python libraries including pandas, numpy, scipy, tensorflow, keras, BeautifulSoup, and requests. Additionally, users can leverage the skython meta library, which enables them to call simple store(key, value) and load(key) functions to persist data across invocations. Persistent data will be user AND function specific. Users can apply custom tags to their functions (“ML”, “scraper”, “NLP” for example) and then choose to publish their functions to the community or keep them private.
- **Endpoints for querying functions.** The point of hosting functions on skython is so that users can call them in the first place. Developers will be able to specify the names of endpoints that clients must query in order to run and retrieve the results of a function.
- **Skybox – a library of community-built skython functions.** Part of the appeal of skython is that developers can view and use scripts created by other users. On the left-hand side of the web UI will be skybox, a searchable listing of all public functions. Skybox is indexed by function name, description, and tags, so developers can easily query for specific functionality. Clicking on a listing will bring up that function’s published code and expected argument/return values. From here, a developer can clone a community function for personal use.

System Architecture

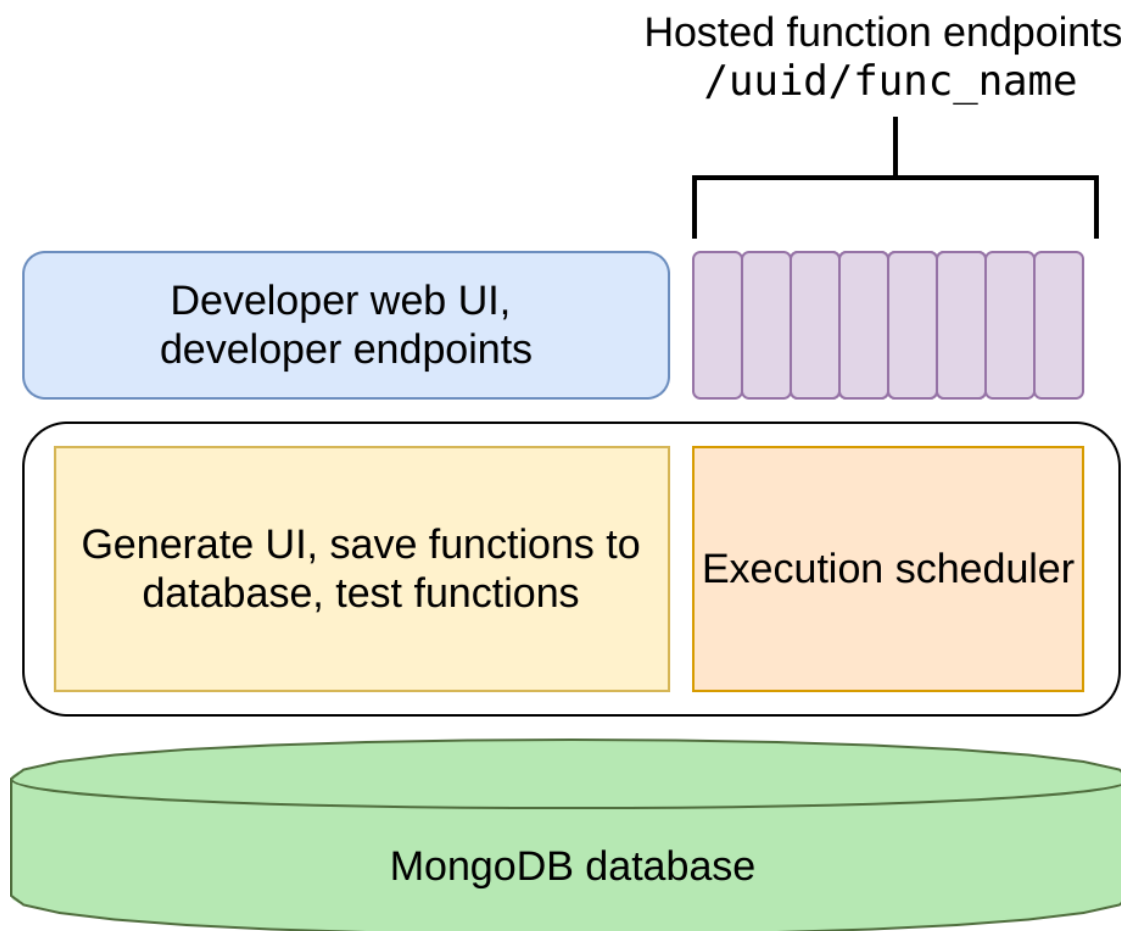


Central server and database. The official instance of skython will run on Google Cloud Platform's Compute Engine, although it can be hosted on any server box or cloud provider. An instance of MongoDB will be running on the same machine as well.

Interaction with developer clients. Developers will be interacting with skython's web UI in order to build new functions and upload them. The web UI will also be populated with community functions in skybox which the developer can query for.

Desktop, mobile, and IoT clients. Applications running on any other platform with the ability to make HTTP requests can query endpoints created by developers in order to run functions with custom parameters and receive a return value through an HTTP response body.

Application Architecture



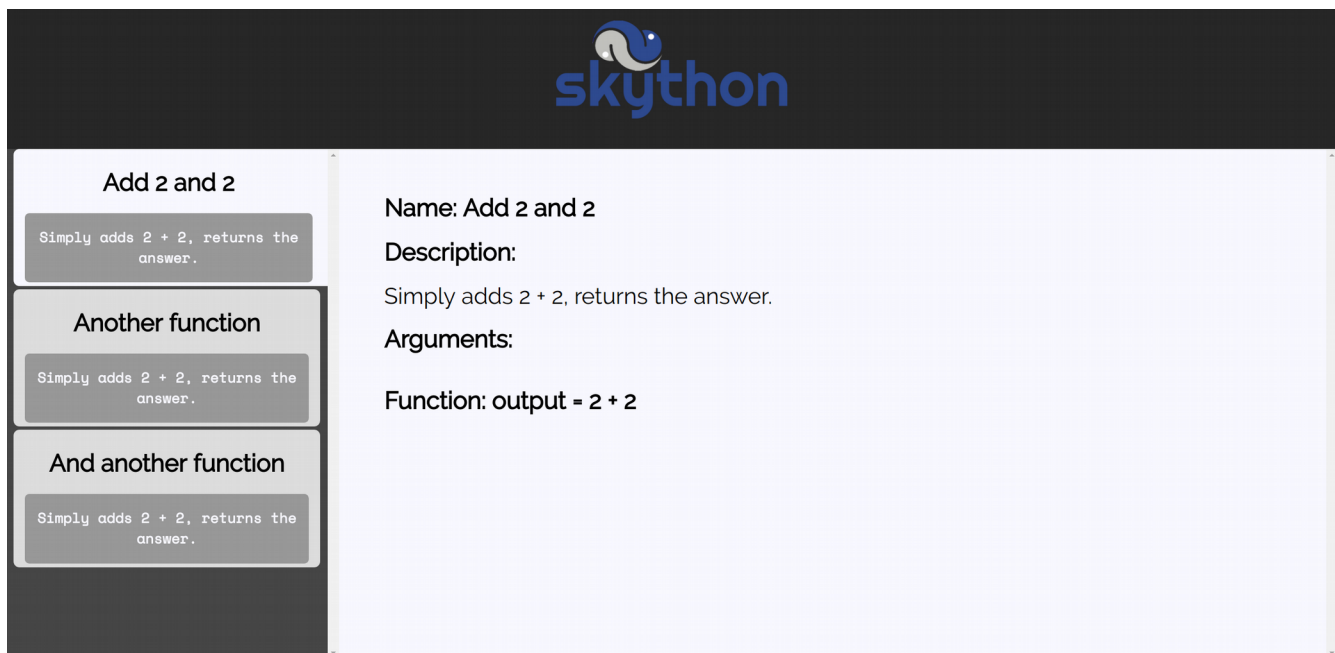
Developer and client endpoints. At the highest level, skython exposes endpoints to developers and users. The web UI calls the developer endpoints needed for uploading functions and loading community functions from skybox. Client endpoints invoke functions created by developers.

Inside the skython application. Once a developer uploads a function or queries skybox, the internals of skython will create or read entries from the underlying MongoDB database. Once a user calls an endpoint, his requested function invocation is queued for execution on the server.

The underlying database. Skython is backed by a Mongo database, which is responsible for storing user data as well as function information (source code, arguments required...). The function objects stored in MongoDB will have the following schema:

Key	Value	Type
(1) ObjectId("59c2ec33184b3f3e1dc68695")	{ 5 fields }	Object
_id	ObjectId("59c2ec33184b3f3e1dc68695")	ObjectId
args	{ 0 fields }	Object
name	Add 2 and 2	String
function	output = 2 + 2	String
description	Simply adds 2 + 2, returns the answer.	String

Proposed Application Layout



Project Deliverables

- Working demo
- Source code
- Documentation and Usage instructions
- Report

Foreseen Risks

- One risk that needs to be addressed in the design of this system is concurrency. Multiple users will be querying functions simultaneously, so the endpoints, scheduler, and database must be built in such a way to allow concurrent access.
- Security is a large concern. For example, certain modules need to be disabled in python to prevent malicious developers from accessing and modifying server files. For example, the OS module needs to be banned.

Planned Schedule

	25- Sept	2- Oct	9- Oct	16- Oct	23- Oct	30- Oct	6- Nov	13- Nov	20- Nov	27- Nov
Create endpoints and their functionality										
Scheduler for function invocations, database										
Accounts										
Function persistent data										
Web UI										
Testing										
Documentation										

Future Work

- There is currently no way of gating who can run a developer's functions. This would mean allowing developers to distribute API keys for their function calls, which is currently out of scope.
- In order for any skython instance to be sustainable, it should be possible to monetize. This could mean introducing a method of charging developers for creation and usage of their functions, a practice used widely by API services.
- Further research needs to be done to prevent malicious executable code from running on the server. This would mean researching and banning python modules and practices which can modify and read server files without permission.

References

- [1] <https://stackoverflow.blog/2017/09/06/incredible-growth-python/>
- [2] <https://medium.com/activewizards-machine-learning-company/top-15-python-libraries-for-data-science-in-in-2017-ab61b4f9b4a7>
- [3] <http://searchaws.techtarget.com/essentialguide/AWS-Lambda-architecture-brings-serverless-to-enterprise-cloud>
- [4] <http://searchcloudcomputing.techtarget.com/news/450296467/Public-cloud-vendors-jump-on-serverless-computing-bandwagon>
- [5] <http://docs.aws.amazon.com/lambda/latest/dg/python-programming-model-handler-types.html>
- [6] <https://devops.com/social-coding/>