Automate your tasks with Python and publish with Chat Apps

That’s the way to build a Virtual Assistant for your team, the way to boost productivity

Anton Chernikov

www.skillpub.org

chernikovanv@gmail.com

Why should experts work with certain routine tasks?

Because they have:

access to applications

access to databases

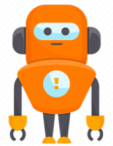
network access

API access to API's

SKILLS Knowledge of how to interpret data and what to do

in a certain subject areas

What if experts create a virtual assistant for themselves? +



- Equip his workplace as their own
- Transfer their routine tasks to the assistant

This will **free up experts time** for new not routine tasks and will **save time** of tasks processing.

Although there are **barriers** — programming a virtual assistant in itself requires additional expertise, time and cost.

What are the routine of operating activities for IT specialists?

- getting / changing attributes of an entity
- collection / analysis of logs
- configuring systems
- incident response

and much more

First step of automation

At first, routine actions are performed **manually** using:

- web interfaces of systems
- client applications of systems
- SQL clients
- API clients
- SSH clients

Then, to save time, employees **automate** their tasks using a high-level (automation-friendly) language like python™

However it is **still a specific employee** who runs and maintains the automation scripts.

What are the options to exclude an expert from routine processes?

1. Integrate routine into interfaces available to other employees (for example, CRM)
    - analysts and developers
    - time and money
    - poor flexibility and speed of implementation of changes
    - one interface is not enough

2. Give access to your interfaces (access to the web interface, databases, API, etc.)
    - dangerous 
    - requires training
    - routine did not go away, we shifted it to others

3. Pass your automation scripts to others
    - dangerous 
    - they need network access, logins/passwords
    - they need an environment for the script to work
    - they need competence

4. Develop a single interface for systems that are managed by a group of experts (API, web, chatbot) and make it available to other employees
    - requires additional expertise from experts
    - and time for integrating scripts into interfaces

we chose this option and removed the cons

Skillpub

We developed a platform for publishing automation scripts within a company and published that platform

How to use it?

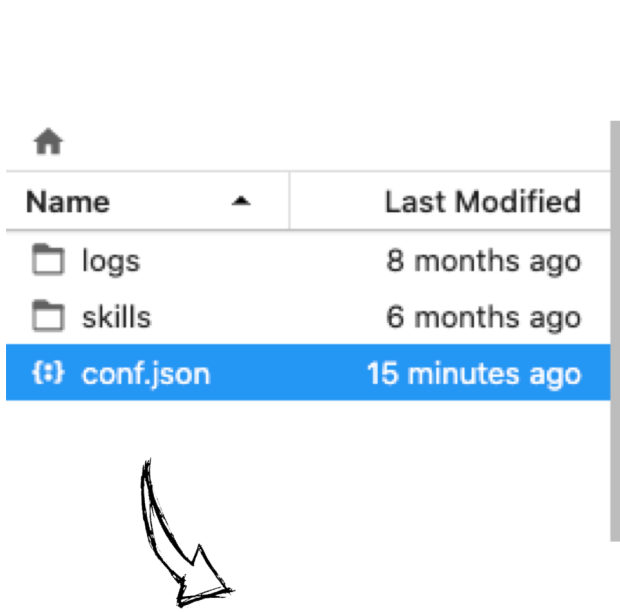



API

Let's take a server, make the necessary network access for it, create credentials in the necessary systems, install the platform Skillpub on the server, create a user-bot in the corporate messenger and give it to the platform.

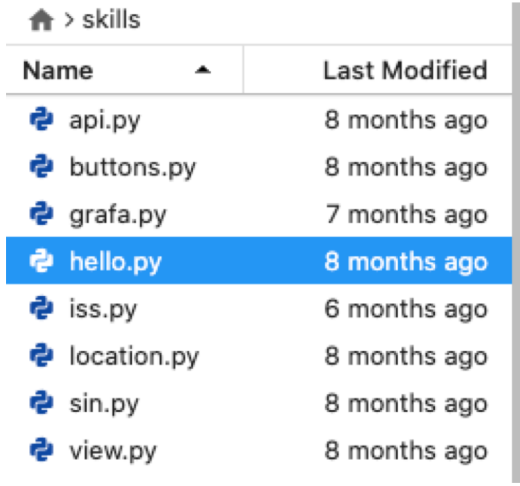
Then the magic ✨ happens! Any* **SKILL** script in a high-level programming language can be used in the platform without adaptation. The platform itself will integrate script with the messenger and JSON API, provide access control, monitoring and logging, only the author of the script determines who can use the **SKILL**.

Let's see how it works with examples!




```
{
    "users": {
        "james": {"channels": {"slack" : "james", "telegram": 1001}}
    },

    "channels": {
        "telegram": {"token": "123456789:QWER_1234567890qwertyu-iopasdfghjklz"},
        "slack": {"token": "xoxb-qwertyuiopa-123456789012-asdfghjklzxcvbnm123456"}
    }
}
```




```
'''the "Hello!" skill'''

print("Hello from Outer Space! :wave:")
```

Real example (telecom) – getting a subscriber profile from the Subscriber Profile Repository

spr.py

```python
from http.client import HTTPConnection
from lxml import etree
import re

argv = ' '.join(sys.argv)

match = re.search("\\b(\\d{11})\\b", argv)
if match is None:
    print("нужен MSISDN абонента")
    sys.exit()

MSISDN = match.group()

spr = HTTPConnection("10.10.10.10:8787", timeout=5)
spr.request('GET', '/rs/msr/sub/msisdn/' + MSISDN)
xml_data = spr.getresponse().read()
spr.close()

data = etree.tostring(etree.fromstring(xml_data), pretty_print=True).decode()
print(data)
```

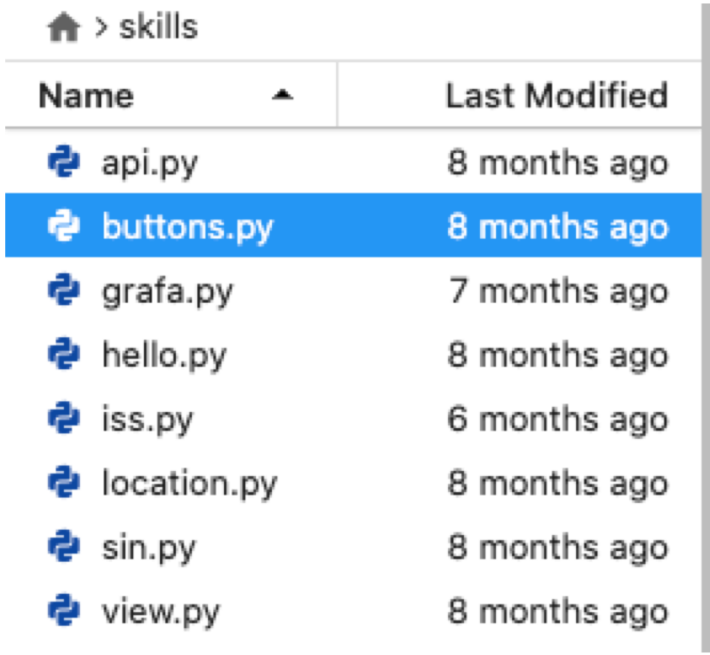
Черников Антон 5:25 PM

spr 79955020016

TelcoHelper APP 5:25 PM

```
<subscriber>
  <field name="MSISDN">79955020016</field>
  <field name="IMSI">250621000399521</field>
  <field name="ContractId">5-29LFTM4R</field>
  <field name="Entitlement">si0_rg0</field>
  <field name="Entitlement">si100_rg1</field>
  <field name="Entitlement">si180_rg1</field>
  <field name="BillingDay">0</field>
  <field name="Entitlement">si10100_rg1</field>
  <field name="Entitlement">si120_rg1</field>
  <field name="Entitlement">si140_rg1</field>
  <field name="Entitlement">si130_rg1</field>
  <field name="Product">20000000000</field>
  <field name="Entitlement">si110_rg1</field>
</subscriber>
```

More features! Data input and buttons

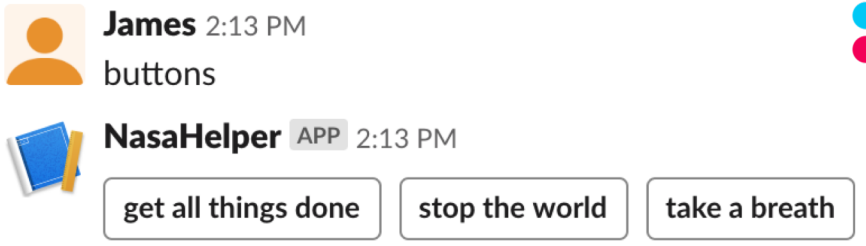



```
'''shows the buttons'''

print({'buttons':['get all things done', 'stop the world', 'take a breath']})

choice = input('your choice?')

if len(choice) > 0:
    print('"'+choice+'"' + ' is great choice!')
else:
    print('not to choose is also a choice')
```



More features! Displaying images

> skills

Name	Last Modified
api.py	8 months ago
buttons.py	8 months ago
grafa.py	7 months ago
hello.py	8 months ago
iss.py	a minute ago
location.py	8 months ago
sin.py	8 months ago
view.py	8 months ago

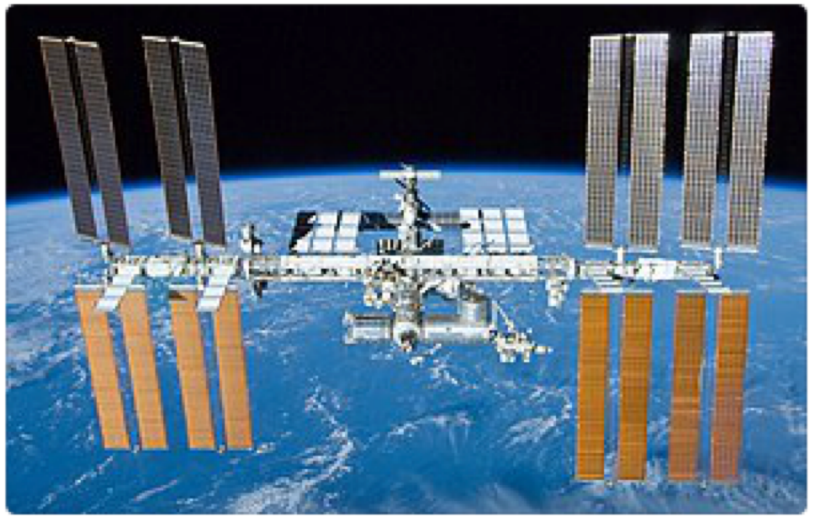
```
1  from urllib.request import urlopen
2  from PIL import Image
3
4  wiki_page = urlopen("https://en.wikipedia.org/wiki/International_Space_Station").read().decode()
5  first_image = urlopen('https:'+re.findall('src="(//upload.wikimedia.org/wikipedia.+?\.jpg)"',wiki_page)[0])
6  image = Image.open(first_image)
7  image.show()
```

James 3:45 PM

view

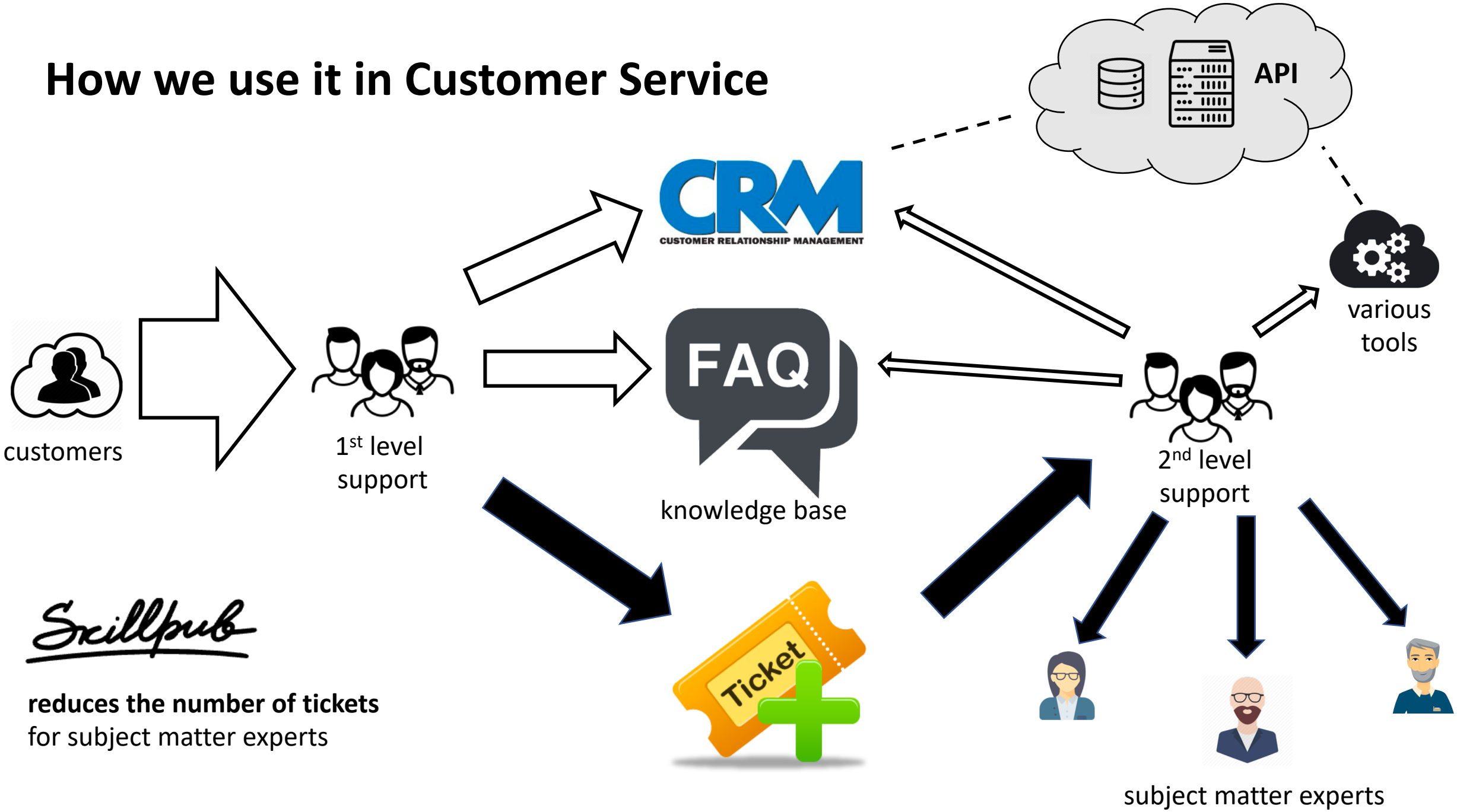
NasaHelper APP 3:46 PM

image.png



More features!

How we use it in Customer Service



Skillpub

reduces the number of tickets
for subject matter experts

More use cases:

- to fetch diagnostic information from multiple data sources into Chat App, gather metrics/logs and analyse them
- to take action in case of incident right from a Chat App, rerouting users requests, server rebooting, launching new instances, and many other actions
- to give easy access via Chat App to APIs for team members and other colleagues
- to provide analytical reports by the request from Chat App, reports with graphs, images, tables, files, etc.

Python community, thank you!

Anton Chernikov

www.skillpub.org

chernikovanv@gmail.com