

A close-up photograph of a computer keyboard. The central focus is a large, bright blue key with the word "Deploy" printed in white, sans-serif font. To the left, several grey keys are visible, including one with a double quote and comma symbol. The lighting is soft, highlighting the texture of the plastic keys.

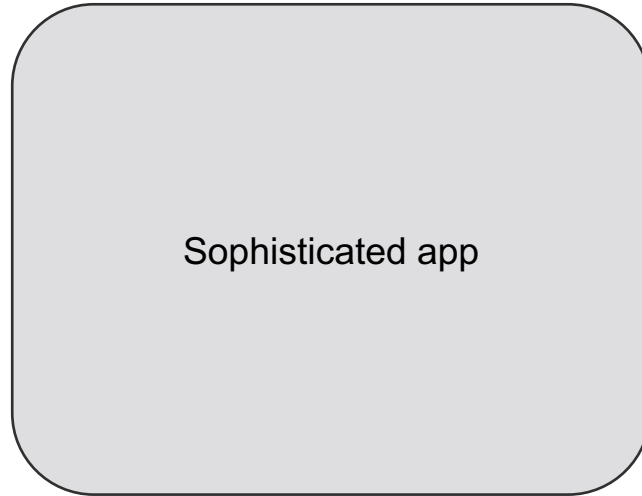
RuFaS Deployment Overview

Pooya Hekmati

November 4, 2022

sh2235@cornell.edu

How good is an app in isolation?



Introducing interfaces

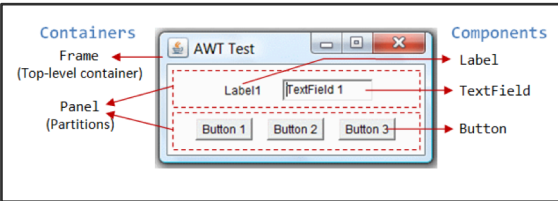


GUI and CLI

```
Administrator: cmd.exe - Shortcut
D:\>ping google.com

Pinging google.com [74.125.236.73] with 32 bytes of data:
Reply from 74.125.236.73: bytes=32 time=83ms TTL=52
Reply from 74.125.236.73: bytes=32 time=78ms TTL=52
Reply from 74.125.236.73: bytes=32 time=109ms TTL=52
Reply from 74.125.236.73: bytes=32 time=107ms TTL=52

Ping statistics for 74.125.236.73:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 78ms, Maximum = 109ms, Average = 94ms
D:\>
```



CLI

GUI

Sophisticated app

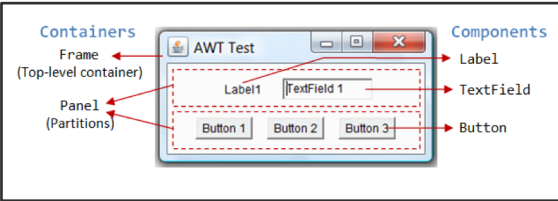
GUI: Graphical User Interface
 CLI: Command Line Interface

What about other apps?

```
Administrator: cmd.exe - Shortcut
D:\>ping google.com

Pinging google.com [74.125.236.73] with 32 bytes of data:
Reply from 74.125.236.73: bytes=32 time=83ms TTL=52
Reply from 74.125.236.73: bytes=32 time=78ms TTL=52
Reply from 74.125.236.73: bytes=32 time=109ms TTL=52
Reply from 74.125.236.73: bytes=32 time=107ms TTL=52

Ping statistics for 74.125.236.73:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 78ms, Maximum = 109ms, Average = 94ms
D:\>
```



CLI

GUI

Sophisticated app

Another
Sophisticated app

GUI: Graphical User Interface

CLI: Command Line Interface

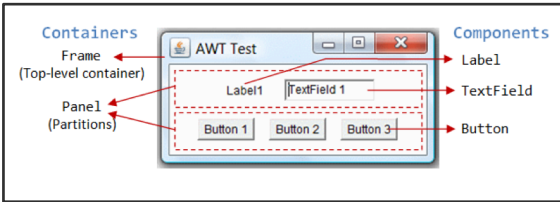
Introducing APIs

```

Administrator: cmd.exe - Shortcut
D:\>ping google.com

Pinging google.com [74.125.236.73] with 32 bytes of data:
Reply from 74.125.236.73: bytes=32 time=83ms TTL=52
Reply from 74.125.236.73: bytes=32 time=78ms TTL=52
Reply from 74.125.236.73: bytes=32 time=109ms TTL=52
Reply from 74.125.236.73: bytes=32 time=107ms TTL=52

Ping statistics for 74.125.236.73:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 78ms, Maximum = 109ms, Average = 94ms
D:\>
  
```



CLI

GUI

Sophisticated app

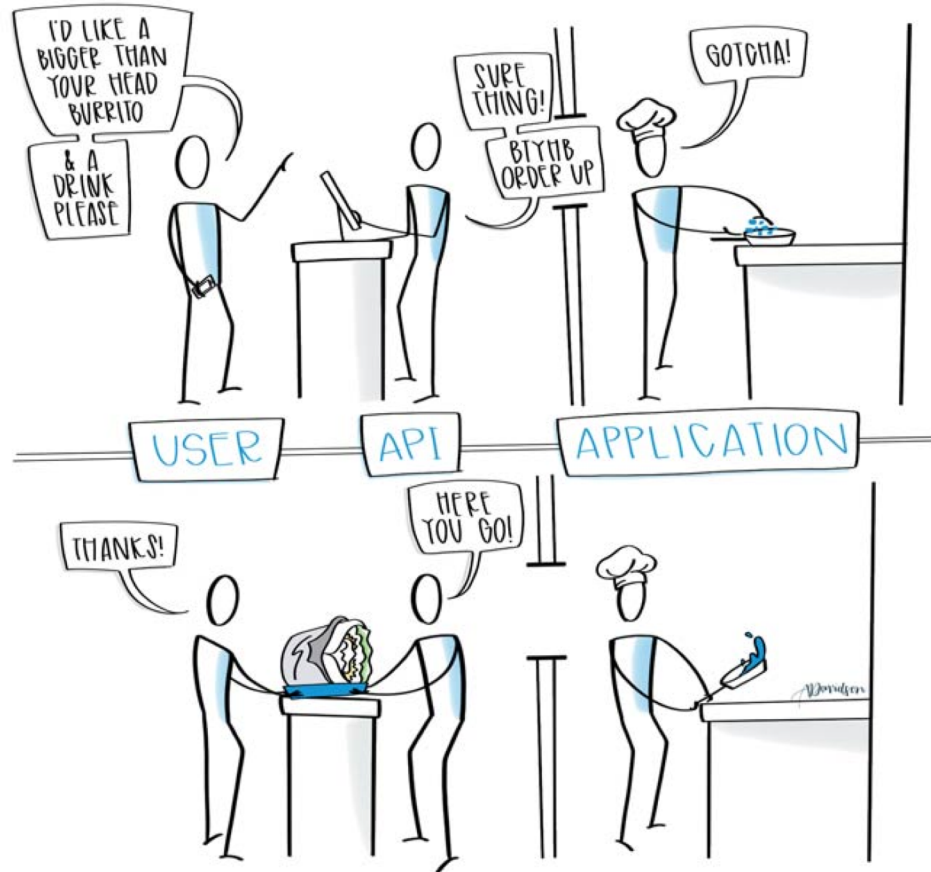
<https://api.rufas.cornell.edu/simulate>

API

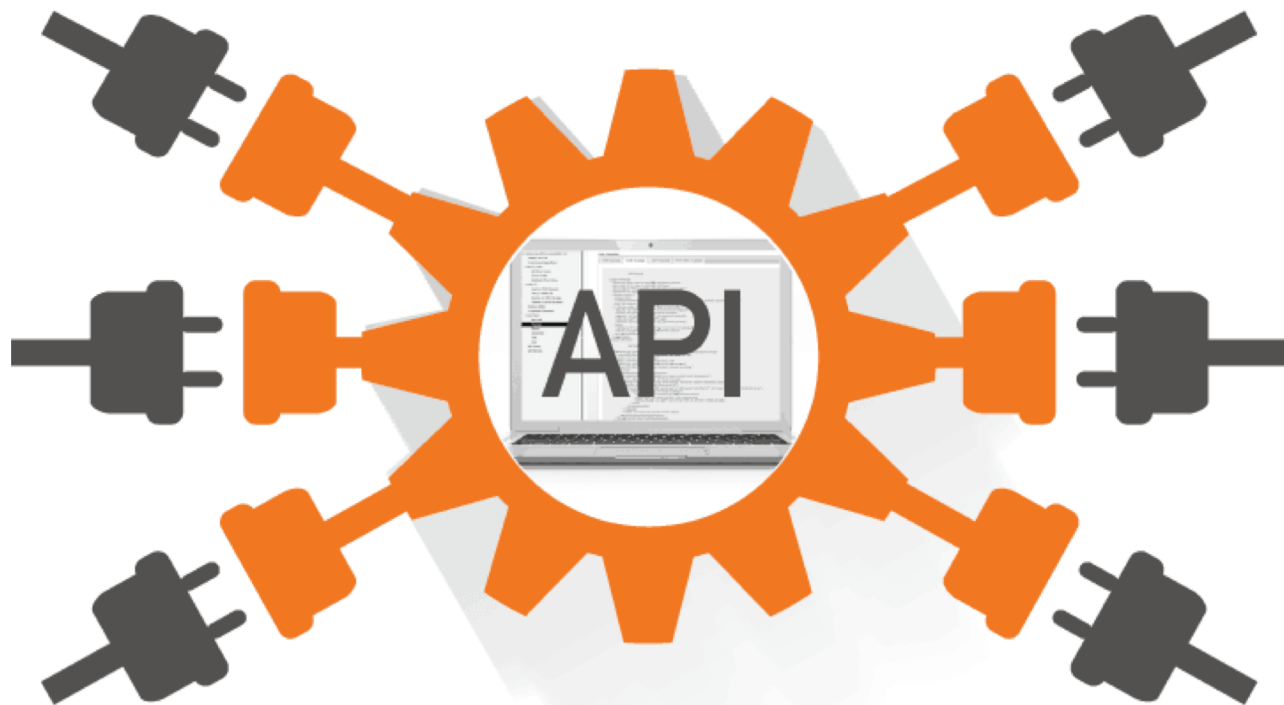
Another Sophisticated app

GUI: Graphical User Interface
 CLI: Command Line Interface
 API: Application Programming Interface

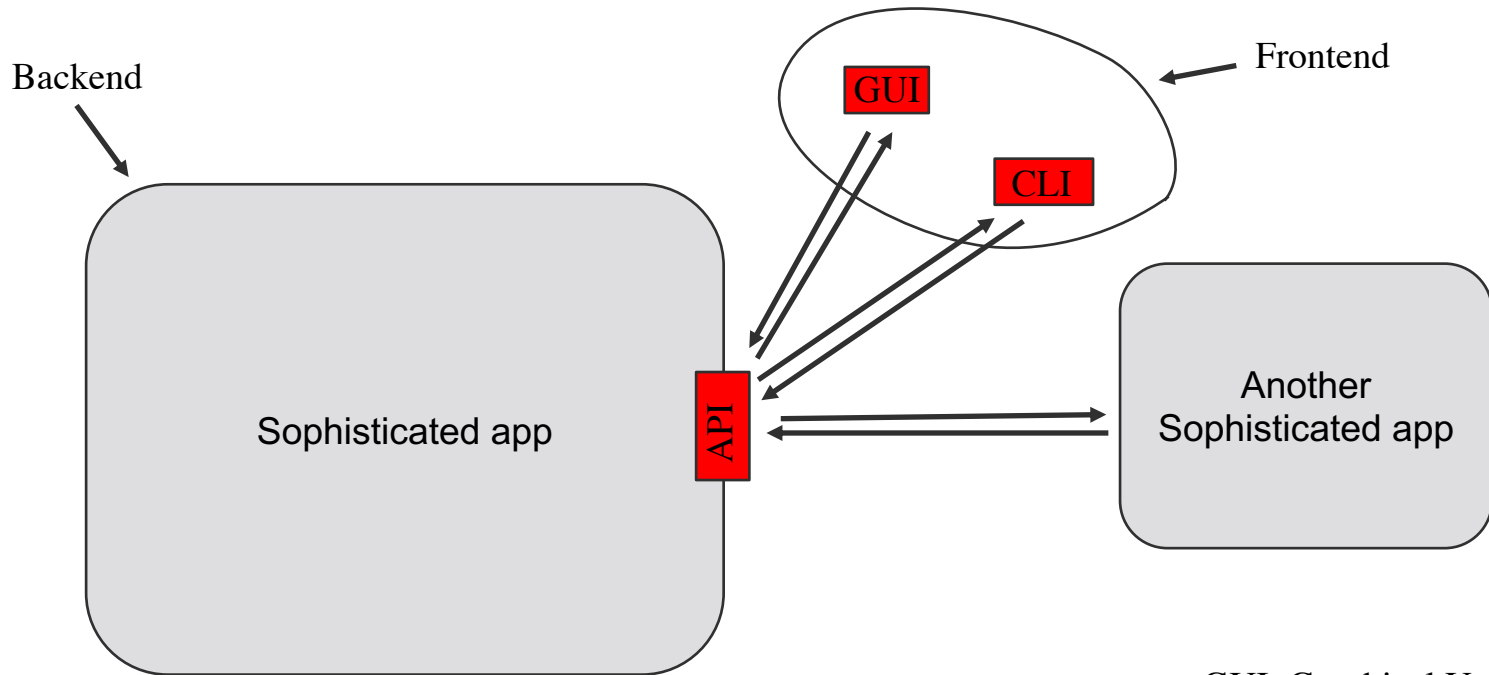
WHAT IS AN API?



APIs make connections better



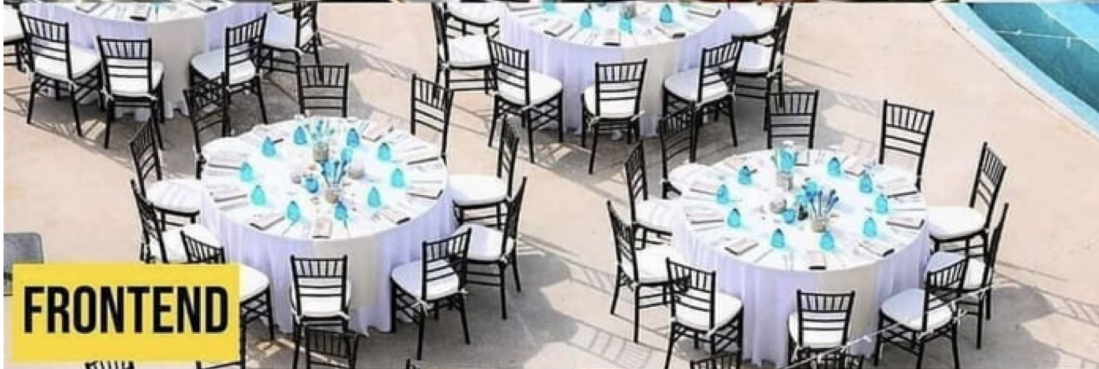
Let's reduce duplications



GUI: Graphical User Interface
CLI: Command Line Interface
API: Application Programming Interface



BACKEND



FRONTEND



API's

Let's standardize things



Type A

- ✦ mainly used in the USA, Canada, Mexico & Japan
- ✦ 2 pins
- ✦ not grounded
- ✦ 15 A
- ✦ almost always 100 - 127 V
- ✦ socket compatible with plug type A



Type B

- ✦ mainly used in the USA, Canada & Mexico
- ✦ 3 pins
- ✦ grounded
- ✦ 15 A
- ✦ almost always 100 - 127 V
- ✦ socket compatible with plug types A & B



Type C

- ✦ commonly used in Europe, South America & Asia
- ✦ 2 pins
- ✦ not grounded
- ✦ 2.5 A, 10 A & 16 A
- ✦ almost always 220 - 240 V
- ✦ socket compatible with plug type C

REST APIs

“A REST API (also known as RESTful API) is an application programming interface (API or web API) that conforms to the constraints of REST architectural style and allows for interaction with RESTful web services. REST stands for representational state transfer and was created by computer scientist Roy Fielding.”

<https://www.redhat.com/en/topics/api/what-is-a-rest-api>

REST Verb	Action	Success	Failure
GET	Fetches a record or set of resources from the server	200	404
OPTIONS	Fetches all available REST operations	200	–
POST	Creates a new set of resources or a resource	201	404, 409
PUT	Updates or replaces the given record	200, 204	404
PATCH	Modifies the given record	200, 204	404
DELETE	Deletes the given resource	200	404

* REST is a set of architectural constraints, not a protocol or a standard. API developers can implement REST in a variety of ways.

What machine runs what?

- Backend and frontend are co-located on the same machine
- A server hosts the backend while user's device (client) runs the frontend



Cloud

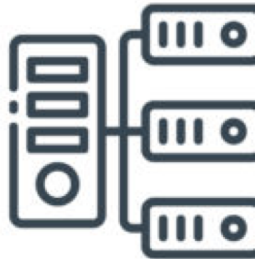
- ✓ **No IT overhead costs**
- ✓ Automatic updates
- ✓ **Effortless maintenance**
- ✓ Scalable
- ✓ **Accessible anywhere**
- ✓ Universal availability of information

act!

vs.

- ✓ **Control over the timing of upgrades and updates**
- ✓ Regulatory compliance
- ✓ **Increased sense of security**
- ✓ Potentially increased speed
- ✓ **Self-managed data recoverability**

On-Premises



Software as a Service (SaaS)



Closing thoughts...

1. Should RuFaS be deployed on cloud or on-premise?
2. If cloud deployment is desired, regulatory compliances should we consider?
3. If on-premise deployment is desired, who owns the hardware? (Cornell, USDA, etc..). Who maintains it?
4. Who pays for the deployment? What constraints do they have?