Designing the MVP…

Entrepreneurship in Technology
Week 06
Dylan Hall
February 8th, 2018

Toto, we’re not in a big company anymore…

1) No defined processes

2) No legacy systems to support

3) No specialists

4) No conventions

5) No hierarchy

6) Oh yeah, no revenue

Startup technology is…

1) Being about the business first
2) Generalizing and not specializing
3) Saving time
4) Speaking up
5) Learning new skills
6) Not being wedded to frameworks

There are many development frameworks…

Test-driven development, agile, lean, etc.

But you simply want to be able to answer:

“Does it work?”

So you can figure out:

“Do people care?”

Mission-based development…

You’ve been selected to start a new country and have been put sent in a ship into the enemy’s territory. The land you would step onto is a small outpost that your enemy country doesn’t value that much. If you’re able to land, you hope to convince the outpost that you’re country is better for them to join your new country. About halfway through the trip, your boat starts leaking, your crew is almost out of food, and if you don’t resolve both issues, you’re all going to die before you reach the outpost.*

*Death in this case means your equity that you built up over 3 years and countless late hours is worth $0 and your beautiful code will sit dormant in a Github repository until someone finally decides to delete your account because no one is using that language anymore and all your fiends just made 6 figures for another straight year and now you have to find another job and now you can’t afford that house you wanted because other people were able to save up money while you were working on a failed startup that your spouse told you was going to fail anyway.

Mission-based development…

Sorry, that was harsh.

There is an urgency in startups.

It should not cause you to panic, but in our example

Food = Cash
Leaks = Time

…you have to manage these things.

What stifles technology productivity…

1) Chasing technology…

2) Administrative overhead…

3) Investors > CEO > Market…

4) Technology uncertainty…

Chasing technology…

Instagram was bought for $1 billion and had 4-5 engineers…

They used technology to their advantage (e.g. AWS)

They prioritized their user and focused on how technology could scale their reach.

On the other hand, blockchain is undergoing the more difficult approach of starting with a technology and trying to create value.

Administrative overhead…

Every additional voice in decision making creates another person to convince when you need to reorient development.

Every additional “tool” creates another thing to interface with and keep up to date.

Every additional communication channel creates one more opportunity for never ending discussion and personal relationship management.

Technology uncertainty…

Don’t see yourself as only knowing a set of languages and things you have been taught in class…

You are a technologist who can learn and apply new skills as evidenced in your classes and at work…

When you find yourself frightened by new technology, figure out how to become comfortable. You can and will learn it…

Your product is a bunch of small steps…

“I need to be able to create an app where people can sell goods, and receive payment for those goods”

“I need to be able to create an app”

THEN

“I need to be able to show goods on the app”

THEN

“I need to allow a user to buy goods on the app”

Your product is a bunch of small steps…

“I need to be able to create an app”

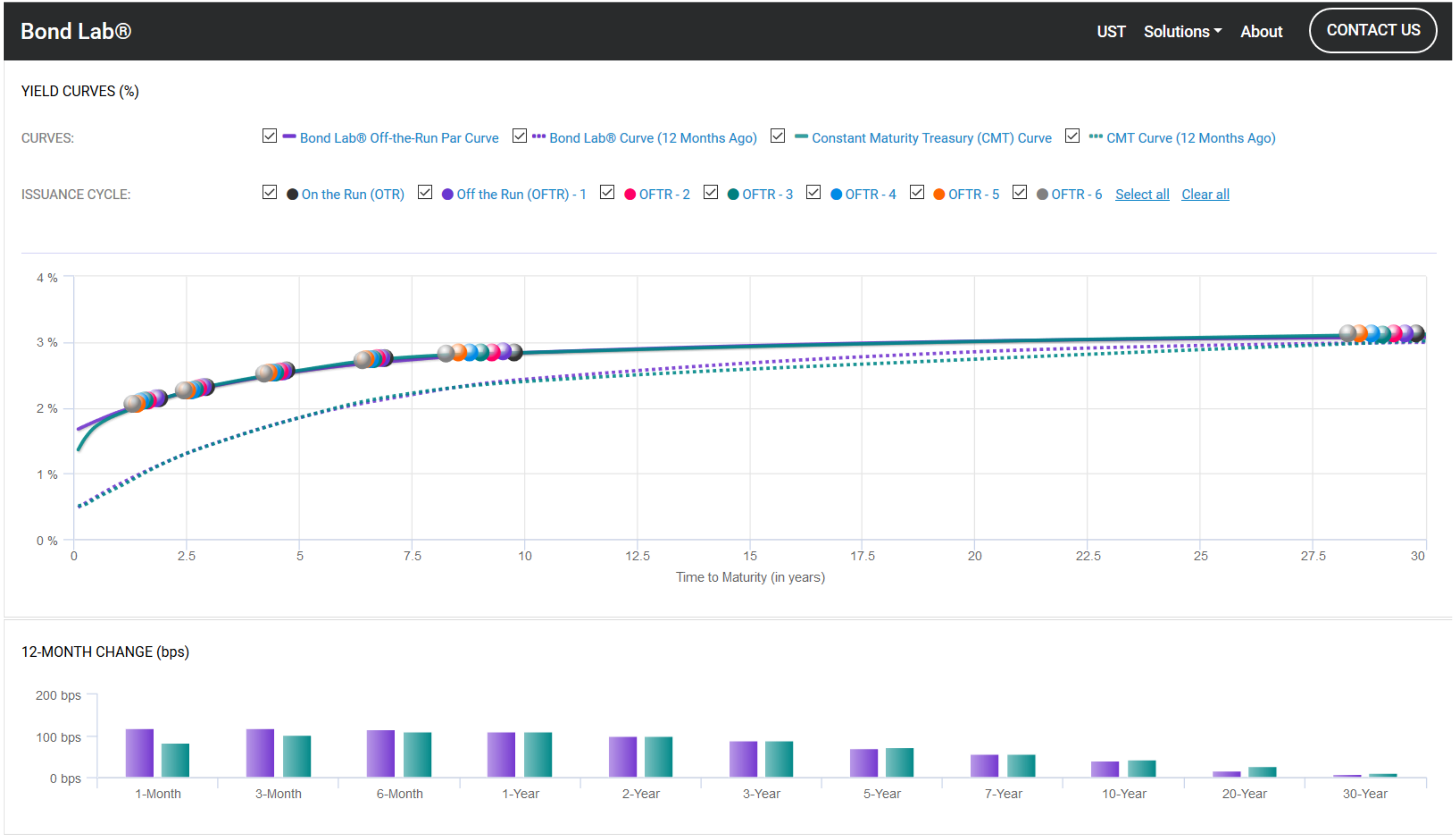
“I need to select a language to develop in.”

THEN

“I need to deploy a Hello World app in that language.”

THEN

“I need to alter my Hello World app to match my business.”



Bond Lab®

UST Solutions ▾ About CONTACT US

YIELD CURVES (%)

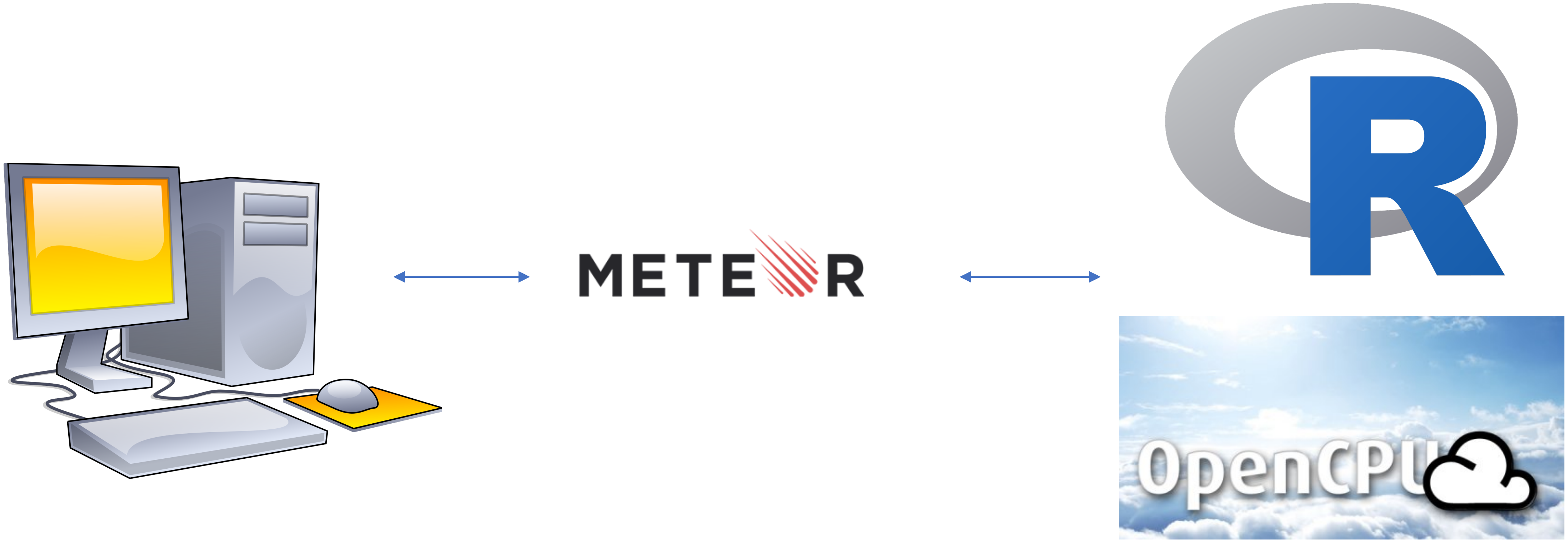
CURVES: ☑ ▬ Bond Lab® Off-the-Run Par Curve ☑ ┅ Bond Lab® Curve (12 Months Ago) ☑ ▬ Constant Maturity Treasury (CMT) Curve ☑ ┅ CMT Curve (12 Months Ago)

ISSUANCE CYCLE: ☑ ● On the Run (OTR) ☑ ● Off the Run (OFTR) - 1 ☑ ● OFTR - 2 ☑ ● OFTR - 3 ☑ ● OFTR - 4 ☑ ● OFTR - 5 ☑ ● OFTR - 6 Select all Clear all

12-MONTH CHANGE (bps)

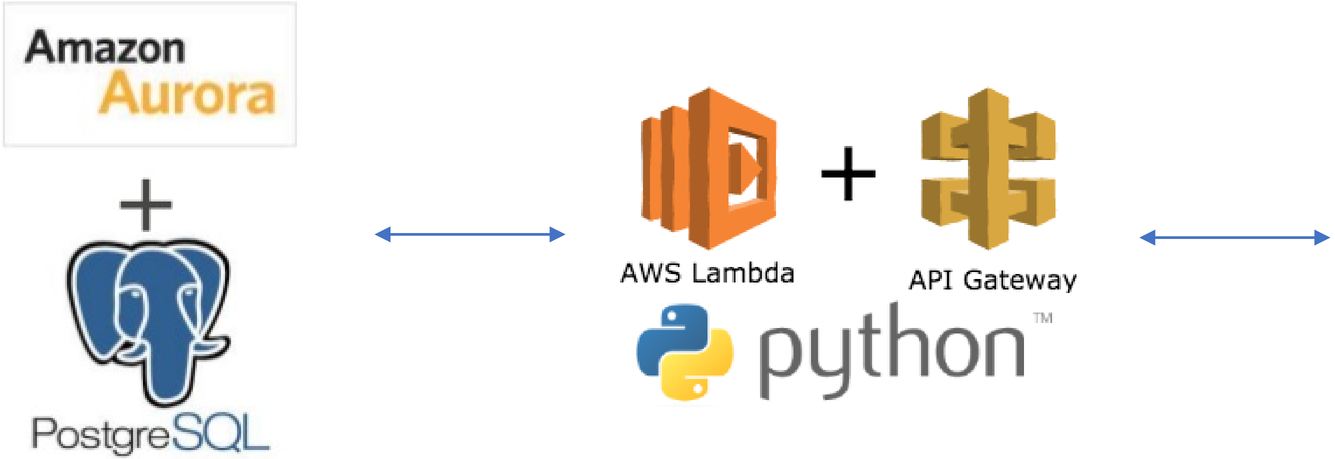
Step 1

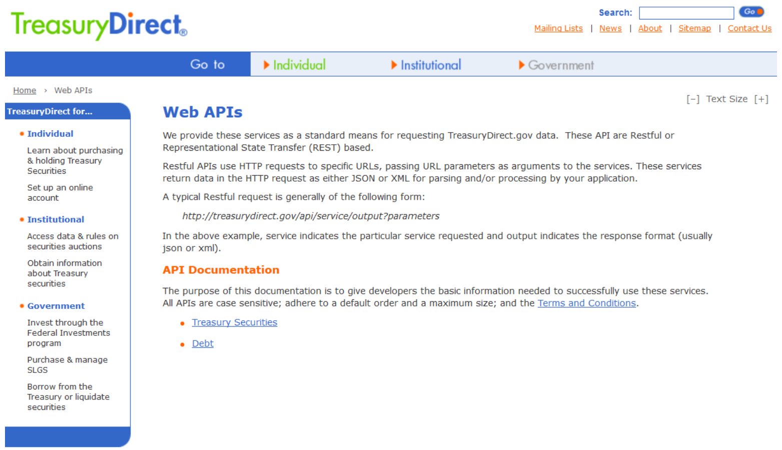
Requirements: I need to make a package written in R accessible to a web application



Step 2

Requirements: I need to pull daily US Treasury data




TreasuryDirect®
Search:
Go
Mailing Lists | News | About | Sitemap | Contact Us
Go to
Individual
Institutional
Government
Home › Web APIs
[–] Text Size [+]
TreasuryDirect for...
Individual
Learn about purchasing & holding Treasury Securities
Set up an online account
Institutional
Access data & rules on securities auctions
Obtain information about Treasury securities
Government
Invest through the Federal Investments program
Purchase & manage SLGS
Borrow from the Treasury or liquidate securities

Web APIs

We provide these services as a standard means for requesting TreasuryDirect.gov data. These API are Restful or Representational State Transfer (REST) based.

Restful APIs use HTTP requests to specific URLs, passing URL parameters as arguments to the services. These services return data in the HTTP request as either JSON or XML for parsing and/or processing by your application.

A typical Restful request is generally of the following form:

http://treasurydirect.gov/api/service/output?parameters

In the above example, service indicates the particular service requested and output indicates the response format (usually json or xml).

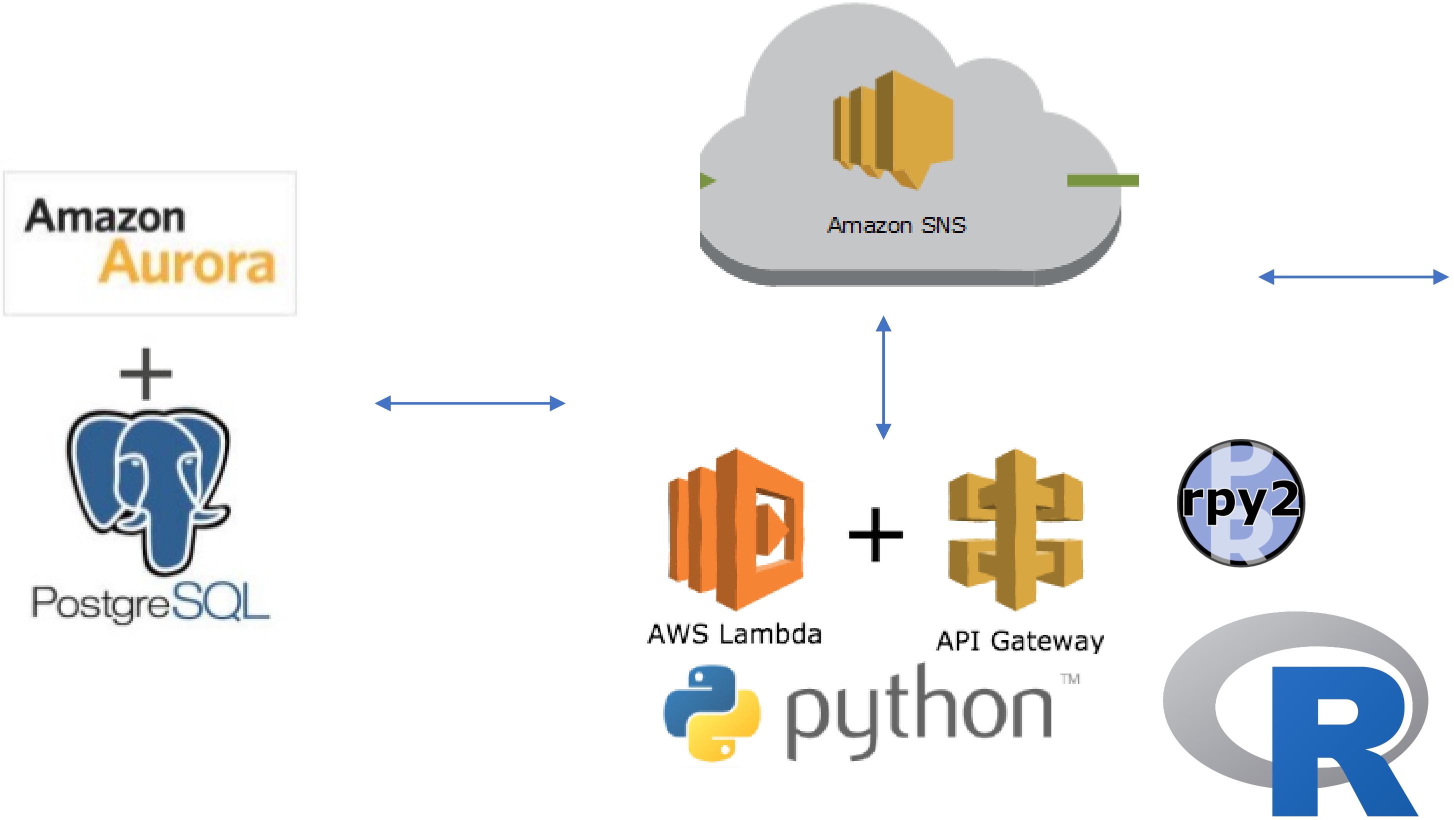
API Documentation

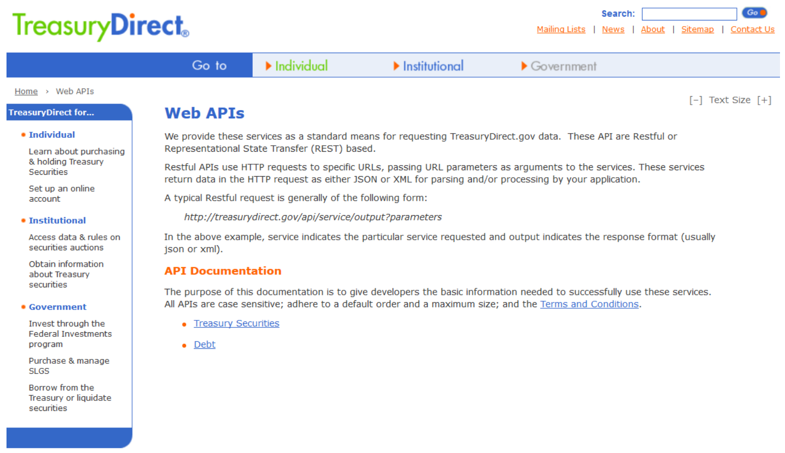
The purpose of this documentation is to give developers the basic information needed to successfully use these services. All APIs are case sensitive; adhere to a default order and a maximum size; and the Terms and Conditions.

- Treasury Securities
- Debt

Step 3

Requirements: I need to automate and perform analysis on many securities





TreasuryDirect for...

- **Individual**
  - Learn about purchasing & holding Treasury Securities
  - Set up an online account
- **Institutional**
  - Access data & rules on securities auctions
  - Obtain information about Treasury securities
- **Government**
  - Invest through the Federal Investments program
  - Purchase & manage SLGS
  - Borrow from the Treasury or liquidate securities

Web APIs

We provide these services as a standard means for requesting TreasuryDirect.gov data. These API are Restful or Representational State Transfer (REST) based.

Restful APIs use HTTP requests to specific URLs, passing URL parameters as arguments to the services. These services return data in the HTTP request as either JSON or XML for parsing and/or processing by your application.

A typical Restful request is generally of the following form:

http://treasurydirect.gov/api/service/output?parameters

In the above example, service indicates the particular service requested and output indicates the response format (usually json or xml).

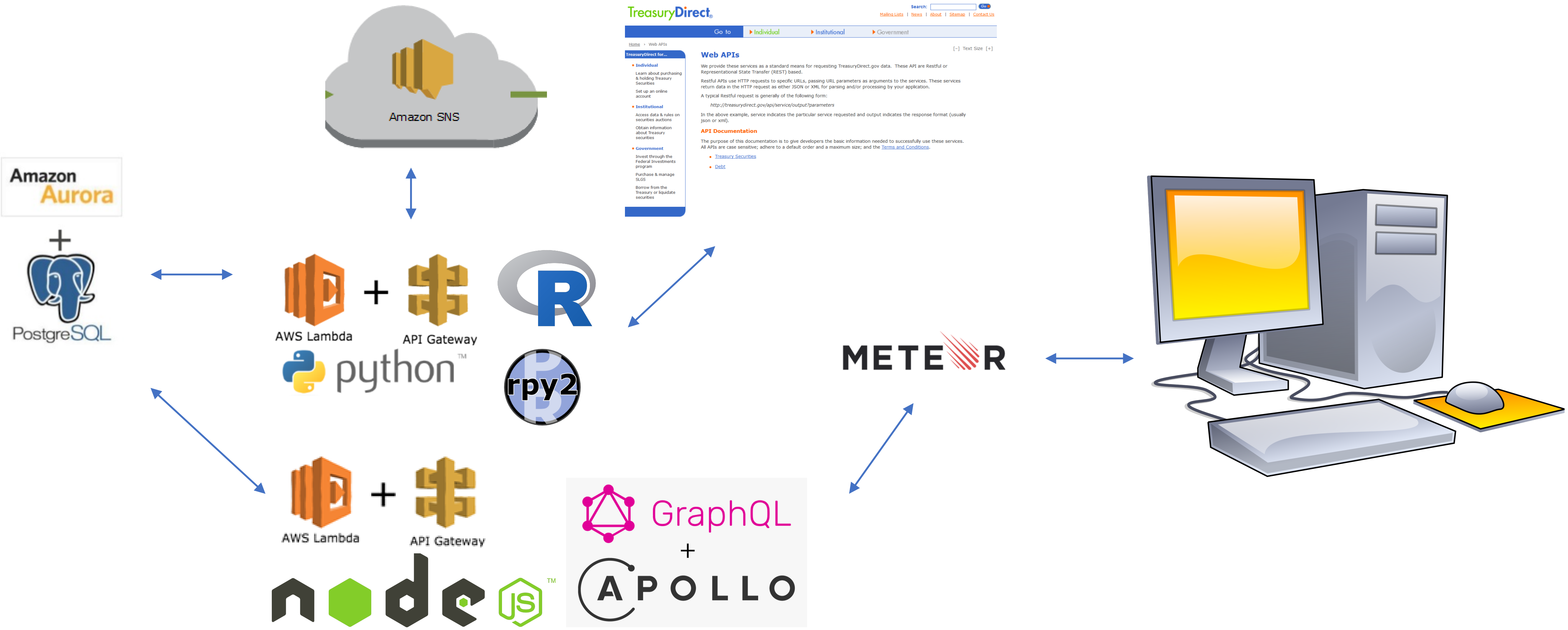
API Documentation

The purpose of this documentation is to give developers the basic information needed to successfully use these services. All APIs are case sensitive; adhere to a default order and a maximum size; and the Terms and Conditions.

- Treasury Securities
- Debt

Step 4

Requirements: I need to disseminate the results to users via an API and display on a website



TreasuryDirect®

Search: Go

Mailing Lists | News | About | Sitemap | Contact Us

Go to ▸ Individual ▸ Institutional ▸ Government

Home › Web APIs

[–] Text Size [+]

TreasuryDirect for...

- Individual
  - Learn about purchasing & holding Treasury Securities
  - Set up an online account
- Institutional
  - Access data & rules on securities auctions
  - Obtain information about Treasury securities
- Government
  - Invest through the Federal Investments program
  - Purchase & manage SLGS
  - Borrow from the Treasury or liquidate securities

Web APIs

We provide these services as a standard means for requesting TreasuryDirect.gov data. These API are Restful or Representational State Transfer (REST) based.

Restful APIs use HTTP requests to specific URLs, passing URL parameters as arguments to the services. These services return data in the HTTP request as either JSON or XML for parsing and/or processing by your application.

A typical Restful request is generally of the following form:

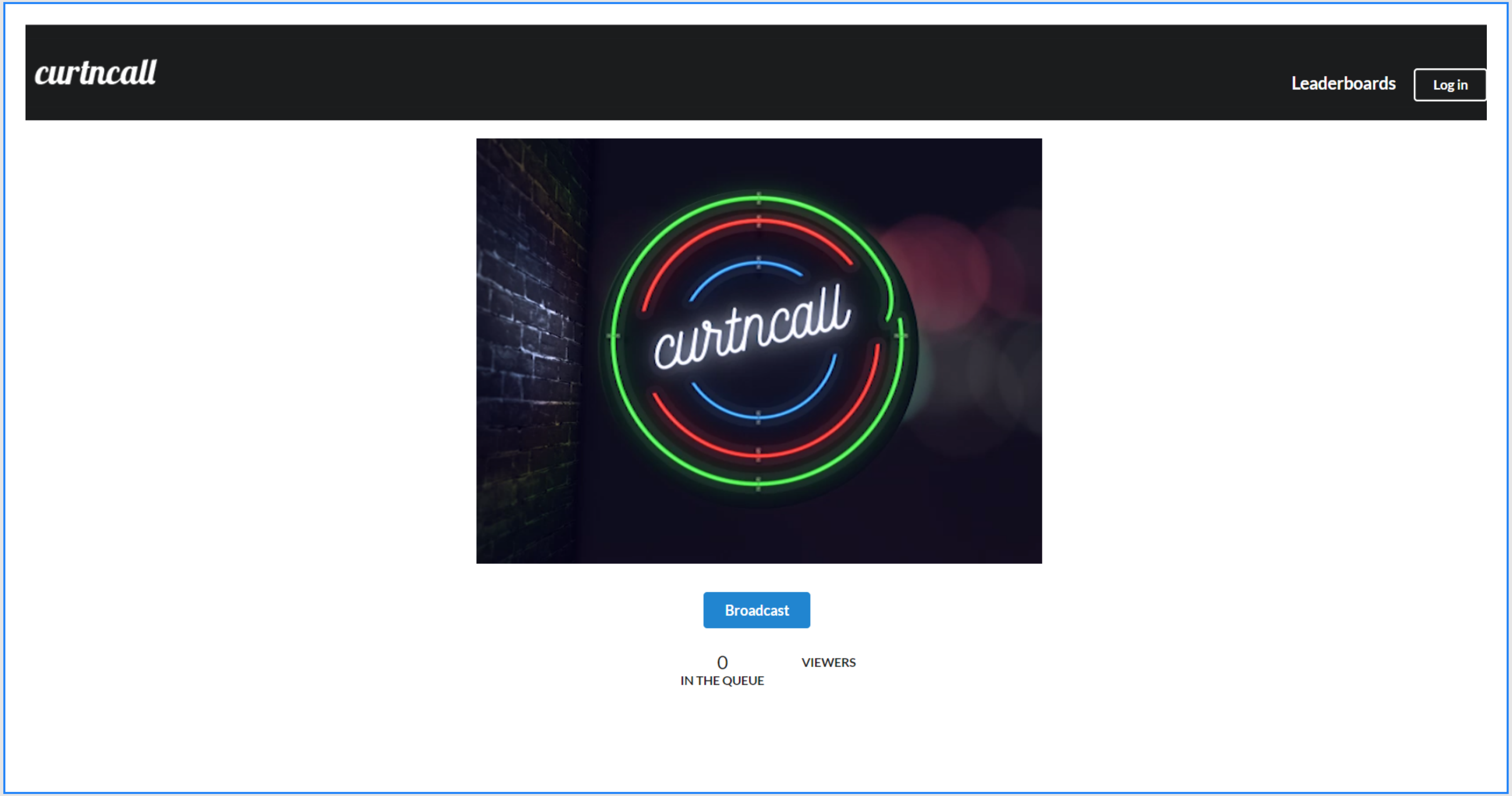
http://treasurydirect.gov/api/service/output?parameters

In the above example, service indicates the particular service requested and output indicates the response format (usually json or xml).

API Documentation

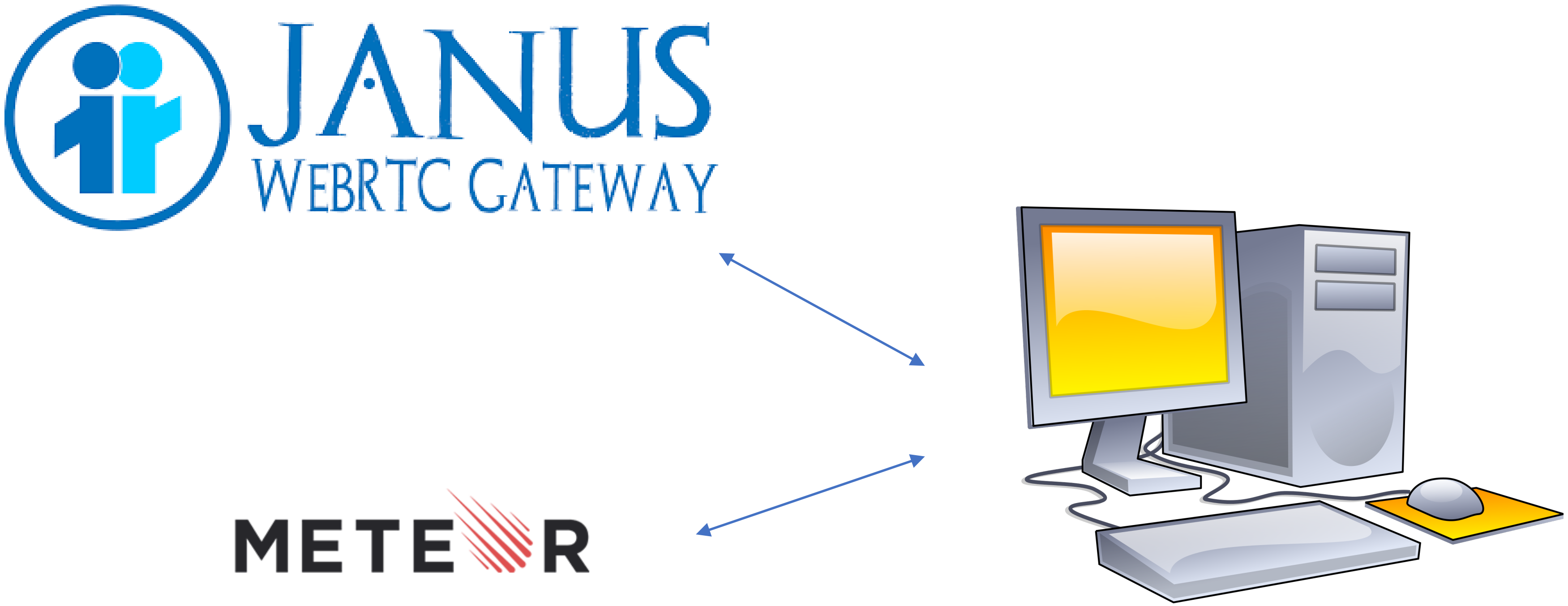
The purpose of this documentation is to give developers the basic information needed to successfully use these services. All APIs are case sensitive; adhere to a default order and a maximum size; and the Terms and Conditions.

- Treasury Securities
- Debt


curtncall
Leaderboards
Log in
curtncall
Broadcast
0
IN THE QUEUE
VIEWERS

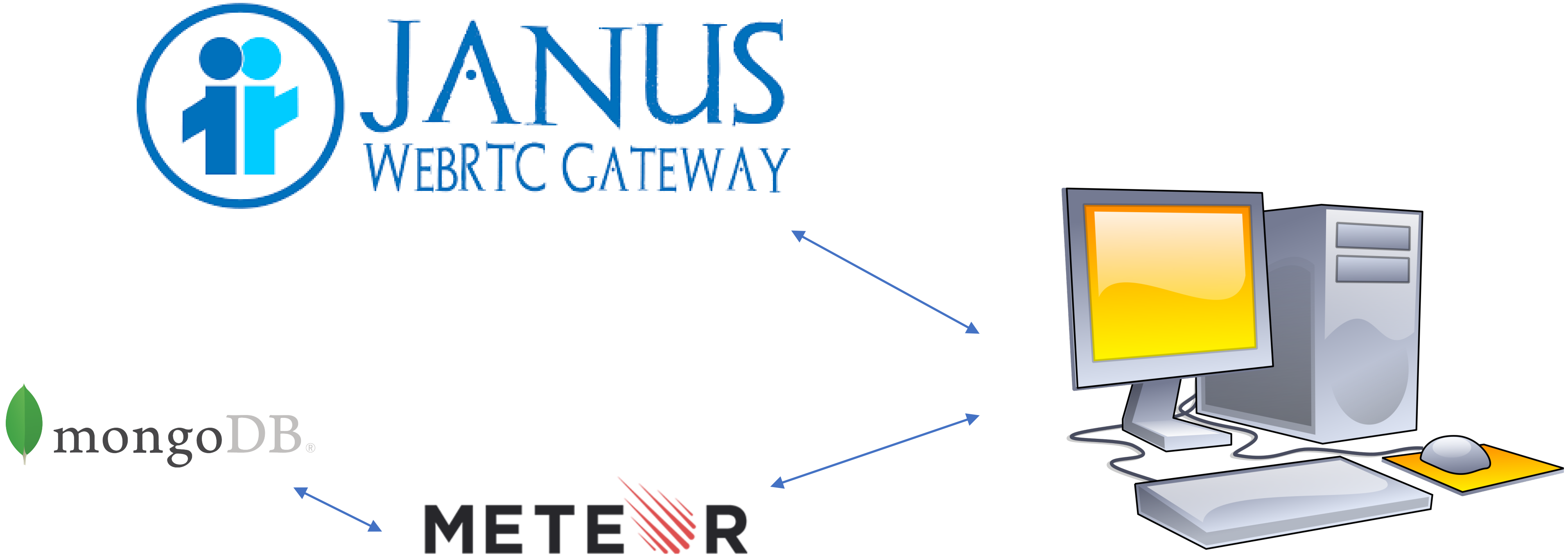
Step 1

Requirements: Show and watch WebRTC from a web application



Step 2

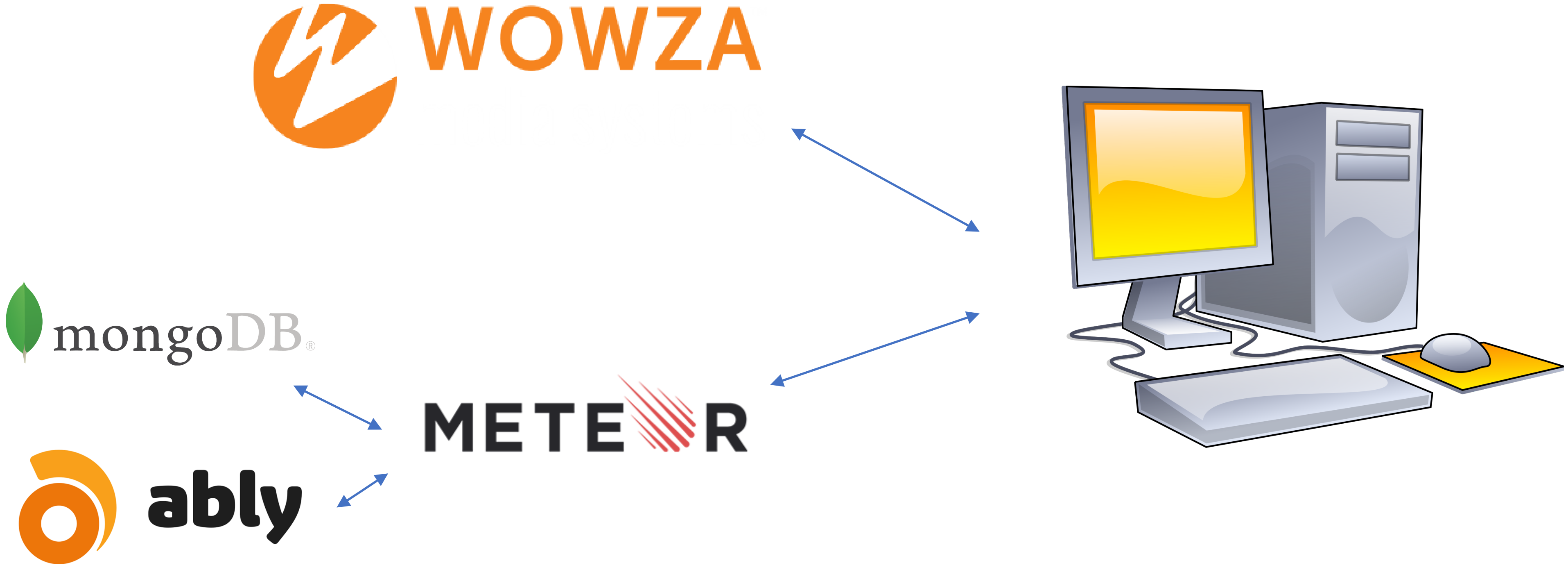
Requirements: Show and watch WebRTC from a web application and store reactions



Step 3

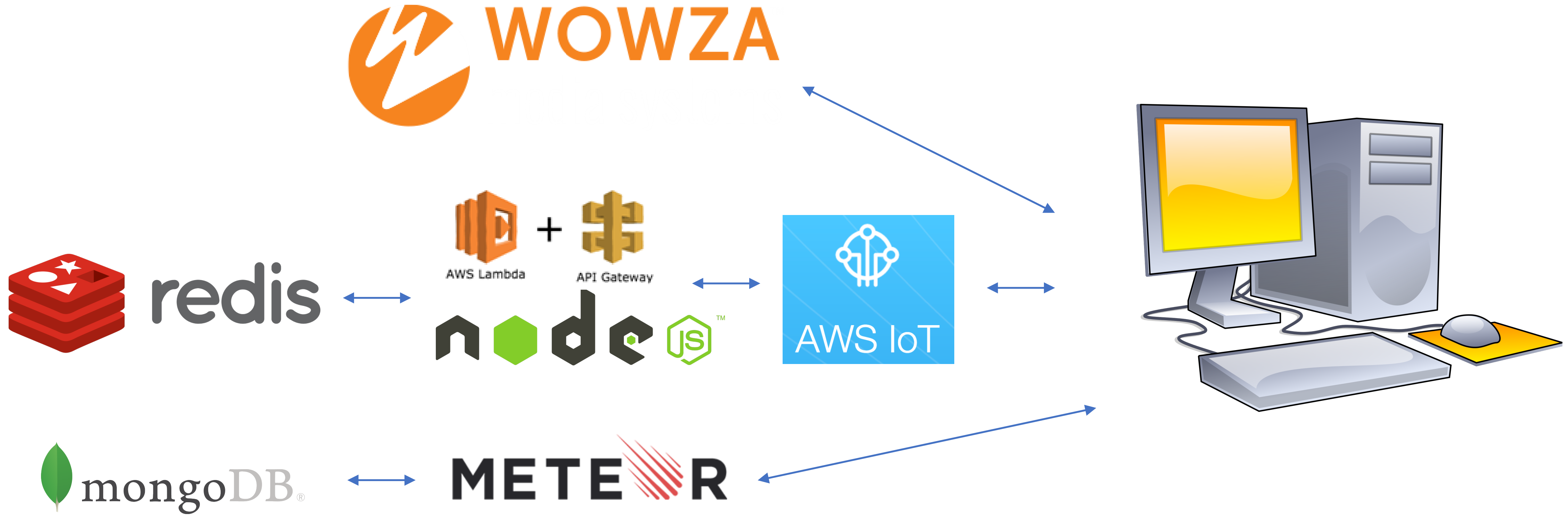
Requirements: Create synchronicity and better performance across broadcasters and viewers

DEMO!



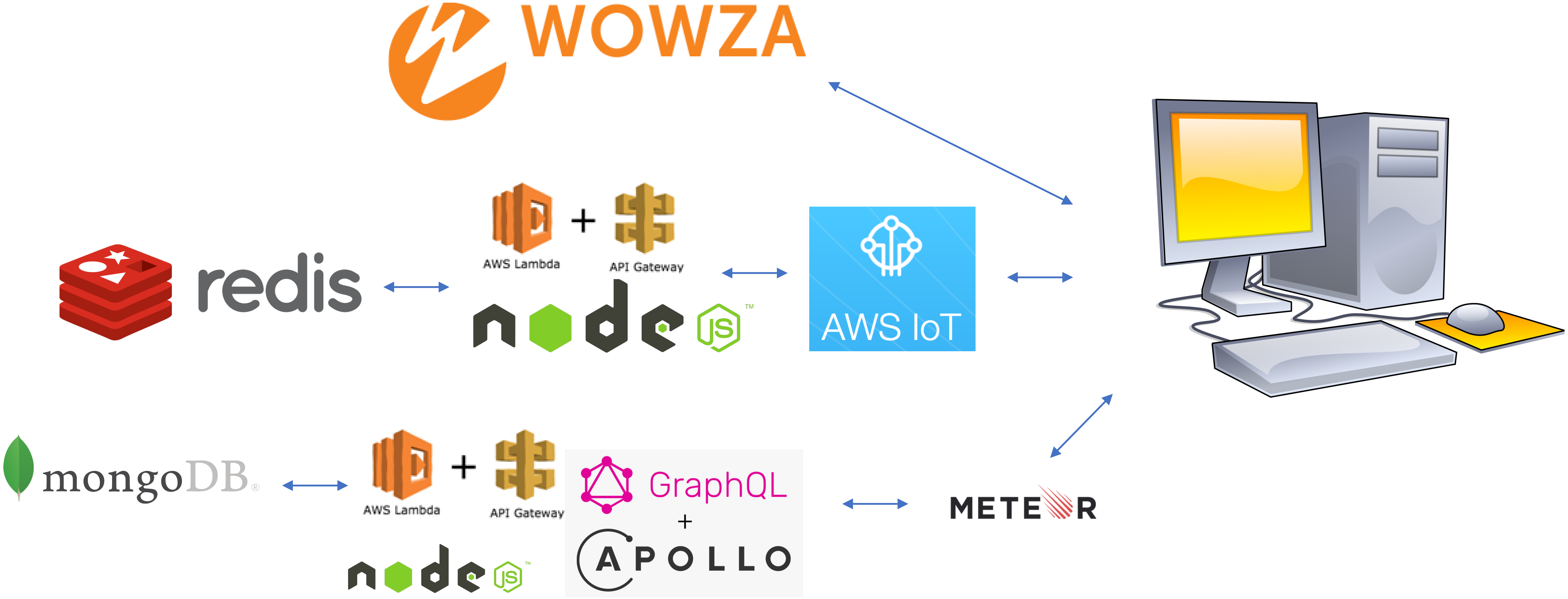
Step 4

Requirements: Move all real-time logic to server side without 3rd party services
DEMO!



Step 5

Requirements: Move all data operations to serverless
DEMO!



Key Takeaways…

Your first “stack” should not be your last “stack”…

You will need to reevaluate your architecture at every phase…

Your goal is to learn more about your product without having to spend too much time and cash on development…

See the sample rubric as a way to measure your features and prioritize your development…

Questions or Thoughts…