NASA/SAO Astrophysics Data System

Beyond “^Author”

Jonny Elliott

Harvard-Smithsonian Centre for Astrophysics

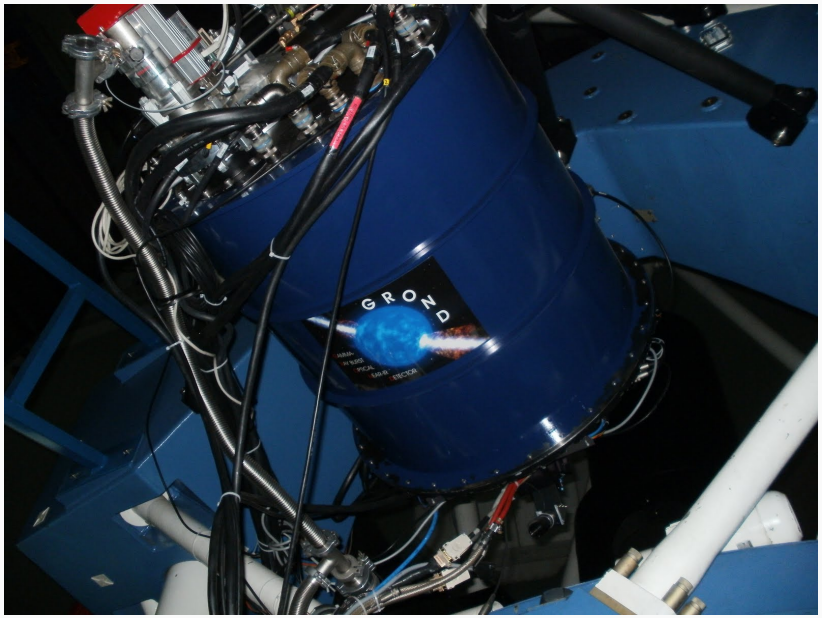
May 10, 2016, MPA Careers Seminar

Table of contents

Who am I and what did I do?

My PhD

My PhD was basically



My PhD

or....

- High Energy Group, GROND team
- Supervisor: Jochen Greiner
- Worked with:
  - i Multi-wavelength photometry of GRB afterglows
  - ii Photometry/spectroscopy of GRB host galaxies
  - iii Semi-analytical modelling/SPH simulations of GRB number counts at high-z
  - iv Machine learning applied to galaxy photometric redshift estimates

When did I decide to switch?

Everyone has their own reasons for leaving science, mine was quite rudimentary: *science* has to be in my top-3 reasons of “Why I stay in science”.

Ordered list of why I stay in science
I like to travel
I like freedom of what to work on
I like writing software
I like ....
...
I like finding scientific results

What did I do?

I began doing things to give me some experience that I could put on my CV aimed at Software Development/Data Science:

1. Took the *Machine Learning* course, Stanford, Andrew Ng
2. Joined *COIN* – Working Group of Cosmostatistics – which develop machine learning tools that utilise statistical techniques that have not yet been applied in Astronomy
3. Did some tutorials for Game Development (Blender/Unity/PyGame)
4. Actively answered questions on StackOverflow

Applied and got a position at the ADS

The interview

For those interested, the interview was of the form:

1. **Informal phone-interview** with the Project Manager. Many questions of the form:

    1. “Tell me a time in which you solved a problem that required pipeline development”
    2. “tell me a time in which you fought for a solution that worked, despite people being against it at the beginning”.
    3. Many that are related to the work done at the ADS “How would you create a pipeline to determine the unique identifiers of a paper from conventional references”, etc.

Lasted around 1.5 hours

The interview

2. **Online technical interview** with the Project Manager and software developers. Three coding questions of the type: **i)**

```
    class Parser(object):
        def __init__(self):
            self.value = None
        def load_bool(self, bl):
            """Turn input into bool, could be of any type"""
```

The interview

ii) Tell us what the following code does, and how you would go about improving the speed of it.

```
    public Elephant (int size) {
        this.size = size;

        int sizeOfElphant(){
            return this.size;
        }

    }
    public main(){
       elephants Elephant[] = new Elephant[100];

       for(int i = 0; i<100; i++ ){
          elephants[i] = Elephant(size=i*100);
          System.out.print(elephants[i].sizeOfElephant());
       }
    }
```

The interview

iii) Can you make a button that dissapears when clicked using JavaScript

```
<html>
<head>
  <title>Button that dissapears</title>
</head>

<body>
    <button type="button">I will dissapear</button>
</body>
</html>
```

1 hour in total

The interview

3. **Formal interview** with all of the group. Only questions that stood out were: “Why do you want to leave science?”.

1 hour

(Before I was contacted again, I contributed to the code base of the ADS)

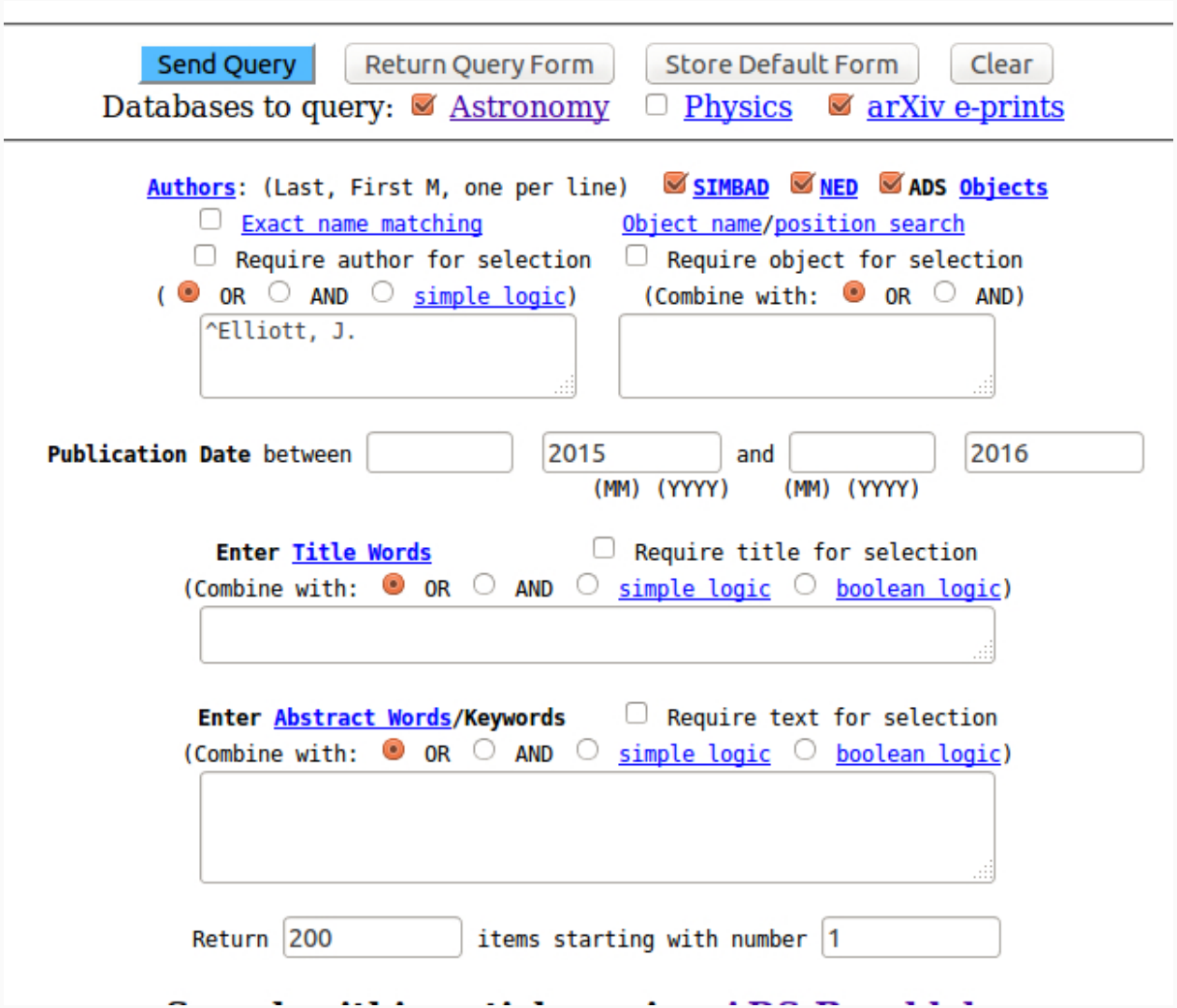
4. **Informal interview** with the Project Manager

1 hour

What do you do at the ADS?

The ADS

For the average astronomer, the ADS is this:

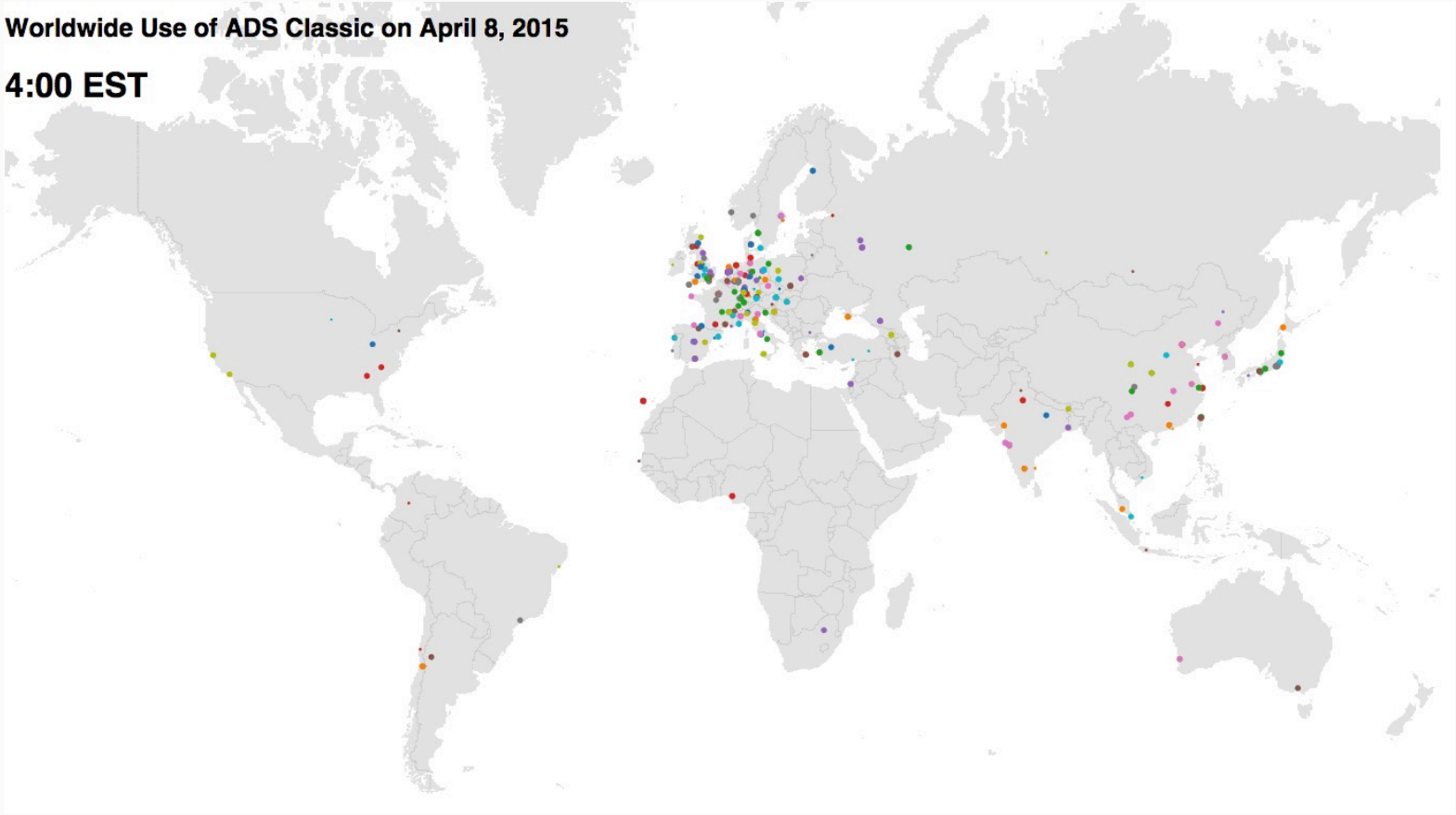


The ADS

To give you an idea of the scope and usage of the ADS, it has around:

- $\sim 50,000$ users daily
- 10 million total users
- 11.2 million documents
- 77 million citation links

The ADS



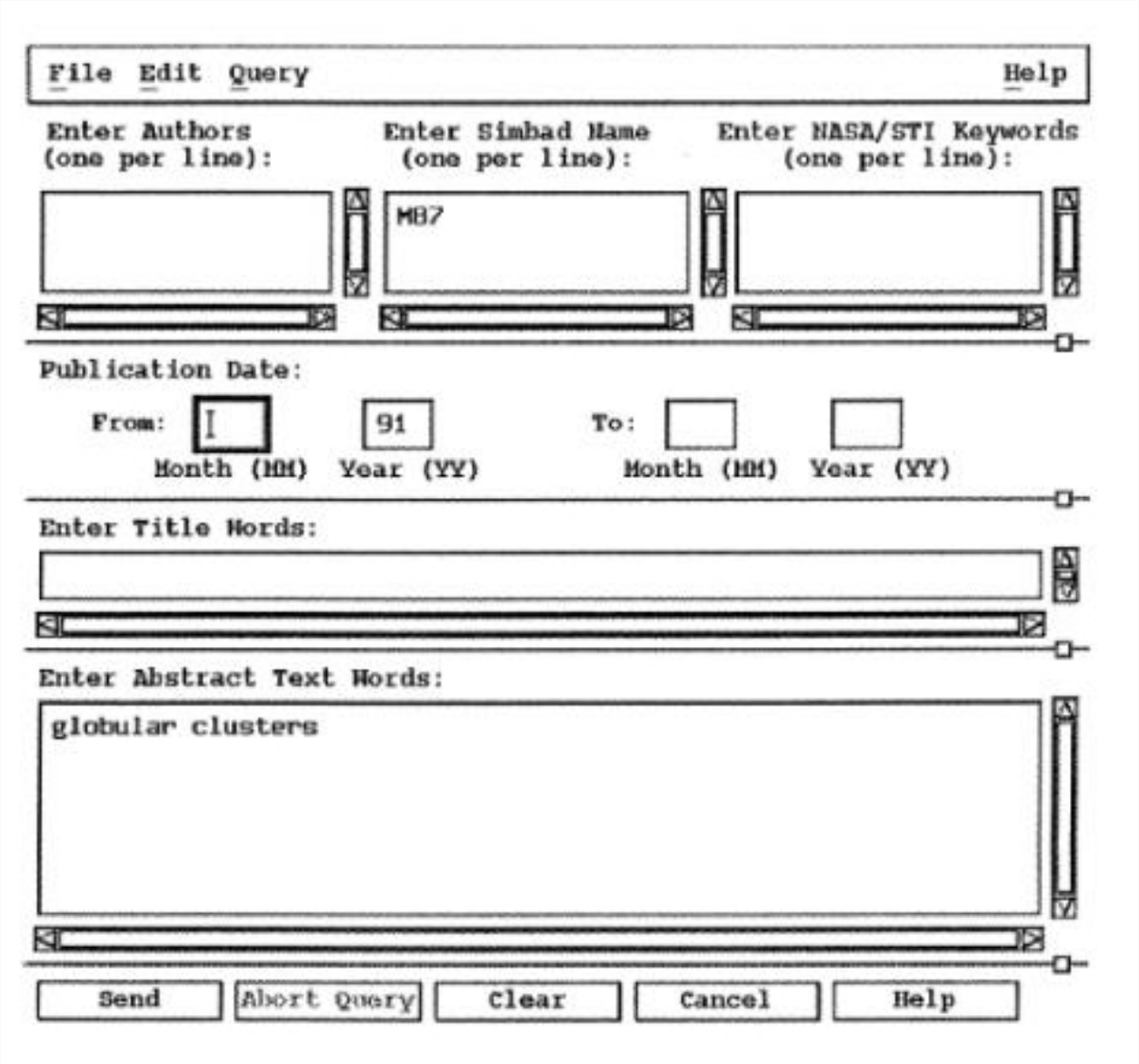
–alex.holachek.com

The ADS

You might think, *“why fix something that isn’t broken?”*, or the ADS works perfectly fine

The ADS

Not changed even since before the World Wide Web was invented



Why fix something that isn't broken?

You might think, *"why fix something that isn't broken?"*, or the ADS works perfectly fine:

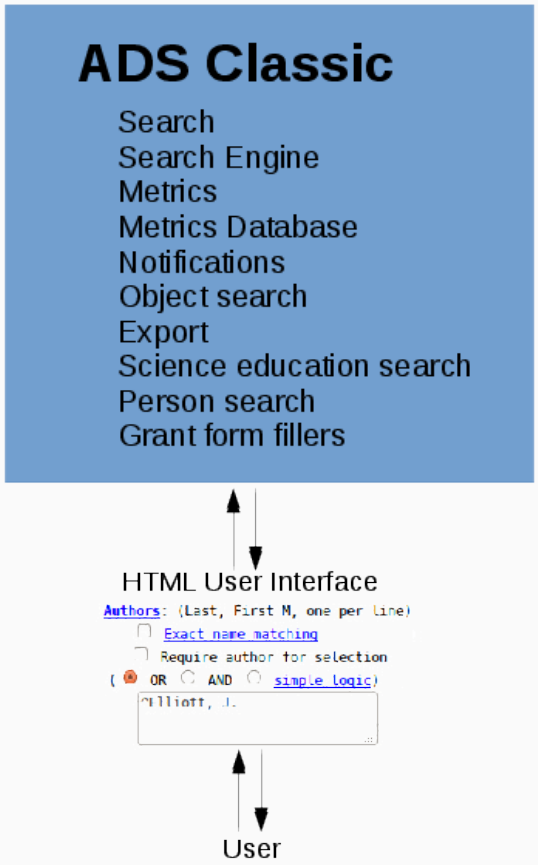
1. The ADS was built in 1992, primarily in `perl`, `C`, and `html`.
2. Service disruption: 1 (or 2) people in the world know how to copy to mirror sites
3. Failure: at most 1 person knows how to fix it if there is a serious issue
4. Extension: 1 or zero people know how to extend the services (this does not necessarily mean nothing will break)
5. Extension: tied to an ecosystem built 25 years ago, can't do anything new that could be useful for researchers

Rebuilding the ADS and its infrastructure

We are currently rebuilding the ADS from the ground-up using state-of-the-art techniques.

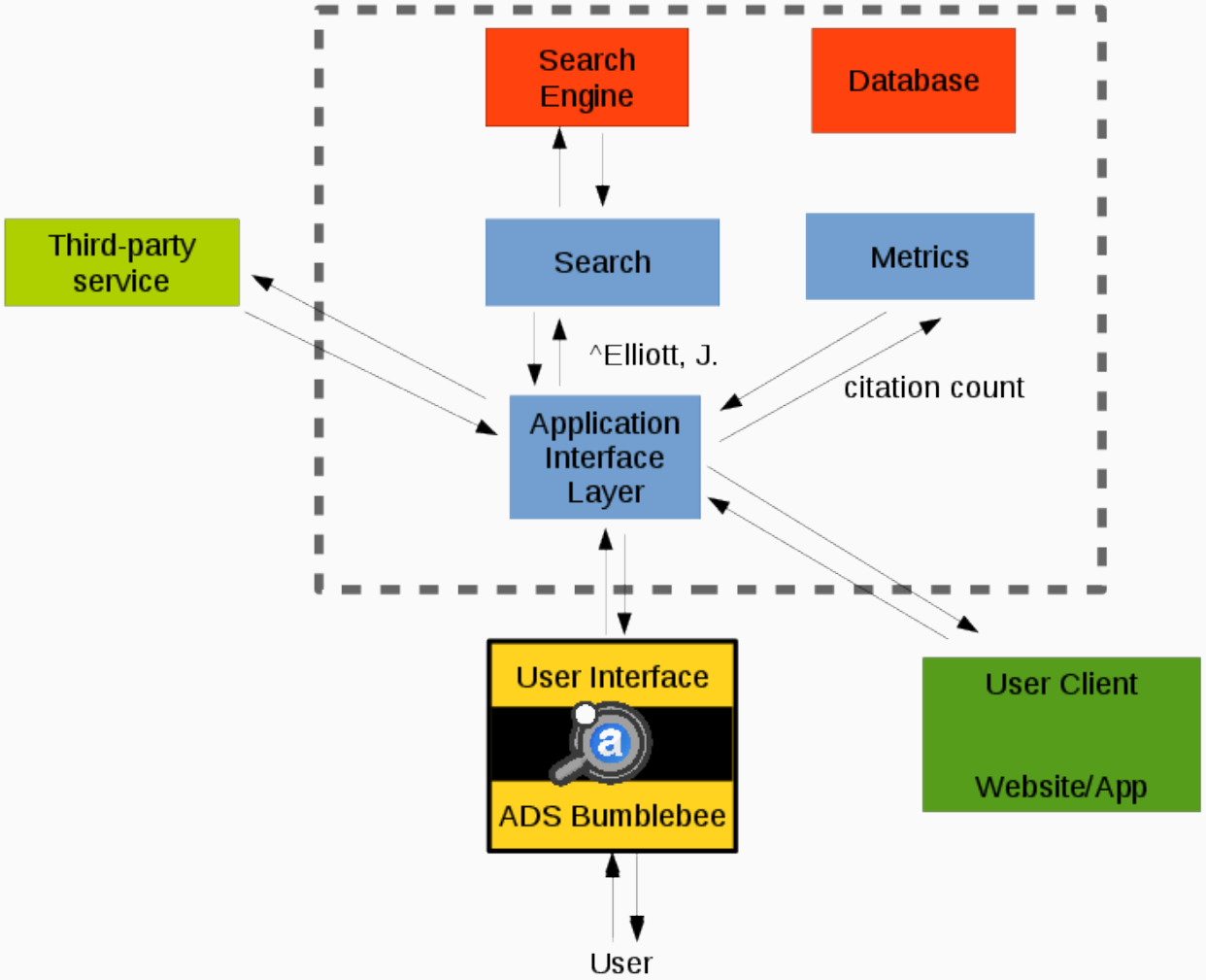
Rebuilding the ADS and its infrastructure

As with most old software, we have a system that is essentially a single monolith application:



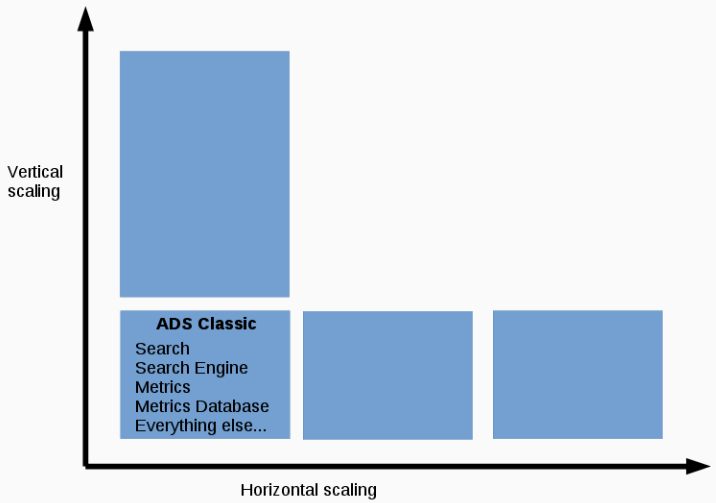
Rebuilding the ADS and its infrastructure

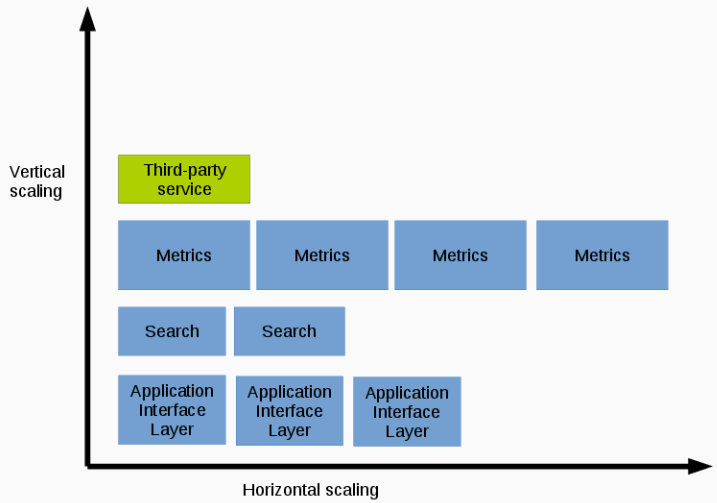
A typical way to deal with this is to switch to a *Microservices Architecture*:



Rebuilding the ADS and its infrastructure

There are major advantages in terms of scalability, maintenance, adding new features/services, bandwith, etc.





But with such an infrastructure requires time to build, test, automate, integrate old systems.

OK, but day-to-day?

–ADS CIRCA 1996

Day-to-day

1. 10 minute stand-up meeting @ 10am
2. Develop software e.g.:
   - 2.1 Frontend: add a widget that allows people to limit searches by object type
   - 2.2 Backend: build a service that alerts users that their citations have increased
   - 2.3 DevOps: build system that can automatically deploy software changes, and test that they work as expected
   - 2.4 Data Science: build service that can automatically generate keywords for a document using machine learning techniques
3. Lunch, talks
4. Develop software
5. 5pm go home

What, where, and why?; Working in software

Working in software

“I’d like to work in software, but what the ADS does isn’t that interesting to me, why is this relevant?”

Such work touches upon a wide range of technologies

Jobs aimed at software

	DevOps	Software Engineer	Data Scientist
Salary	$50k - 150k	$60k - 350k	50k - 250k
Qualif.	–	CS (Msc./PhD)	PhD
Tech.	AWS (cloud services)	Python, Java, Go	Python, R
	Provisioning	Ruby, C++, C,	Stats packages
	Virtualisation	JavaScript	(Pandas, scikitlearn)
	Containerisation	Web Frameworks	“Big data” framework
	SysAdmin	Test-driven Dev.	(Hadoop, Spark)
	Integration		Functional prog.
			(Julia, Scala)

Companies: Google, Intel, Hulu, Netflix, Amazon, Cisco, IBM, Booking.com, Kayak, Microsoft, GitHub, Atlassian, RedHat, etc.

Locations: San Fransisco, Seattle, Boston, New York, London, Berlin, Munich, Amsterdam, Singapore, Hong Kong, Shanghai, Sydney, etc.

What I think is useful to do if you want a software job

What's useful to do

Very simple list of things I think are relevant to know before

1. Object Oriented Programming
2. Can program in the language you claim you can
3. Algorithms and data structures (Cracking the Coding Interview, G. L. McDowell, CareerCup, 2013))
4. Version control (git, SVN)
5. Projects, work, courses that demonstrate your ability related to Software Development (basically provable experience)
6. Read some technical sites:
   `https://news.ycombinator.com/`
7. A lot of jobs are also found on HackerNews:
   `http://whoishiring.io/`
8. Test-Driven Development with Python (Harry Percival, O'Reiley, 2014)
9. Get involved in StackOverflow, or Open Source

Conclusions

Jobs at ADS

- Coming soon: 2-4 Software Engineering/Curation positions
- Salary 60k-120k
- Nice benefits for a Government position (pension, health, dental, transport)
- Visa sponsorship
- Harvard benefits: gym, banks
- Located in Cambridge (Boston - one of the tech hubs of US: meetups, conferences, talks)

Questions

Questions*?

jonathan.elliott@cfa.harvard.edu

github.com/jonnybazookatone

jonnyelliott.org

*Will I increase your citations? – Account: 453235, Routing number: 1039485