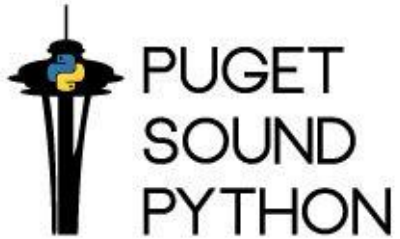
Open Source @ Microsoft

Doug Mahugh
Open Source Programs Office

Microsoft



Wed 3/29 6:00PM
Apptio – Bellevue, WA

Agenda

Open Source @ Microsoft

A tour of the Open Source Programs Office (OSPO)

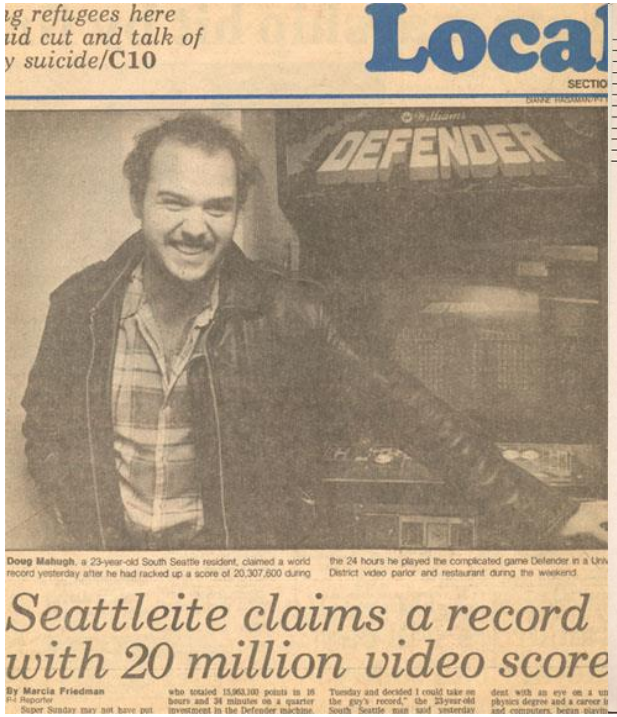
Python FTW

A few examples of the use of Python in the work of the Open Source Programs Office

About @dmahugh

11 years at Microsoft in international standards, developer evangelism, and since 2015 as a founding member of the Open Source Programs Office.

Python is my favorite language for many reasons, including syntactical clarity (curly braces suck), consistency (e.g., *everything's* an object), expansive standard library, and the vibrant Python community. Thinking in Python simplifies complex problems.


g refugees here
id cut and talk of
y suicide/C10

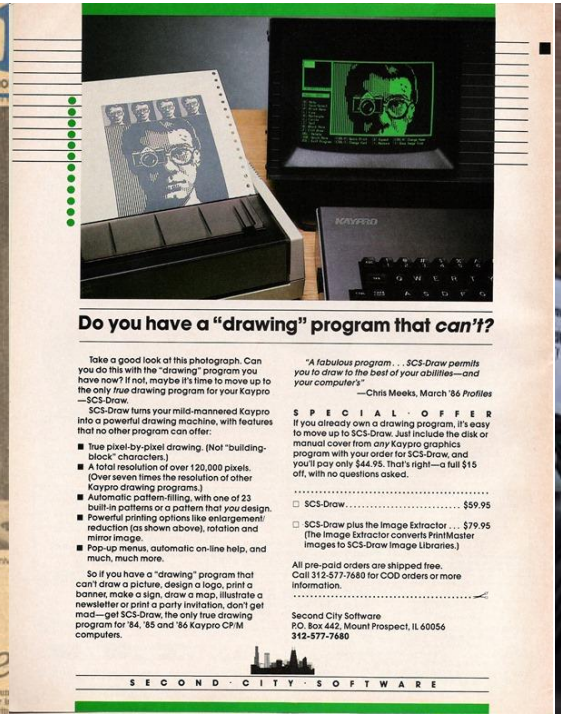
Local

DEFENDER

Doug Mahugh, a 23-year-old South Seattle resident, claimed a world record yesterday after he had racked up a score of 20,307,600 during the 24 hours he played the complicated game Defender in a Uni District video parlor and restaurant during the weekend.

Seattleite claims a record with 20 million video score

By Marcia Friedman

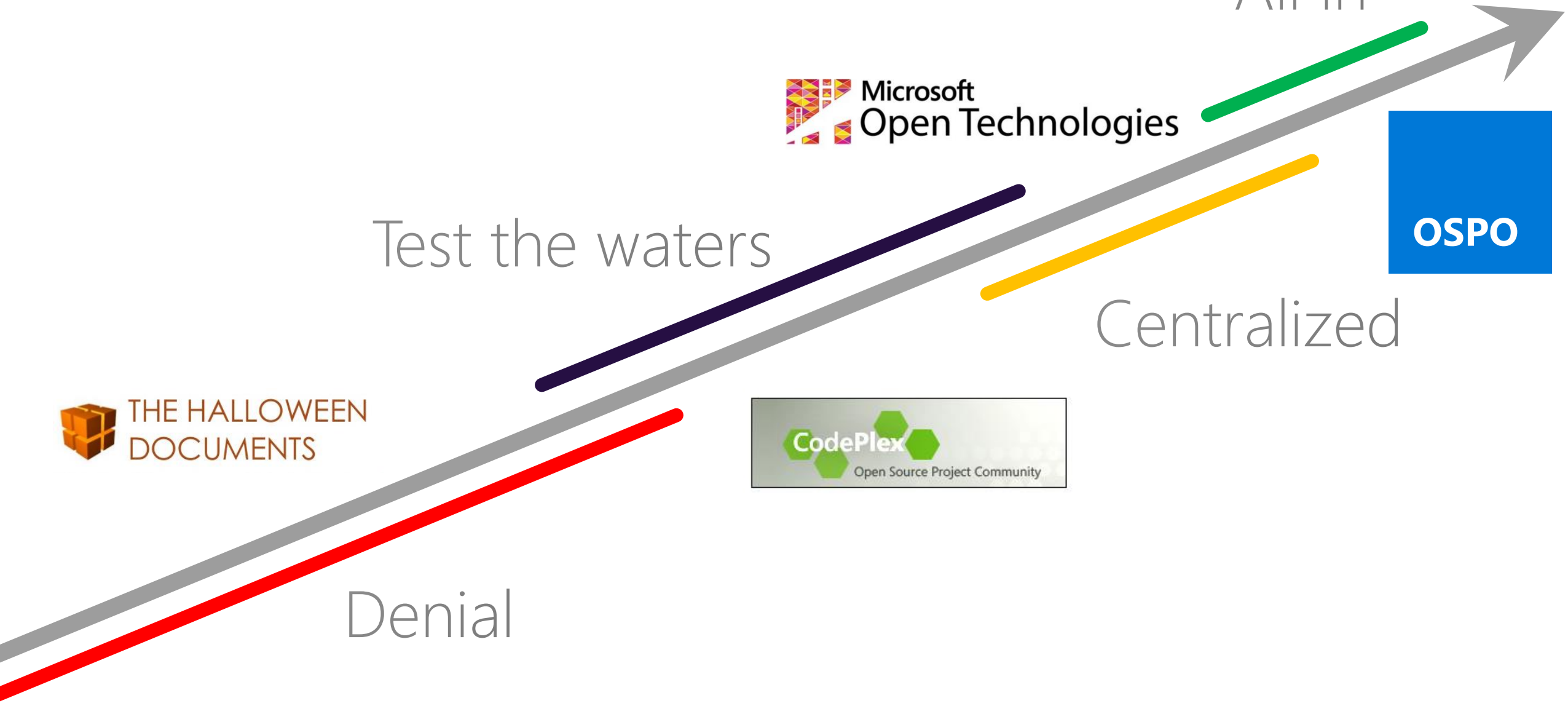


Open Source @ Microsoft

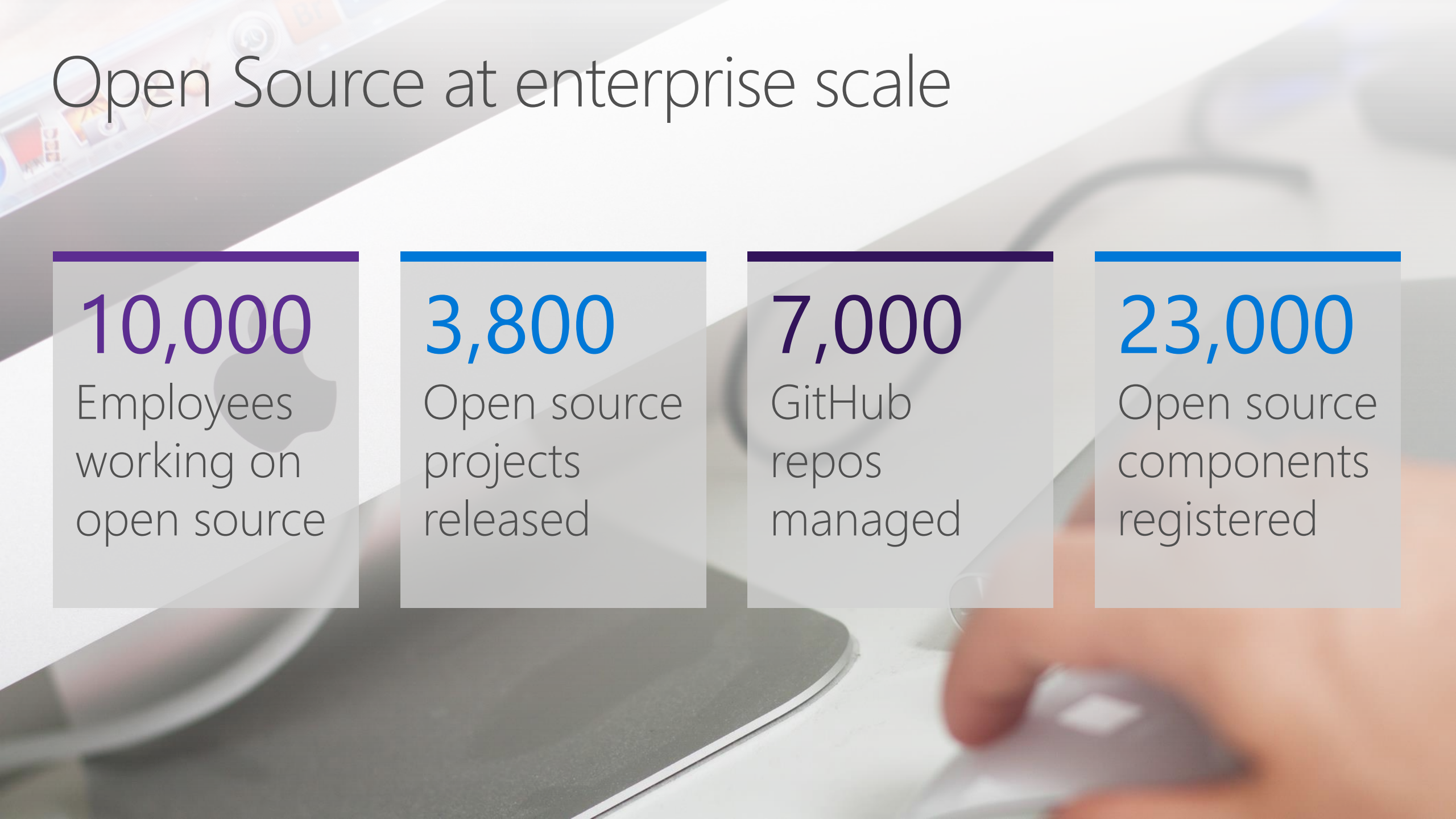
How Microsoft enables enterprise-scale innovation across thousands of engineers integrating, releasing, and contributing to open source projects.

Microsoft

Evolution


"All in"
Microsoft
Open Technologies
OSPO
Test the waters
Centralized
THE HALLOWEEN
DOCUMENTS
CodePlex
Open Source Project Community
Denial

Open Source at enterprise scale



10,000
Employees working on open source

3,800
Open source projects released

7,000
GitHub repos managed

23,000
Open source components registered

Scenarios


Microsoft
Integrate/Enable
Contribute
Release
Open Source

Why?

Why integrate?

Community

Time to market

Cross-platform

Best practices

Recruiting/retention

Why contribute?

Forks are expensive

Interoperability

Know your customer

Why release?

Expand market

Build ecosystem

Enable contributions

Faster feedback

Thought leadership

Develop strategic relationships

Business considerations

What is the core value we're delivering? Can open source help?

Online service, boxed product, or app store?

Internal use, bundled, or shipped to customer?

Each business decision is unique

Consistent experience across public repos (branding, code of conduct, license), but…

Each open source engagement is guided by a particular business unit in pursuit of their specific goals.

Every business group is responsible for their own strategy.

Examples

Quick to market with cross-platform support, Microsoft has made numerous contributions to the Electron codebase.

Microsoft's .NET team gets quick feedback on new features, community makes 60% of code contributions.

Popular services based on Linux, Hadoop, Redis, Node, MongoDB, and other OSS projects. Microsoft is contributing Azure's data center design experience to the Open Compute Project.

Open Source Programs Office

Responsible for simplifying and promoting open source engagement across Microsoft.

Small, agile team, mostly engineers.

Located in the developer tools organization.

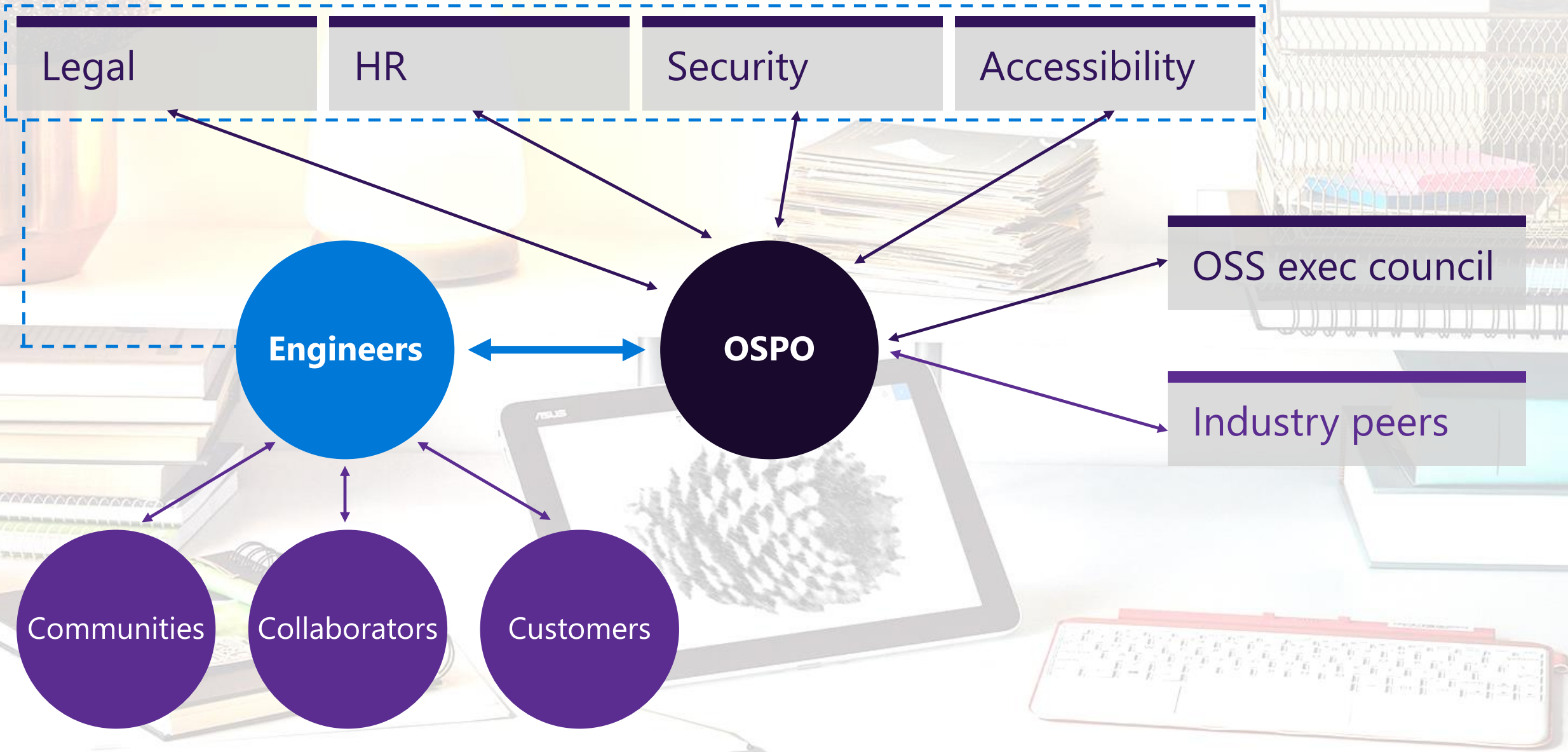
For company-wide consistency, OSPO provides a single source of truth for policies and processes.

Primary customer is engineering teams, but also works with Legal, Security, and others.

Policies and processes

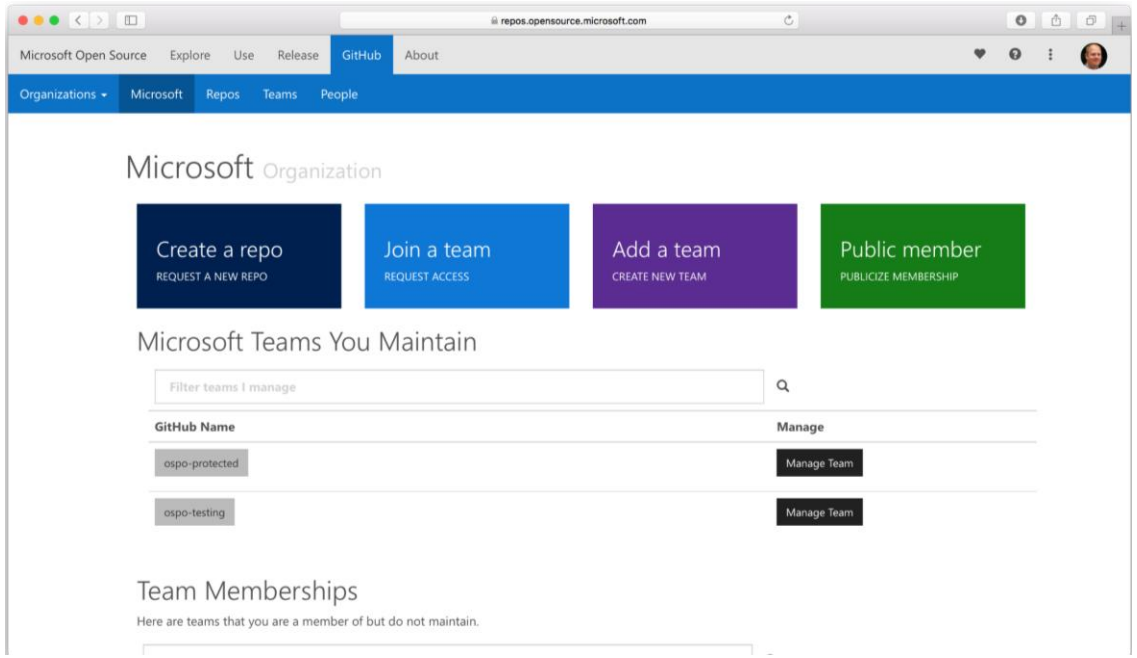
- Component registration and review
- IP scanning
- Repo setup (license, readme)
- Use of public/ private repos
- Security
- GitHub user requirements (2FA, linked identity, etc.)
- Contributor license agreements
- Code of conduct
- Accessibility

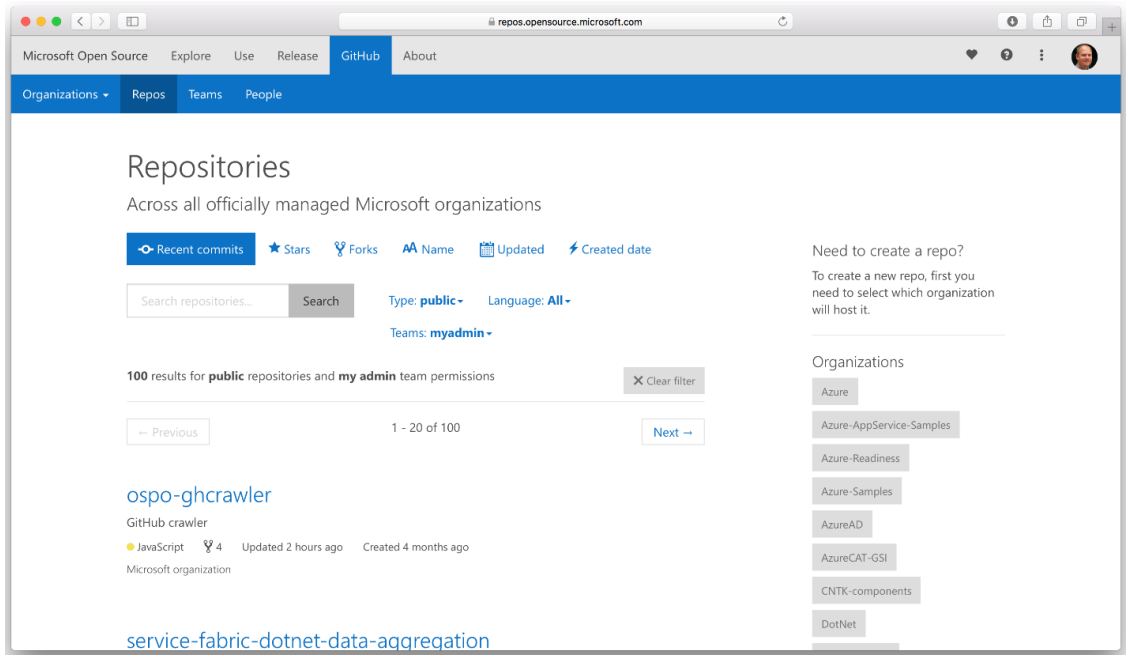
Facilitating communication



OSPO tools for GitHub management

Standardized, consistent approach to creating repos and managing teams





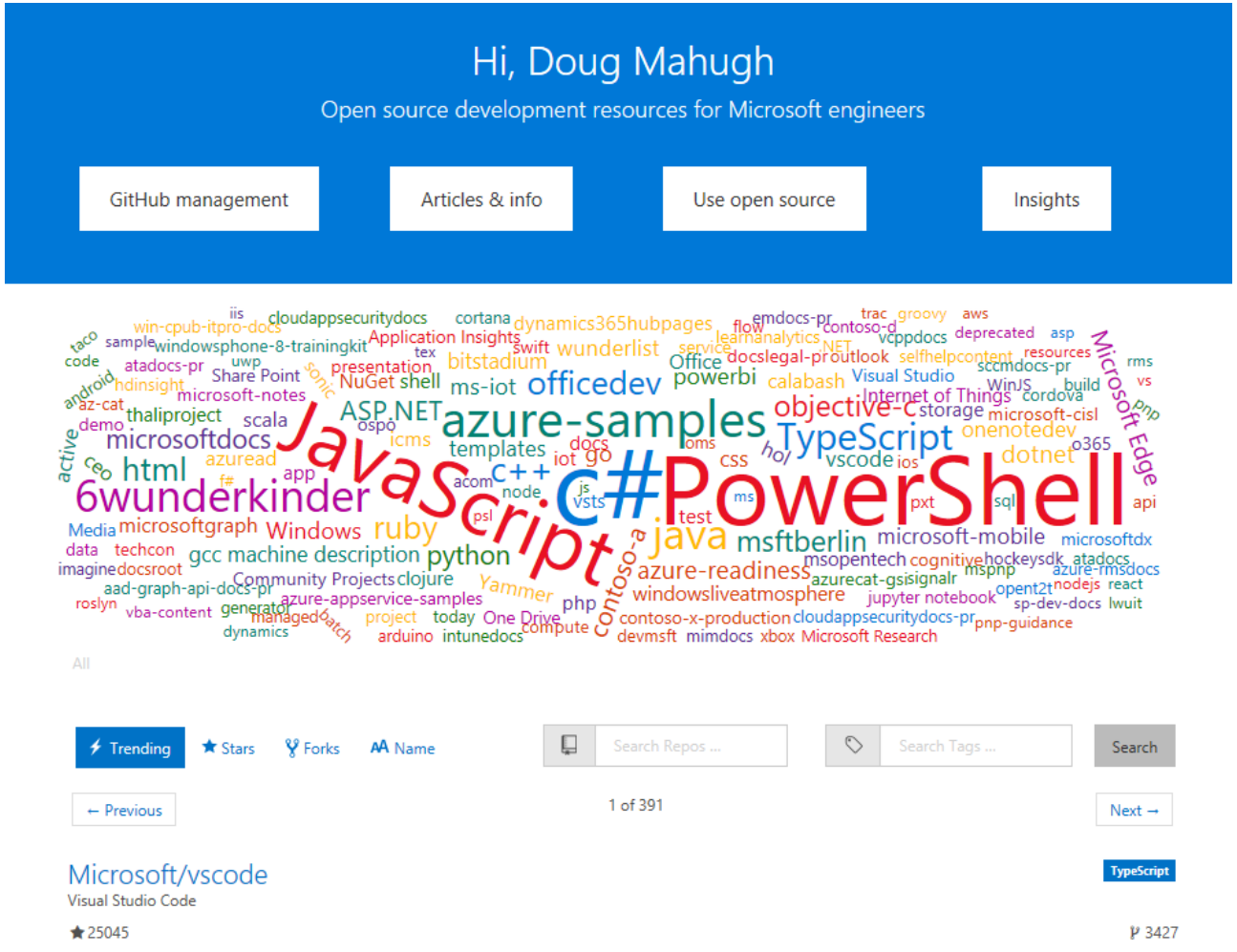
opensource.microsoft.com

Home page for open source *engineering* at Microsoft

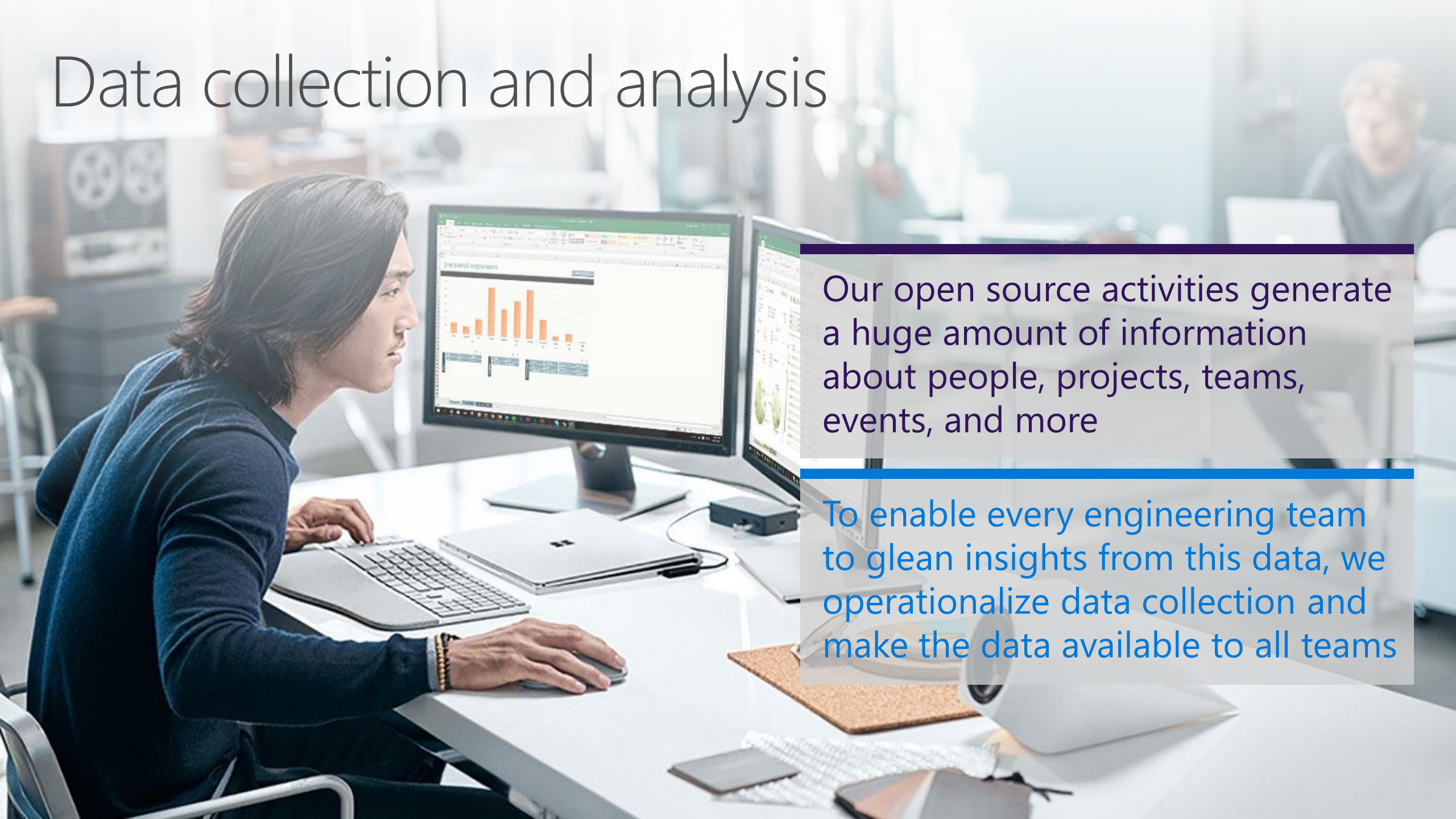
Tag cloud and related tools for repo discoverability

(prototyped in Python, of course!)

Additional functionality provided for Microsoft corporate identities



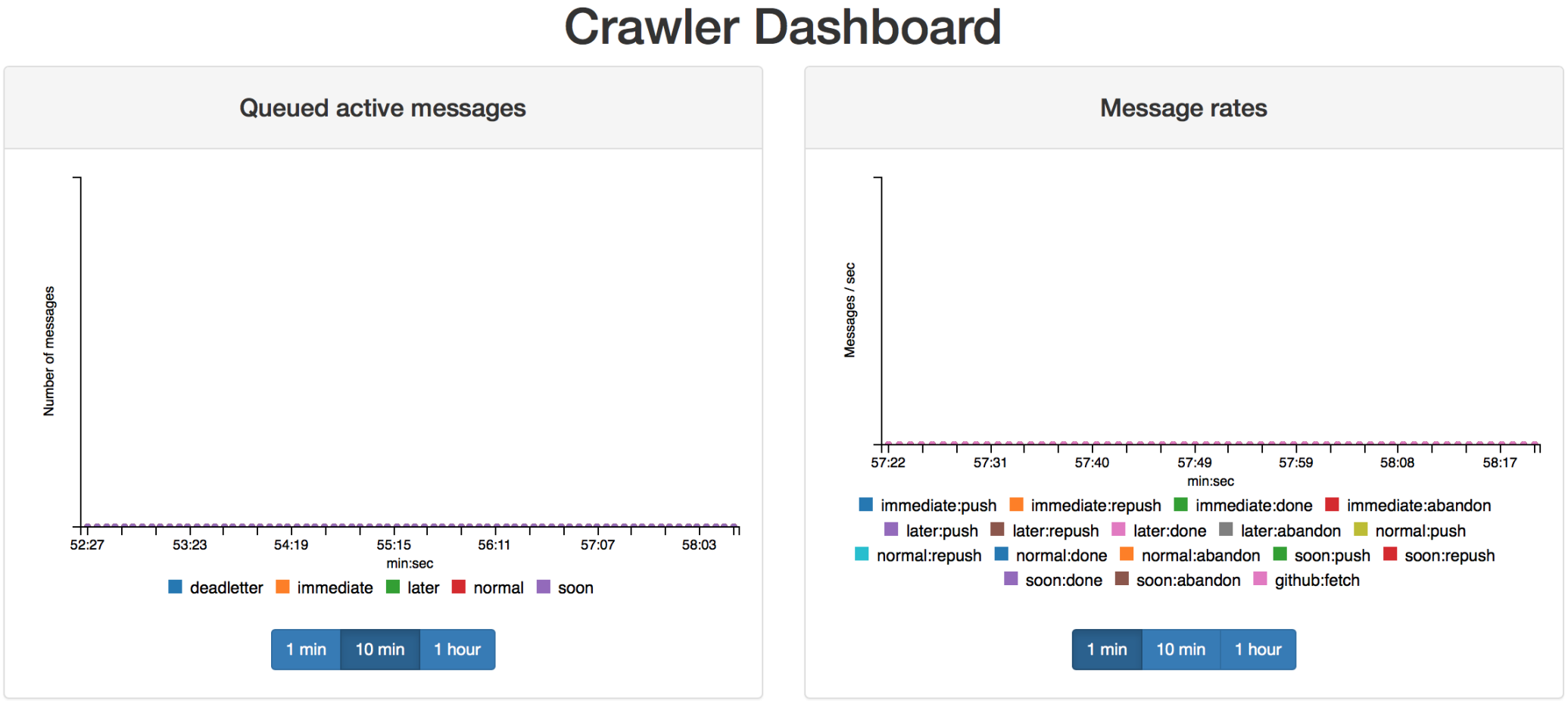
Data collection and analysis



Our open source activities generate a huge amount of information about people, projects, teams, events, and more

To enable every engineering team to glean insights from this data, we operationalize data collection and make the data available to all teams

Crawler in a box



GHCrawler – https://github.com/Microsoft/ghcrawler
Raw data – https://github.com/microsoft/ghinsights

GitHub Insights

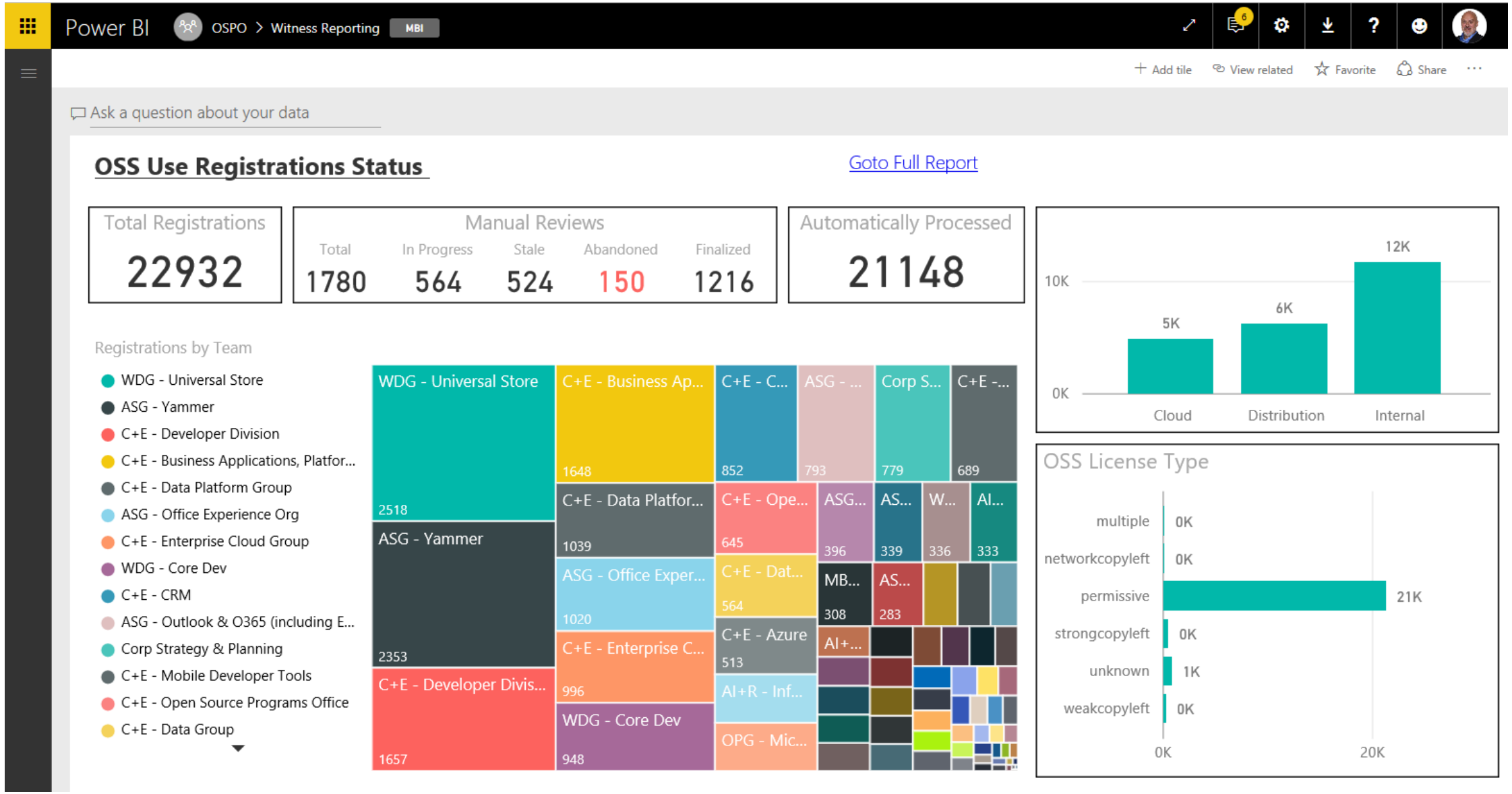
Project Health

Issue Response Time

Source of Contributions

Compliance

Governance Insights



Culture change

Building a sustainable open source culture at Microsoft

Managing fear

Developers fear losing value

Managers fear losing control

Executives fear losing IP

"The oldest and strongest emotion of mankind is fear, and the oldest and strongest kind of fear is fear of the unknown."

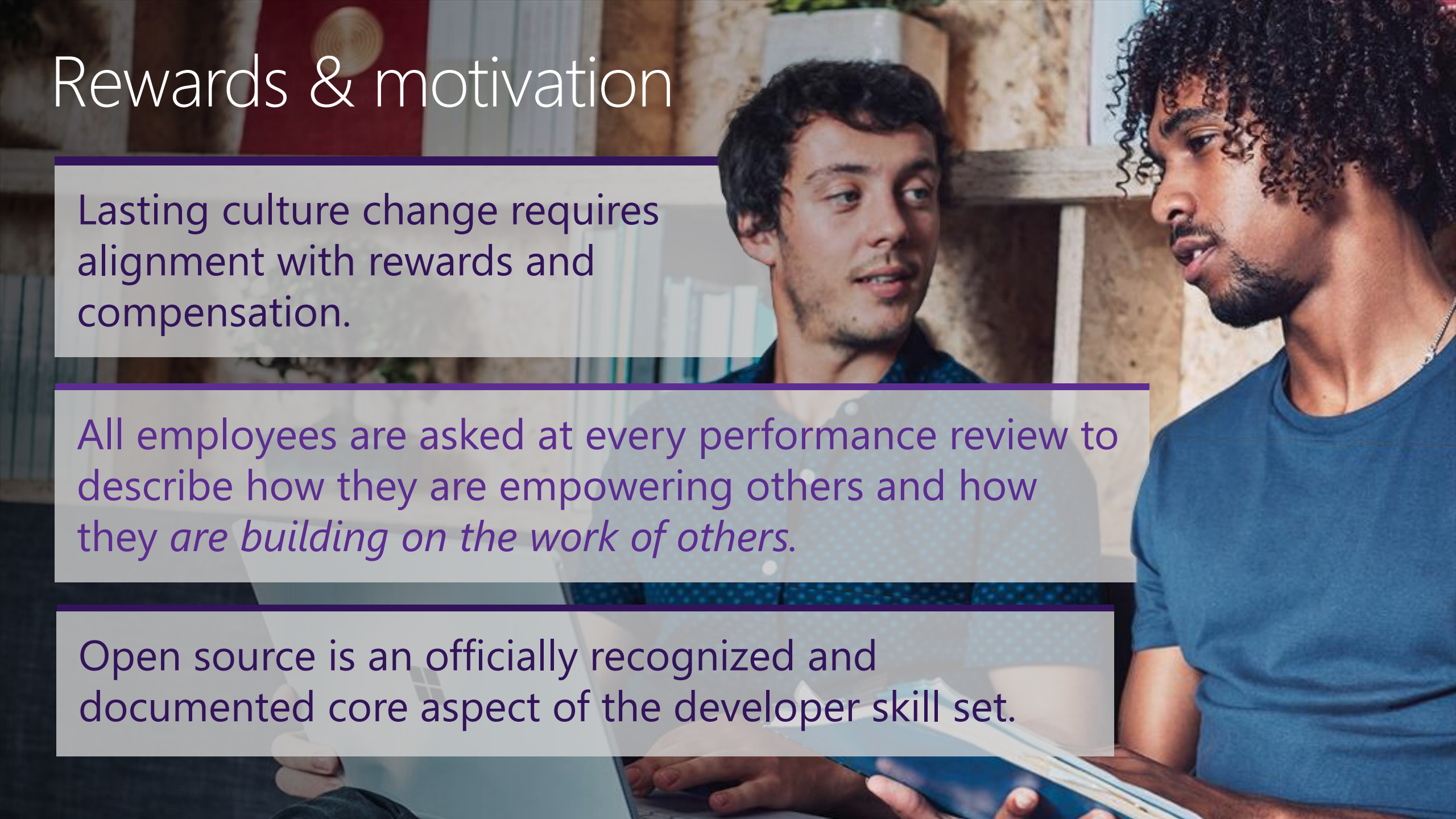
- H. P. Lovecraft

Inner sourcing

In 2014, Microsoft CEO Satya Nadella directed all Microsoft engineers to "open source internally" - anyone at the company can see anyone else's code and use it as needed.

This vision is now a day-to-day reality for Microsoft engineers.

Rewards & motivation



Lasting culture change requires alignment with rewards and compensation.

All employees are asked at every performance review to describe how they are empowering others and how they *are building on the work of others*.

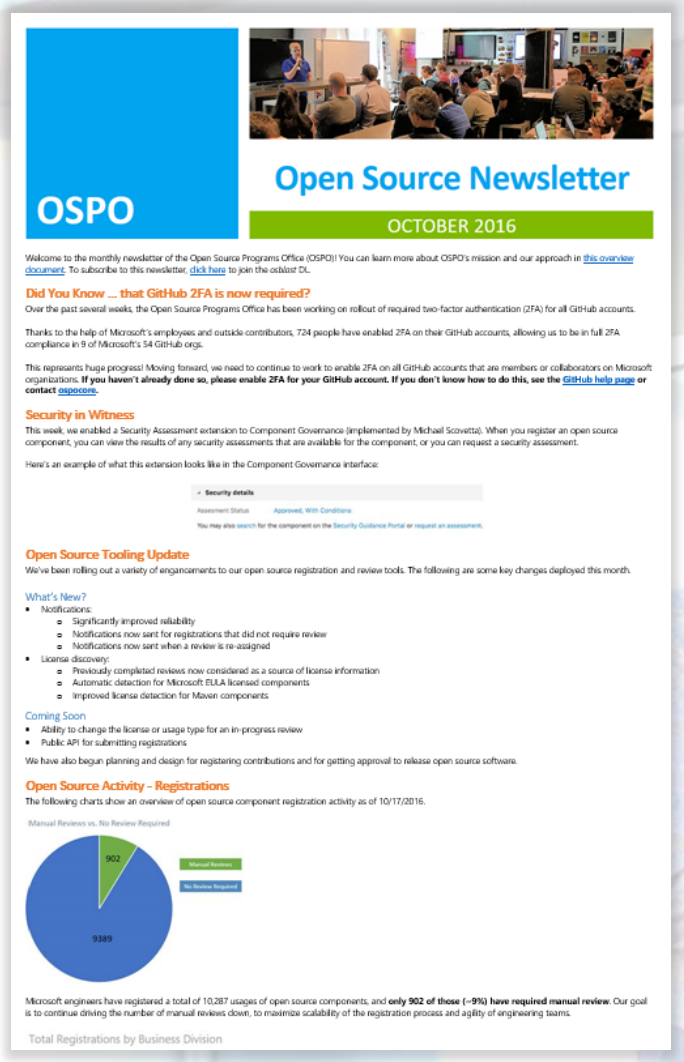
Open source is an officially recognized and documented core aspect of the developer skill set.

Communication

Internal wikis seeded with open source FAQ content, for long-term discoverability/retention

Open Source Night – monthly FTE event, growing attendance and participation

Internal email DLs for us-to-them, peer-to-peer, support questions, policy questions



OSPO

Open Source Newsletter

OCTOBER 2016

Welcome to the monthly newsletter of the Open Source Programs Office (OSPO)! You can learn more about OSPO's mission and our approach in this overview document. To subscribe to this newsletter, click here to join the *osblast* DL.

Did You Know … that GitHub 2FA is now required?

Over the past several weeks, the Open Source Programs Office has been working on rollout of required two-factor authentication (2FA) for all GitHub accounts.

Thanks to the help of Microsoft's employees and outside contributors, 724 people have enabled 2FA on their GitHub accounts, allowing us to be in full 2FA compliance in 9 of Microsoft's 54 GitHub orgs.

This represents huge progress! Moving forward, we need to continue to work to enable 2FA on all GitHub accounts that are members or collaborators on Microsoft organizations. **If you haven't already done so, please enable 2FA for your GitHub account. If you don't know how to do this, see the GitHub help page or contact ospocore.**

Security in Witness

This week, we enabled a Security Assessment extension to Component Governance (implemented by Michael Scovetta). When you register an open source component, you can view the results of any security assessments that are available for the component, or you can request a security assessment.

Here's an example of what this extension looks like in the Component Governance interface:

Security details

Assessment Status: Approved, With Conditions

You may also search for the component on the Security Guidance Portal or request an assessment.

Open Source Tooling Update

We've been rolling out a variety of engancements to our open source registration and review tools. The following are some key changes deployed this month.

What's New?

- Notifications:
  - Significantly improved reliability
  - Notifications now sent for registrations that did not require review
  - Notifications now sent when a review is re-assigned
- License discovery:
  - Previously completed reviews now considered as a source of license information
  - Automatic detection for Microsoft EULA licensed components
  - Improved license detection for Maven components

Coming Soon

- Ability to change the license or usage type for an in-progress review
- Public API for submitting registrations

We have also begun planning and design for registering contributions and for getting approval to release open source software.

Open Source Activity – Registrations

The following charts show an overview of open source component registration activity as of 10/17/2016.

Manual Reviews vs. No Review Required

Microsoft engineers have registered a total of 10,287 usages of open source components, and **only 902 of those (~9%) have required manual review**. Our goal is to continue driving the number of manual reviews down, to maximize scalability of the registration process and agility of engineering teams.

Total Registrations by Business Division

Growing public presence at https://opensource.microsoft.com/

Open Source Night

Monthly open source networking event

New for 2017: open to the public every 3rd time

Python FTW

A few examples of the use of Python
in the Open Source Programs Office


node
node
node
node
node
node
node
node
node
node
node

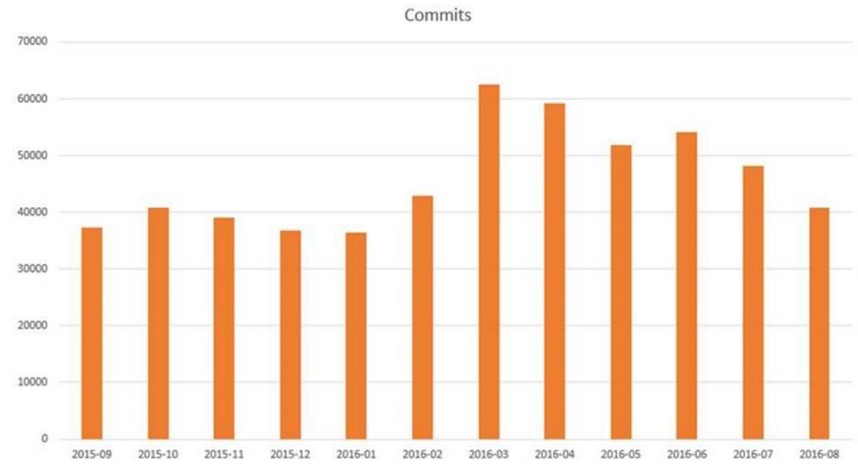
Getting GitHub Data

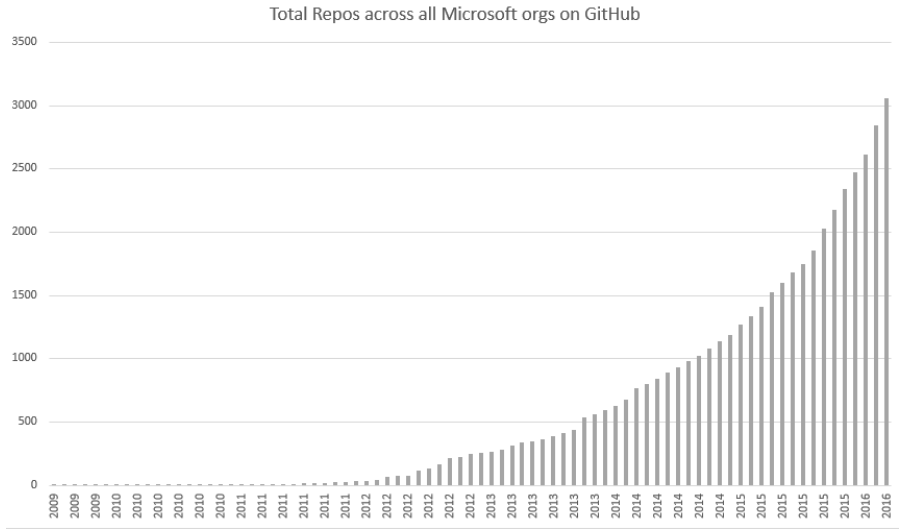
- Problem:
  - recurring need to get data from the GitHub APIs
- Solution:
  - command-line tool based on Requests and Click
  - Handles authentication, pagination, caching
- Next steps:
  - Add a ***commits*** subcommand
  - re-write to take advantage of the GraphQL API (released late 2016)

```
C:\>gitdata -h
Usage: gitdata [options] COMMAND [ARGS]...

  ------------------------------------
  Get information from GitHub REST API
  ------------------------------------
  syntax help: gitdata <subcommand> -h

Options:
  -a, --auth <str>   GitHub username (for configuring access)
  -t, --token <str>  store access token for specified username
  -d, --delete       delete specified username
  --version          Show the version and exit.
  -h, --help         Show this message and exit.

Commands:
  collabs  Get collaborator information for a repo
  members  Get member information by org or team ID
  orgs     Get org memberships for a user
  repos    Get repo information by org or user/owner
  teams    Get team information for an organization

C:\>
```

DEMO

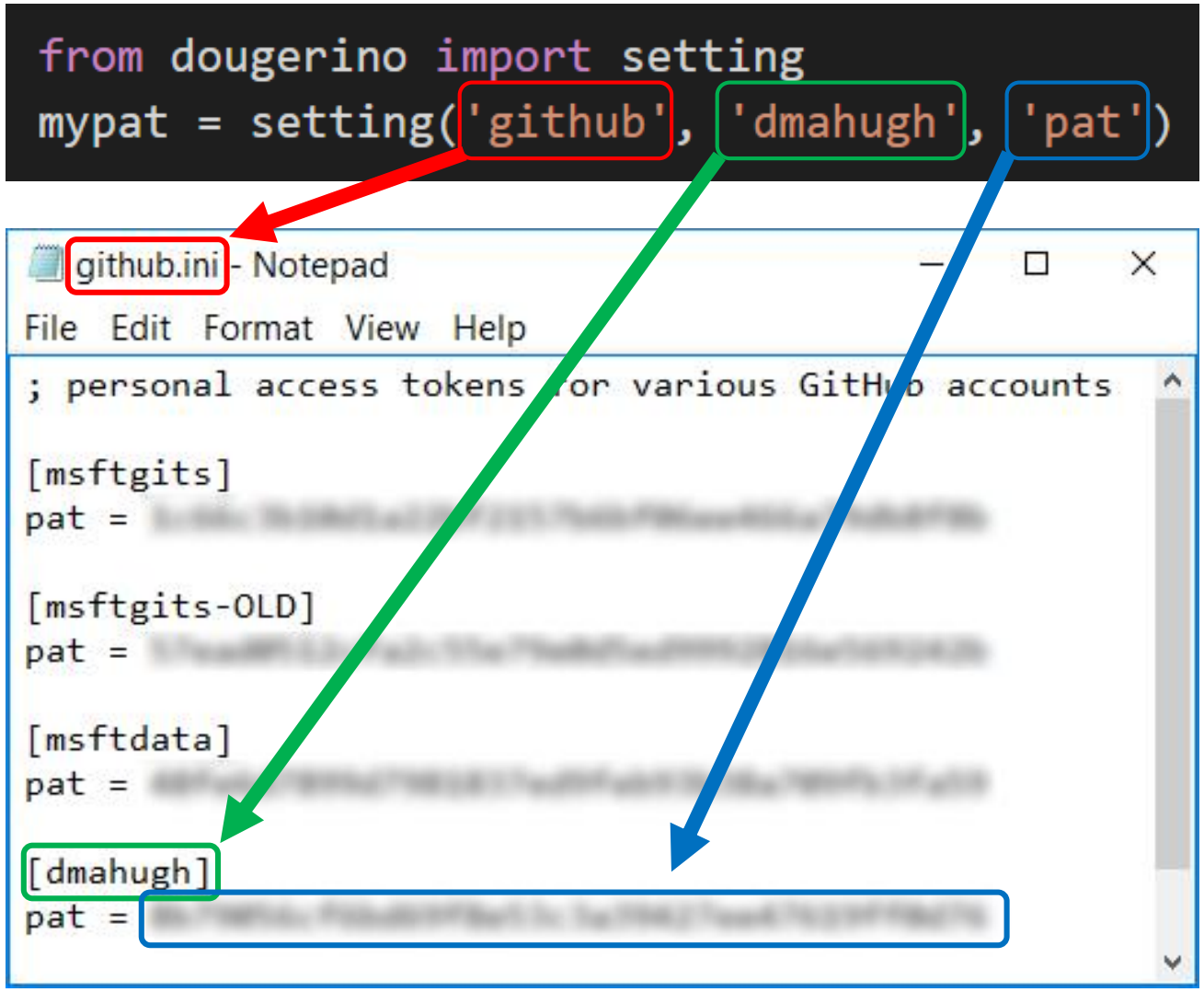
Sample charts created from GitHub data





Authentication

Personal access tokens for GitHub accounts are stored in github.ini in the **`..\_private`** folder.



Pagination – parsing the link header

- The GitHub V3 REST API returns paged results for most entities. (The new GraphQL API, on the other hand, returns *all* requested entities.)

```
from dougerino import github_pagination
link_header = \
    '<https://api.github.com/organizations/6154722/repos?page=2>; rel="next",' + \
    '<https://api.github.com/organizations/6154722/repos?page=98>; rel="last"'
github_pagination(link_header) # returns this dictionary ...

{
    'firstpage': 0, 'firstURL': None,
    'lastpage': '98',
    'lastURL': 'https://api.github.com/organizations/6154722/repos?page=98',
    'prevpage': 0, 'prevURL': None,
    'nextpage': '2',
    'nextURL': 'https://api.github.com/organizations/6154722/repos?page=2',
}
```

- The *github_pagination()* function converts a **`link`** HTTP header to a dictionary that can be used to navigate paged results.

Pagination – returning complete data sets

```
def github_data_from_api(endpoint, auth, headers):
    """Get complete data set from paginated GitHub REST API."""
    payload = []
    page_endpoint = endpoint # endpoint of the current page
    while True:
        response = github_api(page_endpoint, auth, headers)
        payload.extend(json.loads(response.text))
        pagelinks = github_pagination(response)
        if not page_endpoint:
            break # this is the last page, we're done
        else:
            page_endpoint = pagelinks['nextURL'] # next page
    return payload
```

Minimizing GitHub V3 API responses

GitHub response

```
github_response = github_api(endpoint='/repos/microsoft/typescript')
jsondata = json.loads(github_response.text)
```

6084 bytes

Remove fields named *url

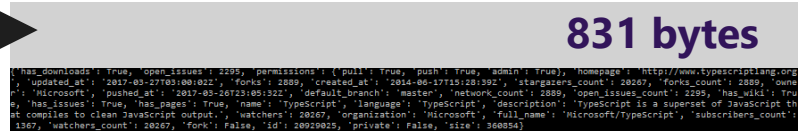
```
# remove fields named *_url, these aren't needed for analysis/reporting
no_urls = {key:value for (key, value) in jsondata.items()
           if not key.endswith('url')}
```

2661 bytes

Replace embedded entities with name only

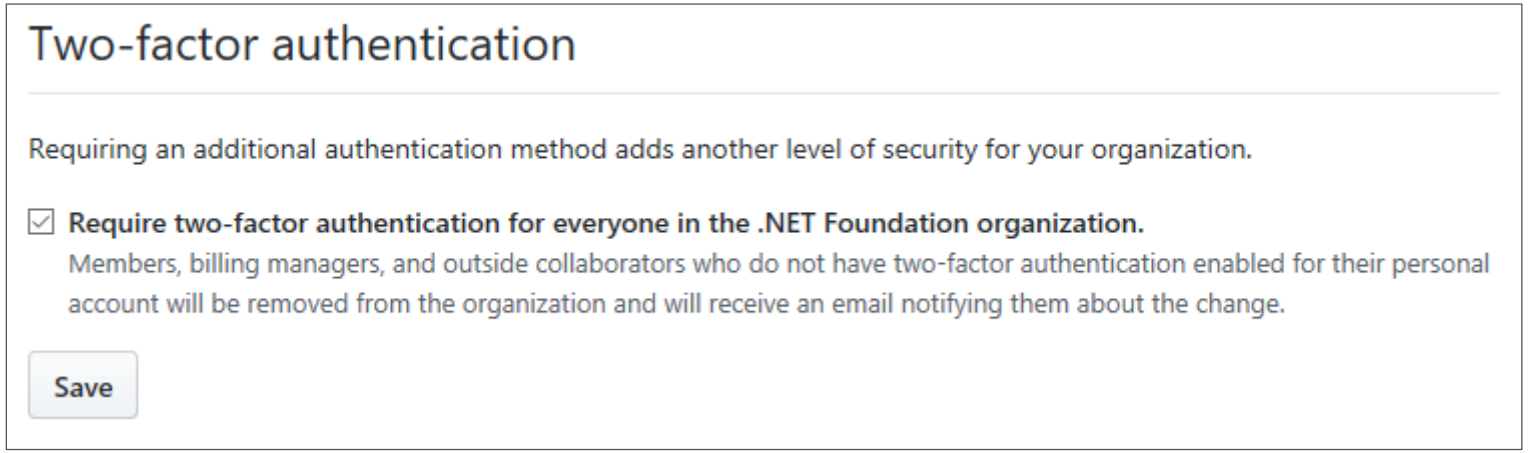
```
# replace embedded entites with just their name instead of a dictionary
logins_only = dict()
for (key, value) in no_urls.items():
    if isinstance(value, dict) and 'login' in value:
        logins_only[key] = value['login']
    else:
        logins_only[key] = value
```

831 bytes



GitHub 2FA requirement for Microsoft orgs

- Problem:
  - To begin enforcing 2FA requirement for all Microsoft orgs, need to
  - determine who will be affected and how to contact them.



- Solution:
  - Collect all available email addresses for members and collaborator without 2FA
- Result:
  - Identified 100% of accounts affected
  - Email addresses found for ~80% of 1530 org members
  - Email addresses found for ~40% of 1182 external collaborators

Determining who doesn't have 2FA

Organization ***members*** – there's an API call to get these ☺

Members list

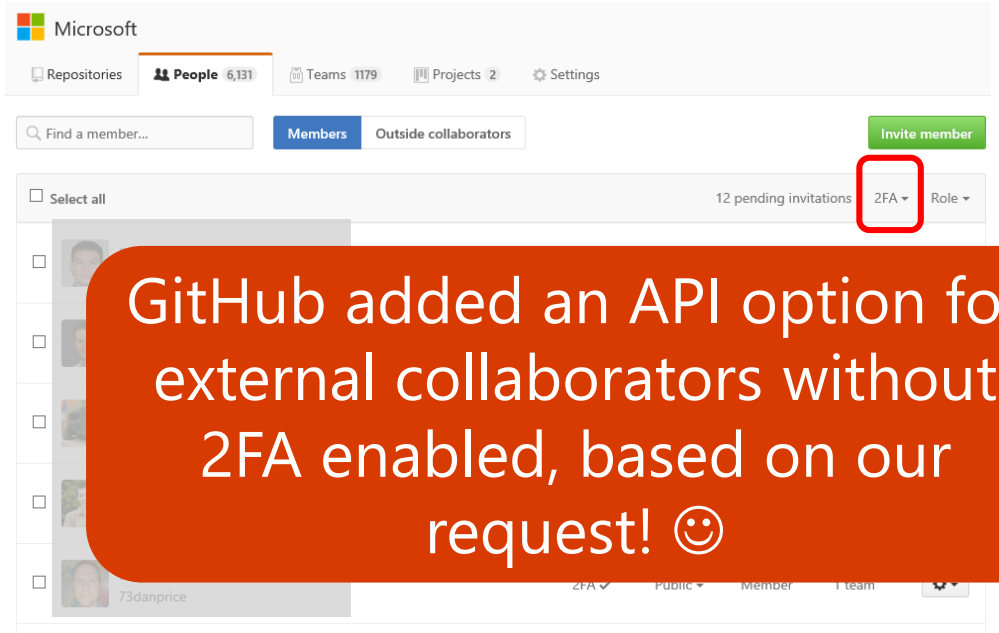
List all users who are members of an organization. If the authenticated user is also a member of this organization then both concealed and public members will be returned.

```
GET /orgs/:org/members
```

Parameters

Name	Type	Description
`filter`	`string`	Filter members returned in the list. Can be one of: * `2fa_disabled` : Members without two-factor authentication enabled. Available for organization owners. * `all` : All members the authenticated user can see. Default: `all`

Repo ***collaborators*** – had to scrape the web UI to get these ☹



GitHub added an API option for external collaborators without 2FA enabled, based on our request! ☺

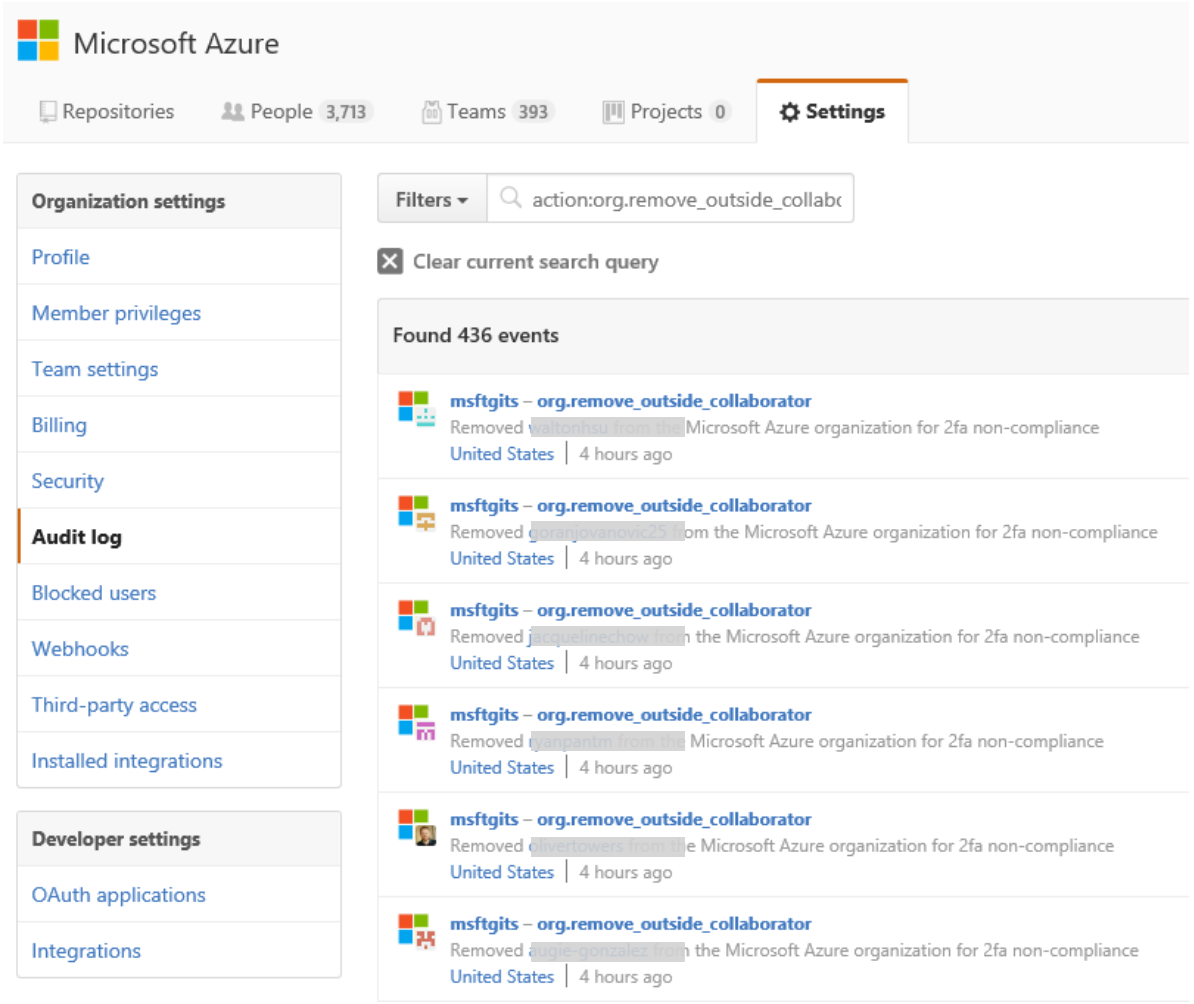
Determining email addresses for those affected

Email addresses were harvested from multiple sources

- Source #1: Linked Microsoft account?
- Source #2: Email appears in public GitHub profile?
- Source #3: In our CLA database?

- Worst case, look to commit history
  - We looked to last 90 days only
  - *Learned that some people do commits for many different email addresses*

GitHub's Audit Log



	action	actor	user	org
116063	org.remove_outside_collaborator	msftgits	ronharper	Azure
116065	org.remove_outside_collaborator	msftgits	boomiazure	Azure
116067	org.remove_outside_collaborator	msftgits	jmulchan	Azure
116069	org.remove_outside_collaborator	msftgits	Prsharma1	Azure
116071	org.remove_outside_collaborator	msftgits	sym-gaurav	Azure
116074	org.remove_outside_collaborator	msftgits	elting	Azure
116075	org.remove_outside_collaborator	msftgits	GaryWinter	Azure
116077	org.remove_outside_collaborator	msftgits	hsirk6	Azure
116079	org.remove_outside_collaborator	msftgits	CeliaMM	Azure
116080	org.remove_outside_collaborator	msftgits	jackxue1206	Azure
116082	org.remove_outside_collaborator	msftgits	SelvaDevaraj	Azure
116085	org.remove_outside_collaborator	msftgits	auxdocstest	Azure
116087	org.remove_outside_collaborator	msftgits	aravind-ca	Azure
116090	org.remove_outside_collaborator	msftgits	KunalLaudSymc	Azure
116092	org.remove_outside_collaborator	msftgits	saadansarithefirst	Azure
116094	org.remove_outside_collaborator	msftgits	sundeepkamathsymc	Azure
116096	org.remove_outside_collaborator	msftgits	tigranatqualys	Azure
116098	org.remove_outside_collaborator	msftgits	sharshavat	Azure
116100	org.remove_outside_collaborator	msftgits	gridpatrik	Azure
116103	org.remove_outside_collaborator	msftgits	serpop	Azure
116105	org.remove_outside_collaborator	msftgits	gsurapaneni	Azure
116108	org.remove_outside_collaborator	msftgits	TransVaultCTO	Azure
116109	org.remove_outside_collaborator	msftgits	boren-ms	Azure
116110	org.remove_outside_collaborator	msftgits	JasonWang-MS	Azure
116112	org.remove_outside_collaborator	msftgits	panzuradev	Azure
116115	org.remove_outside_collaborator	msftgits	kostiantynhontar	Azure
116118	org.remove_outside_collaborator	msftgits	pchaddha100	Azure
116120	org.remove_outside_collaborator	msftgits	tdazurebhavesh	Azure
116122	org.remove_outside_collaborator	msftgits	AkvelonValeriiRadchenko	Azure
116124	org.remove_outside_collaborator	msftgits	timhTD	Azure
116127	org.remove_outside_collaborator	msftgits	ibeyer	Azure
116129	org.remove_outside_collaborator	msftgits	anirudh-striim	Azure
116131	org.remove_outside_collaborator	msftgits	aaronlower-colab	Azure

63 Microsoft organizations on GitHub now have 2FA requirement enabled.

Organizations — New organization

6wunderkinder	owner	Leave
6wunderkinder-ops	owner	2FA Required · Leave
Azure	owner	2FA Required · Leave
Azure-Readiness	owner	2FA Required · Leave
Azure-Samples	owner	2FA Required · Leave
AzureAD	owner	2FA Required · Leave
AzureCAT-GSI	owner	2FA Required · Leave
CNTK-components	owner	2FA Required · Leave
ContosoDev	owner	2FA Required · Leave
ContosoLocal	owner	2FA Required · Leave
ContosoTest	owner	2FA Required · Leave
DynamicsCRM	owner	2FA Required · Leave
Glimpse	owner	2FA Required · Leave
InternetExplorer	owner	2FA Required · Leave
LIS	owner	2FA Required · Leave
MSOpenTech	owner	2FA Required · Leave
MSPowerBI	owner	2FA Required · Leave
Microsoft	member and collaborator on 11 repositories	2FA Required · Leave
Microsoft-CISL	owner	2FA Required · Leave
MicrosoftArchive	owner	2FA Required · Leave
MicrosoftDX	owner	2FA Required · Leave
MicrosoftDocs	owner	2FA Required · Leave
MicrosoftEdge	owner	2FA Required · Leave
MicrosoftResearch	owner	2FA Required · Leave
NuGet	member	Leave
OData	owner	2FA Required · Leave
OSTC	owner	2FA Required · Leave
OfficeDev	owner	2FA Required · Leave
OneDrive	owner	2FA Required · Leave
OneGet	member and collaborator on 1 repository	2FA Required · Leave
OneNoteDev	owner	2FA Required · Leave
PowerBI	owner	2FA Required · Leave
PowerShell	owner	2FA Required · Leave
SharePoint	owner	2FA Required · Leave
SignalR	owner	2FA Required · Leave
WindowsAzure	owner	2FA Required · Leave
WindowsAzure-Toolkits	owner	2FA Required · Leave
WindowsPhone-8-TrainingKit	owner	2FA Required · Leave
Z3Prover	owner	2FA Required · Leave
aspnet	owner	2FA Required · Leave
az-cat	owner	2FA Required · Leave
azure-appservice-samples	owner	2FA Required · Leave
bitstadium	owner	2FA Required · Leave
calabash	owner	2FA Required · Leave
contoso-a	owner	2FA Required · Leave
contoso-c	owner	2FA Required · Leave
contoso-d	owner	2FA Required · Leave
contoso-msft-m-dev	owner	2FA Required · Leave
contoso-x	owner	2FA Required · Leave
contoso-x-production	owner	2FA Required · Leave
deployr	owner	2FA Required · Leave
dotnet	owner	2FA Required · Leave
liveservices	owner	2FA Required · Leave
manifoldjs	owner	2FA Required · Leave
microsoft-hsg	owner	2FA Required · Leave
microsoft-mobile	owner	2FA Required · Leave
microsoft-notes	owner	2FA Required · Leave
microsoftgraph	owner	2FA Required · Leave
ms-iot	owner	2FA Required · Leave
msftberlin	owner	2FA Required · Leave
mspnp	owner	2FA Required · Leave
openT2T	owner	2FA Required · Leave
thaliproject	owner	2FA Required · Leave
winjs	owner	2FA Required · Leave
wunderlist	owner	2FA Required · Leave
yammer	owner	2FA Required · Leave

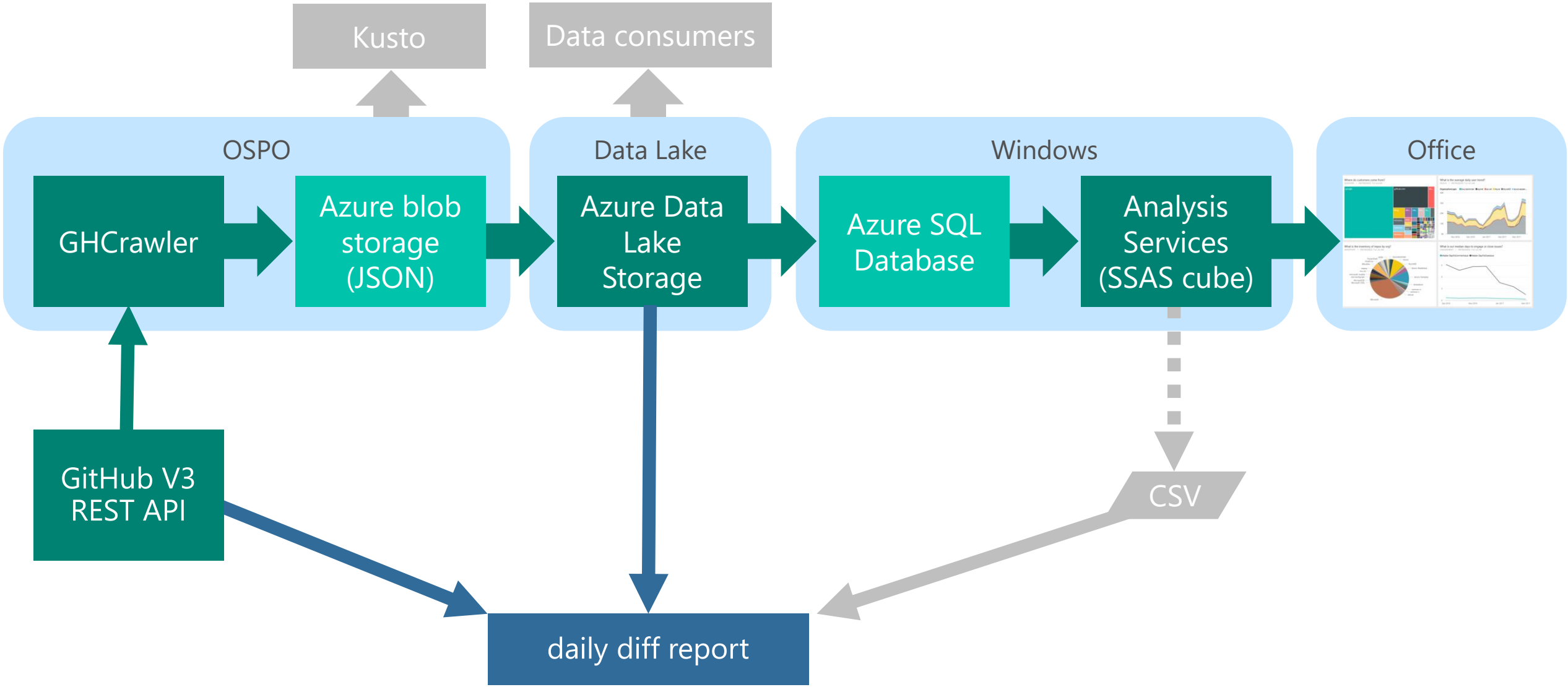
Verifying Data Integrity

- Problem:
  - We're capturing terabytes of GitHub data, and need an ongoing process for verifying the completeness of the data
- Solution:
  - Retrieve data from GitHub API and compare with Azure Data Lake Store
  - For entities with larger counts, compare counts by date range

```python
def github_commit_count(org, repo): #----------------------------------------<<<
    """Return total number of commits for specified org/repo.
    """
    endpoint = 'https://api.github.com/repos/' + org + '/' + repo + '/commits'
    requests_session = requests.session()

    # get first page of results
    firstpage = requests_session.get(endpoint,\
        headers={"Accept": "application/vnd.github.v3+json"})
    if not firstpage.ok:
        return str(firstpage) # 404 errors, etc.
    pagelinks = github_pagination(firstpage)
    json_first = json.loads(firstpage.text)
    pagesize = len(json_first) # of items on the first page of results
    totpages = int(pagelinks['lastpage'])
    lastpage_url = pagelinks['lastURL']

    if not lastpage_url:
        return pagesize # only one page of results, so we're done

    # get last page of results
    lastpage = requests_session.get(lastpage_url,\
        headers={"Accept": "application/vnd.github.v3+json"})
    json_last = json.loads(lastpage.text)
    lastpage_count = len(json_last) # number of items on the last page

    return (pagesize * (totpages - 1)) + lastpage_count
```

```
microsoft/dotnet, total commits = 616
microsoft/vscode, total commits = 16917
microsoft/typescript, total commits = 17014
microsoft/ghcrawler-datalake-etl, total commits = 26
```

GHInsights Processing Pipeline



GHInsights daily diff report - repos

```
---------- Data Verification for 2017-03-27 ----------
Download Repo.csv from Data Lake .......   6.2 seconds, 7,531,988 bytes
Get live data from GitHub API ..........  58.1 seconds, 400,630 bytes
Generate Repo_diff.csv .................   4.2 seconds, 12,305 bytes
                         Missing:       5 Repos
                         Extra:       352 Repos
                         Mismatch:      3 Repos
Upload Repo_diff to Data Lake ..........   3.6 seconds
                       REPO - elapsed time:  72.1 seconds
```

org	repo	issue
CNTK-COMPONENTS	cntk1bitsgd	missing
CONTOSODEV	empty-npm	missing
GLIMPSE	glimpse.docs-vscode	missing
GLIMPSE	glimpse.website-vscode	missing
MICROSOFTDOCS	docs-internal-test	missing
GLIMPSE	glimpse.website	mismatch
MICROSOFT	graphengine	mismatch
OFFICEDEV	microsoft-teams-sample-custombot	mismatch

Thank You!

Q&A

Microsoft

https://github.com/dmahugh/pyladies-march2017