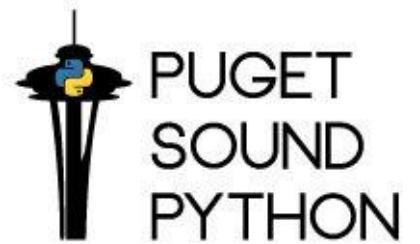


Open Source @ Microsoft

Doug Mahugh
Open Source Programs Office



Wed 3/29 6:00PM
Apptio – Bellevue, WA

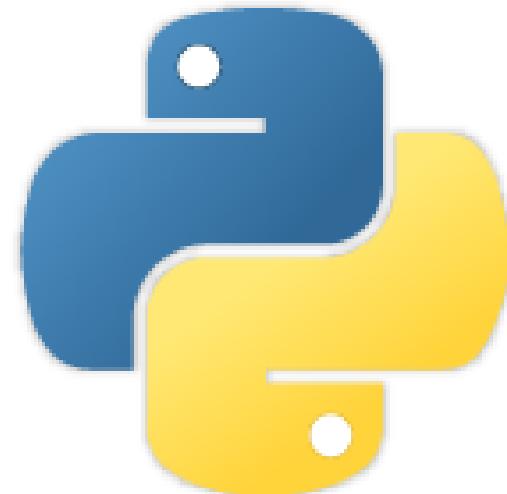
Agenda

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.

ig 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
P-I Reporter
Tuesday and decided I could take on the guy's record," the 23-year-old South Seattle man said yesterday. "I spent 16 hours and 36 minutes on a quarter investment in the Defender machine. I went with an eye on a physics degree and a career in computers, became a software engineer."

Do you have a "drawing" program that can't?

Take a good look at this photograph. Can you do this with the "drawing" program you have now? If not, maybe it's time to move up to the only true drawing program for your Kaypro—SCS-Draw.

SCS-Draw turns your mild-mannered Kaypro into a powerful drawing machine, with features that no other program can offer:

- True pixel-by-pixel drawing. (Not "building-block" character drawing.)
- A total resolution of over 120,000 pixels. (Over seven times the resolution of other Kaypro drawing programs.)
- Built-in patterns, with one of 23 built-in patterns or a pattern that you design.
- Powerful printing options like enlargement/reduction (as shown above), rotation and mirror image, and more.
- Pop-up menus, automatic on-line help, and much, much more.

So if you have a "drawing" program that can't draw a picture, draw a logo, draw a banner, make a sign, draw a map, illustrate a newsletter or print a party invitation, don't get mad—get SCS-Draw, the only true drawing program for 84, 85 and '86 Kaypro CP/M computers.

S E C O N D C I T Y S O F T W A R E

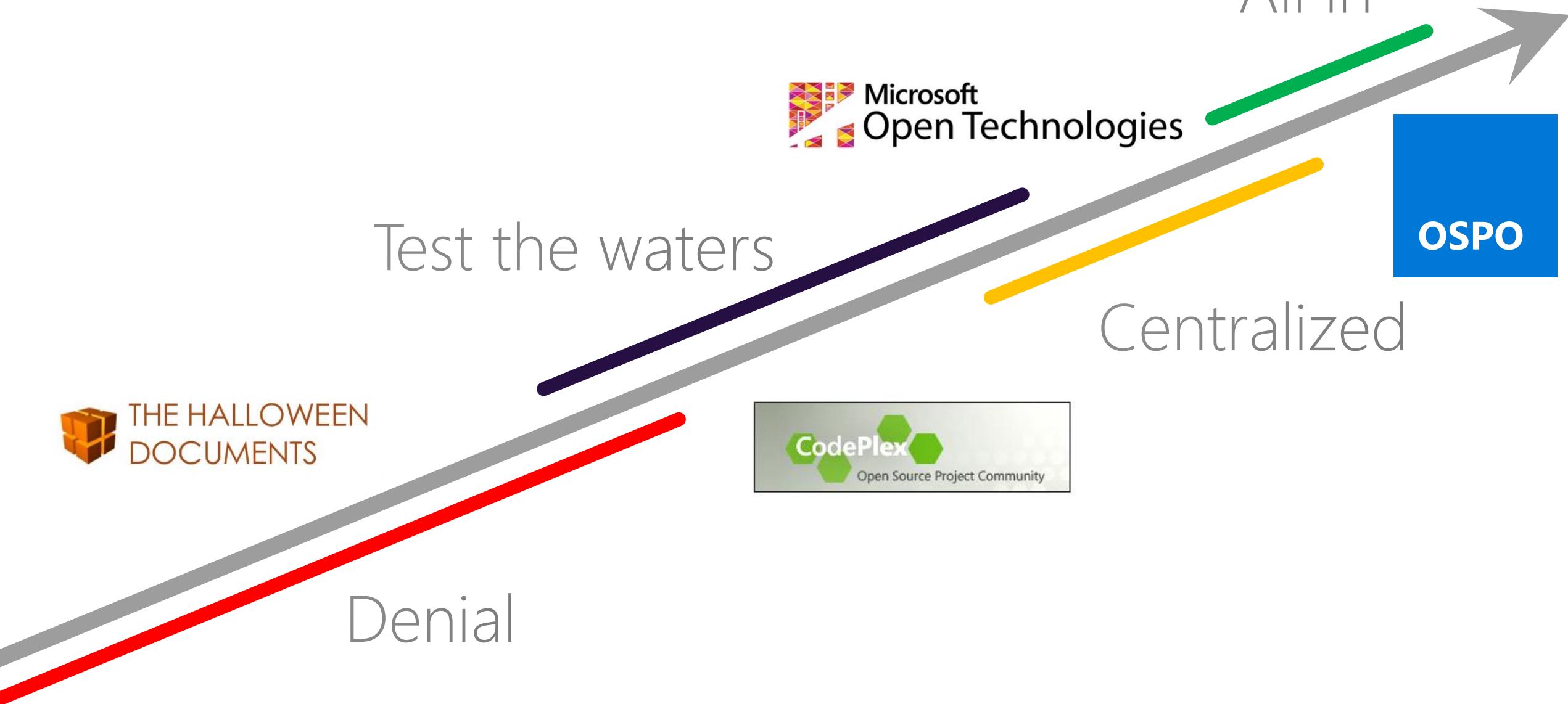


Open Source @ Microsoft

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



Evolution



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

Culture change

Building a sustainable
open source culture
at Microsoft



Scenarios



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



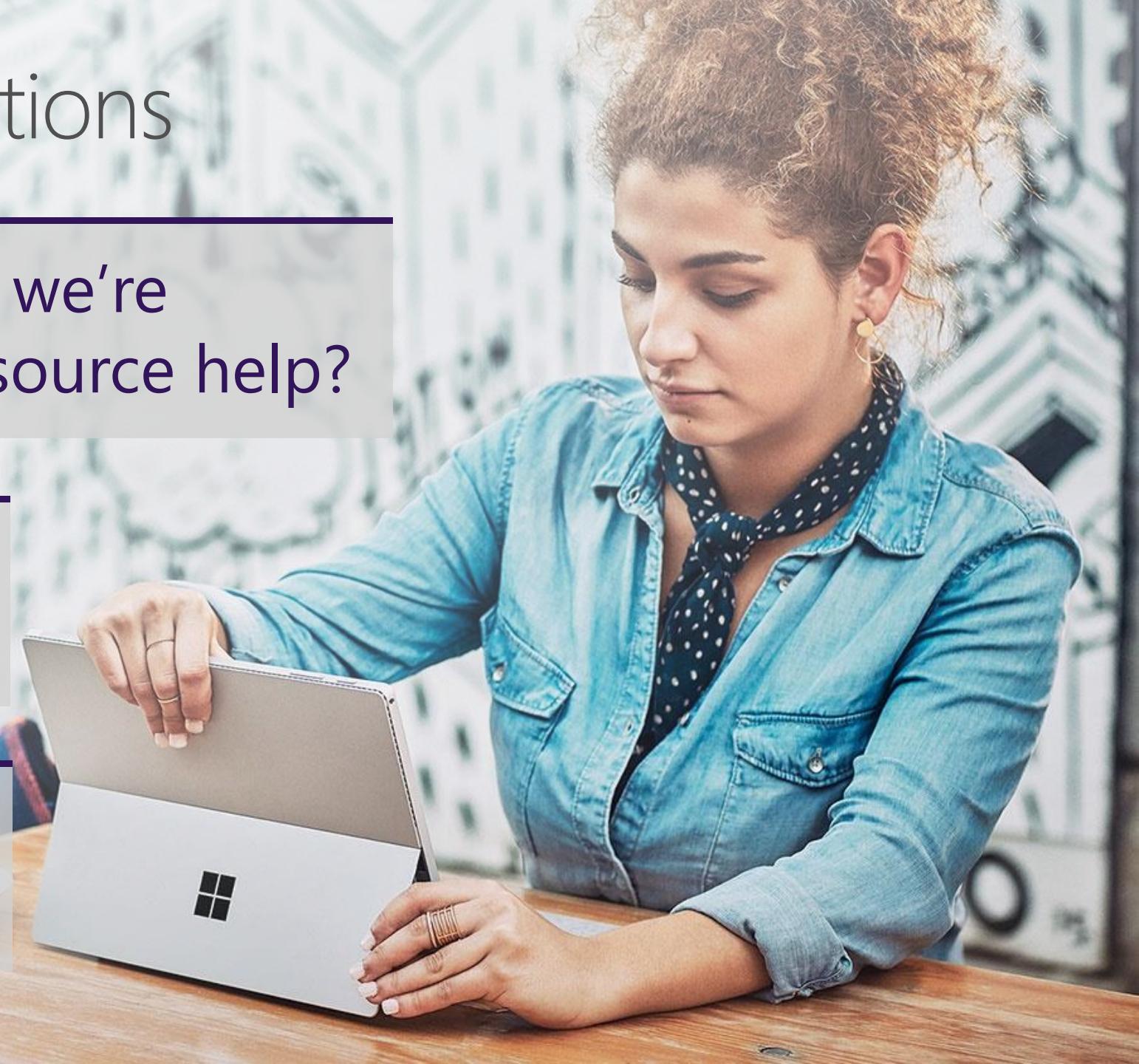
Develop strategic relationships

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
*reducing
friction.*

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

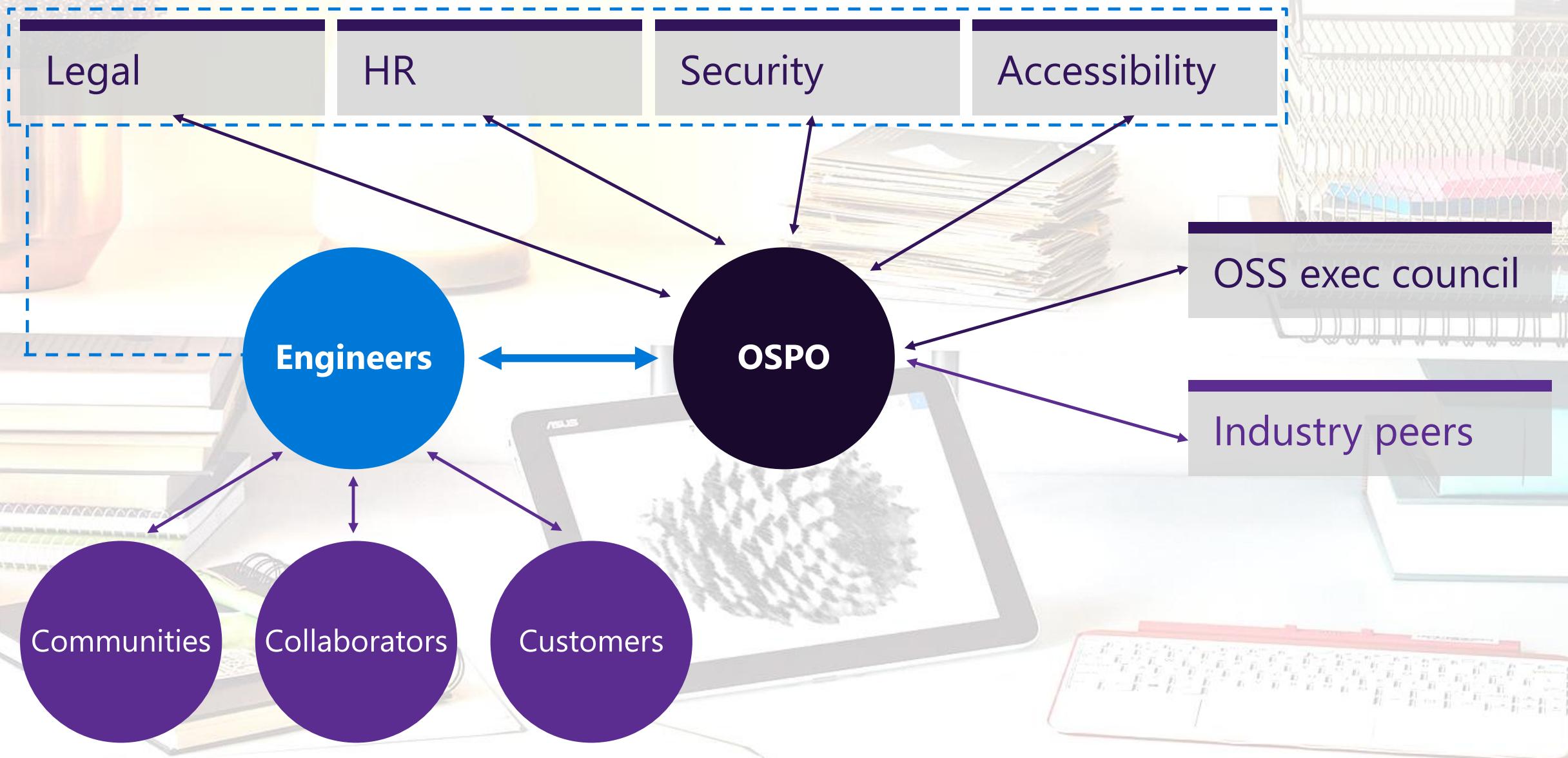
GitHub user
requirements (2FA,
linked identity, etc.)

Contributor license
agreements

Code of conduct

Accessibility

Facilitating communication



OSPO tools for GitHub management

Standardized, consistent approach to creating repos and managing teams

This screenshot shows the Microsoft Open Source GitHub interface. The top navigation bar includes links for Microsoft Open Source, Explore, Use, Release, GitHub (which is selected), and About. Below the navigation is a blue header bar with links for Organizations, Microsoft, Repos, Teams, and People. The main content area is titled "Microsoft Organization" and contains four prominent buttons: "Create a repo" (REQUEST A NEW REPO), "Join a team" (REQUEST ACCESS), "Add a team" (CREATE NEW TEAM), and "Public member" (PUBLICIZE MEMBERSHIP). Below these buttons is a section titled "Microsoft Teams You Maintain" which lists two teams: "ospo-protected" and "ospo-testing". Each team entry has a "Manage" button. At the bottom, there is a section titled "Team Memberships" with a note: "Here are teams that you are a member of but do not maintain.".

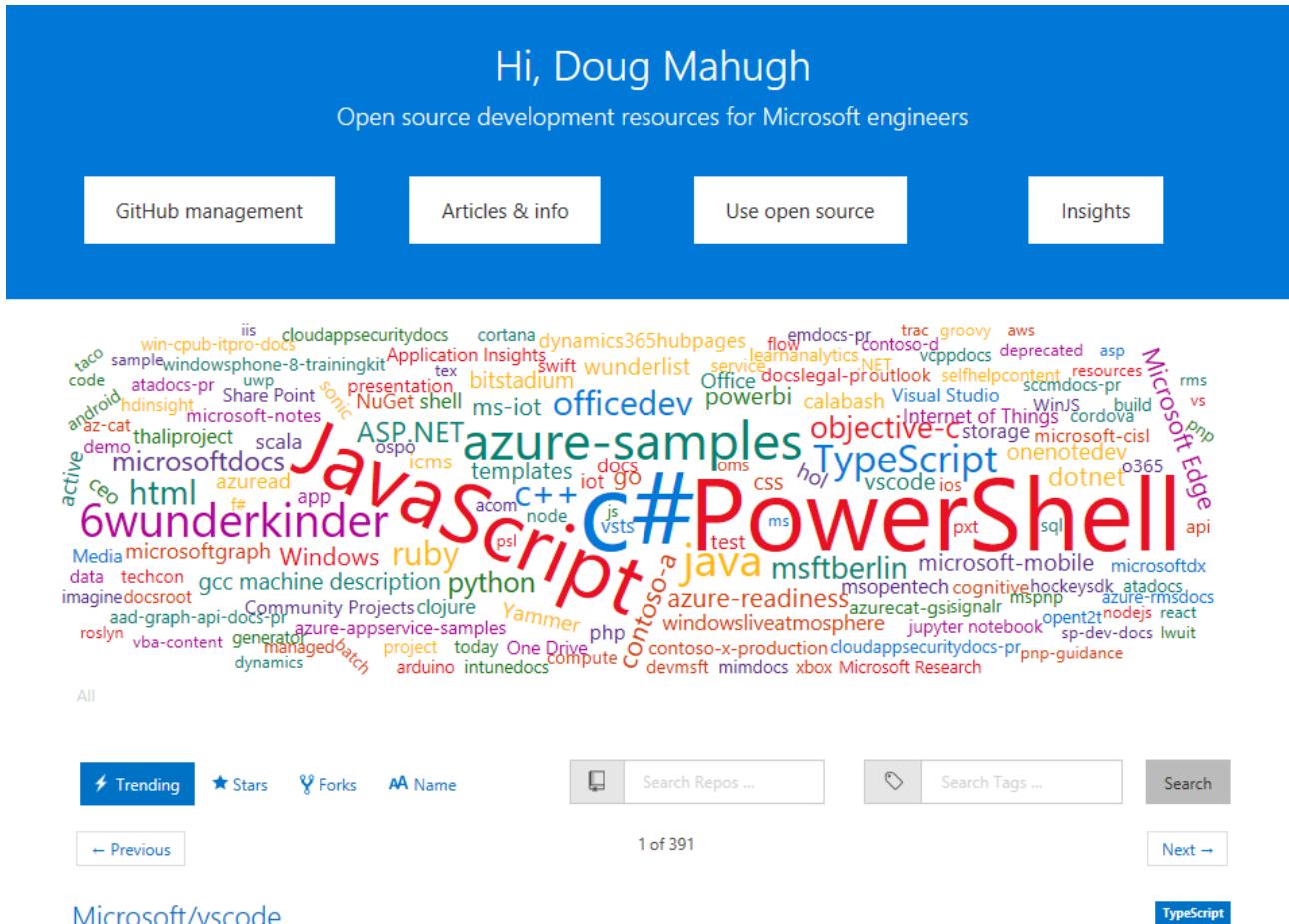
This screenshot shows the Microsoft Open Source GitHub interface. The top navigation bar includes links for Microsoft Open Source, Explore, Use, Release, GitHub (selected), and About. Below the navigation is a blue header bar with links for Organizations, Repos, Teams, and People. The main content area is titled "Repositories" and displays a search bar with filters for "Recent commits", "Stars", "Forks", "Name", "Updated", and "Created date". The search bar also includes fields for "Search repositories..." and "Search". Below the search bar, it says "Teams: myadmin". The results section shows "100 results for public repositories and my admin team permissions". It lists two repositories: "ospo-ghcrawler" (GitHub crawler, JavaScript, 4 forks, updated 2 hours ago, created 4 months ago) and "service-fabric-dotnet-data-aggregation". To the right of the results, there is a sidebar titled "Organizations" which lists several Microsoft-owned GitHub repositories under the "Azure" organization, such as "Azure", "Azure-AppService-Samples", "Azure-Readiness", "Azure-Samples", "AzureAD", "AzureCAT-GSI", "CNTK-components", and "DotNet". A note on the sidebar states: "Need to create a repo? To create a new repo, first you need to select which organization will host it."

<https://opensource.microsoft.com/>

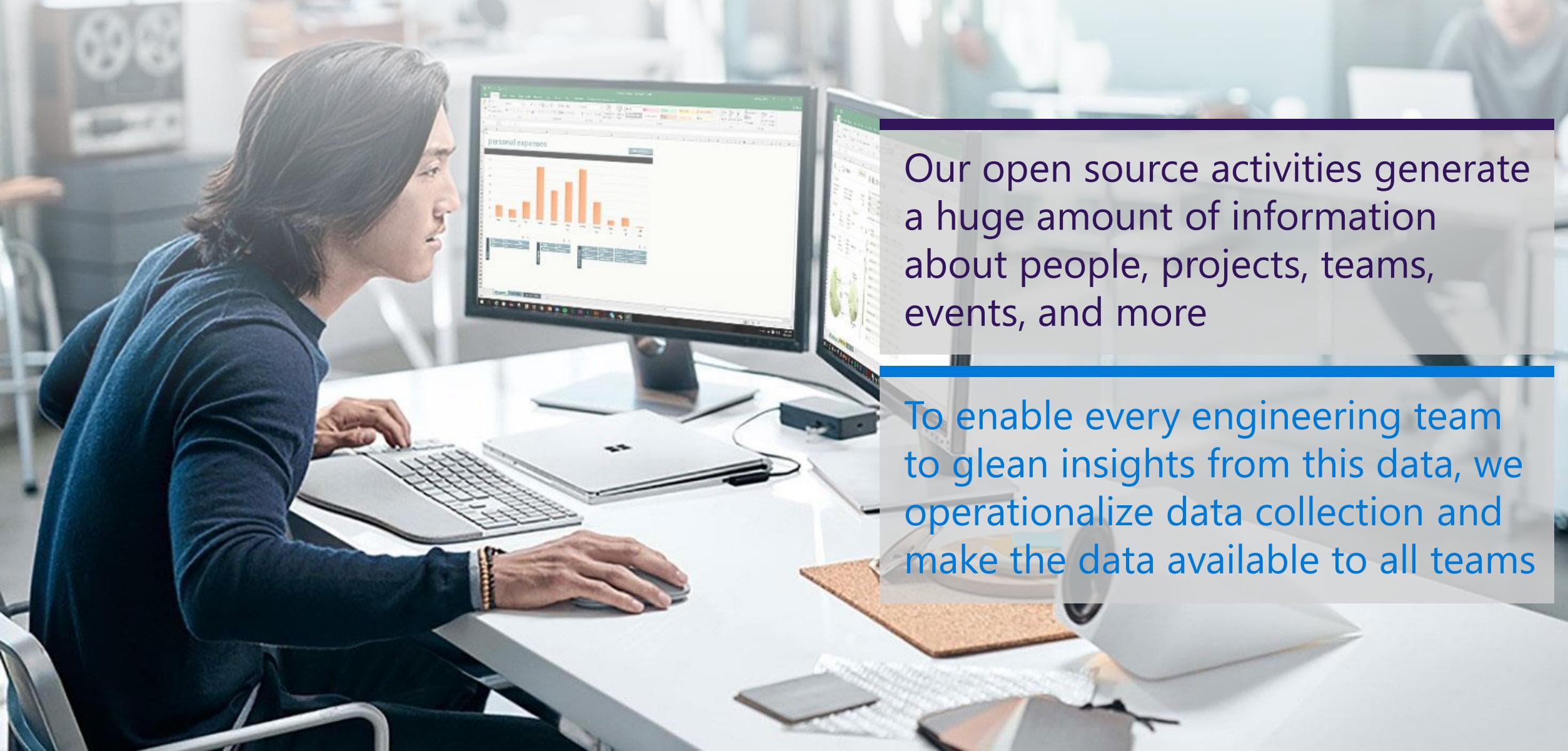
Home page for open source engineering at Microsoft

Tag cloud and related tools for repo discoverability (prototyped in Python, of course!)

Additional functionality provided for Microsoft corporate identities



Data collection and analysis

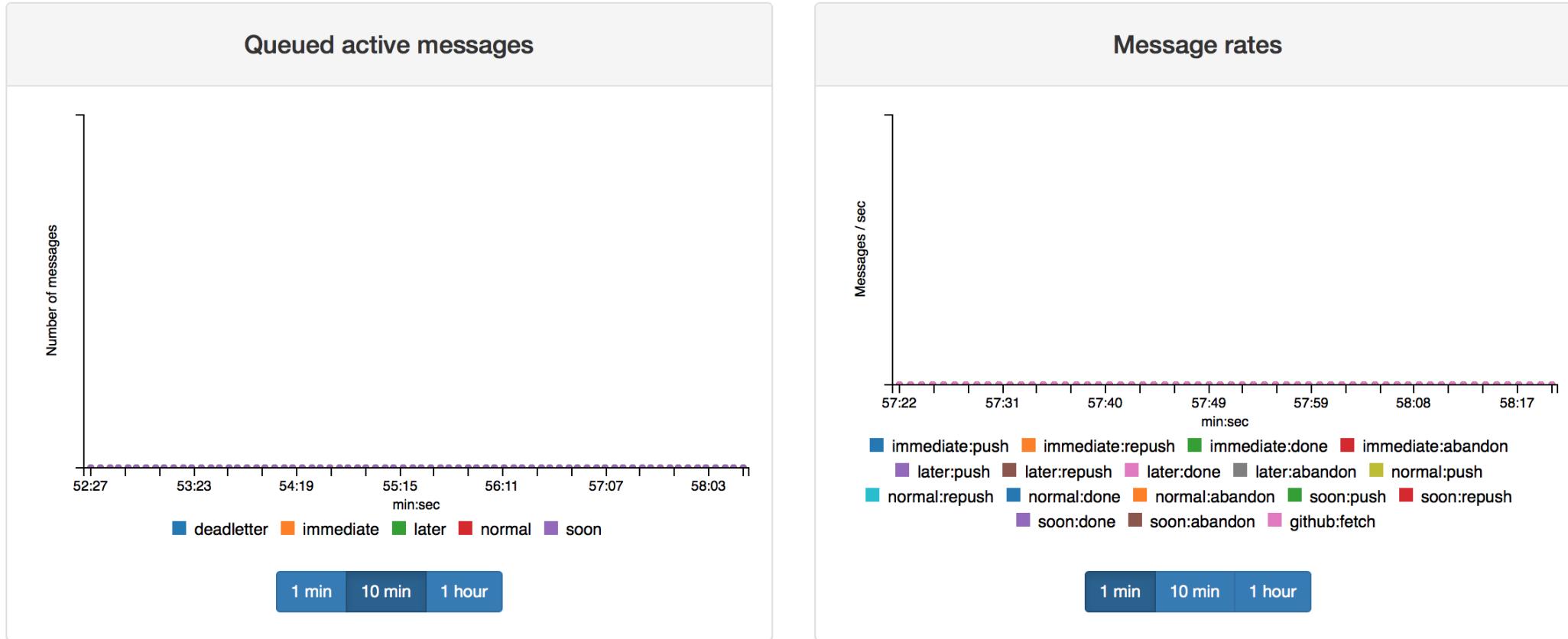


Our open source activities generate a huge amount of information about people, projects, teams, events, and more

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

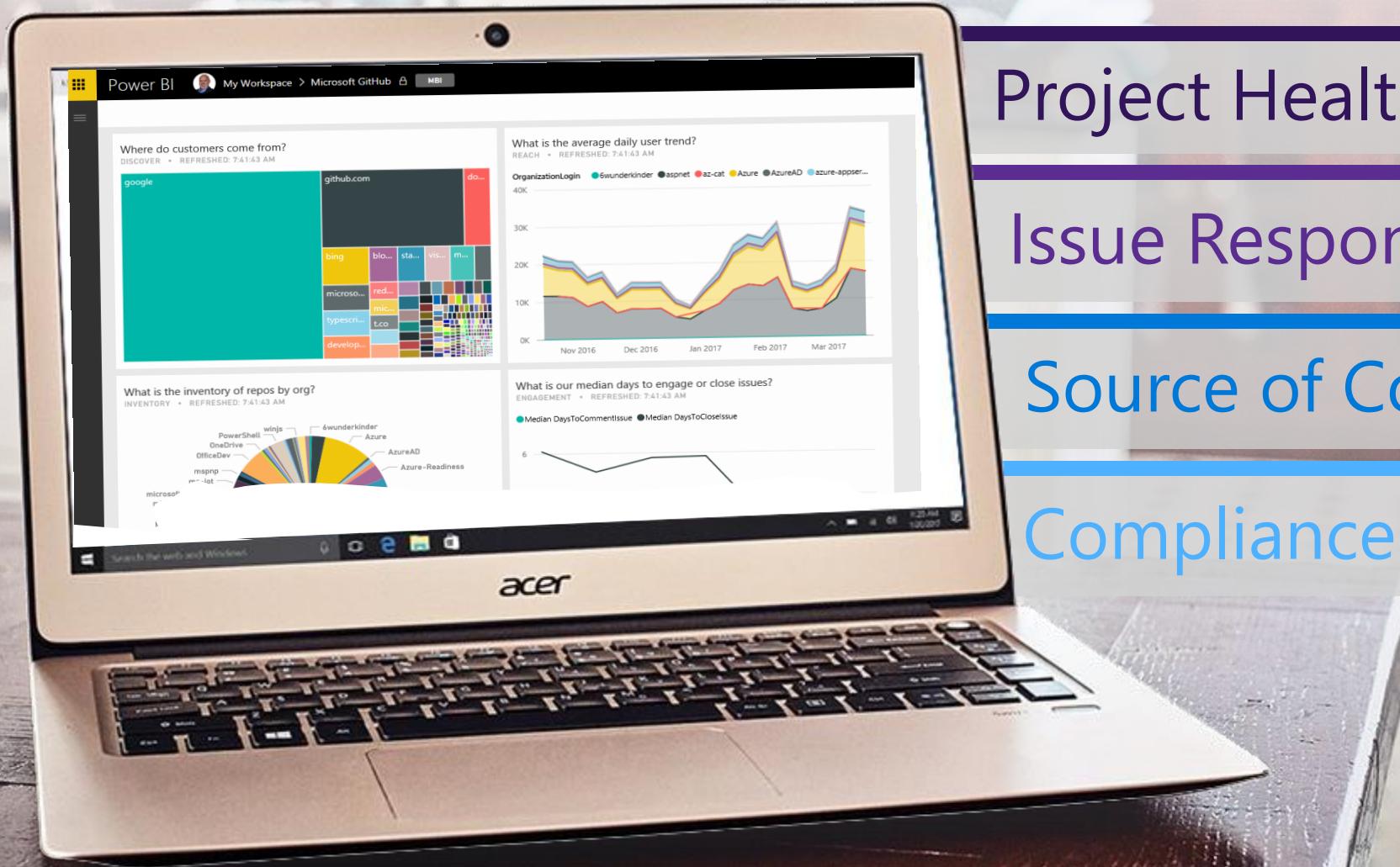
Crawler in a box

Crawler Dashboard



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

GitHub Insights



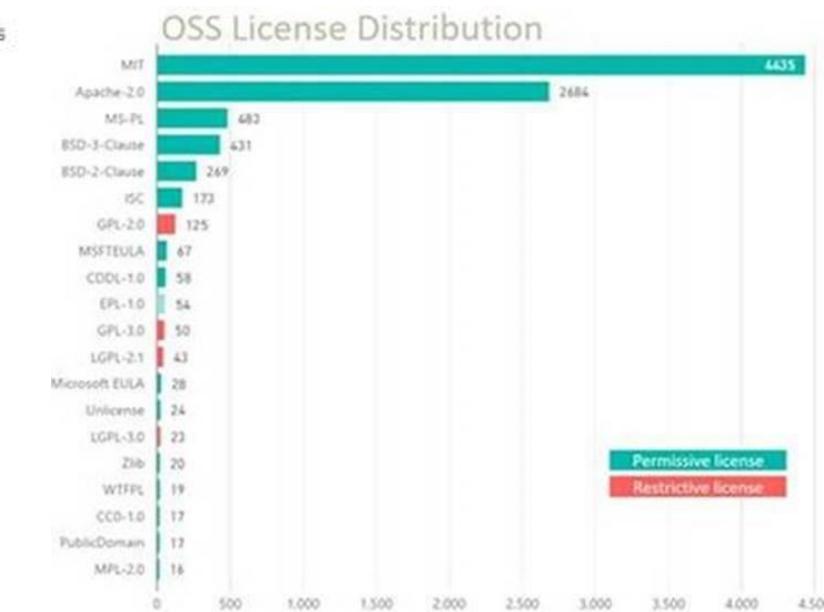
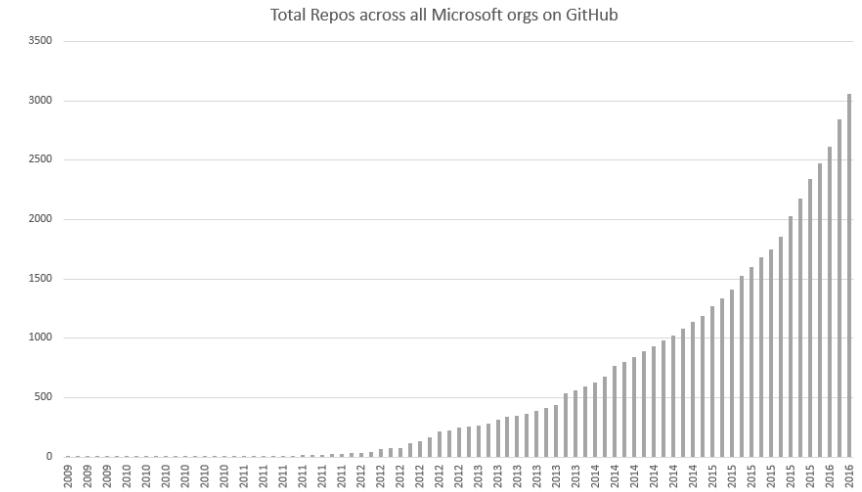
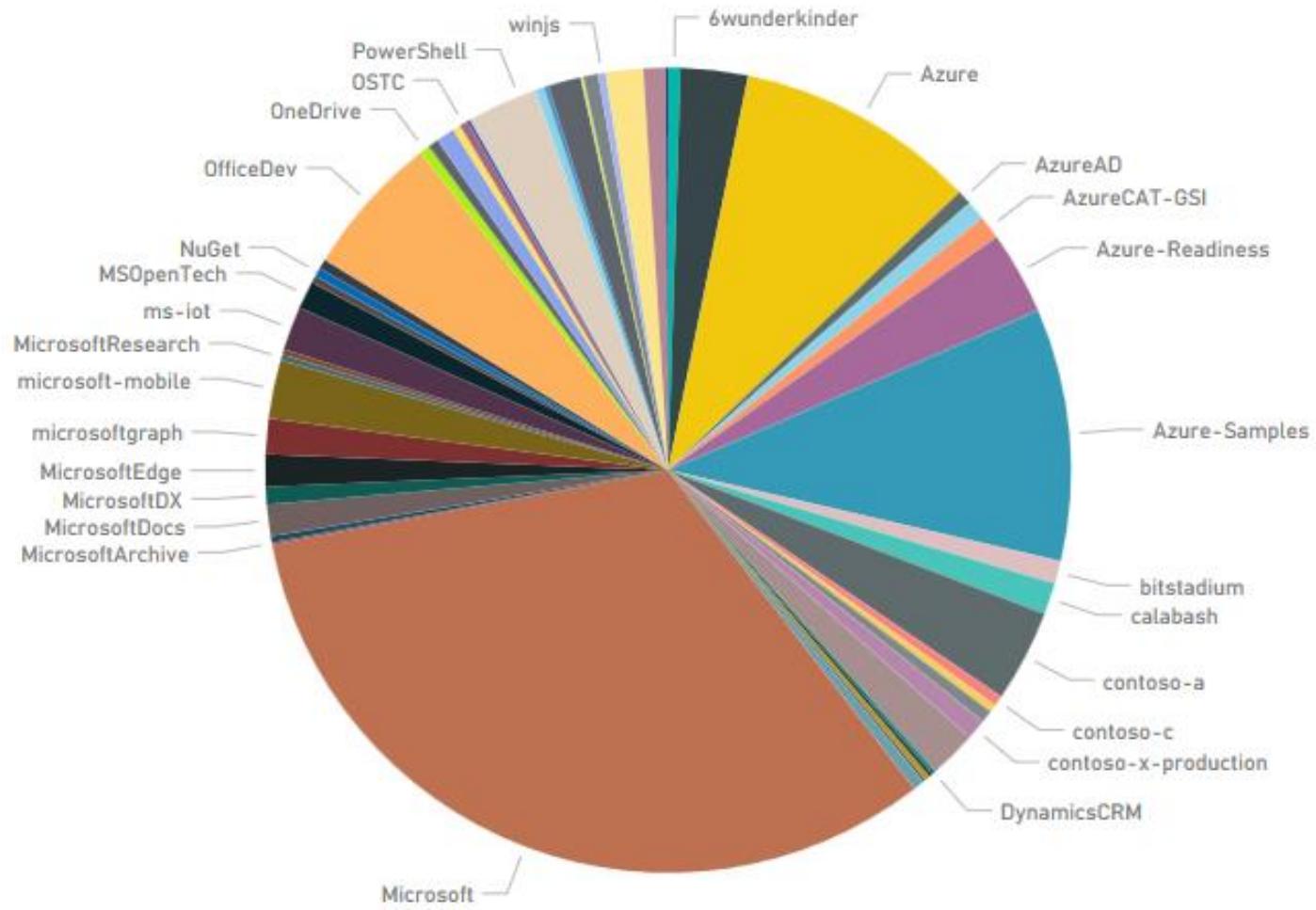
Project Health

Issue Response Time

Source of Contributions

Compliance

Sample charts created from GitHub data



Governance Insights

Power BI OSPO > Witness Reporting MBI

Ask a question about your data

+ Add tile View related Favorite Share ...

OSS Use Registrations Status

Total Registrations **22932**

Total	In Progress	Stale	Abandoned	Finalized
1780	564	524	150	1216

Manual Reviews

Automatically Processed **21148**

Goto Full Report

Registrations by Team

- WDG - Universal Store
- ASG - Yammer
- C+E - Developer Division
- C+E - Business Applications, Platform...
- C+E - Data Platform Group
- ASG - Office Experience Org
- C+E - Enterprise Cloud Group
- WDG - Core Dev
- C+E - CRM
- ASG - Outlook & O365 (including E...
- Corp Strategy & Planning
- C+E - Mobile Developer Tools
- C+E - Open Source Programs Office
- C+E - Data Group

Team	WDG - Universal Store	C+E - Business Ap...	C+E - C...	ASG - ...	Corp S...	C+E - ...
WDG - Universal Store	2518	1648	852	793	779	689
ASG - Yammer	1039	C+E - Data Platfor...	C+E - Ope...	ASG...	AS...	W...
C+E - Developer Division	1020	ASG - Office Exper...	C+E - Dat...	MB...	AS...	AI...
Corp Strategy & Planning	996	C+E - Enterprise C...	564	308	283	
C+E - Data Group	948	WDG - Core Dev	AI+R - Inf...	OPG - Mic...		

OSS License Type

License Type	Count
multiple	0K
network	0K
permissive	21K
strong	0K
unknown	1K
weak	0K

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



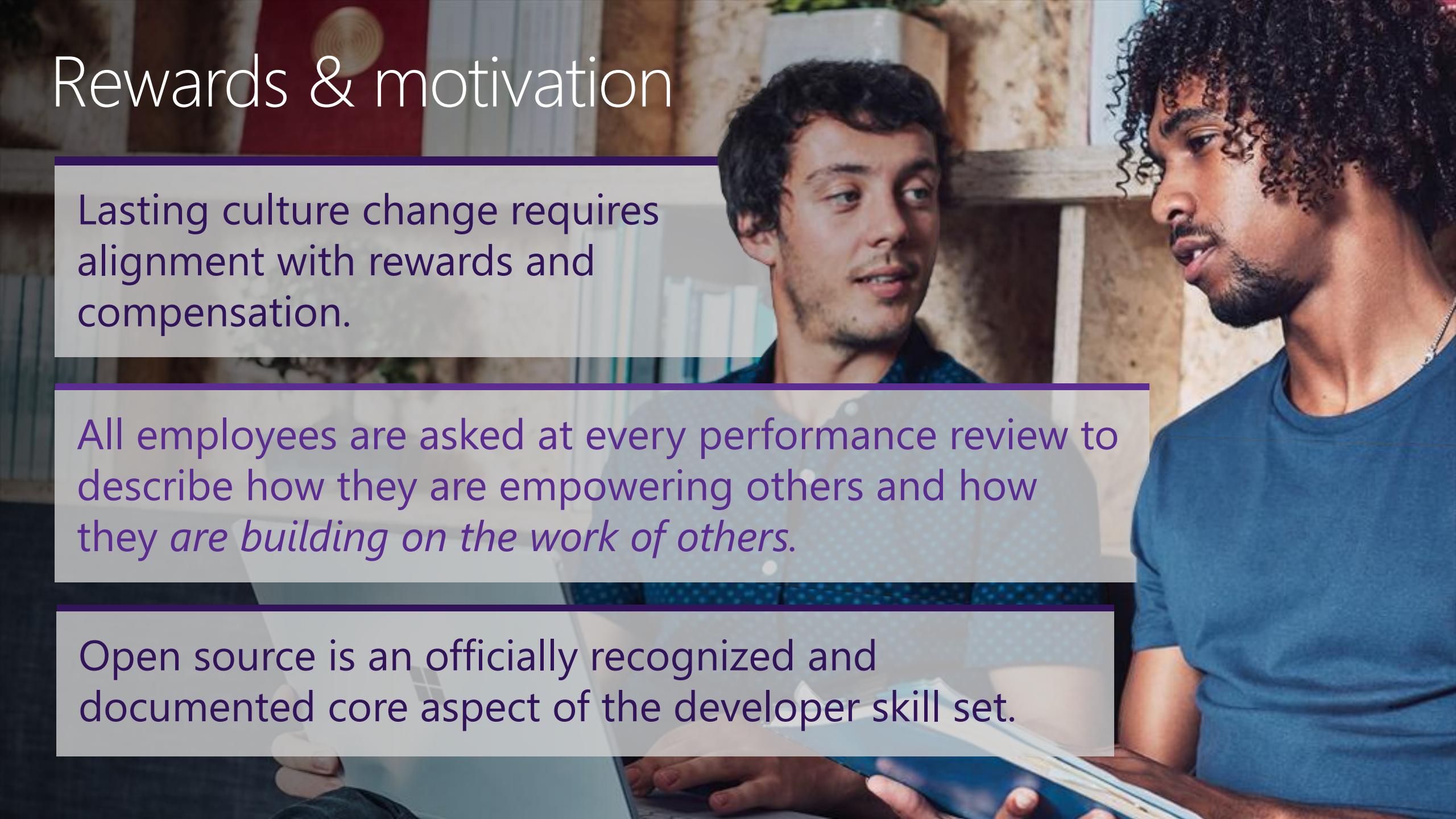
Inner sourcing

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.

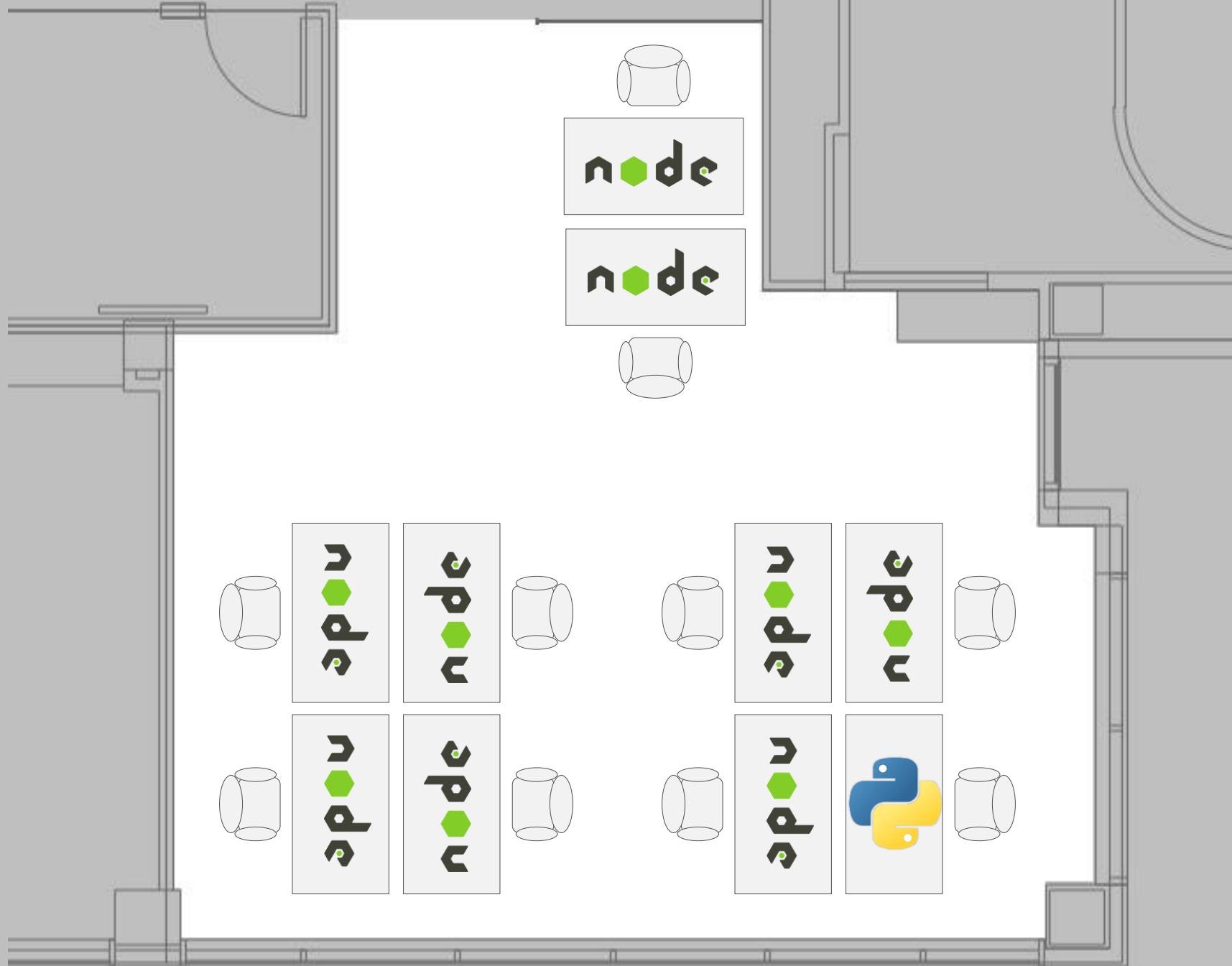


Open Source Night

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



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

Authentication

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

The diagram illustrates the mapping between a Python code snippet and its corresponding configuration in a `github.ini` file. A red box highlights the string `'github'` in the Python code, which points to the `[github]` section in the `github.ini` file. A green box highlights the string `'dmahugh'`, which points to the `[dmahugh]` section. A blue box highlights the string `'pat'`, which points to the `pat` value under the `[dmahugh]` section. The Python code is as follows:

```
from dougerino import setting
mypat = setting('github', 'dmahugh', 'pat')
```

The `github.ini` file content is:

```
github.ini - Notepad
File Edit Format View Help
; personal access tokens for various GitHub accounts

[msftgits]
pat = [REDACTED]

[msftgits-OLD]
pat = [REDACTED]

[msftdata]
pat = [REDACTED]

[dmahugh]
pat = [REDACTED]
```

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 entities 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

`'has_downloads': True, 'open_issues': 2295, 'permissions': {'pull': True, 'push': True, 'admin': True}, 'homepage': 'http://www.typescriptlang.org', 'updated_at': '2013-03-27T03:00:00Z', 'forks': 2889, 'created_at': '2014-06-17T75:28:39Z', 'stargazers_count': 2067, 'forks_count': 2889, 'name': 'Microsoft', 'pushed_at': '2017-05-26T23:05:32Z', 'default_branch': 'master', 'network_count': 2889, 'open_issues_count': 2295, 'has_wiki': True, 'language': 'JavaScript', 'url': 'https://github.com/Microsoft/TypeScript', 'description': 'TypeScript is a superset of JavaScript that compiles to clean JavaScript output.', 'subscribers_count': 1987, 'watchers_count': 2067, 'fork': False, 'id': 39282959, 'private': False, 'size': 3080541}`

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

Two-factor authentication

Requiring an additional authentication method adds another level of security for your organization.

Require two-factor authentication for everyone in the .NET Foundation organization.
Members, billing managers, and outside collaborators who do not have two-factor authentication enabled for their personal account will be removed from the organization and will receive an email notifying them about the change.

Save

Determining who doesn't have 2FA

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

Members list ⓘ

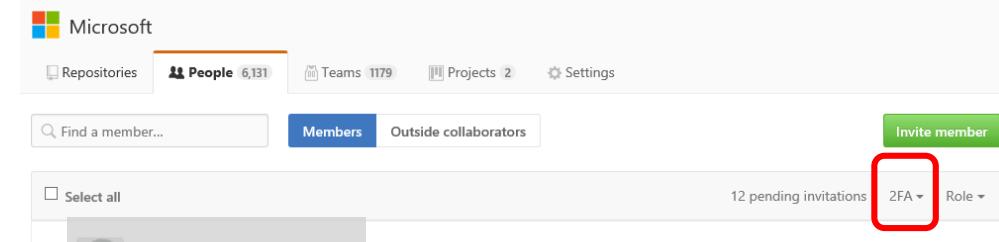
List all users who are members of an organization. If the authenticated user is also a member of this organization then both concealed and public members will be returned.

GET /orgs/:org/members

Parameters

Name	Type	Description
filter	string	Filter members returned in the list. Can be one of: * <code>2fa_disabled</code> : Members without two-factor authentication enabled. Available for organization owners. * <code>a11</code> : All members the authenticated user can see. Default: <code>a11</code>

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?
- If all else fails, look to commit history
 - Looked to last 90 days only
 - *Learned that some people do commits for many different email addresses*

GitHub's Audit Log

Microsoft Azure

Repositories 3,713 Teams 393 Projects 0 Settings

Organization settings Profile Member privileges Team settings Billing Security **Audit log** Blocked users Webhooks Third-party access Installed integrations

Filters ▾ action:org.remove_outside_collaborator

Clear current search query

Found 436 events

msftgits – org.remove_outside_collaborator
Removed [valonharper](#) from the Microsoft Azure organization for 2fa non-compliance
United States | 4 hours ago

msftgits – org.remove_outside_collaborator
Removed [jasonwang-ms](#) from the Microsoft Azure organization for 2fa non-compliance
United States | 4 hours ago

msftgits – org.remove_outside_collaborator
Removed [jacquelinechow](#) from the Microsoft Azure organization for 2fa non-compliance
United States | 4 hours ago

msftgits – org.remove_outside_collaborator
Removed [yanpanm](#) from the Microsoft Azure organization for 2fa non-compliance
United States | 4 hours ago

msftgits – org.remove_outside_collaborator
Removed [olivertowers](#) from the Microsoft Azure organization for 2fa non-compliance
United States | 4 hours ago

msftgits – org.remove_outside_collaborator
Removed [jorgegonzalez](#) from the Microsoft Azure organization for 2fa non-compliance
United States | 4 hours ago

action	actor	user	org
116063 org.remove_outside_collaborator	msftgits	ronharper	Azure
116065 org.remove_outside_collaborator	msftgits	boomiazeure	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	auxdochest	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.



Organization	Owner	2FA Required	Leave
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 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
thailiproject	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

```
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,\n        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']

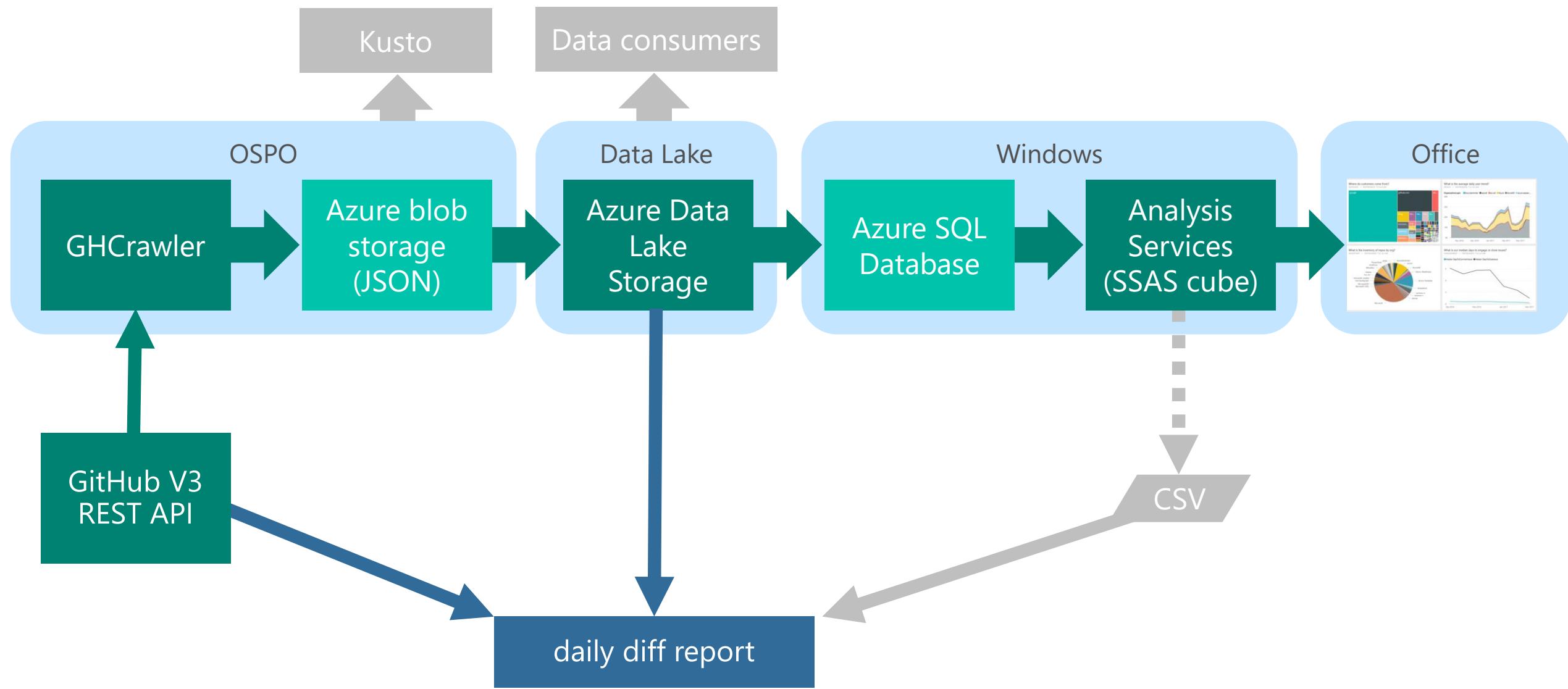
    if not lastpage_url:
        return pagesize # only one page of results, so we're done

    # get last page of results
    lastpage = requests_session.get(lastpage_url,\n        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



© Copyright Microsoft Corporation. All rights reserved.

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