

Supplementary Material

Normalization Rules, Audit Evidence, and Topology Construction for the Unified OAuth Capability Dictionary

August 18, 2026

Abstract

This supplement documents the complete rule-based normalization and topology-construction process used in the accompanying manuscript. It provides the executable rule specification, released-artifact distributions, representative mappings from every provider, difficult and ambiguous cases, edge-construction logic, cross-provider bridge examples, community and centrality summaries, reproducibility instructions, and explicit interpretive boundaries. The purpose is to make every semantic assignment and structural relation auditable without expanding the page-limited main article.

Contents

1	Scope and relationship to the main article	3
2	End-to-end workflow	3
3	Unified record schema	3
4	Complete normalization rule specification	4
4.1	Text normalization	4
4.2	Resource-label extraction	4
4.3	Access-level classification	4
4.4	Sensitivity / functional classification	4
5	Released-artifact coverage and class distributions	5
5.1	Provider coverage	5
5.2	Access-level and sensitivity distributions	5
6	Representative provider-specific mappings	5
7	Difficult and ambiguous normalization cases	6
8	Topology construction rules	7
8.1	Observed edge-rule output	7
9	Cross-provider bridges	8
10	Community and centrality audit summaries	8
10.1	Community detection	8
10.2	Centrality	9
11	Auditability and reproducibility	10

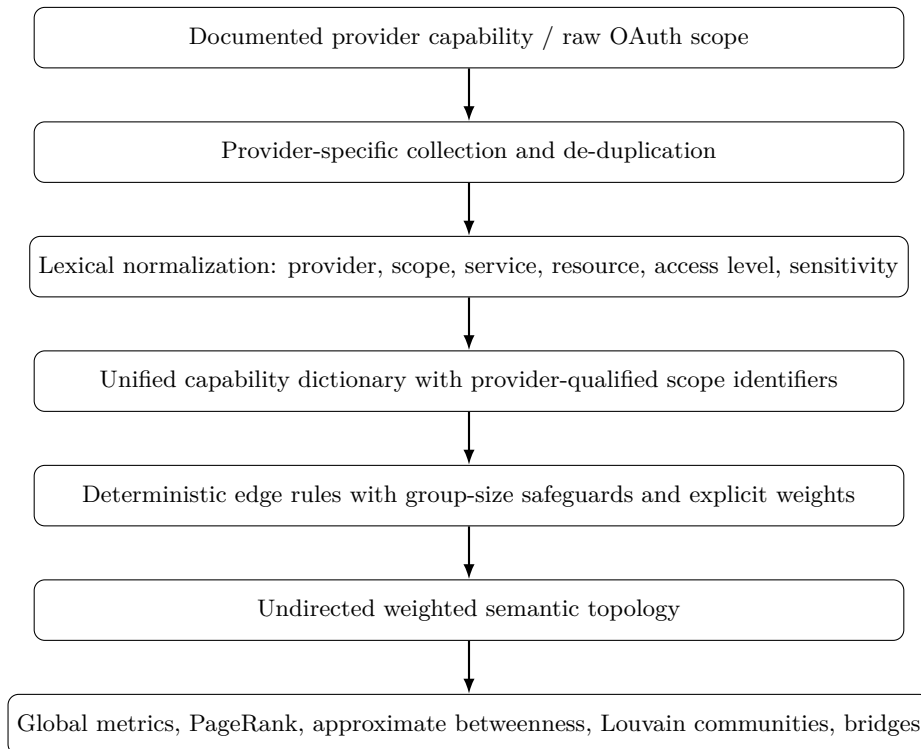
12 Interpretive boundaries and threats to validity	10
13 Recommended citation from the main manuscript	10
A Complete resource-label inventory	10

1 Scope and relationship to the main article

The main article analyzes a structural network whose nodes are provider-specific OAuth capabilities and whose edges are produced by deterministic semantic rules. This supplement does not introduce a second analysis or an alternative dataset. It exposes the operational details that are necessarily abbreviated in the main article. The released dictionary inspected here contains **1,599** capability records from seven providers. The counts in this document describe the released repository snapshot and must be regenerated if live provider documentation is collected again.

This supplement is intended to answer four audit questions: (1) how a raw provider capability becomes a normalized record; (2) which categories and labels are actually present; (3) exactly why two capabilities become adjacent in the graph; and (4) which claims can and cannot be inferred from the resulting topology.

2 End-to-end workflow



3 Unified record schema

Each capability is represented by the following fields:

Field	Meaning
provider	Provider name used as a provenance attribute.
scope	Original documented scope or permission string after whitespace and escaped-slash normalization.
service	Coarse service label; normally derived from resource, with limited GitHub-specific handling.
resource	Lexical resource label extracted by the ordered rules in Section 4.2.
access_level	One of write_or_admin , read , identity , or other .
sensitivity	One of eight functional/sensitivity classes defined in Section 4.4.
description	Provider documentation text when available.
source_url	Documentation or discovery source used during collection.
scope_id	Provider-qualified identifier of the form Provider::scope ; this prevents collisions across providers.

4 Complete normalization rule specification

4.1 Text normalization

The original capability string is stripped of leading and trailing whitespace and escaped slashes are converted to ordinary slashes. Case is retained in the original `scope` field, while rule matching is case-insensitive.

4.2 Resource-label extraction

The `resource` field is a reproducible lexical label, not a manually adjudicated ontology. Rules are applied in the following order:

1. **Google URI rule.** If the lowercased scope contains `googleapis.com/auth/`, take the substring after `/auth/`; remove `.readonly`, `.read`, and `.write`; then use the first segment before a period or slash.
2. **Dot-delimited rule.** If the scope contains a period but contains neither a colon nor a slash, use the first dot-delimited segment.
3. **Colon-delimited rule.** If the scope contains a colon, use the segment before the first colon.
4. **Fallback rule.** Otherwise use the first segment before a period or colon, lowercase it, and replace hyphens with underscores.

These rules create 552 distinct resource labels in the released dictionary. Appendix A lists all labels and frequencies.

4.3 Access-level classification

Access rules are ordered and case-insensitive. Earlier successful rules take precedence.

Class	Trigger tokens	Operational behavior	Count
<code>write_or_admin</code>	<code>readwrite</code> , <code>write</code> , <code>modify</code> , <code>admin</code> , <code>manage</code> , <code>delete</code> , <code>full</code>	Returned when a write/administrative token is present. An outer check also recognizes <code>all</code> , but <code>all</code> alone does not cause assignment unless one of the listed return-trigger tokens is present.	593
<code>read</code>	<code>readonly</code> , <code>read.only</code> , <code>read</code> , <code>view</code>	Returned only if the preceding write/administrative rule did not return.	602
<code>identity</code>	<code>email</code> , <code>profile</code> , <code>openid</code> , <code>identity</code> , <code>signin</code> , <code>user</code>	Returned only if neither prior rule returned.	45
<code>other</code>	None of the above	Fallback class; it does not imply absence of privilege, only absence of recognized lexical markers.	359

4.4 Sensitivity / functional classification

Sensitivity rules are also ordered; the first matching category is retained.

Class	Trigger tokens	Interpretation	Count
<code>communication</code>	<code>mail</code> , <code>gmail</code> , <code>email</code> , <code>message</code> , <code>chat</code>	Messaging and communication capabilities.	178
<code>content</code>	<code>drive</code> , <code>files</code> , <code>file</code> , <code>storage</code> , <code>repo</code> , <code>repository</code>	Stored content, files, storage, and repositories.	93
<code>calendar</code>	<code>calendar</code> , <code>event</code>	Calendar and event capabilities.	42
<code>organization</code>	<code>directory</code> , <code>group</code> , <code>org</code> , <code>organization</code> , <code>team</code> , <code>member</code>	Organizational structure, groups, teams, and membership.	214
<code>identity</code>	<code>user</code> , <code>profile</code> , <code>openid</code> , <code>identity</code> , <code>signin</code>	Identity and profile capabilities.	101
<code>administration</code>	<code>admin</code> , <code>manage</code> , <code>policy</code> , <code>audit</code> , <code>security</code>	Administration, governance, audit, and security.	144
<code>financial</code>	<code>payment</code> , <code>billing</code> , <code>finance</code> , <code>wallet</code>	Financial and billing capabilities.	5

Class	Trigger tokens	Interpretation	Count
general	None of the above	Fallback category; denotes no recognized class token, not low risk.	822

5 Released-artifact coverage and class distributions

5.1 Provider coverage

Table 4: Capability records by provider in the released repository snapshot.

Provider	Records	Share
Microsoft	964	60.29%
Google	467	29.21%
Slack	72	4.50%
GitHub	37	2.31%
Discord	28	1.75%
Dropbox	23	1.44%
Notion	8	0.50%

5.2 Access-level and sensitivity distributions

Access level	Count	Share	Sensitivity	Count	Share
read	602	37.65%	general	822	51.41%
write_or_admin	593	37.09%	organization	214	13.38%
other	359	22.45%	communication	178	11.13%
identity	45	2.81%	administration	144	9.01%
			identity	101	6.32%
			content	93	5.82%
			calendar	42	2.63%
			financial	5	0.31%

The large **general** and **other** groups are reported rather than hidden. They reveal the limit of a deliberately coarse lexical taxonomy and are important when interpreting topology generated from these attributes.

6 Representative provider-specific mappings

Table 5 reports exact rows from the released unified dictionary. It includes every provider and multiple semantic classes.

Table 5: Exact normalization outputs from the released dictionary.

Provider	Original capability	Resource	Access	Sensitivity
Google	https://www.googleapis.com/auth/admin.datatransfer	admin	write_or_admin	administration
Google	https://www.googleapis.com/auth/gmail.readonly	gmail	read	communication
Google	https://www.googleapis.com/auth/calendar.readonly	calendar	read	calendar
Google	https://www.googleapis.com/auth/userinfo.email	userinfo	identity	communication
Microsoft	AdministrativeUnit.Read.All	administrative unit	write_or_admin	administration

Provider	Original capability	Resource	Access	Sensitivity
Microsoft	Mail.Read	mail	read	communication
Microsoft	User.Read	user	read	identity
Microsoft	Files.ReadWrite.All	files	write_or_admin	content
Slack	files:write	files	write_or_admin	content
Slack	remote_files:read	remote_files	read	content
Slack	channels:history	channels	other	general
Slack	users:read.email	users	read	communication
GitHub	repo	repo	other	content
GitHub	read:org	read	read	organization
GitHub	user:email	user	identity	communication
Discord	guilds.members.read	guilds	read	organization
Discord	applications.commands	applications	other	general
Discord	email	email	identity	communication
Dropbox	files.content.write	files	write_or_admin	content
Dropbox	team_data.governance.read	team_data	read	organization
Dropbox	account_info.read	account_info	read	general
Notion	read_user	read_user	read	identity
Notion	update_database	update_databas e	other	general
Notion	create_comment	create_comment	other	general

7 Difficult and ambiguous normalization cases

The following cases are included because they expose rule precedence, lexical limitations, or provider-specific naming conventions. They are not silently corrected after classification; the released outputs remain traceable to the executable rules.

Table 6: Audit of difficult or ambiguous cases.

Provider	Scope	Resource	Access	Sensitivity	Why the case is informative
GitHub	repo	repo	other	content	Bare lexical token; access intent is not encoded by an explicit read/write marker.
Microsoft	User.Read	user	read	identity	Contains both user and read; ordered access rules assign read before identity.
Slack	files:write	files	write_or_admin	content	Colon-delimited scope; resource is the prefix and write drives access classification.
Google	openid	openid	identity	identity	Identity token with no explicit read/write marker.
Google	https://www.googleapis.com/auth/userinfo.email	userinfo	identity	communication	Google URI parsing extracts userinfo; email triggers communication before identity.
Microsoft	AdministrativeUnit.Read.All	administrativeunit	write_or_admin	administration	Contains All and Read; implementation checks write/admin family first but only returns write_or_admin when a write/admin token is also present; resource is the first dot segment.
Slack	channels:history	channels	other	general	No explicit read token; falls to other although the capability is observational in practice.

Provider	Scope	Resource	Access	Sensitivity	Why the case is informative
Discord	<code>applications.commands</code>	<code>applications</code>	<code>other</code>	<code>general</code>	No explicit access token; falls to other.
Notion	<code>read_user</code>	<code>read_user</code>	<code>read</code>	<code>identity</code>	Fallback resource retains the full underscore-delimited token; read takes precedence over identity.
Dropbox	<code>team_data.governance.read</code>	<code>team_data</code>	<code>read</code>	<code>organization</code>	Dot-delimited parsing uses <code>team_data</code> as resource; organization sensitivity is triggered before administration.

These examples establish an important boundary: the normalization is deterministic and inspectable, but it is not claimed to reproduce every provider’s operational authorization semantics perfectly. The topology therefore represents similarity under the stated rules rather than proof of privilege equivalence.

8 Topology construction rules

The graph is undirected and weighted. Every node is a provider-qualified capability. If more than one rule connects the same pair, weights are added and edge-type labels are concatenated. Self-loops are prohibited.

Rule	Edge type	Condition and safeguard	Weight	Interpretation
R1	<code>same_provider_same_resource</code>	Same provider and resource. Applied only when group size is 2–25.	3	Strong within-provider lexical resource relation.
R2	<code>same_provider_same_sensitivity_access</code>	Same provider, sensitivity, and access level. Applied only when group size is 2–15.	1	Weaker attribute-level relation.
R3	<code>cross_provider_resource_bridge</code>	Different providers; same non-general sensitivity; resource strings are equal or one contains the other.	2	Explicit cross-provider semantic bridge.
R4	<code>provider_identity_cluster</code>	Same provider and sensitivity is identity. Applied only when identity group size is 2–30.	2	Within-provider identity grouping.

The upper group-size limits are anti-clique safeguards. They prevent very broad lexical classes from automatically creating extremely dense complete subgraphs. Conversely, they can suppress relations for groups exceeding the thresholds; this is a design choice and an explicit source of sensitivity.

8.1 Observed edge-rule output

Table 8: Edge records by generating rule in the released artifact.

Edge rule	Recorded pairs
<code>same_provider_same_resource</code>	2773
<code>cross_provider_resource_bridge</code>	2307
<code>same_provider_same_sensitivity_access</code>	1062
<code>provider_identity_cluster</code>	157

Because the final graph merges repeated node pairs, the number of recorded rule applications need not equal the number of unique graph edges. In the released graph there are **1599 nodes**, **6049 unique edges**, density **0.004735**, and **514 connected components**. These values are properties of this released snapshot, not universal properties of OAuth ecosystems.

9 Cross-provider bridges

Cross-provider bridges are not independently inferred after graph construction; they are the edges produced by R3 and then extracted by checking whether endpoint providers differ. Thus, bridge counts are consequences of the declared semantic matching rule. They should be interpreted as candidate structural correspondences under the taxonomy, not as certified operational equivalences.

Table 9: Representative exact cross-provider bridges from the released bridge file.

Provider A	Capability A	Provider B	Capability B	Sensitivity
Google	<code>https://www.googleapis.com/auth/admin.dataatransfer</code>	Microsoft	<code>AdministrativeUnit.Read.All</code>	administration
Slack	<code>files:write</code>	Dropbox	<code>files.content.write</code>	content
Slack	<code>remote_files:read</code>	Dropbox	<code>files.content.read</code>	content
Slack	<code>team:read</code>	Dropbox	<code>team_data.governance.write</code>	organization
Google	<code>https://www.googleapis.com/auth/admin.dataatransfer</code>	Microsoft	<code>AdministrativeUnit.Read.All</code>	administration
Google	<code>https://www.googleapis.com/auth/admin.chrome.printers</code>	GitHub	<code>admin:enterprise</code>	administration
Google	<code>https://www.googleapis.com/auth/chat.admin.delete</code>	Slack	<code>chat:write.public</code>	communication
Google	<code>https://www.googleapis.com/auth/user.addresses.read</code>	Notion	<code>read_user</code>	identity
Google	<code>https://www.googleapis.com/auth/groups</code>	Dropbox	<code>groups.read</code>	organization
Microsoft	<code>Chat.Create</code>	Slack	<code>chat:write</code>	communication
Microsoft	<code>CallEvents.Read.All</code>	Dropbox	<code>events.read</code>	calendar
Microsoft	<code>AgentIdUser.ReadWrite.IdentityParentedBy</code>	GitHub	<code>user:follow</code>	identity

The released bridge file contains **2,307** cross-provider edges. Removing R3 is therefore expected to remove most or all such edges; this is evidence of rule dependence, not an independent validation of equivalence.

10 Community and centrality audit summaries

10.1 Community detection

Communities are estimated with Louvain optimization on the weighted graph using random state 42. High modularity must be interpreted in light of the deterministic edge rules and group-size safeguards; it is evidence of partitioned structure in the constructed graph, not proof that providers expose natural, universally accepted capability domains.

Table 10: Largest released communities and their dominant attributes.

ID	Size	Providers	Dominant sensitivity	Dominant access	Highest-PageRank scope
320	117	Dropbox, Google, Microsoft, Slack	organization	read	team:read
11	77	GitHub, Google, Microsoft, Notion, Slack	identity	read	https://www.googleapis.com/auth/user.phonenumbers.read

ID	Size	Providers	Dominant sensitivity	Dominant access	Highest-PageRank scope
26	68	Google, Microsoft, Slack	communication	read	chat:write.customize
240	53	Dropbox, GitHub, Microsoft, Slack	content	write_or_admin	files.metadata.read
3	52	GitHub, Google, Microsoft	organization	write_or_admin	admin:org
24	43	Dropbox, GitHub, Google, Microsoft	calendar	read	https://www.googleapis.com/auth/calendar.freebusy
5	41	Google, Microsoft	content	read	https://www.googleapis.com/auth/drive.admin.labels
28	28	Google	general	other	https://www.googleapis.com/auth/classroom.profile.photos
500	28	Slack	general	read	channels:read
128	28	Discord, Google, Microsoft	communication	other	https://www.googleapis.com/auth/gmail.readonly
521	26	Discord	general	other	applications.commands
475	23	Microsoft	organization	write_or_admin	TeamsTab.Create.Group
125	22	Google	general	read	https://www.googleapis.com/auth/fitness.heart_rate.read
10	15	Google, Microsoft	general	other	https://www.googleapis.com/auth/apps.groups.settings
389	10	Microsoft	general	write_or_admin	PrintJob.Read

10.2 Centrality

PageRank uses edge weights. Approximate betweenness is computed from a sample of up to 100 nodes with seed 42. Consequently, PageRank is deterministic for the released graph, whereas betweenness is an approximation under the stated sampling configuration.

Table 11: Highest-PageRank capabilities in the released graph.

Provider	Scope	PageRank	Degree	Approx. betweenness
Slack	team:read	0.011714	99	0.055720
GitHub	admin:org_hook	0.004443	30	0.003696
GitHub	admin:org	0.004443	30	0.003696
Google	https://www.googleapis.com/auth/user.phonenumbers.read	0.002195	61	0.004586
Google	https://www.googleapis.com/auth/user.gender.read	0.002195	61	0.004586
Google	https://www.googleapis.com/auth/user.birthday.read	0.002195	61	0.004586
Google	https://www.googleapis.com/auth/user.addresses.read	0.002195	61	0.004586
GitHub	admin:public_key	0.001896	19	0.002800
GitHub	admin:enterprise	0.001896	19	0.002800
GitHub	user	0.001677	48	0.001183
GitHub	user:follow	0.001677	48	0.001183
Microsoft	TeamsTab.Create.Group	0.001559	36	0.001941
Microsoft	TeamsTab.Create	0.001559	36	0.001941
Dropbox	files.metadata.read	0.001363	27	0.000238
Dropbox	files.metadata.write	0.001363	27	0.000238

11 Auditability and reproducibility

Every mapping can be audited in `unified_oauth_scope_dictionary.csv`; every rule application can be audited in `semantic_topology_edges.csv`; final cross-provider pairs are in `cross_provider_bridges.csv`; node-level metrics are in `scope_centralty_metrics.csv`; community summaries are in `community_summary.csv`; and the complete weighted topology is in GraphML format.

A minimal reproduction sequence is:

```
python -m venv .venv
# activate the environment, then install pandas, requests, networkx,
# matplotlib, python-louvain, and tqdm
python build_topology.py
```

The collector accesses live public documentation. Therefore, exact capability counts may change as providers modify documentation. For strict reproduction of the manuscript snapshot, use the released CSV and GraphML artifacts rather than recollecting live sources.

12 Interpretive boundaries and threats to validity

1. **Lexical rather than operational semantics.** Similar names do not guarantee identical authorization effects, resource granularity, consent behavior, or API-side enforcement.
2. **Ordered substring effects.** Compound names may match an earlier category even when a later category is also plausible. Section 7 provides concrete cases.
3. **Provider imbalance.** Microsoft and Google contribute most nodes; provider-removal analyses are therefore necessary to separate broad structure from dominant-provider effects.
4. **Rule-induced topology.** Modularity, bridges, and communities partially reflect the construction rules. Structural results must be discussed as properties of the resulting semantic representation.
5. **Collection drift.** Live documentation can change; the released snapshot is the authoritative artifact for the reported run.
6. **No claim of authorization equivalence.** R3 creates candidate cross-provider semantic bridges, not proof that two capabilities are substitutable or security-equivalent.
7. **Coarse classes.** The fallback categories `general` and `other` preserve unmatched capabilities but limit semantic resolution.

13 Recommended citation from the main manuscript

The main article may refer to this file with the following concise statement:

The complete normalization rules, category inventories, representative and ambiguous mapping cases, topology-construction rules, and released-artifact audit tables are provided in the accompanying Supplementary Material.

A Complete resource-label inventory

The table below lists all distinct resource labels and their frequencies in the released unified dictionary. These are lexical outputs of the extraction rules, not manually merged ontology concepts.

Resource label	Frequency
<code>accessreview</code>	3
<code>account_info</code>	1
<code>acronym</code>	1
<code>activities</code>	2
<code>adexchange</code>	1

Resource label	Frequency
admin	37
administrativeunit	2
admob	2
adsense	2
advanced	3
adwords	1
agentcard	3
agentcardmanifest	3
agentcollection	7
agentidentity	6
agentidentityblueprint	7
agentidentityblueprintprincipal	5
agentiduser	2
agentinstance	3
agentregistration	2
agentregistry-only	1
agentregistry-write	1
agreement	2
agreementacceptance	2
aienterpriseinteraction	3
analytics	6
androidenterprise	1
androidmanagement	1
androidpublisher	1
anon	2
apiconnectors	2
app_mentions	1
appcatalog	3
appcerttrustconfiguration	2
appengine	1
application	4
applications	7
aproleassignment	1
approvalsolution	4
approvalsolutionresponse	1
apps	7
appsandservices	2
appsmarketplace	1
attacksimulation	2
audit_log	1
auditactivity	2
auditlog	1
auditlogsquery	1
authenticationcontext	2
authorized-buyers-marketplace	1
background	1
bigquery	2
bigtable	4
billingconfiguration	1
bitlockerkey	2
blogger	2
bookings	3
bookingsappointment	1
bookmark	1

Resource label	Frequency
bookmarks	2
books	1
border	1
bot	1
bottom	1
browsersitelists	2
businessscenarioconfig	4
businessscenarioconfigdata	2
calendar	17
calendars	6
callaiinsights	2
calldelegation	4
callevents	2
callrecordings	2
callrecords	1
calls	7
calltranscripts	2
canvases	2
ces	1
changemanagement	1
channel	5
channelmeeting	1
channelmeetingnotification	1
channelmeetingparticipant	1
channelmeetingrecording	1
channelmeetingtranscript	1
channelmember	4
channelmessage	7
channels	4
channelsettings	4
chat	53
chatmember	7
chatmessage	5
chatmessagereadreceipt	1
chatsettings	2
chrome	7
chromewebstore	2
classroom	24
cloud-bigtable	3
cloud-billing	2
cloud-healthcare	1
cloud-identity	11
cloud-language	1
cloud-netapp	1
cloud-platform	1
cloud-platform-only	1
cloud-translation	1
cloud-vision	1
cloud_search	9
cloudcomposer	2
cloudkms	1
cloudpc	2
cloudruntimeconfig	1
codespace	1

Resource label	Frequency
commands	1
community	2
compute	2
configuration	2
configurationmonitoring	2
connections	1
consentrequest	5
contacts	8
container	1
container-only	1
content	3
contentactivity	2
contracts	1
control	2
conversation	2
conversations.connect	2
converttointernal	1
copilot	1
copilotconversation	1
copilotpackages	2
copilotpolicysettings	2
create_comment	1
crm	1
crosstenantinformation	1
crosstenantuserprofilessharing	4
customauthenticationextension	3
customdetection	2
customext	2
customsecattributeassignment	2
customsecattributeauditlogs	1
customsecattributedefinition	2
customsecattributeprovisioning	2
customtags	2
dataconnector	3
datalineage	1
datalineage-write	1
datamanager	2
dataplex	1
dataplex-write	1
dataportability	72
dataproc	1
dataproc-only	1
datastore	3
ddmconversions	1
delegatedadminrelationship	2
delegatedpermissiongrant	2
delete	1
delete_repo	1
device	5
devicelocalcredential	2
devicemanagementapps	2
devicemanagementcloudca	2
devicemanagementconfiguration	2
devicemanagementmanageddevices	3

Resource label	Frequency
devicemanagementrbac	2
devicemanagementscripts	2
devicemanagementserviceconfig	2
devicetemplate	3
devstorage	1
devstorage_only	1
devstorage_write	1
dfareporting	1
dfatrafficking	1
dialogflow	1
directory	4
directoryrecommendations	2
discovery	1
discoveryengine	2
discoveryenginewrite	1
display	2
display-video	1
display-video-mediaplanning	1
display-video-user-management	1
dm_channels	1
docsearch	3
documents	2
domain	2
doubleclickbidmanager	1
doubleclicksearch	1
drive	16
dynamicsvoice	2
eas	1
ediscovery	2
eduadministration	4
eduassignments	8
educurricula	4
eduroster	6
email	5
emergency	1
endpoint	1
engagementconversation	3
engagementexport	1
engagementmeetingconversation	1
engagementrole	3
entitlementmanagement	2
entra	1
entrabackup	3
eventlistener	2
events	1
ews	1
exchange	1
exchangemessagetrace	1
export	2
external	4
externalconnection	3
externalitem	3
externaluserprofile	2
factchecktools	1

Resource label	Frequency
family	1
file_requests	2
fileingestion	1
fileingestionhybridonboarding	1
files	14
filestoragecontainer	2
filestoragecontainertype	1
filestoragecontainertypereg	2
financials	1
firebase	3
fitness	22
forms	7
games	1
gdm	1
gist	1
gmail	13
googlehealth	8
group	3
groupmember	2
groups	6
groupsettings	2
guests	2
guilds	3
hardwareoath	4
healthmonitoringalert	2
healthmonitoringalertconfig	2
homegraph	1
https	3
identify	1
identitynotifications	2
identityprovider	2
identityriskevent	2
identityriskyagent	2
identityriskyserviceprincipal	2
identityriskyuser	2
identityuserflow	2
im	3
imap	1
inboundflow	2
incoming_webhook	1
indexing	1
industrydata	1
informationprotectionconfig	2
informationprotectioncontent	2
informationprotectionpolicy	2
insert_content	1
internalfederation	2
invitation	2
jobs	1
keep	2
learningassignedcourse	3
learningcontent	2
learningprovider	2
learningselfinitiatedcourse	3

Resource label	Frequency
left	1
licenseassignment	2
lifecycleinfo	2
lifecycleworkflows	2
limitedmode	2
links	2
listitems	1
lists	1
logging	3
mail	10
mailboxconfigitem	2
mailboxfolder	4
mailboxitem	6
mailboxsettings	2
mailtips	2
manage_billing	1
manage_runners	1
managedtenants	2
manufacturercenter	1
maps-platform	7
marketingplatformadmin	2
meetings	3
member	2
members	2
messages	1
microsoft365install	2
microsoft365web	1
microsoftauthapp	4
monitor	1
monitoring	3
mpim	3
multitenantorganization	3
mutualtlssoauthconfiguration	2
ndev	4
nestingsupport	1
networkaccess	2
networkaccessbranch	2
networkaccesspolicy	2
notes	6
notifications	2
og	4
onedrive	1
onlinemeeting	3
onlinemeetinggaininsight	2
onlinemeetingartifact	2
onlinemeetingnotification	1
onlinemeetingparticipant	1
onlinemeetingrecording	2
onlinemeetings	4
onlinemeetingtranscript	2
onpremsdirectorysynchronization	2
onpremisespublishingprofiles	1
onpremisesyncbehavior	1
openid	1

Resource label	Frequency
organization	2
organizationalbranding	2
orgcontact	1
outboundflow	2
owner	1
padding	2
partnerbilling	1
partnersecurity	2
passkey	4
password	4
passwordprofile	1
pendingexternaluserprofile	2
people	2
peoplesettings	2
phone	4
pins	2
place	2
placedevice	2
placedevicetelemetry	1
platformcred	4
playdeveloperreporting	1
playintegrity	1
policy	26
policytemplate	2
pop	1
position	1
postmaster	4
presence	4
presentations	2
printconnector	2
printer	4
printershare	3
printjob	10
printsettings	2
printtaskdefinition	1
privilegedaccess	6
privilegedassignmentschedule	5
privilegedeligibilityschedule	5
profile	2
profilephoto	2
programcontrol	2
protectionscope	2
provisioninglog	1
pstncalls	1
public_repo	1
publickeyinfrastructure	2
pubsub	1
qna	1
qr	4
radius	1
reactions	2
read	8
read_comment	1
read_content	1

Resource label	Frequency
read_database	1
read_user	1
reading	2
readwrite	1
realtime-bidding	1
realtimeactivityfeed	1
recordsmanagement	2
redis-only	1
redis-write	1
referencedefinition	2
reflect	2
relatedtenant	2
relationship	2
relationships	1
reminders	2
remote_files	3
remotedesktopconfig	1
repo	3
repo_deployment	1
reports	1
reportsettings	2
request	2
requests	1
resourcespecificpermissiongrant	6
restore	2
right	1
riskpreventionproviders	2
role_connections	1
roleassignmentschedule	3
roleeligibilityschedule	3
rolemanagement	9
rolemanagementalert	2
rolemanagementpolicy	6
rpc	5
rule	1
run	4
sasportal	1
schedule	2
schedulepermissions	1
scim	1
script	6
sdm	1
search	1
searchconfiguration	2
security_events	1
securityactions	2
securityalert	2
securityanalyzedmessage	2
securitycopilotworkspaces	2
securityevents	2
securityidentitiesaccount	1
securityidentitiesactions	1
securityidentitiesautoconfig	2
securityidentitieshealth	2

Resource label	Frequency
securityidentitiesmigration	2
securityidentitiessensors	2
securityidentitiesuseractions	2
securityincident	2
sensitivitylabel	3
sensitivitylabels	1
sentimentsurvey	1
service	2
servicecontrol	1
servicehealth	1
servicemessage	1
servicemessageviewpoint	1
serviceprincipalendpoint	2
sessions	2
setting	2
sharepoint	1
sharepointcrosstenantmigration	2
sharepointtenantsettings	2
sharing	2
shortnotes	4
signinidentifier	2
sites	7
siteverification	2
size	1
smtp	1
softwareoath	4
sourcesystem	2
spanner	2
spiffetrustdomain	2
spreadsheets	2
sqlservice	1
storyline	1
streetviewpublish	1
subjectaccess	1
subjectrightsrequest	2
subscribewithgoogle	1
subscription	1
synchronization	2
tagmanager	7
tap	4
tasks	8
team	4
team_data	5
team_info	1
teammember	4
teams	1
teamsactivity	6
teamsappinstallation	52
teamsettings	5
teamspolicyuserassign	1
teamsresourceaccount	1
teamstab	23
teamstelephonenumber	2
teamsuserconfiguration	1

Resource label	Frequency
teamtemplates	2
teamwork	2
teamworkappsettings	2
teamworkdevice	2
teamworksection	4
teamworktag	4
teamworktargetedmessage	2
teamworkuserinteraction	1
termstore	2
threatassessment	2
treathunting	1
threatindicators	2
threatintelligence	1
threatsubmission	4
threatsubmissionpolicy	1
timeperiod	2
todo	2
topic	1
trace	2
trustframeworkkeyset	2
twitter	1
unifiedgroupmember	1
update_content	1
update_database	1
user	25
useractivity	1
userauthenticationmethod	4
usercloudclipboard	1
userinfo	2
usermetric	1
usernotification	1
users	3
users.profile	2
usershiftpreferences	2
userteamwork	2
usertimelineactivity	1
userwindowssettings	2
verifiedaccess	1
virtualappointment	4
virtualappointmentnotification	1
virtualevent	4
visibility	1
voice	1
wallet_object	1
webhook	1
webmasters	2
width	1
windowshello	4
windowsupdates	2
workflow	5
workflow.steps	1
workforceintegration	2
workingtime	1
write	5

Resource label	Frequency
xml	1
xtenantidentitysync	1
youtube	5
youtubepartner	1
youtubepartner-channel-audit	1
yt-analytics	1
yt-analytics-monetary	1