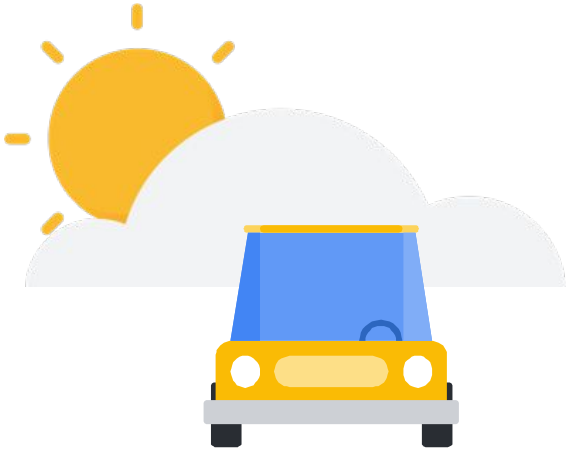
Carolyn Huynh | Google Cloud Security

Self Driving Cloud

Troubleshooter



care0hlyn.github.com

Lead UX Designer

Role

I was the first designer to join the Self Driving Cloud team and ever since, I've been the Lead UX Designer and thought leader behind the suite of SDC products three years now.

Troubleshooter is the second product within the self-driving cloud initiative that I have lead from inception to launch.

The Problem Statement:

Troubleshooting 403 error messages is hard.
It's done through lots of trial & error.

Admins troubleshoot by giving *more* permissions, which makes it hard to maintain least privilege.

What are we building?

Troubleshooter is the second product launch within the Self Driving Cloud suite of products. This product is meant to help over granting permissions by helping admins debug denial error messages.

Built on the IAM Recommender API, it was a sophisticated backend that needed to be unpacked simply in the UI.

Example User Scenario

Anita is a developer in your organization and gets an error message informing her that she *does not* have the relevant permissions when she tries to delete images in a bucket.

Anita goes to Mike, her security admin for help, who has to manually check the policy himself, and will end up expediting the process by just granting Anita a bunch of unnecessary roles.

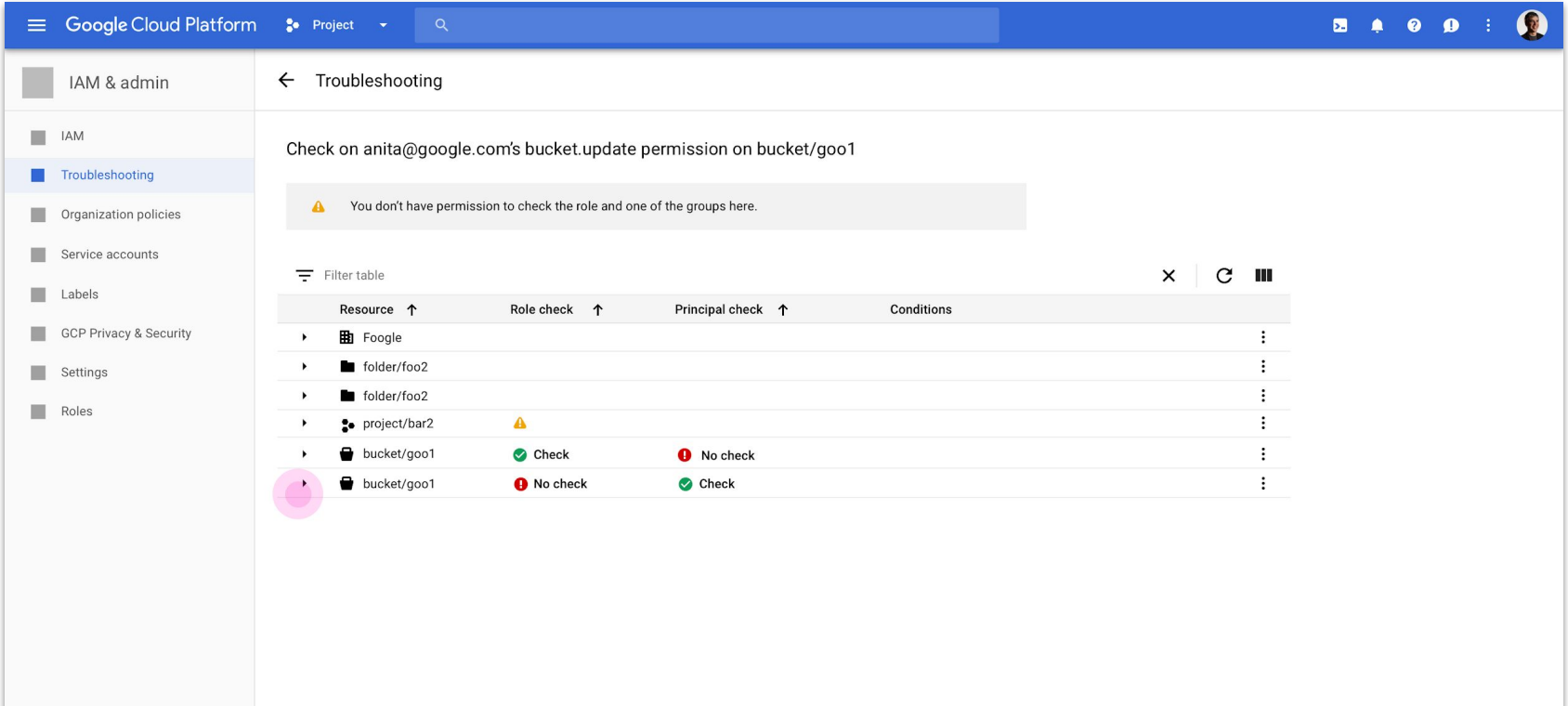
Translating the API to GUI

The engineers had approached me with a predetermined sketch in mind.

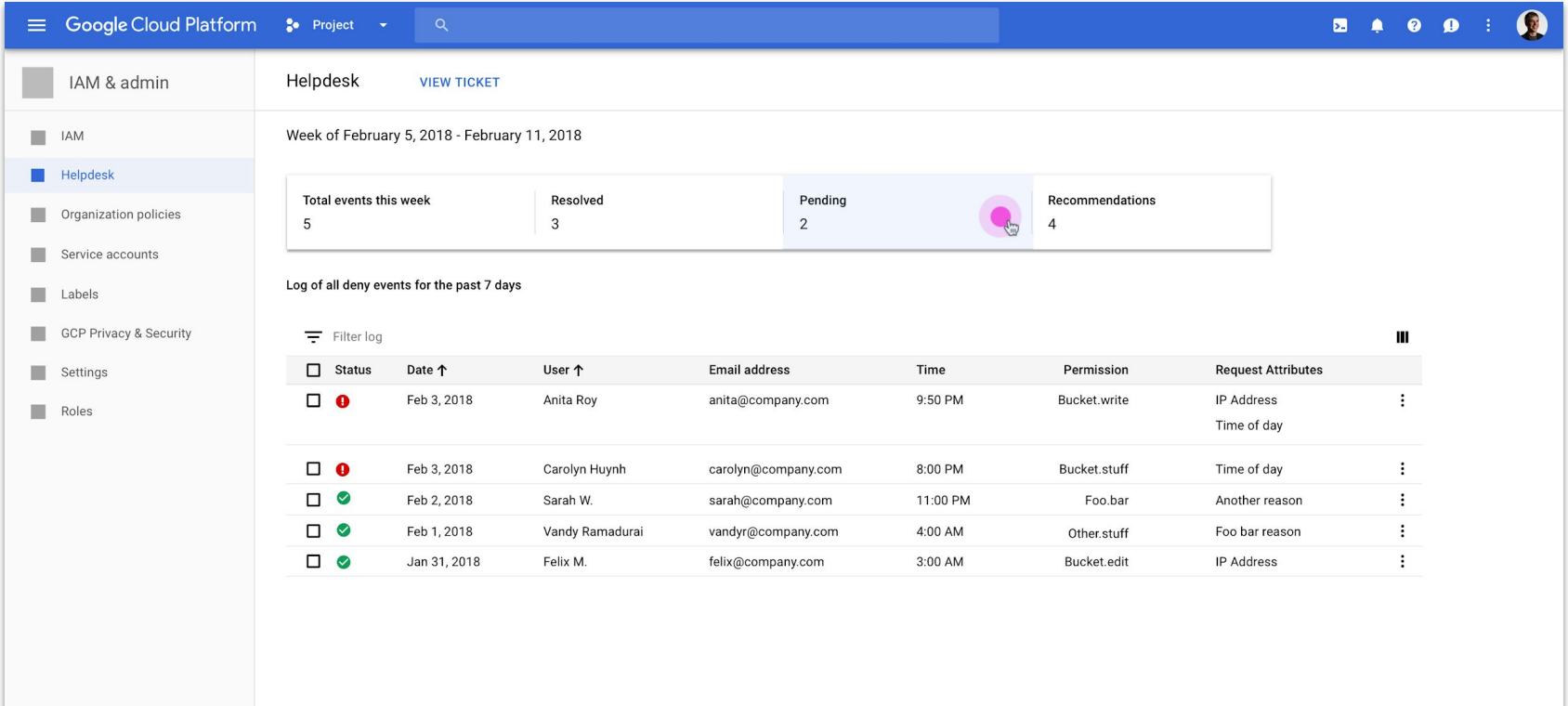
After going through many rounds of the engineers' design solution, I was convinced that their way was not the way to go. Through lots of research, I changed product direction by designing a different way to display IAM bindings by showing raw json policy so that advanced users would be able to click through bindings via a tree hierarchy.

I was able to change product direction with multiple stakeholders with design thinking and creating a more visual tool for developers.

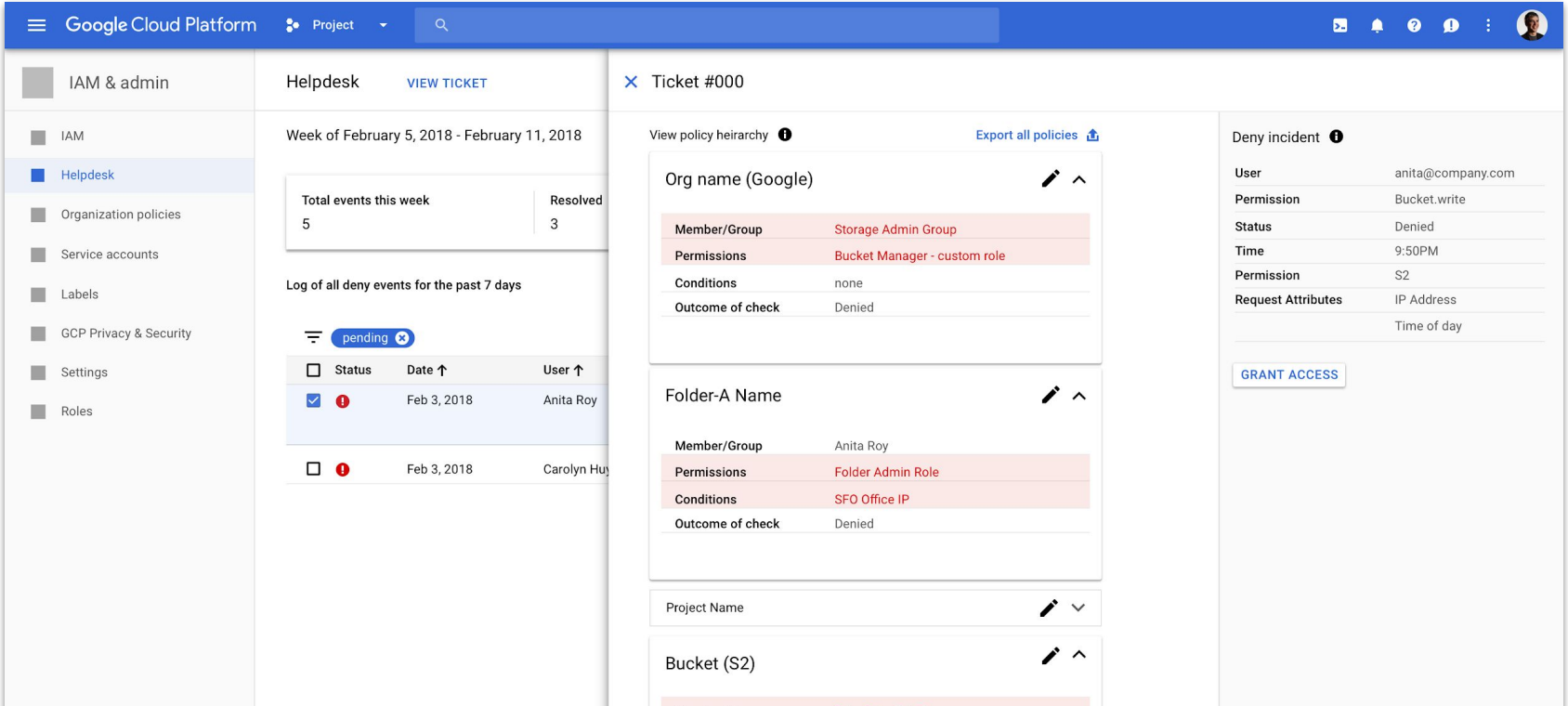
...FYI, here was the engineer's idea



...another example.

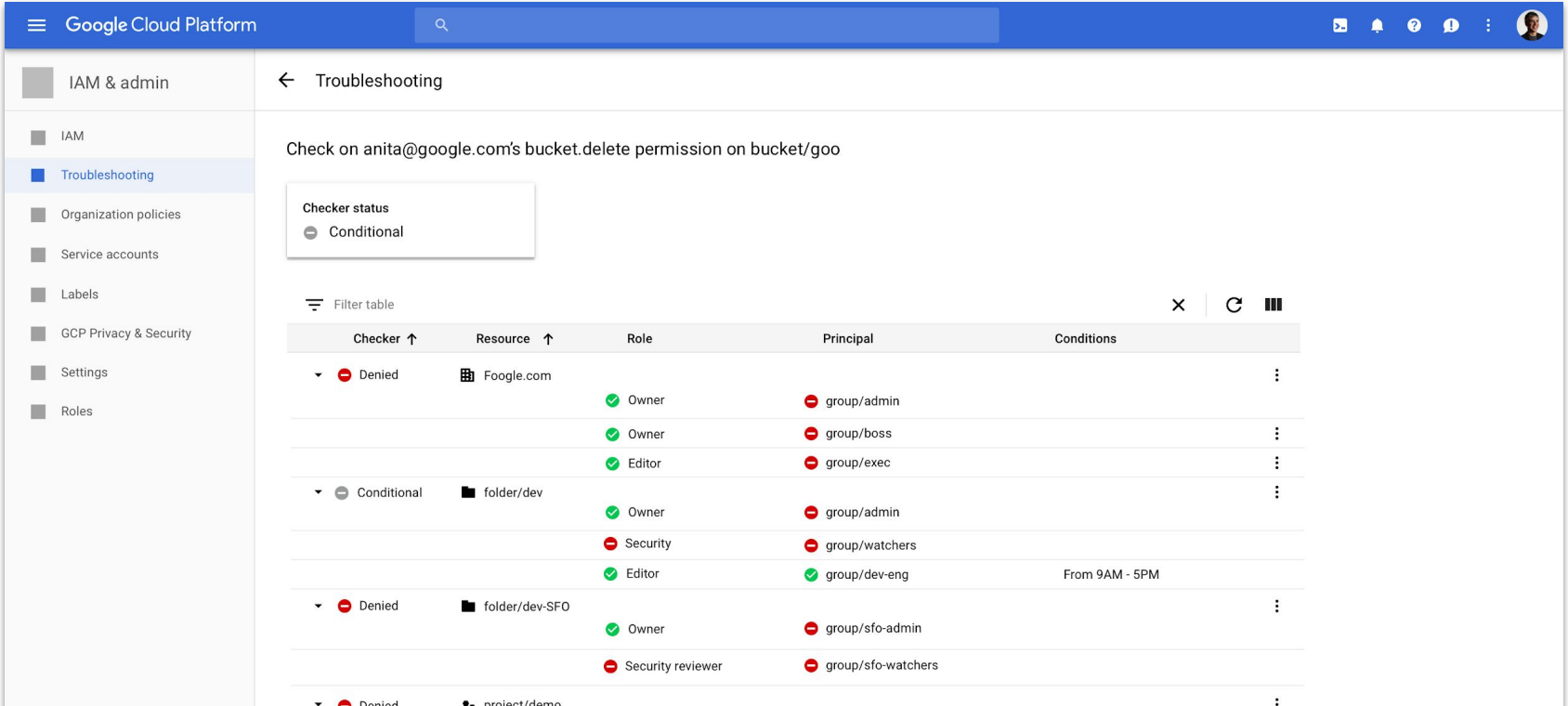


...annnd one more.



No matter how many versions of tables I designed, the problem was that developers still couldn’t understand and debug readable text.

...jk, last one.


Google Cloud Platform
IAM & admin
IAM
Troubleshooting
Organization policies
Service accounts
Labels
GCP Privacy & Security
Settings
Roles
Troubleshooting
Check on anita@google.com's bucket.delete permission on bucket/goo
Checker status
Conditional
Filter table
Checker
Resource
Role
Principal
Conditions
Denied
Foogle.com
Owner
group/admin
Owner
group/boss
Editor
group/exec
Conditional
folder/dev
Owner
group/admin
Security
group/watchers
Editor
group/dev-eng
From 9AM - 5PM
Denied
folder/dev-SFO
Owner
group/sfo-admin
Security reviewer
group/sfo-watchers

```
{
  "bindings": [
    {
      "members": [
        "user:jie@example.com"
      ],
      "role": "roles/resourcemanager.organizationAdmin"
    },
    {
      "members": [
        "user:divya@example.com",
        "user:jie@example.com"
      ],
      "role": "roles/resourcemanager.projectCreator"
    }
  ],
  "etag": "BwUjMhCsNvY=",
  "version": 1
}
```


Research

Research had shown that the original idea behind the engineer's mocks for the product had one consistent theme:

Nobody could tell that each row in the table represented a binding within an IAM policy.

So, I literally went through the API and the subsequent raw json policy and began scrubbing it, in an attempt to find UI components that would expliciting show what each section of the policy meant.

I decided to be as explicit as possible and expose raw json policy, just like how developers see it today.

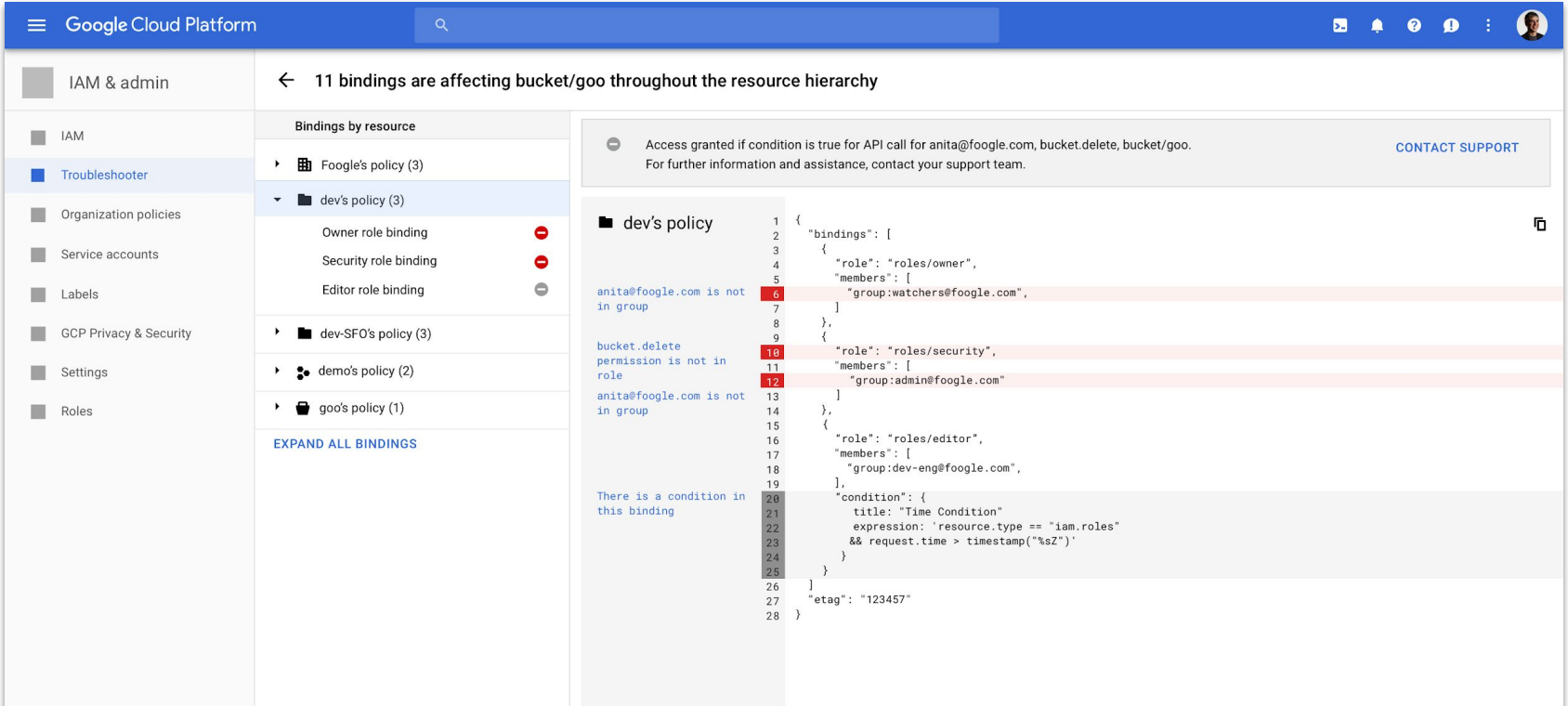
I took a look at some of the most competitive products on the market and understood a rising trend in policy-as-code.

I decided to mimic the policy-as-code trend as best as I could, while understanding the constraints of the API.

We couldn't provide a way to version control policy, but we *could* provide a way for customers to integrate with 3rd party tools that will allow them to.

```
{
  "bindings": [
    {
      "members": [
        "user:jie@example.com"
      ],
      "role": "roles/resourcemanager.organizationAdmin"
    },
    {
      "members": [
        "user:divya@example.com",
        "user:jie@example.com"
      ],
      "role": "roles/resourcemanager.projectCreator"
    }
  ],
  "etag": "BwUjMhCsNvY=",
  "version": 1
}
```


A preview of the end solution I designed.



Research 2.0

My researcher ran with my idea of exposing raw json (something that hadn't been done before within GCP), and we ran lots of customer sessions, often times changing up the design on the spot.

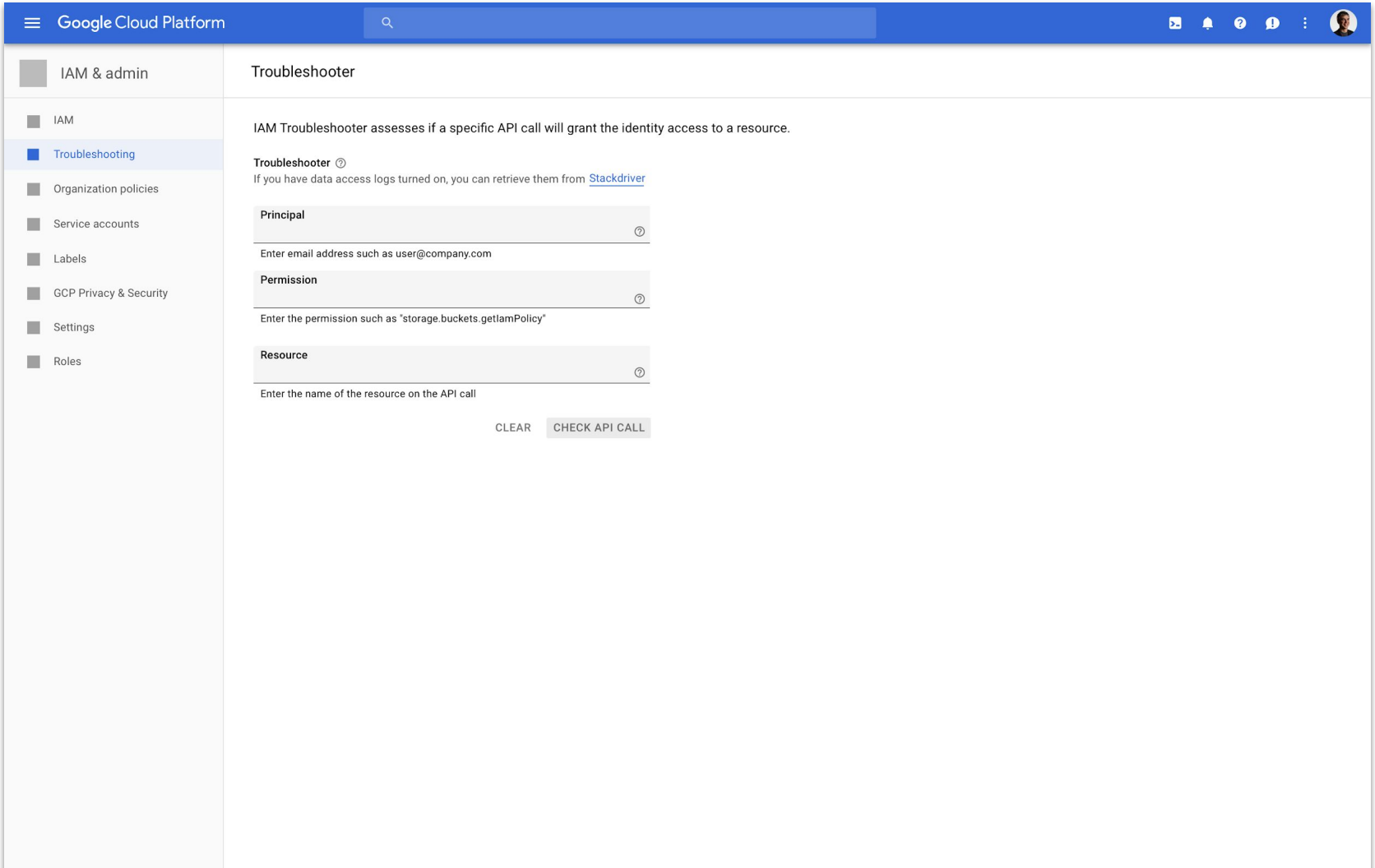
The PM of Troubleshooter was impressed by the amount of praise the new UI was getting from customers, and began to take the idea of exposing the raw policy and reusing the pattern in other areas of self driving cloud.

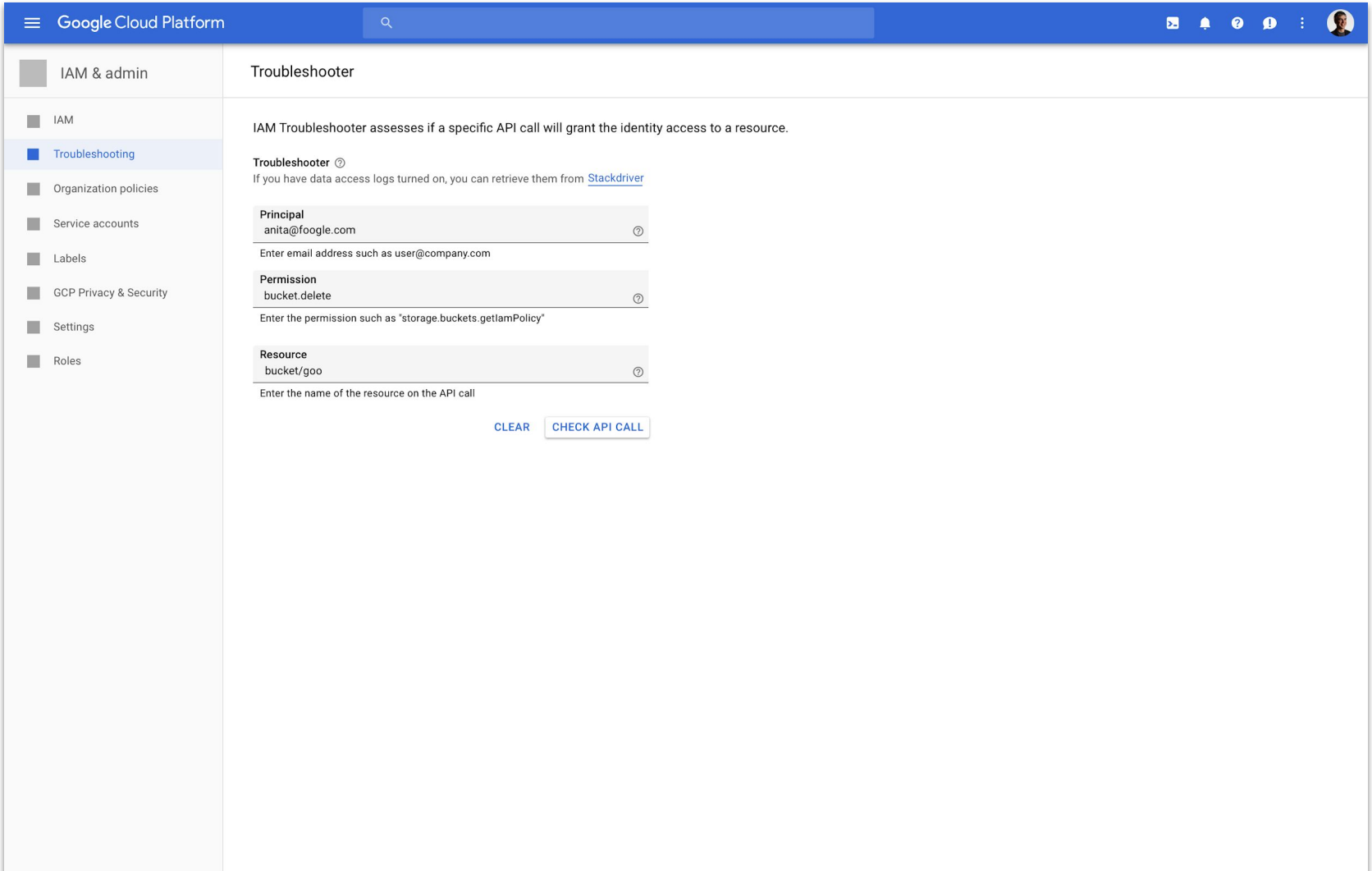
Expert interface. Looks just like an API response.

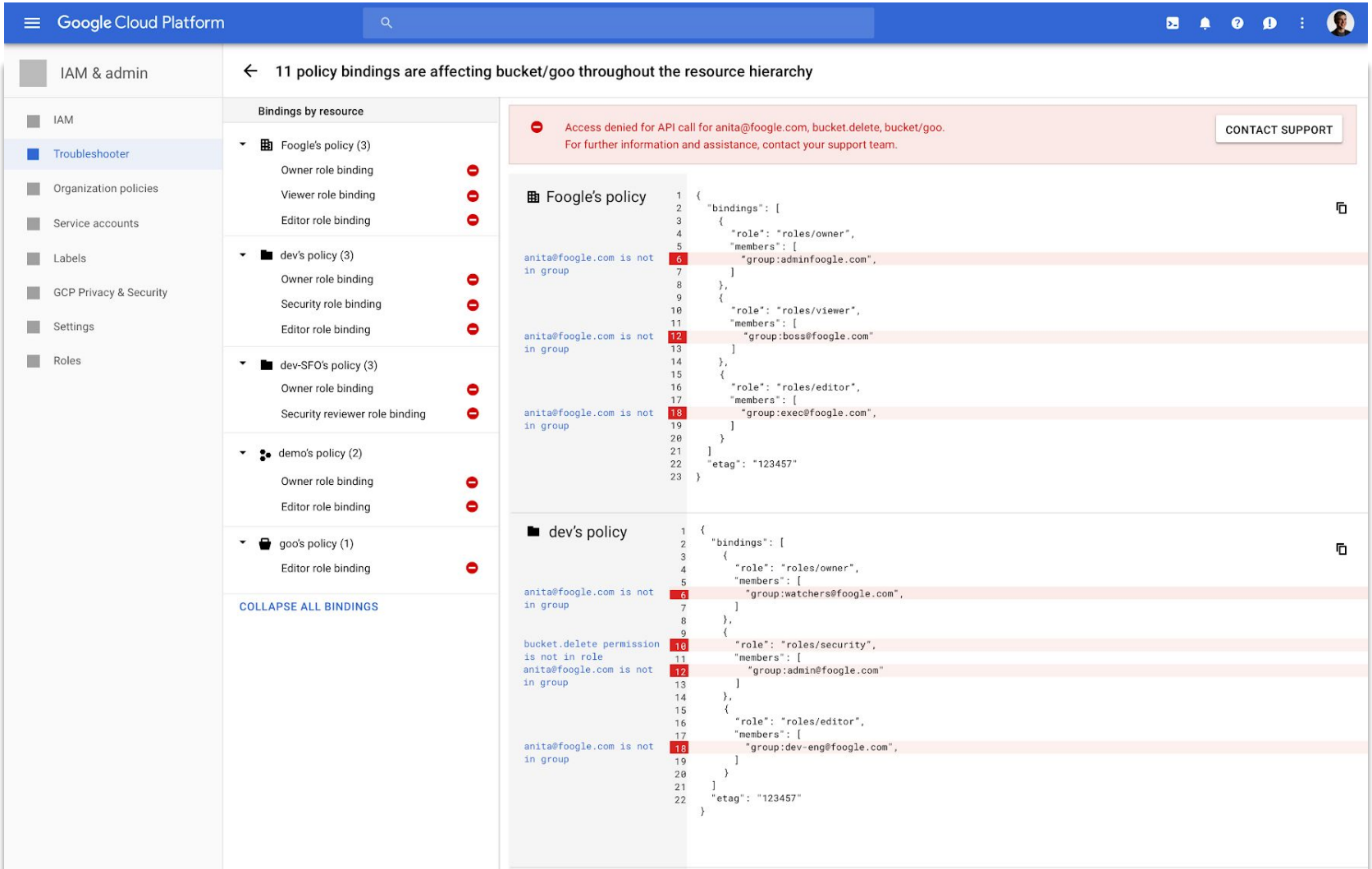
Customer on GCP
Name and Company redacted

(I swear they said this).

Now, let’s walk through the whole product from start to finish.


Google Cloud Platform
IAM & admin
IAM
Troubleshooting
Organization policies
Service accounts
Labels
GCP Privacy & Security
Settings
Roles
Troubleshooter
IAM Troubleshooter assesses if a specific API call will grant the identity access to a resource.
Troubleshooter
If you have data access logs turned on, you can retrieve them from Stackdriver
Principal
Enter email address such as user@company.com
Permission
Enter the permission such as "storage.buckets.getIamPolicy"
Resource
Enter the name of the resource on the API call
CLEAR
CHECK API CALL


Google Cloud Platform
IAM & admin
IAM
Troubleshooting
Organization policies
Service accounts
Labels
GCP Privacy & Security
Settings
Roles
Troubleshooter
IAM Troubleshooter assesses if a specific API call will grant the identity access to a resource.
Troubleshooter
If you have data access logs turned on, you can retrieve them from Stackdriver
Principal
anita@foogle.com
Enter email address such as user@company.com
Permission
bucket.delete
Enter the permission such as "storage.buckets.getIamPolicy"
Resource
bucket/goo
Enter the name of the resource on the API call
CLEAR
CHECK API CALL


Google Cloud Platform
IAM & admin
IAM
Troubleshooter
Organization policies
Service accounts
Labels
GCP Privacy & Security
Settings
Roles
11 policy bindings are affecting bucket/goo throughout the resource hierarchy
Bindings by resource
Foogle's policy (3)
Owner role binding
Viewer role binding
Editor role binding
dev's policy (3)
Owner role binding
Security role binding
Editor role binding
dev-SFO's policy (3)
Owner role binding
Security reviewer role binding
demo's policy (2)
Owner role binding
Editor role binding
goo's policy (1)
Editor role binding
COLLAPSE ALL BINDINGS
Access denied for API call for anita@foogle.com, bucket.delete, bucket/goo.
For further information and assistance, contact your support team.
CONTACT SUPPORT
Foogle's policy
anita@foogle.com is not in group
anita@foogle.com is not in group
anita@foogle.com is not in group
{
"bindings": [
{
"role": "roles/owner",
"members": [
"group:adminfoogle.com",
]
},
{
"role": "roles/viewer",
"members": [
"group:boss@foogle.com"
]
},
{
"role": "roles/editor",
"members": [
"group:exec@foogle.com",
]
}
]
"etag": "123457"
}
dev's policy
anita@foogle.com is not in group
bucket.delete permission is not in role
anita@foogle.com is not in group
anita@foogle.com is not in group
{
"bindings": [
{
"role": "roles/owner",
"members": [
"group:watchers@foogle.com",
]
},
{
"role": "roles/security",
"members": [
"group:admin@foogle.com"
]
},
{
"role": "roles/editor",
"members": [
"group:dev-eng@foogle.com",
]
}
]
"etag": "123457"
}

Roles

dev-SFO's policy (3)

Owner role binding

Security reviewer role binding

demo's policy (2)

Owner role binding

Editor role binding

goo's policy (1)

Editor role binding

COLLAPSE ALL BINDINGS

dev json policy copied

anita@foogle.com is not in group

```
{
  "bindings": [
    {
      "role": "roles/owner",
      "members": [
        "group:watchers@foogle.com",
      ]
    },
    {
      "role": "roles/security",
      "members": [
        "group:admin@foogle.com"
      ]
    },
    {
      "role": "roles/editor",
      "members": [
        "group:dev-eng@foogle.com",
      ]
    }
  ]
  "etag": "123457"
}
```

dev's policy

anita@foogle.com is not in group

bucket.delete permission is not in role

anita@foogle.com is not in group

anita@foogle.com is not in group

```
{
  "bindings": [
    {
      "role": "roles/owner",
      "members": [
        "group:watchers@foogle.com",
      ]
    },
    {
      "role": "roles/security",
      "members": [
        "group:admin@foogle.com"
      ]
    },
    {
      "role": "roles/editor",
      "members": [
        "group:exec@foogle.com",
      ]
    }
  ]
  "etag": "123457"
}
```

dev-SFO's policy

anita@foogle.com is not in group

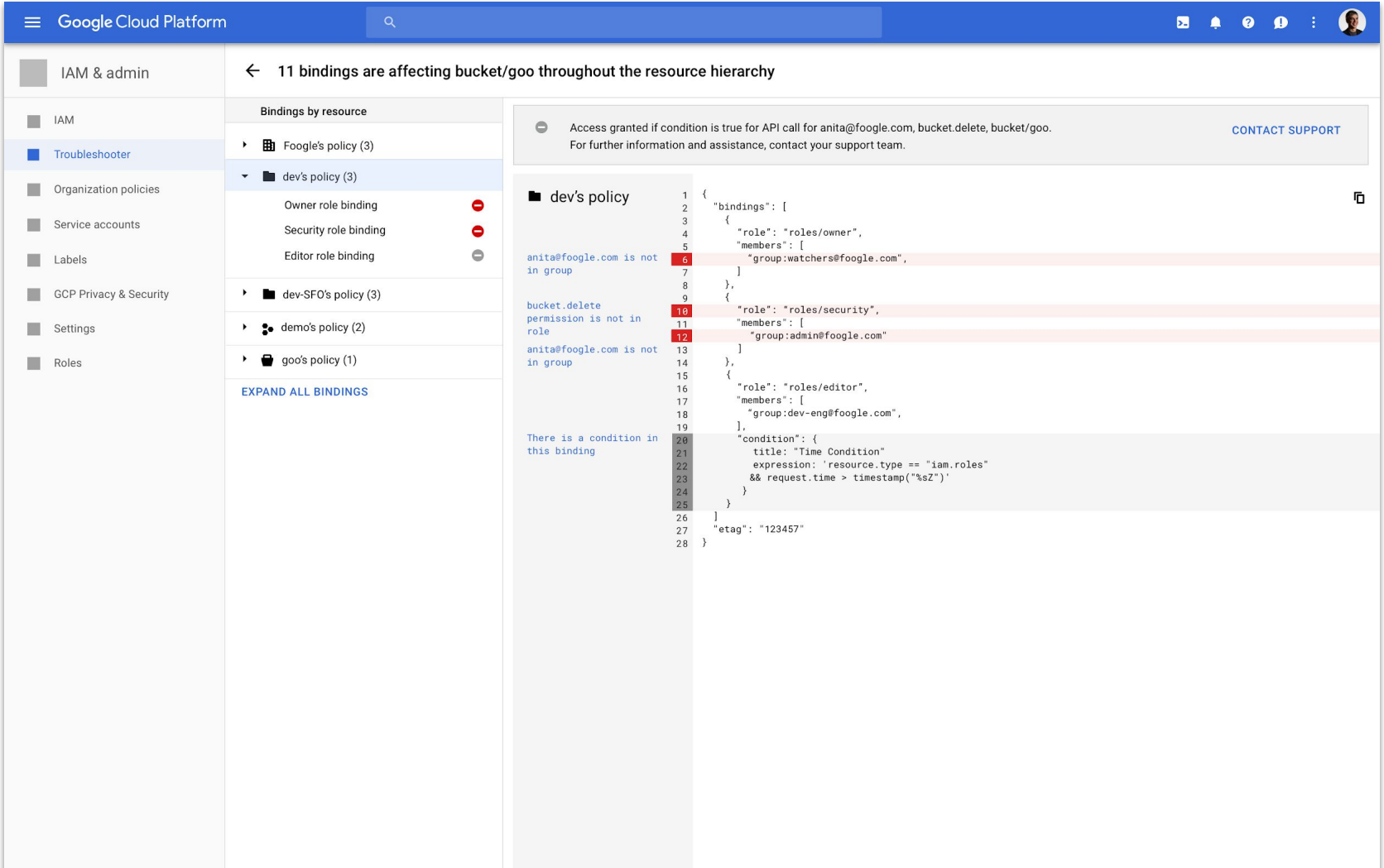
bucket.delete permission is not in role

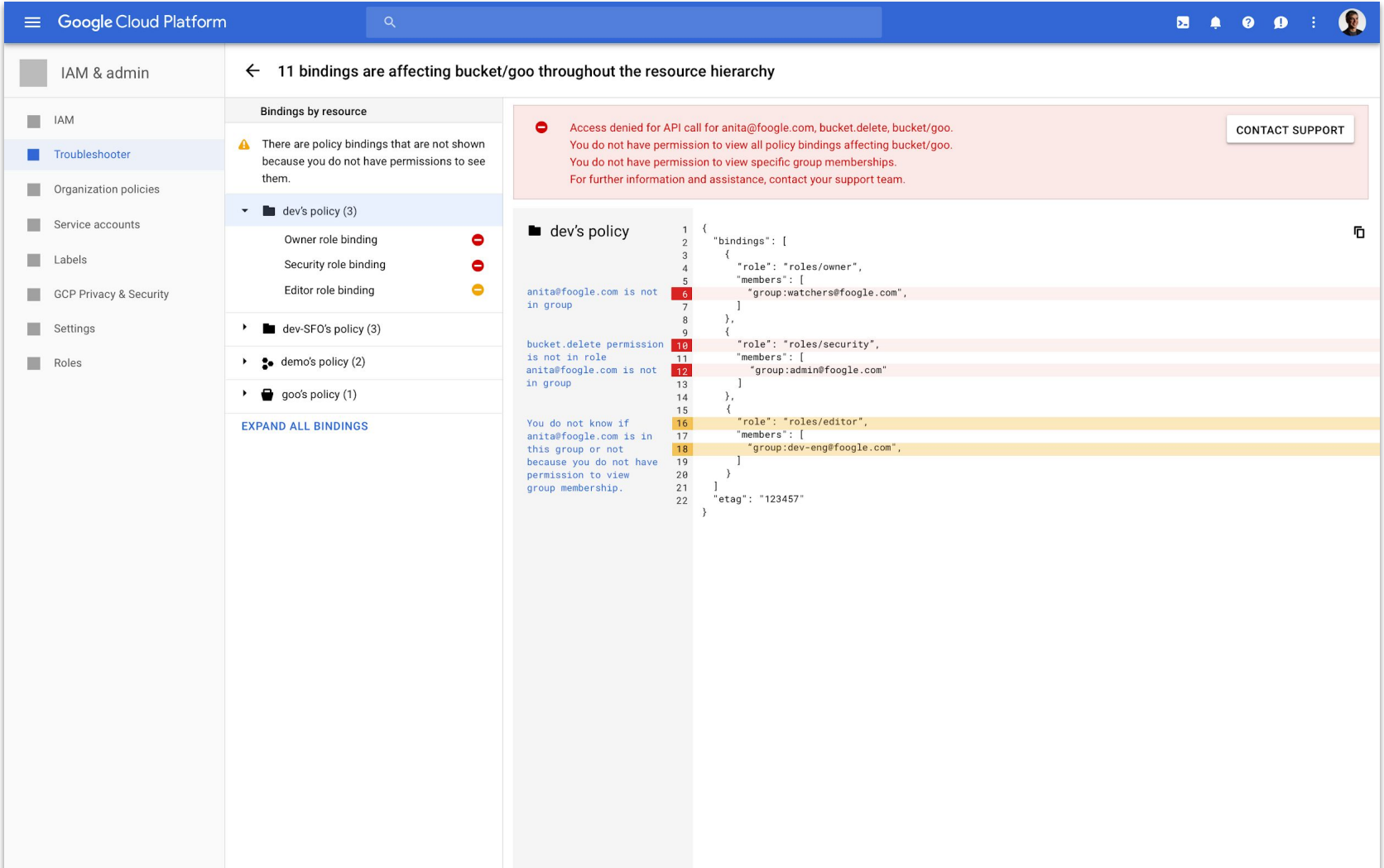
anita@foogle.com is not in group

```
{
  "bindings": [
    {
      "role": "roles/owner",
      "members": [
        "group:sfo-admin@foogle.com",
      ]
    },
    {
      "role": "roles/security-reviewer",
      "members": [
        "group:sfo-watchers@foogle.com"
      ]
    }
  ]
  "etag": "123457"
}
```

demo's policy

```
{
  "bindings": [
    {
```


Google Cloud Platform
IAM & admin
IAM
Troubleshooter
Organization policies
Service accounts
Labels
GCP Privacy & Security
Settings
Roles
11 bindings are affecting bucket/goo throughout the resource hierarchy
Bindings by resource
Foogle's policy (3)
dev's policy (3)
Owner role binding
Security role binding
Editor role binding
dev-SFO's policy (3)
demo's policy (2)
goo's policy (1)
EXPAND ALL BINDINGS
Access granted if condition is true for API call for anita@foogle.com, bucket.delete, bucket/goo.
For further information and assistance, contact your support team.
CONTACT SUPPORT
dev's policy
anita@foogle.com is not in group
bucket.delete permission is not in role
anita@foogle.com is not in group
There is a condition in this binding
{
"bindings": [
{
"role": "roles/owner",
"members": [
"group:watchers@foogle.com",
]
},
{
"role": "roles/security",
"members": [
"group:admin@foogle.com"
]
},
{
"role": "roles/editor",
"members": [
"group:dev-eng@foogle.com",
],
"condition": {
title: "Time Condition"
expression: 'resource.type == "iam.roles"
&& request.time > timestamp("%sZ")'
}
}
]
"etag": "123457"
}


Google Cloud Platform
IAM & admin
IAM
Troubleshooter
Organization policies
Service accounts
Labels
GCP Privacy & Security
Settings
Roles
11 bindings are affecting bucket/goo throughout the resource hierarchy
Bindings by resource
There are policy bindings that are not shown because you do not have permissions to see them.
dev's policy (3)
Owner role binding
Security role binding
Editor role binding
dev-SFO's policy (3)
demo's policy (2)
goo's policy (1)
EXPAND ALL BINDINGS
Access denied for API call for anita@foogle.com, bucket.delete, bucket/goo.
You do not have permission to view all policy bindings affecting bucket/goo.
You do not have permission to view specific group memberships.
For further information and assistance, contact your support team.
CONTACT SUPPORT
dev's policy
anita@foogle.com is not in group
bucket.delete permission is not in role
anita@foogle.com is not in group
You do not know if anita@foogle.com is in this group or not because you do not have permission to view group membership.
{
"bindings": [
{
"role": "roles/owner",
"members": [
"group:watchers@foogle.com",
]
},
{
"role": "roles/security",
"members": [
"group:admin@foogle.com"
]
},
{
"role": "roles/editor",
"members": [
"group:dev-eng@foogle.com",
]
}
]
"etag": "123457"
}

Debuted on stage at Google Cloud NEXT.

One of the top most used products within GCP, making close to a [redacted #] API calls a month.

Key Takeaways & Reflections

As the Lead Designer (and *only* designer) to ship a second self driving cloud product...

1

Accessibility

Since this is highly visual tool that really (and literally) highlights areas that helps developers debug 403 messages, in a mad rush to ship, I had forgotten the most *foundational* part of shipping any visual tool: accessibility.

I hadn't taken into account those were were colorblind.

2

Accessibility

I knew my mistake immediately after we shipped.

Unlike role recommender which had the traditional +/- pattern, (which by the way, is the standard way for those who are colorblind to go through a code diff), Troubleshooter didn't have that pattern.

3

Accessibility

I knew that icons were needed, along with a hidden tag within each icon that would describe what each denial message meant.

I deeply regretted this error, and as someone who prides myself n being an inclusive designer, I knew I had excluded a large portion of users..

Though I talk a lot about inclusivity in design, I hadn't even held myself accountable, and had prioritized shipping > doing right by the user. It ended up creating tech debt to fix the problem.

I will do much, *much* better next time.

As we all should.

Thank You