

The background features a vibrant, abstract design. On the left, there are horizontal, wavy bands of color in shades of red, orange, yellow, and green. On the right, a bright white light source emits a series of colorful rays in shades of blue, cyan, and yellow, creating a sunburst effect.

cisco *Live!*

Let's go

#CiscoLiveAPJC



The bridge to possible

API design principles

and considerations for massive scale

John M. Kuchta, Product Architect for Meraki Cloud Platform & API
TKIPisalegacypipher @ GitHub
DEVNET-1646



#CiscoLiveAPJC

Cisco Webex App

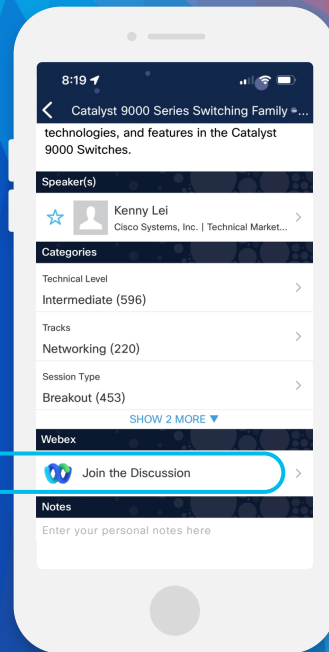
Questions?

Use Cisco Webex App to chat with the speaker after the session

How

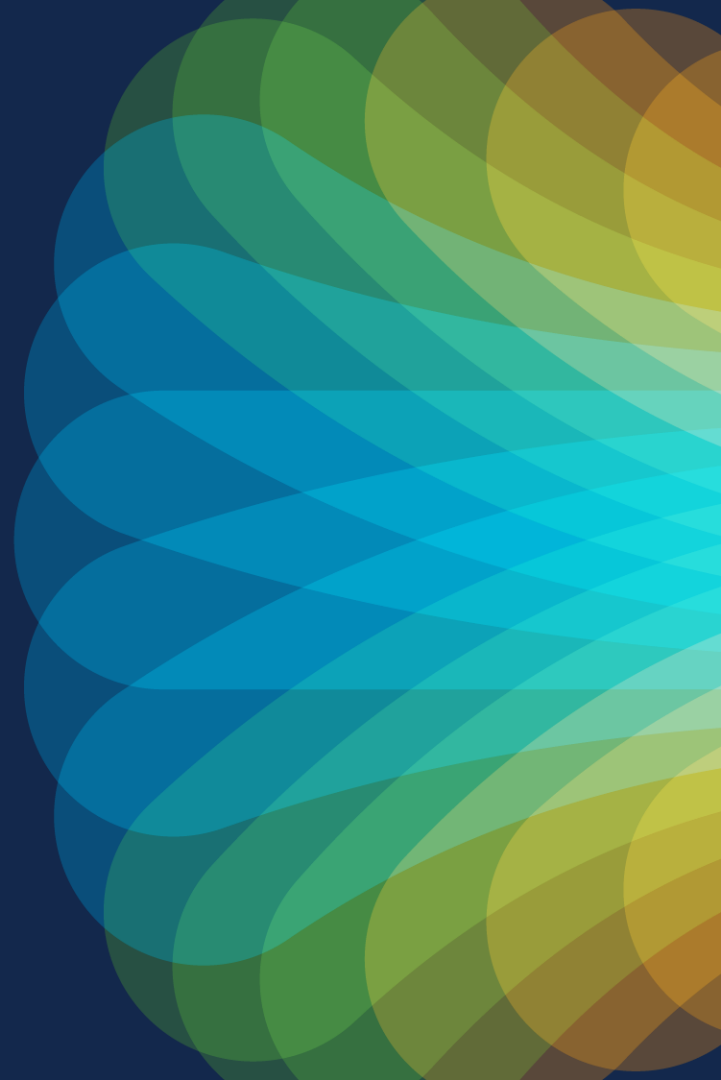
- 1 Find this session in the Cisco Live Mobile App
- 2 Click “Join the Discussion”
- 3 Install the Webex App or go directly to the Webex space
- 4 Enter messages/questions in the Webex space

Webex spaces will be moderated by the speaker until December 22, 2023.



<https://ciscolive.ciscoevents.com/ciscolivebot/#DEVNET-1646>

Introduction



What is this about?

- Software design
 - Meraki dashboard & REST API operations
 - Your own projects and applications consuming Meraki APIs or otherwise
 - Your own applications offering their own APIs
- User experience
 - When leveraging API operations in your own software, Meraki or otherwise
 - When Meraki leverages APIs in our own software (e.g. web & mobile apps)

Agenda

- Introduction
- 1. Naming matters
- 2. Signal is better than data
- 3. Aggregate by default
- 4. Filters are dope
- Conclusion
- Q&A

Who is this guy?



- John M. Kuchta
- Product Architect for Meraki Cloud Platform & APIs
- Cisco API strategy development & product management
- Developer consulting for partners, customers and Cisco engineering teams
- 12+ years in network and IT management

Why talk about this?

- Scalable solutions require creativity and holistic thinking.
- Better software means happier customers. Happier developers means better adoption.
- Reflection is key to growth. Our learnings, evidenced in our newest APIs, can benefit your developers.

Scale considerations for Meraki APIs

Business outcomes

- Does the API operation accelerate business outcomes?
- Is the API operation relevant to ecosystem partner, enterprise customer and M/SP use cases?

Large environments

- Does the API operation perform well for customers with large environments?
- Will customers with large environments find the API operation efficient?

Developer experience

- Is the API operation intuitive?
- Can developers adopt the API operation with minimal training or prior vendor-specific context?

Naming matters

Naming matters

Documentation is everything but naming is the most important part

- Poor naming complaints

- “It’s hard to know what a thing does, because its name doesn’t explain what it is.”
- “It’s hard to adopt a feature because learning the knobs and dials takes too much time.”

- Observations

- Developers don’t like using APIs that are confusing.
- Easy-to-read APIs make them easy to learn, trust, and adopt.

getNetwork WirelessSsid

```
1  [
2      {
3          "number": 0,
4          "name": "My SSID",
5          "enabled": true,
6          "ssidAdminAccessible": false,
7          "visible": true,
8          "localAuth": false,
9          "authMode": "8021x-radius",
10         "encryptionMode": "wpa",
11         "wpaEncryptionMode": "WPA2 only",
12         "splashPage": "Click-through splash page",
13         "adminSplashUrl": "http://example.com",
14         "splashTimeout": "30 minutes",
15         "walledGardenEnabled": true,
16         "walledGardenRanges": [
17             "example.com",
18             "1.1.1.1/32"
19         ],
20         "radiusEnabled": true,
21         "radiusServers": [
22             {
23                 "host": "0.0.0.0",
24                 "port": 3000,
25                 "openRoamingCertificateId": 2,
26                 "caCertificate": "-----BEGIN CERTIF
27             }
```

Naming matters

Guidelines for naming well

- A great name is **explicit**, **unambiguous** and **self-documenting**.
- Group related attributes into objects to **illustrate their relationships**.
- **Avoid abbreviations, acronyms**, and **homegrown shorthand**, each of which can confuse outside developers.

getNetwork WirelessSsid

Reimagined

```
3      "number": 0,  
4      "name": "My SSID",  
5      "enabled": true,  
6      "advertisement": {  
7          "enabled": true  
8      },  
9      "accessControl": {  
10         "encryption": {  
11             "type": "WPA2"  
12         },  
13         "authentication": {  
14             "mode": "RADIUS",  
15             "radius": {  
16                 "provider": "Meraki"  
17             }  
18         },  
19         "captivePortal": {  
20             "enabled": true,  
21             "type": "Click-through splash",  
22             "splash": {  
23                 "url": "http://example.com",  
24                 "timeout": 30,  
25                 "walledGarden": {  
26                     "enabled": true,  
27                     "addresses": [
```

getOrganizationDevicesUplinksAddressesByDevice

Why might you use this operation?

Naming matters

getOrganizationDevicesUplinksAddressesByDevice

- Attribute names are explicit and unambiguous. You don't need to be a Meraki expert to grok the data.
- Related attributes are grouped accordingly. A single device may have multiple uplinks, so we group the uplinks into an array within the device object.
- We've avoided any unnecessary abbreviations.

```
1  [
2    {
3      "mac": "00:11:22:33:44:55",
4      "name": "My MX",
5      "network": { "id": "N_24329156" },
6      "productType": "switch",
7      "serial": "Q234-ABCD-5678",
8      "tags": [ "tag1", "tag2" ],
9      "uplinks": [
10         {
11           "interface": "wan1",
12           "addresses": [
13             {
14               "protocol": "ipv6",
15               "assignmentMode": "static",
16               "address": "2001:db8:c15:c0:d0c::101:ca75",
17               "gateway": "fe80:db8:c15:c0:d0c::10ca:1d02",
18               "public": {
19                 "address": "2001:db8:c15:c0:d0c:15e:15e:bab1"
20               }
21             }
22           ]
23         }
24       ]
25     }
26   ]
27 ]
```

Naming matters

Takeaways

- Great names enable quick adoption and reduce op-ex.
- A great name is **explicit**, **unambiguous** and **self-documenting**.
- Group related attributes into objects to **illustrate their relationships**.
- **Avoid abbreviations, acronyms**, and **homegrown shorthand**, each of which can confuse outside developers.

Signal is better
than data

Signal is better than data

Improving SNR yields better returns than increasing data volume

- Poor SNR complaints

- “Finding the information that I need is like finding a needle in a haystack.”
- “I need more API budget in order to deal with the amount of noise in the data.”

- Observations

- A sea of positive data points can drown out the actionable pieces of negative data.
- Developers accustomed to working with noisy data usually ask for more API budget—to process more noise—but they don’t usually ask for better SNR on the data.
- You can reduce the number of calls necessary to deliver an insight if you increase the SNR of your data.

getOrganization DevicesStatuses

```
1  [
2      {
3          "name": "My AP",
4          "serial": "Q234-ABCD-5678",
5          "mac": "00:11:22:33:44:55",
6          "publicIp": "123.123.123.1",
7          "networkId": "N_24329156",
8          "status": "online",
9          "lastReportedAt": "2018-02-11T00:00:00.090210Z",
10         "lanIp": "1.2.3.4",
11         "gateway": "1.2.3.5",
12         "ipType": "dhcp",
13         "primaryDns": "8.8.8.8",
14         "secondaryDns": "8.8.4.4",
15         "productType": "wireless",
16         "components": {
17             "powerSupplies": [
18                 {
19                     "slot": 1,
20                     "serial": "QABC-1234-5678",
21                     "model": "PWR-MS320-1025WAC",
22                     "status": "powering",
23                     "poe": {
24                         "unit": "watts",
25                         "maximum": 740
26                     }
27                 }
28             ]
29         },
30         "model": "MR34",
31         "tags": [ "tag1", "tag2" ]
32     }
33 ]
34
```

Signal is better than data

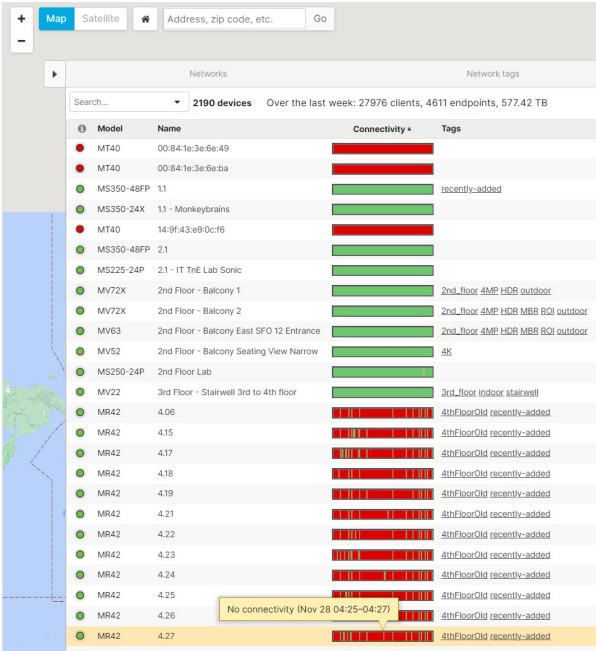
Raising the bar with higher SNR

- Identify the business use case to recognize the signal.
- Signal drives efficient outcomes.
- More data is not always better—less is often more!

What is availability?

A more explicit name for “status” or “connectivity”

- Availability measures the dashboard’s ability to interact with the device.
- Like “status” (online, alerting, offline) except:
 - Explicitly from the perspective of the dashboard, ignoring local device uptime or powered status
 - Used instead of “connectivity” which is frequently confused with other concepts such as a client’s web connectivity.



The screenshot shows a network management interface. At the top, there's a search bar with "Address, zip code, etc." and a "Go" button. Below that, a summary bar indicates "2190 devices" and "Over the last week: 27976 clients, 4611 endpoints, 577.42 TB". The main table lists devices with columns for Model, Name, Connectivity, and Tags. The Connectivity column uses a visual indicator: a green bar for online devices and a red bar for offline devices. A tooltip for device MR42 at 4.27 shows "No connectivity (Nov 28 04:25-04:27)".

Model	Name	Connectivity	Tags
MT40	00:84:1e:3e:6e:49	Online	
MT40	00:84:1e:3e:6e:ba	Online	
MS350-48FP	1.1	Online	recently-added
MS350-24X	1.1 - Monkeybrains	Online	
MT40	14:9f:43:e9:0c:f6	Online	
MS350-48FP	2.1	Online	
MS225-24P	2.1 - IT TrE Lab Sonic	Online	
MV72X	2nd Floor - Balcony 1	Online	2nd_floor 4MP HDR outdoor
MV72X	2nd Floor - Balcony 2	Online	2nd_floor 4MP HDR MBB BCI outdoor
MV63	2nd Floor - Balcony East SFO 12 Entrance	Online	2nd_floor 4MP HDR MBB BCI outdoor
MV52	2nd Floor - Balcony Seating View Narrow	Online	4K
MS250-24P	2nd Floor Lab	Online	
MV22	3rd Floor - Stairwell 3rd to 4th floor	Online	3rd_floor indoor stairwell
MR42	4.06	Offline	4thFloorOld recently-added
MR42	4.15	Offline	4thFloorOld recently-added
MR42	4.17	Offline	4thFloorOld recently-added
MR42	4.18	Offline	4thFloorOld recently-added
MR42	4.19	Offline	4thFloorOld recently-added
MR42	4.21	Offline	4thFloorOld recently-added
MR42	4.22	Offline	4thFloorOld recently-added
MR42	4.23	Offline	4thFloorOld recently-added
MR42	4.24	Offline	4thFloorOld recently-added
MR42	4.25	Offline	4thFloorOld recently-added
MR42	4.26	Offline	4thFloorOld recently-added
MR42	4.27	Offline	4thFloorOld recently-added

CISCO *Live!*

DEVNET-1646

getOrganization
Devices
Availabilities
ChangeHistory

**Statuses,
reimagined**

```
1  [
2  {
3
4      "ts": "2018-02-11T00:00:00.090210Z",
5      "device": {
6          "serial": "Q234-ABCD-5678",
7          "name": "My AP",
8          "productType": "wireless",
9          "model": "MR34"
10     },
11     "details": {
12         "old": [
13             {
14                 "name": "status",
15                 "value": "online"
16             }
17         ],
18         "new": [
19             {
20                 "name": "status",
21                 "value": "offline"
22             }
23         ]
24     },
25     "network": {
26         "id": "N_24329156",
27         "name": "Main Office",
28         "url": "https://n1.meraki.com/n//manage/nod
29     },
30     "tags": [ "tag1", "tag2" ]
31 }
```

getOrganization Configuration Changes

```
1  [
2      {
3          "ts": "2018-02-11T00:00:00.090210Z",
4          "adminName": "Miles Meraki",
5          "adminEmail": "miles@meraki.com",
6          "adminId": "212406",
7          "networkName": "Main Office",
8          "networkId": "N_24329156",
9          "ssidName": "My SSID",
10         "ssidNumber": 1,
11         "page": "via API",
12         "label": "PUT /api/v1/organizations/293",
13         "oldValue": "{\"id\":\"2930418\",\"name",
14         "newValue": "{\"id\":\"2930418\",\"name",
15     }
16 ]
```

Signal is better than data

Takeaways

- Signal drives efficient outcomes and efficient development.
- Identify the business use case to recognize the signal.
- Signal drives efficient outcomes.
- More data is not always better—less is often more!

Aggregate by default

Aggregate by default

API clients will aggregate data inefficiently given no other option

- Complaints from a lack of aggregation
 - “Collecting these datasets takes too much time.”
 - “Iterating over devices or networks in order to collect the complete picture takes too many API calls.”
- Observations
 - Data aggregation is inevitable.
 - The data provider is best positioned to aggregate the data.
 - Developers appreciate when you do the work for them.
 - If you don’t do the work, they will, and they’ll do it inefficiently.

getOrganization ApplianceUplinks UsageByNetwork

```
1  [
2      {
3          "networkId": "N_24329156",
4          "name": "Main Office",
5          "byUplink": [
6              {
7                  "serial": "Q234-ABCD-5678",
8                  "interface": "wan1",
9                  "sent": 200,
10                 "received": 400
11             },
12             {
13                 "serial": "Q234-ABCD-5678",
14                 "interface": "wan2",
15                 "sent": 150,
16                 "received": 10
17             },
18             {
19                 "serial": "Q234-ABCD-5678",
20                 "interface": "cellular",
21                 "sent": 20,
22                 "received": 22
23             }
24         ]
25     },
26     {
27         "networkId": "N_809546",
28         "name": "Branch",
29         "byUplink": [
30             {
31                 "serial": "Q234-ABCD-5691",
32                 "interface": "wan1",
```

Filters are dope

Filters are dope

Offer filters for your data and developers will love you for it.

- Complaints from a lack of filters
 - “Wading through these datasets takes too much time, or too many API calls.”
 - “Sometimes I need the whole picture, but usually, I only need a narrow slice to make a decision. Pulling down the whole dataset is a waste of bandwidth.”
- Observations
 - API consumers love filters but building them into APIs takes time and effort.
 - Nobody likes doing extra work.
 - Building filters can save you op-ex.

getOrganizationDevicesAvailabilitiesChangeHistory

Built-in filters

- Timespan, t0, t1
- Product types (switch, appliance, camera, etc.)
- Network IDs
- Statuses

```
1  [
2    {
3      "ts": "2018-02-11T00:00:00.090210Z",
4      "device": {
5        "serial": "Q234-ABCD-5678",
6        "name": "My AP",
7        "productType": "wireless",
8        "model": "MR34"
9      },
10     "details": {
11       "old": [
12         {
13           "name": "status",
14           "value": "online"
15         }
16       ],
17       "new": [
18         {
19           "name": "status",
20           "value": "offline"
21         }
22       ]
23     },
24     "network": {
25       "id": "N_24329156",
26       "name": "Main Office",
27       "url": "https://n1.meraki.com/n//manage/nodes/list",
28       "tags": [ "tag1", "tag2" ]
29     }
30   }
31 ]
```

getNetwork Settings

updateNetwork Settings

```
1  {
2      "localStatusPageEnabled": true,
3      "remoteStatusPageEnabled": true,
4      "localStatusPage": {
5          "authentication": {
6              "enabled": false,
7              "username": "admin"
8          }
9      },
10     "securePort": { "enabled": false },
11     "fips": { "enabled": true },
12     "namedVlans": { "enabled": true }
13 }
14
```

Resource actions for network configurations

Common implementation

- Read
 - **show** (GET) action to retrieve one resource
- Update (when required)
 - One PUT action to update one resource

Resource actions for network configurations

Improved implementation

- Read
 - **show** (GET) action to retrieve one resource
 - **index** (GET) action to retrieve **all the resources, aggregated**
- Update
 - One PUT action to update one resource

Resource actions for network configurations

Improved and efficient implementation

- Read
 - **index** (GET) action to retrieve all the resources, aggregated, **with optional filters!**
- Update
 - One PUT action to update one resource

updateNetwork ApplianceTraffic ShapingVpn Exclusions

```
1  {
2      "custom": [
3          {
4              "protocol": "tcp",
5              "destination": "192.168.3.0/24",
6              "port": "8000"
7          }
8      ],
9      "majorApplications": [
10         {
11             "id": "meraki:vpnExclusion/appli
12             "name": "Office 365 Sharepoint"
13         }
14     ]
15 }
16 |
```

getOrganizationApplianceTrafficShapingVpnExclusionsByNetwork

Why might you use this operation?

getOrganization ApplianceTraffic ShapingVpn ExclusionsBy Network

```
1  [
2      {
3          "networkId": "N_24329156",
4          "networkName": "Main Office",
5          "custom": [
6              {
7                  "protocol": "tcp",
8                  "destination": "192.168.3.0",
9                  "port": "8000"
10             }
11         ],
12         "majorApplications": [
13             {
14                 "id": "meraki:vpnExclusion/",
15                 "name": "Office 365 Sharepo
16             }
17         ]
18     }
19 ]
```

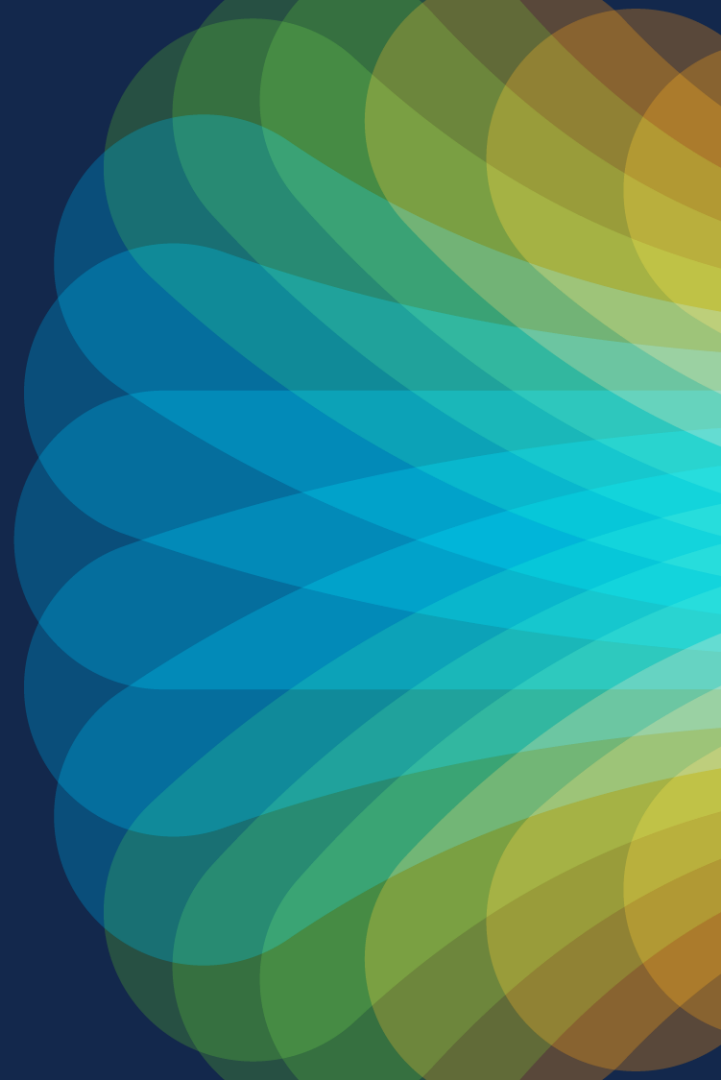
getOrganization SwitchPortsBy Switch

```
2   [
3   {
4       "name": "Example Switch",
5       "serial": "Q555-5555-5555",
6       "mac": "01:23:45:67",
7       "network": {
8           "name": "Example Network",
9           "id": "N_12345"
10      },
11      "model": "MS120-8",
12      "ports": [
13          {
14              "portId": "1",
15              "name": "My switch port",
16              "tags": [
17                  "tag1",
18                  "tag2"
19              ],
20              "enabled": true,
21              "poeEnabled": true,
22              "type": "access",
23              "vlan": 10,
24              "voiceVlan": 20,
25              "allowedVlans": "1,3,5-10",
26              "rstpEnabled": true,
27              "stpGuard": "disabled",
28              "linkNegotiation": "Auto negotiate",
29              "accessPolicyType": "Sticky MAC allow list",
30              "stickyMacAllowList": [
31                  "34:56:fe:ce:8e:b0",
32                  "34:56:fe:ce:8e:b1"
33              ]
34          }
35      ]
36  }
```

Aggregating & filtering

- Do the work for the customer
- Data providers are best positioned to aggregate data.
- Filters can multiply the utility of your API operations.
- Leveraging Meraki aggregates and filters can massively improve your application's efficiency.
- Shipping filterable aggregate API operations can reduce op-ex.

Conclusion



Recap

Four principles to remember

- Naming matters
- Signal is better than data
- Aggregate by default
- Filters are dope



Did you know?

You can have a
one-on-one session with
a technical expert!

Visit Meet the Expert in The HUB
to meet, greet, whiteboard & gain
insights about your unique questions
with the best of the best.



Meet the Expert Opening Hours:

Tuesday	3:00pm – 7:00pm
Wednesday	11:15am – 7:00pm
Thursday	9:30am – 4:00pm
Friday	10:30am – 1:30pm

Session Surveys

We would love to know your feedback on this session!

- Complete a minimum of four session surveys and the overall event surveys to claim a Cisco Live T-Shirt



Continue your education



- Visit the Ecosystem Station at the Cisco Networking booth for a chance to win an MV63 camera!
- Visit the Cisco Showcase for related demos
- Book your one-on-one Meet the Expert meeting
- Attend the interactive education with DevNet, Capture the Flag, and Walk-in Labs
- Visit the On-Demand Library for more sessions at www.CiscoLive.com/on-demand



The bridge to possible

Thank you

CISCO *Live!*

#CiscoLiveAPJC

The background features a vibrant, multi-colored abstract design. On the left, there are horizontal, wavy bands of color in shades of red, orange, yellow, and green. On the right, a bright white light source emits a series of colorful rays in shades of blue, green, and yellow, creating a sunburst effect.

cisco *Live!*

Let's go

#CiscoLiveAPJC