CISCO

You make **possible**

Introduction to Developing Apps for Cisco ACI App Center

Alec Chamberlain
Damon Li

Devnet-1136

Barcelona | January 27-31, 2020

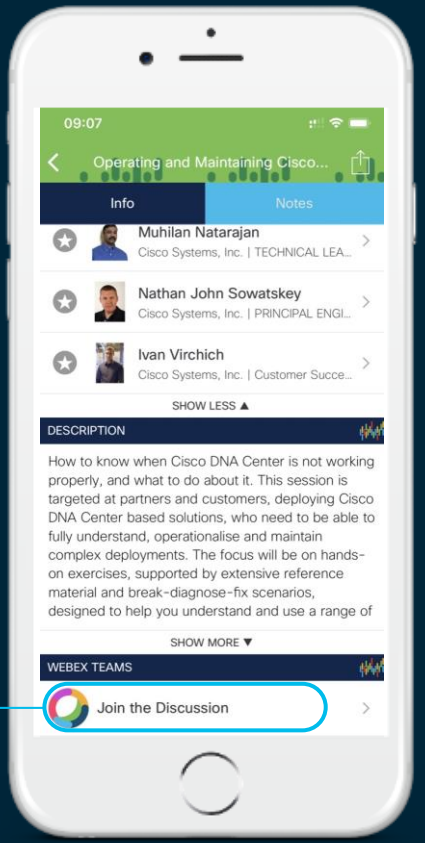
Cisco Webex Teams

Questions?

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

How

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



Agenda

- Developing a stateless app
- Developing a stateful app
- Packaging and uploading an app
- Use Cases
- Network Insights
- Key Takeaways

What is ACI App Center

- An open and programmable infrastructure
  - Open API for Value Added applications
- Driving factors – openness and extensibility of ACI
- Download and get started:

aciappcenter.cisco.com

developer.cisco.com/site/aci/docs/app-center/getting-started/

Why ACI App Center

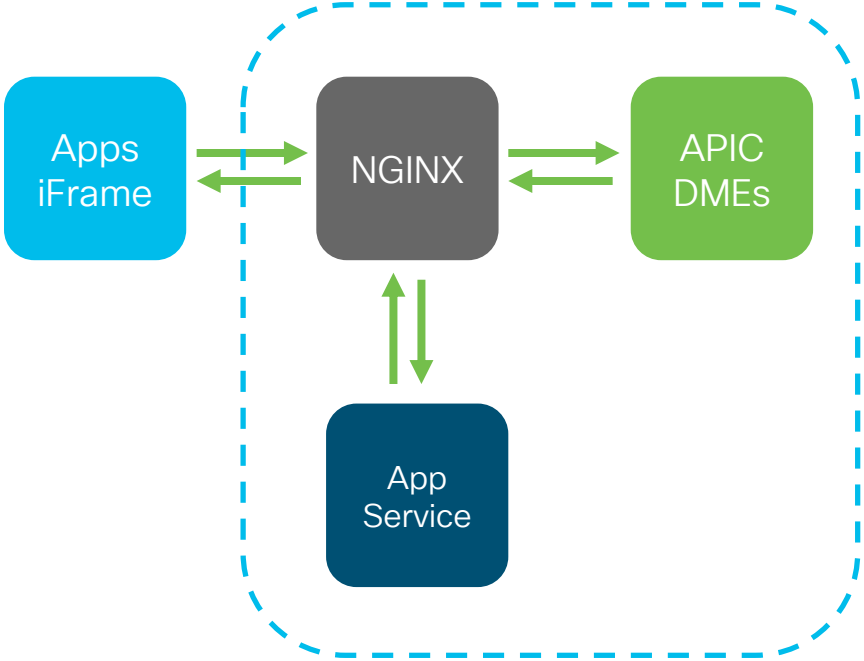
- Custom applications based on *your* business requirements
- Embed custom apps within the APIC GUI
- Who writes them?
  - Partners
  - Community
  - Customers
  - Third Party
- Augment base software functionality
- Simplify software development, delivery and consumption

Developing a stateless app

Type of Apps

Stateless Apps

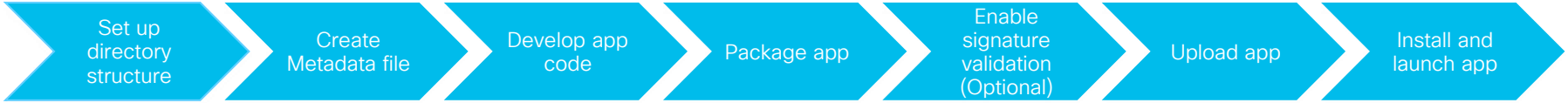
Stateful Apps



What is a stateless app

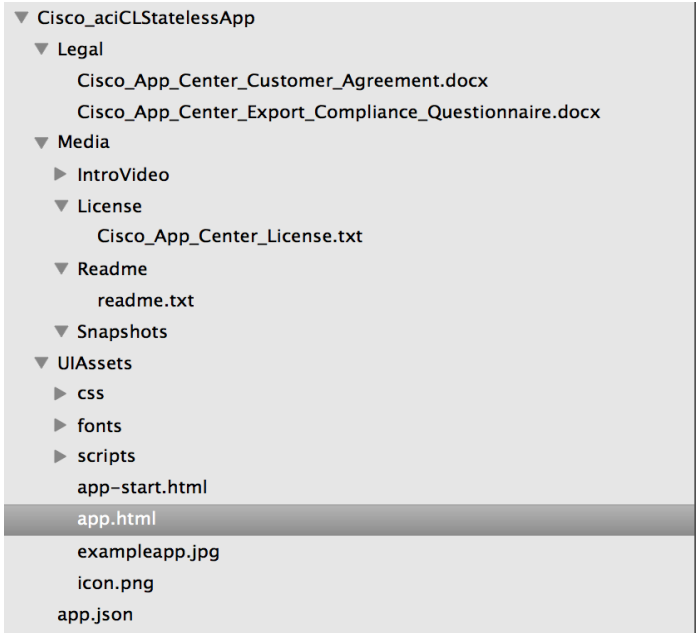
- Stateless apps are primarily HTML, CSS and JS front-end.
- Run as part of the APIC UI inserted as an iFRAME
- Can be inserted in any part of the APIC UI or Apps Tab
- You develop your custom apps on your own development environment, not directly on APIC.
- APIC expects a zipped file with .aci extension. A packager utility should be used to package your app into .aci extension.

Stateless app development workflow



Setup the directory structure

- Stateless apps must follow a specific directory structure.
- On your development box, create a simple directory structure as the example:



Create metadata

- app.json contains your app's metadata
- It must adhere to a specific format and contain certain mandatory variables
- The metadata describes what your app does, its insertion point in the UI, the privileges required, the minimum APIC version it requires, its revision number, who authored it, its category, etc.
- By default, custom apps are displayed only under the App navbar menu in APIC UI

Sample metadata

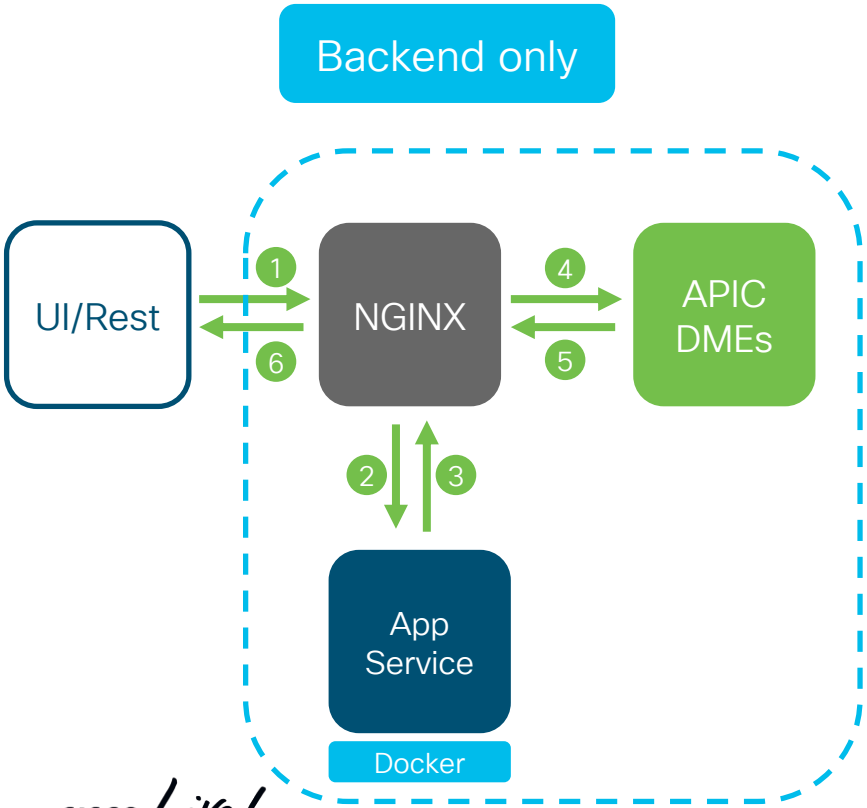
```
{
    "apicversion":"2.2(1n)",
    "appid":"aciCLStatelessApp",
    "author":"Christine Lakits",
    "category":[
        "Tools and Utilities",
        "Visibility and Monitoring"
    ],
    "contact":{
        "contact-email":"clakits@cisco.com"
    },
    "iconfile":"icon.png",
    "name":"aciCLStatelessApp",
    "permissions":[
        "admin"
    ],
    "permissionslevel":"read",
    "shortdescr":"Example Stateless App",
    "vendor":"Cisco",
    "vendordomain":"Cisco",
    "version":"1.0"
}
```

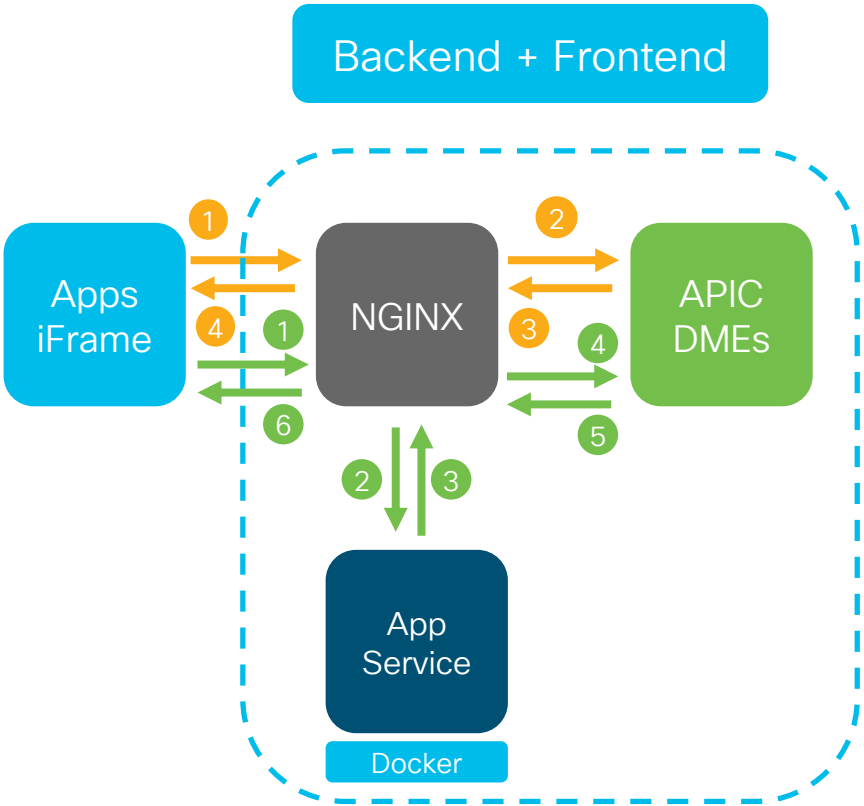
Mandatory parameters:

- appid
- version
- iconfile
- name
- shortdescr
- vendor
- vendordomain
- apicversion
- permissions
- permissionslevel
- api (only for the stateful app)
- author
- category

Developing a stateful app

Stateful Apps – with or without front end





Stateful app development workflow

Setup Directory structure

- Stateful apps must follow a specific directory structure.
- On your development box, create a simple directory structure as the example:

```
▼ Cisco_aciCLStatefulApp
  ▼ Image
      aci_appcenter_docker_image.tgz
  ▼ Legal   <- Files are added automatically with the packager
      Cisco_App_Center_Customer_Agreement.docx
      Cisco_App_Center_Export_Compliance_Questionnaire.docx
  ▶ Media   <- Cisco license file / Help for the App
  ▼ Service
      server.py
      start.sh
  ▼ UIAssets
    ▶ css
    ▶ fonts
    ▶ scripts
      app-start.html   <- From Cisco DevNet for token mechanism
      app.html   <- Main HTML file
      icon.png
    app.json   <- Application Metadata
```

Create the docker image

- You have the options to use reference image provided by Cisco, or create your own image (should not exceed 1 GB)
- One Docker image per application and communication is not allowed between different containers
- You spawn a container from the image on your dev machine
  - install libraries you need in it, commit and save

Developing the Code

The `start.sh` starts the backend service – `server.py` in this case – It calls the python Flask server. You can have another name for your backend app script.

```
#!/bin/sh

# Run the server
python /home/app/src/Service/server.py

```

Developing the code

```python
from flask import Flask
from cobra.mit.access import MoDirectory
from cobra.mit.session import CertSession
from cobra.mit.session import LoginSession
from cobra.model.pol import Uni as PolUni
from cobra.model.aaa import UserEp as AaaUserEp
from cobra.model.aaa import AppUser as AaaAppUser
from cobra.model.aaa import UserCert as AaaUserCert
from cobra.internal.codec.jsoncodec import toJSONStr, fromJSONStr
from cobra.internal.codec.xmlcodec import _toXMLStr, fromXMLStr

import json
import logging

app = Flask(__name__)
```

```python
@app.route('/')
def hello_world():
    ''' Test the connectivity.
    '''
    logging.info('Received API Request from Client - /')
    return 'Cisco HelloACI PlugIn Version 1.0.'
```

```python
if __name__ == '__main__':
    # Setup logging
    fStr='%(asctime)s %(levelname)5s %(message)s'
    logging.basicConfig(filename='/home/app/log/helloaci.log', format=fStr, level=logging.DEBUG)

    # Run app flask server
    app.run(host='0.0.0.0', port=80)
```

Sample Code Snippet

```python
@app.route('/testAPI.json')
def hello_world():
    ''' Test the connectivity.
    '''
    logging.info('Received API Request from Client - /')
    return "You have reached the docker container, it\'s alive!"

@app.route('/getTenant.json')
def get_tenant():
    try:
        cookie = requestAppToken()

        reply = getClass('fvTenant', cookie)

        tenants = []

        for tenant in reply['imdata']:
            tenants.append(tenant['fvTenant']['attributes']['name'])

        return json.dumps(tenants)

    except Exception as e:
        import traceback
        logging.info(e)
        logging.info(traceback.format_exc())
        return 'Error: \n{} \n{}'.format(e, traceback.format_exc())



if __name__ == '__main__':
    # Setup logging
    fStr='%(asctime)s %(levelname)5s %(message)s'
    logging.basicConfig(filename='/home/app/log/server.log', format=fStr, level=logging.DEBUG)

    # Run app flask server
    app.run(host='0.0.0.0', port=80)
```

```html
<script type="text/javascript">
  function testAPI() {
    var query_url = window.BACKEND_QUERY_URL + '/testAPI.json'
    $.ajax({
      url: query_url,
      type: 'GET',
      headers: {'DevCookie': window.APIC_DEV_COOKIE, 'APIC-challenge': window.APIC_URL_TOKEN},
      success: function(data){
        console.log(data);
        $("#docker-running-ok").css('display', '');
        $("#reply_testAPI").text(data)
      },
      error: function(error){
        console.log("===ERROR");
        console.log(error);
        $("#docker-running-nok").css('display', '');
      },
    });
  }

  testAPI();
</script>
```

```
window.BACKEND_QUERY_URL = document.location.origin + "/appcenter/Cisco/aciCLStatefulApp";
```


AppCreator

- Automatically create skeleton (HelloWorld) stateless and stateful app
- Includes all required directories and files
- Includes docker image for stateful app

```
*****************************************
*              ACI App Creator          *
*****************************************
Welcome! This tool will guide you through the creation of a fully functional ACI App Center application.
The information that you will provide can be changed later on, please read the ACI App Center Developer Guide to learn more about it.

--- General information ---
Let's begin! What should be the name of your application? (e.g. "TestingApp", "MyFirstApp",...)
Note: The name of the application will also be used as the application ID.
This can be changed afterwards in "app.json".
> (Application name)
```

Packaging the App

Packaging your app

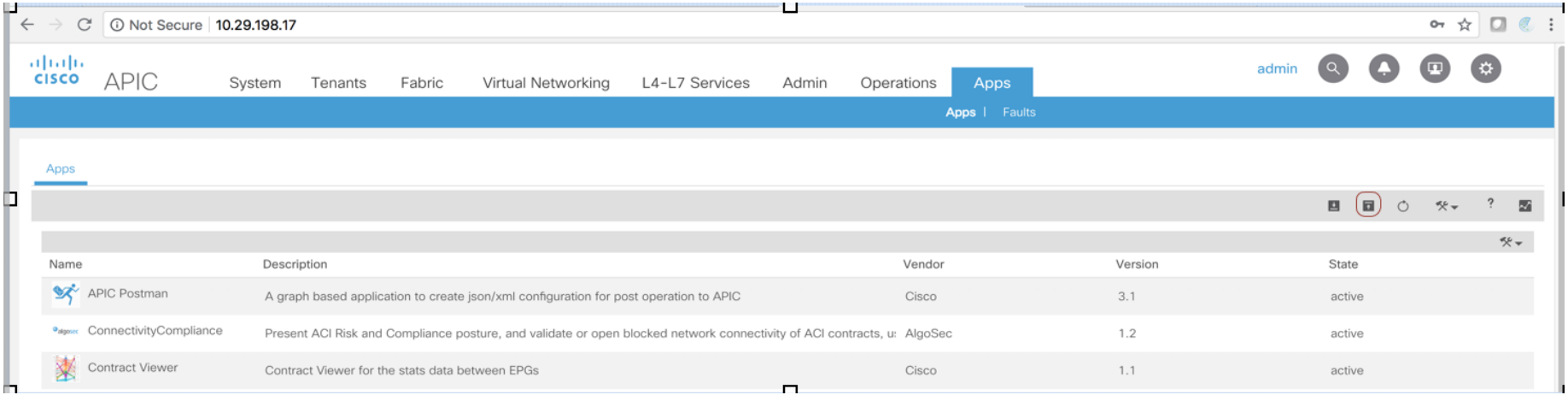
- Make sure you have `app.json` in the main directory, check your directory structure, check readme and license files, and verify that you have `app.html` and `app-start.html` and you app's icon under `UIAssets`.
- Mare sure you have the docker image in the image directory, and `start.sh` and `server.py` in the service directory if it's a stateful app.
- Get packager cisco_aci_app_packager-1.0.tar.gz from Cisco DevNet.
- Run the packager to package your application.

```
$ python aci_app_packager.py -F /Users/vyordano/CiscoLive/Devnet-1136/ACI_APP
```

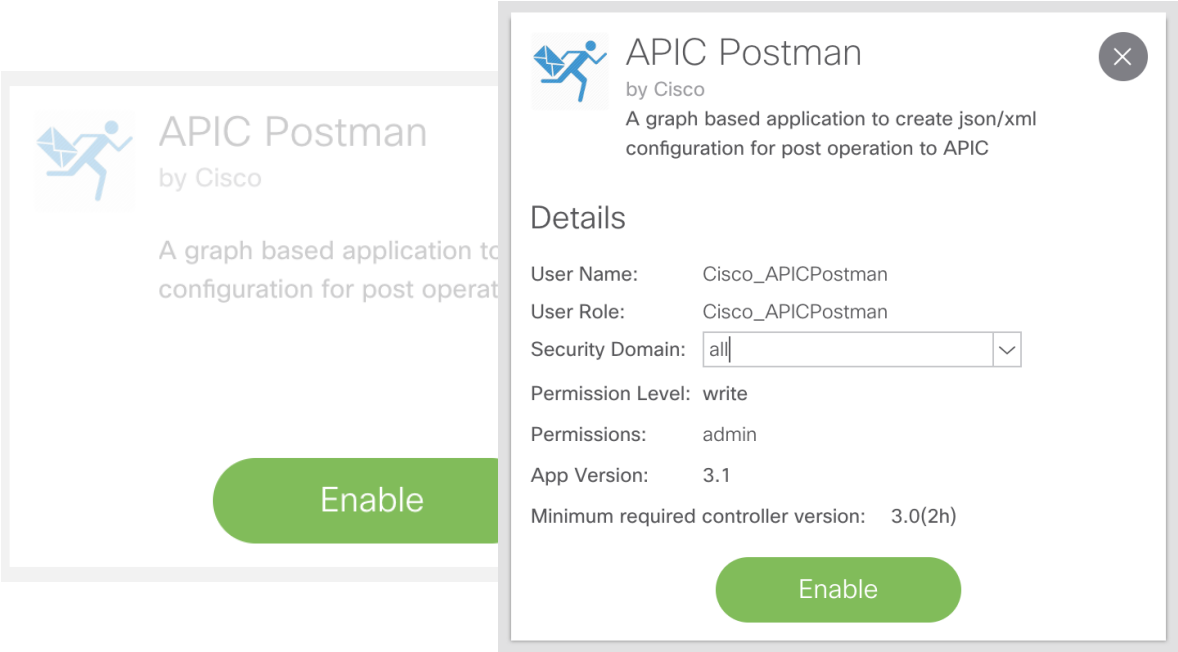
Upload and Install

Uploading the App

Upload the package to APIC (Apps → Apps → upload sign), then you will see your new app



Enabling the app



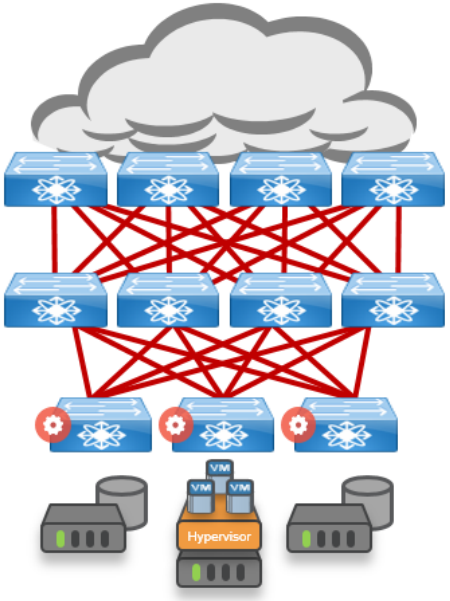
Use Cases

Typical use cases

Deployment Automation	Monitoring	Troubleshooting	Deployment Verification
Automate common network provisioning tasks so they don't require human intervention	Monitor things typically not available via standard mechanisms	Pull data out of the fabric that provides the necessary context for troubleshooting	Make sure there haven't been mistakes in the deployment

Real World App Ideas

Network Visibility Is Hard



SNMP

syslog

CLI

- Hard to Operationalize
- Incomplete
- Unstructured
- Device-Specific
- Slow

Ripe For Disruption

What has not changed

Use Cases

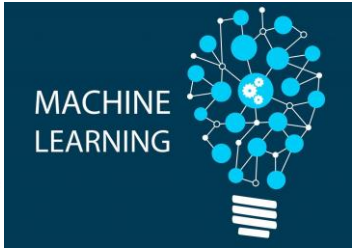
- Network Health
- Anomaly detection
- Troubleshooting/Remediation
- SLA, Performance Tuning
- Capacity Planning
- Security

What has changed

Industry Trends

- Real time statistics
- Software Defined Networking
- Distributed Computing

Technology Advancements



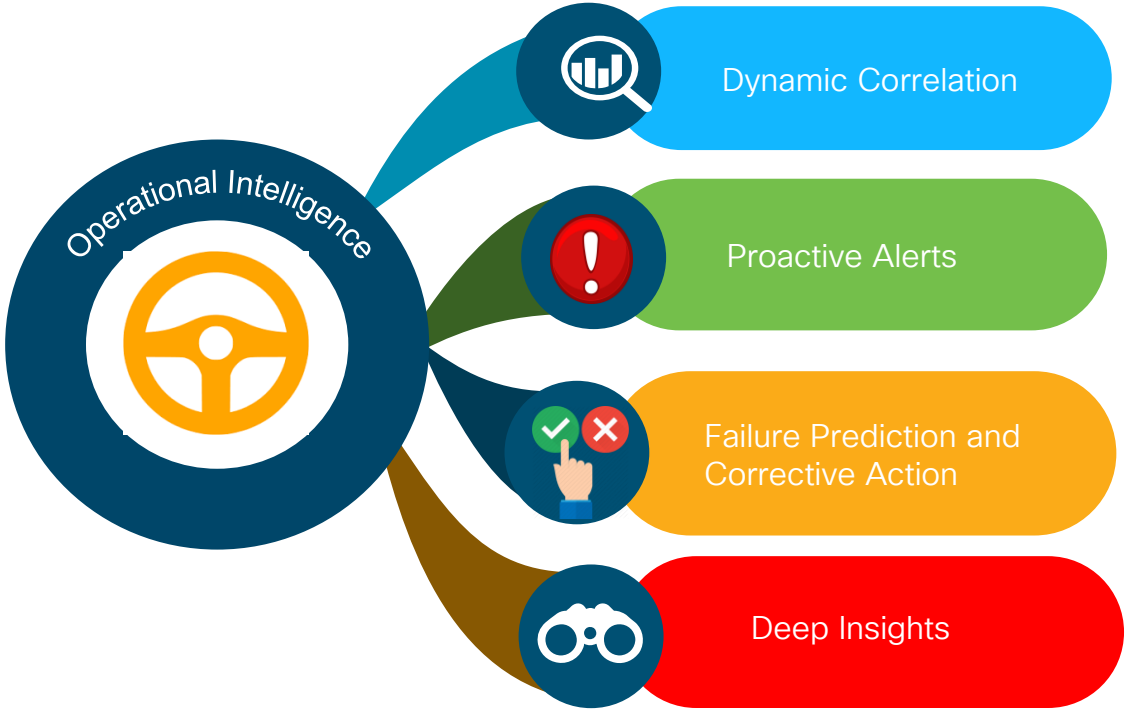
Operation Intelligence Engine

Dynamic Correlation
Correlate information across data sources

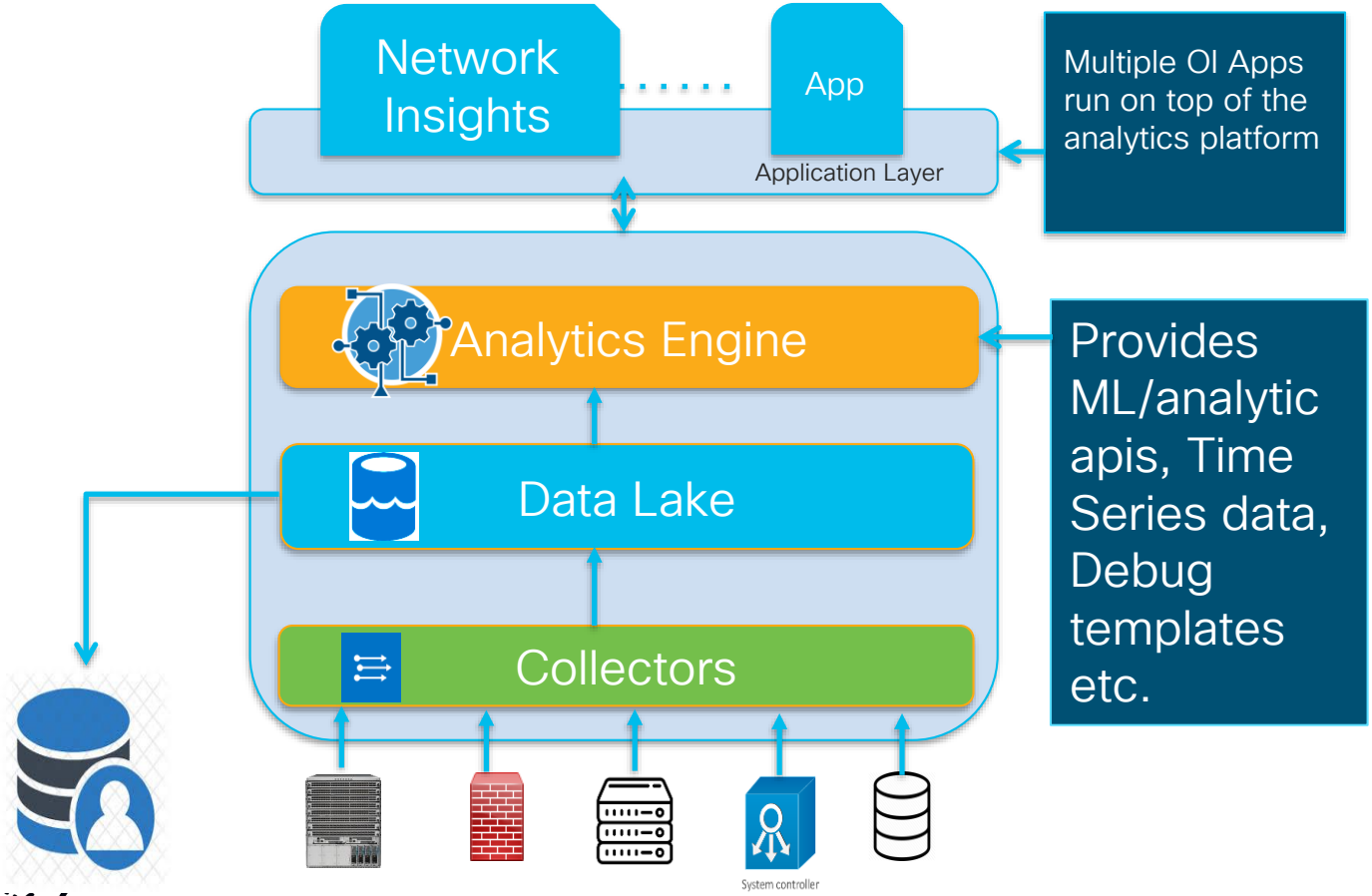
Proactive Alerts
See problems before end users do and alert

Failure Prediction & Corrective Action
Ability to predict failure and provide corrective action

Deep Insights
Ability to discover information with ease



Operation Intelligence as a Service



Data Center Use Cases

Health Reporting

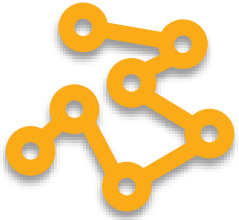
- Network Health
- Resource Utilization with Forecasting alerts
- Protocol and Interface state/events with correlations
- Environmental data
- Trend analysis on Resource usage

Performance Monitoring

- Network Performance
- Interface utilization
- Buffer monitoring
- Microburst detection
- Drop event correlation
- QoS Stats

Visibility

- Path and Latency Measurement
- End-to-end visibility
- Path tracing over time
- Flow latency monitoring
- Ftriage & Trace Route Support



Network Insights Resources

Network Insights Resources - NIR

Monitoring, Capacity Planning & Troubleshooting

Solution Visibility

- Switch Resources
- End-to-end-resources
- Event Correlation
- Resource & Event History

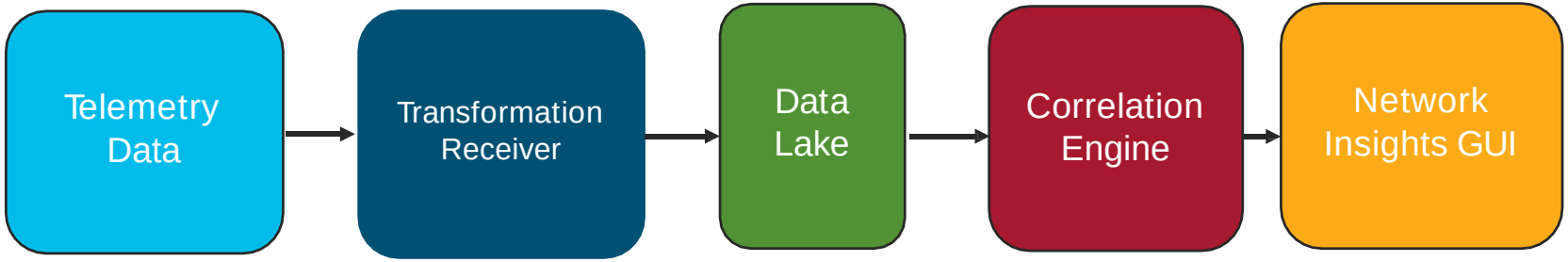
Troubleshooting/Operations

- Flow-issues
- Event analysis
- Statistics (interfaces etc..)

Network Insights Resources – Customer Benefits

System	Resource Utilization	Fabric-Wide Capacity Planning, Trend Monitoring
System	Environmental Monitoring	Avoid Environmental (CPU, Power, Memory, Fan, Storage Related Failures
Operations	Statistics	Identify/Predict Failing Devices
Operations	Flow Analytics	Troubleshoot Application Latency Identify Traffic/Protocol behavior
Operations	Event Analytics	Identify Subtle Path-Related issues

Network Insights Resources App Architecture



- GPB over Kafka
- Flow tables over UDP
- Buffer and queue stats over UDP

Collect and Normalize Telemetry data

Store correlated data for use by GUI

Analyze

Visualize

Microservices

NIR Demo

Learn more about the new DevNet Certifications and how you can prepare now!

	Associate Level	Specialist Level	Professional Level	Expert Level
Engineering	CISCO CERTIFIED CCNA	CISCO CERTIFIED SPECIALIST	CISCO CERTIFIED CCNP	CISCO CERTIFIED CCIE
Software	CISCO CERTIFIED DEVNET Associate	CISCO CERTIFIED DEVNET SPECIALIST	CISCO CERTIFIED DEVNET Professional	CISCO CERTIFIED DEVNET Expert

Future Offering

Start Here | Upcoming Cisco DevNet Certifications

- Start at **Meet DevNet**

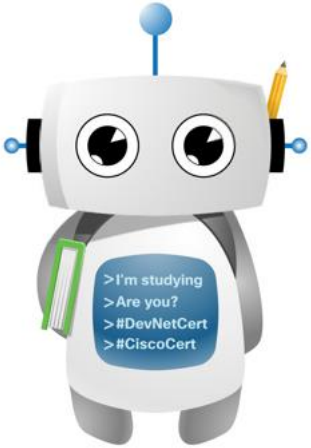
  DEVNET-2864: Getting ready for Cisco DevNet Certifications
  Offered daily at 9am, 1pm & 4pm at Meet DevNet

- Attend a **brownbag session**

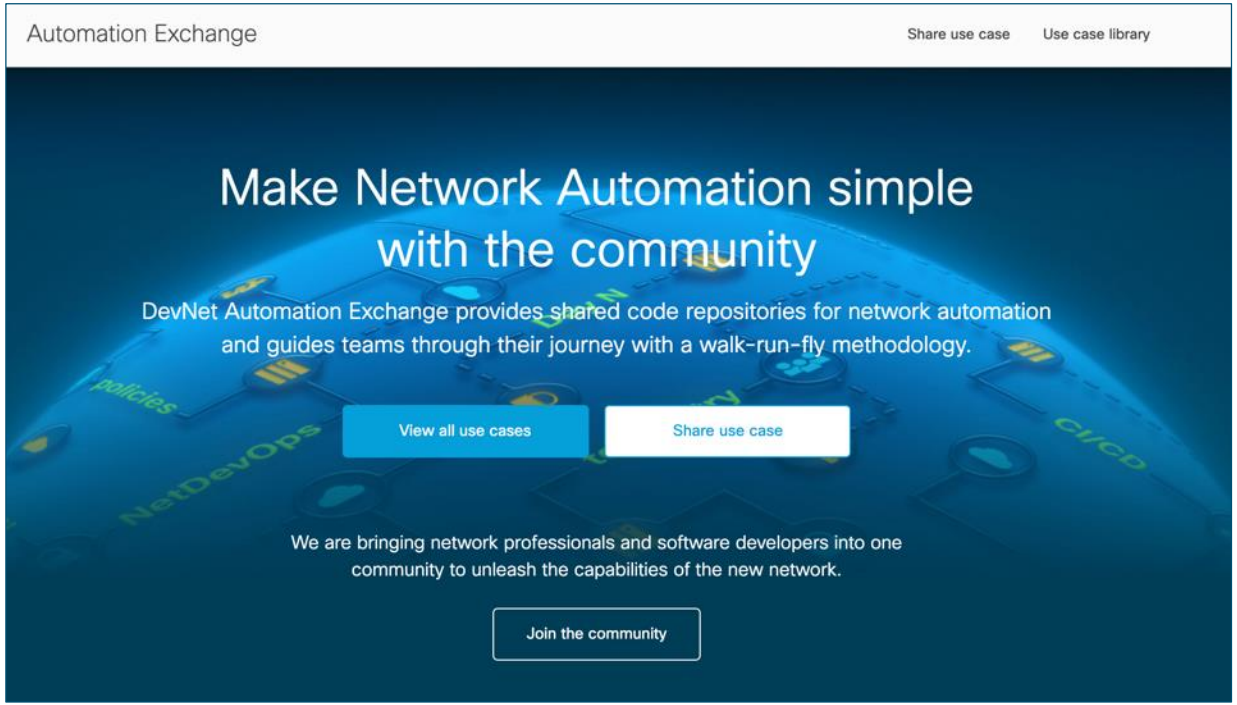
  DEVNET-4099: DevNet Certifications: Bringing software practices & software skills to networking
  Offered daily 12:15-12:45 in the DevNet Zone Theater

- Visit the **Learning@Cisco** booth

- Scan this code to **sign up** for the latest updates or go to http://cs.co/20eur02



Find shared code repositories of use cases for network automation & more!



Don't miss our 5 **Automate Infrastructure demos** in the DevNet Zone!

Scan this code or go to the URL to **learn more**

http://cs.co/20eur01

Complete your online session survey

- Please complete your session survey after each session. Your feedback is very important.
- Complete a minimum of 4 session surveys and the Overall Conference survey (starting on Thursday) to receive your Cisco Live t-shirt.
- All surveys can be taken in the Cisco Events Mobile App or by logging in to the Content Catalog on ciscolive.com/emea.

Cisco Live sessions will be available for viewing on demand after the event at ciscolive.com.

Continue your education

Demos in the Cisco Showcase

Walk-In Labs

Meet the Engineer 1:1 meetings

Related sessions

Thank you

CISCO

You make **possible**