



**BERUFLICHE SCHULE ITECH**

Elbinsel Wilhelmsburg

# Emerging Technologies Workshop

Network Programmability with Cisco APIC-EM  
(Instruktoren-Qualifizierung)

Ulrich Stritzel

Berufliche Schule ITECH Elbinsel Wilhelmsburg, Hamburg

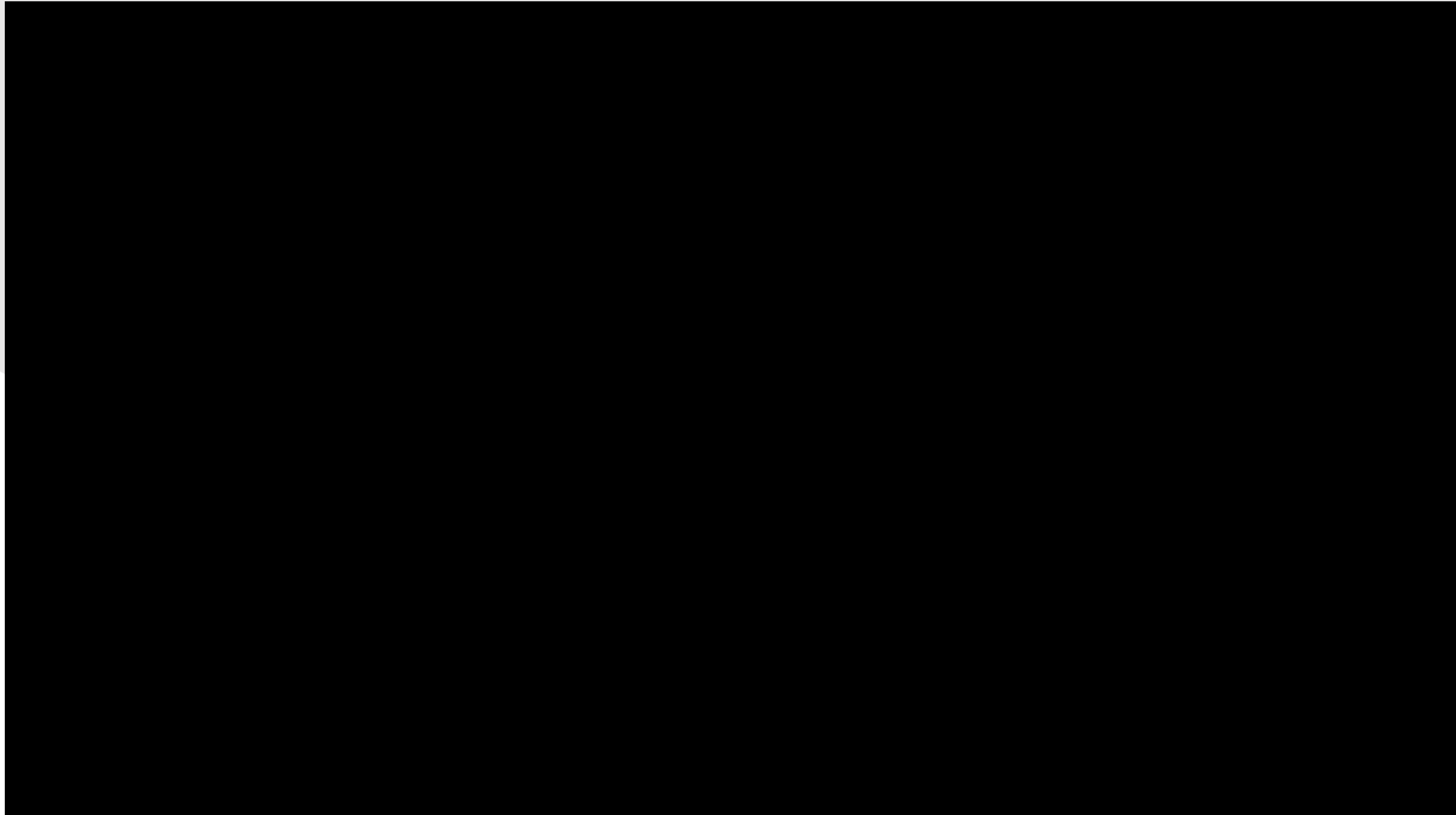
[ulrich.stritzel@itech-bs14.de](mailto:ulrich.stritzel@itech-bs14.de)

04.05.2019

## Agenda

- Chapter 1: Introduction to Basic Programming with Python
  - Motivation zu diesem Kurs
  - Getting (your equipment) ready
  - Begriffsklärung
  - Writing some code
- Chapter 2: Programming the APIC-EM
  - Begriffsklärung
  - Das APIC-EM REST Interface
  - Klicken und gucken
- Final Exam

# Chapter 1: Introduction to Basic Programming with Python





**BERUFLICHE SCHULE ITECH**

Elbinsel Wilhelmsburg

# **From Zero to Hero**

In 120 minutes

# Chapter 1: Introduction to Basic Programming with Python

## Getting (your equipment) ready

- Schreiben Sie sich in den ITC-Kurs per Seat Token ein.
- Starten Sie den Kurs
- Folgen Sie den Schritten in Kap. 1.0.1.2 (Lab - PC Setup for Workshop)
- Laden Sie sich die „Script and data files“ herunter!



**BERUFLICHE SCHULE ITECH**

Elbinsel Wilhelmsburg

**Demo...**

# Chapter 1: Introduction to Basic Programming with Python

## Begrifflichkeiten: CoPs = Community of Practice

- *GitHub* ist ein Onlinedienst, der Software-Entwicklungsprojekte auf seinen Servern bereitstellt.
- *Stack Overflow* ist eine Internetplattform, auf der angemeldete Benutzer Fragen zum Thema Softwareentwicklung diskutieren können.
- *Cisco DevNet* is Cisco's Entwickler Programm, vorrangig für Anwendungen und Integration für und on Cisco Produkte.



# Chapter 1: Introduction to Basic Programming with Python

## Begrifflichkeiten: Swagger

Swag = Jugendwort des Jahres 2011.  
Beschreibt in der Jugendsprache eine  
„beneidenswerte lässig-coole Ausstrahlung“.



[Bildquelle](#)



# Chapter 1: Introduction to Basic Programming with Python

## Begrifflichkeiten: Swagger

Swag = Jugendwort des Jahres 2011.  
Beschreibt in der Jugendsprache eine  
„beneidenswerte lässig-coole Ausstrahlung“.



[Bildquelle](#)

Komm erst garnicht auf [LACHSCHON.DE](http://LACHSCHON.DE)

# Chapter 1: Introduction to Basic Programming with Python

## Begrifflichkeiten: Swagger

Das Swagger Framework ist ein Open-Source-Software-Framework, um RESTful Webservices zu entwerfen, zu erstellen, zu dokumentieren und zu nutzen.

RESTful Webservices (REST = Representational State Transfer) meint dabei Webdienste, die APIs anbieten (API = Application Programming Interface).

→ **Fokus: Standardisierte Form einer API-Dokumentation**

# Chapter 1: Introduction to Basic Programming with Python

<https://www.programmableweb.com/apis/directory>

API Name	Description	Category	Submitted
<a href="#">Google Maps</a>	[This API is no longer available. Google Maps' services have been split into multiple APIs, including the <a href="#">Static Maps API</a> ,	<a href="#">Mapping</a>	12.05.2005
<a href="#">Twitter</a>	[This API is no longer available. It has been split into multiple APIs, including the <a href="#">Twitter Ads API</a> , <a href="#">Twitter Search Tweets...</a>	<a href="#">Social</a>	12.08.2006
<a href="#">YouTube</a>	The Data API allows users to integrate their program with YouTube and allow it to perform many of the operations available on the website. It provides the capability to search for videos, retrieve...	<a href="#">Video</a>	02.08.2006



**BERUFLICHE SCHULE ITECH**  
Elbinsel Wilhelmsburg

## Beispiel: International Space Station (ISS)

<http://api.open-notify.org>

<https://www.latlong.net>

<https://www.epochconverter.com>



**BERUFLICHE SCHULE ITECH**

Elbinsel Wilhelmsburg

**Let's write some code...**



## Chapter 1: Introduction to Basic Programming with Python

- Variablentypen (integer, float, string, boolean)
- Vergleichsoperatoren (>, <, ==, !=, >=, <=)
- Verknüpfen und ändern von Variablen und Variablentypen
- Listen und Verzeichnisse
- Eingaben
- If-Funktionen und Schleifen
- Zugriff auf externe Dateien
- Zugriff auf APIs

## Chapter 1: Introduction to Basic Programming with Python

- Variablentypen (integer, float, string, boolean)
- Vergleichsoperatoren (>, <, ==, !=, >=, <=)
- Verknüpfen und ändern von Variablen und Variablentypen
- **Listen und Verzeichnisse**
- **Eingaben**
- If-Funktionen und Schleifen
- Zugriff auf externe Dateien
- Zugriff auf APIs





**BERUFLICHE SCHULE ITECH**  
Elbinsel Wilhelmsburg

## **Chapter 2: Programming the APIC-EM**

## JSON vs. XML

- JSON = JavaScript Objekt Notation
- XML = Extensible Markup Language
- JSON ist kompakter als XML (Name-Werte-Paare statt Open-Close-Tags)
- JSON produziert kleinere Datenmengen
- JSON ist ähnlich wie JavaScript (bessere Lesbarkeit)

→ **JSON ist besser für API-Daten geeignet!**

## JSON data funktioniert wie Python-Listen und -Dictionarys

Die JSON-Daten rechts sind in Python in der Variable `json_data` gespeichert worden. Wie lautet die Anfrage, die den Namen der dritten Person in der Liste in der Variable `iss_astronaut` speichern soll?

?

```
{  
  message: "success",  
  number: 6,  
  - people: [  
    - {  
      craft: "ISS",  
      name: "Oleg Kononenko"  
    },  
    - {  
      craft: "ISS",  
      name: "David Saint-Jacques"  
    },  
    - {  
      craft: "ISS",  
      name: "Anne McClain"  
    },  
    - {  
      craft: "ISS",  
      name: "Alexey Ovchinin"  
    },  
    - {  
      craft: "ISS",  
      name: "Nick Hague"  
    },  
    - {  
      craft: "ISS",  
      name: "Christina Koch"  
    }  
  ]  
}
```

## JSON data funktioniert wie Python-Listen und -Dictionarys

Die JSON-Daten rechts sind in Python in der Variable `json_data` gespeichert worden. Wie lautet die Anfrage, die den Namen der dritten Person in der Liste in der Variable `iss_astronaut` speichern soll?

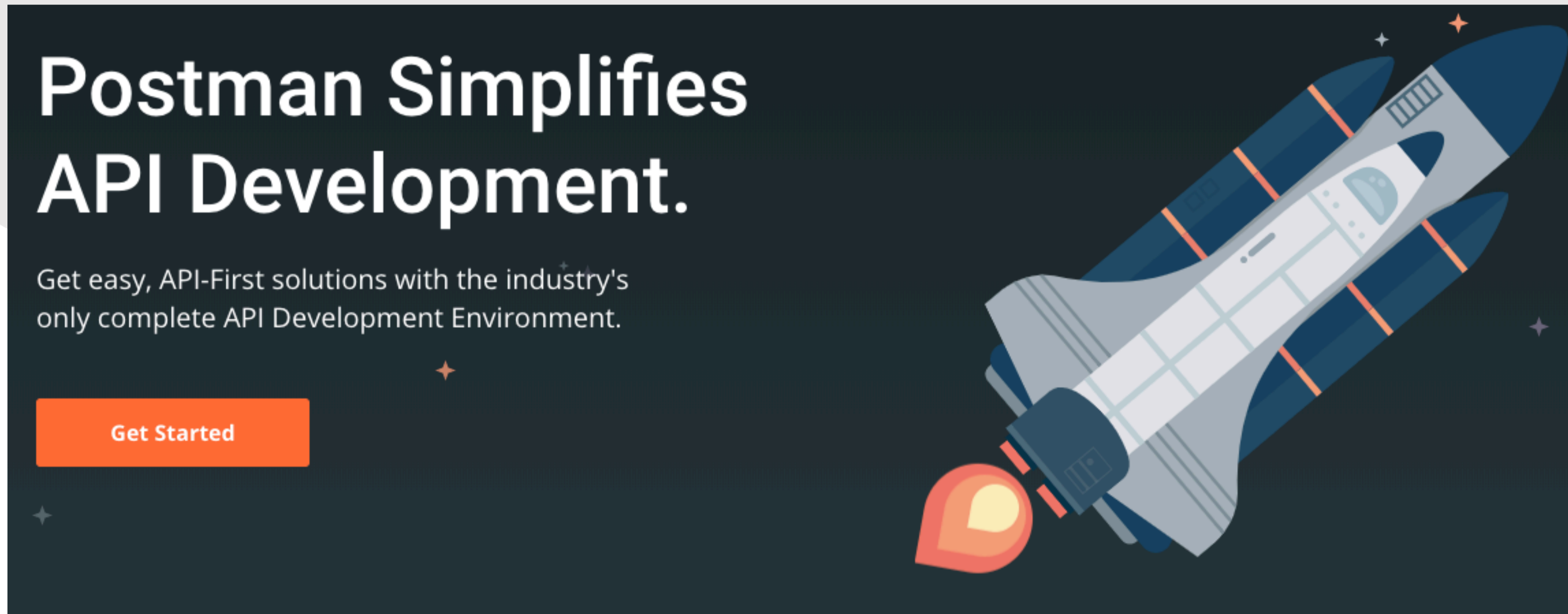
`iss_astronaut=`

`json_data[,'people'][2][name]`

```
{
  message: "success",
  number: 6,
  - people: [
    - {
      craft: "ISS",
      name: "Oleg Kononenko"
    },
    - {
      craft: "ISS",
      name: "David Saint-Jacques"
    },
    - {
      craft: "ISS",
      name: "Anne McClain"
    },
    - {
      craft: "ISS",
      name: "Alexey Ovchinin"
    },
    - {
      craft: "ISS",
      name: "Nick Hague"
    },
    - {
      craft: "ISS",
      name: "Christina Koch"
    }
  ]
}
```

## Postman

... ist eine API-Entwicklungsumgebung um z.B. API-Requests durchzuführen und REST-APIs auf Funktion zu prüfen.



**Postman Simplifies  
API Development.**

Get easy, API-First solutions with the industry's  
only complete API Development Environment.

[Get Started](#)

The graphic features a stylized rocket launch against a dark blue space background with stars. The rocket is white and blue, angled upwards from the bottom left towards the top right. A bright orange and yellow flame is at the base of the rocket. The text is in white, and a prominent orange button with white text is located in the lower-left quadrant.

## Chapter 2: Programming the APIC-EM

**Cisco APIC-EM?**



## Chapter 2: Programming the APIC-EM

**Cisco APIC-EM =**

**Cisco Application Policy Infrastructure Controller - Enterprise Modul**



## Chapter 2: Programming the APIC-EM

*Cisco Application Policy Infrastructure Controller Enterprise-Modul –  
Netzwerkkonfiguration und Bereitstellung vereinfachen*

Der Cisco Application Policy Infrastructure Controller (APIC) bildet die Basis für die anwendungszentrische Infrastruktur (Application Centric Infrastructure, ACI) der Cisco ONE-Plattform. Das Cisco APIC Enterprise-Modul dient dabei zur Erweiterung des Cisco APIC vom Rechenzentrum auf WAN- und Access-Domänen. Dadurch kann der Cisco APIC das Netzwerk im Rahmen der Cisco ONE-Plattform auch für WAN und Zugriff abstrahieren und automatisieren. Das Cisco APIC Enterprise-Modul bringt SDN (Software-defined Networking) in die ACI und bildet die Steuerungsebene der Cisco ONE Enterprise Networks Architecture. Der Controller ist umfassend programmierbar. So kann die IT-Abteilung die Konfiguration des Netzwerks automatisieren und schnell auf neue Geschäftsmöglichkeiten reagieren.

## Chapter 2: Programming the APIC-EM

*Cisco Application Policy Infrastructure Controller Enterprise-Modul –  
Netzwerkkonfiguration und Bereitstellung vereinfachen*

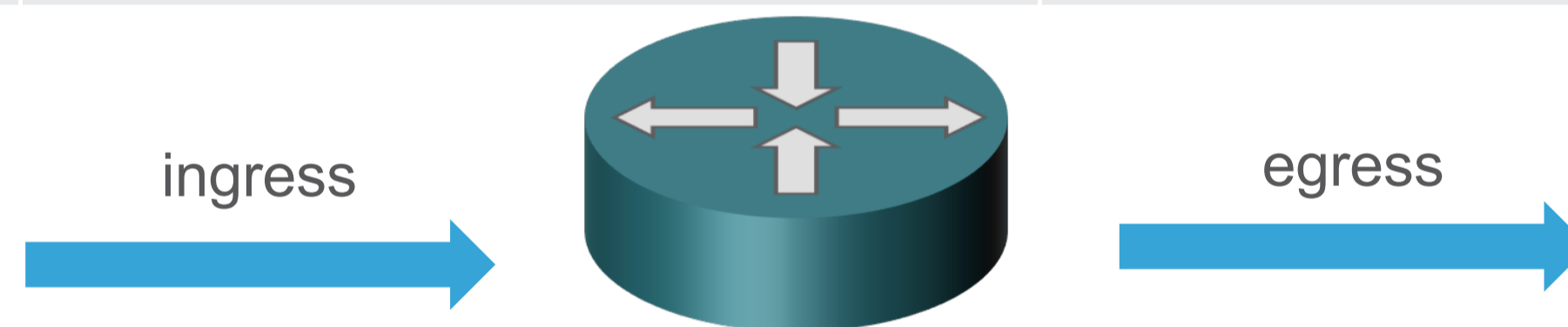
Der Cisco Application Policy Infrastructure Controller (APIC) bildet die Basis für die anwendungszentrische Infrastruktur (Application Centric Infrastructure, ACI) der Cisco ONE-Plattform. Das Cisco APIC Enterprise-Modul dient dabei zur Erweiterung des Cisco APIC vom Rechenzentrum auf WAN- und Access-Domänen. Dadurch kann der Cisco APIC das Netzwerk im Rahmen der Cisco ONE-Plattform auch für WAN und Zugriff abstrahieren und automatisieren. Das Cisco APIC Enterprise-Modul bringt **SDN (Software-defined Networking)** in die ACI und bildet die Steuerungsebene der Cisco ONE Enterprise Networks Architecture. Der Controller ist umfassend programmierbar. So kann die IT-Abteilung die Konfiguration des Netzwerks automatisieren und schnell auf neue Geschäftsmöglichkeiten reagieren.

# Network programmability concepts

## SDN: Control Plane and Data Plane

### Control Plane

Hardware	Purpose	Example Processes
Device CPU	makes decisions about where traffic is sent	routing protocols, spanning tree, AAA, SNMP, CLI

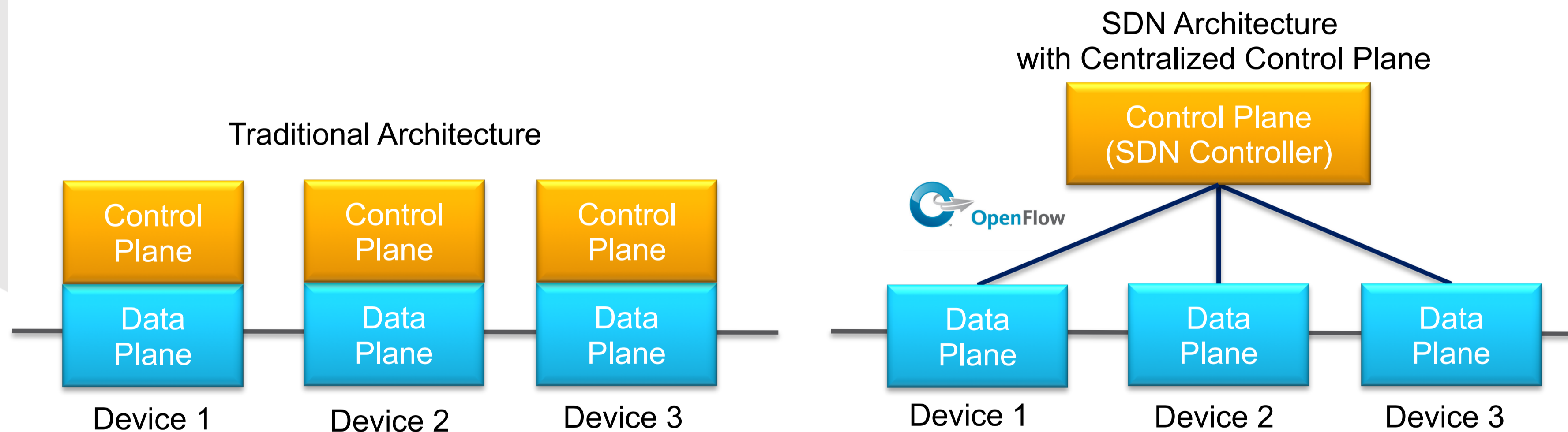


### Data Plane

Hardware	Purpose	Example Processes
Dedicated ASICs	forwards traffic to the selected destination	packet switching, L2 switching, QoS, policies, ACLs

# Network programmability concepts

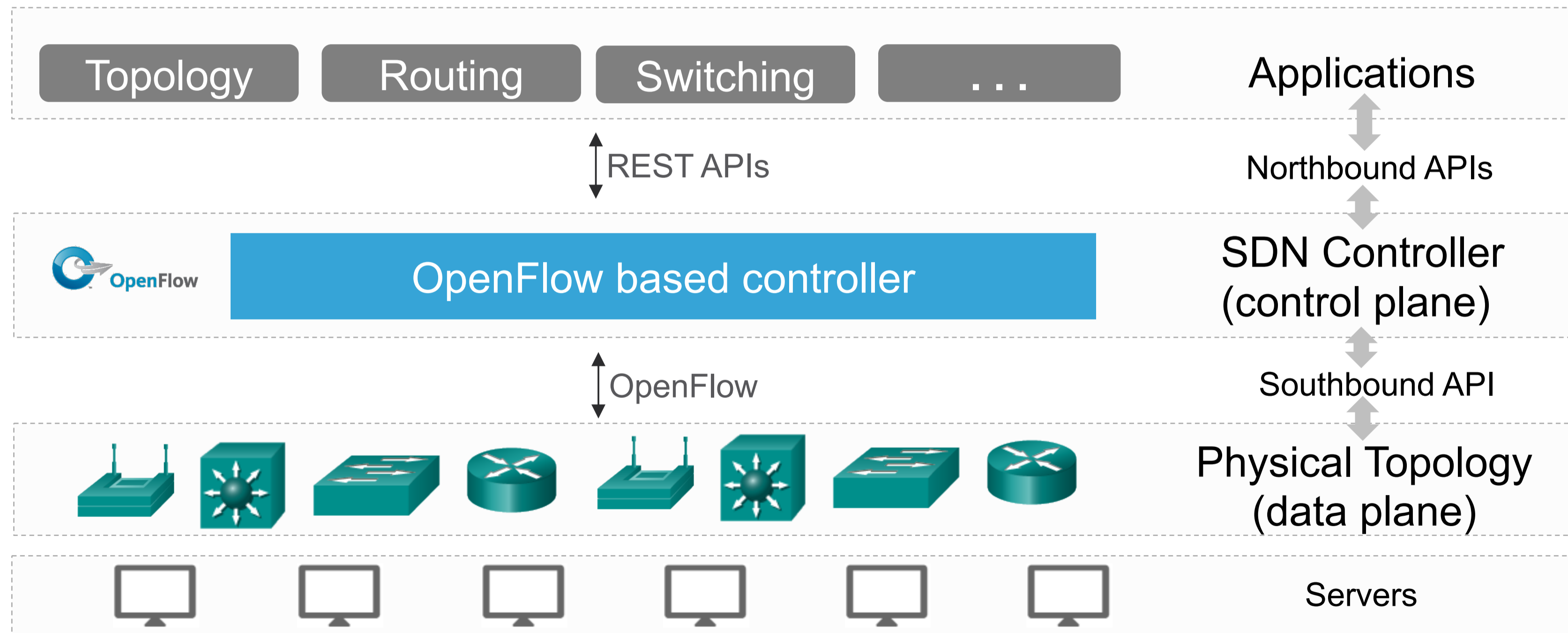
## Traditional and SDN Architectures



OpenFlow is a protocol between SDN controllers and network devices, as well as a specification of the logical structure of the network switch functions.

# Network programmability concepts

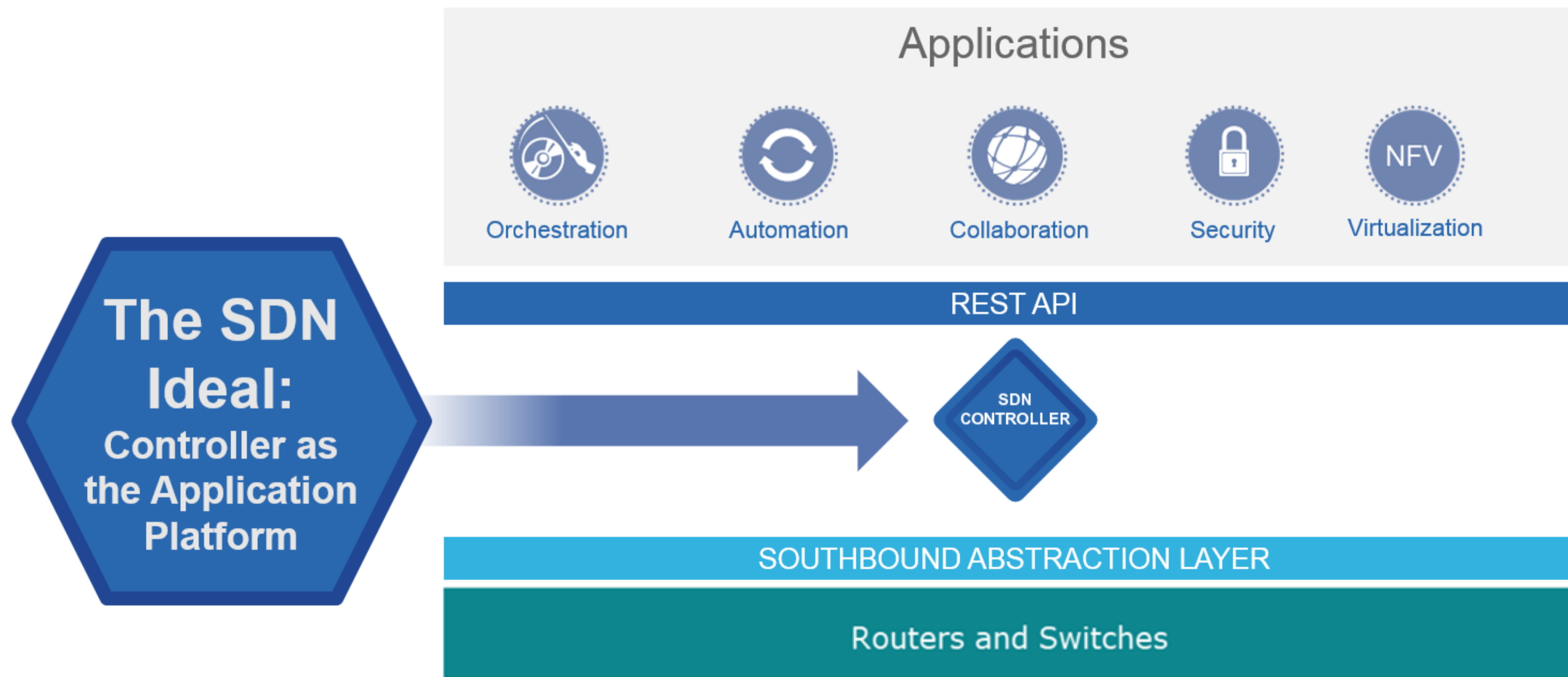
## OpenFlow SDN Model





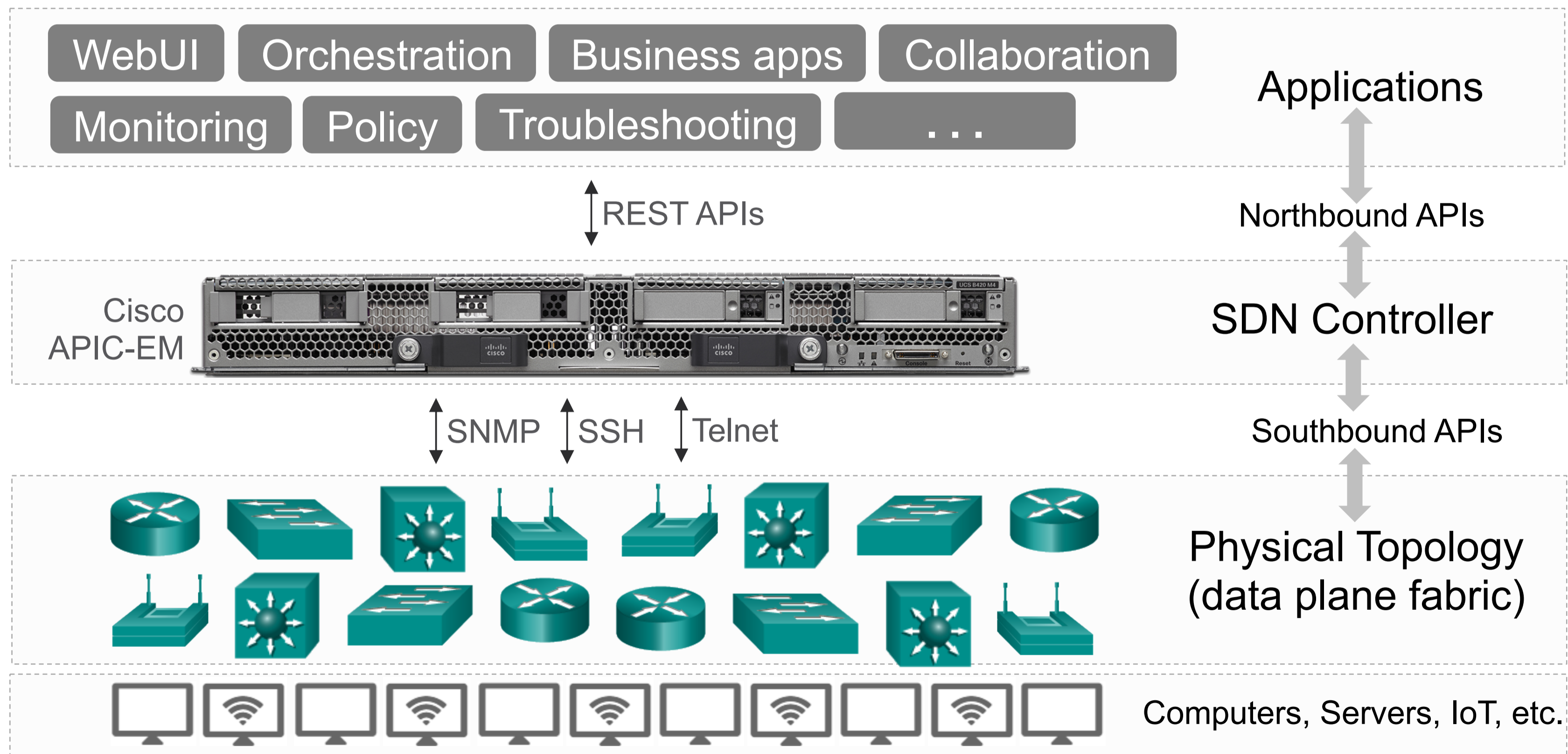
# Network programmability concepts

## Network-Wide Abstractions Simplify the Network



# Network programmability concepts

## Cisco SDN Model with APIC-EM





## Chapter 2: Programming the APIC-EM

### Interacting mit APIC-EM – Wie geht das?

- **API** (Application Programming Interface/Anwendungsprogrammierschnittstelle)
- **Requests** (sending request, requesting token (bei APIC-EM: „service ticket“), building request)
  - In Python: modul „requests“ muss importiert werden
- **Erfolgreicher** request! („HTTP-Status-Code“)

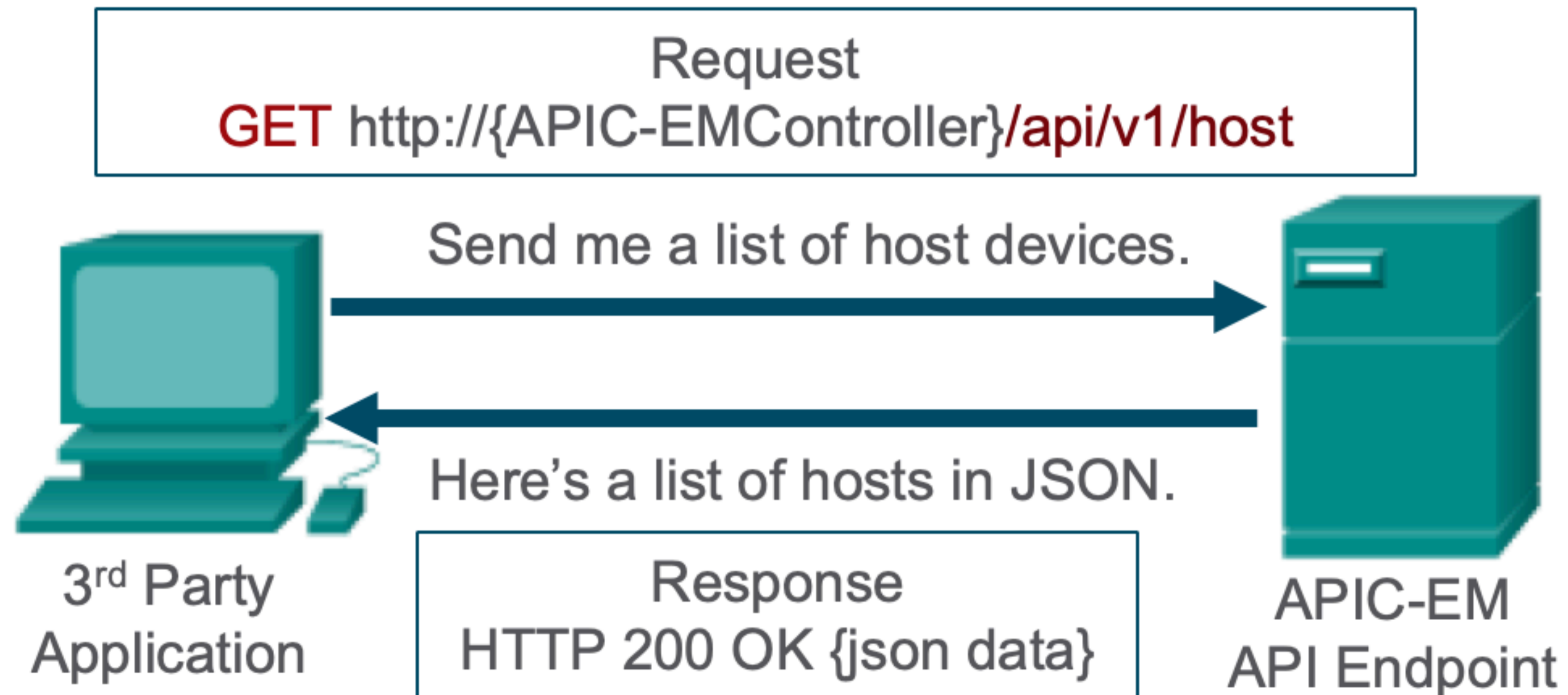
## Chapter 2: Programming the APIC-EM

### Die wichtigsten HTTP-Status-Codes (vgl. <https://www.restapitutorial.com/httpstatuscodes.html>)

<b>2xx Success</b>	<b>4xx Client Error</b>
200 OK	400 Bad Request
201 Created	401 Unauthorized
204 No Content	403 Forbidden
	404 Not Found
<b>3xx Redirection</b>	409 Conflict
304 Not Modified	
	<b>5xx Server Error</b>
	500 Internal Server Error

## Chapter 2: Programming the APIC-EM

### Wie funktioniert das?



## Chapter 2: Programming the APIC-EM

# Anatomy of a REST Request

REST requests require the following elements (requirements may differ depending on the API):

### Method

- GET (retrieve), POST (create), PUT (update), DELETE (remove)

### URL

- Example: `http://{APIC-EMController}/api/v1/host`

### Authentication

- Basic HTTP, OAuth, none, Custom

### Custom Headers

- HTTP Headers
- Example: `Content-Type: application/json`

### Request Body

- JSON or XML containing data needed to complete request

## Chapter 2: Programming the APIC-EM

### What is in the Response?

#### HTTP Status Codes

- <http://www.w3.org/Protocols/HTTP/HTRESP.html>
- 200 OK
- 201 Created
- 401, 403 Authorization error
- 404 Resource not found
- 500 Internal Error

#### Headers

#### Body

- JSON
- XML



```
1 {
2   "version": "0.0",
3   "response": "349117ce-3c7f-4e14-bc6f-83071e990198: Acl Policy
Appended Successfully on the Device : 9cb0df12-b9f7-4551-932e-
3391974da58f"
4 }
```

Example output of a HTTP response in the Postman application

# Network programmability concepts

## APIC-EM – Log in

Virtualized APIC-EM Controllers are available in several DevNet Sandboxes:

Always On, NetAcad instances

- For NetAcad users only
- <https://DevNetSBX-NetAcad-APICEM-1.cisco.com>
- <https://DevNetSBX-NetAcad-APICEM-2.cisco.com>
- <https://DevNetSBX-NetAcad-APICEM-3.cisco.com>
- **User/PW:** Ask your instructor

Always on, public instance

- For to all DevNet users
- <https://SandBoxAPICEM.cisco.com>
- **User:** *devnetuser* **PW:** *Cisco123!*





**BERUFLICHE SCHULE ITECH**  
Elbinsel Wilhelmsburg

**Demo...**



## Und jetzt ans Eingemachte...

- 1.0.1.2: Lab - PC Setup for Workshop
- 1.2.1.5: Activity - Write, Save, and Run Your First Program
- 1.2.3.3: Activity - Troubleshoot List and Dictionary Code
- 1.2.6.4: Activity – Create a Script to Allow User to Add Devices
- 1.3.3: Parsing JSON with Python (alle Activities)
- 2.1.2.4: Activity - APIC-EM Topology Page
- 2.2.1.6: Activity - APIC-EM Swagger Documentation
- 2.3: Putting It All Together



**BERUFLICHE SCHULE ITECH**

Elbinsel Wilhelmsburg

**Vielen Dank  
und  
weiterhin einen spannenden Akademietag!**