Networking Academy

it bildungsnetz

Akademie für Lehrerfortbildung und Personalführung

Meraki: Der Zauber der Adhoc-Methodik mit API-Administration

Kilian Hümpfer, Cisco Systems

20. Nationaler Akademietag der Bildungsinitiative Networking
22./23. April 2021

Agenda

- Lehrer an die Macht?
- Einführung Meraki
- Einführung APIs
- Workshop: Postman API‘s
- Live Demo: Lehrer-App
- Nächste Schritte

Lehrer an die Macht?

Lehrer an die Macht?

KLUG MACHT DUMM!
aber ohne Sinn und Verstand
Wir streiken für
bessere Arbeitsbedingungen
bessere Bildung und gute Schulen
artüffeln,

Quelle: SHZ.de

Quelle:
econocom.de

Einführung Meraki

Die Meraki Platform

Eine Oberfläche für alle Systeme

MR
Wireless Access Points

MS
Ethernet Switches

MX
Security & SD-WAN Appliances

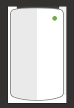
MG
Cellular Gateways

MI
Insight [Application & WAN]

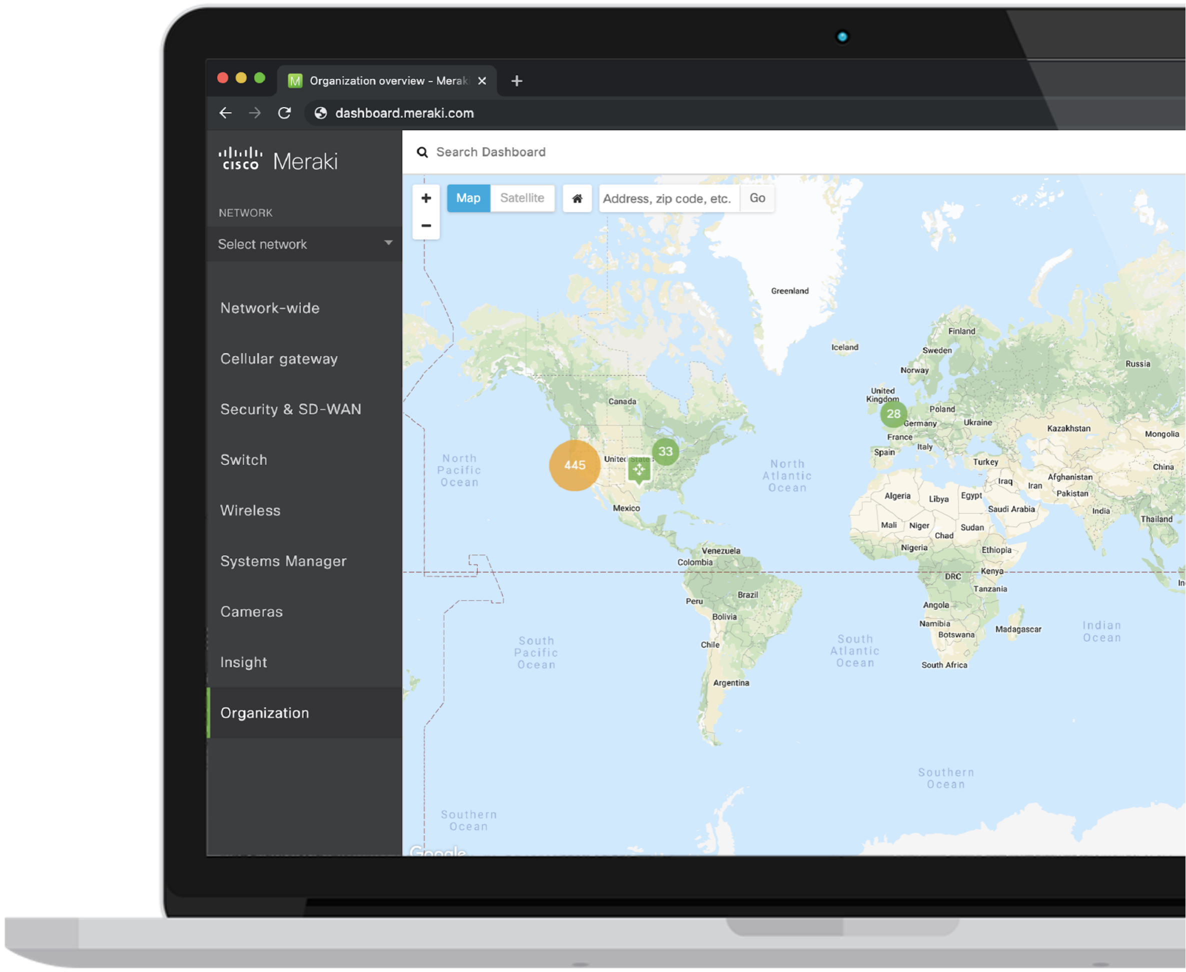
SM

SM
Endpoint Management

MV
Smart Cameras

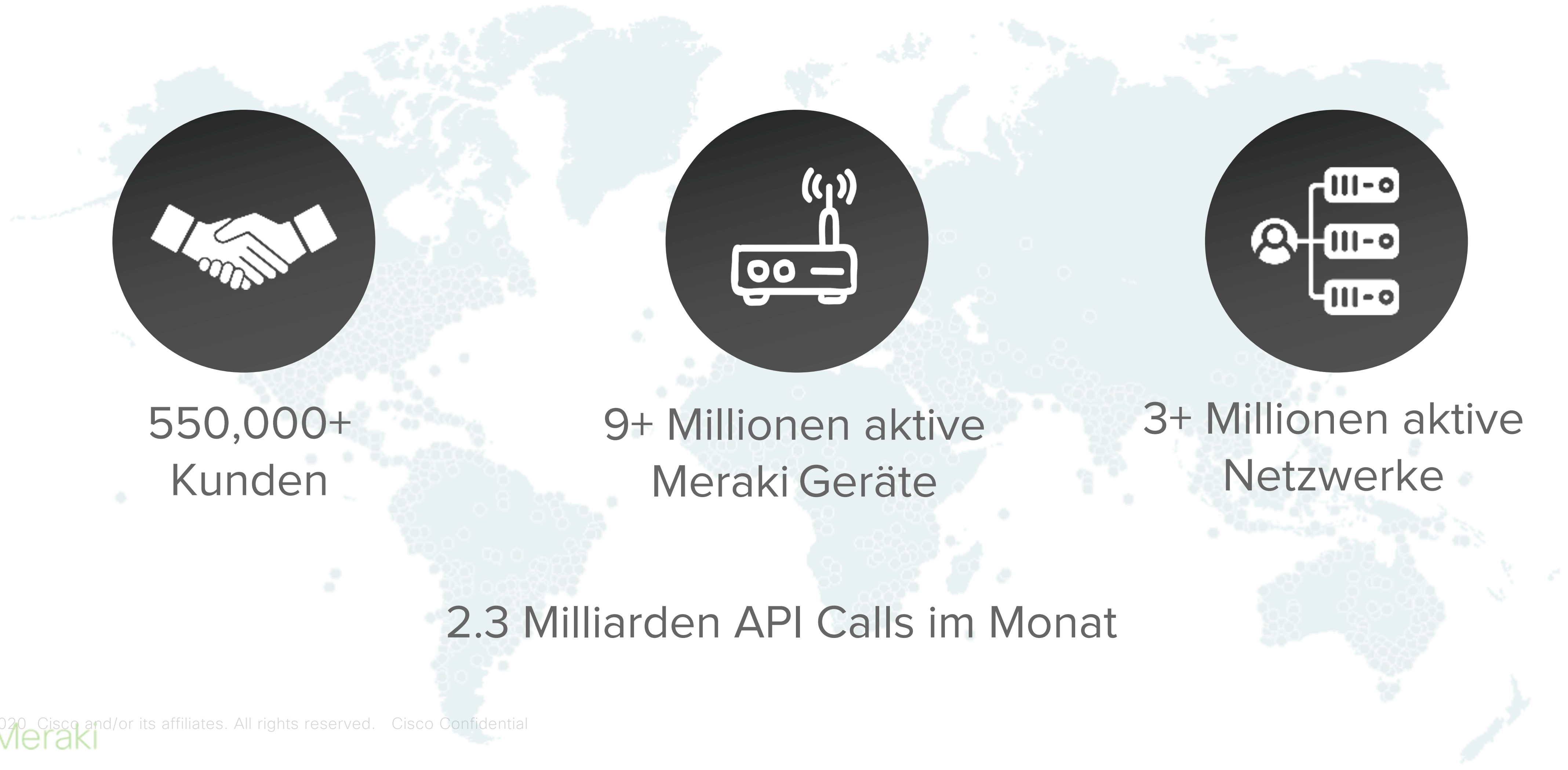


MT
Smart Sensors



Meraki Cloud

Der weltweit größte Anbieter von cloudverwalteten Netzwerken



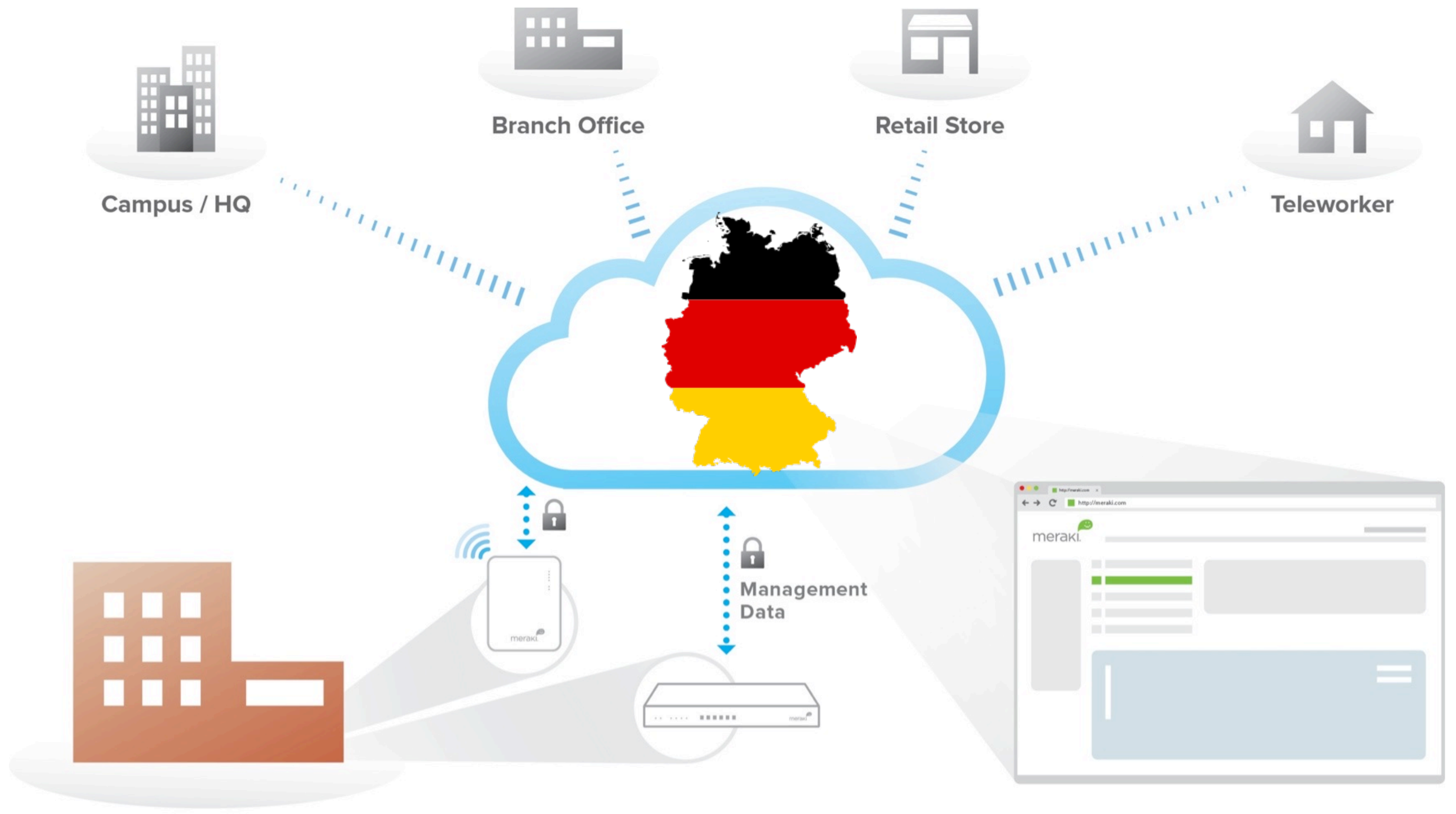
550,000+
Kunden

9+ Millionen aktive
Meraki Geräte

3+ Millionen aktive
Netzwerke

2.3 Milliarden API Calls im Monat

Cloud-verwaltete Netzwerkarchitektur



Sicheres Out—of-Band Management: Kein Benutzerdatenverkehr durchläuft die Cloud

Cloud-gehostete zentralisierte Verwaltungsplattform

Intuitives **browserbasiertes** Dashboard

Demo Time!!!

Lehrer an die Macht? Dilemma!

KLUG MACHT DUMM!
aber ohne Sinn und Verstand
Wir streiken für
bessere Arbeitsbedingungen
bessere Bildung und gute Schulen
artüffeln,

Quelle: SHZ.de

Einführung APIs

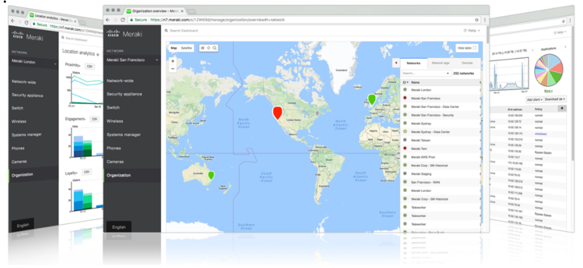
Integration into existing environments

Cisco
ISE & Prime

RADIUS &
SNMP

DNA Spaces

Location Scanning API



Provisioning system

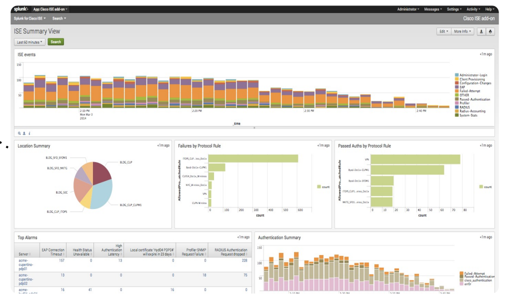
Dashboard
API

Webhooks

servicenow

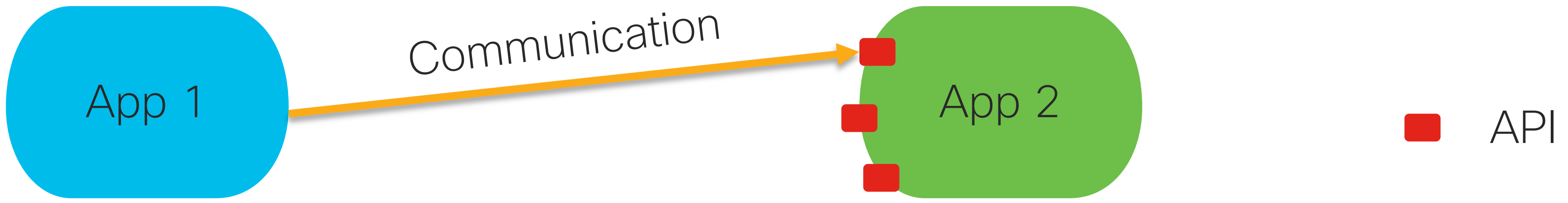
Alarm & Ticketing

Syslog/Netflow



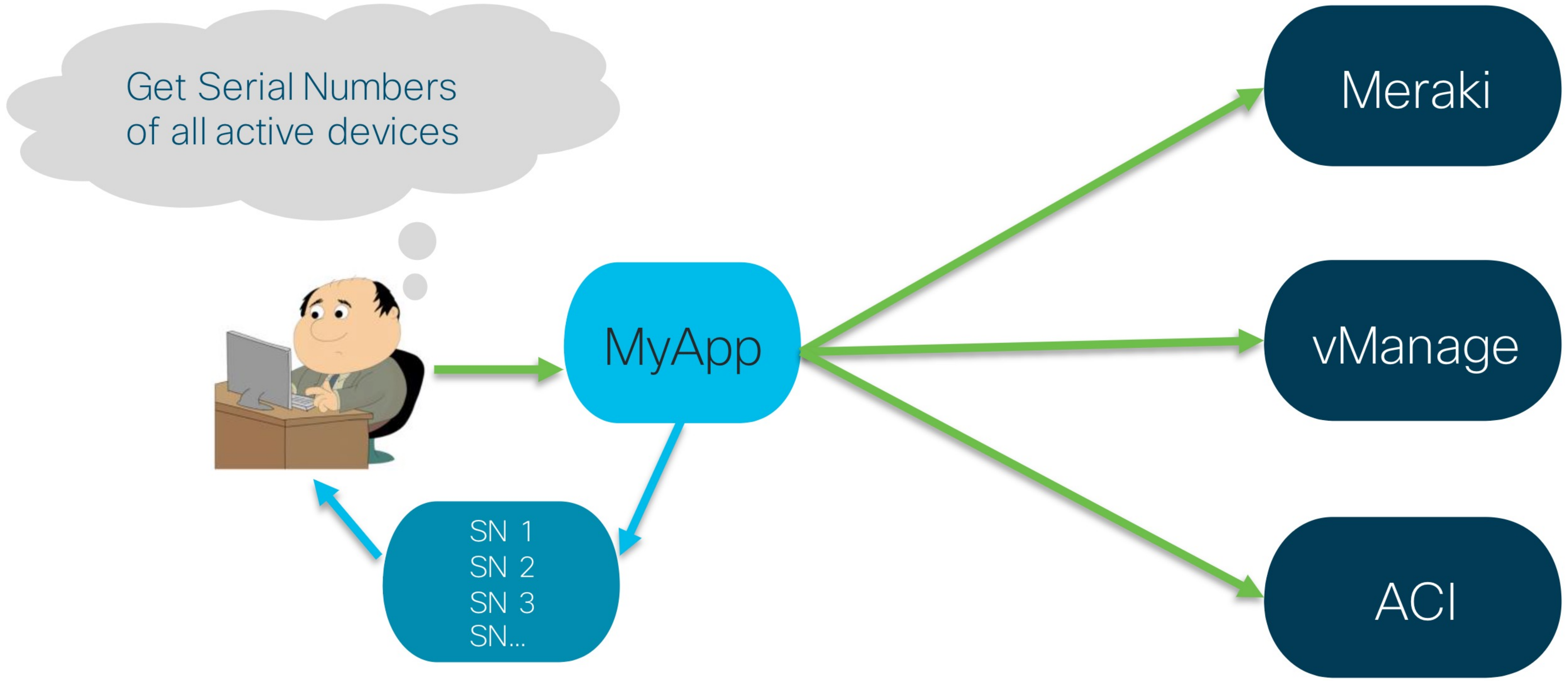
Logging platform

API: application programming interface



API is a way for two applications to talk to each other
GUI is an interface for Humans, API is the interface for Machines

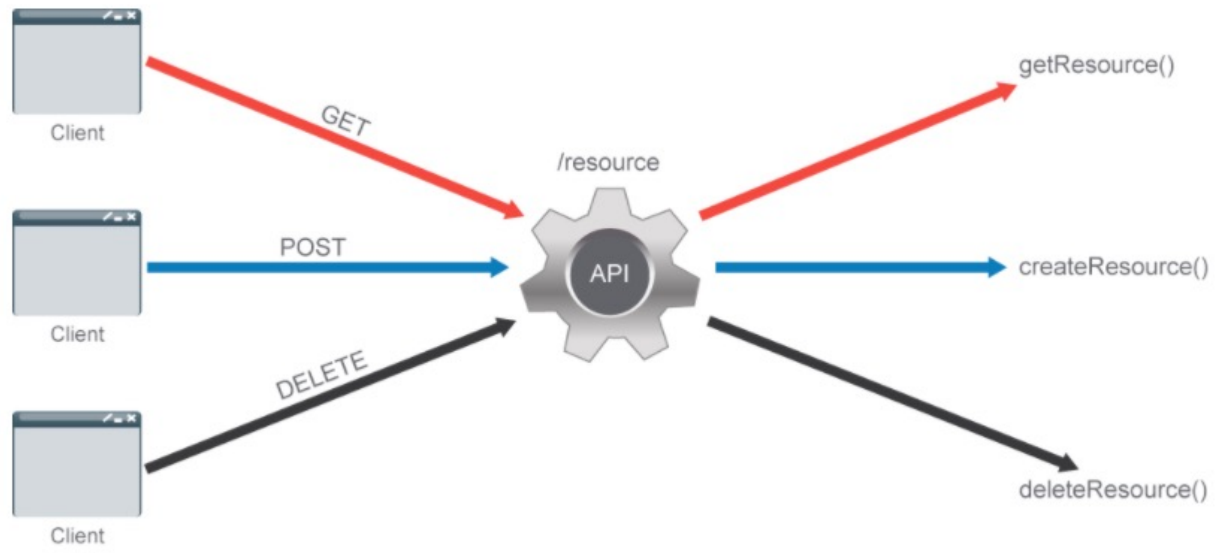
Example: Serial Numbers application



What is REST?

- Representational State Transfer
- REST APIs most commonly leverage HTTP(s)

HTTP and REST



Follows 6 principles:

- Client server architecture
- Stateless
- Cacheable
- Layered system
- Uniform interface
- Code on demand

HTTP Overview

- Based on Client-Server Model (Request – Response)
- Stateless protocol by default

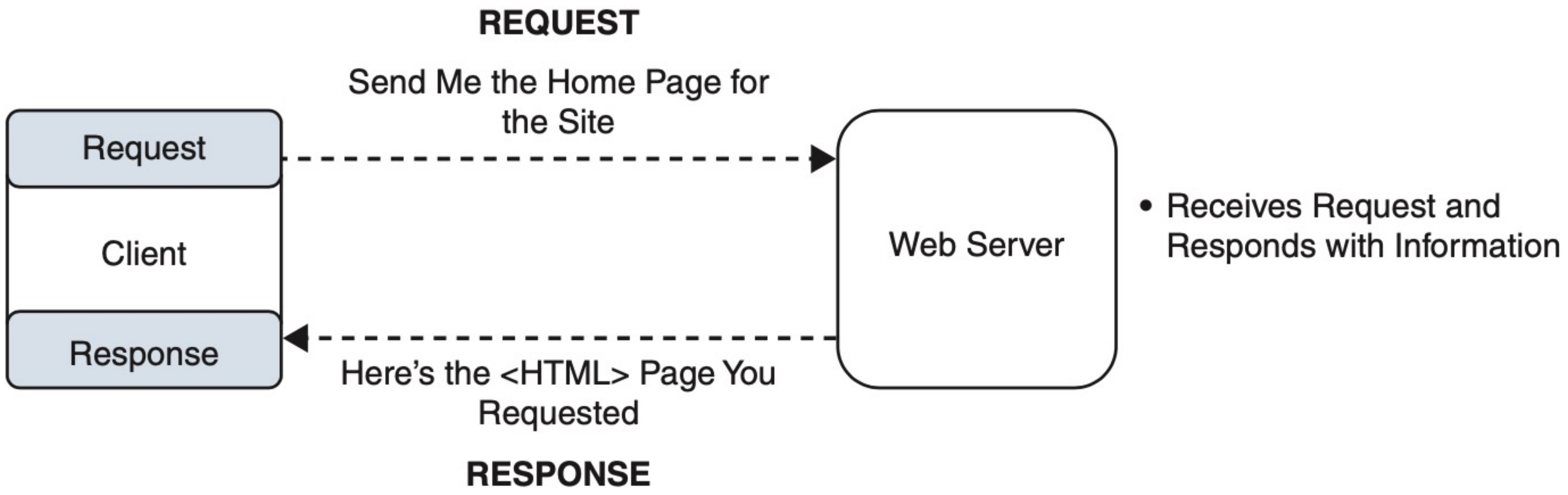
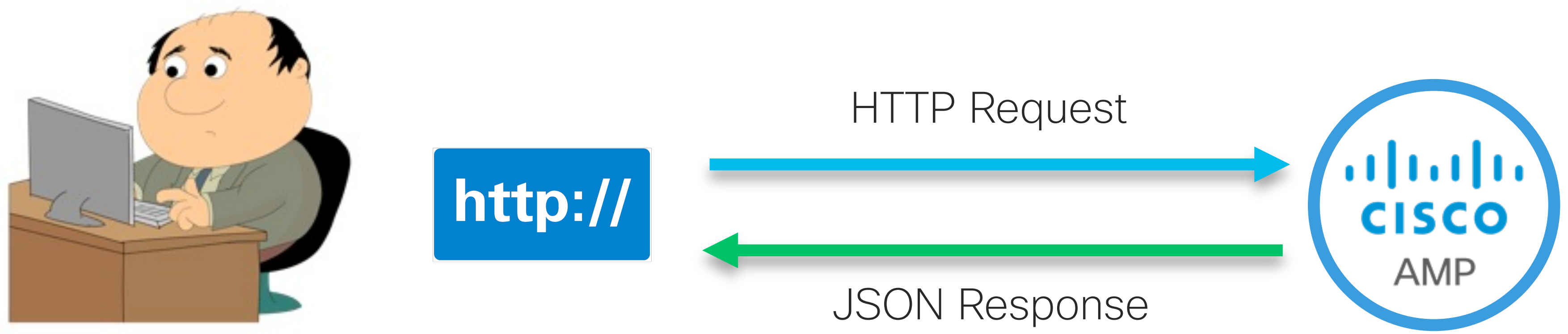

REQUEST
Send Me the Home Page for the Site
Request
Client
Response
Web Server
• Receives Request and Responds with Information
Here's the <HTML> Page You Requested
RESPONSE

Figure 7-5 *Simple HTTP Request/Response Cycle*

REST APIs

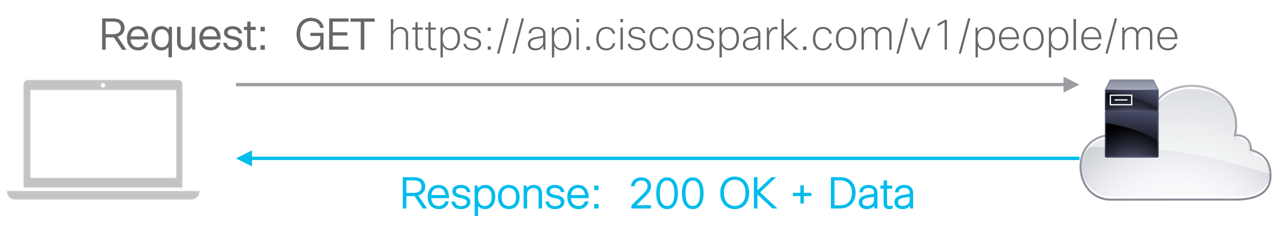
- Mechanism by which clients can communicate with the REST Server
- HTTP Calls (GET, PATCH, DELETE, POST, PUT)



REST APIs Authentication

- API key authentication: client adds a pregenerated key to HTTP headers (Base 64 Encoded). Keys are generated on the server on demand (normally limited per user or per service). Keys can also be added in the headers as a cookie or as a URL parameter
- Token authentication: uses a dynamically generated token. The authentication server validates the credentials. If they are valid, a custom, time-limited, and signed authentication token is issued and returned to the user. The API service validates the token and serves the required data.
- HTTP authentication: uses built-in HTTP authentication

Request/Response



HTTP Request

```
GET /v1/people/me HTTP/1.1
```

Request Headers

```
Host: api.ciscospark.com
Authorization: Bearer <redacted>
Accept: */*
Accept-Encoding: gzip, deflate, sdch
Connection: keep-alive
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_4) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/49.0.2623.112 Safari/537.36
```

HTTP Response

```
HTTP/1.1 200 OK
```

Response Headers

```
Date: Fri, 08 Apr 2016 16:59:20 GMT
Content-Type: application/json;charset=UTF-8
Content-Encoding: gzip
Content-Length: 323
Trackingid: NA_514181f9-7885-4716-bbfb-fe9a54f2248a
Vary: Accept-Encoding
X-Cf-Requestid: 8634487a-8c9e-417e-60bf-06ead6ffe950
```

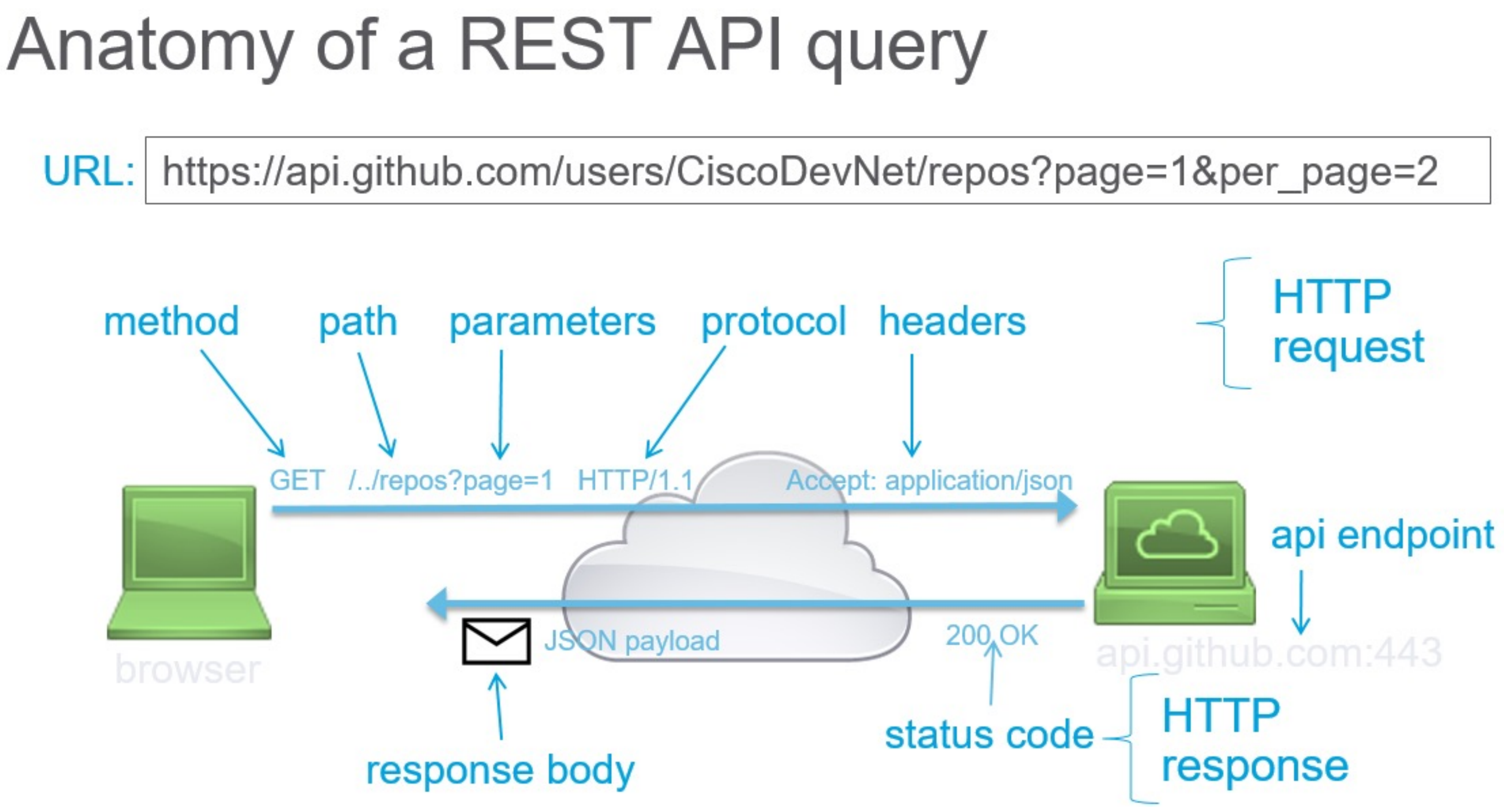
<blank line>

Response Payload

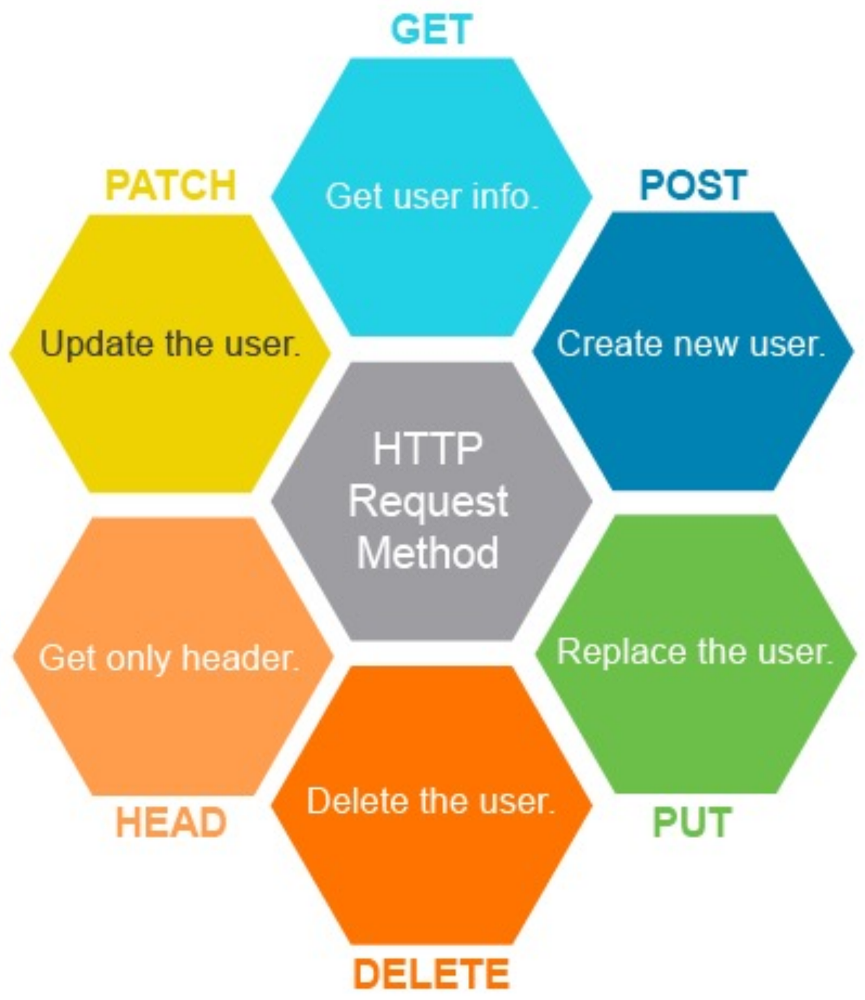
```
{
   "id":
   "Y2lzY29zcGFyazovL3VzL1BFT1BMRS9mZjhlZTZmYi1hZmVmLTRhNGQtOTJiMS1kNmIyMTZiNTg5NDk
   ",
   "emails": [    "chrlunsf@cisco.com"  ],
   "displayName": "Chris Lunsford (chrlunsf)",
   "avatar": "https://1efa7a94ed216783e352-
   c62266528714497a17239ececf39e9e2.ssl.cf1.rackcdn.com/V1~ba1ecf557a7e0b7cc3081998
   df965aad~cNFKqEjAQ5aQkyt_l1zsCQ==~1600",
   "created": "2012-06-15T20:36:48.914Z"
}
```

Note: This is all exchanged as simple text over a TCP/TLS connection.

How does it work?



HTTP Methods



4) **Status Codes:**
== (in response) what was the status of the request?

Status Code	Status Message	Meaning
200	OK	All looks good
201	Created	New resource created
400	Bad Request	Request was invalid
401	Unauthorized	Authentication missing or incorrect
403	Forbidden	Request was understood, but not allowed
404	Not Found	Resource not found
500	Internal Server Error	Something wrong with the server
503	Service Unavailable	Server is unable to complete request

- 2xx: ALL OK
- 4xx: Client-side error
- 5xx: Server-side error

More extensive List - https://developer.mozilla.org/en-US/docs/Web/HTTP/Sta

APIs sind überall

API Demo – Chrome Developer Tools

API Demo – Meraki Mobile App

Workshop

Herausforderung

Lehrer soll WLAN selber an- bzw. ausschalten können.

Zusätzlich soll er Youtube blockieren können.

Lehrer soll keinen Admin-Zugriff bekommen.

Many Options for Working with REST APIs

curl

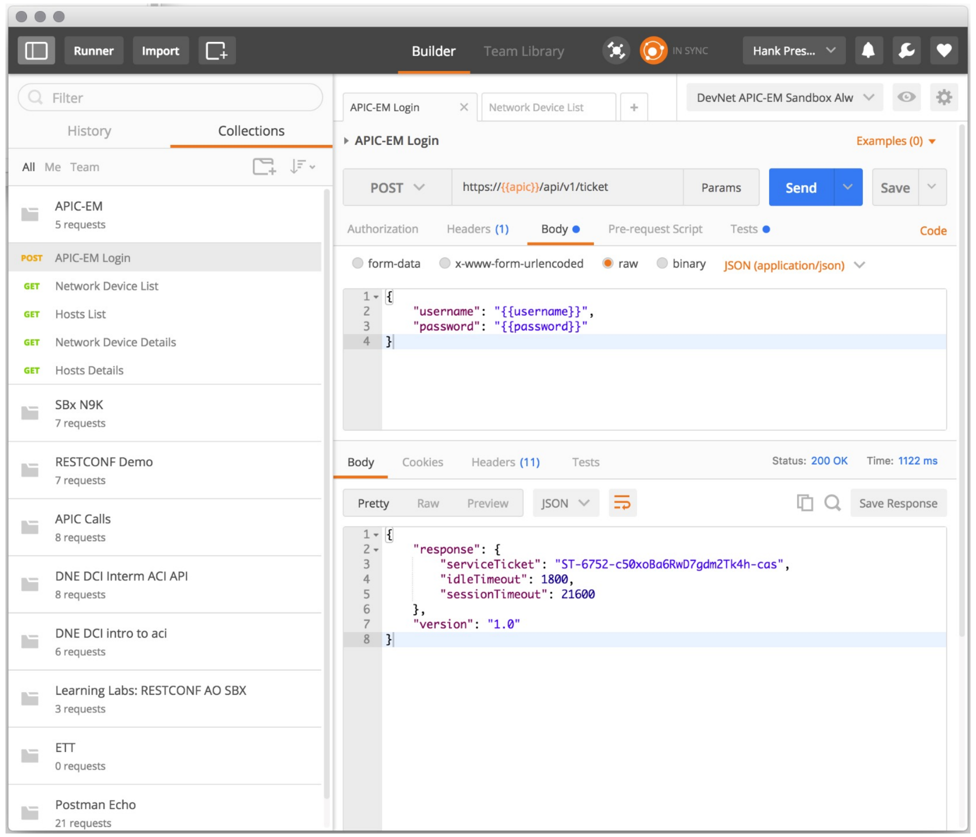
- Linux command line application

Postman

- API testing desktop App and framework

Programming Language Library

- Like *Python Requests*
- *programmatically* executing API calls



APIC-EM Login

POST https://{{apic}}/api/v1/ticket

Body: ○ form-data ○ x-www-form-urlencoded ● raw ○ binary JSON (application/json)

```
{
    "username": "{{username}}",
    "password": "{{password}}"
}
```

Status: 200 OK Time: 1122 ms

Body — Pretty — JSON

```
{
    "response": {
        "serviceTicket": "ST-6752-c50xoBa6RwD7gdm2Tk4h-cas",
        "idleTimeout": 1800,
        "sessionTimeout": 21600
    },
    "version": "1.0"
}
```

Hands On!

What you learned in this module…

- An API is an Application Programming Interface.
- REST stands for Representational State Transfer and uses HTTP or HTTPS to send requests and receive responses.
- REST APIs use standard verbs like GET, PUT, POST, and DELETE that may correspond to Create, Read, Update, Delete.
- Tools like curl, Postman, and Python requests are commonly used for REST API calls.
- Postman is a great utility for testing and trying out REST APIs.
- You can use collections and environments for testing REST APIs with Postman.

Nächste Schritte
→ Kasimir

cisco

The bridge to possible