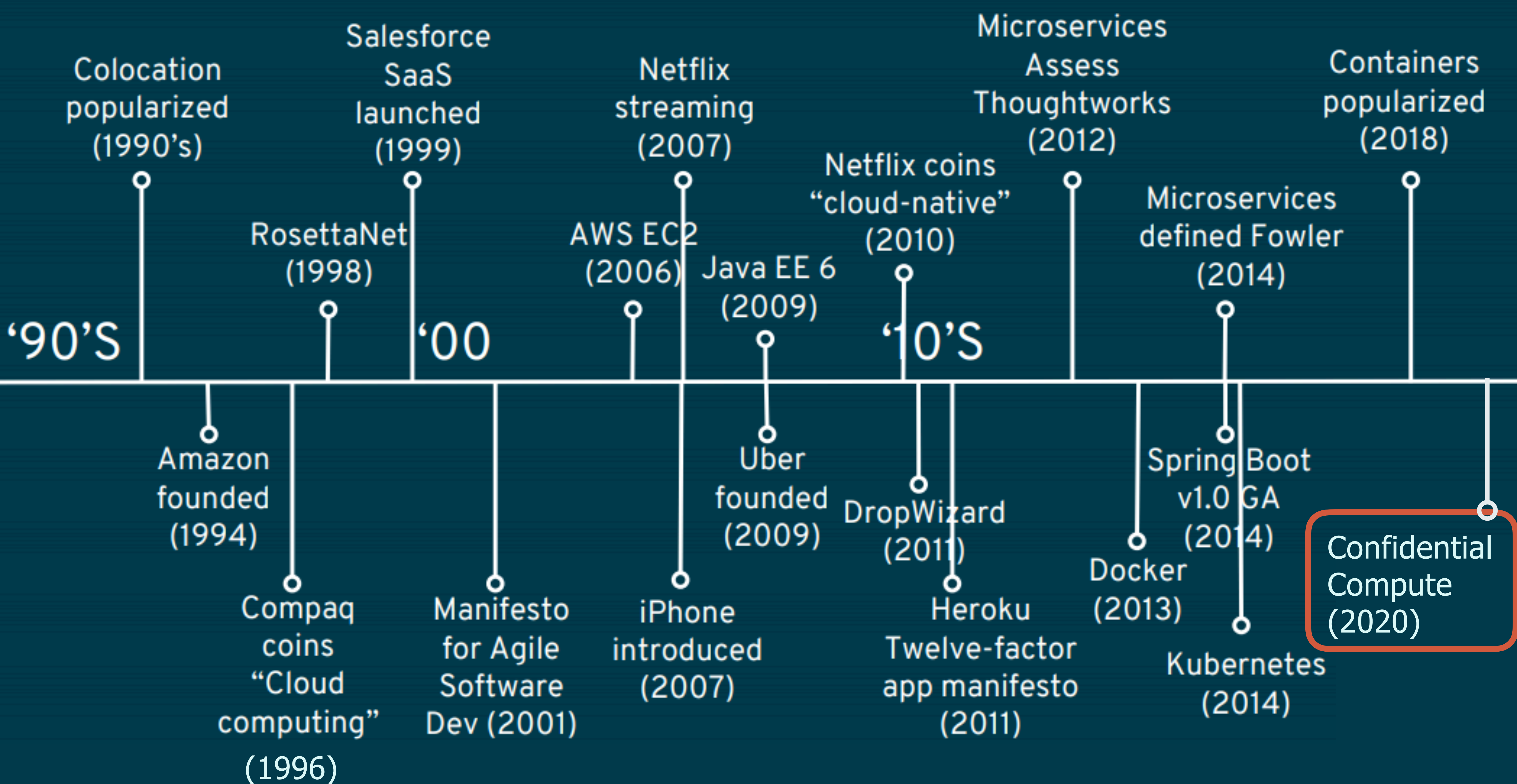
BUILDING CONFIDENTIAL CLOUD-NATIVE APPLICATIONS WITH THE SCONE PLATFORM

Christof Fetzer, PhD
christof.fetzer@scontain.com
https://scontain.com

2020: CONFIDENTIAL COMPUTING



Adapted from Qvarnstrom, Saavedra: „PATH TO CLOUD-NATIVE APP DEV"

christof.fetzer@scontain.com

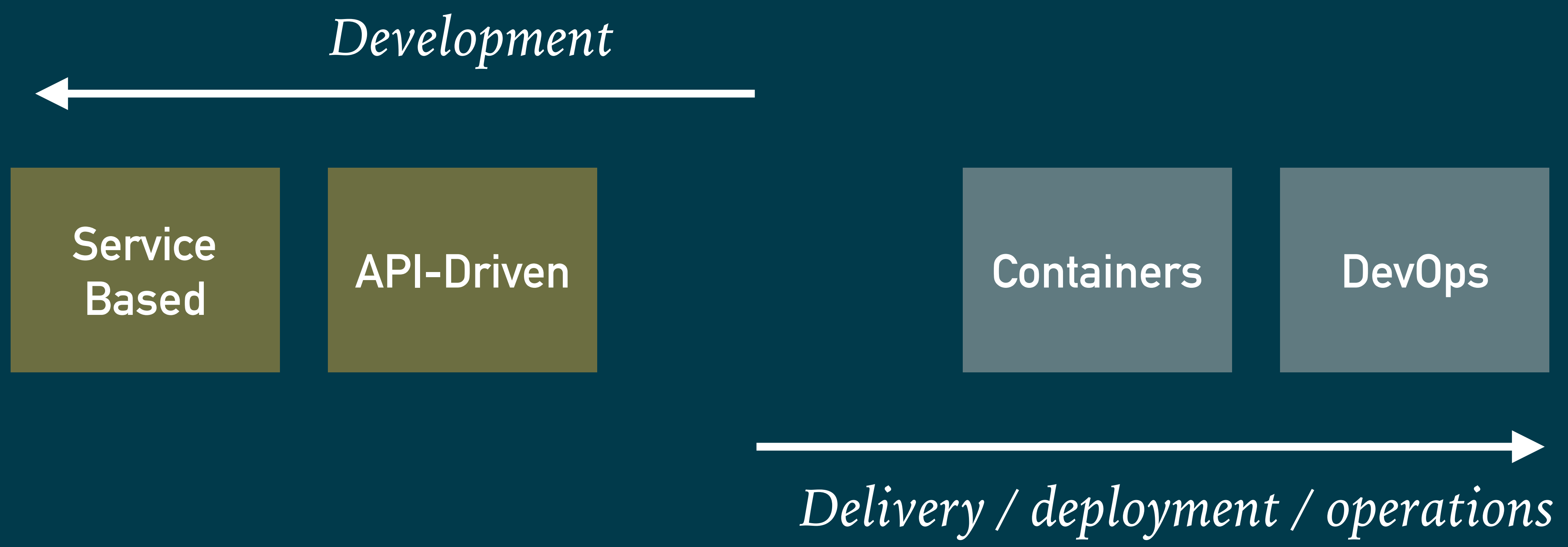
CLOUD-NATIVE APPLICATIONS VS TRADITIONAL APPLICATIONS

	Cloud-Native Application Development	Traditional Development
Focus	Speed to market	Longevity & stability
Development Methodology	Agile development, DevOps	Waterfall, semi-agile development
Teams	Collaborative DevOps team	Isolated dev, operations, QA, and security teams
Delivery Cycle	Short and continuous	Long
Application Architecture	Loosely coupled, service-based, API-based	Tightly-coupled, monolithic
Infrastructure	Container-centric, portable, scales horizontally, on-demand capacity, on premise & cloud	Server-centric, infrastructure dependent, scales vertically, provisioned for peak capacity, on premise

Adapted from Qvarnstrom, Saavedra: „PATH TO CLOUD-NATIVE APP DEV"

christof.fetzer@scontain.com

CLOUD-NATIVE APPLICATIONS



Cloud-Native Application

- an application developed and operated using the cloud-native development/operation model

Adapted from Qvarnstrom, Saavedra: „PATH TO CLOUD-NATIVE APP DEV"

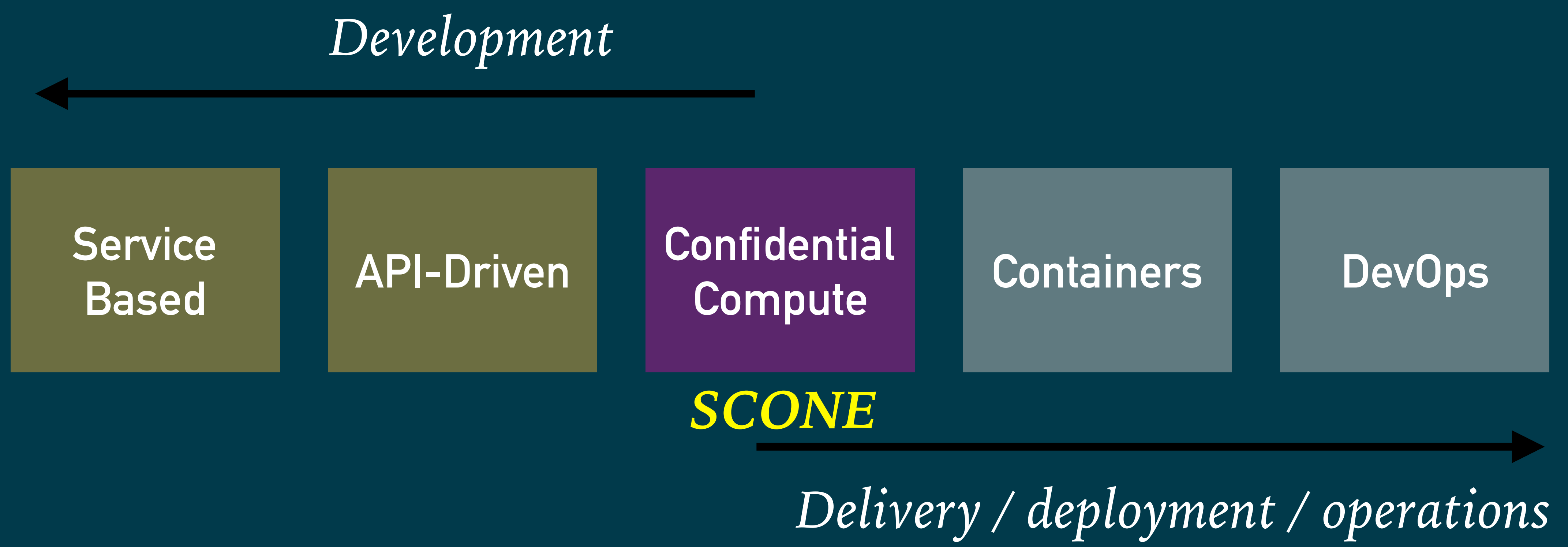
christof.fetzer@scontain.com

CONFIDENTIAL CLOUD-NATIVE APPLICATIONS

	Confidential Cloud-Native Application Development
Security	**data**, **code**, and **keys** are **always encrypted** *NEW* - at rest, in transit, in main memory -
Focus	Speed to market
Development Methodology	Agile development, DevOps
Teams	Collaborative DevOps team
Delivery Cycle	Short and continuous
Application Architecture	Loosely coupled, service-based, API-based communication
Infrastructure	Container-centric, portable, scales horizontally, on-demand capacity

christof.fetzer@scontain.com

CONFIDENTIAL CLOUD-NATIVE APPLICATIONS

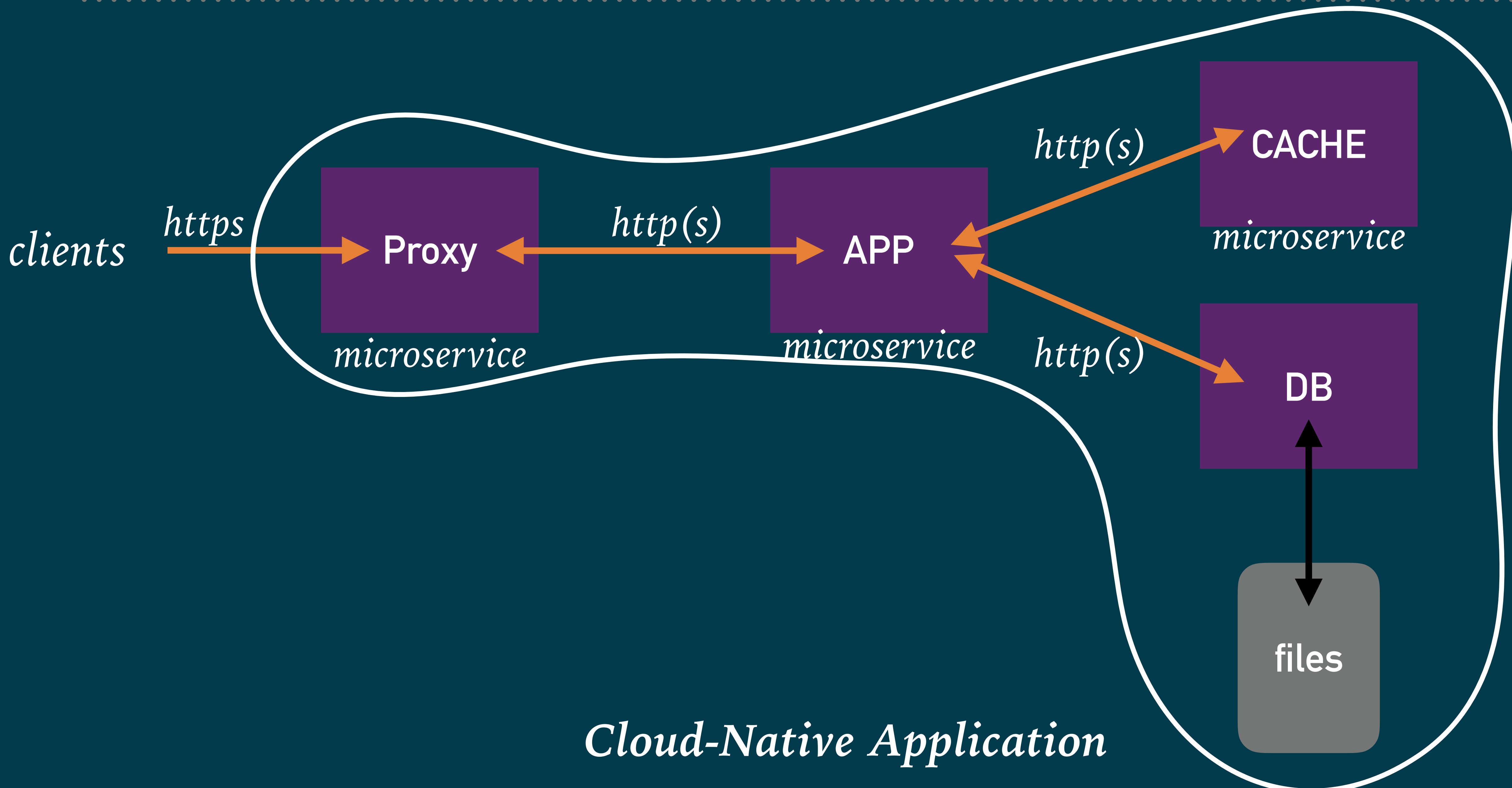


Confidential Cloud-Native Application

- *cloud-native application*
- *protect code, data and keys of application*

christof.fetzer@scontain.com

CLOUD-NATIVE APPLICATION

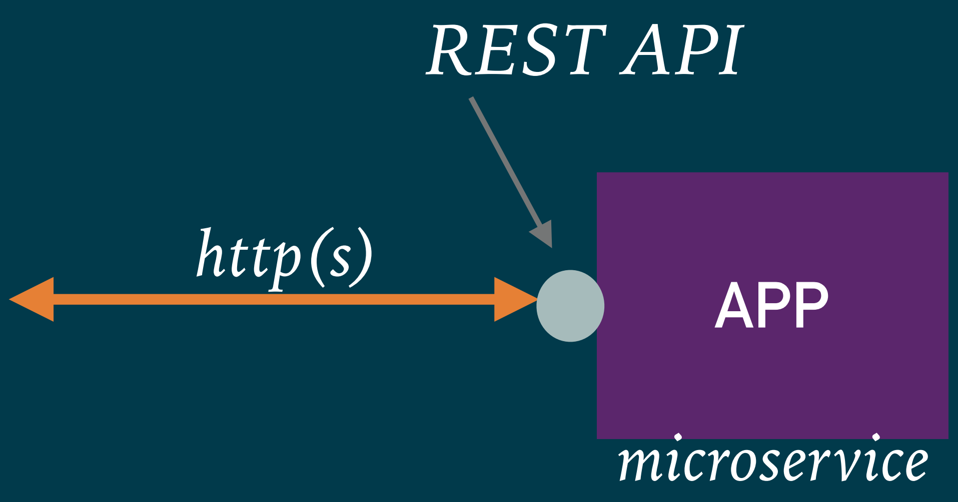


Cloud-Native Application

- an application developed and operated using the cloud-native development/operation model

christof.fetzer@scontain.com

MICROSERVICE



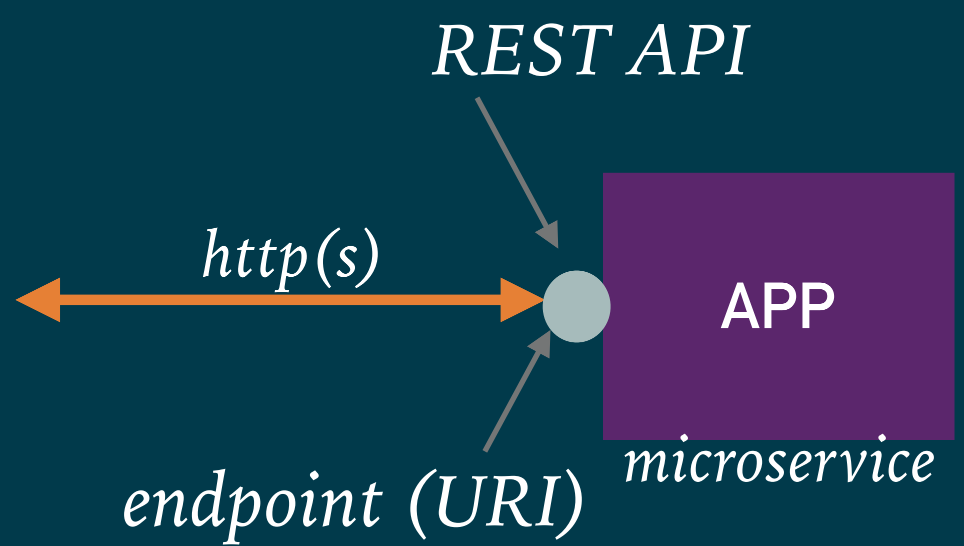
Cloud-Native Application

microservice

- *focus on a single aspect*
- *microservices are small, autonomous services that work together*

christof.fetzer@scontain.com

REST API



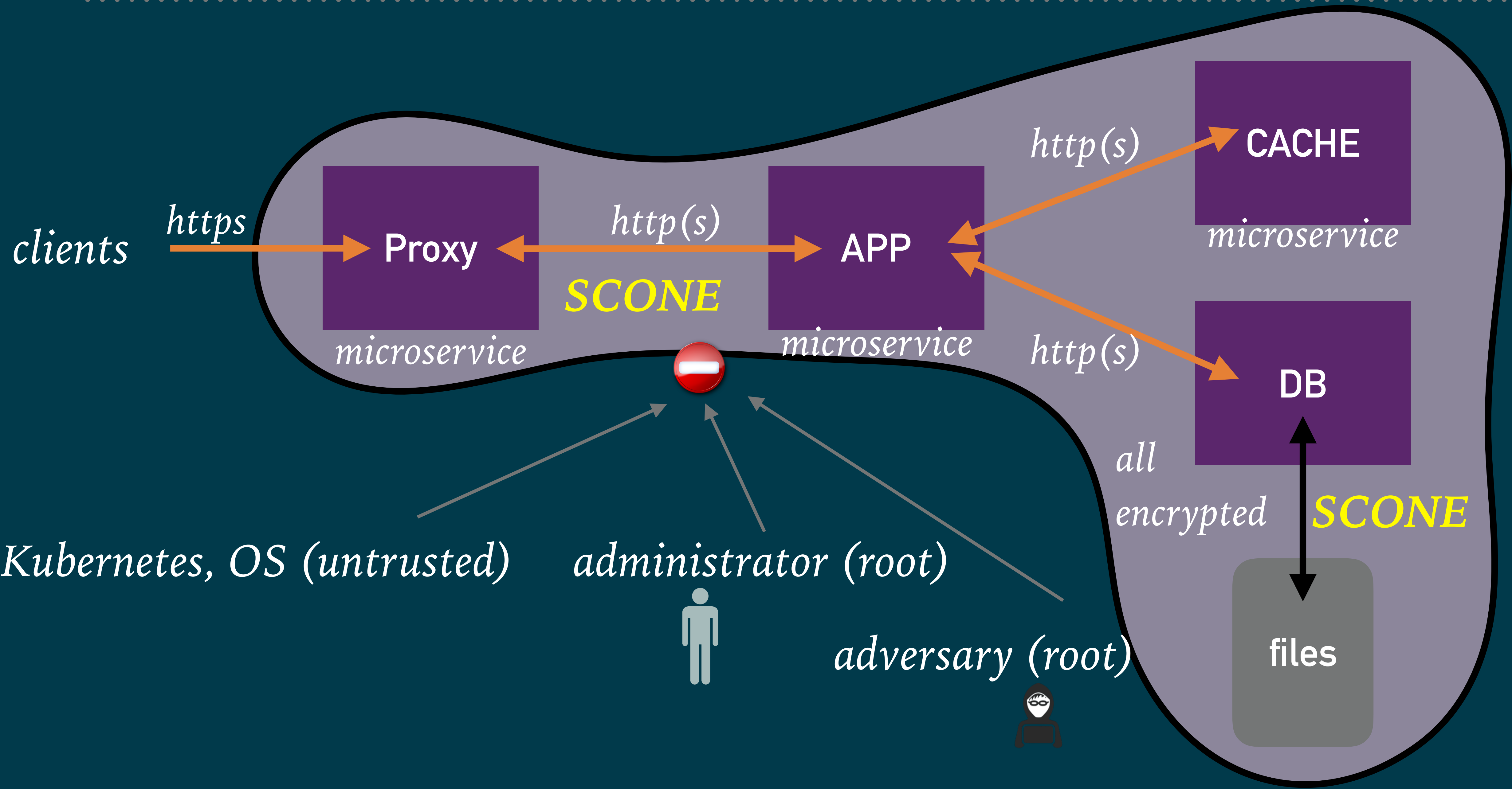
Cloud-Native Application

REST = Representational state transfer

REST API

- identify resource in request
- Fixed methods (from http):
  - **Create (POST)**
  - **Retrieve (GET)**
  - **Update (PUT)**
  - **Delete (DELETE)**

CONFIDENTIAL CLOUD-NATIVE APPLICATION



Confidential Cloud-Native Application

- *cloud-native application*
- *protect code, data and keys of application*

christof.fetzer@scontain.com

PROTECTION GOALS OF CONFIDENTIAL COMPUTE

SCONE

- Protection of
  - **Confidentiality:** information is not made available or disclosed to unauthorized individuals, entities, or processes
  - **Integrity:** information cannot be modified by unauthorized individuals, entities, or processes
  - **Freshness:** information cannot be replaced by old information by unauthorized individuals, entities, or processes

Kubernetes, Ceph

- **Additional Protection goals**:
  - **Availability**: probability that information is available when it is needed
  - **Durability**: probability that information will survive for one year

christof.fetzer@scontain.com

WHAT INFORMATION TO PROTECT?

- Protection of
  - **Code**, e.g., modern AI programs written in Python
  - **Data**, e.g., training data to create AI models
  - **Keys**, e.g., keys used to encrypt databases

christof.fetzer@scontain.com

WHAT INFORMATION TO PROTECT?

- Protection of
  - **Code**, e.g., modern AI programs written in Python
  - **Data**, e.g., training data to create AI models
  - **Keys**, e.g., key used to encrypt database
- **Example**:
  - *MariaDB* supports encryption of database
  - encryption key is stored in configuration file
  - configuration file protected via access control:
    - i.e., can be read and written by MariaDB (user) as well as any root (=privileged) user

christof.fetzer@scontain.com

IS ACCESS CONTROL SUFFICIENT?

- Sufficient if we define that
  - only **authorized user** can become root users

IS ACCESS CONTROL SUFFICIENT?

- Sufficient if we define that
  - only **authorized user** can become root users
- What about
  - **adversary** gaining root access (e.g., stealing credentials)?
  - **authorized** user laid off -> could become an **adversary**?

christof.fetzer@scontain.com

IS ACCESS CONTROL SUFFICIENT?

- Sufficient if we define that
  - only **authorized user** can become root users
- What about
  - **adversary** gaining root access: **must be addressed**
  - **insider attacks: must be addressed**
- **Solution:**
  - Use a **threat model** that gives adversary more power!

christof.fetzer@scontain.com

THREAT MODEL

ADVERSARY HAS ROOT & HW ACCESS!

christof.fetzer@scontain.com

WHY ASSUME ADVERSARY HAS ROOT ACCESS?

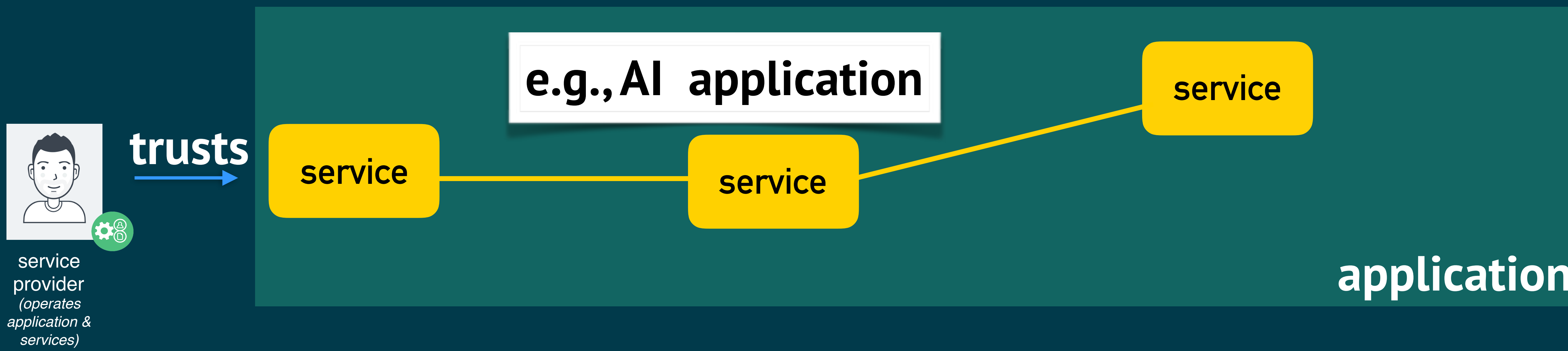
- **Reasons:**
  - **Legal**:
    - cloud provider might be legally required to provide access to the data
  - **Liability**:
    - too expensive to err on the threat model
  - **Limits of access control:**
    - How do we know that we can trust individual user?
  - **Software complexity**: cannot assume that software is correct (see **Defender's dilemma**)
  - **Hardware complexity**: cannot assume that hardware is correct (see BMC, firmware, …)
  - …

christof.fetzer@scontain.com

LEGAL REASON

Definition: TRUST

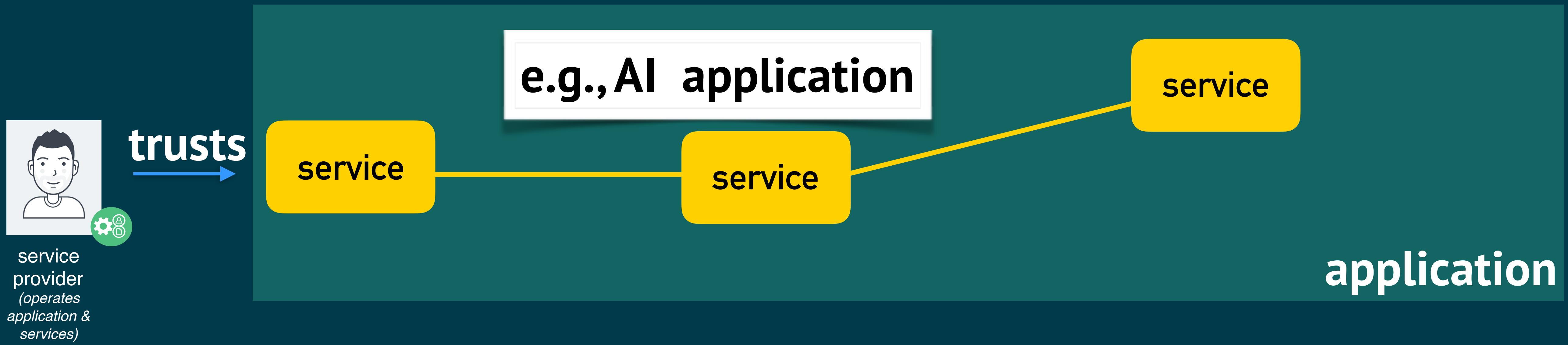
*we can **justifiably assume** the **security**, **reliability**, or …*
*of someone or **something***



service provider trusts that the services do what they supposed to do

christof.fetzer@scontain.com

APPLICATION-ORIENTED VIEW

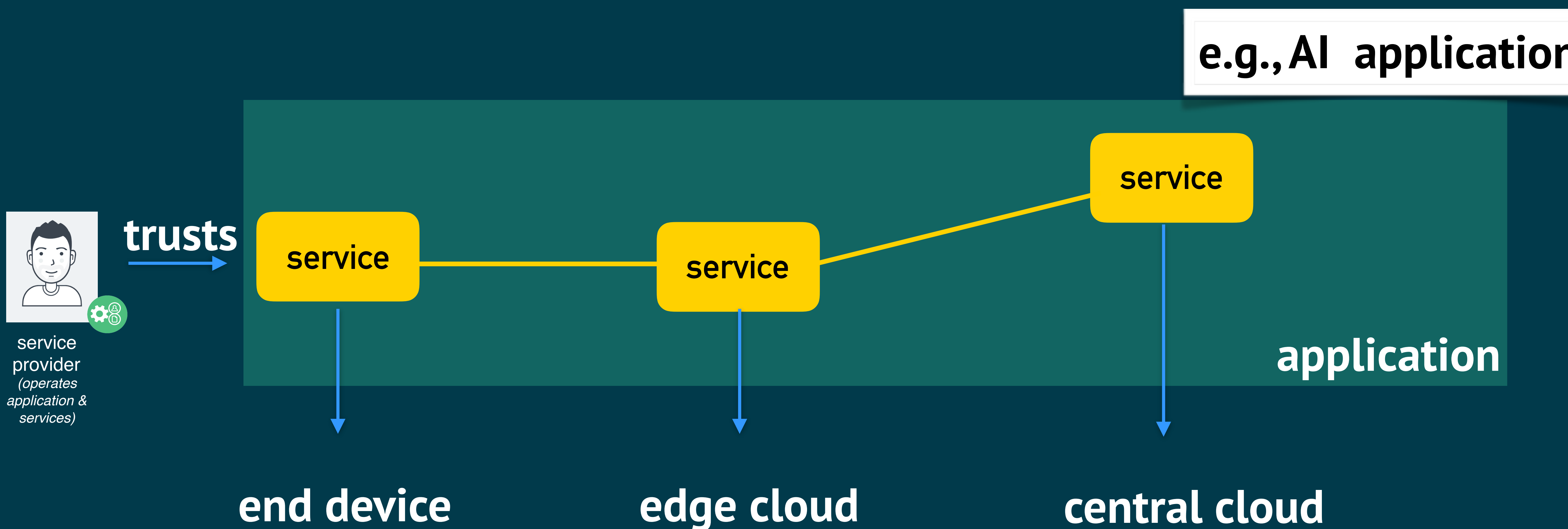


service provider trusts that the services do what they supposed to do

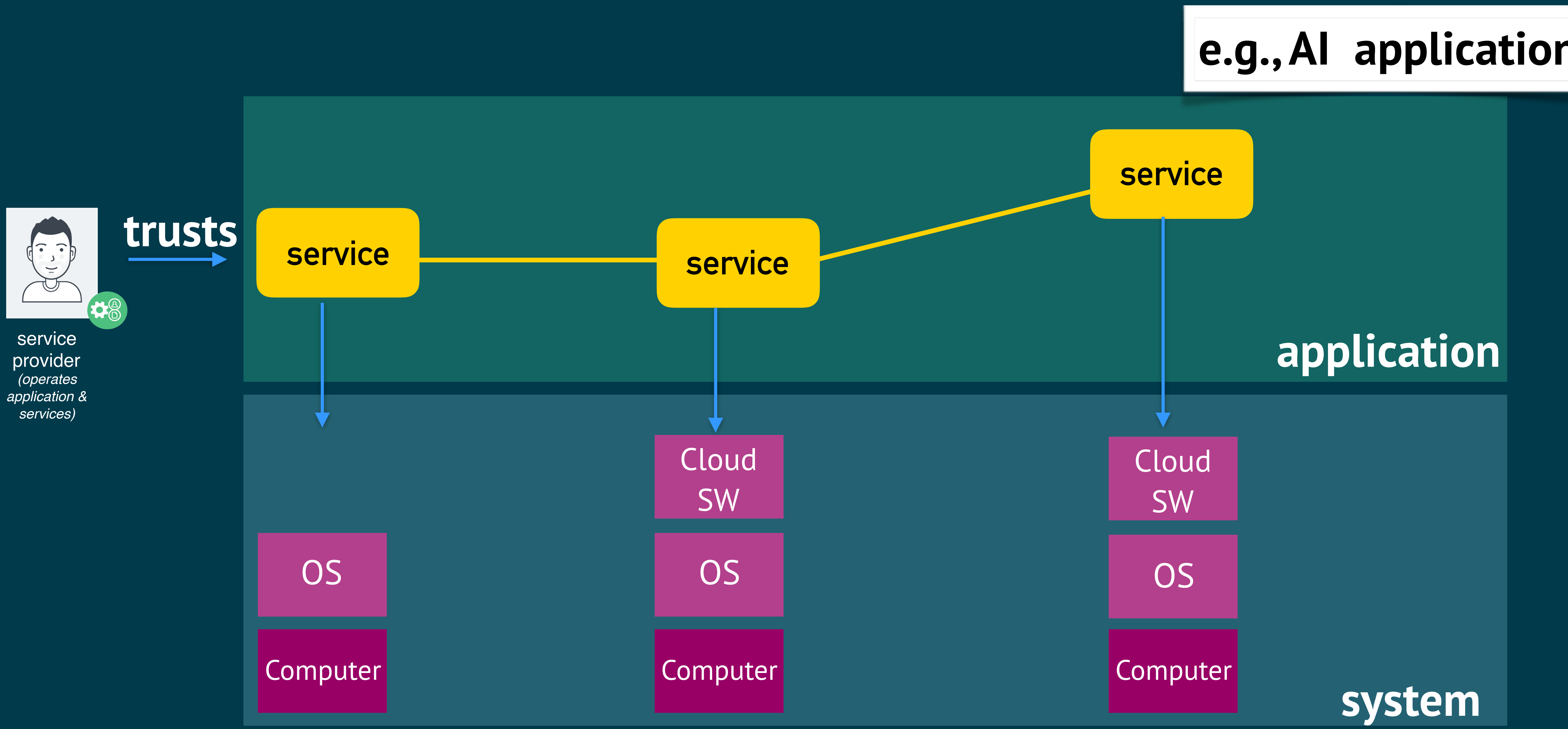
Why?

- service provider controls the source code used in the services

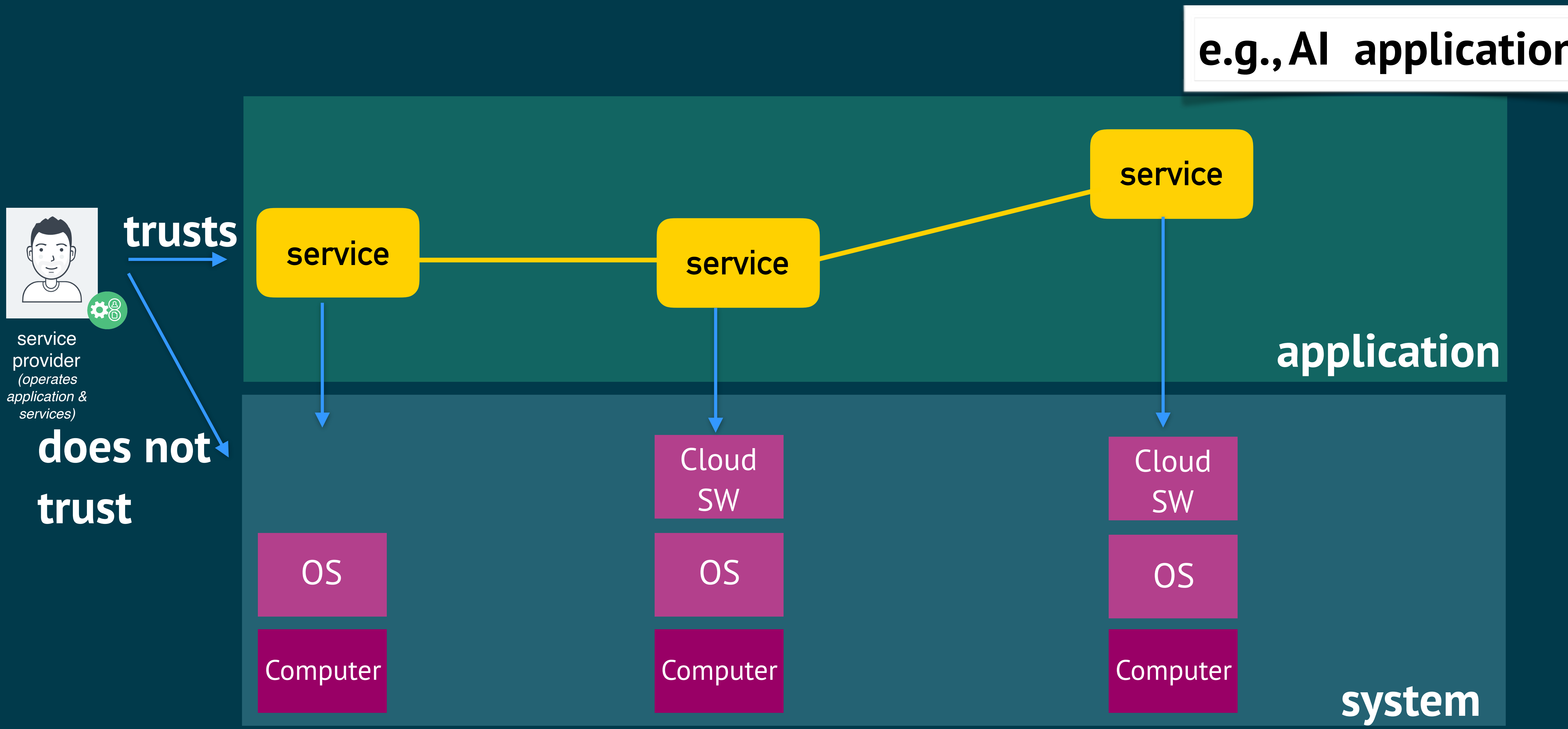
APPLICATION-ORIENTED VIEW



APPLICATION-ORIENTED VIEW



PROBLEM: TRUST

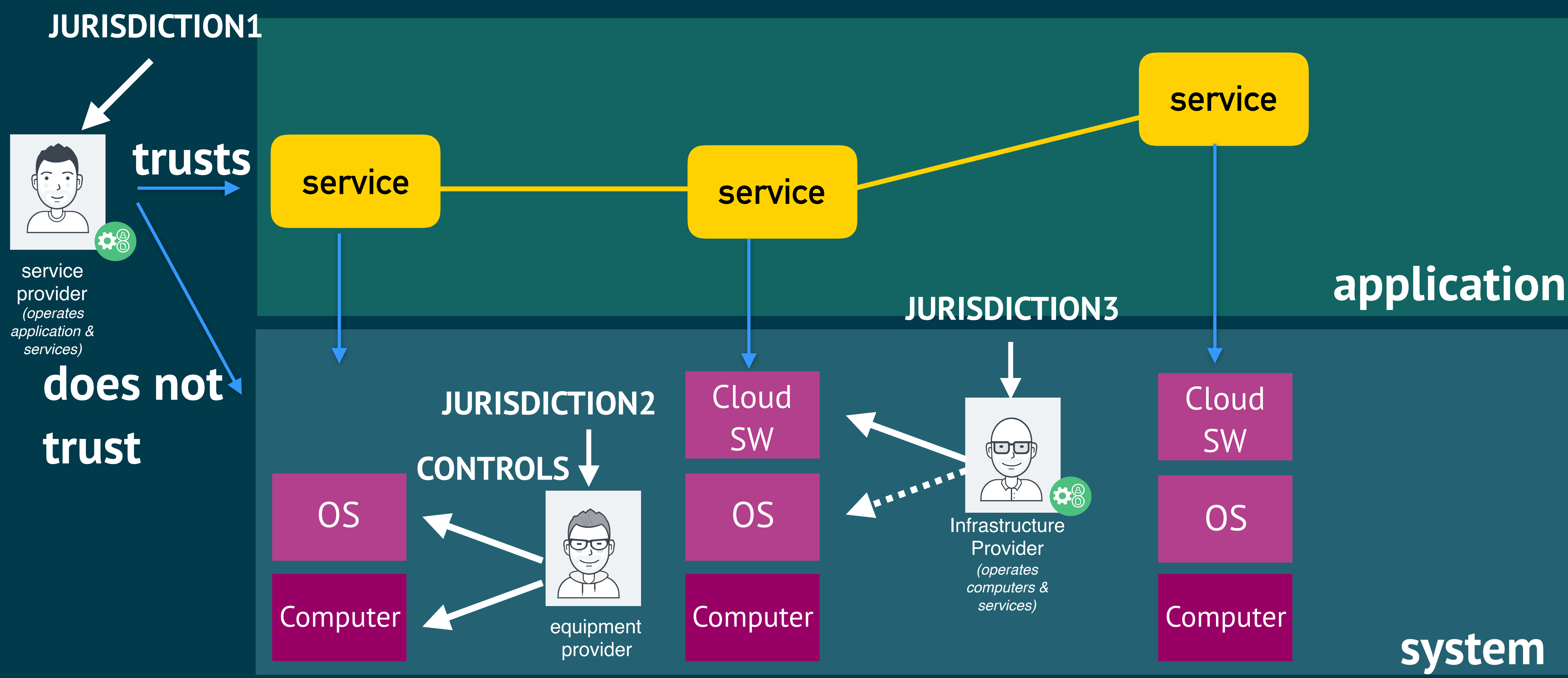


christof.fetzer@scontain.com

PROBLEM: TRUST

Why?

- service provider not in control of the system! Different jurisdictions.

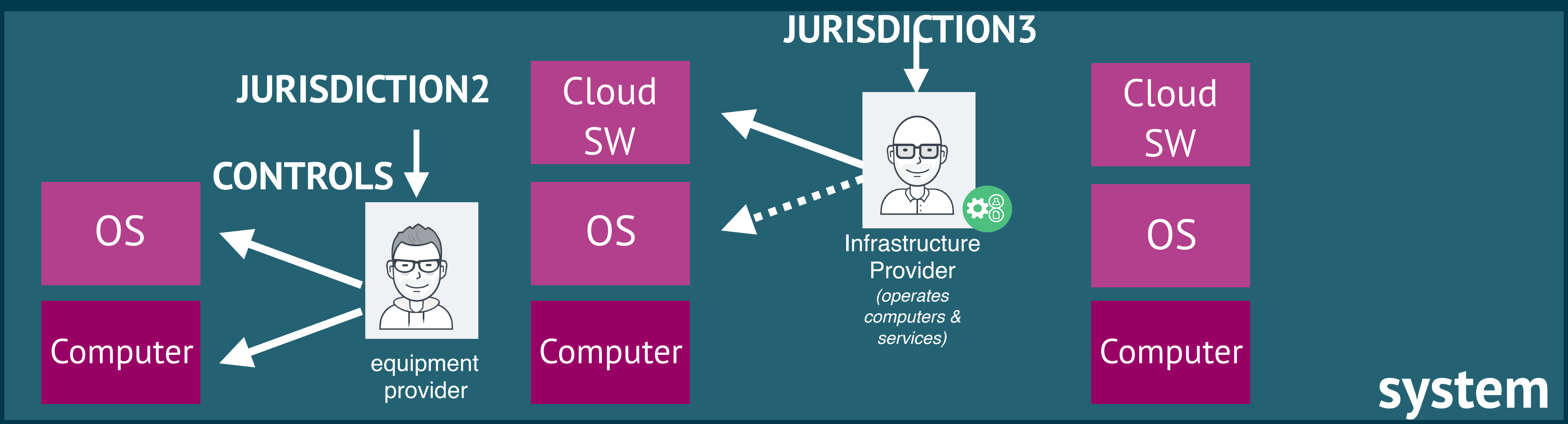


christof.fetzer@scontain.com

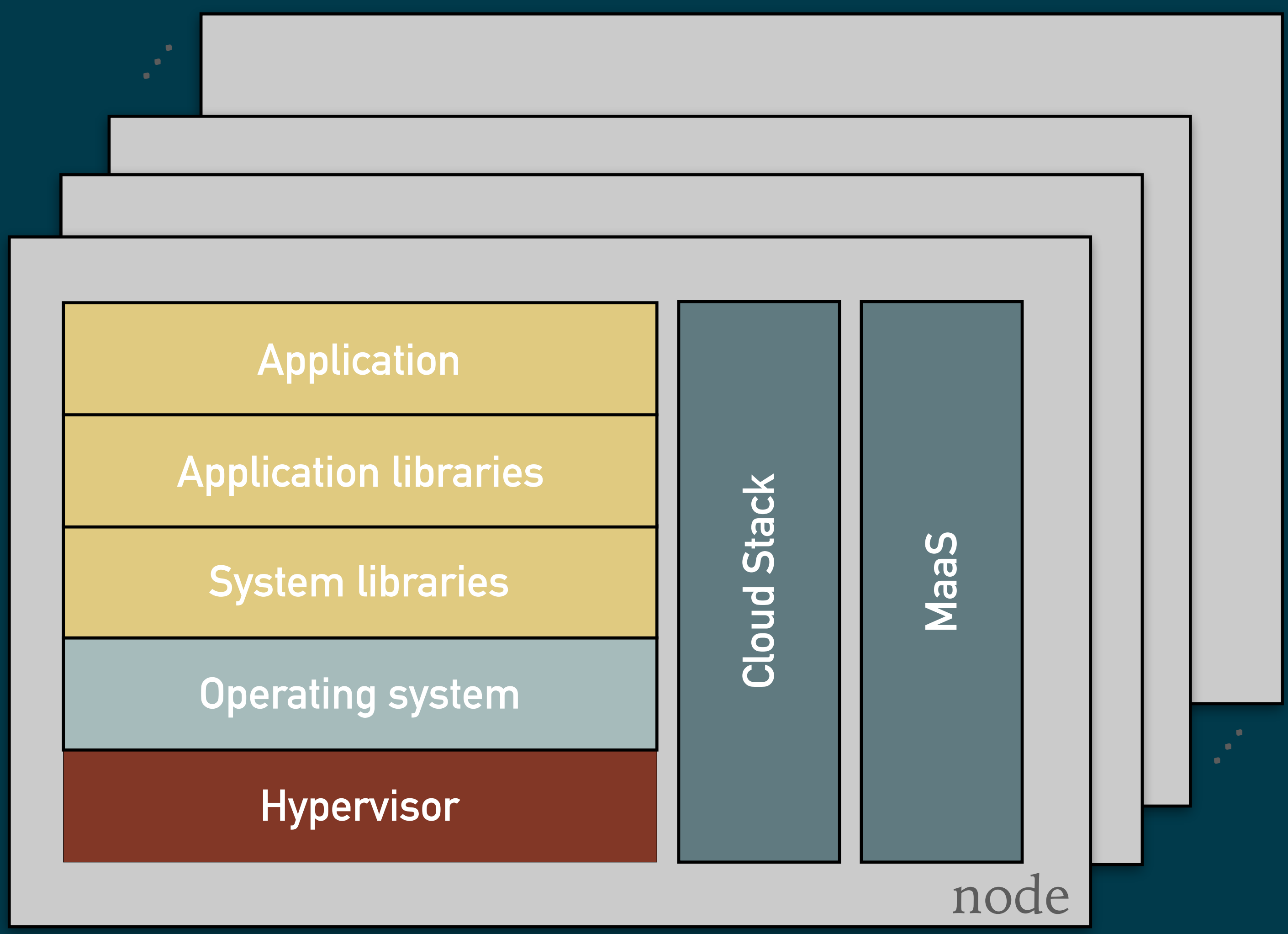
TRADITIONAL APPROACH: SECURITY CERTIFICATION

Why not depend on security certification?

- **system to complex to understand the security!**
- **security certification typically shallow & historic software/firmware version**



SOFTWARE COMPLEXITY



cloud software stack

DEFENDER'S DILEMMA

- **Attackers**:
  - success by exploiting a single vulnerability
- **Defender**:
  - must protect against every vulnerability
    - **not only in application**
  - millions of lines of source code
  - not all known by service provider

christof.fetzer@scontain.com

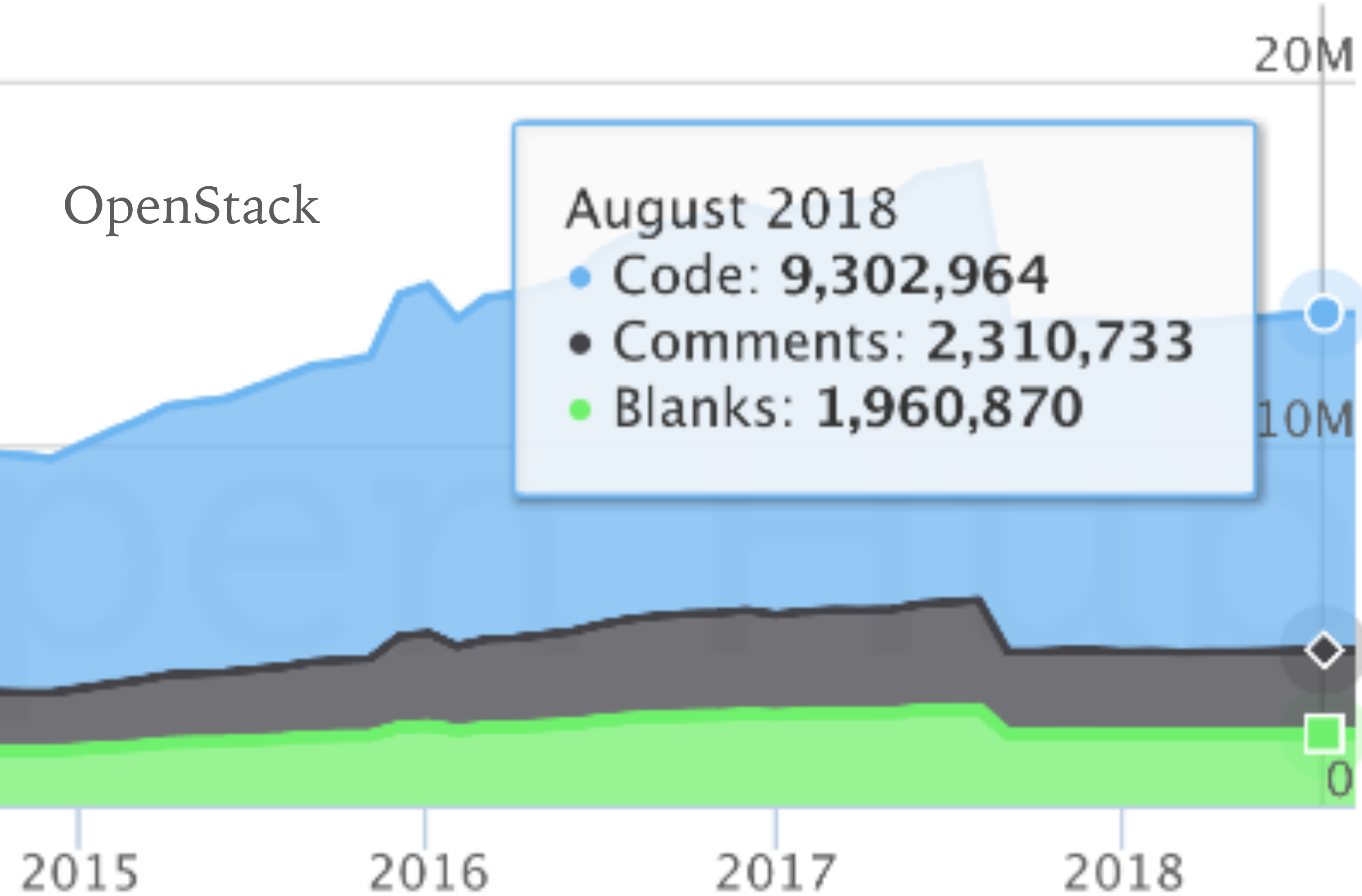
CLOUD SOFTWARE STACK

Applications run on top of software stack

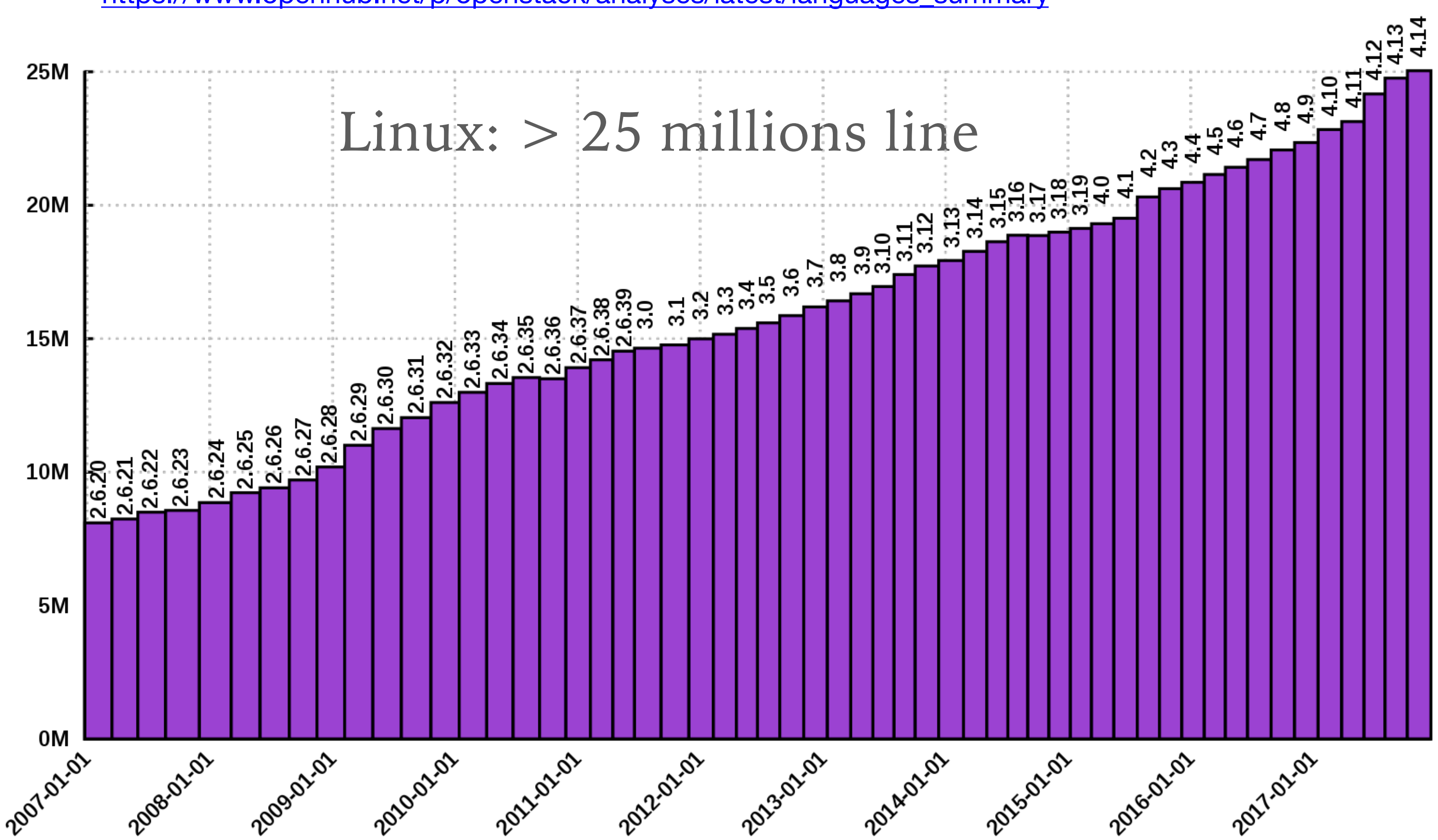
- millions of lines of code

Cloud stack consists of

- VM/container engine
- operating system
- hypervisor
- node management service



https://www.openhub.net/p/openstack/analyses/latest/languages_summary



StefanPohl, CC0, https://commons.wikimedia.org/w/index.php?curid=41549243

christof.fetzer@scontain.com

VULNERABILITIES

- **Coverity reports:**
  - 1 defect per approx.1700 lines of code
- **Kernel self protection project:**
  - 500 security bugs fixed in Linux during the last 5 years
  - each bug stayed about 5 years inside kernel
- Xen hypervisor
  - 184 vulnerabilities (2012-2016)
    [http://www.cvedetails.com/product/23463/XEN-XEN.html?vendor_id=6276]
- **Coverity:**
  - quality of commercial software is not better than open source software

[KSPP] Kees Cook, The State of Kernel Self Protection Project, Linux Security Summit (LSS), 2016

[Coverity] Open Source Report 2014 - Coverity, go.coverity.com/rs/157-LQW.../2014-Coverity-Scan-Report.pdf

christof.fetzer@scontain.com

VULNERABILITIES

- **Coverity reports:**
  - 1 defect per approx.1700 lines of code
- **Kernel self protection project:**
  - 500 security bugs fixed in Linux during the last 5 years
  - each bug stayed about 5 years inside kernel
- Xen hypervisor
  - 184 vulnerabilities (2012-2016)

    [http://www.cvedetails.com/product/23463/XEN-XEN.html?vendor_id=6276]

Bugs are especially bad in privileged software as it may result in unrestricted access to the system

christof.fetzer@scontain.com

HARDWARE PROTECTED MODE NOT SUFFICIENT

- Protected mode (rings) protects OS from applications, and applications from one another…
  - … until a malicious applications exploits a flaw to gain full privileges and then tampers with the OS or other applications
  - Applications not protected from privileged code attacks
- The attack surface is the whole software stack
  - Applications, OS, VMM, drivers, BIOS…

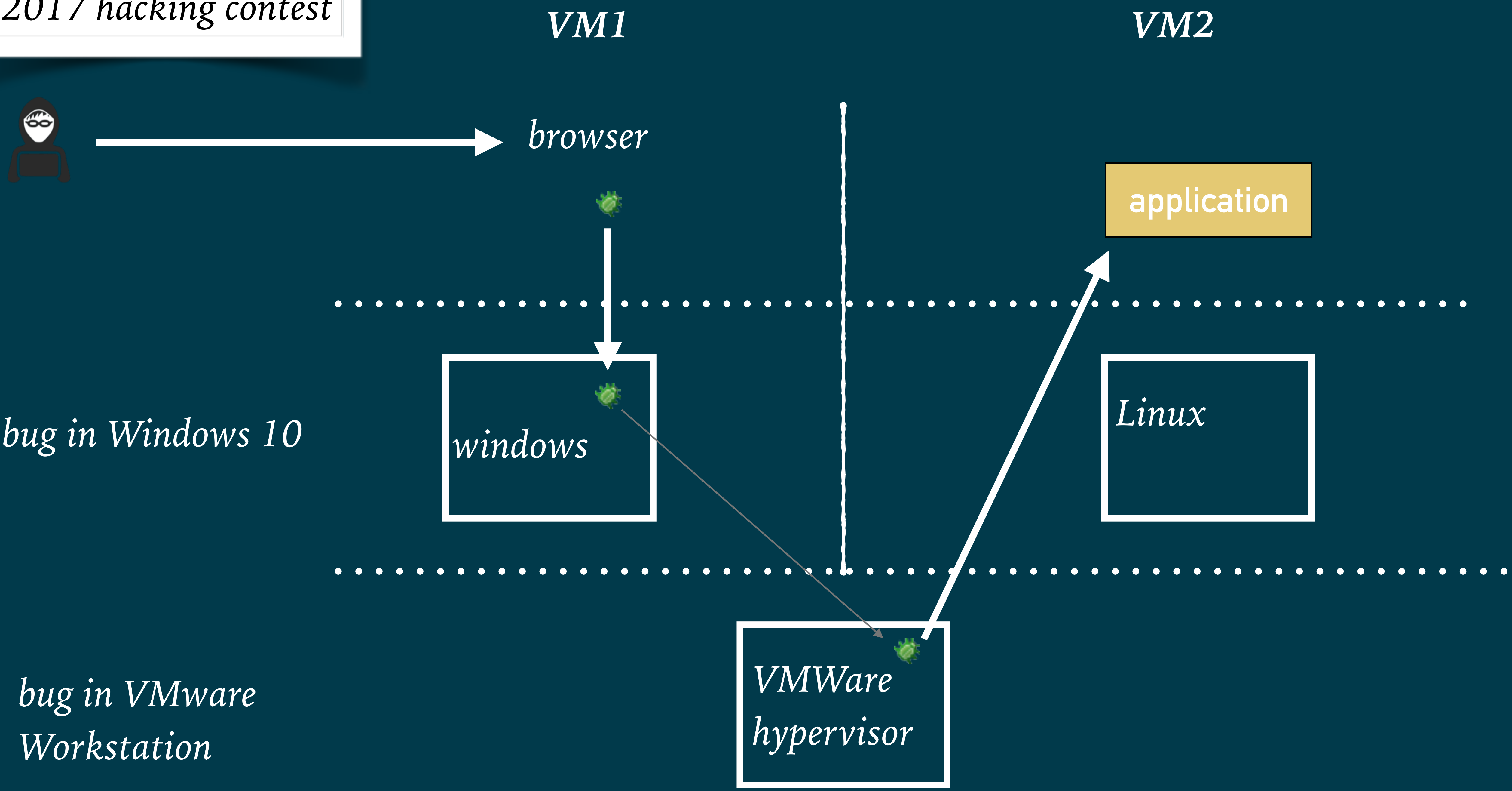
christof.fetzer@scontain.com

ATTACKING VIRTUAL MACHINES

2017 hacking contest

bug in Windows 10

bug in VMware Workstation



https://arstechnica.com/security/2017/03/hack-that-escapes-vm-by-exploiting-edge-browser-fetches-105000-at-pwn2own/

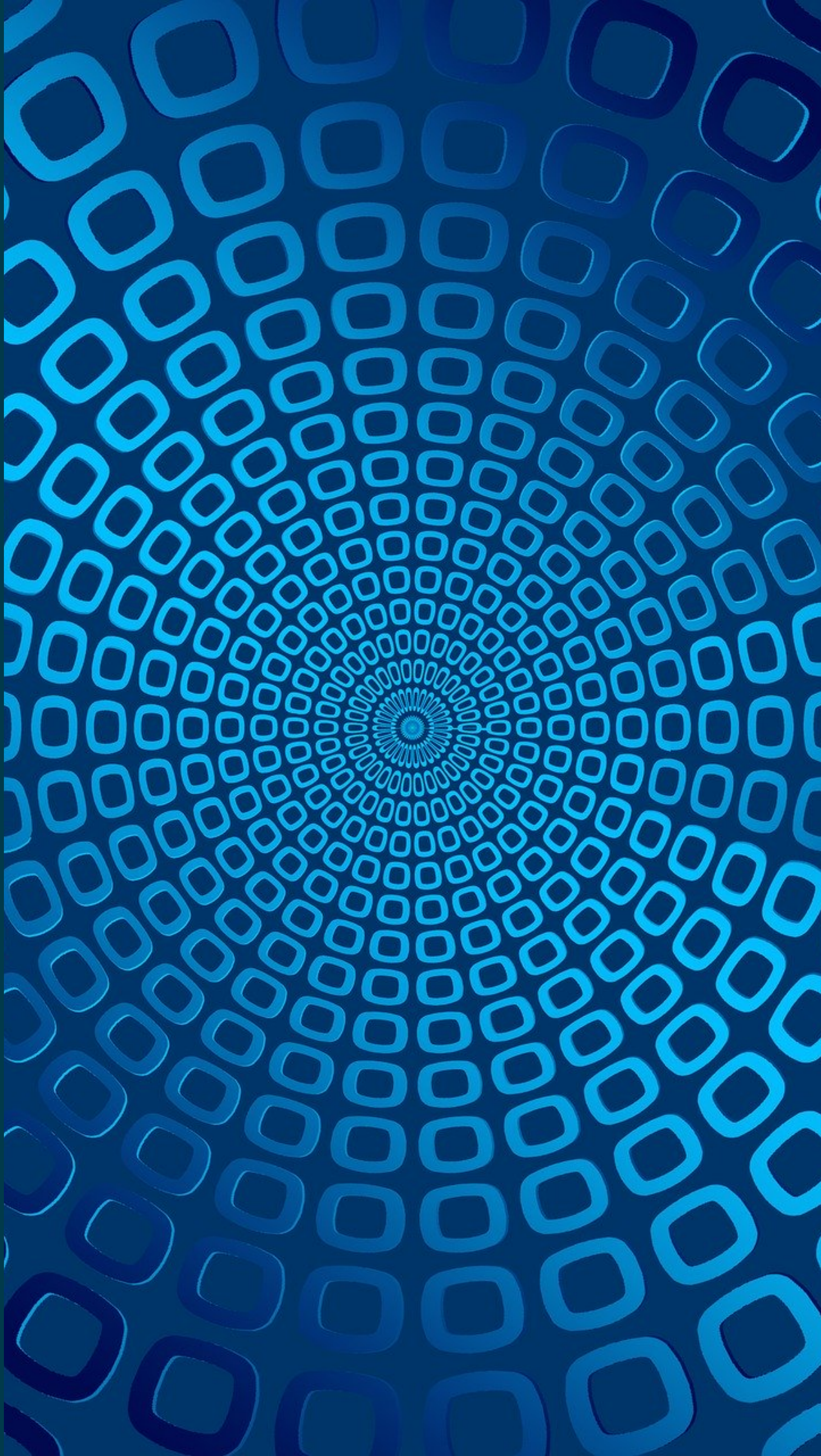
EXAMPLE: HEARTBLEED BUG

- Serious vulnerability in the popular OpenSSL cryptographic software library
    - Very widely used: apache/nginx (66% of Web servers), email servers, chat servers, VPN, etc.
- Buffer overrun when replying to a heartbeat message
- Allows anyone on the Internet to read the memory of the systems protected by the vulnerable versions of the OpenSSL software
    - The attacker can obtain sensitive data from server's memory: passwords, private keys, …

SCONE

HARDWARE, FIRMWARE, MANAGEMENT FEATURES, …

Buggy on multiple levels…



CONFIDENTIALITY

information is not made available or disclosed to unauthorized individuals, entities, or processes

service

keys, data, code

SECRETS IN MAIN MEMORY

christof.fetzer@scontain.com

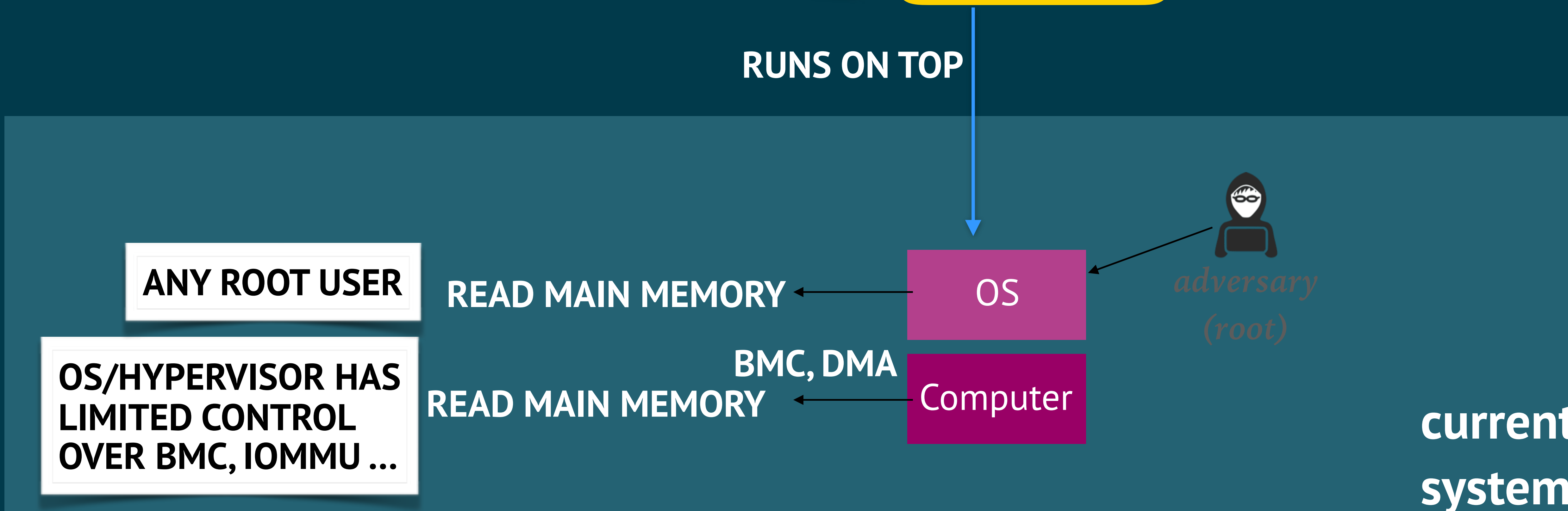
CONFIDENTIALITY

information is not made available or disclosed to unauthorized individuals, entities, or processes

service

keys, data, code

SECRETS IN MAIN MEMORY



BMC, INTEL ME, … : MANAGEMENT INTERFACE TO ACCESS REMOTE MACHINE / MEMORY

CONFIDENTIALITY

information is not made available or disclosed to unauthorized individuals, entities, or processes

service

keys, data, code

SECRETS IN MAIN MEMORY

RUNS ON TOP

ANY ROOT USER

READ MAIN MEMORY

OS

adversary (root)

OS/HYPERVISOR HAS LIMITED CONTROL OVER BMC, IOMMU …

BMC, DMA READ MAIN MEMORY

Computer

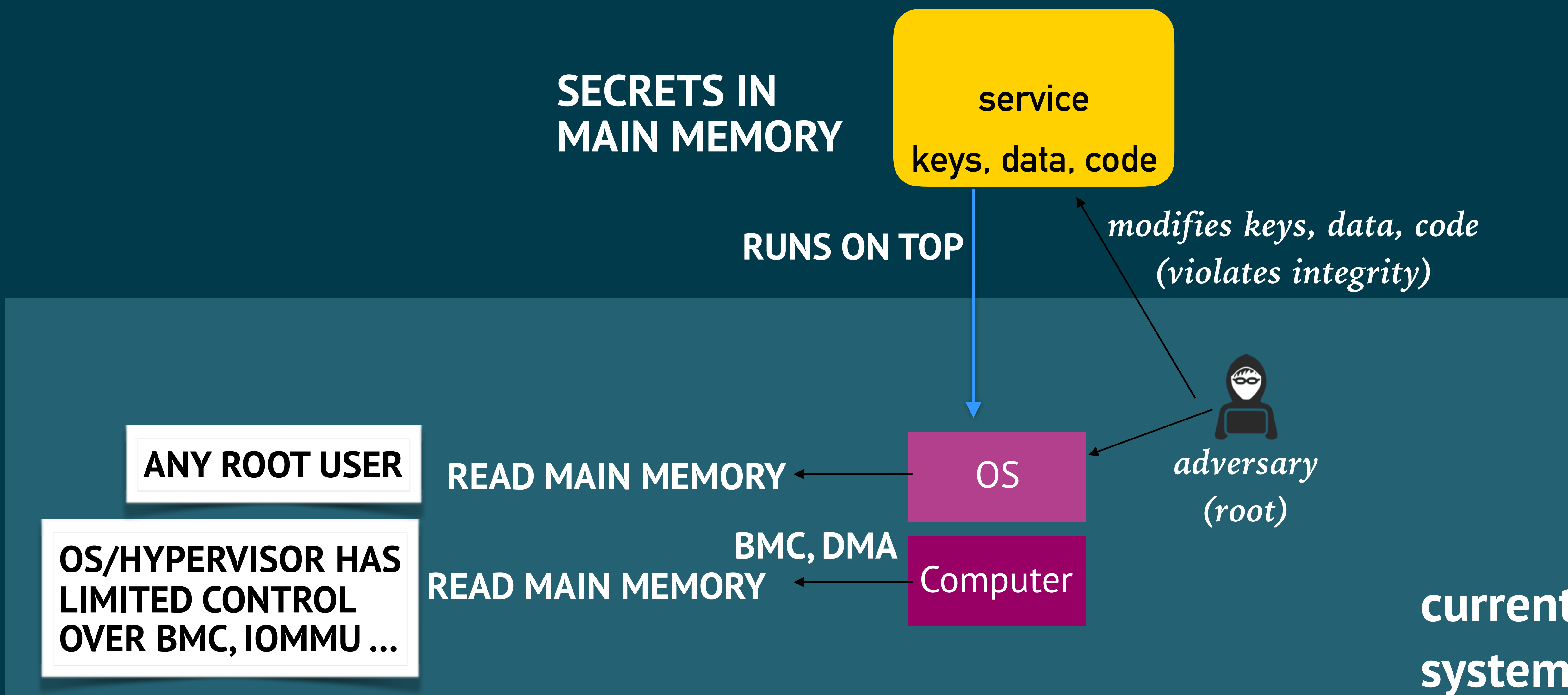
current system

BMC, INTEL ME, … : MANAGEMENT INTERFACE TO ACCESS REMOTE MACHINE / MEMORY

AN APPLICATION CANNOT PROTECT CONFIDENTIALITY OF ITS SECRETS AGAINST ADVERSARIES WITH ROOT ACCESS (TRADITIONAL SYSTEMS)

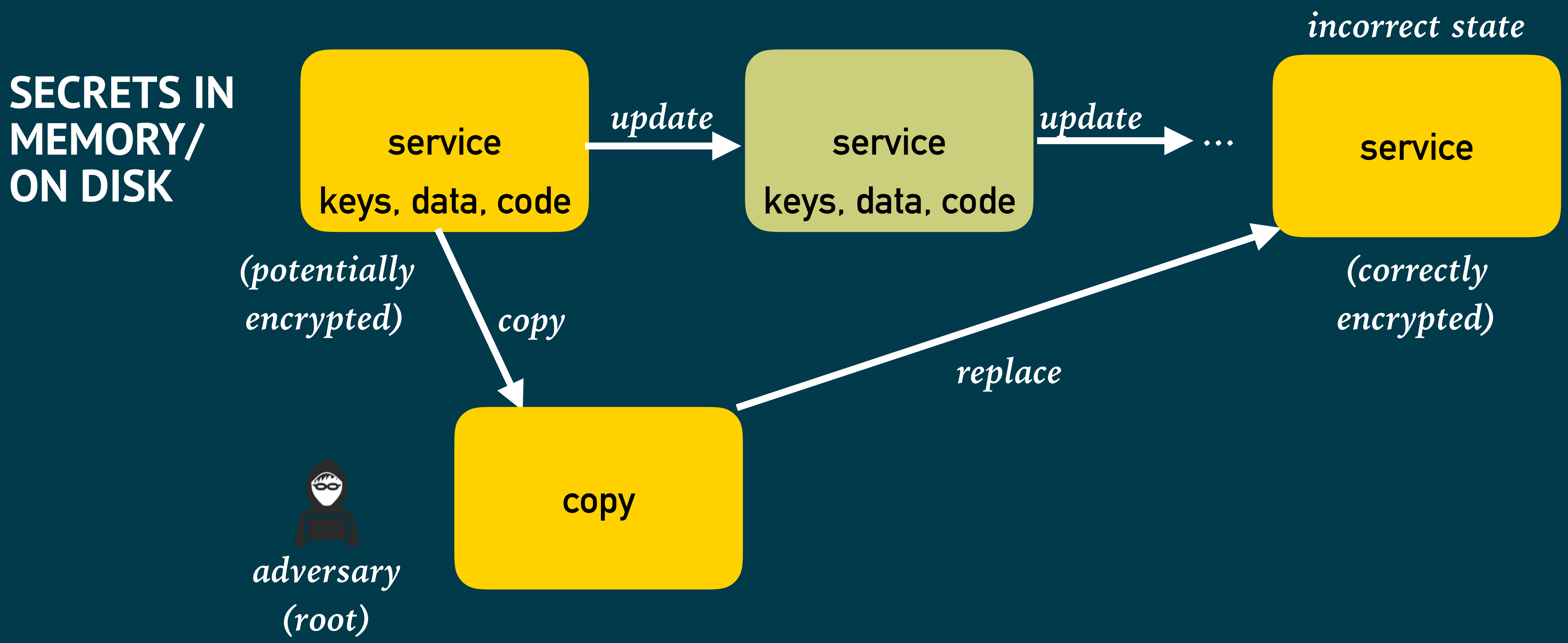
christof.fetzer@scontain.com

INTEGRITY



AN APPLICATION CANNOT PROTECT INTEGRITY OF ITS SECRETS AGAINST ADVERSARIES WITH ROOT ACCESS (TRADITIONAL SYSTEMS)

FRESHNESS

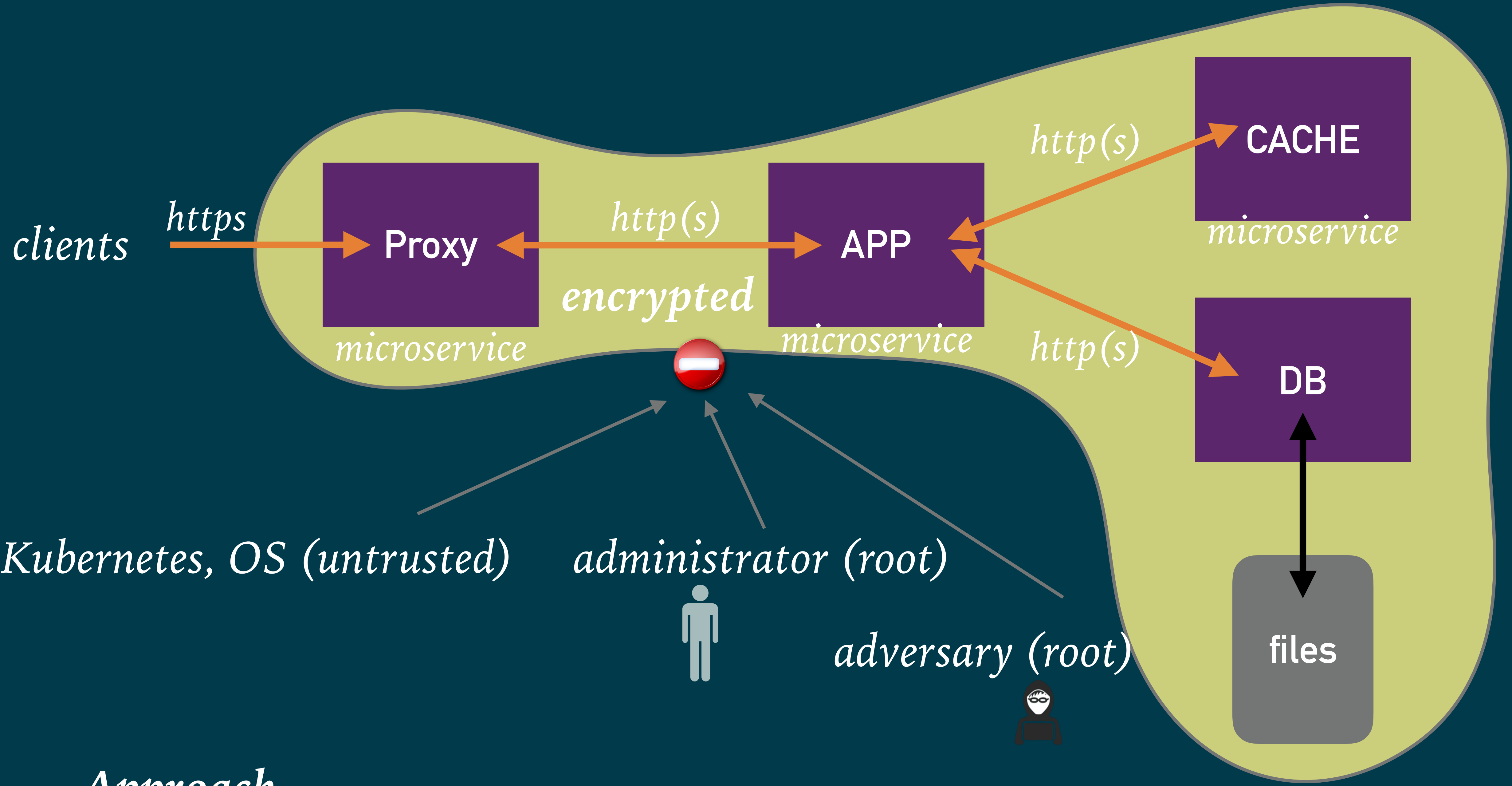


freshness: *information cannot be rolled-back to an old state that has already been replaced*

PROBLEM: HOW TO ENSURE CONFIDENTIALITY, INTEGRITY AND FRESHNESS IF ADVERSARY HAS ROOT AND HARDWARE ACCESS?

christof.fetzer@scontain.com

CONFIDENTIAL CLOUD-NATIVE APPLICATION

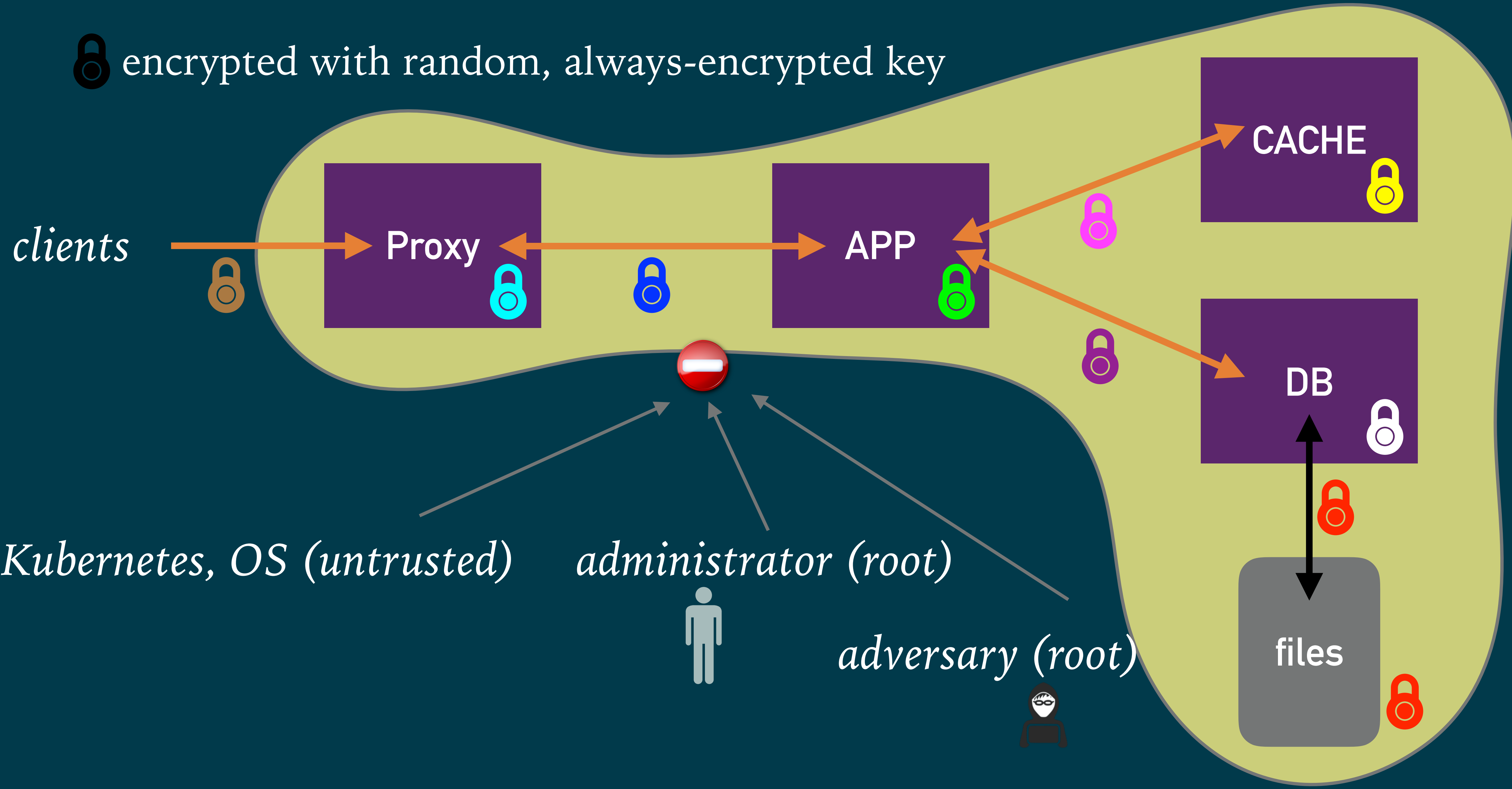


Approach

- *Encrypt everything (code, data, keys), and*
- *is never unencrypted: always encrypted at least once*

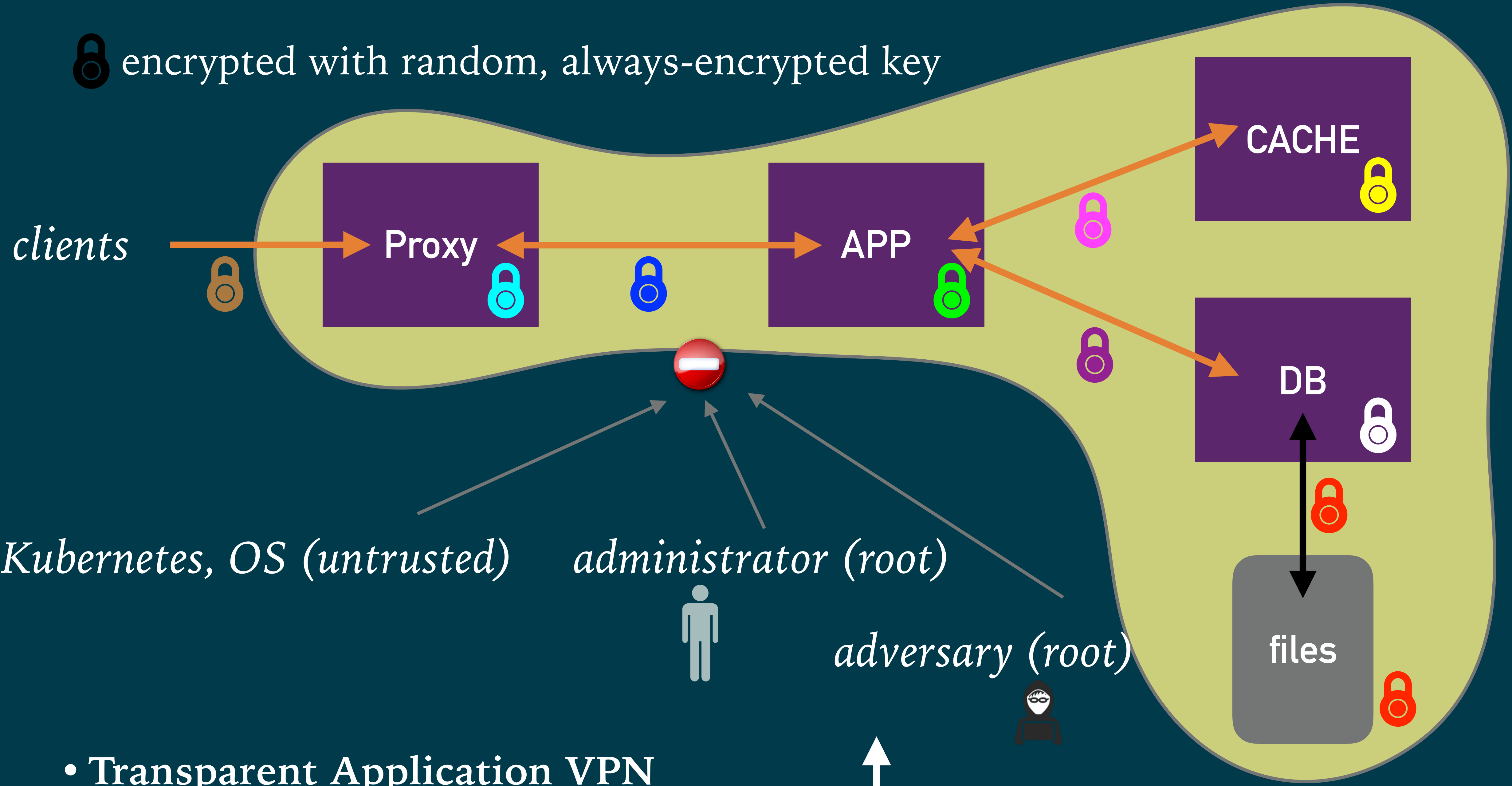
christof.fetzer@scontain.com

SCONE ENCRYPTS DATA IN TRANSIT, AT REST AND IN MAIN MEMORY



christof.fetzer@scontain.com

SCONE ENCRYPTS DATA IN TRANSIT, AT REST AND IN MAIN MEMORY



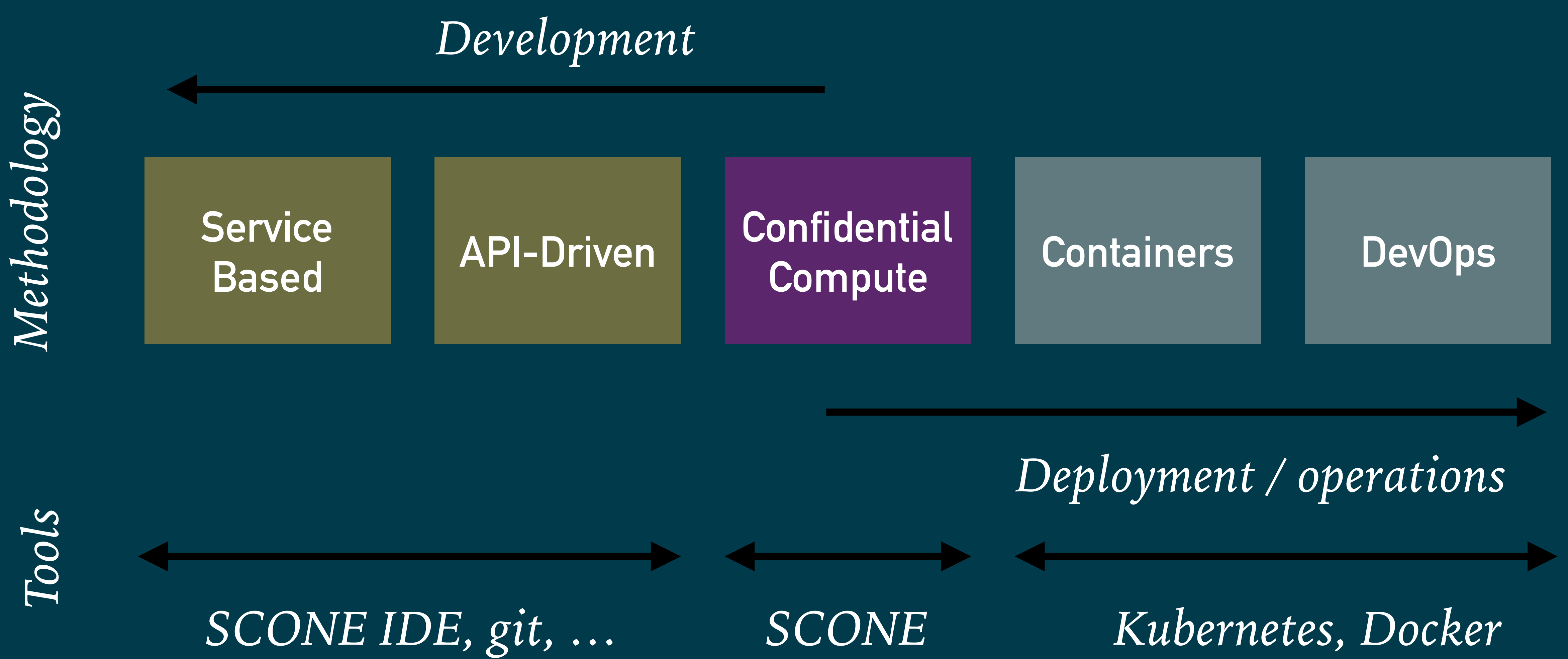
- Transparent Application VPN
- Transparent encryption of files
- Transparent execution inside of TEE

No source code changes required!

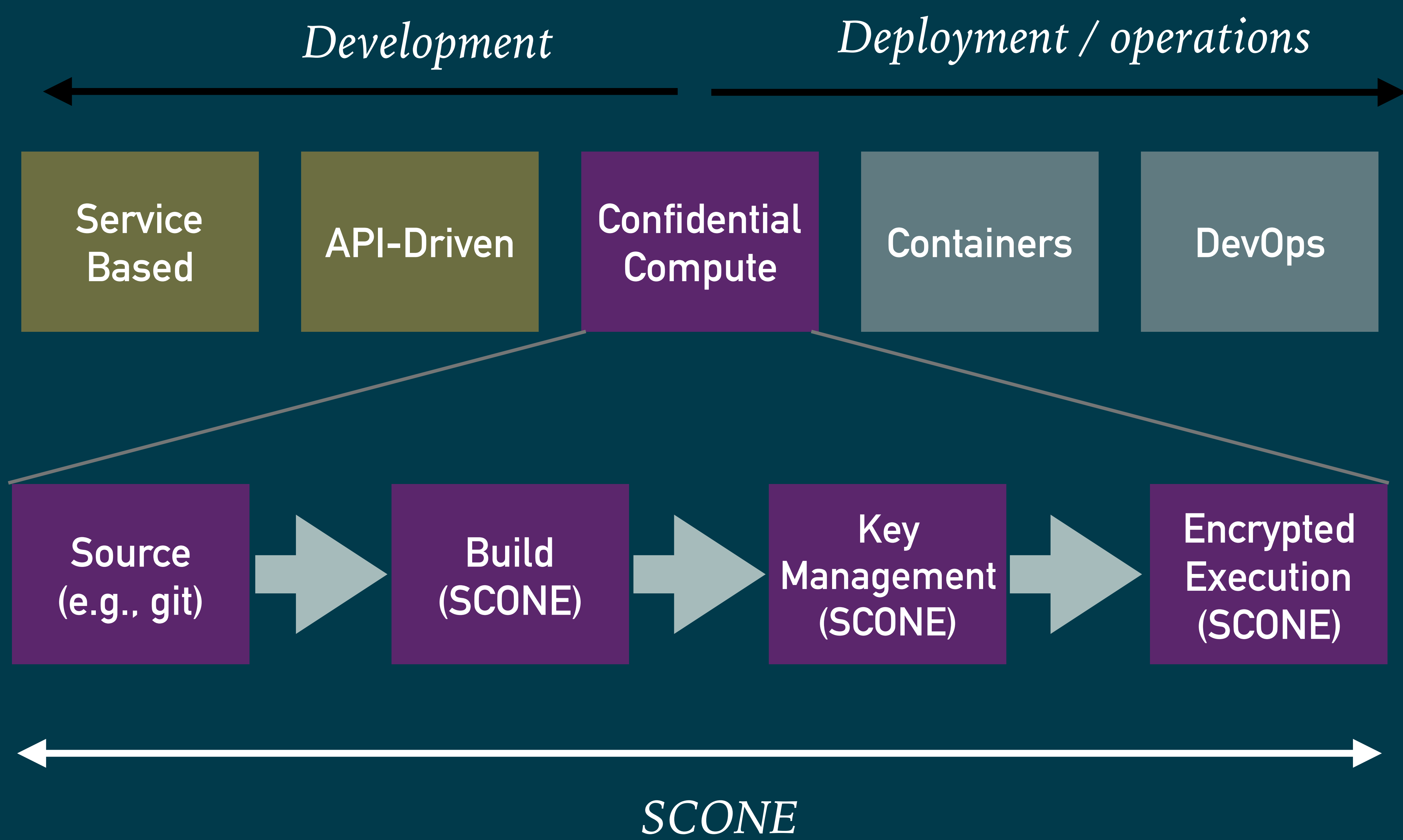
christof.fetzer@scontain.com

HOW?

CONFIDENTIAL CLOUD-NATIVE APPLICATION DEVELOPMENT

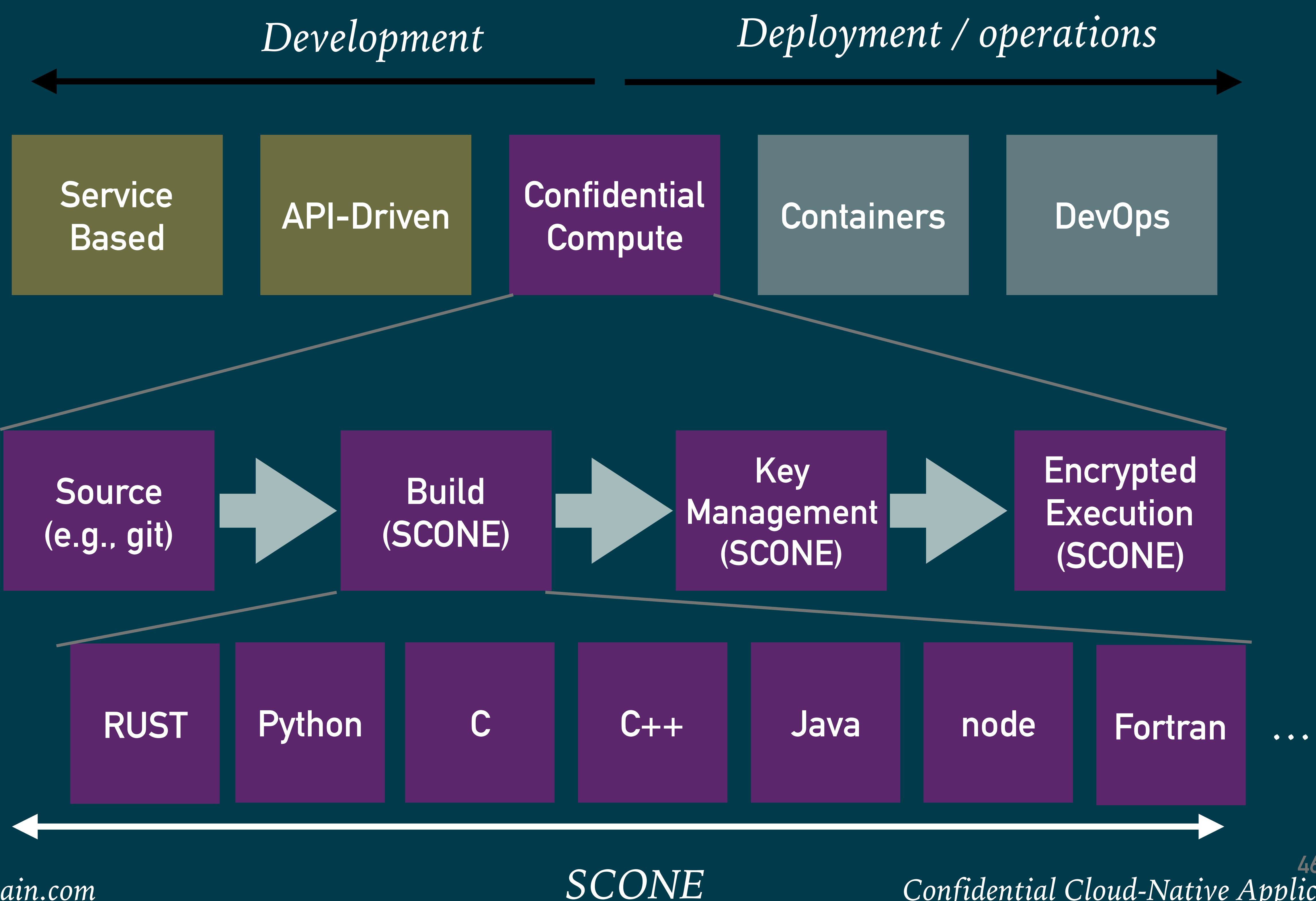


CONFIDENTIAL CLOUD-NATIVE APPLICATIONS



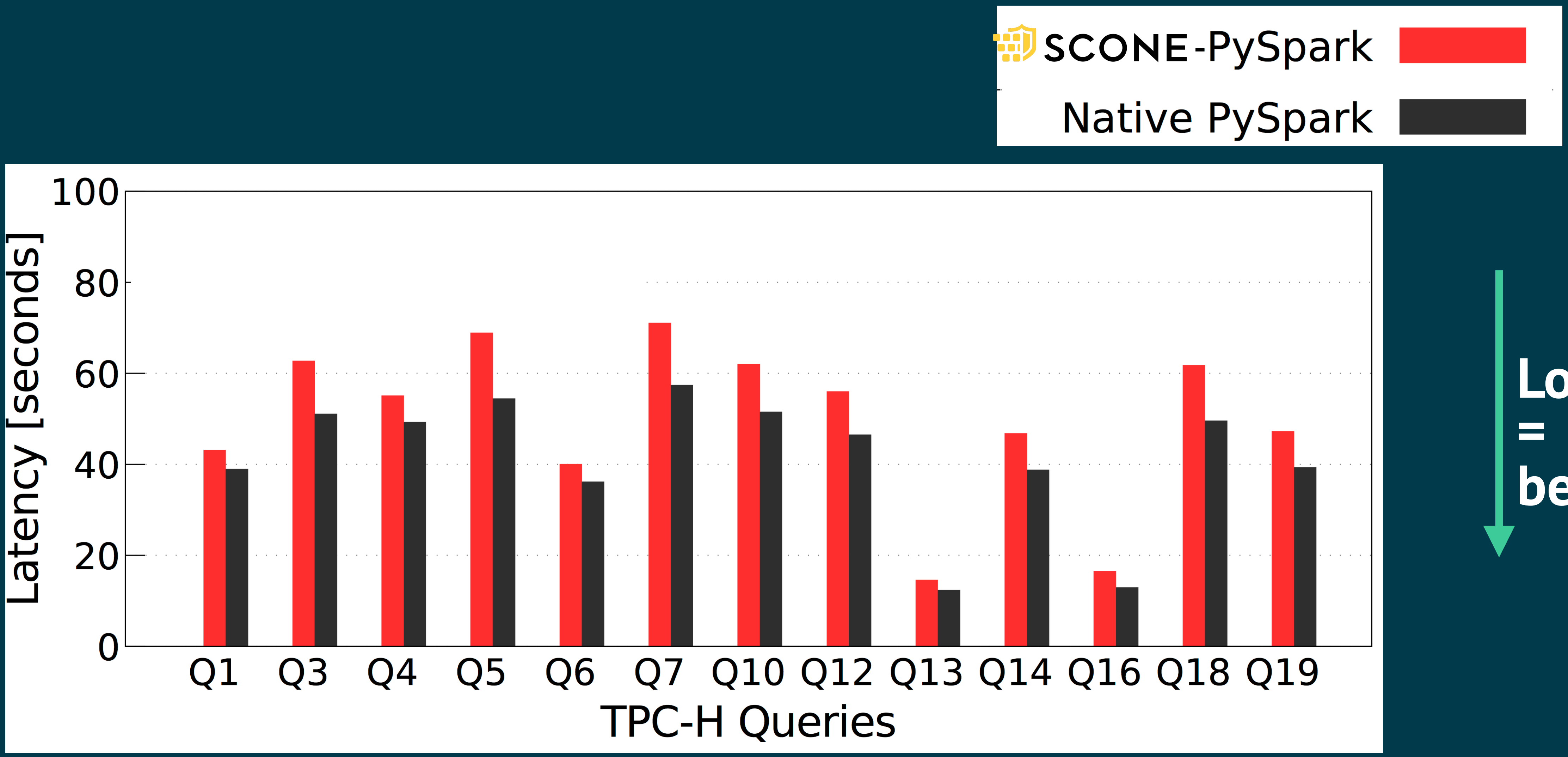
christof.fetzer@scontain.com

CONFIDENTIAL CLOUD-NATIVE APPLICATIONS



PERFORMANCE?

PERFORMANCE



Lower = better

< 22 % overhead compared to native execution

USE CASES?

EHEALTH

- we have built the Electronic Patient Record service
    - a **confidential cloud-native application** on top of Kubernetes
- highly scalable and efficient:
    - We scaled to 6000 parallel confidential microservices in a single Kubernetes cluster

christof.fetzer@scontain.com

AI FRAMEWORKS

- SCONE supports
  - TensorFlow
  - TensorFlow Lite
  - PyTorch
  - OpenVino
  - Scikit-Learn
  - …

SCONE

CONCLUSION

SCONE supports the development & operation of **confidential cloud-native applications**

SCONE

christof.fetzer@scontain.com

https://scontain.com
https://sconedocs.github.io/