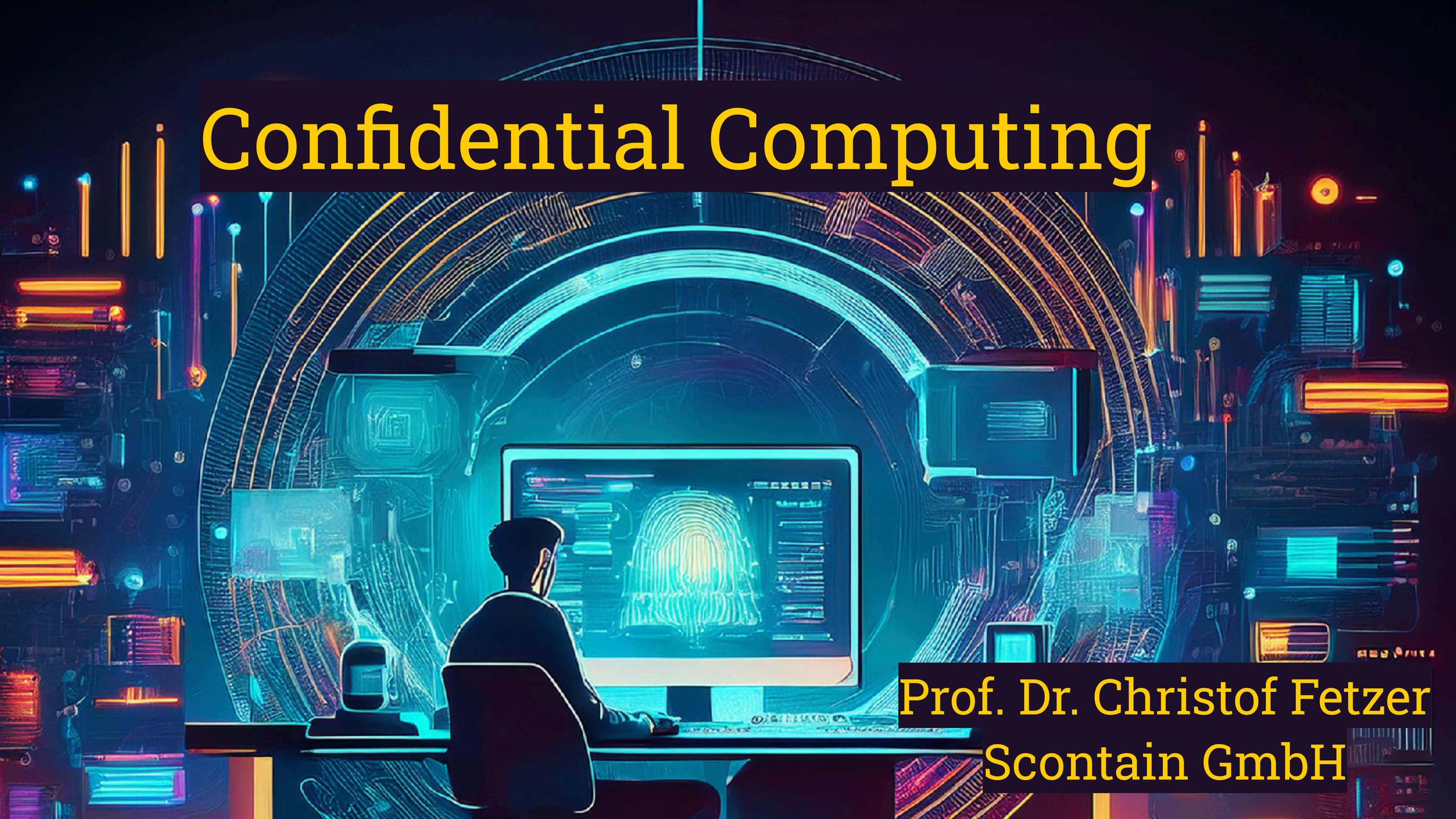
Confidential Computing



Prof. Dr. Christof Fetzer
Scontain GmbH

Confidential Computing

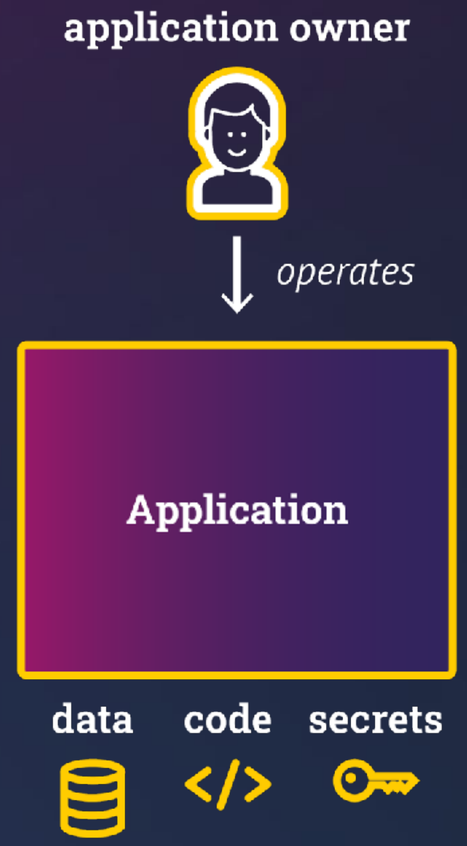
- Motivation -

Motivation

- **Role**: application owner

Objectives:

- provides an application to clients
- protects **data**, **code**, and **secrets** (e.g., keys) of the application



A **role** defines set of responsibilities, duties, and expectations for a particular position within a development team or organization

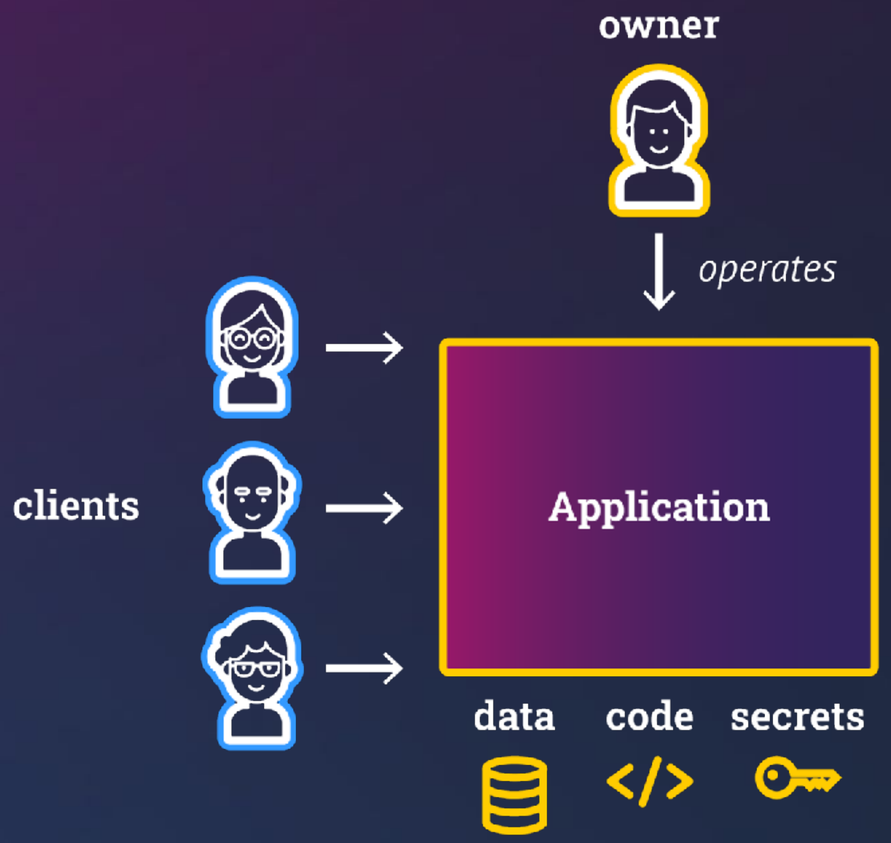
Motivation

- **Role: application owner**

Objectives:

  - provides an application to clients
  - protects **data, code, and secrets** of the application

- **Role: clients**
  - can connect to the application
  - access their data



Requirements

- Example: eHealth domain -

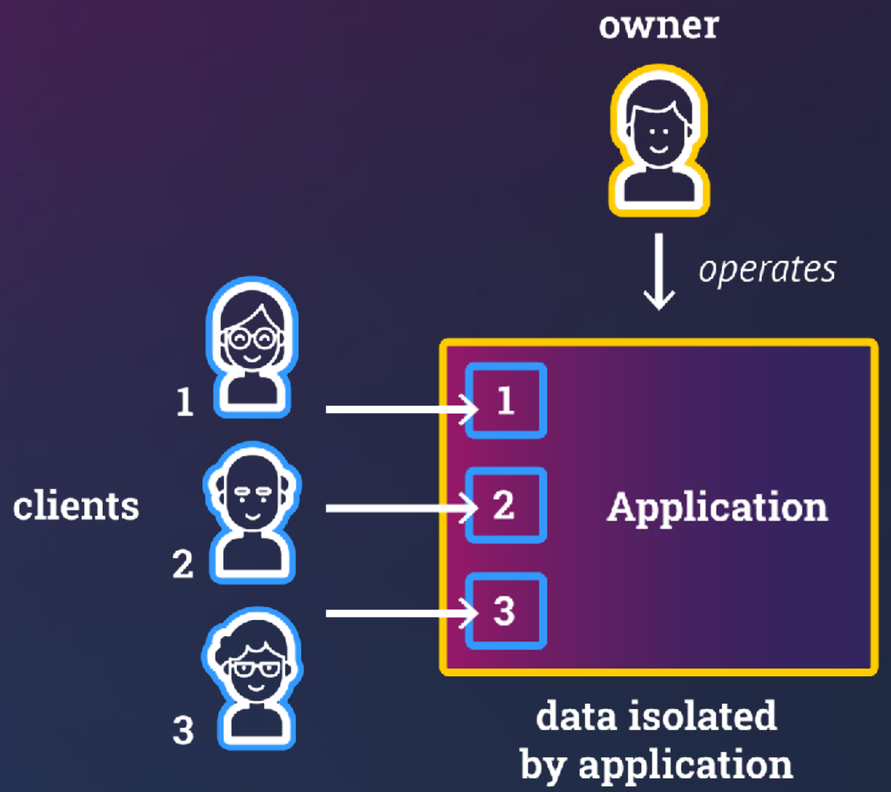
Example Requirement: Isolation of data

- **Role: application owner**

Objectives:

  - provide an application to clients
  - protect **data, code, and secrets** of the application

- **Role:** clients
  - can connect to the application
  - access their data
- **Requirement**:
  - **application isolates data of clients**



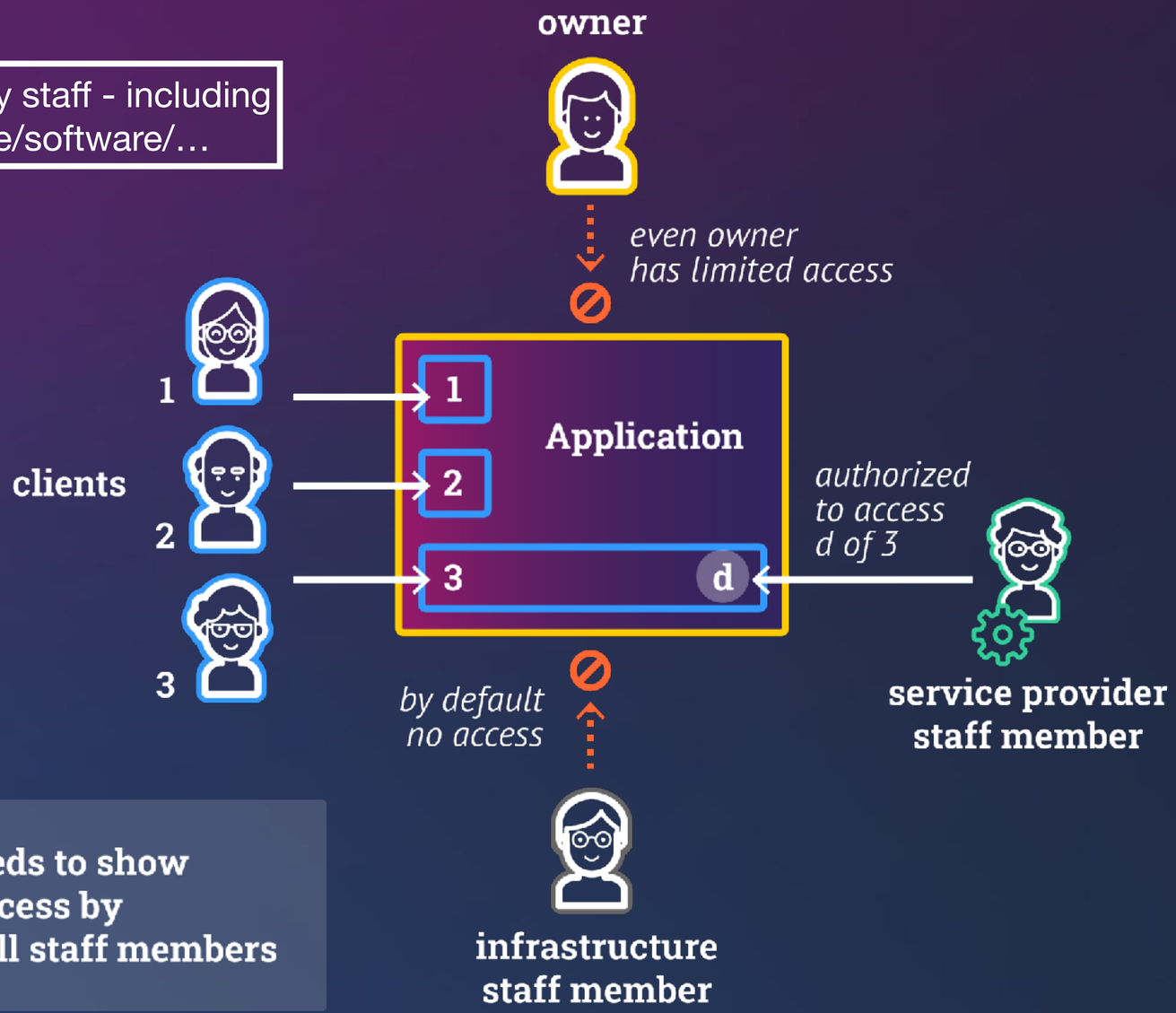
An **isolation** of client data implies that a client A cannot access the data of any other client B. This must be guaranteed (in some cases) even if the application is buggy.

Definition: **Isolation**

- (**Data**) **isolation** is primarily the process of segregating data sets to prevent **unauthorized access** or **leakage**. It involves separating sensitive information to ensure it is accessed only for authorized purposes, thereby reducing the risk of unauthorized access.

Limit Access by Owner & Staff

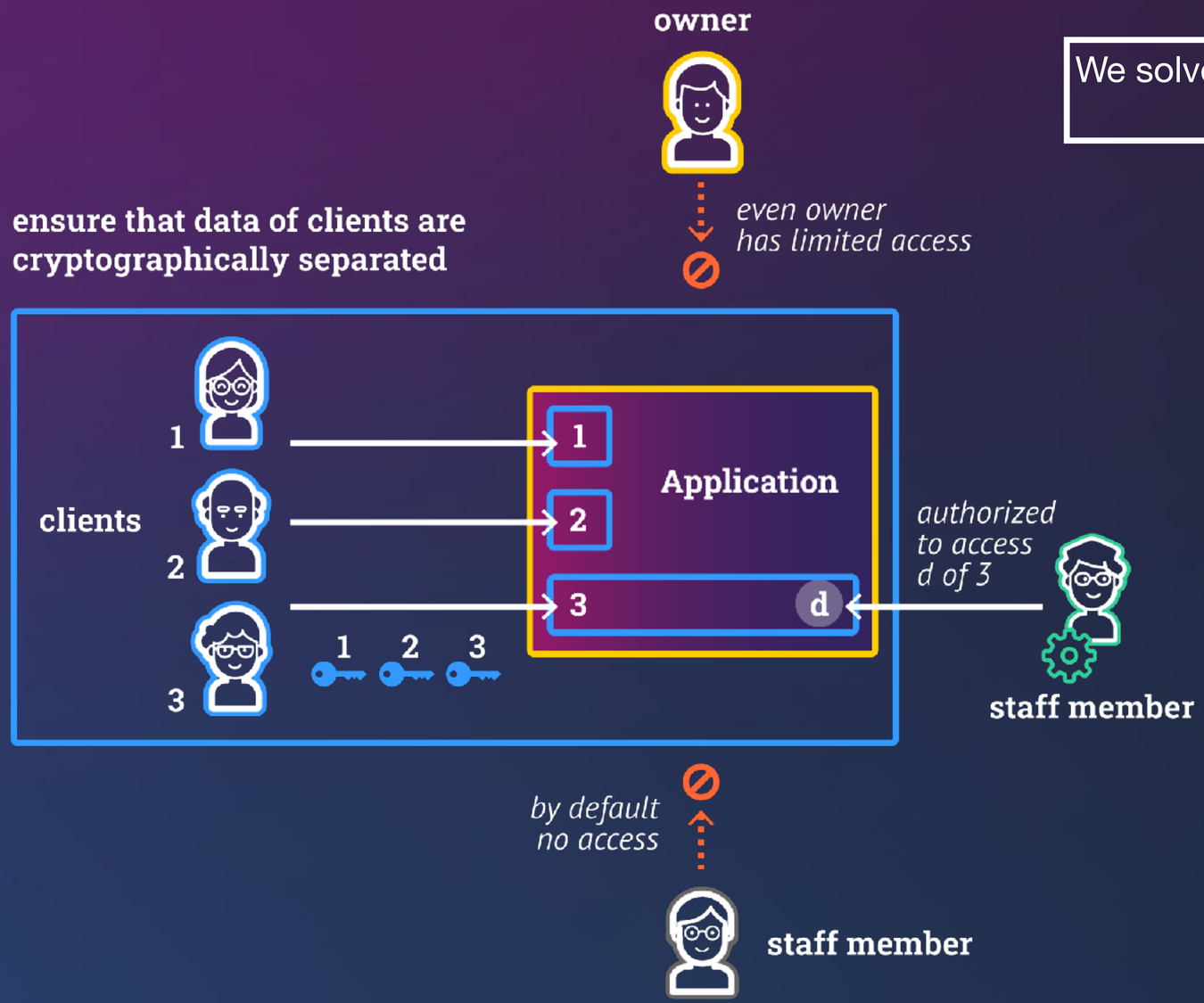
Requirement: Limit access by staff - including administrators of hardware/software/…



owner needs to show limited access by owner & all staff members

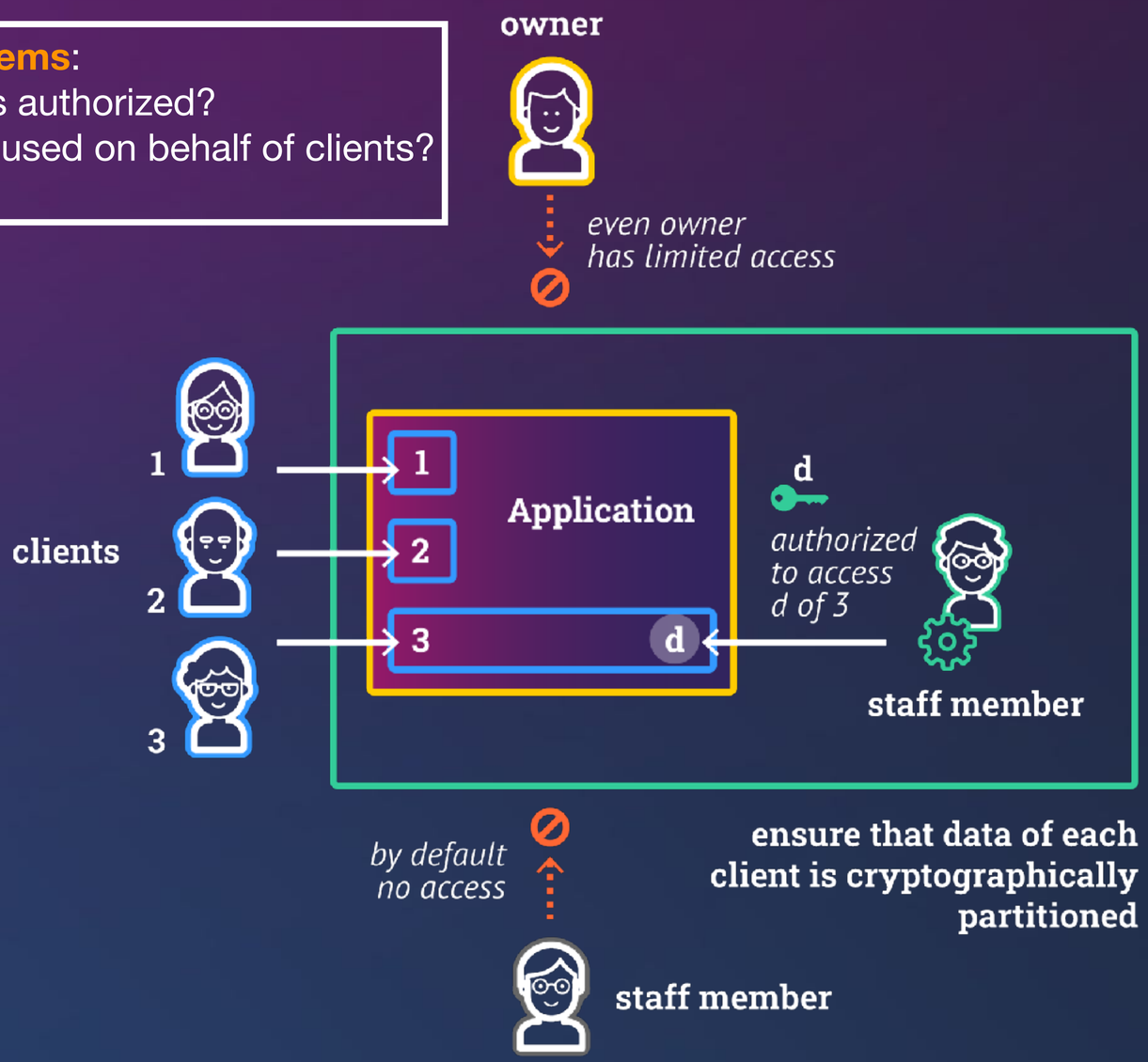
Approach: Divide and Conquer

We solve the problem by splitting in smaller problems

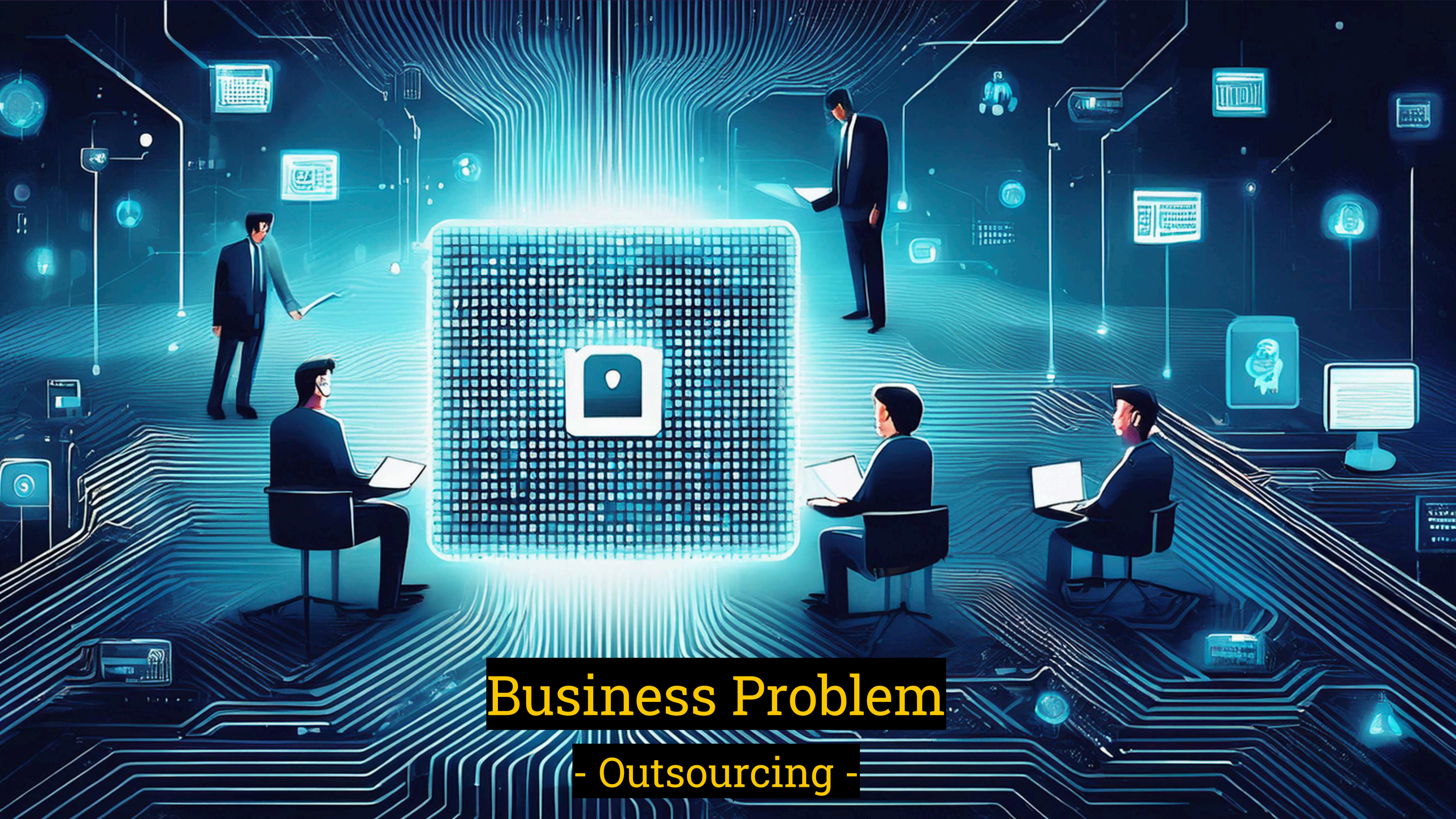


Divide and Conquer

Problems:
1. How to determine who is authorized?
2. How to protect the keys used on behalf of clients?



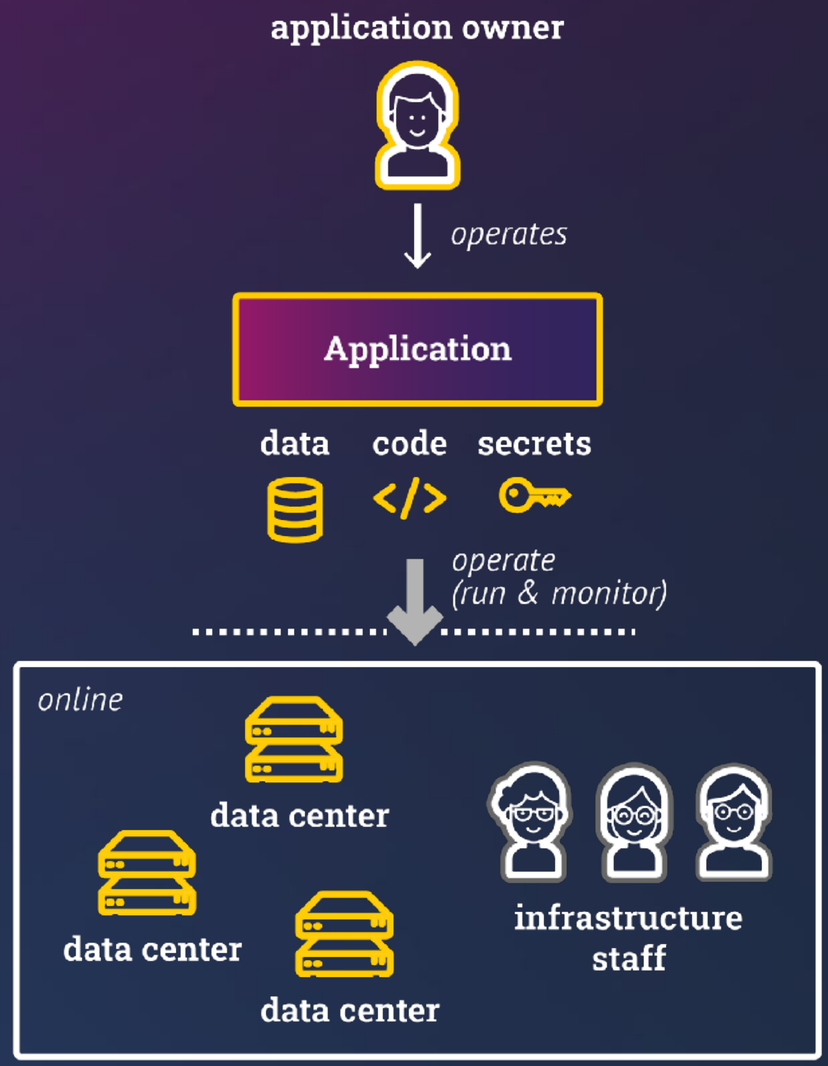
Client data is further split


Business Problem
- Outsourcing -

Problem Description

Problem: application owner cannot operate the application

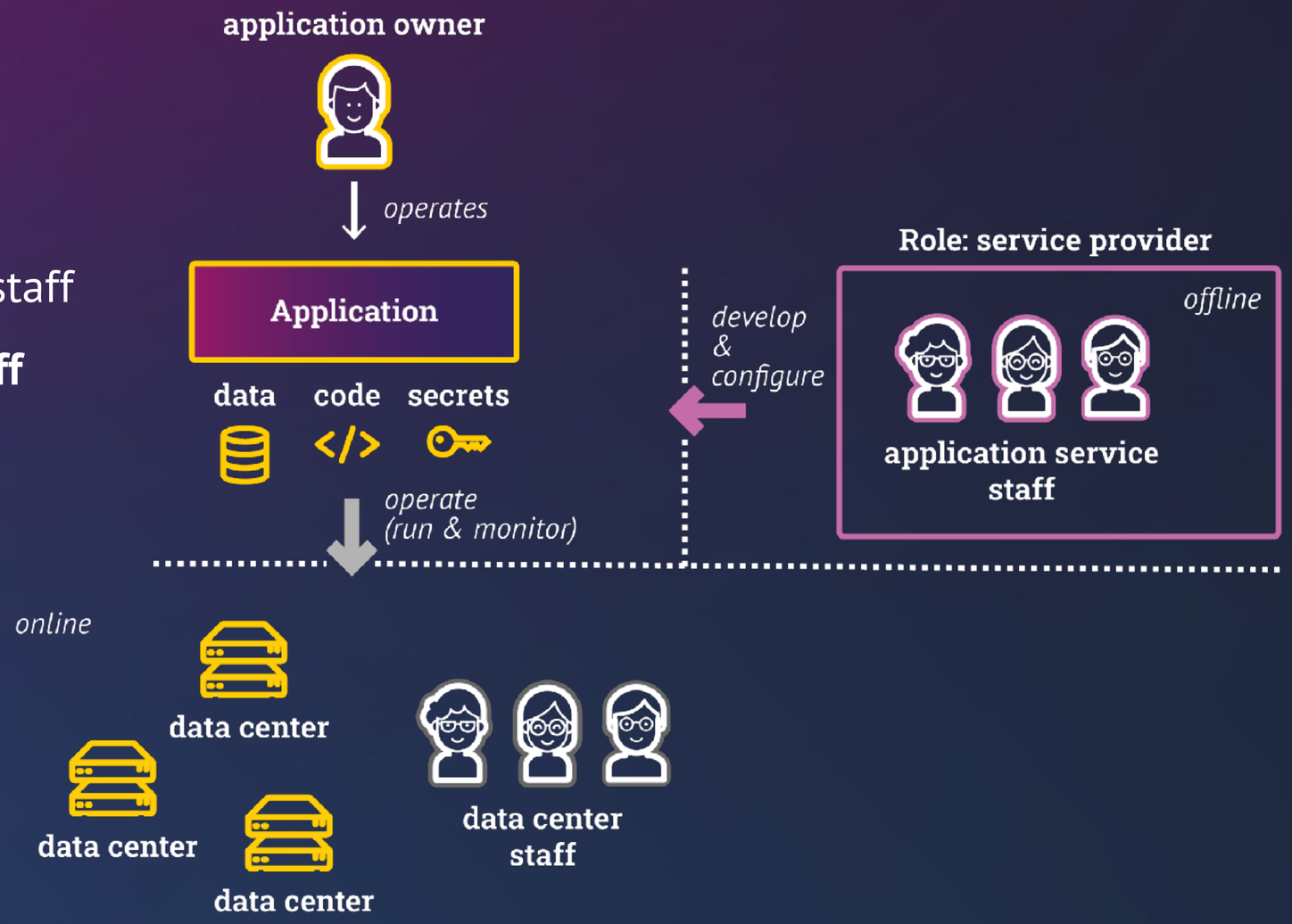
- **lack of data centers**
- **lack of trusted infrastructure staff**
- lack of application service staff



Problem Description

Problem: application owner cannot operate the application

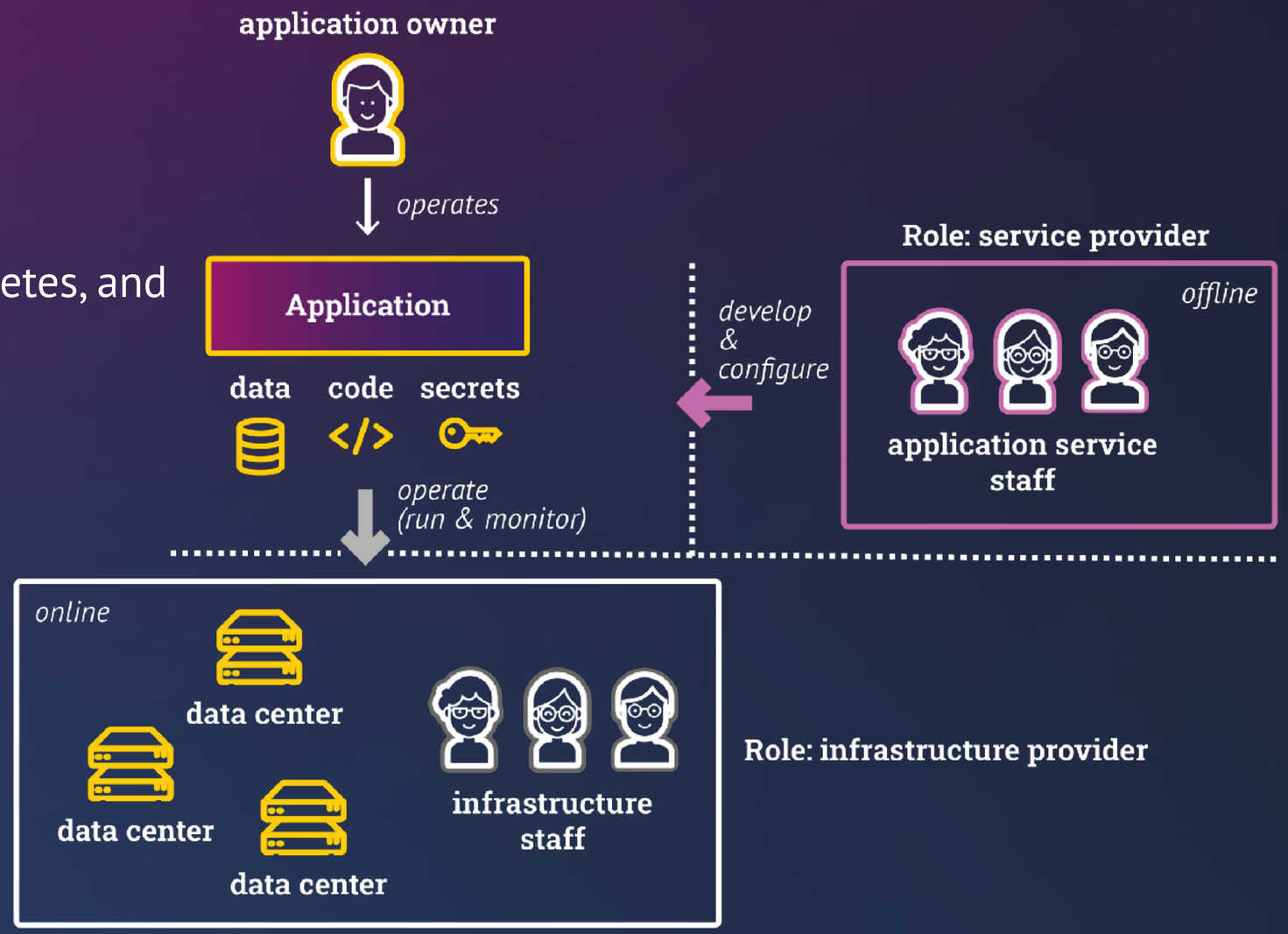
- lack of data centers
- lack of trusted infrastructure staff
- **lack of application service staff**



Approach: Outsource!

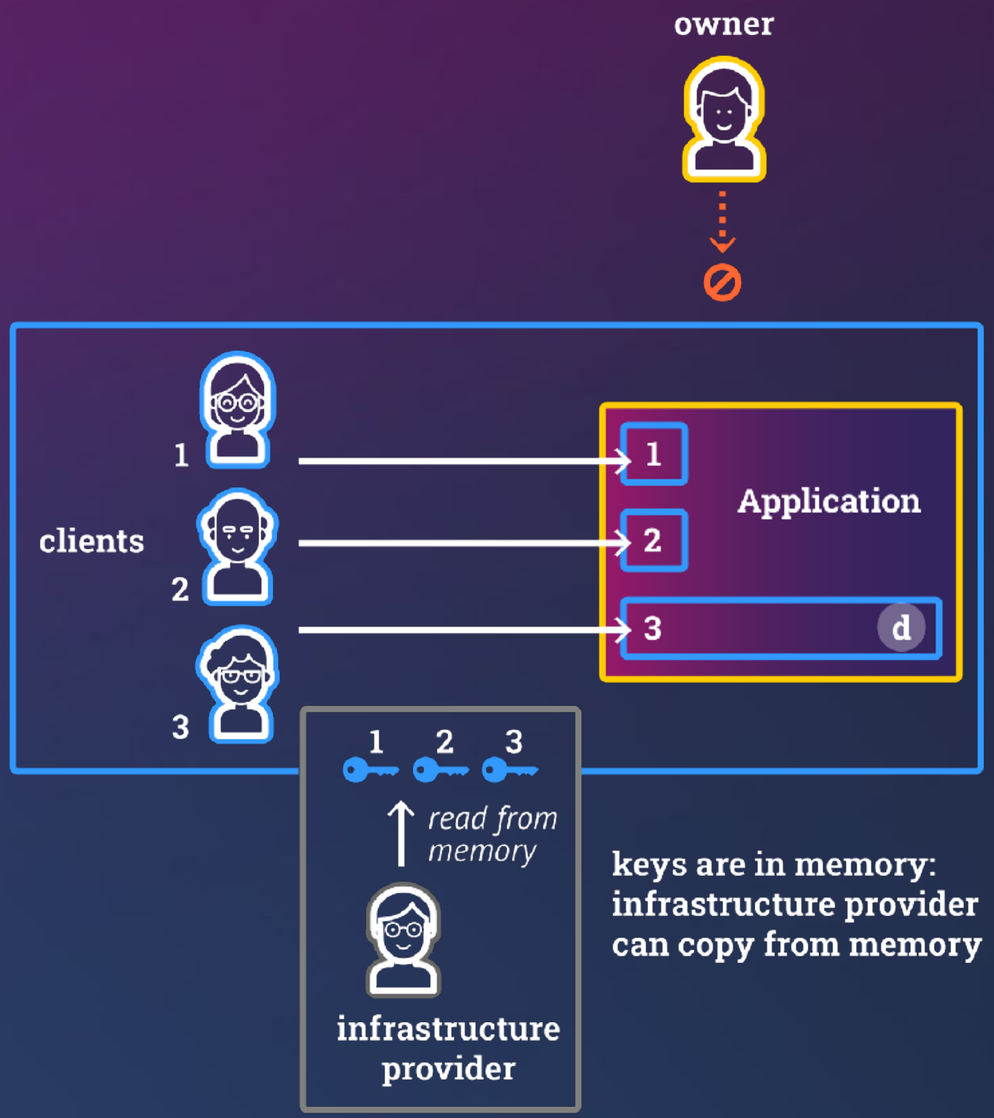
Approach: external entities

- operate data centers, Kubernetes, and
- manage application development

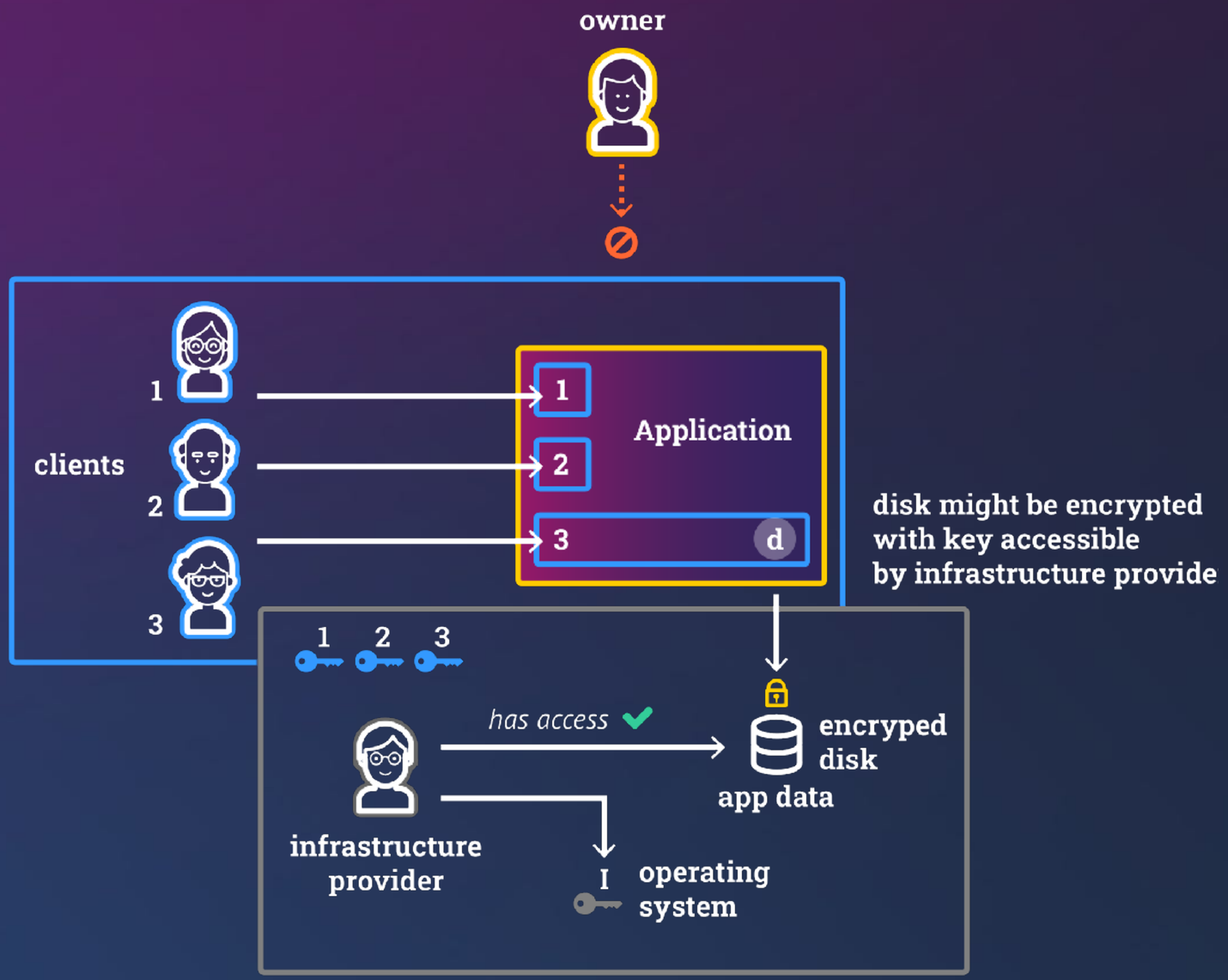


Technical Problem Description

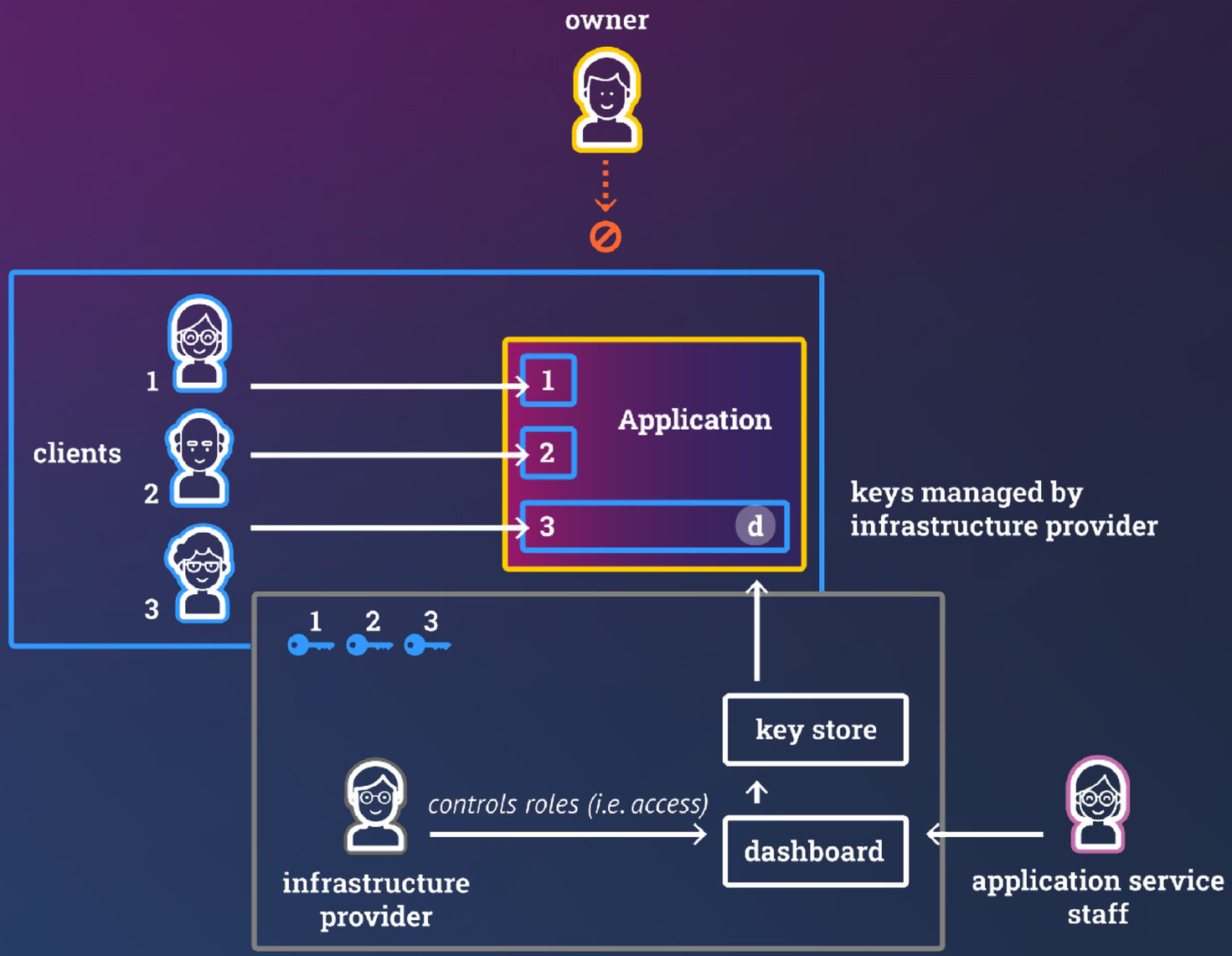
Problem: Hardware & Admin Access



Problem: Encrypted Disks



Problem: Key Management



Threat Model & Implications

- we might trust the CPU -

Who to trust?

owner

infrastructure quickly evolving, application owner cannot vouch for security

Implication:
We need to ensure no access to source code, data or any keys

infrastructure provider

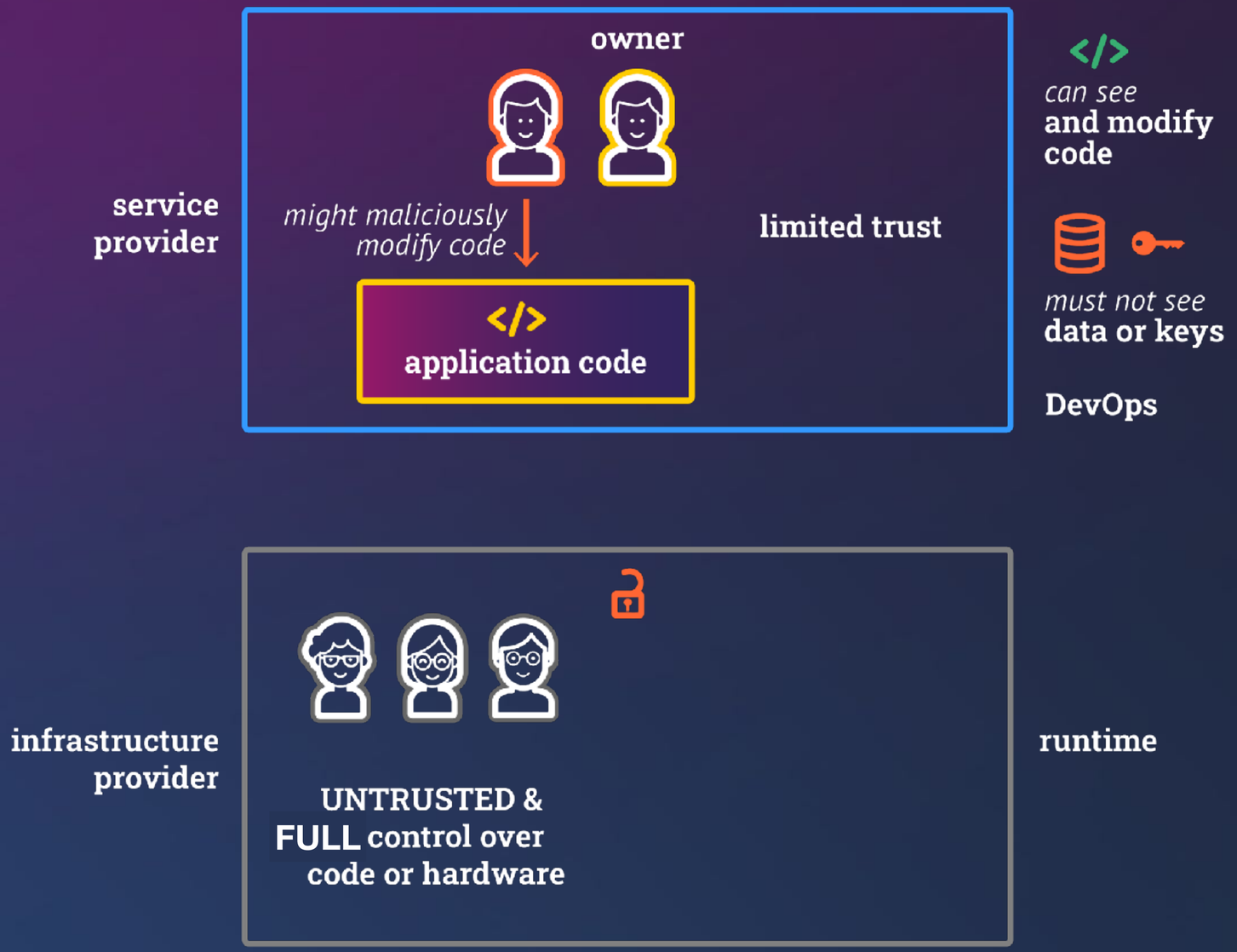
UNTRUSTED & **FULL** control over code or hardware

managed hypervisor, operating system, Kubernetes, key store, access control, ...staff members

are ALL UNTRUSTED

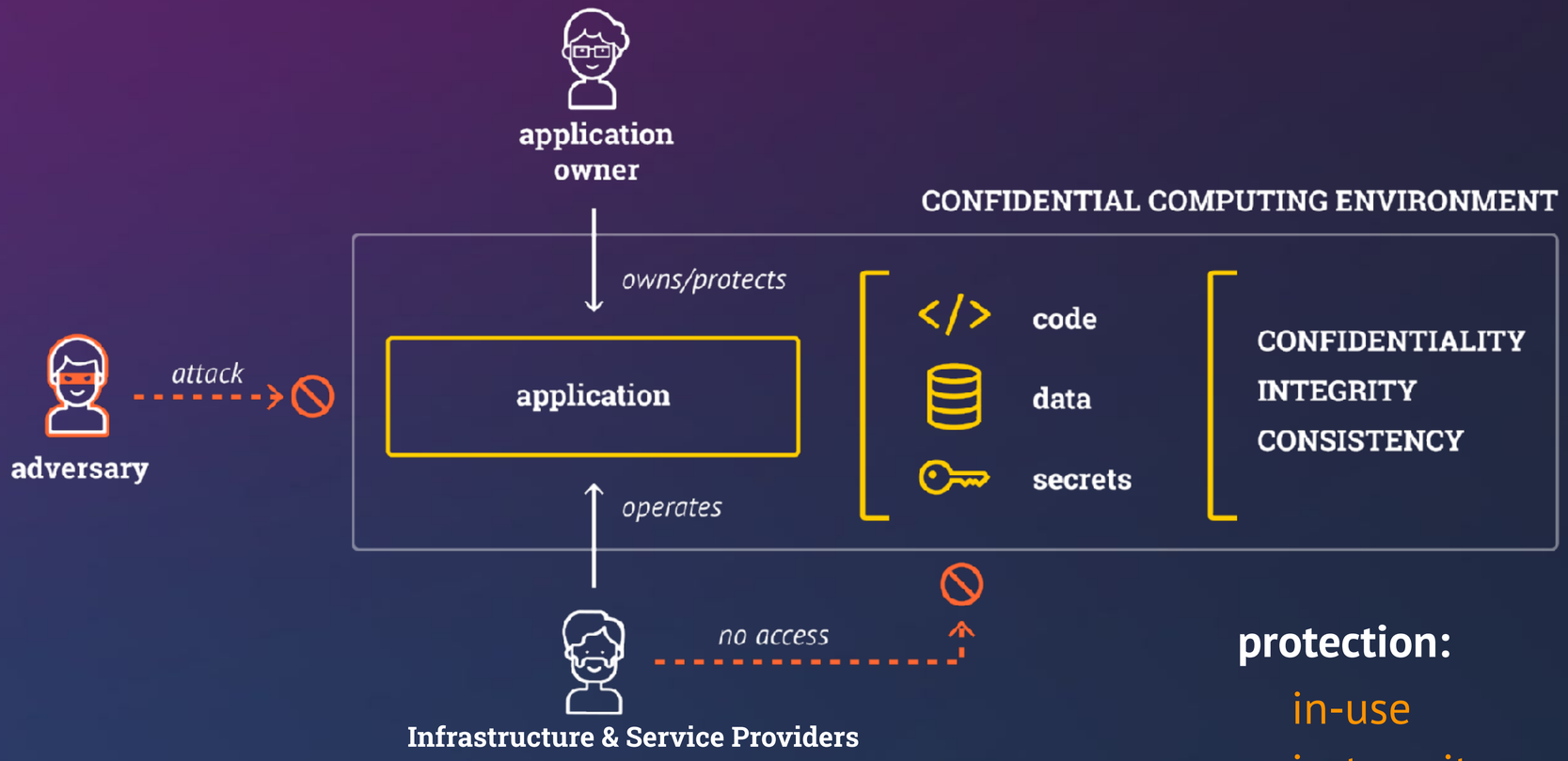
runtime

Threat Model




Approach

Approach: Confidential Computing



protection:

in-use

in-transit

at rest

„application-oriented security“