

# DevOpsCon

by  devmio

**Zero Trust**  
**The hard way?**



# Who we are



## Daniel Pötzing

Daniel Pötzing has many years of experience in the development and architecture of Enterprise Web Applications. He has worked with many great self-organized agile teams and knows how collaboration and mutual inspiration – together with the right technologies and patterns – makes software projects successful and solving challenges fun.

In love with DDD since 2008



## Kevin Schu

Kevin Schu is a DevOps enthusiast and Infrastructure Automation Specialist. Currently, he is Director for Cloud and DevOps Consulting at AOE in Wiesbaden, supporting customers moving into the cloud, helping them sharpen their cloud strategy as well as building high performing DevOps organizations.



**150 mio. customer data leaked**  
**700\$ mio. compensation fee**

**Do you remember the “Equifax Hack”?**



## Avoidable!

A widely known vulnerability in Apache Struts

A missing network segmentation

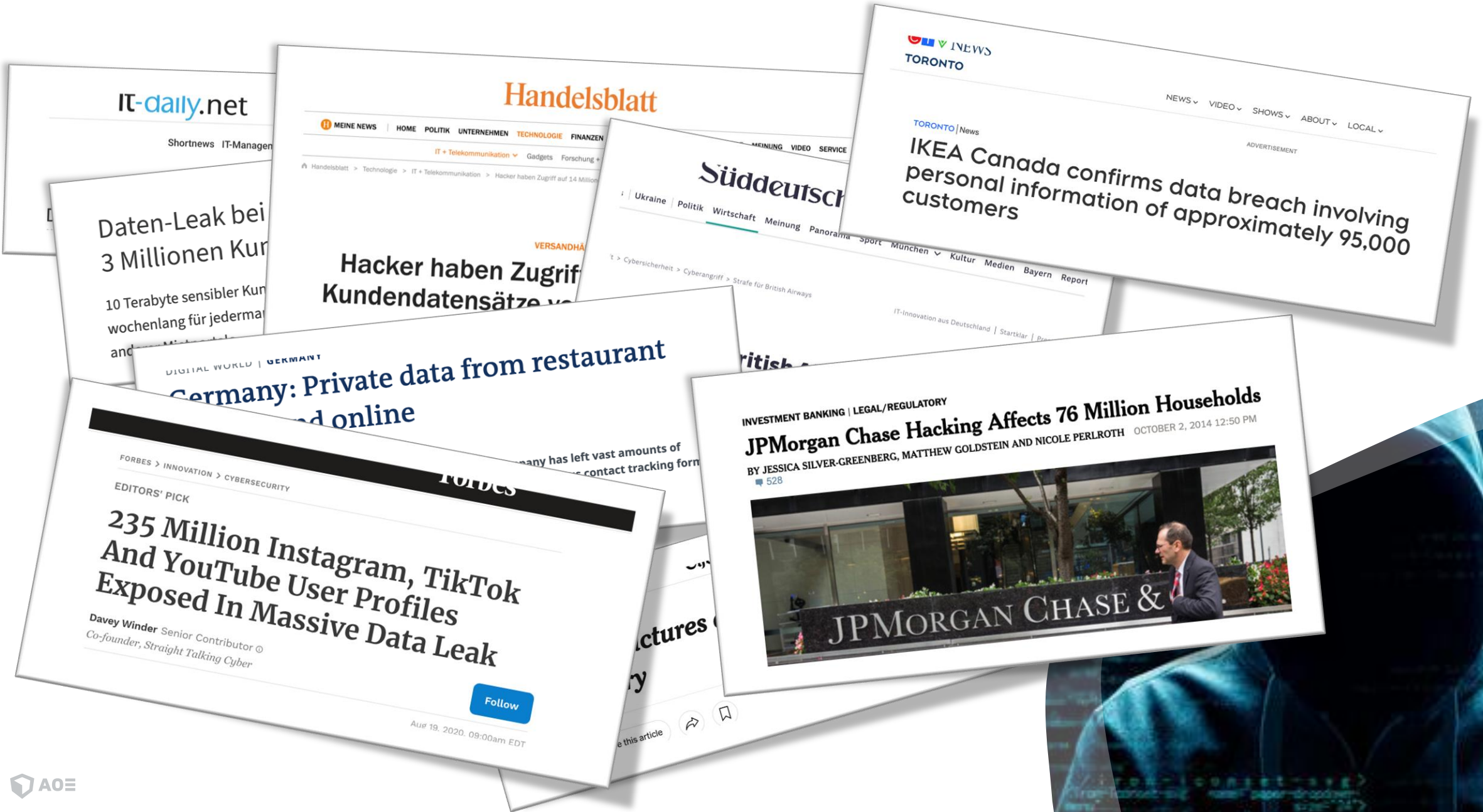
Unencrypted personal credentials on network shares

Unencrypted data

A broken intrusion detection



# Not the only one



IT-daily.net

Shortnews IT-Manager

Daten-Leak bei  
3 Millionen Kur  
10 Terabyte sensibler Kur  
wochenlang für jederma  
and

Handelsblatt

MEINE NEWS | HOME POLITIK UNTERNEHMEN TECHNOLOGIE FINANZEN

IT + Telekommunikation | Gadgets Forschung +

Handelsblatt > Technologie > IT + Telekommunikation > Hacker haben Zugriff auf 1,4 Millionen

Hacker haben Zugriff  
Kundendatensätze

Süddeutsche

Ukraine | Politik | Wirtschaft | Meinung | Panorama | Sport | München | Kultur | Medien | Bayern | Report

NEWS  
TORONTO

NEWS | VIDEO | SHOWS | ABOUT | LOCAL

TORONTO | News  
ADVERTISEMENT  
IKEA Canada confirms data breach involving  
personal information of approximately 95,000  
customers

DIGITAL WORLD | GERMANY  
Germany: Private data from restaurant  
and online

FORBES > INNOVATION > CYBERSECURITY

EDITORS' PICK

235 Million Instagram, TikTok  
And YouTube User Profiles  
Exposed In Massive Data Leak

Davey Winder Senior Contributor @  
Co-founder, Straight Talking Cyber

Follow

Aug 19, 2020, 09:00am EDT

INVESTMENT BANKING | LEGAL/REGULATORY

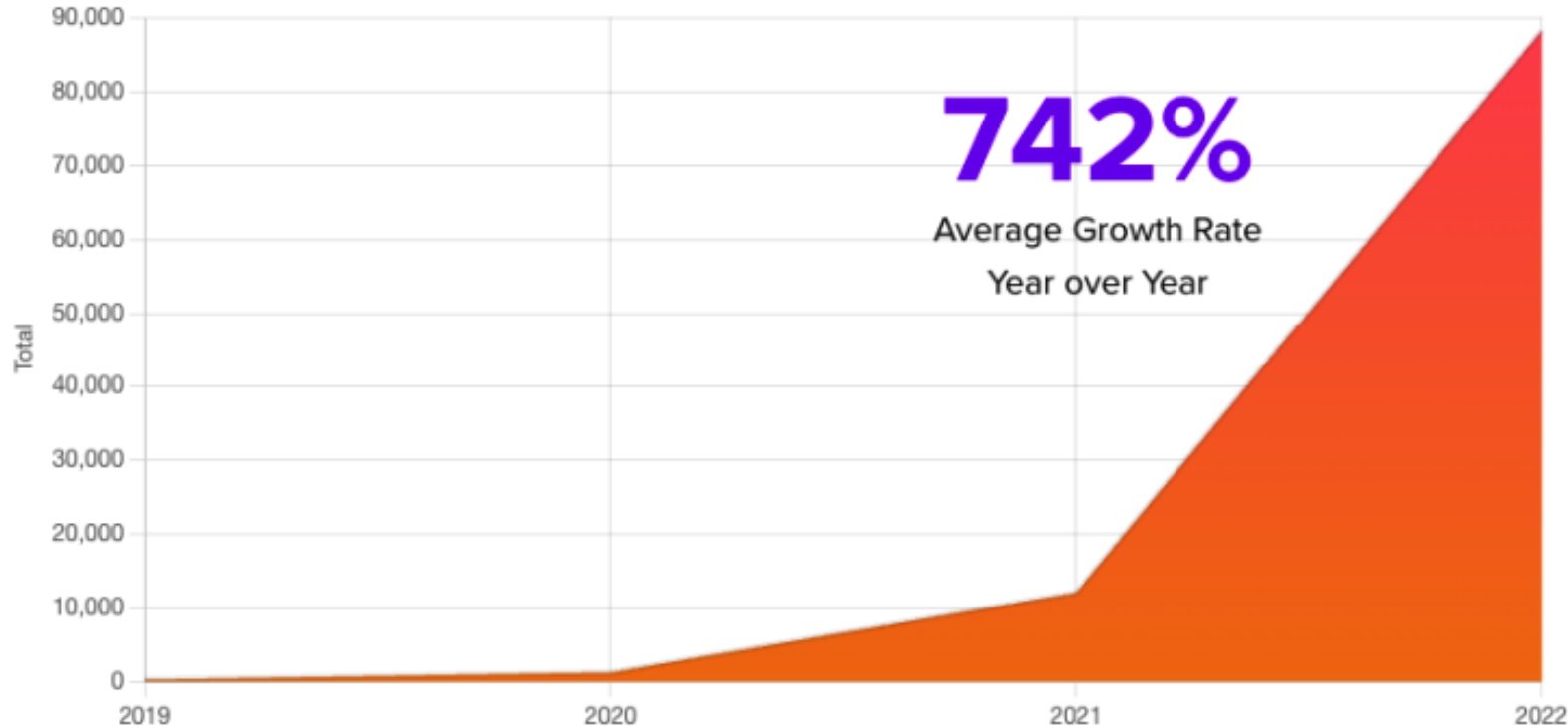
JPMorgan Chase Hacking Affects 76 Million Households  
BY JESSICA SILVER-GREENBERG, MATTHEW GOLDSTEIN AND NICOLE PERLROTH OCTOBER 2, 2014 12:50 PM



# Many statistics show a dangerous trend

## Hacking is a business model and more and more automated

Amount of supply chain attacks (like log4J)



<https://blog.sonatype.com/2023-predictions-software-supply-chain-governance>



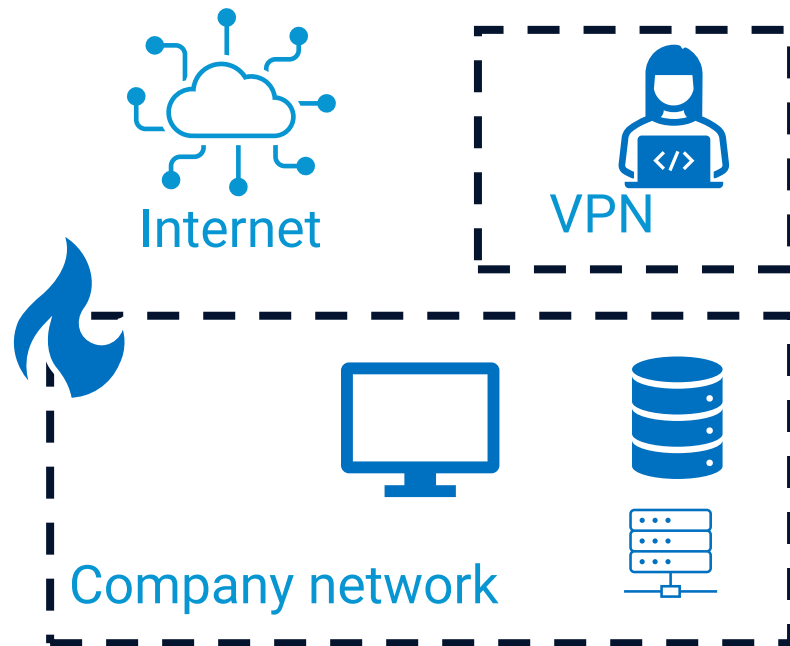




**Security is based on trust**



# Trust used to be based on network perimeters (location)



- High risk of breaches (once you are in you are in)
- User, devices, data and applications are moving “out”

A new security thinking is required  
**Zero Trust: Never trust, always verify!**



least privilege



assume breach



strong identity verification



verify explicitly

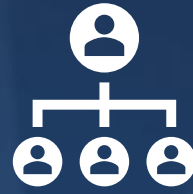


Increase security & reduce risk



Work and access from anywhere / anytime

# Zero Trust Aspects



Organization and Culture



Secure Development and Delivery



Security Monitoring & Automation

Identities & Identity Awareness

Device & Device Authentication

Networking & Firewall

Application Security

Infrastructure Security

Secure Data Handling







**Why is it hard?**





**53%** security teams say it's harder to keep up with security requirements

report ongoing **88%** talent challenges

**64%** frustratingly changing from one security tool to the next





## Security Leader vs. Non-Leaders\*

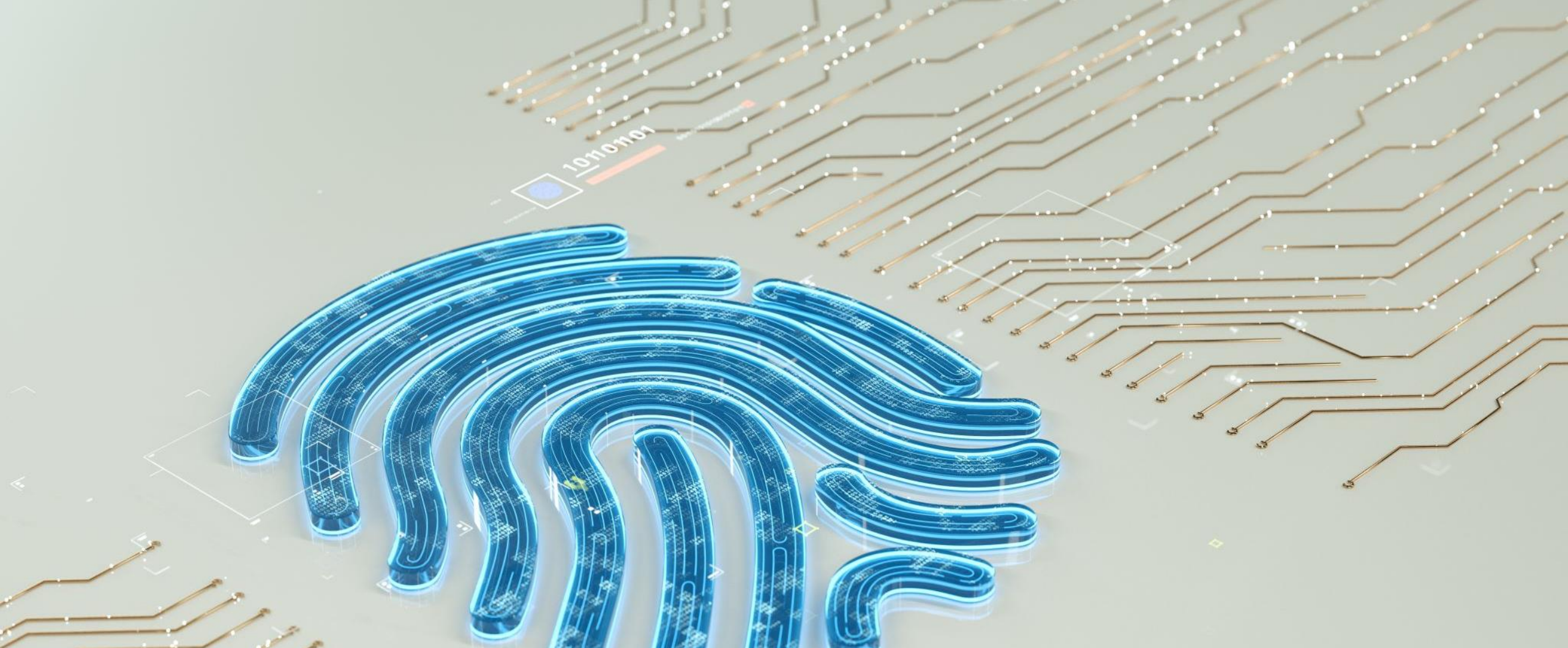
**4x**

more effective at blocking attacks

more likely to detect breaches

**3x**

better at closing breaches



**Identity & Identity Awareness**  
**Strong identity, strong security**



# Identity

A unique entity (person, device, application) that can be authenticated and authorized to access resources.

## AuthN = Authentication

Prove identity – are you who you claim to be?

## AuthZ = Authorization

Are you allowed to do what you want to do?



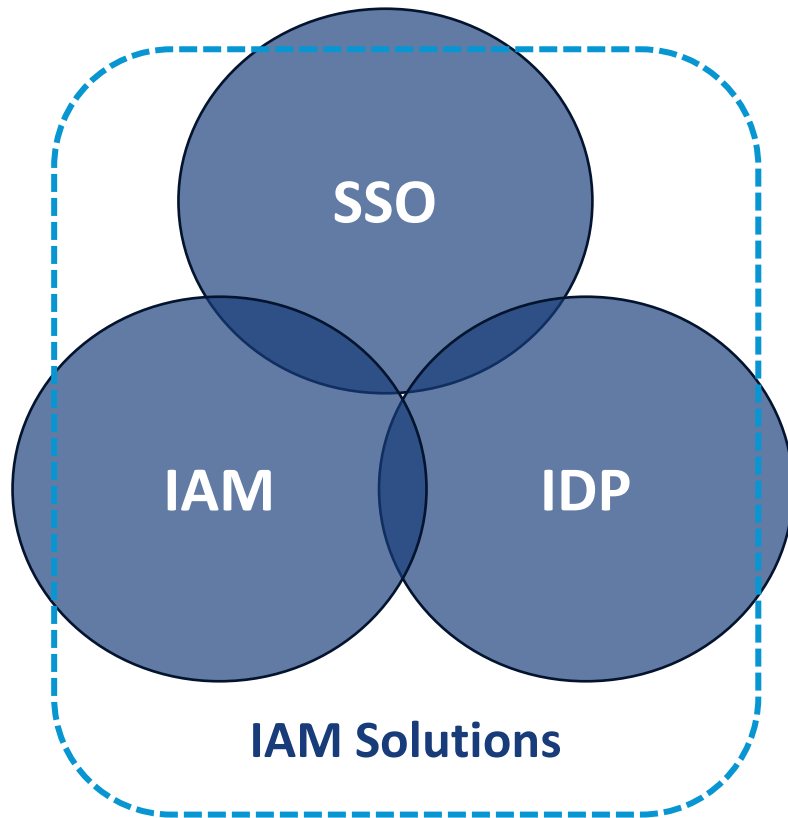


## SSO is key

- On average, employees have to switch between ten apps every hour



## IAM – A wider perspective



- Seamless user experience
- Centralized source of identities, roles and groups
- Auditability
- Policy Enforcements (password strength, session lifetimes, ...)
- Ability to block a compromised identity

SSO: Single Sign On

IDP: Identity Provider

IAM: Identity and Access Management



### BIG 3

- Google IAM
- AWS IAM Identity Center
- Azure AD

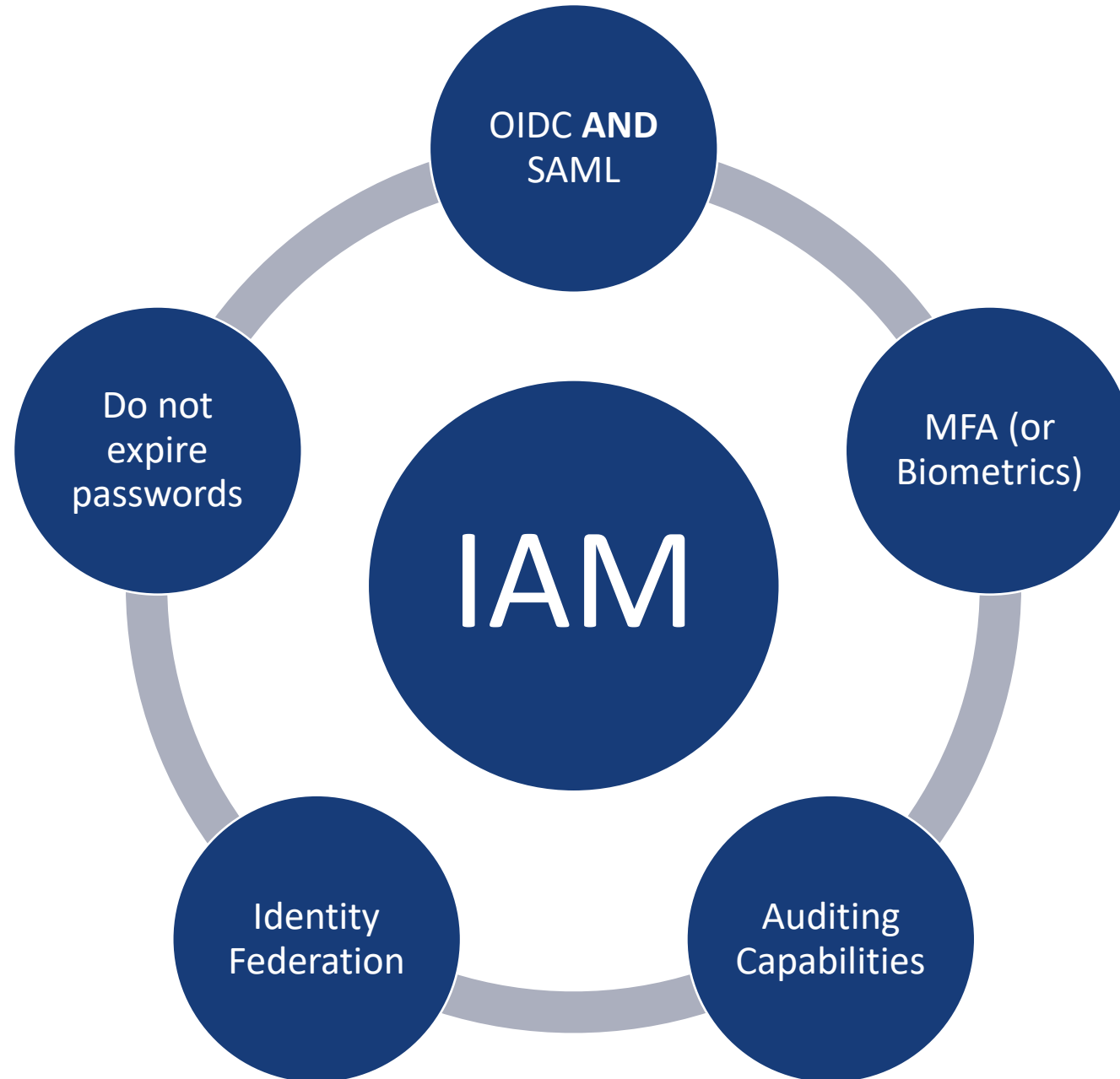
### Open Source

- KeyCloak
- FreeIPA

### Cloud Solutions

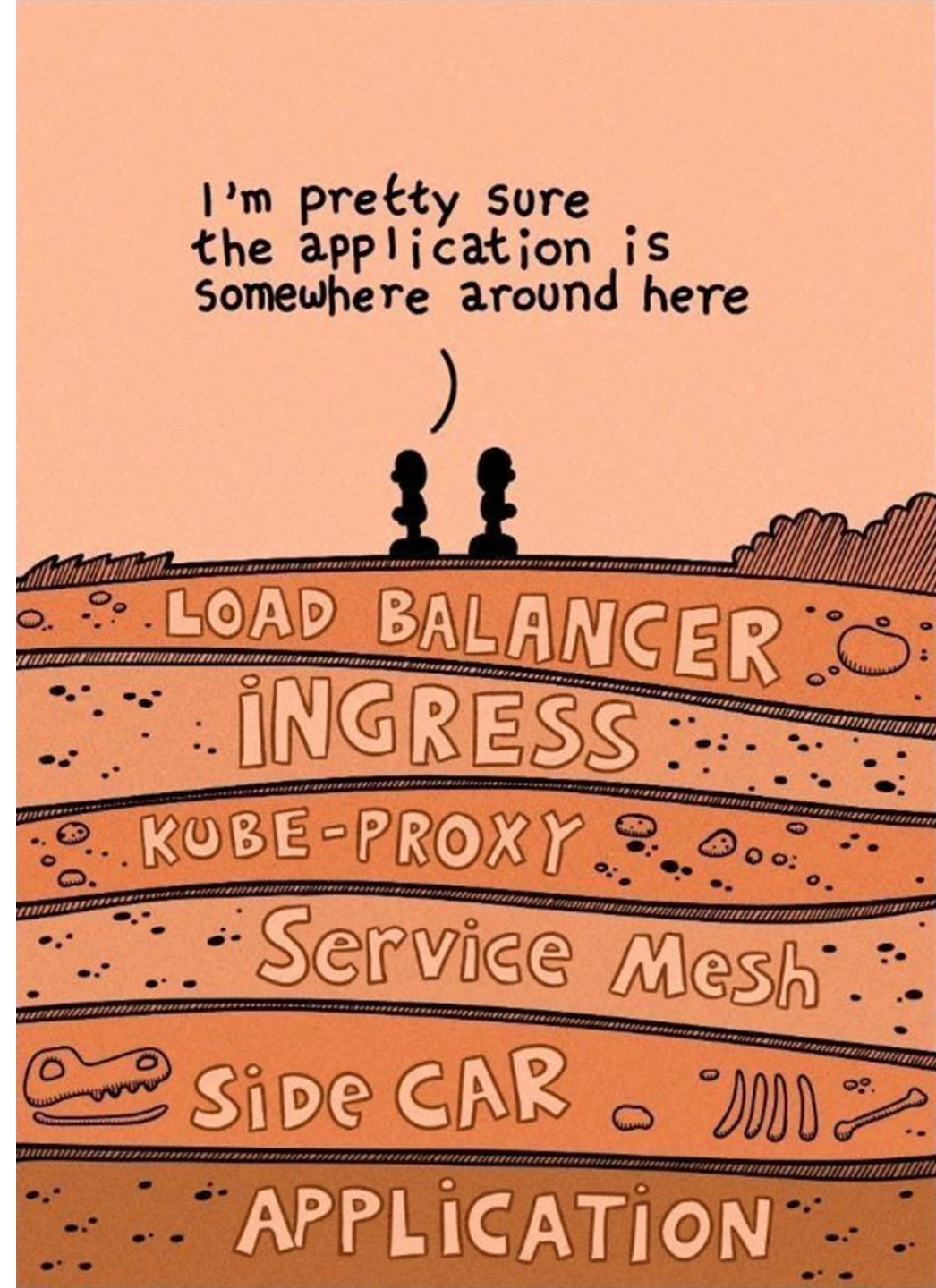
- Okta
- Ping Identity
- Bare.ID



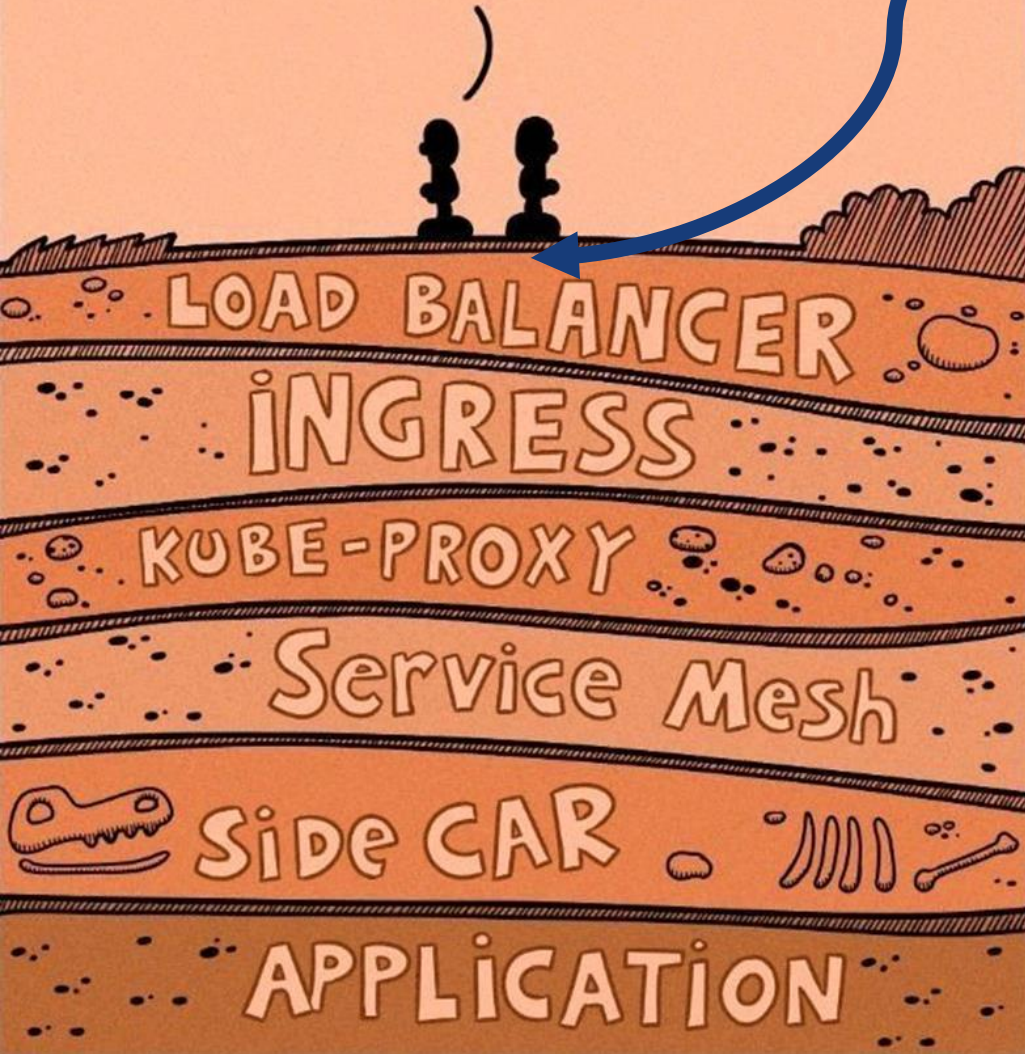


# Identity Awareness

Identity Aware Proxy (IAP) enhances identity awareness and strengthens security controls.



I'm pretty sure  
the application is  
somewhere around here



## Identity Aware Proxy - IAP

- At the very beginning of your request chain
- L7 HTTP Proxy – Software Defined Perimeter
- Enforces authentication for all incoming requests
- Integrates seamlessly into IAM solutions using common protocols like OIDC or SAML
- Ideally already enforces high level policy-based authorization
- Also enables authentication for applications that do not support SSO natively
- First authentication, not the last





## BIG 3

- Google IAP
- AWS Verified Access
- Azure Conditional Access

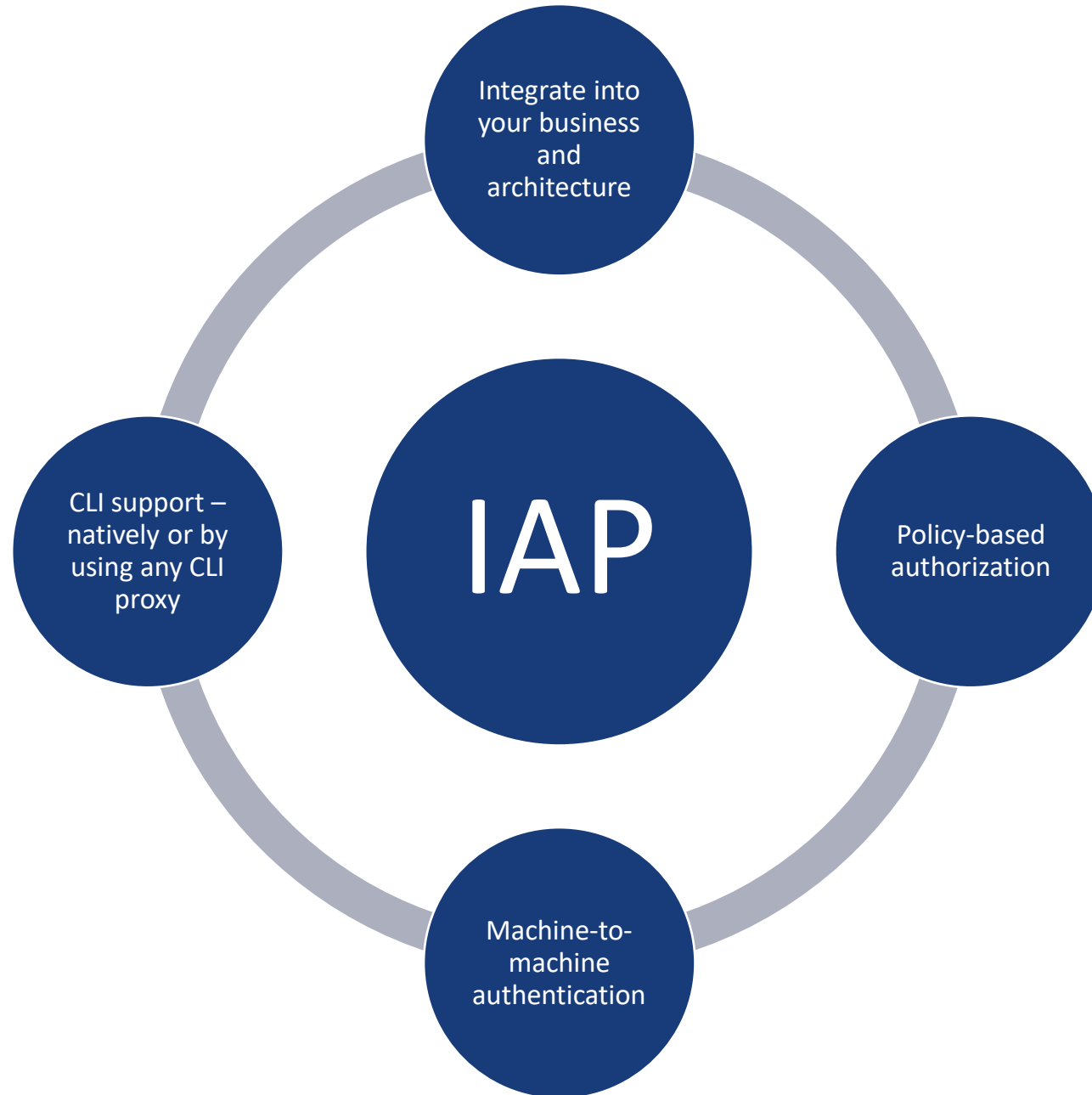
## Open Source

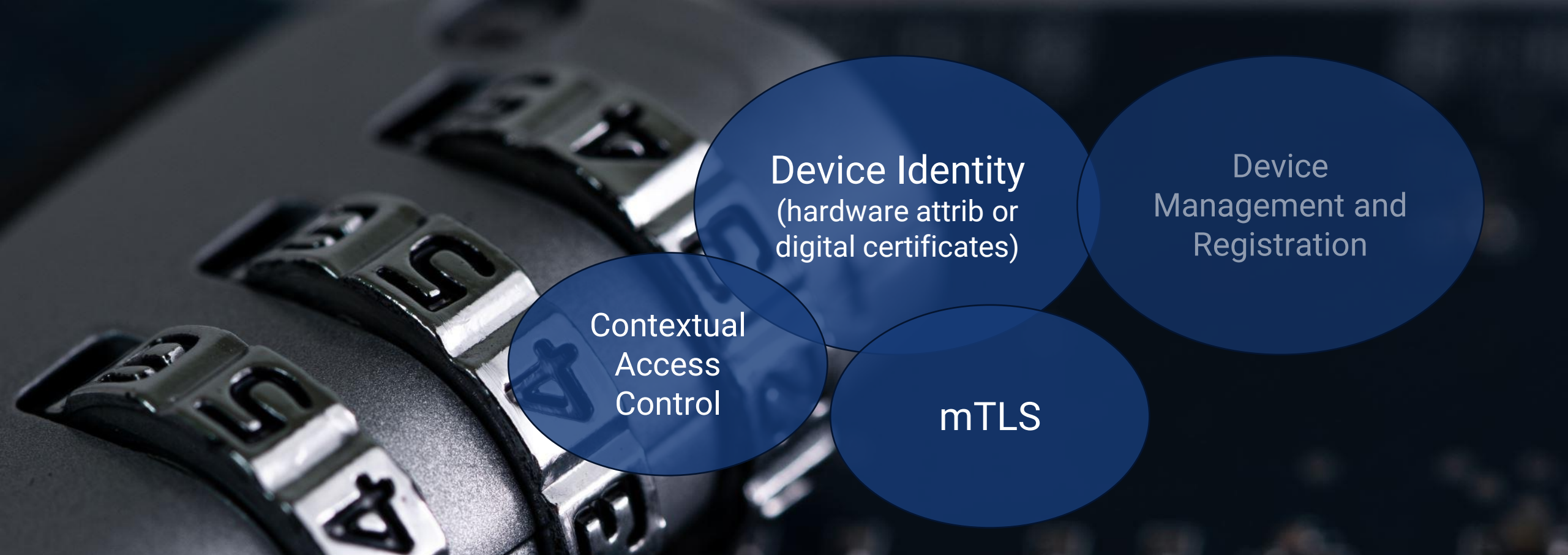
- Pomerium
- Teleport
- Hashicorp Boundary
- (oauth2-proxy)

## Cloud Solutions

- Strongdm
- Okta Workforce



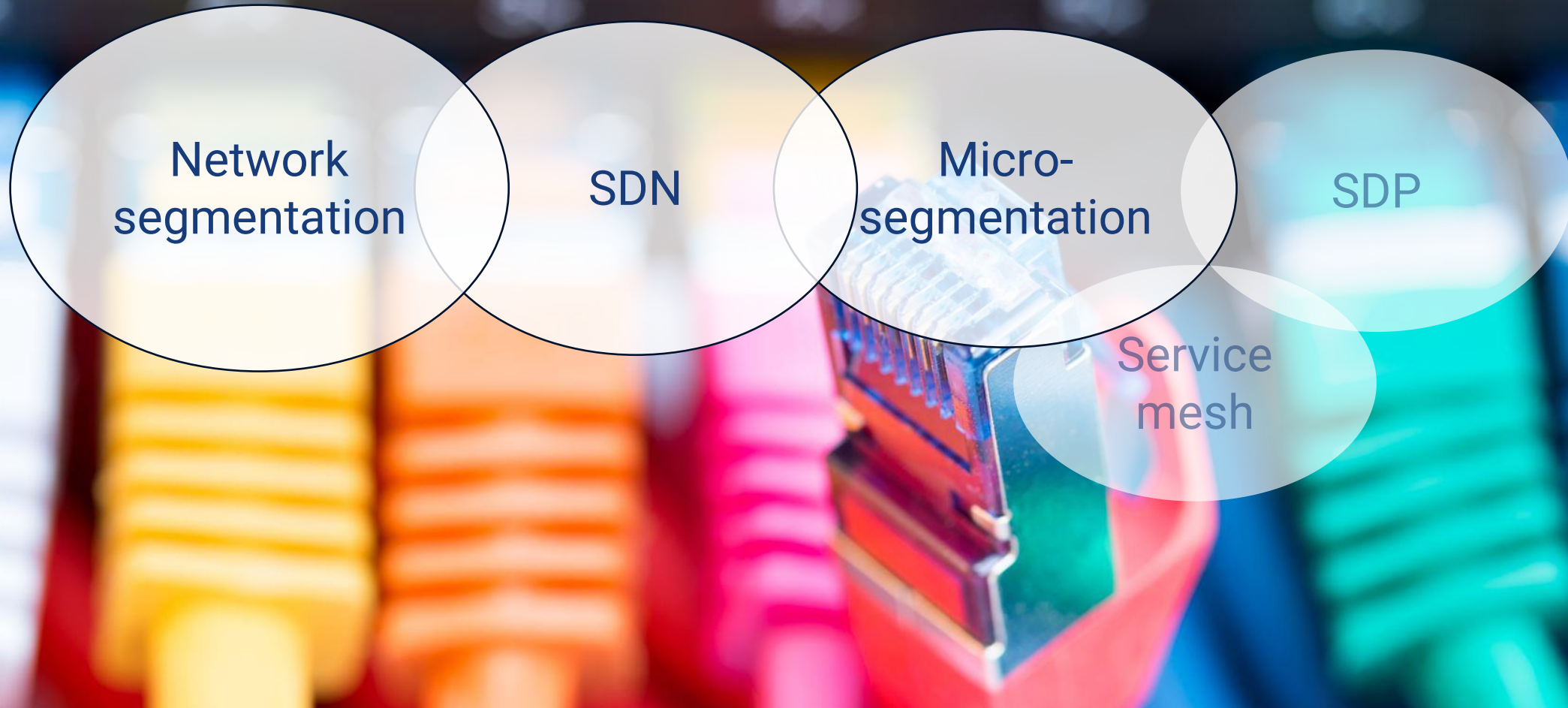




# Device Authentication

**Fortifying Access:**

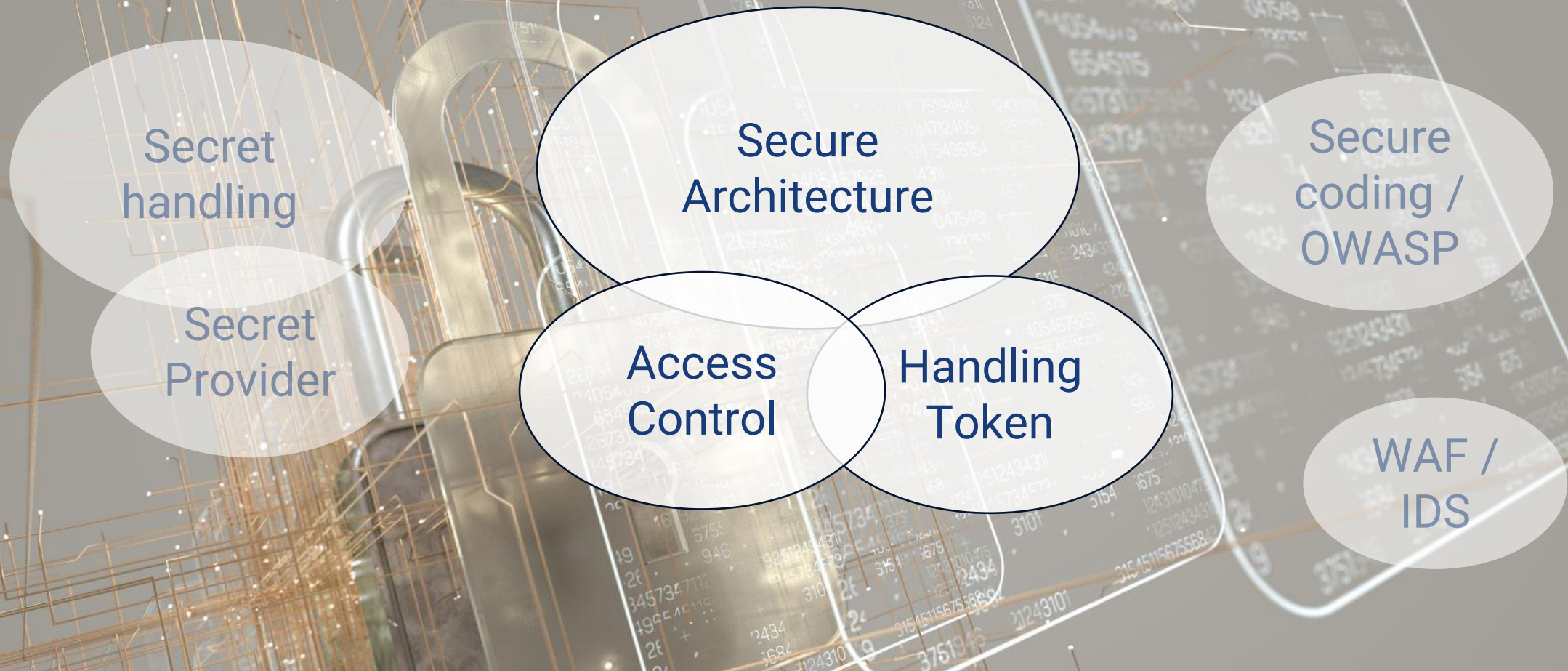
**Device Authentication as first line of defense**



## **Networking & Firewall**

**Boundaries are essential, microsegmentation is powerful**



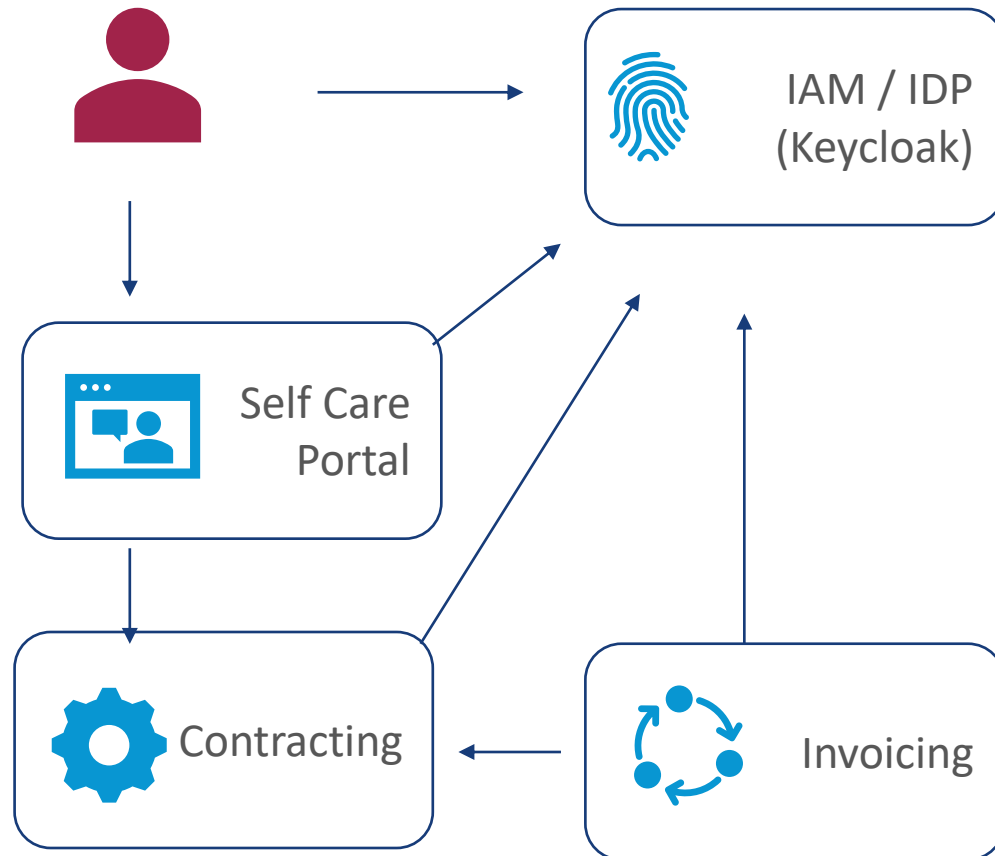


# Application Security

Code strong, shield stronger: From access doubts to certainty



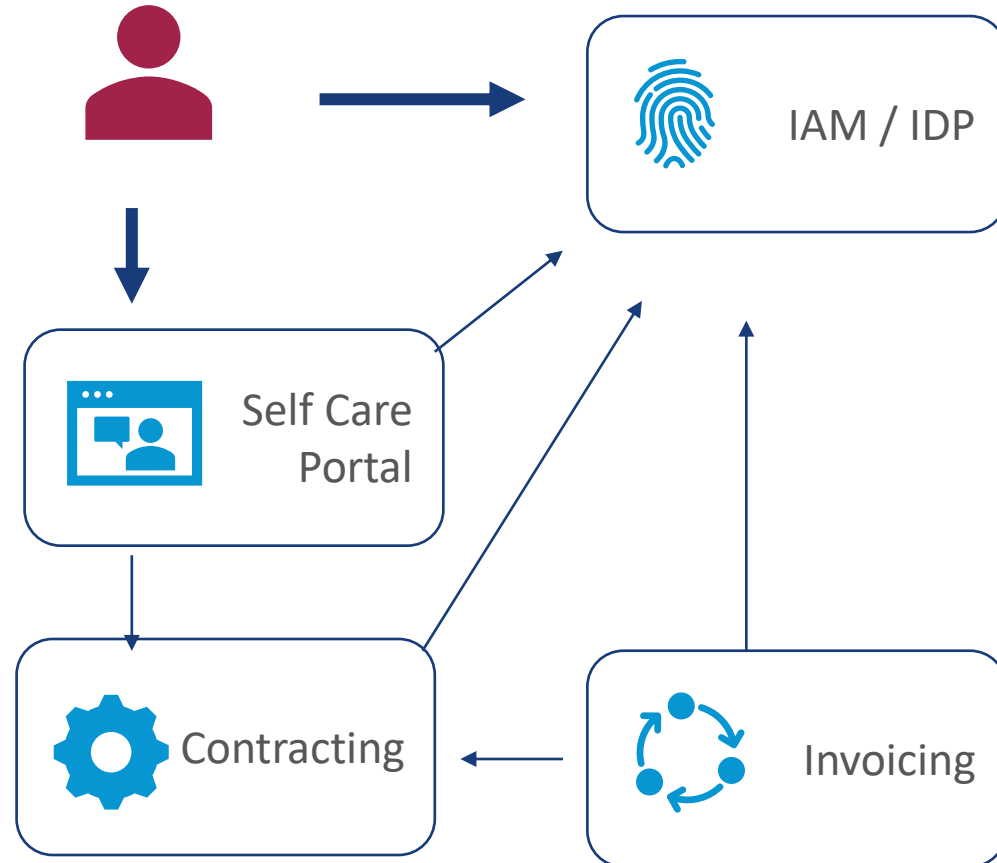
## Security by design – Secure software architecture



Some example enterprise solution - of course designed as distributed software architecture with useful bounded contexts.

All involved apps are configured as own „client“ in the IAM solution.

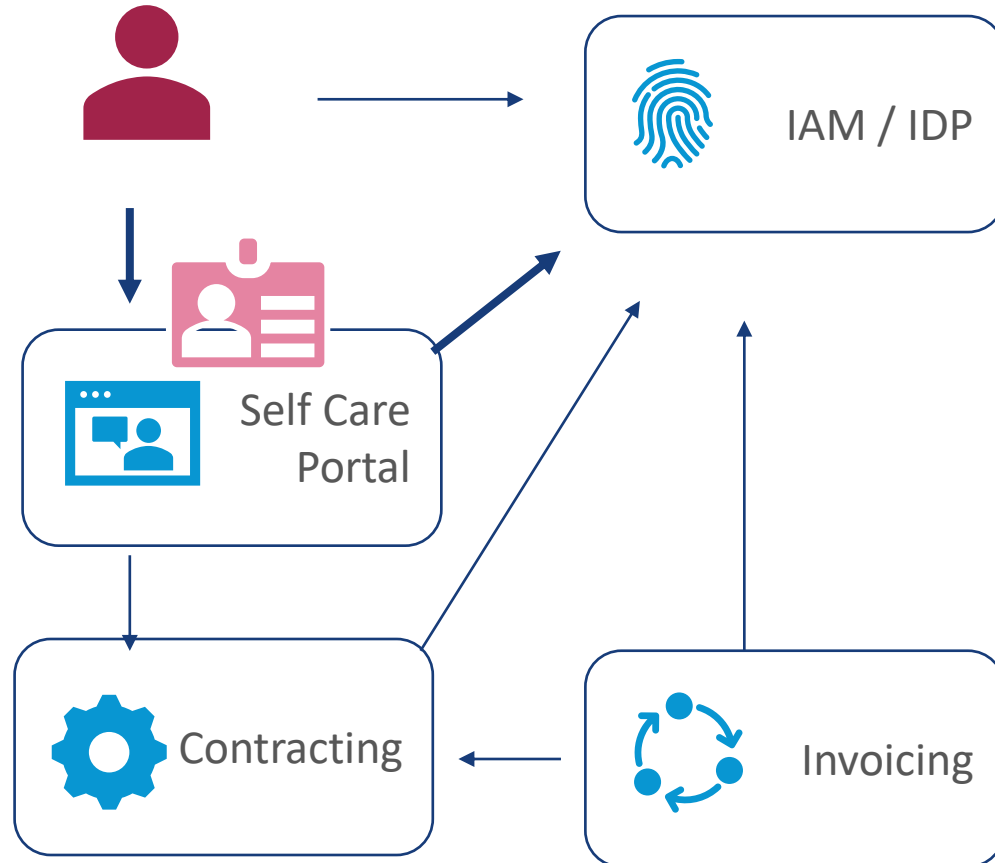
## Security by design – Secure software architecture



User want to access a secured resource in the self care portal:

- Portal redirects to IAM solution – using OpenID Connect
- IAM takes care of Authn: In this case the user needs to provide **username/password**.

## Security by design – Secure software architecture

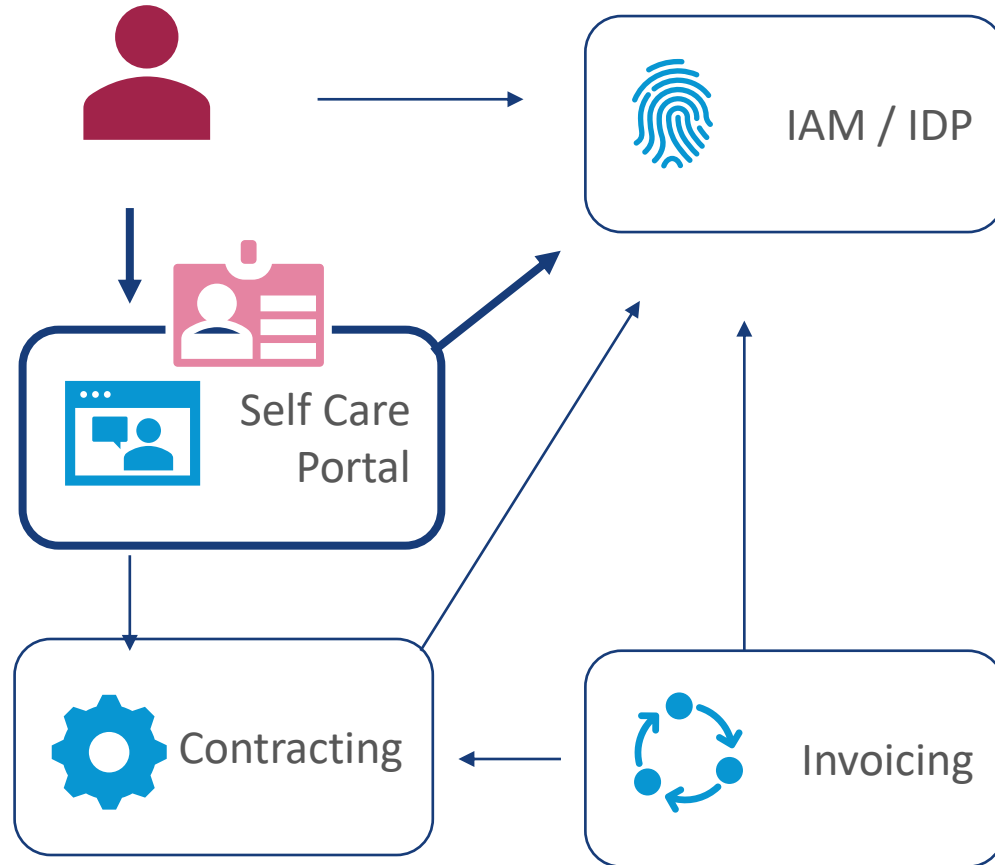


The OpenID Connect / OAuth 2.0 „**Authorization Code grant**“ is used:

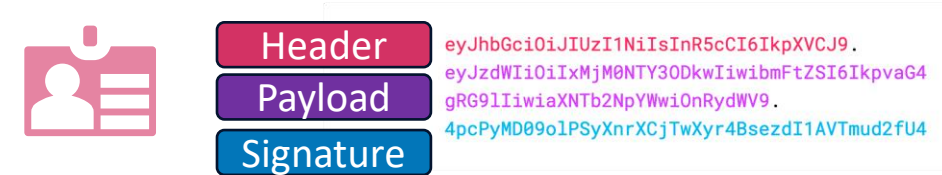
The self care portal receives **ID Token, Refresh Token and Access Token** – they represent the authorization information for the authenticated user



# Security by design – Secure software architecture



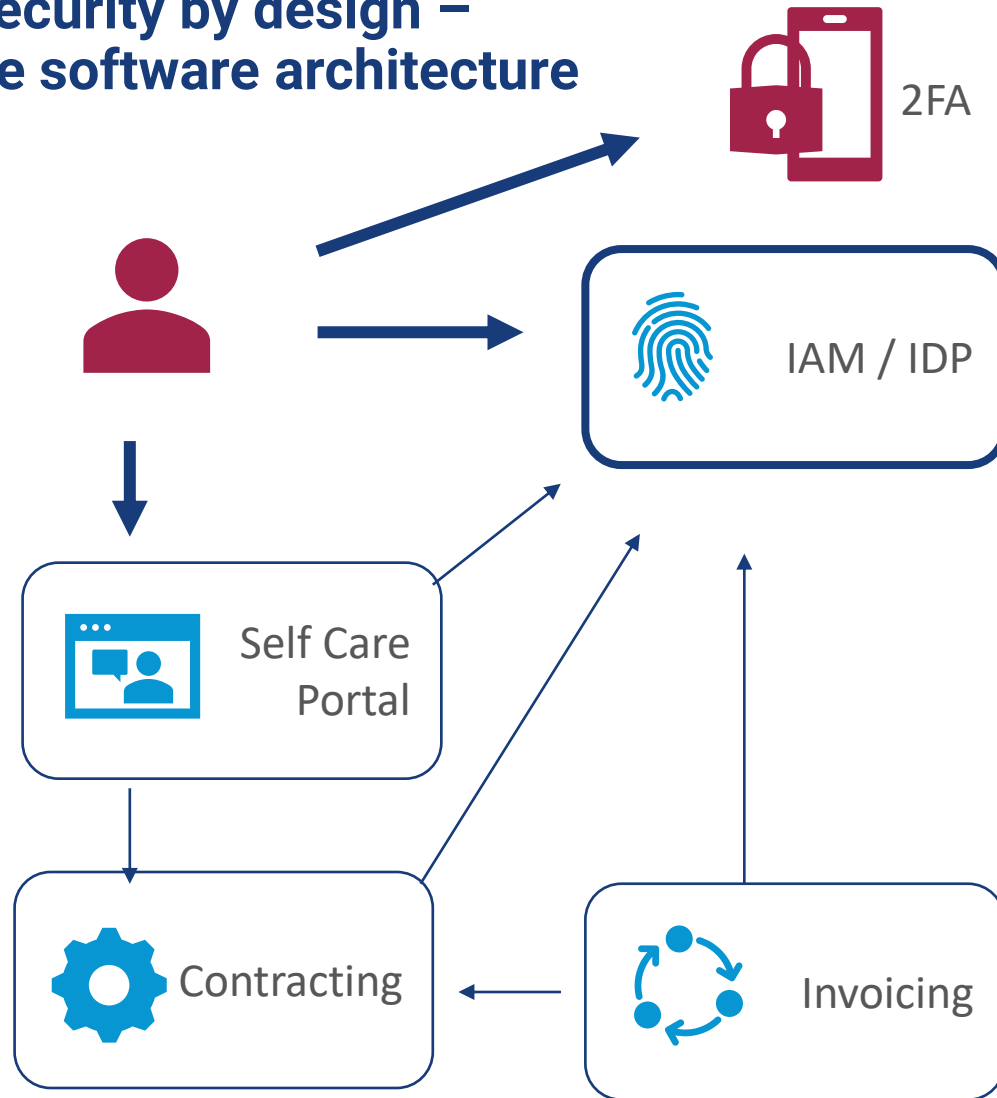
## JWT Token details and verification



The self care portal:

- **Validates** the tokens:
  - Signature
  - Expiration
  - Audience
- **Authorisation** based on RBAC
  - Role is a custom Claim in the payload of the token.

## Security by design – Secure software architecture

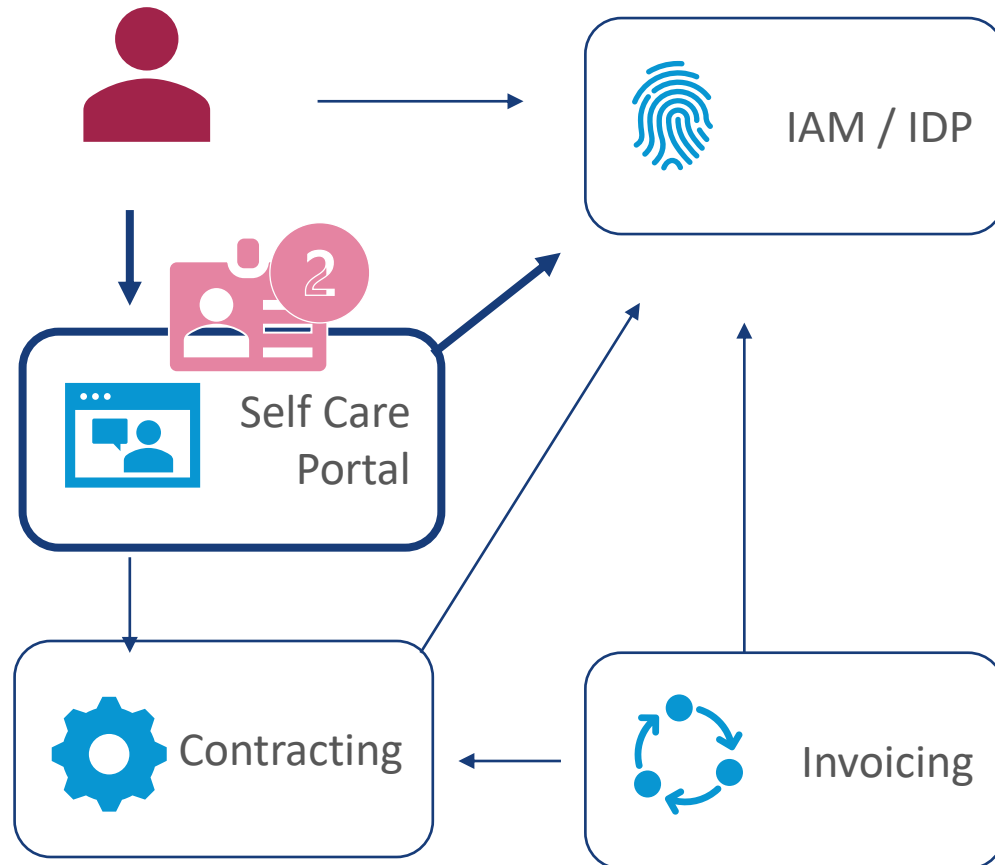


### Step Up Authentication:

User wants to access a critical resource in the self care portal.

- The „level of assurance“ is not high enough (acr claim)
- The self care portal starts a **re-authentication** with the required level of assurance
- The IAM is configured to require a **second factor** – and the user provides the second factor

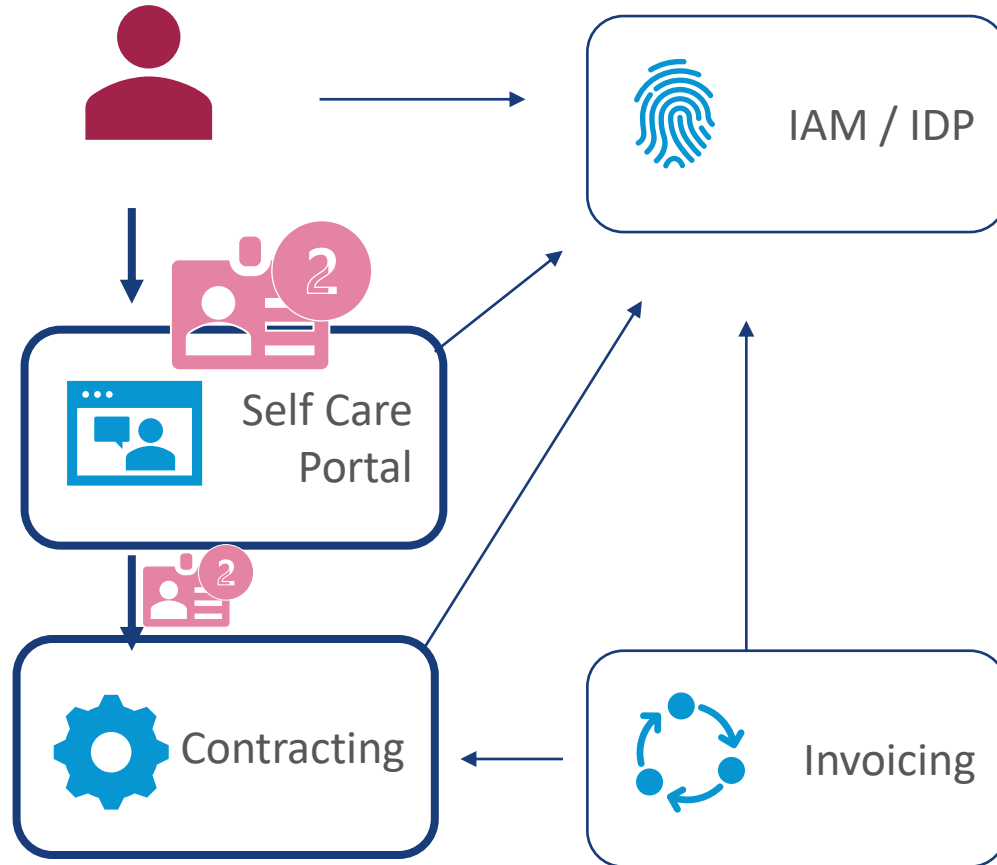
## Security by design – Secure software architecture



The self care portal receives new **ID Token, Refresh Token and Access Token as before** – now with higher level of assurance



## Security by design – Secure software architecture

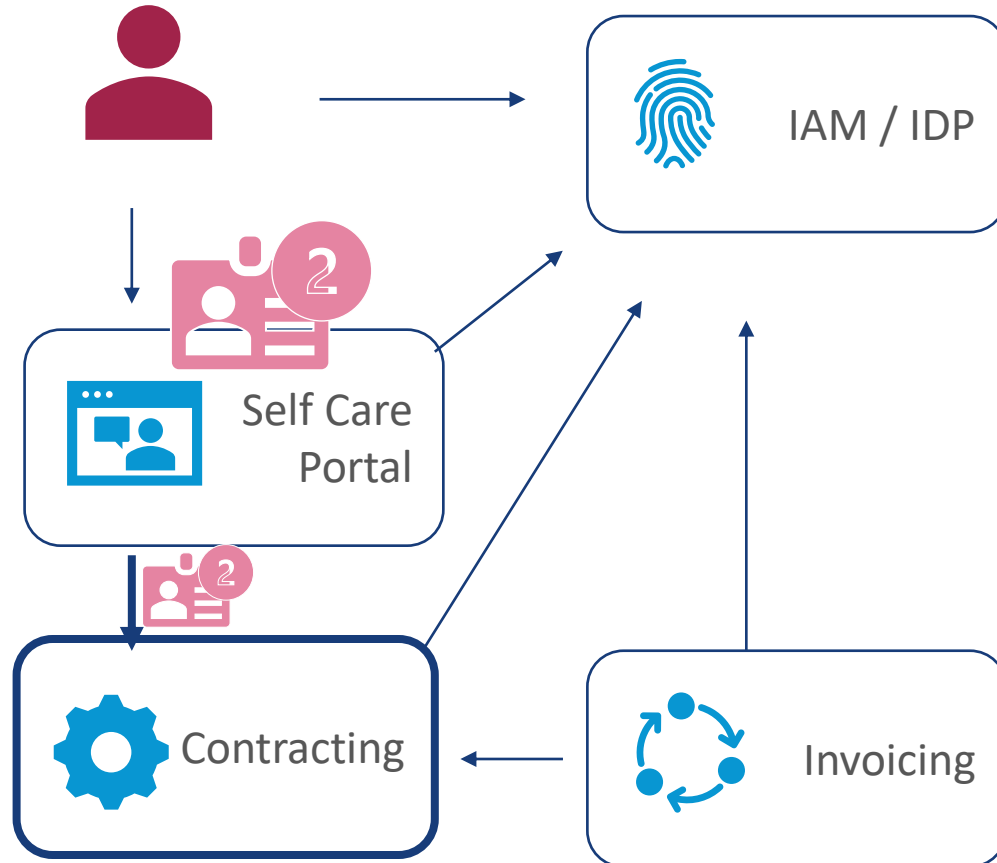


The self care portal needs sensitive user data from the **contracting** service.

### Pass identity upstream:

The API is called and the API request contains the **AccessToken**.

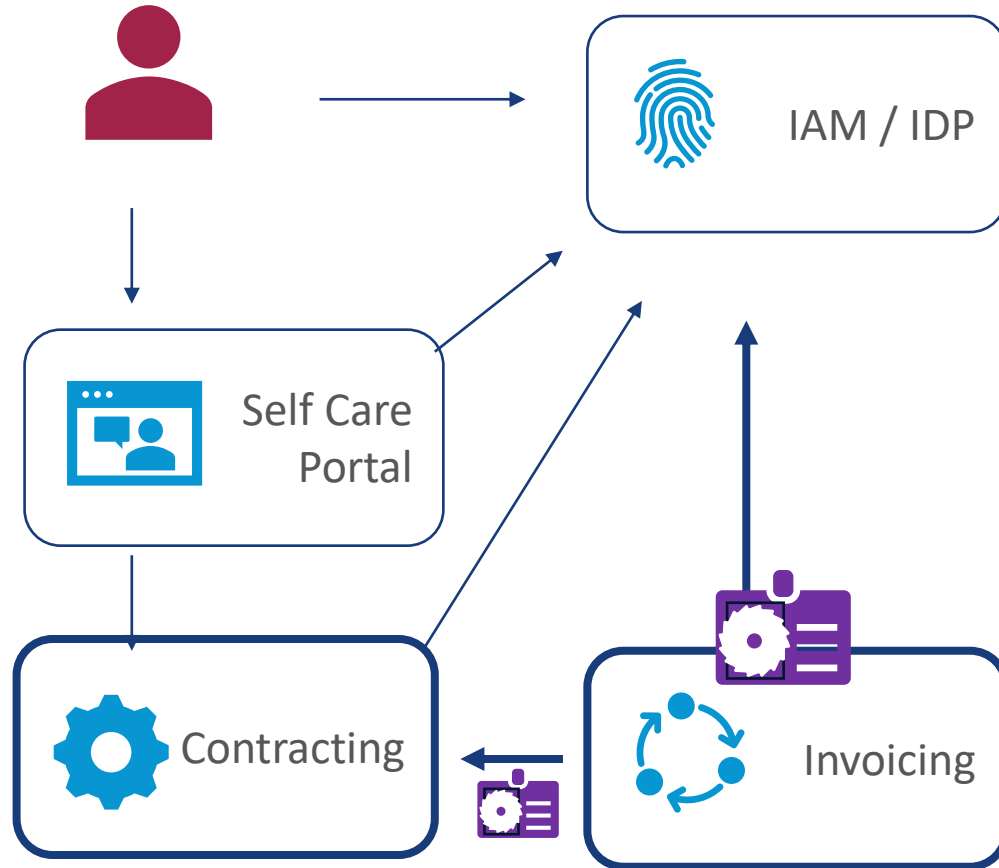
# Security by design – Secure software architecture



## Protected API of the contracting service:

- Passed as Bearer in the Authorization http header
- **Validates Token**
- **Authorization** based on **ABAC**
  - Checking the „customerid“ claim before providing access to restricted data.

## Security by design – Secure software architecture



### Service to service communication:

Invoicing service needs data from contracting service:

- Using the „**Client Credentials Grant**“ flow
- And presenting the **AccessToken** to the according API endpoint

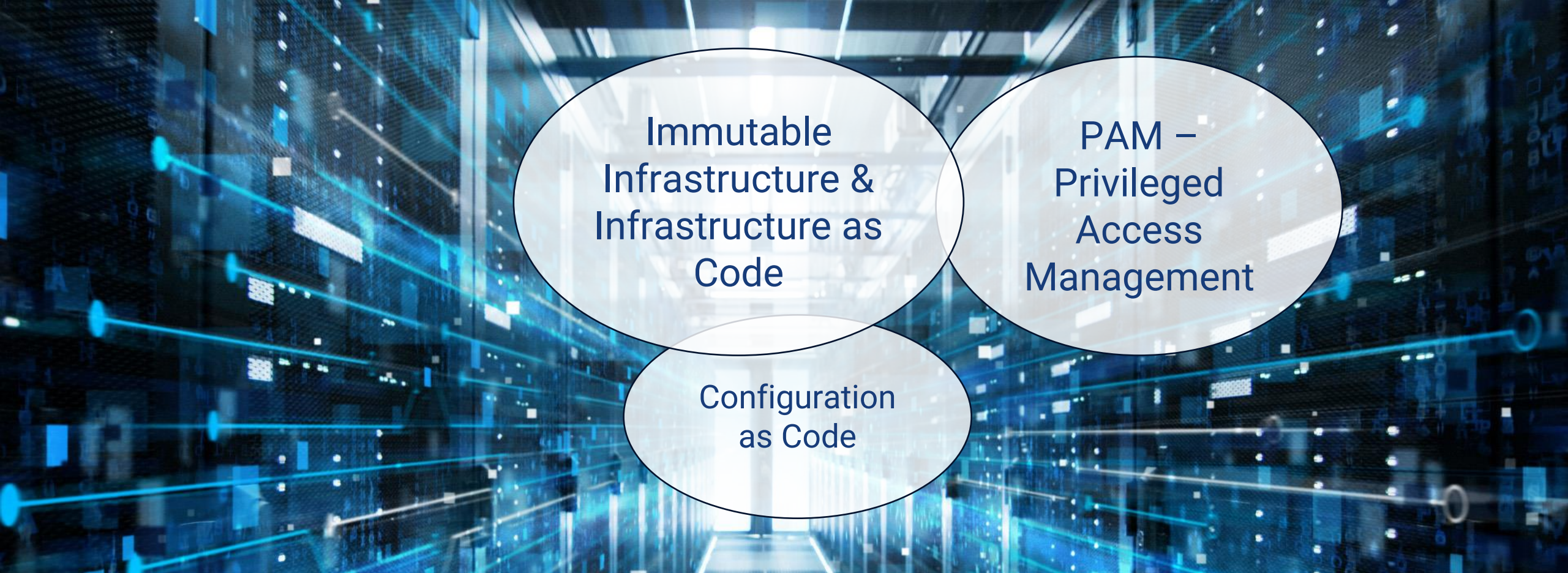




## Application Security - Summary

Build apps as if they are public:  
Don't rely on perimeter security

- Enforce authentication and authorization also for internal APIs
  - Use strong authn where useful
  - Always verify – also downstream
  - Consider RBAC or ABAC
- Ensure Service to Service communication is also
  - Encrypted
  - Authenticated



Immutable  
Infrastructure &  
Infrastructure as  
Code

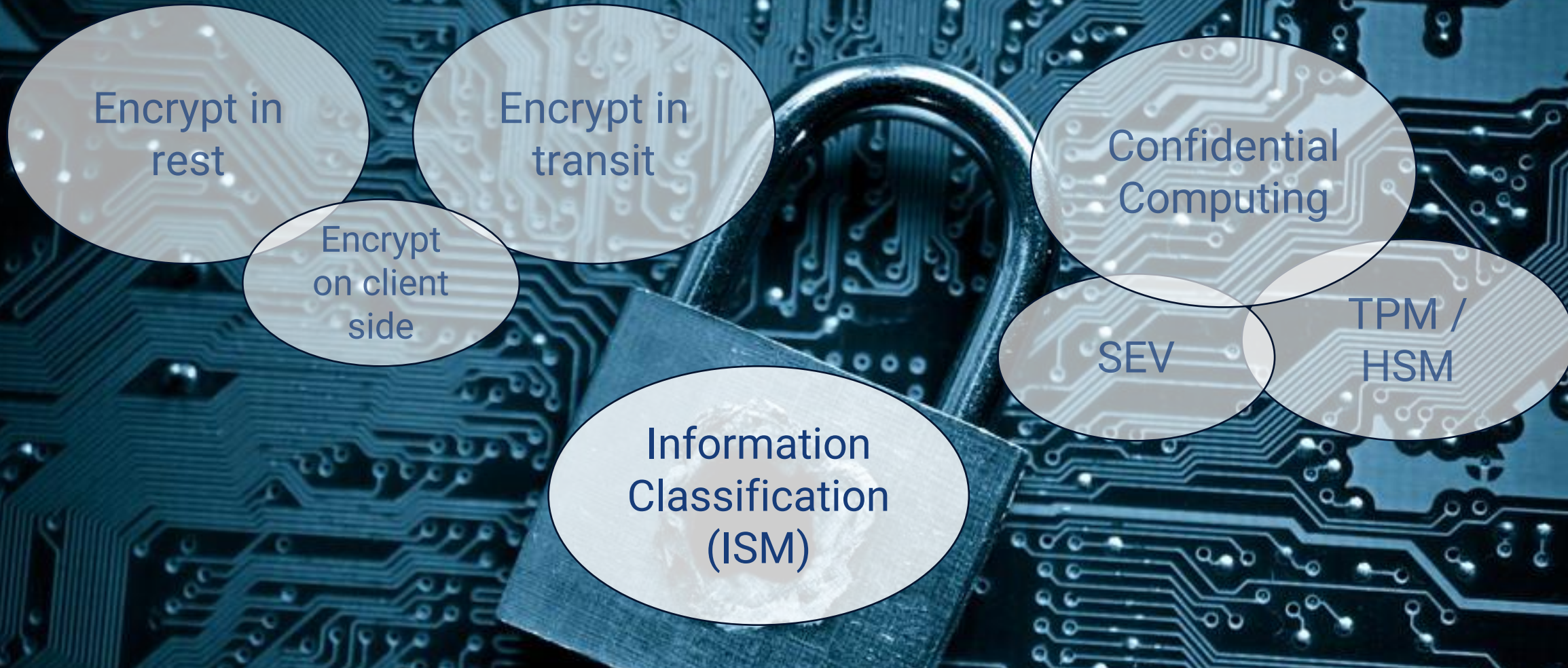
PAM –  
Privileged  
Access  
Management

Configuration  
as Code

## Infrastructure Security

**Defend the Core: Building Robust Infrastructure**





# Secure Data Handling

**Classify, Protect and Safeguard: Locking Down your Data**





Threat  
Modelling

Risk  
Management

Continuous  
X and  
Scanner

Git  
Security

**Secure Development and Delivery**  
**Defend the Pipeline, Trust the Process**



# Secure Development and Delivery

Git Credentials  
example



Commit

Build

Test

Package / Deploy

SAST

Dependency Scanning

Container Scanning

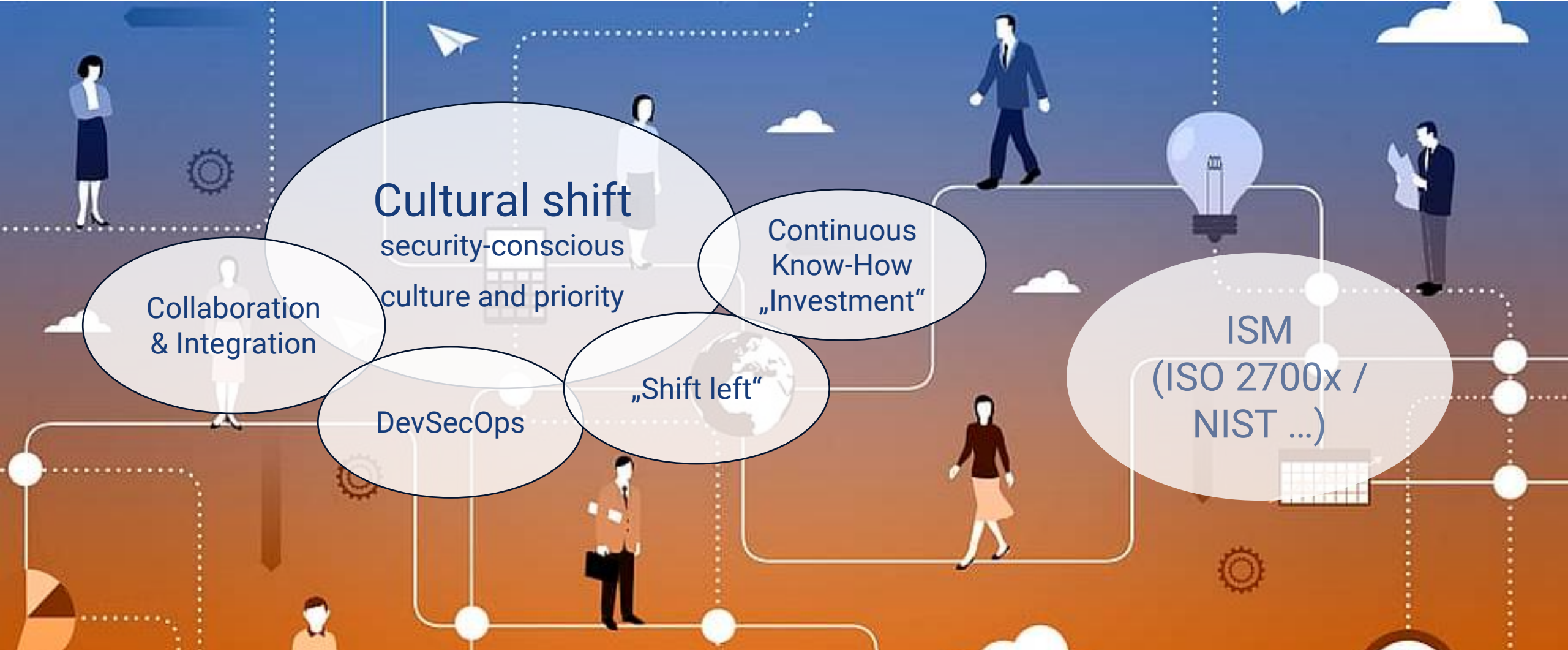
DAST

Pentest GPT 😊



**Security Monitoring**  
**From Detection to Action**

**Automate everything**  
**Reduce errors, win traceability**



# Organization and Culture





# Looking back to “Equifax”

## Looking back to “Equifax”

A widely known vulnerability in Apache Struts

A missing network segmentation

Unencrypted personal credentials on network shares

Unencrypted data

A broken intrusion detection



# Hacks and breaches explode: Security is business-critical



**Zero Trust is multidimensional & hard - but it's worth it.**





Build apps as if they are public



Don't stop, keep up

Talk to an expert



Strong identity / Know the standards



Never trust, always verify!



**Thank you!**

**- meet us at the Expo**

