Experts Live Austria

Azure Core Infrastructure

How to find the right/best way to a manageable **enterprise** Azure core infrastructure.

Thanks to our Sponsors!

base it

ACP IT for innovators.

INFOTECH
[IT & Communication]

IT Management Solutions
www.nice.de

Lenovo

mann&mouse
WE LIVE IT

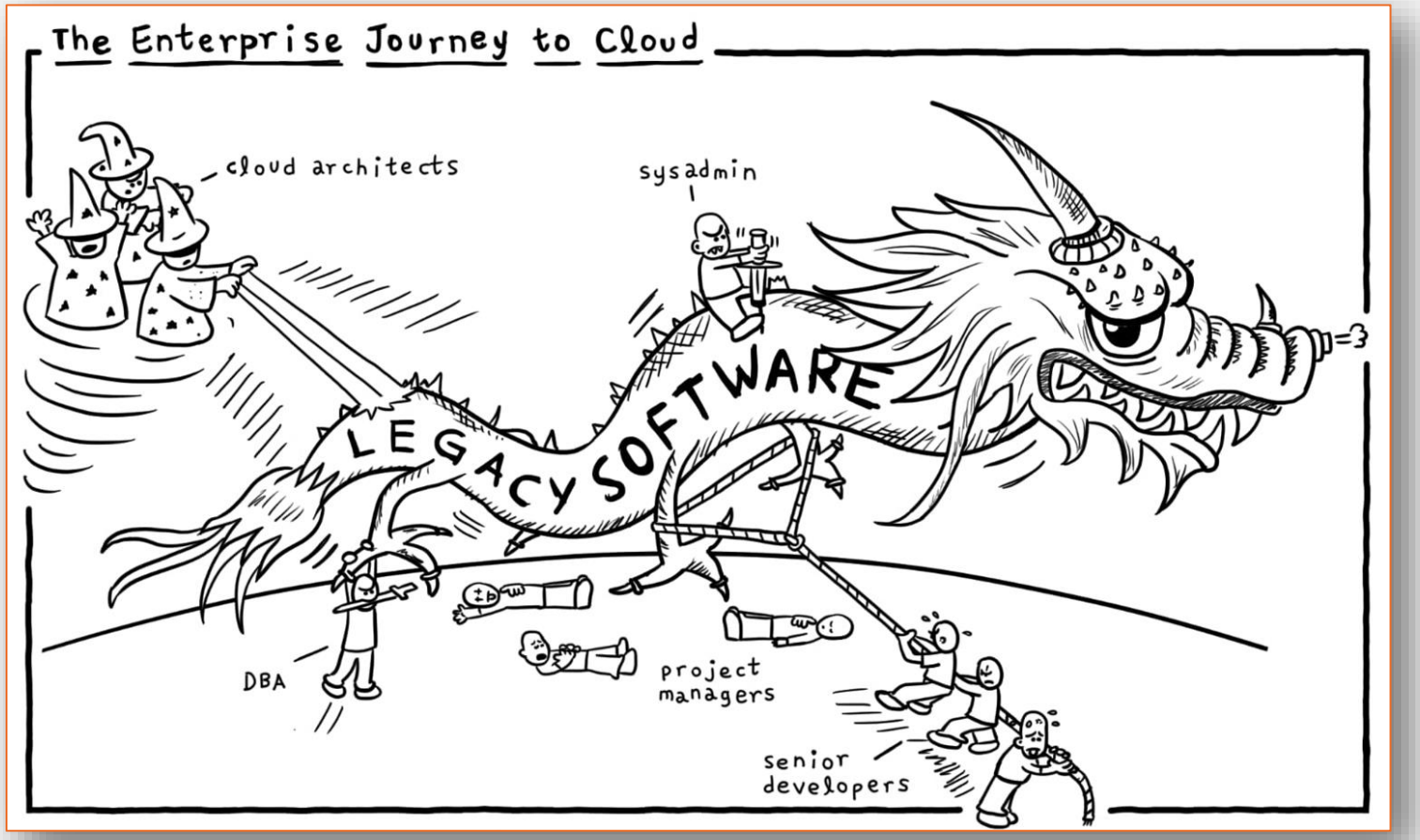
secureguard

ACP IT for innovators.

Status quo – The "*usual*" way to the cloud

Source: https://twitter.com/The__Sina/status/1240007957145743360

The "*successful*" Enterprise Cloud Journey

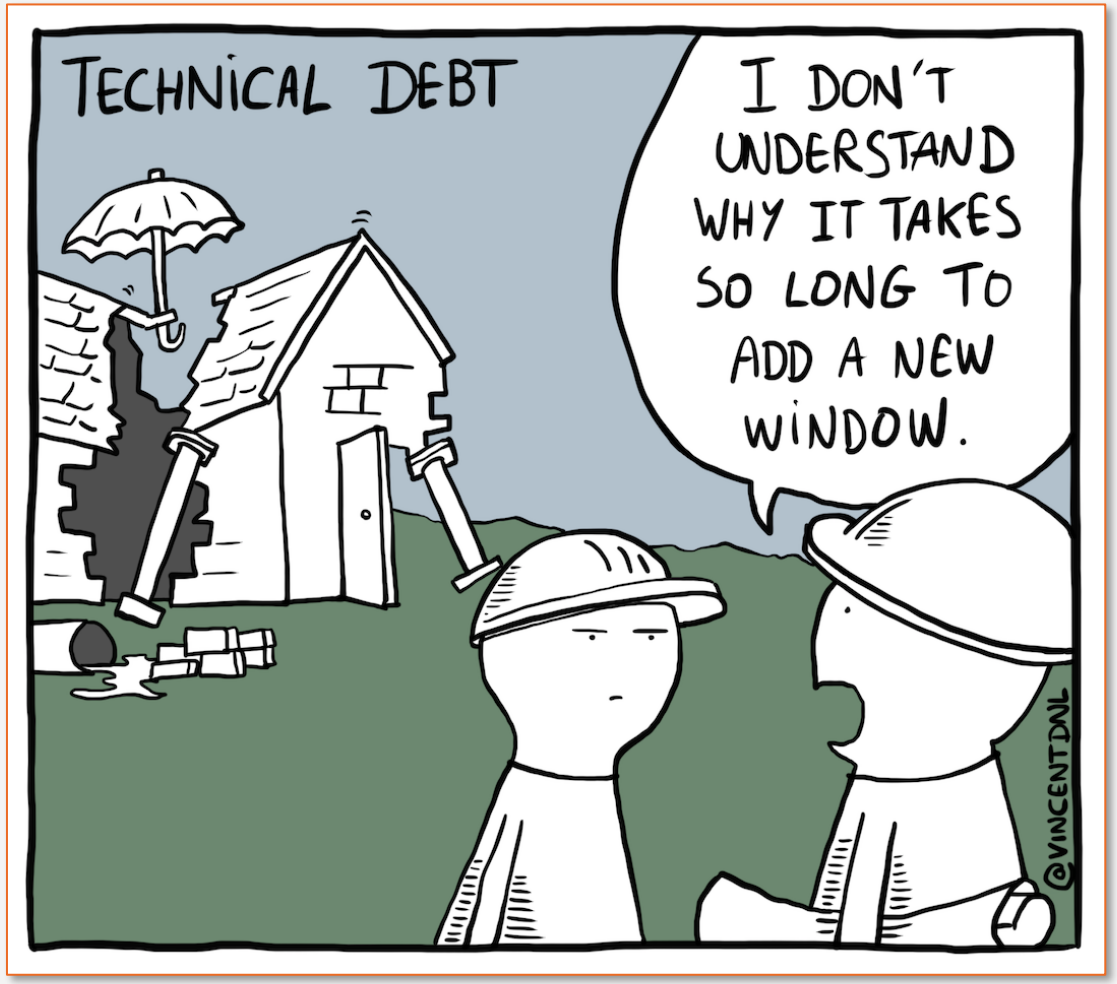


Main Challenges of Customers:

- Governance & Compliance
- High Complexity
- Increasing Costs
- Lack of experience
- Missing Cloud Skills (Partner)

Source: https://tanzu.vmware.com/content/blog/enterprise-journey-to-the-cloud

Avoid “technical debt” in the cloud



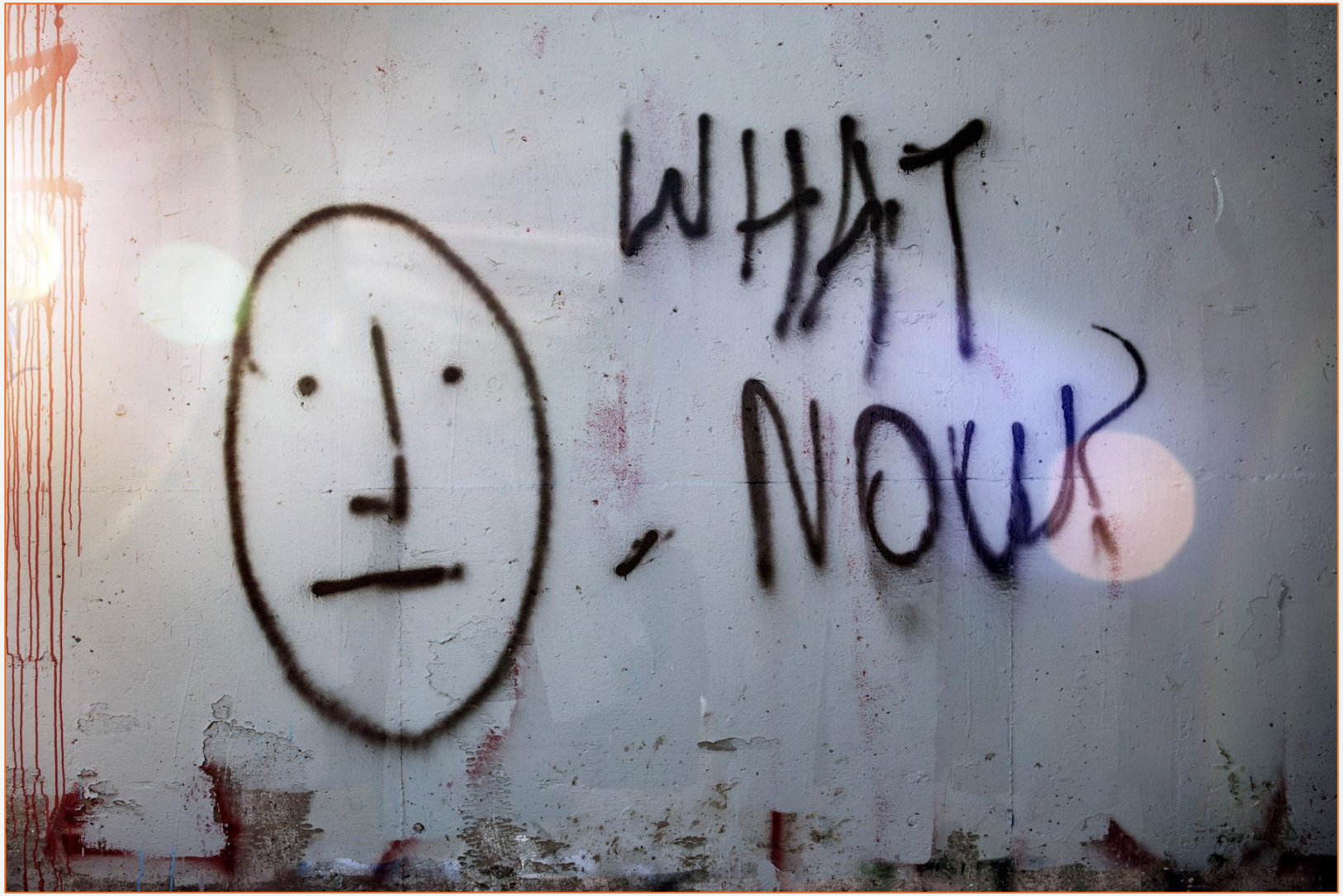
Source: https://vincentdnl.com/drawings/technical-debt

Definition of “*technical debt*”:

In software development, technical debt […] is the **implied cost of future reworking required** when choosing an easy but **limited** solution instead of a better approach that could take more time.

Source: *Wikipedia*

Find your right way to the Azure Cloud



Source: https://unsplash.com/photos/KZcWygxZ_J4

1. Cloud Strategy

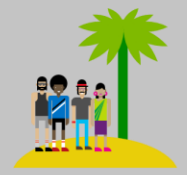
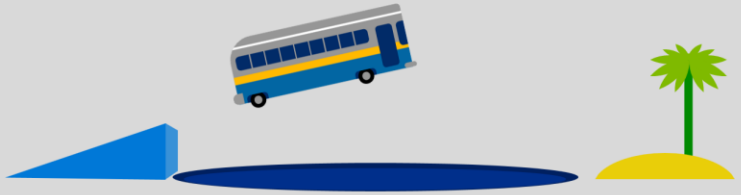
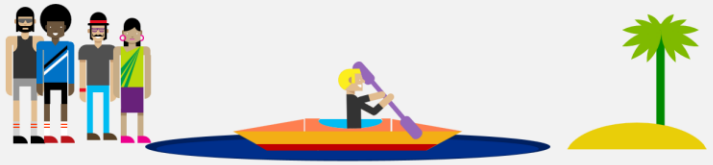
Know where you want to go in the Cloud

› *"Cloud can not deliver what it promises, if you do not know what to expect from it."*

Source: https://unsplash.com/photos/UDIeHDOhBZ8

Cloud Strategy – “Road-2-Cloud”

› How to shift workloads to the cloud?

Cloud-first		A new organization starting with modern workplace
Big Switch Transition		Many workloads need to be modernized at the same time
Group by Group Transition		Doesn't address the needs of the full organization
Iterative (Co-management)		Iteratively move workloads to modern

Source: Microsoft (Slide)

Public Cloud – Strategy

- Procedure and content of cloud strategy definition

Cloud Strategy – Overall:

- Company overall strategy
- Align organization’s goals
- Stakeholder identification
- Obstacles & Dependencies
- Time frame & Milestones
- Internal & external prerequisites

Cloud Strategy – Details:

- Cloud first, Cloud native, **Cloud smart**, etc.
- Multi cloud scenarios (hyperscaler, platform comparison, exit strategy, vendor lock-in, etc.)
- Governance/Compliance (law, rules, audits, etc.)
- Cloud automation (IaC)
- Application Migration Strategy
- Application Modernization (Refactor, Rehost, Relocate, Repurchase, Retire, Retain, etc.)
- Organization (skillset, staffing, business processes, CCoE)

1. Cloud Strategy

2. Azure Governance

Why is a Governance Team needed?

Source: https://pixabay.com/photos/chess-board-game-strategy-toys-3467512/

Define the
"Rules of the Game"

Source: https://unsplash.com/photos/NrS53eUKgiE

Source: https://pixabay.com/photos/white-male-3d-model-isolated-3d-1834094/

Governance Team

- The official guidance in the CAF is to **always** create a cloud governance team. At first, the team might be very **small**.
- A cloud governance team ensures that cloud-adoption **risks** and risk tolerance are properly **evaluated** and **managed**. The team identifies risks that **can't be tolerated by the business**, and it converts risks into governing corporate policies. (Link)

Azure Governance Topics & Aspects

Source: https://www.forbes.com/sites/sap/2014/04/01/10-characteristics-of-the-evolving-cmo/?sh=59f0b30127c4

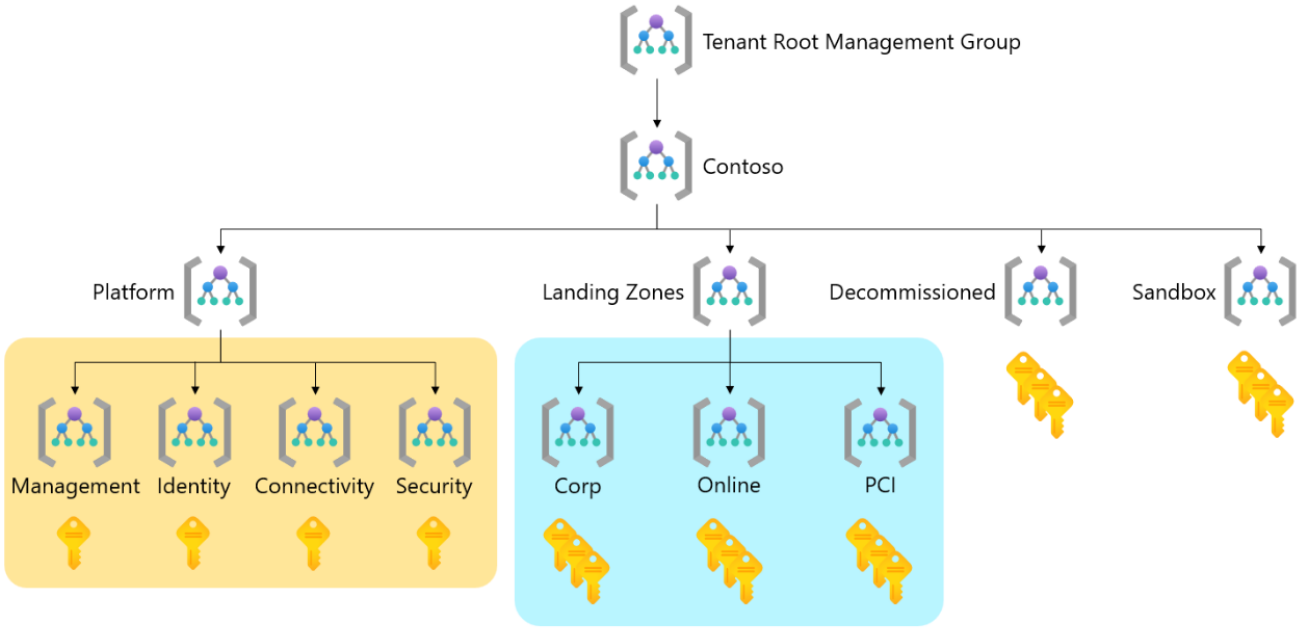
Example: Governance – Azure Hierarchy

CAF – Azure Hierarchy – “Tailored Example”

Link: https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/ready/landing-zone/tailoring-alz#example-of-a-tailored-azure-landing-zone-hierarchy

Example of a tailored Azure landing zone hierarchy

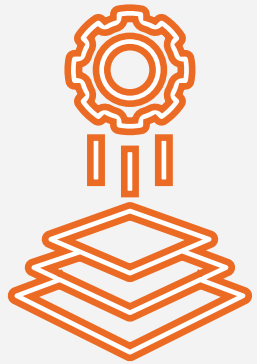
The following diagram shows a tailored Azure landing zone hierarchy. It uses examples from the preceding diagram.



Azure Governance – “Tailor” all design areas before

Transfer

Enterprise



Plan & Specification
(*Commitment*)

2. Azure Governance

3. Azure Core Infrastructure

“Landing Zone”

Microsoft CAF for Azure

- Microsoft Cloud Adoption Framework (CAF) for Azure – Link
- **Goal**: Achieving digitization goals in the cloud (*Guidance*, *Strategy* & *Technology*)
- **Focus**: Best practices, tools, documentations, examples, check lists and templates
- Predefined/structured approach for **cloud transformation** (*quality* & *speed*)

Successful “*Cloud Journey*” ✈

Source: https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/overview#understand-the-lifecycle

Why the Microsoft CAF is not enough?

*"A **framework** is a generic term commonly referring to an essential supporting structure which other things are built on top of."*

(Source: Framework – Wikipedia)

Framework = Construction Kit

Customer Requirements

Microsoft CAF for Azure

Analysis, Design,
Results & Implementation

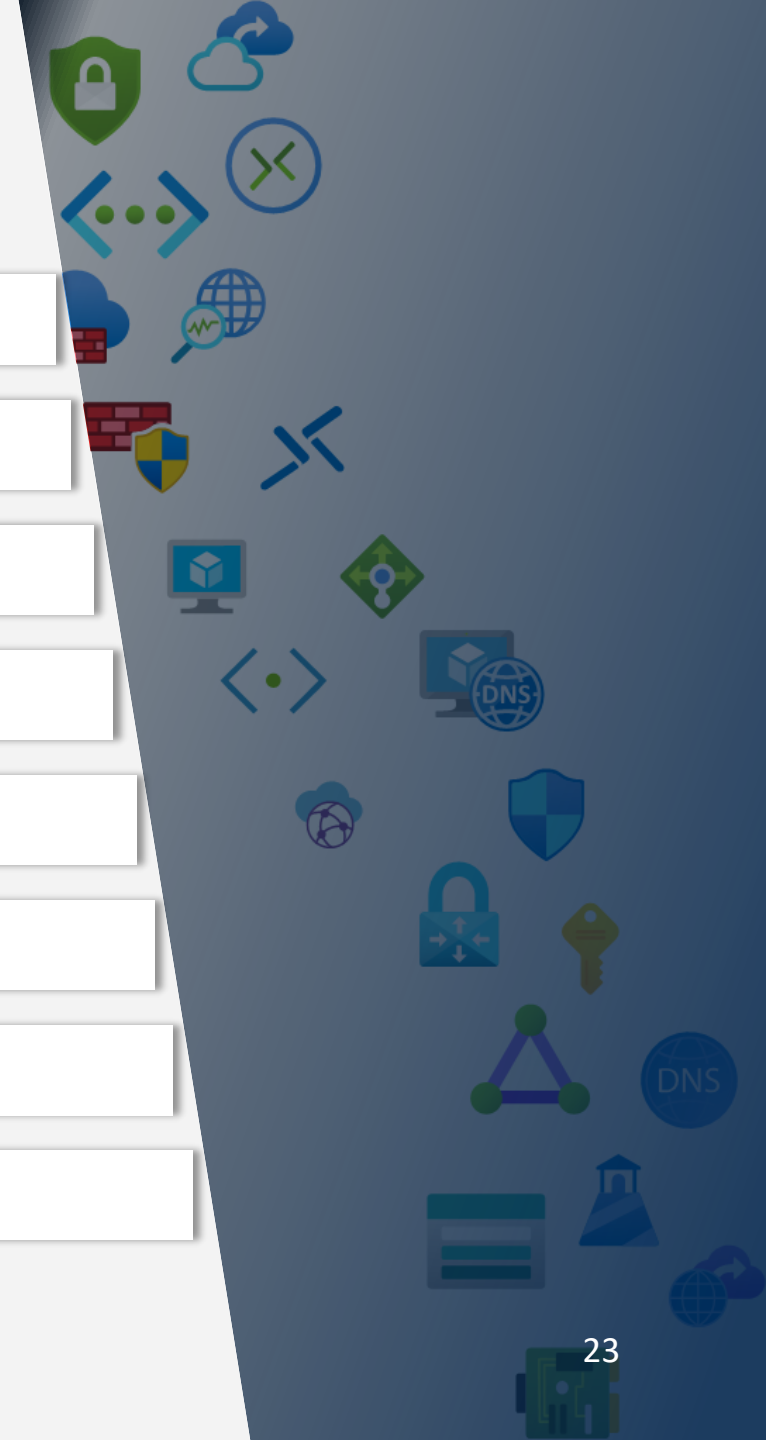
Cloud Adoption **Templates** and standard-based **blueprints** for Azure

Source: https://www.lokalkompass.de/unna/c-lk-gemeinschaft/messe-dortmund-intermodellbau-2016_a650074

Example: Azure Core Infrastructure

Azure Basic/Core General Design

- **Requirements** of the core layer (“*Landing Zone*”).
- Alignment of Azure **Governance** & **Architecture**.
- Design overall **network topology** in Azure (“on-premise to Azure”).
- Consideration **SLAs** and "*single point of failure*“ (SPOF).
- **Responsibilities** for the core layer (interdisciplinary teams & CCoE).
- Azure **Regions**, Availability **Zones** & Region **Pairs**.
- Implementation of all core/basic resources with **IaC** approach (*Terraform*).
- Definition of IT Cloud **Operations** & **Managed Services.**

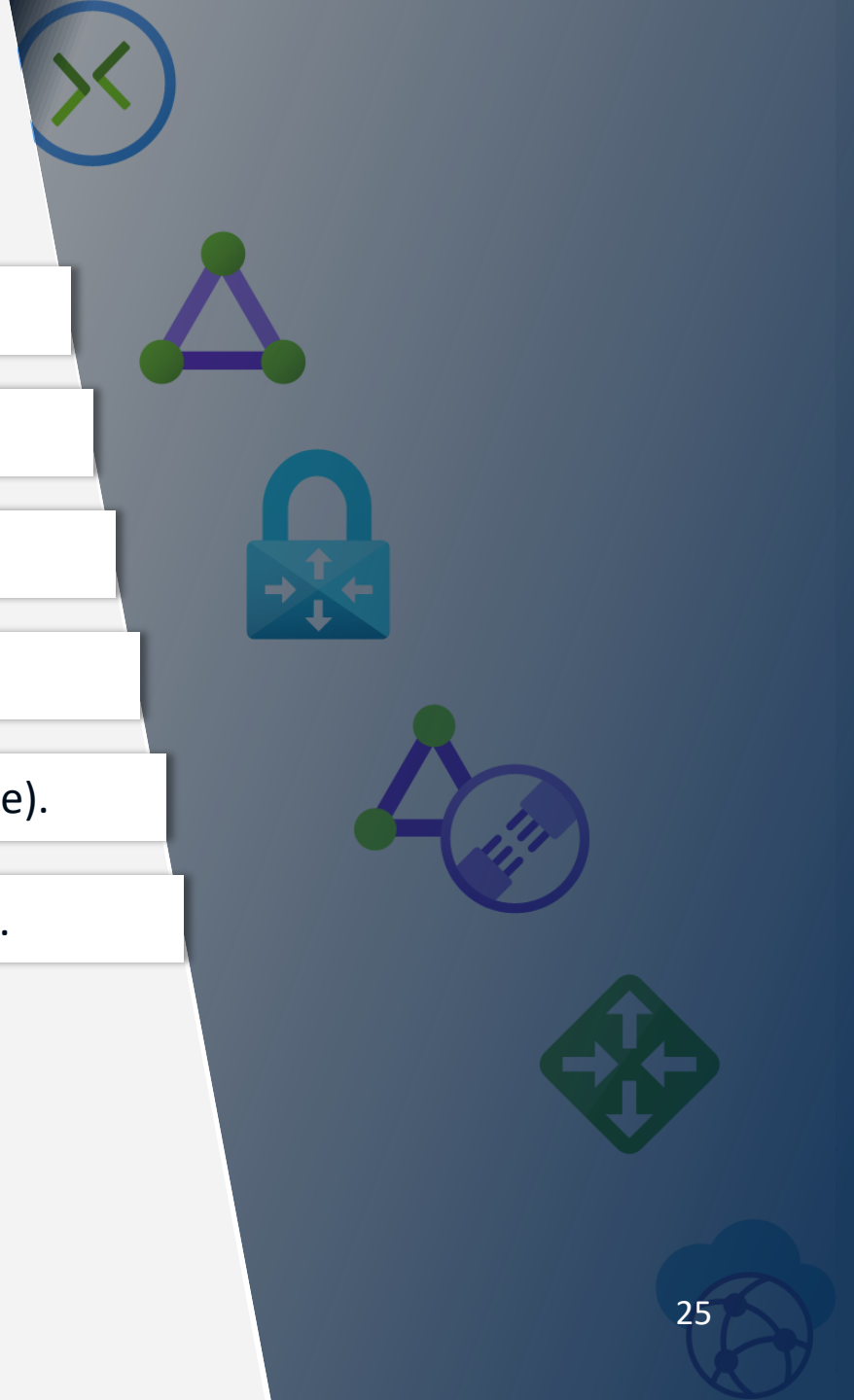


Azure Network Architecture

- Analysis of detailed network requirements (hub & spoke, latency, vWAN, etc.).
- Address range definition and design (CIDR, VNets, Subnets, Peerings, etc.).
- Network **segmentation** & security **zones**.
- Design of a **routing concept** including standardized **traffic flow.**
- **DNS** resolution (possibility to use VMs or Azure DNS Private Resolver).
- **Process** for network adjustments, troubleshooting and expansion requests.
- **Mapping** of the Azure Network in a standardized **IaC library** (e.g., Terraform).

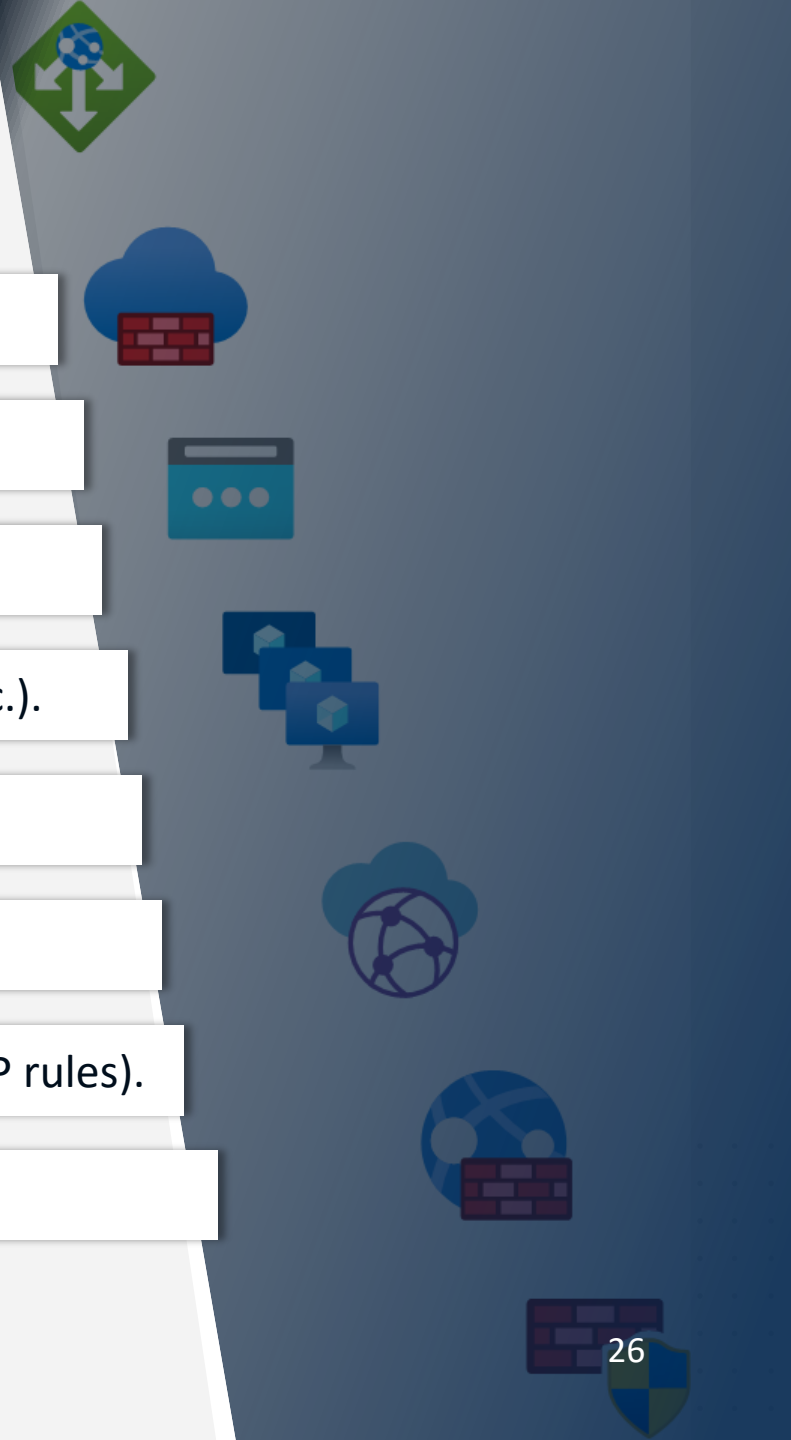
Azure Hybrid Connection

- Analysis of the **local** (*on-premises*) data centers as a starting point.
- Detailed connection **requirements** (bandwidth, outbound traffic, etc.).
- Define **shared services** in the hub network (as the first "*landing zone*").
- Alignment Azure network and hybrid connection (intersection).
- Costs & Definition of a roadmap for hybrid connections (VPN vs. ExpressRoute).
- **Mapping** of the Azure Network in a standardized **IaC library** (e.g., Terraform).



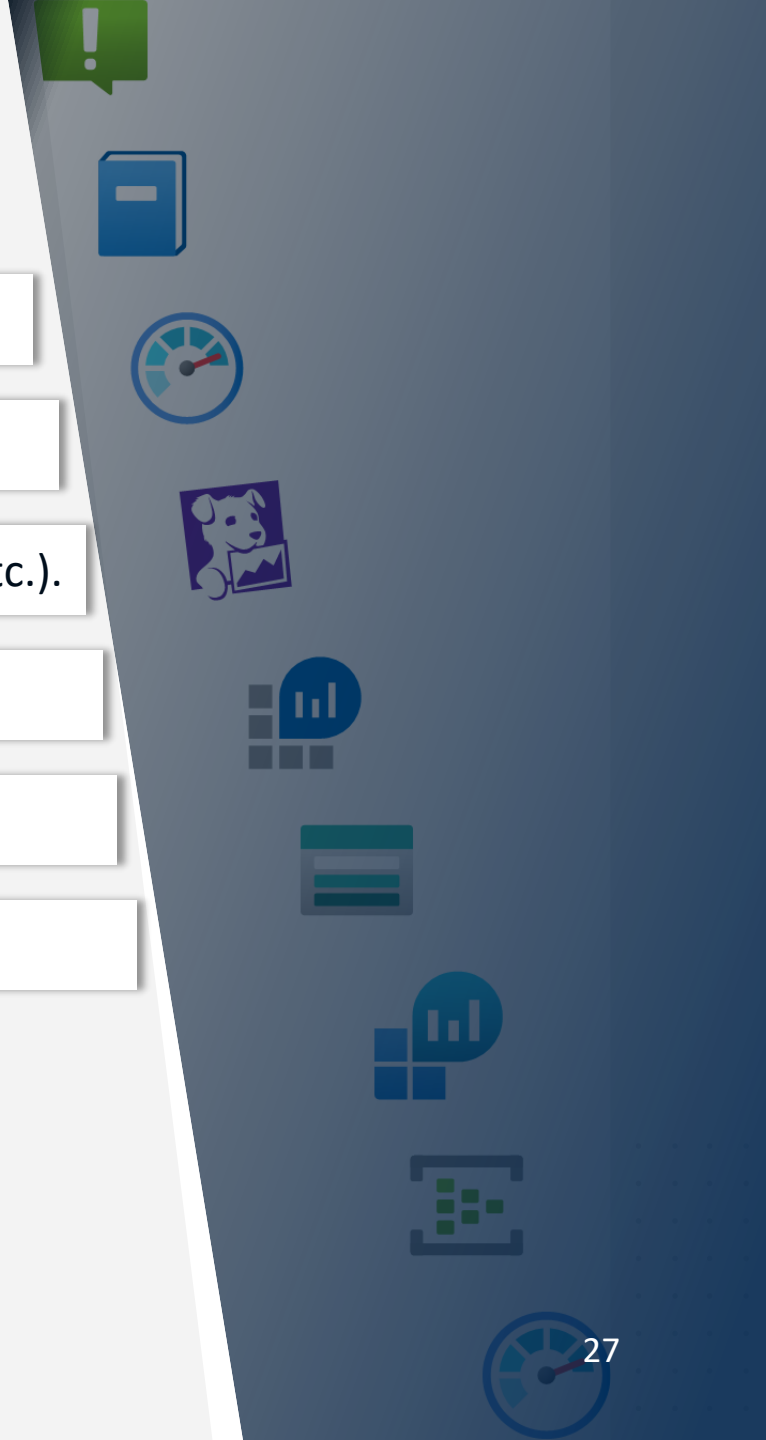
Azure Firewall & Azure NVA

- Requirements for an **Azure NVA** (IaaS) or **Azure Firewall** (PaaS).
- Definition of (high) **availability** for the firewall solution in Azure.
- Avoidance of a single point of failure (SPOF) because of **redundancy.**
- Design firewall **architecture** (VNETs, CIDR blocks, Public IPs, Forced Tunneling, etc.).
- Challenge of the **routing concept** from the Azure Network (*traffic flow*).
- Definition needed **features** (e.g., Proxy, Threat Int., IDPS, TLS inspections, etc.).
- Template standardized settings & rule sets (FW Policies – RCG with NAT, NET, APP rules).
- Deployment of the Firewall solution with **IaC library** (e.g., Terraform).



Logging & Monitoring

- Requirements regarding a **monitoring solution** for Azure resources.
- Application/solution monitoring, infrastructure monitoring.
- Specification of the KPI for Logs & Metrics in Azure (CPU, RAM, Disk, Availability, etc.).
- Use of current tools in combination with a monitoring concept (alerting pipeline).
- Configuration of the diagnostic settings for detailed analysis & monitoring.
- Creation of a dedicated concept for mapping a monitoring strategy.



3. Azure Core Infrastructure

Extended Design Areas & Specifications

- **Backup**-Strategy (on-premise, Azure Backup, etc.).
- Azure Site Recovery (**BCDR**).
- **Shared**-Services (e.g.: DC, DNS, Jumphosts, Bastion, Firewall, etc.).
- Key-/Secret- & Certificate-Management (**KSC**).
- **Network Security** (e.g.: Internet-Breakout, Environments, Endpoints).
- **Serverbereitstellung** (e.g.: VMs, Images, Agents, SKUs, Update, Encryption, etc.).
- Microsoft Defender for Cloud (*Azure Security Center*).
- Orientation: Azure Reference Architecture (Link).

3.x Azure Core Infrastructure

“Extended”

4. Cloud Automation

"Infrastructure as Code" (IaC)

What is IaC?

Managing/Provisioning of infrastructures using **code**.

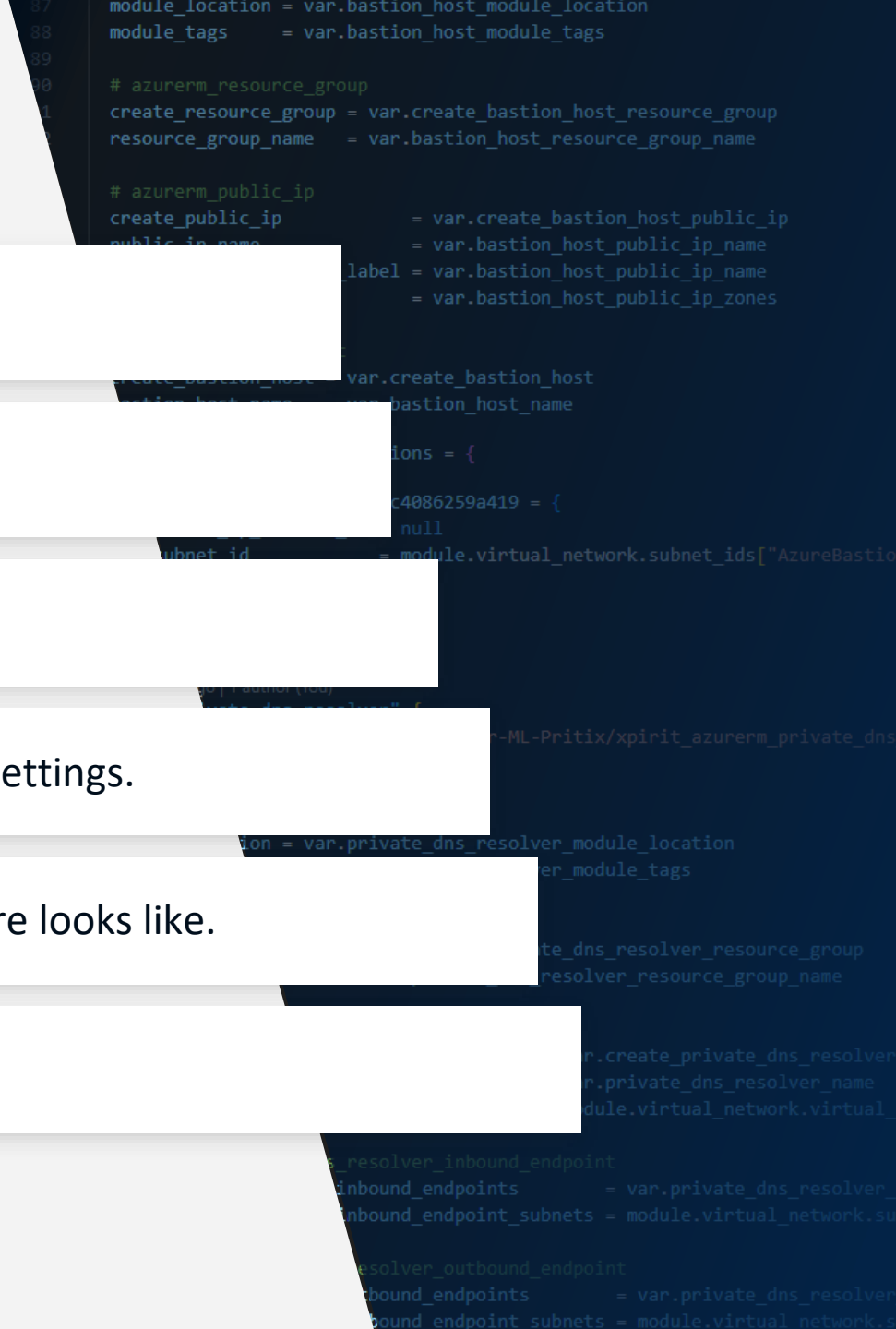
How resources, applications and environments are **configured**.

Code describes the resources and the whole **architecture** landscapes.

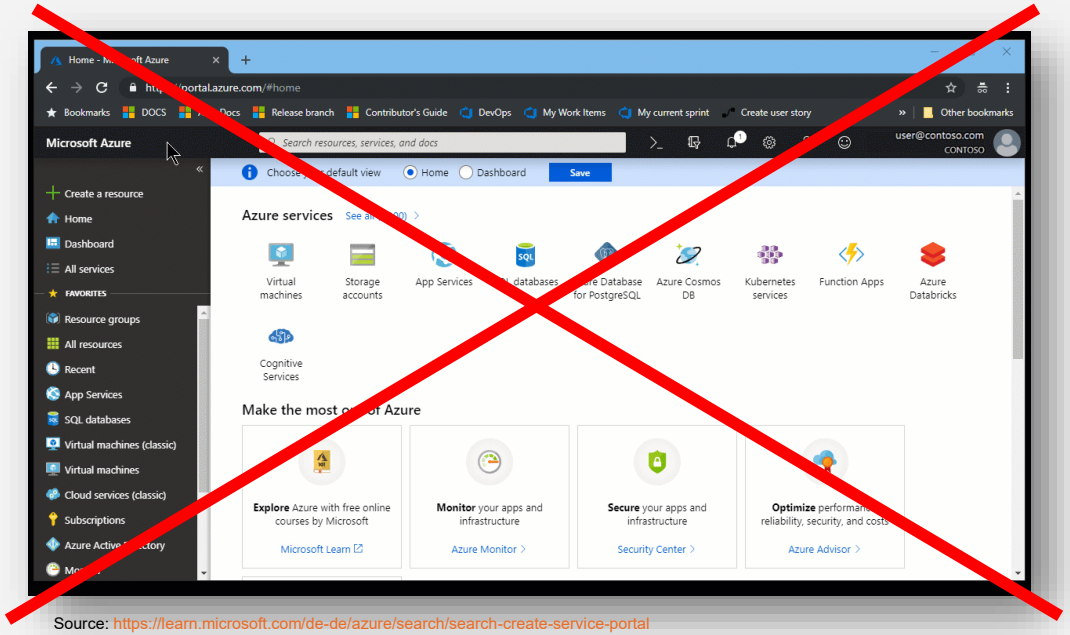
Declarative syntax can be used to specify the resources and detailed settings.

Human readable code files (`main.tf`) describe how the infrastructure looks like.

Definition of **variables** and their **values** of the environment.



Mind change administration interface in the Cloud



Source: https://learn.microsoft.com/de-de/azure/search/search-create-service-portal

```
You, 4 weeks ago | 1 author (You)
resource "azurerm_resource_group" "this" {      You, 2 months ago • feat: create initial code base …
  count    = var.create_resource_group ? 1 : 0
  name     = var.resource_group_name
  location = var.module_location

  tags = merge(
    var.module_tags,
    var.resource_group_tags
  )
}

You, 4 weeks ago | 1 author (You)
resource "azurerm_virtual_network" "this" {
  count                = var.create_virtual_network ? 1 : 0
  location             = var.module_location
  resource_group_name  = try(azurerm_resource_group.this[0].name, var.resource_group_name)
  name                 = var.virtual_network_name
  address_space        = var.virtual_network_address_space
  bgp_community        = var.virtual_network_bgp_community
  dns_servers          = var.virtual_network_dns_servers
  edge_zone            = var.virtual_network_edge_zone
  flow_timeout_in_minutes = var.virtual_network_flow_timeout_in_minutes

  You, 2 months ago | 1 author (You)
  dynamic "ddos_protection_plan" {
    for_each = var.virtual_network_ddos_protection_plan != null ? var.virtual_network_ddos_protection_plan : {}
```

Avoid:
„Click-Click-Cloud“
„Clicky-Bunti“

„Transition 2 cloud“
→ Shift from “*static*” to “*dynamic*” infrastructure

Cloud Operating Model

Using **IaC** deployment **templates** & **modules** in building out **your** individual Cloud Infrastructure!

Demo: Terraform Code

Advantages of using an IaC approach

- Reusability - Creation of "repeating" infrastructures (**standardization/template**).
- Speed - **Faster** creation of resources and environments.
- Parallelization - Several developers can work on the same infrastructure **simultaneously**.
- Documentation - The **code definition file** is also the documentation of the environment.
- Consistency - The code reflects the **overall state** of the environment.
- Tracking - Use of source code management (**version control**).
- Order - Order of resource creation is managed **automatically**. Automate changes.
- Error/state handling - **No check** necessary if resources already exist ("*Desired State*").
- Quality - Less **human error** by using the same code ("*no-brainer*").
- Reproducibility - The **same code** always leads to the **same result**.
- Cost Reduction - Remove **manual** tasks. People refocus their efforts on enterprise tasks.

4. Cloud Automation ☑

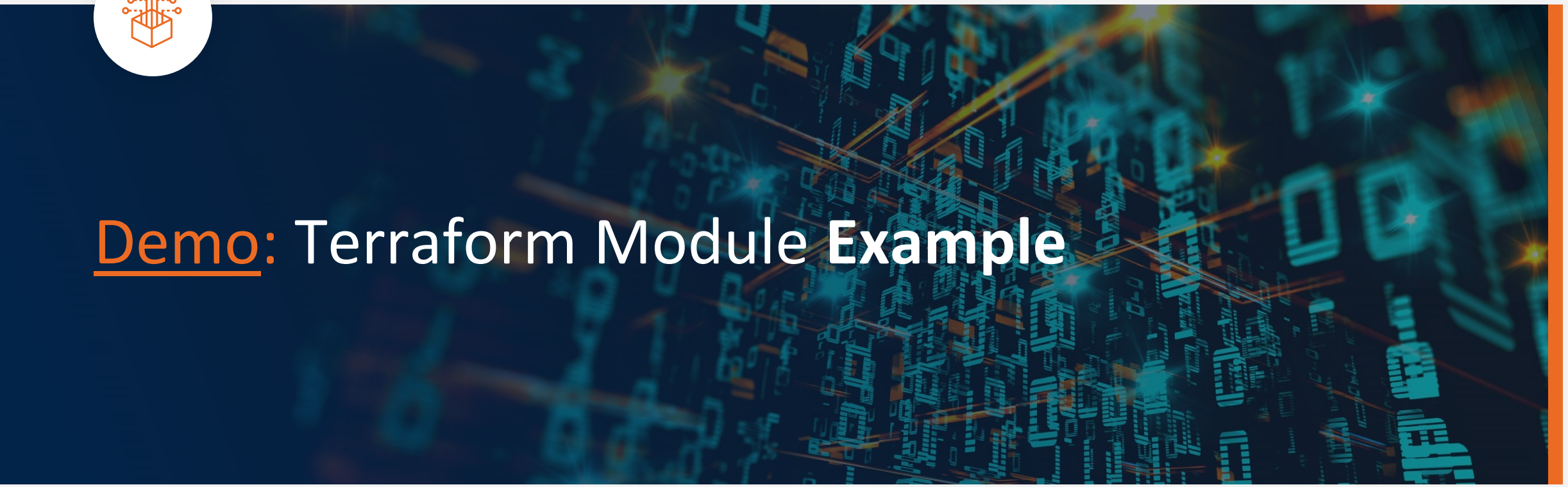
“Infrastructure as Code (IaC)”

5. “IaC” Module Library

What is a module?

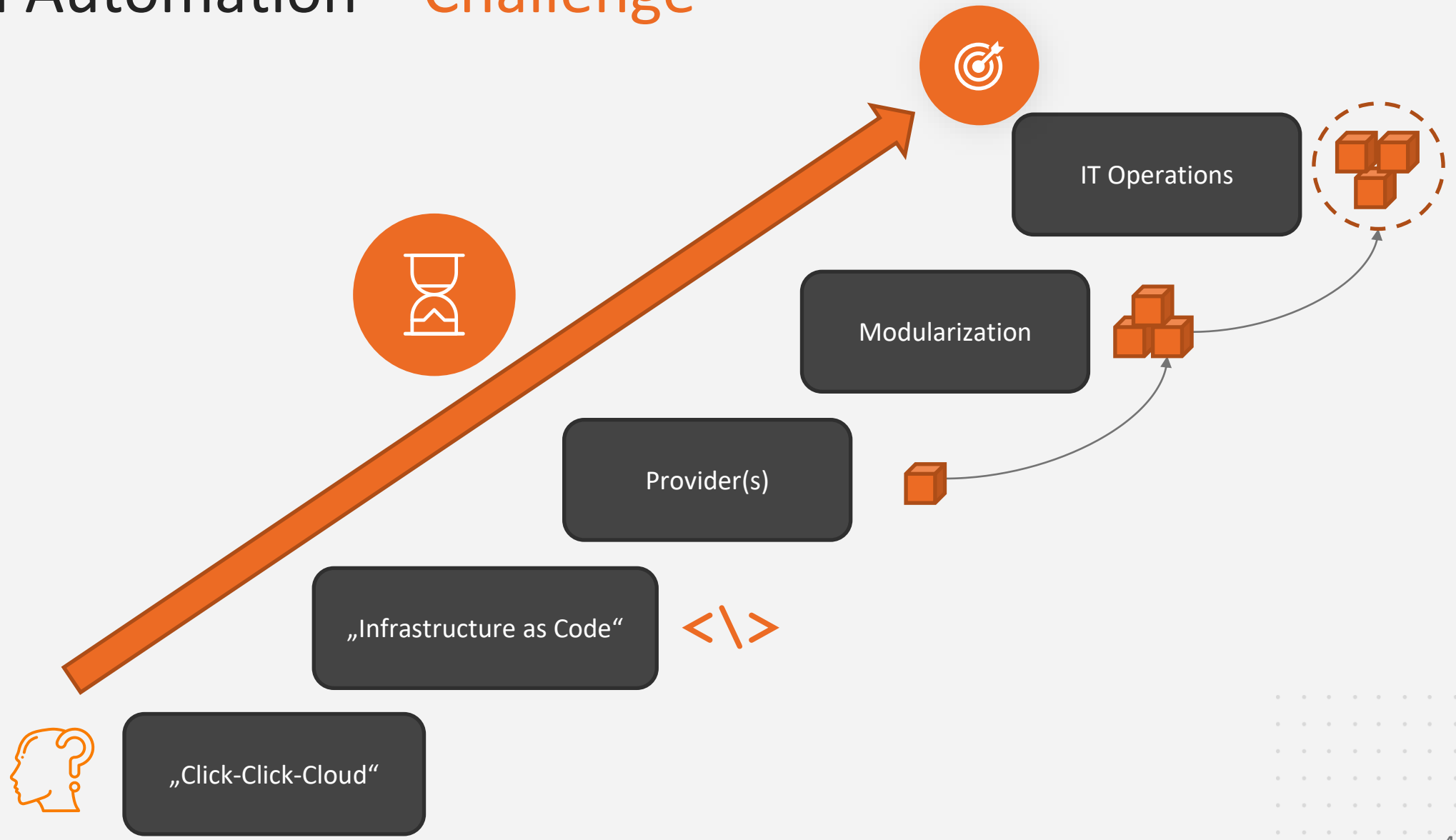
- Create lightweight **abstractions**
- Don’t describe **physical** objects
- Describe infrastructure in terms of its **architecture**
- Package and **reuse** resource configurations
- Separated in code **repositories**
- **Independently** usable & testable

> **“A module is a container for multiple resources that are used together.”**



Demo: Terraform Module **Example**

Cloud Automation – Challenge

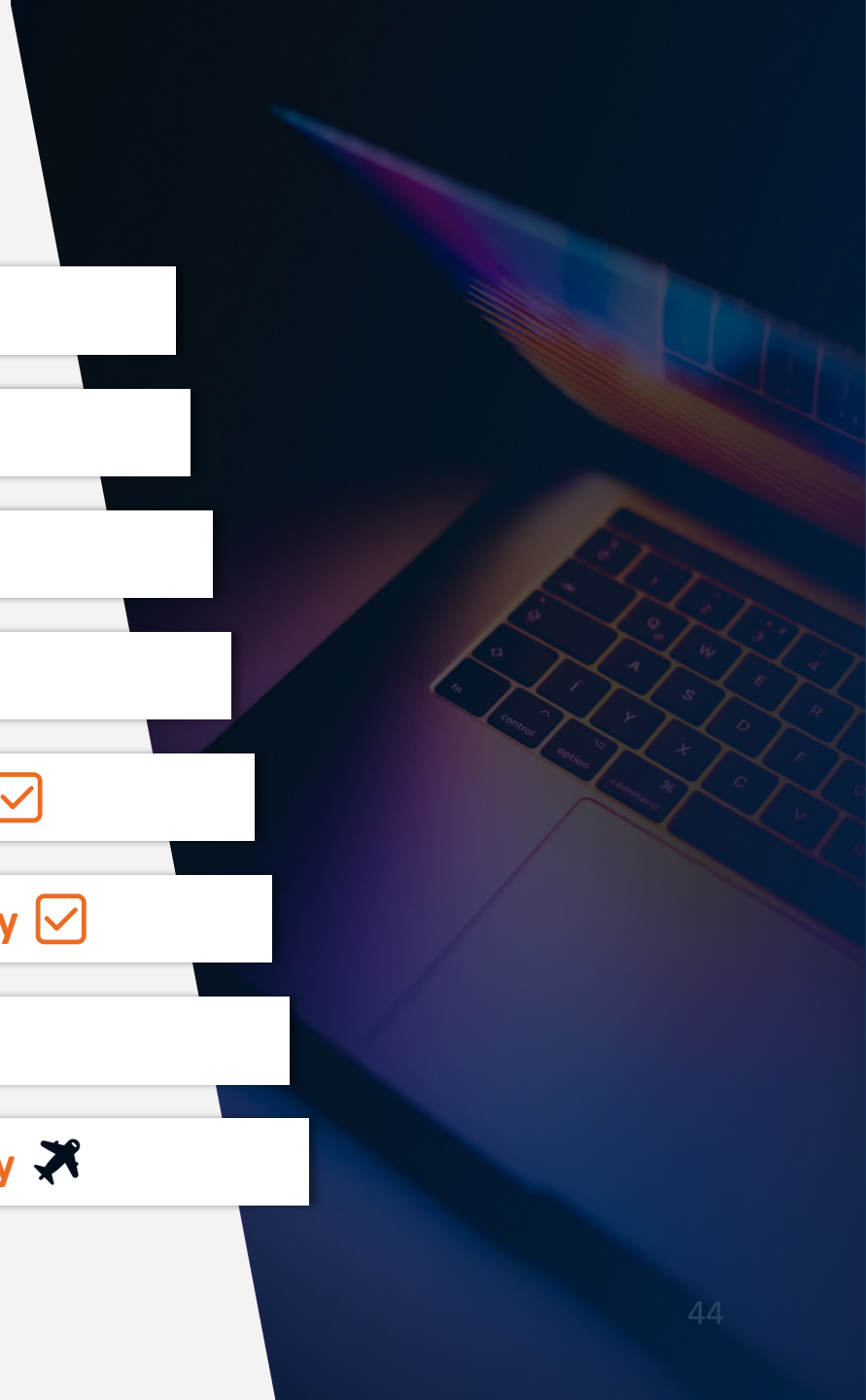


4. Cloud Automation

“Modularization”

Conclusion – Design Principals

Know your goals/destination	→ **Cloud Strategy** ☑
Don’t do things “later”	→ **Technical Dept** ☑
Define the rules & guardrails	→ **Governance** Concept & Team ☑
Get orientation & rethink	→ Microsoft **CAF** for Azure ☑
Design & Architecture	→ **Core** Infrastructure (“*Extended*”) ☑
Modularization	→ Provide shared IaC **Module Library** ☑
Don’t do “*Click-Click-Cloud*”	→ Infrastructure as Code – **IaC** ☑
Stay on Track	→ Follow you planned **Cloud Journey** ✈



Stefan Rapp

Cloud Solution Architect (CSA)
Xpirit Germany GmbH

https://www.linkedin.com/in/rapster83

@rapster83

https://blog.misterazure.com

- 15 years in IT Consulting
- 6 years MS Development & Infrastructure
- Since 2018 Azure Governance & Infrastructure
- Application Modernization towards Azure
- Pushing IaC (Terraform) at customers

45

Thanks to our Sponsors!

base it

ACP IT for innovators.

INFOTECH
[IT & Communication]

IT Management Solutions
www.nice.de

Lenovo

mann&mouse
WE LIVE IT

secureguard

ACP IT for innovators.

Thank you (Q&A)

Experts Live Austria

Blog : https://blog.misterazure.com

GitHub : @rapster83

LinkedIn : https://www.linkedin.com/in/rapster83/