Æ-DIR - Authorized Entities Directory

- Paranoide Benutzerverwaltung mit OpenLDAP -

Tübix 2018

Zur Person

- Michael Ströder <michael@stroeder.com>, Freiberufler
- Schwerpunkte
  - Identity & Access Management, Verzeichnisdienste (LDAP)
  - Single Sign-On, Zwei-Faktor Authentifizierung
  - PKI (X.509, SSH), Verschlüsselung, dig. Signatur
- Open Source / Freie Software:
  Æ-DIR, OATH-LDAP, web2ldap

Agenda

- Ziele
- Architektur
- Datenmodell
- Berechtigungen
- Anwendungsbeispiele

Ziele

- Prinzipien
  - Need-to-know
  - Least Privilege
  - Separation of Duties
- Delegierte Administration überschaubarer Bereiche
- Aussagekräftiger Audit Trail
- Basis für Compliance-Checks

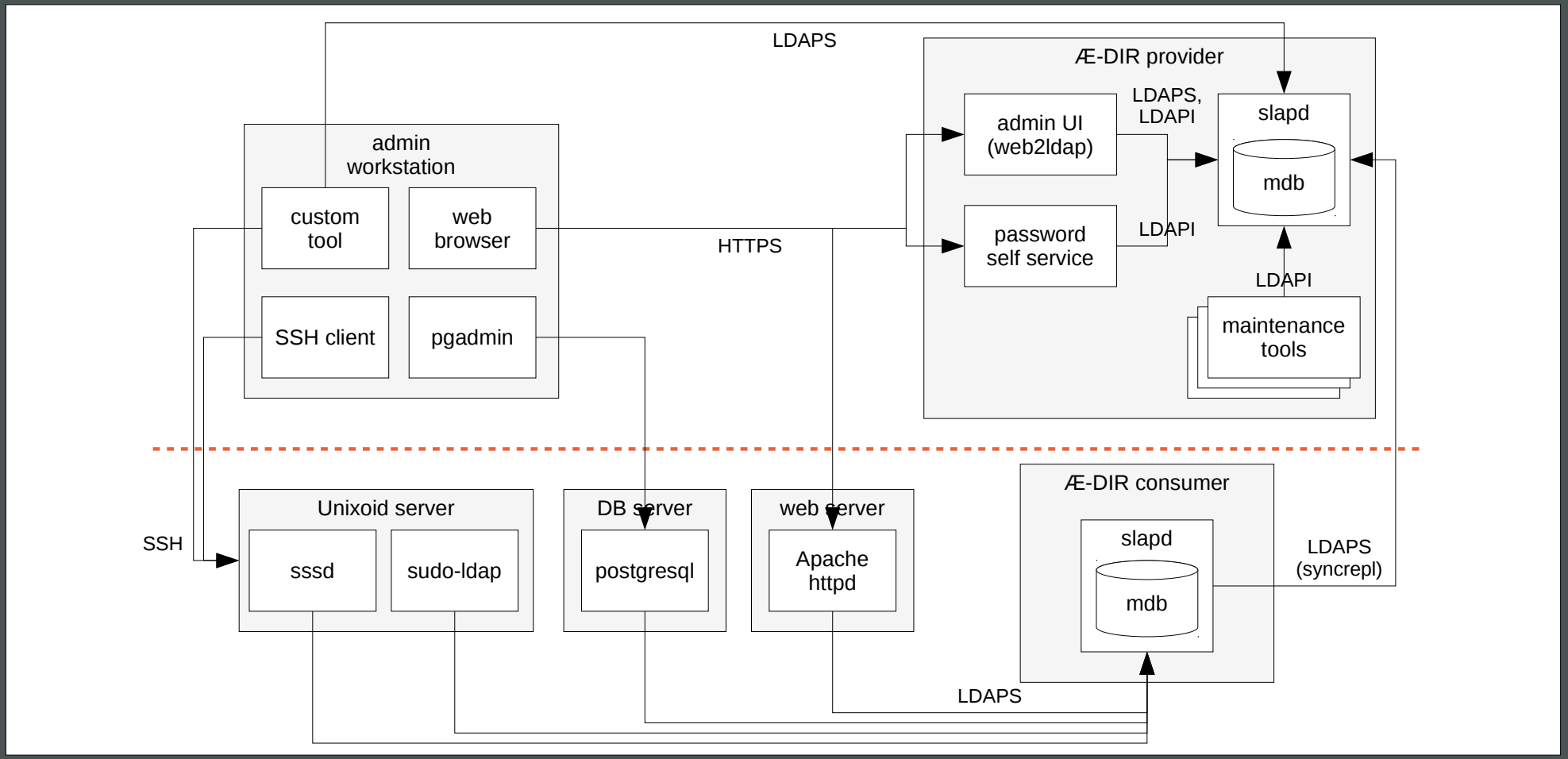
Paradigmen (1)

- Explizit ist besser als implizit
- Keine sichere Autorisierung ohne sichere Authentifizierung
- Keine anonymen Zugriffe
- Individuelle Authentifizierung
- Keine allmächtige Stellvertreterrollen
- Rechtevergabe immer basierend auf Gruppenzugehörigkeit

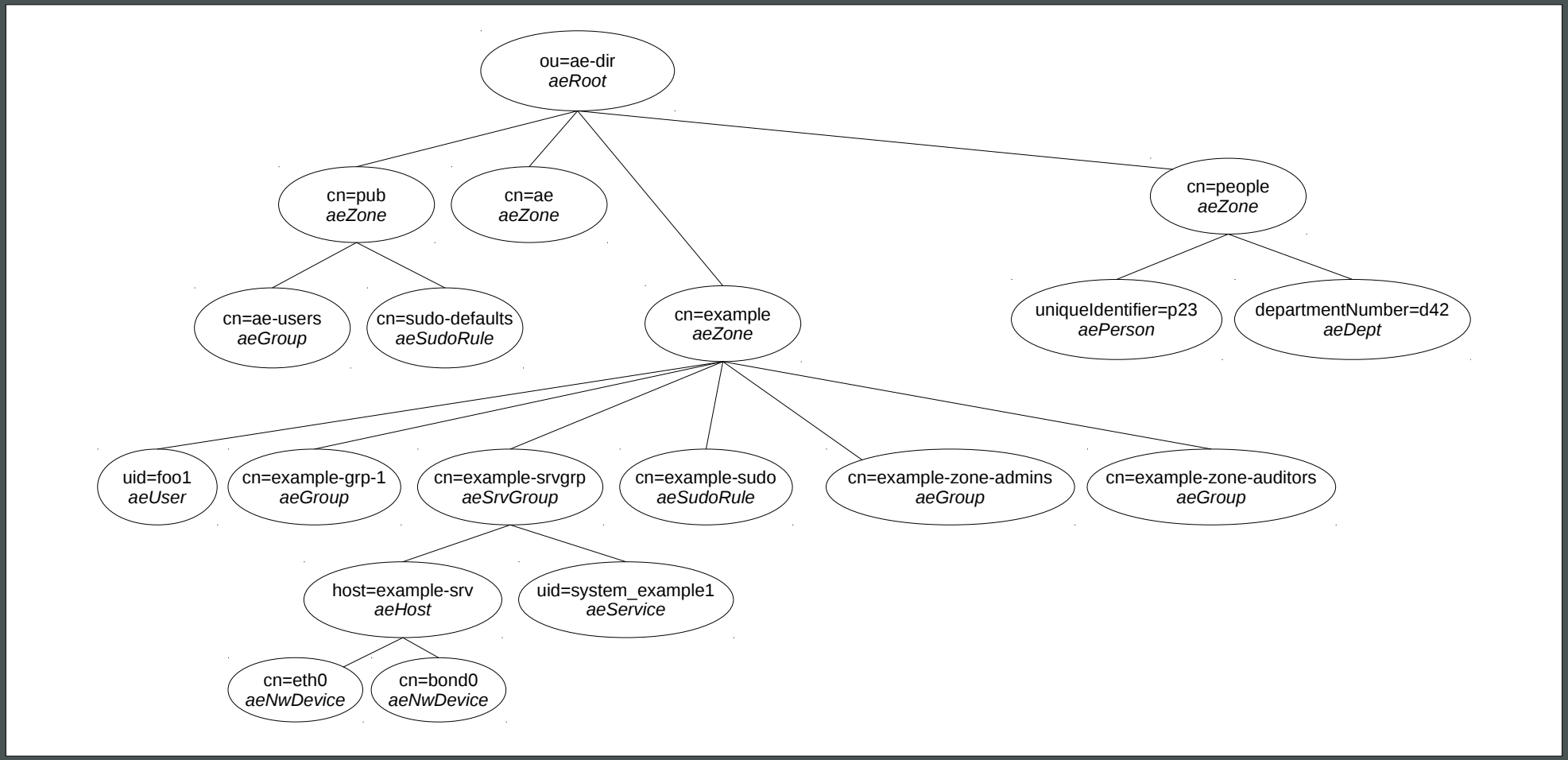
Paradigmen (2)

- Keine hierarchische Struktur erforderlich
- Eine Person ist kein Benutzer
- Rollentrennung mit mehreren Benutzern je Person
- Persistente IDs (keine Wiederverwendung)
- Nur verschlüsselter Netzwerkverkehr (TLS und SSH)
- Wohldefinierte Semantik und Syntax aller Objekte und ihrer Attribute (besser keine Daten als schlechte Daten)

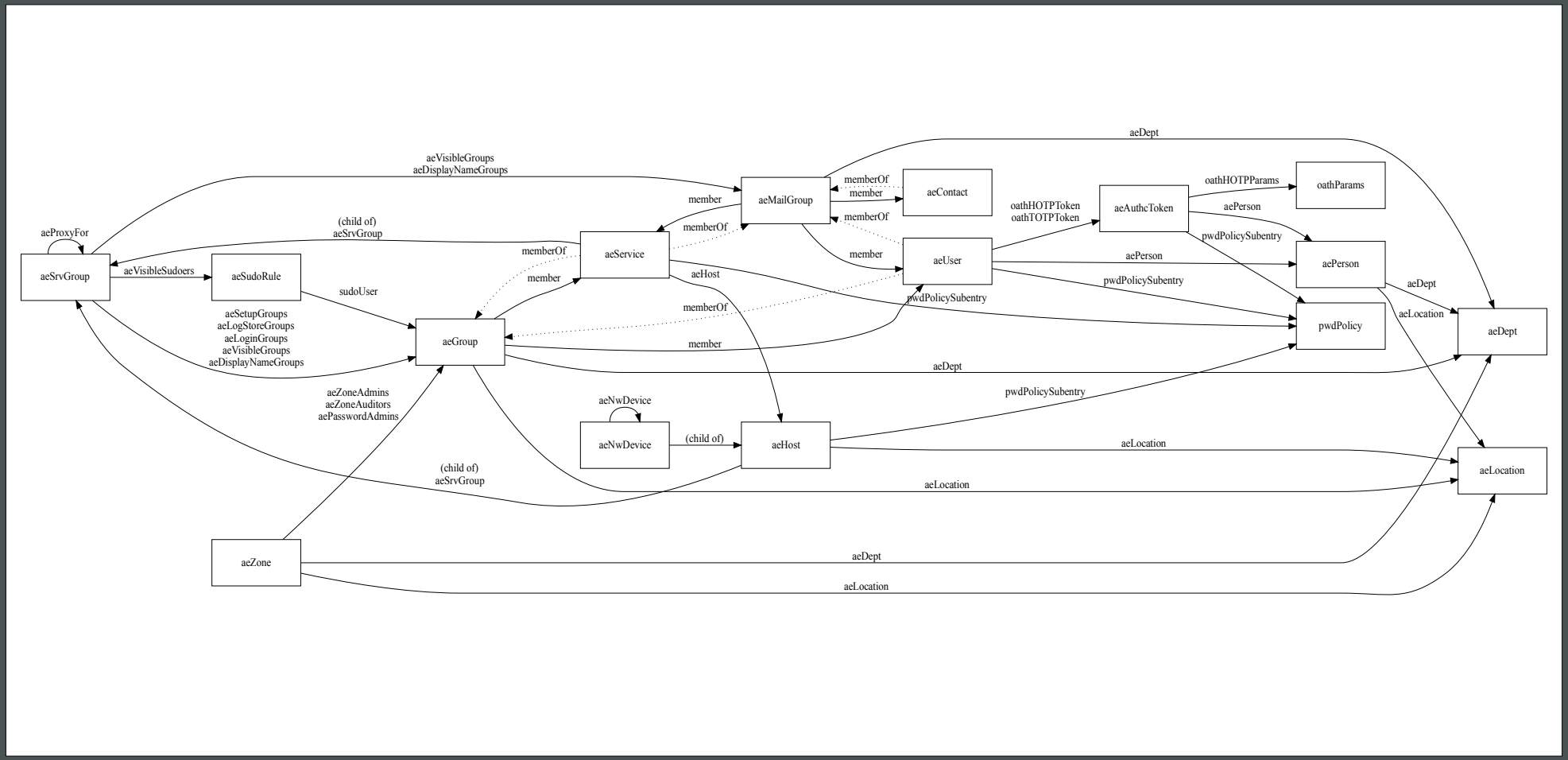
2-stufige Architektur



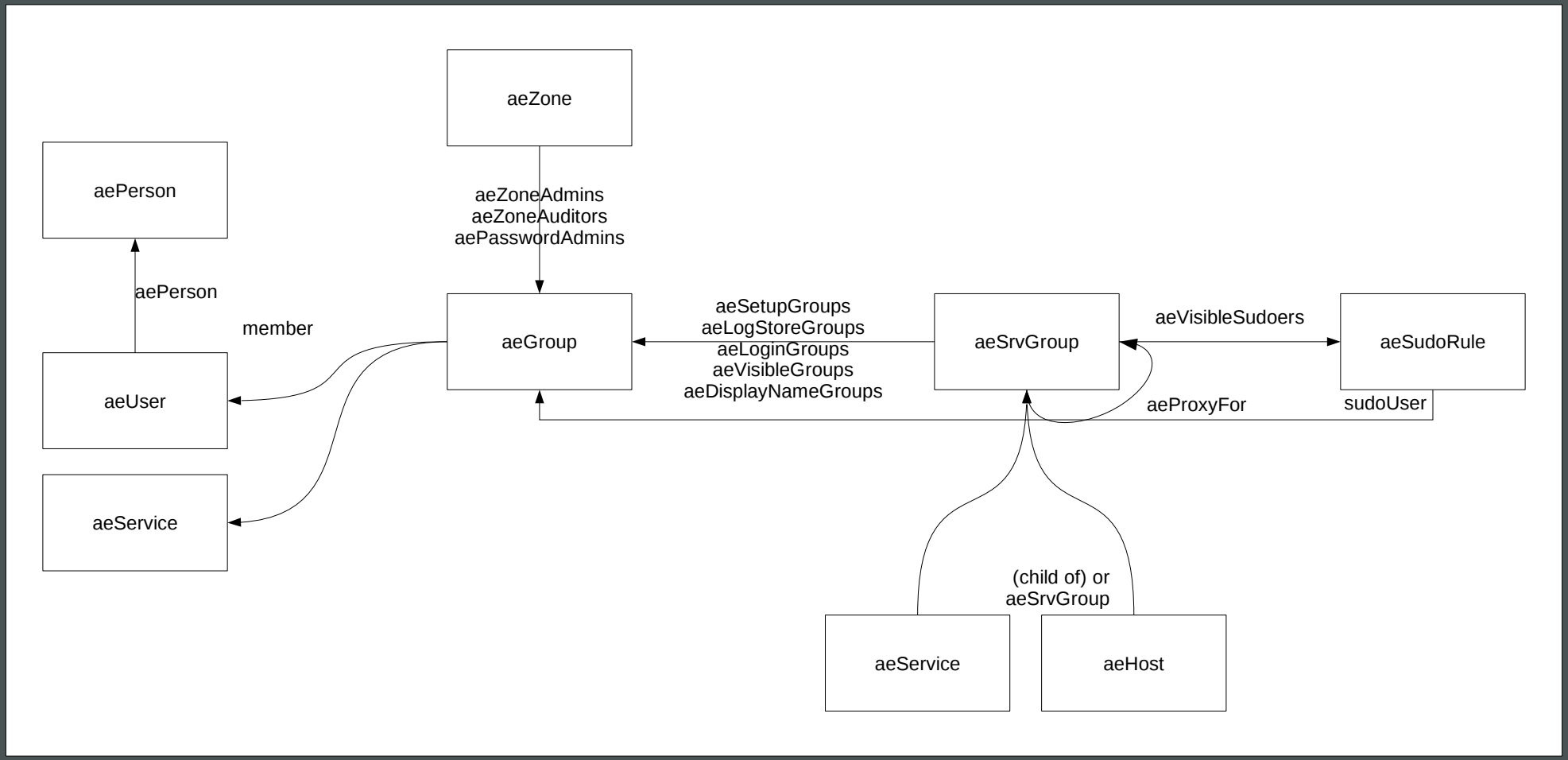
2-stufige Architektur



Entitäten (EER vollständig)



Berechtigungsbeziehungen (EER Authz)



Installation Æ-DIR Server

- *ansible*-Rolle zieht alle Dienste hoch
- Basiskonfiguration separat
- site-spezifische Variablen anpassen
- Kommentare in lesen!
  ansible/roles/ae-dir-server/defaults/main.yml
- site-Verzeichnis anlegen, siehe *ansible/example/*
- *ansible*-Rolle zieht immer wieder alles glatt

Sicherheitsbarrieren -- Defense in Depth

- Sichere Voreinstellungen, keine Dienstepassworte!
- Mit sich selbst integriert
- Separate Dienste, Unix Domain Sockets (Peer Credentials)
- *systemd*-Optionen zur Härtung (mount points etc.)
- Strikte *AppArmor*-Profile für alle Dienste
  (optional, *targeted* und nur für SUSE und Debian)
- 2-Faktor-Anmeldung: yubikey basierend auf *OATH-LDAP*
- Bald für Apache: Regelsatz für *mod_security*

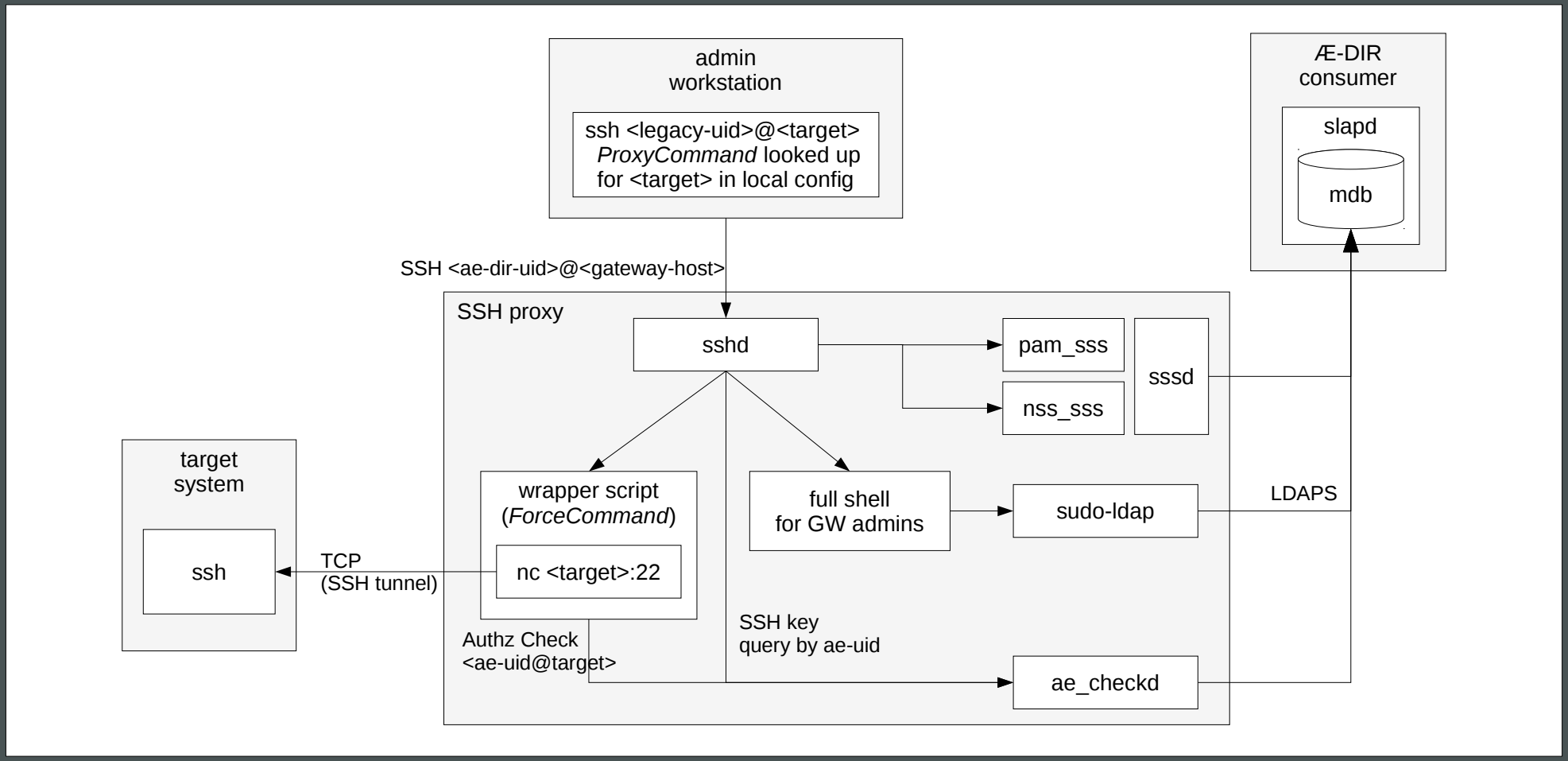
Client-Konfigurationen

- siehe client-examples/
- postfix/dovecot: Mailboxen, Mail-Verteiler
- Apache
- FreeRADIUS
- sshd
- MacOS getestet!
- Linux-Integration mit sssd oder nss-pam-ldapd (nslcd): Fork *ansible/roles/ae-dir-linux-client/*

In Arbeit: aehostd

- Simpler, angepasster Host-Demon kennt Datenmodell
- noch weniger Client-Konfiguration
- optimierte Suche nach Benutzern und Gruppen
- virtuelle Gruppen (Benutzer-GID, Rollengruppen)
- LDAP Session Tracking Control f. besseres Logging
- *hosts* map
- sudoers-Dateien via *cvtsudoers* (sudo 1.8.23+)
- weniger Code, verschlankter *pynslcd(8)*

SSH Proxy mit Autorisierung



Fazit

- Security by Design ist möglich
- Ja, ist auch anstrengend
- Benutzer brauchen Starthilfe und schlüssige Erklärungen
- Rückhalt durch Führungskräfte sehr hilfreich (Budget!)
- Ursprüngliche Sicherheitsversprechen nicht brechen
  => vor Änderungen immer gründlich nachdenken

Ausblick

- Weitere Ideen reichlich vorhanden
- Engere Integration
  - DevOps (ansible, puppet, o.ä.)
  - X.509 PKI für Server-Zertifikate, SSH Key Signing
  - WebSSO-Integration
  - Netzwerk Access Control
  - Deployment-Infrastruktur (PXE, TFTP-Boot, DHCP, Sys-DNS)
- Dokumentvorlagen für Compliance-Standards

Links

- Doku:
  https://ae-dir.com
- Spielt mit rum!
  https://ae-dir.com/demo.html
- OATH-LDAP:
  https://oath-ldap.stroeder.com

:-/

? … !