Æ-DIR - Authorized Entities Directory

openSUSE Conference 2019

2019-05-26

Michael Ströder <michael@stroeder.com>

- Freelancer
- Topics the last 20 years
  - Identity & Access Management, Directory Services (LDAP)
  - Single Sign-On, Multi-Factor Authentication
  - PKI (X.509, SSH), Applied Crypto
- Open Source / Free Software:
  Æ-DIR, OATH-LDAP, web2ldap

General Security Requirements

- Principles
  - Need-to-know
  - Least Privilege
  - Separation of Duties
- Delegated administration of manageable small areas
- Meaningful audit trails
- Compliance checks

Secure DevOps

- Well-defined security policies
- Teams / Projects
- Network separation (infra, frontend, middleware, backend)
- DevOps staging environments
  - dev: all devs have full control
  - test: some devs have debug capabilities
  - prod: full access only for ops, temporary access for devs

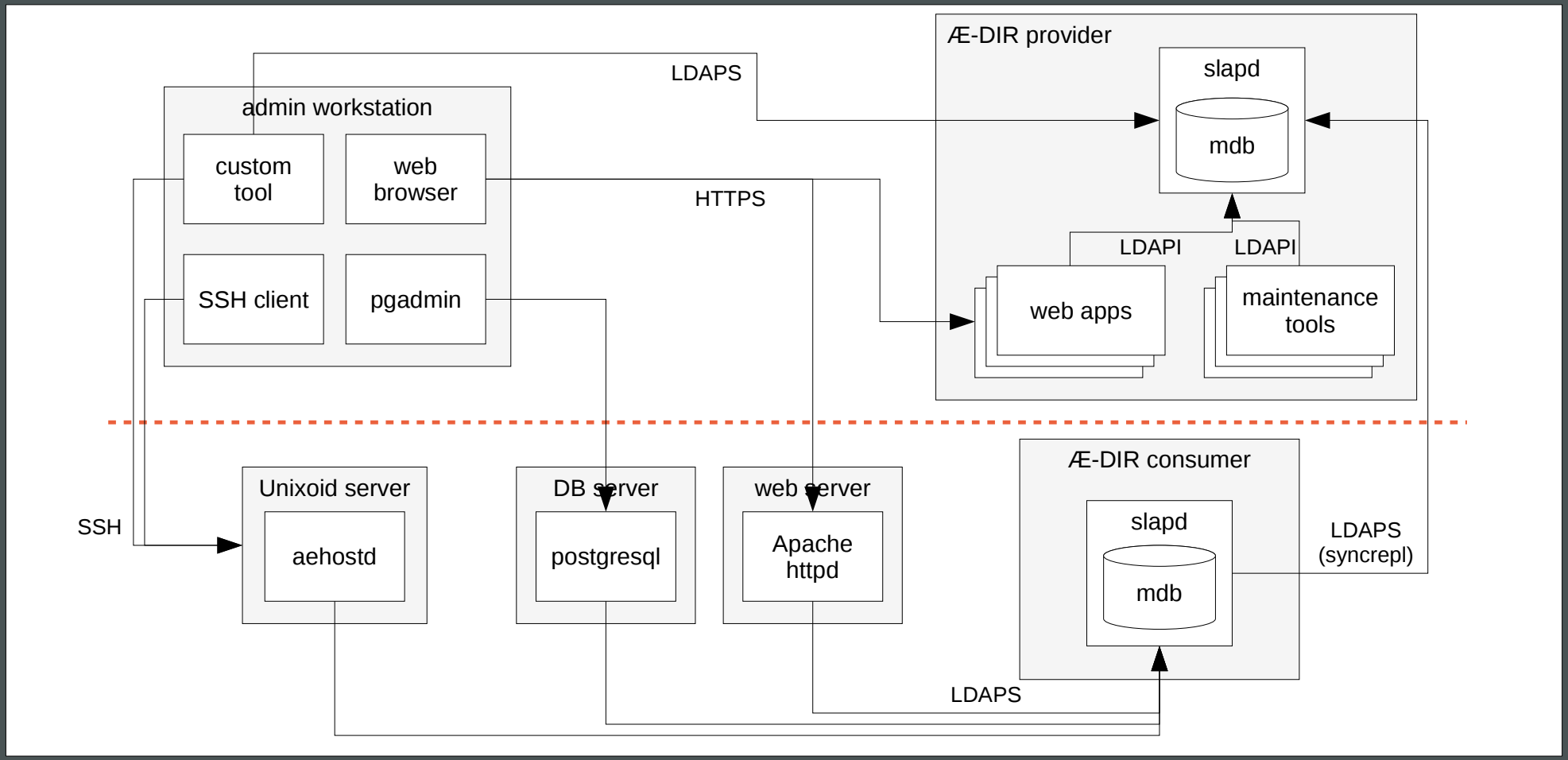
Agile DevOps

- Waiting for workflow progress is not agile .....
- Someone authorized to decide should simply do it
- Let's eliminate workflows!
- Fine-grained authorization is needed for data maintenance
- Constraints to prevent false input
- Same rules for web UI and LDAP write access

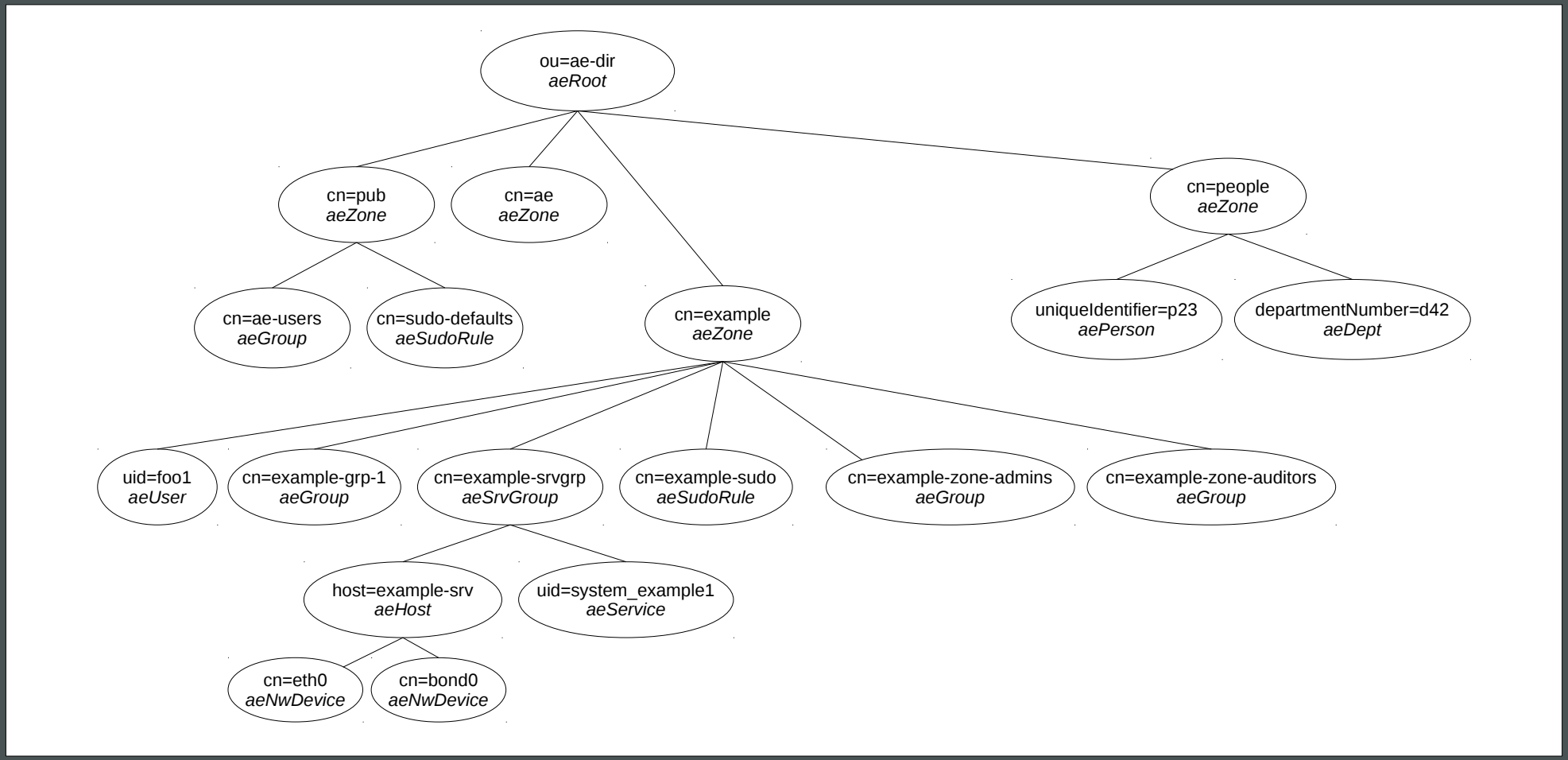
Paradigms

- Explicit is better than implicit
- Secure authorization requires secure authentication
- Avoid all-mighty proxy roles and workflows
- Do not assume hierarchical structure
- A person is not an user account
- Multiple user accounts per person
- Persistent IDs (never re-used) for reliable audit trails

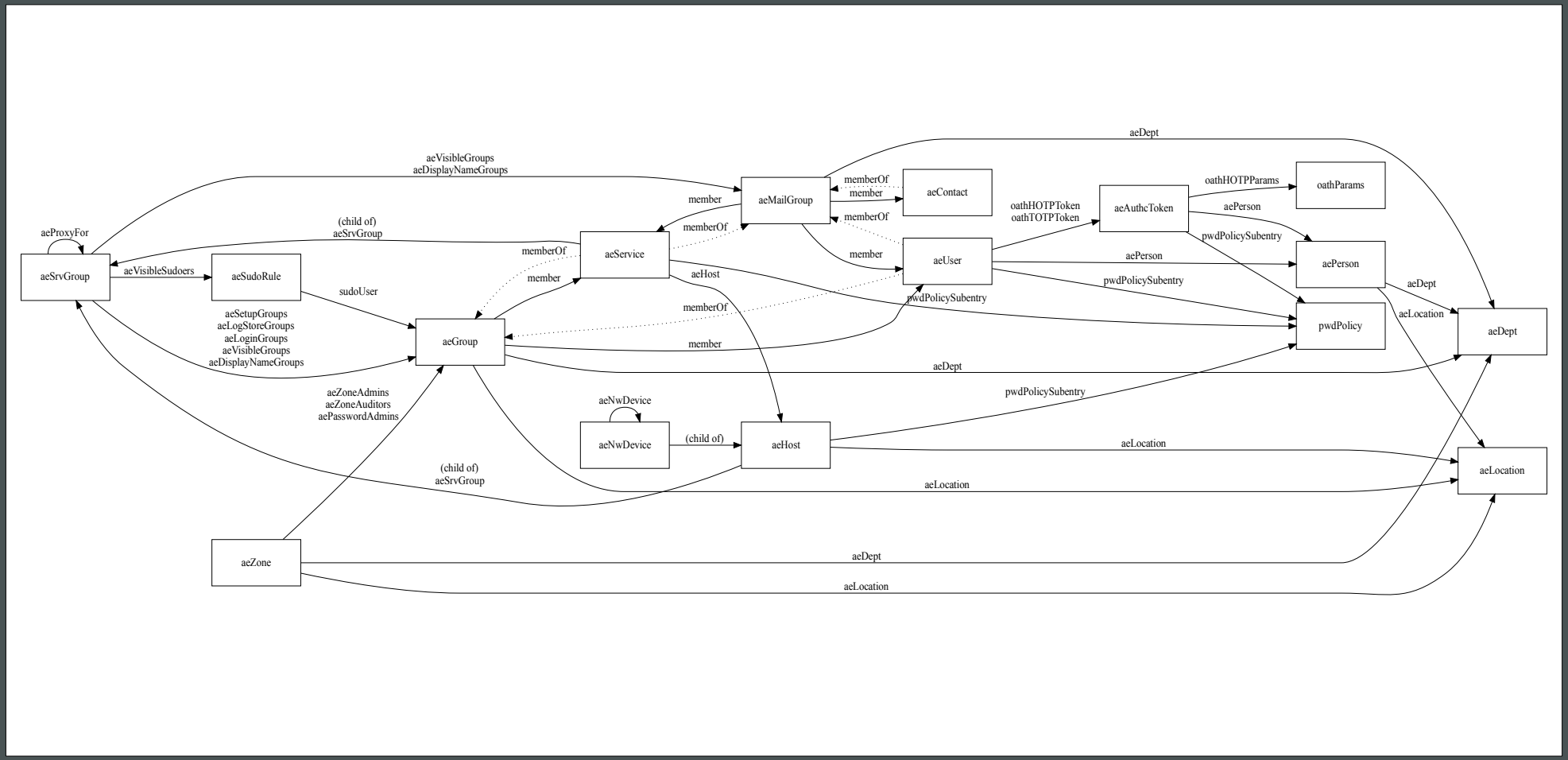
2-tier architecture



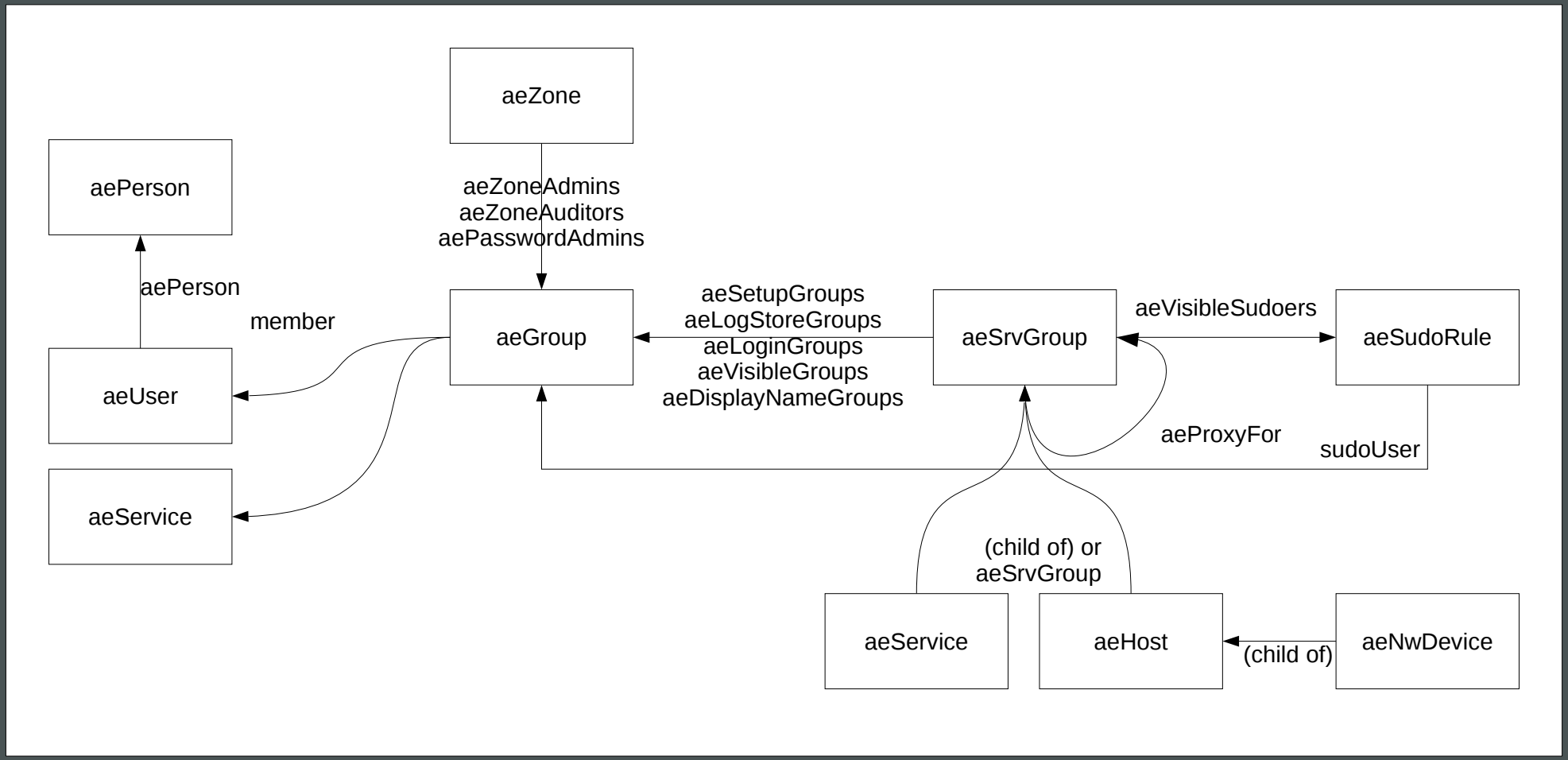
Directory Information Tree (DIT)



Entity relationships



Entity relationships for access control



Roles

- Æ admins delegate zones, fix broken entries,
  but they do **not** maintain zones
- Æ auditors may read (almost) everything
- Zone admins are the maintainers doing the daily work
- Zone auditors may read anything within a zone
- Setup admins maintain hosts/services within service groups
- Users may read own entries, see members of own groups,
  change own password

Schema basics

- Based on standard schema (inetOrgPerson, RFC 2307, ..)
- Definitions prefixed with “ae”
- Æ-DIR schema not used in client configuration
- Common meta data in abstract object class *aeObject*
- Hybrid group class aeGroup (multiple inheritance):
  - *memberUid* (RFC2307) and *member* (RFC2307bis)
  - empty groups based on *groupOfEntries*

Unique identifier

- Unique user and system IDs already needed for secure authorization
- Many unique constraints enforced:
  uid, uidNumber, gidNumber, cn, aeFqdn, macAddress…
- Adding an entry means claiming a unique ID
- Existing unique ID “owned” by zone admins
- Also connects system IDs to delegated administration, useful for deployment, DNS/DHCP, NAC, PKI, etc.

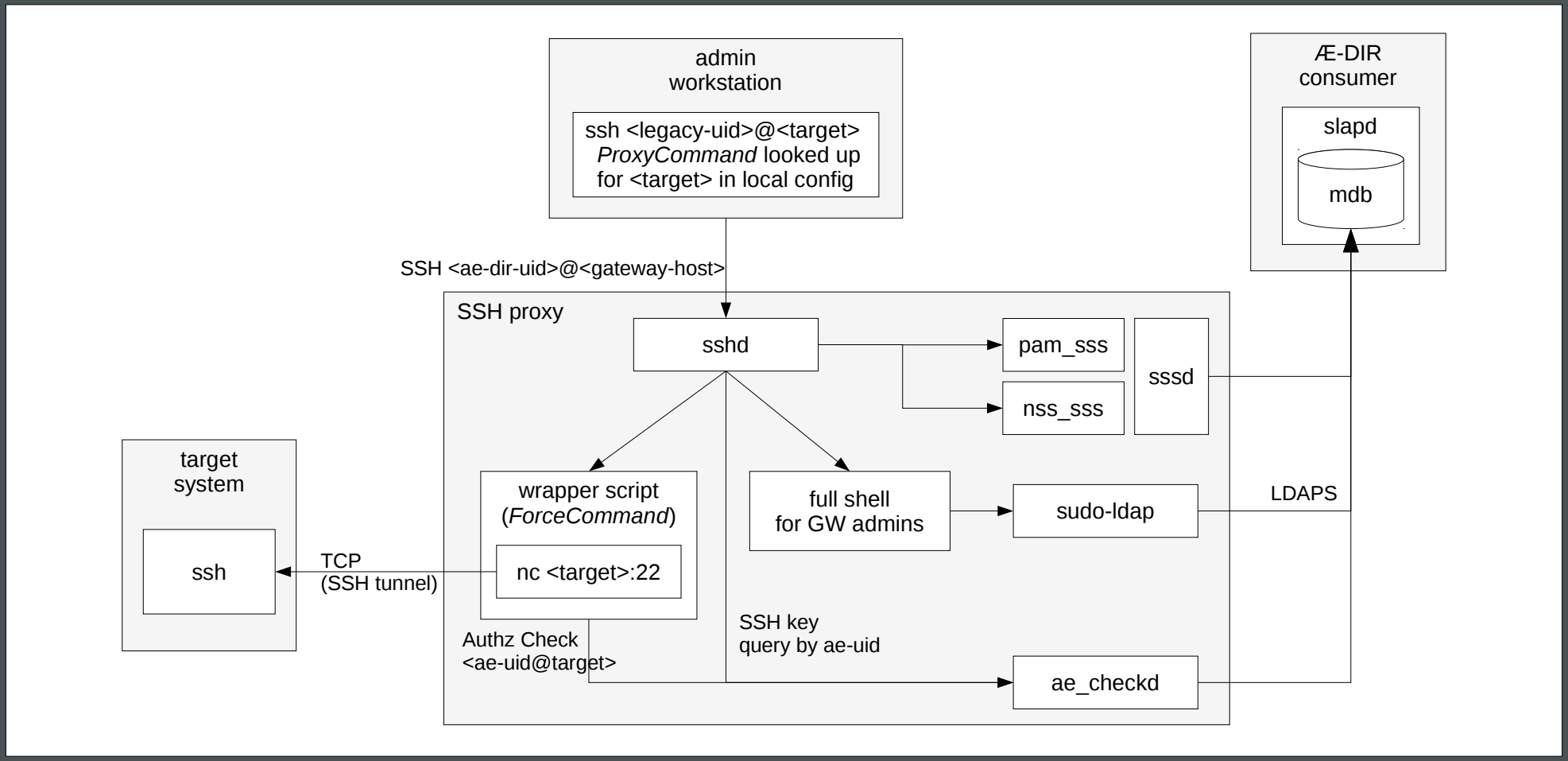
Installation Æ-DIR server

- *ansible* role installs replicas and all services
- base configuration to be done separately
- site-specific ansible variablen
- Read the comments!
  ansible/roles/ae-dir-server/defaults/main.yml
- Create site directory, see *ansible/example/*
- If things went wrong ansible role corrects it

Defense in Depth

- Secure defaults
- Self-contained (zone *ae*)
- Service separated, Unix domain sockets (Peer Credentials)
- *systemd.exec(5)* options for hardening (mount points etc.)
- Strict *AppArmor* profiles for all services
  (optional, *targeted* and only for SUSE and Debian)
- 2-faktor-authc: yubikey based on *OATH-LDAP*

SSH proxy authz



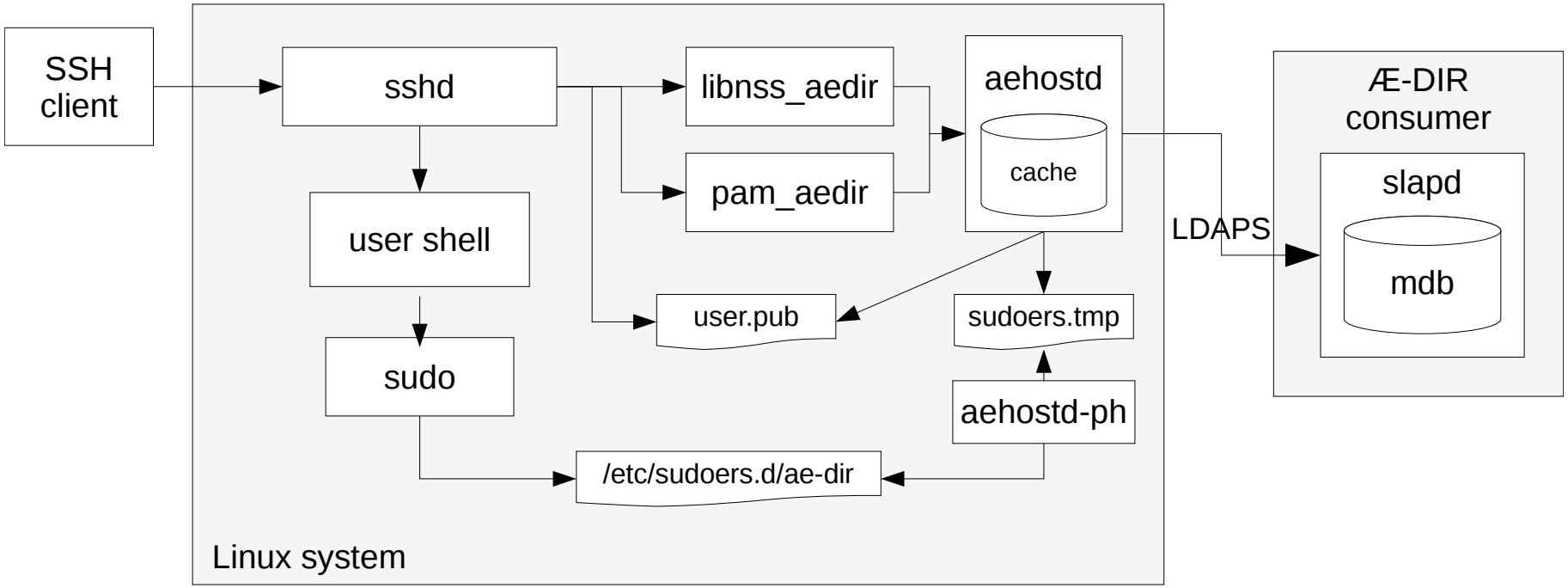
aehostd - Why?

- Æ-DIR's slapd burns CPU cycles with set-based ACLs
- Better automated enrollment needed (host password)
- *sudo-ldap* causing lots of parallel TLS connections
- Connection behaviour unpredictable
- LDAPI support for NSS/PAM on Æ-DIR servers
- Fed up by asking others for simple features

aehostd - Goals

- Better performance
- Better behaviour for lots of NSS clients:
  - Load-balancing
  - Update timing
- Enrollment automation with pseudo SSH login
- Simple! Less configuration, less code, less dependencies, less privileges

aehostd / aehost-ph



aehostd - Specific Features

- Virtual groups:
  - primary user GIDs
  - role groups
- Syncing of SSH authorized keys
- LDAP session tracking control for better logging
- *hosts* map based on *aeNwDevice* entries
- Enrollment via pseudo login with password

```
ssh aehost-init@host.example.com
```

Conclusion

- Security by design is possible
- Yes, it’s painful sometimes
- Admins need help in the beginning
- Backing of management helps (budget!)
- Don’t break former security promises later!
  → think twice or more before changing something

Links

- Docs:
  https://ae-dir.com
- Play with it!
  https://ae-dir.com/demo.html
- OATH-LDAP:
  https://oath-ldap.stroeder.com

:-/

? … !