Photo by James Eades on Unsplash

mLinux: Building a Linux Client at Siemens

Roger Meier (backup for Felipe Herrera Martinez) & Markus Legner

Open Source @ Siemens, 2022-05-18

SIEMENS

Why even talk about this at an open-source event?

SIEMENS

What is special about mLinux?

Closed-source software	Open-source software
Traditional IT services	

SIEMENS

Linux at Siemens

Why do we need an official Linux client?

What are the challenges?

SIEMENS

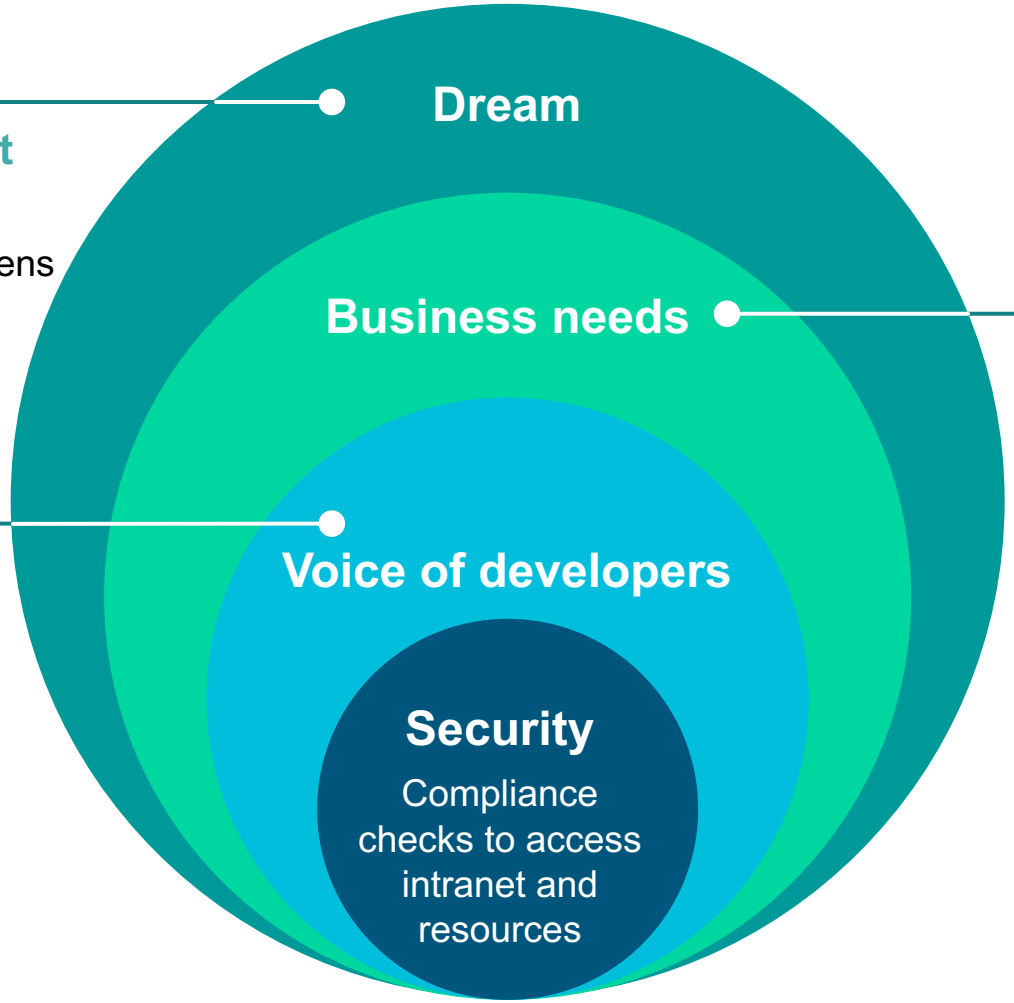
Why do we need an official Linux client?

Make Linux a real user choice at Siemens

- Facilitate the usage of Linux at Siemens
- No exceptions needed
- Same user experience as other OS choices at Siemens

Why must I feel like a hacker when I want to use Linux?

- Other OSes (Windows, MacOS) are ready to use
- Complex and tedious process to get a Linux for daily work at Siemens.
- Lots of exceptions required
- Missing collaboration functionalities available in other OS options



Linux is part of our core business

- Align efforts and create synergies between business and IT
- Thousands of products/solutions are based on Linux
- Eat our own dogfood: use what we sell
- Free choice of OS for developers

Our vision: make Linux a 1st class citizen at Siemens!

SIEMENS

What are the challenges?

Proprietary software/hardware

- Siemens uses proprietary software to achieve security and enable collaboration
- Limited Linux support for many of these applications
- Sending and receiving encrypted/signed emails requires support for PKI and smart cards

Security requirements

- Access to company network and intranet require certificates and application of security rules
- Some security tools do not support Linux (or only some distributions)
- Identity and access management based on Microsoft (Azure) Active Directory

Diverse user preferences and use cases

- Every user has their favorite distribution and applications
- Broad spectrum of users:
  - Some just want a running system
  - Others want to choose exactly which software is installed
- → **Unmaintainable without community contributions and open-source tools**

Onboarding without intranet access to support remote work scenarios

SIEMENS

Principles

Open development
Automatization
Open standards
Upstream contributions

Image by John Martinez Pavliga (CC BY 2.0)

SIEMENS

Open development, planning, and community interactions → IT service as inner-source project

Planning and development open on code.siemens.com

- Milestones for monthly and long-term planning
- Issues to track new features and bugs
- All code is available in the GitLab project and reviewable by anyone
- Everyone can weigh in on decisions

Community interactions on code.siemens.com

- **No separate IT helpdesk**
- Bugs and **support on the issue tracker** → handled by the development team
- All documentation built with MkDocs and served via GitLab pages

Benefits: scalability through an active and growing community of mLinux users

- Many users are **Linux experts**
- Provide **valuable feedback**, bug reports, and improvement suggestions
- Users help with debugging, add documentation, and answer other users' questions

SIEMENS

Automatization based on GitOps

Merge requests for all changes

- Require approval and merge by second team member
- **Automatic linting and testing** with GitLab CI ensure quality of changes
- Automatically create Puppet environment for new features and bug fixes to enable tests

Automatic deployment and user management

- Default branch “production” is **automatically deployed** via CI job
- Users are **automatically validated** and added to relevant Active Directory (AD) groups

Benefits: transparency and reliability despite agility

- Changes are **transparent and reviewable**
- Deploy new features as soon as they are completed
- Ensure the latest version is always functional and **no new bugs are introduced**

SIEMENS

Use open standards and open-source tools

Device management and monitoring via Puppet

- Puppet provides all features required for device management and compliance checks
- Puppet code is open-source
- Puppet enterprise just provides additional convenience features

Use existing (Linux) tools wherever possible

- OpenSC for smartcard support
- Evolution for signed and encrypted emails
- p11-kit for company-internal certificates
- Puppet modules by dev-sec.io for OS hardening
- Clevis for automatically unlocking encrypted disk

Benefits: reusability and access to resources

- Solution **can be reused** in other contexts
- Debugging is much easier if source code is available
- **Fixing bugs and adding features** is possible without having to wait for the vendor
- Often better documentation and **support by the community**

SIEMENS

Contribute upstream instead of creating forks and patches

Upstream first: do not create forks or local patches

- Report bugs to the upstream open-source project
- Contribute important features or bug fixes ourselves

Rationale: give back to the community

- We are using other people's work for free → **let's provide something in return**
- Other people may have the same problems and require the same features as we

Benefits: maintainability and code review

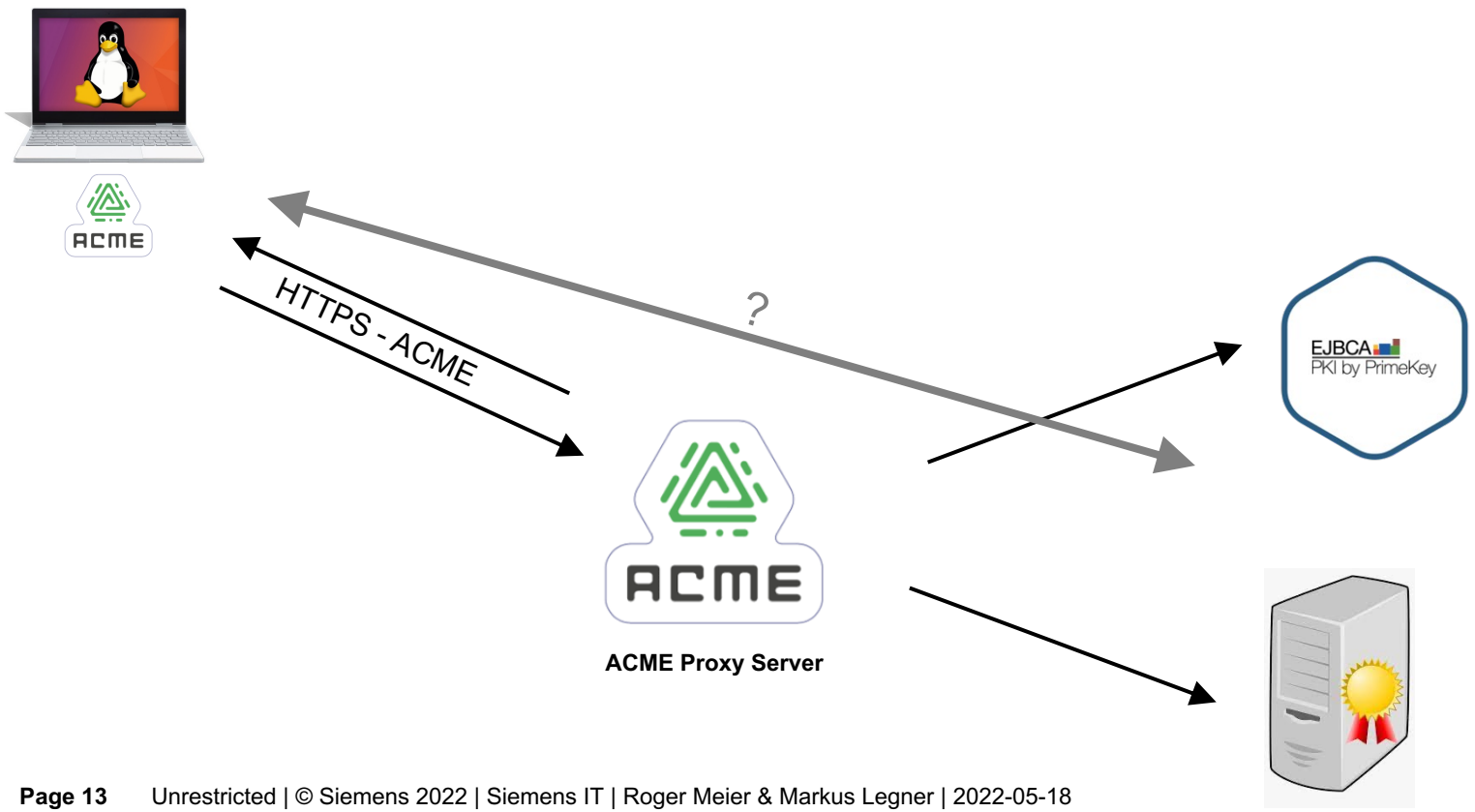
- Much **easier to maintain**: directly use the upstream code / containers
- Additional input and **improvements** on our own code

SIEMENS

Case Study: Certificate Life-Cycle Management

SIEMENS

Managing device certificates using open standards



Goal: secure system to issue device certificates based on **open standards**

Problem: multiple backend systems that use **different (proprietary) protocols**

Solution: ACME proxy server (acme2certifier) provides **an ACME façade**:

- serves the ACME protocol for clients
- interacts with backend servers to issue certificates

SIEMENS

ACME: an open standard for automatically managing X.509 certificates

ACME does **not** refer to the fictional corporation that features prominently in the Road Runner/Wile E. Coyote cartoons

The Automatic Certificate Management Environment (ACME) was developed by the Internet Security Research Group (ISRG) to automatically provide X.509 certificates in their free service *Let's Encrypt*

Standardized by IETF in 2019 as RFC 8555

Allows web servers to request certificates from CAs, who validate domain ownership through standardized **HTTP or DNS challenges**

Image from Wile E. Coyote and the Road Runner (Warner Bros.)

ISRG

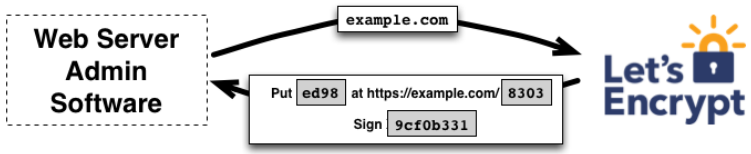
Let's Encrypt

Image source: ISRG, letsencrypt.org

SIEMENS

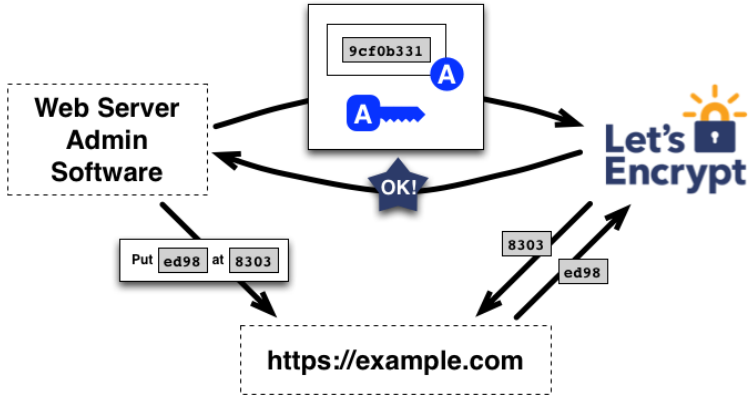
RFC8555 (ACME): an open standard for automatically managing X.509 certificates

Client onboarding



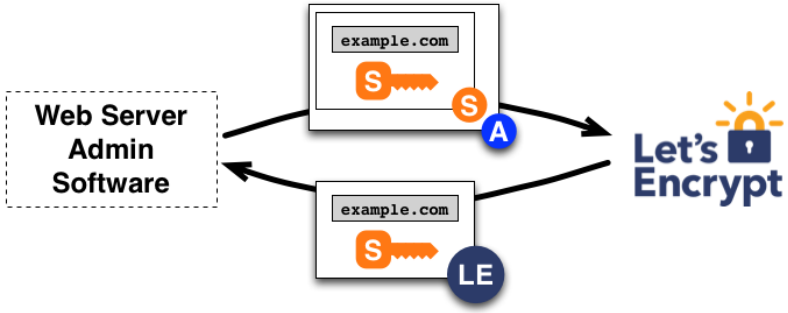
1. Client: create account key pair and submit order for a new certificate
2. Server: **issue a DNS or HTTP challenge** for the client to demonstrate control over their domain (example.org)

Proof of domain ownership



1. Client: add the challenge in a DNS TXT record (or serve it via HTTP) to **prove its control over the domain** (_acme-challenge.example.org. 300 IN TXT "gfj9Xq...Rg85nM")
2. Server: **verify** that the correct entry has been added

Issuance and renewal

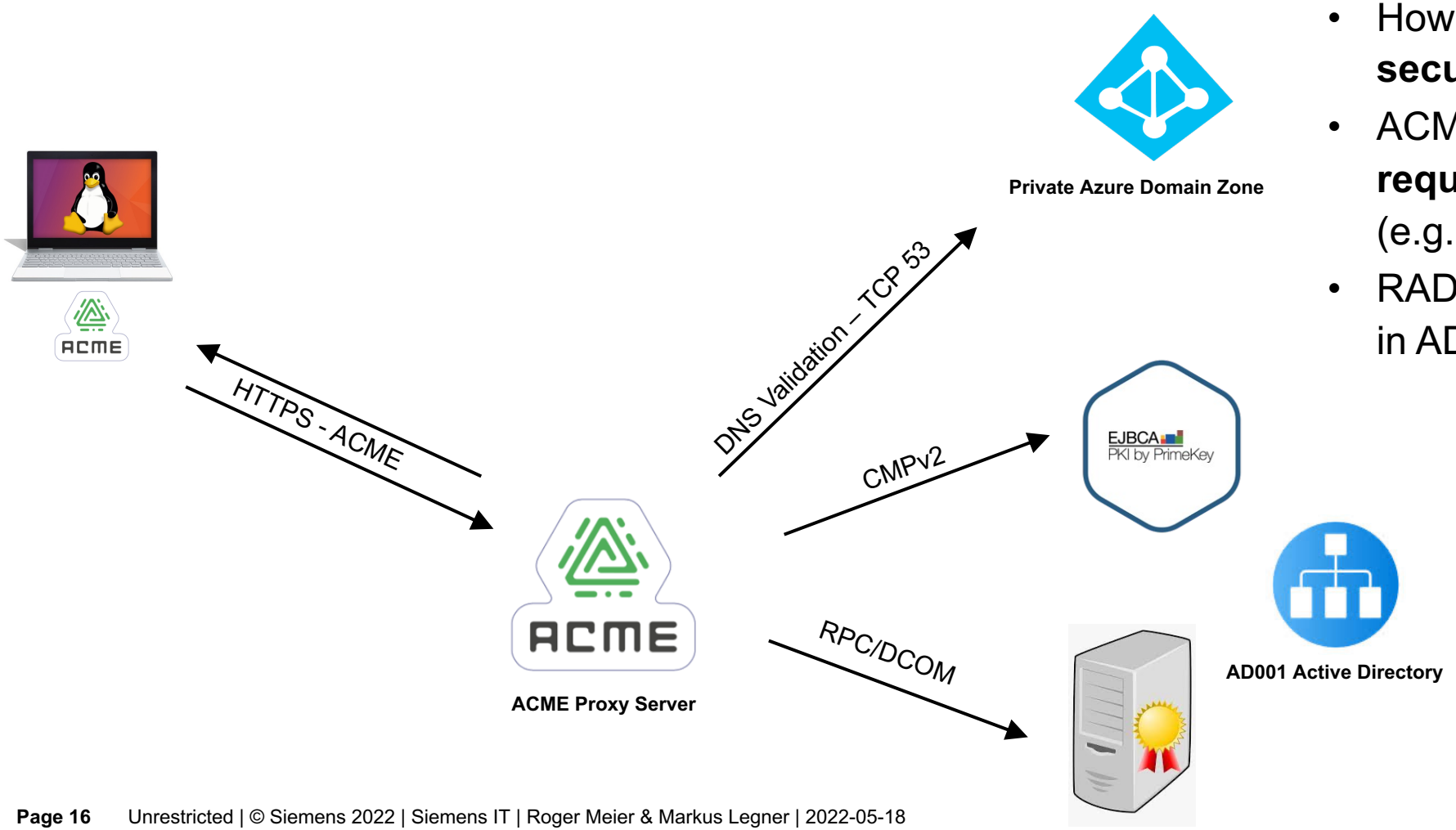


1. Client: **send a certificate signing request** (CSR)
2. The CSR is signed by the corresponding private key and the client's account key
3. Server: verify both signatures, **issue a certificate** for the authorized domain

Image source: letsencrypt.org

SIEMENS

Using ACME in the context of mLinux to issue device certificates

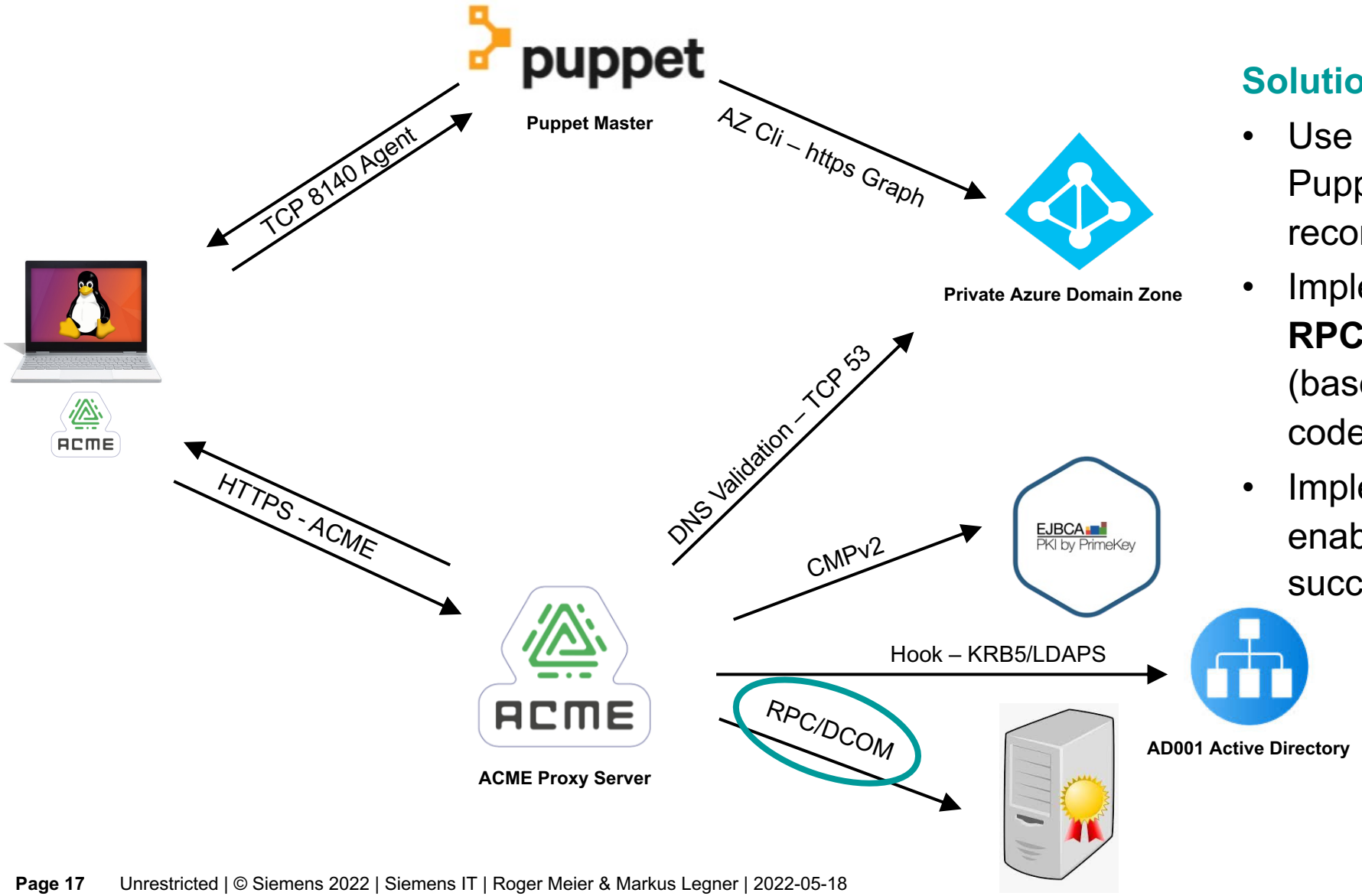


Challenges:

- How to **fulfill DNS challenge in a secure way**?
- ACME proxy **doesn't support all required protocols** (e.g., RPC/DCOM)
- RADIUS **requires device object** in AD

SIEMENS

Using ACME in the context of mLinux to issue device certificates



Solutions:

- Use **existing secure channel** to Puppet master to create DNS record
- Implement **new handler for RPC/DCOM** communication (based on existing open-source code)
- Implement **hooks feature** to enable AD object creation after successful certificate issuance

SIEMENS

Current State and Future Work

SIEMENS

What have we learned while building mLinux?

Building an IT service based on open-source tools and workflows has many benefits

- Feedback and insights from the community
- Developers appreciate direct communication channels with maintainers
- Improve scalability, maintainability, stability, agility, and transparency

This approach doesn’t solve everything

- Sometimes conflicting requirements and preferences
- Limited resources require prioritizing features
- Challenges with proprietary software remain

Photos by Thomas Oxford and Stephanie LeBlanc on Unsplash

SIEMENS

Current state of mLinux

- In production since end of April: **every Siemens employee can now use this service**
- Support for Ubuntu and Debian
- Approximately 150 users
- Core development team of five people (some only part-time)

Next steps

- **Infrastructure as code** (IaC) for the complete infrastructure
- Support even more use cases and **increase flexibility**
- Further **strengthen cooperation** with Linux community and other Linux projects within Siemens
- **Increase community involvement** in decisions and development
- Further **upstream improvements**

We are hiring, join us!
Senior Architect for managed Linux client (Job-ID 298523)

SIEMENS

Contact

Felipe Herrera Martinez
End Point Protection Architect
felipe.herrera_martinez@siemens.com

Further resources (internal only):

- https://mlinux.siemens.io
- https://code.siemens.com/mlinux/mlinux

Markus Legner
Senior Open-Source Specialist
markus.legner@siemens.com

Roger Meier
Principal Key Expert
Service Owner code.siemens.com
r.meier@siemens.com

SIEMENS