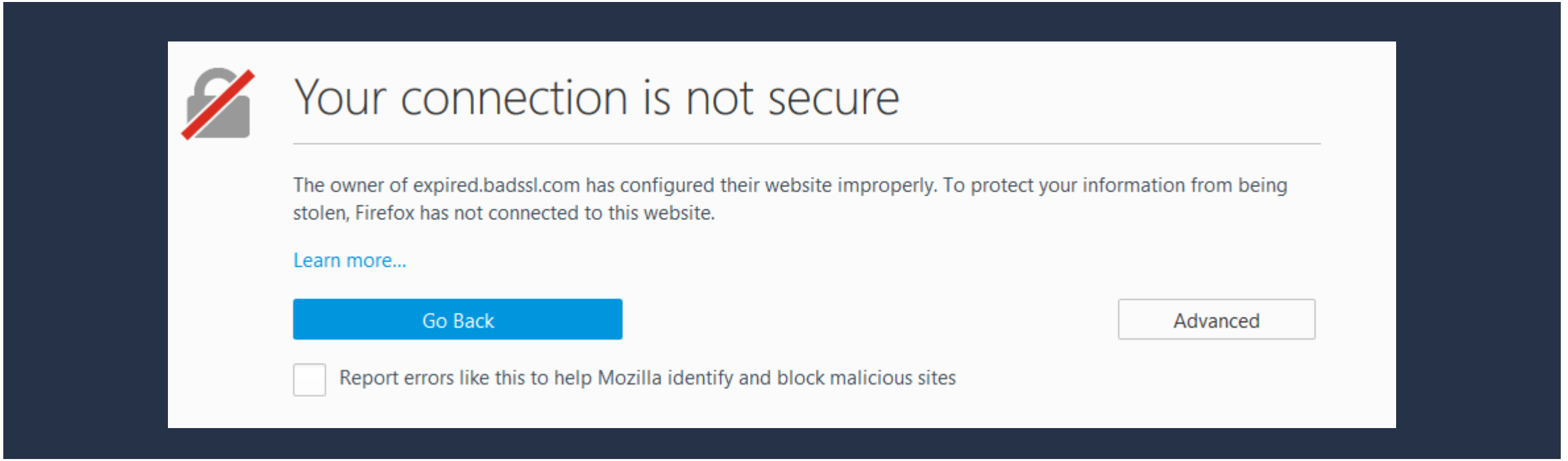
ZERTIFIKAT: SCHON WIEDER ABGELAUFEN ?!?

ZERTIFIKATSNUTZUNG KOMFORTABEL UND SICHER MIT ACME & CO

Andreas Kühne
trustable solutions UG

ZERTIFIKAT: SCHON WIEDER ABGELAUFEN ?!?

ZERTIFIKATSNUTZUNG KOMFORTABEL UND SICHER MIT ACME & CO



WARUM LAUFEN ZERTIFIKATE ÜBERHAUPT AB?
GRÜNDE FÜR ENDLICHE LAUFZEIT

- ‘Abnutzung’ des Schlüssels durch seine Verwendung (historisch)
- Schwächung von Algorithmen (Crypto-Analyse, Hardware-Fortschritte, …)
- Komprommitierung des Schlüssels
  - Unachtsamkeit (Autofill beim eMail-Empfänger)
  - Vorsatz (‚disgruntled employee‘)
  - Angriffe (Ransomware inkl. Disclosure-Androhung)
  - Soft- und Hardware-Bugs (z.B. ROCA, HeartBleed)
- Re-Validierung der Zuordnung
- 'Abzocke‘ durch CAs

WOZU BRAUCHEN WIR ZERTIFIKATE ÜBERHAUPT?

- Zertifikate sind zentrale Bausteine einer PKI
- Einfache Art, Vertrauensbeziehungen herzustellen
- Zertifikate binden öffentlichen Schlüssel an eine Identität (z.B. DNS-Namen, eMail-Adresse, Person)
- Signatur auf definiertes Dokument (X.509: TBS-Struktur)
- PKI erlaubt effektive Nutzung von
  - Verschlüsselung
  - Digitale Signatur
  - Authentisierung
  - TLS

ALTERNATIVEN ZUR PKI
‘MAN MÜSSTE DOCH NUR …’

- Direct Trust
  - Hoher manueller Aufwand
  - Fragile Prozesse
- Web of Trust
  - Kollaborativer Ansatz
  - Unzuverlässige Infrastruktur
  - Probleme bei PGP offensichtlich
- Block Chains
  - Teil-Lösung des Trust-Problematik (genutzt z.B. bei TSL)
  - BC-artige Lösungen kranken an Miner-Motivation (‘Wie wird man da reich?’)

TOP-ANWENUNGSFALL ‘TLS’

- Vertrauliche Kommunikation im Internet (**und** Intranet) unabdingbar
  - Personenbeziehbare Daten
  - Medizinische Daten
  - Finanzdaten
  - VS NfD
  - KRITIS
  - BSI Grundschutz
- Let‘s Encrypt stellt 200 Mio. Zertifikate pro Tag aus
- Browser lehnen HTTP-Hosts (bald) ab
- ‚Überwachung‘ im Internet: https://keychest.net/
- **Vorsicht:** TLS Client Zertifikate sind ‚anders‘ !

AUTO-ENROLLMENT-PROTOKOLLE

- ACME
  - Standardisiertes Protokoll: RFC 8555
  - Großes Angebot von Clients (Certbot, acme.sh, …) und Bibliotheken (acme4j)
  - Let‘s Encrypt : Kostenloses Angebot im Internet
- SCEP (NDES, EST)
  - Unter- und Über-standardisiertes Protokoll: IETF (viele Versionen), RFC 8894
  - Wenig explizite Client-Programme, aber gute Java-Bibliothek (jscep)
  - Unterstützung in Geräten integriert (Drucker, Telefonanlagen)
  - Broker(z.B. MDM)
- CEP/CES
  - Windows-Protokoll, SOAP-basiert
  - Eng in AD-Infrastruktur integriert
  - Nutzung ohne AD-Domäne / unter Linux möglich
- K8s et al.
  - Bereitstellung von Zertifikaten und Schlüsseln durch Vaults
  - Wrapping der App durch Proxies
  - Einige Helm-Charts erwarten ACME-Server
  - Ingres / Gateway-Server können ACME-Kandidaten sein

ACME / LET'S ENCRYPT

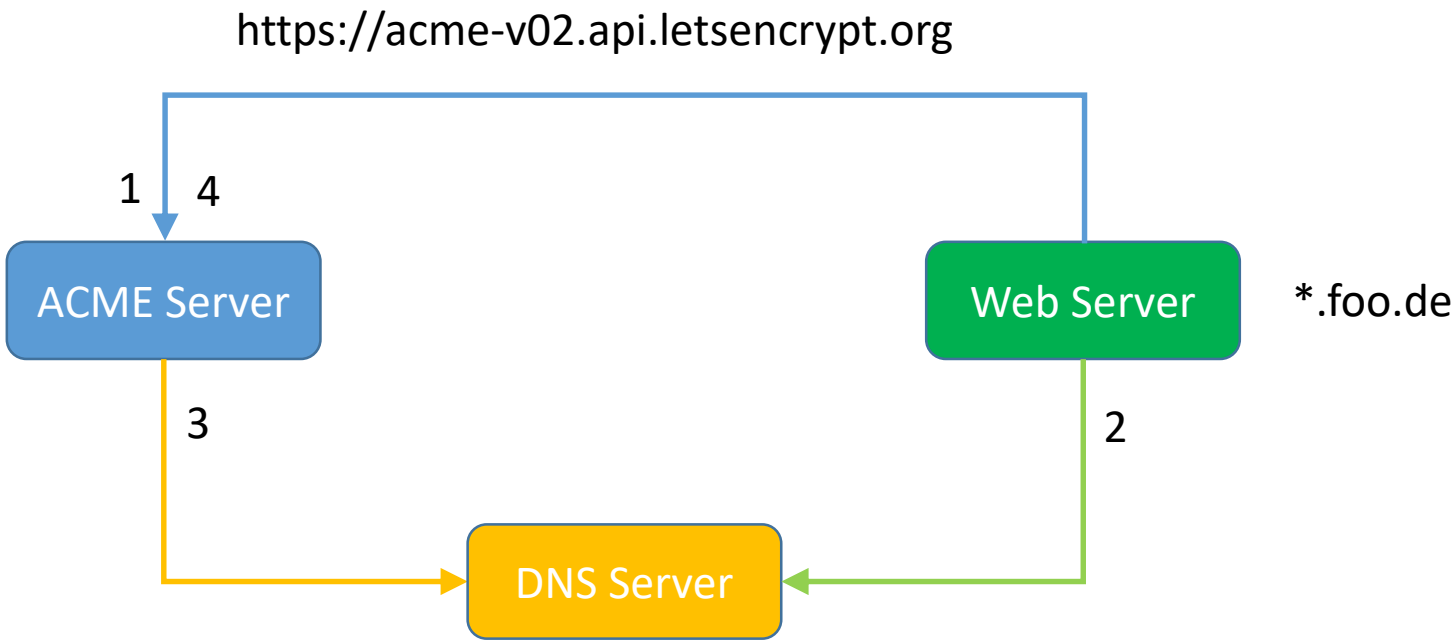
- Let's Encrypt: Kostenlose CA, betrieben von der ISRG (https://abetterinternet.org/)
- LE spezifizierte auch das ‚**Automatic** Certificate Management Environment' (ACME, RFC 8555)
  - Relativ komplexes Protokoll
  - TLS mandatorisch, Payload signiert
- Automatisierbarkeit basiert auf ‚Proof of control':
  - Kontrolle des Webinhaltes der gewünschten Domäne (HTTP Challenge)
  - Kontrolle über den zug. DNS-Eintrag (DNS Challenge), erlaubt Wildcard Zertifikate
- Niedriges Vertrauensniveau (Level ‚Domain Validation')
- Sehr praxis-tauglich! Apache/nginx & Certbot: Install'n'forget

ACME IM INTRANET

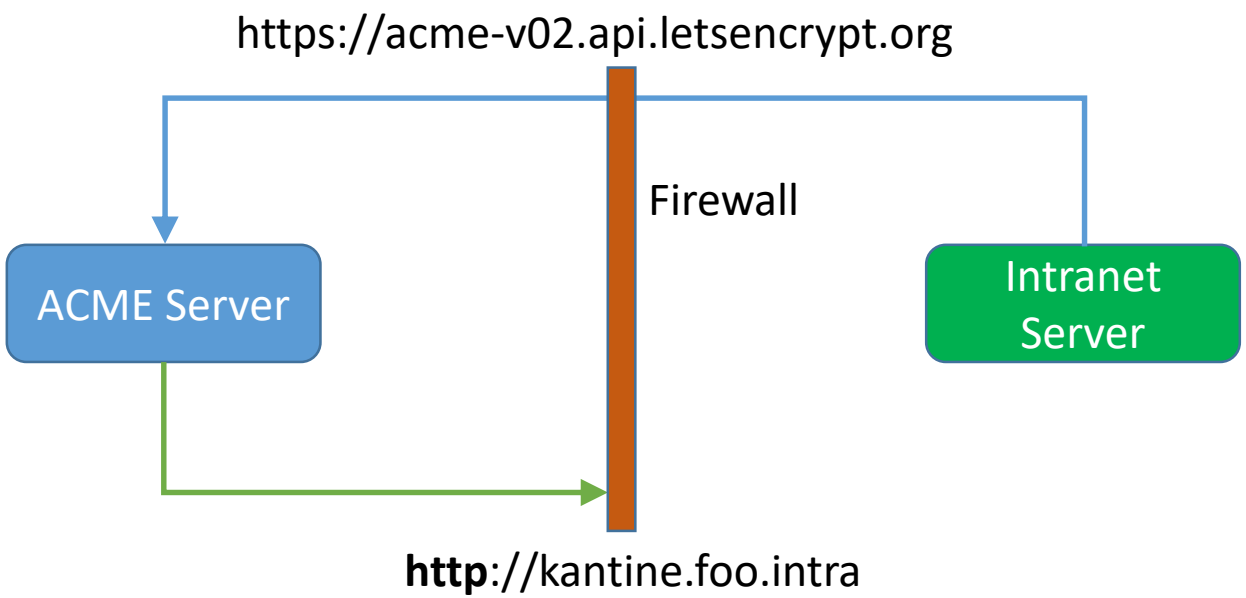
- Meist mehr Server im Intranet als im Internet sichtbar!
- Zertifikatsablauf intern ebenso fatal für die Betriebsabläufe.
- Anforderungen ähnlich im Internet, Lösung könnte ähnlich sein.
- Let‘s Encrypt ggf. erreichbar, aber nicht nutzbar (siehe folgende Folien)
- Also eigener ACME-Server!
  - Boulder & Pebble
  - Kommerzielle Produkte
  - Java / Open Source: ACME-Subprojekt von ID4me (https://gitlab.com/ID4me/Acme)
  - C# / Windows / ADCS : ACME-Server-ACDS (https://github.com/glatzert/ACME-Server-ACDS)

LETSENCRYPT
PLAIN VANILLA

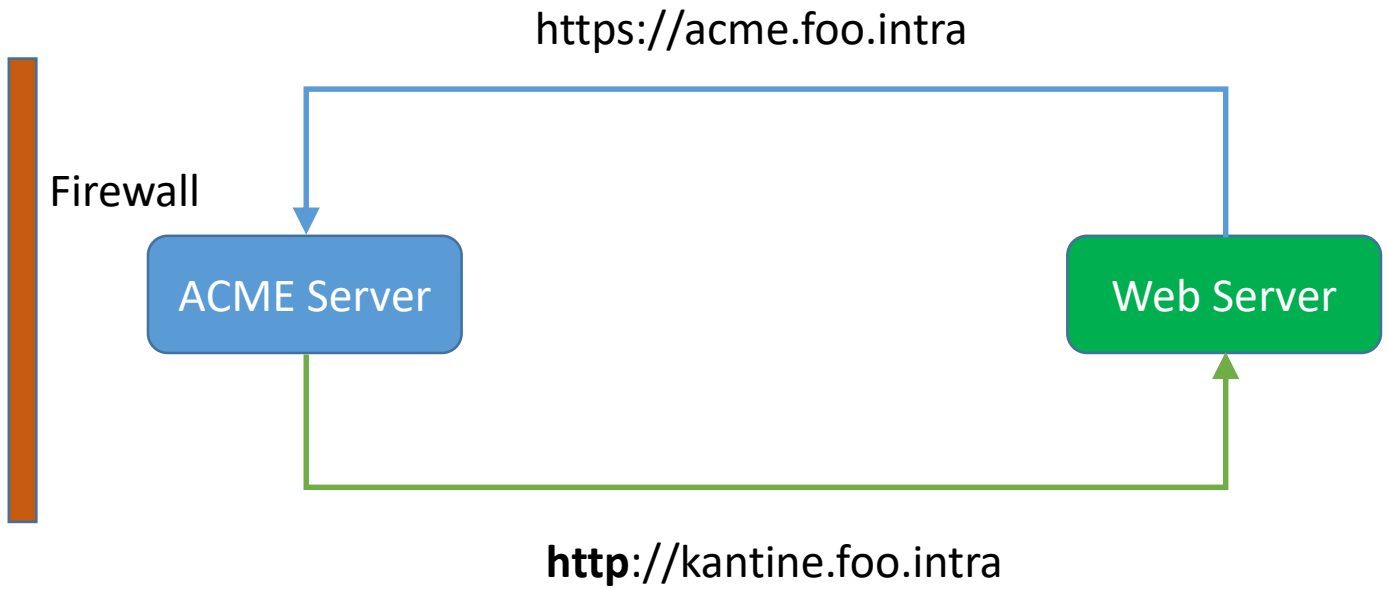
LETSENCRYPT
WILDCARDS !



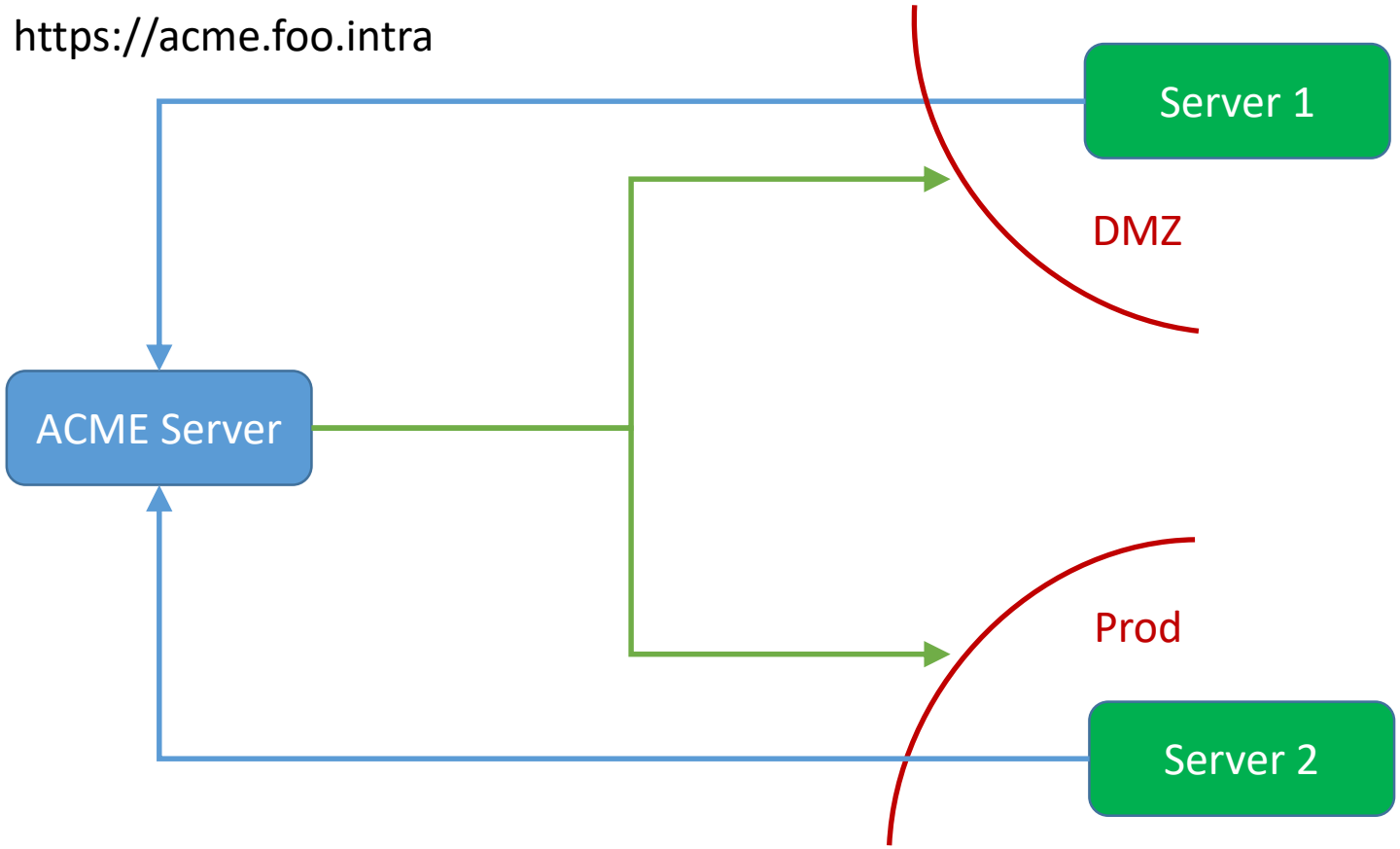
LETSENCRYPT & INTRANET
CHALLENGE BECOMES CHALLENGING



LETSENCRYPT & INTRANET



ACME IN KOMPLEXEN INTRANETS



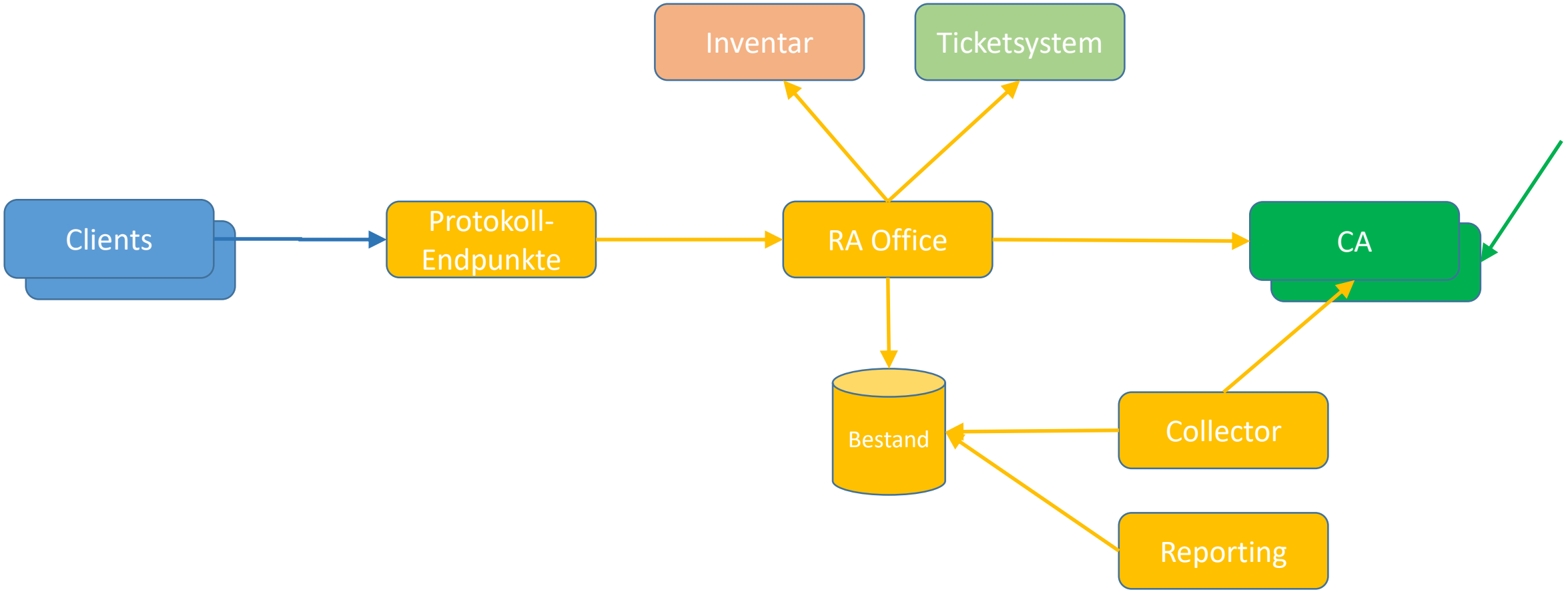
ACME IM INTRANET: PROBLEME

- Netz-Trennung üblich
  - Büro-Anwendungen / Mobile Geräte / Produktion
  - Entwicklung / Test / Betrieb
- Zuordnung zu Verantwortlichem / Abteilung / Anwendung
  - Option mittels ‚External Account Binding'
- Existierende Freigabeprozesse integrieren
- Nutzung für andere Verwendungszwecke (z.B. für TLS Clients) noch als Draft

SCEP

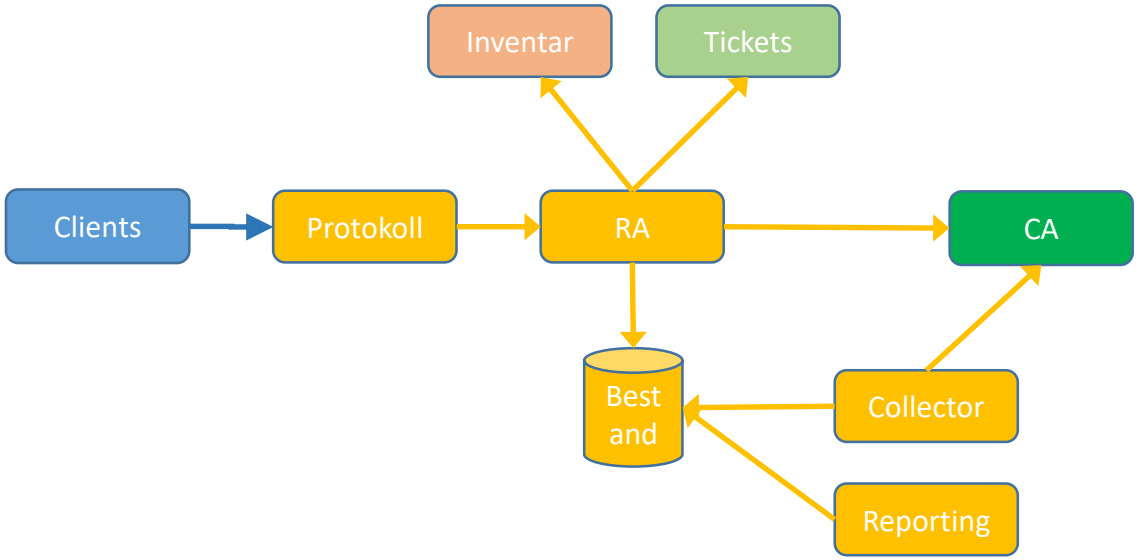
- Netz-Trennung weniger problematisch
  - Client initiiert die Verbindung
- Initiale Ausstellung auf Basis von Passwort
  - NDES-Admin erzeugt sich Passwort und nutzt es länger …
- Erneuerung mittels ‚Proof of Possession' (PoP), i.d.R. Besitz des ‚alten' Schlüssels
- Integration eines Freigabeprozesses notwendig
- Nachfolger ‚Enrollment over Secure Transport' (EST, RFC 7030)
  - Sicheres Transport-Protokoll
  - Wenig Adaption (vornehmlich Cisco-Router)

KOMPONENTEN-ÜBERBLICK



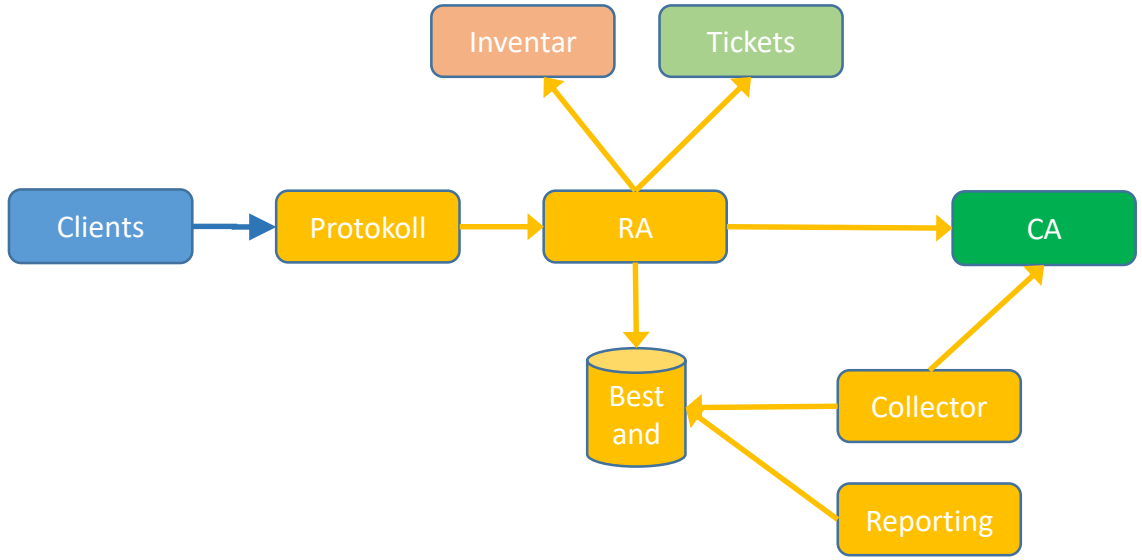
ZERTIFIKATSANTRAG
AUFGABEN DER KOMPONENTE ‘RA’

- Trivialer Ansatz: RA-Officer prüft und gibt Request ggf. frei
- Einfache Plausibilisierung bei ACME erfolgreich umgesetzt
- Komplexe Prüfprozesse abbildbar
  - Proprietäre Lösungen (z.B. ejbca, xpki, appviewX)
  - BPMN-Abläufe
  - Anbindung an Ticket-Systeme
  - Einbindung interner Services (z.B. Bestandsführende Systeme)
  - Entartet schnell in Integrations-Projekt



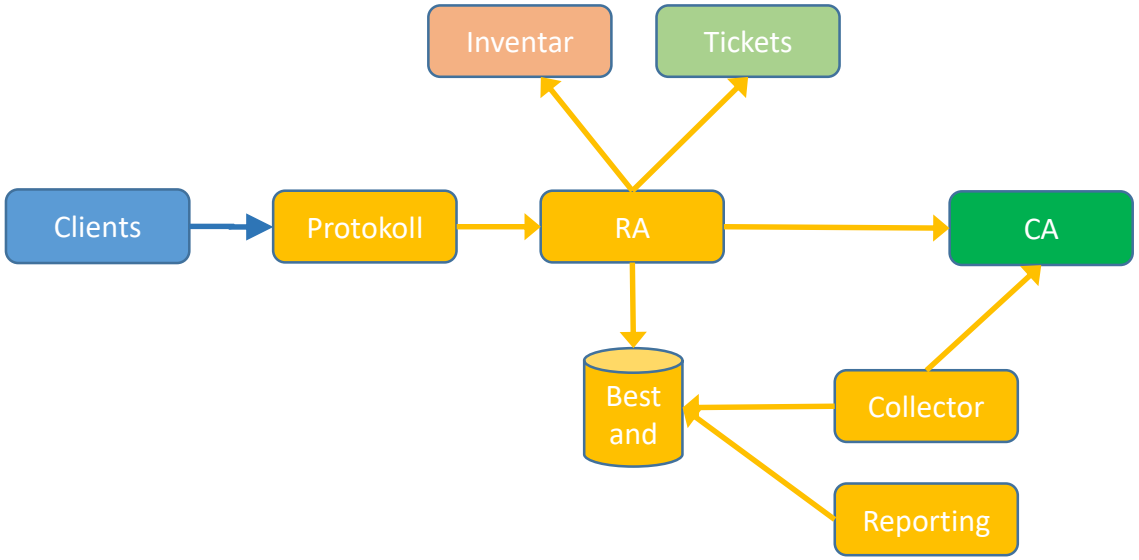
ANBINDUNG CA

- Verbindung zur CA
  - Zertifikatsaustellung
  - Rückruf
  - Bestandsabfrage
- Protokolle
  - Certificate Management Protocol (CMP), RFC 2510
  - Windows: WCCE / ADCS API
  - Proprietäre Protokolle (SOAP / REST)



ZERTIFIKATSBESTAND

- **Ein** Inventar aller ausgestellter Zertifikate
- Basis für IT Governance / Asset Management
- Behandlung verschiedener Quellen:
  - Eigene, interne CAs (z.B. für Dev, Test, Prod, Infrastruktur, …)
  - ADCS (Windows-Domäne)
  - CAB/F-Zertifikate (‘öffentliche’ Zertifikate)
  - (qualifizierte) Zertifikate von akkred. Anbietern
- Quelle auch für den Ausstellungsprozess (‚unique key‘)



ZERTIFIKATSMANAGEMENT
AUSKUNFTSFÄHIG SEIN

- Analyse von Zertifikatsaspekten
  - Ablaufdaten und erfolgte Erneuerungen, ggf. Benachrichtigungen versenden
  - Algorithmen (Verschlüsselung, Hashing, Padding)
  - Schlüssellängen und Parameter (v.a. bei ECC)
  - Nutzende Systeme (z.B. bei Bugs wie HeartBleed, Kompromittierungsverdacht)
- Analyse der Beantragungs-, Ausstellungs- und Rückruf-Aktion
  - Anomalie-Erkennung
  - KPIs

TOOL-ÜBERSICHT

	ACME-ADCS-Server	ejbca	Nexus	keyfactor	Venafi
ACME	+	Comm.	+	+	+
SCEP		+	+	+	+
Web-Form		+	+	+	+
CMP		+	+	+	+
ADCS-Adapter	+		+	+	+
Zert-Bestand		+	+	+	+
Reporting			+	+	+
Workflow		Prop.	BPMN	Prop.	Prop.
Open Source	+	(+)			

TOOL-ÜBERSICHT

Werbeblock !

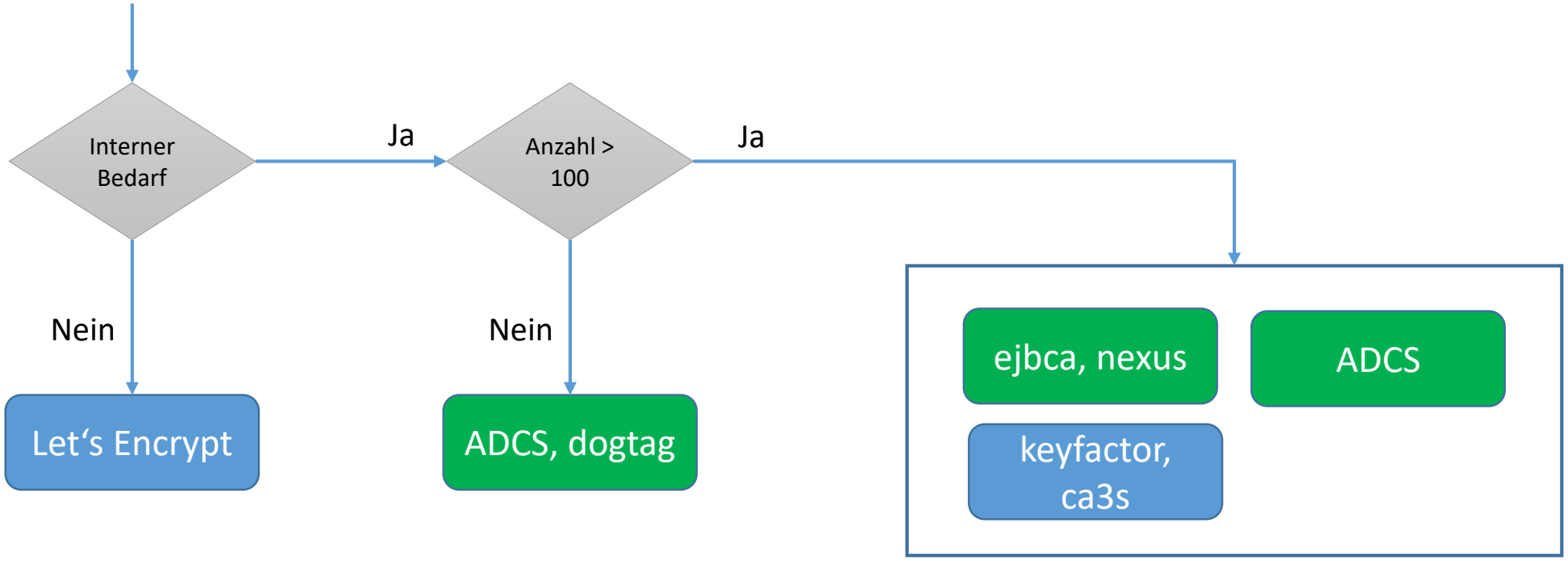
	ACME-ADCS-Server	ejbca	Nexus	keyfactor	Venafi	ca3s
ACME	+	Comm.	+	+	+	+
SCEP		+	+	+	+	+
Web-Form		+	+	+	+	+
CMP		+	+	+	+	+
ADCS-Adapter	+		+	+	+	+
Zert-Bestand		+	+	+	+	+
Reporting			+	+	+	+
Workflow		Prop.	BPMN.	Prop.	Prop.	BPMN
Open Source	+	(+)				+

OPEN SOURCE ‘CA3S’

Werbeblock !

- Start in 2016
- Motivation: Unzufrieden mit prop. Software
- Pure Java (plus Typescript, BPMN), Spring Boot, jHipster
- Hosted on github: https://github.com/kuehne-trustable-de/ca3sCore
- Lizenz: EUPL (GPL mit Übersetzungsservice)
- Im produktiven Einsatz
- ACME-Proxy auf WebSocket-Basis im Bau
- Volunteers welcome!

ENTSCHEIDUNGSBAUM
BRAUCHE ICH EIN ZERTIFIKATSMANAGEMENT?



EMPFEHLUNGEN

- Zertifikatsproblematik ernst nehmen!
- Zertifikatsablauf ist **kein** Schicksalsschlag! Ausreden werden nicht mehr akzeptiert!
- Let‘s Encrypt nutzen!
- > 100 Zertifikate: Management-Software nutzen
- Open Source & Standards einsetzen & unterstützen!
- Danach: TLS-Config & Trust Management aufbauen

FRAGEN ?

- kuehne@trustable.de

- https://github.com/kuehne-trustable-de/ca3sCore
- https://www.oasis-open.org/apps/org/workgroup/dss-x