ACME Auto Discovery

draft-vanbrouwershaven-acme-auto-discovery

Mike Ounsworth, Paul van Brouwershaven

ACME WG

IETF 119 – Brisbane | March 2024

ENTRUST

SECURING A WORLD IN MOTION

New collaborators and authors

- Tim Hollebeek (DigiCert),
- Corey Bonnell (DigiCert),
- Q Missel,
- Inigo Barreira (Sectigo),
- Wayne Thayer (Fastly)

ENTRUST

Overview

- draft-vanbrouwershaven-acme-auto-discovery
  - Provides a DNS-based way for ACME clients to learn the preferred ACME server for a given domain in cases where explicit ACME client configuration is not possible.
  - All open design issues resolved since Prague.

- draft-vanbrouwershaven-acme-client-discovery
  - Registers `.well-known/acme-keys` where a cloud provider can publish the set of ACME client keys that belong to its ACME bots.
  - Acts as a level of indirection so that client keys don't get permanently pinned in CA allow-lists and thus prevent rotation, adding new clients with new keys, etc.

ENTRUST

Terminology

- **ACME Account** ::= the account used by the ACME client, represented by an asymmetric keypair.
- **CA Account** ::= an administrative or organization account in the UI of the CA – could be used for associating out-of-band validation, billing, and other types of info that is “out-of-band” from ACME.

ENTRUST

Design point we've been stuck on #1

Multiple accounts at the CA

ENTRUST

Multiple account disambiguation

Problem: What if multiple subscriber accounts at the CA are authorized to issue for the same domain?

Solution: RFC 8657 defines the “accounturi” parameter for CAA DNS record.

We added a new section explaining how to use “accounturi” for account disambiguation for auto-discovered ACME requests, and how “accounturi” interacts with the “priority” CAA parameter defined in this draft.

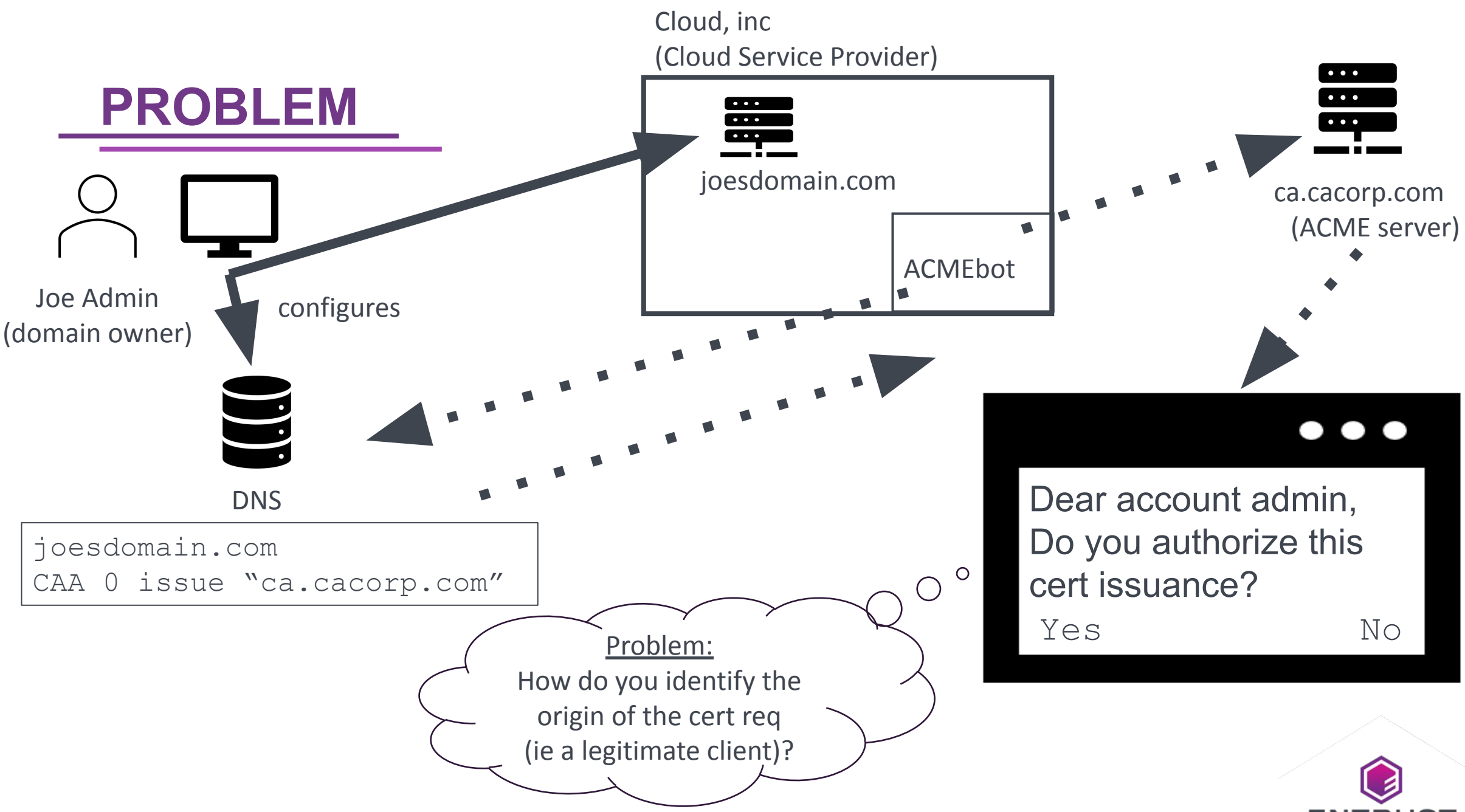
SOLVED!

ENTRUST

Design point we've been stuck on #2

Authorization of the client

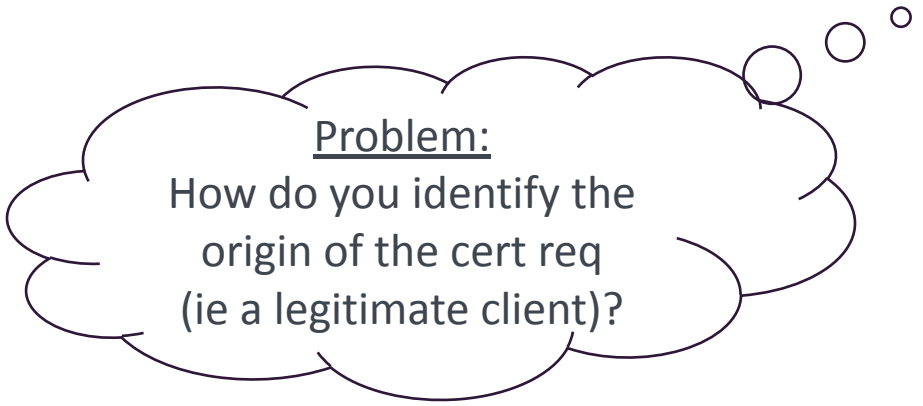
PROBLEM

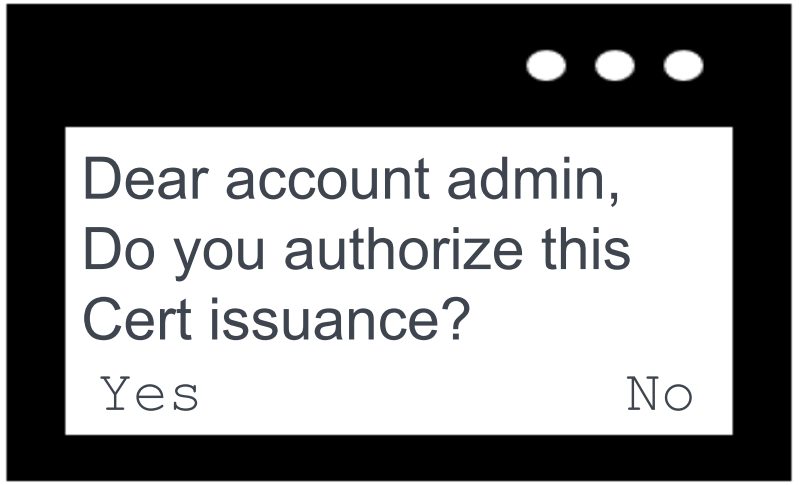


8

ENTRUST

Problem



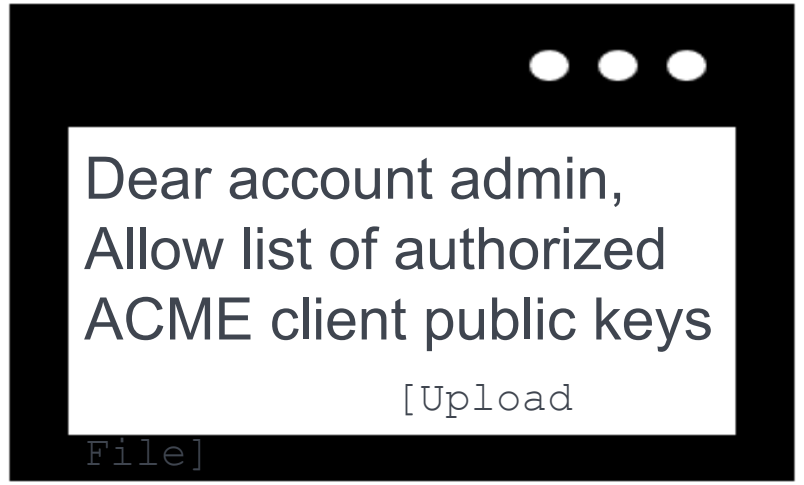


- For certificates that contain information that can be validated entirely by ACME validation methods, there is no problem: the ACME DV check is sufficient authorization.
- For certificates that contain information validated out-of-band from ACME: we need some kind of client authentication and authorization against the requested domain.

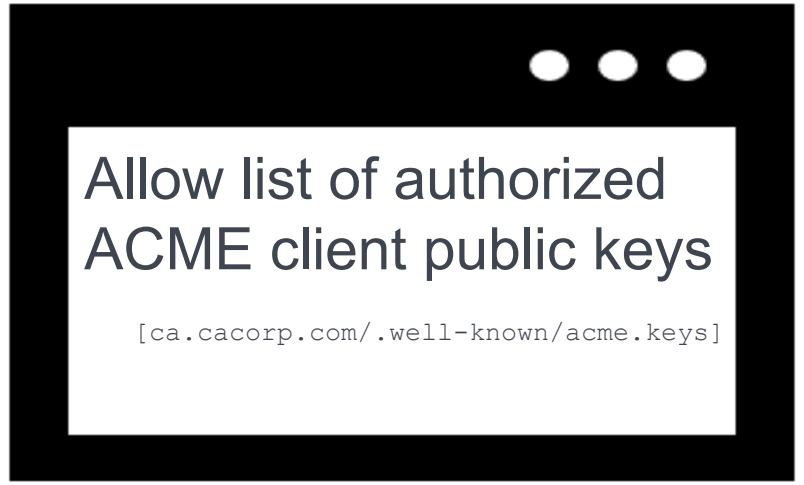
- Ok. How?
  - ACME has the External Account Binding (EAB) for this; however if the Cloud Service Provider (CSP) makes a UI for a subscriber to enter an EAB key, then you've now done explicit ACME configuration, so you don't need Auto-Discovery at all.
  - That means we have to do it with only the ACME account client key.

ENTRUST

Proposed Solution



- Subscriber pre-configures with the CA a set of ACME account public keys that may initiate requests to this account.
- Discussions around:
  1. CSPs must publish their ACME account pub keys .. which means rolling them over is difficult. … ok, so add a layer of dynamic binding:
  2. Do we need a separate draft for “ACME Account Key Auto-Discovery”?
  3. Is this even CA/B F BR compliant?
     - We have been doing very careful readings of the TLS Server Cert BRs section 3.2.5.
       This *should* be allowed, but maybe the BRs need a tweak to make this explicitly allowed?



ENTRUST

draft-vanbrouwershaven-acme-client-discovery

*“Specifically, this document registers the URI "**/.well-known/acme-keys**" where all compliant service providers can publish their ACME client public keys. This mechanism not only enhances the trust relationship by allowing the ACME server to identify the specific service provider but also provides flexibility to service providers. They can use multiple keys and rotate them as often as they like, thereby improving security and control over their ACME client configurations.*

Moreover, this mechanism empowers CA customers by giving them the ability to specifically authorize which service providers can request certificates on their behalf.”

ENTRUST

Summary

Summary

- In Prague, the chairs asked us to form a design team to solve the outstanding problems.
- We have.

- It resulted in realizing that we have two separate discovery problems here:
  - ACME client auto-discovering which ACME server the domain-owner wants it to use (via DNS CAA record).
  - CA having an allow-list of ACME account keys that are authorized to issue for this CA account.
    - The Cloud Service Provider (CSP) can place these at https://csp.com/.well-known/acme-keys so that they can be rotated by the CSP.

ENTRUST

Adoption?

- We should probably ask the adoption question separately:
  - **Adoption** draft-vanbrouwershaven-acme-auto-discovery **?**
    - Provides a DNS-based way for ACME clients to learn the preferred ACME server for a given domain in cases where explicit ACME client configuration is not possible.
    - All open design issues resolved since Prague.
  - **Adoption** draft-vanbrouwershaven-acme-client-discovery **?**
    - Less mature, feedback from CSPs wanted.
    - Registers `.well-known/acme-keys` where a cloud provider can publish the set of ACME client keys that belong to its ACME bots.
    - Acts as a level of indirection so that client keys don't get permanently pinned in CA allow-lists and thus prevent rotation,
      adding new clients with new keys, etc.

ENTRUST

Thank You

Mike.Ounsworth@entrust.com

Paul.vanBrouwershaven@entrust.com

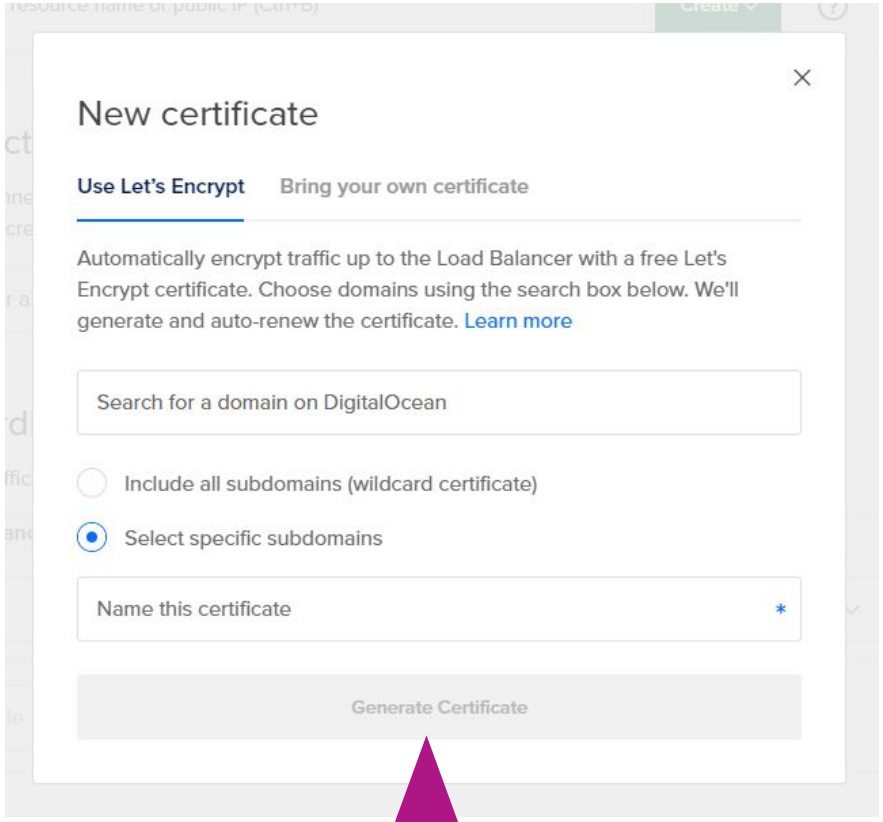
entrust.com

ENTRUST

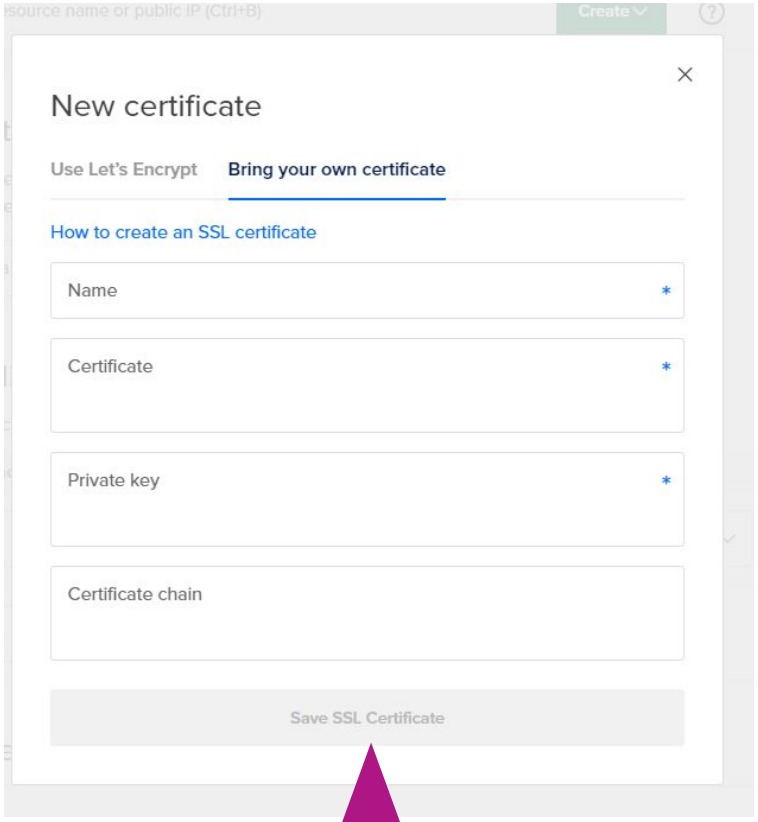
SECURING A WORLD IN MOTION

Problem refresher from 117

DIGITALOCEAN - LOAD BALANCER



You can use Let's Encrypt (ACME), provide some configuration, but you **can not** specify your own ACME server or account binding.



Or you can upload a custom certificate.

17

PROBLEM

- A certificate with a validity of 90-days ‘requires’ automation
  - Renewing a certificate manually 4-6 times will not be ‘appreciated’
- When subscribers can’t specify their preferred ACME server, the **default** will become the norm!
- If the default is the norm, we **lack issuer diversity** which risks becoming a single point of failure.
- (side-benefit: prioritized list of fallback ACME servers for a given domain)

How do we automate discovery of the domain owner’s preferred CA?

18

ENTRUST

PROBLEM

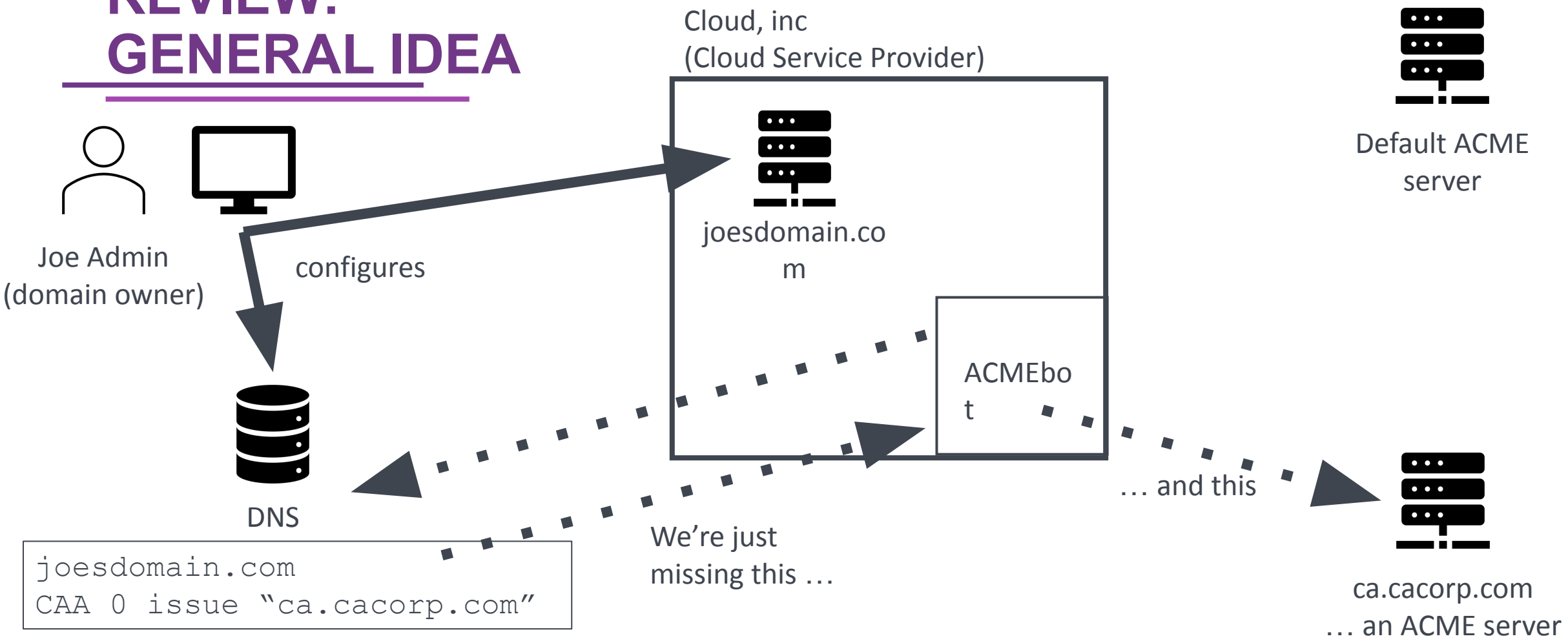
- A certificate with a validity of 90-days ‘requires’ automation
  - Renewing a certificate manually 4-6 times will not be ‘appreciated’
- When subscribers can’t specify their preferred ACME server, the **default** will become the norm!
- If the default is the norm, we **lack issuer diversity** which risks becoming a single point of failure.
- (side-benefit: prioritized list of fallback ACME servers for a given domain)

How do we automate discovery of the domain owner’s preferred CA?

19

ENTRUST

REVIEW: GENERAL IDEA



… you would think there's enough info here
to send ACMEbot to the Joe's preferred ACME server …

20

ENTRUST

Current Status

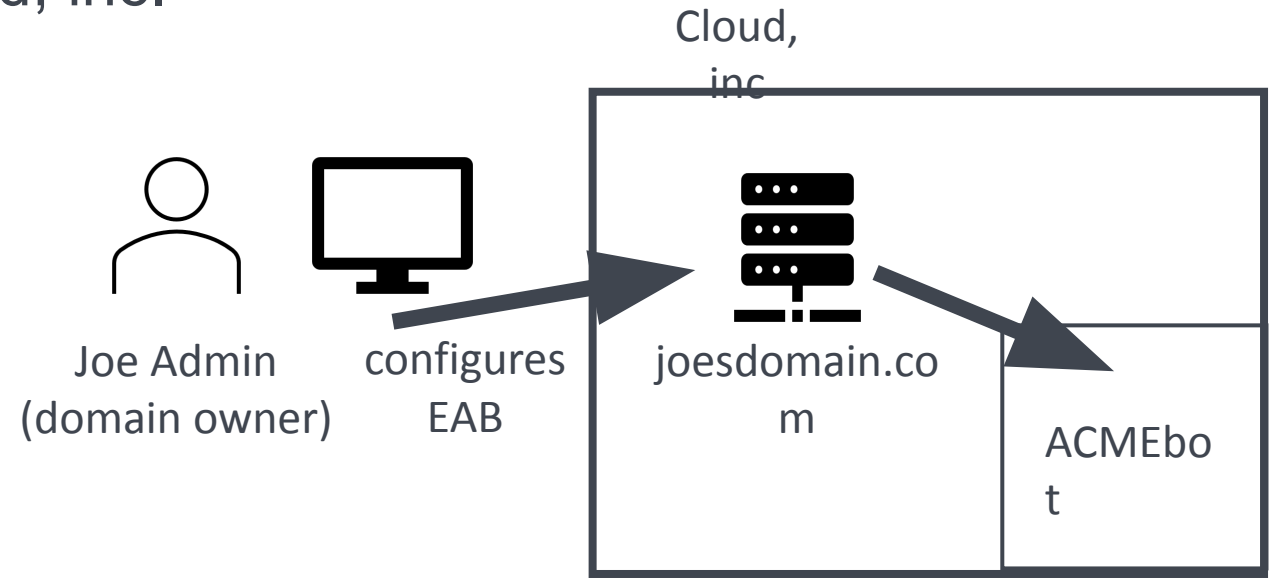
Status

- A new draft (-02) was released incorporating the feedback received.
- We have identified (and are attempting to solve) more challenges around the external/internal account binding mechanisms.
  - General problem: How to associate incoming ACME requests with the correct CA account?
  - Sub-Problem 1: The ACME account will be owned by the CSP and may either be re-used across all customers they manage, or may be a fresh account per ACME request.
    - So we cannot use ACME account to retrieve the appropriate CA account.
  - Sub-Problem 2: multiple CA accounts are authorized to issue for the same domain.
    - So we cannot use requested domain to retrieve the appropriate CA account.

ENTRUST

Problem 0: External Account Binding keys

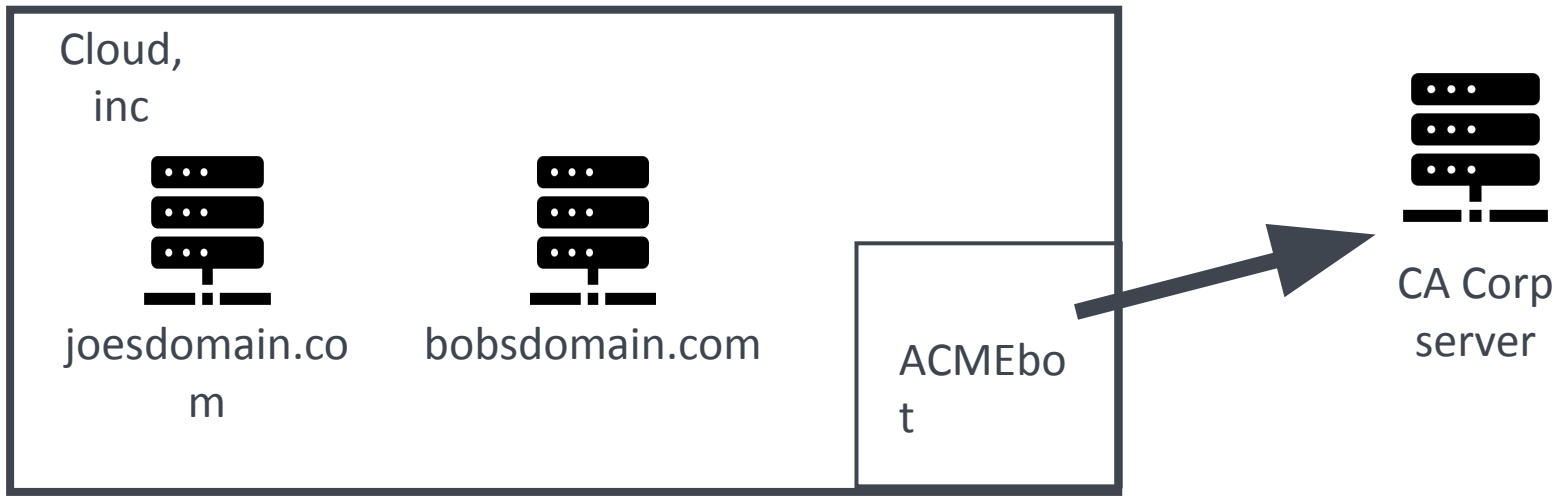
- ACME already has External Account Binding keys, but they can't be leveraged here because:
  1. Passing Joe's EAB key down to ACMEBot requires UI changes in Cloud, inc.
  2. Joe's EAB key may have more permissions than Joe really wants to share with Cloud, inc.



ENTRUST

Problem 1: ACME accounts are not unique per CA account

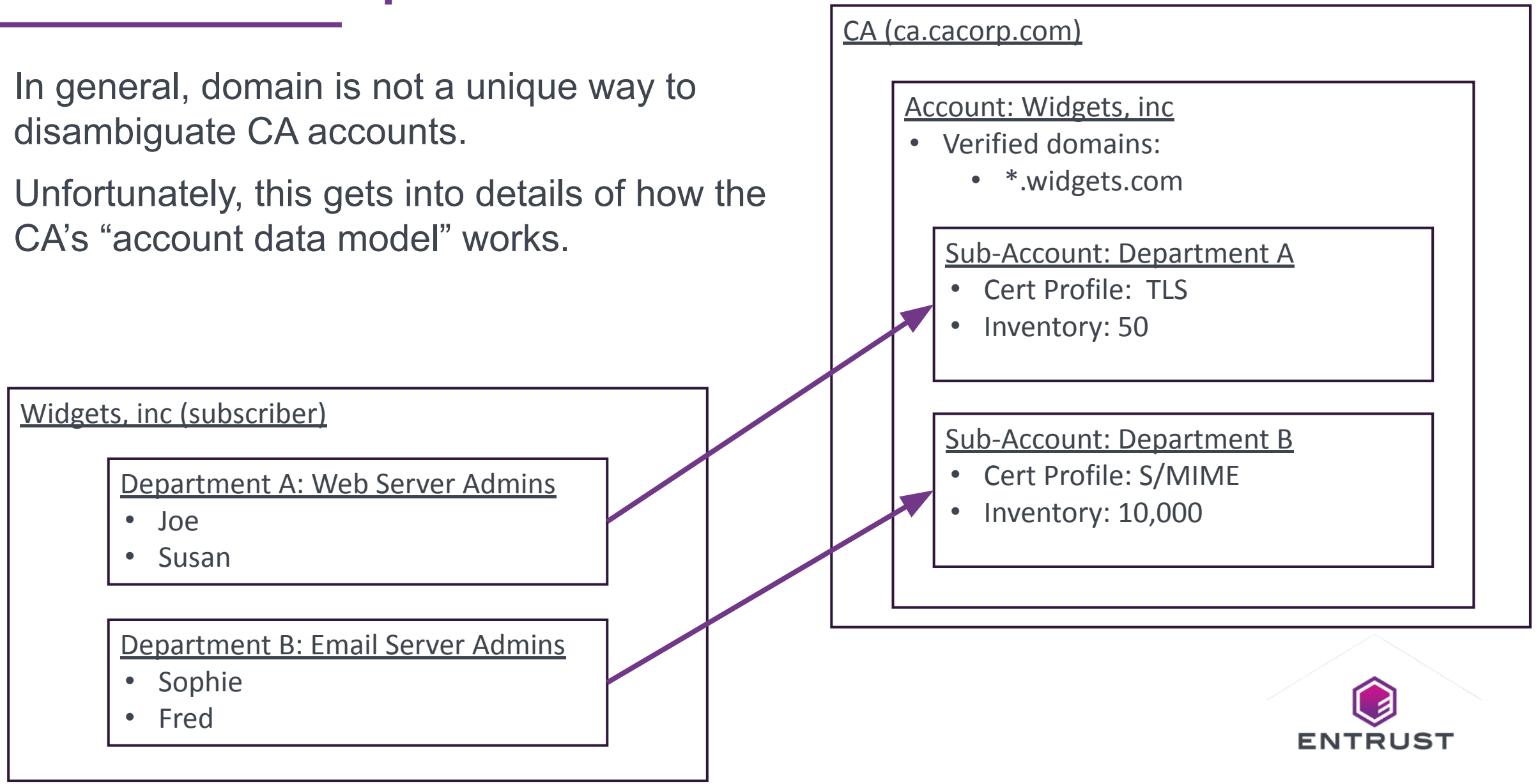
- Most service providers currently work by either having a single ACME account per CA, or generating throwaway ACME accounts – ex.: Certbot automatically creates a new account for each ACME server but doesn't know anything about users, actually, Cerbot creates the account keys in a shared config folder by default.
- This problem is described in section 9.3 of the security considerations of the draft.



ENTRUST

Problem 2: Multiple CA accounts for the same domain

- In general, domain is not a unique way to disambiguate CA accounts.
- Unfortunately, this gets into details of how the CA's "account data model" works.



ENTRUST

Potential Account Binding (AB) Mechanisms

External AB

- Not supported by Cloud Service Providers (CSP).
- Unlikely to gain support as it requires interface and implementation changes by the CSP.
- Requires a unique account per CSP customer.

Internal AB (email)

- Described in section 7.1.2 of the draft.
- Prone to phishing attacks.
- Easier to implement than the EAB as required information (email) is already known by the CSP.
- Requires a unique account per CSP customer.

Internal AB (DV)

- Described in section 7.1.1 of the draft.
- Does not require any CSP changes.
- Requires a unique account.

ENTRUST

Potential Account Binding (AB) Mechanisms

External AB

- Not supported by Cloud Service Providers (CSP)
- Unlikely to gain support as it requires interface and implementation changes by the CSP
- Requires a unique account per CSP customer

Internal AB (email)

- Described in section … of the draft
- Prone to phishi…
- Easier to imple… the EAB as re… information (e… already known…
- Requires a uniq… per CSP custom…

Int… AB (DV)

… 7.1.1

Design is still ongoing, we're not sure this is right yet.

More vendor input is needed here!

For example, is email really the right mechanism? What about a UUID in the CAA DNS record to disambiguate accounts? Or maybe {domain + cert profile} is unique? More design needed.

ENTRUST

Shared Account Binding

- Not described in the draft, looking for feedback
- Similar to where the CSP (Cloud Service Provider) is a reseller of the CA and uses one set of API credentials for multiple customers, except there would be no contract between the CA and the CSP
- The ACME key could identify the CSP, to allow CA customers to enable specific CSP
  - The CSP could publish its public key(s) in its well-known directory
  - The CSP could obtain a certificate for it's ACME key and include it in the x5u parameter of the JWK
    - less likely to see broad adoption, involves validation costs and renewal procedures
  - A challenge response with the account key email address could be performed (based on the CSP domain, e.g., @aws.com)
    - less likely to see broad adoption, requires (automated) acknowledgement on the CSP side
- Domain Control Validation determines if the CSP is authorized to issue this certificate

ENTRUST

Summary & Next Steps

- This draft **slowed down** when we realized there's a hard problem buried in here.
- We need more design iteration on how to disambiguate which CA account a given ACME request should be associated with – we may need to consider authentication and authorization separately.

- This may need **a design group** of CAs and CSPs to make sure we've captured and addressed the sticky cases properly (some of which may be CA-specific).

ENTRUST