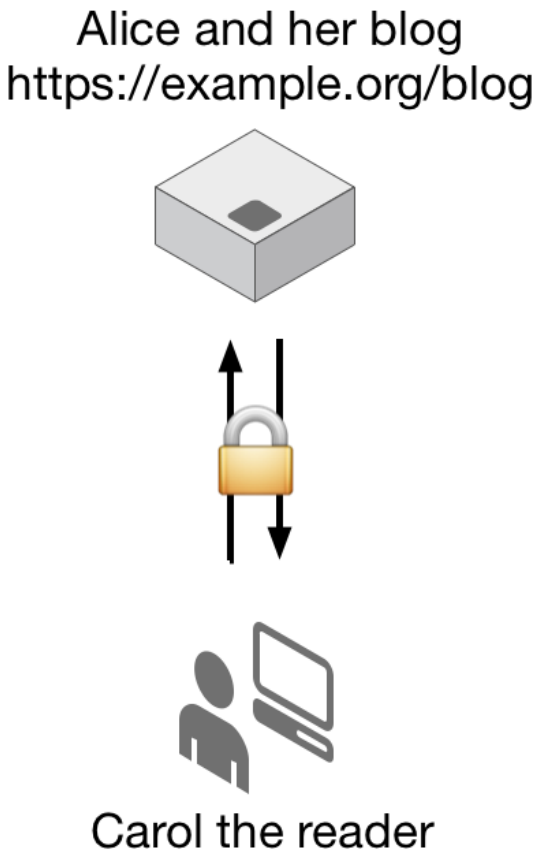
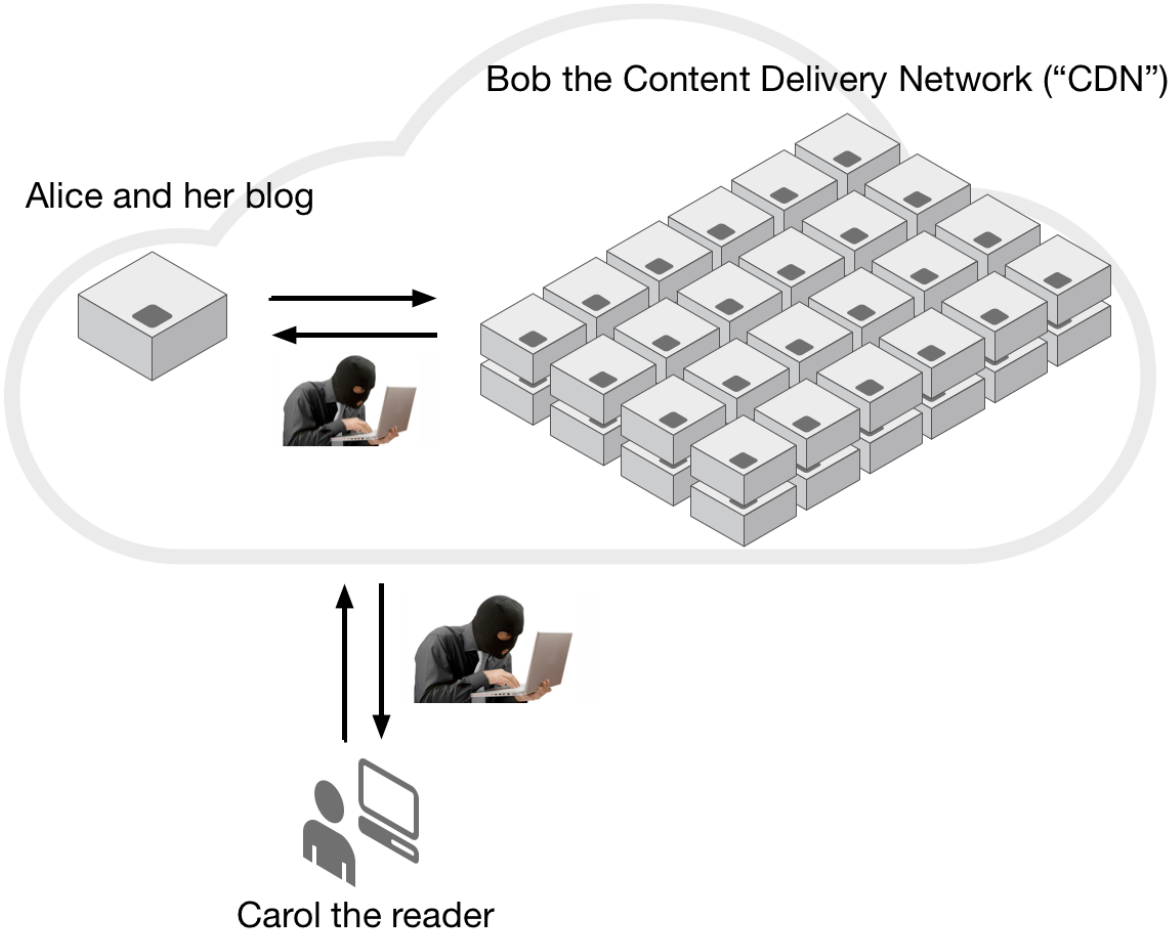
When HTTPS Meets CDN: A Case of Authentication in Delegated Services

J. Liang, J. Jiang, H. Duan, K. Li, T. Wan, J. Wu

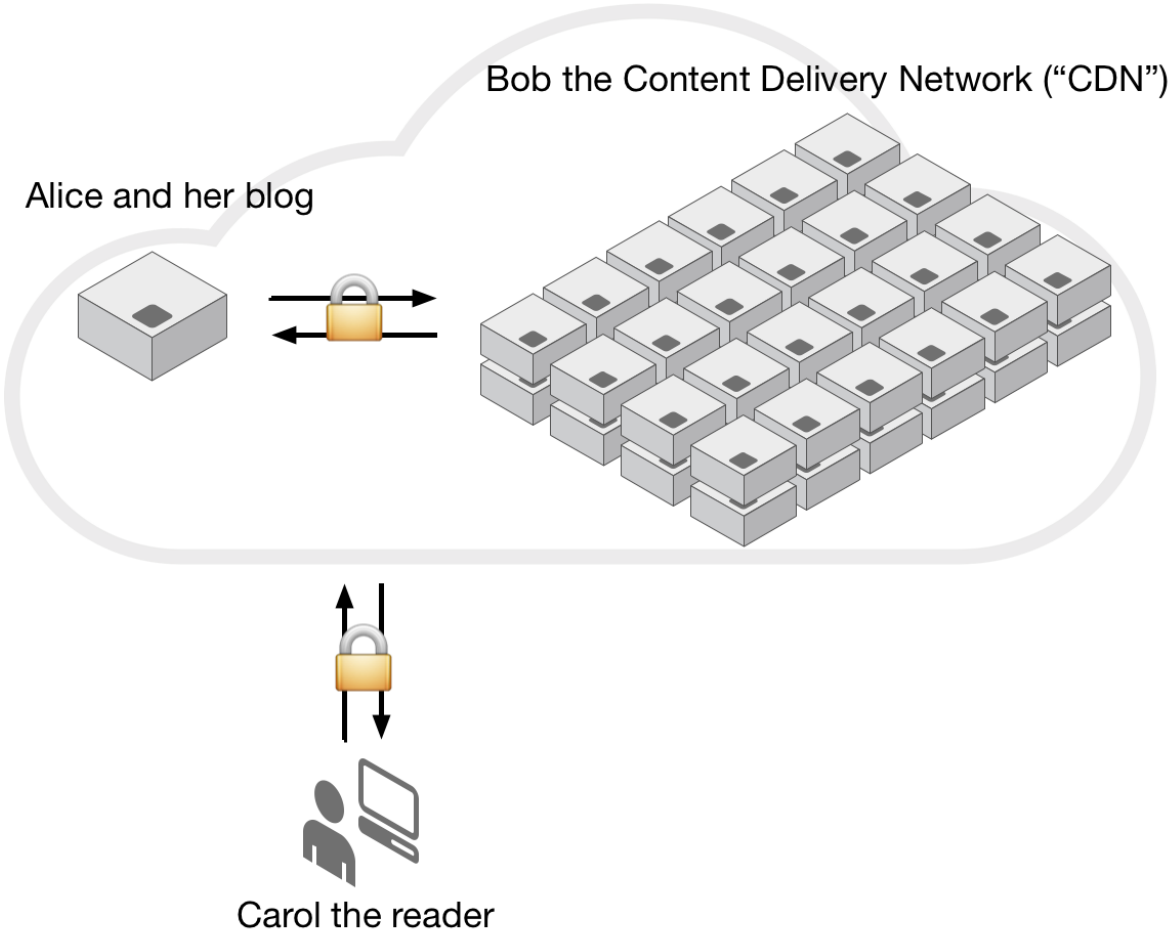
Problem statement: TLS, an End-to-End Protocol



Problem Statement: End-to-End Protocol and Three Parties



Problem Statement: Aim for Secure Front-End and Back-End Communication



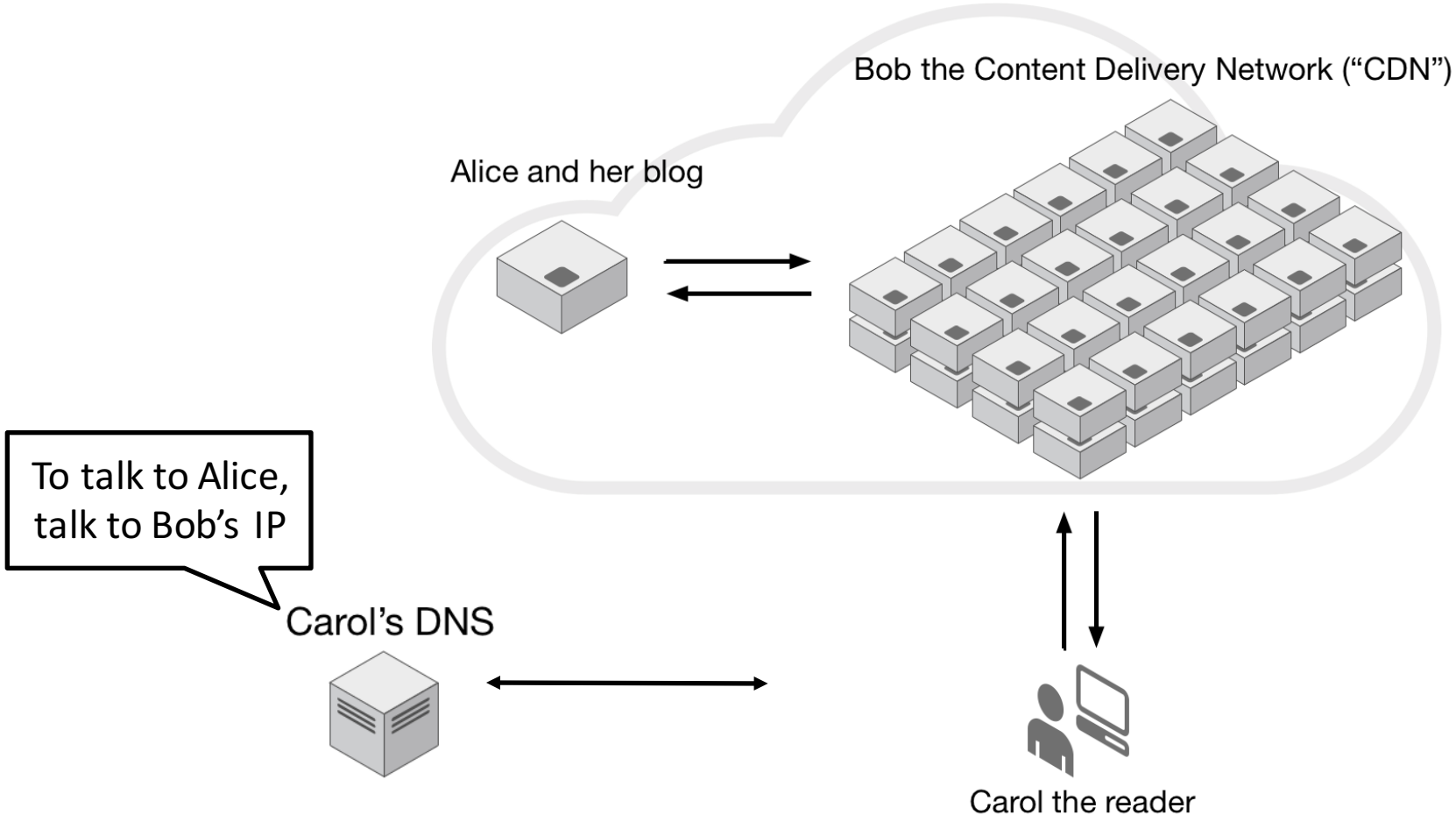
Presentation Topics

- Problem and requirements for possible solution
- Current solutions in practice
- Possible X.509 out-of-box solution
- Proposed solution
- Securing back-end communication
- Final remarks

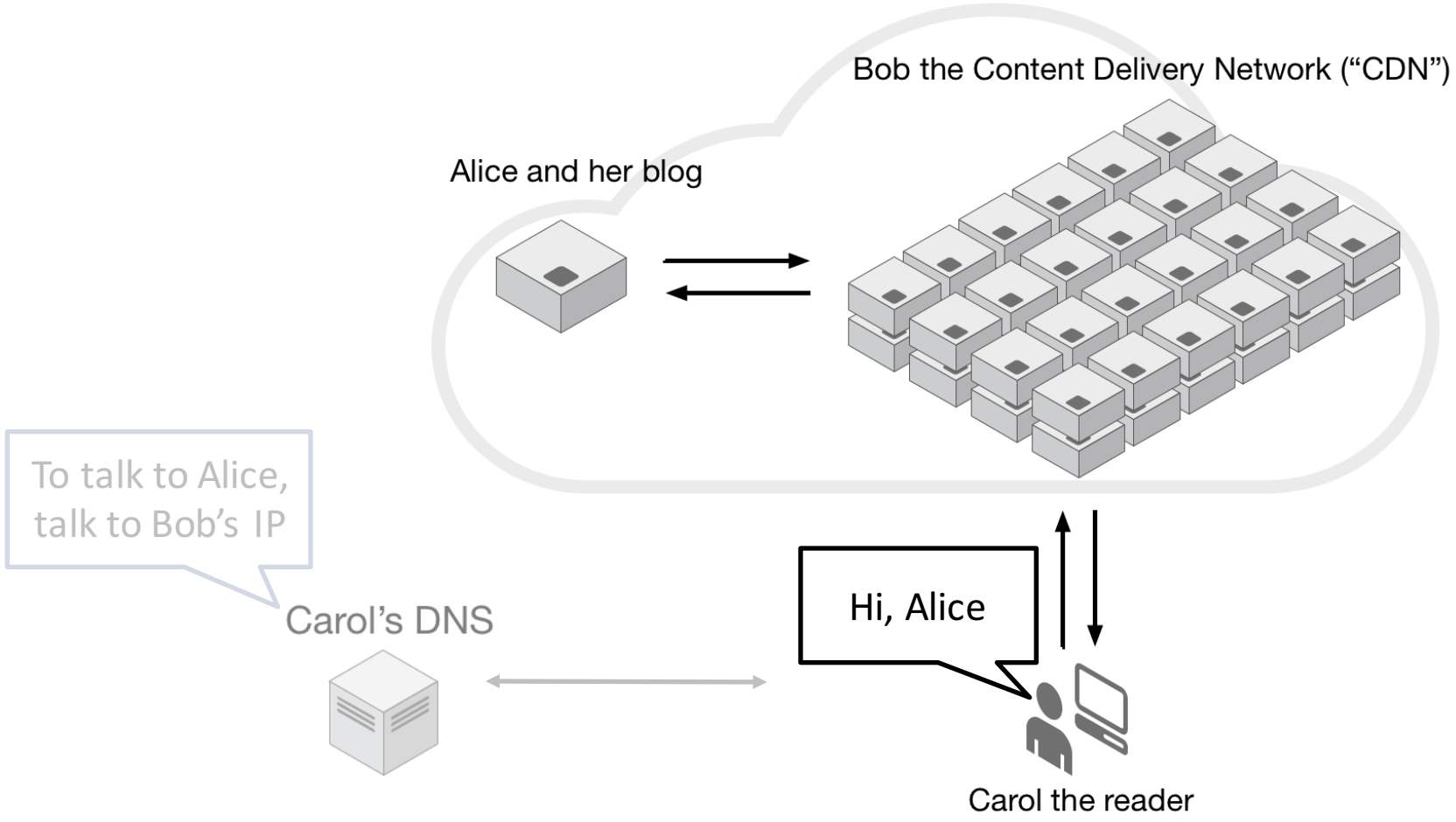
Presentation Topics

- Problem and requirements for possible solution
- Current solutions in practice
- Possible X.509 out-of-box solution
- Proposed solution
- Securing back-end communication
- Final remarks

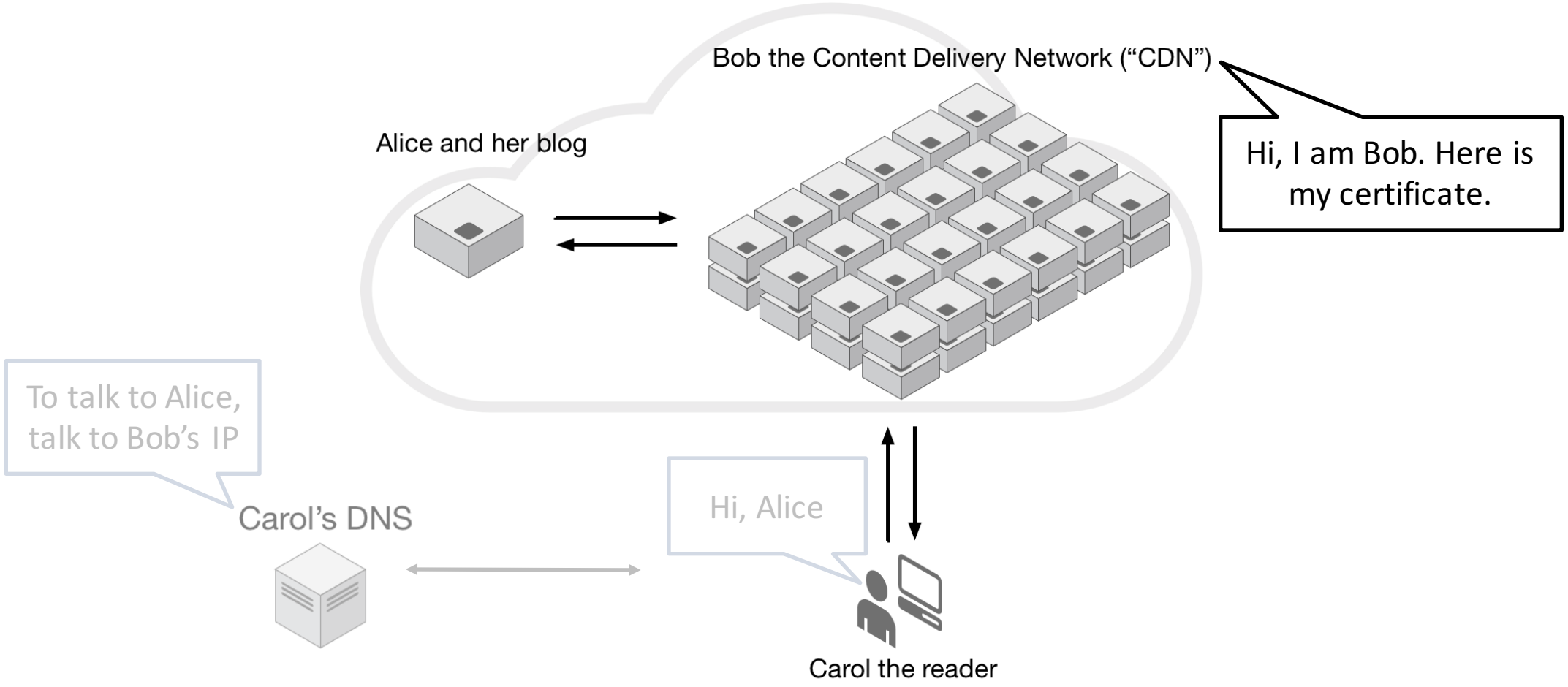
Problem Statement: Front-End



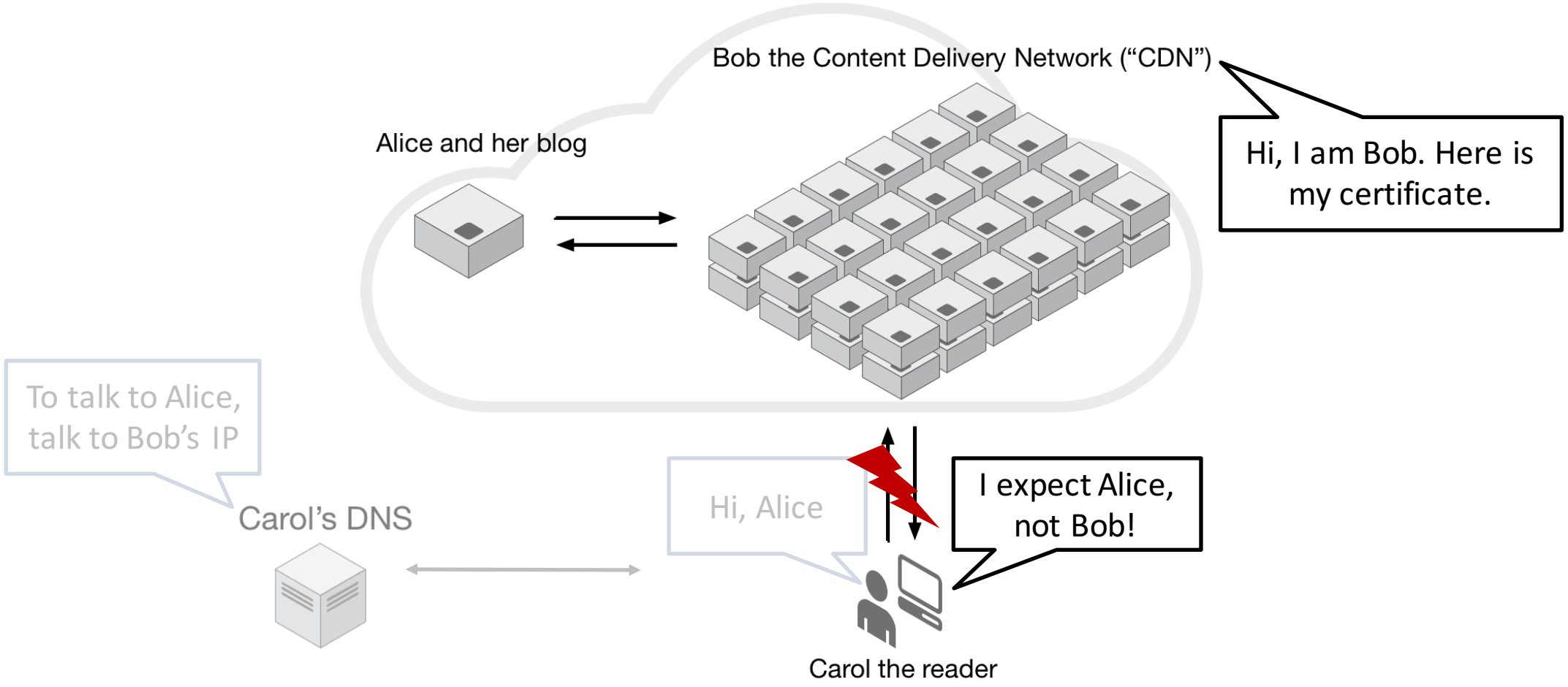
Problem Statement: Front-End



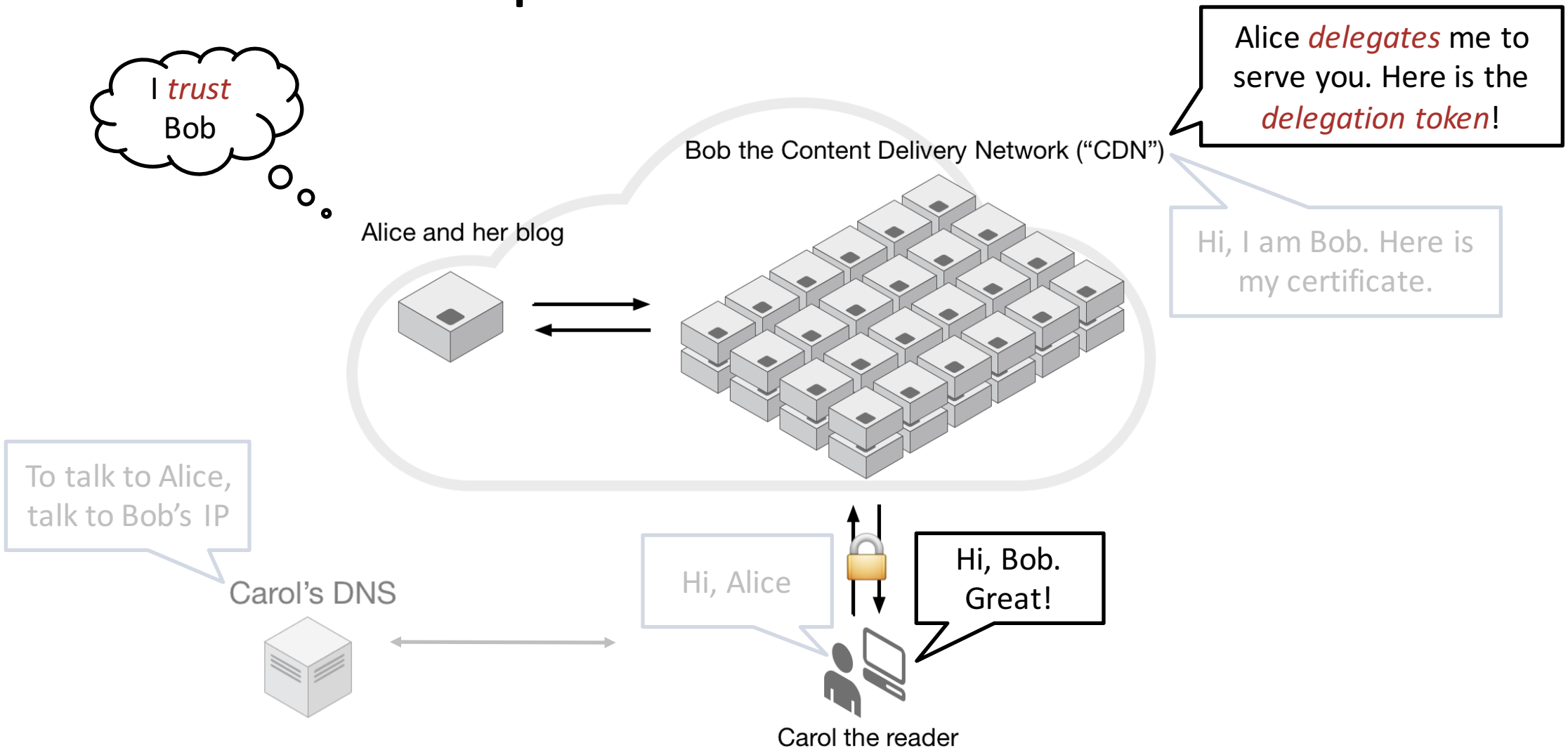
Problem Statement: Front-End



Problem Statement: Front-End



Problem Statement: Concept for Front-End Communication



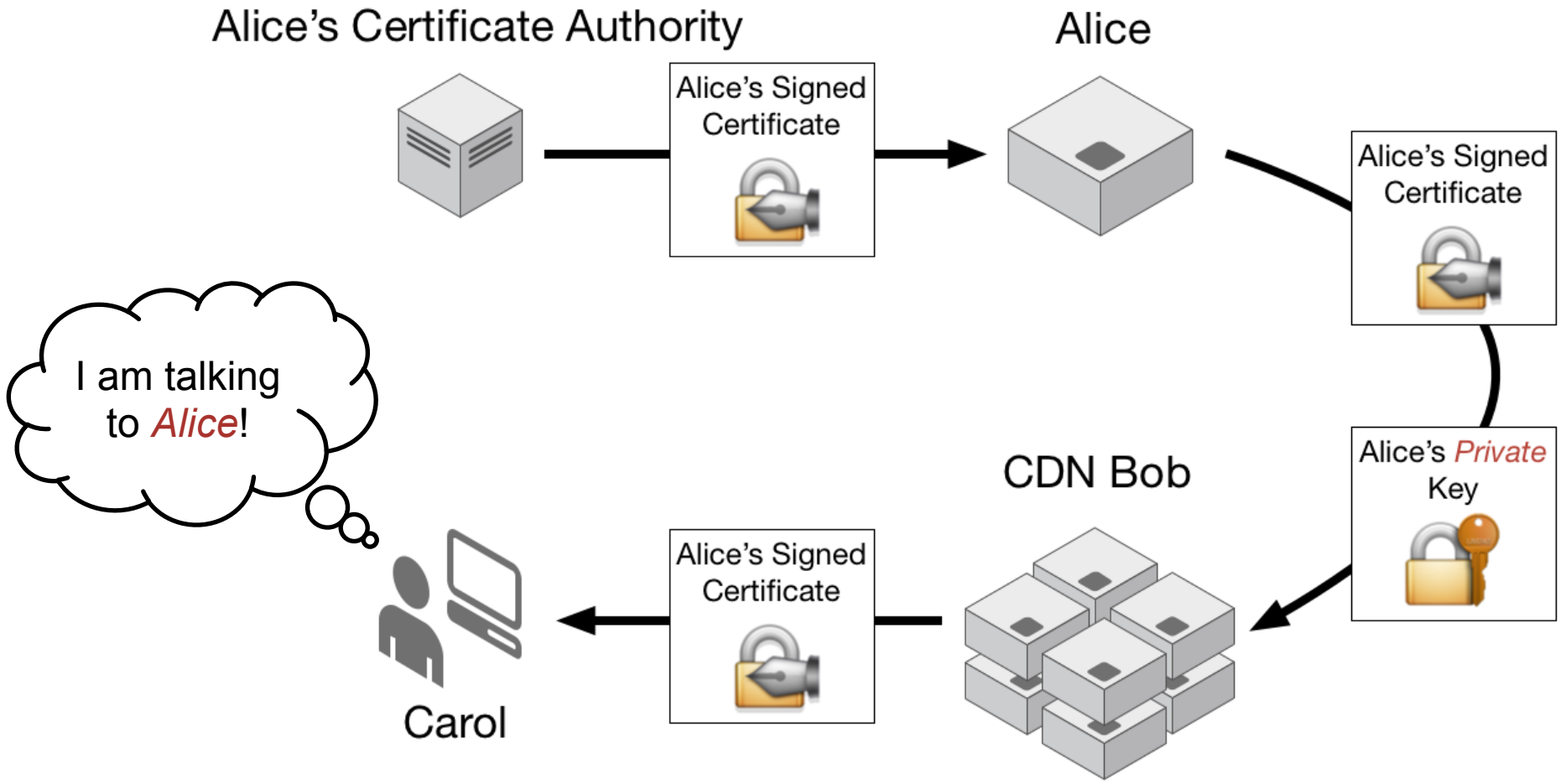
Proposed Requirements

1. *Unforgeable* delegation token
2. Delegator can *issue* and *revoke* delegation token *independently* and *efficiently*
3. Delegation token includes *complete identification* of delegator

Presentation Topics

- Problem and requirements for possible solution
- Current solutions in practice
- Possible X.509 out-of-box solution
- Proposed solution
- Securing back-end communication
- Final remarks

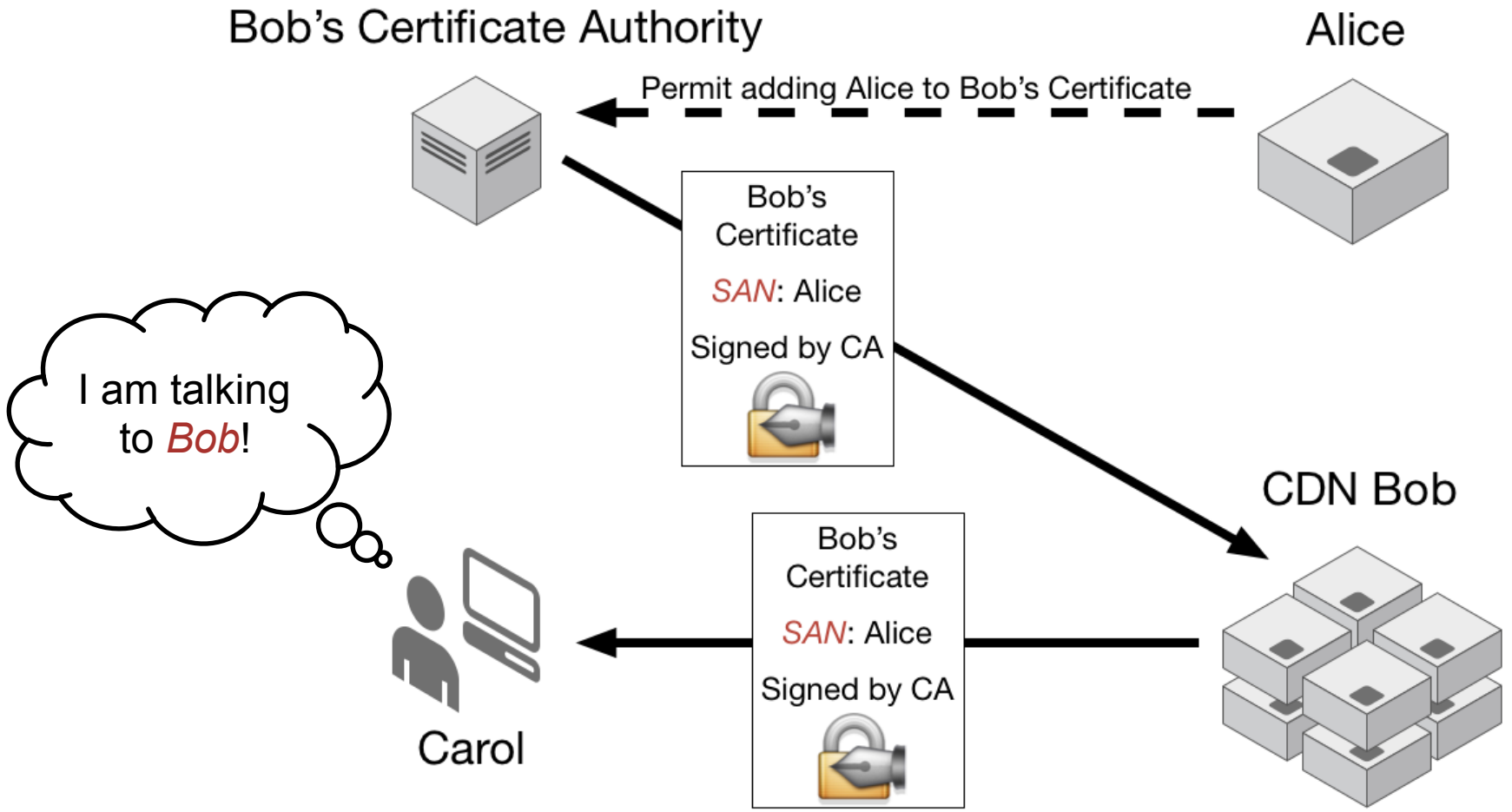
Current Solutions: “Custom Certificate” (Shared Private Key)



Current Solutions: “Custom Certificate” (Shared Private Key)

- Private key is given to CDN and distributed inside the CDN
- No guaranteed, independent and efficient revocation possible
- No efficient way to create the delegation
- No complete identification
- Much larger attack surface on private key, much more can go wrong!

Current Solutions: Shared Certificate



Current Solutions: Shared Certificate

- Observed 1’198 sites using shared certificates for 3 months
- 1’865 certificate changes observed
  - Mainly due to joining and leaving customers
- *None* of the abandoned shared certificates were *revoked*
  - Checked against CRL and OCSP (“Online Certificate Status Protocol”) servers
- Each shared certificate contains *many different domains* at once!
  - Great value for an attacker to get hold of!

Current Solutions: Statistics

- Interested in DNS based request routing
- 20 well-known CDN providers surveyed
- 19 CDNs support HTTPS with DNS (most use `CNAME`)
- 10’721 sites evaluated

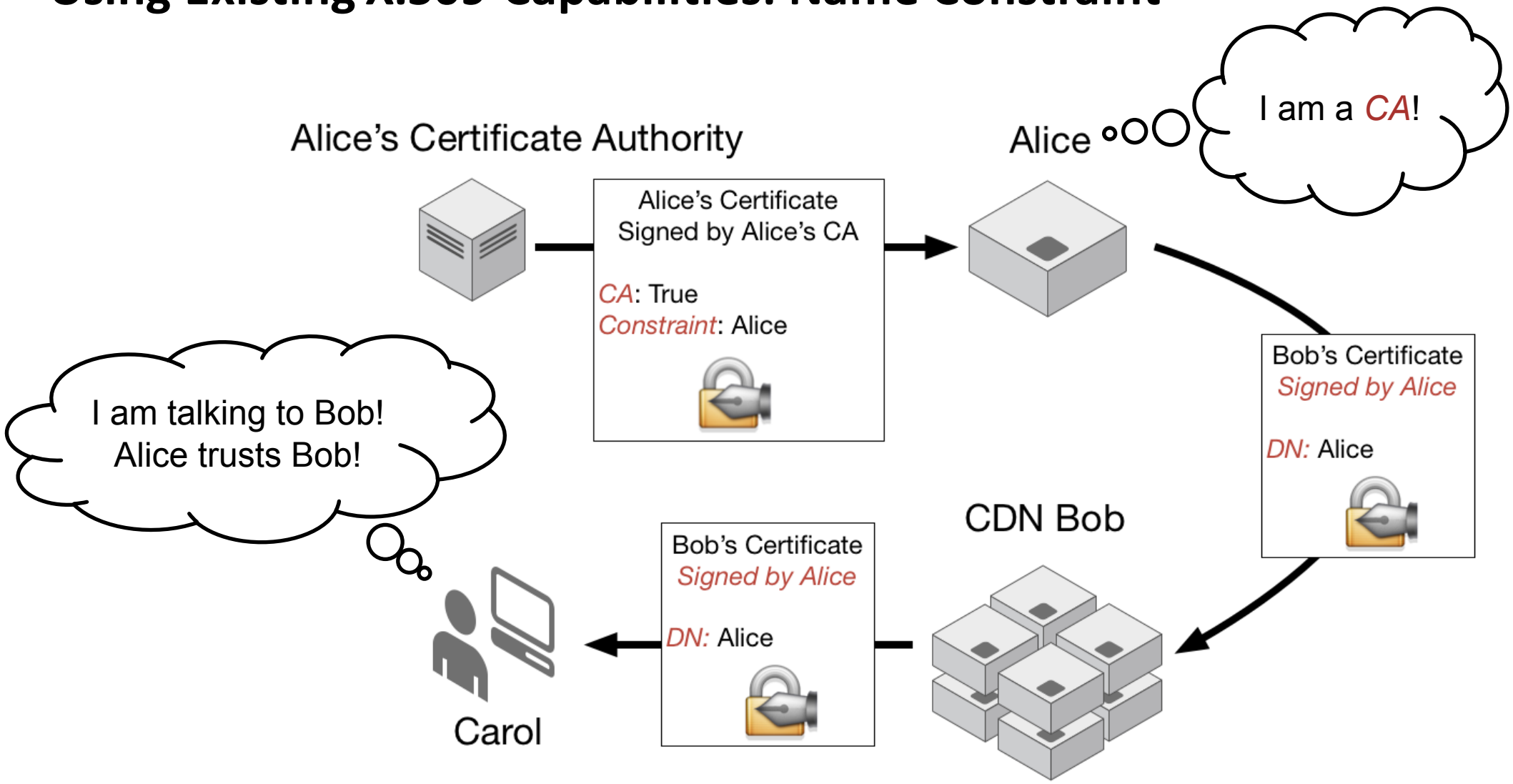
Current Solutions: Statistics

HTTPS Status		# of web sites	%
Valid certificate	Custom Certificate	2’152	20.1%
	Shared Certificate	1’198	11.1%
Invalid certificate	HTTP Status 200	1’637	15.3%
	Others	5’734	53.5%
	Total	*10’721*	*100%*

Presentation Topics

- Problem and requirements for possible solution
- Current solutions in practice
- Possible X.509 out-of-box solution
- Proposed solution
- Securing back-end communication
- Final remarks

Using Existing X.509 Capabilities: Name Constraint



X.509 Name Constraint: Found Problems

- Missing support in libraries / browsers

- Standard conform:
  - Alice can sign a certificate for `alice.com` ***and*** `google.com`
    (single certificate with two `CN` fields)
  - Works even if Alice's CA restricts Alice to sign only certificates for `alice.com`
  - This certificate is valid for `google.com` and will be accepted by all browsers
    (except Firefox)

X.509 Name Constraint: Even more problems…

- Extensive security requirements for a subordinate CA
  - Costs a lot!
- No positive incentive for current CAs
  - Less certificates sold ⇒ smaller income
- Out of 1.5 million HTTPS certificates *none* contained Name Constraints[1]
- Problems outweigh the benefits!

[1] ICSI Notary Certificate Database

Presentation Topics

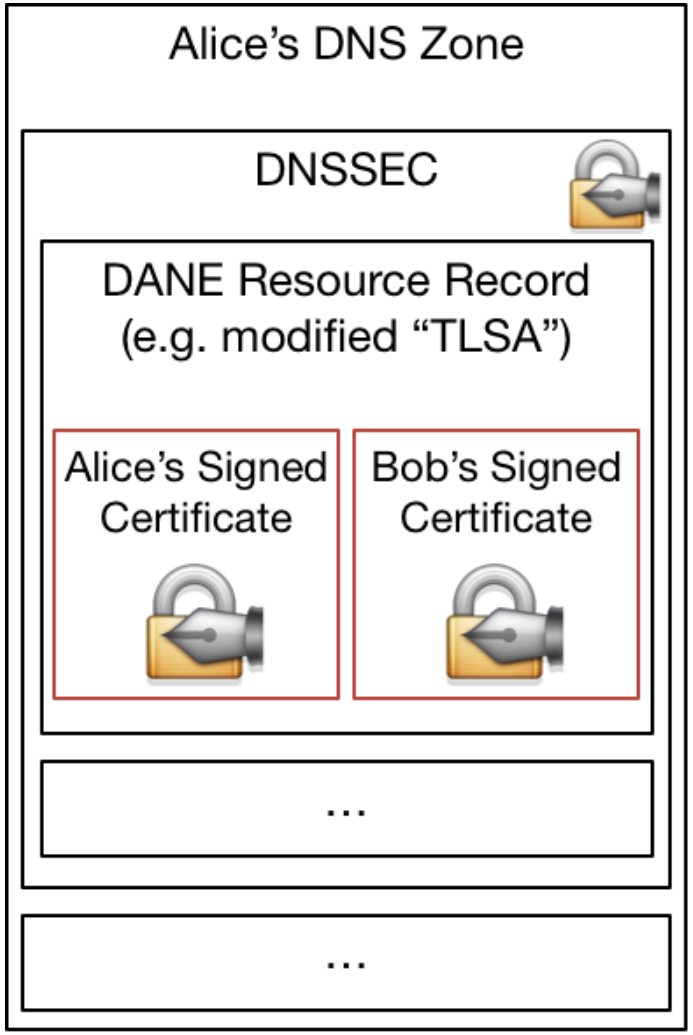
- Problem and requirements for possible solution
- Current solutions in practice
- Possible X.509 out-of-box solution
- Proposed solution
- Securing back-end communication
- Final remarks

Proposed Solution: Necessary Technology

- Leverage upcoming *DNS* technology
- DNSSEC: *integrity* and *authenticity* for DNS Resource Records
- DANE: *Bind* X.509 *certificates to names* using DNS
  - Requires DNSSEC

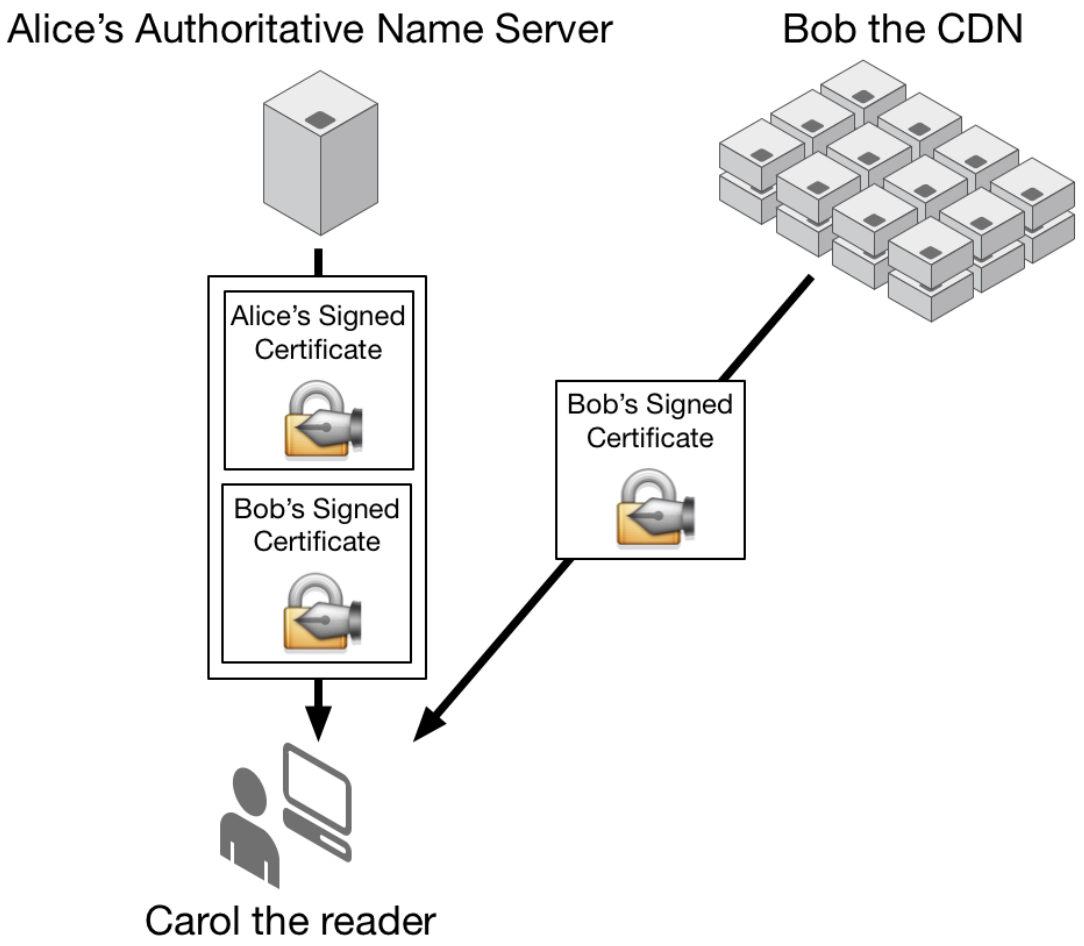
Proposed Solution

- Alice adds *Alice's certificate and Bob's certificate* as a DANE Resource Record to Alice's DNS Zone
- Unforgeable due to DNSSEC
- Issuing and revoking delegation by changing Alice's DNS Zone
- Complete identification possible due to both certificates being shipped



Proposed Solution: Client's View

- Carol knows:
  - Alice's certificate
  - Alice delegates to Bob
  - Expected certificate from Bob
- Carol trusts Bob's certificate
- Carol can see security indicators based on Alice's Certificate



Proposed Solution: Discussed Drawbacks

- Replay attack using stale DNS Resource Record possible
  - Inherent problem in DNSSEC
  - Possible solutions: short DNSSEC signature expiration dates, DNSCurve[1]
- At least one more key to protect due to DNSSEC
- Overhead due to certificate chain for Alice's certificate + CDN's certificate
  - Nearly all responses likely to be larger than 4'096 bytes leads to "first-UDP-then-TCP" overhead[2]
  - Likely to become general problem for DANE/DNSSEC
  - Possible solution: Use TCP right away

[1] http://dnscurve.org/in-benefits.html

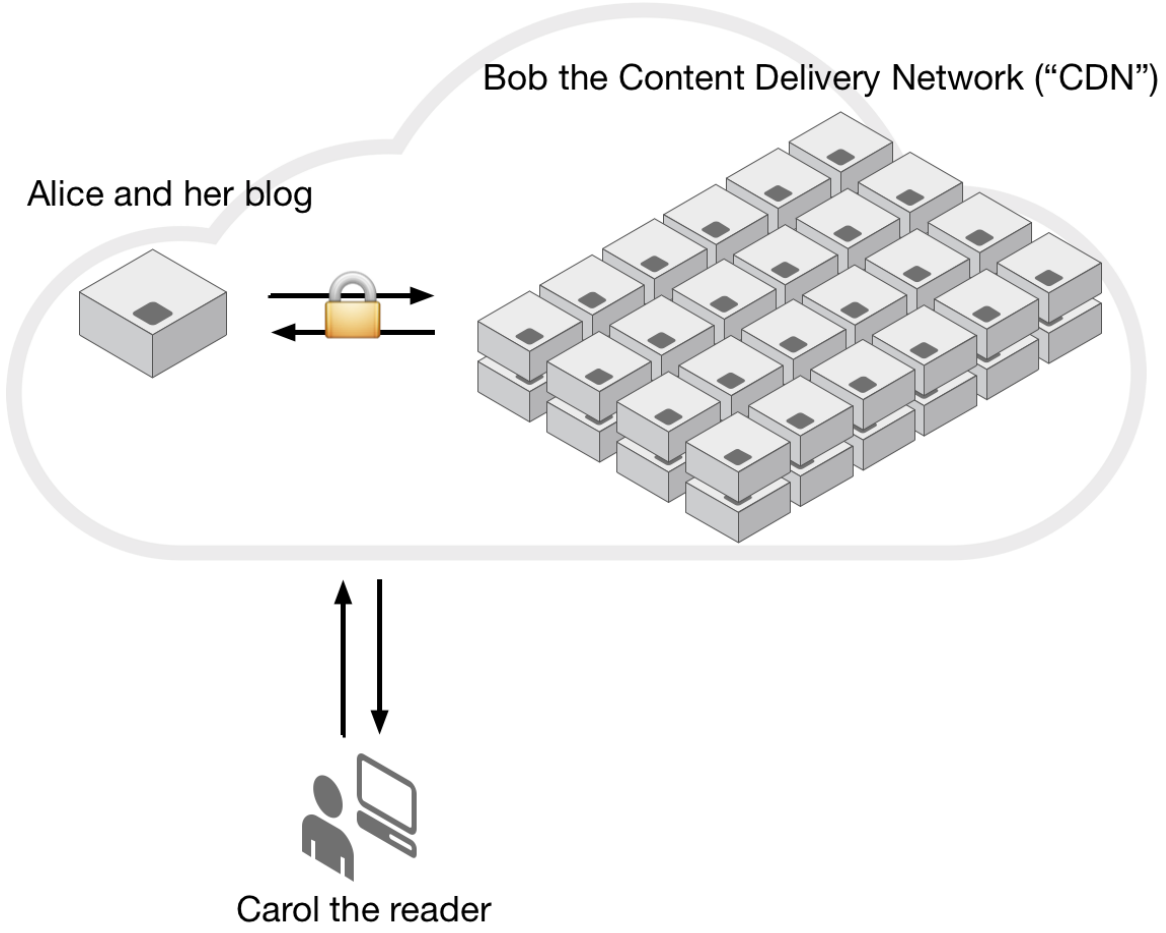
[2] Damas, J., Graff, M., and P. Vixie, "Extension Mechanisms for DNS (EDNS(0))", STD 75, RFC 6891, DOI 10.17487/RFC6891, April 2013, http://www.rfc-editor.org/info/rfc6891

Presentation Topics

- Problem and requirements for possible solution
- Current solutions in practice
- Possible X.509 out-of-box solution
- Proposed solution
- Securing back-end communication
- Final remarks

What about the backend? Well, theoretically it's simple

- Simple, just use HTTPS
- *However*, out of 5 CDNs…
  - 2 do not support/use HTTPS
  - 2 do not perform certificate authentication
  - 1 does not check `CN` field against domain name



Presentation Topics

- Problem and requirements for possible solution
- Current solutions in practice
- Possible X.509 out-of-box solution
- Proposed solution
- Securing back-end communication
- Final remarks

Final Remarks

- Nearly all TLDs support DNSSEC[1], but is it used by Domain owners?
  - 510'640 out of 120'167'319 .com Domains use DNSSEC, i.e. only 0.42%[2]
- "[…] a ***simple*** extension of DANE […]"
- "[…] our proposal ***broadens the semantics*** of DANE […]"
- What motivates CDNs to push such a feature?
- How to present the danger of current methods to customers?
- How to communicate all this to the user?

[1] ICANN Research "TLD DNSSEC Report" as of 2015-11-20 00:02:19: http://stats.research.icann.org/dns/tld_report/
[2] Statdns "TLD Zone File Statistics – November 2015": www.statdns.com

Thank You

Request-routing Mechanisms

- URL Rewriting
  - Resources used by web site point to CDN
  - E.g. `<img src="smiley.gif">` ⟼ `<img src="//alice.bob.com/smiley.gif">`
- CNAME
  - DNS `CNAME` Resource Record based request-routing
  - Basically an alias for a domain: "Ask for `alice.bob.com` if you want to access `alice.com`"
- Domain Hosting
  - CDN's DNS server acts as Authoritative DNS Server
  - Domains DNS zone managed by CDN