Public Key Infrastructure (PKI)

The Basics of Public Key Infrastructures

- A PKI
  - Binds public key to identity
  - Enables other entities to verify key-identity binding
  - Provides services for management of keys in a distributed system
- Goal:
  - Protect and distribute information that is needed in a widely distributed environment, where the users, resources and stake-holders may all be in different places at different times

The Basics of Public Key Infrastructures cont.

- Consists of
  - Hardware, software, policies, services, programming interfaces, cryptographic algorithms, protocols, users, and utilities
- Provides:
  - Data integrity
  - Data confidentiality
  - Authentication
- Integrates
  - Public key cryptography
  - Digital certificates
  - Certification authorities

Components of PKI

- Certificate/Certification Authority (CA)
  - Confirms the identity of entities by issuing certificates
- Registration Authority (RA)
  - Trusted by CA to authenticate users requesting digital certificates from CA
- Validation Authority / Repository (VA)
  - Provides services used to validate a certificate
  - Database of active digital certificates for a CA
- Archive
  - Stores and protects sufficient information to determine if a digital signature on an old document should be trusted
- Certificates
  - Includes public key, identity, and other information

KEY TERMS

- Authority revocation list (ARL)
- CA certificate
- Certificate
- Certificate Authority (CA)
- Certificate path
- Certificate repository
- Certificate Revocation List (CRL)
- Certificate server
- Certificate signing request (CSR)
- Certification practices statement (CPS)
- Cross-certification certificate

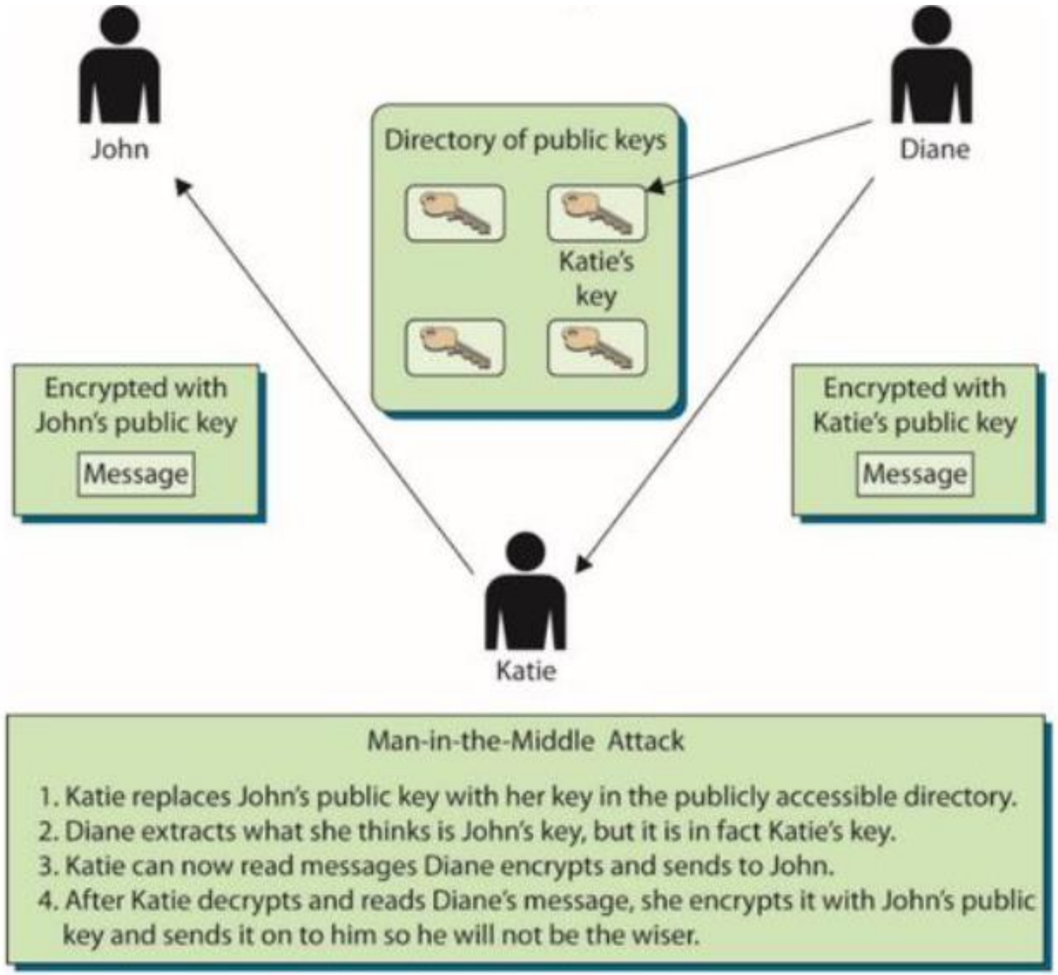
KEY TERMS

- Digital certificate
- Dual control
- End-entity certificate
- Hardware security module (HSM)
- Hierarchical trust model
- Hybrid trust model
- Internet Security Association and Key Management Protocol (ISAKMP)
- Key archiving
- Key escrow
- Key recovery
- Local registration authority (LRA)

KEY TERMS

- Online Certificate Status Protocol (OCSP)
- Peer-to-peer trust model
- Policy certificate
- Public key infrastructure (PKI)
- Registration authority (RA)
- X.509

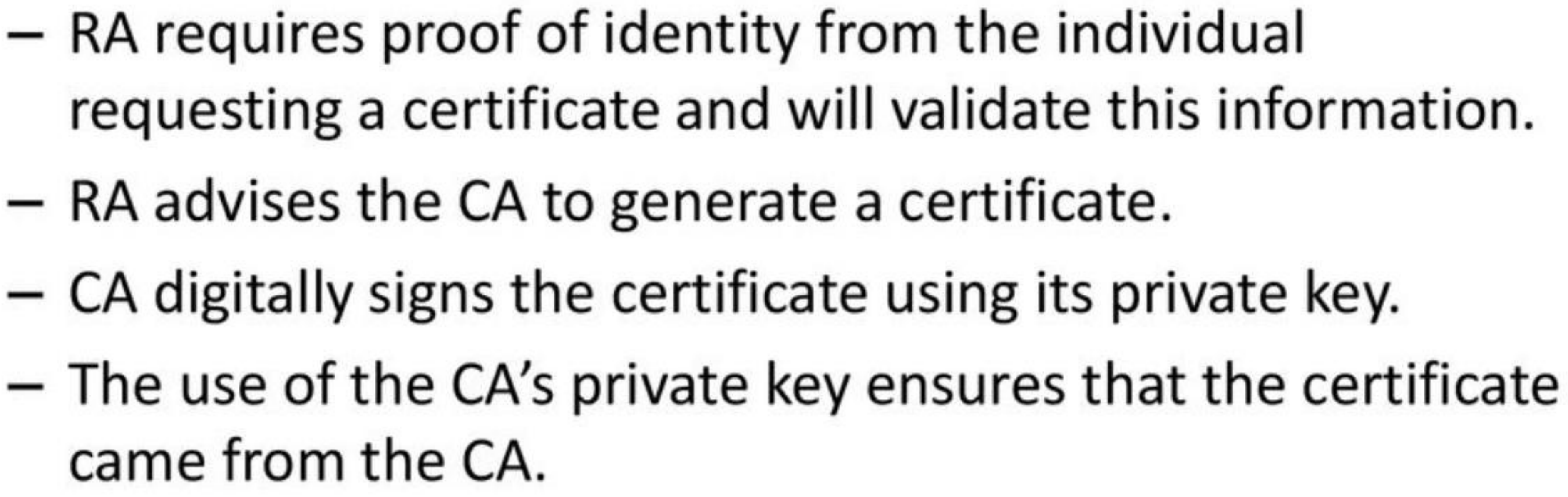
Without PKI, individuals could spoof others’ identities


John
Directory of public keys
Diane
Katie’s key
Encrypted with John’s public key
Message
Encrypted with Katie’s public key
Message
Katie

Man-in-the-Middle Attack

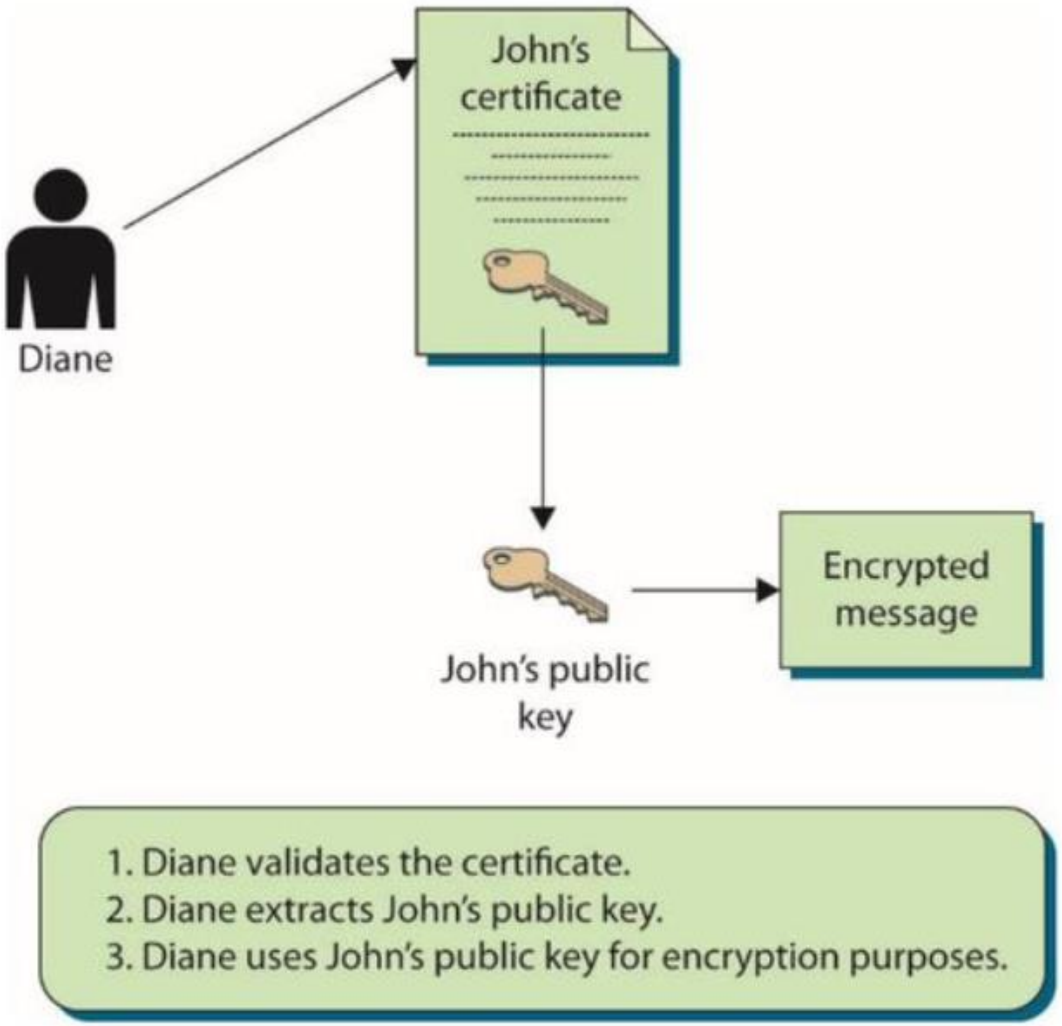
1. Katie replaces John’s public key with her key in the publicly accessible directory.
2. Diane extracts what she thinks is John’s key, but it is in fact Katie’s key.
3. Katie can now read messages Diane encrypts and sends to John.
4. After Katie decrypts and reads Diane’s message, she encrypts it with John’s public key and sends it on to him so he will not be the wiser.

Third-party trust model



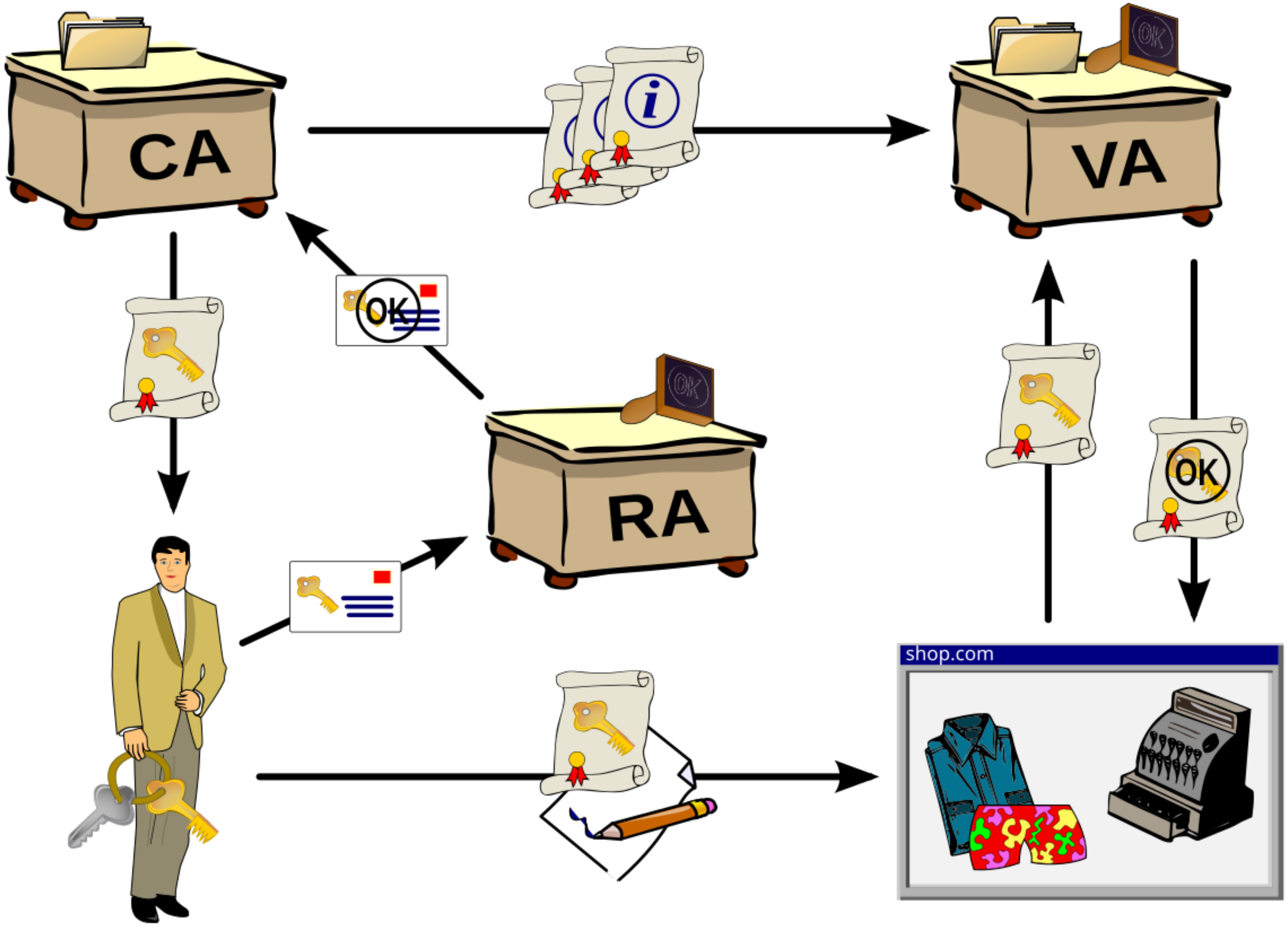
- RA requires proof of identity from the individual requesting a certificate and will validate this information.
- RA advises the CA to generate a certificate.
- CA digitally signs the certificate using its private key.
- The use of the CA’s private key ensures that the certificate came from the CA.

Public keys are components of digital certificates



1. Diane validates the certificate.
2. Diane extracts John's public key.
3. Diane uses John's public key for encryption purposes.

PKI Interaction Diagram



What does Infrastructure really mean?

- Generating key-pairs and validating certificates does not a PKI make
- No 3rd party trusted identifier → trust each other and/or the channel
- PKI provides trust that *you* cannot / don’t provide
- Infrastructure – sustaining groundwork upon which other things can be built.
  - Low level, predictable, uniform
  - Supports high-level applications

Certificate Authorities

- A **certificate authority (CA)** is a trusted authority that certifies individuals' identities and creates electronic documents indicating that individuals are who they say they are.
- The electronic document is referred to as a **digital certificate**, and it establishes an association between the subject's identity and a public key.
- The private key that is paired with the public key in the certificate is stored separately.

Certificate Authorities

- If one CA component is compromised, it can negatively affect the CA integrity overall.
- Every CA should have a **certification practices statement** (**CPS**).
    - It outlines how identities are verified; the steps the CA follows to generate, maintain, and transmit certificates; and why the CA can be trusted to fulfill its responsibilities.
- A **certificate server** is the actual service that issues certificates based on the data provided during the initial registration process.

Registration Authorities

- A **registration authority (RA)** is the PKI component that accepts a request for a digital certificate and performs the necessary steps of registering and authenticating the person requesting the certificate.
- The authentication requirements differ depending on the type of certificate being requested.
- Most CAs offer a series of classes of certificates with increasing trust by class.

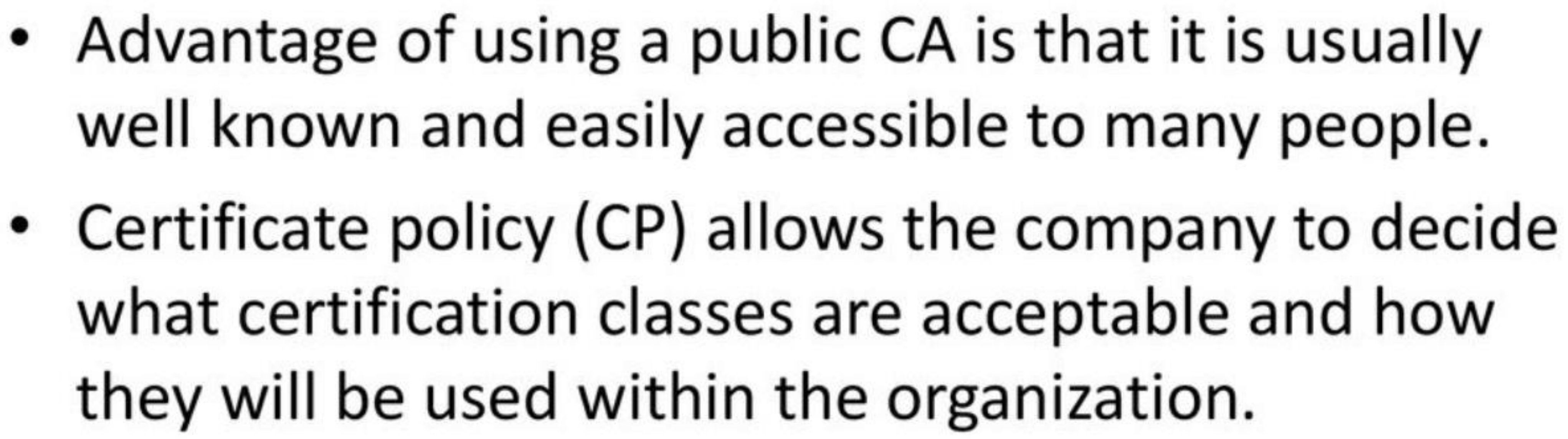
Local Registration Authorities

- A **local registration authority (LRA)** performs the same functions as an RA.
  - It is closer to the end users and reduces WAN traffic.
  - It is implemented in companies with their own internal PKIs and in companies with distributed sites
  - It performs identification, verification, registration functions; sends request, along with the user's public key, to a centralized CA so that the certificate can be generated.
  - It acts as an interface between the users and the CA.
  - LRAs simplify the RA/CA process for entities that desire certificates only for in-house use.

Public Certificate Authorities

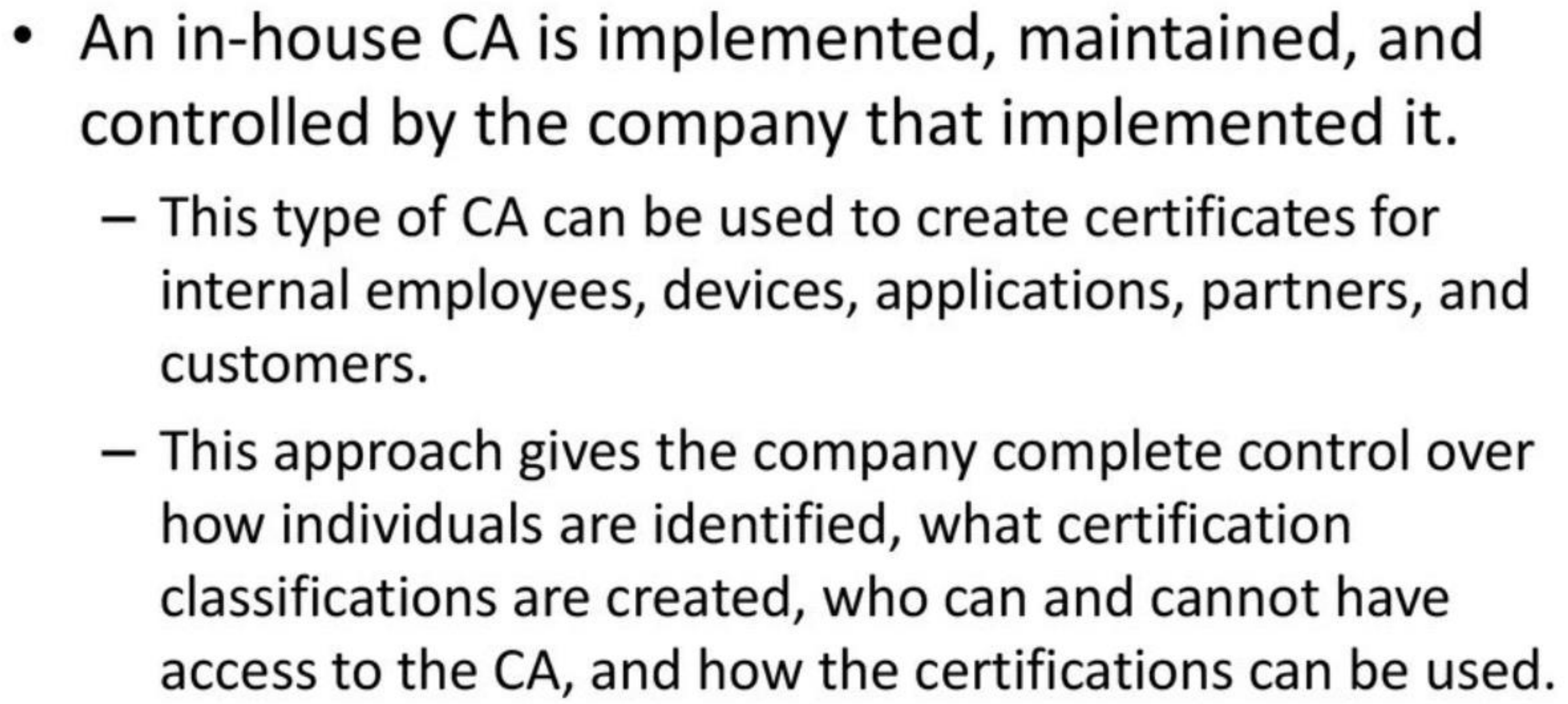
- Public CAs are already established and being used by many other individuals and companies.
  - Specialize in verifying individual identities and creating and maintaining their certificates
  - Issue certificates that are not bound to specific companies or departments
- Examples of public CAs include:
  - VeriSign (including GeoTrust and Thawte), Entrust, and Go Daddy

Public Certificate Authorities

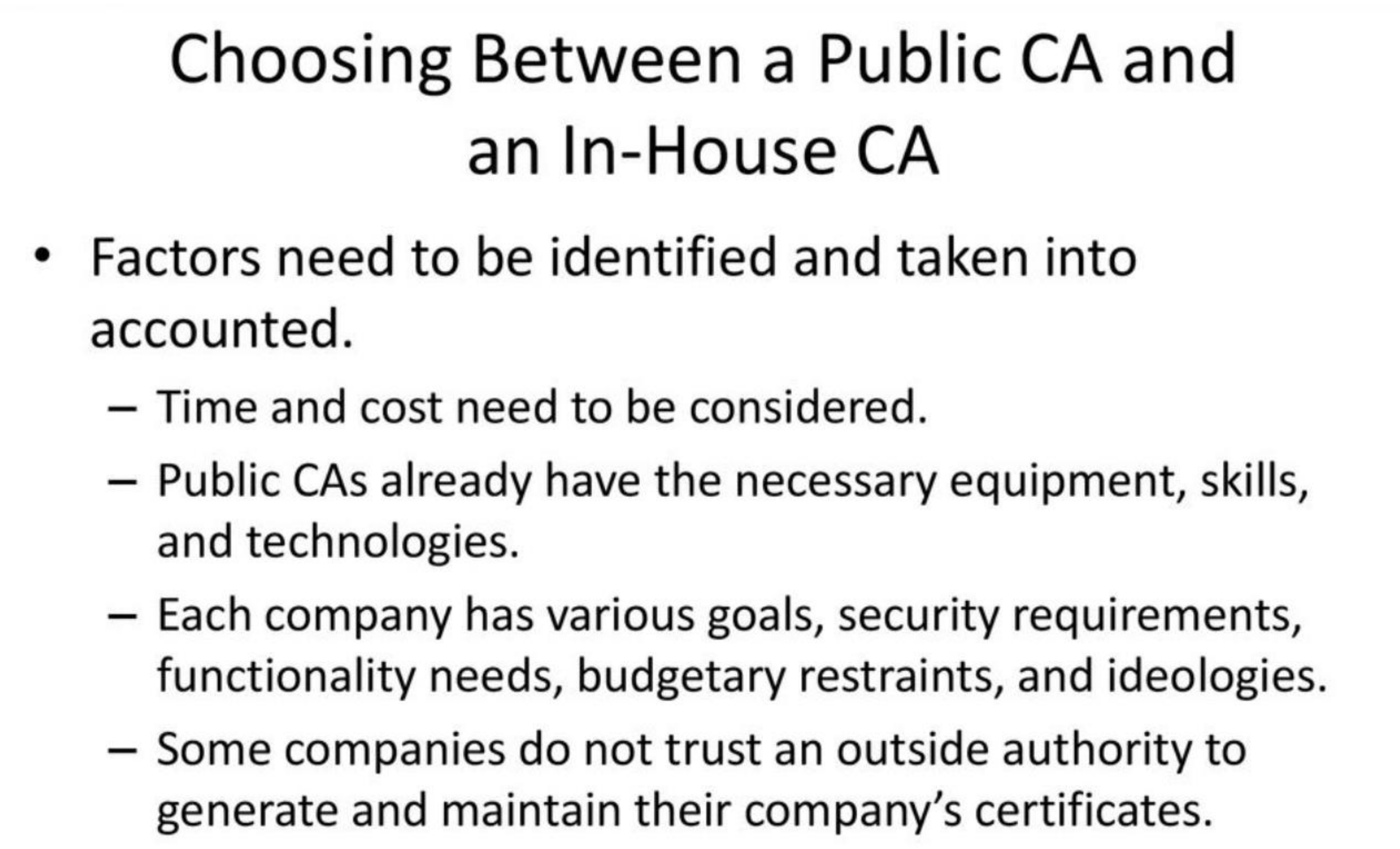


- Advantage of using a public CA is that it is usually well known and easily accessible to many people.
- Certificate policy (CP) allows the company to decide what certification classes are acceptable and how they will be used within the organization.

In-House Certificate Authorities



- An in-house CA is implemented, maintained, and controlled by the company that implemented it.
  - – This type of CA can be used to create certificates for internal employees, devices, applications, partners, and customers.
  - – This approach gives the company complete control over how individuals are identified, what certification classifications are created, who can and cannot have access to the CA, and how the certifications can be used.



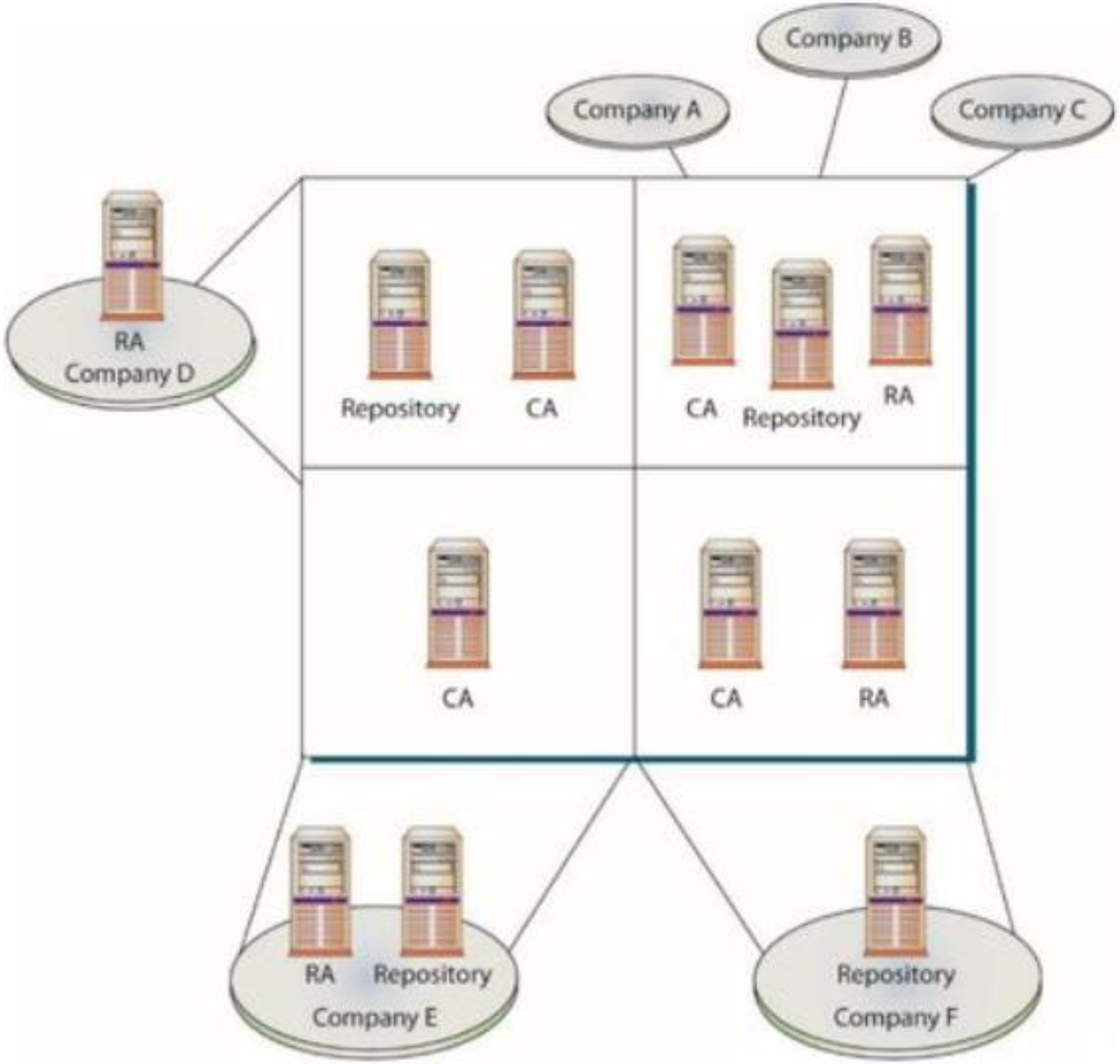
Choosing Between a Public CA and an In-House CA

- Factors need to be identified and taken into accounted.
  - Time and cost need to be considered.
  - Public CAs already have the necessary equipment, skills, and technologies.
  - Each company has various goals, security requirements, functionality needs, budgetary restraints, and ideologies.
  - Some companies do not trust an outside authority to generate and maintain their company’s certificates.

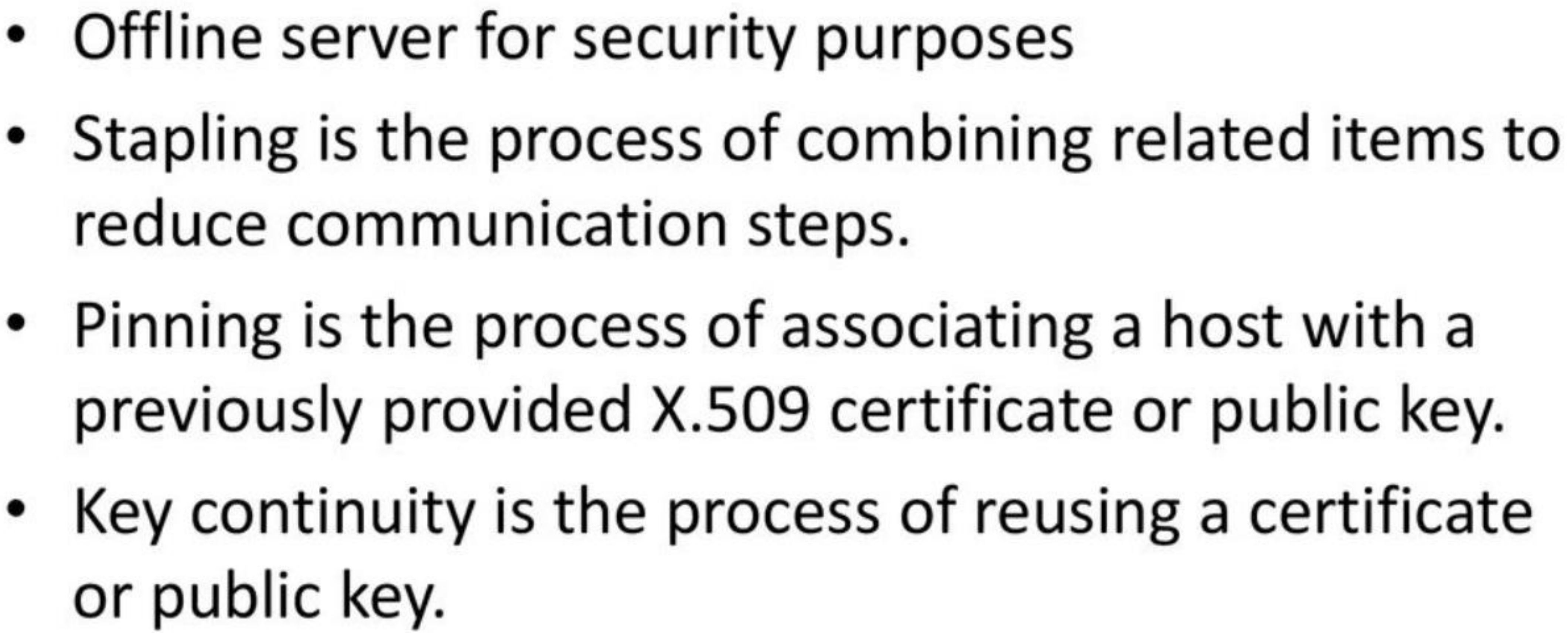
Outsourced Certificate Authorities

- Usually, the more complex parts (the CA, RA, CRL, and key recovery mechanisms) are outsourced.
  - Level of trust the company willing to give to service provider and level of risk willing to accept must be determined.
  - Large vertical markets can have their own outsourced PKI environments set up to split costs and follow industry-specific standards.
  - A set of standards can be drawn up about how each different facility should integrate its own infrastructure and how it should integrate with the centralized PKI components.

Outsourced Certificate Authorities



Outsourced Certificate Authorities



- Offline server for security purposes
- Stapling is the process of combining related items to reduce communication steps.
- Pinning is the process of associating a host with a previously provided X.509 certificate or public key.
- Key continuity is the process of reusing a certificate or public key.

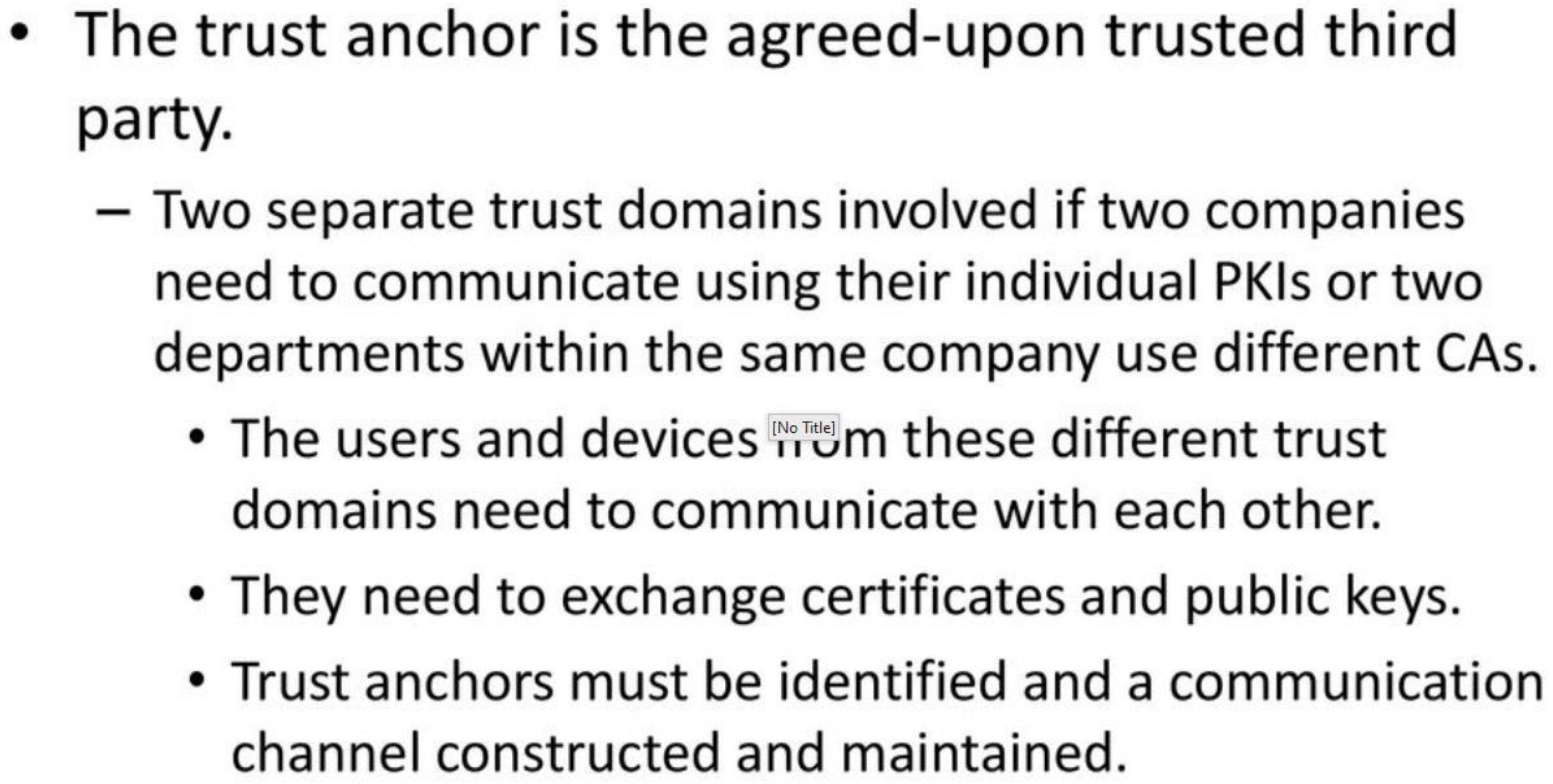
Pinning

- Process of associating a host with a previously provided X.509 certificate or public key.
  - Save the cert for later
- If pinned cert and host cert (from TLS) don’t match → refuse to connect
- Protects against misissuance, CA compromise, man-in-the-middle

Trust Models

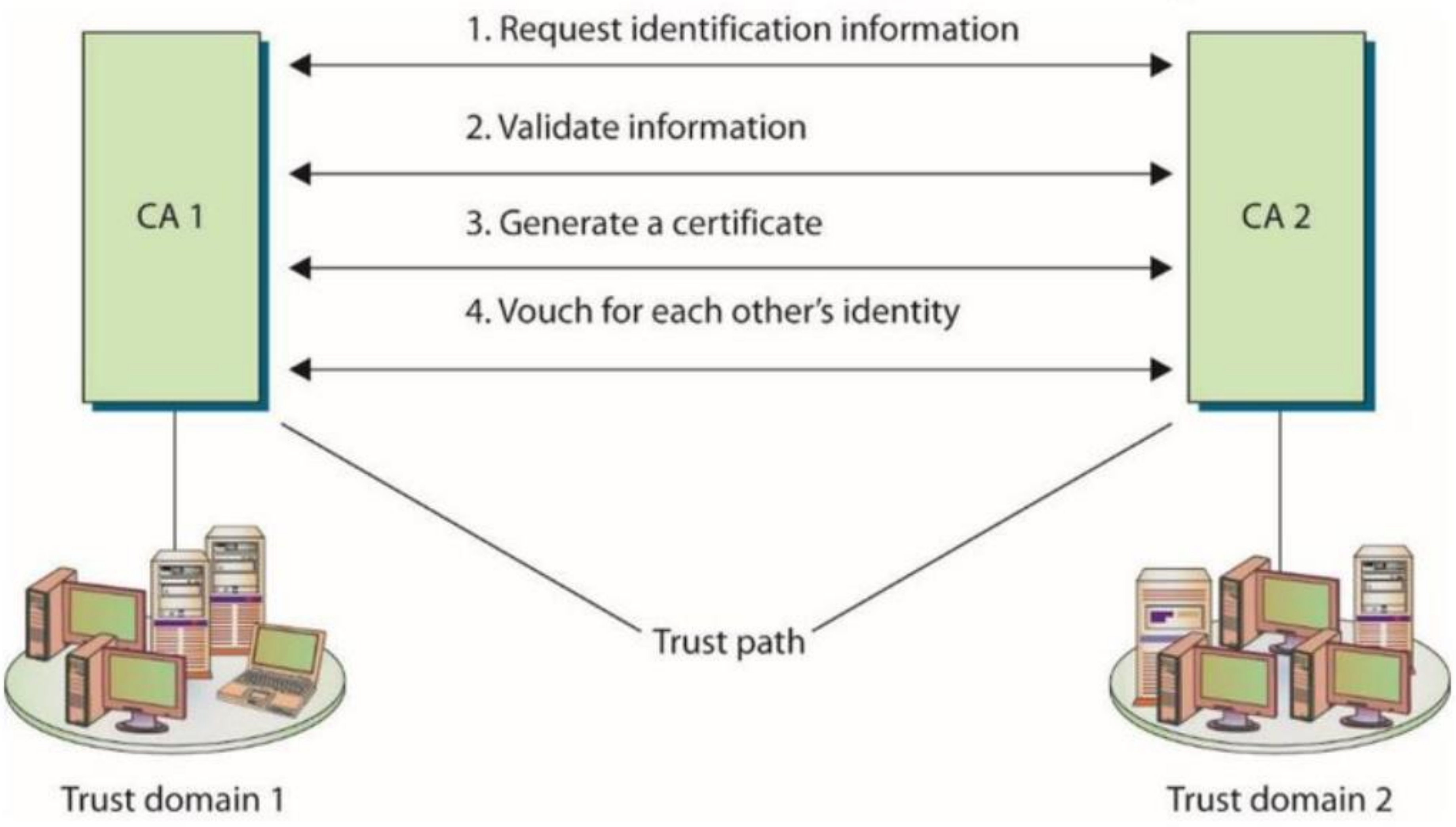
- A trust domain is a construct of systems, personnel, applications, protocols, technologies, and policies that work together to provide a certain level of protection.
  - Most trust domains need to communicate with other, less-trusted domains.
  - Must figure out how much two different domains should trust each other, and how to implement and configure an infrastructure that would allow these two domains to communicate in a way that will not allow security compromises or breaches.

Trust Anchor



- The trust anchor is the agreed-upon trusted third party.
  - Two separate trust domains involved if two companies need to communicate using their individual PKIs or two departments within the same company use different CAs.
    - The users and devices from these different trust domains need to communicate with each other.
    - They need to exchange certificates and public keys.
    - Trust anchors must be identified and a communication channel constructed and maintained.

Trust Relationship


1. Request identification information
2. Validate information
3. Generate a certificate
4. Vouch for each other's identity
CA 1
CA 2
Trust path
Trust domain 1
Trust domain 2

Trust Models

- Trust models describe and outline the trust relationships between the different CAs and different environments.
  - Indicate where the trust paths reside
- Trust models and paths need to be thought out before implementation.
  - Restrict and control access properly
  - Ensure as few trust paths as possible are used
- Several different trust models can be used:
  - Hierarchical, peer-to-peer, and hybrid models

Trust Models

- A trust relationship must be established between two issuing authorities (CAs).
  - CA issues a certificate for the other CA’s public key.
  - Each CA validates the other CA’s identification info and generates a certificate containing a public key for that CA.
    - A trust path established between the two entities.
    - The trust path can be unidirectional or bidirectional – either the two CAs trust each other (bidirectional) or only one trusts the other (unidirectional).
- Certificate chain is a chain of trust

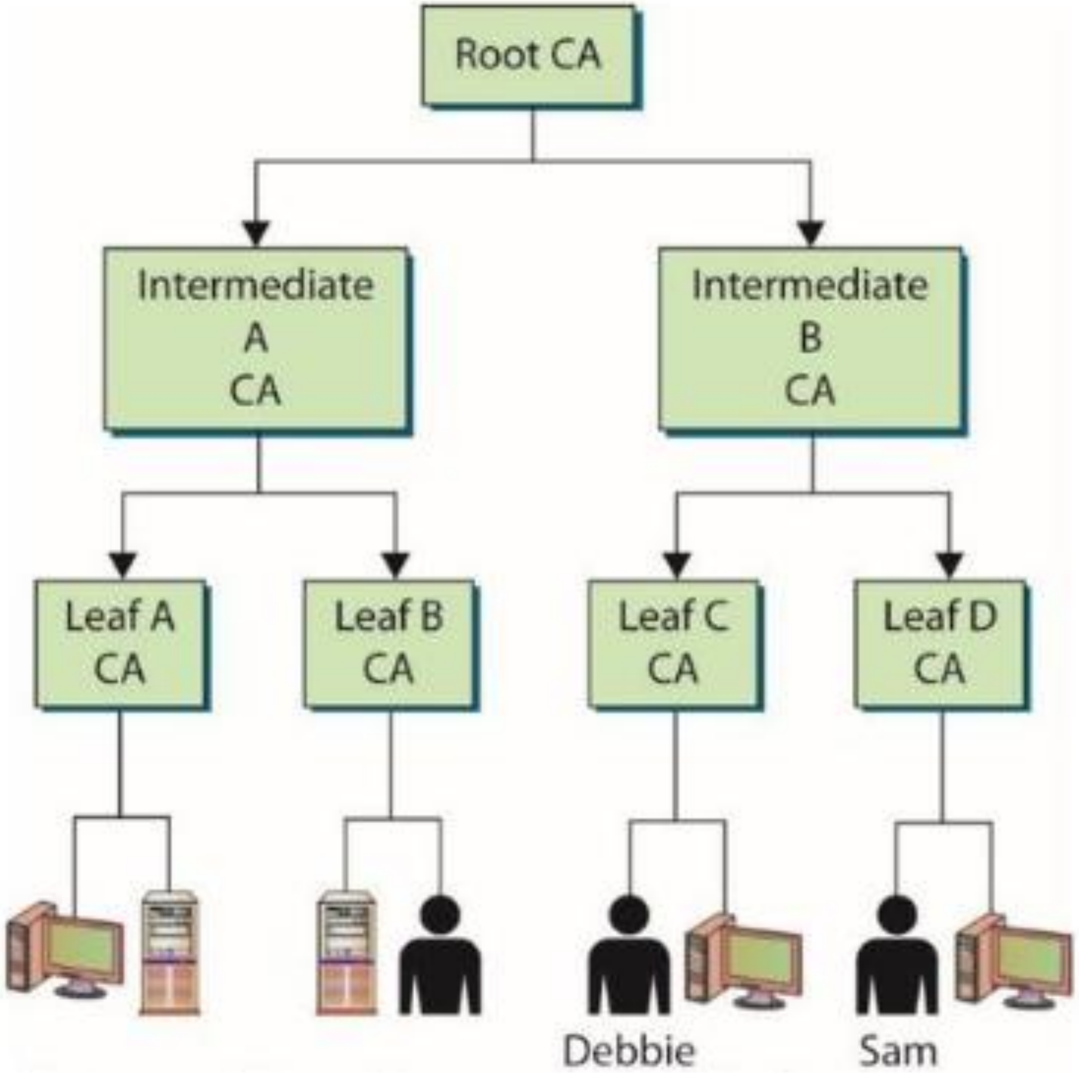
Certificate Chaining

- Certs bind identity to public key
- Why trust a cert?
  - Because of the chain
- Certificate chain is a chain of trust
- Chain/Intermediate certs sit in the middle
- Root cert sits at the top/end of the chain
  - Root cert self-signed by root CA
- Valid sigs + trusted root = trusted certs

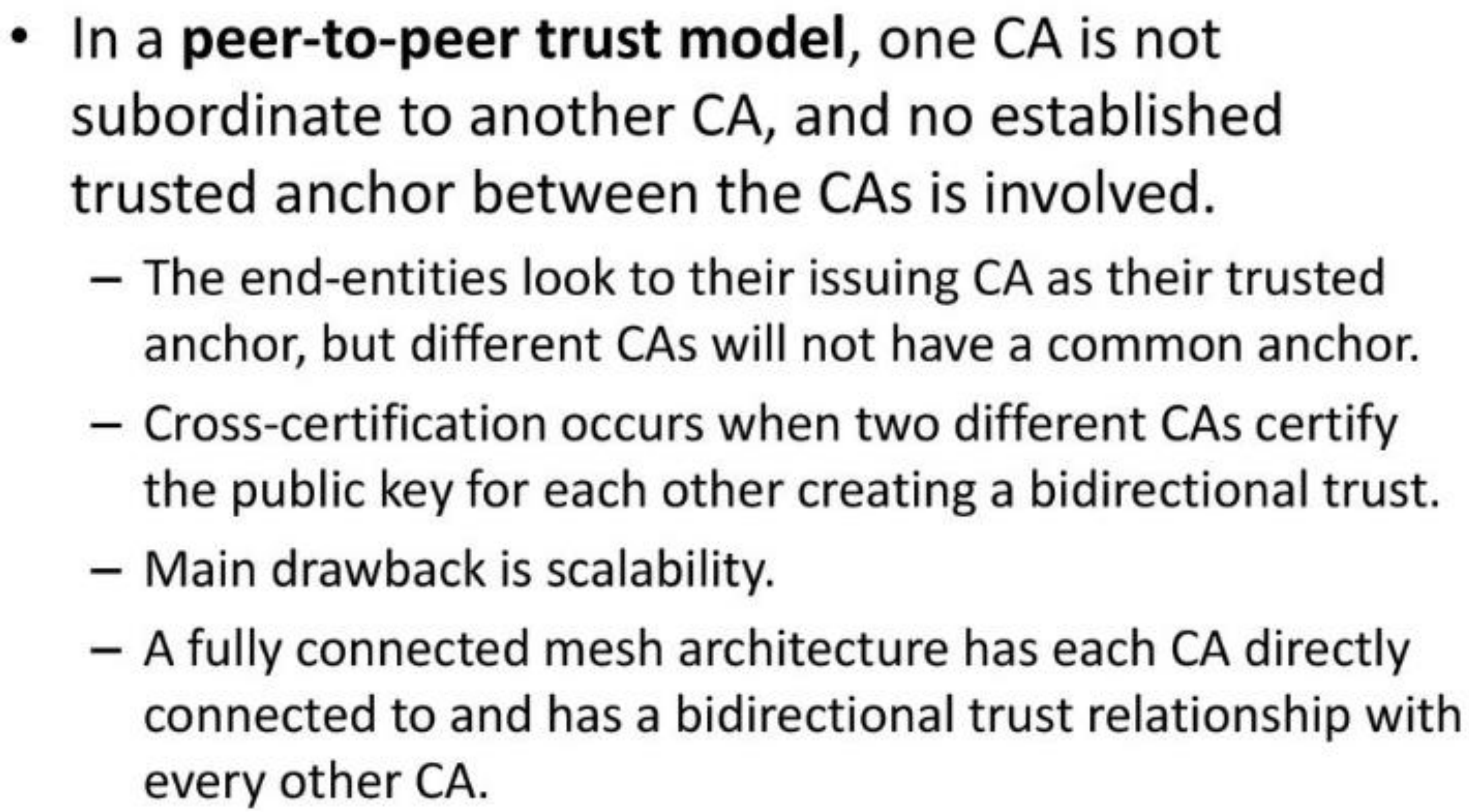
Trust Models

- The **hierarchical trust model** is a basic hierarchical structure that contains a root CA, intermediate CAs, leaf CAs, and end-entities.
  - The configuration is that of an inverted tree.
  - The root CA is the ultimate trust anchor for all other entities in this infrastructure.
    - Root CA generates certificates for the intermediate CAs, which in turn generate certificates for the leaf CAs; leaf CAs generate certificates for the end-entities.
    - *Subordinate CAs* are subordinate to the CA they reference.

Hierarchical Trust Models

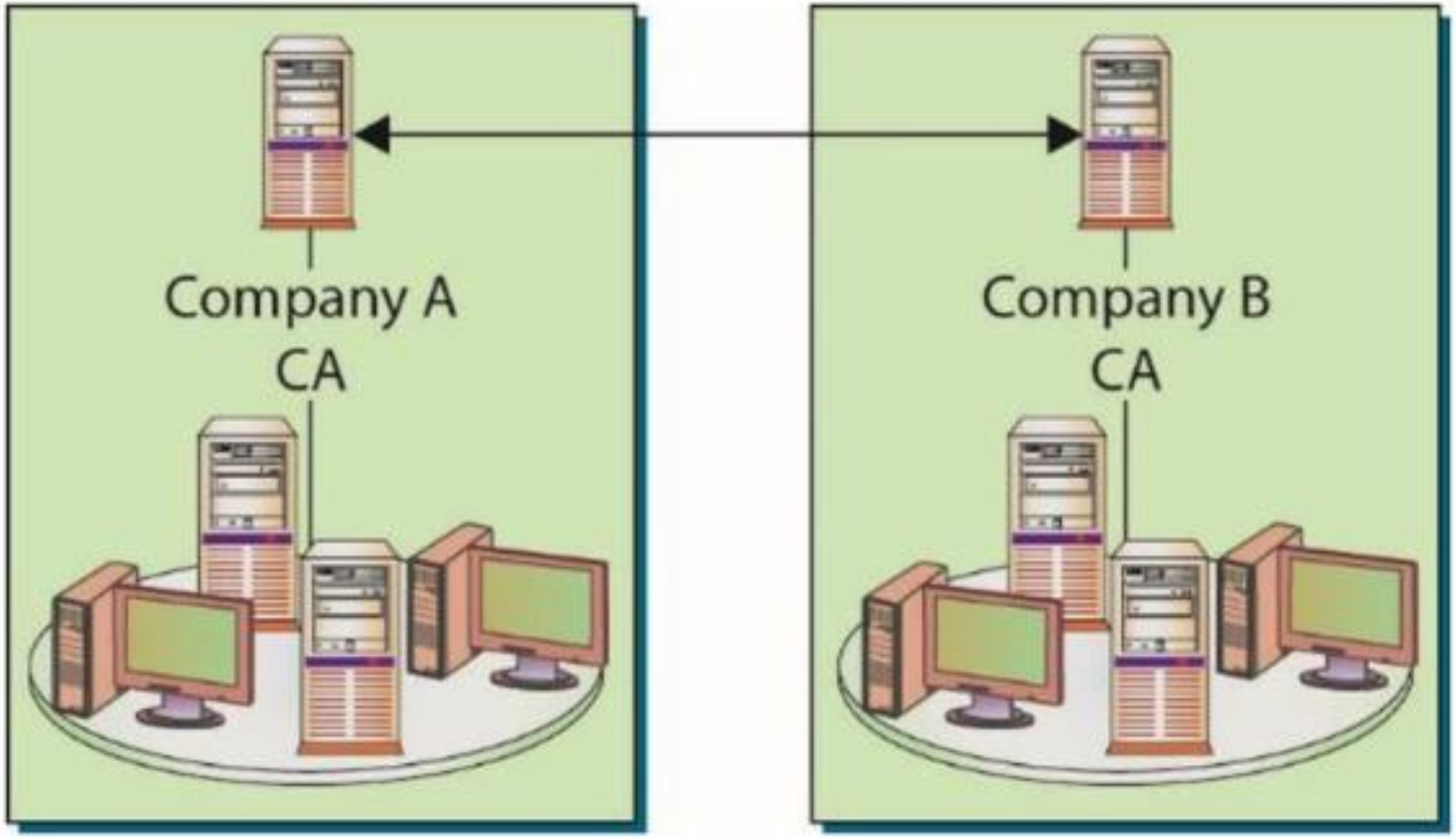


Trust Models

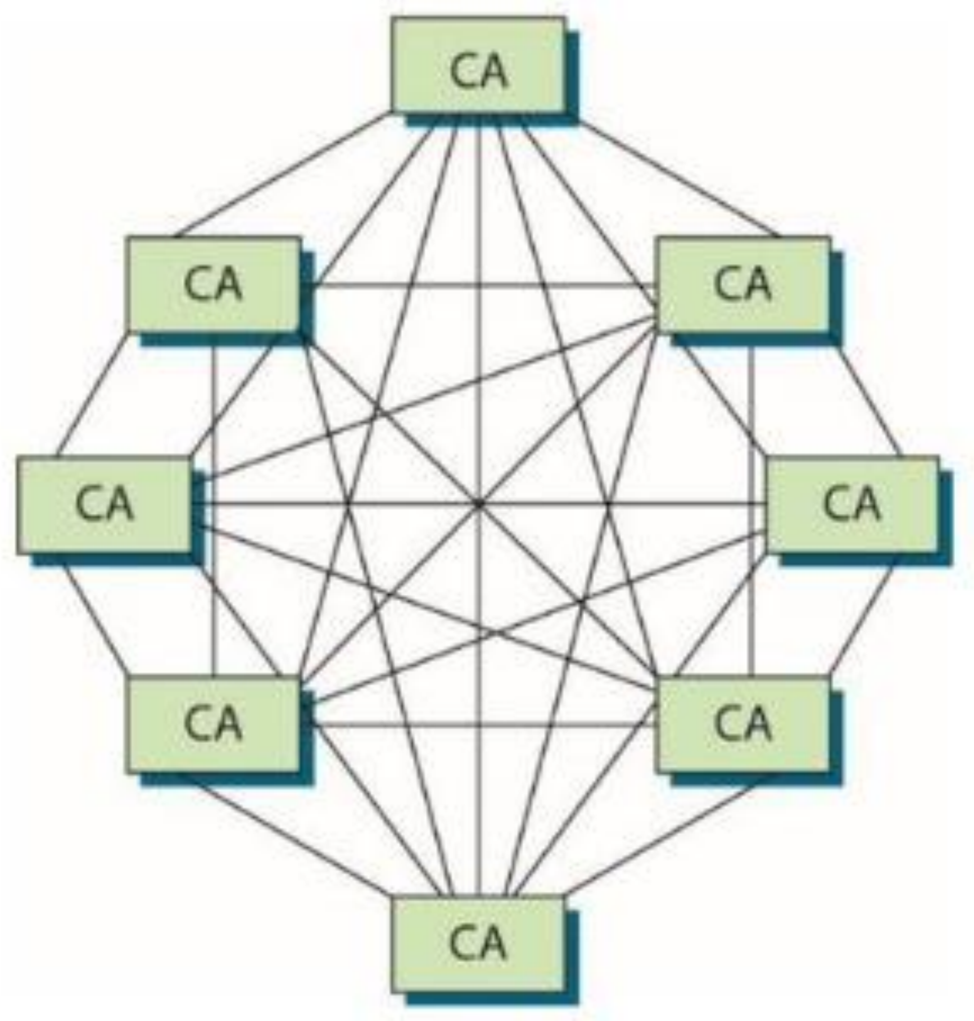


- In a **peer-to-peer trust model**, one CA is not subordinate to another CA, and no established trusted anchor between the CAs is involved.
  - The end-entities look to their issuing CA as their trusted anchor, but different CAs will not have a common anchor.
  - Cross-certification occurs when two different CAs certify the public key for each other creating a bidirectional trust.
  - Main drawback is scalability.
  - A fully connected mesh architecture has each CA directly connected to and has a bidirectional trust relationship with every other CA.

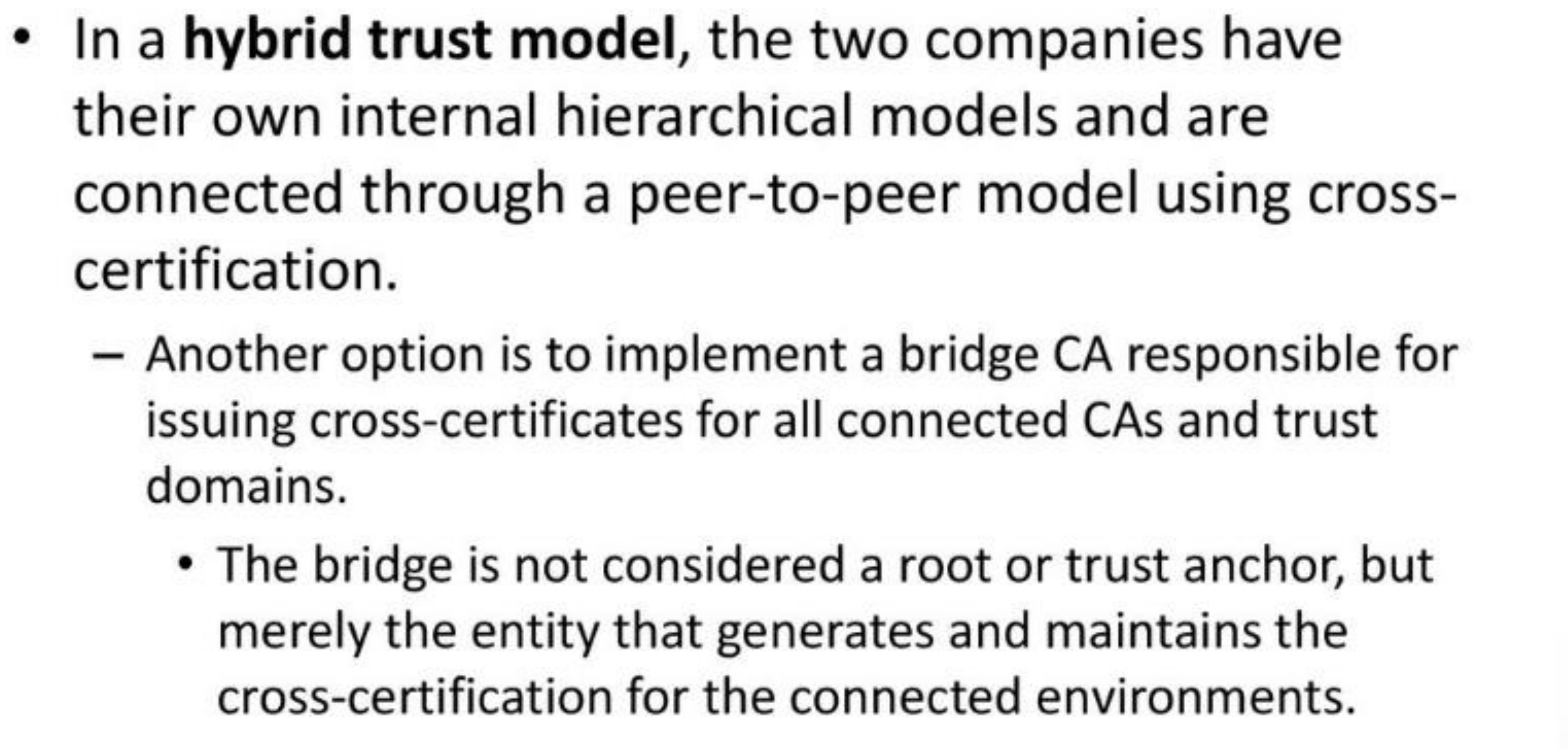
Trust Models



Trust Models

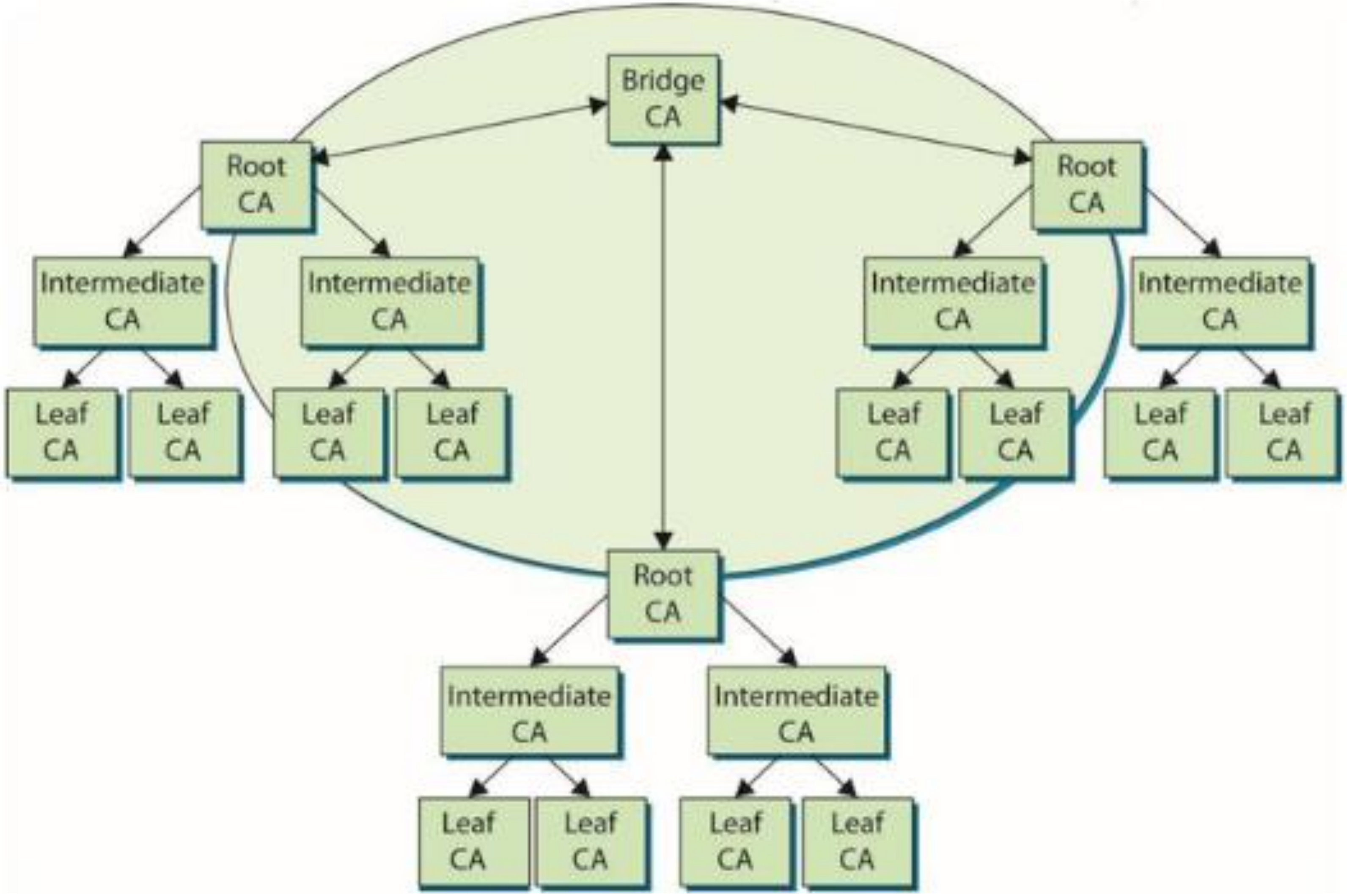


Trust Models



- In a **hybrid trust model**, the two companies have their own internal hierarchical models and are connected through a peer-to-peer model using cross-certification.
  - Another option is to implement a bridge CA responsible for issuing cross-certificates for all connected CAs and trust domains.
    - The bridge is not considered a root or trust anchor, but merely the entity that generates and maintains the cross-certification for the connected environments.

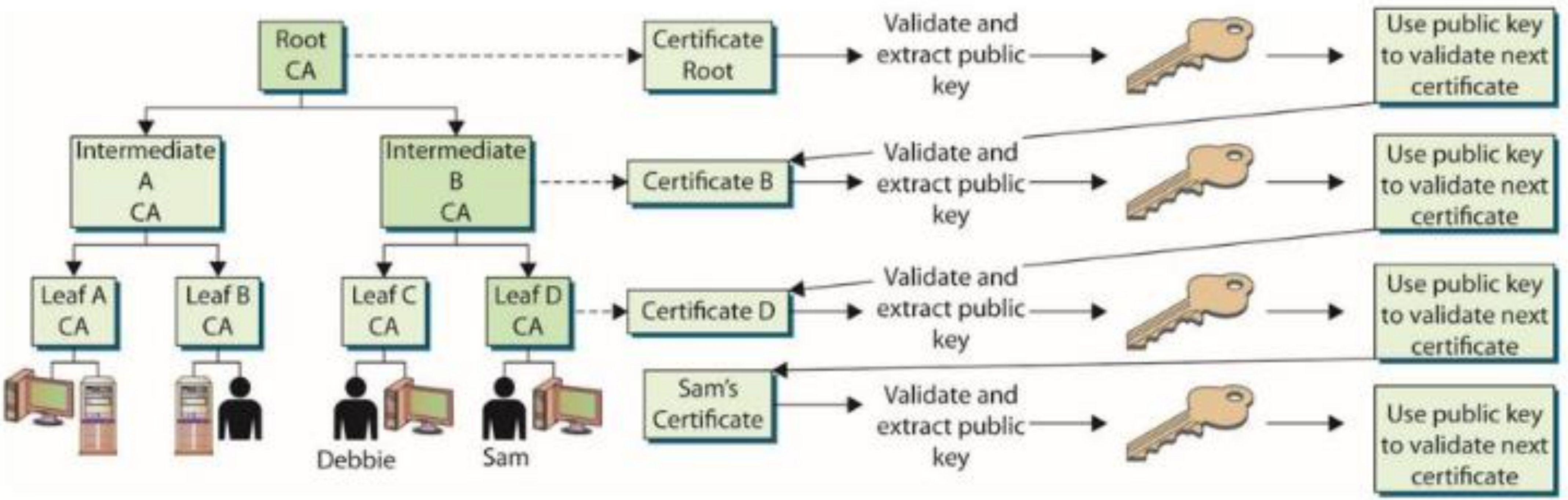
Trust Models



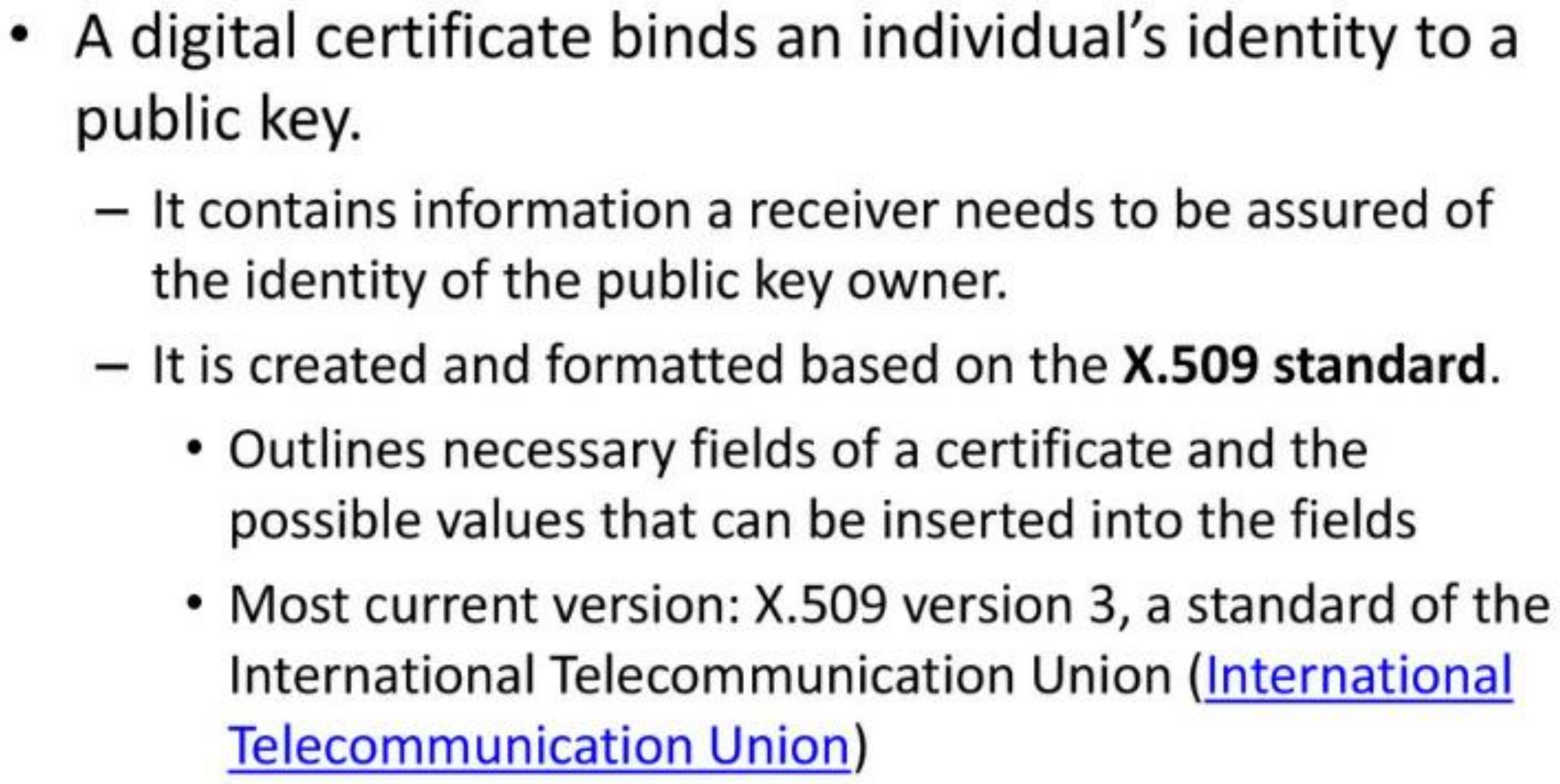
Trust Models

- Walking the certificate path
  - Following the **certificate path** means the client software had to continue to track down and collect certificates until it came upon a self-signed certificate.
  - A *self-signed certificate* indicates that it was signed by a root CA.
  - Simplistic trust model works well within an enterprise that easily follows a hierarchical organizational chart.
    - Many companies cannot use this trust model because different departments or offices require their own trust anchors.

Walking the certificate path


Root CA
Certificate Root
Validate and extract public key
Use public key to validate next certificate
Intermediate A CA
Intermediate B CA
Certificate B
Validate and extract public key
Use public key to validate next certificate
Leaf A CA
Leaf B CA
Leaf C CA
Leaf D CA
Certificate D
Validate and extract public key
Use public key to validate next certificate
Sam's Certificate
Validate and extract public key
Use public key to validate next certificate
Debbie
Sam

Digital Certificates



- A digital certificate binds an individual's identity to a public key.
  - It contains information a receiver needs to be assured of the identity of the public key owner.
  - It is created and formatted based on the **X.509 standard**.
    - Outlines necessary fields of a certificate and the possible values that can be inserted into the fields
    - Most current version: X.509 version 3, a standard of the International Telecommunication Union (International Telecommunication Union)

Digital Certificates

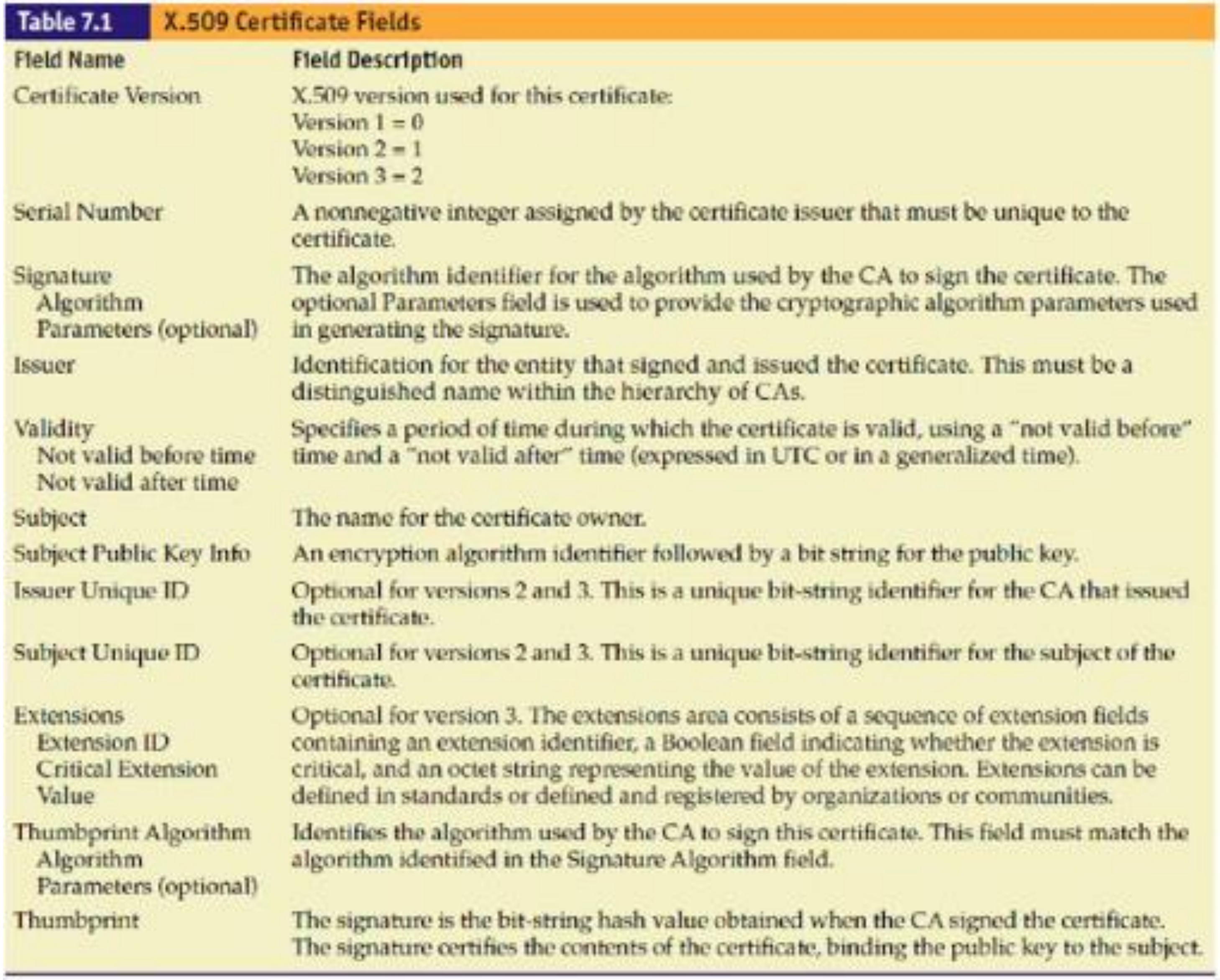
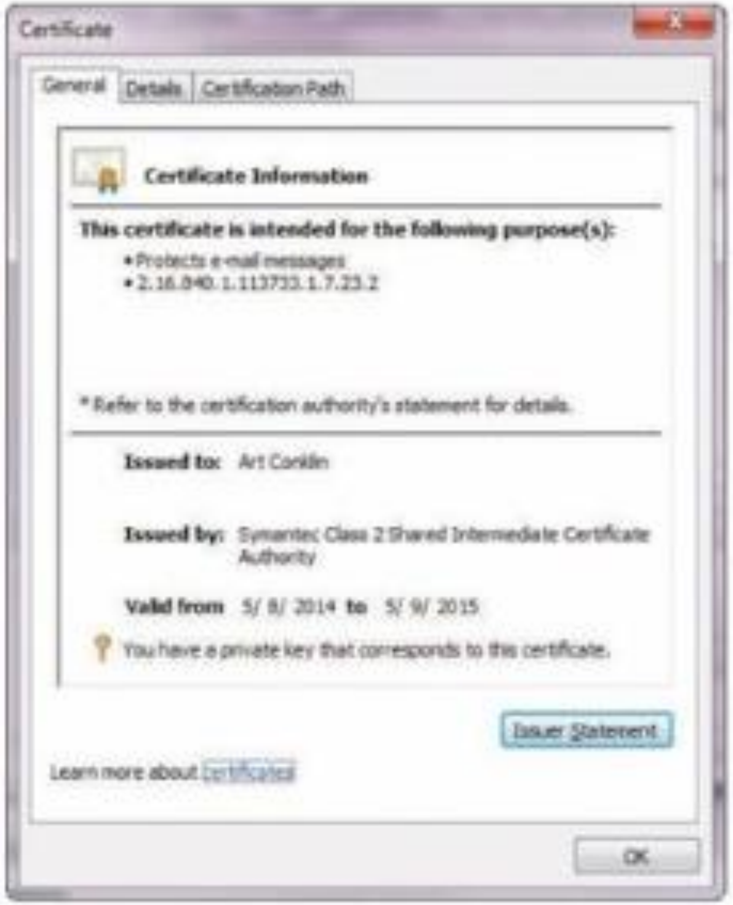


Table 7.1 X.509 Certificate Fields

Field Name	Field Description
Certificate Version	X.509 version used for this certificate: Version 1 = 0 Version 2 = 1 Version 3 = 2
Serial Number	A nonnegative integer assigned by the certificate issuer that must be unique to the certificate.
Signature Algorithm Parameters (optional)	The algorithm identifier for the algorithm used by the CA to sign the certificate. The optional Parameters field is used to provide the cryptographic algorithm parameters used in generating the signature.
Issuer	Identification for the entity that signed and issued the certificate. This must be a distinguished name within the hierarchy of CAs.
Validity Not valid before time Not valid after time	Specifies a period of time during which the certificate is valid, using a "not valid before" time and a "not valid after" time (expressed in UTC or in a generalized time).
Subject	The name for the certificate owner.
Subject Public Key Info	An encryption algorithm identifier followed by a bit string for the public key.
Issuer Unique ID	Optional for versions 2 and 3. This is a unique bit-string identifier for the CA that issued the certificate.
Subject Unique ID	Optional for versions 2 and 3. This is a unique bit-string identifier for the subject of the certificate.
Extensions Extension ID Critical Extension Value	Optional for version 3. The extensions area consists of a sequence of extension fields containing an extension identifier, a Boolean field indicating whether the extension is critical, and an octet string representing the value of the extension. Extensions can be defined in standards or defined and registered by organizations or communities.
Thumbprint Algorithm Algorithm Parameters (optional)	Identifies the algorithm used by the CA to sign this certificate. This field must match the algorithm identified in the Signature Algorithm field.
Thumbprint	The signature is the bit-string hash value obtained when the CA signed the certificate. The signature certifies the contents of the certificate, binding the public key to the subject.

Digital Certificates



Certificate Information

This certificate is intended for the following purpose(s):

- Protects e-mail messages
- 2.16.840.1.113733.1.7.23.2

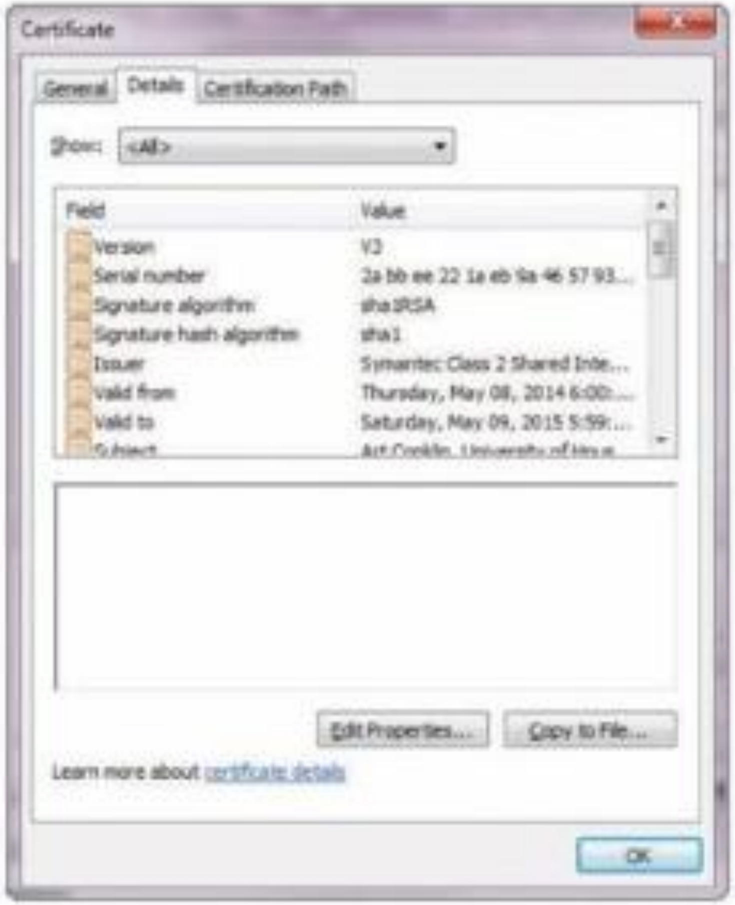
* Refer to the certification authority's statement for details.

Issued to: Art Conklin

Issued by: Symantec Class 2 Shared Intermediate Certificate Authority

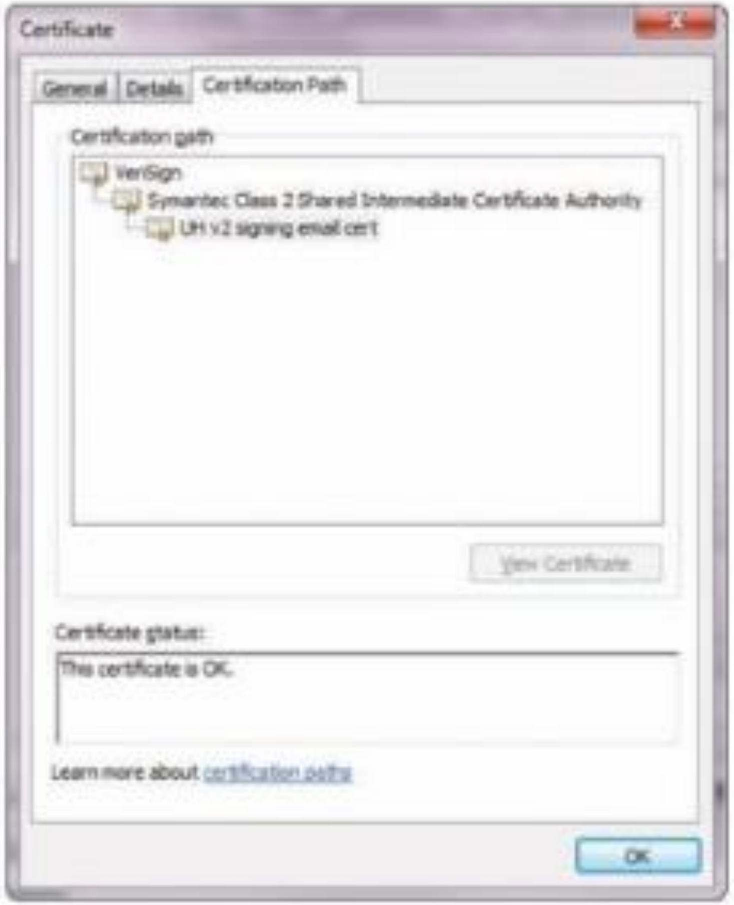
Valid from 5/ 8/ 2014 **to** 5/ 9/ 2015

You have a private key that corresponds to this certificate.



Show: <All>

Field	Value
Version	V3
Serial number	2a bb ee 22 1a eb 9a 46 57 93...
Signature algorithm	sha1RSA
Signature hash algorithm	sha1
Issuer	Symantec Class 2 Shared Inte...
Valid from	Thursday, May 08, 2014 6:00:...
Valid to	Saturday, May 09, 2015 5:59:...
Subject	Art Conklin, University of Hous...



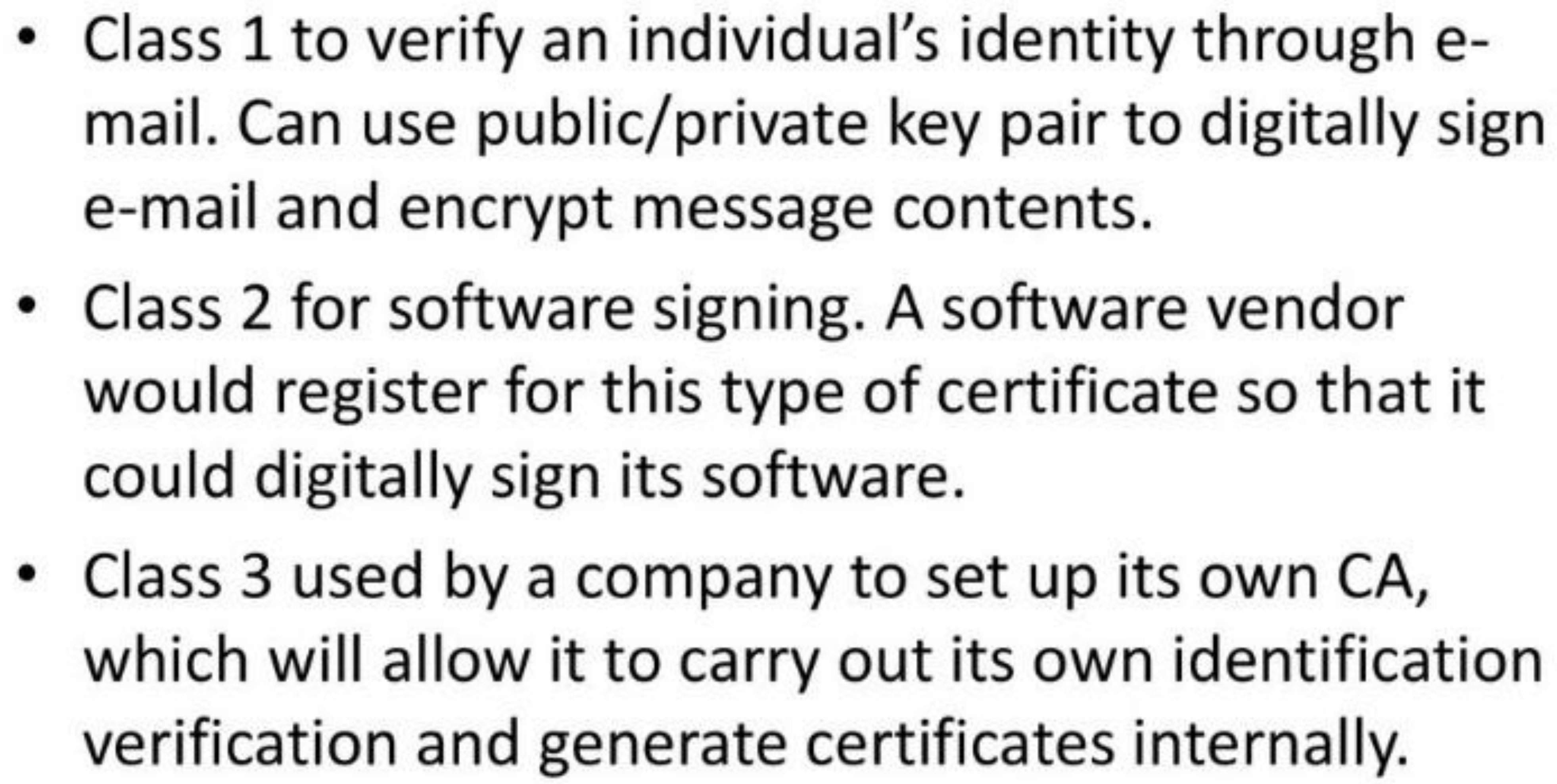
Certification path

- VeriSign
  - Symantec Class 2 Shared Intermediate Certificate Authority
    - UH v2 signing email cert

Certificate status:

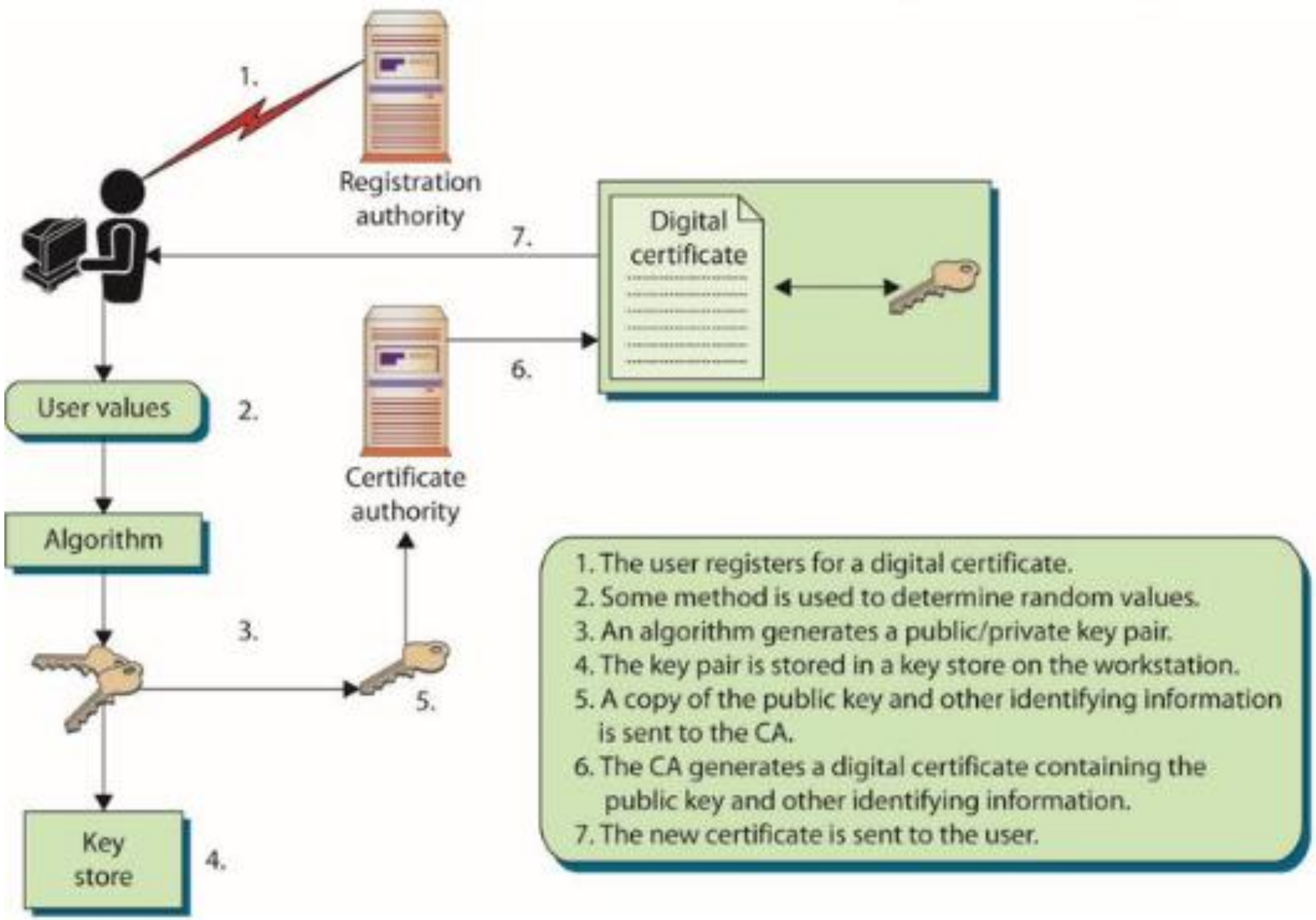
This certificate is OK.

Certificates Classes



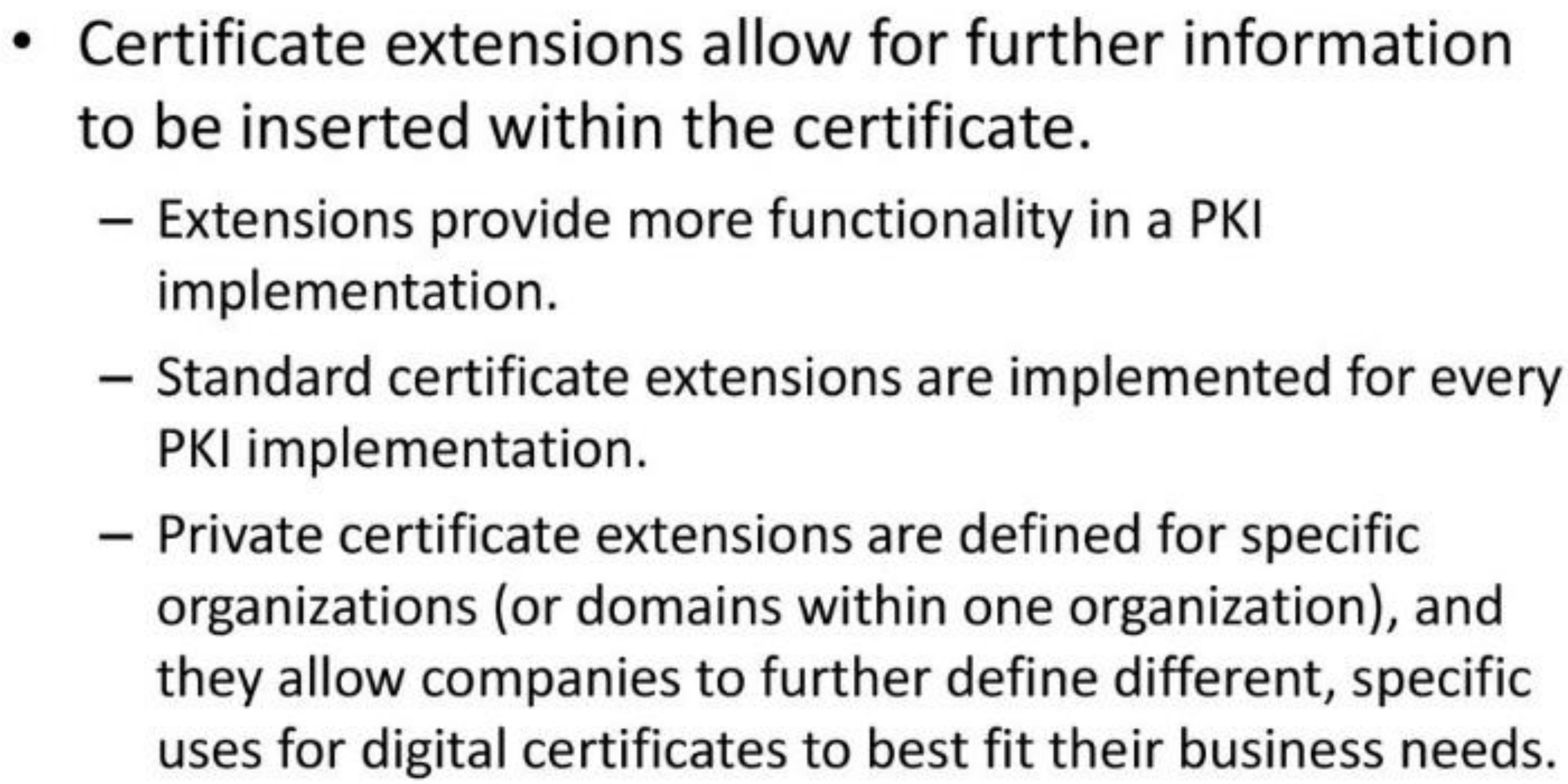
- Class 1 to verify an individual's identity through e-mail. Can use public/private key pair to digitally sign e-mail and encrypt message contents.
- Class 2 for software signing. A software vendor would register for this type of certificate so that it could digitally sign its software.
- Class 3 used by a company to set up its own CA, which will allow it to carry out its own identification verification and generate certificates internally.

Certificates Classes



1. The user registers for a digital certificate.
2. Some method is used to determine random values.
3. An algorithm generates a public/private key pair.
4. The key pair is stored in a key store on the workstation.
5. A copy of the public key and other identifying information is sent to the CA.
6. The CA generates a digital certificate containing the public key and other identifying information.
7. The new certificate is sent to the user.

Certificates Extensions



- Certificate extensions allow for further information to be inserted within the certificate.
  - Extensions provide more functionality in a PKI implementation.
  - Standard certificate extensions are implemented for every PKI implementation.
  - Private certificate extensions are defined for specific organizations (or domains within one organization), and they allow companies to further define different, specific uses for digital certificates to best fit their business needs.