AllSeen Summit 2015:

IoT: Taking PKI Where No PKI Has Gone Before

Presented by: *Scott Rea – DigiCert Sr. PKI Architect*

digicert® | YOUR SUCCESS IS BUILT ON TRUST®

ALLSEEN ALLIANCE

Agenda

Establishing Trust

What is Trust?

- Confidence or assurance that a person, system, thing will behave exactly as you expect, or alternatively, in your best interests

Trust cannot be established by technology alone

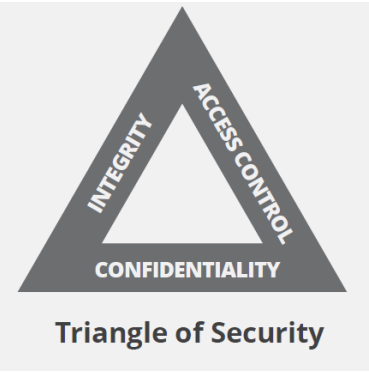
- A framework for trust requires the following attributes:
  - Technology & Tokens (secure, interoperable, audited)
  - Policy & Procedures (published and proven)
  - Relationships & Responsibilities (legal agreements or trust framework)

What is PKI?

- Public Key Infrastructure (**PKI**) underpins the security and trust infrastructure of the Internet, and while the implementation of the protocol has manifest chinks in its armor from time to time, the protocol itself has stood the test of time as an effective mechanism for establishing trust and ensuring confidentiality, integrity and authenticating previously unknown participants to secure Internet-based transactions.

What is PKI?

- Comprehensive security technology, policies, and defined relationships that uses cryptography and standards to enable users to:
  - Identify (authenticate) themselves to network services, access policies, and each other to prove source of origin and destination i.e. to ensure resources are made available only to those authorized to access them;
  - Digitally sign electronic documents, email and other data to provide authorization and prove integrity;
  - Encrypt email, data, and other documents to prevent unauthorized access and ensure confidentiality.
- Security practitioners often describe the above services CIA i.e. Confidentiality, Integrity, Availability



Triangle of Security

PKI Services: Confidentiality

- PKI encryption prevents need for shared secrets.
- Anyone encrypts with public key of recipient.
- Requires some mechanism for discovering intended recipient's public key i.e. distribution of their Certificate
- Only the recipient can decrypt with their private key.
- Private key is secret, so no one else can read encrypted data.

Triangle of Security

PKI Services: Integrity

- Compute message digest, encrypt with your private key.
- User Agent decrypts with your public key from your certificate.
- Re-compute the digest and verify match with original – guarantees no one has modified signed data.
- Only signer has private key, so no one else can spoof their digital signature.

Triangle of Security

PKI Services: Availability

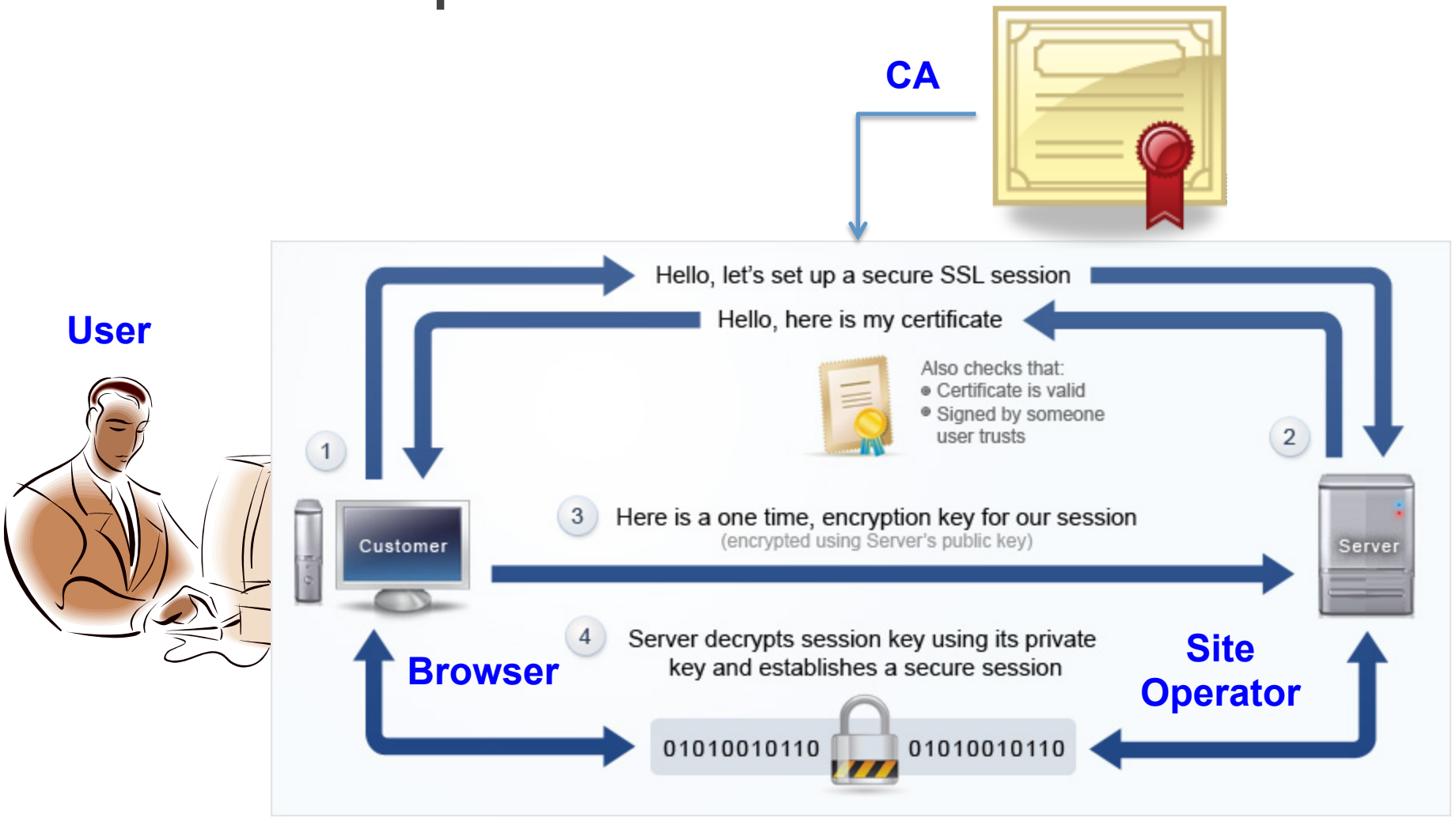
- The signature of the CA on the issued certificate containing the public key, acts a seal of authenticity of the identity of the holder of the private key
- Identity attributes are included in the certificate
- Availability decisions can be facilitated by relying upon the authentication of identity and identity binding to the asymmetric key pair that took place during the issuance process

Triangle of Security

Web Security: Example of a PKI Implementation

- We can use the well known SSL/TLS protocol as implemented to establish security between Internet users and site hosts as a prime example of a practical PKI implementation…

SSL/TLS: The process



Web Security: a Multi-party Responsibility

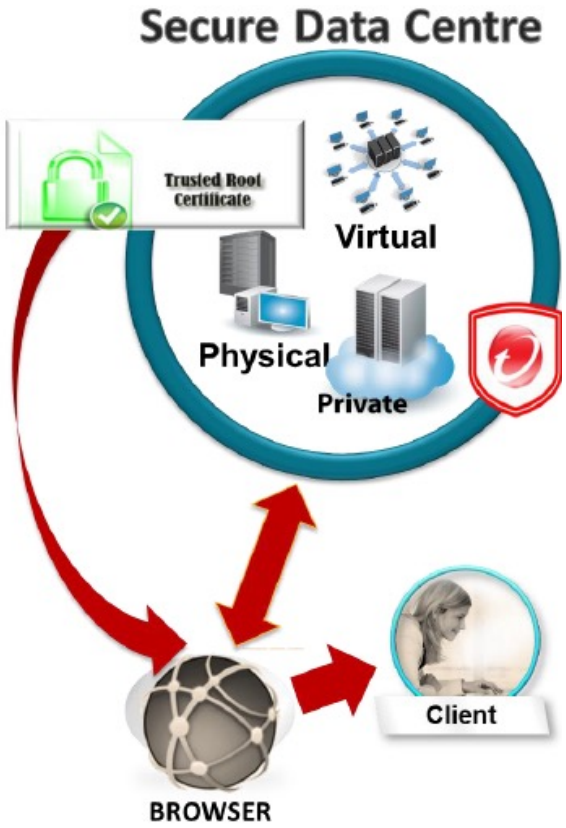
- Effective PKI requires appropriate implementation, and **ANY** of the parties involved can contribute to whether effective trust is achieved or not :
  - the policies and practices of the **Certification Authorities** (CA) issuing certificates,
  - the practices and capabilities of the **Browser** managing certificates,
  - the practices of the site administrator or **Site Operator** configuring and utilizing certificates,
  - and the practices of the web **User** relying upon certificates.

What is a Certificate Authority (CA)

- An organization that creates, publishes, and revokes certificates.
- Verifies the information in the certificate.
- Protects general security and policies of the system and its records.
- Allows you to check certificates so you can decide whether to use them in business/security transactions.
- Has one or more trusted Roots, called a trust anchor embedded in applications
- Agree to abide by a known Certificate Policy (CP) and publish a corresponding Certificate Practices Statement (CPS) for audit and independent verification of practices
- E.g. DigiCert Web PKI CPS can be found here http://www.digicert.com/ssl-cps-repository.htm

Certification Authority Responsibilities

- CA generates “roots” in secure environment – ceremony, video recorded, audited, keys on HSMs
- CA undergoes rigorous third party audit of operations and policy
- CA private keys are held under extreme protections and used to sign web site certificates and status information
- CA applies for corresponding root certificates to be included into trusted root stores
- CA policy and operations must be in compliance with Browser root store rules in order to be trusted by default, and may be distributed by software updates



Certification Authority Responsibilities

- When issuing a SSL/TLS cert to a web site, the CA verifies certain information relating to ownership of the site with the respective domain and verifies control of keys being used.

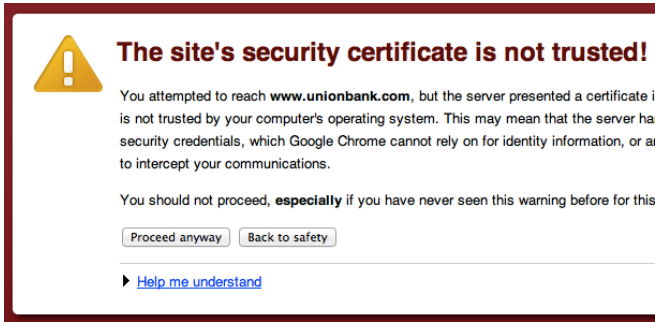
- The strongest verification of site and domain ownership with multiple verification of direct contacts etc., allows issuance of the highest standard of assurance for SSL certificates
  - This highest tier of verification is called Extended Validation or EV
  - EV issued certs are recognized in browser GUI e.g. green bar

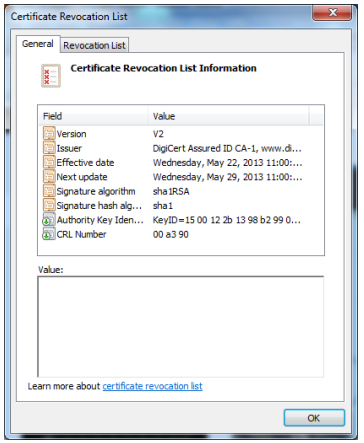
SSL/TLS: Trust Basis – CA Role

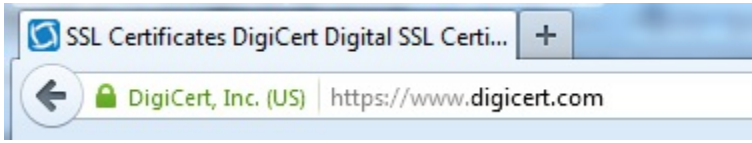
- CA provides certificates to customers chaining to trusted roots embedded in Browsers (who may in turn rely upon roots embedded in Operating Systems)
- CA verifies the certificate subject according to published policies included in the CP, and binds subject to Public Key embedded in the certificate
- Site Operators generate PKI key pairs, provide Public Key to CA for cert creation, install certs on their servers for secure web pages, and protect corresponding Private Key from unauthorized access
- Site Operator may require certificate based authentication of Users for 2-way trust
- Users go to secure web pages HTTPS://, User Agent checks for CA's root inclusion in browser trusted root store
- If CA's root is in browser's trusted store: encrypted session, favorable padlock UI (including EV green bar)

SSL/TLS: Trust Basis – CA Role

- If CA root not in client trusted root store for browser – warning displayed
- CAs and browsers have the ability to revoke roots, sub-CAs, and certificates for any problems
- CAs publish revocation lists (CRLs) or provide updated certificate status information online (OCSP)
- If certificate revoked or expired – warning displayed
- CAs must complete annual audits and follow CA/B Forum rules to remain in browser trusted root stores
- Stronger rules and higher CA standards are set for green Extended Validations or “EV” display





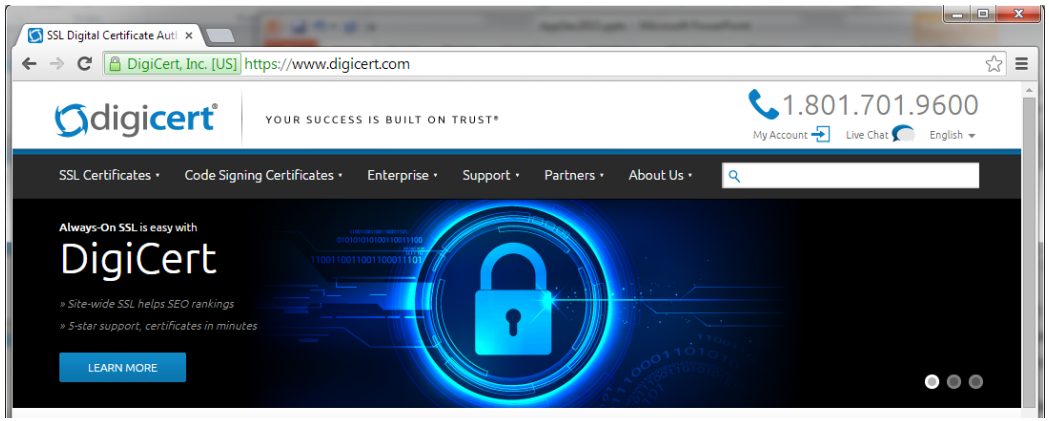


SSL/TLS: Trust Basis – Browser Role

- Browser publishes a set of requirements for inclusion of trusted CAs into its root store
- Browser evaluates CA applicants for inclusion against published criteria
- Browser update process allows for inclusion or removal of trusted roots
- Browser also allows Users to manually manage which roots they choose to trust and for which purposes
- Browser periodically reviews CA audit data to ensure that included roots are still in compliance with trust program

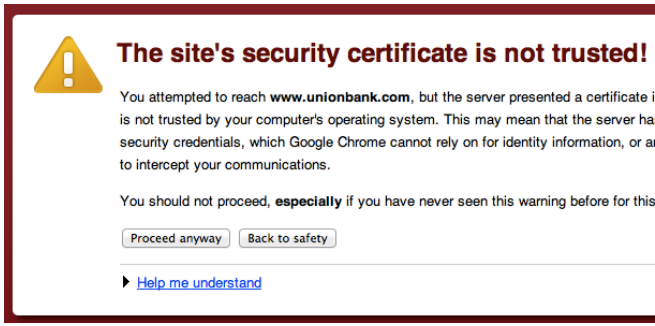
SSL/TLS: Trust Basis – Browser Role

- Browser users go to secure web pages HTTPS://,
- Browser performs the following checks on the site certificate being used:
  - matches name in location bar to those certified in the site certificate (subjectAlternativeName)
  - verifies that the site has control of the private key corresponding to the public key included in the site certificate
  - checks that the site certificate is within its published validity period, and that it is being used for appropriate purposes
  - checks for inclusion of CA's root and or intermediate sub-CAs (the chain) in the trusted root store
  - checks the current status of the site certificate (and its chain) with the CA using OCSP or CRL



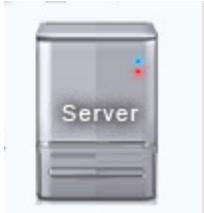
SSL/TLS: Trust Basis – Browser Role

- If CA not in browser trusted root store, or any of the checks pre-defined above fails for any cert in the chain
  - warning displayed
- If CA's root is in browser's trusted store, and all other checks are passed:
  - encrypted session established, favorable padlock
  - UI (including EV green bar)
- Browser messaging and indicators are critical



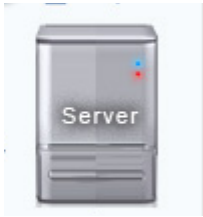
SSL/TLS: Trust Basis – Site Operator Role

- Site Operators must enable SSL/TLS protocols on the applications running on their site
  - consideration for the protocol parameters used should be based on user capabilities and security expectations
- Keys must be generated and used to provide a CSR to the chosen CA
  - When creating a request, you generate a key pair (a private key and a public key)
  - Public keys are included in the CSR and embedded into the certificate by the CA,
  - Private keys MUST be secured on the site because that is how you prove you are the one authorized behind the public certificate that represents you
  - Use strong algorithms and secure processes to generate appropriate keys
- Choose a CA that is trusted in the browsers used by the applications expected users
  - Not all CAs are the same: Review certifications and practices
  - Understand performance capabilities and services
- Choose the right certificate type for the information or data that is being protected
- Install certificate from CA in application and configure trust chains



SSL/TLS: Trust Basis – Site Operator Role

- Ensure data being transferred between site and browsers is appropriately protected
  - Always On SSL should be configured so that not just initial authentication of Users is enabled, but protection of transactions from User login through User logout is in place
  - Configure back-end database or third party communications to also be secured so that would-be attackers cannot simply go around the TLS secured channels
- Maintain DNS records with appropriate and up-to-date information
- Monitor site
  - Watch for inappropriate traffic
  - Revoke certificate if private key becomes compromised
  - Update certificate BEFORE it expires
  - Keep application software patched and up-to-date
- Only use private key and certificate in accordance with intended usages
  - Avoid over-exposure of key material
- Employ network and physical protections on private keys
  - Use firewalls and intrusion and/or extrusion detection services to ensure private data and cryptographic keys are being protected
  - Use HSM to store private keys when physical protections are not sufficient



SSL/TLS: Trust Basis – User Role

- How can a web User be sure they are or remain protected in the SSL/TLS process?
  - Keep browser up-to-date
  - Watch for appropriate queues in the browser
    - Name in location bar is where you expected to go/be
    - Protocol being used is HTTPS
    - Lock to indicate a secure connection is established
    - DO NOT ignore pop-up warnings in the browser chrome, review carefully and respond appropriately
    - Validate that content is appropriate and expected (Logos – site or association, site seals, expected link contents and values)
    - Review information in security certificate upon initial visit and periodically thereafter
  - Utilize browser tools or plugins to restrict active content for new or relatively new sites visited
  - Only enter appropriate sensitive data once you are confident you have the right site

SSL/TLS: Trust Basis – User Role

- How can a web User be sure they are or remain protected in the SSL/TLS process?
  - Don't enable global security exceptions inappropriately
    - If choosing to trust a previously unknown or untrusted CA or server certificate, consider doing so just for that session
    - Consider why the site is not using an already trusted TLS credential
  - Don't manually install CA roots or intermediate CAs from untrusted sites, or if they were not obtained via a secure and reliable mechanism
  - Don't install plugins or add-ons from untrusted sites
  - Consider using different browser instances for casual browsing vs secure transactions
  - Run AV and other scanning services to interrogate active content before launching it on your system

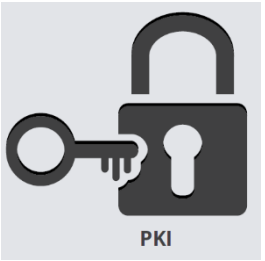
Web Security: a Multi-party Responsibility

- Effective PKI requires appropriate implementation, and **ANY** of the parties involved can contribute to whether effective trust is achieved or not :
  - the policies and practices of the **Certification Authorities** (CA) issuing certificates,
  - the practices and capabilities of the **Browser** managing certificates,
  - the practices of the site administrator or **Site Operator** configuring and utilizing certificates,
  - and the practices of the web **User** relying upon certificates.

Web Security: a Multi-party Responsibility

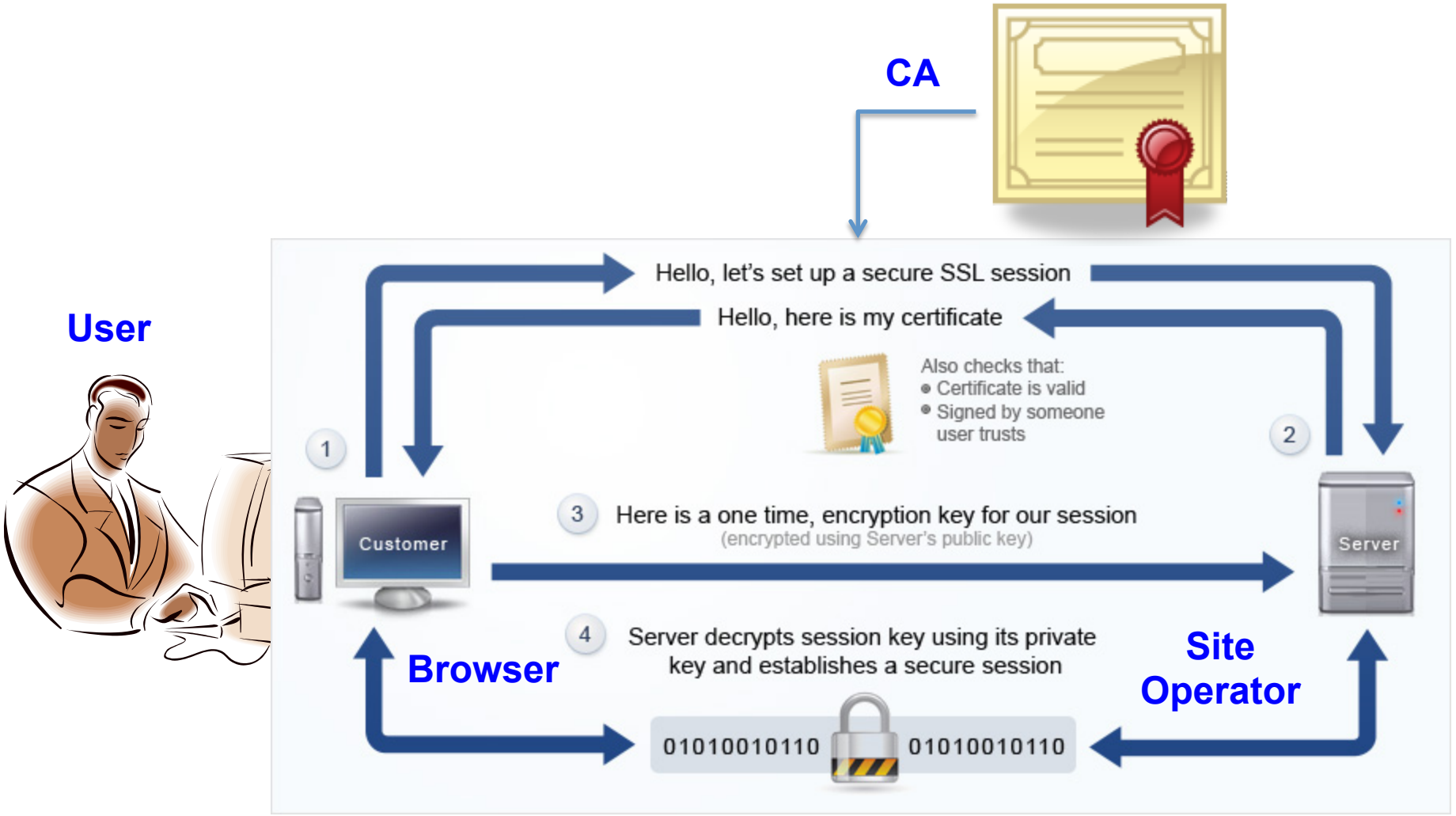
- You can't just implement a technology and expect trust to automatically be established
- Choices made about implementation options will impact security and privacy
- What system remains in constant state?
  - What is the impact to trust based on system configuration changes or system behavior over time?
  - Who is verifying that?
- The actions of Users over time will impact the assurance of trust
  - Do they clearly understand their responsibilities within the system?
- When something does go wrong – who is liable/responsible?

Roles in a Generalized PKI Model

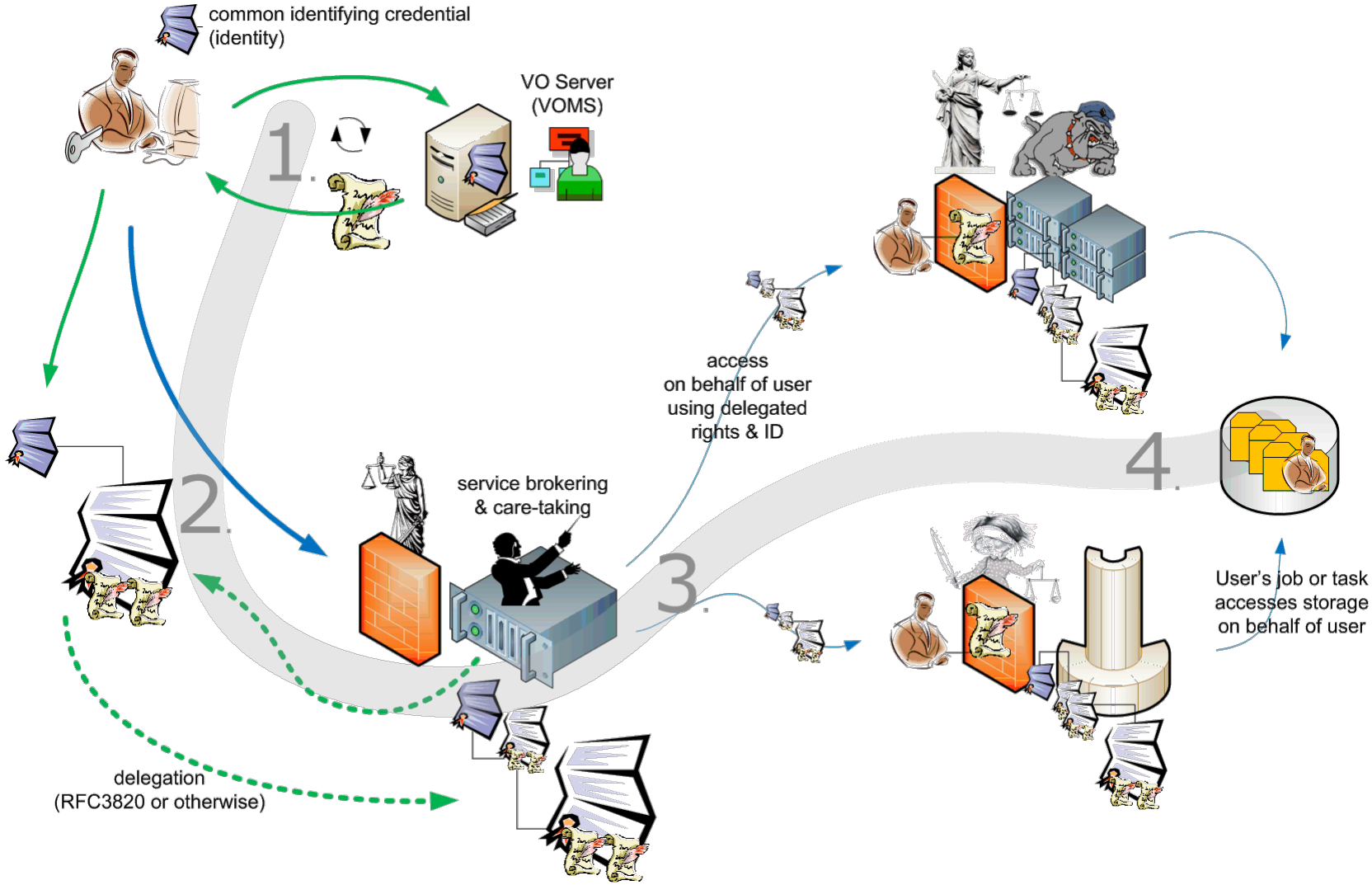


- Certification Authority:
  - Trusted Introducer i.e. between previously unknown parties, or when two parties who are remote to each other, need assurance that the network identity corresponds to the expected physical entity or trusted process
  - binds known identities to keys used to establish network identities
  - establishes trust relationship with User Agents by embedding Roots
- User
  - Party that wishes to remotely exchange data in a trusted manner over an open network
  - May be an individual, a process, or a device
  - Identity and attributes are verified by the CA
- User Agent:
  - Application e.g. Browser, that allows remote Users to establish a secure channel over an open network
  - Verifies Users and attributes using local trust policies and embedded Roots
  - Facilitates the encoding and decoding of data over the secure channel
- Site/Site Operator:
  - Service that has data to exchange in a trusted manner over an open network
  - Identity and attributes are verified by the CA
  - Verifies remote Users and attributes using local trust policies and embedded Roots
  - Facilitates the encoding and decoding of data over the secure channel

PKI Facilitates TLS



Supercomputing: Grid Authorization Flow Example



Graphic:David Groep, Nikef NL

PKI Challenges for IoT Device Interaction

- When the User is a device – who speaks authoritatively for that device?
  - Is it the manufacturer or the owner or the current user of the device or a designated device admin?
  - How does a CA verify authenticity of the relationship between device and responsible party?
  - What happens when the responsible party changes?
- What constitutes a device identity?
  - Is it just a unique identifier associated with the device?
  - Is that sufficient to enable Authorization to perform various transactions
    - Often the location of the device, or the designated responsible party at any given point in time, or the configuration of the device etc. are all catalysts to modifying what permissions or capabilities the device might be trusted with.
    - A single Identity based PKI does not handle the additional authorization permissions very well unless those authorizations are long lived, similar to the Identity cert validity (in which case a new Identity cert could be issued)

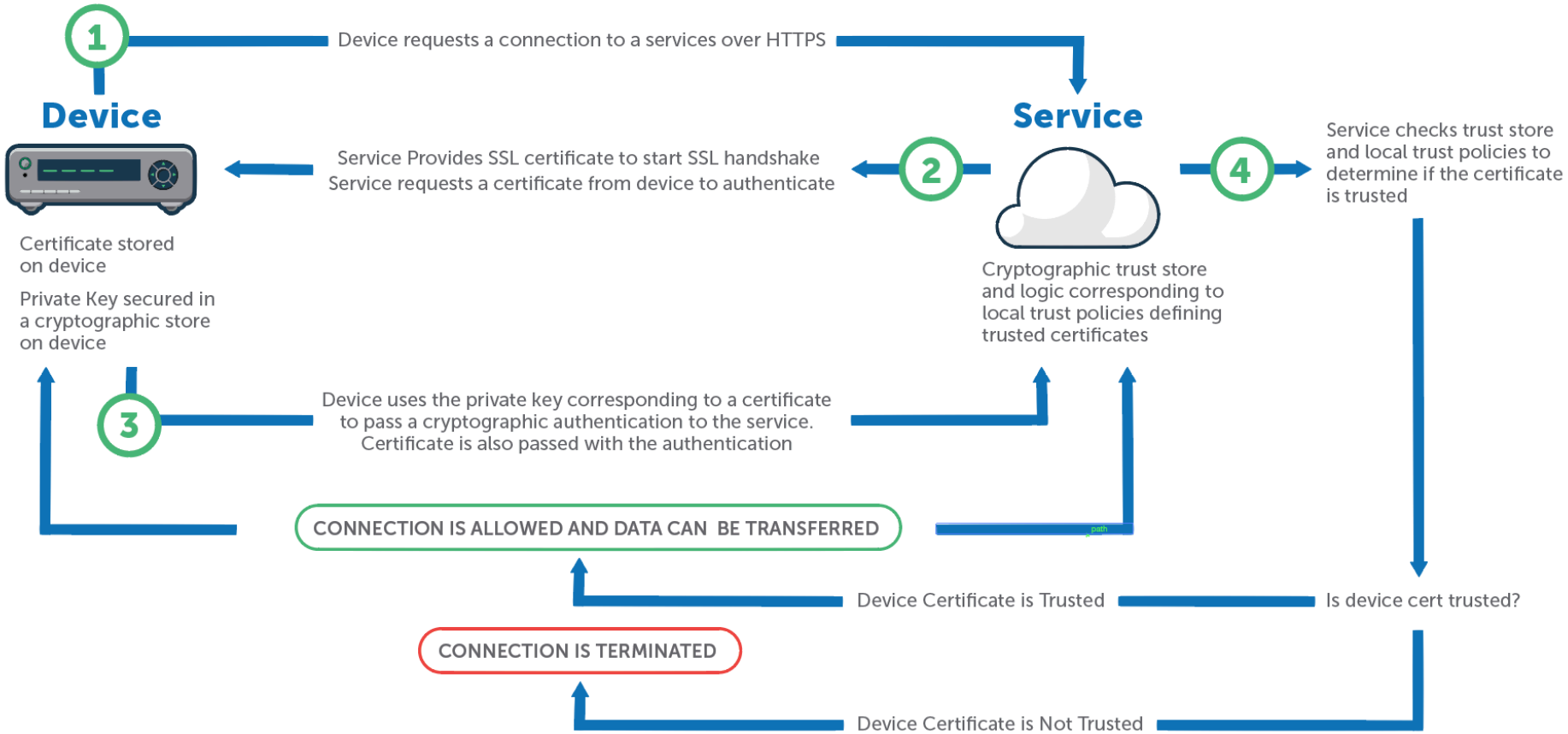
PKI Challenges for IoT Device Interaction

- Devices whose attributes change over time may have different authorizations in the same network
  - Physical Device Identity has not changed, but ownership or location or configuration changes often necessitate a change in allowable permissions
  - When the period/frequency of change does not correspond with the identity based certificate issued, then it becomes wasteful and inefficient to utilize the identity certificate for authorizations
  - In this instance it is more efficient to utilize a separate authorization mechanism that is linked to the identity certificate
  - If the authorization mechanism is another certificate: What is the format of that authorizations certificate? Who issues it? How is it verified?
  - If the authorizations mechanism is a local database: Who manages that database? How is it secured? How does that scale to IoT proportions?

PKI Challenges for IoT Device Interaction

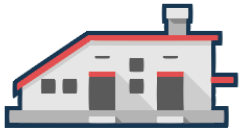
- No matter which approach is taken to the transitory authorizations that become associated with IoT devices (local DB or separate authorizations certificate), there is a change required to be implemented for traditional PKI.
- Local Databases do not scale, and are problematic to administer and secure
  - How do you determine which DB is authoritative for a given device in a particular instance?
- Attribute certificates (see RFC 3820) scale but are not well supported for all life cycle aspects in cryptographic libraries
  - When an Identity certificate is revoked, how do you find and revoke all the Attribute certificates associated with that?
- Other attribute certificate types e.g. SAML assertions signed by an authoritative source provide a better implementation capability
  - Still the question is how to identify and manage the set of authoritative sources

PKI for Securing Device Communication Channel



PKI for Device Firmware Updates

Device Manufacturer



Certificate stored on device

Private Key secured in a cryptographic store on device

1. Firmware is created for a device by the manufacturer
2. A hash of the firmware is created and encrypted with the private key corresponding to the code signing certificate
3. Encrypted hash is tied to the firmware

   Corresponding certificate is tied to the firmware to identify manufacturer is the code signer

Device

Cryptographic trust store and logic corresponding to logical trust policies defining trusted firmware

4. Firmware is presented to the devices for update
5. Device checks code signing certificate used to sign the firmware

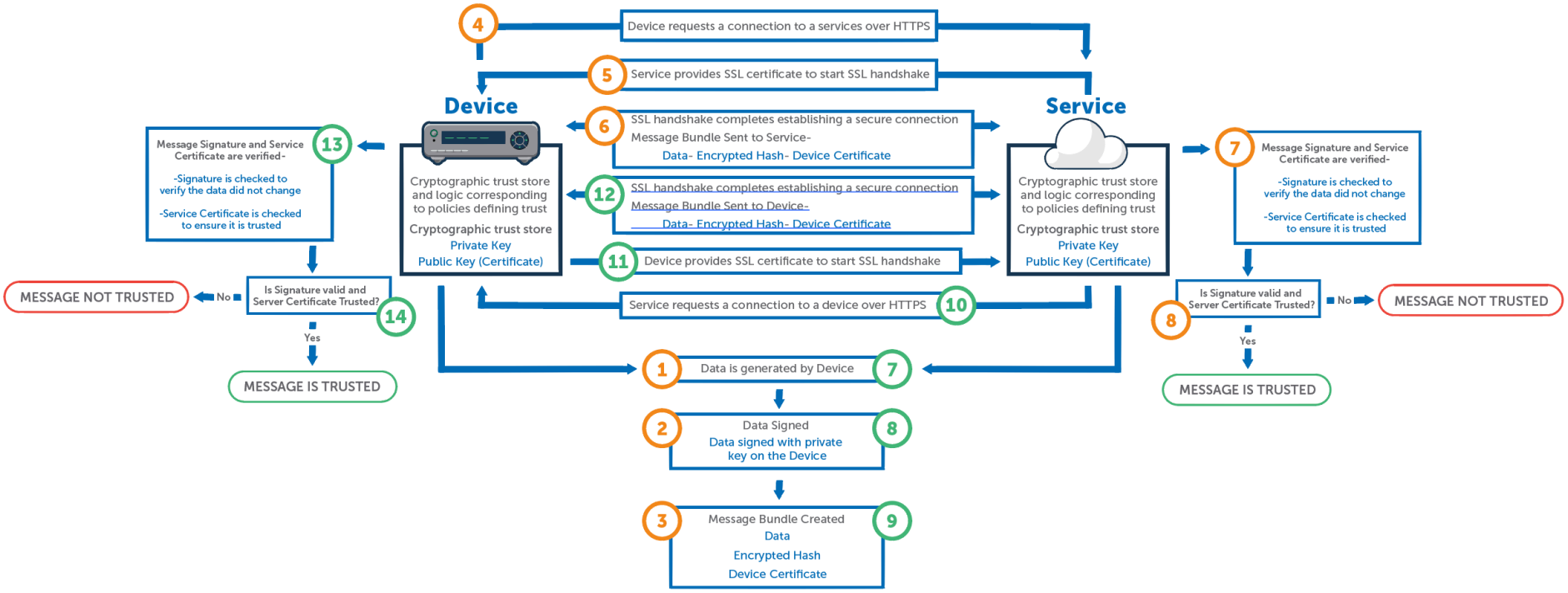
   Checks against its cryptographic trust store and local trust policies to determine if the certificate used to sign the code is trusted

6. Is certificate trusted?
   - If not trusted → FIRMWARE IS NOT INSTALLED
   - If trusted
7. A hash of the firmware is created by the device

   Hash encrypted by the manufacturer is decrypted using the code singing certificate public key

8. The hash of the firmware created by the device is compared with the hash of the firmware created when the manufacturer signed the file
9. Do the hashes match?
   - No match means → FIRMWARE IS NOT INSTALLED
   - If hashes match → FIRMWARE IS TRUSTED AND INSTALLED

PKI for Securing Device Data/Messages



PKI for Securing Device Data/Messages

Signing Data

Hash of Data Encrypted

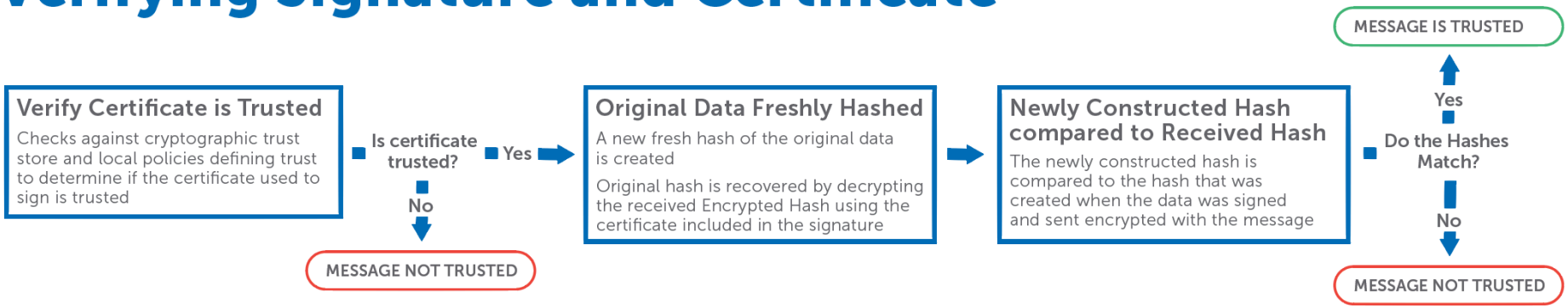
A hash of the data is created and encrypted with the private key

→

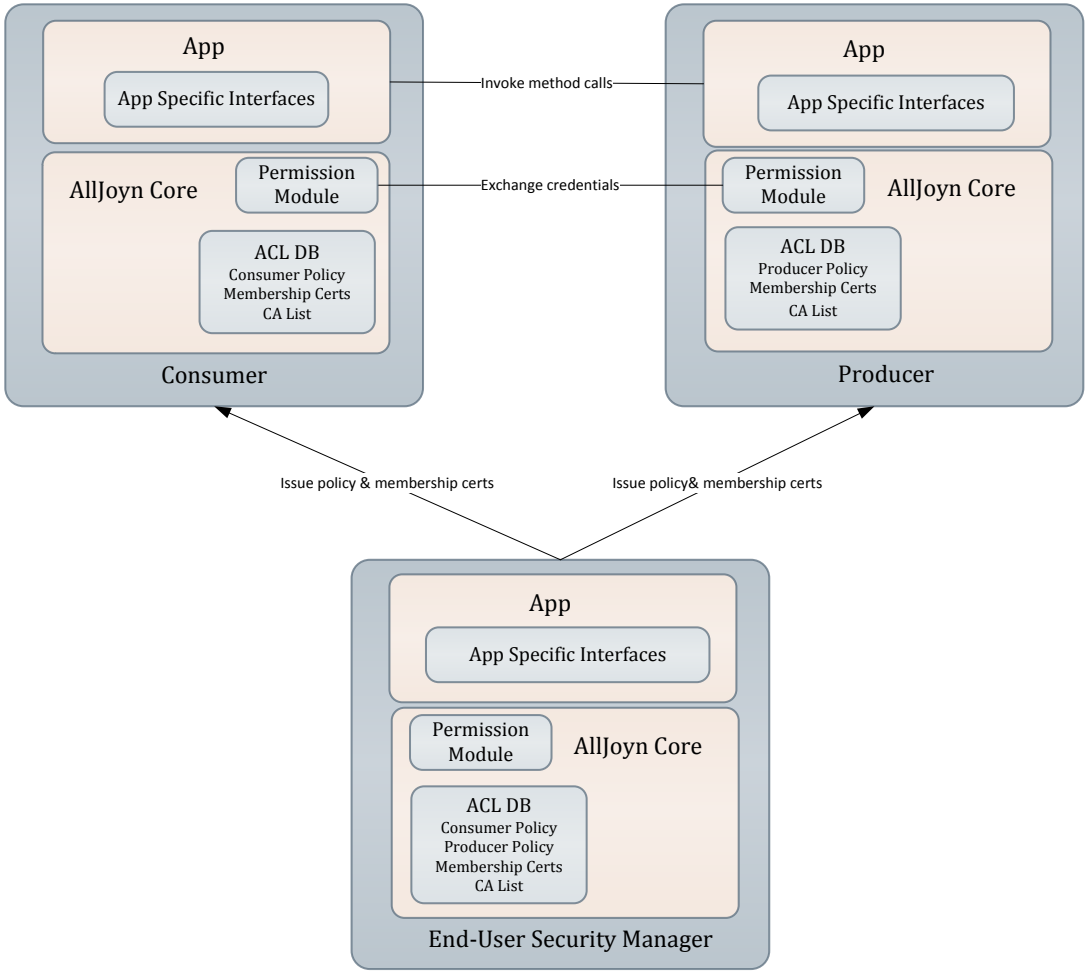
Certificate Corresponding to Private Key Attached

Corresponding certificate is tied to the signature

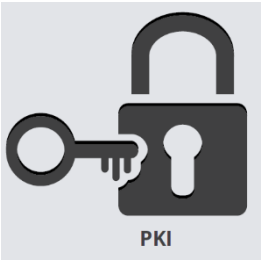
Verifying Signature and Certificate



AllJoyn Security 2.0



AllJoyn Security 2.0 – A Model of Generalized PKI



Generalized PKI

- Certificate Authority
- User/User Agent:
- Site/Site Operator:

AllJoyn Security 2.0

- Security Group Authority
  - A security group authority is the user or application that defines the security group and grants membership certificates to others. The security group authority is the certificate authority for that group.
- AllJoyn Consumer
  - An AllJoyn application consuming services on the AllJoyn network.
- AllJoyn Producer
  - An AllJoyn application providing services on the AllJoyn network

What should IoT Developers do?

- Utilize PKI – it the best technology we have for securing messages, data, and identities across open networks
- Understand that PKI is not just implementing the technology, but establishing trust also means defining the processes and procedures for each actor in the PKI, and clearly communicating the responsibilities and acknowledgement of liabilities of each actor in the system
- When IoT device attributes are important to the local trust configuration, ensure the PKI processes and procedures can cater for attribute changes
  - This may involve establishing Attribute Authority trust architectures and procedures as well as Certification Authority trust architectures and procedures
    - NOTE: The AllJoyn Security Group Authority is a good example of an AA
  - The CAs should enable trust of AA service providers

What should IoT Developers do?

- Trusting Attribute Authorities should follow similar established processes for trusting Certification Authorities:
  - A policy for AA's to operate under should be established (this includes attribute release and management policies)
  - An AA may then publish a practices statement that demonstrates how it meets the policy requirements (this includes operational and security controls)
  - The CA may then verify (or through an audit from a certification body) that the AA is operating in accordance with established policy and its published practices
  - The CA may then issue an identity certificate to the AA to be used for making attribute assertions. This assertion signer is trusted by the App community by being issued through an intermediate that chains to a community root.
  - The AA follows its published policies to make and manage assertions for devices enrolled with it
  - When device attributes change, the local AA provides updated attribute assertions regarding that device
  - If the AA misbehaves, the CA may revoke it assertion signer, thus de-authorizing it from the network

What should IoT Developers do?

- Implementing the AllJoyn Security 2.0 Model
  - Local CA vs commercial CA
    - Partner who knows how to scale
    - Is trusted and audited across broadest scale
    - Reduces your implementation effort
  - Embed Identity in device during OEM process
  - Regulate Attribute Authorities (AllJoyn Security Group Admins)
    - Trusted CA for assertion signers
    - Published policies available to ecosystem
    - Accreditation/audit of operations
- **Trust cannot be established by technology alone**
    - Technology & Tokens (secure, interoperable, audited)
    - Policy & Procedures (published and proven)
    - Relationships & Responsibilities (legal agreements or trust framework)

Questions?

Contact Info:

scott.rea@digicert.com

https://www.digicert.com

(801) 701-9636