
KubeCon
CloudNativeCon
North America 2022
BUILDING FOR THE ROAD AHEAD
DETROIT 2022
TiKV
TiKV
HELM

You Like It Or Not; You Need It!

PKI And Certificate Management

KubeCon | CloudNativeCon

North America 2022

BUILDING FOR THE ROAD AHEAD

DETROIT 2022

October 24-28, 2021

Shweta Vohra

Senior Architect, IBM

Contents

PKI and Certificate Management – Vocabulary and Refresher

Case Study - Hybrid Common yet Complex Scenario and Demonstration

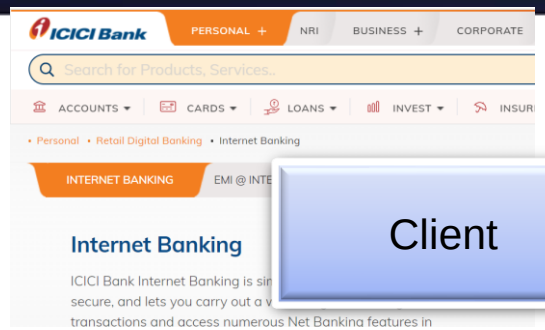
5 PKI Design Decisions – You must know!

BUILDING FOR THE ROAD AHEAD

DETROIT 2022

PKI and Certificate Management – Vocabulary & Refresher

Public Key Infrastructure (PKI)



SSL/TLS

Client

Encrypted Data

1. Trust Establishment

Server

Decrypted Data

2. Exchange of Encrypted Data over Network

Request to Issue/ Sign Certificate

Registration Authority

Verify Signature

Signature OK

Certificate DB/ Store

Request Forward/ Signing

Certificate + Private Key

Verification Authority

Certificate Authority

Certificate + Public Key

Public Key Infrastructure

What it Provides?

- Unique Cryptographic Identity to Endpoints
- Encryption of data over Network's communication channels

How it Provides?

- By governing issuance and management of digital certificates

Key Components

- Digital Certificate (X.509 certificates)
- Certificate Authority (CA)
- Registration authority (RA)
- Verification Authority (VA)
- Certificate database/ Store

PKI is a **Framework** not a specific **Technology**!

Cryptography - Public & Private Keys

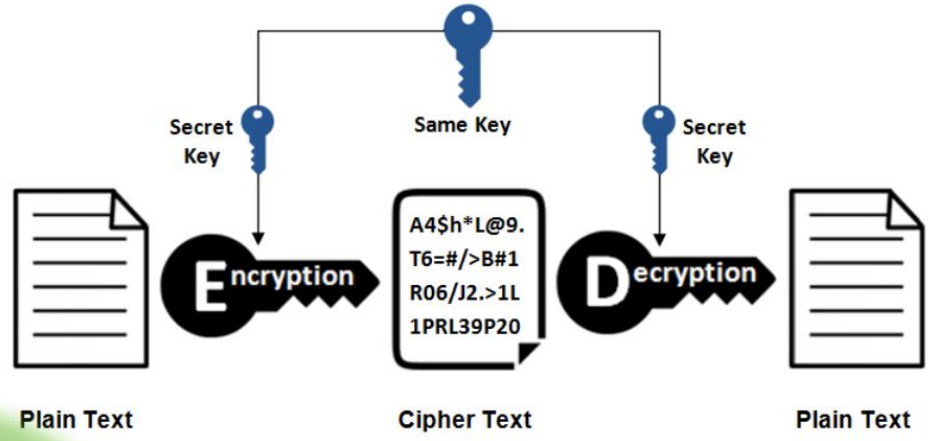
Cryptography is the science of secret writing with the intention of keeping the data secret. Cryptography is classified into symmetric cryptography, asymmetric cryptography, and hashing.

- Symmetric or Private Key Cryptography
  - Same key is used for encryption and decryption
  - Comparatively faster but less secure
  - RC4, AES, DES, 3DES, and QUAD

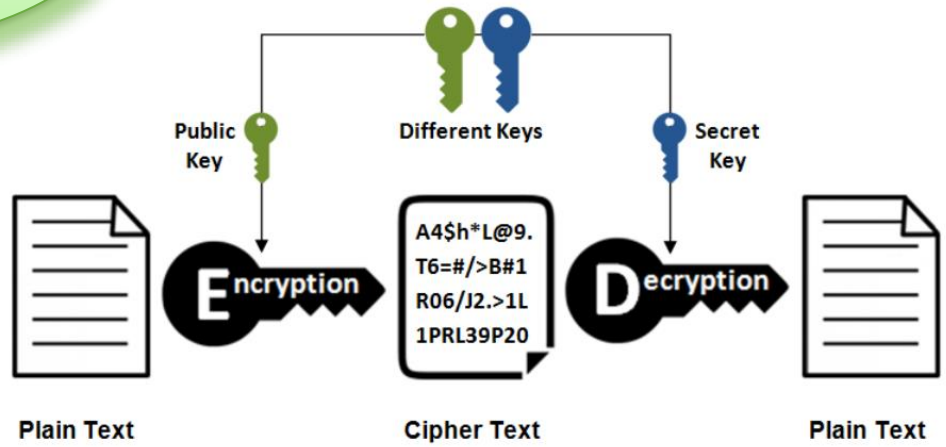
PKI uses Public Key Cryptography Framework

- Asymmetric or Public Key Cryptography
  - Different Set of keys used for encryption and decryption
  - Comparatively slower but tough to break
  - RSA, Diffie-Hellman, ECC algorithms

Private key (Symmetric encryption)



Public key (Asymmetric encryption)



Digital Certificates

What it is?

- A digital certificate is a digital document that binds the identity to a public key infrastructure (PKI)
- Used in many Internet protocols, including TLS/SSL, which is the basis for HTTPS - the secure protocol for browsing the web

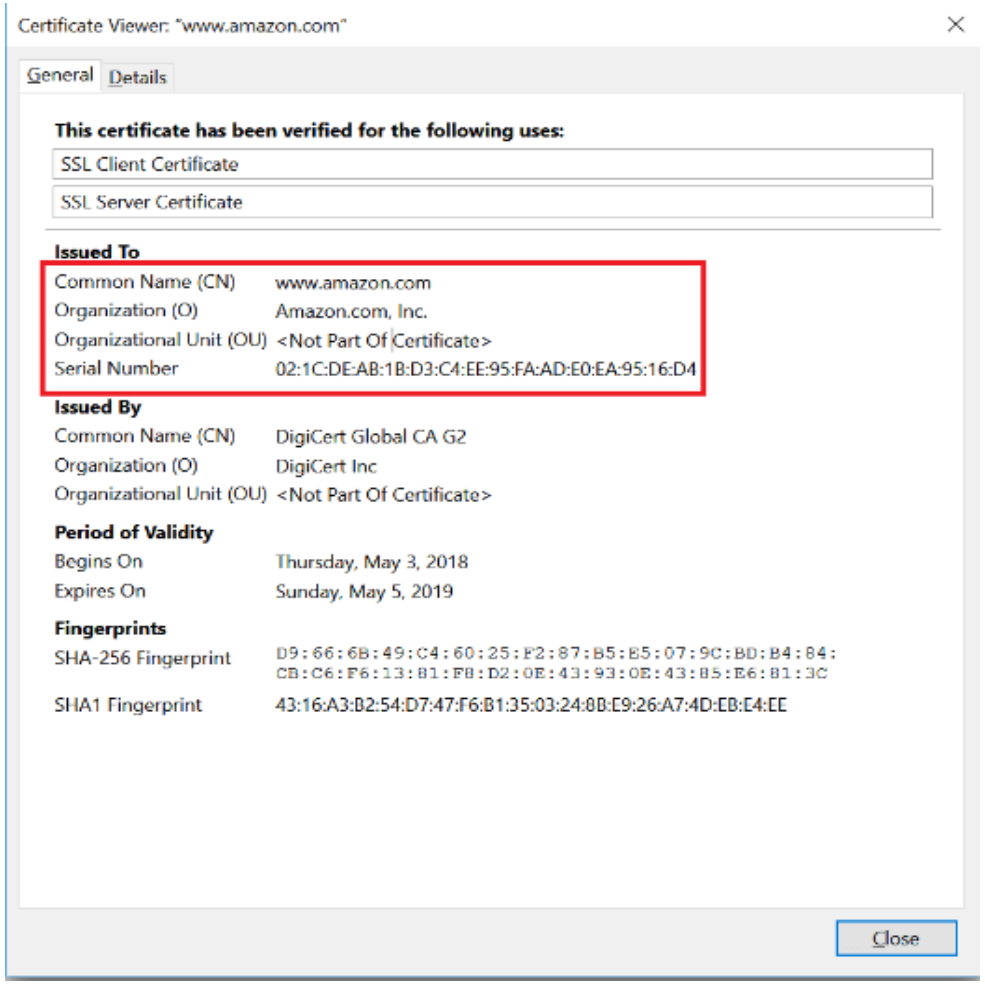
How to Generate?

- Many options that provide widely accepted international standard X.509 standard for example:
- OpenSSL tool
- LibreSSL tool
- 3rd Party RA/ VA e.g. Let's Encrypt, DigiCert etc.
- Cloud Services e.g. AWS CM, Google CM,

Key Activities

- Digital Certificates lifecycle management requires:
  - Certificate Generation and Issuance
  - Certificate Distribution/ Deployment
  - Certificate Revocation
  - Certificate Renewals/Rotation
  - Certificate Scanning and Monitoring

Sample Certificate



Windows Certificate View

Use certmgr.msc to open on your windows box

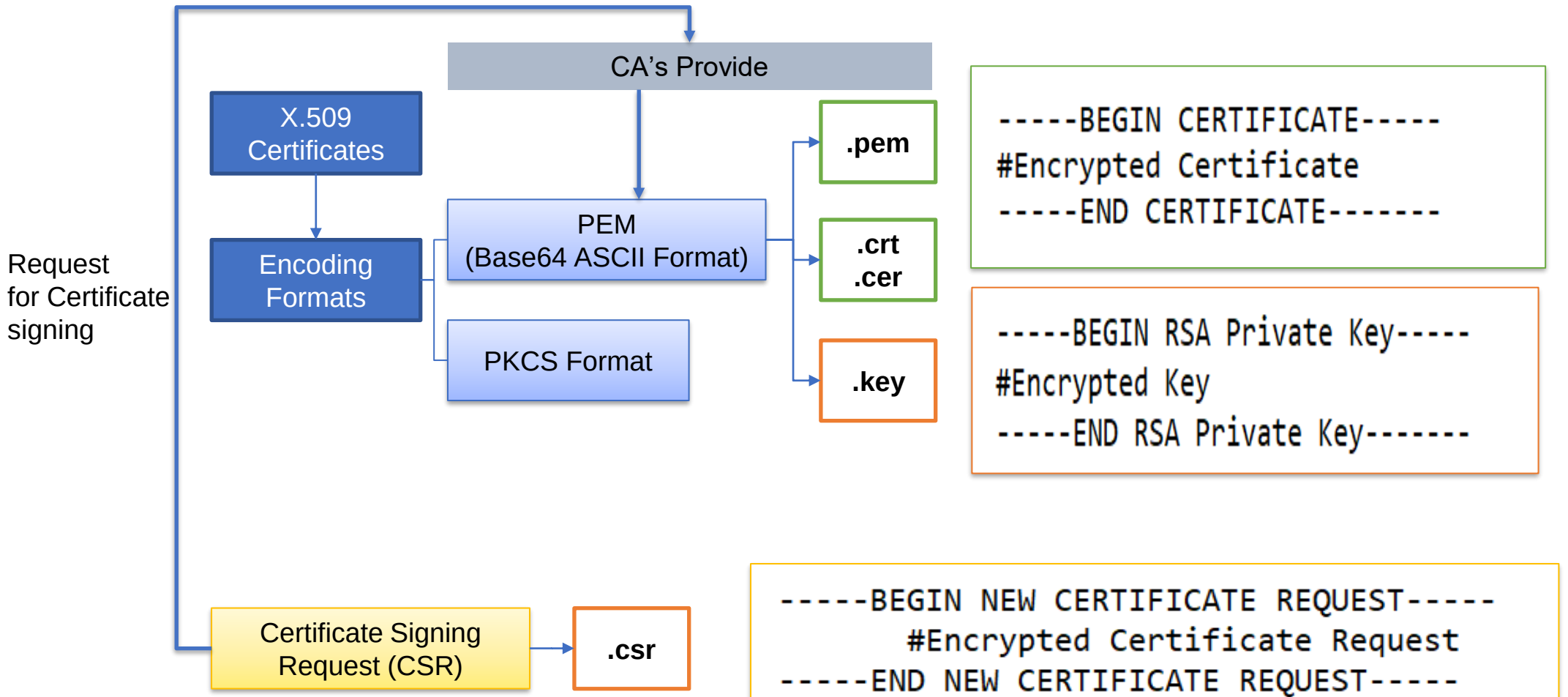
```
$ openssl x509 -text -noout -in certificate.pem
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            75:16:c7:f8:49:48:f2:22:e3:9d:1f:5e:27:00:a9:21:4f:1e:f3:16
        Signature Algorithm: ecdsa-with-SHA256
        Issuer: CN = client.example.com
        Validity
            Not Before: Jun 27 11:38:03 2020 GMT
            Not After : Jul 27 11:38:03 2020 GMT
        Subject: CN = client.example.com
        Subject Public Key Info:
            Public Key Algorithm: id-ecPublicKey
                Public-Key: (256 bit)
                pub:
                    04:b8:41:c9:0a:c5:2b:82:f3:d6:5f:43:24:d6:3f:
                    e3:34:d5:05:1c:25:14:52:1c:6f:e8:62:11:44:53:
                    91:4e:a1:22:46:83:16:60:25:e5:05:52:09:44:dc:
                    78:93:fc:d0:ac:76:3f:02:39:60:c2:4e:a3:fd:23:
                    3d:a8:10:39:f3
                ASN1 OID: prime256v1
                NIST CURVE: P-256
        X509v3 extensions:
            X509v3 Subject Key Identifier:
                96:8C:28:0D:B6:78:A8:8C:5C:6B:D2:A2:37:A8:2C:60:A1:70:00:5C
            X509v3 Authority Key Identifier:
                keyid:96:8C:28:0D:B6:78:A8:8C:5C:6B:D2:A2:37:A8:2C:60:A1:70:00:5C

            X509v3 Basic Constraints: critical
                CA:TRUE
    Signature Algorithm: ecdsa-with-SHA256
         30:45:02:20:37:e6:ba:45:bb:ce:cf:ed:f6:a8:e3:a0:2a:76:
         d0:07:cb:12:55:e0:f4:82:f4:68:44:ad:77:66:e7:6e:71:7e:
         02:21:00:e4:83:37:35:04:7a:10:27:a3:db:cb:76:8e:6f:20:
```

Public Key Info

Linux Certificate View

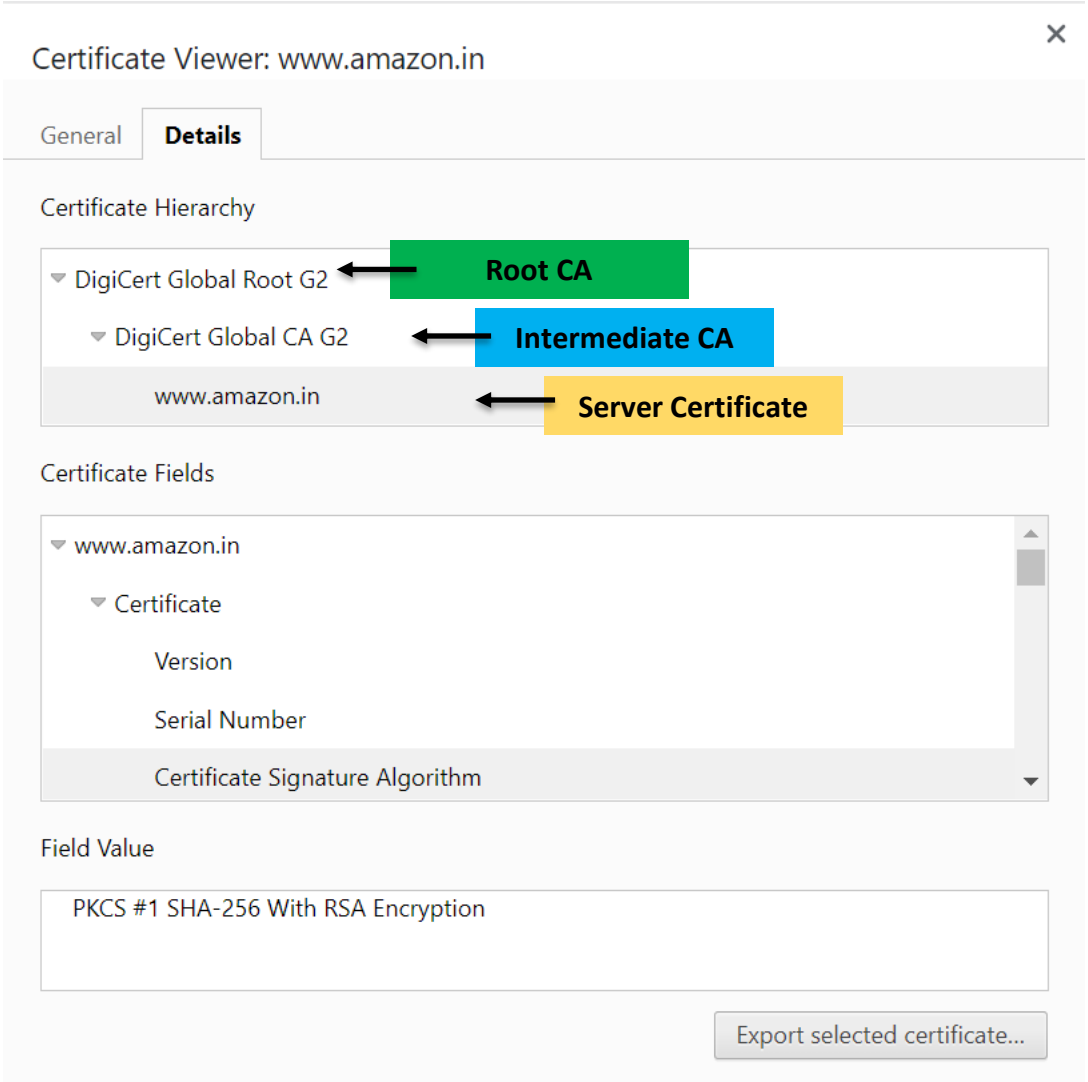
Use OpenSSL or other compatible utilities

Certificate Encoding and Files

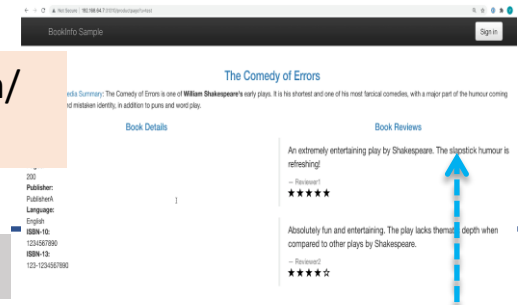


Certificate Hierarchy (Chain)

- Hierarchy of certificates from Root CA to leaf certificate is called “Certificate hierarchy” or “Chain of trust”
- It can be single-layer or multi-layered certificate authorities for protecting Root CA (from middle man attacks etc.) and separating concerns



Case Study – Hybrid Common yet Complex Production Scenario



Other Application/ Interfaces

External Users

Internal User

Cloud Provider

Network Proxy Layer

API Gateway

ELB

Ingress Gateway

Corporate Data Centre

Network Proxy Layer

Compute Layer

VMs/ Serverless Compute

Container Orchestrator

pod

pod

pod

kubernetes

Service Mesh

Ingress/ Services

Control Plane

istiod

Certificate authority

Authentication policies

Network configuration

Authorization policies

API server configuration

Data Plane

Envoy

Requests

Ingress Envoy

python

Product page

Java

Reviews-v1

Reviews-v2

pod

pod

pod

pod

Bookinfo Application

You need PKI & Digital Certificates!

External & Public CA

TLS Root CA

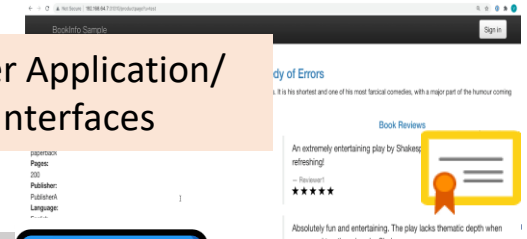
GoDaddy

Let's Encrypt

digicert

R53 & ACM

Other Application/ Interfaces



External Users

Cloud Provider

TLS - ICA

AWS Certificate Manager

Network Proxy Layer

API Gateway

ELB

Ingress Gateway

Corporate Data Centre

Private CA

TLS Root CA

Compute Layer

VMs/ Serverless Compute

Conta Orches

TLS - ICA

kubernetes

Service Mesh

Ingress/ Services

Control Plane

istiod

Certificate authority

Authentication policies

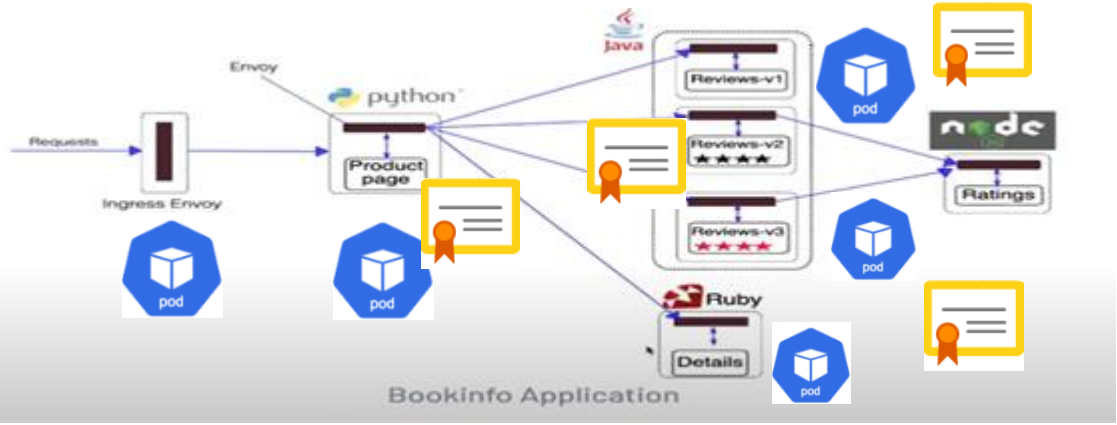
Network configuration

Authorization policies

API server configuration

TLS - ICA

Data Plane



External & Public CA

TLS Root CA

GoDaddy

Let's Encrypt

digicert

R53 & ACM

Other Application/ Interfaces

External Users

Cloud Provider

Network Proxy Layer

API Gateway

ELB

Ingress Gateway

Corporate Data Centre

Private CA

TLS Root CA

Compute Layer

VMs/ Serverless Compute

Container Orchestrator

pod

pod

pod

kubernetes

Service Mesh

Ingress/ Services

Control Plane

istiod

Certificate authority

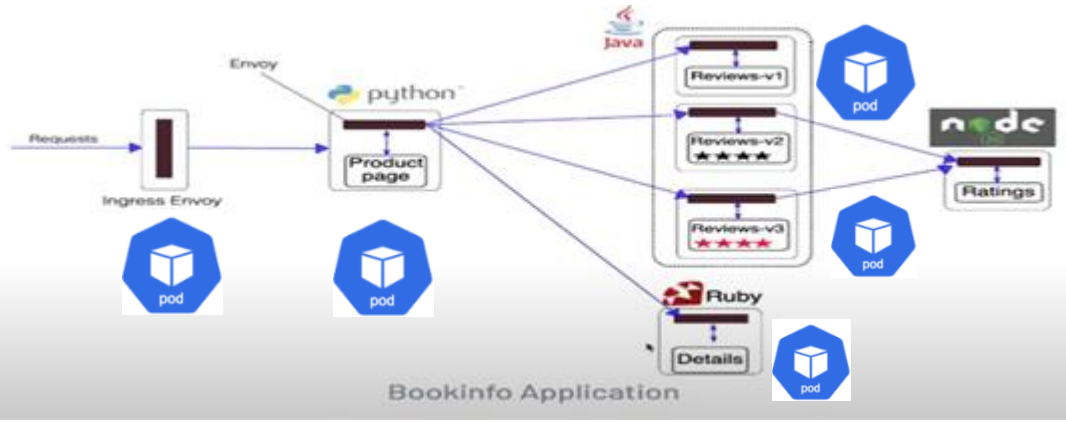
Authentication policies

Network configuration

Authorization policies

API server configuration

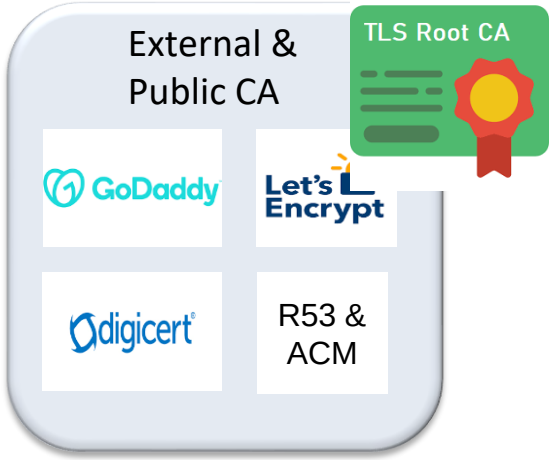
Data Plane



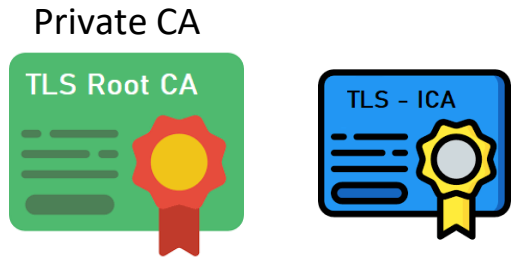
Case Study - Steps and Tools to Setup PKI

For this case study scenario:

- Utilize Public CA services - Used for external facing websites and interfaces
- DigiCert, Let's Encrypt, GoDaddy or Cloud provided services etc.



- Private and Intermediate CA – Used for internal trust establishment and communication
- OpenSSL, Cert-manager, SPIFFE/ SPIRE etc.



- Plan Certificate lifecycle management - Issuance, distribution, rotation, revocation etc.
- SPIFFE/ SPIRE, AWS or GCP Certificate Manager etc.

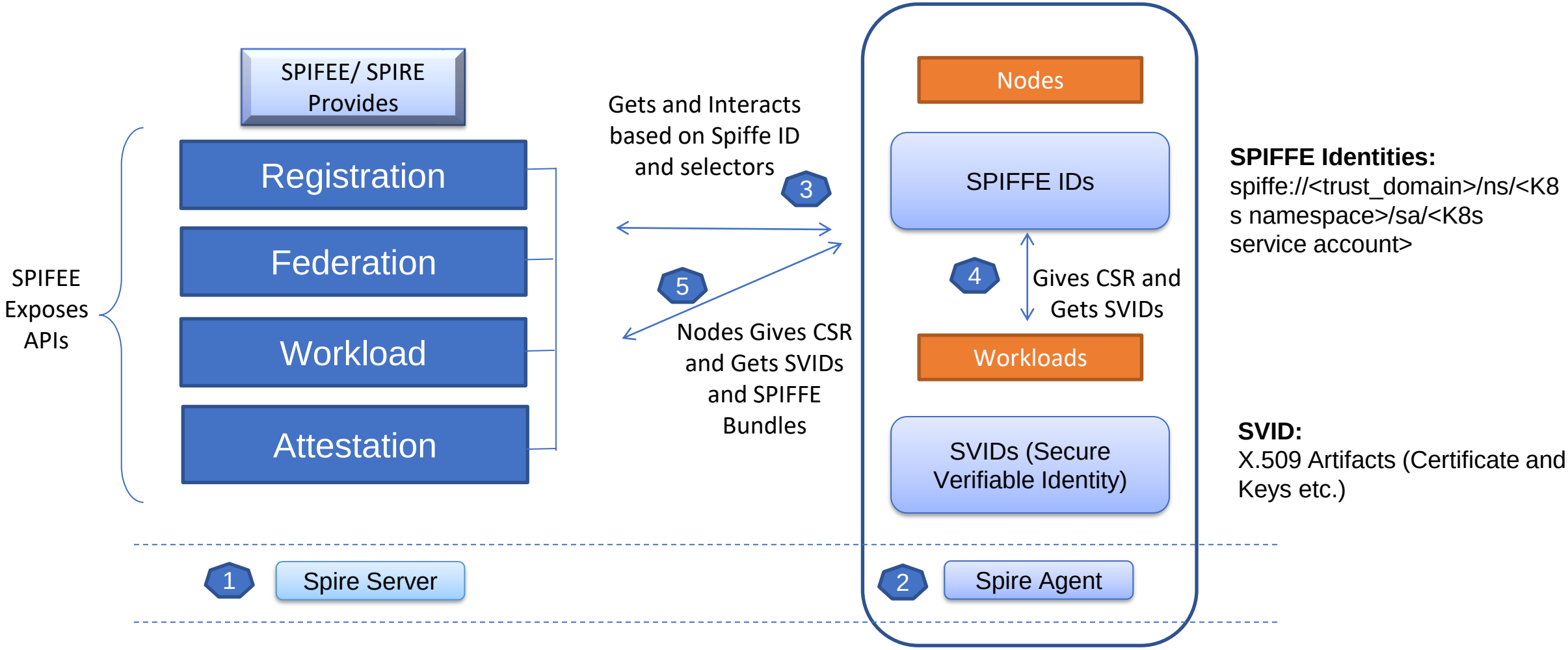
Note: Consider your security requirements and plan your PKI

Demo!

SPIFEE and SPIRE

SPIFEE – Secure Production Identity Framework For Everyone

SPIRE - is a production-ready implementation of the SPIFFE APIs



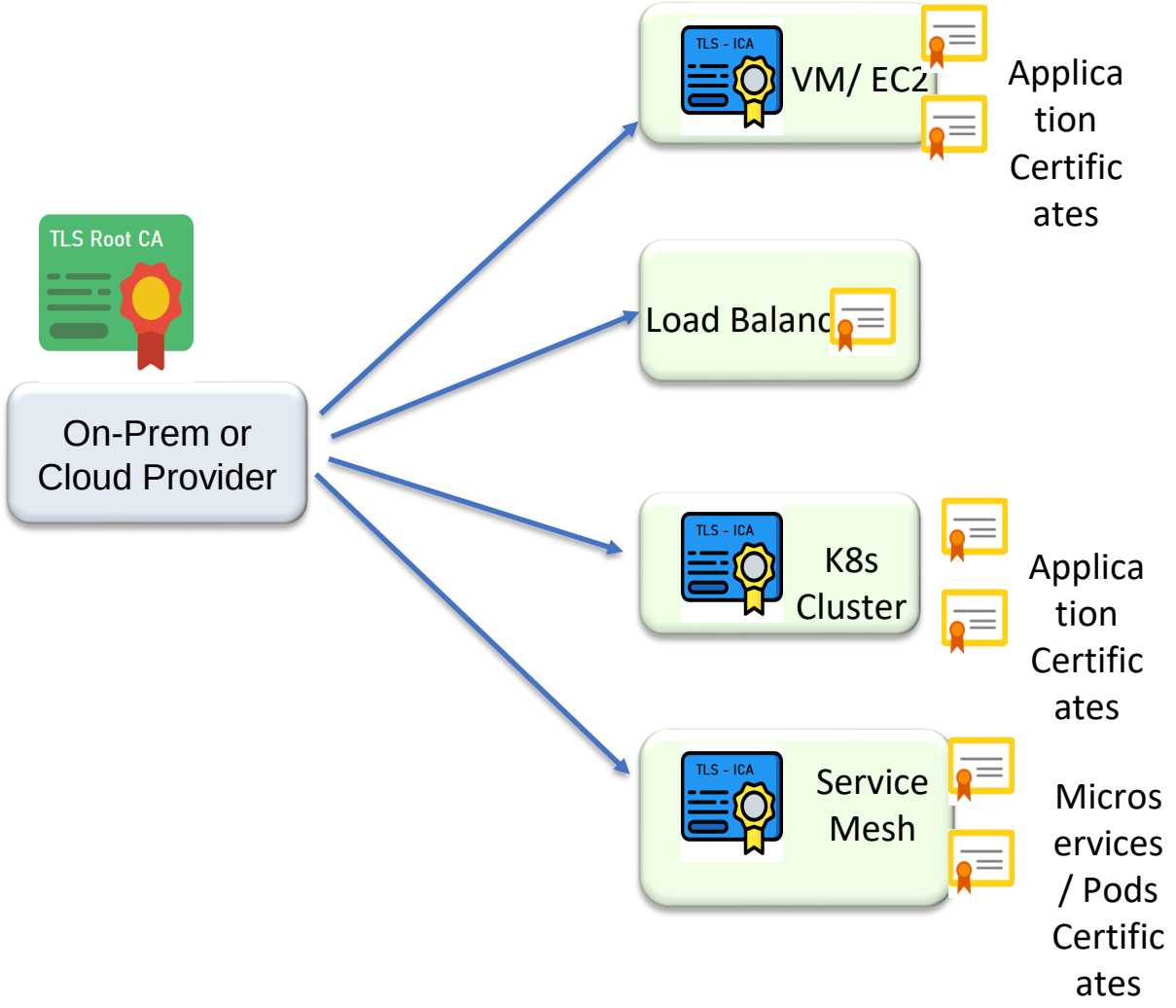
SPIFFE Identities:
spiffe://<trust_domain>/ns/<K8s namespace>/sa/<K8s service account>

SVID:
X.509 Artifacts (Certificate and Keys etc.)

5 PKI Design Decisions – You must know!

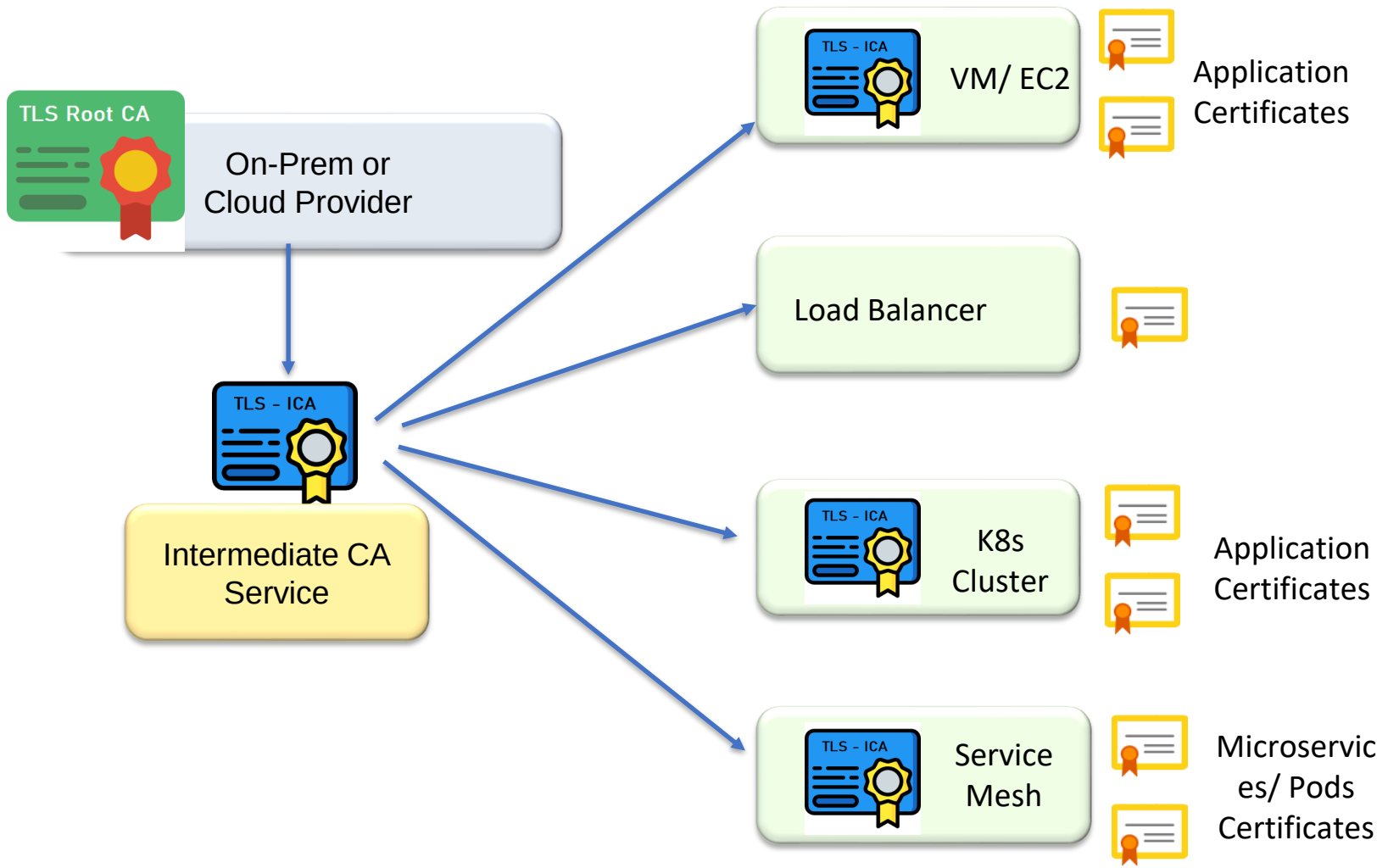
1 - Design (or Know) your certificates chain and hierarchy

Single Root CA Authority



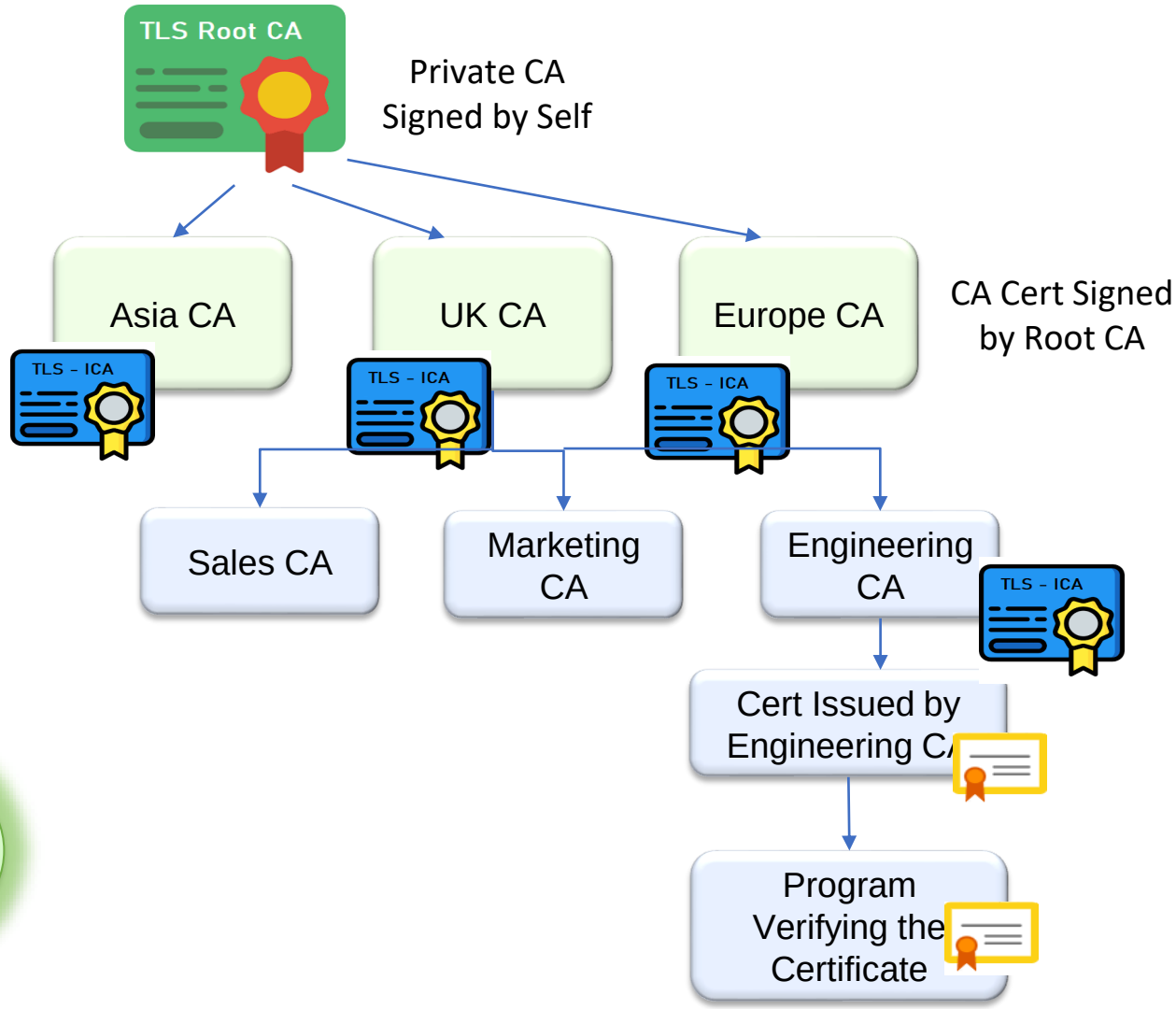
1 - Design (or Know) your certificates chain and hierarchy

Multiple Intermediate CAs



1 - Design (or Know) your certificates chain and hierarchy

Geographical or Departmental Distribution



Design Wisely!
Based on Size, Criticality, Levels of Security, Operational Capacity

2 - Where to terminate your certificates?

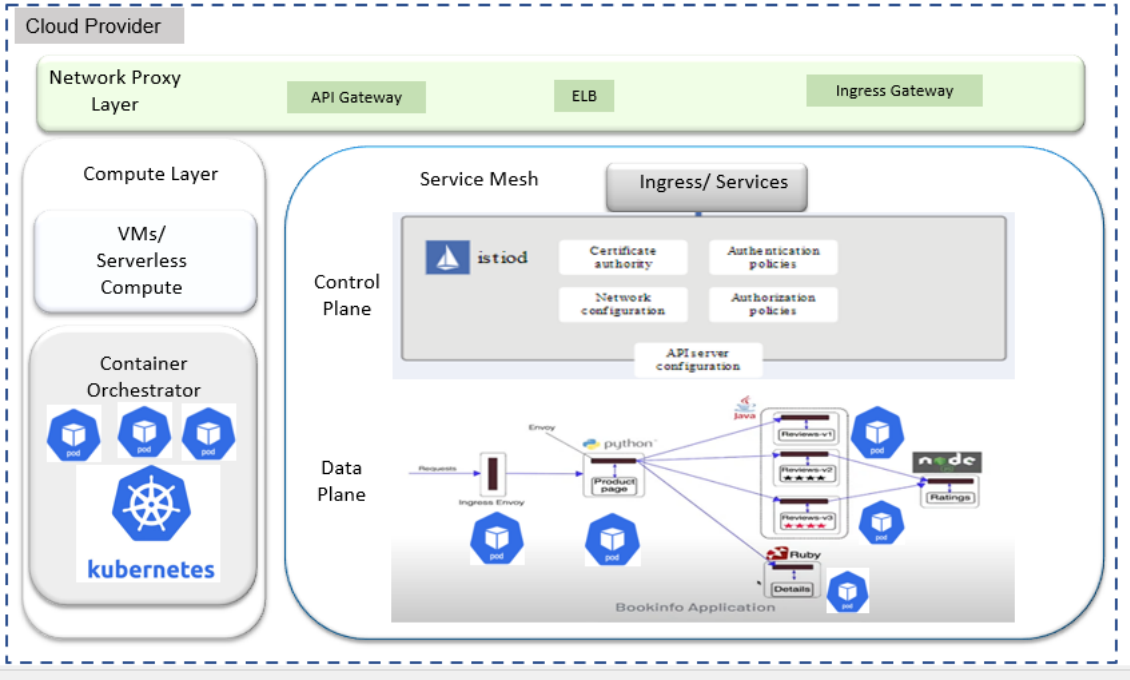
- Wise to probe where to terminate certificates in your application context:

Network Layer

- Terminating TLS at API Gateway
- Terminating TLS at Load balancer
- Terminating TLS at Ingress controller

Compute layer

- Terminating TLS at VM (e.g. EC2)
- Terminating TLS at Pod/ Container



3 - TLS version mis-matches & design decision

- How to deal with TLS version mis-matches (in complex hybrid systems)?
  - For example what if TLS and mTLS needs to co-exist
  - TLS – Works on one way trust verification
  - mTLS – Establishes two way trust. It further enhances the security with two way trust and handshake
    - Some infrastructure services provide mTLS by default such as Istio service mesh. It automatically configures workload sidecars to use mutual TLS when calling other workloads

```yaml
apiVersion: security.istio.io/v1beta1
kind: PeerAuthentication
metadata:
  name: default
  namespace: foo
spec:
  mtls:
    mode: PERMISSIVE
---
apiVersion: security.istio.io/v1beta1
kind: PeerAuthentication
metadata:
  name: default
  namespace: foo
spec:
  selector:
    matchLabels:
      app: finance
  mtls:
    mode: STRICT
```

- In such cases know the ways to co-exist
- Example: Use mTLS mode for peer authentication:
  - **STRICT** – Strictly enforce same protocol across
  - **PERMISSIVE** – Use for mTLS to TLS relaxation (on port or service)
  - **DISABLE** – connection is disabled and not tunnelled

4 – Certificate Revocation Methods & Design

CRL
(Certificate Revocation List)

- CRL contains a list of revoked certificates essentially, all certificates that have been revoked by the CA or owner and should no longer be trusted

OCSP
(Online Certificate Status Protocol)

- OCSP service checks for certificate status
- An OCSP response contains one of three values: “good”, “revoked”, or “unknown”

OCSP - Stapling
(Online Certificate Status Protocol – Stapling)

- Its an enhancement to the standard OCSP protocol.
- It eliminates the need for a browser to send OCSP requests directly to the CA.
- Instead, the web server caches the OSCP response from the CA and when a TLS handshake is initiated by the client, the web server “staples” the OSCP response to the certificate it sends to the browser.

Relevance:
- CRL Design is often overlooked aspect of PKI design
- Consider - How to revoke certificates and keep all informed when required?

Recommendations:
- For internal Private CAs – you might want to use CRL and custom automation
- For Public CA – OCSP or OCSP Stapling can be used

5 - Certificate Automation and Monitoring

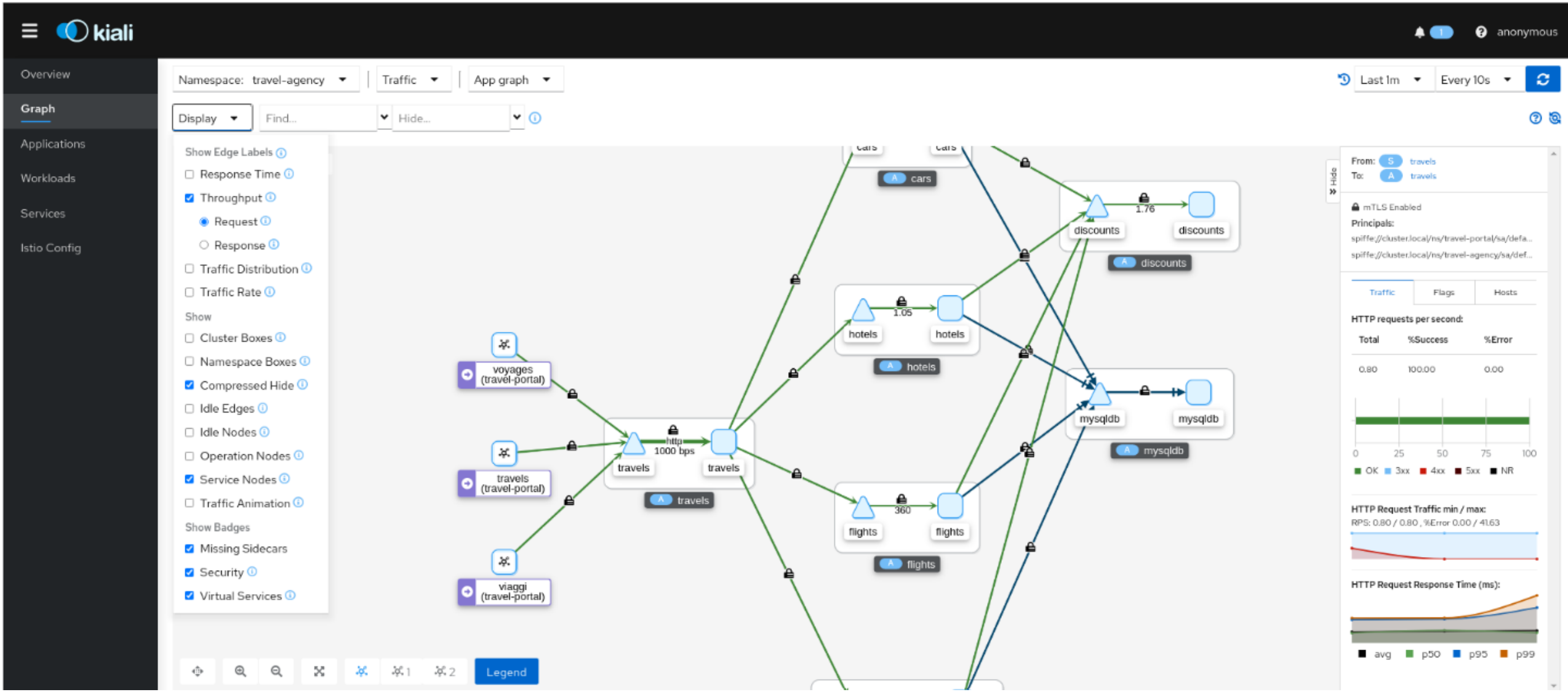
- This is an areas which is many a times unplanned in PKI setup
- What & How much to automate Operations? – Try to strike the balance.

- Few Suggestions:
  - Create and Using Certificate Template

  - Utilize certificates rolling deployment – to avoid downtime due to certification renewal or expiry
    - Use tools that automatically take care of certification lifecycle such as:
      - Cloud provider services - AWS or Google Certificate manager
      - SPIFFE/SPIRE that takes away burden of certification regeneration and rotation etc.

  - Use tools for Monitoring, Tracing and Alerting for better and easy certificates management


SSL Monitoring (x509_cert)
Last 6 hours
SSL Certificate
All
https://freenas.jorge...
-2.93 day
https://veeamone.jor...
9.98 year
https://veeamtech.dd...
2.60 week
https://www.jorgedel...
34.81 week
tcp://veeamtech.ddn...
2.61 week
tcp://veeamtech.dd
2.60 wee
SSL Certificate
SSL Expiry in
tcp://veeamtech.ddns.net:8086/
2.60 week
tcp://veeamtech.ddns.net:8086
2.61 week
https://www.jorgedelacruz.es:443/
34.81 week
https://veeamtech.ddns.net:3000/
2.60 week
https://veeamone.jorgedelacruz.es:1239/
9.98 year
https://freenas.jorgedelacruz.es:443/
-2.93 day


kiali
anonymous
Overview
Graph
Applications
Workloads
Services
Istio Config
Namespace: travel-agency
Traffic
App graph
Last 1m
Every 10s
Display
Find...
Hide...
Show Edge Labels
Response Time
Throughput
Request
Response
Traffic Distribution
Traffic Rate
Show
Cluster Boxes
Namespace Boxes
Compressed Hide
Idle Edges
Idle Nodes
Operation Nodes
Service Nodes
Traffic Animation
Show Badges
Missing Sidecars
Security
Virtual Services
voyages (travel-portal)
travels (travel-portal)
viaggi (travel-portal)
travels
http 1000 bps
travels
travels
cars
discounts
1.76
discounts
discounts
hotels
1.05
hotels
hotels
mysqldb
mysqldb
mysqldb
flights
360
flights
flights
Legend
From: S travels
To: A travels
mTLS Enabled
Principals:
Traffic
Flags
Hosts
HTTP requests per second:
Total
%Success
%Error
0.80
100.00
0.00
OK
3xx
4xx
5xx
NR
HTTP Request Traffic min / max:
RPS: 0.80 / 0.80 , %Error 0.00 / 41.63
HTTP Request Response Time (ms):
avg
p50
p95
p99

KubeCon | CloudNativeCon North America 2022

BUILDING FOR THE ROAD AHEAD

DETROIT 2022

Question?
& Feedback

Please scan the QR Code above to leave feedback on this session