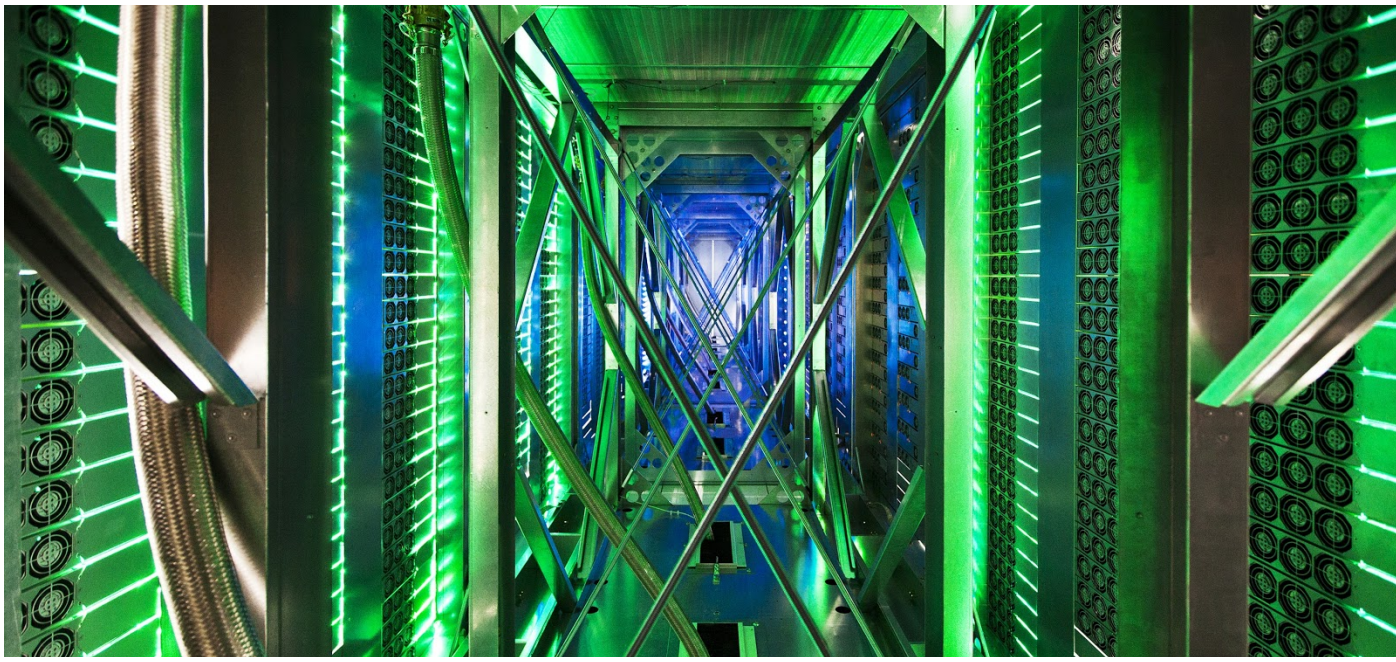
Google Cloud

Google Cloud Whitepaper
February 2021

How to deploy a secure and reliable public key infrastructure with Google Cloud Certificate Authority Service



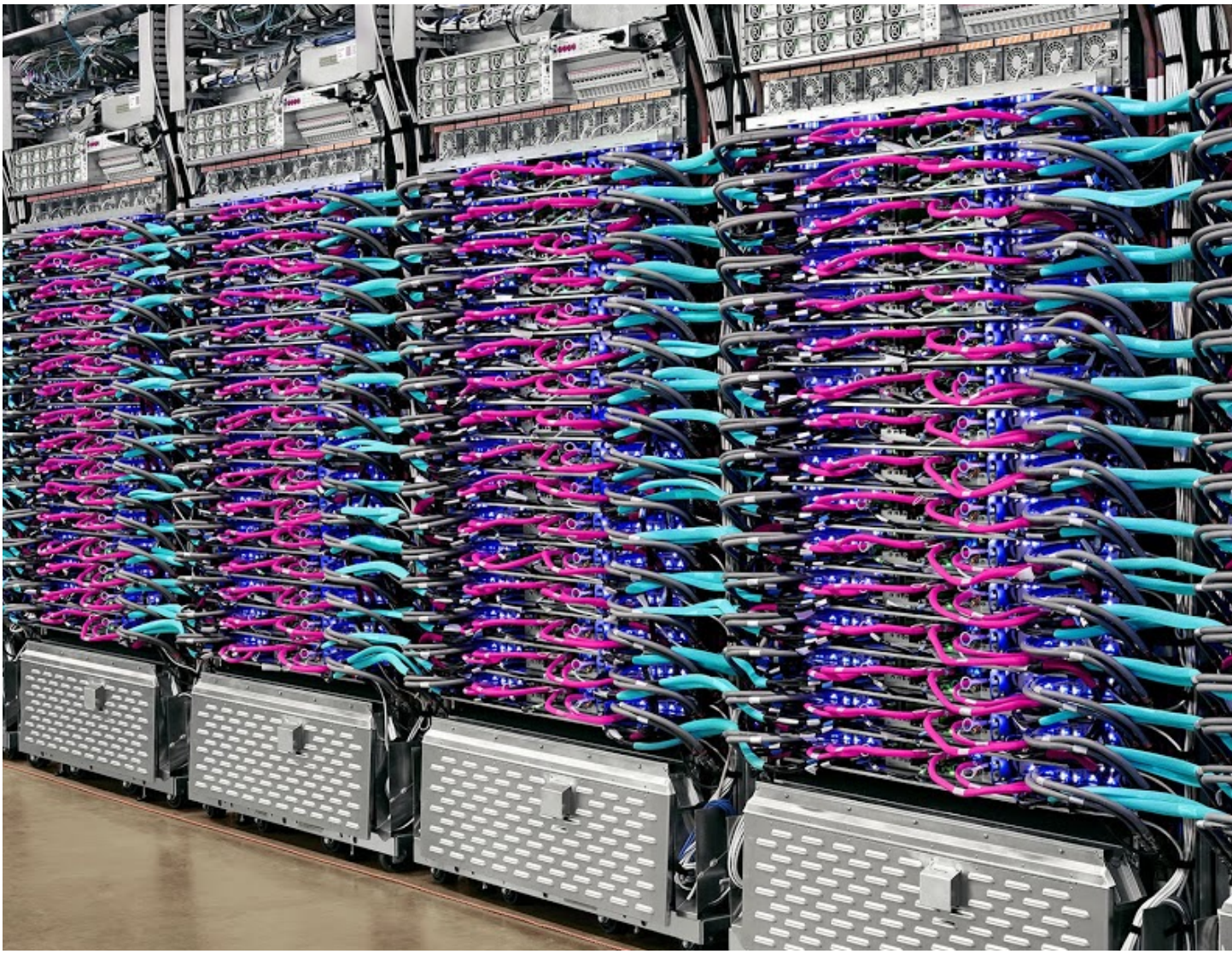
Anoosh Saboory, Product Manager, Google Cloud
Mark Cooper, Founder, PKI Solutions

Google Cloud

Google Cloud

Table of Contents

Google Cloud



Abstract

This white paper is designed to provide security and architectural recommendations to organizations for the use of the Google Cloud Certificate Authority Service (CA Service or CAS). It describes critical concepts to securing and deploying a PKI and specific recommendations for configuring CAS to improve security, provide operational high availability and to leverage the capabilities of CAS.

DISCLAIMER: This whitepaper applies to Google Cloud products described at cloud.google.com. The content contained herein represents the status quo as of the time it was written. Google's security policies and systems may change going forward, as we continually improve protection for our customers.

Google Cloud

Certificate trusts – public versus private PKI

The creation or use of a public key infrastructure (PKI) to issue certificates is largely dependent on the environment in which the PKI issued certificates will be used. For common internet facing services such as a website or host where visitors to the site are largely unknown to the host, a certificate that is trusted by the visitor is needed to ensure a seamless validation of the host. If a visitor's browser hasn't been configured to trust the PKI from which the certificate was issued, an error will occur. To facilitate this process, publicly trusted certificate authorities provide the role of trust issuers of certificates that can be broadly trusted throughout the world. However, their structure, identity requirements, certificate restrictions, and certificate cost make them ineffective for certificate needs within an organizational or private ecosystem – such as the internet of things (IoT) or DevOps.

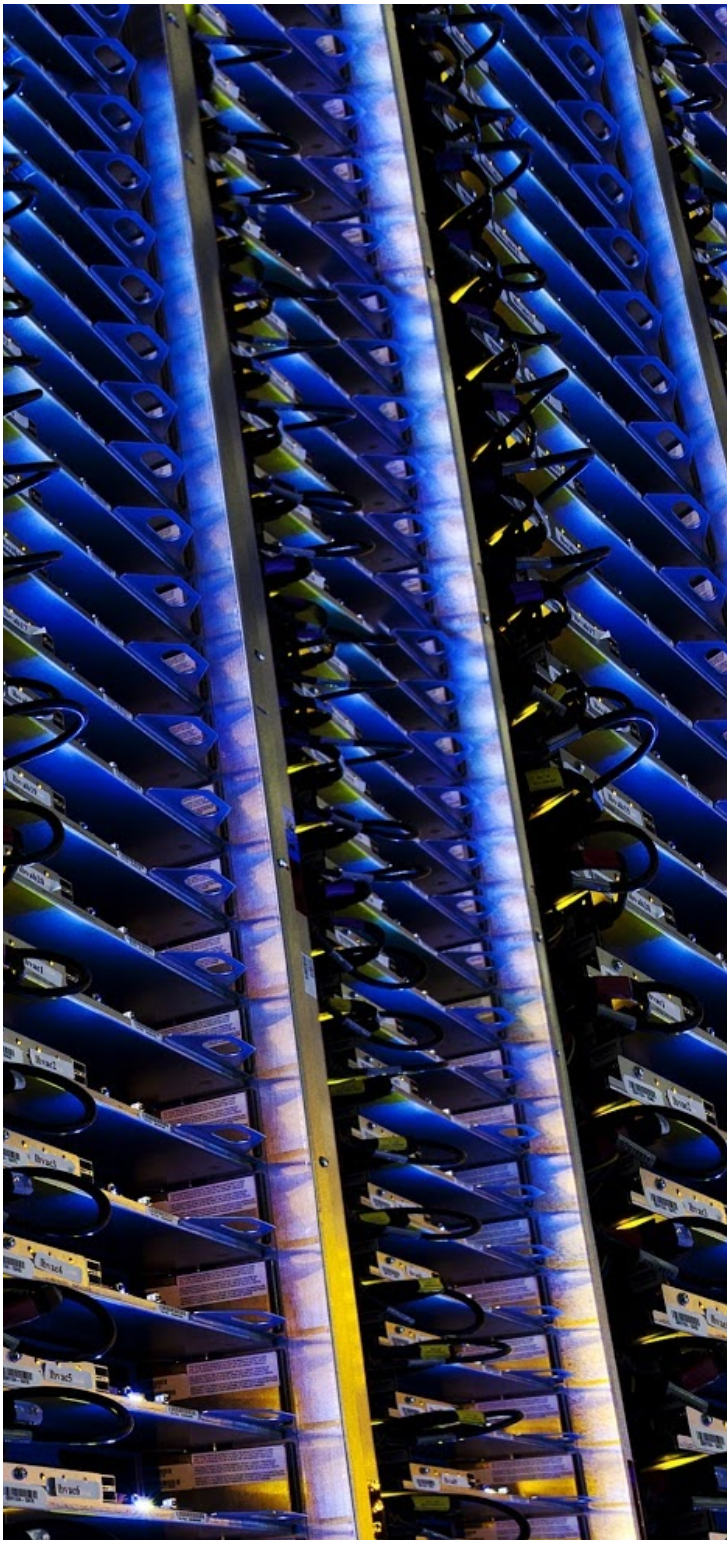
Organizations that have a need for internally trusted certificates and little to no need for external trust of certificates can have more flexibility, control, and security of their certificates without a per certificate charge from commercial providers.

A private PKI can be configured to issue the certificates an organization needs to meet a wide range of use cases and can be configured to do so on a large scale, automated basis. Additionally, an organization can be assured that externally issued certificates cannot be used to access or connect to organizational resources.

The Google Cloud Certificate Authority Service (CAS) allows organizations to establish, secure and operate their own private PKI. Certificates issued by CAS will be trusted only by the devices and services an organization can configure to trust the PKI. The certificates will not be trusted by external parties or broadly on the internet. Accordingly the certificates issued by CAS are not part of the Google Trust Services program.

Google Cloud enabled PKI architectures

CAS enables organizations to flexibly expand, integrate or establish a PKI for their needs. CAS can be used to establish and operate as an organization's entire PKI or can be used to act as one or more CA components in the PKI along with on-premises or other CAs.



PKI is based on a hierarchy model, with one or more Certificate Authorities (CA) in a parent/child relationship. The top tier CA in any PKI is the root CA and is responsible for forming the Trust Anchor of the PKI. To properly participate and use certificates in a PKI, a device, software, or component needs to trust the PKI – this is typically accomplished by explicitly configuring the component to trust the root CA. As a result, all the certificates issued by the CA are trusted.

Subordinate or Intermediate CAs are generally responsible for issuing certificates directly to end-entities such as users, computers and devices. These CAs are cryptographically signed by a parent CA, often the root CA. As a result, systems that trust the root CA, automatically trust the subordinate CAs and end-entity certificates they issue.

A root CA should always be carefully designed and configured to ensure it is securely operated. The entire trust of the PKI will be dependent on how well the root (and subordinate CAs) are protected. The use of proper access controls, roles and practices are critical to maintain the security of a CA. The use of Hardware Security Modules is highly recommended – especially on root CAs. Refer to the Security Considerations section for more details.

There are several architectures that could be implemented to achieve goals within your organization and your PKI.

Cloud architecture

This architecture leverages CAS for both the root and issuing CA roles. CAS will be used to create and operate the root CA as well as all the subordinate CAs. This model is ideal for organizations that wish to have a fully functional PKI without requiring on-premises equipment, workloads, or security appliances and associated CapEx/OpEx costs. This simplified architecture can be easily expanded to address high availability and regional requirements as detailed in section Building for a Regional or Global Footprint.

> **Recommendation:** For organizations that wish to deploy a PKI with no on-premises equipment, this approach provides a full PKI structure with no physical or logical deployment requirements in the organization. However, additional work or components may be needed to fully integrate enrollment to endpoints and users as appropriate.

Benefits

Full cloud-based workload for PKI – easily deployed

No on-premises hardware, controls or appliance required

Root and subordinate CAs can leverage Google Cloud Key Management (KMS) with supported Hardware Security Module (HSM) for key protection

CA roles protected through role-based access controls (Cloud IAM)

CAs can be regionalized for higher availability

CA Integration with Google Cloud monitoring products such as Cloud Audit Log and Cloud Monitoring

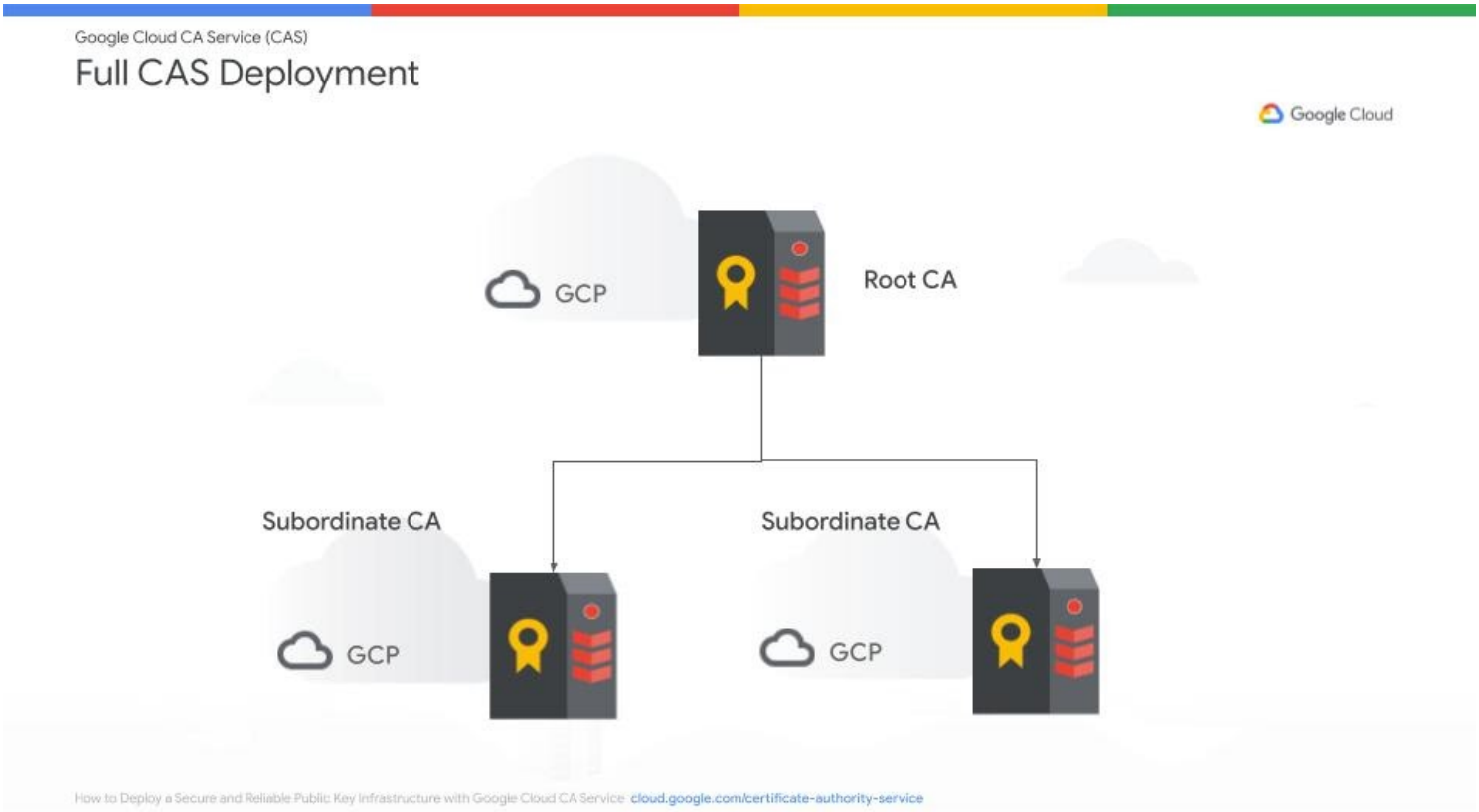
CAS infrastructure is maintained and secured to industry standards SOC, ISO 27001, and FedRAMP (where applicable).

Cons

Integration and enrollment for some environments will take additional integration work or the use of third-party products

Existing on-premises CAs are not easily migrated if currently protected with Hardware Security Modules

New trust anchors need to be distributed to clients



Cloud issuing CAs

Organizations that prefer to operate and secure a root CA within their own organizations can configure and deploy the CA on the platform of their choice and utilize CAS to create subordinate CAs. This direction allows an organization to physically own and control their root CA keys as well as all the controls and access to the root CA. Typically these root CAs are air-gapped from the network and closely controlled and secured to ensure unauthorized access or use is prevented. As root CAs are only used a few times a year, their offline operation is facilitated through manual operations. CAS is then used to create the necessary subordinate/issuing CA infrastructure for the issuance to end-entities in the organization or ecosystem. The root CA would be used to sign the subordinate CA certificate.

> **Recommendation:** For organizations that wish to create and maintain their root CAs and associated signing keys, but leverage CAS for issuance, this model provides an architecture with minimal on-premises equipment while leveraging CAS for end-entity certificate needs.

Benefits

Root CA is created and operated by organizations and is 100% under their control, the trust anchor continues to work across environments

Air-gapped/offline CA provides physical and logical security around the root CA

Physical HSMs can be used to secure root CA

Subordinate CAs can leverage Google's Cloud HSM service for key protection

Subordinate CA roles protected through IAM

Subordinate CAs can be deployed regionally to provide high availability

Cloud/DevOps integration is provided by Subordinate CAs

CAS infrastructure is maintained and secured to industry standards SOC, ISO 27001, and FedRAMP (where applicable).

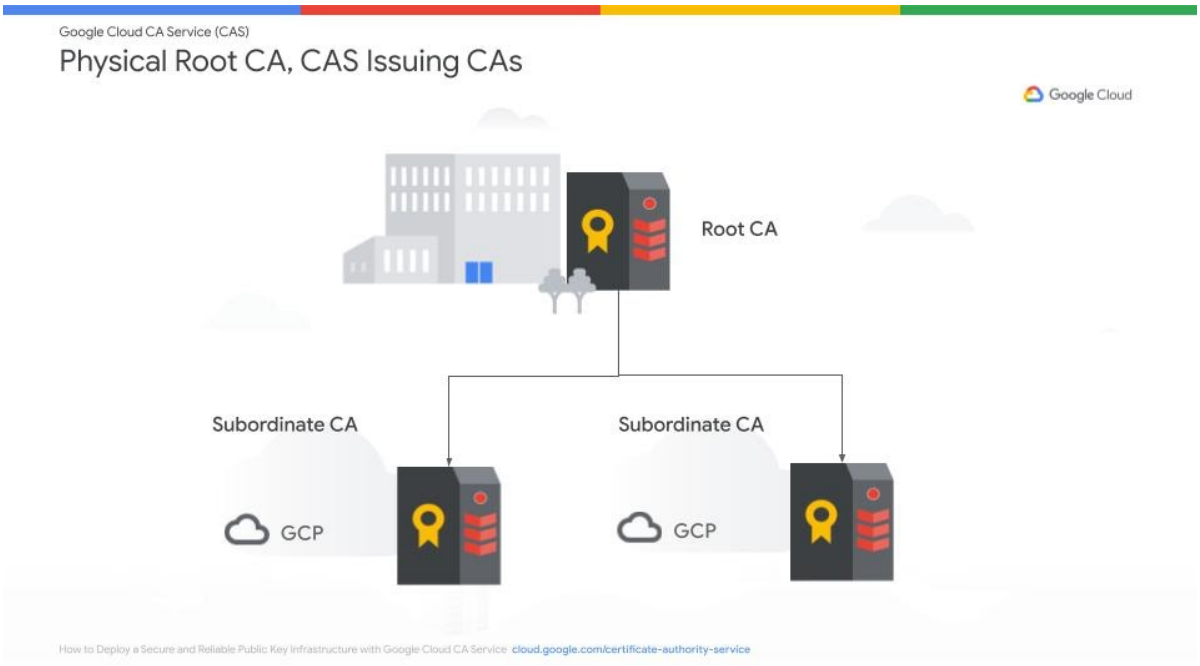
Cons

Integration and enrollment for some environments will take additional integration work or the use of third-party products

On-premises hardware, security controls, licenses and HSMs are required for root CA

Organizations will need to manage both a cloud-based CA and on-premises CA with different operational controls and processes (and plan for on-premises CAs redundancy, recovery, and global high-availability)

CA constraints and policies need to be carefully designed and applied to CAs in both environments to limit potential exposure due to compromise or misuse.



Cloud root CAs

For organizations that have a need or desire to operate on-premises subordinate/intermediate CAs, CAS can be leveraged to create and operate a root CA securely. Root CAs have little operational responsibilities once established and are easily facilitated through manual access through the CAS interface. The root CA can be easily configured to use Google's Cloud HSM for key protections and IAM can limit administrative staff allowed to operate and interact with the root CA. To provide additional security, the root CA can be set to "inactive" to mimic an offline root CA - this would prevent the CA from issuing new certificates until the CA is re-enabled. During this time CRLs would still be created by the root CA.

> **Recommendation:** This approach can be an excellent choice for organizations that do not wish to manage an offline/air-gapped offline CAs such as root CAs and associated maintenance tasks. For organizations moving away from physical data center systems, the addition of a physical root CA may not be desired. Additionally it eliminates the need to purchase physical HSMs to secure and protect the CA signing keys.

Benefits

Root CA can be easily configured and deployed in CAS

No air gapped/offline CA to monitor and track within organization

No on-premises hardware, controls or appliances required for root CA

Root CA can leverage Google's Cloud HSM service for key protection

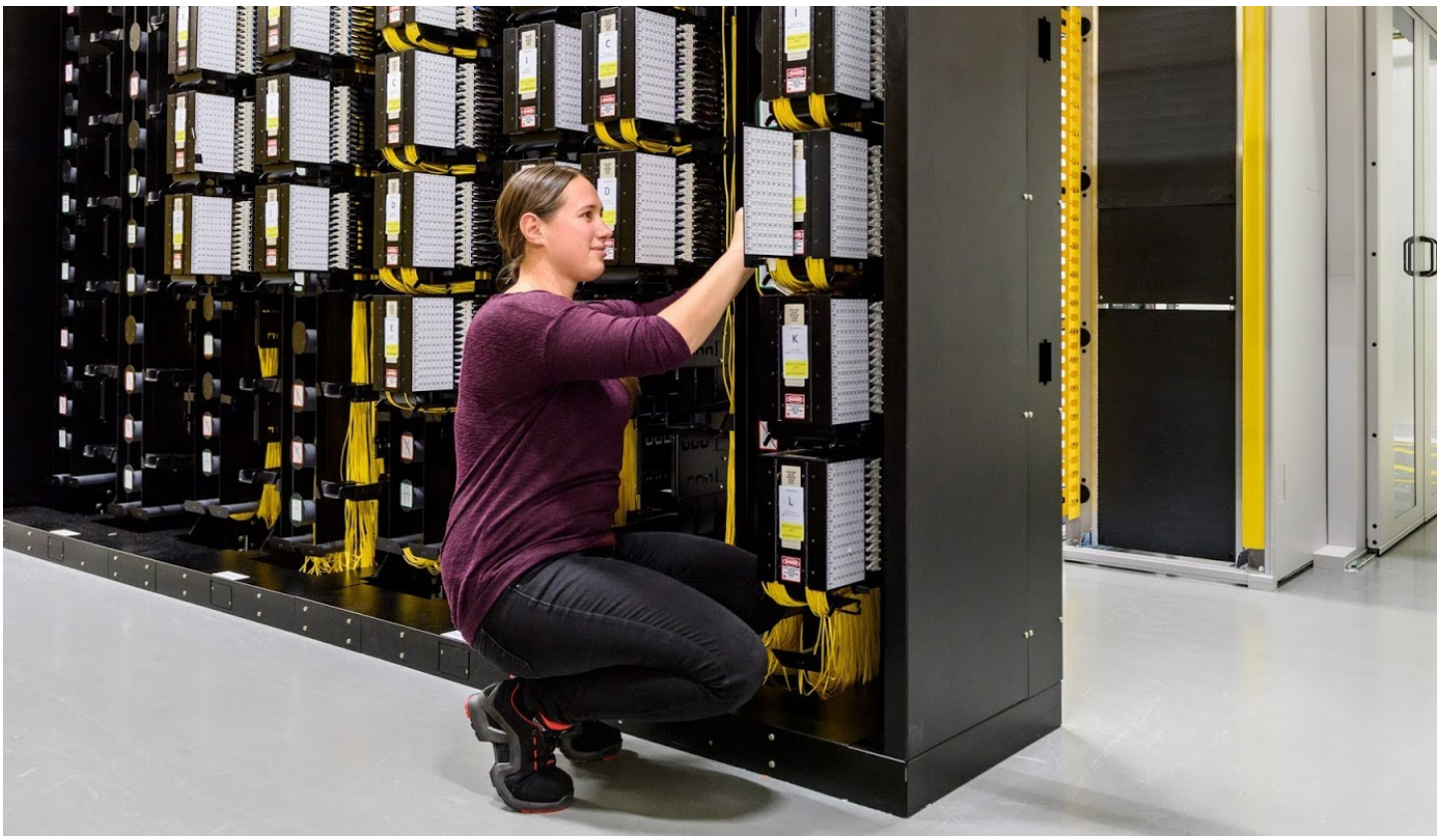
Organizations can deploy integrated subordinate CAs within their environments to facilitate automated enrollments

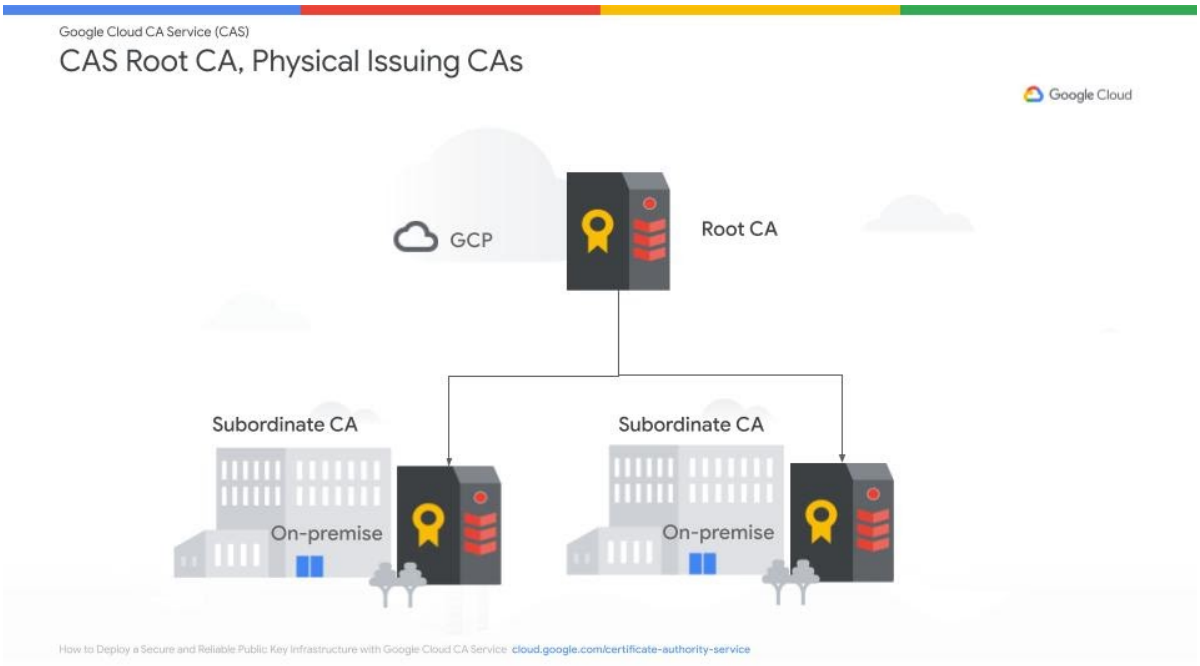
On-premises CAs would be physically and logically closer to workloads which could provide better latency

Cons

Organizations will need to manage both a cloud-based CA and an on-premises CA with different operational controls and processes

Existing on-premises root CAs cannot be migrated if currently protected with Hardware Security Module





Mixed cloud and on-premises

To maximize the benefits of each deployment approach, the Hybrid Architecture leverages both on-premises and CAS based roles to provide better security, integration, and availability. In this architecture, the root CA is deployed either on-premises or CAS and subordinate CAs are both on-premises and CAS based. The on-premises CAs can provide better integration with existing authentication solutions such as Active Directory and provide automatic enrollment for supporting systems. CAS based CAs can be leveraged to provide specialized certificates, certificates for web services, IoT, DevOps and for locations requiring regional availability without the complexity of deploying on-premises CAs in multiple locations.

> **Recommendation:** For organizations that have a need for both on-premises integration as well as cloud-based issuance, the hybrid model provides more flexibility. Organizations can choose to deploy the root CA on-premises if they desire to directly own the security and operation of the CA.

Benefits

Organizations can deploy root CA in CAS or on-premises based on security needs

CAs can leverage cloud or on-premises HSMs as desired

Air-gapped/offline CA can help provide better physical and logical security around the root CA

Subordinate CAs are deployed based on various integration and workload needs

CAS Subordinate CA roles protected through IAM

On-premises CAs integrate directly with organization environment, directories, and products

CAS Subordinates can be deployed regionally to provide high availability

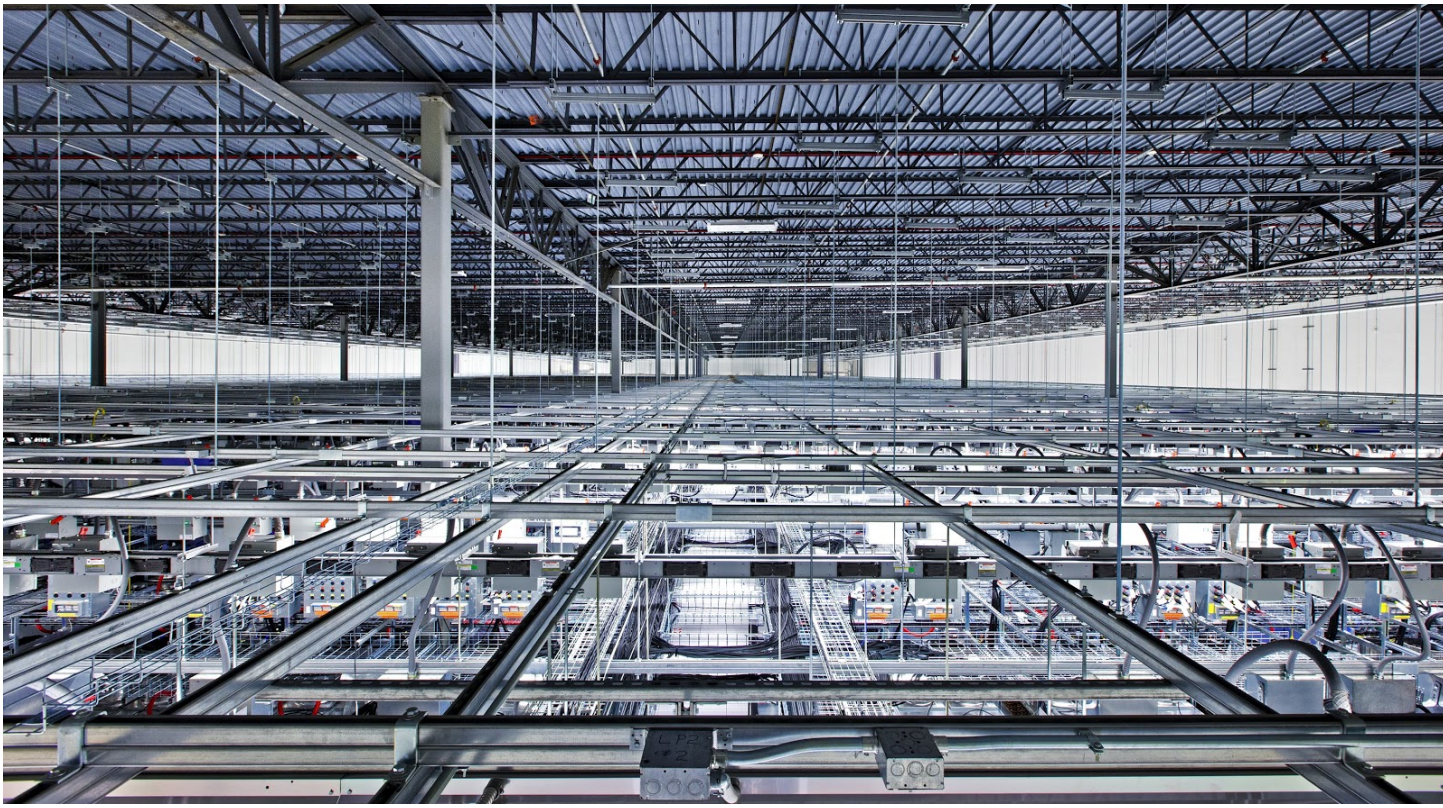
Cloud/DevOps integration is provided by CAS Subordinates

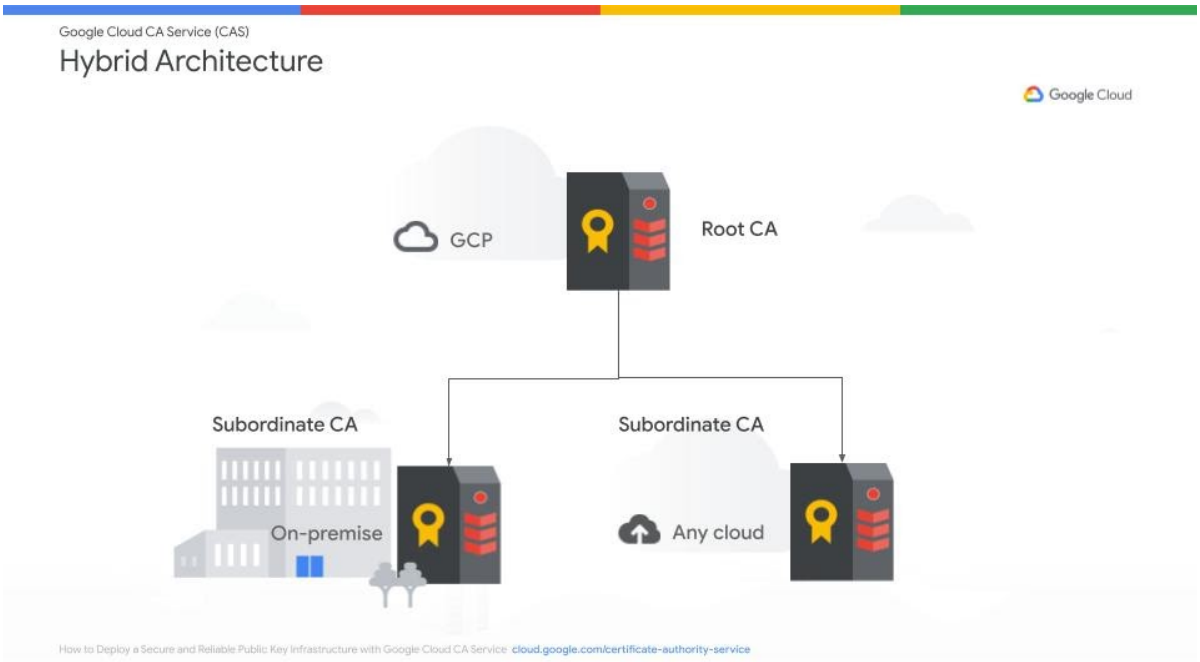
Cons

Organizations will need to manage multiple CA technologies and access controls. Areas such as training, procedures, controls, access rights, role assignment, certificate lifecycle management and others would all require additional oversight.

On-premises hardware, security controls, licenses and HSMs are required for some CAs

On-premises and CAS do not provide redundancy or high availability for the other.





Building for a regional or global footprint

Designing and deploying a PKI for a large organization that is dispersed across distant geographies requires careful consideration to two core areas of the PKI, the issuance of certificates, and the validation of certificates – specifically revocation checking. Together, these two components represent the most challenging and often difficult parts of designing and deploying a PKI. While a CA is responsible for both tasks, the method to achieve the reliability and operational security you desire are based on organization needs. Not every organization has a business-critical need to always issue certificates. Many organizations simply want to make sure existing certificates continue to function if a CA is unavailable. Adding more CAs doesn't achieve this goal – as each CA creates its own revocation information which is needed for validation. On the other hand, if you have a need to issue certificates with a high level of certainty – DevOps, IoT, Network Access Control (802.1x), then the use of multiple CAs is critical.

Regional CAs

Providing a dispersed and highly available PKI for your organization can be greatly simplified through CAS Regionalization. When deploying your CA, you can easily specify the location of your CA. With this approach, you can ensure that you deploy your CAs to locations closest to your organizational needs for issuance. Regional CAs can be leveraged to achieve two distinct design requirements – locality and high availability.

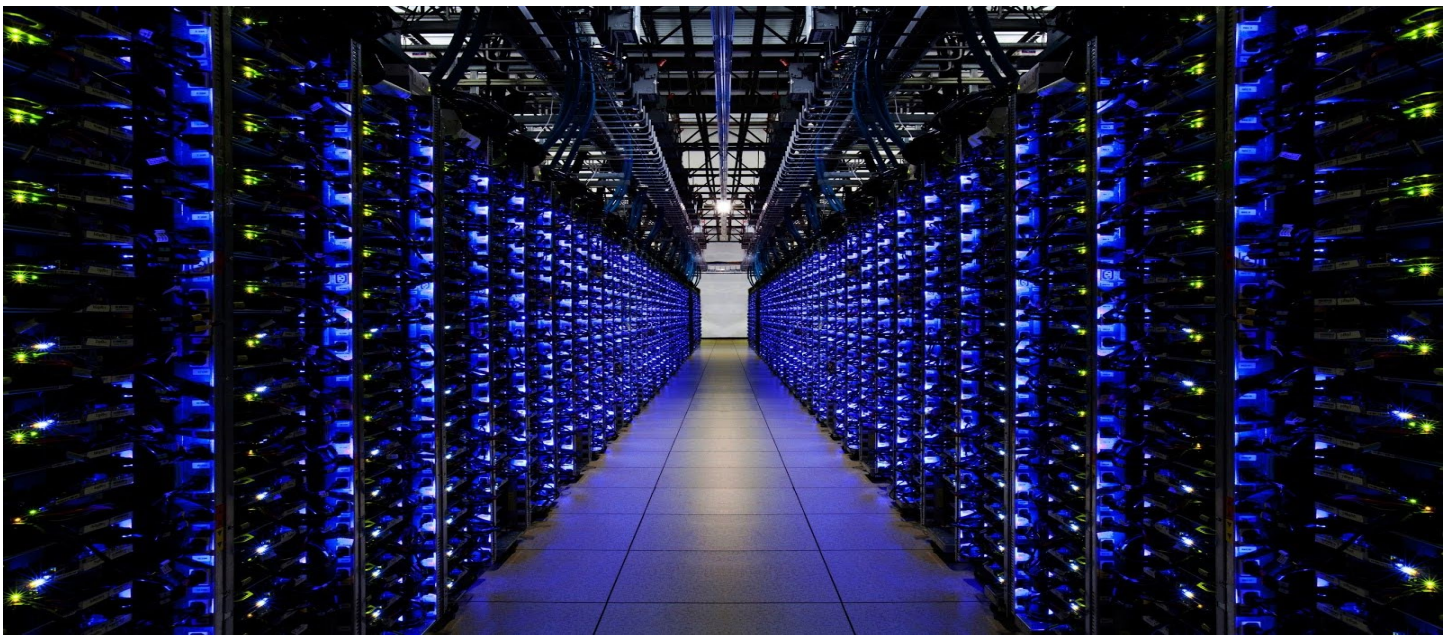
Locality and performance

Organizations that have a need to issue certificates to end-entities spread across a continent or globally can benefit by placing a subordinate CA in multiple regions to provide faster response and reduce the likelihood of network issues preventing a certificate from being issued. Additionally, since the CA creates and publishes its artifacts in a Cloud Storage Bucket in the same zone as the deployed CA, it ensures devices have access to revocation (CRL distribution point) and Authoritative Information Access (AIA) chaining certificates in the fastest possible time without backhauling over long WAN links to services in a remote region. CAS requires KMS signing keys to be placed in the same regional location as the CA – which also reduces network issues and improves response times.

High availability

By configuring and deploying multiple subordinate CAs, organizations can benefit from higher availability for certificate issuance. Multiple CAs configured to issue similar certificates will provide redundancy if another CA was inaccessible or unable to issue a certificate. Additionally, using discrete Google Cloud regions can help geographic separation to provide a higher availability rate. This is particularly important for organizations using DevOps, IoT, Manufacturing or other certificate needs with a very high demand on around the clock issuance.

Placing multiple CAs in different regions also provides additional protection against an outage in any single GCP location or loss of communications. Organizations can also implement multiple internet connections to provide redundancy in communicating with GCP regions to further enhance availability.



CA signing keys

One of the primary security considerations for any PKI is the security and controls for the underlying cryptographic private keys associated with certificates. The proper control of the underlying cryptographic key pair for CA certificates determines the security and integrity afforded by the PKI. Without proper consideration or controls, a PKI may appear to provide enhanced identity and encryption but provide little value if the keys are not securely held and associated with the identity holder.

When designing and deploying a PKI, it is critically important to think about the CA keys, including root and all subordinate CAs. The keys should be protected appropriately for their importance to the PKI as well as ensuring organizational threat profiles are mapped to key types and protections.

Hardware Security Modules (HSM)

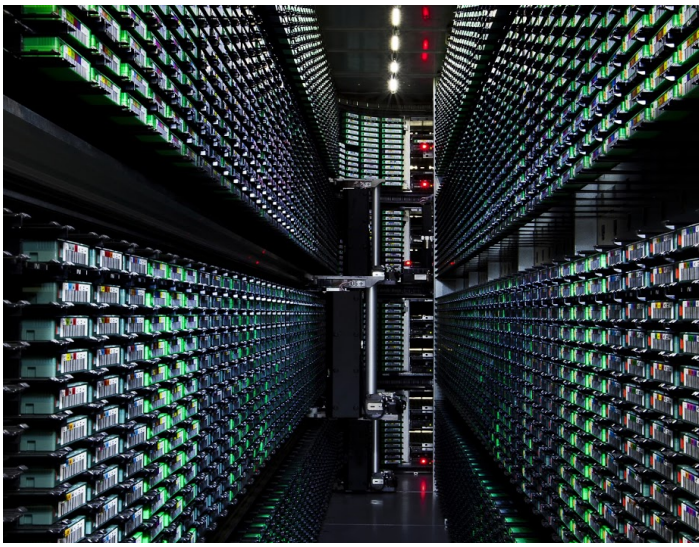
One of the best ways to protect CA signing keys is to ensure they can't be duplicated, extracted or used on an unauthorized device. Any software based cryptographic key is vulnerable to backups, extraction or copy operations. Separate hardware-based mechanisms provide a cryptographically secure enclave to generate, store and utilize cryptographic keys. They are designed to prevent the extraction of keys and can limit access and use of keys to authorized devices and processes. If a CA is configured to use an HSM, one can be reasonably assured its key exists in one known place – the HSM.

For example, Hardware Security Modules are generally designed to achieve security requirements defined in the Federal Information Processing Standard (FIPS) 140-2 (https://csrc.nist.gov/publications/detail/fips/140/2/final) which offers a variety of protections for cryptographic keys. HSMs generally conform to Level 3 and Level 4 protections for keys that are important for compliance regimes (e.g., PCI DSS or FedRAMP). HSM manufacturers submit their devices to authorized testing labs to be independently certified to these standards. A newer standard FIPS 140-3 has been introduced to replace FIPS 140-2 but has not seen widespread adoption as of 2021.

Recommendation: CAS can be automatically configured to use Google managed keys which leverages Google's Cloud HSM for generating, storing and utilization of keys. However, if you wish to use your own key or leverage another key already stored in KMS, you can import the key to KMS (with or without Google's Cloud HSM protection) and choose to use the key during the setup of the CA. Note, once you create or import a key to an HSM, it is generally impossible to extract or migrate that key to another platform.

Google managed vs customer managed keys

When configuring and deploying a new CA with CAS, you are required to specify the CA key size and algorithm for the CA. When an organization has cryptographic needs that align with the default options, the Google Managed key provides easier creation, storage, and protection of the CA key. The service will automatically leverage Google's Cloud HSM to generate and secure the CA key. However, there may be some scenarios where the organization wishes to use a customer managed key. This could include a scenario where you already have a keypair for the CA on-premises - for instance if you are migrating an existing CA into CAS. You can import the key into Cloud KMS and then specify the key when creating a CA in CAS.



Google managed keys are using Google's Cloud HSM and as a result are not accessible or usable by any other organization. Access and use of Google's Cloud HSM signing keys are auditable through Cloud Audit Logs.

Recommendation: If you do not have a specific security or operational requirement for a customer managed signing key, the Google managed key provides a simplified key generation, storage, and utilization system.

Google Cloud

Customer Managed Keys

CAS provides flexibility for the creation and use of CA signing keys through both Google Managed and Customer Managed Keys. This is further enhanced with the option to Bring Your Own Key (BYOK) that enables organizations to import a key into Cloud KMS (with or without Cloud HSM) and the signing keys for their CAs in CAS. This can provide flexibility to meet specific requirements for organizations.

Importing external CA signing keys

Utilizing bring-your-own-key approach or External Key Manager, organizations can leverage an existing signing key for a CA and migrate that CA to CAS. By exporting the existing CA key and importing to Cloud KMS, CAS can be configured to assume the identity of an on-premises CA. Once configured, the certificates issued by CAS will be signed with the same keypair as an on-premises CA. This can be used to continue to publish a CRL after a CA is decommissioned or can be used to fully assume the responsibilities of a former CA. Note, it is not possible to import previously issued certificates into CAS. As a result, it is not recommended to migrate an existing external CA with issued certificates into CAS.

Key escrow

BYOK can also be used to ensure your organization retains control and access to the signing keys for your CA. When CAS manages a CA key, it uses Google's Cloud HSM which provides protections from export or extraction of the key. An organization that wishes to maintain a copy of their CA keys can generate keys using on-premises tools and import them to Cloud KMS for use with CAS. You can then safely escrow the keys and maintain possession until needed in the future. Note, while you will import a copy of the key to Cloud KMS, which can be protected by Cloud HSM, your copy of the key will still exist and must be always safeguarded.

CA key sizes and algorithms

Cryptographic key sizes and algorithms define the type and strength of the asymmetric keypair that will be used to prove possession of a certificate and to provide the keys for encrypting information. As CAs live for a relatively long period of time (2-20 years for instance), the strength of their keys must be strong enough to be secure throughout the length of their intended lifetime.

When selecting a CA key size, it is important to understand the viability of a key over the period of intended use. A key size and complexity should align with the sensitivity of information it is protecting as well as the length of time that information needs to be protected. Short lived session-based keys can be much smaller as the amount of information protected with the key is relatively short-lived and the usefulness of the information is transactional in nature. CAs on the other hand sign other CA certificates, revocation lists and end-entity certificates which are valid for longer periods of time.

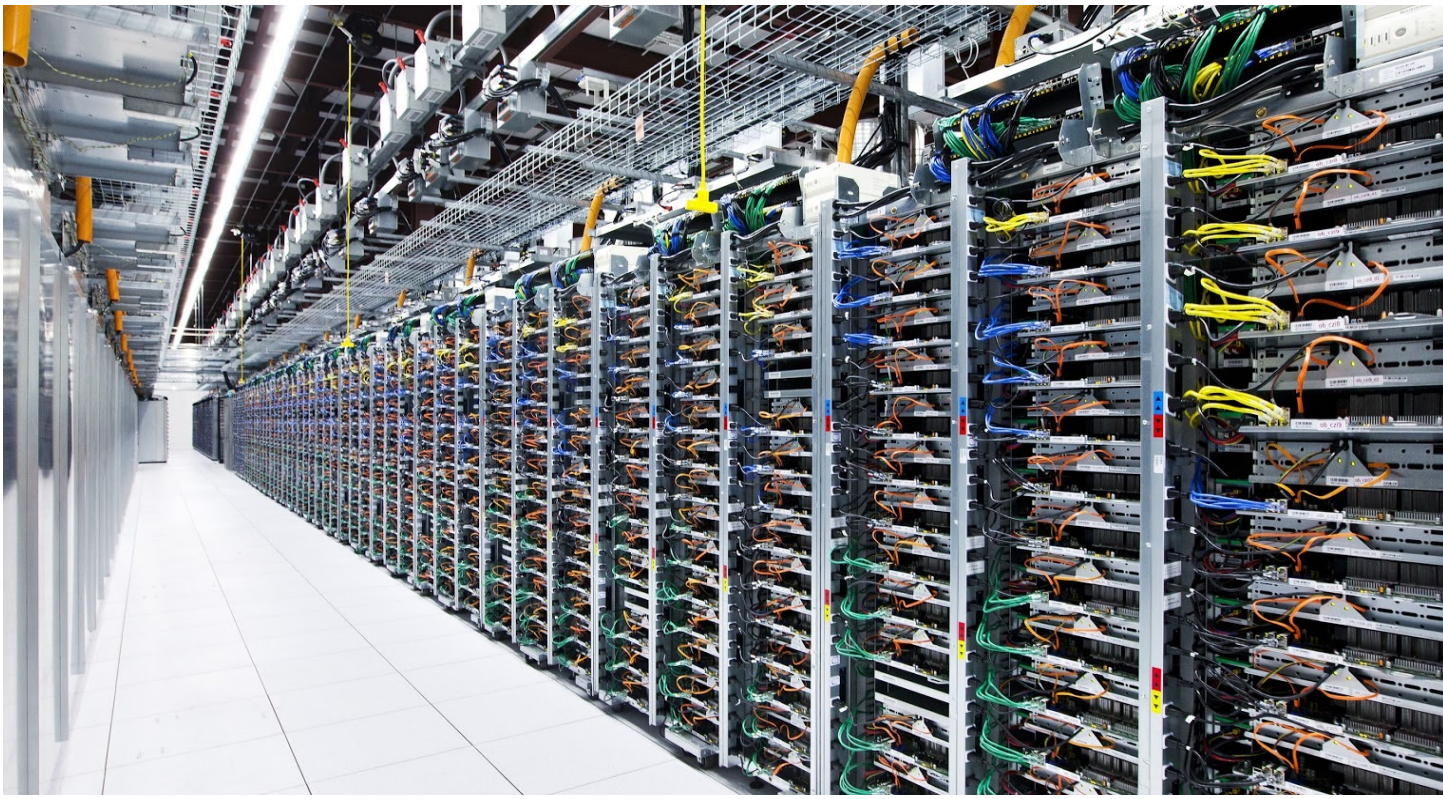
The cryptographic algorithm determines how a keypair is created and the overall strength of a key. The two most common types of algorithms used in PKI today are RSA and ECDSA – both of which are available in CAS. A given key size in RSA is not comparable in strength to ECDSA and vice versa. In general, ECDSA can offer a great tradeoff between performance, security and signature size. However, ECDSA is not supported on all systems and organizations must carefully determine if the intended recipients of certificates can support ECDSA as well as any other device or component that will need to validate a ECDSA certificate. When mixing algorithms in a PKI chain, client compatibility must be considered to ensure certificates will be trusted and validated.

Good sources to determine the longevity and usefulness of cryptographic keys are standards bodies around the world such as ANSSI, BSI, and NIST which publishes the SP 800-57 Recommendation for Key Management.

When selecting a key size for a CA, select as high of a key size as is compatible with your intended certificate holders and validating devices. This should be balanced with anticipated processor load and application specific needs. An RSA 2048 key is widely supported but offers less long-term protection than RSA 4096.

Choosing a key size at the bottom of the commercially viable scale such as RSA 2048 and ECDSA P-256 are acceptable. However, it does present a risk when cryptographic standards change, and recommendations are to use stronger keys. Where possible, strive to use a key size at least one step above minimally viable options. You can also use a tiered approach with larger key sizes for root CAs to correspond to their longer lifetimes, and smaller key sizes for subordinate CAs which have shorter lifetimes and perform renewals to generate new keys much more often than root CAs.

> **Recommendation:** If you have a well-defined PKI environment with modern devices, ECDSA offers can offer a great tradeoff between performance, security and signature size. In organizations with a wide range of systems and uncertainty about key support, it could be sufficient to use RSA based keys. There is generally minimal difference in compatibility between 2048 and 3072 RSA keys but some older systems may have compatibility issues with RSA 4096 keys so they should be used cautiously.



CA service tiers

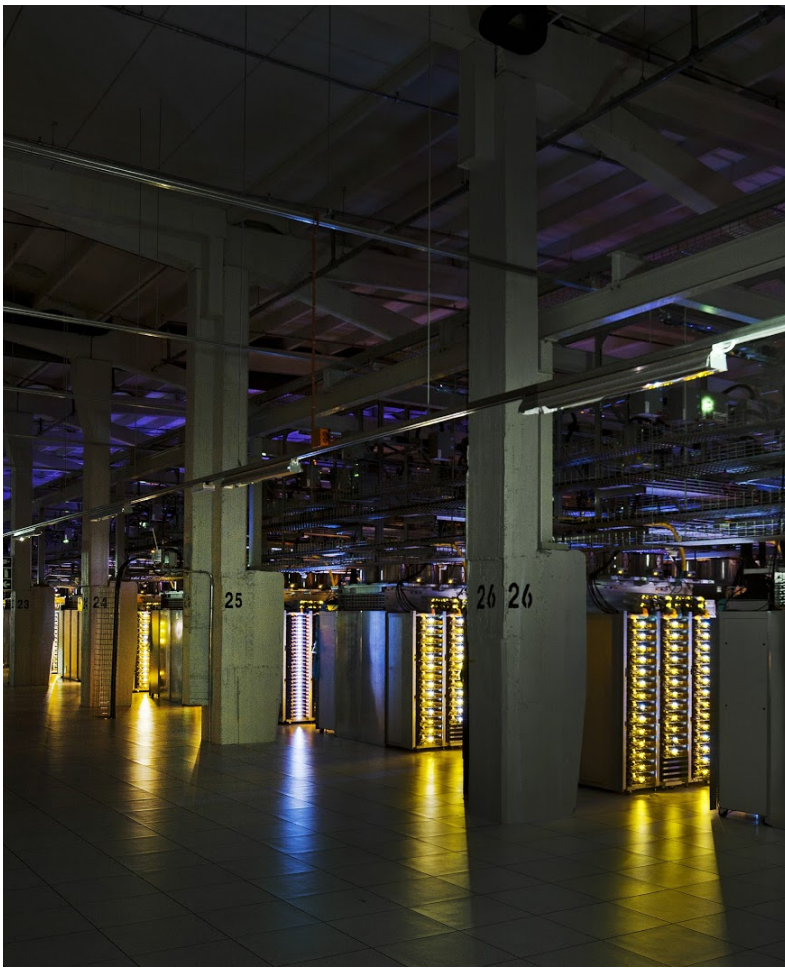
CAS provides two operational service tiers for a CA – DevOps and Enterprise. These two tiers provide organizations with a balance of performance and security based on operational requirements. DevOps is designed to provide faster issuance of certificates at a higher throughput which is helpful in high volume use case scenarios. However, these certificates are not tracked in the CA database and subsequently can't be revoked. Additionally, DevOps CAs only support Google managed keys – customer managed KMS keys are not supported.

Enterprise provides traditional CA operations such as tracking and revoking certificates. It also supports customer managed Cloud KMS signing keys as well as Google managed keys.

> **Recommendation**: The use of DevOps mode should be carefully considered as it does not provide for certification revocation. Since issued certificates are not stored in the CA, the only way to track certificates will be by reviewing Cloud Audit Logs. It should only be used for use cases with short lived certificates that will not need to be revoked – such as microservices, containers, session certificates, non-persistent virtual machines and other isolated needs. A PKI can consist of a combination of CAs providing DevOps and Enterprise to meet a variety of needs. Root CAs should generally be configured as Enterprise as well as any subordinate CAs that will be issuing certificates to other CAs and end-entities such as users and computers.

Google Cloud

CA policies



CA policies enforce standards for the operation of the CA and the certificates it issues. Policies can be thought of as a set of rules that define how the CA will issue certificates and which parameters can be included in a request and which values are accepted. A CA without policies will sign any certificate request that is properly formatted and submitted by an individual with permission to do so. To ensure the CA issues certificates within defined parameters, CA policies should be used to ensure all certificates adhere to standards.

CAS provides CA Policies to control items such as maximum certificate lifetime, subject names, subject alternative names (SAN), and whether preconfigured certificate requests (CSR/REQ) can be submitted.

Every organization will have a variety of CA policies based on issuance needs and control of the CA. Some example policies that can be defined:

- Ensure subject names match organizational details
- Restrict a CA from issuing subordinate CA certificates
- Overwrite subject name properties to enforce organizational details
- Set maximum issued certificate lifetimes
- Require submission of CSR certificate request file to improve private key protection and proof of possession.
- Restrict subject alternative names to predefined list

Recommendation: Organizations should enforce a hard limit on the lifetime of all certificates from a CA. Certificate lifetimes are balanced with the cryptographic strength of the certificate key pair as well as possible certificate misuse. End-entity certificates default to a validity period of 30 days but should be set to the shortest possible lifetime while ensuring usability and reliability for each use case. Additionally, where possible, organizations should define and implement a CA policy for use throughout the organization to ensure details are properly defined for certificates.

Google Cloud

Roles and access controls

To protect the security and integrity of a CA, a separation of duties should be implemented to ensure that no one individual can affect the CA, request certificates, issue certificates and audit the system. Through the implementation of discrete roles, organizations can better manage the security of the CA and the overall PKI itself. Roles can be assigned at the project level or for specific CAs. Individual role account holders can be assigned to one of five predefined CAS roles by using IAM.

CA Service Operation Manager

This role can create, update, and delete CAs. They can also revoke certificates and create storage buckets. The role also includes the same abilities as the CA Service Auditor. This role should be assigned to role owners who will be responsible for configuring and deploying CAs in the organization.

CA Service Certificate Manager

This role can submit requests to a CA like the CA Service Certificate Requester, but also inherits the permissions of the CA Service Auditor. It should be assigned to individuals accountable for creating, tracking and reviewing certificate requests on a CA – such as a manager or lead engineer.

CA Service Admin

This admin role inherits permissions from the CA Service Operation Manager and CA Service Certificate Manager. This role can perform all actions within CAS and should be seldom assigned once the service is established. In this role, individuals can perform all aspects of administration including assigning rights to others and managing certificate requests in CAS. Special control and access to this account should be implemented to prevent unauthorized access or use.

CA Service Auditor

The Service Auditor has read-only access to all CAS resources and can retrieve and list properties of the CA, certificates, revocation lists, reusable configs, IAM policies and projects. This role should be assigned to individuals that are accountable for validating the proper security and operations of the CA and require no day-to-day responsibility in administering the service.

CA Service Certificate Requester

This role can submit certificate requests to a CA – either certificate request files or through the CAS User Interface. This role should be granted to trusted individuals who should be allowed to request certificates for end-entities in the organization.

> **Recommendation:** Individuals should not be assigned to more than one role at any time. In addition, everyone holding an assigned role should be properly briefed and trained on their responsibilities and security practices.

Google Cloud

Summary

Organizations can extend and expand their PKIs through the use of CAS and on-premises systems. As more services are moved to the cloud, the ability to securely issue and manage certificates expands beyond traditional enterprise boundaries. By carefully selecting the deployment architecture that fits their short and long term needs, organizations can greatly benefit by leveraging CAS to enable business requirements and improve operational efficiency.