CITI TECHNOLOGY OFFICE
FINANCIAL SERVICE
COMMON CLOUD CONTROLS (CCC) STANDARD

DISCUSSION DOCUMENT

CONTACT(S):

Jim Adams
jim.b.adams@citi.com
Jon Meadows
jonathan.meadows@citi.com
Moe Matar
moe.matar@citi.com
Jason Nelson
jason.nelson@citi.com

ISSUE DATE:

24th April 2023

VERSION: 1.0

Contents

1 BACKGROUND

Driven by changing technology requirements and a need to modernize existing infrastructure and applications, Financial Services are rapidly embracing Cloud-native capabilities to accelerate innovation and improve customer experience. Cloud Service Providers (CSP) offer solutions that differentiate from existing "on-premises" services, by providing the promise of improved operating resilience, hyper-scale and advanced, on-demand compute and data services. In addition, modern development practices accelerate "time-to-market" whilst also providing an environment that helps attract and retain engineering staff in an ever more competitive market.

The move to CSPs does, however, present significant challenges. The recent U.S. Department of Treasury report, *The Financial Services Sector's Adoption of Cloud Services*[1], outlines six thematic challenges:

- Insufficient Transparency to Support Due Diligence and Monitoring by Financial Institutions
- Gaps in Human Capital and Tools to Securely Deploy Cloud Services
- Exposure to Potential Operating Incidents, Including Those Originating at a CSP
- Potential Impact of Market Concentration in Cloud Service Offerings on the Sector's Resilience
- Dynamics in Contract Negotiations Given Market Concentration
- International Landscape and Regulatory Fragmentation

Each of these thematic challenges is described in detail in the report. Many are exacerbated given the variations in services and implementations provided across CSPs. For instance, the variation in technical approaches further increases the skills gap referenced in *Gaps in Human Capital and Tools to Securely Deploy Cloud Services,* as the skills are not necessarily portable across CSPs. Similarly, those same variations create complexity when attempting to shift workloads from one CSP to another, as reflected in *Potential Impact of Market Concentration in Cloud Service Offerings on the Sector's Resilience*. Likewise, the absence of a consistent definition of services and controls across CSPs results in the fragmentation and complexity referenced in *International Landscape and Regulatory Fragmentation*.

A key consideration as the Financial Services Sector leverages the services provided by CSPs is an understanding of the implications of the *Shared Responsibility* model. To meet the security, resilience, and operational needs necessary to operate an environment compliant with the existing risk management principles and practices necessary in the financial services sector[2], both the CSP and the financial services firm share the responsibilities. The CSP operate, manage, and control the physical security of the facilities and the management and operation of the systems and services it provides. The financial services firm maintains responsibility for the configuration of the services, hygiene, and

[1] See https://home.treasury.gov/system/files/136/Treasury-Cloud-Report.pdf
[2] OCC Bulletin 2020-46 Cybersecurity: Joint Statement on Security in a Cloud Computing Environment

security patching activities on its operating system and the development of the application software. It is the financial services firm's ultimate responsibility to ensure compliance of the complete solution with applicable laws and regulations.

Amazon Web Services (AWS) describes this segregation of responsibilities as AWS being responsible for the *Security of the Cloud* whereas the customer is responsible for the *Security in the Cloud*. The following diagram shows this succinctly:

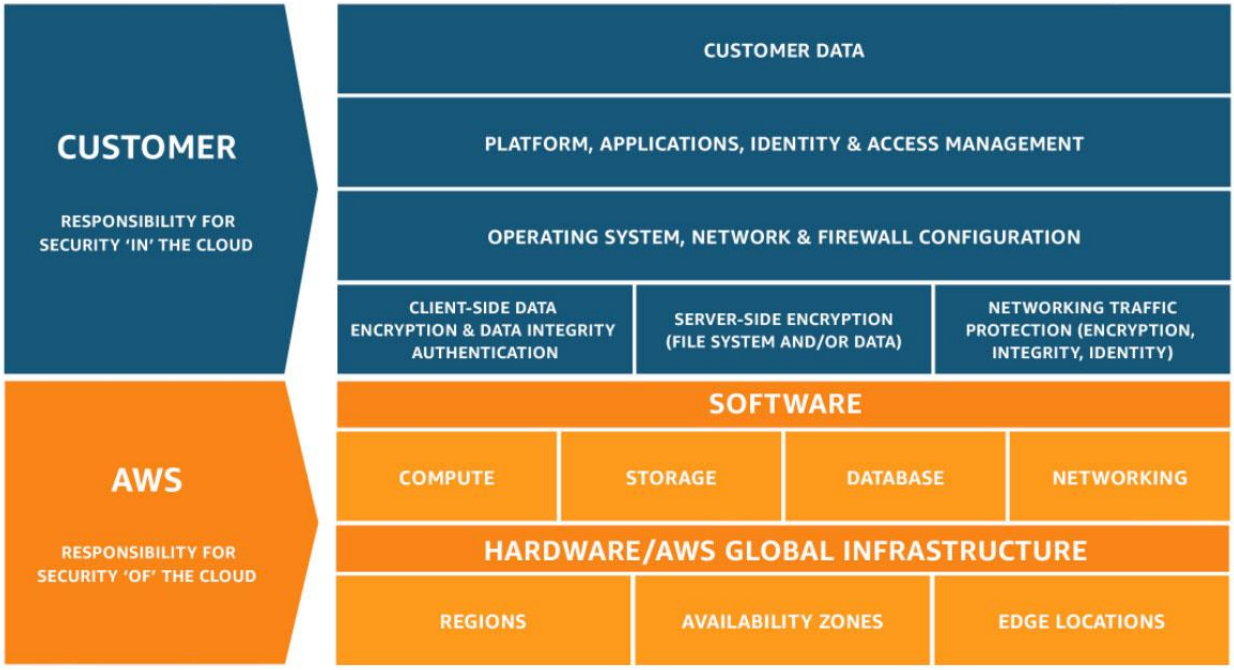


Figure 1: AWS Shared Responsibility Model[3]

The challenge this presents is that the majority of the CSPs, although providing highly secure cloud environments[4], only provide *access to* the necessary controls to create secure and compliant solutions. It is still the financial services firm's responsibility to determine *how* these controls are applied to the services they leverage, and *which* controls are needed to secure the service. And although some CSPs are looking to create dedicated Financial Services Solutions, this could reinforce the thematic concern *Potential Impact of Market Concentration in Cloud Service Offerings on the Sector's Resilience.* If firms operate within this environment, portability to other CSPs will be challenging to evidence like-for-like controls.

To address these challenges, it is Citi's belief that there is a need to establish *an industry minimum standard*. The standard would describe the consistent controls for the subset of

[3] See https://aws.amazon.com/compliance/shared-responsibility-model/
[4] See AWS Compliance and Security for Financial Services and Google Cloud Security Compliance

critical services that are common across CSPs that firms use when delivering compliant solutions within the existing risk management principles and practices necessary for the financial services sector. This standard must be agnostic to any specific regulatory body, but be able to evidence lineage to the same, ensuring it does not compound the thematic concern of *International Landscape and Regulatory Fragmentation*. It must be independent of any CSP specific implementation to ensure applicability and consistency across providers. It must be able to evolve as cyber adversaries develop new techniques and threats to the critical services covered, with transparency and traceability.

Such a minimum standard would provide a benchmark of controls for critical services that would transcend CSPs, allow evidence of controls consistency when migrating solutions from one CSP's service to another, and provide a level of assurance and compliance to regulatory requirements. Applying such a standard would significantly reduce the impediments to migrate applications between CSPs, ultimately reducing the concentration risk, although there will be a need for additional cybersecurity considerations outside this proposed standard. Establishing a minimum standard would also provide an opportunity for smaller financial services companies, that perhaps have less available investment, to secure their critical services at a CSP. It will provide a "level playing field" with those firms that can make the necessary investment in order to leverage public cloud solutions.

2 FINANCIAL SERVICES COMMON CLOUD CONTROLS STANDARD

In line with the shared responsibility model, financial services firms must ultimately bear the responsibility for defining the appropriate controls to manage the CSP services they are leveraging, be it *Infrastructure as a Service (IaaS)* or *Platform as a Service (PaaS)*[5]. CSPs often offer comparable services, such as relational databases or object storage, but in each case, the financial services firm must assess the necessary controls for the service on that CSP and then determine how best to implement those controls at their chosen CSP. Existing frameworks, such as the *Cloud Security Alliance Cloud Controls Matrix*[6], look to provide clarity on the controls objectives. This can be used as a tool for the systematic assessment of a cloud implementation providing guidance on which security controls should be implemented and by which accountable owner within the shared responsibility. However, the fidelity on the threats that these controls would mitigate at a point in time – effectively the *why* these are the appropriate controls for a specific cloud service – is less clear.

To mitigate this, it is proposed that a consistent *threat modelling practice* be followed to determine the necessary controls that a *generalized* version of a critical service would need to establish. For an identified *Service*, an initial understanding of the architecture of the service must be described as a logical *Service Data Flow Diagram*. This would be a generalized flow but should provide sufficient detail to understand any common attack vectors in the service design where *Threats Techniques* may require *Mitigations*. These mitigations are then addressed by the presence of one or more logically defined *Controls*.

As such, we propose the Financial Services Common Cloud Controls Standard to provide consistency of four related components:

1. **Cloud Services Taxonomy** – a consistent taxonomy for critical services provided by a specific CSPs to facilitate identification and classification of similar services across CSPs
2. **Service Specific Data Flow Diagram** – a high-level data flow of a generic service, providing sufficient details to understand common attack vectors in the service
3. **Threat Catalogue** – a consistent taxonomy of common threat techniques and associated mitigations that may occur across services exploiting potential weaknesses
4. **Logical Controls Description** – a logical control that when implemented would address one or more specific threats

The following diagram provides a representation of the relationship between these components:

[5] It could be applicable to *Software as a Service (SaaS)* when it is hosted directly on a CSP (as opposed to in a *community cloud*[5] where it would be the service owner's responsibility)

[6] See CSA Cloud Controls Matrix

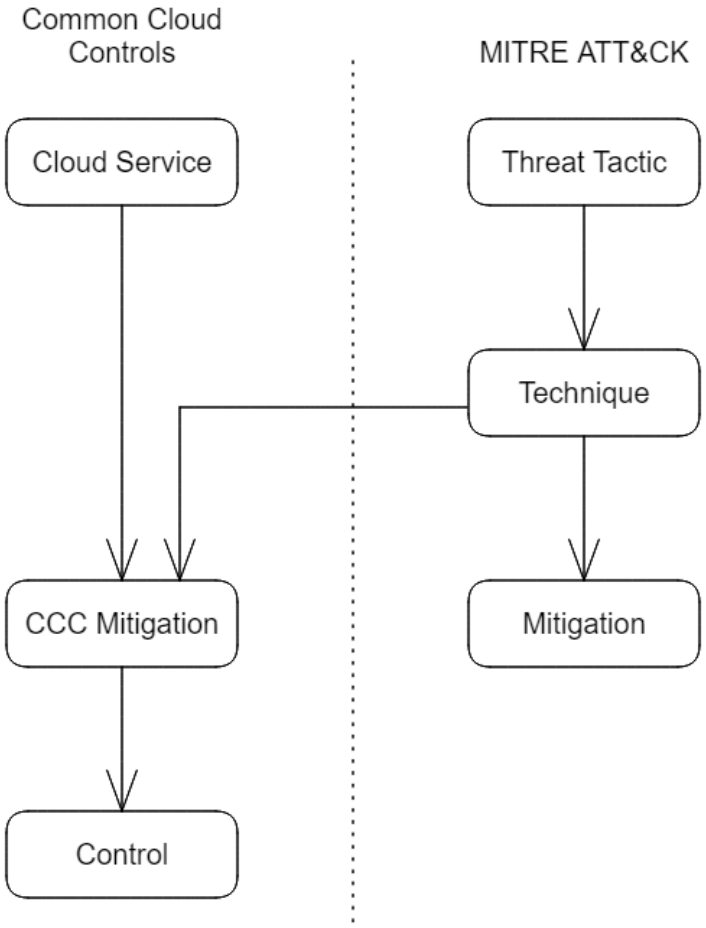


Figure 2- CCC Standard Components

For each Cloud Services in scope for CCC, the relevant Threat Techniques applicable to that Service are identified (above leveraging the MITRE ATT&CK® framework), for which one or more mitigations would be required, and these would be addressed by one or more controls specific to the service.

The authors recognize that implementation of *only* this set of logical controls would be insufficient to secure a service *implementation specific to a CSP,* as there may be other threats specific to a CSP's service that may need to be addressed. It does, however, describe the common threats that type of service would need to address and subsequently the associated controls that a service implementation must be able to provide, and this is expected to represent a significant number of the threats. Using an analogy of object orientated design, consider the CCC defined threats and controls as the *abstract class* for a type of cloud service, which then needs to be specialized and made *concrete* for a specific implementation of that service. This specialization would need to add the specific threats and controls applicable to that implementation of the service.

As each CSP categorizes its services consistently and demonstrates their service can meet the controls outlined in the CCC Standard specific to that service, the challenges in migrating between CSPs will be significantly reduced, as there would be assurances that the service

can meet the baseline of controls needed. Other considerations are necessary to provide resilient and secure solutions in the cloud outside of the remit of this minimum standard.

2.1 CLOUD SERVICES TAXONOMY

The first challenge that the CCC Standard looks to address to facilitate migration between CSPs, is a comprehensive understanding of the equivalent service offerings provided by each provider. Every CSP provides a multitude of products, each categorized according to its own classifications[7]. Some of these services are CSP specific – their differentiators – but many represent the foundational critical services that many solutions in the cloud require – compute, storage, networking, databases etc. Various resources do currently exist that provide a cursory comparison of the different services across providers. They tend to use simple categorization to group related services, as evidenced by resources such as the *Public Cloud Service Comparison* hosted by comparecloud.in and Google's comparison of its services to Azure and AWS[8].

Service category	Service type	Google Cloud product	Google Cloud product description	AWS offering	Azure offering
App modernization	CI/CD	Cloud Build	Build, test, and deploy on Google Cloud serverless CI/CD platform	AWS CodeBuild, AWS CodeDeploy, AWS CodePipeline	Azure DevOps, GitHub Enterprise
App modernization	CI/CD	Google Cloud Deploy	Deliver continuously to Google Kubernetes Engine and Anthos.	AWS CodeCommit, AWS CodeBuild, AWS CodeDeploy	Azure DevOps

Figure 3: Google comparison with Azure and AWS

To provide appropriate foundations for the CCC Standard, a curated taxonomy of critical services needs to be created *and* adopted by the various CSPs to classify their services consistently. Prior resources may assist in establishing this taxonomy, although a richer description may be required to support accurate classification, potentially including *qualifiers* that describe features of the service to enable accurate comparisons.

It should also be noted that it is only the common services across CSPs that the taxonomy would need to focus on. As previously stated, each CSP may provide other distinct and

[7] See Google Cloud Products, AWS Cloud Products, Azure Product Categories and IBM Cloud Catalog
[8] See Compare AWS and Azure services to Google Cloud

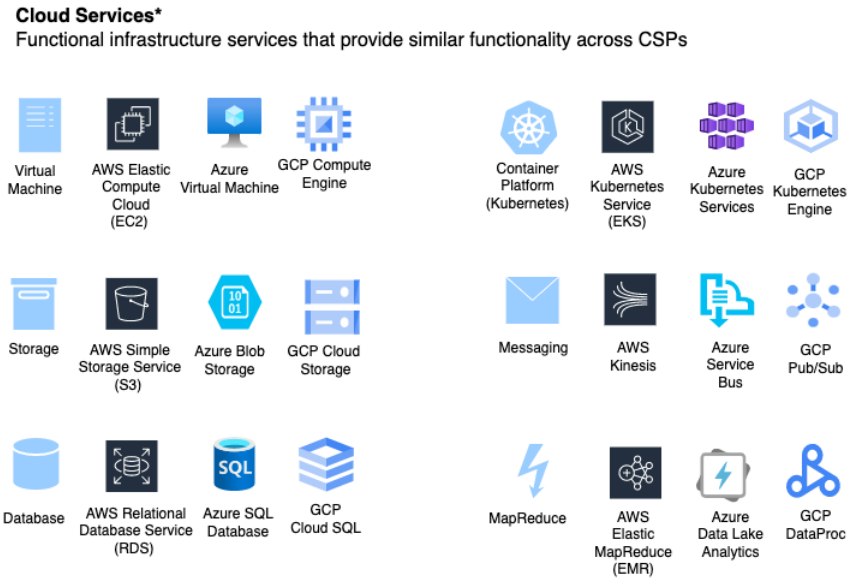
differentiated services specific to their environment, for which there may not be an equivalent service at other CSPs. These could be classified against the taxonomy, although there is little immediate benefit in doing so until more than one CSP offers the same type of service.

2.2 GENERALIZED SERVICE FLOW DIAGRAM

To create a threat model for a generalized cloud service, a *data-flow diagram* (DFD)[9] for the service is needed showing the different paths that information flows through the service and the critical points in the flow. The data-flow diagram will show how data logically moves through the service and allows the identification of critical components that the service may depend on.

Much like the cloud services themselves, today each CSP provides resources to support the development of architecture diagrams[10] that are aligned to its services. As can be seen in the sample legend below, although they cover similar concepts, there is variation in terminology and styles across CSPs. This necessitates a requirement to create CSP agnostic architecture icons to support the Generalized Service Flow Diagram, with a lineage to the CSP specific resources to support the detailed CSP specific architecture diagrams necessary for detailed design and implementation of the service.

It should also be noted that some CSPs provide dedicated diagramming tooling[11], or libraries to be added to architecture tooling, that may be useful to be extended or leveraged.



[9] See Data-flow diagram - Wikipedia for a description of Data-flow diagrams
[10] See AWS Architecture Icons
[11] See Introducing a Google Cloud architecture diagramming tool

2.3 THREAT CATALOGUE

The CCC Standard will provide foundational threat models for the services that it covers. To ensure consistency across services and traceability to the controls that mitigate the threat, a taxonomy of these threats and an associated description must be created. It should be noted that threat modelling is only one but a key component of establishing a comprehensive cyber security solution. Various cyber security frameworks exist to ensure organizations establish appropriate governance, risk mitigation and security procedures and the CCC Standard does not look to replace any of these. In fact, it will look to leverage existing knowledge bases.

2.3.1 MITRE ADVERSARIAL TACTICS, TECHNIQUES, AND COMMON KNOWLEDGE (ATT&CK®)

There are several cyber security frameworks that could be beneficial to leverage. The *MITRE Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK®)*[12] provides a curated knowledge base and model for cyber adversary behavior, providing a common taxonomy of individual adversary actions understood by both offensive and defensive sides of cybersecurity. This taxonomy is widely used in the Cyber Security industry to understand the threats and techniques being used to exploit systems. Whilst the original taxonomy was based upon identifying Windows attacks, subsequent taxonomies have been created to focus on specific domains, such as the cloud.

The ATT&CK® is open and available to any person or organization for use at no charge[13]. Specifically, the *ATT&CK® Matrix for Enterprise*[14] provides a rich classification of the tactics and techniques for *Windows*, *macOS*, *Linux*, *PRE*, *Azure AD*, *Office 365*, *Google Workspace*, *SaaS*, *IaaS*, *Network* and *Containers*. This taxonomy aids the threat modelling process by identifying known threat techniques and associated mitigations that have been used against cloud systems.

[12] See https://attack.mitre.org/
[13] See https://attack.mitre.org/resources/terms-of-use/
[14] See Matrix - Enterprise | MITRE ATT&CK®

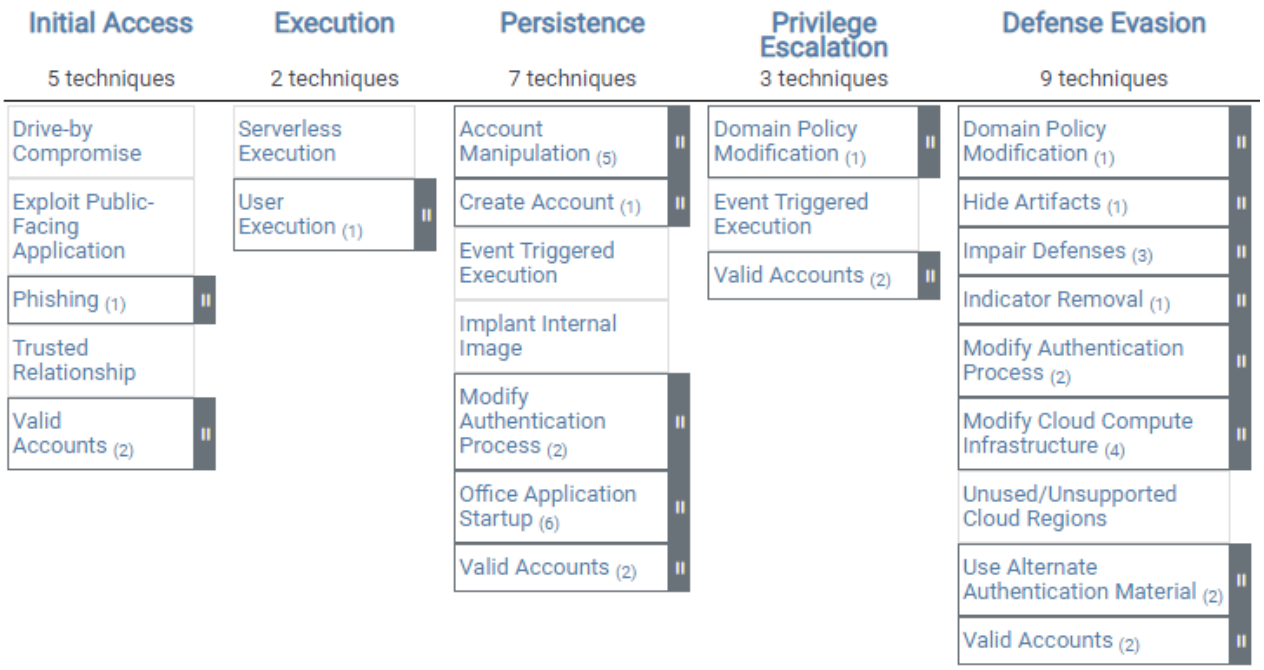


Figure 4 – An extract from the ATT&CK® Matrix for the Enterprise

It is proposed that a subset of the threat *techniques* and associated *mitigations* be used in the CCC Threat Models. As an example, one threat technique within the MITRE ATT&CK® framework within the *Collection* threat tactic, is *Data from Cloud Storage* (T1530)[15] which describes when adversaries may access data from improperly secured cloud storage.

The following Mitigations are described:

[15] See Data from Cloud Storage, Technique T1530 - Enterprise | MITRE ATT&CK®

Mitigations

ID	Mitigation	Description
M1047	Audit	Frequently check permissions on cloud storage to ensure proper permissions are set to deny open or unprivileged access to resources.[5]
M1041	Encrypt Sensitive Information	Encrypt data stored at rest in cloud storage.[5][6] Managed encryption keys can be rotated by most providers. At a minimum, ensure an incident response plan to storage breach includes rotating the keys and test for impact on client applications.[14]
M1037	Filter Network Traffic	Cloud service providers support IP-based restrictions when accessing cloud resources. Consider using IP allowlisting along with user account management to ensure that data access is restricted not only to valid users but only from expected IP ranges to mitigate the use of stolen credentials to access data.
M1032	Multi-factor Authentication	Consider using multi-factor authentication to restrict access to resources and cloud storage APIs.[5]
M1022	Restrict File and Directory Permissions	Use access control lists on storage systems and objects.
M1018	User Account Management	Configure user permissions groups and roles for access to cloud storage.[6] Implement strict Identity and Access Management (IAM) controls to prevent access to storage solutions except for the applications, users, and services that require access.[5] Ensure that temporary access tokens are issued rather than permanent credentials, especially when access is being granted to entities outside of the internal security boundary.[15]

Figure 5- Data from Cloud Storage Mitigations

Expanding upon this approach the CCC would provide service specific mitigations to the threats identified in the threat model. These mitigations would then be further elaborated with one or more associated *Logical Controls*, maintaining referential integrity. It may be that during the development of the CCC standard, the MITRE ATT&CK® framework needs to be expanded to include other mitigations.

2.3.2 OTHER RESOURCES TO BE CONSIDERED

Another resource that may be useful to leverage, although not specific to Public Cloud, is the *NIST Mobile Threat Catalogue*[16]. It provides threats specific to mobile applications in a rich and detailed catalogue, including details of the threat, its origin, exploit examples and potential counter measures. This may provide a foundation for the CCC Threat Catalogue.

[16] See NIST Mobile Threat Catalogue and

The *Cloud Security Alliance (CSA)* provides a report of the *Top Threats*[17] applicable to the creation and operation of public cloud solutions. This will be beneficial when considering the catalogue of threats.

As previously stated, the CCC Standard does not look to replace any of these cybersecurity frameworks and resources in any way. As such, the CCC Standard should leverage these resources and potentially influence them to focus on the public cloud domain.

2.4 Logical Controls and Controls Taxonomy

Once threat techniques to specific cloud services have been identified along with the associated mitigations, it is necessary to describe the logical control or set of controls that will provide the mitigation. The control definition must be detailed enough to ensure that the control information can be used by the CSPs to identify the appropriate control *implementation* and ultimately allow the *assessment* of the control efficacy.

2.4.1 Open Security Controls Assessment Language (OSCAL)

The *Open Security Controls Assessment Language (OSCAL)*[18] is a machine-readable data format used to define a control policy. This is a NIST standard that is maturing with controls now available to define the NIST 800-153 cloud standard. However, until recently, this has predominantly been the definition of controls and policies without definition of the test case or implementation.

Whilst the ability to provide a machine-readable definition is useful to transfer between control points and users, the real benefit comes from being able to define a test requirement that can be understood as a clear machine executable way to validate the control has been satisfied. It is proposed that as part of CCC Standard that OSCAL be leveraged to provide the controls definition, albeit with an extension to describe how the control could be validated. This will require an update to be made to the OSCAL standard to allow codified controls to be added to the component definition and test cases to be represented in the assessment data. These changes will allow OSCAL to be used not only to represent the control language, but also to supply a capability that would validate the efficacy of the control implementation.

An example extension is presented below, leveraging two proposed elements to the OSCAL standard, namely “rules” and “validate” in the control-implementation. Here, the validate element defines a codified example of how the controls can be validated. The exact implementation of that control is left to be implemented by the CSP for that specific service. However, the text details sufficient data to demonstrate how the control can be tested to prove its efficacy. This may require the implementation of a DSL to detail the test. In this case, we have leveraged a Gherkin[19] format.

---

[17] See Cloud Security Alliance Top Threats to Cloud Computing
[18] See NIST OSCAL
[19] See Gherkin Reference - Cucumber Documentation

```
- id: M1032_asm-validate.CCC.C2
    name: validate
    props:
      - name: method
        value: (CCC.C2)
        class: preventative
  prose: "GIVEN cloud service WHEN principle attempts authentication to the service with
local credential THEN service should deny action for all principles"
```

3 EXAMPLE OF FINANCIAL SERVICE CLOUD STANDARD – RELATIONAL DATABASE SERVICE

This section provides an example of how the CCC standard could be applied to a single common service across CSPs. It should not be considered definitive but merely representative of how the standard could be applied. Details will be provided on how each component of the standard would be populated. The example will then be further extended to show how to leverage and further extend for a specific instance of a service at a CSP.

3.1 PUBLIC CLOUD SERVICES TAXONOMY

The service would need to align with the CCC Public Cloud Service Taxonomy, which may be represented as follows:

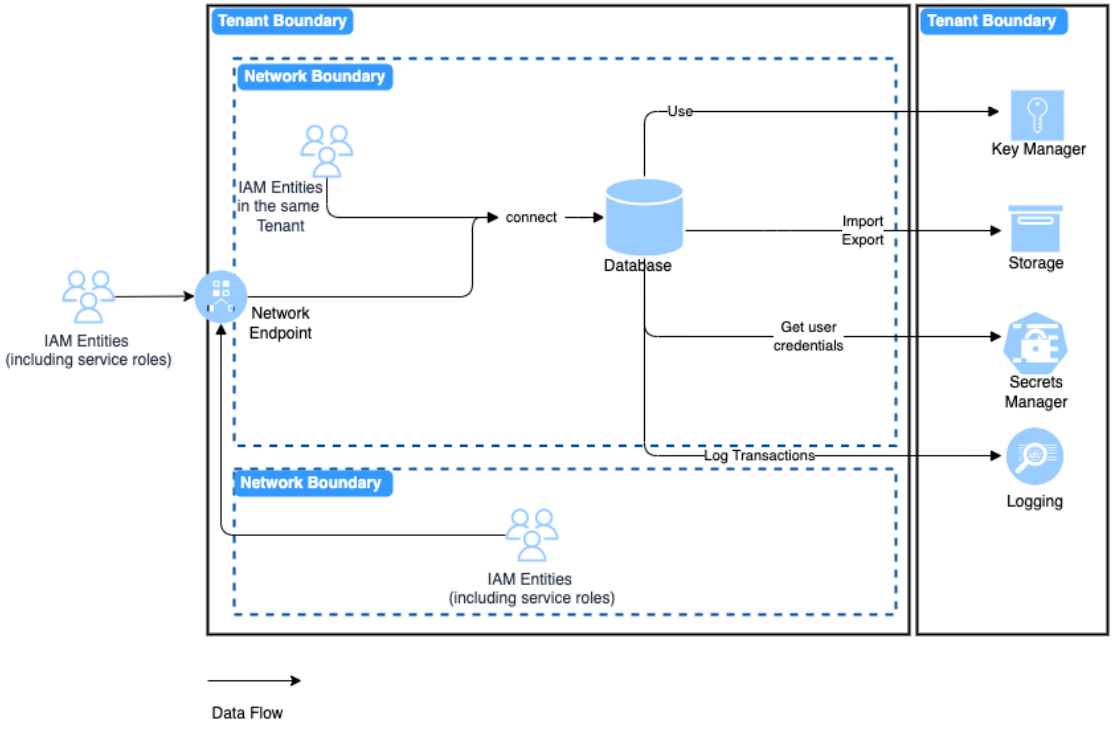
```
Data-At-Rest->Database->Relational Database
```

Other services within the Database taxonomy could be Document Database, Graph Database, Time-series Database, NoSQL Database, and In-memory Database. In addition, qualifiers may be needed, such as confirming the transaction boundaries and replication capabilities.

3.2 GENERIC SERVICE DATA FLOW DIAGRAM – RELATIONAL DATABASE

The following high-level data flow diagram shows the appropriate flows for a generic cloud relational database service. Subsequent diagrams show the associated threats and mitigations for that service specific architecture.

Figure 6 - Generic Data-flow for Relational Database



3.3 THREAT CATALOGUE FOR RELATIONAL DATABASE

For each service, specific tactics and techniques can be documented leveraging the MITRE ATT&CK® taxonomy. The following diagram highlights a subset of tactics that a Relational Database must defend against and maps to the list of threats identified below the diagram.

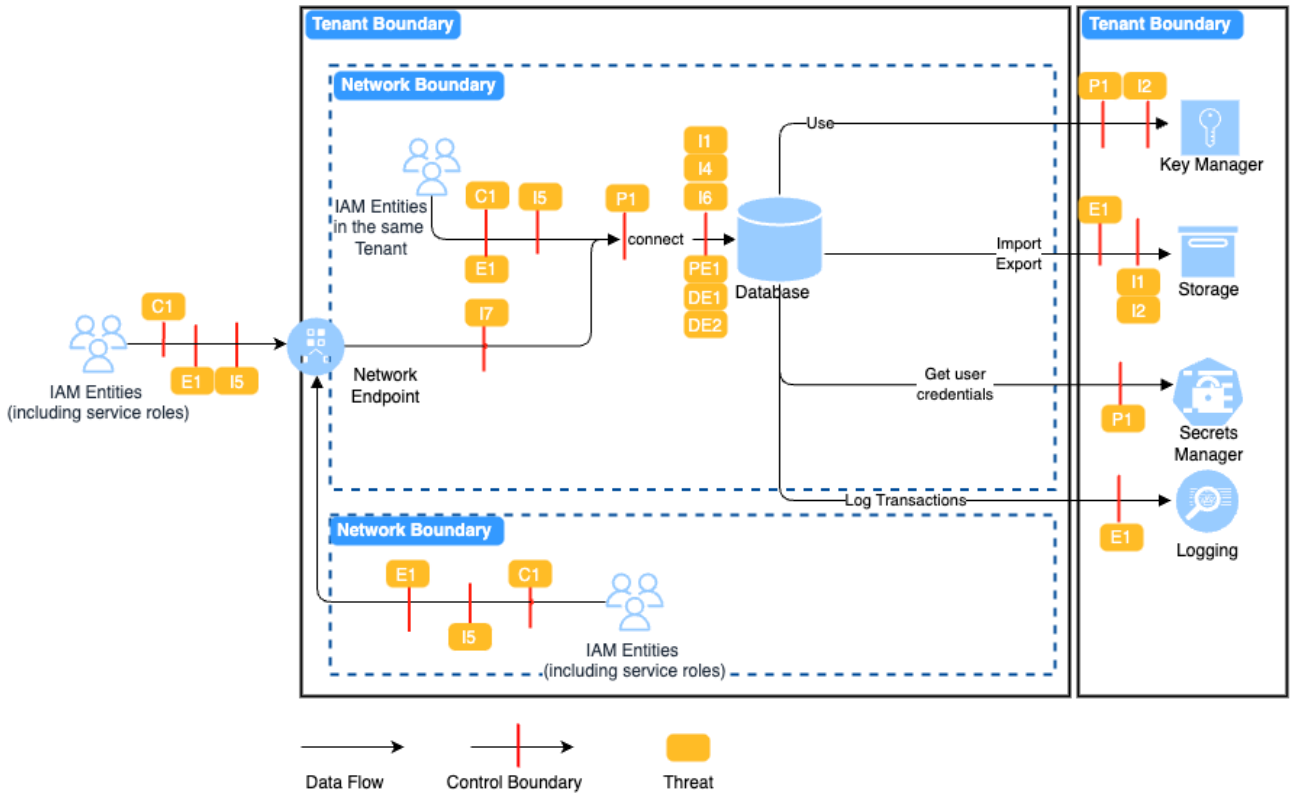


Figure 7 – Threats represented in Generic Data-low for Relational Databases

The table below shows details on the threat Tactic and relevant Techniques that applies to the service, referenced from MITRE ATT&CK® framework:

Collection		
C1	T1530: Data from Cloud storage	Adversaries may access data from improperly secured cloud storage. Many cloud service providers offer solutions for online data object storage such as Amazon S3, Azure Storage, and Google Cloud Storage. These solutions differ from other storage solutions (such as Relational Databases) in that there is no overarching application. Data from these services can be retrieved directly using the cloud provider's APIs.

Persistence		
P1	T1078.001: Valid Accounts: Default Account	Adversaries may obtain and abuse credentials of a default account as a means of gaining Initial Access, Persistence, Privilege Escalation, or Defense Evasion. Default accounts are those that are built-into an OS, such as the Guest or Administrator accounts on Windows systems.
	T1078.002: Domain Accounts	Adversaries may obtain and abuse credentials of a domain account as a means of gaining Initial Access, Persistence, Privilege Escalation, or Defense Evasion. Domain accounts are those managed by Active Directory Domain Services where access and permissions are configured across systems and services that are part of that domain. Domain accounts can cover users, administrators, and services.
	T1078.003: Local Accounts	Adversaries may obtain and abuse credentials of a local account as a means of gaining Initial Access, Persistence, Privilege Escalation, or Defense Evasion. Local accounts are those configured by an organization for use by users, remote support, services, or for administration on a single system or service.
	T1078.004: Cloud Support Accounts	Adversaries may obtain and abuse credentials of a cloud account as a means of gaining Initial Access, Persistence, Privilege Escalation, or Defense Evasion. Cloud accounts are those created and configured by an organization for use by users, remote support, services, or for administration of resources within a cloud service provider or SaaS application. In some cases, cloud accounts may be federated with traditional identity management system, such as Window Active Directory.
	T1098: Additional Cloud Roles	An adversary may add additional roles or permissions to an adversary-controlled cloud account to maintain persistent access to a tenant.

Privilege Escalation		
PE1	T1078.001: Valid Accounts: Default Account	Adversaries may obtain and abuse credentials of a default account as a means of gaining Initial Access, Persistence, Privilege Escalation, or Defense Evasion. Default accounts are those that are built-into an OS, such as the Guest or Administrator accounts on Windows systems.
	T1078.002: Domain Accounts	Adversaries may obtain and abuse credentials of a domain account as a means of gaining Initial Access, Persistence, Privilege Escalation, or Defense Evasion. Domain accounts are those managed by Active Directory Domain Services where access and permissions are configured across systems and services that are part of that domain. Domain accounts can cover users, administrators, and services.
	T1078.003: Local Accounts	Adversaries may obtain and abuse credentials of a local account as a means of gaining Initial Access, Persistence, Privilege

Privilege Escalation		
		Escalation, or Defense Evasion. Local accounts are those configured by an organization for use by users, remote support, services, or for administration on a single system or service.
	T1078.004: Cloud Support Accounts	Adversaries may obtain and abuse credentials of a cloud account as a means of gaining Initial Access, Persistence, Privilege Escalation, or Defense Evasion. Cloud accounts are those created and configured by an organization for use by users, remote support, services, or for administration of resources within a cloud service provider or SaaS application. In some cases, cloud accounts may be federated with traditional identity management system, such as Window Active Directory.

Defense Evasion		
DE1	T1578.001 Create Snapshot	An adversary may create a snapshot or data backup within a cloud account to evade defenses. A snapshot is a point-in-time copy of an existing cloud compute component such as a virtual machine (VM), virtual hard drive, or volume. An adversary may leverage permissions to create a snapshot to bypass restrictions that prevent access to existing compute service infrastructure
	T1578.002 Create Cloud Instance	An adversary may create a new instance or virtual machine (VM) within the compute service of a cloud account to evade defenses. Creating a new instance may allow an adversary to bypass firewall rules and permissions that exist on instances currently residing within an account. An adversary may Create Snapshot of one or more volumes in an account, create a new instance, mount the snapshots, and then apply a less restrictive security policy to collect Data from Local System or for Remote Data Staging.
DE2	T1550.001: Use Alternate Authentication Material	Adversaries may use stolen application access tokens to bypass the typical authentication process and access restricted accounts, information, or services on remote systems. These tokens are typically stolen from users or services and used in lieu of login credentials. Application access tokens are used to make authorized API requests on behalf of a user or service and are commonly used to access resources in cloud and container-based applications and software-as-a-service (SaaS).

Exfiltration		
E1	T1537: Transfer Data to Cloud Account	Adversaries may exfiltrate data by transferring the data, including backups of cloud environments, to another cloud account they control on the same service to avoid typical file transfers/downloads and network-based exfiltration detection.

Impact		
I1	T1485: Data Destruction: Deletion of data, database, or snapshot	Adversaries may destroy data and files on specific systems or in large numbers on a network to interrupt availability to systems, services, and network resources. Data destruction is likely to render stored data irrecoverable by forensic techniques through overwriting files or data on local and remote drives. Common operating system file deletion commands such as del and rm often only remove pointers to files without wiping the contents of the files themselves, making the files recoverable by proper forensic methodology. This behavior is distinct from Disk Content Wipe and Disk Structure Wipe because individual files are destroyed rather than sections of a storage disk or the disk's logical structure. Adversaries may attempt to overwrite files and directories with randomly generated data to make it irrecoverable. In some cases, politically oriented image files have been used to overwrite data.
I2	T1486: Data Encrypted for Impact> Encryption with adversary key	Adversaries may encrypt data on target systems or on large numbers of systems in a network to interrupt availability to system and network resources. They can attempt to render stored data inaccessible by encrypting files or data on local and remote drives and withholding access to a decryption key. This may be done to extract monetary compensation from a victim in exchange for decryption or a decryption key (ransomware) or to render data permanently inaccessible in cases where the key is not saved or transmitted.
I3	T1490: Inhibit System Recovery: Deletion of snapshot, modify snapshot retention	Adversaries may delete or remove built-in operating system data and turn off services designed to aid in the recovery of a corrupted system to prevent recovery. This may deny access to available backups and recovery options.
I4	T1489: Service Stop: Unauthorised shutdown or restart database	Adversaries may stop or disable services on a system to render those services unavailable to legitimate users. Stopping critical services or processes can inhibit or stop response to an incident or aid in the adversary's overall objectives to cause damage to the environment
I5	T1531: Account Access Removal: Modify account or service access. Modify network access	Adversaries may interrupt availability of system and network resources by inhibiting access to accounts utilized by legitimate users. Accounts may be deleted, locked, or manipulated (ex: changed credentials) to remove access to accounts. Adversaries may also subsequently log off and/or perform a System Shutdown/Reboot to set malicious changes into place.
I6	T1565: Data Manipulation: Add or modify data in database	Adversaries may insert, delete, or manipulate data to influence external outcomes or hide activity, thus threatening the integrity of the data. By manipulating data, adversaries may attempt to affect a business process, organizational understanding, or decision making.

3.4 THREATS TO CCC MITIGATIONS

From the techniques specified in the MITRE ATT&CK® framework above we can drill down into the mitigations for those threats specific to the service. A Mitigation is a set of controls that combine to mitigate the identified threat partially or fully.

The table below shows the Mitigations required to address the modelled threats. Where the mitigation exists within the MITRE ATT&CK® framework it is referenced (Mxxxx). If this is a CCC service specific mitigation, then it is identified (CCC.xx):

Threat	Mitigation ID	Mitigation	Description
P1	M1032	Multi-factor Authentication	Use two or more pieces of evidence to authenticate to a system, such as username and password in addition to a token from a physical smart card or token generator.
	M1026	Privileged Account Management	Manage the creation, modification, use, and permissions associated to privileged accounts.
	M1018	User Account Management	Manage the creation, modification, use, and permissions associated to non-privileged user accounts
	CCC.M1	Organization level Authorization Origin Policy	Define actions that are allowed for cloud accounts subscribed to an organization. Ensure policy set to enforce MFA for console and API actions for IAM principles.

3.5 MITIGATIONS MAPPED TO LOGICAL CONTROLS

Taking *M1032: Multi-factor Authentication* as an example mitigation, one can then define the set of Logical Controls required to achieve this mitigation.

P1 > Mitigation M1032 : Multi-factor Authentication	
Control Examples	Control Type
CCC.C1: Platform must enforce authentication with MFA	Preventative
CCC.C2: Local Authentication must be disabled or restricted	Preventative
CCC.C3: Monitor API action that disables MFA	Detective
CCC.C4: Monitor for database actions related to local authentication	Detective

3.6 LOGICAL CONTROLS TO EXTENDED OSCAL

The controls defined above can then be represented using the NIST OSCAL language for control definition, as the example below suggests. The example below proposes a modification OSCAL to introduce a set of validate statements, written in Gherkin language, to offer a human-readable set of tests that can be performed to validate these controls.

```
controls:
  - id: M1032
    class: MITRE Att&ck
    title: Multi-Factor Authentication
    params:
      props:
        - name: label
          value: M1032
          class: MITRE Att&ck
        - name: sort-id
          value: M1032
    links:
      - href: "#M1032"
        rel: required
      - href: "#M1026"
        rel: related
      - href: "#M1018"
        rel: related
      - href: "#CCC.M1"
        rel: related
    parts:
      - id: M1032_smt
        name: statement
        prose: "Multi Factor Authentication"
        parts:
          - id: M1032_smt.CCC.C1
            name: item
            props:
              - name: label
                value: (CCC.C1)
            prose: "Platform must enforce authentication with MFA."
          - id: M1032_smt.CCC.C2
            name: item
            props:
              - name: label
                value: (CCC.C2)
            prose: " Local authentication must be disabled or restricted."
          - id: M1032_smt.CCC.C3
            name: item
            props:
              - name: label
                value: (CCC.C3)
            prose: "Monitor API action that disables MFA."
          - id: M1032_smt.CCC.C4
            name: item
            props:
              - name: label
                value: (CCC.C4)
            prose: "Monitor for database actions to related to local authentication."
      - id: M1032_gdn
        name: guidance
        prose: "Use two or more pieces of evidence to authenticate to a system; such as username and
password in addition to a token from a physical smart card or token generator."
      - id: M1032_obj
        name: assessment-objective
        props:
          - name: label
            value: M1032
            class: CCC
        parts:
```

```
          - id: M1032_obj.CCC.C1
            name: assessment-objective
            class: preventative
            props:
              - name: label
                value: (CCC.C1)
            prose: "Use two or more pieces of evidence to authenticate to a system; such as username
and password in addition to a token from a physical smart card or token generator."
          - id: M1032_obj.CCC.C2
            name: assessment-objective
            class: preventative
            props:
              - name: label
                value: (CCC.C2)
            prose: "Local Authentication must be disabled and only allow multi-factor
authentication."
          - id: M1032_obj.CCC.C3
            name: assessment-objective
            class: detective
            props:
              - name: label
                value: (CCC.C3)
            prose: "Attempt to disable multi-factor authentication are recorded, monitored and
alerted on."
          - id: M1032_obj.CCC.C4
            name: assessment-objective
            class: detective
            props:
              - name: label
                value: (CCC.C4)
            prose: "Successful attempt to use local authentication to access a resource should be
monitored and alerted on."
      - id: M1032_asm-examine
        name: assessment-method
        props:
          - name: method
            ns: ... #namespace to CCC
            value: EXAMINE
          - name: label
            value: M1032-Examine
            class: CCC
        parts:
          - name: assessment-objects
            prose: >-
              CSP Identification and authentication policy
              Cloud Service Authentication policy
              Procedures addressing authenticator management
              System security plan
              System design documentation
              System configuration settings and associated documentation
              Other relevant documents or records
      - id: M1032_asm-validate
        name: assessment-method
        props:
          - name: method
            ns: ... #namespace to FSC2
            value: Validate
          - name: label
            value: M1032-Validate
            class: CCC
        parts:
          - name: assessment-objects
            prose: Mechanisms supporting and/or implementing password-based
              authenticator management capability

      - id: M1032_asm-validate
        name: assessment-method
        props:
          - name: method
            value: Validate
            ns: ... #namespace to CCC
```

```
      class: CCC
    - name: label
      value: M1032-Validate
      class: CCC
  parts:
    - id: M1032_asm-validate.CCC.C1
      name: validate
      props:
        - name: method
          value: (CCC.C1)
          class: preventative
      prose: "GIVEN Cloud Platform WHEN principle attempts authentication to perform API action
              THEN platform must enforce multi-factor authentication for each user"
    - id: M1032_asm-validate.CCC.C2
      name: validate
      props:
        - name: method
          value: (CCC.C2)
          class: preventative
      prose: "GIVEN cloud service WHEN principle attempts authentication to the service with
             local credential THEN service should deny action for all principles"
    - id: M1032_asm-validate.CCC.C3
      name: validate
      props:
        - name: method
          value: (CCC.C3)
          class: detective
      prose: "GIVEN MFA is enabled on cloud service WHEN authentication attempt is successful.
              and MFA age is '0' or 'null' and MFA authentication token present=false THEN
              record the action and alert"
    - id: M1032_asm-validate.CCC.C4
      name: validate
      props:
        - name: method
          value: (CCC.C4)
          class: detective
      prose: "GIVEN cloud service WHEN principle attempts authentication to service with
             local credential THEN record the action and alert"
```

4 EXPECTATIONS OF A CSP TO ADHERE TO THE CCC STANDARD

For the CCC Standard to be successful, commitments would be required by each Cloud Service Provider so their services can be certified as meeting the standard:

1. **Classification of relevant CSP services to Cloud Services Taxonomy** – the identification of which services align to the Cloud Service Taxonomy is a foundational step in establishing the CCC Standard. It allows the CSP to associate their service with the appropriate CCC Threat Model.

2. **Identify the controls that may be applied to a service to address the Threats** – for classified services, the CSP must describe the appropriate controls that *could be* applied to mitigate the identified threats for that service. It must be noted, that in line with the shared responsibility model, it would still ultimately be the consumer of the services responsibility to ensure those controls are applied and maintained correctly and are necessary for the solution they are delivering.

3. **Reevaluate the necessary controls when their services are enhanced** – a CSP may introduce new capabilities to a *service*, or fundamentally change its architecture, which may inadvertently create new attack vectors or invalidate the existing Threat Model and associated Mitigations. To guard against this, CSPs must revalidate the efficacy of CCC controls as part of the service release process. Major enhancements or changes to a service may be introduced with feature-toggles, allowing users of the service to *opt-in* to enable a feature, once a new Threat Model has been produced, and additional mitigations are defined and applied.

By adhering to the above conditions, each CSP can evidence compliance to the CCC Standard for each supported service by providing:

1. An *executable Infrastructure-as-Code package* that deploys an instance of the service, along with its codified controls which implement the Mitigations defined in the CCC Standard.
2. An *executable package which performs the set of tests* outlined for each Control in its OSCAL definition.

3. *Documentation* that describes the contents of each of the aforementioned packages, and any execution instructions, including inputs and outputs expected, aligned to a specific version of their service.

5 CONSIDERATIONS FOR CSP CERTIFICATION TO THE CCC STANDARD

To ensure the integrity of adherence to the standard, an independent, reputable certifying authority must exist. This authority should be responsible for certifying a CSP's adherence to the CCC standard, for each supported service. It could also certify the use of these services by a CSP client (i.e., a Financial Institution) to ensure CCC mitigations have been applied correctly.

Certification should be time-based, requiring regular recertification, and should be performed against a specific version of a service Threat Model as defined in CCC, for a specific version of a CSP's service.