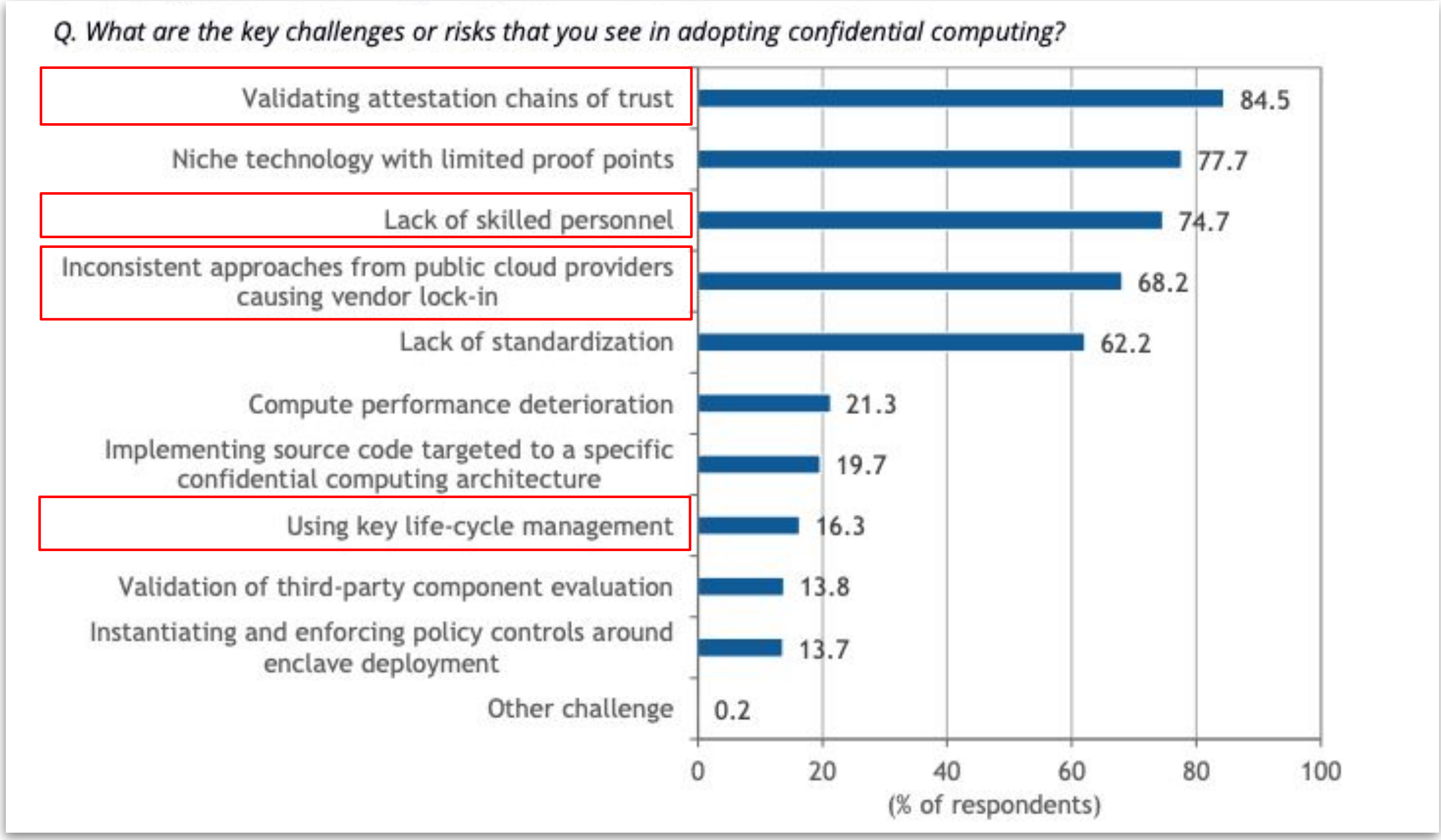
Trustee

A general purpose attestation verifier
and relying party services broker

Attestation is a Hurdle



Q. What are the key challenges or risks that you see in adopting confidential computing?

Trustee - Unified Attestation

Trustee..

- Provides a **common attestation abstraction** and protocol
- Collaborates with CCC Projects **Veraison** and **Virtee**
- Supports **vendor verifiers** and even non-CC usages
- May fulfill CCC/WIMSE **workload identity** goals
- Customer-driven development

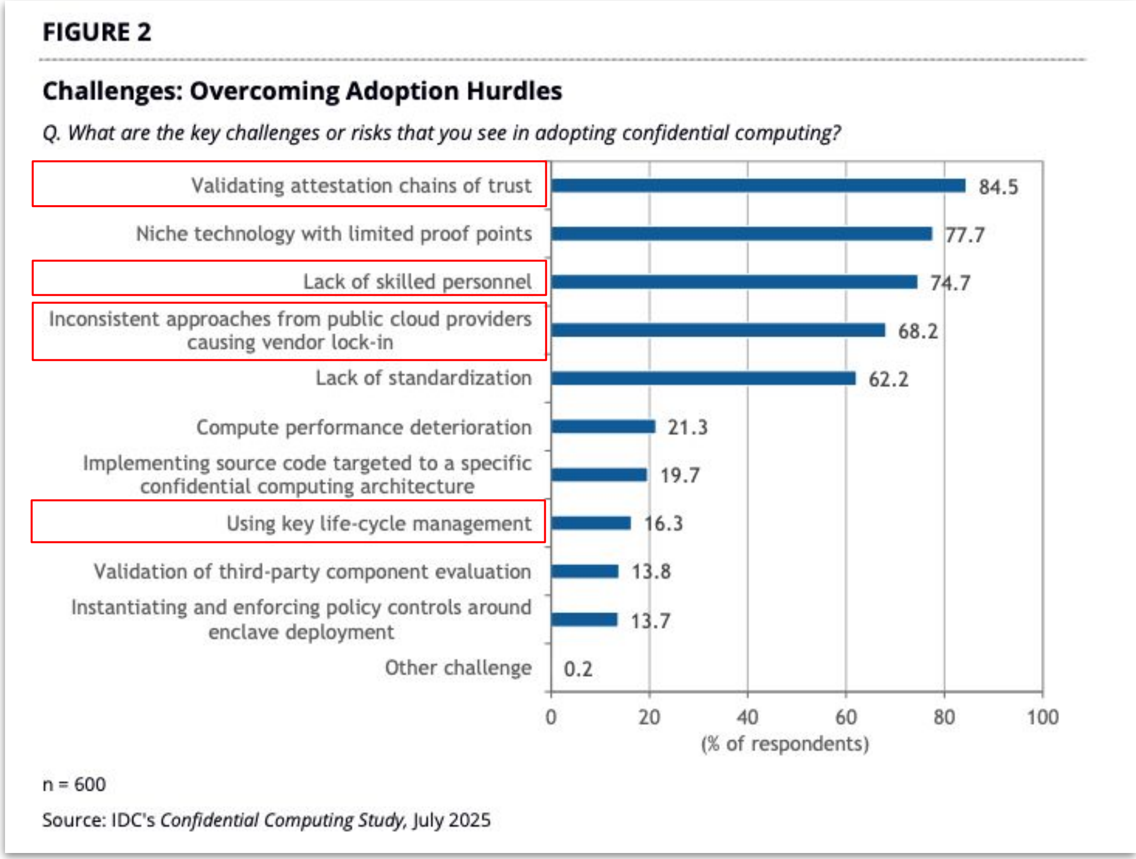


FIGURE 2

Challenges: Overcoming Adoption Hurdles

Q. What are the key challenges or risks that you see in adopting confidential computing?

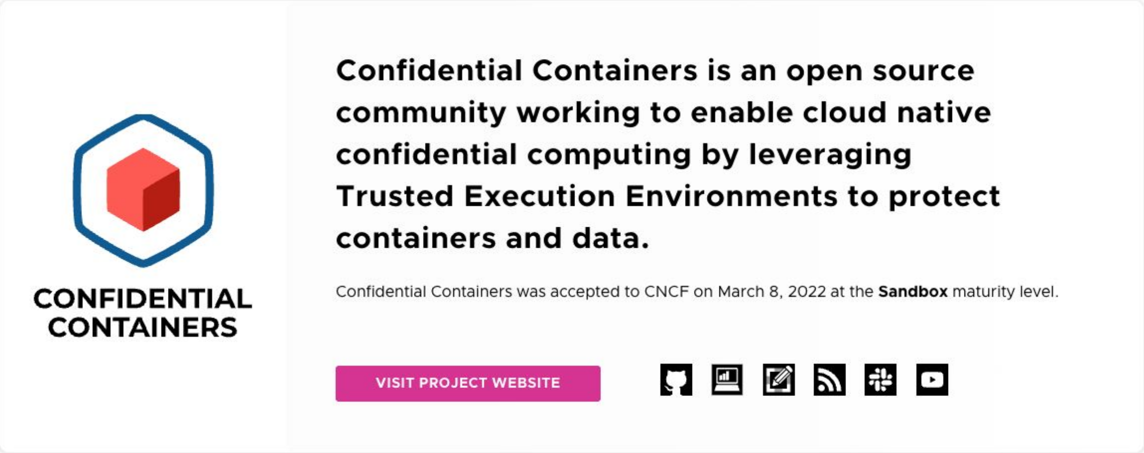
n = 600

Source: IDC's *Confidential Computing Study*, July 2025

Customer driven development

Contributors and Adopters:

- Alibaba Cloud
- Edgeless Systems
- IBM
- NanhuLab
- KubeArmor
- Red Hat
- ByteDance
- Intel
- Switchboard
- JDCloud
- Kubermatic
- NVIDIA
- Azure

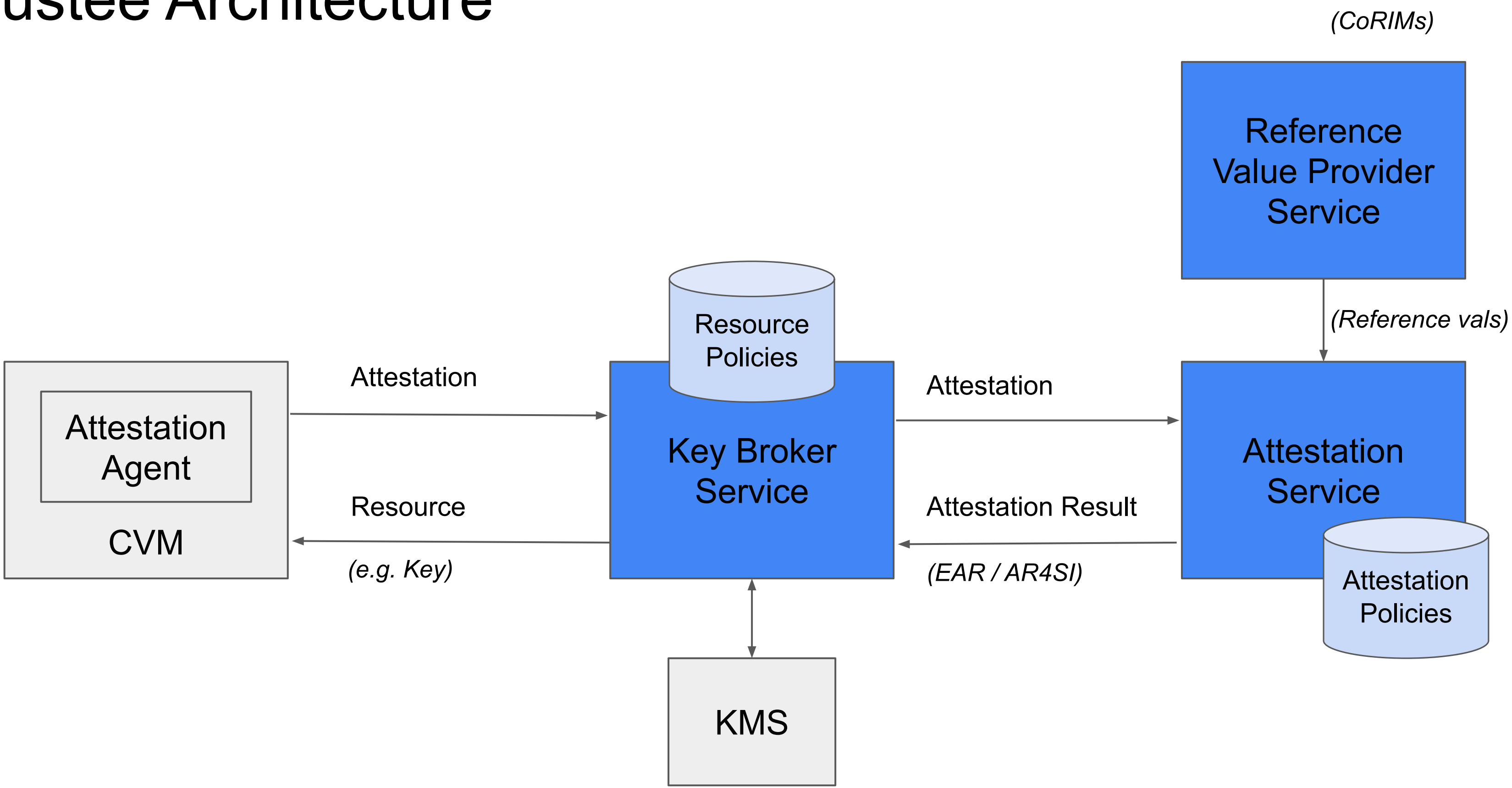


Confidential Containers is an open source community working to enable cloud native confidential computing by leveraging Trusted Execution Environments to protect containers and data.

Confidential Containers was accepted to CNCF on March 8, 2022 at the **Sandbox** maturity level.

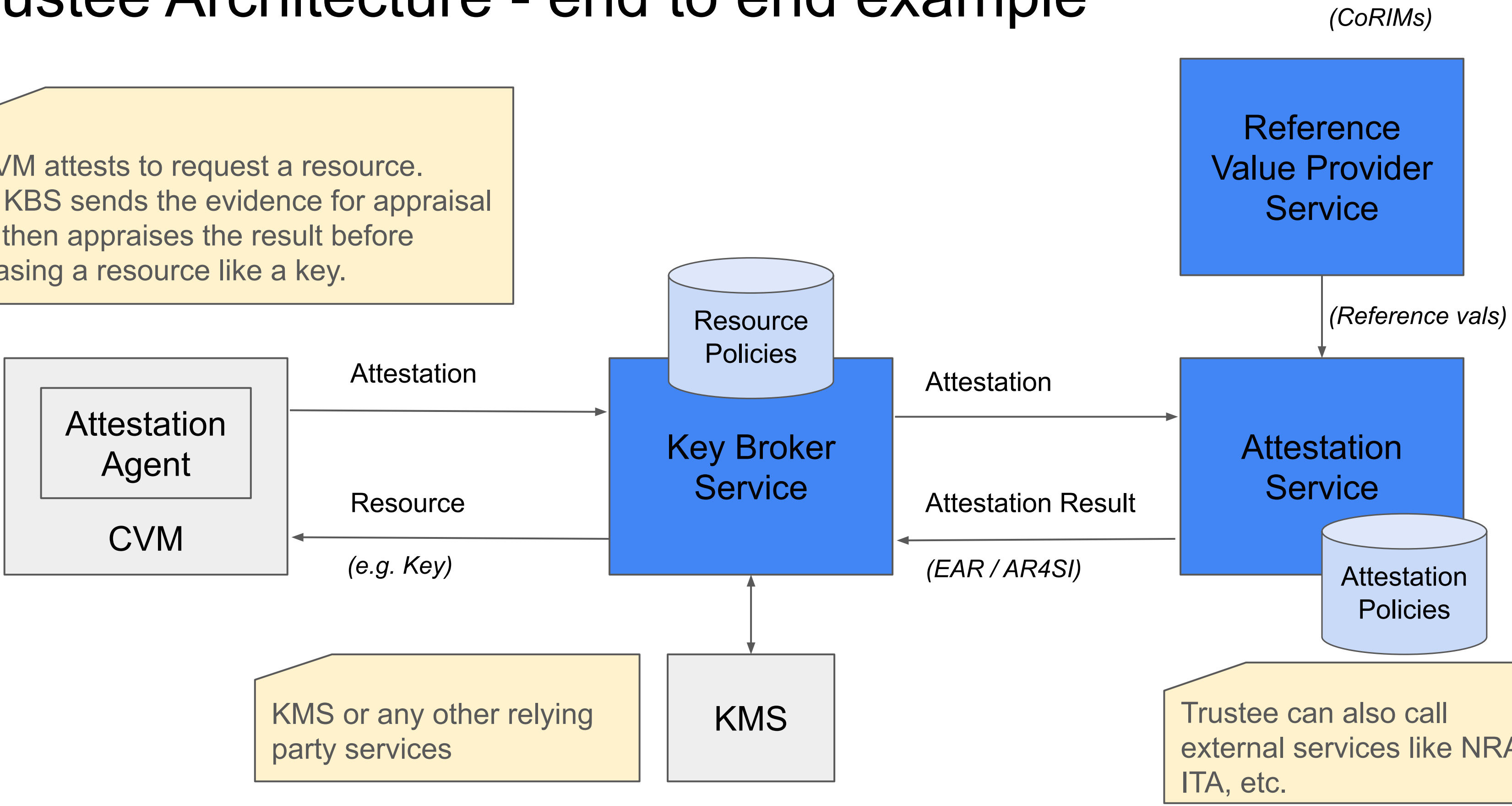
Launched in 2021, CNCF **Sandbox** Stage since 2022
Progressing to **Incubation** soon

Trustee Architecture



Trustee Architecture - end to end example

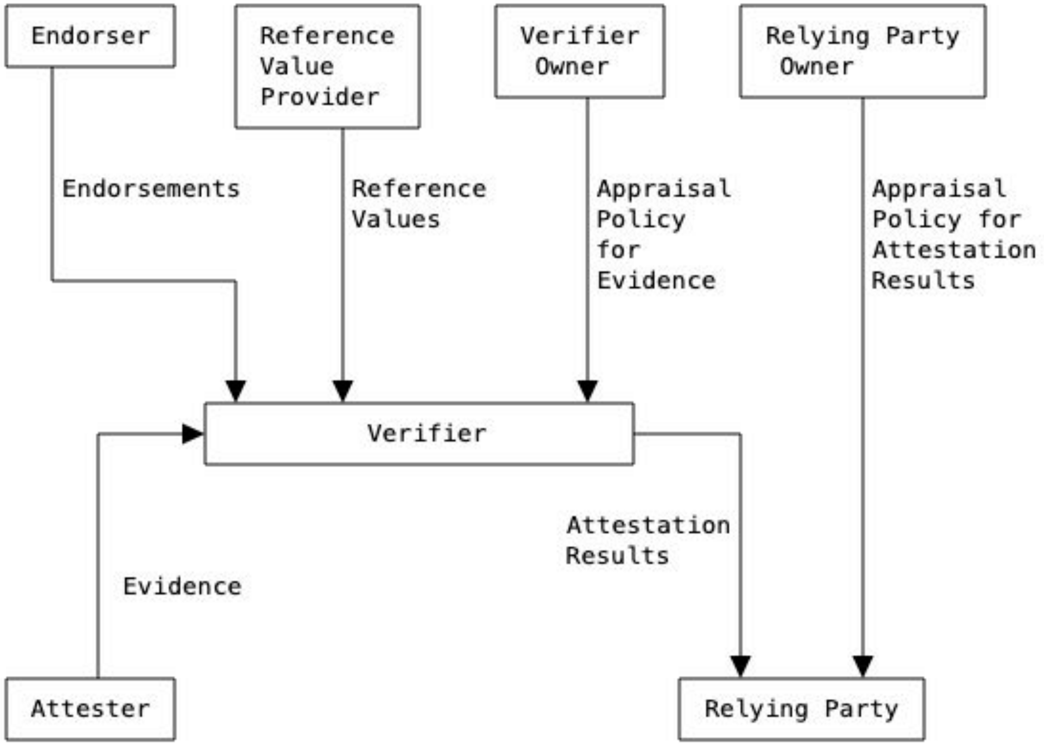
A CVM attests to request a resource. The KBS sends the evidence for appraisal and then appraises the result before releasing a resource like a key.



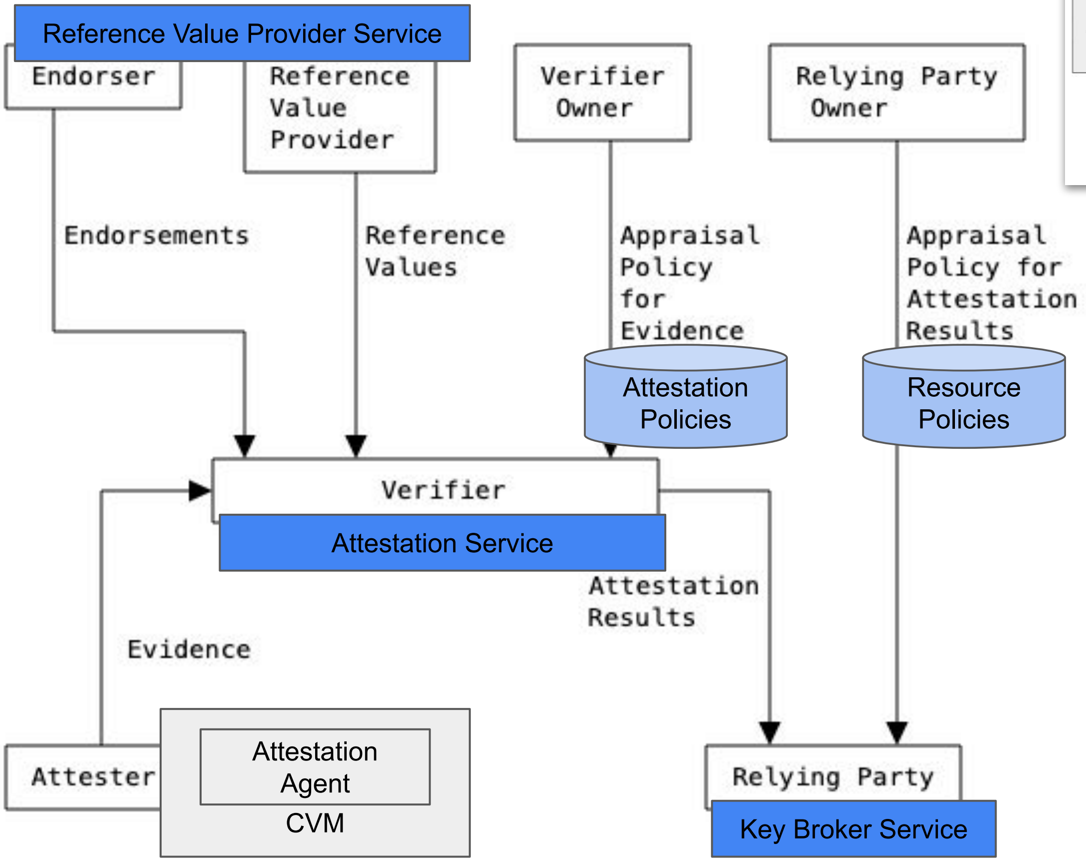
KMS or any other relying party services

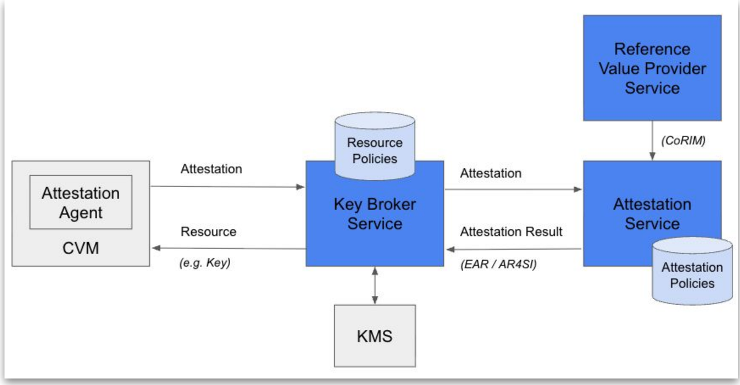
Trustee can also call external services like NRAS, ITA, etc.

RATS reminder



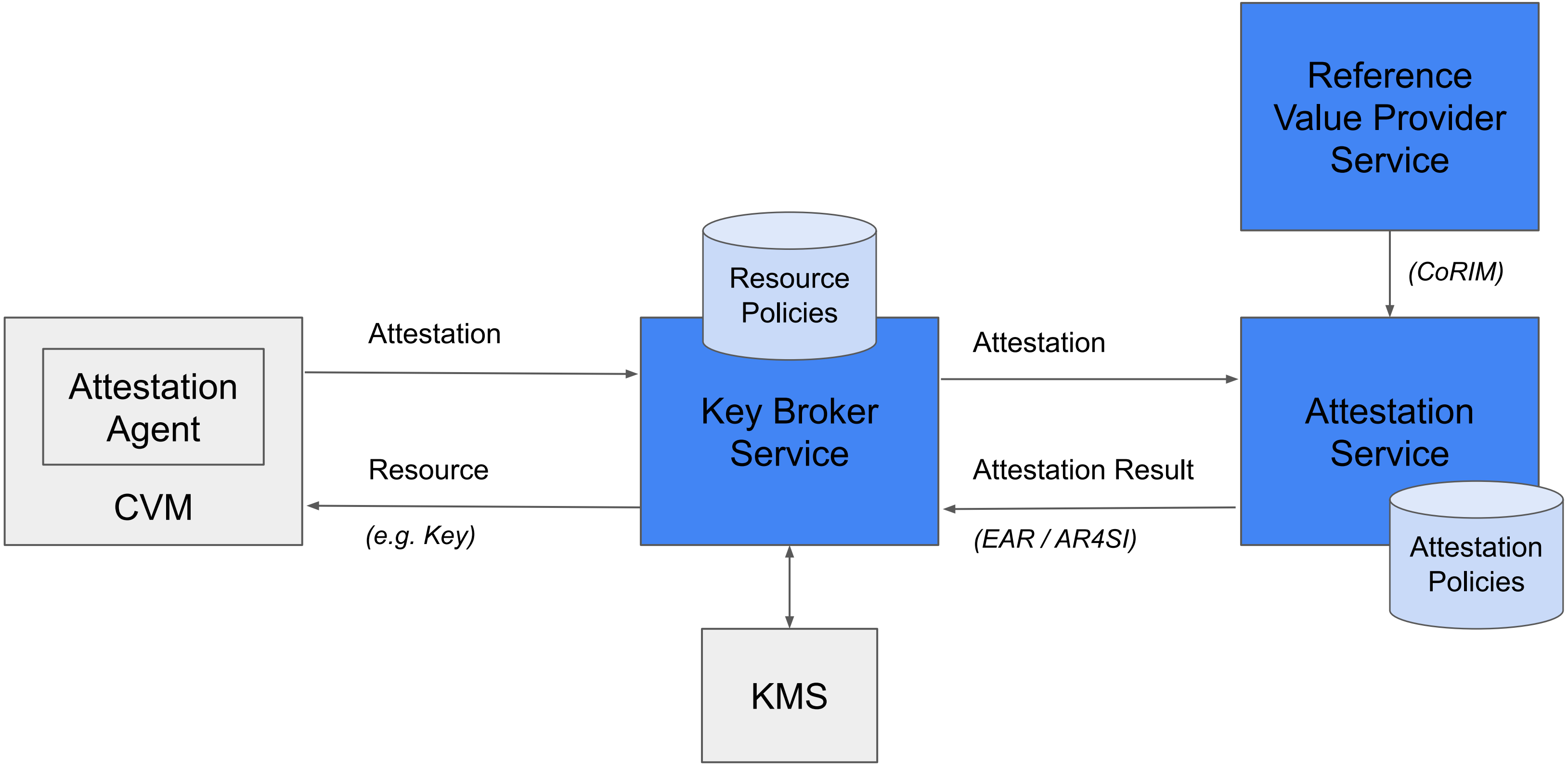
You Are Here



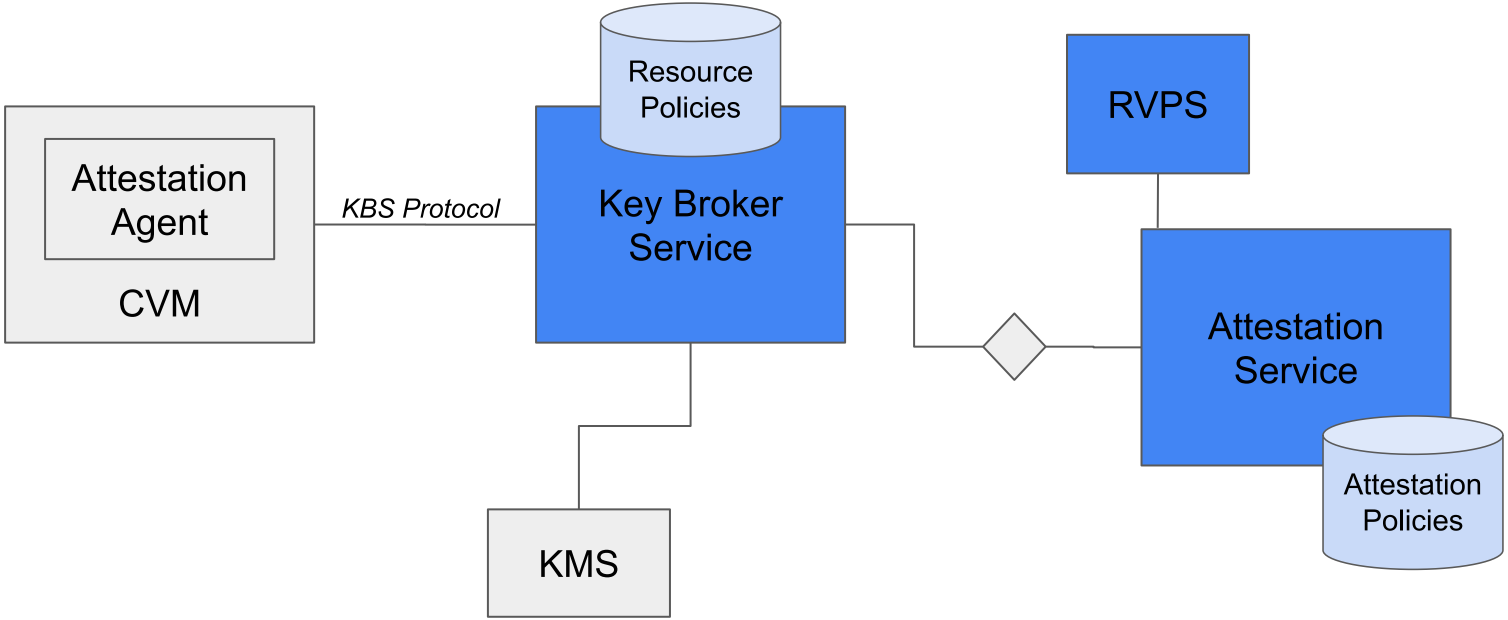


Mapping Trustee Architecture on RATs

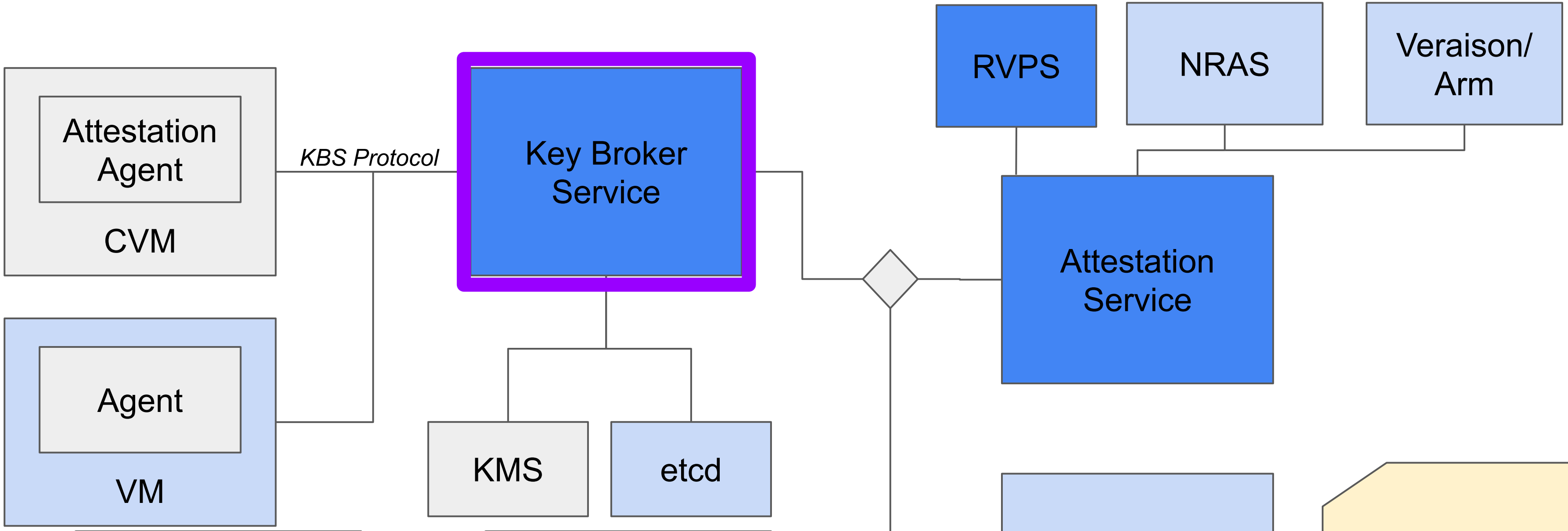
Trustee Architecture



Deployment Versatility



Deployment Versatility



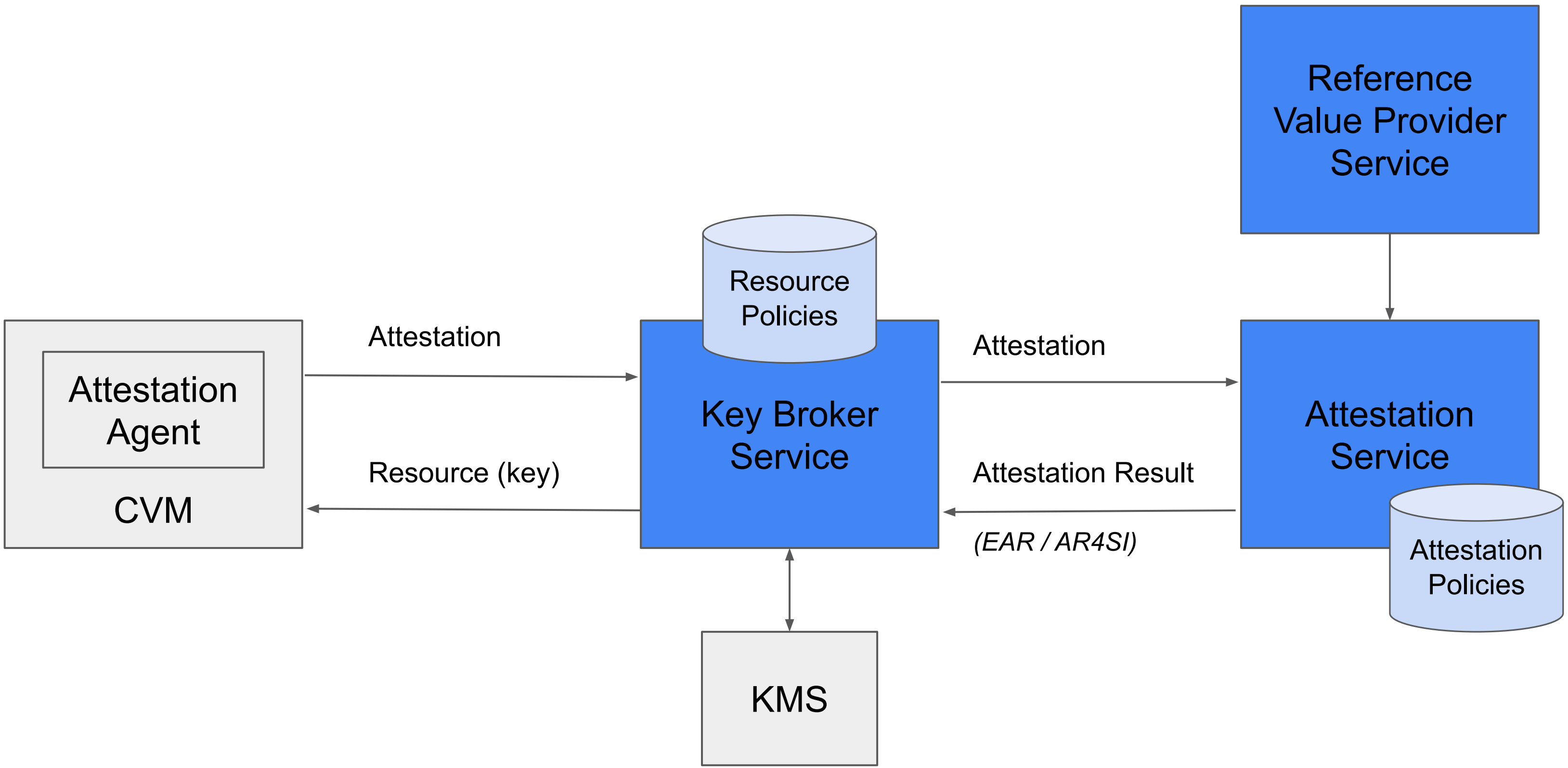
Trustee AS supports vendor plugins for integrated or remote verifiers

Non-CC workloads can still use the KBS protocol to access resources with appropriate policies

Trustee can broker resource access by URI. including e.g. k8s based secrets encrypted to etcd

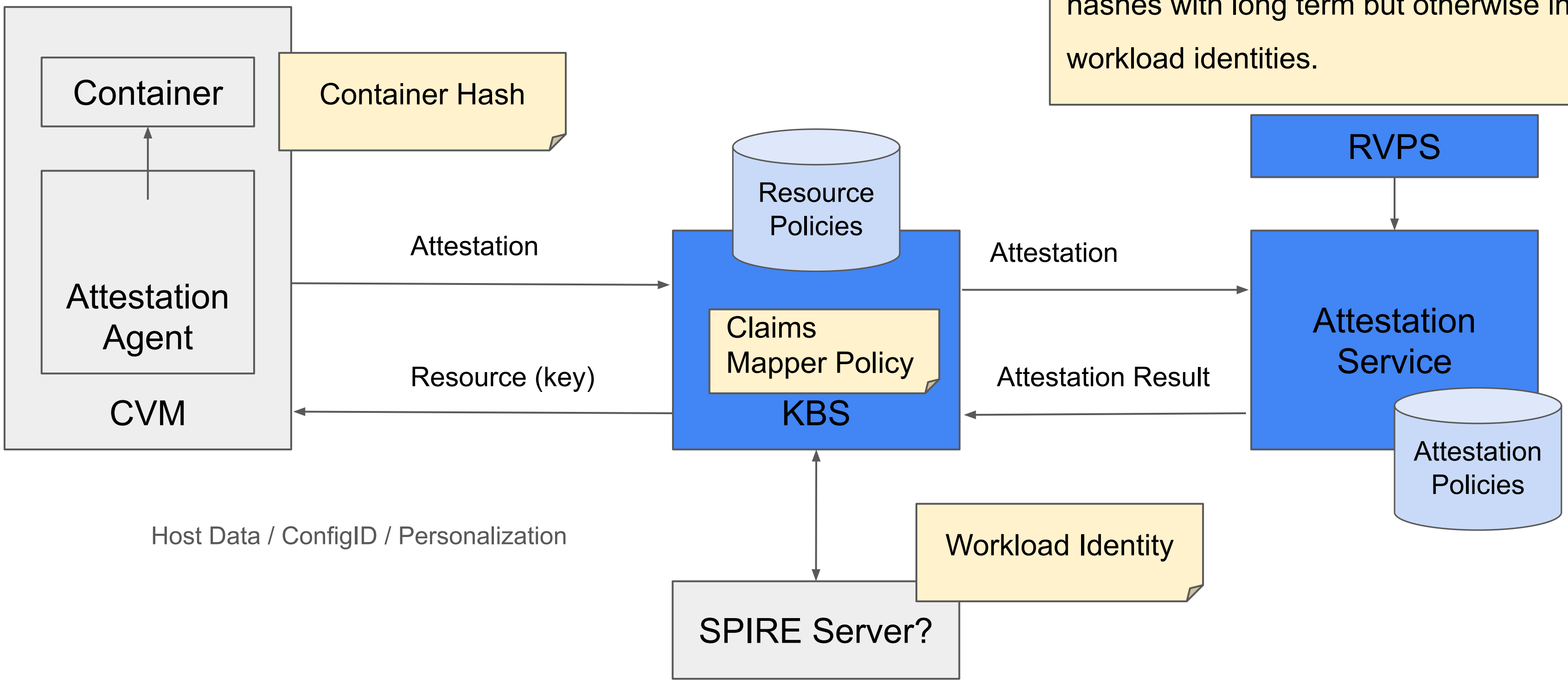
Trustee can directly call 3rd party attestation services

Trustee Architecture



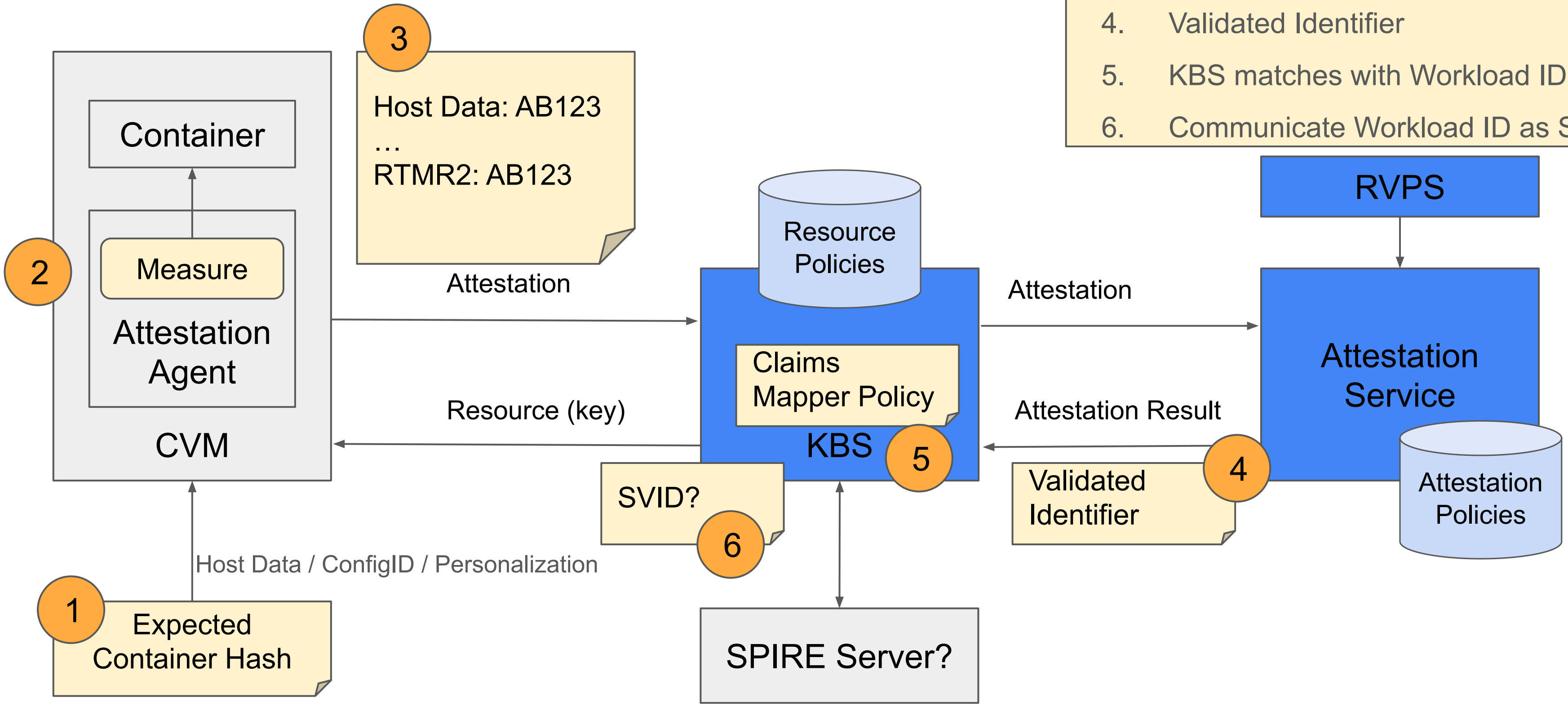
Validated Identifier / Workload Identity

Trustee could realize the **Claims Mapper** function from **CCC Workload Identity SIG**. This would map brittle but secure container hashes with long term but otherwise insecure workload identities.



Validated Identifier / Workload Identity

1. Create CVM with expected container
2. Agent enforces expectation
3. Attestation includes measurements
4. Validated Identifier
5. KBS matches with Workload ID policy
6. Communicate Workload ID as SVID

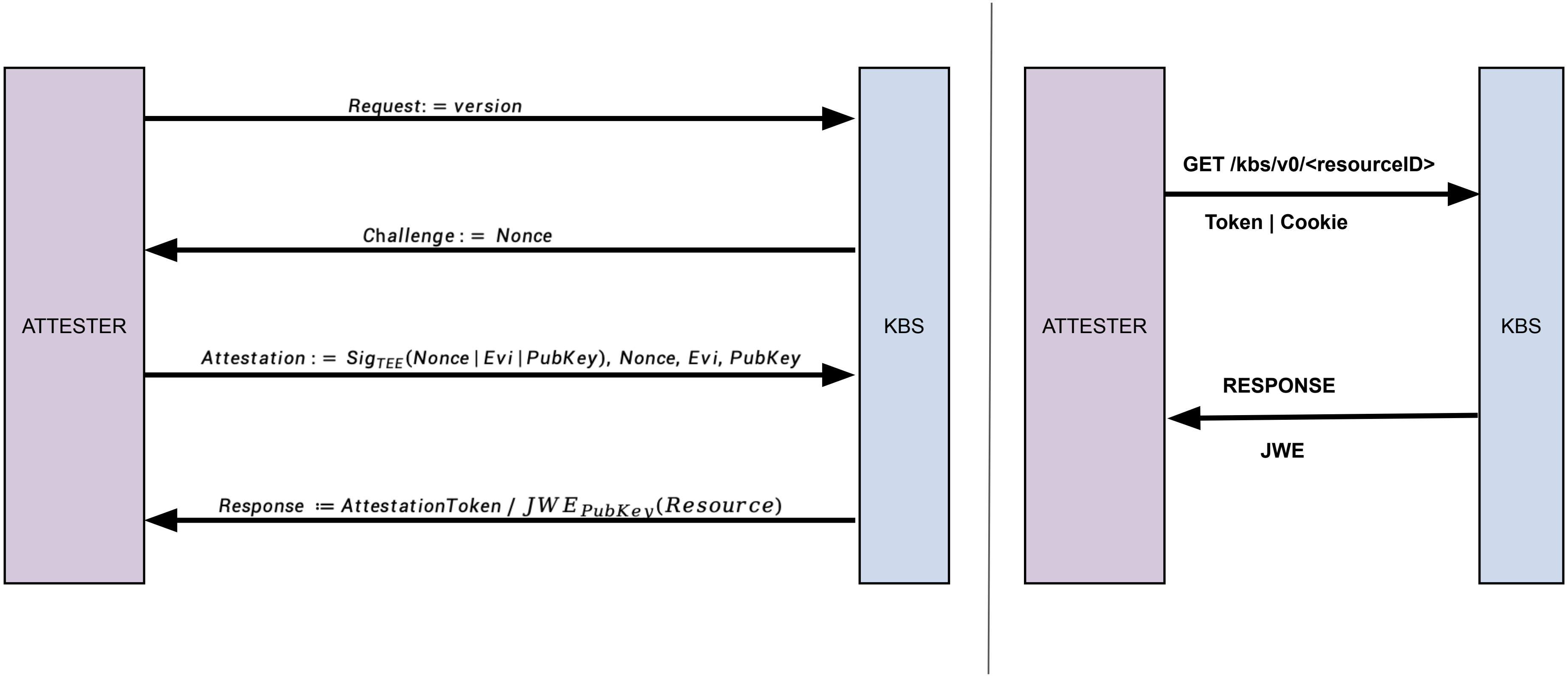


KBS 2-Phase Protocol

1. **Authenticate**
   RCAR Handshake (Request-Challenge-Attestation-Response)
   Result: Attestation Token or Session Cookie

2. **Request Resource**
   Present: Token | Cookie
   Retrieve: kbs://<repository>/<type>/<tag>

KBS Protocol: Phase 1 RCAR | Phase 2 Get Resource



ATTESTER

$Request := version$

$Challenge := Nonce$

$Attestation := Sig_{TEE}(Nonce \mid Evi \mid PubKey),\ Nonce,\ Evi,\ PubKey$

$Response \coloneqq AttestationToken\ /\ JWE_{PubKey}(Resource)$

KBS

ATTESTER

GET /kbs/v0/<resourceID>

Token | Cookie

RESPONSE

JWE

KBS

Coming soon

Trustee in an Enclave

Establishing RoT in Trustee Enclave

New verifiers as new platforms emerge

Continue collaborating on Workload Identity

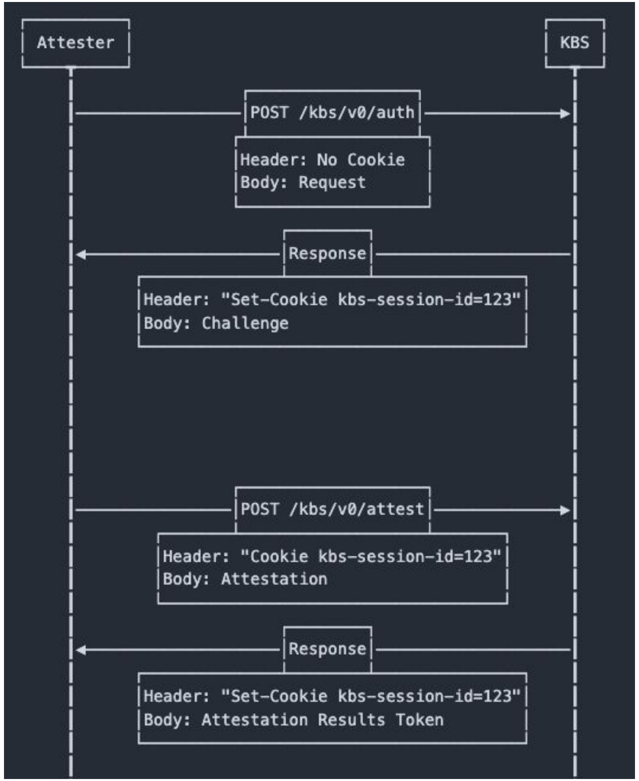
Standardize on KBS Protocol through adoption

Questions?

BACKUP

KBS Protocol

Authenticate



Attester

KBS

POST /kbs/v0/auth

Header: No Cookie
Body: Request

Response

Header: "Set-Cookie kbs-session-id=123"
Body: Challenge

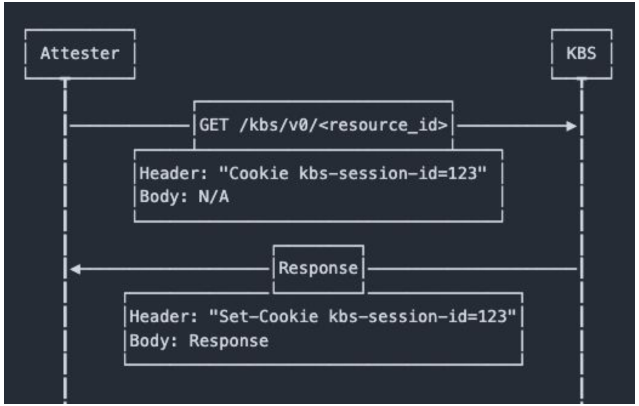
POST /kbs/v0/attest

Header: "Cookie kbs-session-id=123"
Body: Attestation

Response

Header: "Set-Cookie kbs-session-id=123"
Body: Attestation Results Token

Request Resource



Attester

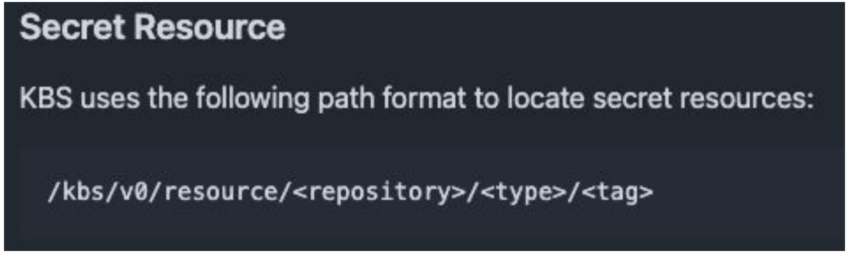
KBS

GET /kbs/v0/<resource_id>

Header: "Cookie kbs-session-id=123"
Body: N/A

Response

Header: "Set-Cookie kbs-session-id=123"
Body: Response



Secret Resource

KBS uses the following path format to locate secret resources:

```
/kbs/v0/resource/<repository>/<type>/<tag>
```

https://github.com/confidential-containers/trustee/blob/main/kbs/docs/kbs_attestation_protocol.md