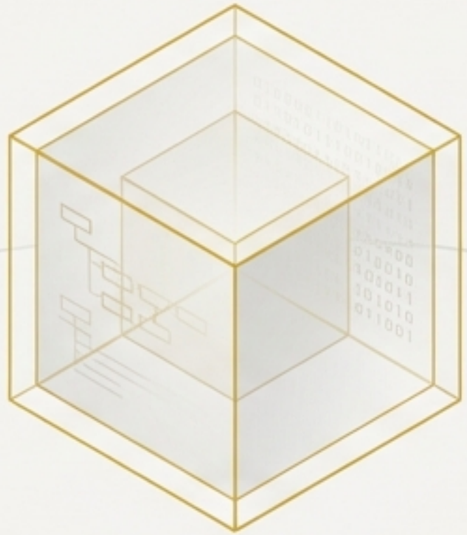
The KERI/ACDC/A2A Stack: A New Foundation for Secure Digital Interaction

KERI (THE LOCK)



ACDC (THE PACKAGE)

A2A (THE TRANSPORT)

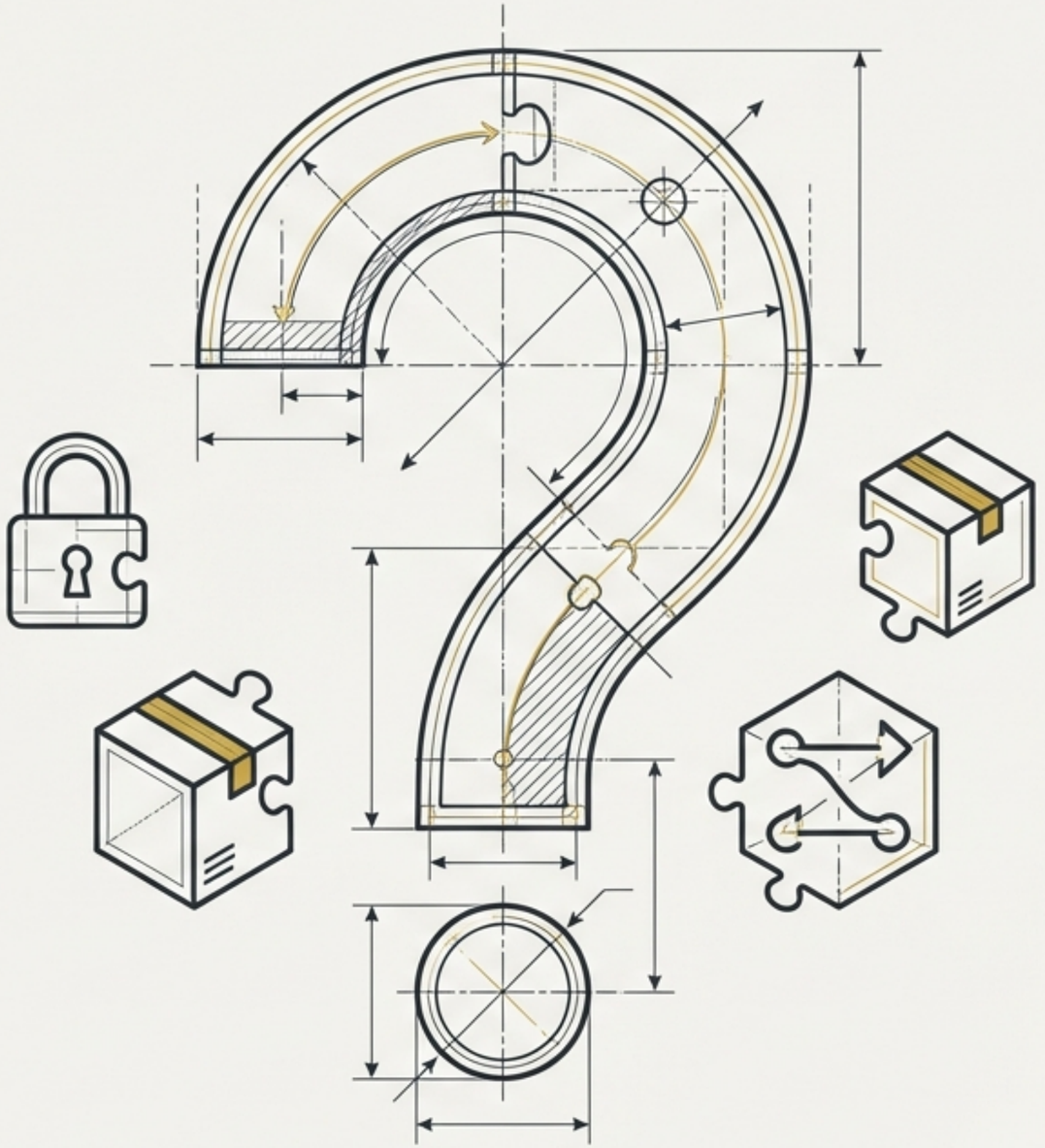
An integrated approach to verifiable data exchange, built on cryptographic certainty.

Three Powerful Technologies, One Integrated Solution

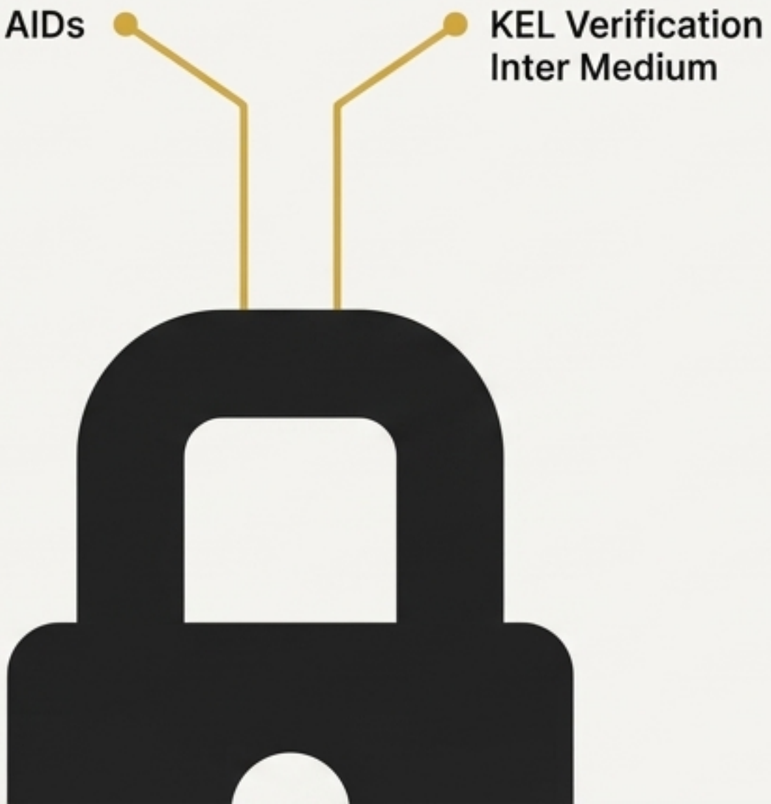
The digital world relies on the exchange of information, but current methods often lack cryptographic proof. We have powerful, distinct solutions:

- **KERI**: An immutable root of trust for identity.
- **ACDC**: Secure, chained data containers.
- **A2A**: A protocol for peer-to-peer interaction.

The critical question is not what they do individually, but how they combine to form a uniquely secure and verifiable data stack.



The Foundation: KERI is the Unquestionable Trust Spanning Layer

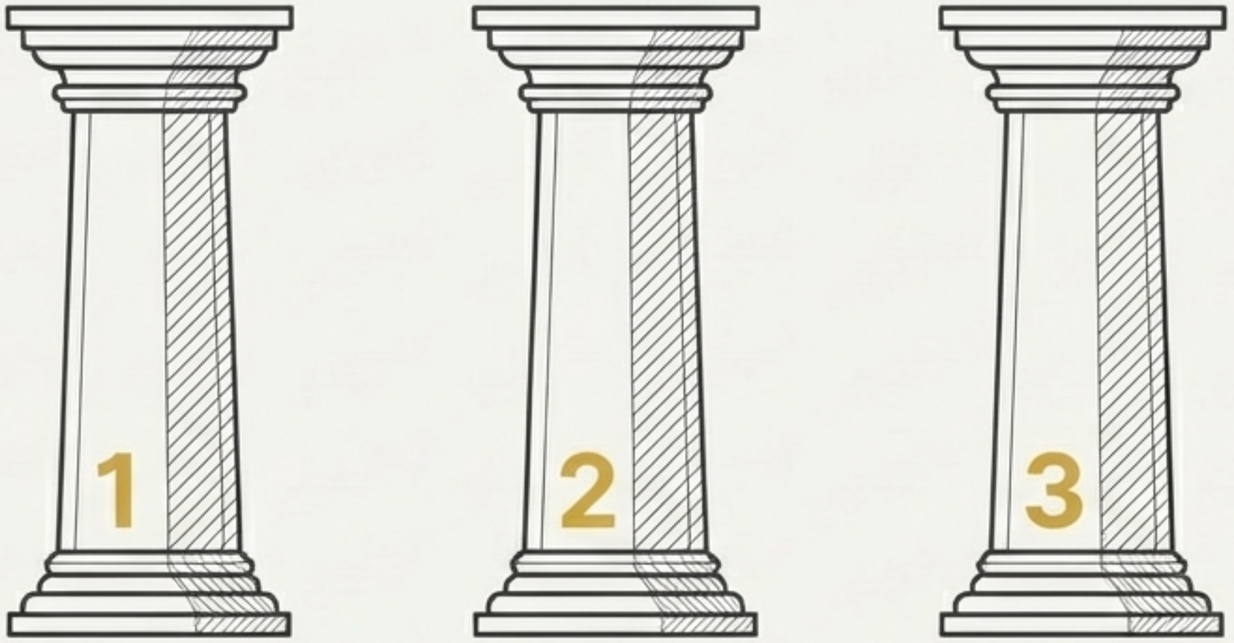


KERI provides the cryptographic certainty that underpins the entire stack. It is not an add-on; it is the anchor.

- **For Identity:** A2A Agents use KERI Autonomic Identifiers (AIDs) to establish who they are.
- **For Verification:** Signatures on ACDCs are verified against the issuer's KERI Key Event Log (KEL), ensuring the keys were valid at the precise moment of issuance.

With Trust Established, ACDC and A2A Erect Three Pillars of Capability

The integration of Authentic Chained Data Containers (ACDCs) with the Agent-to-Agent (A2A) protocol delivers three critical, verificale capabilities. We will We will explore each one.



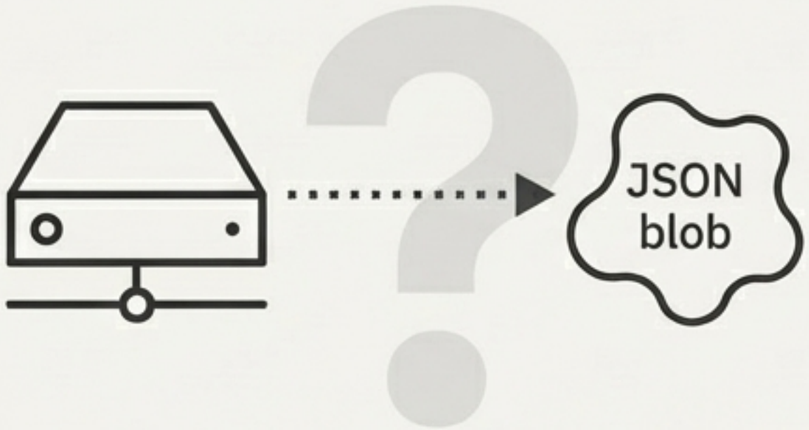
1. Provenance
IBM Plex Serif

2. Authorization

3. Privacy
IBM Plex Serif

Pillar 1: ACDCs Transform A2A Artifacts into Objects of Verifiable Provenance

The Old Way



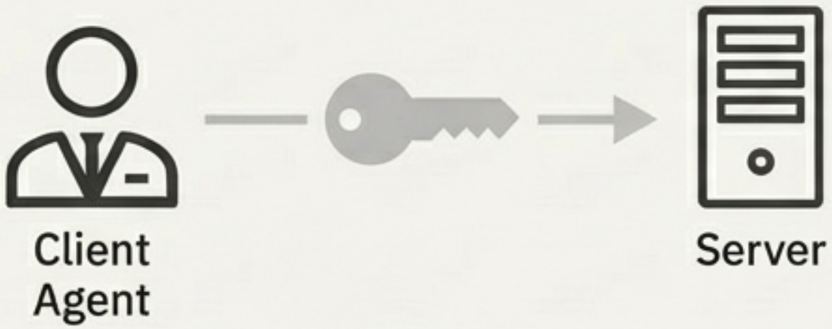
Without ACDCs, an A2A task produces an Artifact that is just a JSON blob. Its integrity relies entirely on trusting the agent's server security.

The New Way

The A2A Artifact is an ACDC. The receiver gets a cryptographically verifiable container, anchored to the issuer's KERI log. This provides **provenance**: definitive proof of *who* issued the data and that it is untampered.

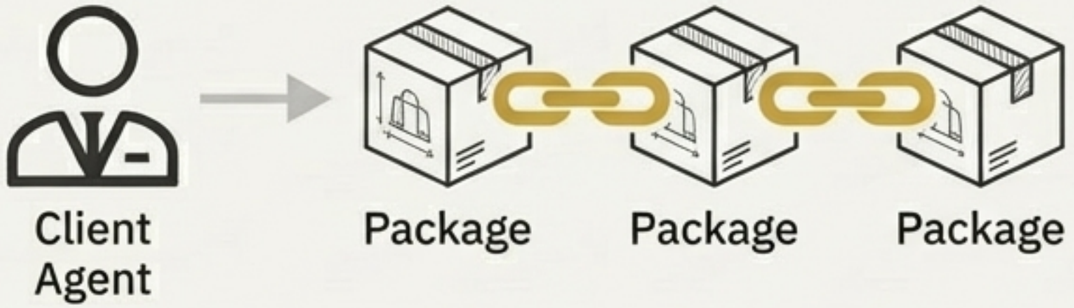
Pillar 2: ACDCs Enable Verifiable, Delegated Authorization within A2A

Opaque Authority



Traditional authorization relies on simple secrets like API keys, which offer no verifiable proof of the authority's origin or its delegation chain.

Provable Delegation



Instead of a key, the agent sends an ACDC chain in the A2A metadata. This chain provides a verifiable, cryptographic **audit trail** of delegated authority.

The Mechanism of Authority: How ACDC Chaining Creates a Verifiable Trust Network

Because ACDCs natively support chaining, an agent can present a sequence of credentials that cryptographically prove a delegation path. This moves authorization from a simple binary check check to a rich, auditable narrative.

Root Authority X issues credential to...

Manager Y, who issues credential to...

Agent Z, who can now perform a specific action.

This entire chain can be presented within a single A2A task, allowing for immediate, trustless verification of complex authorization.

Pillar 3: A2A and ACDC Combine for Controlled, Private Data Disclosure

Un-permissioned data exploitation is a critical risk. The A2A protocol's support for multi-turn interactions, combined with the ACDC concept of **Chain-Link Confidentiality**, enables a robust, consent-based disclosure process.

This ensures data is only revealed after terms are explicitly agreed upon.

The Privacy Protocol in Action: A Three-Step Exchange

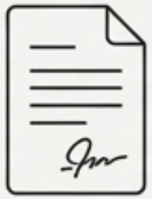
Sender Agent

Receiver Agent

Step 1: **Offer**

The Sender sends a stripped-down ACDC revealing only the data schema and issuer, but no private information.

1



Step 2: **Agree**

The Receiver reviews the terms and sends back a cryptographically signed agreement (e.g., 'I agree not to resell this data').

2

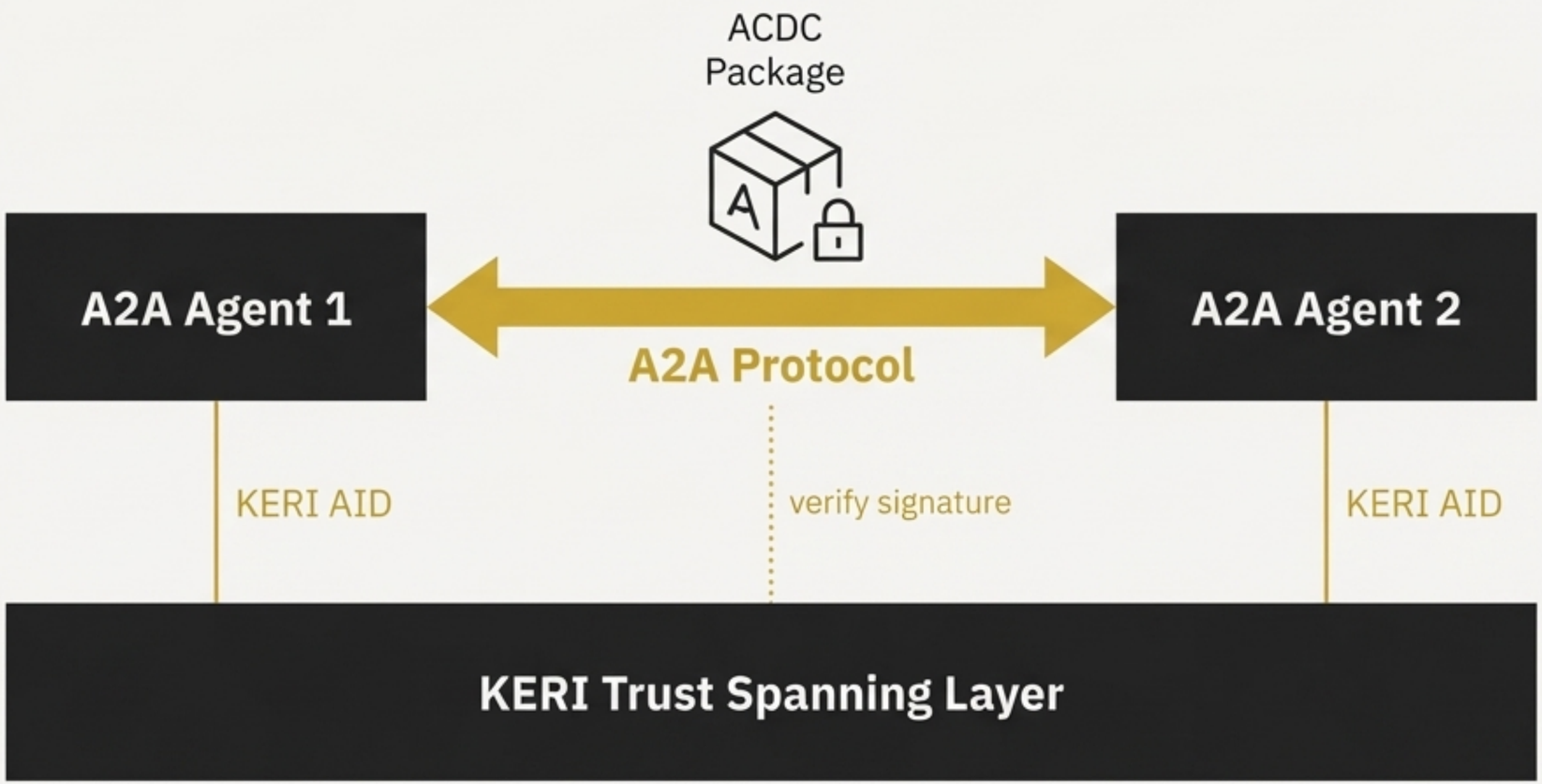
Step 3: **Disclose**

Upon verifying the agreement, the Sender sends the Full ACDC containing the private data.

3

A Unified Architecture for Secure Exchange Secure Blueprint

These components do not operate in isolation. They form a complete, layered architecture where identity, data, and transport are seamlessly and securely integrated.
KERI provides the trust anchor for
A2A agent identities and ACDC issuance, while A2A provides the secure channel to exchange and negotiate access to ACDCs.



The Core Analogy: Lock, Package, Transport

KERI handles the Lock.

It provides the immutable, cryptographic root of trust for identity and verification.

ACDC handles the Package.

It is the secure, verifiable data container that holds the information being exchanged.

A2A handles the Transport.

It is the peer-to-peer protocol that manages the communication, negotiation, and secure exchange.

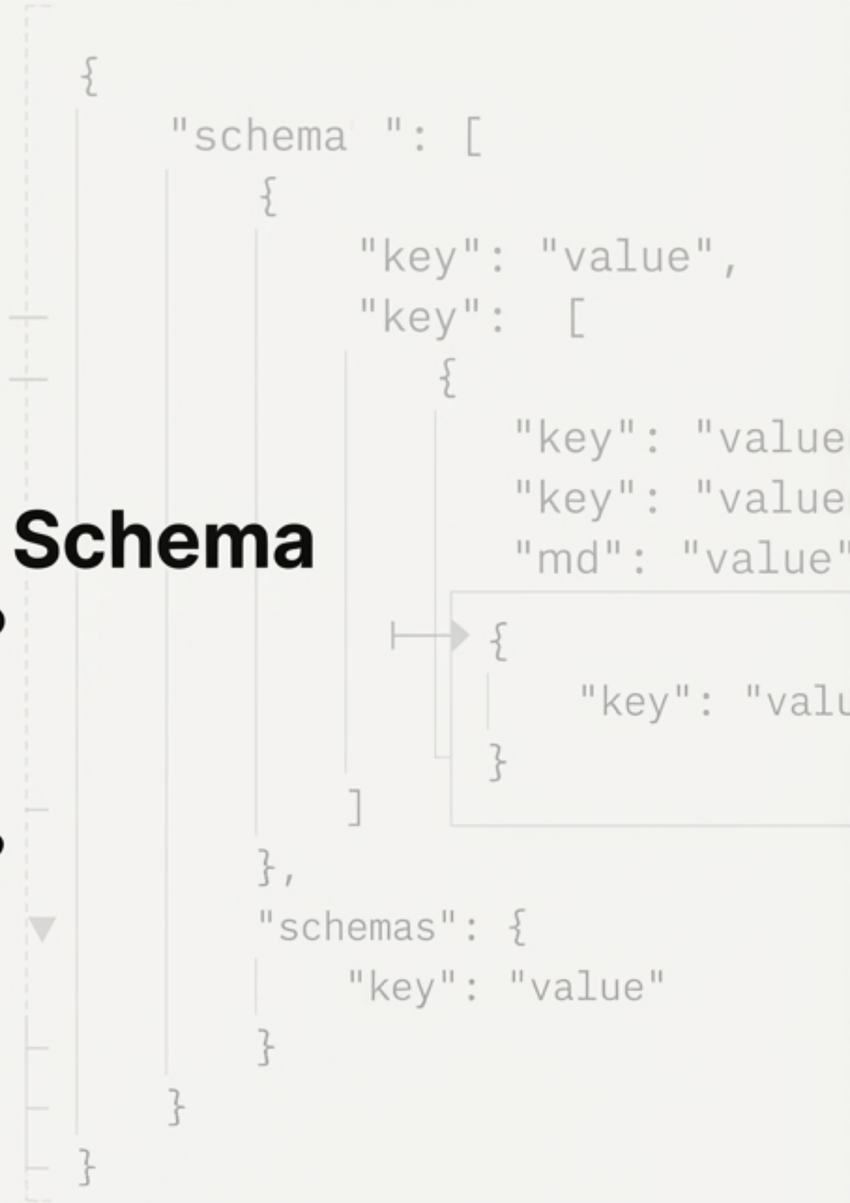
The Complete Workflow: From Cryptographic Identity to Secure Transport

Identity (KERI) Data (ACDC) Transport (A2A)

This is the complete picture of the KERI/ACDC/A2A stack. A seamless integration moving from cryptographic identity, through containerised verifiable data, to a secure peer-to-peer transport layer.

From Architecture to Application

With the architectural foundation established, the next stage is implementation. The power of the stack is realised in its application to specific use cases.



How would you define an ACDC Schema for a specific A2A task?

– A ‘Verified Purchase Order’?
– A ‘Validated Medical Result’?
– A ‘Delegated Access Right’?