CESR for ~~Muggles~~ First Year Wizards

IIW #35
Day 1 - Session #_
15 November 2022

https://keri.one

CESR (Composable Event Streaming Representation) is a data exchange format fundamental to the work on KERI and ACDC at the Trust Over IP (ToIP) ACDC Task Force.

https://trustoverip.org/

The purpose of this session to explain the basic features of CESR to anyone who wants to quickly gain a high-level understanding.

Format

- First, a little background from ACDC chair Sam Smith
- Then 6 minutes for each of the 7 main features
  - A quick explanation of the basic idea
  - Questions for Sam—but only about that feature
  - STRICT CUT-OFF AT SIX MINUTES
- Close with general Q&A

Meet ACDC chair Dr. Sam Smith

Note: CESR is part of KERI and ACDC

See the companion slide decks:

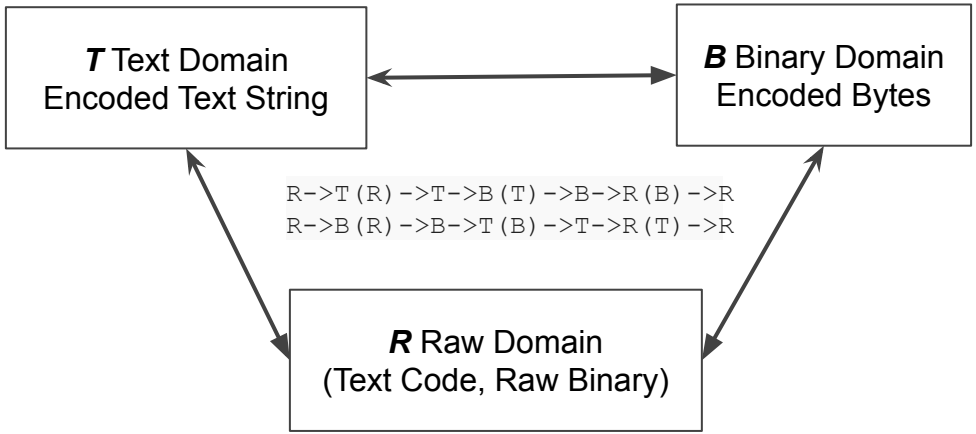
- **KERI for Muggles** (originally shown at IIW #32)
- **ACDC for Muggles** (originally shown at IIW #34)
- **https://keri.one/** - look under “Resources”

CESR in Seven Easy Steps

#1: Text and Binary Composability

Any composition of data primitives (or groups of primitives) can be converted en mass between text and binary without losing its decomposability

Example

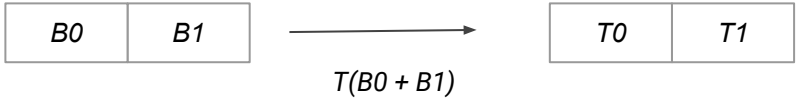


B(t[0] + t[1]) = B(t[0]) + B(t[1]) = b[0] + b[1]

T(b[0] + b[1]) = T(b[0]) + T(b[1]) = t[0] + t[1].

Example

Base64 Encode

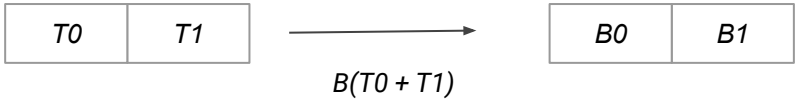


4694e894769e6c3267e8b477c2590284cd647dd42ef6007d254fce1cd2e9be423 +

10b0b92f7881543efb77f3186d8186094420a90063bb5a38c7551dfb3dac2febb1

BGlOiUdp5sMmfotHfCWQKEzWR91C72AH0lT84c0um-QjELC5L3iBVD77d_MYbYGGCUQgqQBju1o4x1Ud-z2sL-ux

Base64 Decode



BGlOiUdp5sMmfotHfCWQKEzWR91C72AH0lT84c0um-Qj + ELC5L3iBVD77d_MYbYGGCUQgqQBju1o4x1Ud-z2sL-ux

4694e894769e6c3267e8b477c2590284cd647dd42ef6007d254fce1cd2e9be42310b0b92f7881543efb77f3186d8186094420a90063bb5a38c7551dfb3dac2febb1

Why text & binary composability?

1. You can design, manage, code, and understand any content in the text domain — and then when you want to transmit it, you can convert it to binary and get a **linear reduction in the bandwidth** without any conversion or other work
2. **No more text vs. binary wars** — for example, no more JSON vs CBOR or JOSE vs COSE standards battles

Benefit #1

You get all the advantages of text and of binary encoding without having to pick — you get both!

#2: Readability and ease of use for developers

In CESR, all types of primitives — both cryptographic (hashes, keys, signatures) and others (numbers, special types) — are expressed as strings, not data structures

Ed25519 Public Key: BGlOiUdp5sMmfotHfCWQKEzWR91C72AH0lT84c0um-Qj

Blake3-256 Digest: ELC5L3iBVD77d_MYbYGGCUQgqQBju1o4x1Ud-z2sL-ux

Ed25519 Signature:
0BCdI8OSQkMJ9r-xigjEByEjIua7LHH3AOJ22PQKqljMhuhcgh9nGRcKnsz5KvKd7K_H9-1298F4Id1DxvIoEmCQ

Salt: 0AAwMTIzNDU2Nzg5YWJjZGVm

Number Short: MAAB

Number Big: NP__________

DateTime: 1AAG2020-08-22T17c50c09d988921p00c00

OOBI URl:
http://127.0.0.1:5642/oobi/BBilc4-L3tFUnfM_wJr4S4OJanAv_VmF_dJNN6vkf2Ha/controller

DID: did:keri:EEBp64Aw2rsjdJpAR0e2qCq3jX7q7gLld3LjAwZgaLXU

Group with nested group (Indexed Trans AID Sigs for Keystate at event)
Pre + snu + dig + sigs group :
-FABEAzjKx3hSVJArKpIOVt2KfTRjq8st22hL25Ho9vnNodz0AAAAAAAAAAAAAAAAAAAAAAEAzjKx3hSVJArKpIOVt
2KfTRjq8st22hL25Ho9vnNodz-AABAAD-iI61odpZQjzm0fN9ZATjHx-KjQ9W3-CIlvhowwUaPC5KnQAIGYFuWJyRgA
QalYVSEWoyMK2id_ONTFUE-NcF

ACDC with CESR Primitives:


```
{
 "v": "ACDC10JSON00011c_",
 "d": "EBdXt3gIXOf2BBWNHdSXCJnFJL5OuQPyM5K0neuniccM",
 "i": "did:keri:EmkPreYpZfFk66jpf3uFv7vklXKhzBrAqjsKAn2EDIPM",
 "ri": "did:keri:EymRy7xMwsxUelUauaXtMxTfPAMPAI6FkekwlOjkggt",
 "s": "E46jrVPTzlSkUPqGGeIZ8a8FWS7a6s4reAXRZOkogZ2A",
 "a": "EgveY4-9XgOcLxUderzwLIr9Bf7V_NHwY1lkFrn9y2PY",
 "e": "EIl3MORH3dCdoFOLe71iheqcywJcnjtJtQIYPvAu6DZA",
 "r": "EDIai3Wkd-Z_4cezz9nYEcCK3KNH5saLvZoS_84JL6NU"
}
```


Attached CESR Signature or Reference to TEL (Issuance and Revocation Registry) Anchor


```
-VAX-BABAABtOhjlKo8WhJQ3EXMIMaQ_IH6yeyxs7_JuO4RioH1NUTtzTuV1bbuB
7eoNhEj20VJYa4947ZMVrOxKhzI6EqUH
```


VC with JWT/JWK/JWS :


```
{
  "sub": "did:example:ebfeb1f712ebc6f1c276e12ec21",
  "nbf": 1560711419,
  "iss": "did:example:76e12ec712ebc6f1c221ebfeb1f",
  "exp": 1560797819,
  "vc": {
    "@context": [
      "https://www.w3.org/2018/credentials/v1",
      "https://www.w3.org/2018/credentials/examples/v1"
    ],
    "type": [
      "VerifiableCredential",
      "UniversityDegreeCredential"
    ],
    "credentialSubject": {
      "college": "Test University",
      "degree": {
        "name": "Bachelor of Science and Arts",
        "type": "BachelorDegree"
      }
    }
  },
  "jti": "http://example.edu/credentials/3732"
 }
```


Attached JWS

Why text-based primitives?

1. Easy to read and embed in text, documents, logs, and identifiers
2. Any cryptographic material (keys, hashes, signatures, etc.) can be embedded in a namespace
3. Any namespace may be extended with one or more cryptographic primitives as elements (enabling cryptographically-agile, future-proofed namespaces)
4. Every single primitive can be used as a address on in a URL

Benefit #2

CESR’s all-string format makes it easy for developers to use for all types of data exchange tasks

#3: Ease of streaming and pipelining data

CESR works for all kinds of data transmission — it can handle streaming or pipelined data as easily as fixed or batch data

Streaming Text:


```
{"v":"KERI10JSON0000fd_","t":"icp","d":"EBXqe7Xzsw2aolT09Ouh5Zw9kNn2sgoHmo4zCn7Q7ZSC","i":"
BAqph4mAWcf7mkIgk1Xrpvr7dWT7YvHIam_hqUAT2rqw","s":"0","kt":"1","k":["BAqph4mAWcf7mkIgk1Xrpv
r7dWT7YvHIam_hqUAT2rqw"],"nt":"0","n":[],"bt":"0","b":[],"c":[],"a":[]}

-AABAAB7WHPA5UPHhV5DRKUU93pXnwp4bPGDQ-DIrFsVr6kPIpHByaM2WPC7SgHXVn3MMGjsdJc1Ul8LrvUc1VrV46c
L{"v":"KERI10JSON000091_","t":"rct","d":"ELfp9ZhqQCGov3wPRLa6vn5VkIQjug2sb2QD17T-TIpY","i":
"ELfp9ZhqQCGov3wPRLa6vn5VkIQjug2sb2QD17T-TIpY","s":"0"}-CABBAqph4mAWcf7mkIgk1Xrpvr7dWT7YvHI
am_hqUAT2rqw0BDaa9nAkQ2-M2_Mr4Kecfa9Y-rR9WD3IKDV3AG4USGCP-wA2rIAzw6vBABM9eCIs6mETGykfX04DCW
avJsrfjMK{"v":"KERI10JSON000091_","t":"rct","d":"EFECUzlLZ3IKG9Kvkj51a0RYPYXnUeZ5SIpw8x3SPS
1E","i":"ELfp9ZhqQCGov3wPRLa6vn5VkIQjug2sb2QD17T-TIpY","s":"1"}-CABBAqph4mAWcf7mkIgk1Xrpvr7
dWT7YvHIam_hqUAT2rqw0BAEUr8d1CAe9HYGOXeqbgBPf9IFM0L1iNw6ZgMlfJ4djVvZ8Fl250sAh4thrsOFaaNqYCV
g8uWRS3YtpEu5yfEE{"v":"KERI10JSON000091_","t":"rct","d":"EELHnIwzGaJ-twKTfXtsPMteqsIVmDpiwV
O574h6LRXD","i":"ELfp9ZhqQCGov3wPRLa6vn5VkIQjug2sb2QD17T-TIpY","s":"2"}-CABBAqph4mAWcf7mkIg
k1Xrpvr7dWT7YvHIam_hqUAT2rqw0BCt52hAdsQw3LONIIzelwVZpGZX6vqxZEFp3nxtz657xS8Y92ngcGhYK30Wc1_
-y8baTDb-NAsL3pLJyn_czSUJ{"v":"KERI10JSON000091_","t":"rct","d":"ECWvVQFFqxmAW-vpSLwWj4yPO0
4nGA-6l8cifNBlc3gK","i":"ELfp9ZhqQCGov3wPRLa6vn5VkIQjug2sb2QD17T-TIpY","s":"3"}-CABBAqph4mA
Wcf7mkIgk1Xrpvr7dWT7YvHIam_hqUAT2rqw0BCc8xIDPi9H1kpSMjELYByC51ULre7Y0m_9Ftti23NtrRmbOV8RwgL
E4mzrbwwSOksKhzNoqX3QXZsDjXU5N1UL{"v":"KERI10JSON000091_","t":"rct","d":"EFcoQIrpd4_NMcnL7S
vVqUSLfPZOzkAGbtQcE3JVMn7D","i":"ELfp9ZhqQCGov3wPRLa6vn5VkIQjug2sb2QD17T-TIpY","s":"4"}-CAB
BAqph4mAWcf7mkIgk1Xrpvr7dWT7YvHIam_hqUAT2rqw0BBeLpMLawI0J7ZzESvU4M86mTCGPcuYlGux9en-6PBIO7x
ZGMuVMCFwqWFmkMlhJ_ZEIUVm9ZgSp-P8jtTU4yUL{"v":"KERI10JSON000091_","t":"rct","d":"ELVXLfglCi
mN6Y-HkpoLoLiQkR1v65rrg7JRDhcToXVn","i":"ELfp9ZhqQCGov3wPRLa6vn5VkIQjug2sb2QD17T-TIpY","s":
"5"}-CABBAqph4mAWcf7mkIgk1Xrpvr7dWT7YvHIam_hqUAT2rqw0BCUPQtStUkkvgVW-Aq4mNzVpT0PNvSjrLjR024
98Z4AiM7lbmkJTDPL1gU4yuu_G_Lc7q6V_EWsZUxfMyw3HysP{"v":"KERI10JSON000091_","t":"rct","d":"EC
SAoB-QcY3Vnia2G80NLVMkiGssUV70JoWxwJbqx9gL","i":"ELfp9ZhqQCGov3wPRLa6vn5VkIQjug2sb2QD17T-TI
```

Streaming Text:

{"v":"KERI10JSON0000fd_","t":"icp","d":"EBXqe7Xzsw2aolT09Ouh5Zw9kNn2sgoHmo4zCn7Q7ZSC","i":"BAqph4mAWcf7mkIgk1Xrpvr7dWT7YvHIam_hqUAT2rqw","s":"0","kt":"1","k":["BAqph4mAWcf7mkIgk1Xrpvr7dWT7YvHIam_hqUAT2rqw"],"nt":"0","n":[],"bt":"0","b":[],"c":[],"a":[]}

-AABAAB7WHPA5UPHhV5DRKUU93pXnwp4bPGDQ-DIrFsVr6kPIpHByaM2WPC7SgHXVn3MMGjsdJc1Ul8LrvUc1VrV46cL

{"v":"KERI10JSON000091_","t":"rct","d":"ELfp9ZhqQCGov3wPRLa6vn5VkIQjug2sb2QD17T-TIpY","i":"ELfp9ZhqQCGov3wPRLa6vn5VkIQjug2sb2QD17T-TIpY","s":"0"}

-CABBAqph4mAWcf7mkIgk1Xrpvr7dWT7YvHIam_hqUAT2rqw0BDaa9nAkQ2-M2_Mr4Kecfa9Y-rR9WD3IKDV3AG4USGCP-wA2rIAzw6vBABM9eCIs6mETGykfX04DCWavJsrfjMK

{"v":"KERI10JSON000091_","t":"rct","d":"EFECUzlLZ3IKG9Kvkj51a0RYPYXnUeZ5SIpw8x3SPS1E","i":"ELfp9ZhqQCGov3wPRLa6vn5VkIQjug2sb2QD17T-TIpY","s":"1"}

-CABBAqph4mAWcf7mkIgk1Xrpvr7dWT7YvHIam_hqUAT2rqw0BAEUr8d1CAe9HYGOXeqbgBPf9IFM0L1iNw6ZgMlfJ4djVvZ8Fl250sAh4thrsOFaaNqYCVg8uWRS3YtpEu5yfEE

{"v":"KERI10JSON000091_","t":"rct","d":"EELHnIwzGaJ-twKTfXtsPMteqsIVmDpiwVO574h6LRXD","i":"ELfp9ZhqQCGov3wPRLa6vn5VkIQjug2sb2QD17T-TIpY","s":"2"}

-CABBAqph4mAWcf7mkIgk1Xrpvr7dWT7YvHIam_hqUAT2rqw0BCt52hAdsQw3LONIIzelwVZpGZX6vqxZEFp3nxtz657xS8Y92ngcGhYK30Wc1_-y8baTDb-NAsL3pLJyn_czSUJ{"v":"KERI10JSON000091_","t":"rct","d":"ECWvVQFFqxmAW-vpSLwWj4yPO04nGA-6l8cifNBlc3gK","i":"ELfp9ZhqQCGov3wPRLa6vn5VkIQjug2sb2QD17T-TIpY",

Why streaming and pipelining data?

1. Special self-framing grouping or count codes means a stream parser can:
    a. Know how many characters or bytes to offload for a given primitive from the code
    b. Offload a group of primitives without parsing each primitive in the group
2. Supports:
    a. Core affinity concurrent off-loading for core affinity
    b. Hierarchical composition of groups of primitives
3. Enables interleaved:
    a. Text and binary CESR in same stream
    b. CESR with JSON, CBOR, and or MGPK in same stream
4. Future proofed scalability

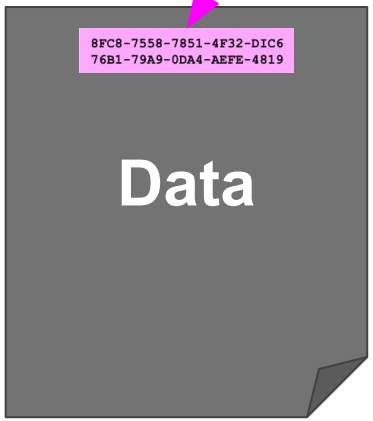
Benefit #3

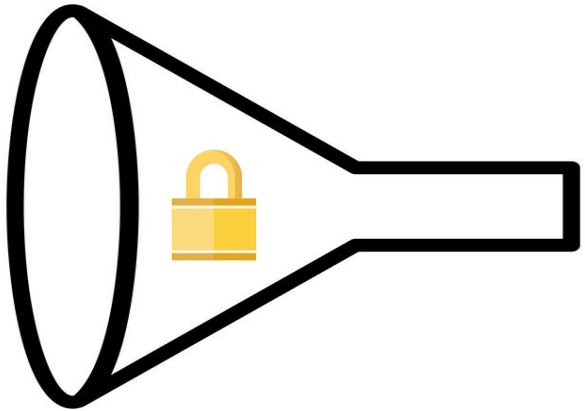
CESR can work as a universal data interchange format for all types of data

#4: Self-addressing data structures

A SAID (Self-Addressed ID) is a self-referential digest (hash) embedded as a CESR-encoded strings inside text data structures, thereby making the data structure a SAD (Self-Addressed Data)

A SAID is a cryptographic digest *bound to its content*





8FC8-7558-7851-4F32-DIC6
76B1-79A9-0DA4-AEFE-4819

SAID

Input = digital content that will include the SAID *after* hashing

Apply cryptographic digest algorithm

Output = short, globally-unique identifier of the content

Why self-addressing data structures?

1. SAIDs provide globally unique, cryptographically agile identifiers for all data
2. Any data with a SAID can have verifiable integrity
3. SAIDS enable:
    a. permissionless registries of data types, data schemas, and other data structures
    b. deduplicated verifiable text data

Benefit #4

SAIDs mean any data object anywhere can be addressed with a verifiable identifier

#5: Legally valid embedded digital text signatures

The relevant legislation, namely, the USA Electronic Signatures in Global and National Commerce Act (ESIGN), the USA Uniform Electronic Transactions Act (UETA) and the EU Regulation for Electronic Identification and Electronic Trust Services (eIDAS) enable digital signatures as legally compliant signatures equivalent in force to wet signatures.

CESR Encoding of signatures more conveniently enables embedding that signatures in text based document envelopes as verifiable test data structures.

Why legally valid digital signatures?

1. Text-based digital signatures are legally binding with recourse.
2. Non-repudiable signature(s) enables data to have verifiable authenticity (duplicity evident) at rest (Zero-Trust Architecture)
3. This includes both:
    a. Hash linked signed data
    b. Hash chained signed data (verifiable data structures)

Benefit #5

CESR signed data enables individuals and businesses to engage in all types of legally valid transactions

#6: Verifiable archival documents

CESR documents of all kinds can be re-verified at any point in the future.

The ISO ISO 14641:2018 standard for the preservation of electronic documents, lists four important features of a legally defensible archive: long-term preservation, data integrity, data security, and traceability.

https://www.gleif.org/assets/components/xbrl-viewer/gleif-annual-report-2021/ixbrl-report-2021-viewer.html

Why verifiable archival documents?

1. Granular accountability of data sourcing in a given document
2. Hierarchical document archives as hash chained signed verifiable data structures in text
3. Examples:
    a. Signed annual reports
    b. Signed XBRL
    c. Verifiable email or chat messages
    d. Complex documents with attachments

Benefit #6

CESR documents provide legally valid evidence of commitments and transactions

#7: Verifiable audit logs

Electronic Code of Federal Regulations: Electronic Signatures (E-CFR) regulation requires audit trails on electronic data including nonredudiable attribution via digital signatures. Relevant clauses are provided below as follows [25]:

> E-CFR
>
> (e) Use of secure, computer-generated, time-stamped audit trails to independently record the date and time of operator entries and actions that create, modify, or delete electronic records. Record changes shall not obscure previously recorded information. Such audit trail documentation shall be retained for a period at least as long as that required for the subject electronic records and shall be available for agency review and copying.
>
> (g) Use of authority checks to ensure that only authorized individuals can use the system, electronically sign a record, access the operation or computer system input or output device, alter a record, or perform the operation at hand.
>
> (j) The establishment of, and adherence to, written policies that hold individuals accountable and responsible for actions initiated under their electronic signatures, in order to deter record and signature falsification.

```
-FAB     # Trans Indexed Sig Groups counter code 1 following group

E_T2_p83_gRSuAYvGhqV3S0JzYEF2dIa-OCPLbIhBO7Y    # trans prefix of signer for sigs

-EAB0AAAAAAAAAAAAAAAAAAAAAB    # sequence number of est event of signer's public keys for sigs

EwmQtlcszNoEIDfqD-Zih3N6o5B3humRKvBBln2juTEM      # digest of est event of signer's public keys for sigs

-AAD     # Controller Indexed Sigs counter code 3 following sigs

AA5267UlFg1jHee4Dauht77SzGl8WUC_0oimYG5If3SdIOSzWM8Qs9SFajAilQcozXJVnbkY5stG_K4NbKdNB4AQ # sig 0

ABBgeqntZW3Gu4HL0h3odYz6LaZ_SMfmITL-Btoq_7OZFe3L16jmOe49Ur108wH7mnBaq2E_0U0N0c5vgrJtDpAQ # sig 1

ACTD7NDX93ZGTkZBBuSeSGsAQ7u0hngpNTZTK_Um7rUZGnLRNJvo5oOnnC1J2iBQHuxoq8PyjdT3BHS2LiPrs2Cg # sig 2
```

Why verifiable audit logs?

1. CESR verifiable data structures are self-auditing
2. Embedded cryptographic primitives for hashes and/or signatures make it tamper evident and/or duplicity evident
3. Non-base-64 characters can be used to annotate CESR text to improve readability (e.g., add comments), yet are easily stripped for cryptographic processing.

Benefit #7

Using CESR enables low-friction compliance that makes it harder to commit fraud or cybercrimes

More questions for Sam?

For more about CESR, ACDC and KERI:

https://keri.one

Thank you!

May your keys be with you!