DIDCOMM SUPER STACK (DIDSS)

CREATING HIGHLY SCALABLE DIDCOMM AGENTS USING .NET, TRINITY, AND OKAPI WITH EASE (FT. VCTPS PROTOCOL)

MICHAEL HERMAN

TRUSTED DIGITAL WEB PROJECT

HYPERONOMY DIGITAL IDENTITY LAB

PARALLELSPACE CORPORATION

ALBERTA, CANADA

MWHERMAN@PARALLELSPACE.NET

Trusted Digital Web

trinsic-id/**okapi**

Okapi

2 Contributors | 0 Issues | 0 Stars | 0 Forks

microsoft/**GraphEngine**

Microsoft Graph Engine

17 Contributors | 63 Issues | 2k Stars | 316 Forks

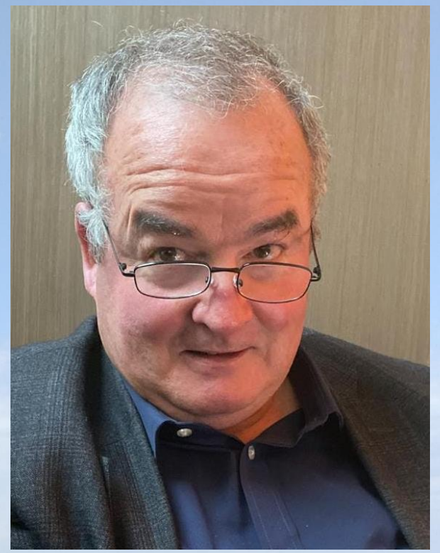
microsoft/**dotnet**

This repo is the official home of .NET on GitHub. It's a great starting point to find many .NET OSS...

309 Contributors | 412 Used by | 13k Stars | 2k Forks

MICHAEL HERMAN

SELF-SOVEREIGN BLOCKCHAIN ARCHITECT AND DEVELOPER
HYPERONOMY DIGITAL IDENTITY LAB
PARALLELSPACE CORPORATION
ALBERTA, CANADA



SINGULAR GOAL

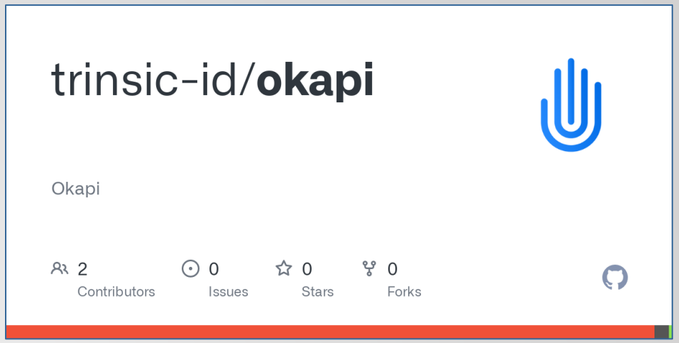
- Making decentralized agent design and development easier for C# and .NET developers

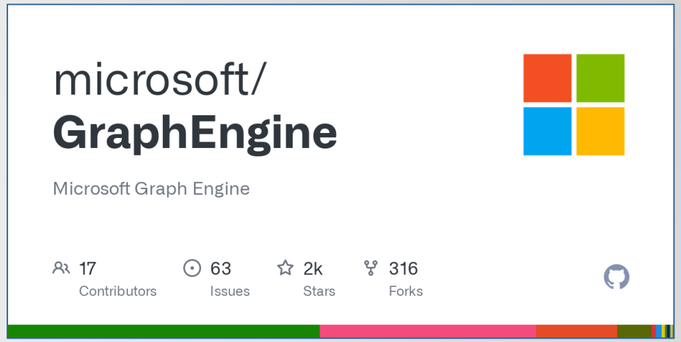
1. Model-based Automatic Code Generation
2. Structured Credential Model
3. Trinsic DID and DIDCOMM Libraries

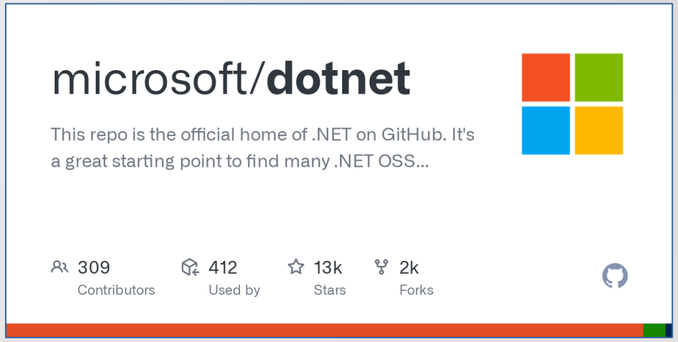
DIDCOMM SUPER STACK (DIDSS)

- Application framework for creating Verifiable Capability Authorization-enabled, highly scalable decentralized agents using .NET and DIDCOMM (featuring the VCTPS DIDCOMM Protocol)

Requirement	Open Source Project
3. DID key generation, DIDCOMM messaging, and credential signing	Trinsic-id Okapi Library
2. Decentralized agent modeling and code generation	Microsoft "Trinity" GraphEngine
1. Development and execution platform	Microsoft .NET Platform







MICROSOFT "TRINITY" GRAPHENGINE

- Trinity Specification Language (TSL)
  - Agent protocol and message structure definition
- TSL Codegen
  - Automatic code generation of complete C# projects
  - Objects, messages, and message handlers
  - Automatic JSON serialization/deserialization
- Very fast, highly optimized in-memory object graph database
- Highly performant and scalable clustered deployment model
- Simple, easy and efficient to use

trinsic-id/**okapi**

Okapi

2 Contributors | 0 Issues | 0 Stars | 0 Forks

microsoft/
GraphEngine

Microsoft Graph Engine

17 Contributors | 63 Issues | 2k Stars | 316 Forks

microsoft/**dotnet**

This repo is the official home of .NET on GitHub. It's a great starting point to find many .NET OSS...

309 Contributors | 412 Used by | 13k Stars | 2k Forks

TRINSIC-ID OKAPI API

- DIDKey
  - generate
  - resolve
- DIDCOMM
  - pack
  - unpack
  - sign
  - verify
- Oberon
  - create_key
  - create_token
  - blind_token
  - unblind_token
  - create_proof
  - verify_proof

trinsic-id/**okapi**

Okapi

2 Contributors | 0 Issues | 0 Stars | 0 Forks

microsoft/**GraphEngine**

Microsoft Graph Engine

17 Contributors | 63 Issues | 2k Stars | 316 Forks

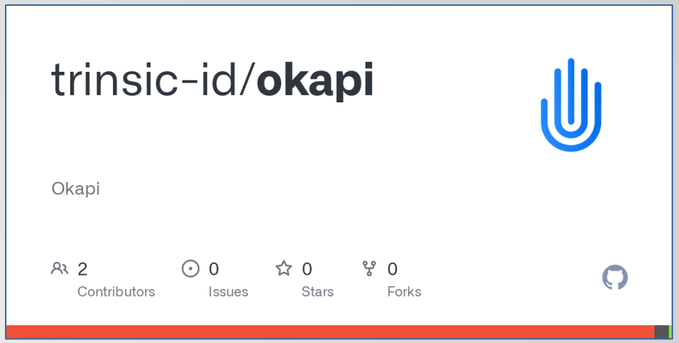
microsoft/**dotnet**

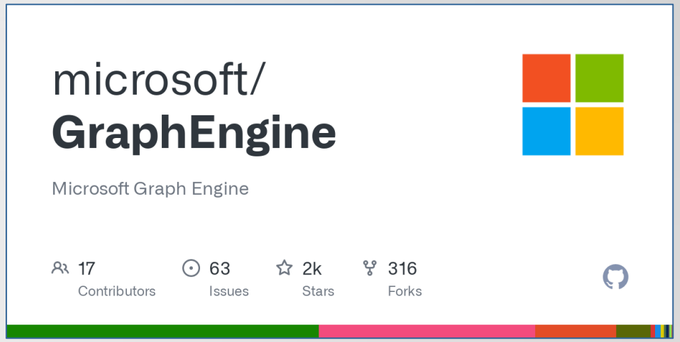
This repo is the official home of .NET on GitHub. It's a great starting point to find many .NET OSS...

309 Contributors | 412 Used by | 13k Stars | 2k Forks

MICROSOFT .NET PLATFORM

- Build and execution platform
- Visual Studio 2022 IDE
- Visual Studio Code w/GitHub integration
- C# version 10
- Microsoft Common Language Runtime (MSCLR)





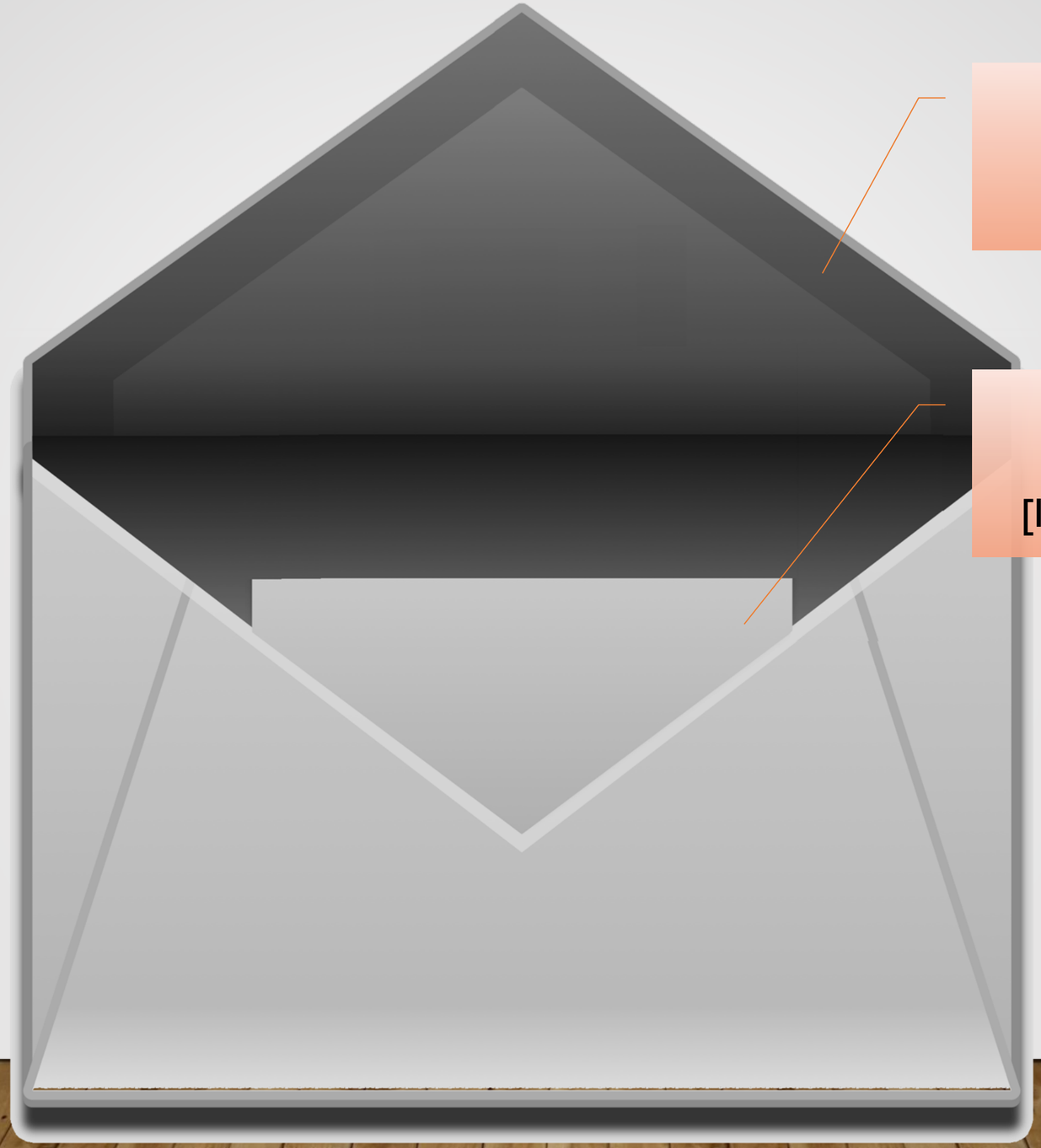
WHAT IS A STRUCTURED CREDENTIAL?

CREDENTIAL ENVELOPE
CREDENTIAL CONTENT
PACKING LABEL
ENVELOPE SEAL

Trusted
Digital
Web

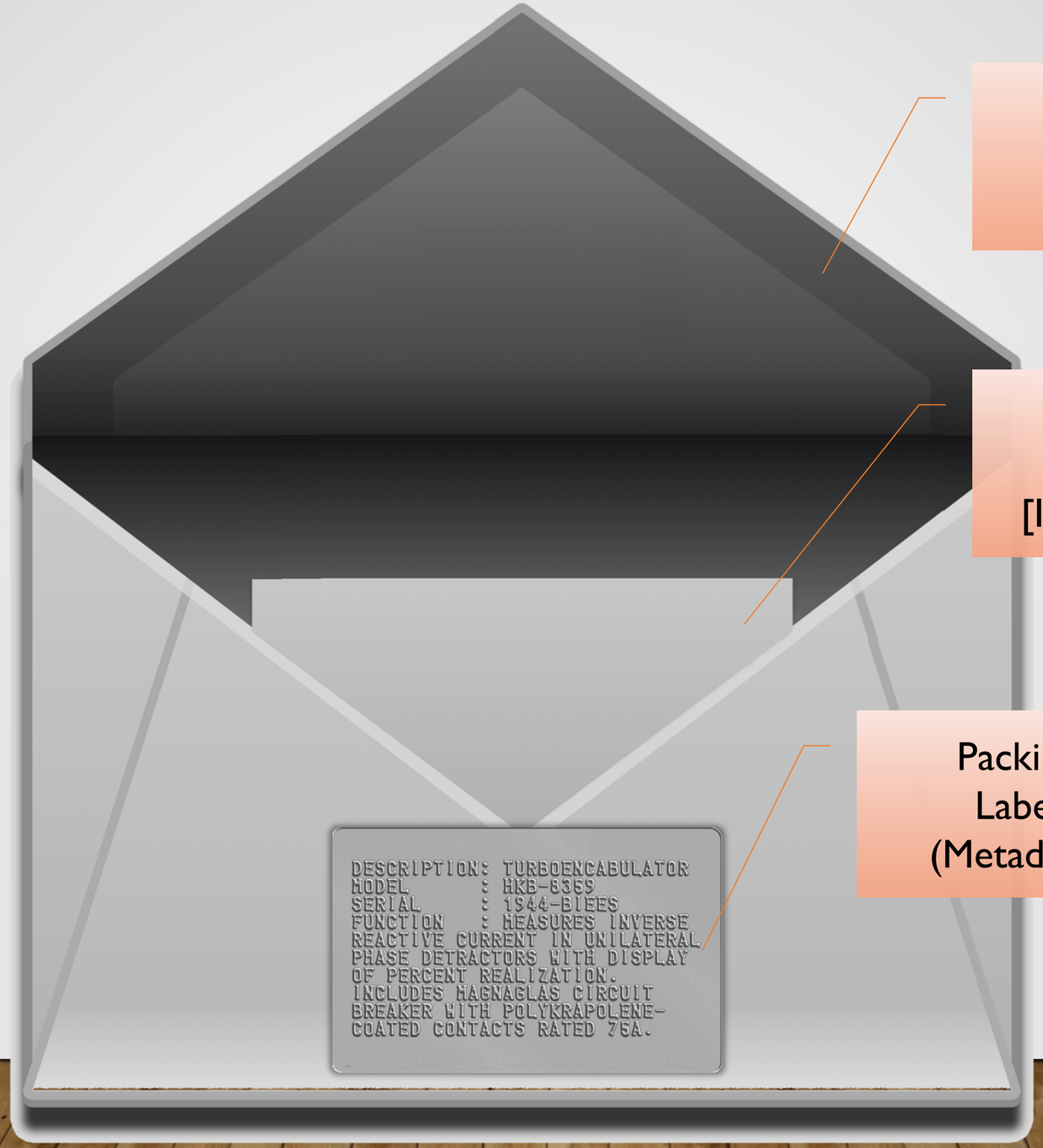
12

Credential Envelope (with ID)



Credential Envelope (with ID)

Credential Content [ID + Claims]



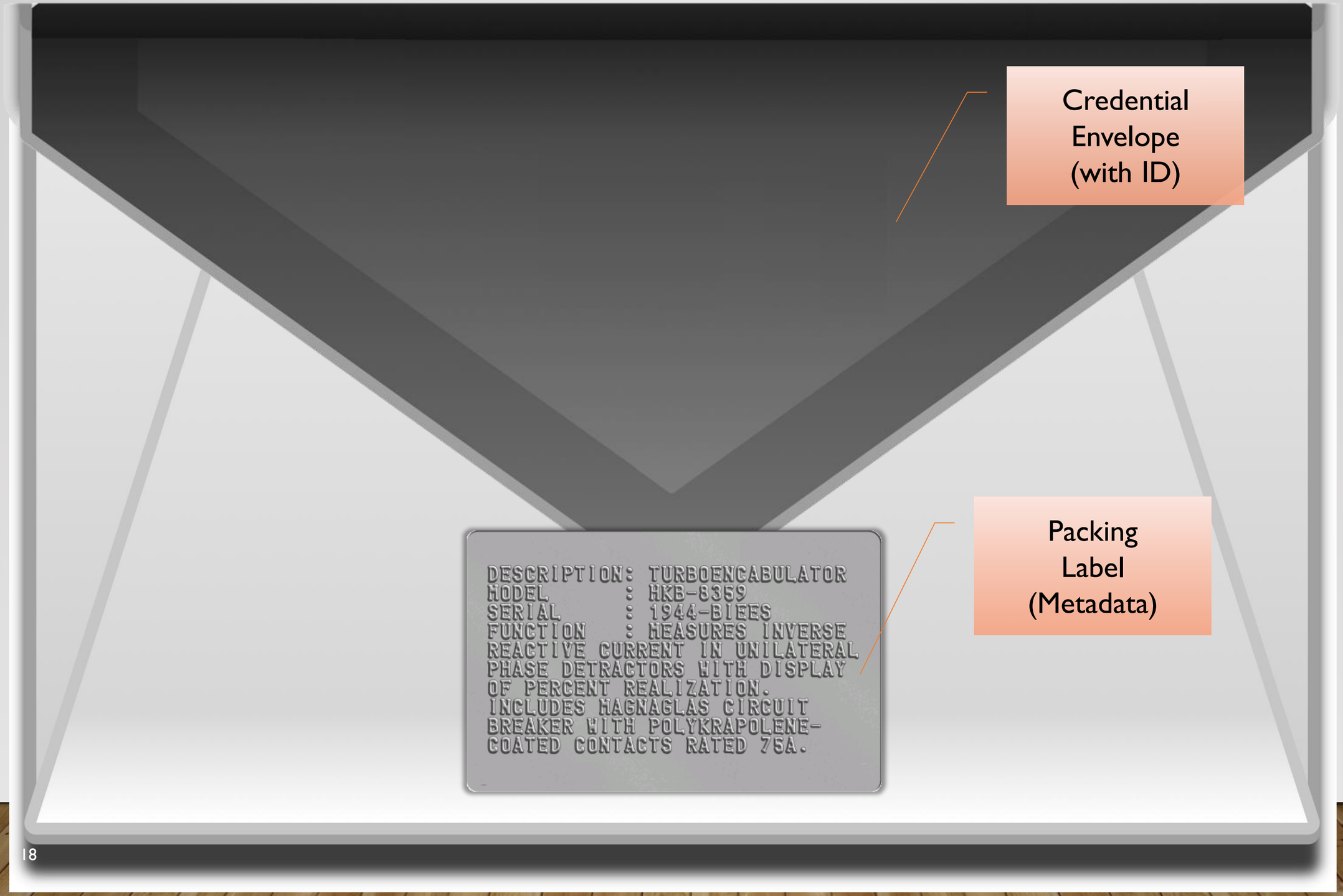
Credential Envelope (with ID)

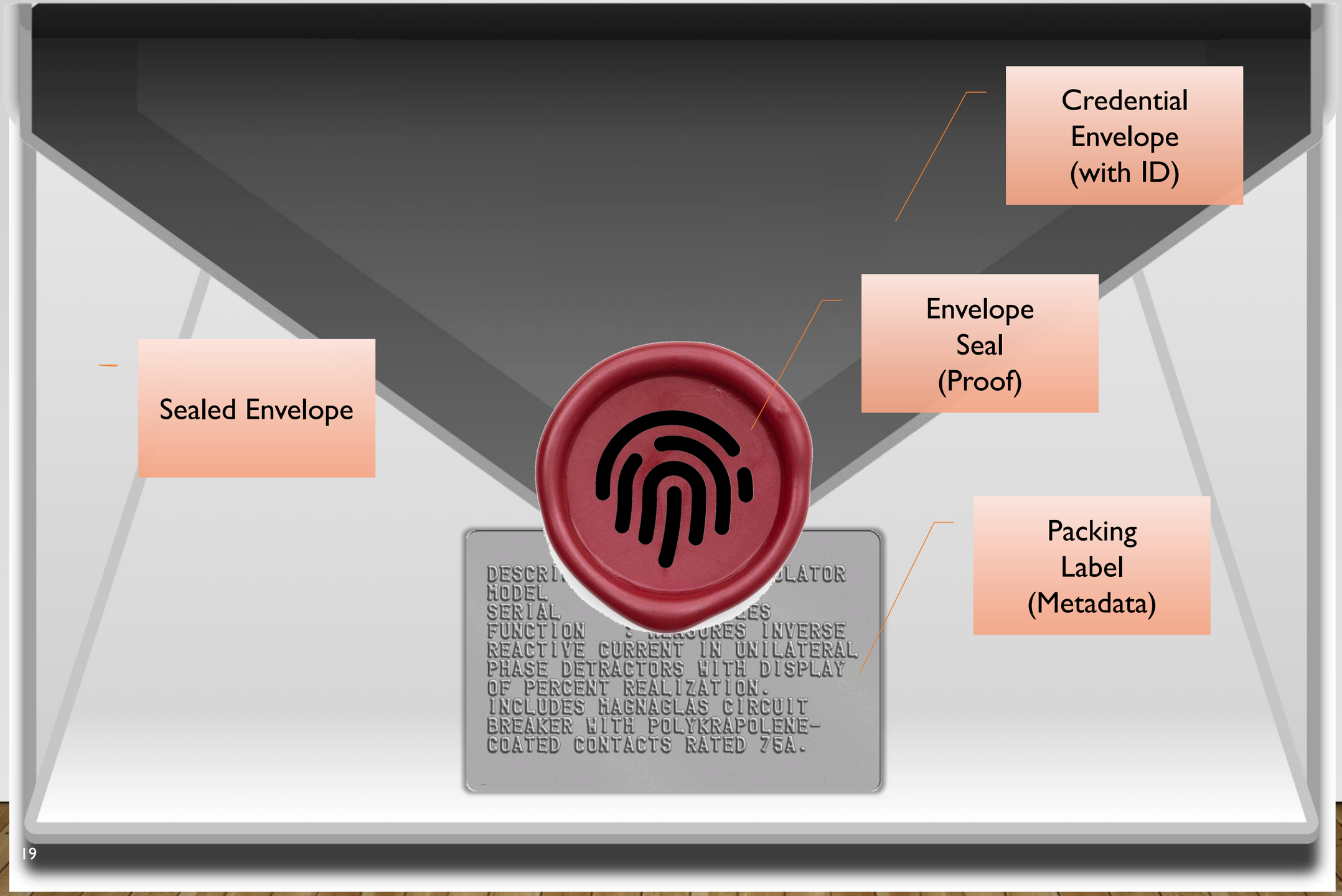
Credential Content [ID + Claims]

Packing Label (Metadata)

DESCRIPTION: TURBOENCABULATOR
MODEL : HKB-8359
SERIAL : 1944-BIEES
FUNCTION : MEASURES INVERSE
REACTIVE CURRENT IN UNILATERAL
PHASE DETRACTORS WITH DISPLAY
OF PERCENT REALIZATION.
INCLUDES MAGNAGLAS CIRCUIT
BREAKER WITH POLYKRAPOLENE-
COATED CONTACTS RATED 75A.


Credential
Envelope
(with ID)
DESCRIPTION: TURBOENCABULATOR
MODEL : HKB-8359
SERIAL : 1944-BIEES
FUNCTION : MEASURES INVERSE
REACTIVE CURRENT IN UNILATERAL
PHASE DETRACTORS WITH DISPLAY
OF PERCENT REALIZATION.
INCLUDES MAGNAGLAS CIRCUIT
BREAKER WITH POLYKRAPOLENE-
COATED CONTACTS RATED 75A.


Credential Envelope (with ID)
DESCRIPTION: TURBOENCABULATOR
MODEL : HKB-8359
SERIAL : 1944-BIEES
FUNCTION : MEASURES INVERSE
REACTIVE CURRENT IN UNILATERAL
PHASE DETRACTORS WITH DISPLAY
OF PERCENT REALIZATION.
INCLUDES MAGNAGLAS CIRCUIT
BREAKER WITH POLYKRAPOLENE-
COATED CONTACTS RATED 75A.
Packing Label (Metadata)


Credential Envelope (with ID)
Envelope Seal (Proof)
Sealed Envelope
Packing Label (Metadata)
DESCRI LATOR
MODEL
SERIAL
FUNCTION : MEASURES INVERSE
REACTIVE CURRENT IN UNILATERAL
PHASE DETRACTORS WITH DISPLAY
OF PERCENT REALIZATION.
INCLUDES MAGNAGLAS CIRCUIT
BREAKER WITH POLYKRAPOLENE-
COATED CONTACTS RATED 75A.

PRIMARY COLORS: BOUND CREDENTIAL EXAMPLE

C: > TDW > TDW.Documentation > {} primarycolors-unbound-vc2.json >

Credential Envelope

Packing Label

Credential Content

Claims

Envelope Seal

```
{
    "id": "did:colors:primarycolorpallette",
    "type": [
        "VerifiableCredential",
        "ColorPalette"
    ],
    "@context": [
        "https://www.w3.org/2018/credentials/v1",
        "https://www.w3.org/2018/credentials/examples/v1",
      {
        "@context": {
            "claims": {
                "@id": "https://example.com/claims",
                "@type": "@json"
            }
        }
    }],
    "credentialSubject": {
        "id": "did:colorpallettes:1234",
        "claims": {
            "colors": [
                "red",
                "green",
                "blue"
            ]
        }
    },
    "proof": {}
}
```


20

```
struct BTT_Invoice_EnvelopeContent
{
    string udid;
    List<string> context;
    optional string credentialsubjectudid; // bound credential
    Cac_Invoice claims;
    optional BTTEncryptedClaims encryptedclaims;
}
struct BTT_Invoice_Envelope
{
    string udid;
    BTTGenericCredential_PackingLabel label;
    BTT_Invoice_EnvelopeContent content;
}
struct BTT_Invoice_SealedEnvelope
{
    BTT_Invoice_Envelope envelope;
    BTTGenericCredential_EnvelopeSeal envelopeseal;
}
```

VCTPS PROTOCOL

Verifiable Capability Authorization (VCA) based

Secure Verifiable Credential Transport Protocol

trinsic-id/**okapi**

Okapi

2 Contributors | 0 Issues | 0 Stars | 0 Forks

microsoft/
GraphEngine

Microsoft Graph Engine

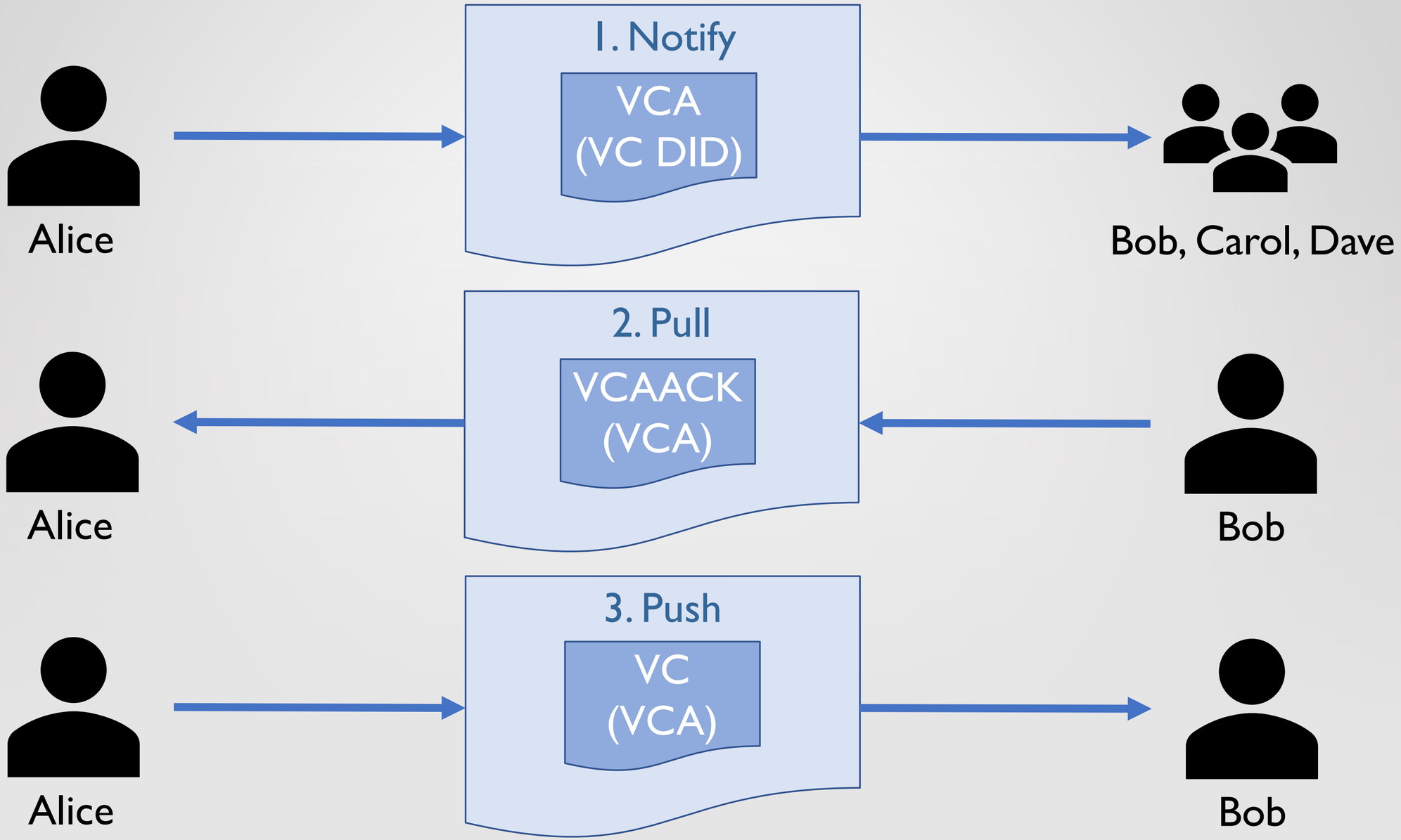
17 Contributors | 63 Issues | 2k Stars | 316 Forks

microsoft/**dotnet**

This repo is the official home of .NET on GitHub. It's a great starting point to find many .NET OSS...

309 Contributors | 412 Used by | 13k Stars | 2k Forks

VCTPS DIDCOMM PROTOCOL – SEQUENCE DIAGRAM



TRUSTED DIGITAL WEB AND VCTPS PROTOCOL MODELS

Enterprise Architecture Domains	Extended OSI Model	Trusted Digital Web	VCPTS Protocol
Business Domain/Architecture	Business	did:bluetoquebizdoc	BPMN Workflows
Application Domain/Architecture	Application	did:vca + did:vcaack	VCTPS Protocol
	Presentation	did:key + did:object	DIDCOMM Messaging
Technology/Infrastructure Domain/Architecture	Session	Trinity Graph Engine	REST/HTTP
	Transport	TCP & UDP Protocols	TCP Protocol
	Network	IP (Logical Addressing)	IP (Logical Addressing)
	Data Link	Physical Addressing (MAC)	Physical Addressing (MAC)
	Physical	Binary Transmission	Binary Transmission

DEMOS

trinsic-id/**okapi**

Okapi

2 Contributors | 0 Issues | 0 Stars | 0 Forks

microsoft/
GraphEngine

Microsoft Graph Engine

17 Contributors | 63 Issues | 2k Stars | 316 Forks

microsoft/**dotnet**

This repo is the official home of .NET on GitHub. It's a great starting point to find many .NET OSS...

309 Contributors | 412 Used by | 13k Stars | 2k Forks

DEMOS

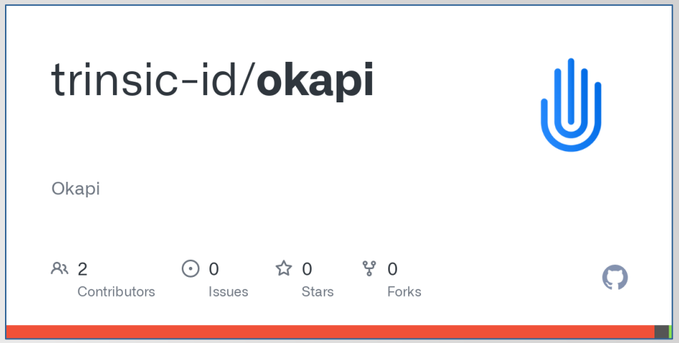
- Prototype1 Goals
  - Create a DIDCOMM agent that accepts encrypted DIDCOMM messages
  - Model the message structures, DIDCOMM protocols, and server implementation using the Trinity Specification Language (TSL)
  - Use automatic code generation to build the message structures and server implementation using C#
  - Send an "empty" DIDCOMM message to the DIDCOMM Agent
- Prototype 2 Goals
  - Use the Trinsic-id Okapi library to create an encrypted DIDCOMM message containing a (dummy) Verifiable Capability Authorization (VCA) resembling a Notify message in the VCTPS Protocol
  - Send the Notify VCTPS Protocol message (w/multiple recipients) to the Prototype 1 DIDCOMM agent
  - Have the DIDCOMM agent receive, accept, and decrypt the Notify message
  - Show a simple example of saving a Trinity structure (cell) to Trinity Local Storage

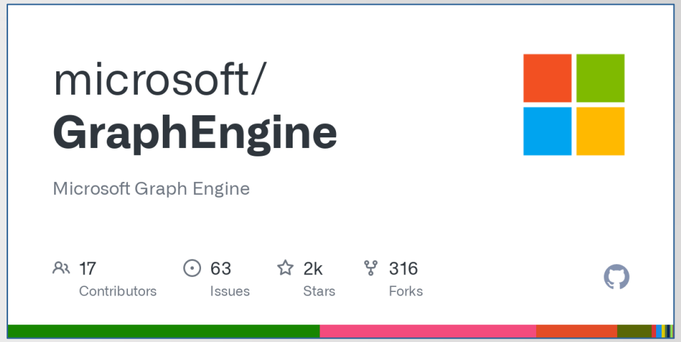
DEMOS

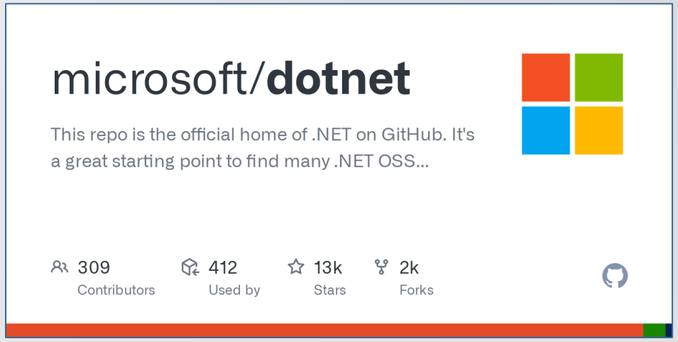
- Prototype 5 Goals
  - Structured Credential Programming Model using TSL autogenerated entities (C# classes)
  - Model Verifiable Capability Authorization (VCA) using TSL (and VCA ACK)
  - Replace dummy VCA used in Prototype2 with the VCTPS Protocol model expressed using TSL
  - Create helper methods for VCAs (as well as VCAACKs and UBL Invoices, Parties, and Items)
  - Create a VCA VC

RESOURCES

- Trinsic-id Okapi DID Libraries
  - GitHub: https://github.com/trinsic-id/okapi
  - Examples: https://github.com/trinsic-id/okapi/tree/main/dotnet/Tests/Okapi.Tests
- Microsoft "Trinity" GraphEngine
  - GitHub: https://github.com/microsoft/GraphEngine
  - Documentation: https://www.graphengine.io/docs/manual/index.html
  - Examples: https://www.graphengine.io/docs/manual/DemoApps/index.html
- DIDCOMM Super Stack (DIDSS)
  - Examples: https://github.com/mwherman2000/VCTPSPrototypes
- Structured Credential Model
  - https://www.youtube.com/playlist?list=PLU-rWqHm5p45dzXF2LJZjuNVJrOUR6DaD







QUESTIONS?

MICHAEL HERMAN
TRUSTED DIGITAL WEB PROJECT
HYPERONOMY DIGITAL IDENTITY LAB
PARALLELSPACE CORPORATION
ALBERTA, CANADA

MWHERMAN@PARALLELSPACE.NET

Trusted Digital Web