



Decentralized Identity and Zero Trust Enforcement for Industrial Robot Maintenance

Howest University of Applied Sciences

sirris

howest
cyber 3 lab

Interreg Fr-Wa-Vl - Secuweb

- Oct. 2024 - Sept. 2028
- Focus: next generation of the internet
- Technologies: 5G, AI, Blockchain, IoT, Quantum, Solid
- Domains: Food, E-Health, Industry 4.0, Mobility
- Technology Building Blocks, Demonstrators, Use Cases
- CITC, Eurasanté, Howest, Sirris, TUA West, UC Louvain, UGent, UPHF

Interreg



Cofinancé par
l'Union Européenne
Medegefinancierd door
de Europese Unie

France - Wallonie - Vlaanderen



A common industrial challenge

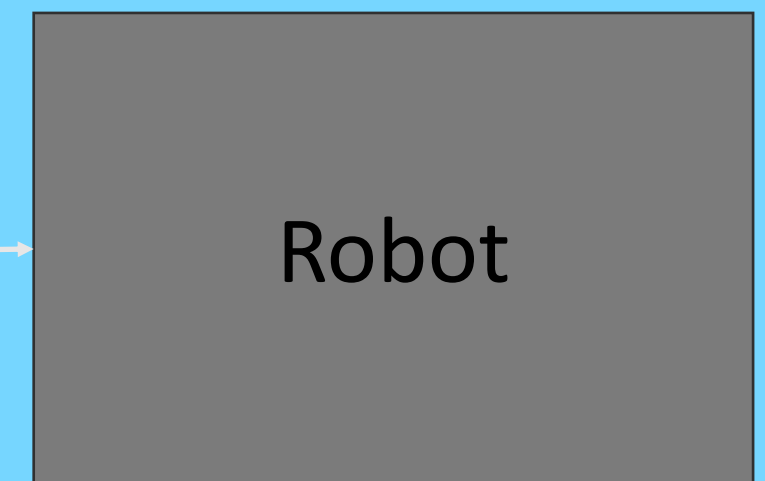
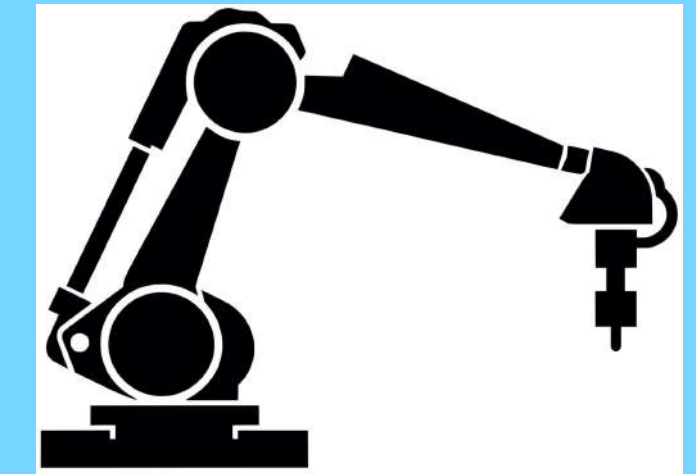
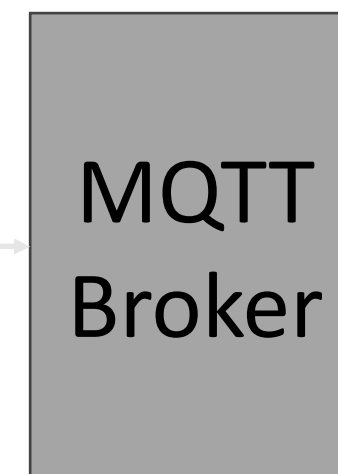
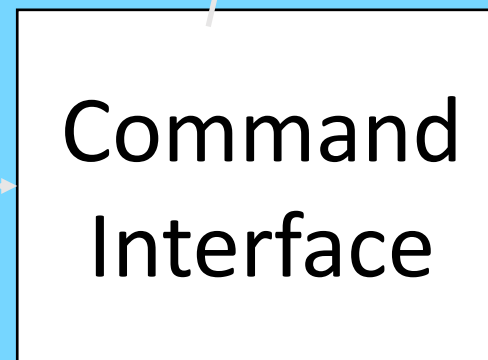
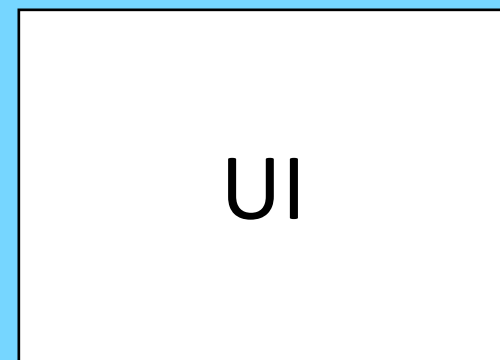
On **Acme's** factory floor, a **robotic arm** from **Robotics Inc.** is a key asset.

Access is tightly controlled within the industrial environment, with networked sensors and security zones protecting critical systems.



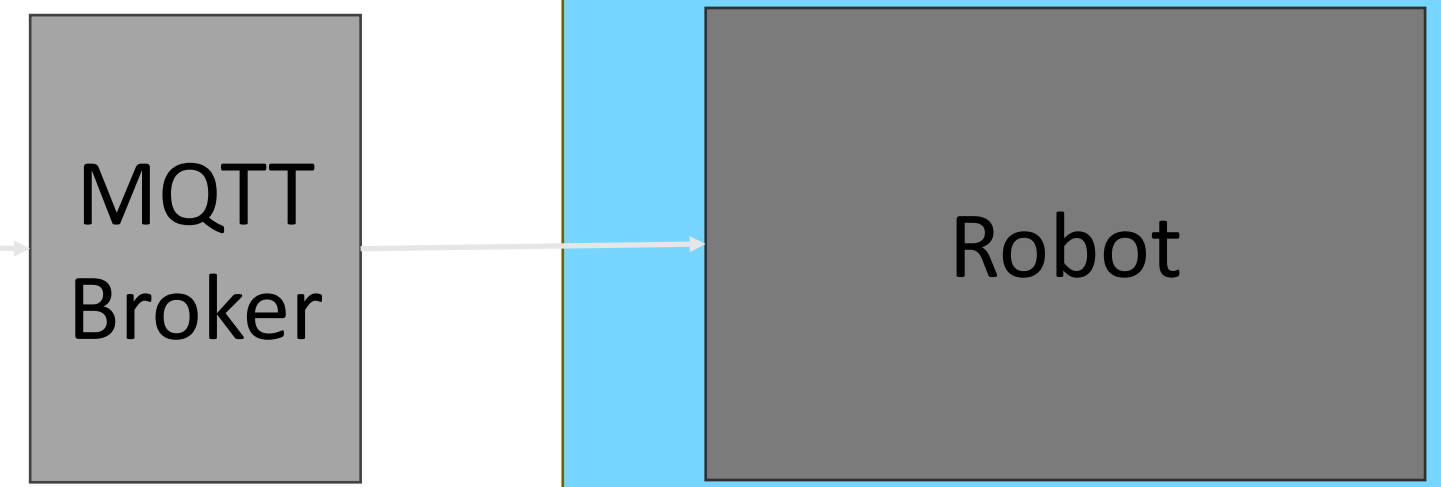
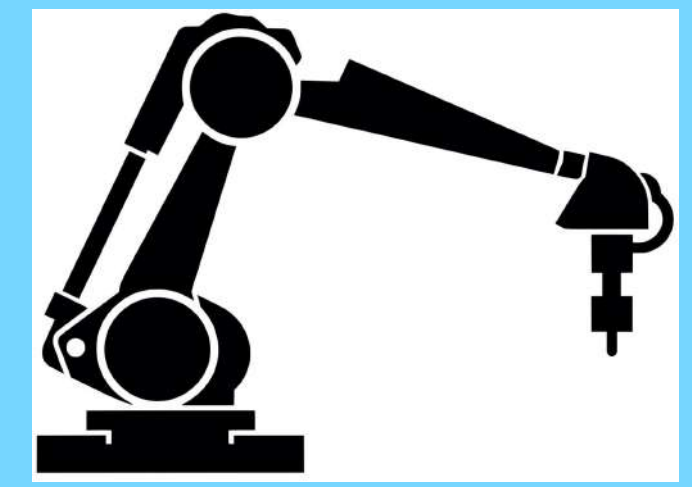
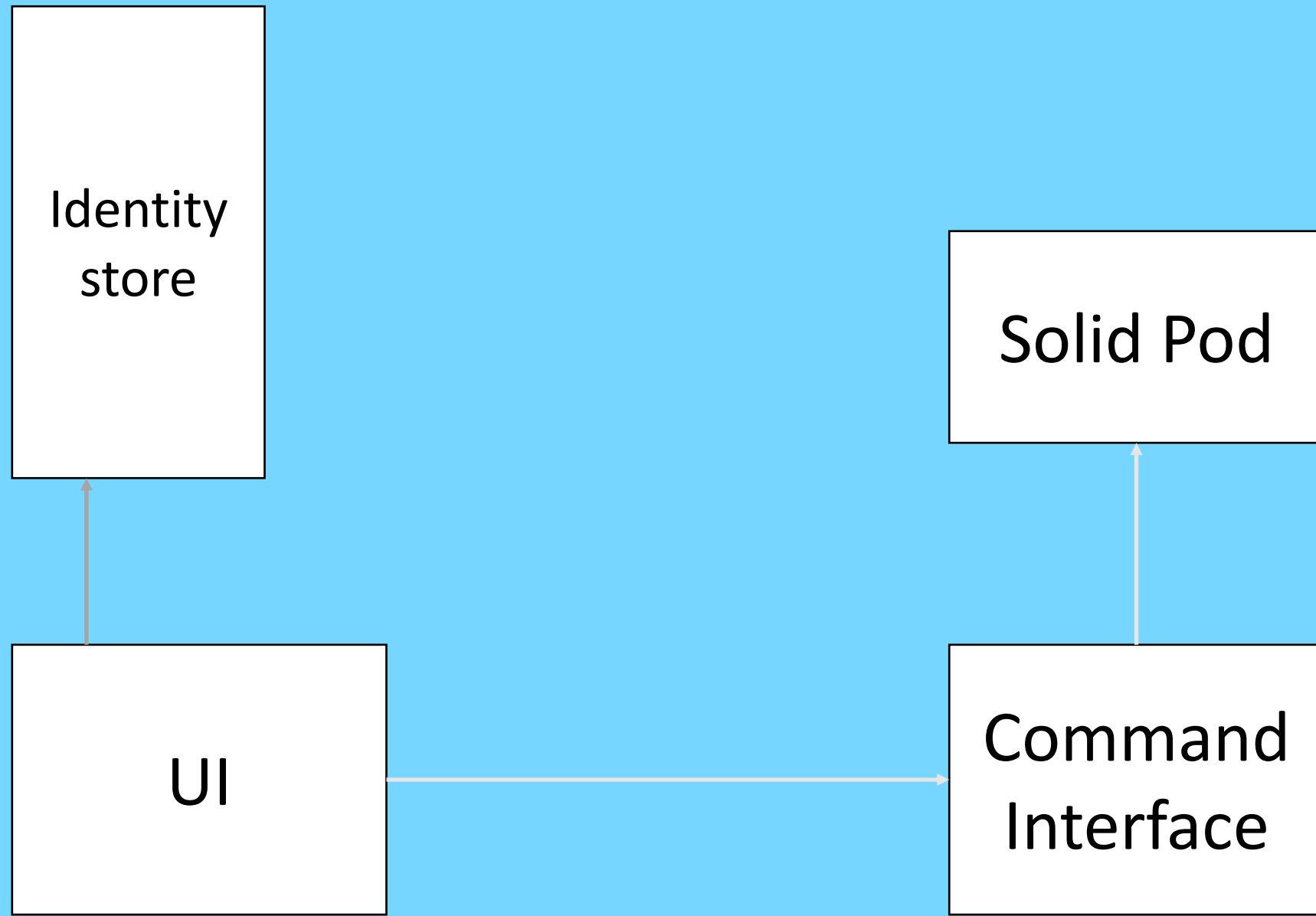
Internal Network 1

Internal Network 2



Internal Network 1

Internal Network 2

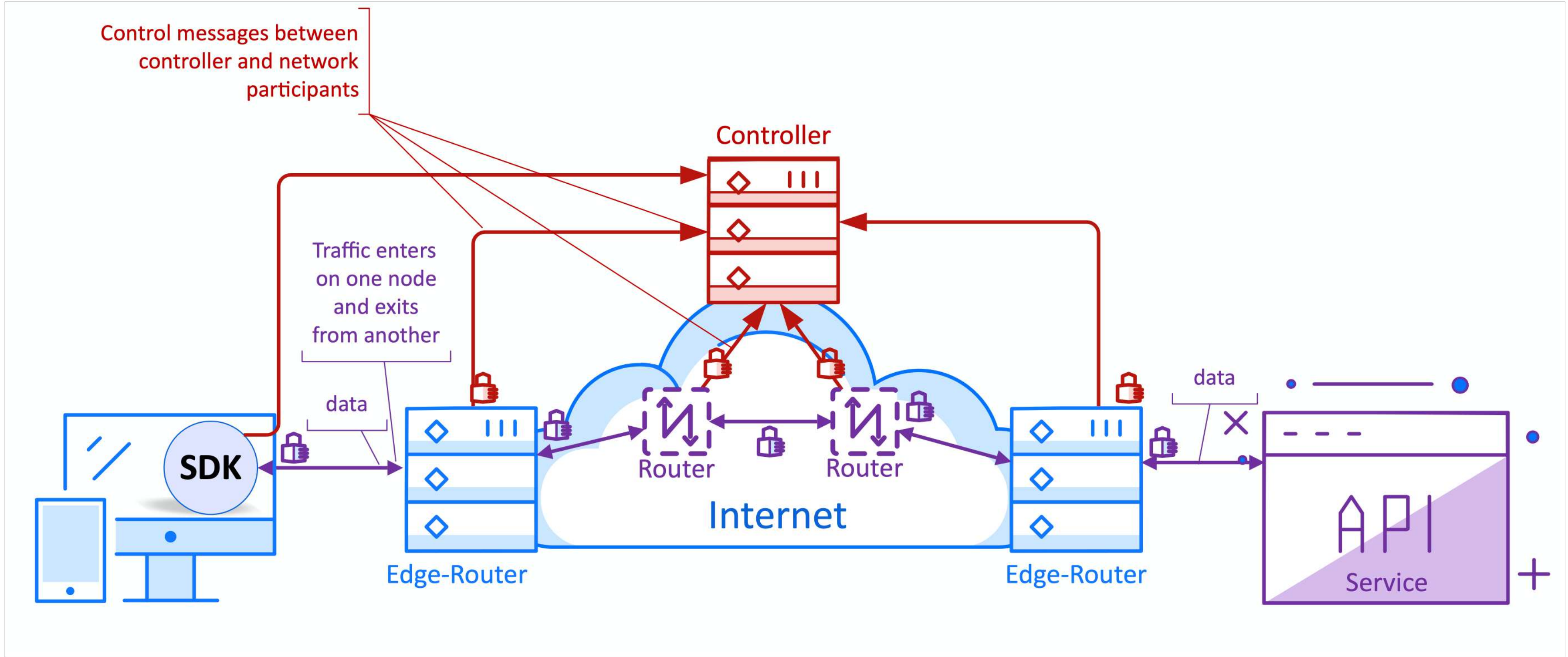


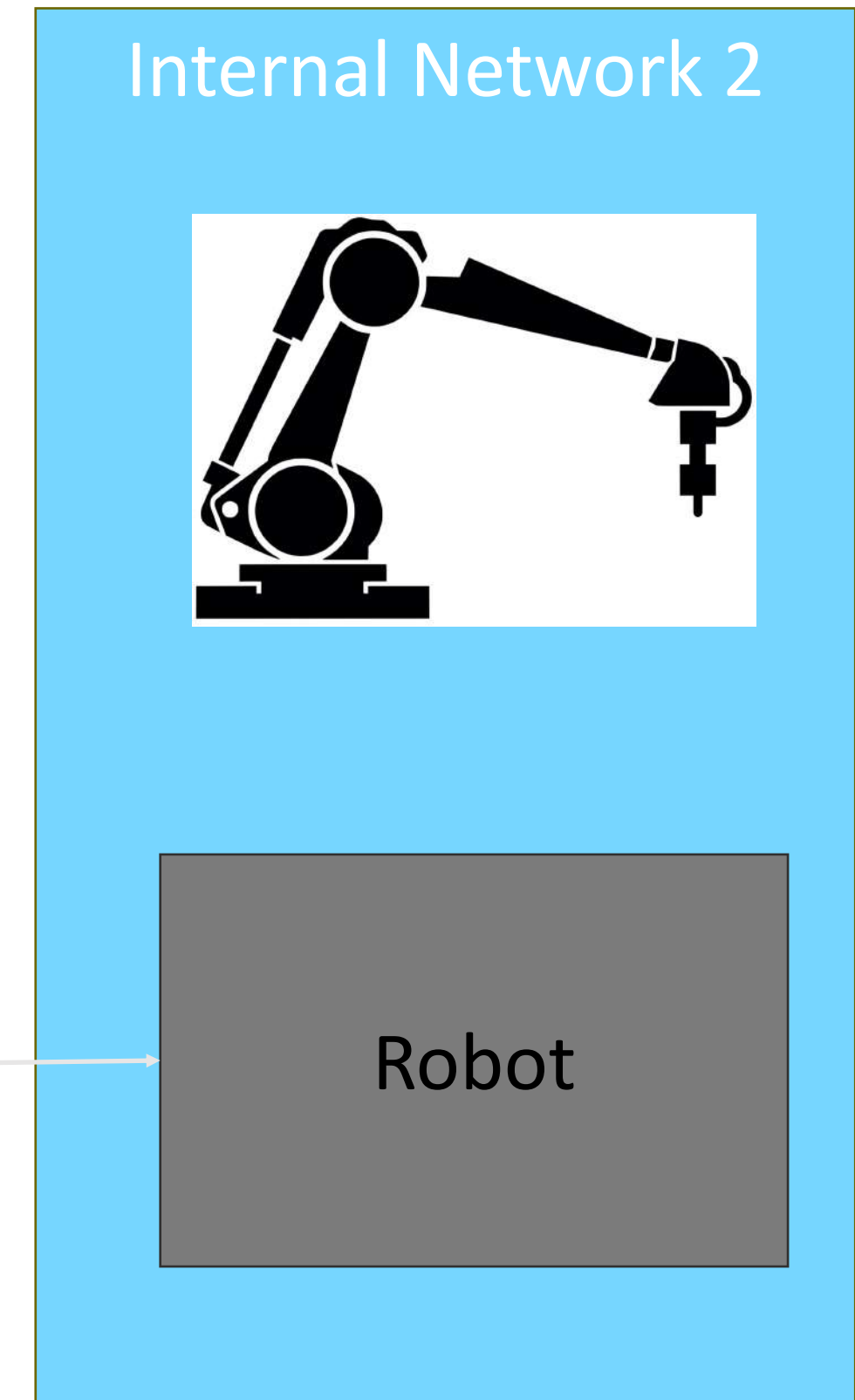
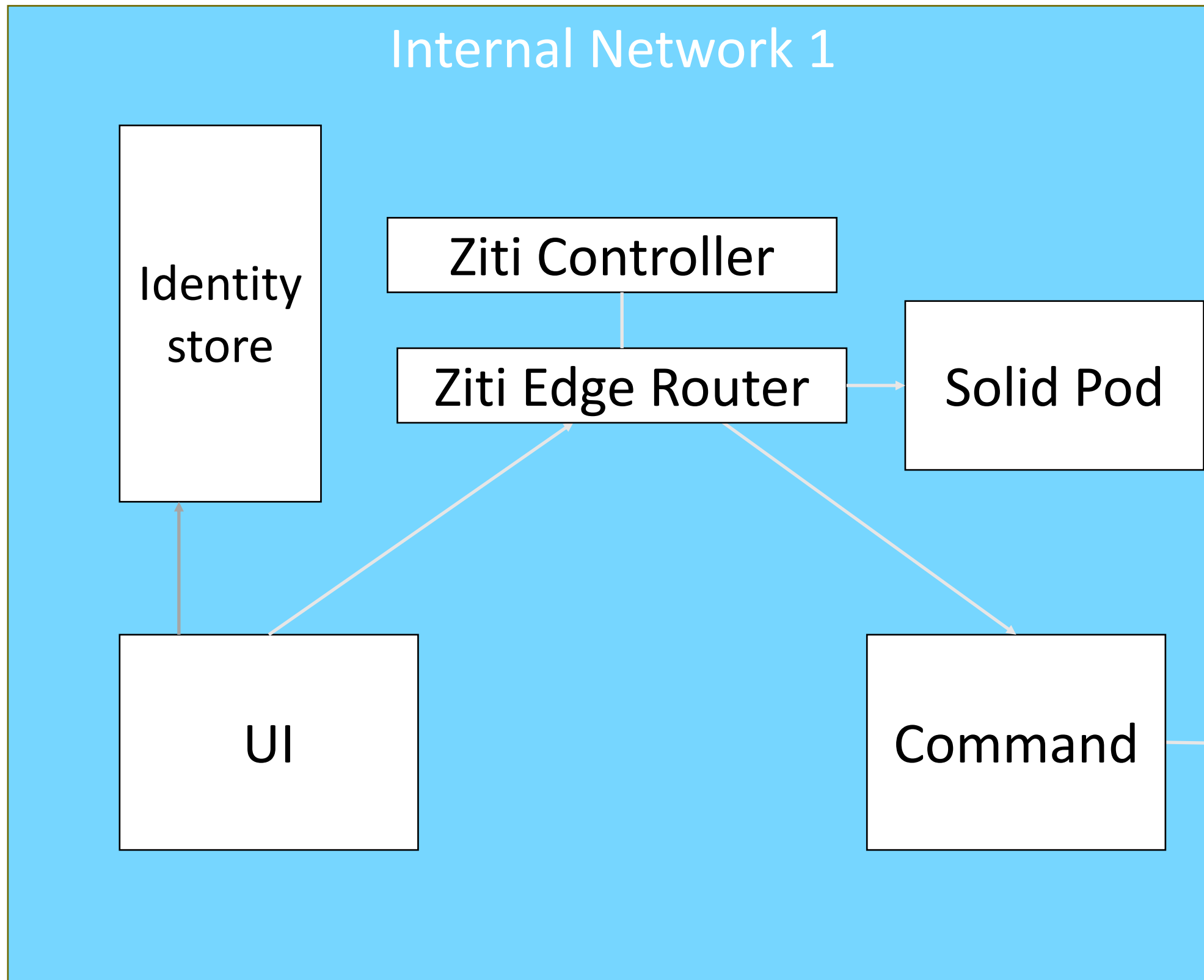
Zero Trust

Zero Trust is a security framework based on the principle:
“**Never trust, always verify.**”

It requires **strict identity verification** for **every user or device** trying to access resources, regardless of location.

This mitigates risks of insider threats, compromised credentials, and lateral movement within networks.





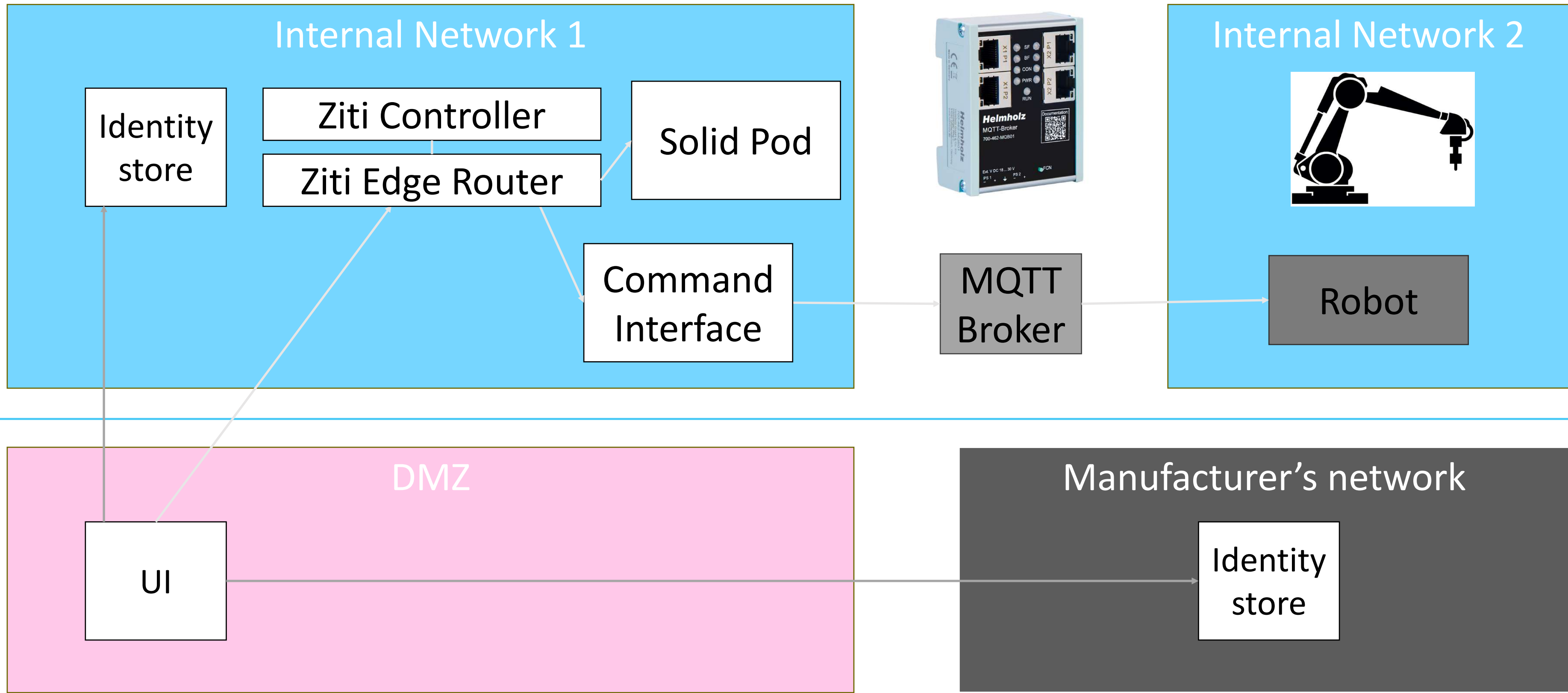
Meet Alice, an External Technician

Alice arrives claiming to be a technician from **Robotics Inc.** to service the robot.

But can we really **trust** her identity?

If Alice can impersonate a trusted technician, she could gain unauthorized access to critical systems.



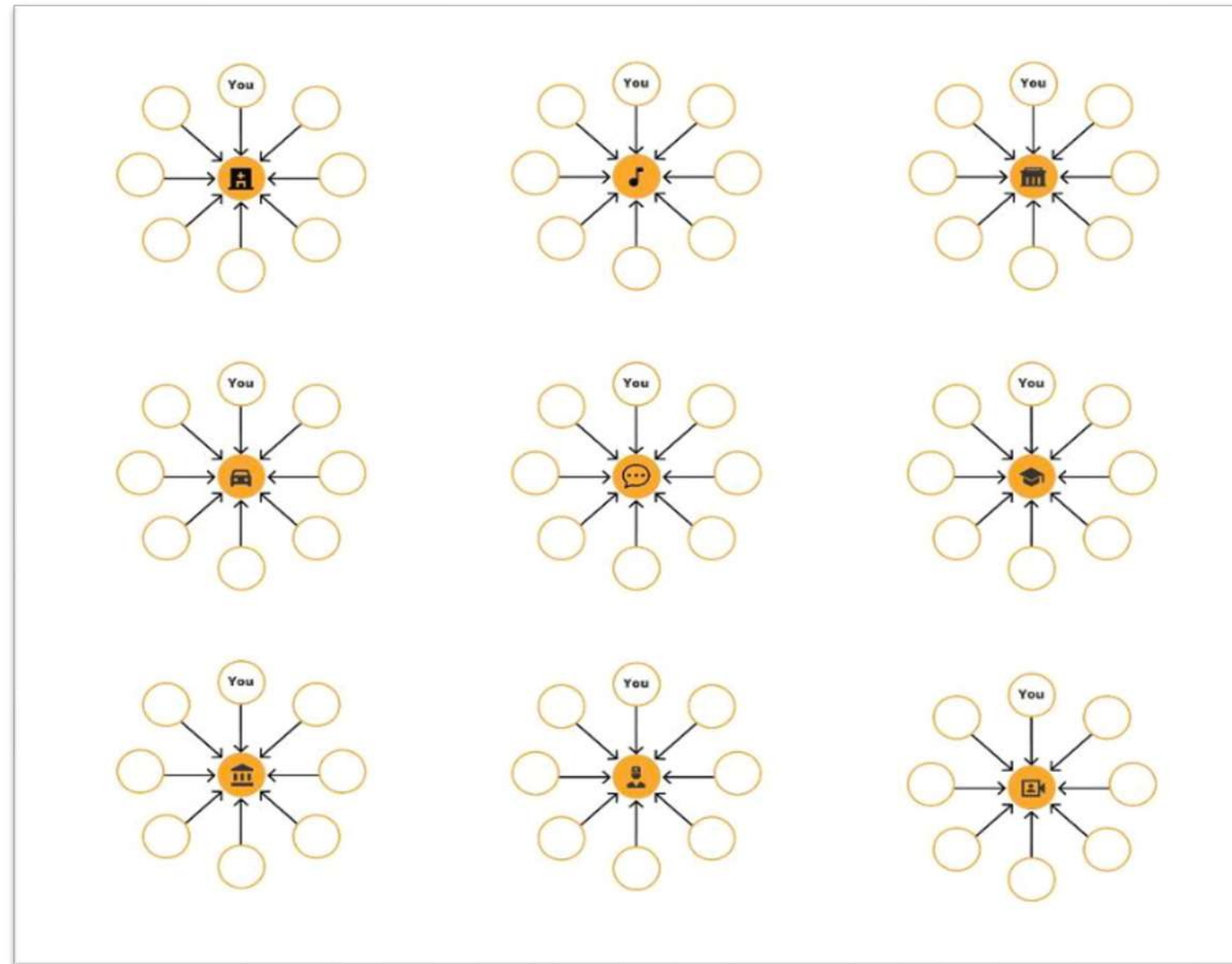


This is troublesome

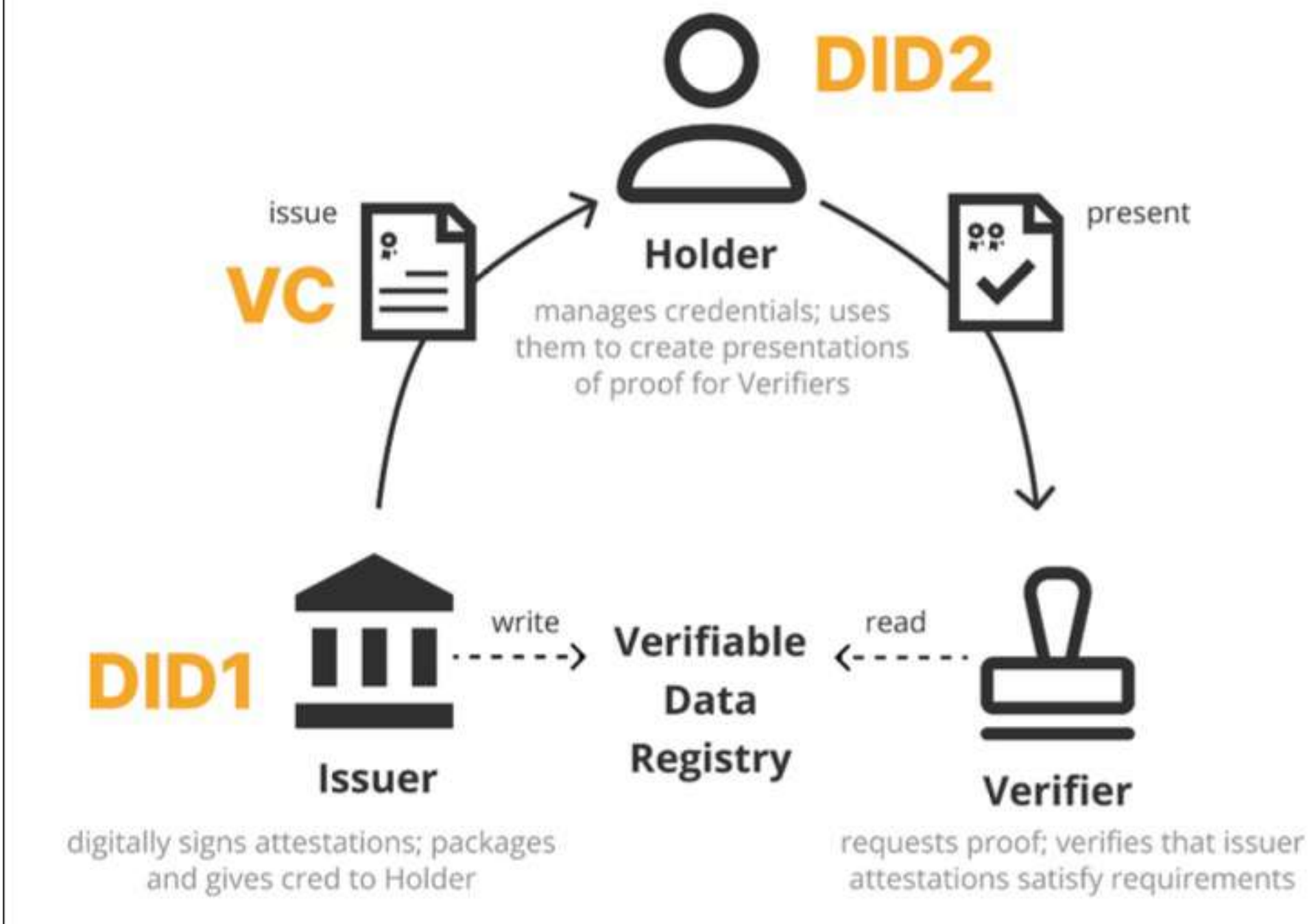
This doesn't scale well.

- Requires **custom integrations** with each external supplier's system
- Depends on **uptime** of external system
- Depends on **security** of external system

What if we approached this differently?



Self Sovereign Identity



Verifiable Credential & Verifiable Presentation

Verifiable Credential (VC)

```
{
  "@context": ["https://www.w3.org/2018/credentials/v1", ... ],
  "type": ["VerifiableCredential", "ComcastCredential"],
  .....

  // (1) ID of Credential Issuer.
  // Resolved to machine-readable info about the issuer, e.g. Public Keys
  "issuer": "did:example:565049",
  ...

  // (2) claims about the subject of the credential (holder)
  "credentialSubject": {
    // ID of credential subject (holder)
    // Resolved to machine-readable info about the holder, e.g. Public Keys
    "id": "did:example:ebfeb1f712ebc6f1c276e12ec21",
    "name": "John",
    "contractNum": 123456,
    .....
  },

  // (3) digital proof that makes the credential tamper-evident
  "proof": {
    "type": "Ed25519Signature2020",
    "created": "2022-02-25T14:58:42Z",
    "verificationMethod": "did:example:565049#key-1",
    "proofPurpose": "assertionMethod",
    "proofValue": "z3FjecWufY46yg5LhxhueARiKBk9czhSePTFehP..."
  }
}
```

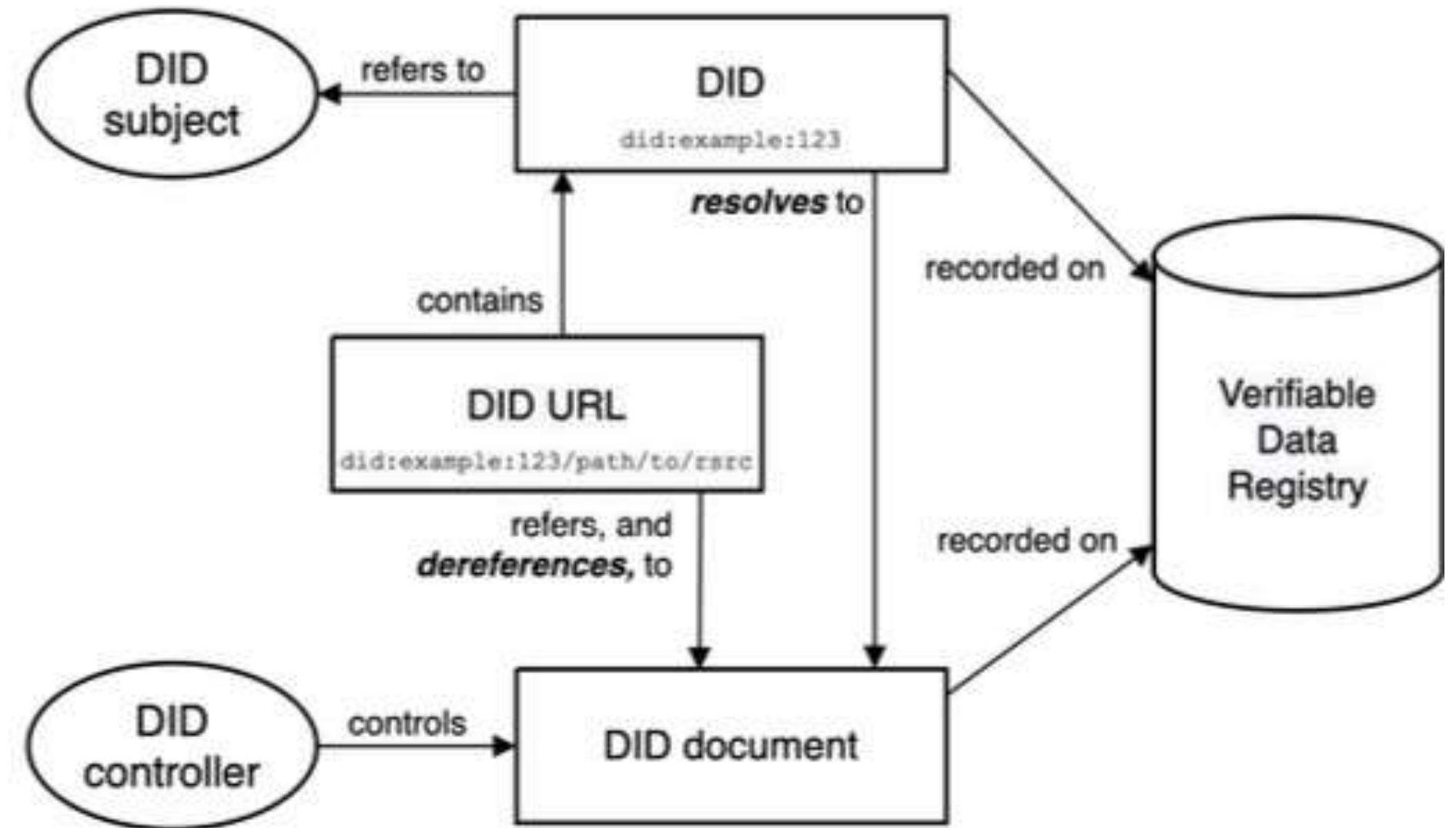
Verifiable Presentation (VP)

```
{
  "@context": ["https://www.w3.org/2018/credentials/v1", ... ],
  "type": "VerifiablePresentation",

  // (1) Verifiable Credential, see example at the left.
  "verifiableCredential": [{
    "issuer": "did:example:565049",
    ....
    "credentialSubject": {
      "id": "did:example:ebfeb1f712ebc6f1c276e12ec21",
      "name": "John",
      "contractNum": 123456,
      .....
    },
    "proof": { .... }
  }],

  // (2) digital signature by the credential holder
  // (proof of key possession)
  "proof": {
    "type": "Ed25519Signature2020",
    "created": "2022-03-25T16:37:21Z",
    "verificationMethod": "did:example:ebfeb1f712ebc6f1c276e12ec21#key-1",
    "proofPurpose": "assertionMethod",
    "proofValue": "f3FjecWufY46yg5Lhxhue244Jsdfs,PsdgIO2d..."
  }
}
```

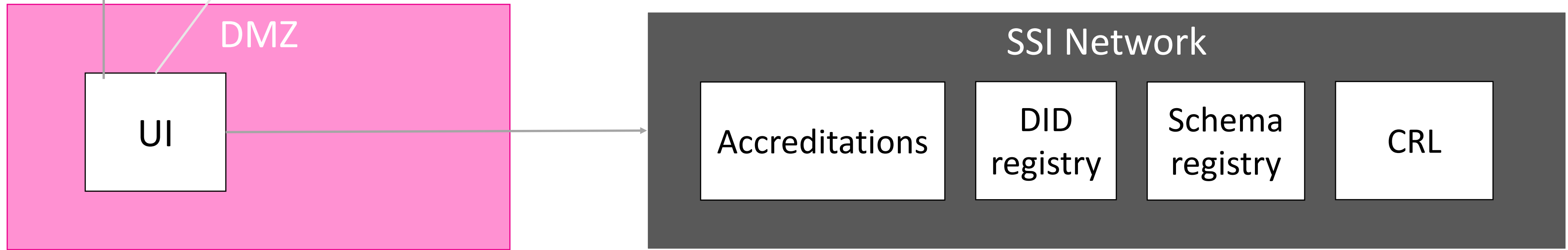
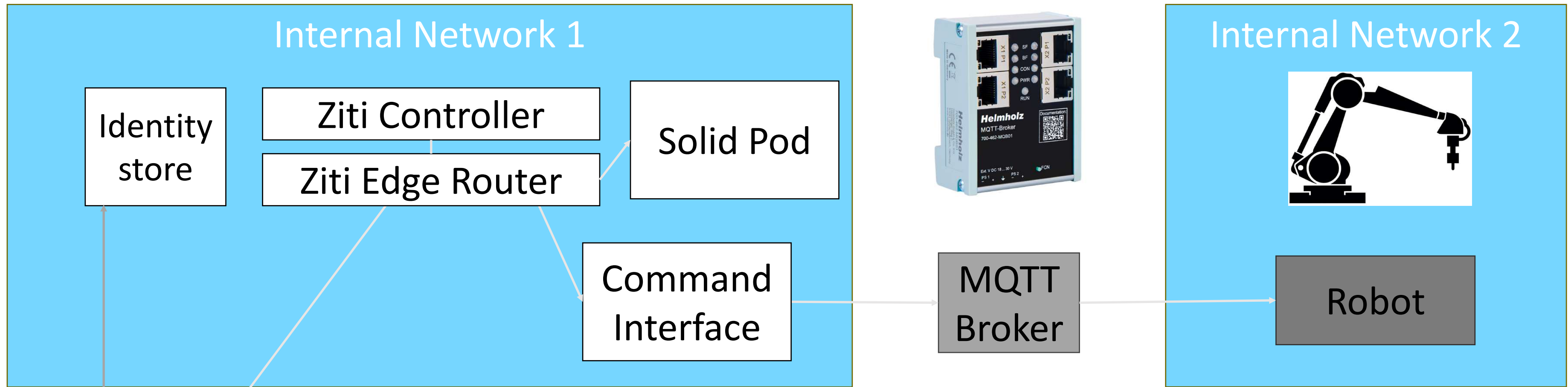
Decentralized Identity



Trusted Registry

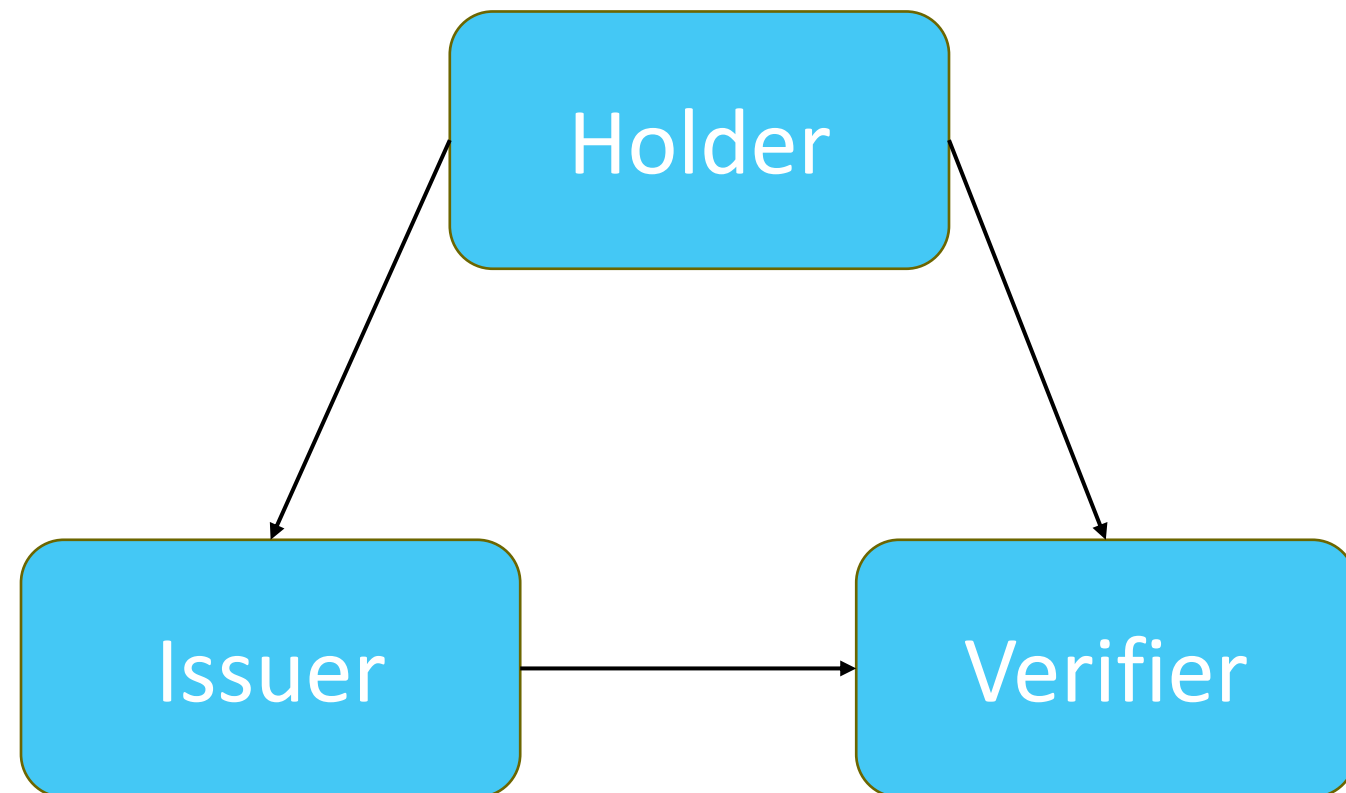
Contains:

- DIDs (Decentralized identities)
- VC Schemas
- Accreditations

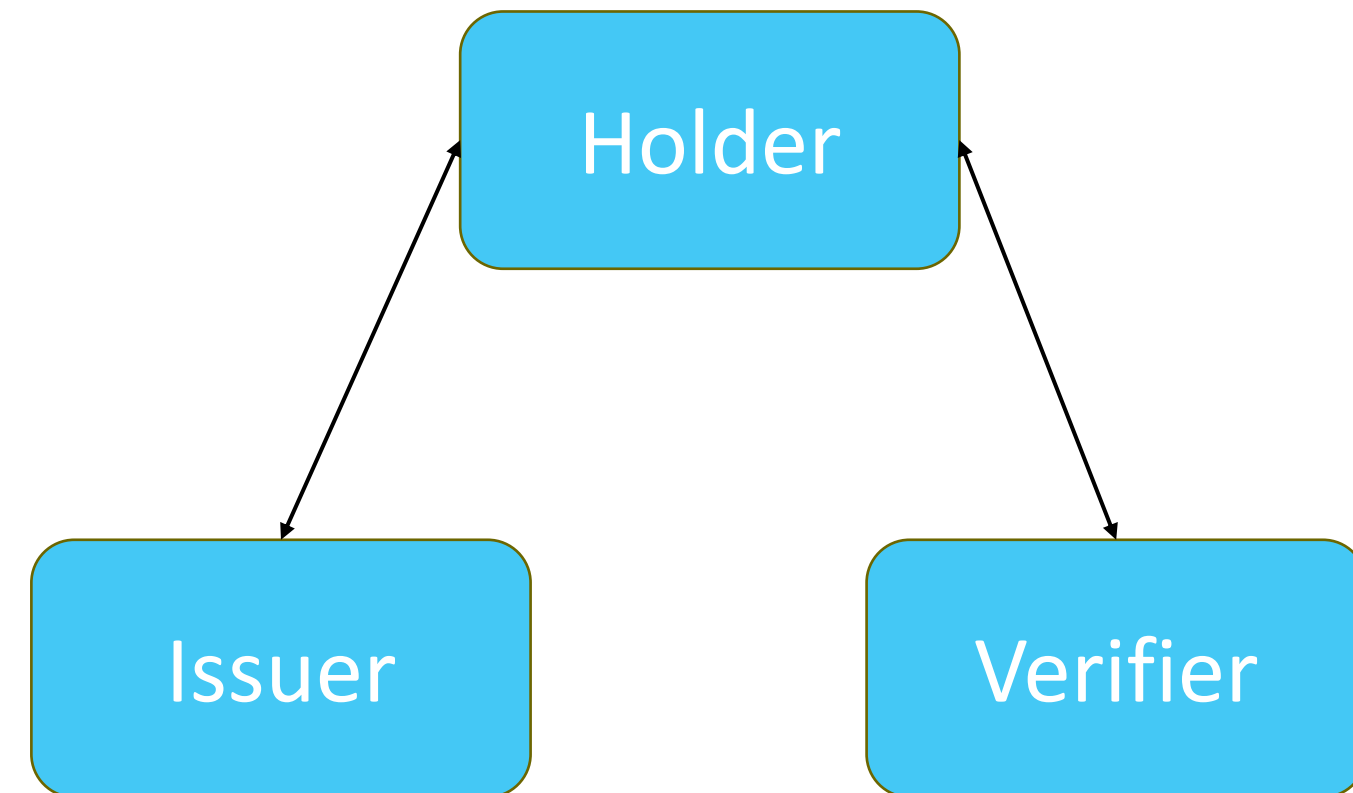


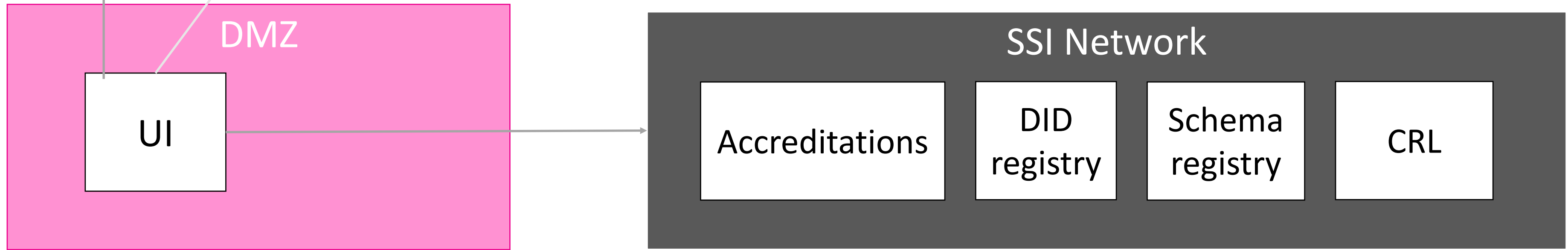
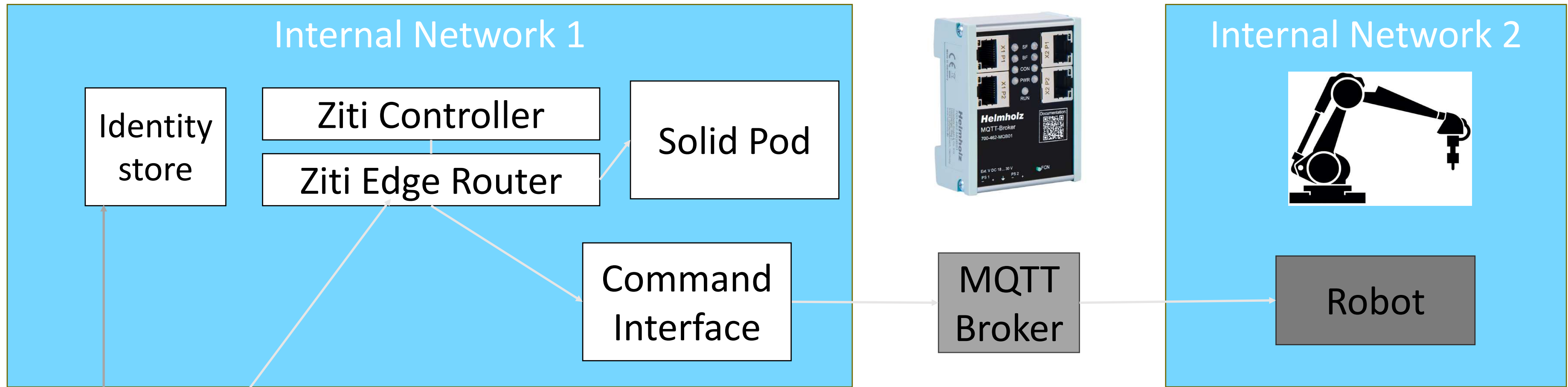
OIDC (OpenID Connect) vs. OIDC4VP

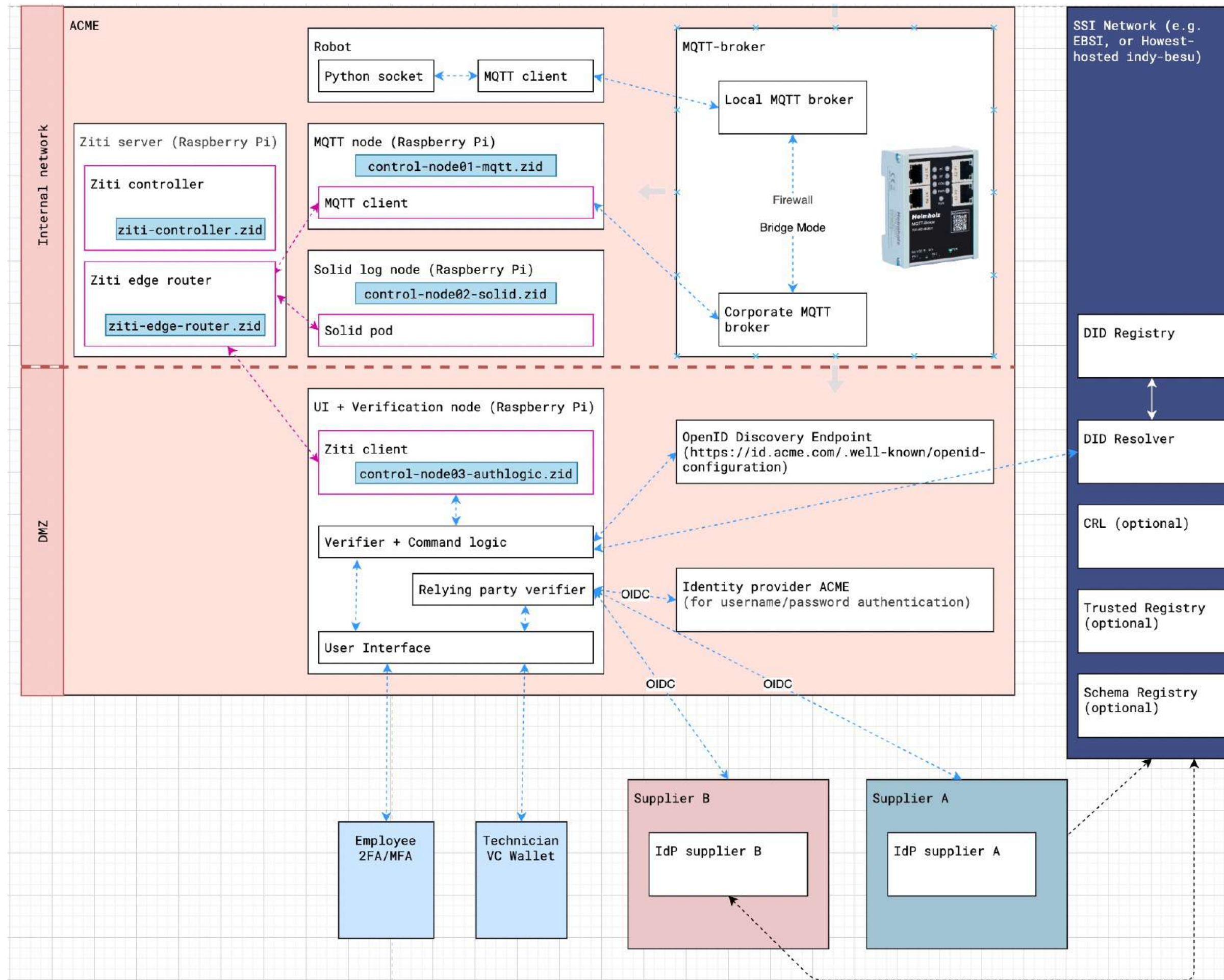
OIDC



OIDC4VP







Security Benefits of Layered Approach

- Enforces strict **identity verification** on all users, internal or external
- Provides strong **auditability** and **revocation capabilities** for **external identities**
- Aligns with **Zero Trust** by never assuming implicit trust based on network location or user origin

Key Takeaways

- Industrial environments face unique challenges managing identities for both **internal** and **external users**.
- Traditional systems like **LDAP/OIDC** work well **internally** but fall short for external technicians.
- Combining **Self-Sovereign Identity** with trust registries enables scalable, secure **external** identity verification.
- Layered identity management strengthens security and supports Zero Trust principles.
- This approach integrates seamlessly with legacy protocols and infrastructure, enabling practical, future-ready solutions.

Thank you for your attention!