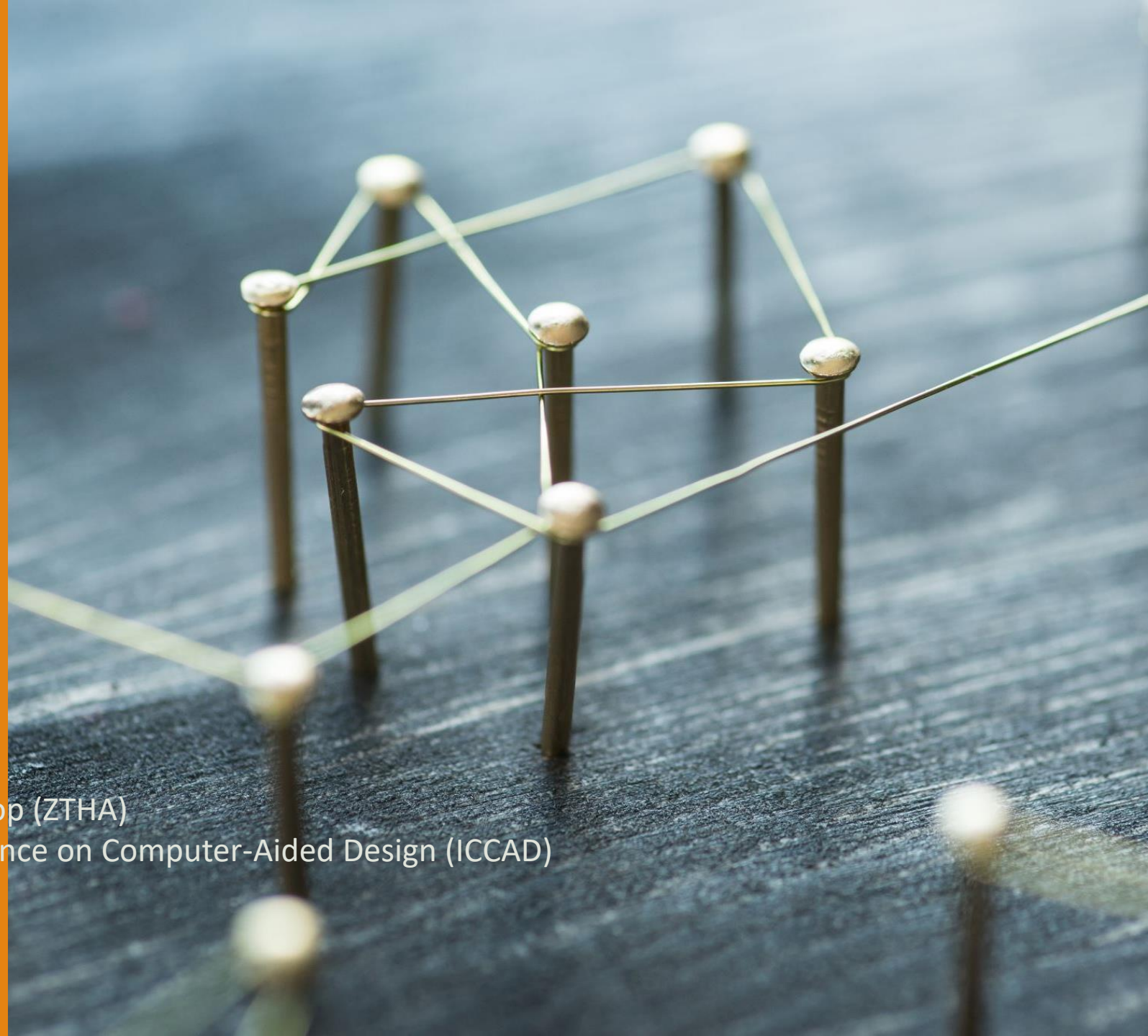
Blockchain-based Community Corroboration for Zero Trust Architectures and Systems

Z Chen
zxchen@ieee.org

Zero Trust Hardware Architectures Workshop (ZTHA)
Co-located with 2022 International Conference on Computer-Aided Design (ICCAD)
Thursday, November 3, 2022



Challenges

Increased connectivity of people and devices to the Internet and to each other has created an ever-expanding attack surface that extends throughout the world and into almost every American home

Attack: SolarWinds, Colonial Pipeline, log4j,...

- Vulnerability of cyber environment as the notion of perimeter is no longer valid.
- Fragility of software and hardware supply chain from apps down to microelectronics
- Profitability of successful attacks

Trust Paradigm

From **trust but verify** (Challenging & Response or Ask & Verify)
To **never trust, always verify**

Total stranger + Verification !=> Trust

community corroboration

- building mutual trust gradually
- assessing and verifying any claims intermittently, and
- delegating trust among communities on demand

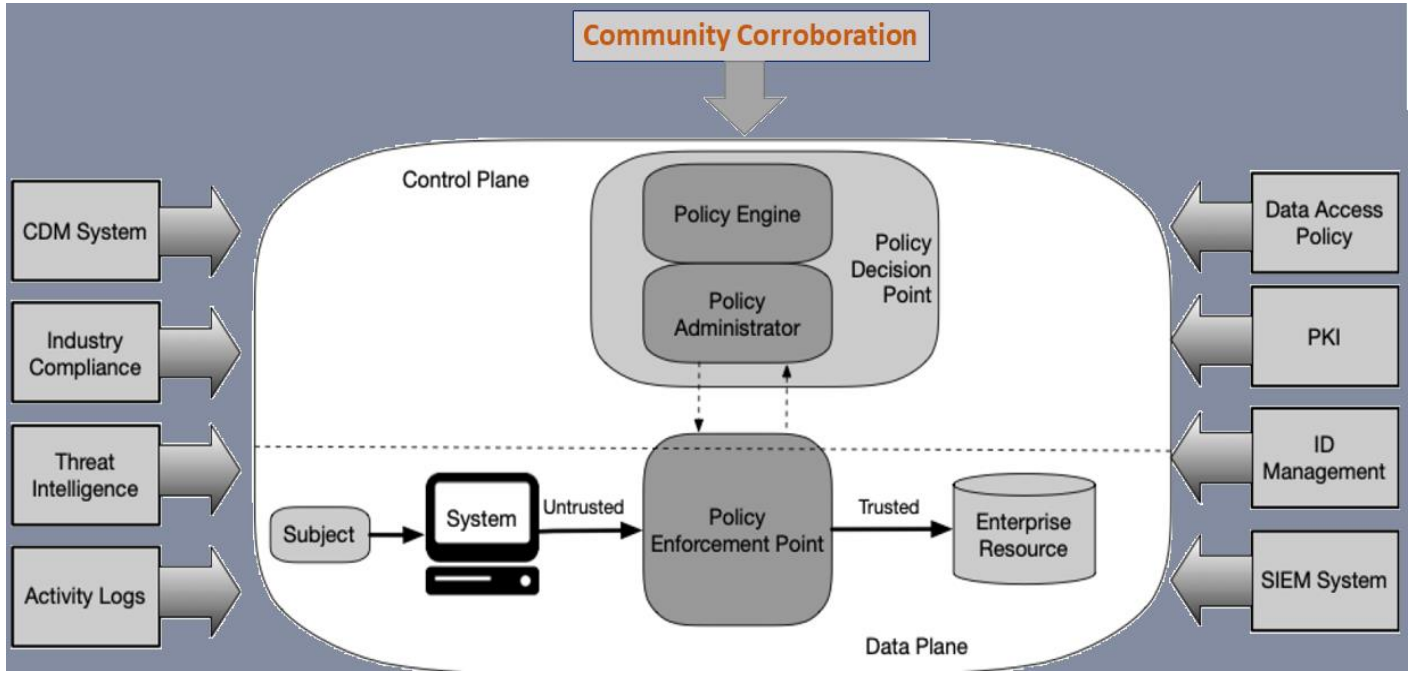
Subject awareness + Verification => as one component to trust

Zero Trust Tenets

01) Resource protection
02) Universal authentication
03) Dynamic authorization
04) Least privilege
05) Monitoring and adapting
06) Encryption at rest
07) Assurance process iteratively
08) Provenance over a period
09) Traceability over lifecycle
10) **Subject awareness**

Zero Trust Architecture

Community Corroboration = Experts Insight + Crowd Audit



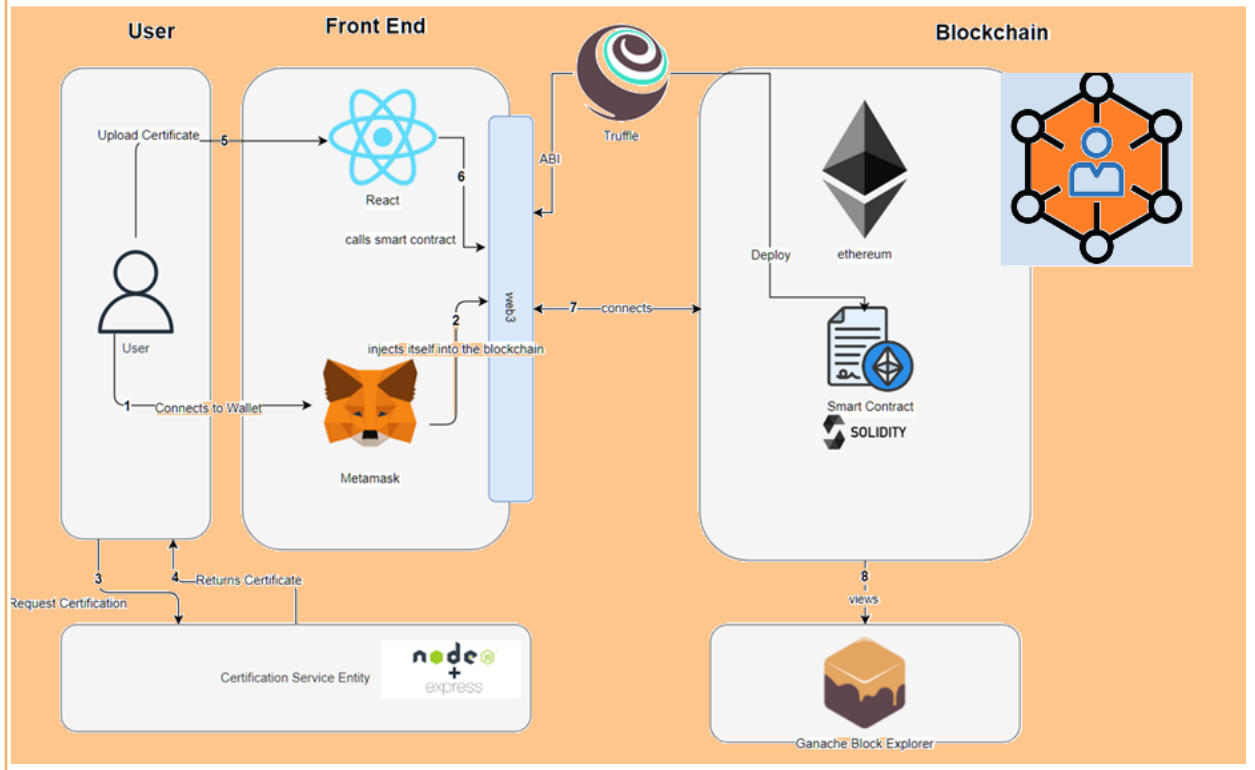
Core Logical Components

PASS+ Pitch

Subject aware
- community corroboration

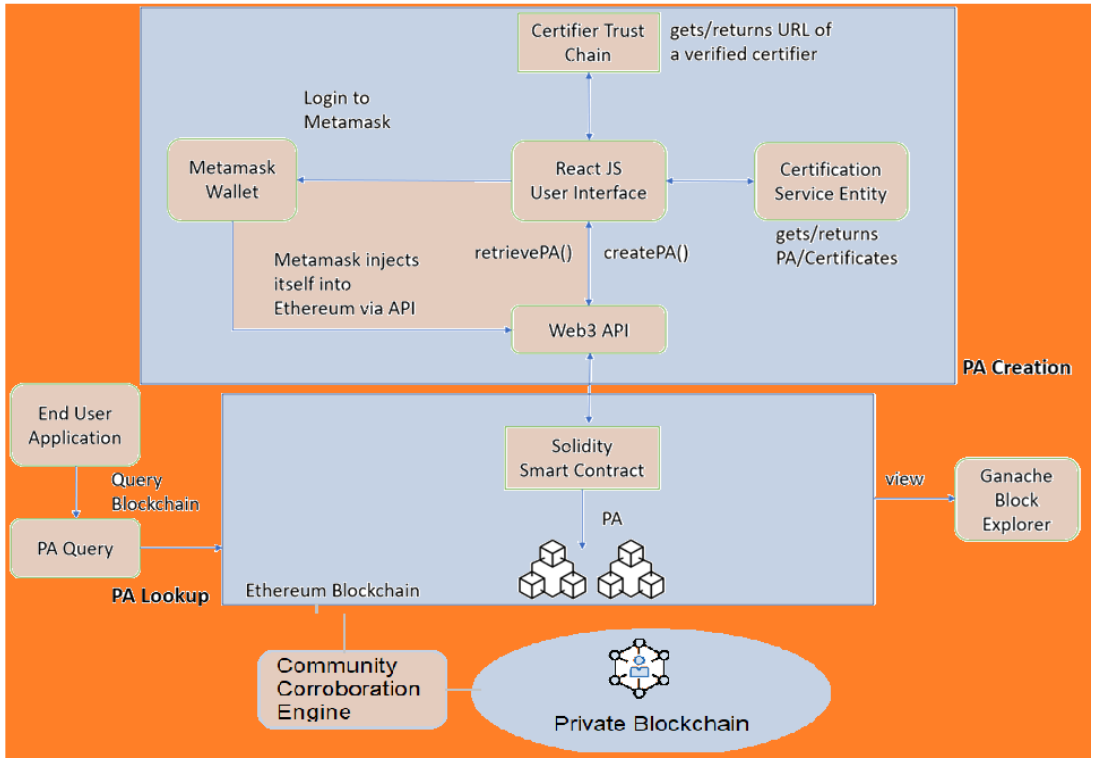
Building blocks
- portfolio artifacts (PAs)

Infrastructure for storing and managing PAs :
- Blockchain =
  Consortium Permissioned
  +
  Public Permissionless



PASS+ Claims

- zero trust =+ community corroboration
- existing groups + crowd auditing
- management and control
- PAs: accumulation over a period
- PAs: anonymous if needed
- PAs: hardened once they are in a chain
- self-sovereign identity (SSI)
- economic considerations



PASS+ to Zero Trust

? **Just because PASS+ has a write-once-read-many data storage, and any data modifications will be detected, would it be sufficient for trust?**

PASS+ to Zero Trust

? Just because PASS+ has a write-once-read-many data storage, and any data modifications will be detected, would it be sufficient for trust?

? **What information must be recorded in this data storage to enable a user to establish/verify trust as a component?**

PASS+ to Zero Trust

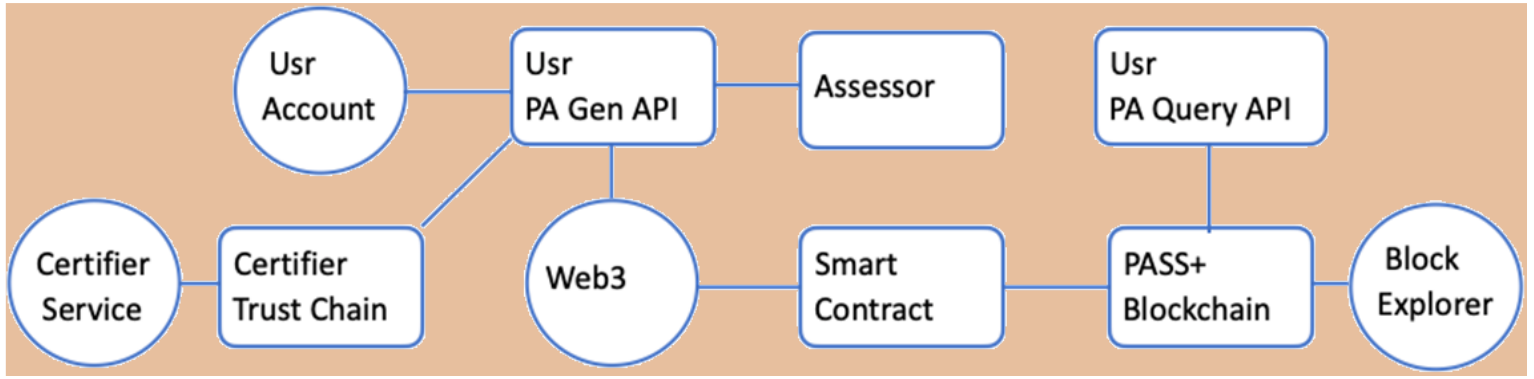
? Just because PASS+ has a write-once-read-many data storage, and any data modifications will be detected, would it be sufficient for trust?

? What information must be recorded in this data storage to enable a user to establish/verify trust as a component?

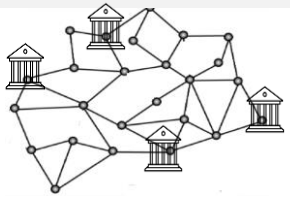
? **Just because something is open, supported/verified by a community, does it mean that it is correct? The assumption of "community validation equivalent to trust" is flawed.**

PASS+ to Zero Trust

? Just because PASS+ has a write-once-read-many data storage, and any data modifications will be detected, would it be sufficient for trust?
? What information must be recorded in this data storage to enable a user to establish/verify trust as a component?
? Just because something is open, supported/verified by a community, does it mean that it is correct? The assumption of “community validation equivalent to trust” is flawed.
? **Why not trust experts more than the general public community in this very specialized software and/or hardware supply chain from apps down to microelectronics?**



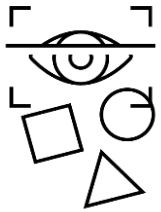
Certifier Trust Chain



Consortium Permissioned
Certifiers:
- Registration service
- Signing Service
- Approving Service

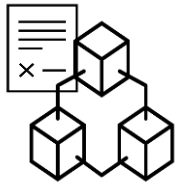
Certifier Chain:
- Verification Service
- Mapping Service
- List Service

Assessor



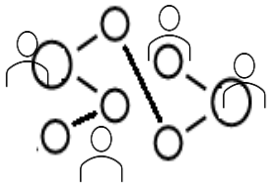
Expert Sys Engine
ML Models
Inferences

Smart Contract



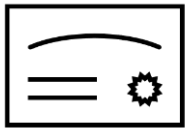
Creation Call
Retrieval Call
Encryption Util
ABI Decode Util

PA Dev API

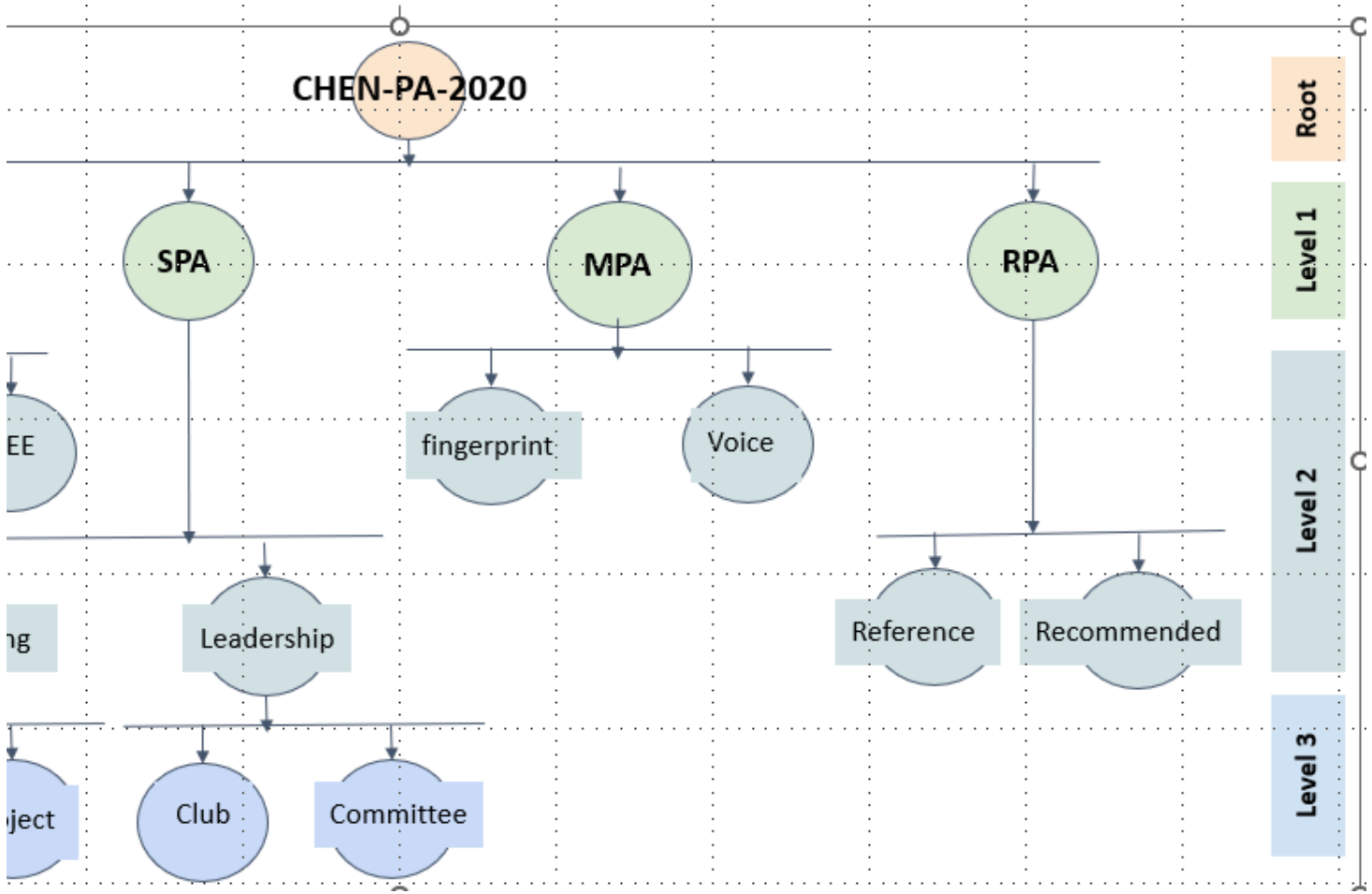


Account Mgnt
PA Setup
Query Service

Portfolio Artifact

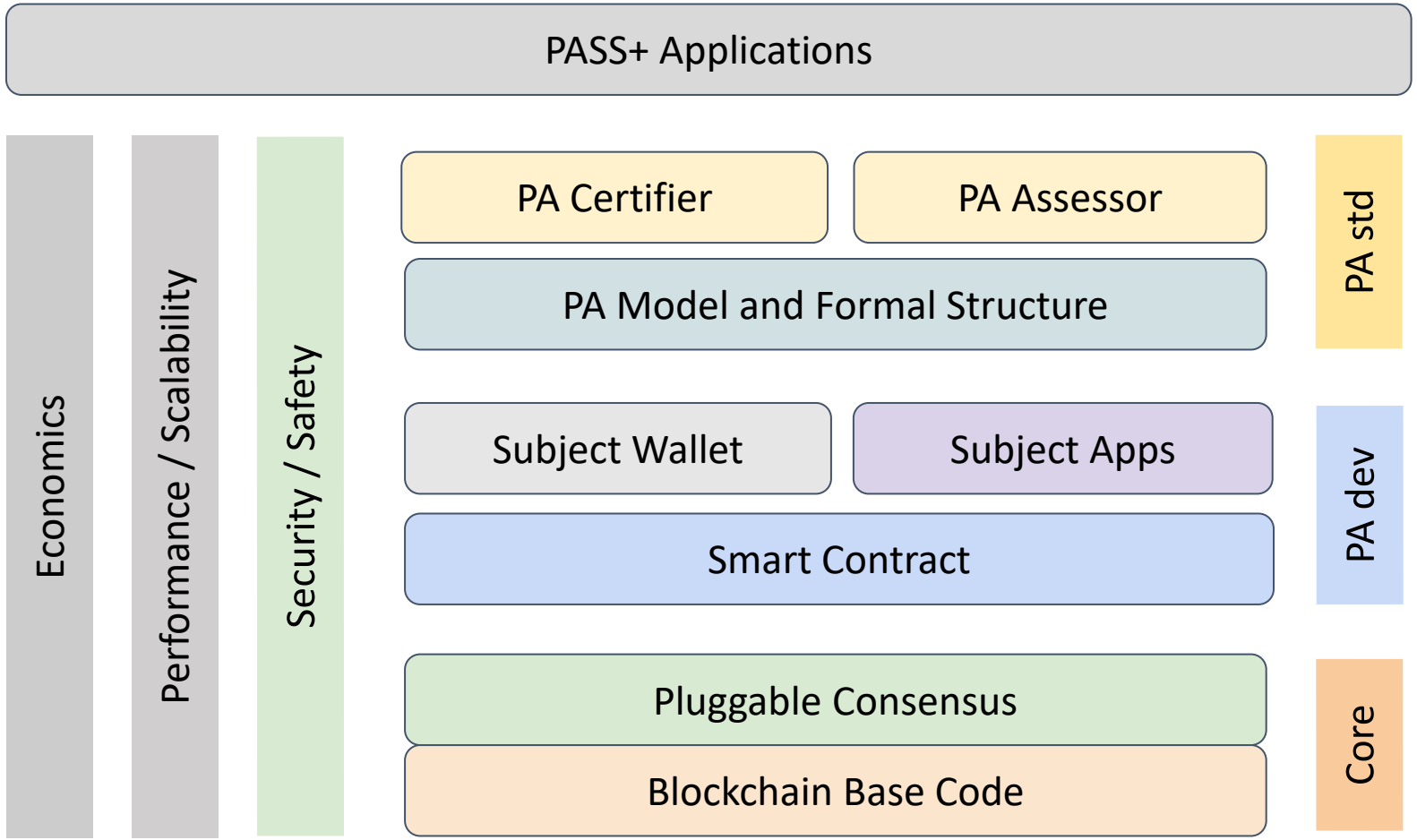


PA Model
Ontology
JSON-LD
PA JSON Input
Internal Verification
External Validation

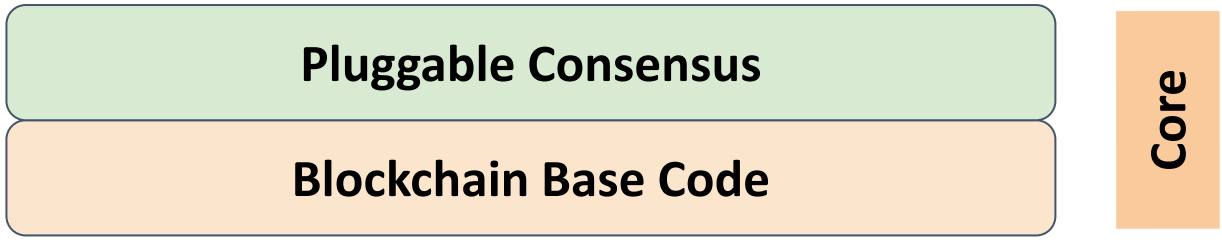

CHEN-PA-2020
SPA
MPA
RPA
EE
fingerprint
Voice
ng
Leadership
Reference
Recommended
ject
Club
Committee
Root
Level 1
Level 2
Level 3

PAs for Z

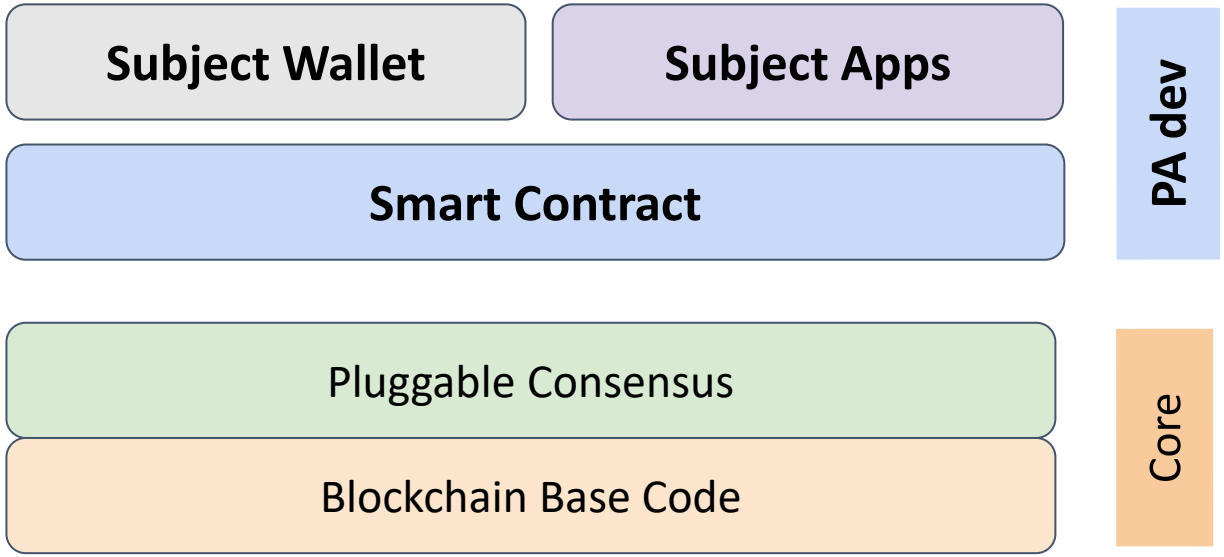
PASS+ Public Chain Prototype Overview



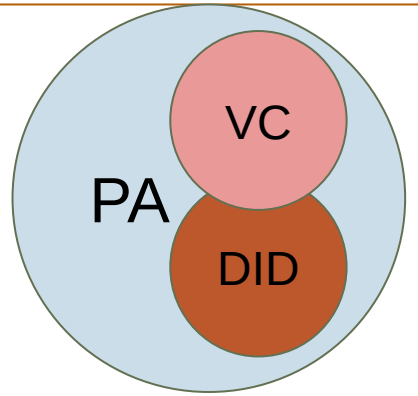
- Develop and plug in our consensus protocol
  - Proof of work (PoW) - not sustainable
  - Proof of stack (PoS) - hard to quantify stack
  - Proof of authority (PoA) - centralized decentralized trap, bribery vulnerability
  - **Add randomness and incentives**
- Control the generation blocks based on the time elapse, size, and utility



- Create, store, and retrieve PAs
- Create and manage subject accounts
- Visualize accounts
- Communicate with other web applications certifiers / assessors



- Human subject - holistic collection
- Non-human subject - identifiable features



- Standardization
- Not too cumbersome

- Certification vs Evaluation
- Self-sovereign identity (SSI)

PA Certifier | **PA Assessor**

PA Model and Formal Structure

Subject Wallet | Subject Apps

Smart Contract

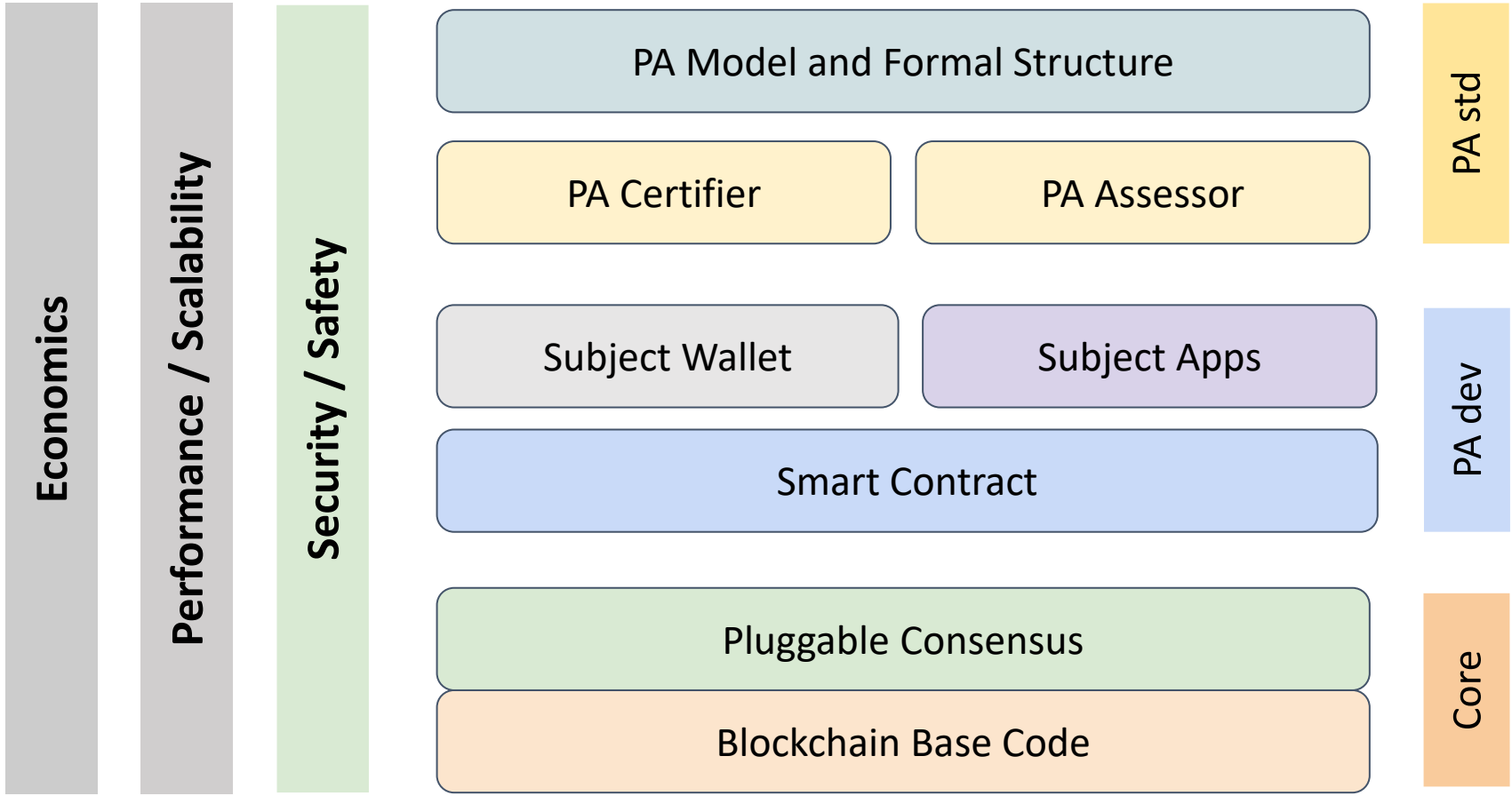
Pluggable Consensus

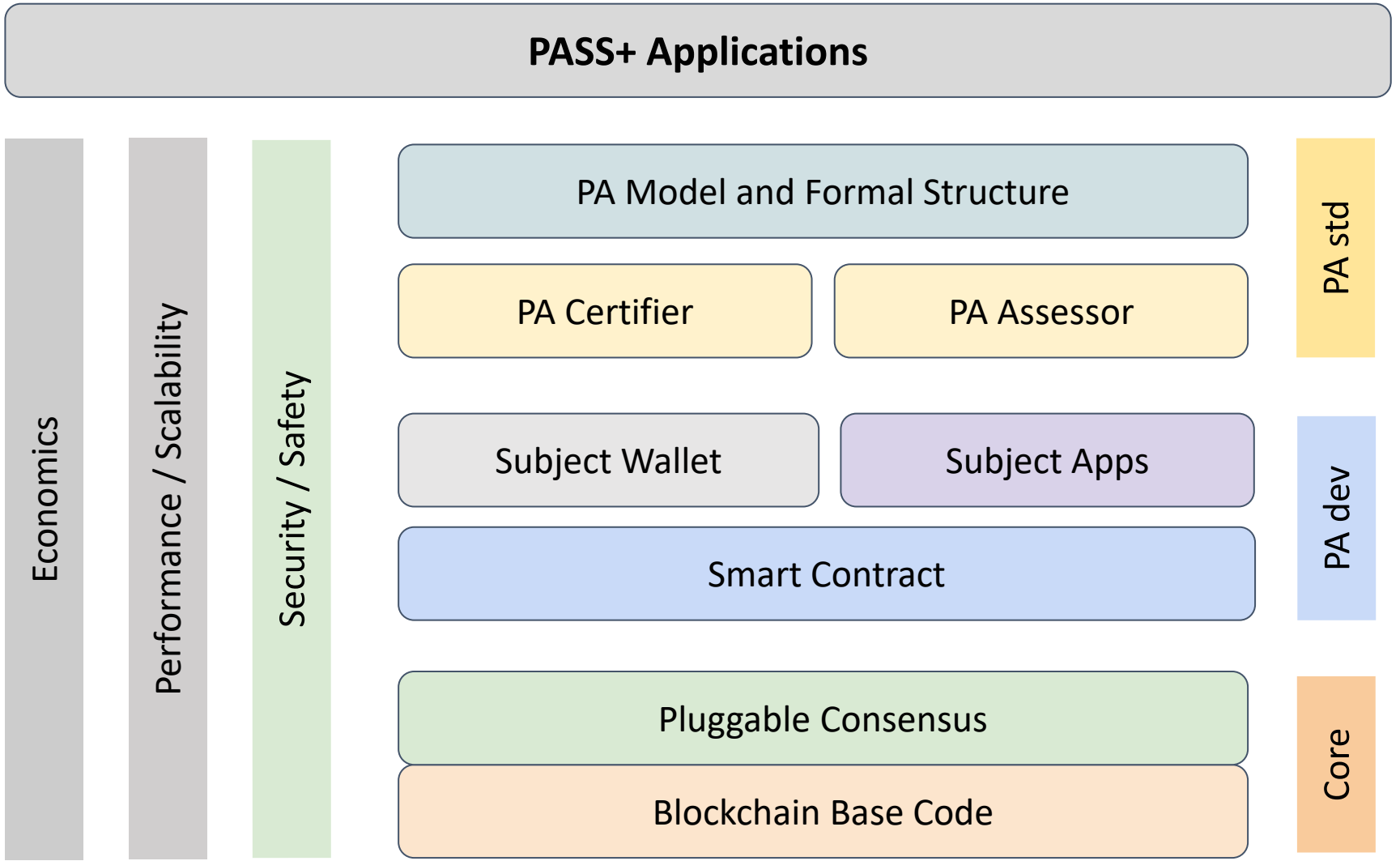
Blockchain Base Code

PA std

PA dev

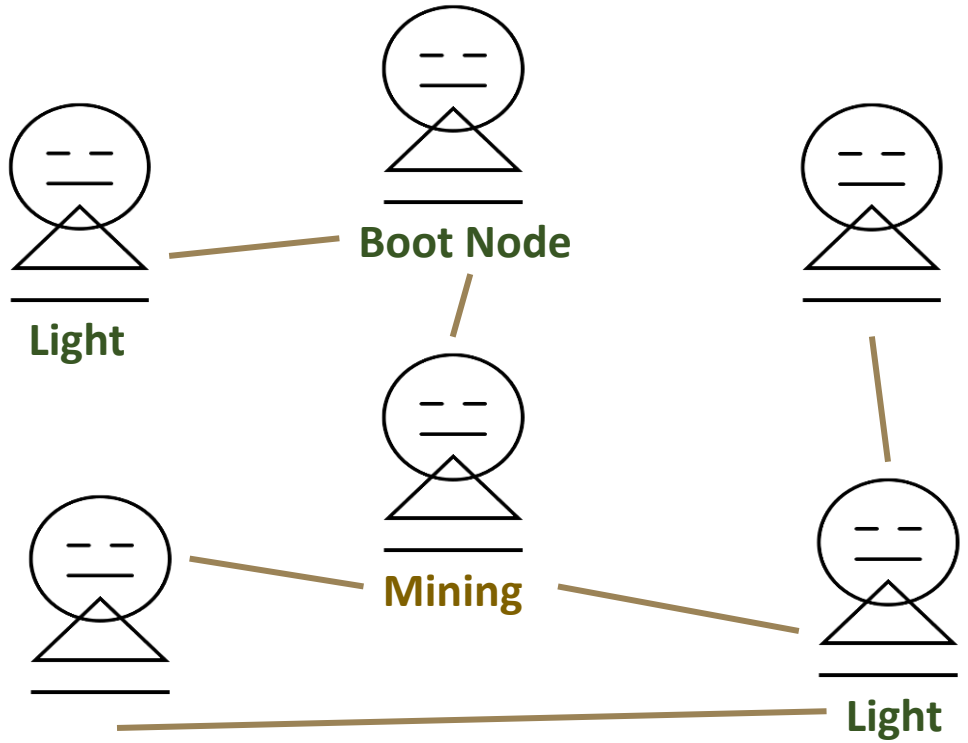
Core


Economics
Performance / Scalability
Security / Safety
PA Model and Formal Structure
PA Certifier
PA Assessor
Subject Wallet
Subject Apps
Smart Contract
Pluggable Consensus
Blockchain Base Code
PA std
PA dev
Core


PASS+ Applications
Economics
Performance / Scalability
Security / Safety
PA Model and Formal Structure
PA Certifier
PA Assessor
Subject Wallet
Subject Apps
Smart Contract
Pluggable Consensus
Blockchain Base Code
PA std
PA dev
Core

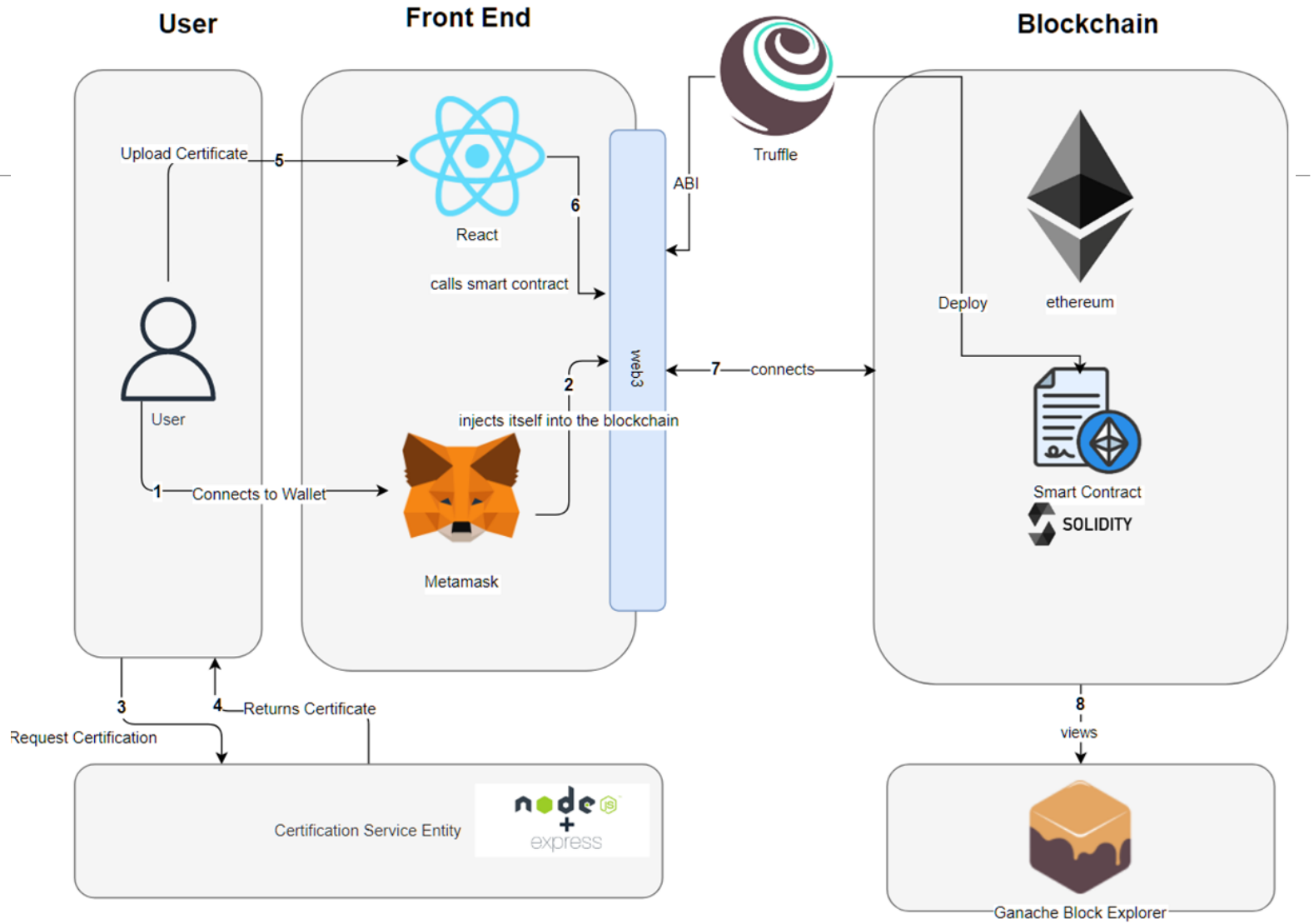
Core Implementation / Proof of Concept

1. Clone source code
2. Coding – adding consensus protocol code
3. Build the executable
4. Design test configuration
5. Make dirs
6. Create node accounts
7. Generate genesis.json
8. Init test net
9. Create node accounts and start
10. Generate boot key and bind it to the boot node
11. Start all nodes
12. Setup test env
13. Testing

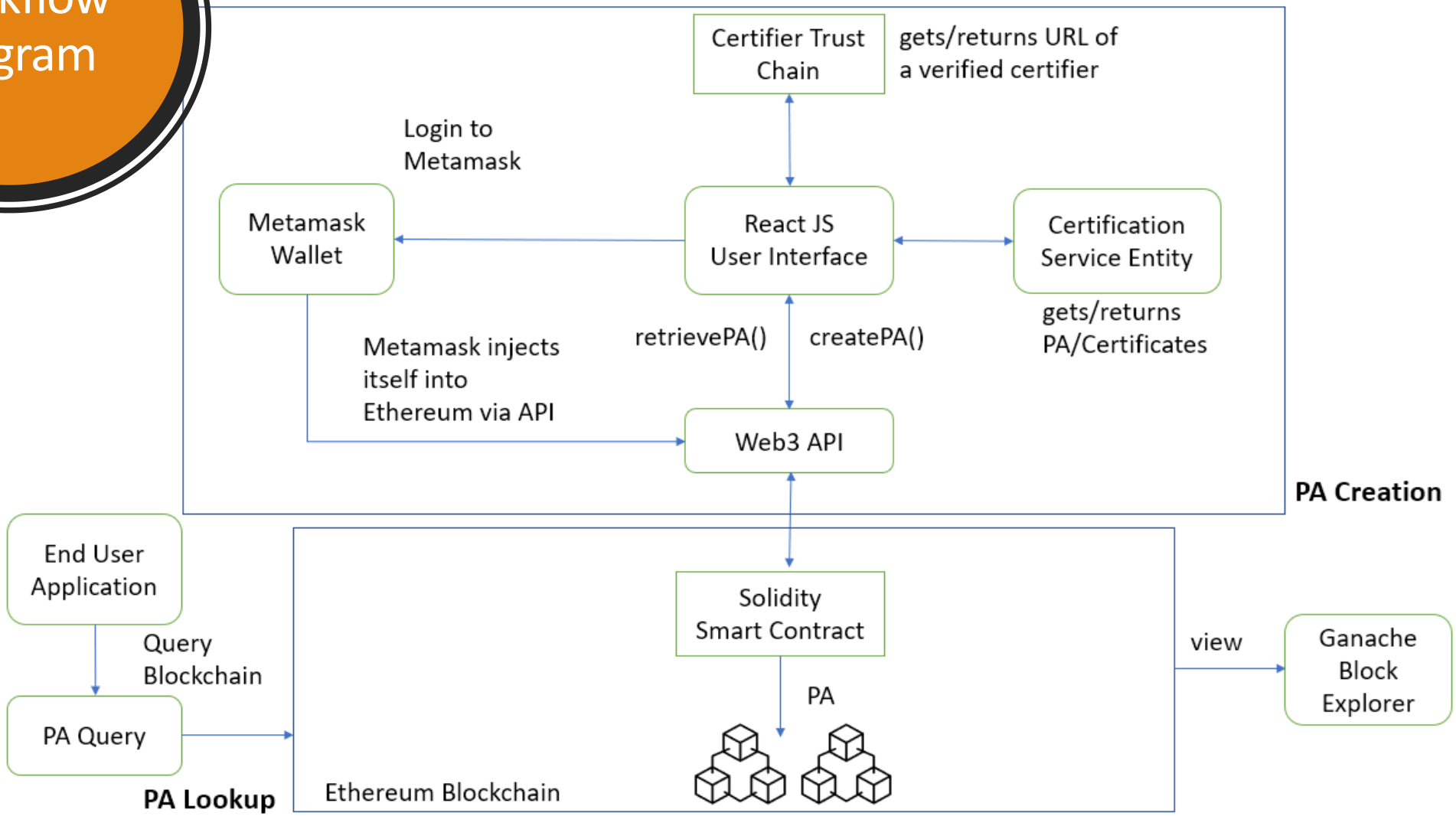


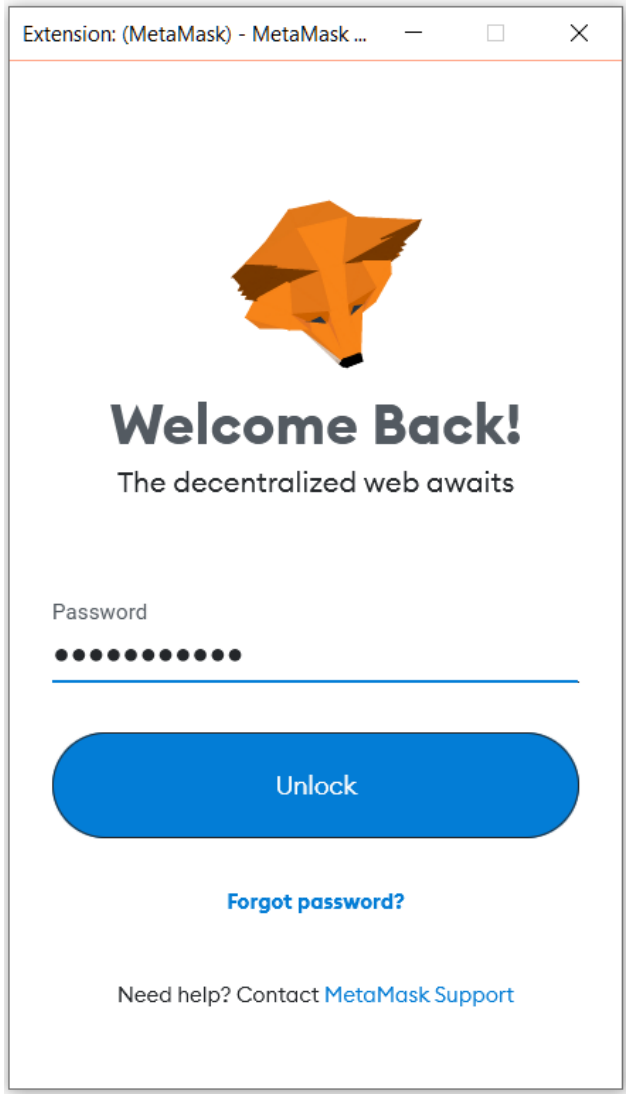
Tools

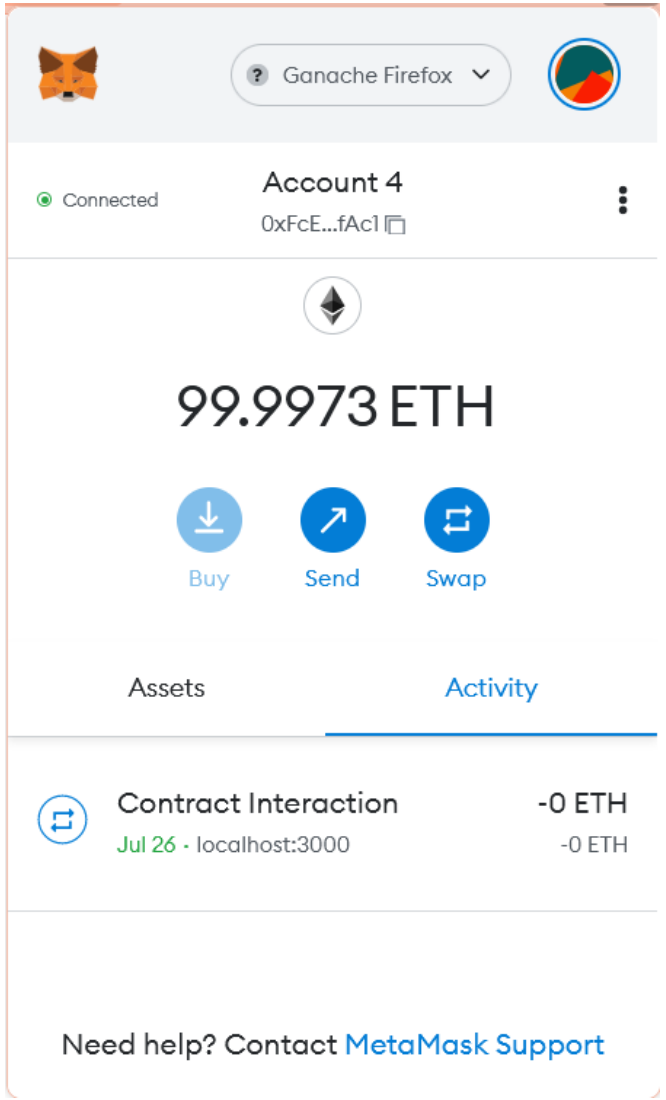
- **Development**
  - Ethereum
  - Truffle
  - Ganache
  - React
  - web3.js
  - MetaMask
  - Express
- **Languages**
  - JavaScript
  - Solidity
  - Go
  - Rust



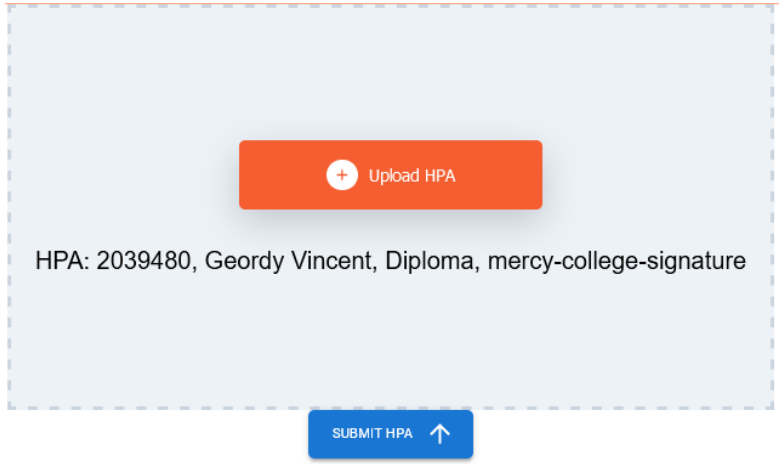
Workflow Diagram

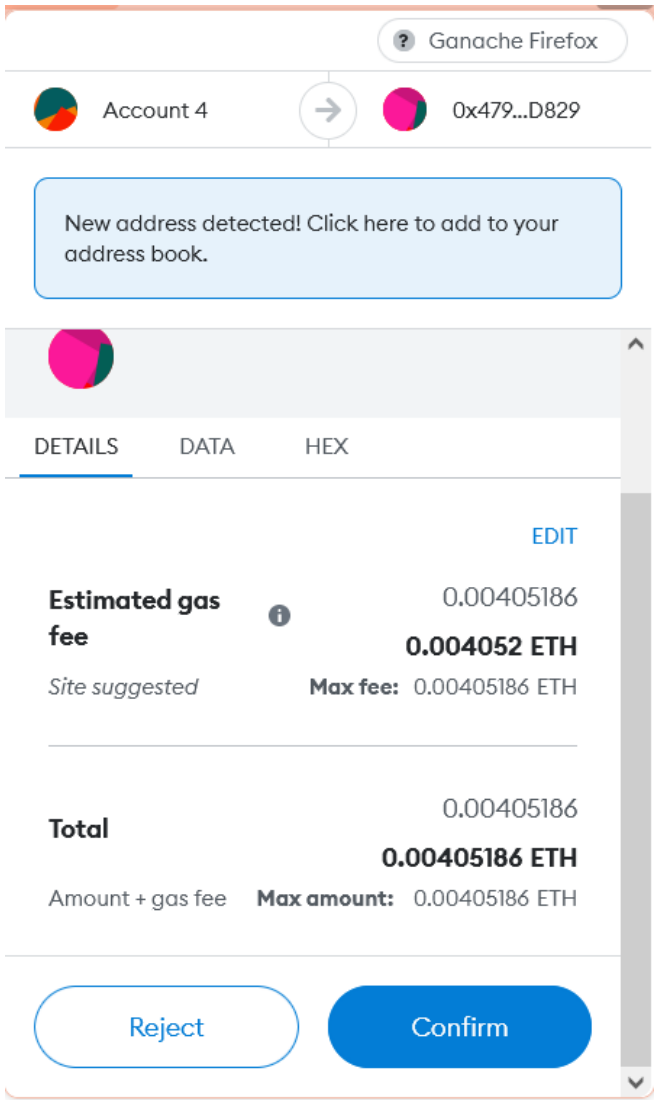



Extension: (MetaMask) - MetaMask ...
Welcome Back!
The decentralized web awaits
Password
Unlock
Forgot password?
Need help? Contact MetaMask Support


Ganache Firefox
Connected
Account 4
0xFcE...fAc1
99.9973 ETH
Buy
Send
Swap
Assets
Activity
Contract Interaction
-0 ETH
Jul 26 · localhost:3000
-0 ETH
Need help? Contact MetaMask Support

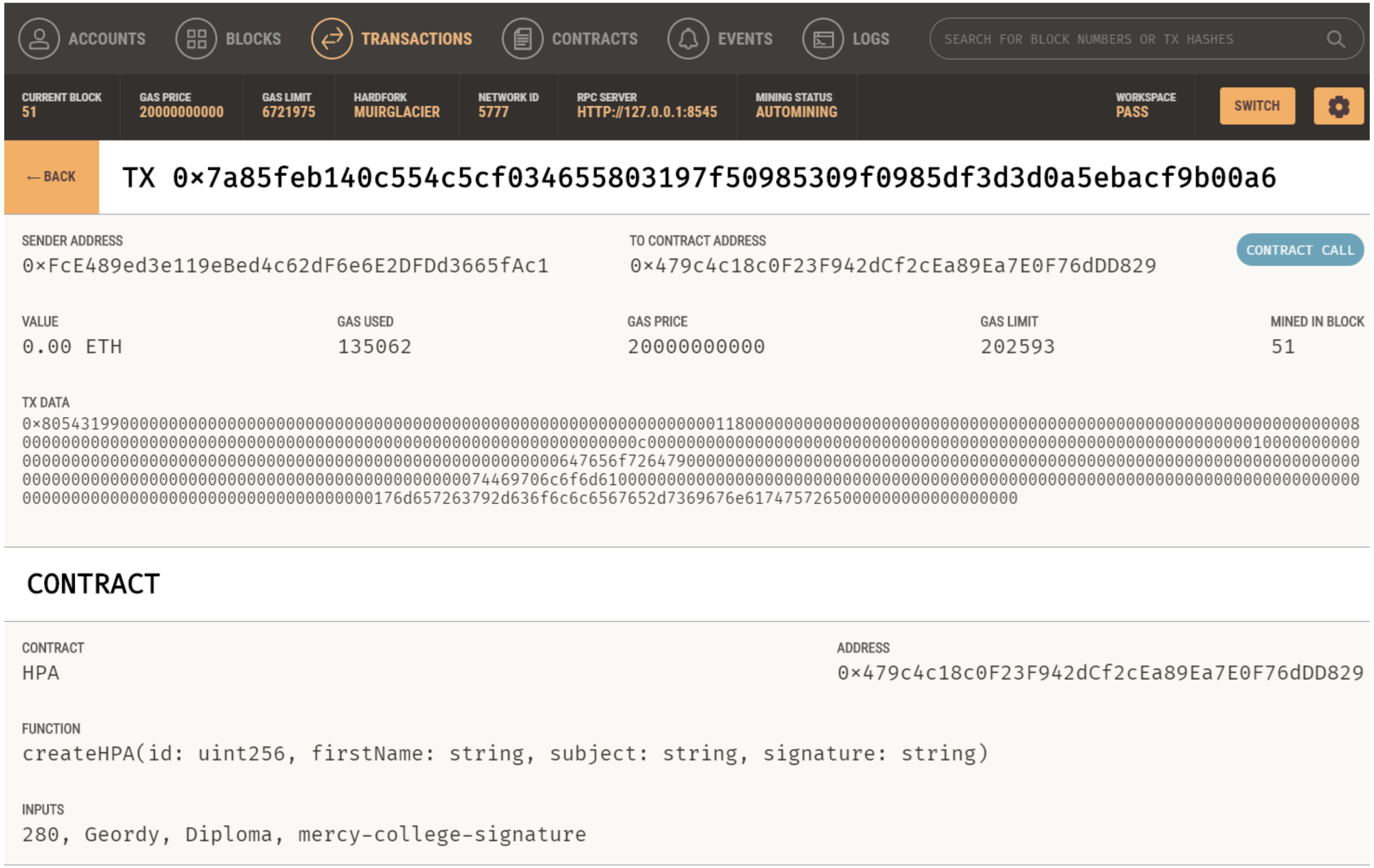
User Wallet

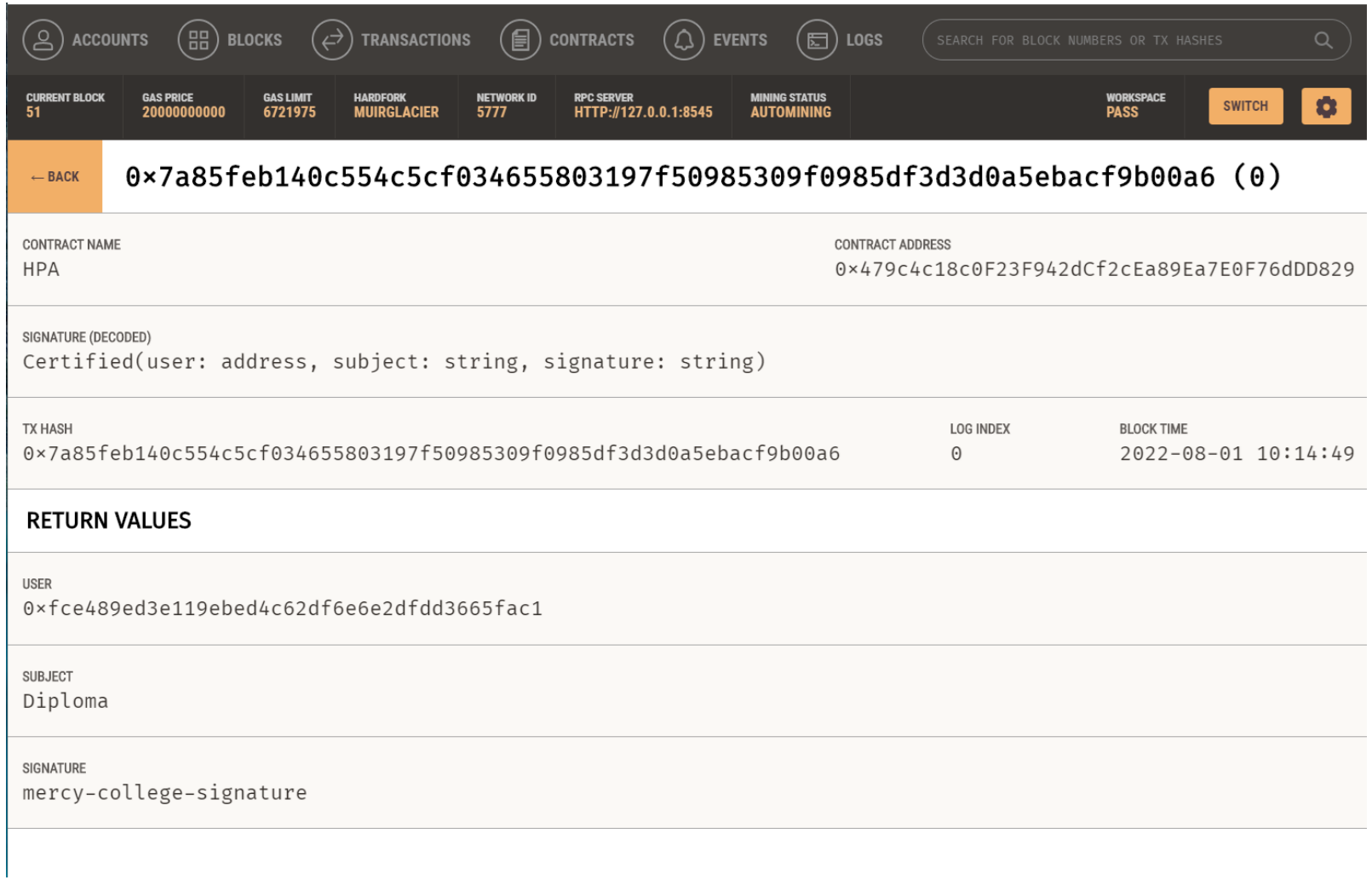

Upload HPA
HPA: 2039480, Geordy Vincent, Diploma, mercy-college-signature
SUBMIT HPA


Ganache Firefox
Account 4
0x479...D829
New address detected! Click here to add to your address book.
DETAILS
DATA
HEX
EDIT
Estimated gas fee
0.00405186
0.004052 ETH
Site suggested
Max fee: 0.00405186 ETH
Total
0.00405186
0.00405186 ETH
Amount + gas fee
Max amount: 0.00405186 ETH
Reject
Confirm

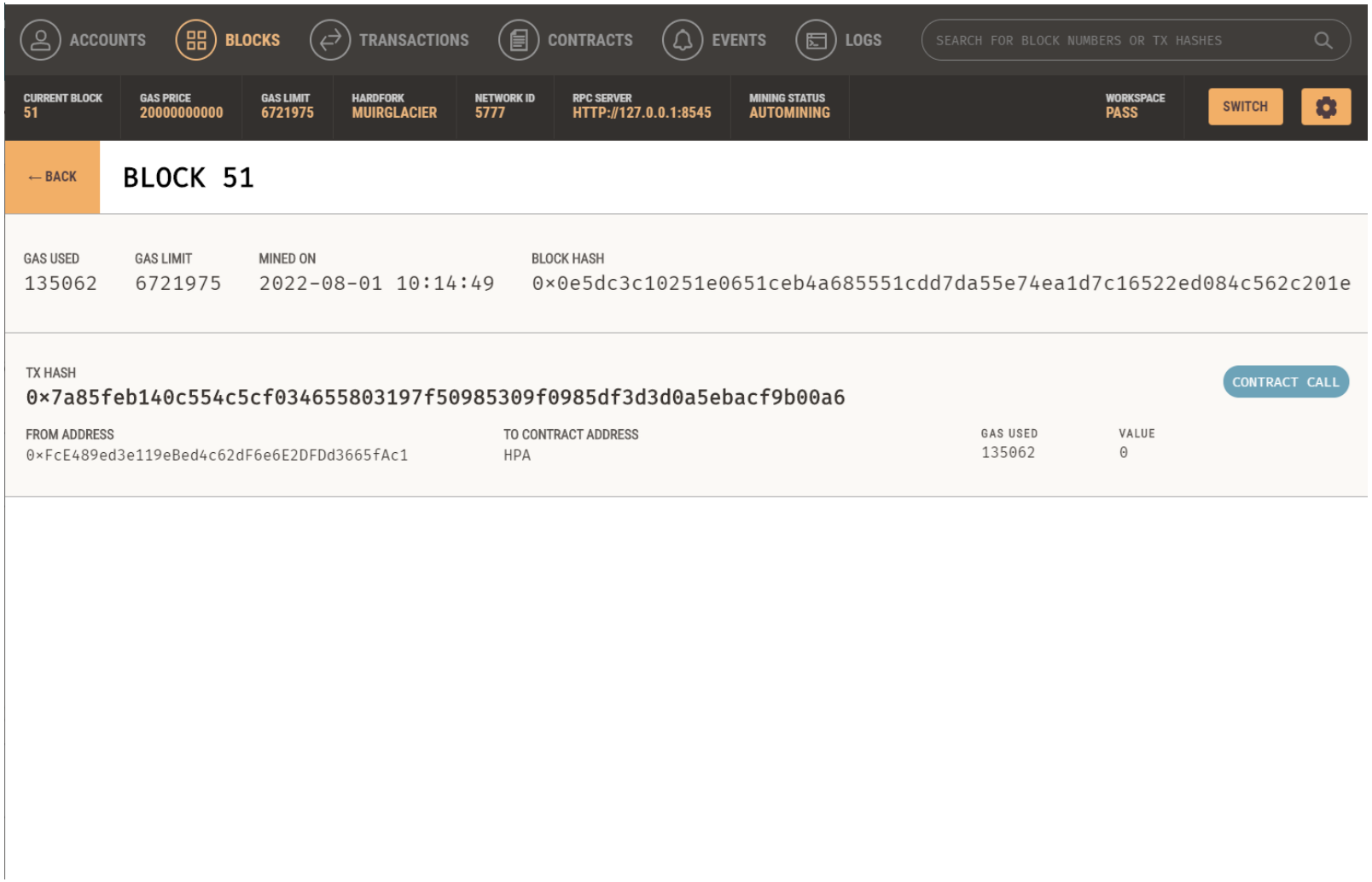
PA Upload

PA Transaction



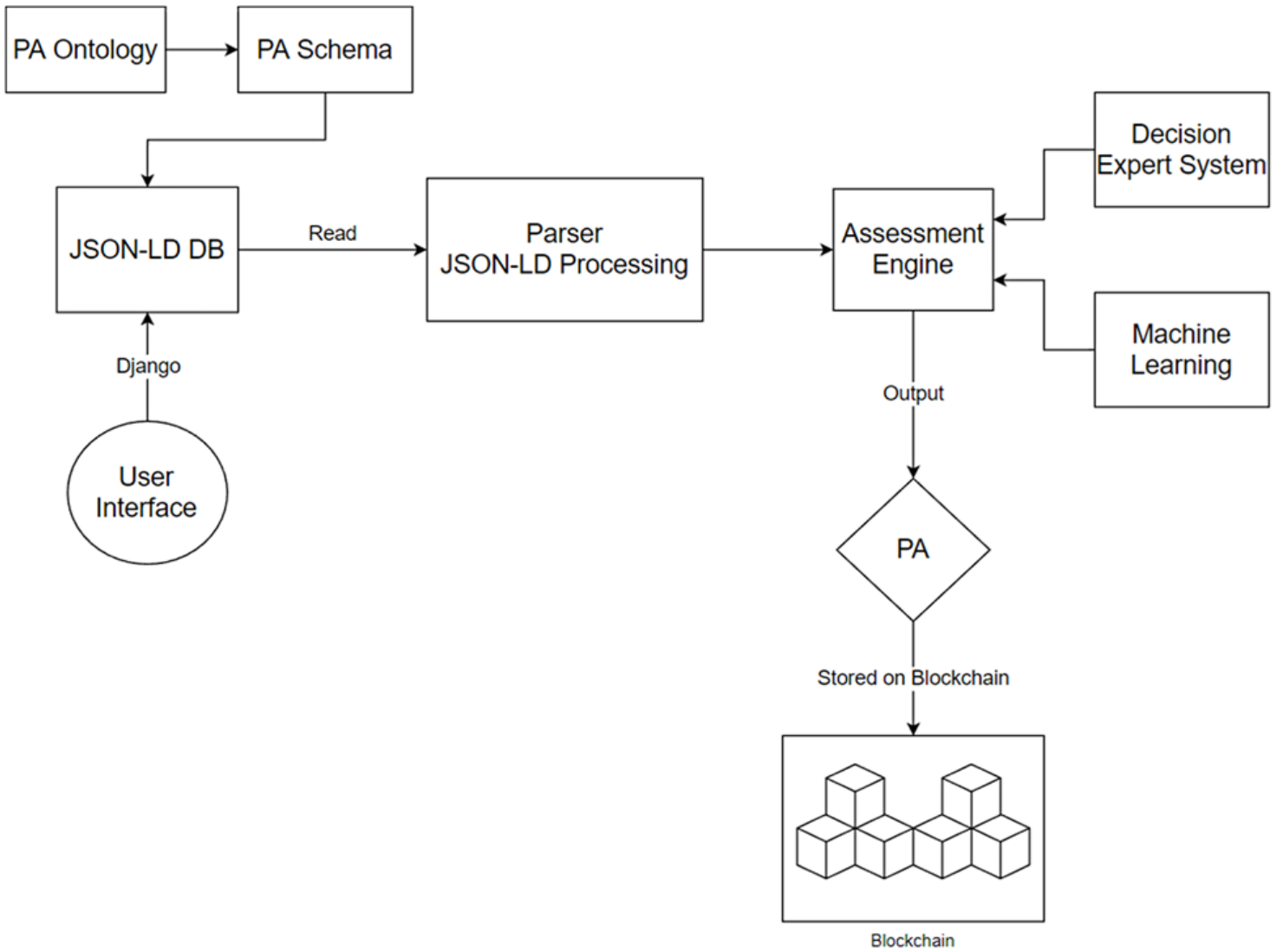

ACCOUNTS
BLOCKS
TRANSACTIONS
CONTRACTS
EVENTS
LOGS
SEARCH FOR BLOCK NUMBERS OR TX HASHES
CURRENT BLOCK 51
GAS PRICE 20000000000
GAS LIMIT 6721975
HARDFORK MUIRGLACIER
NETWORK ID 5777
RPC SERVER HTTP://127.0.0.1:8545
MINING STATUS AUTOMINING
WORKSPACE PASS
SWITCH
← BACK
0×7a85feb140c554c5cf034655803197f50985309f0985df3d3d0a5ebacf9b00a6 (0)
CONTRACT NAME
HPA
CONTRACT ADDRESS
0×479c4c18c0F23F942dCf2cEa89Ea7E0F76dDD829
SIGNATURE (DECODED)
Certified(user: address, subject: string, signature: string)
TX HASH
0×7a85feb140c554c5cf034655803197f50985309f0985df3d3d0a5ebacf9b00a6
LOG INDEX
0
BLOCK TIME
2022-08-01 10:14:49
RETURN VALUES
USER
0×fce489ed3e119ebed4c62df6e6e2dfdd3665fac1
SUBJECT
Diploma
SIGNATURE
mercy-college-signature

PA Event

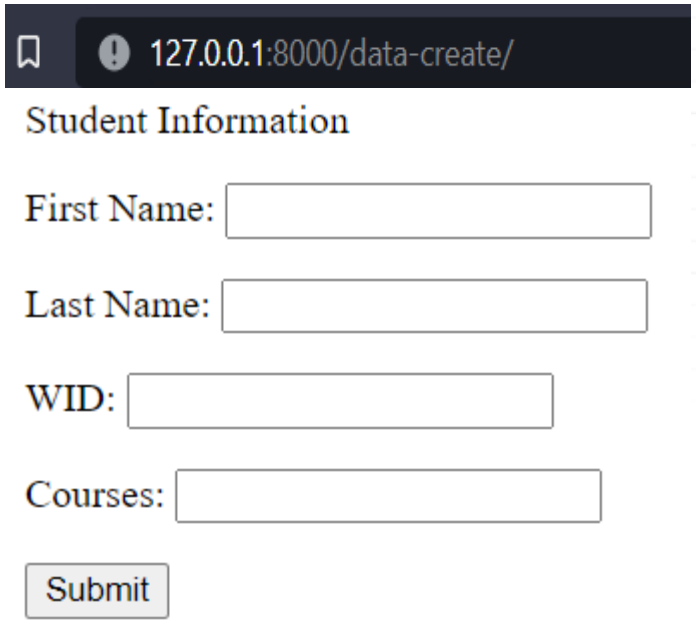

ACCOUNTS
BLOCKS
TRANSACTIONS
CONTRACTS
EVENTS
LOGS
SEARCH FOR BLOCK NUMBERS OR TX HASHES
CURRENT BLOCK 51
GAS PRICE 20000000000
GAS LIMIT 6721975
HARDFORK MUIRGLACIER
NETWORK ID 5777
RPC SERVER HTTP://127.0.0.1:8545
MINING STATUS AUTOMINING
WORKSPACE PASS
SWITCH
← BACK
BLOCK 51
GAS USED 135062
GAS LIMIT 6721975
MINED ON 2022-08-01 10:14:49
BLOCK HASH 0×0e5dc3c10251e0651ceb4a685551cdd7da55e74ea1d7c16522ed084c562c201e
TX HASH
0×7a85feb140c554c5cf034655803197f50985309f0985df3d3d0a5ebacf9b00a6
CONTRACT CALL
FROM ADDRESS
0×FcE489ed3e119eBed4c62dF6e6E2DFDd3665fAc1
TO CONTRACT ADDRESS
HPA
GAS USED
135062
VALUE
0

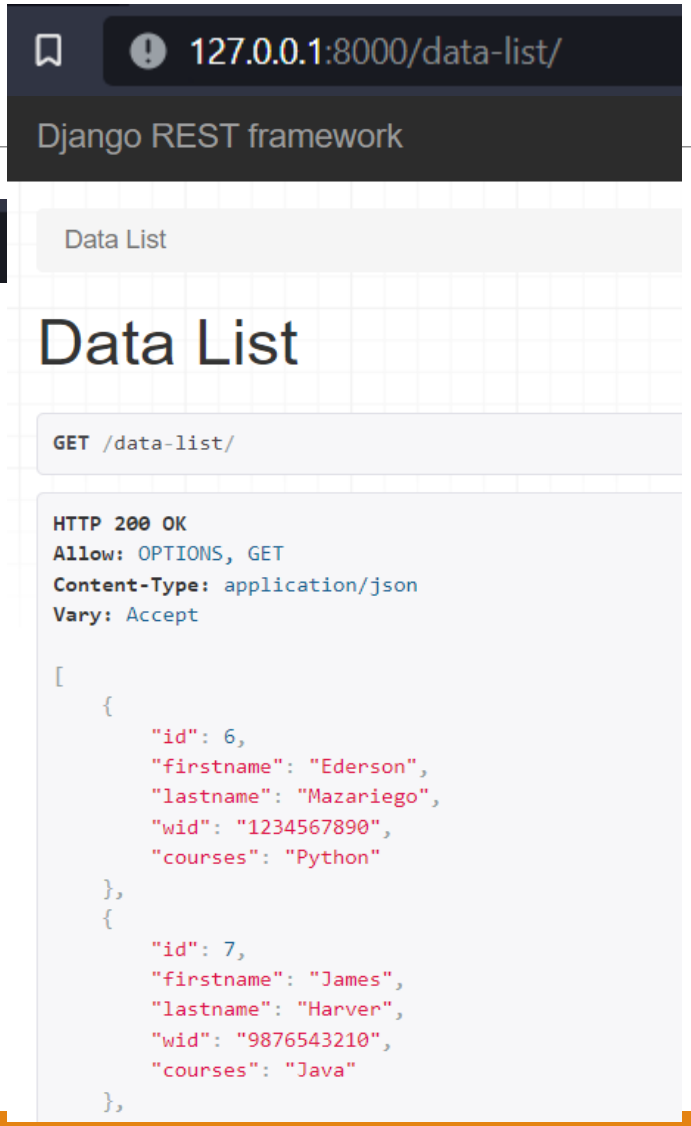
Block Mined

Workflow Diagram



User Data Input





User Competency Assessment Output

jsonldex2 > [ ] @graph > {} 0 > [ ] courses > {} 0

```
{
    "@context": "https://jsonldresume.github.io/skill/context.json",
    "@graph": [
        {
            "@id": "https://maryjane.github.io/resume/",
            "@type": "skill:Resume",
            "owner": {
                "@id": "https://maryjane.github.io"
            },
            "name": [
                {
                    "@id": "https://maryjane.github.io",
                    "@type": "Person",
                    "givenName": "Mary",
                    "familyName": "Jane"
                }
            ],
            "courses": [
                {
                    "@id": "https://maryjane.github.io/resume/",
                    "@type": "Course",
                    "name": "Intro to Python"
                }
            ],
            "projects": [
                {
                    "@id": "https://maryjane.github.io",
                    "@type": "CreativeWork",
                    "abstract": "Projects completed during academic years.",
                    "name": "Python Project"
                }
            ]

        }
    ]
}
```

jsonldex2extract.py > ...

```python
import json

f = open('jsonldex2')
data = json.load(f)

x = 0

courses_db = ('Intro to Python', 'Java Intro')
projects_db = ('Python Project', 'Java Project')

data2 = data['@graph'][0]["courses"]
for course in data2:
    if course['name'] in courses_db:
        level = x+1
        print('Course:',course['name'])
        #print(level)
    else:
        print('Course not found.')
data3 = data['@graph'][0]["projects"]
for project in data3:
    if project['name'] in projects_db:
        level2 = level+1
        print('Project:',project['name'])
        #print(level2)
    else:
        print('Project not found.')

if level2 == 0:
    print ("User level is 0.")
elif level2 == 1:
    print ("User level is Beginner.")
elif level2 == 2:
    print ("User level is Intermediate.")
else:
    print("User level cannot be defined.")

f.close()
```

```
Course: Intro to Python
Project: Python Project
User level is Intermediate.
```

JSON-LD Sample

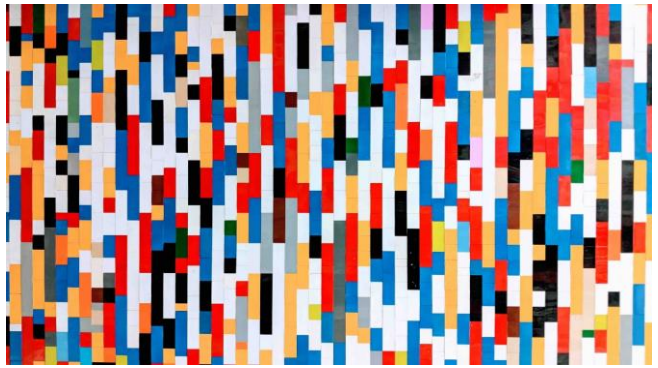
```
{
	"@context": "https://jsonldresume.github.io/skill/context.json",
	"@graph": [
		{
			"@id": "https://maryjane.github.io/resume/",
			"@type": "skill:Resume",
			"owner": {
				"@id": "https://maryjane.github.io"
			},
		"name": [
			{
				"@id": "https://maryjane.github.io",
				"@type": "Person",
				"givenName": "Mary",
				"familyName": "Jane"
			}
		],
```

```
		"courses": [
			{
				"@id": "https://maryjane.github.io/resume/",
				"@type": "Course",
				"name": "Intro to Python"
			}
		],
		"projects": [
			{
				"@id": "https://maryjane.github.io",
				"@type": "CreativeWork",
				"abstract": "Projects completed in semester.",
				"name": "Python Project"
			}
		]

	}
  ]
}
```

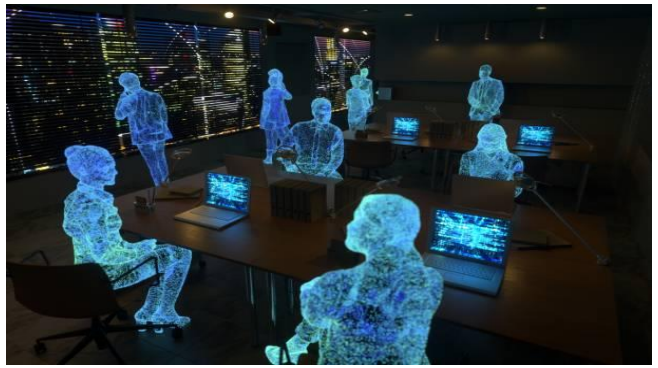
PASS+ Applications

Task Assignment & Delegation



NCL Skill Testing & Training

Bot Id & Content Digger



Vehicle Authn & Proof of Vicinity

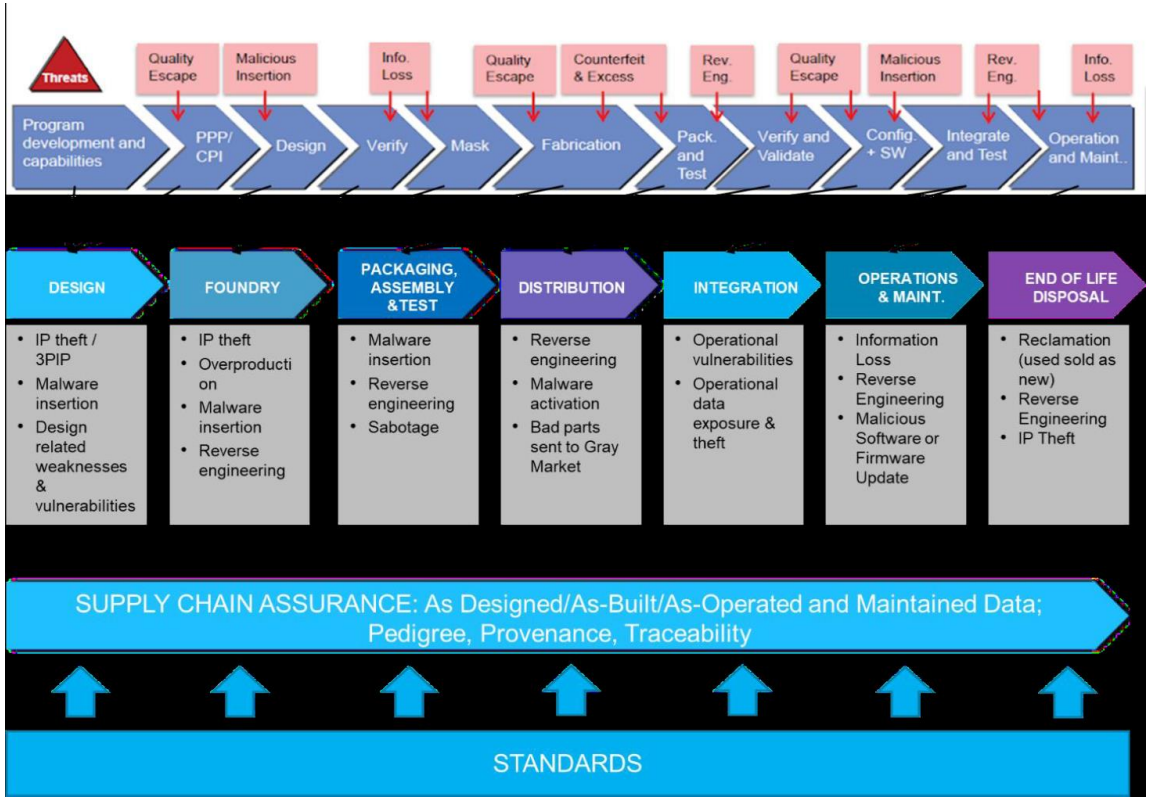
Digital Community Binding

Digital Life & Afterlife

Special Consideration- microelectronics up

Execution Stack

- counterfeit chips
- trojan implanted hardware
- infected OS
- malicious firmware
- middleware,
- supply chain software
- applications
- bad communication.



Future Work

Development

- Re-examine ontology analysis
- Create JSON-LD schema
- Code block query
- Develop block generation
- Improve the code of subject API
- Generalize smart contract
- Release PASS+

Research

- Study economic side of the blockchain for PASS+
- Organize PAs more efficiently
- Explore ML model for assessor
- Improve proof-of-me consensus and test it
- Support advanced query
- Support multimodality

Application

- NCL skill testing and training
- Vehicle authentication and proof of vicinity
- Bot identification for content creator analysis
- Task assignment and delegation
- Job hunting or recruiting
- Digital community binding
- Digital life and afterlife
- Zero trust hardware

Deming Chen
dchen@illinois.edu

Geordy Vincent
gvincent@mercy.edu

Ederson Mazariego
emazariego@mercy.edu

Acknowledgement

This research was performed under an appointment to the U.S. Department of Homeland Security (DHS) Science & Technology (S&T) Directorate Office of University Programs Summer Research Team Program for Minority Serving Institutions, administered by the Oak Ridge Institute for Science and Education (ORISE) through an interagency agreement between the U.S. Department of Energy (DOE) and DHS. ORISE is managed by ORAU under DOE contract number DE-SC0014664. All opinions expressed in this paper are the author's and do not necessarily reflect the policies and views of DHS, DOE or ORAU/ORISE.