Credential Vault: User Registration & Update Blockchain Flow (1A & 2A Focus)

An open architectural challenge led by Abhishek Kumar ? April 2025

PHASED FLOW OVERVIEW

USER ACTIONS & BLOCKCHAIN EVENTS:

1A: User Registration

- User provides required and optional information.
- System verifies `mandatory fields`: Email, Phone, DOB.
- System optionally stores `optional fields`: Full Name, Initials, Location, etc.
- System generates:
  - `registration` block with:
    - Block timestamp (UTC)
    - Global `blockchain index`
    - `user_id`
    - Encrypted/hashed data split into `mandatory` and `optional`
    - `previous_hash` (genesis = "0")
    - `block_hash` (SHA-256 of the block)
  - Logs entry in `userregister.log`

2A: User Update

- User selects "Update Info" after registration.
- Provides correction(s) or additions.
- System validates update and generates:
  - `update` block with:
    - Timestamp (UTC)

    - `user_id`
    - Corrected fields only
    - Reference to `registration_block_hash`
    - New `block_hash`
  - Logs entry in `userupdate.log`

FIELD CLASSIFICATION CONTEXT (from Registration Interface)

Mandatory (Required at submission):

- Email
- Phone
- Date of Birth

Optional (User-provided, but not enforced by system):

- Full Name
- Initials
- Personal ID
- Preferred ID Type
- Location
- Registration Date

Later, when the user proceeds to 3.A (access), they are prompted to select at least 3 fields to verify their identity. These selections may include both mandatory and optional fields.

This distinction is important: 'mandatory' does not mean immutable or final ? only that the system expects them for processing. The actual user acknowledgement and correction path determines finality.

BLOCK STRUCTURES

1. Registration Block (1A)

{...sample JSON...}

2. Update Block (2A)

{...sample JSON...}

TIMESTAMP & SEQUENCING STRATEGY

- Use datetime.utcnow().isoformat() to generate timestamp for every block.
- Each block has:
  - index (global blockchain index)
  - previous_hash (ensures order integrity)
  - Internal user_id reference to identify user-specific chains.

USER JOURNEY SNAPSHOT

```
[ Registration Block (1A) ]
       ??????????????????
                      ?
              (optional updates)?
                      ?
             [ Update Block (2A) ]
```

LOGGING SCHEME

Log File	Description
userregister.log	Captures initial registration events
userupdate.log	Tracks corrections/updates to user data

NEXT STEPS

- Implement register_user() and update_user_info() methods.

- Add validation rules for mandatory fields before block generation.

- Auto-log block creation events.

- Prepare the system to later branch into 3.A (Access) after full 1A/2A logic is locked.

Status:

? Actively working on Phase 1 and 2 of user identity blockchain framework (1A and 2A).

? 3.A (Access & Smart Contract) deferred until after full 1A/2A implementation.

Now Open for Public Comments:

This architectural challenge is part of an open-source review to improve user identity finalization logic in blockchain systems. Contributions and proposals are welcome.

Lead Contributor: Abhishek Kumar

Project: CredentialVault

Date: April 2025