The human-blockchain identification system aims to provide a secure and immutable way to handle identities on a blockchain. It replaces traditional, centralized forms of identification with a decentralized model. Here's how it would generally work and be used, step by step:

+Step 1: User Onboarding
1. +Create Cryptographic Keys: A user begins by generating a public and private key pair. The public key is openly shared, while the private key is kept secret and used to sign transactions to prove ownership.

+Step 2: Creating a Digital Identity
2. +Input Identity Information: The user provides identity information which could include a username, email, and potentially other attributes. This data is input through a user interface, such as a web application.

3. +Hash Identity Data: The application hashes the identity information to create a unique digital fingerprint. This hash is what will be stored on the blockchain, not the actual data, to preserve privacy.

+Step 3: Storing Identity on the Blockchain
4. +Create Identity Record: The hash and the user's public key are bundled into an identity record.

5. +Store on Blockchain: This identity record is sent to a smart contract on the blockchain. The smart contract records the hash associated with the public key, effectively creating a blockchain-based identity.

+Step 4: Verification
6. +Identity Verification: When identity verification is required, the user provides their identity information to the verifier.

7. +Hash Verification: The verifier hashes the user’s information and checks the blockchain to confirm the hash matches the one stored. If it matches, the identity is verified.

+Step 5: Interaction with Other Systems
8. +Perform Transactions: With a verified identity, users can sign transactions using their private key, and other parties can trust these transactions since they can verify the signer's identity on the blockchain.

9. +Update or Revoke: If users need to update their identity information, they can send a new transaction to the smart contract with a new hash. To revoke access, they can remove or replace their identity hash in the smart contract.

+User Experience
'User Interface (UI): Users interact with the system through a web or mobile application that abstracts away the complexity of blockchain interactions.
'Wallet Integration: Users utilize digital wallets to manage their keys and sign transactions securely.
'Notifications: The system notifies users of important events, like when their identity is verified or when they need to confirm a transaction.

+Innovation Aspects

'Immutability: Once an identity record is on the blockchain, it cannot be altered, providing strong protection against identity theft and fraud.
'Decentralization: Removes the need for a central authority to manage identities, potentially reducing costs and increasing user control over personal information.
'Privacy: By using hashes instead of actual data and leveraging privacy-preserving technologies like zero-knowledge proofs, users' personal data can remain private.

+Step-by-Step Use Case Example
1. +Alice Signs Up: Alice generates a key pair, inputs her identity information, and creates an account through the dApp interface.
2. +Hashing: The dApp hashes Alice’s information and creates an identity record.
3. +Blockchain Record: The dApp sends a transaction to the smart contract to store Alice’s identity hash with her public key.
4. +Verification Request: Bob needs to verify Alice’s identity for a service.
5. +Alice Provides Data: Alice inputs her identity information into Bob's verification system.
6. +Verification Process: Bob’s system hashes the information provided by Alice and checks it against the hash stored on the blockchain.
7. +Confirmation: If the hashes match, Bob is confident in Alice's identity and can proceed with the transaction.

This is a simplified representation, and actual implementations could be more complex, particularly when it comes to protecting user privacy and ensuring compliance with regulations.