

## Chapter 16:

### Decentralized Blockchain Technology and Police Accountability

Decentralized blockchain technology has the potential to revolutionize the way we handle public documents related to police accountability, such as public complaints against police personnel. In traditional systems, these documents can be obscured or unreported, making it difficult to hold police officers accountable for misconduct. However, by using decentralized blockchain technology, we can create a more transparent and accountable system that is resistant to manipulation and censorship.

Decentralized blockchain technology is a distributed ledger system that allows for secure, transparent, and tamper-proof record-keeping. This technology can be used to create a decentralized database of public complaints against police personnel, which can be accessed by the public and law enforcement agencies alike. By using blockchain technology, we can ensure that these documents are secure and tamper-proof, and that they can be easily accessed by anyone who needs them.

One potential use case for decentralized blockchain technology in police accountability is the creation of a public oversight committee. This committee could use blockchain technology to create a public database of police complaints and disciplinary actions, which would be accessible to the public and law enforcement agencies alike. This would increase transparency and accountability in policing, and would provide a mechanism for community input into policing practices.

Another potential use case for decentralized blockchain technology in police accountability is the creation of a system for reporting police misconduct. By using blockchain technology, we can create a system that is resistant to censorship and manipulation, and that allows for secure and anonymous reporting of police misconduct. This would make it easier for individuals to report incidents of police misconduct, and would help to ensure that these incidents are properly investigated and addressed.

However, there are also challenges associated with the implementation of decentralized blockchain technology in police accountability. One such challenge is the need to ensure that the technology is accessible to everyone, including those who may not have access to the internet or modern decentralized web3 technology. Additionally, there may be concerns about the privacy and security of individuals who report incidents of police misconduct, and steps must be taken to ensure that their identities are protected.

Overall, decentralized blockchain technology has the potential to play an important role in police accountability, by increasing transparency, accountability, and accessibility. While there are challenges associated with its implementation, the potential benefits are significant, and it is an

area that deserves further exploration and investment. The code below is written in a OOP or object oriented programming language called PHP. The final implementation however, is likely to use Python, which seems to be the preferred OOP language for handling both blockchain ledgers and modern artificial intelligence APIs such as GPT 4's API created by Open AI with funding from microsoft. Both PHP and Python are types of server-side scripting languages used to move information between web servers and personal devices.

```
<?php

if(isset($_POST['submit'])){

    $submission_date = date('Y-m-d H:i:s');
    $to = $_POST['email'];
    $from = "admin@newlondonvoice.com";
    $firstname = $_POST['firstName'];
    $lastname = $_POST['lastName'];
    $address = $_POST['address'];
    $email = $_POST['email'];
    $complaint = $_POST['complaint'];
    $subject = $_POST['complaint'];
    $ipAddress = $_POST['ipAddress'];
    $documentnumber = $_POST['documentNum'];

    // Create an array of form data
    $data = array(
        'date_submitted' => $submission_date,
        'first_name' => $firstname,
        'last_name' => $lastname,
        'email' => $email,
        'address' => $address,
        'complaint' => $complaint,
        'ip_address' => $ipAddress,
        'int_doc_number' => $documentnumber,
        'block_added' => false // new boolean variable
    );

    // Append the form data to the complaintdata.json file
    $file = 'complaintdata.json';
    $current_data = file_get_contents($file);
    $array_data = json_decode($current_data, true);
```

```

    // Check if there are two submitted forms that have not been added to
a block
    $count = 0;
    foreach ($array_data as $form) {
        if ($form['block_added'] == false) {
            $count++;
        }
    }
    if ($count == 2) {
        // Create a new block by reading the last 3 forms in
complaintdata.json and save it to blockchain.json
        $last_three_forms = array_slice($array_data, -3);
        $block = array('forms' => $last_three_forms);
        $block_data = json_encode($block);

        $blockchain_file = 'blockchain.json';
        $current_blockchain_data = file_get_contents($blockchain_file);
        $array_blockchain_data = json_decode($current_blockchain_data,
true);
        $array_blockchain_data[] = $block_data;
        $final_blockchain_data = json_encode($array_blockchain_data);
        file_put_contents($blockchain_file, $final_blockchain_data);

        // Set the 'block_added' variable to true for each form in the new
block
        foreach ($last_three_forms as &$form) {
            $form['block_added'] = true;
        }
        unset($form);
    }

    // Add the new form data to the complaintdata.json file
    $array_data[] = $data;
    $final_data = json_encode($array_data);
    file_put_contents($file, $final_data);

    // Increment the count of forms written and check if it's evenly
divisible by 3
    $forms_written = count($array_data);
    if ($forms_written % 3 == 0) {

```

```
        // Run hashMaker.php to read complaintdata.json and create a new
block
        exec('php hashMaker.php');
    }
    //          $form['block_added'] = true;

    $headers = 'From: '.$from . "\r\n";
    $body = "First name : ".$firstName . "\r\n" .
        "Address : ".$address . "\r\n" .
        "Message : ".$complaint;

    if(mail($to, $subject, $body, $headers)){
        echo "Mail Sent!";
    } else {
        echo "Failed To Send Mail";
    }
}
```