


HF- ABAC with FL learning

 Introduction

User Perspective

An user can have a **owner (admin)**, **family members (roles)**, **Guest (roles)**, **Neighbors(roles)**, **Delivery man (temp roles)**. Each individual will have different access. Lets say, if an user has multiple house, she needs different access for each. A **neighbor** will have access certain of the day. Lets say no one at home and a delivery came, neighbor can give access to the delivery man and it will be store in Hyper-ledger fabric ledger. But, it will be number attempts.

Access Control Perspective

The homeowner is the primary owner (admin) and assigns different roles to individuals: family members, guests, neighbors, and temporary roles such as a delivery man. For example, **a neighbor might have restricted access to certain areas** only during **specific times of the day**. When **no one is at home and a delivery arrives**, the **neighbor can temporarily authorize access** for the delivery man through the ABAC system. This authorization is context-aware—it factors in the **roles of the individuals** involved, the **current occupancy of the house**, and the **time of day**. The **temporary access** granted to the delivery man is then recorded on the Hyperledger Fabric ledger along with the number of **access attempts**, ensuring transparent and immutable tracking of all transactions. This setup allows the homeowner to maintain granular control over who accesses which property and under what conditions, while also providing an auditable trail of all access events.

Device Perspective

When a camera initiates a **model update**, an attribute-based access control (ABAC) system verifies several context-aware attributes: the camera's operational status (ensuring it's not malfunctioning), environmental conditions (such as adequate lighting and weather conditions affecting image quality), and network connectivity. Additionally, each **camera's update is tagged with its location** and role (for example, indoor versus outdoor coverage), and the update is only accepted if it occurs during a predefined time window. Once validated, the update is securely transmitted and recorded on a Hyperledger Fabric ledger, which logs the update along with **metadata** like the **number of update attempts**. This comprehensive approach ensures that only reliable, contextually appropriate model updates are aggregated, enhancing the overall accuracy of the surveillance system while maintaining an auditable, tamper-proof record of all transactions.

