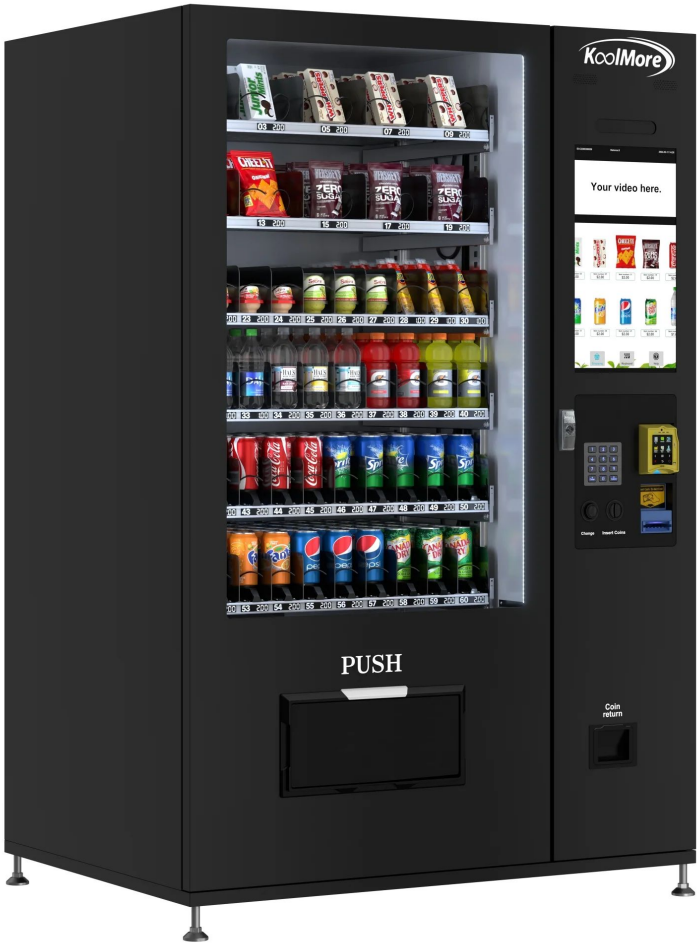
CSC 116 Blockchain Smart Contracts / Blockchain Applications



The contracts are codes


BLOCKCHAIN
NETWORK
SELLER
BLOCKCHAIN
NETWORK
BUYER
SMART CONTRACT

What is a Smart Contract?

- **Definition:**
  Self-executing code stored on a blockchain that automatically runs when certain conditions are met.

- **Why are they special?**
  - No middlemen.
  - Immutable (can’t be changed).
  - Transparent (everyone can see the code).

Smart Contract

Check the new patient infos

```
type Patient struct {
    ID string `json:"id"`
    Age int `json:"age"`
    Gender string `json:"gender"`
    Precondition string `json:"precondition"`
    Visit_list []string `json:"visit_list"`
}

func (s *SmartContract) createPatient(APIstub shim.ChaincodeStubInterface,
        args []string) sc.Response {

    if (len(args) != 4){
        return shim.Error("Improper number of args")
    }

    id := args[0]
    age,_ := strconv.Atoi(args[1])
    gender := args[2]
    precondition := args[3]

    age_constraint_lower := 0
    age_constraint_upper := 120
    gender_constraint := ""
    precondition_constraint := ""

    if(age >= age_constraint_lower && age <= age_constraint_upper
    && strings.Contains(gender, gender_constraint)
    && precondition == precondition_constraint){
        newPatient := Patient{ID:id, Age:age, Gender: gender,
                              Precondition: precondition}
        patientAsBytes, _ := json.Marshal(newPatient)
        APIstub.PutState(id, patientAsBytes)
    } else {
        return shim.Error("Invalid Patient Info")
    }

    return shim.Success(nil)
}
```

Customized Smart Contract Use Cases

Role-Based Access Control:
Smart contracts can verify the roles of the users, only the right users can query/insert data.

Multi-Signature Authorization:
Smart contracts send notifications to all admin users for approval. If all admins approve, the query will proceed.