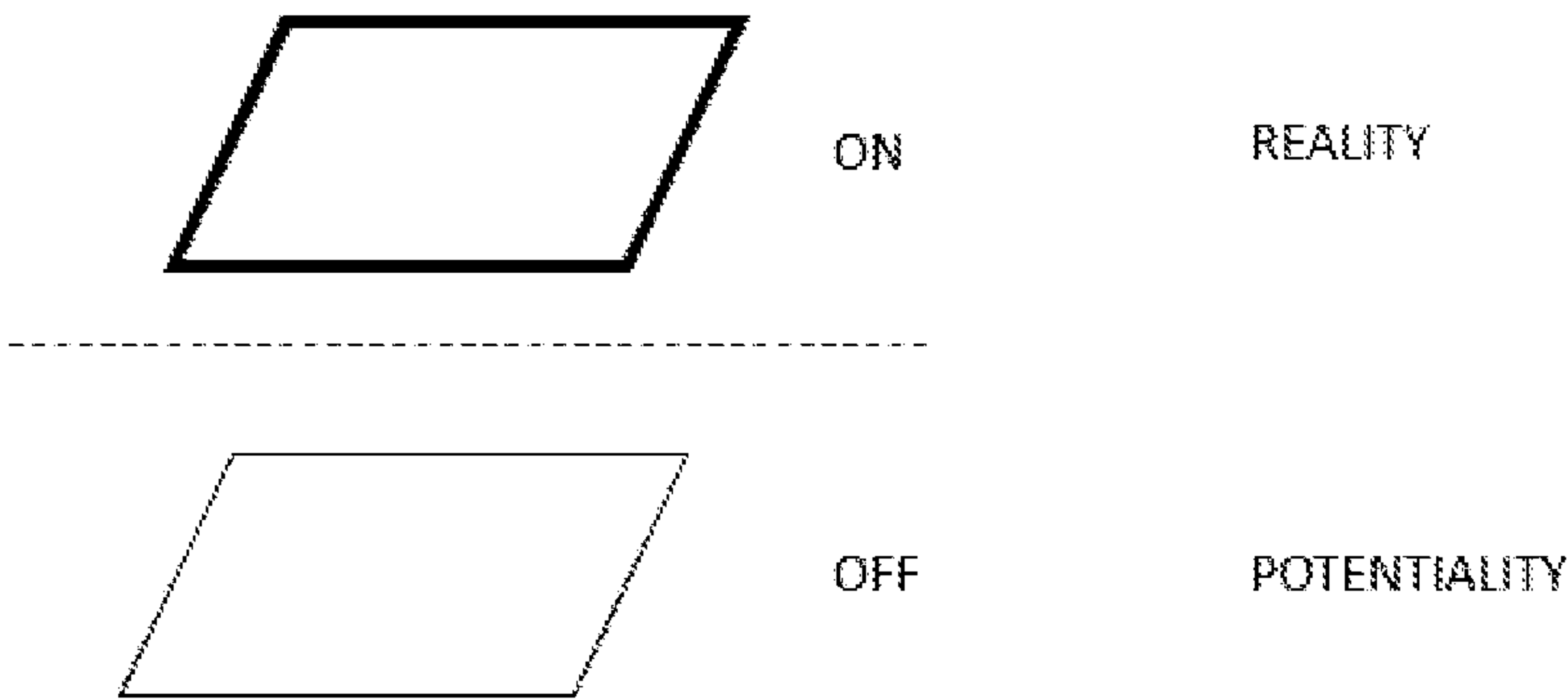


DRAWINGS

Drawing 1:



Drawing 2:

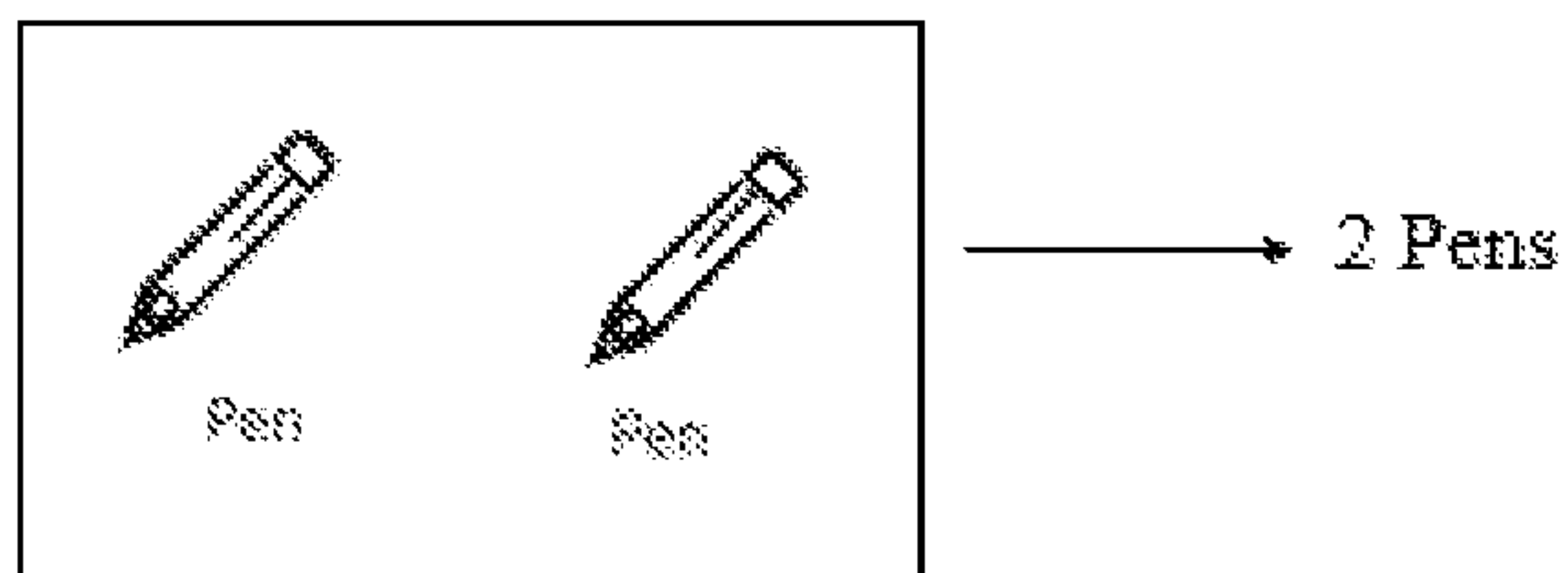


Drawing 3:



2/79

Drawing 4:

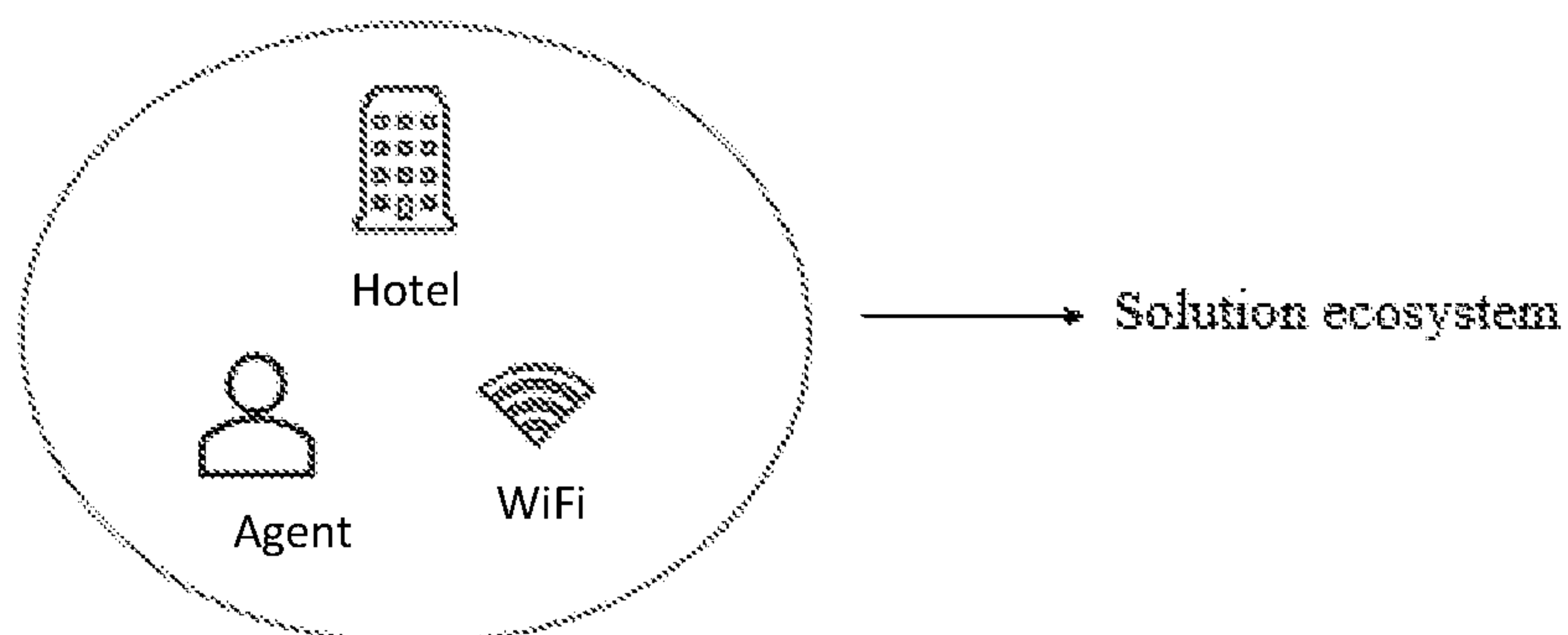


Drawing 5:

Book room

Check room availability. Get customer information. Get room stay information. Get transport information. Get laundry information. Get price. Book room.

Drawing 6:



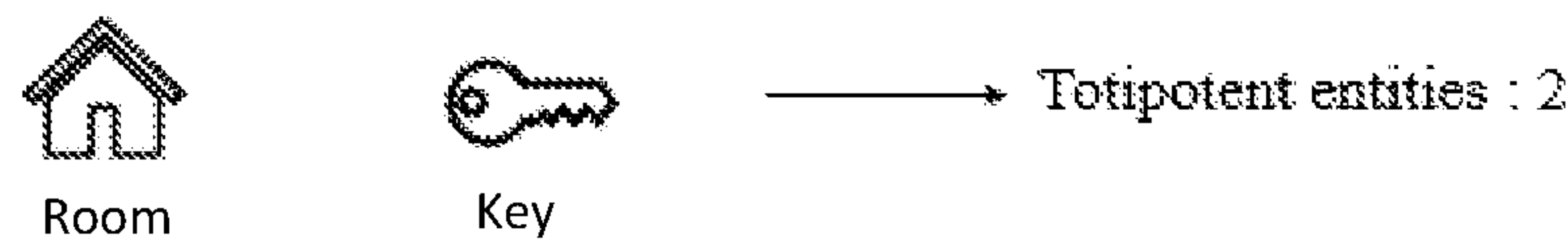
Drawing 7:

Book room.

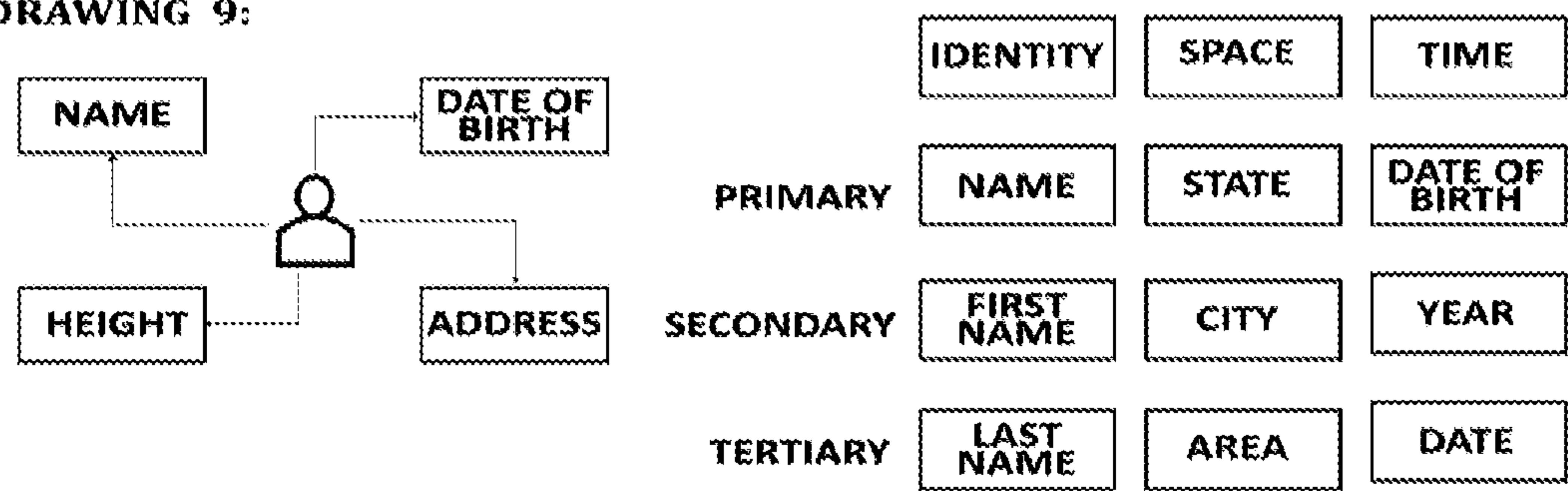
Check room availability. **Get customer information.** **Get room stay information.** **Get transport information.** **Get laundry information.** **Get place.** **Book hotel room.**

Note: Bold and underline represents reality state. In NSL, they are in red.

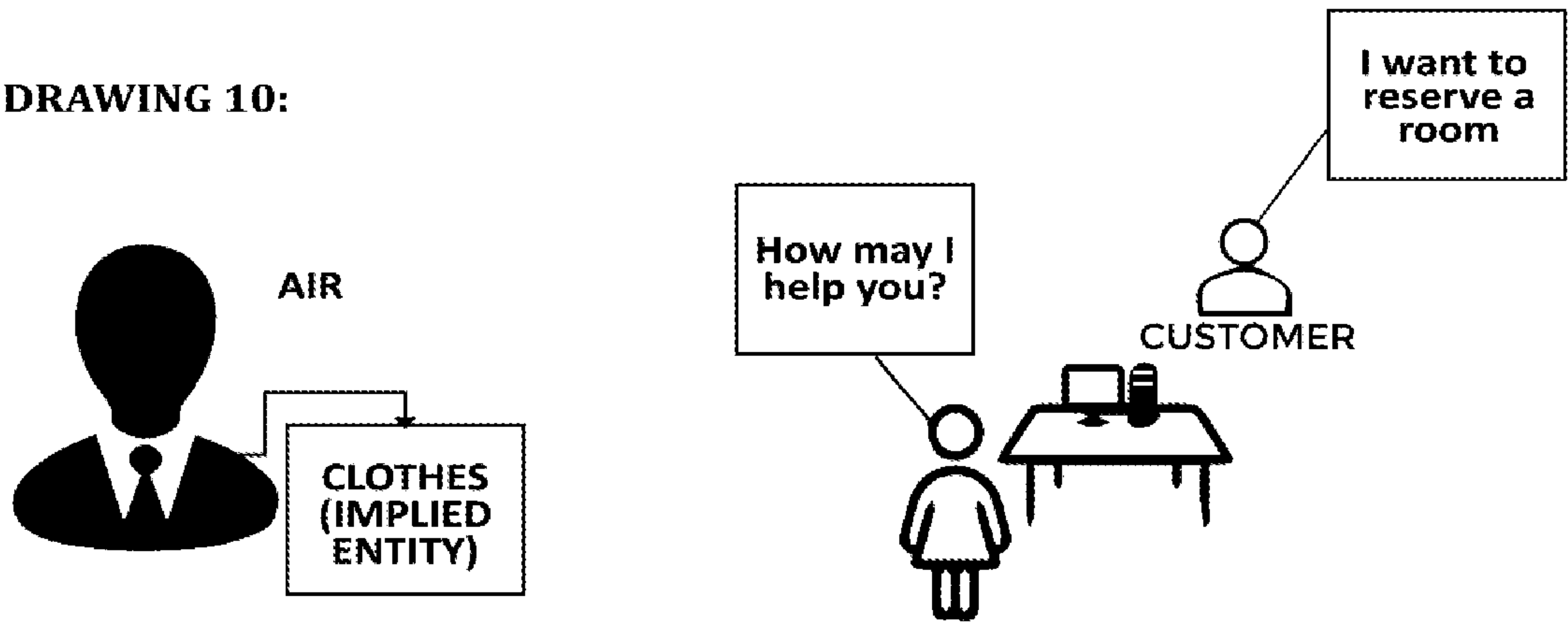
Drawing 8:



DRAWING 9:

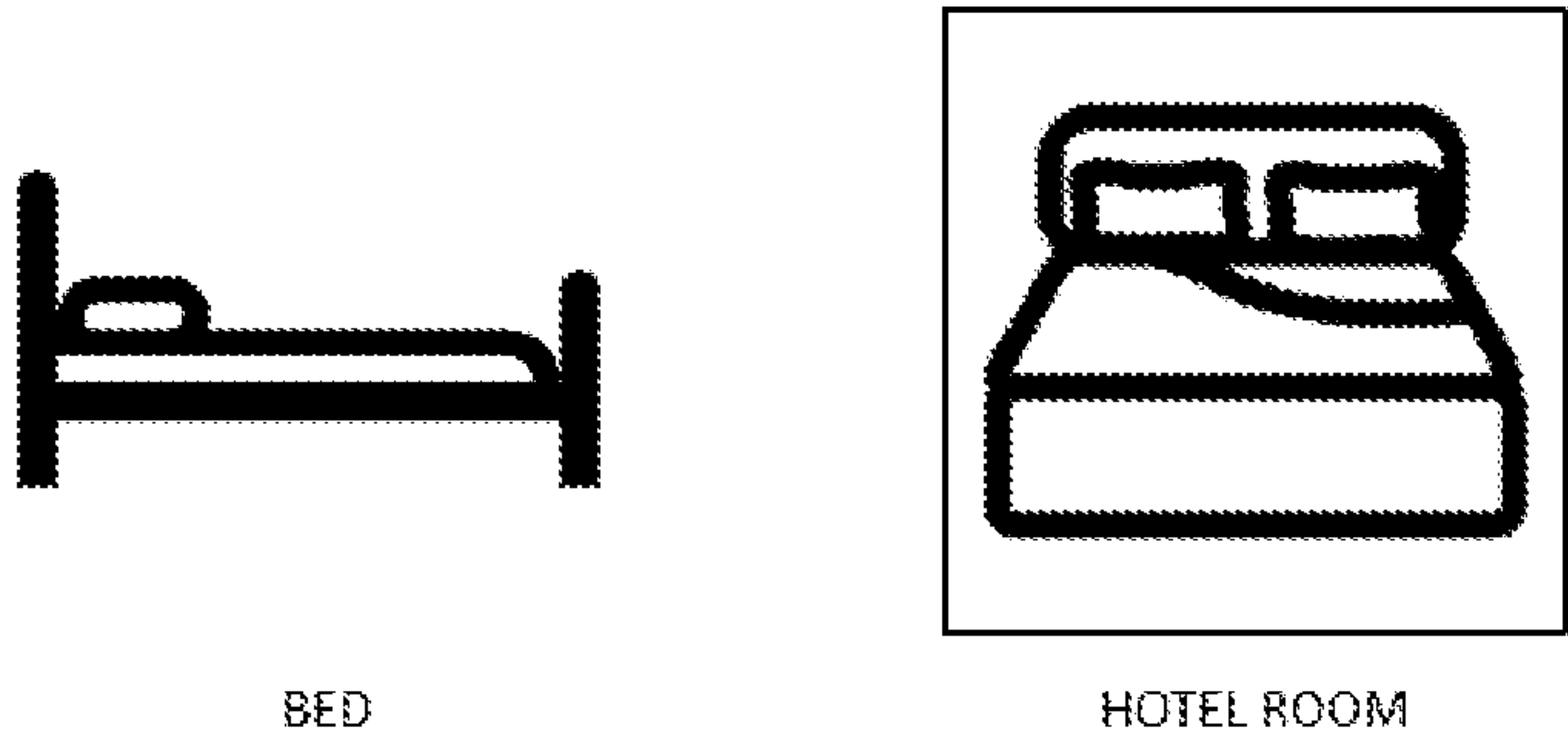


DRAWING 10:

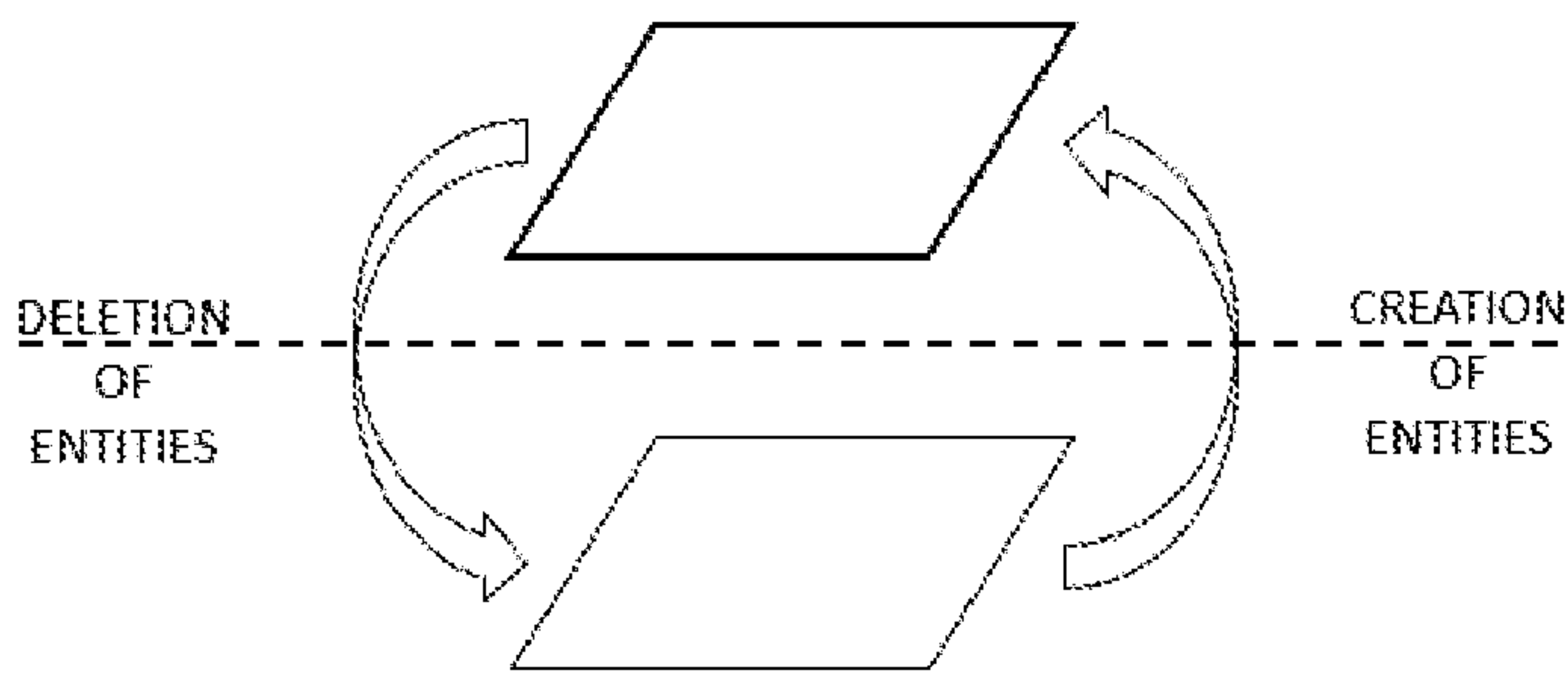


Drawing 11:

Frozen entities:

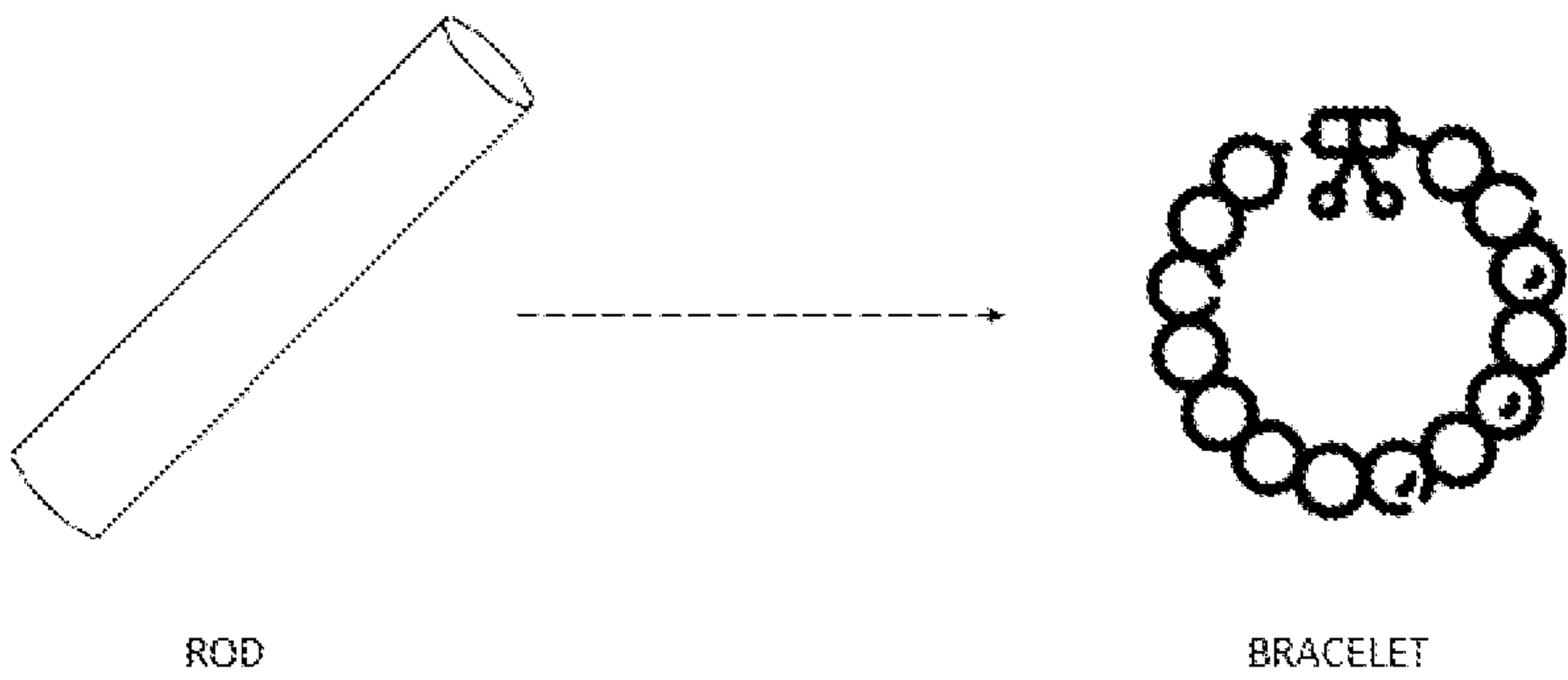


Drawing 12:

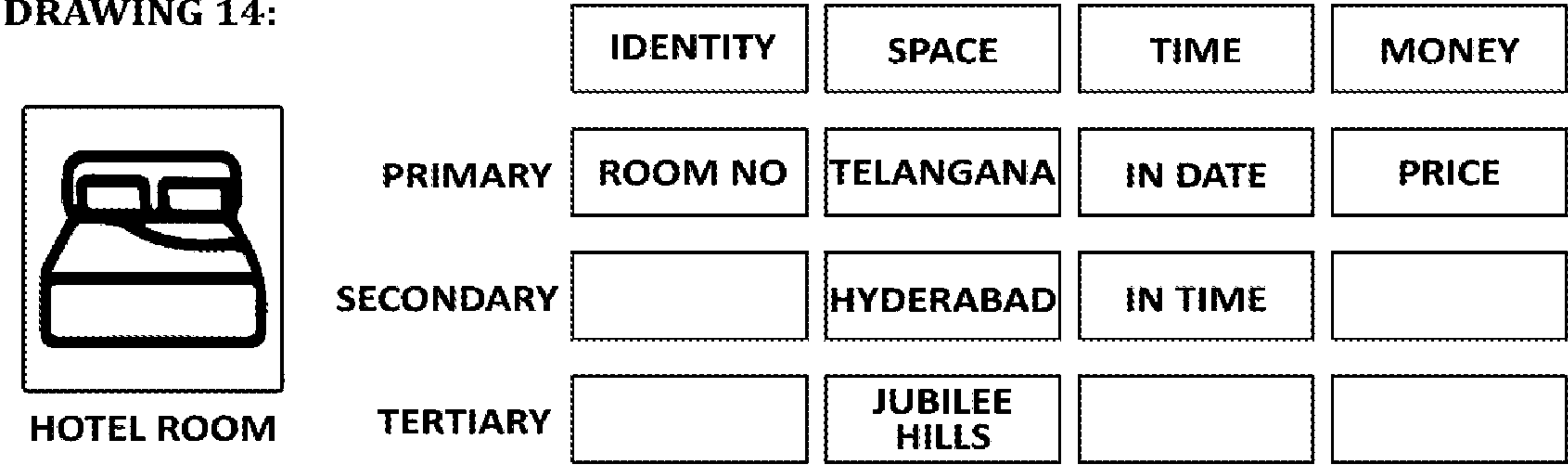


Drawing 13:

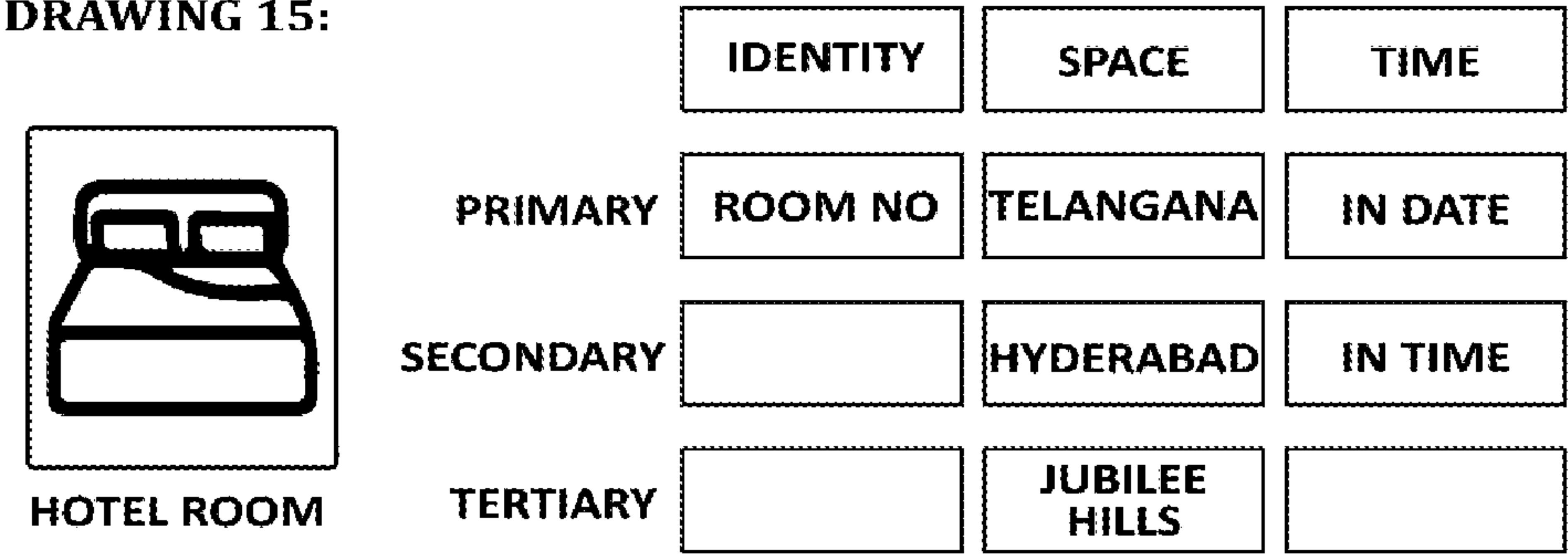
Modification of entities:



DRAWING 14:

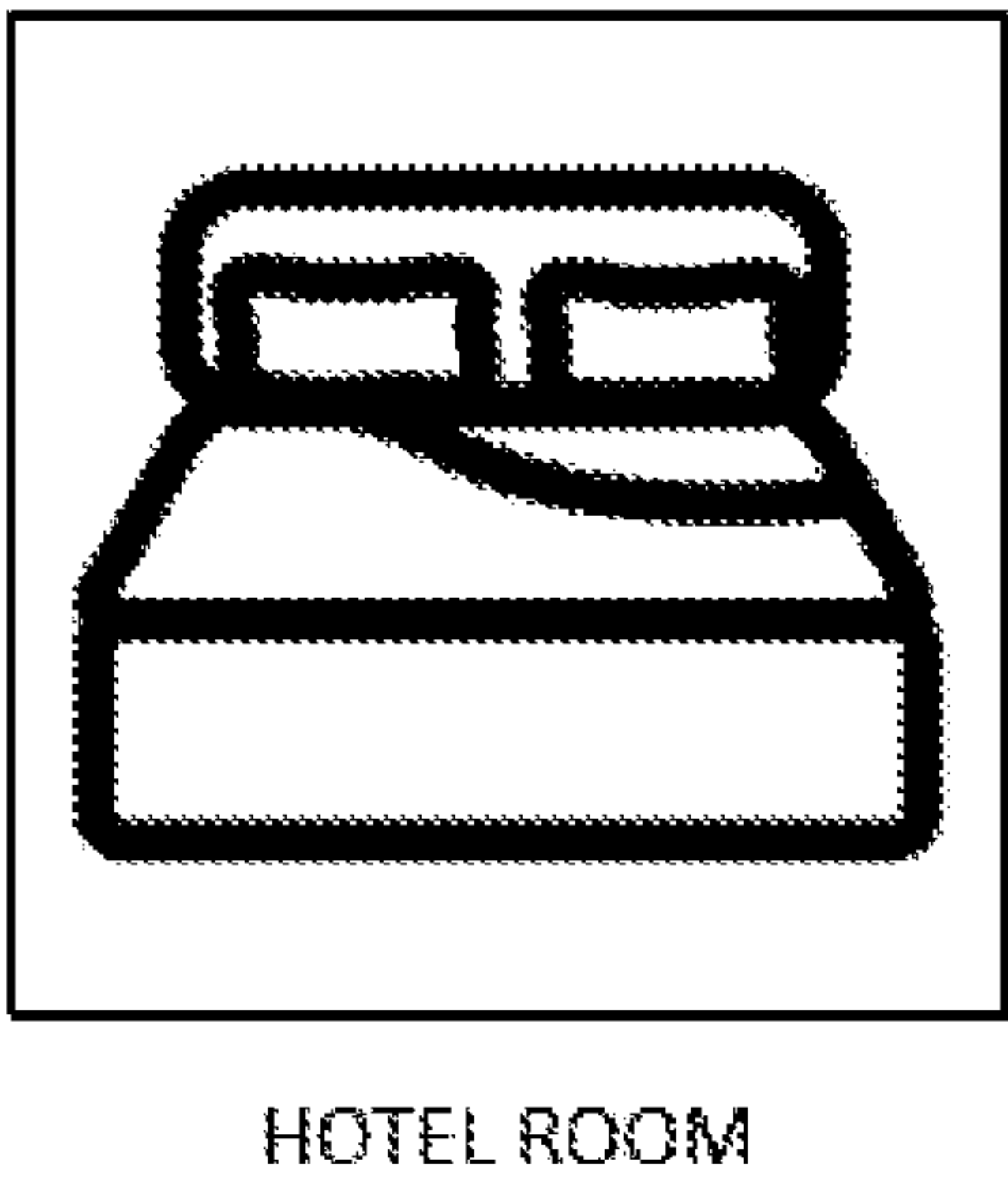


DRAWING 15:



Drawing 16:

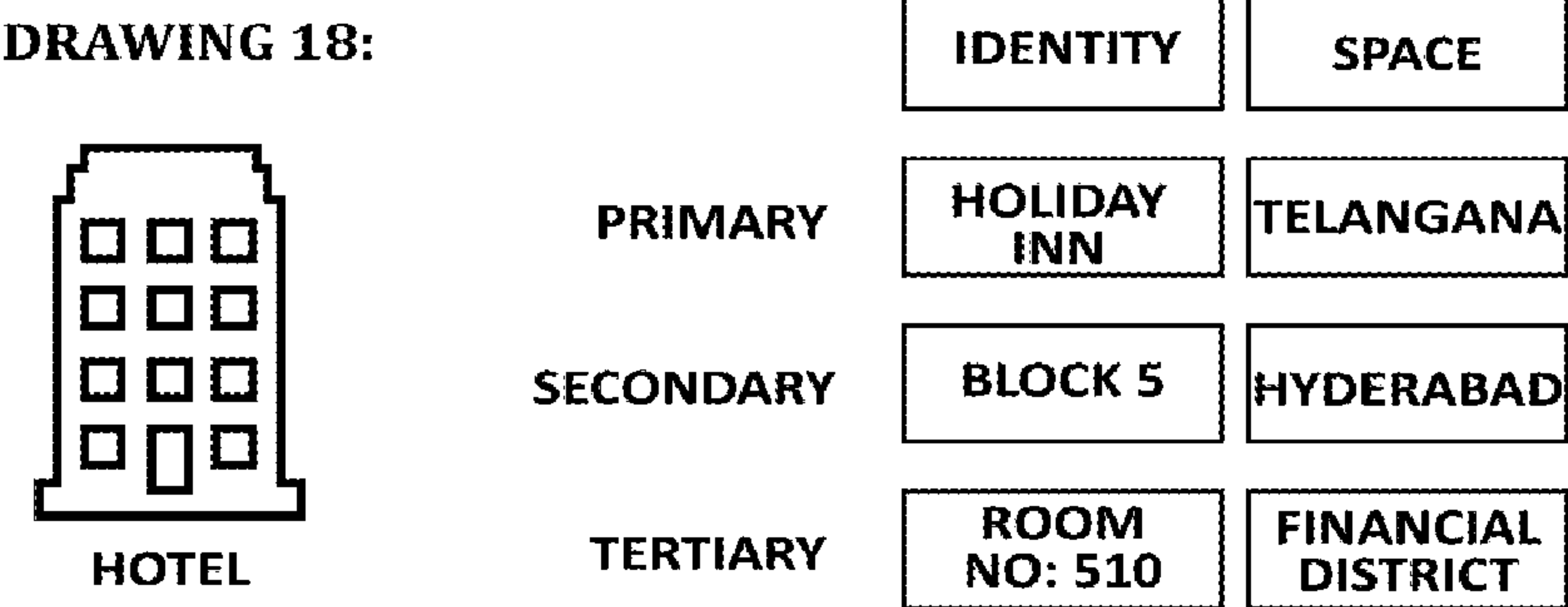
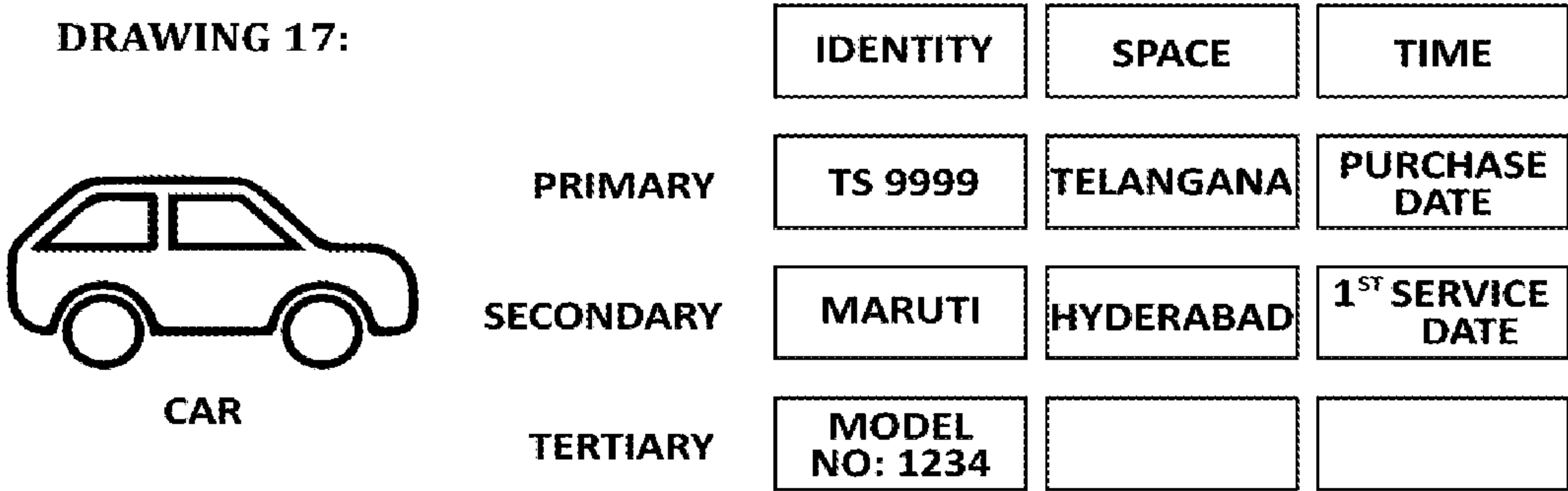
Example: All the hotel rooms will carry the following standard attributes:- Price, Wi-Fi



PRICE



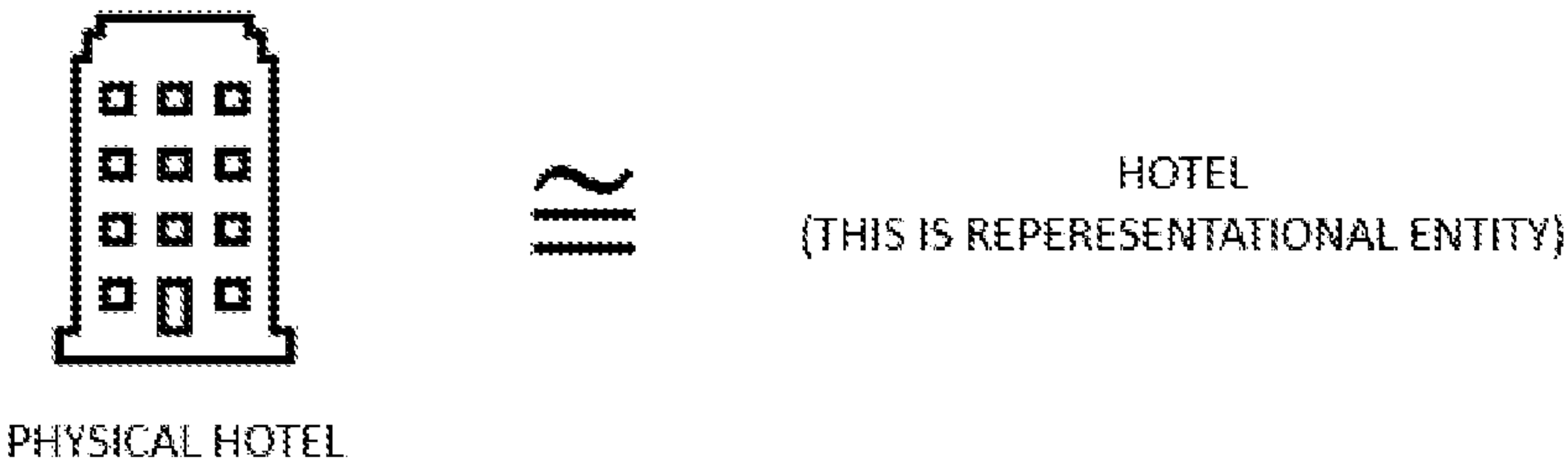
Wi-Fi



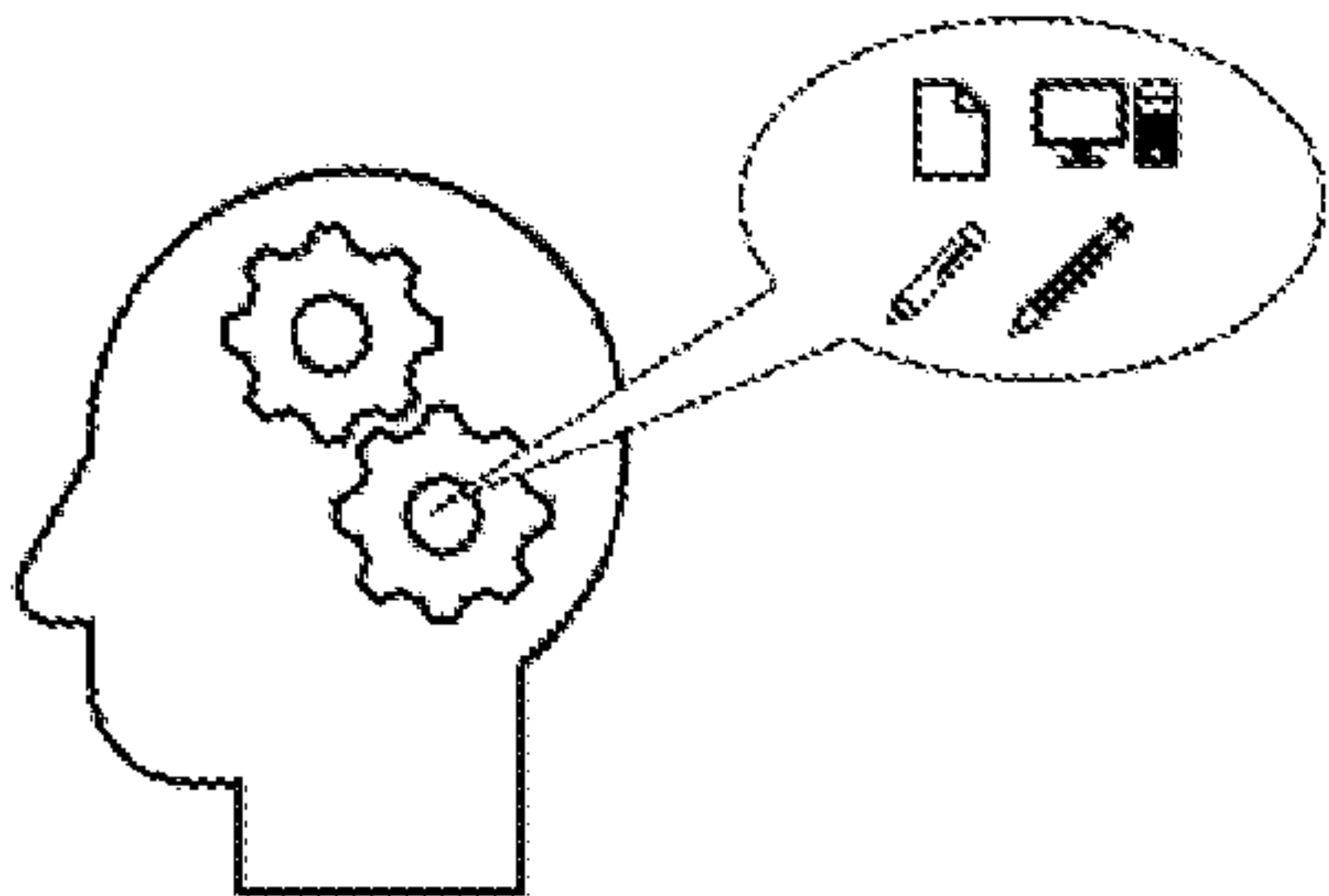
Drawing 19:



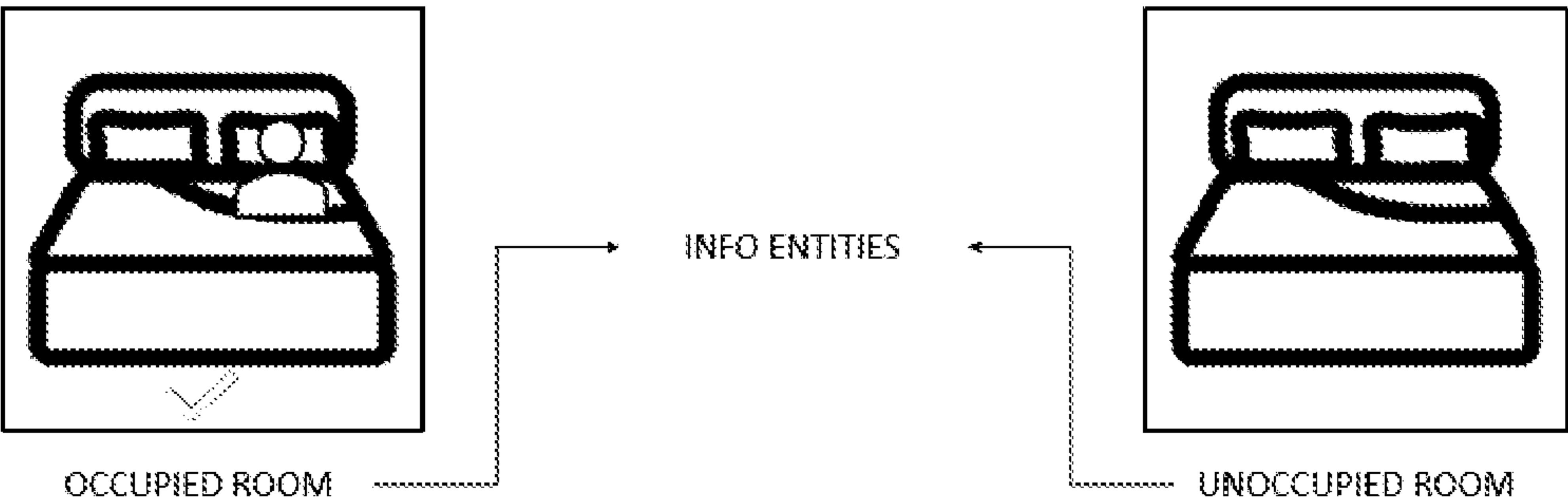
Drawing 20:



Drawing 21:



Drawing 22:



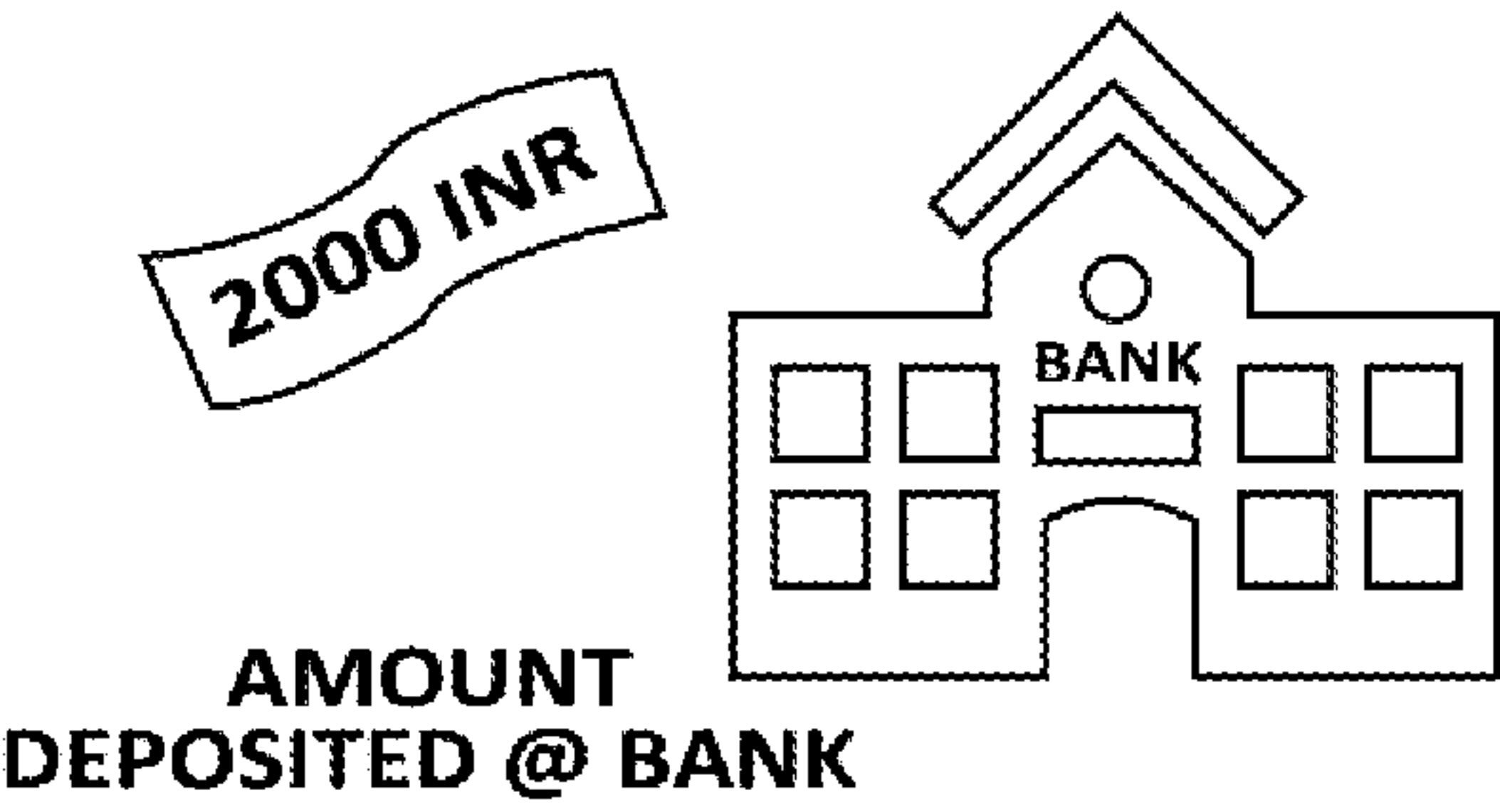
DRAWING 23:

Example:
18th Dec collection
from customers

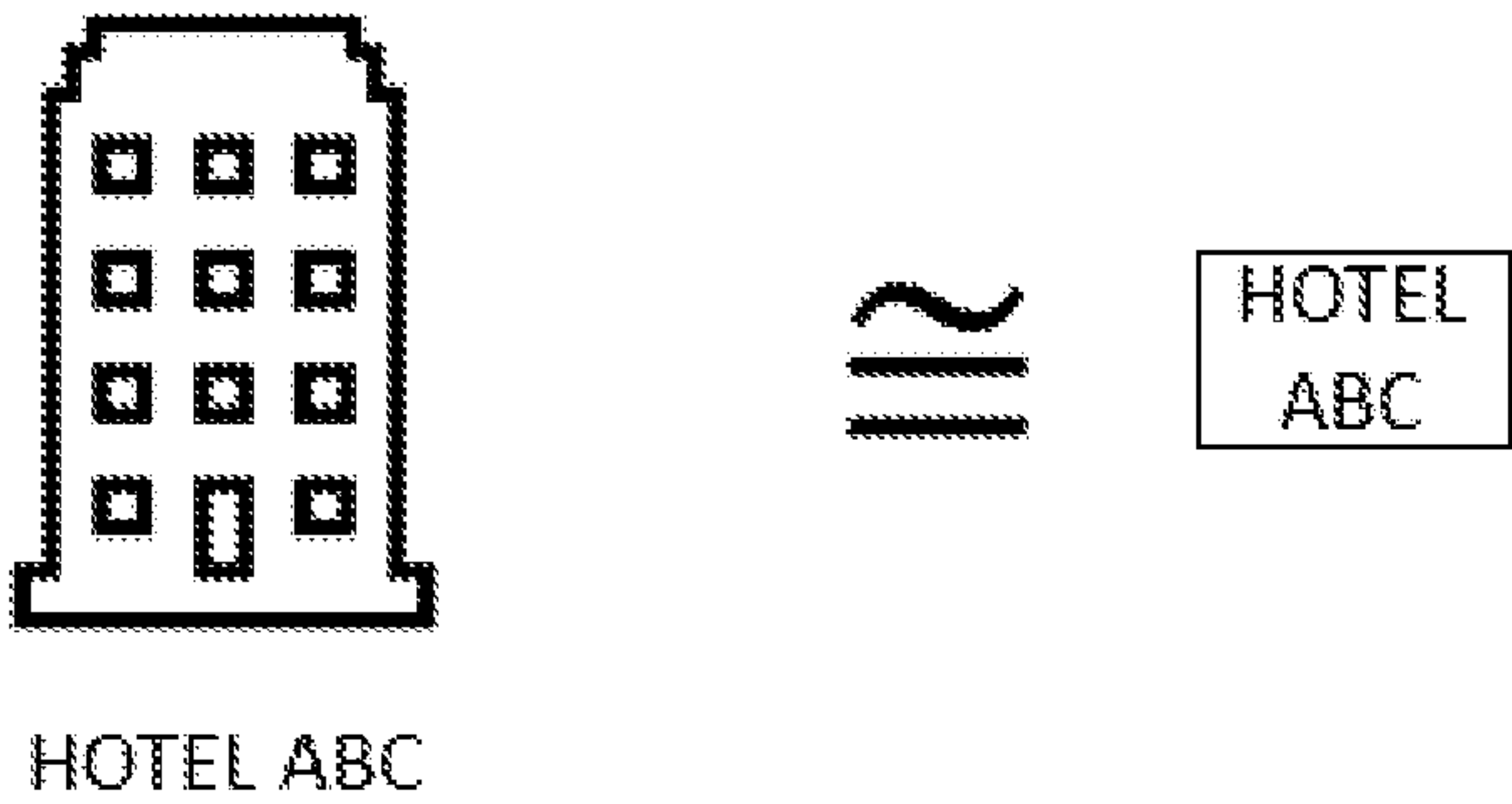
December

Su	Mo	Tu	We	Th	Fr	Sa
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

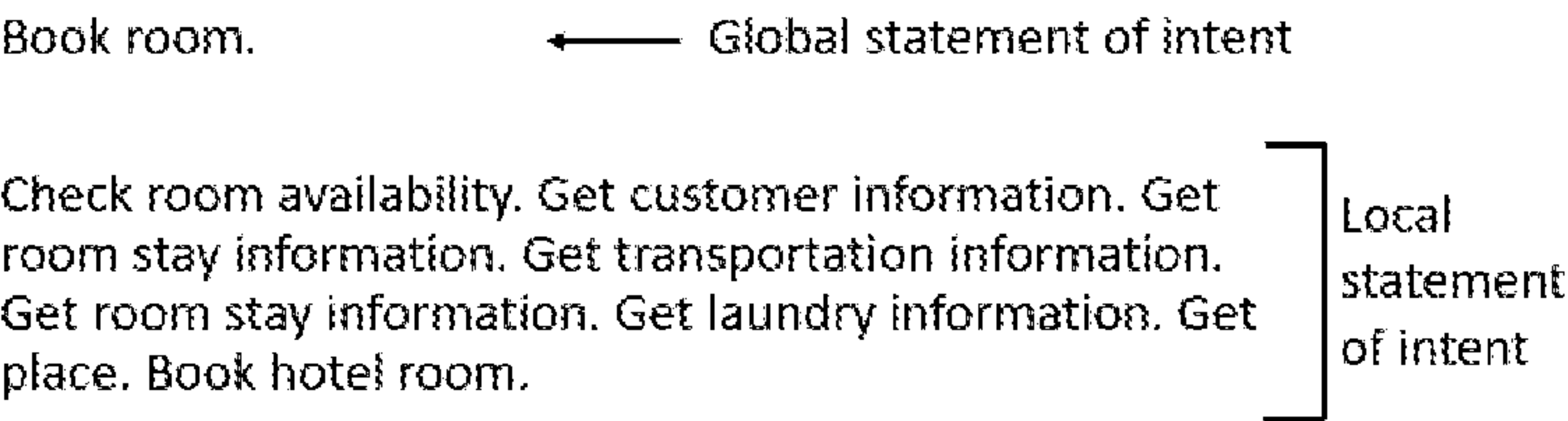
Total Collection = 2000 INR



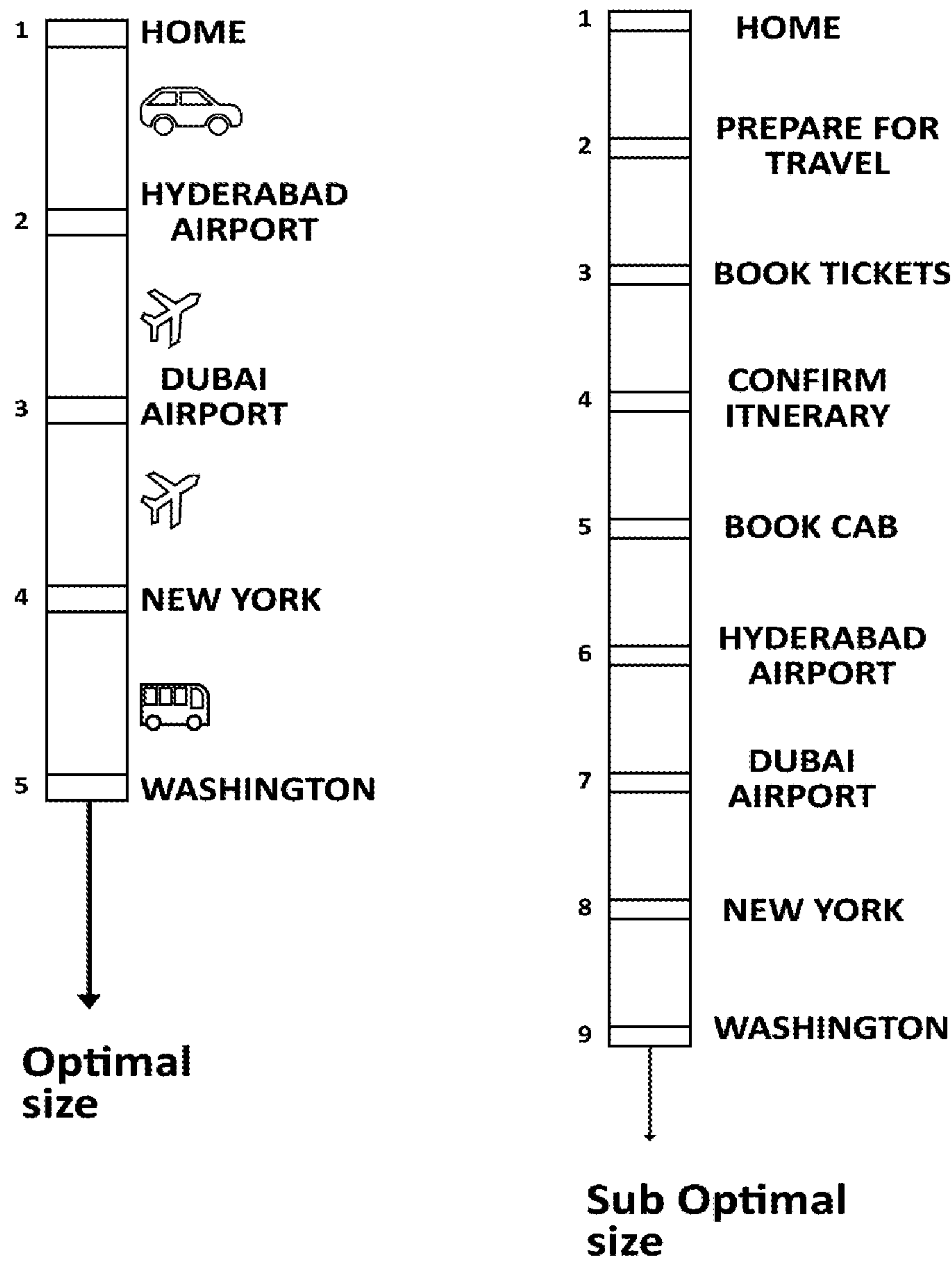
Drawing 24:



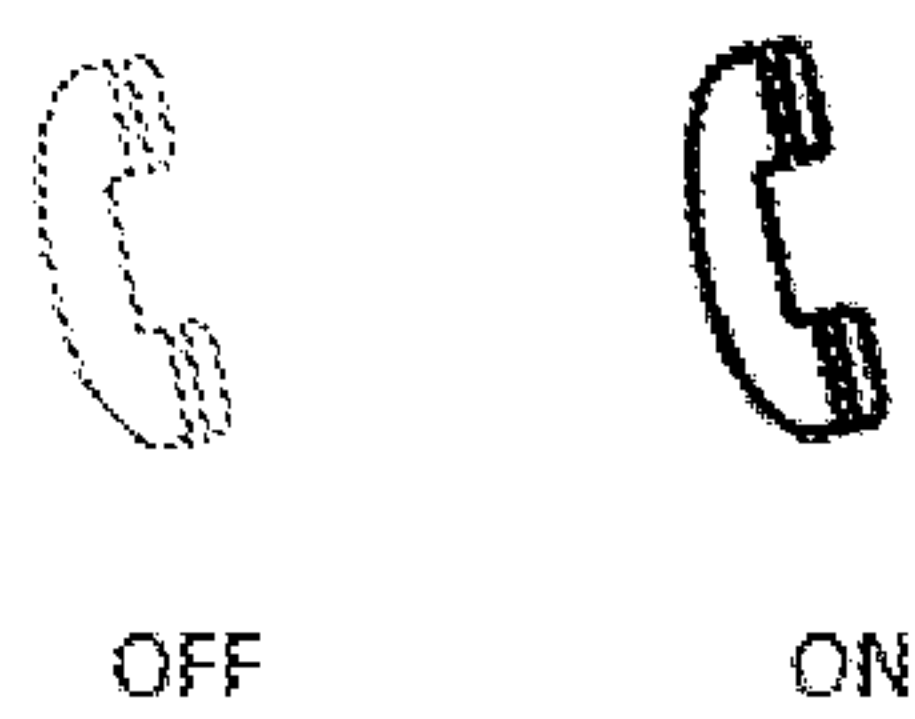
Drawing 25:



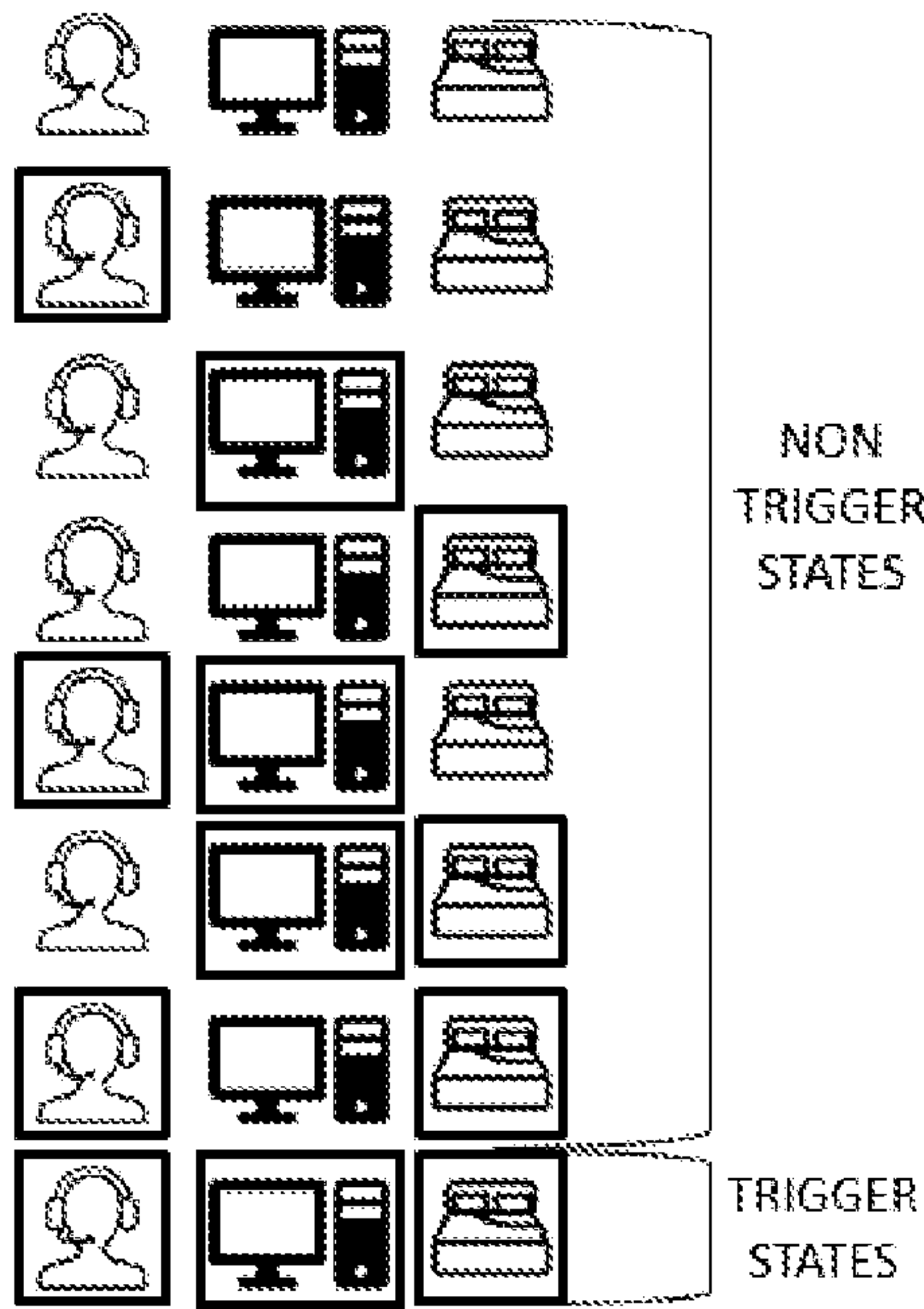
DRAWING 26:



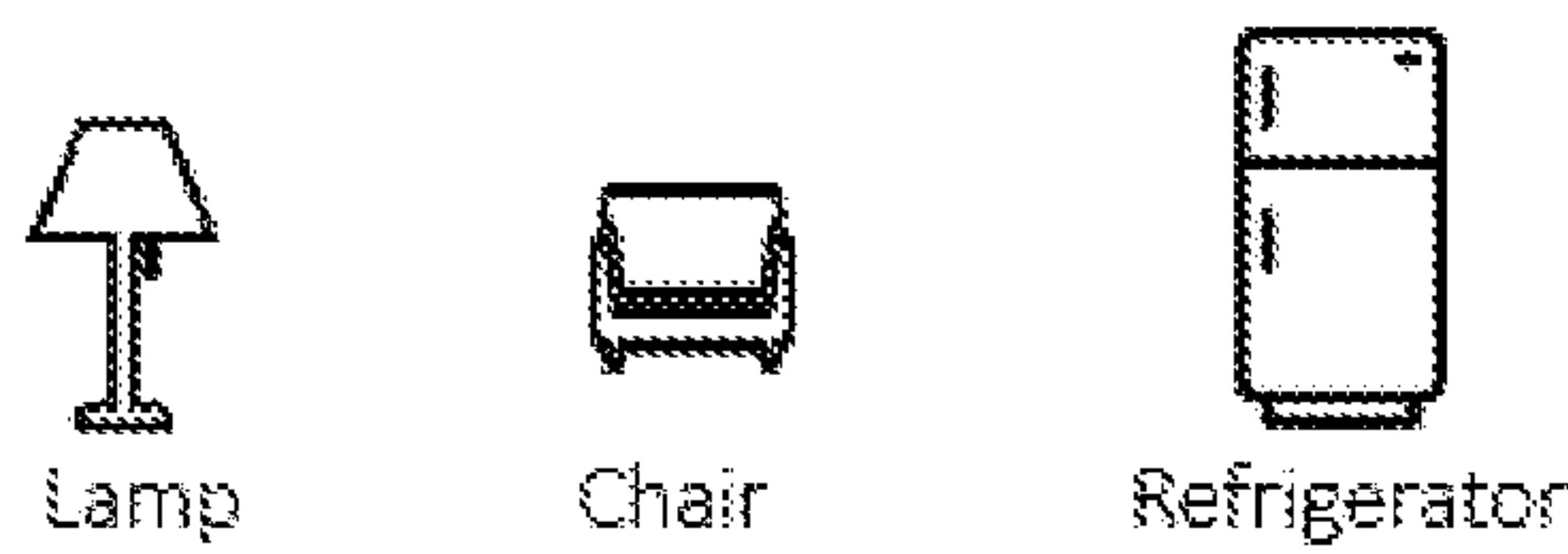
Drawing 27:



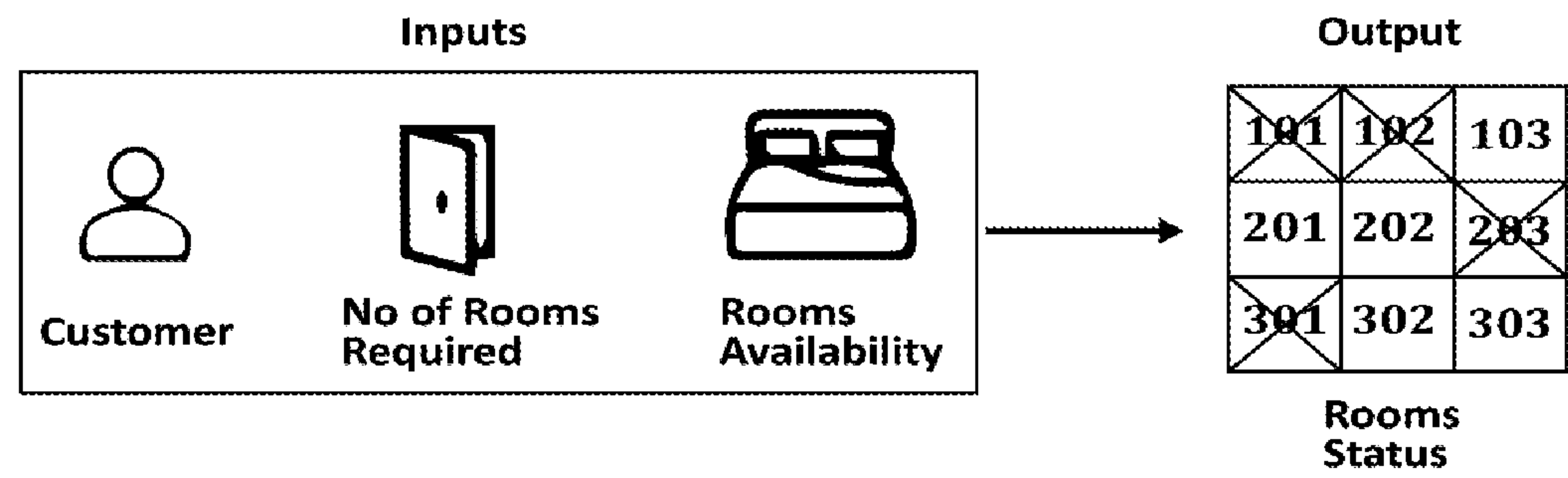
Drawing 28:



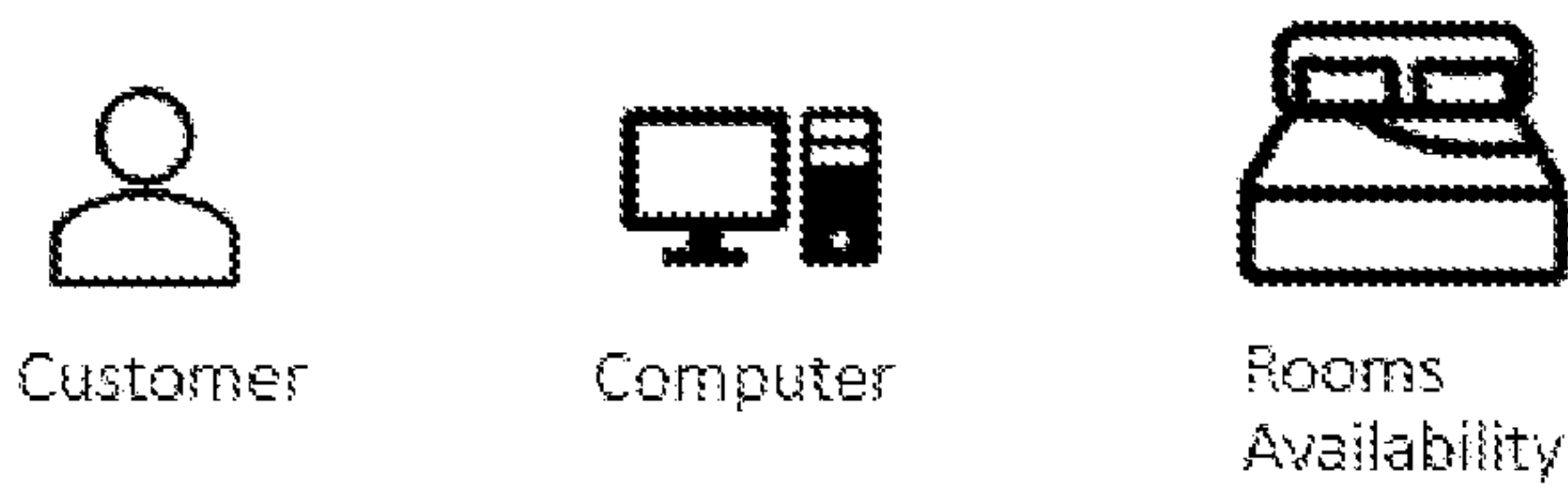
Drawing 29:



DRAWING 30:

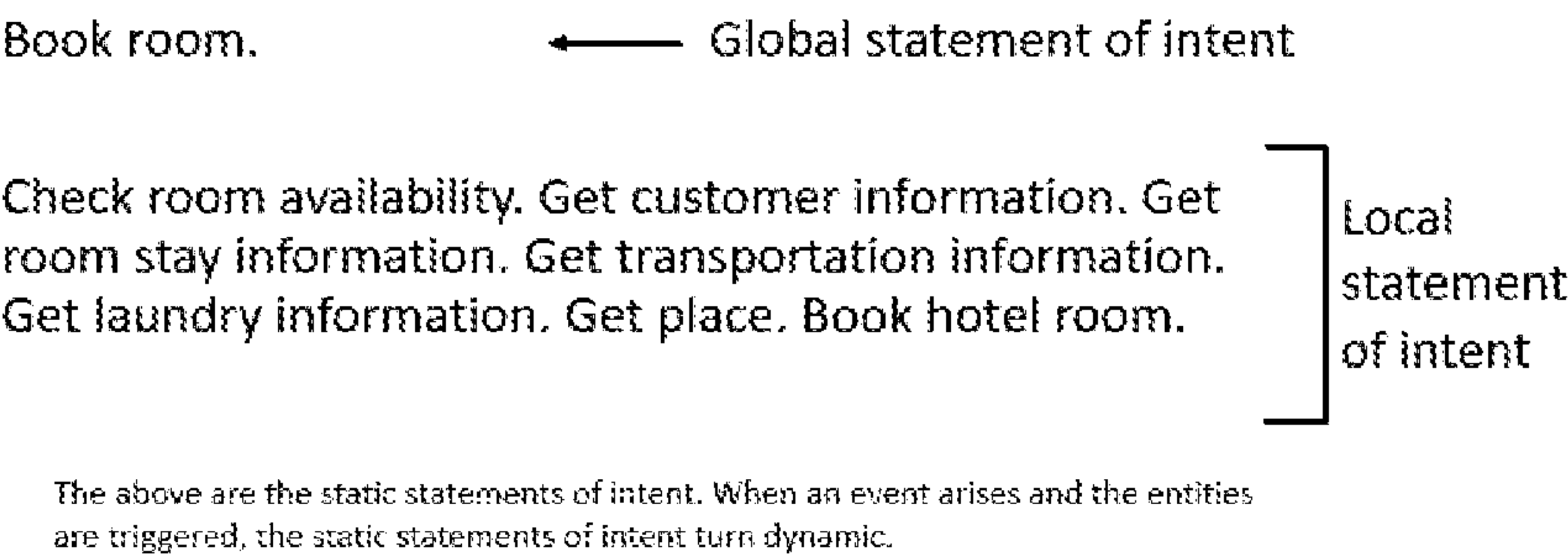


Drawing 31:

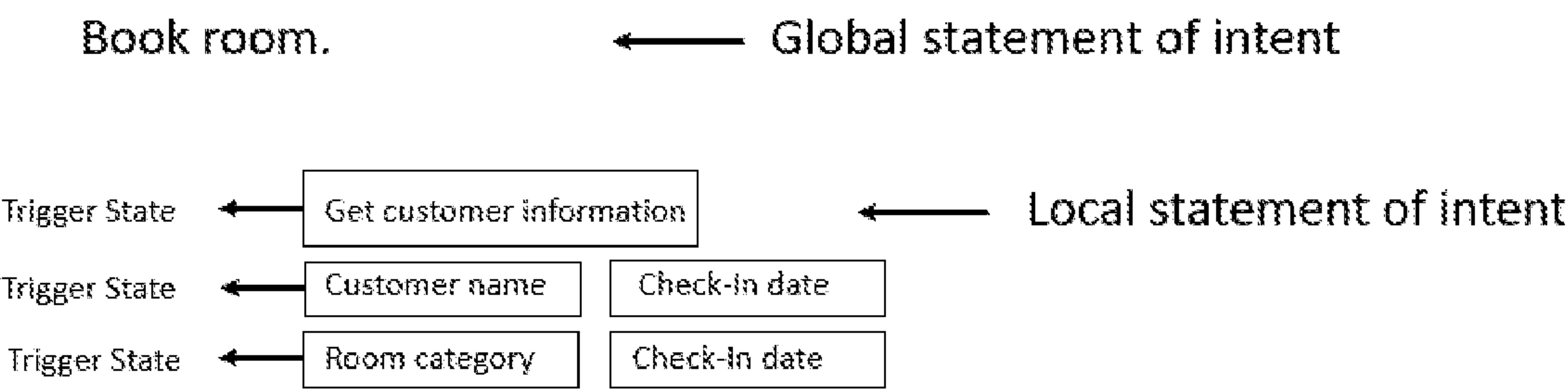


Natural Language: Customer intends to book a hotel room online
Natural Solutions Language: Customer book hotel room

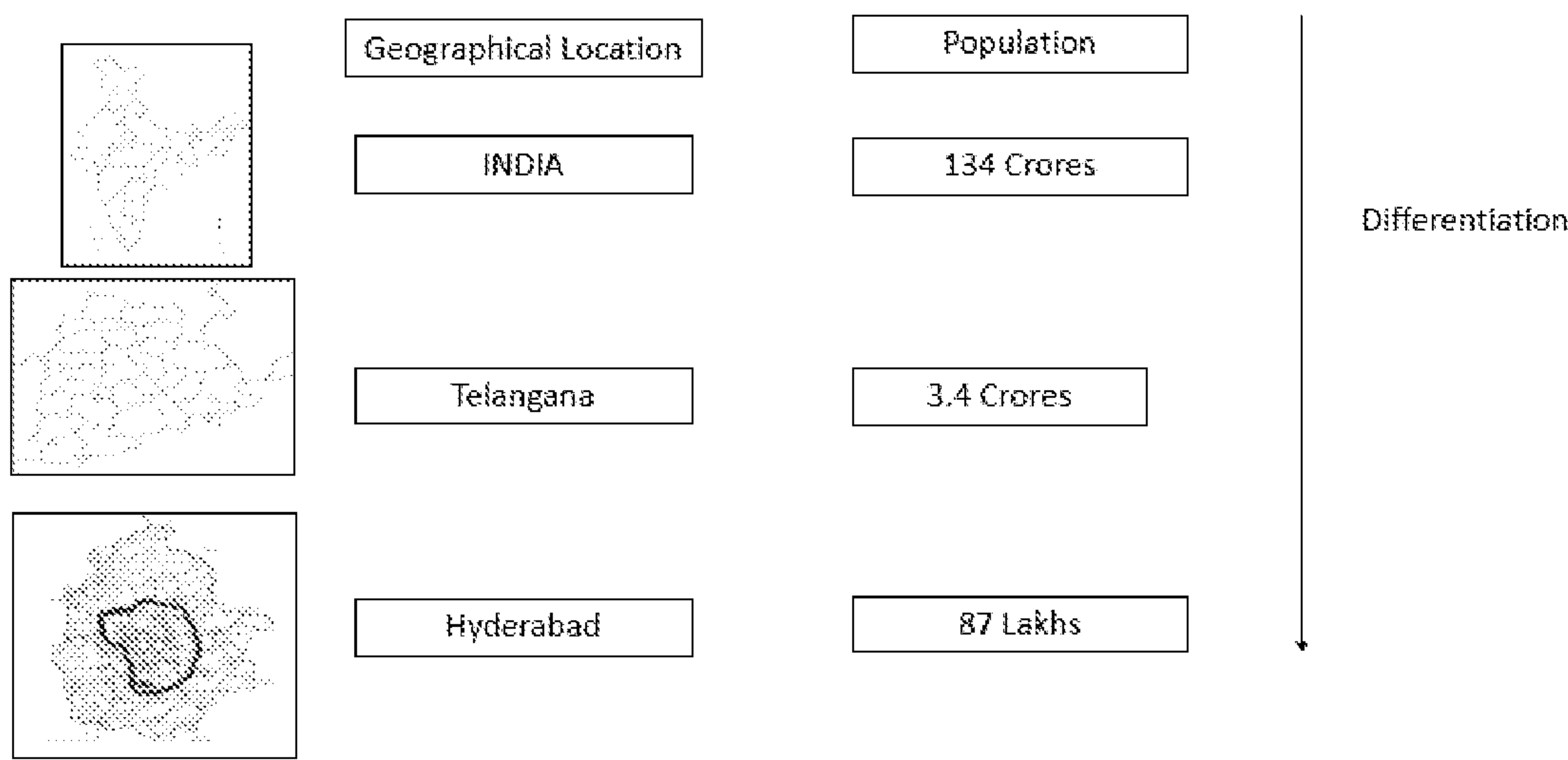
Drawing 32:



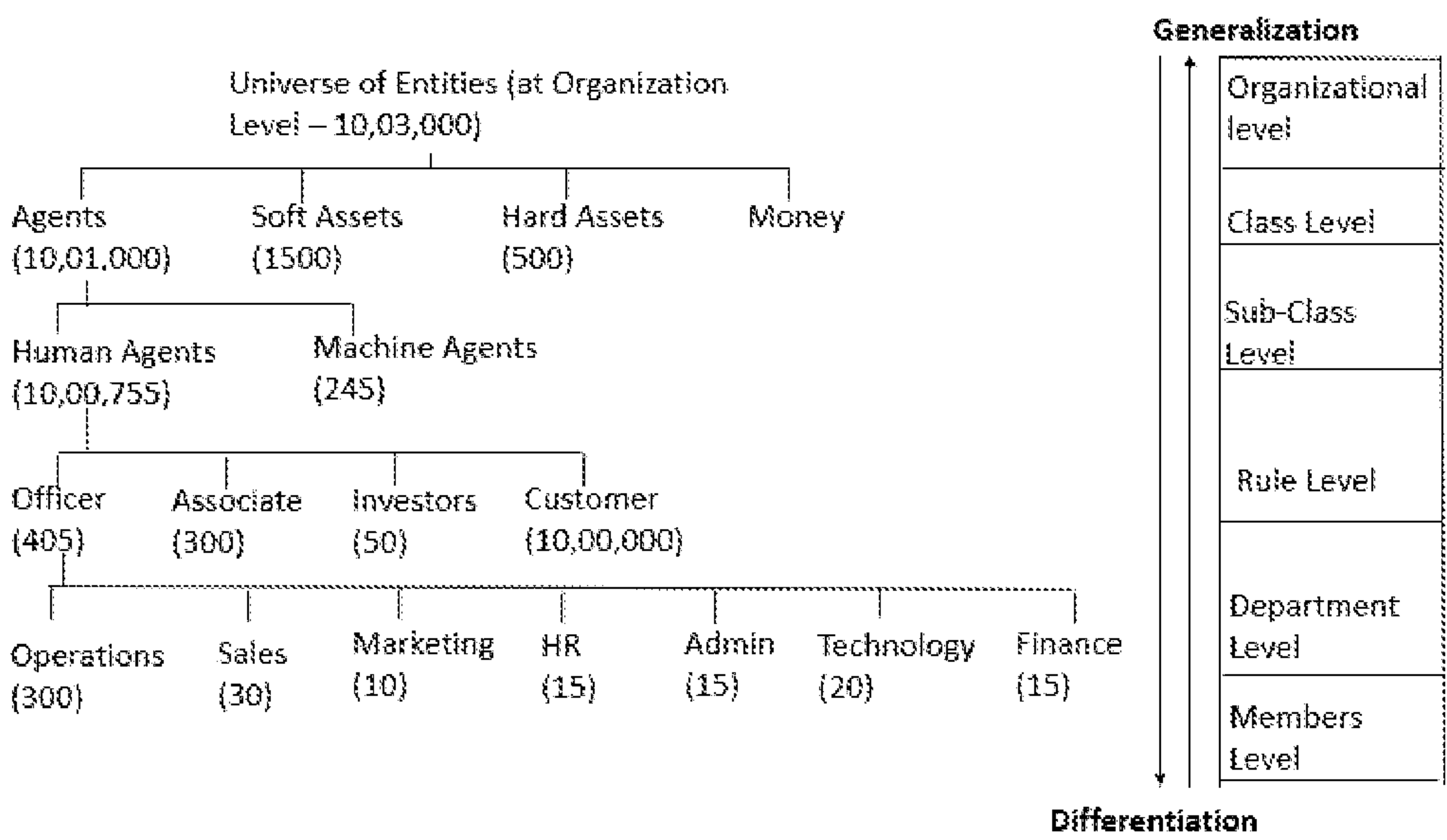
Drawing 33:



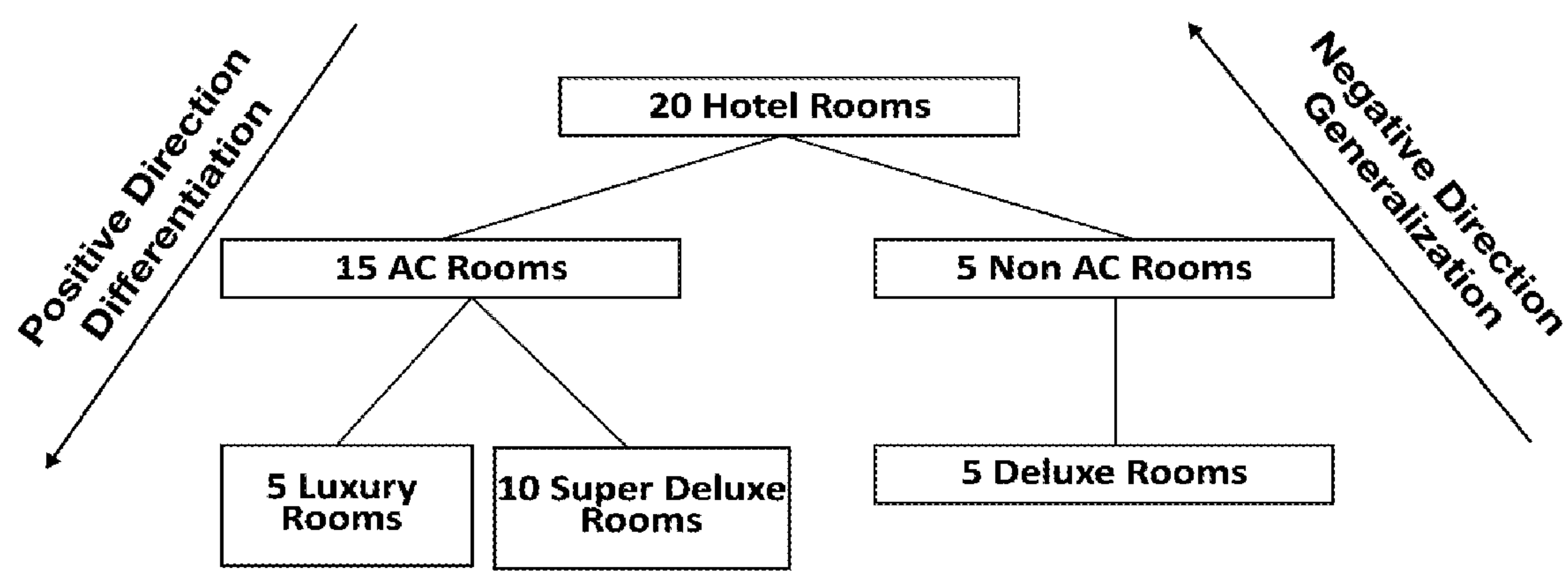
Drawing 34:



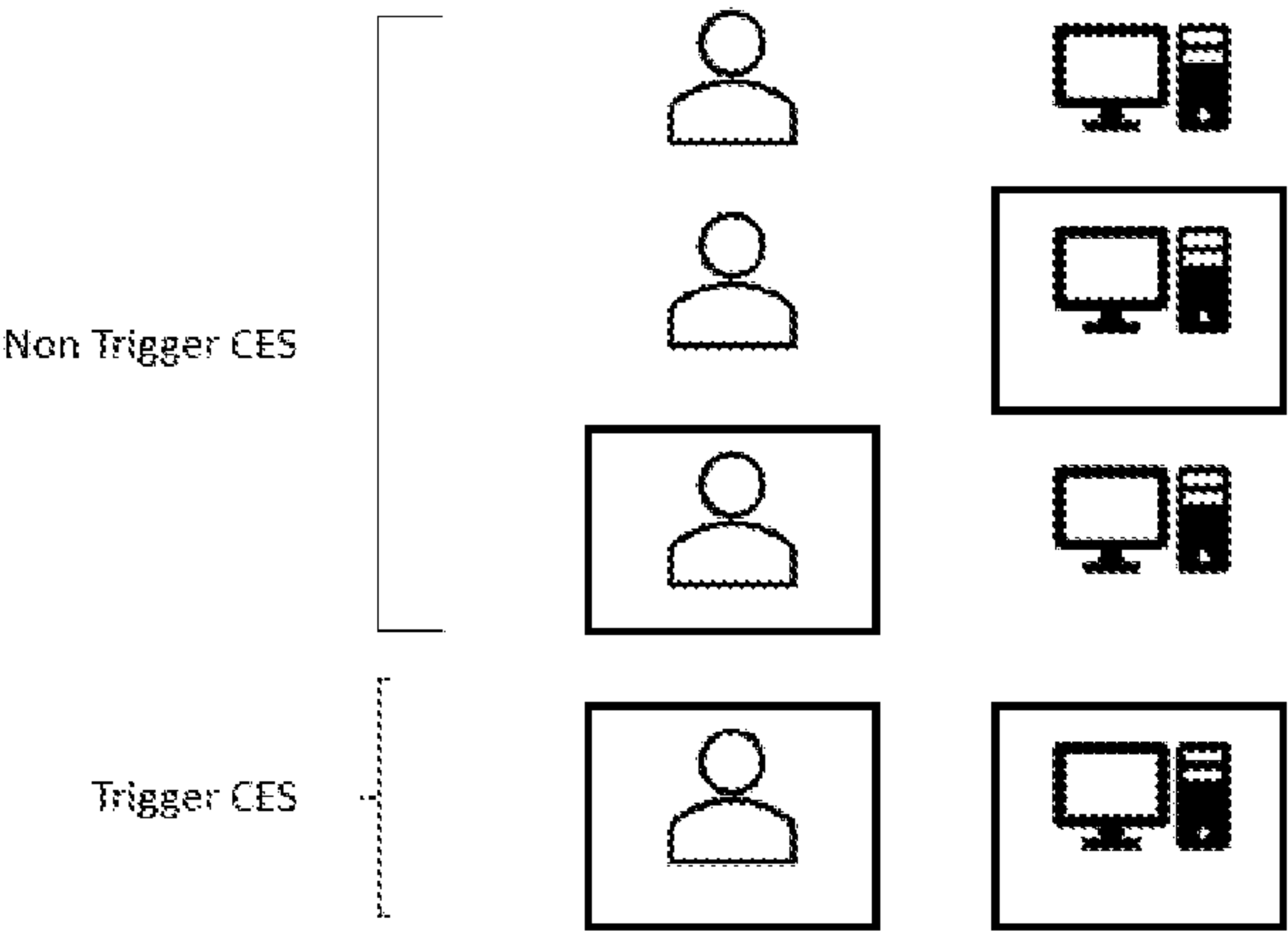
Drawing 35:



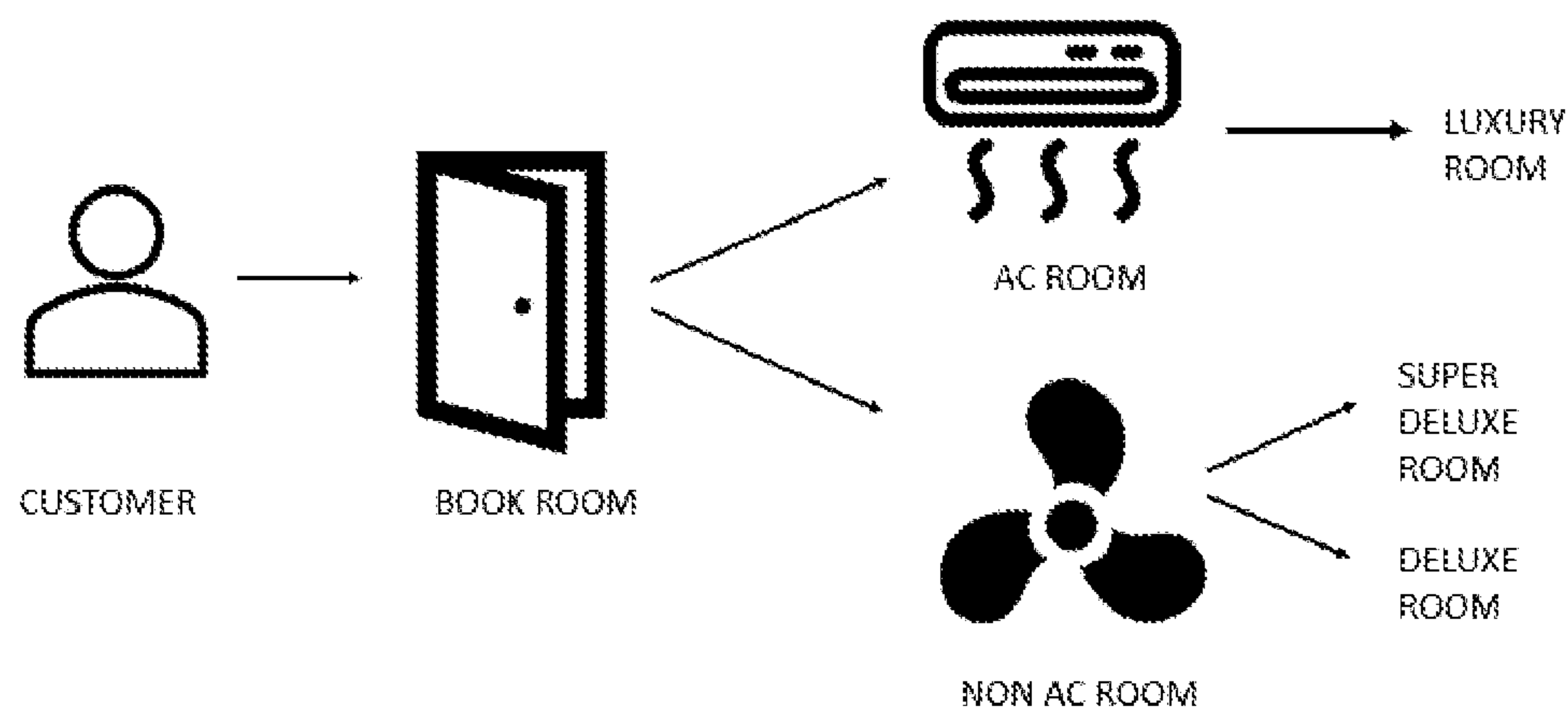
DRAWING 36:



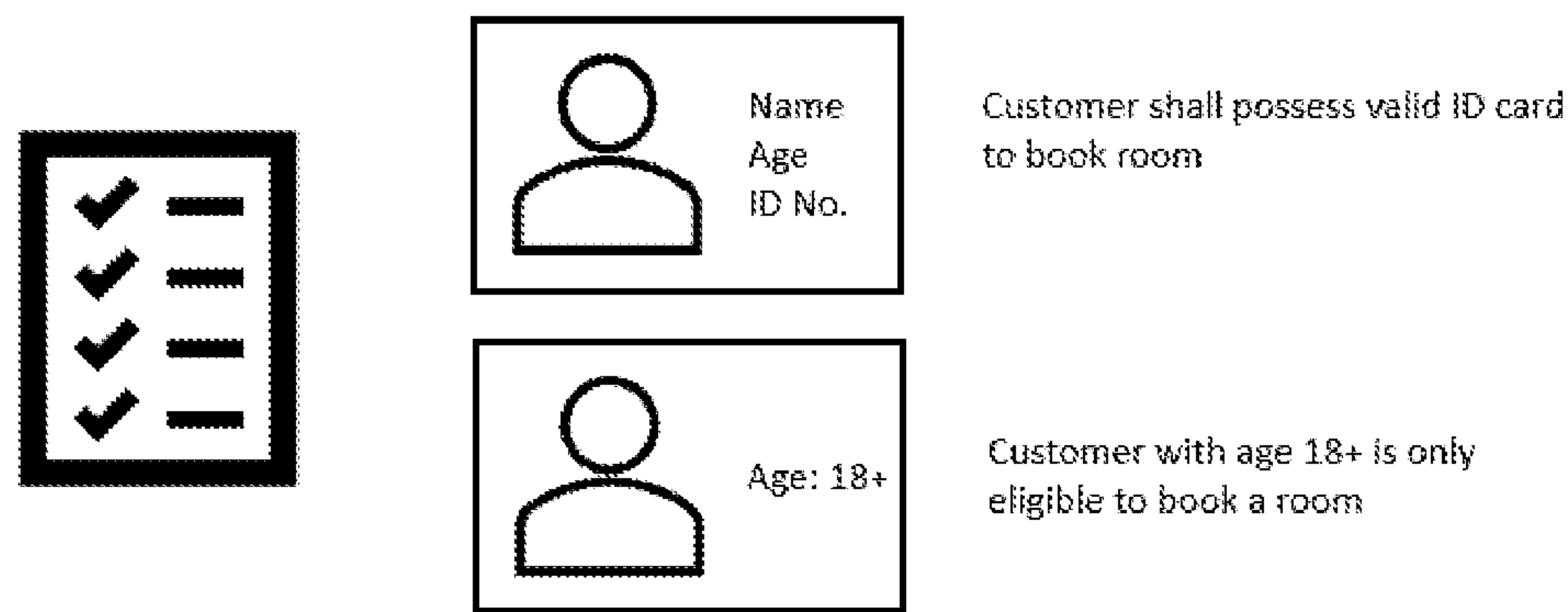
Drawing 37:



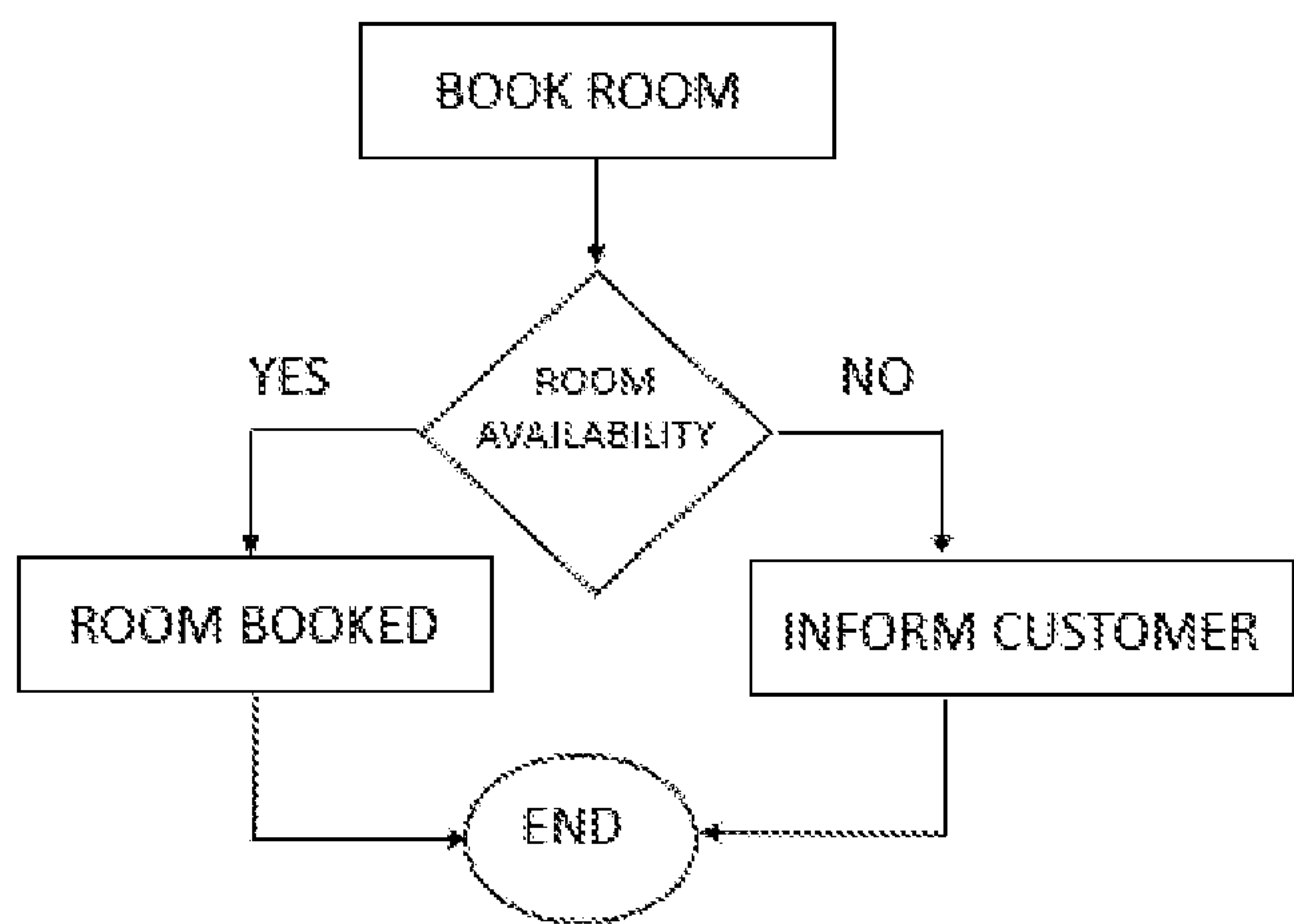
Drawing 38:



Drawing 39:



Drawing 40:



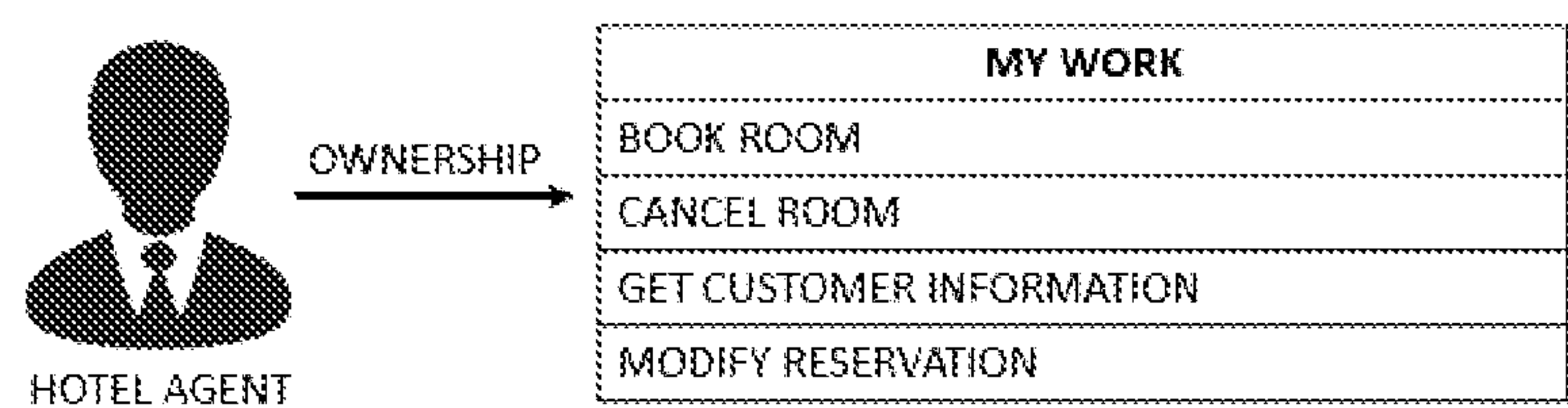
Drawing 41:



Drawing 42:



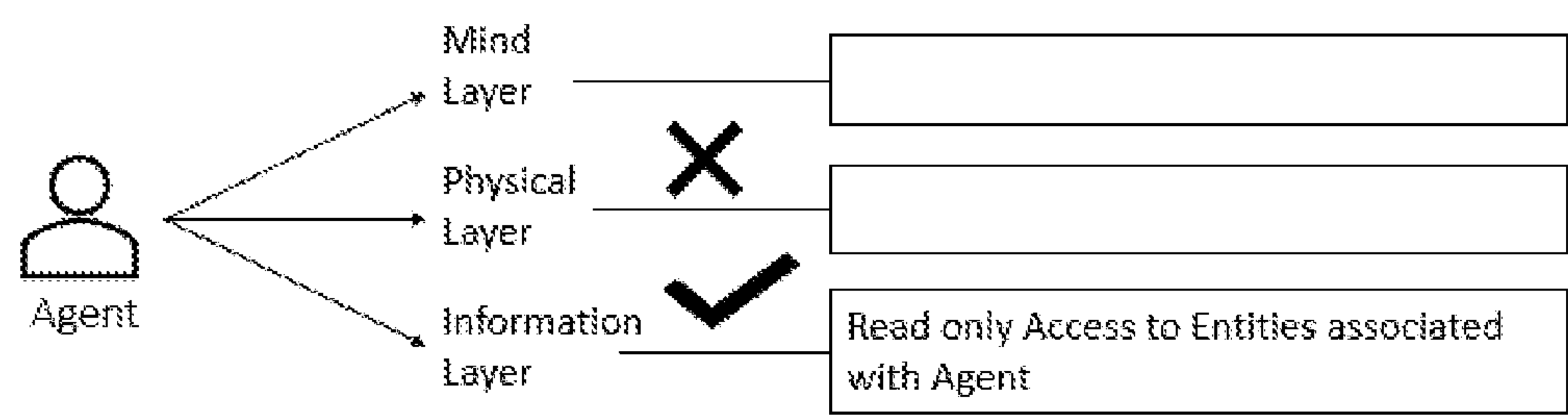
Drawing 43:



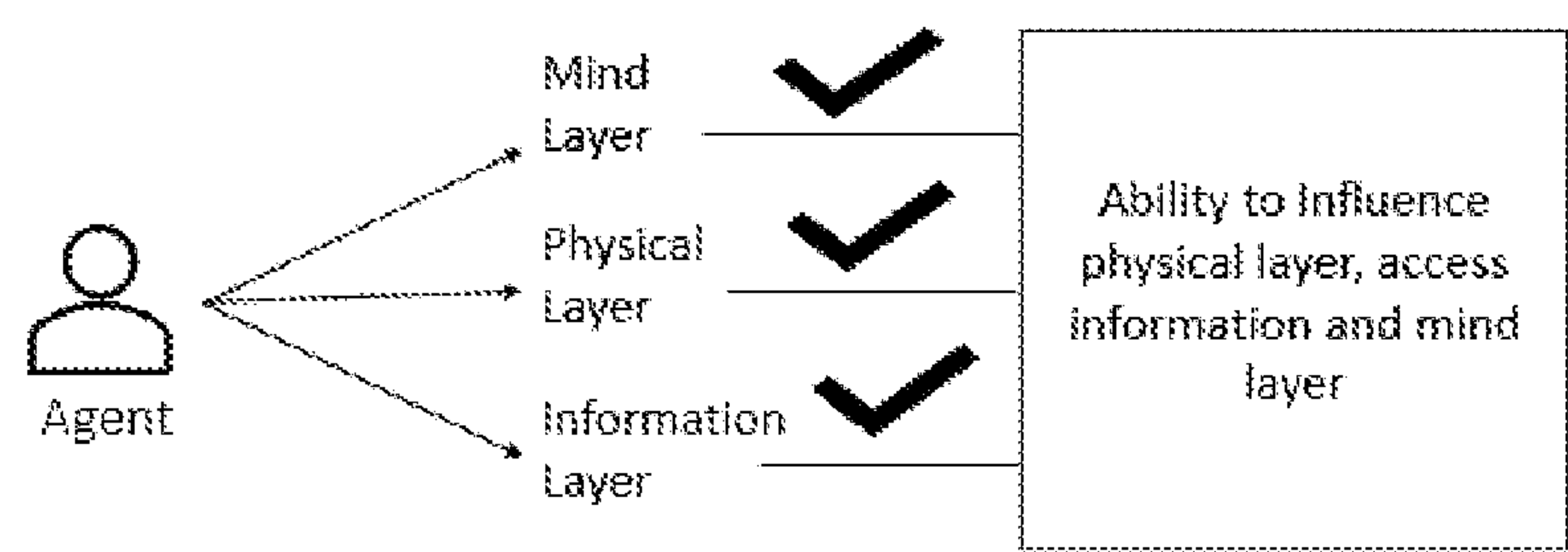
Drawing 44:

Physical Functions	 Book Room Check Room Availability, Get Customer Information, Get Room Stay Info, Get Price. Book Room.			
Mind Functions	Availability of Rooms on Weekends			
Information Functions	<table><tr><td><u>Rooms Available</u> Deluxe – 3 Super Deluxe – 8 Luxury – 10</td><td>Deluxe Super Deluxe Luxury</td><td>Wi-Fi Cost, LED TV; Size-200 Square Feet Free Wi-Fi, LCD TV; Size-250 Square Feet Free Wi-Fi, 60 inch LCD TV; Size-300 Square Feet</td></tr></table>	<u>Rooms Available</u> Deluxe – 3 Super Deluxe – 8 Luxury – 10	Deluxe Super Deluxe Luxury	Wi-Fi Cost, LED TV; Size-200 Square Feet Free Wi-Fi, LCD TV; Size-250 Square Feet Free Wi-Fi, 60 inch LCD TV; Size-300 Square Feet
<u>Rooms Available</u> Deluxe – 3 Super Deluxe – 8 Luxury – 10	Deluxe Super Deluxe Luxury	Wi-Fi Cost, LED TV; Size-200 Square Feet Free Wi-Fi, LCD TV; Size-250 Square Feet Free Wi-Fi, 60 inch LCD TV; Size-300 Square Feet		

Drawing 45:



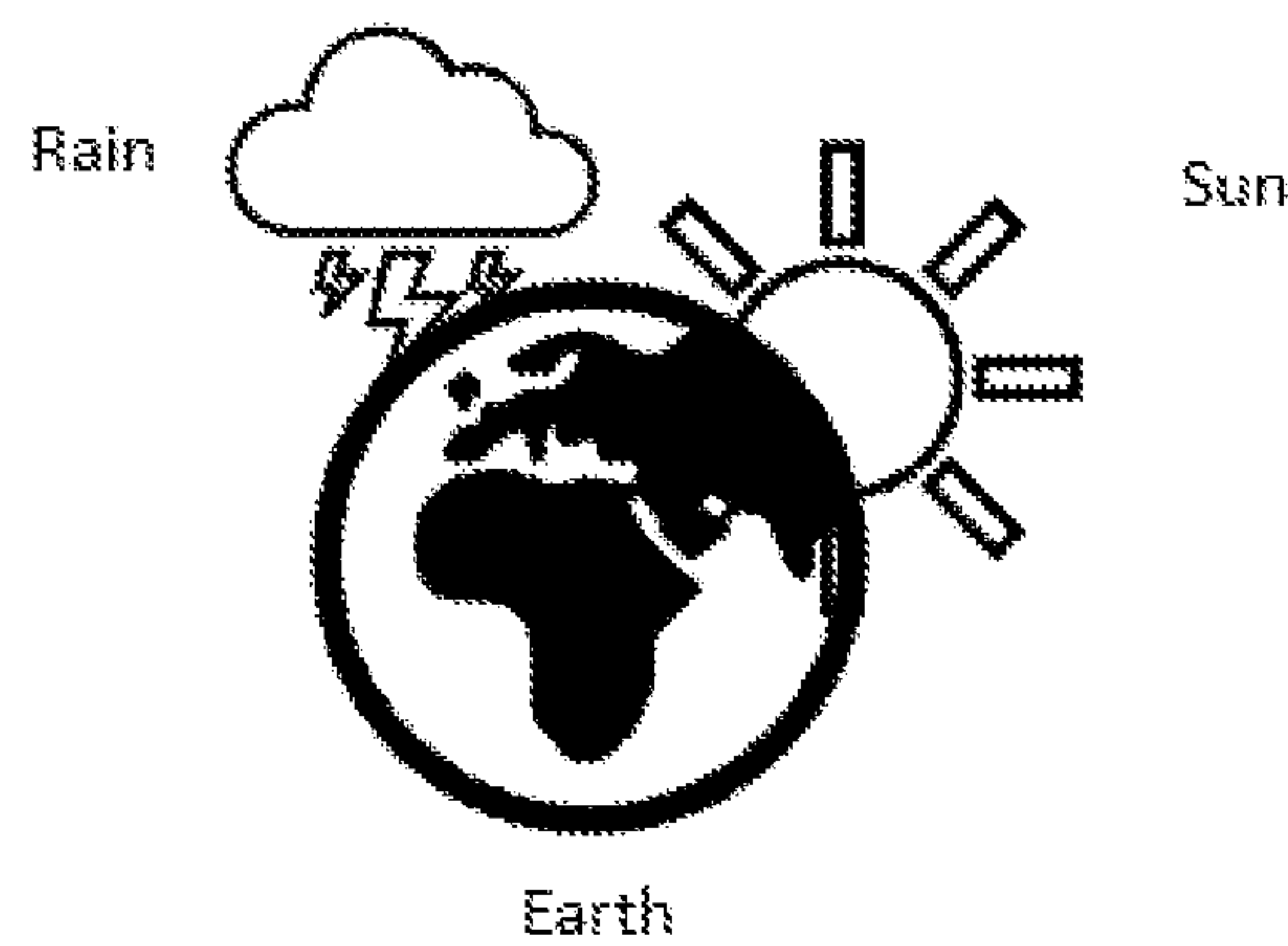
Drawing 46:



Drawing 47:



Drawing 48:



Drawing 49:

Book room. ← Global statement of intent

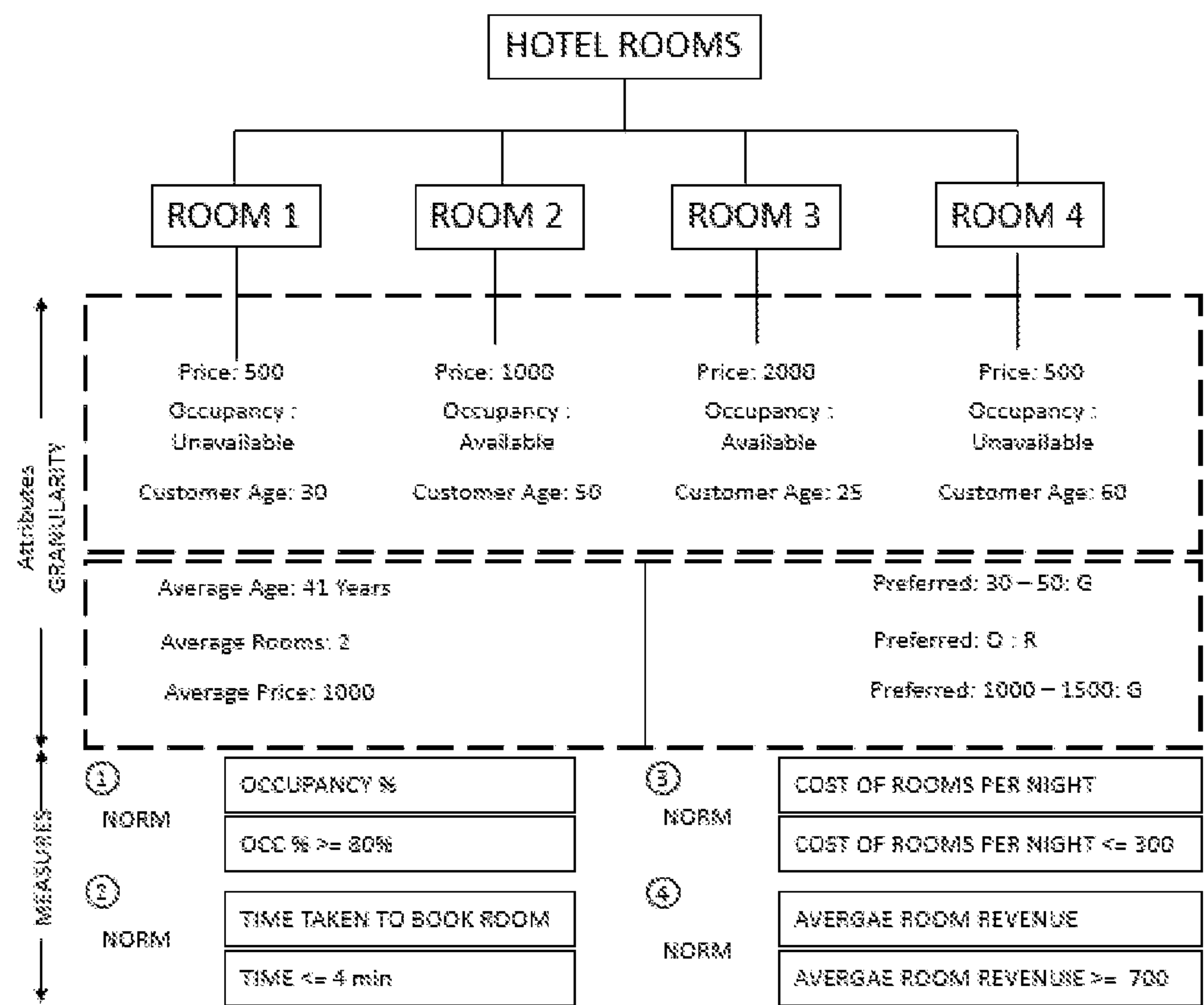
Paragraph-I:-

Check room availability. Get customer information. Get room stay information. Get place. Book hotel room.

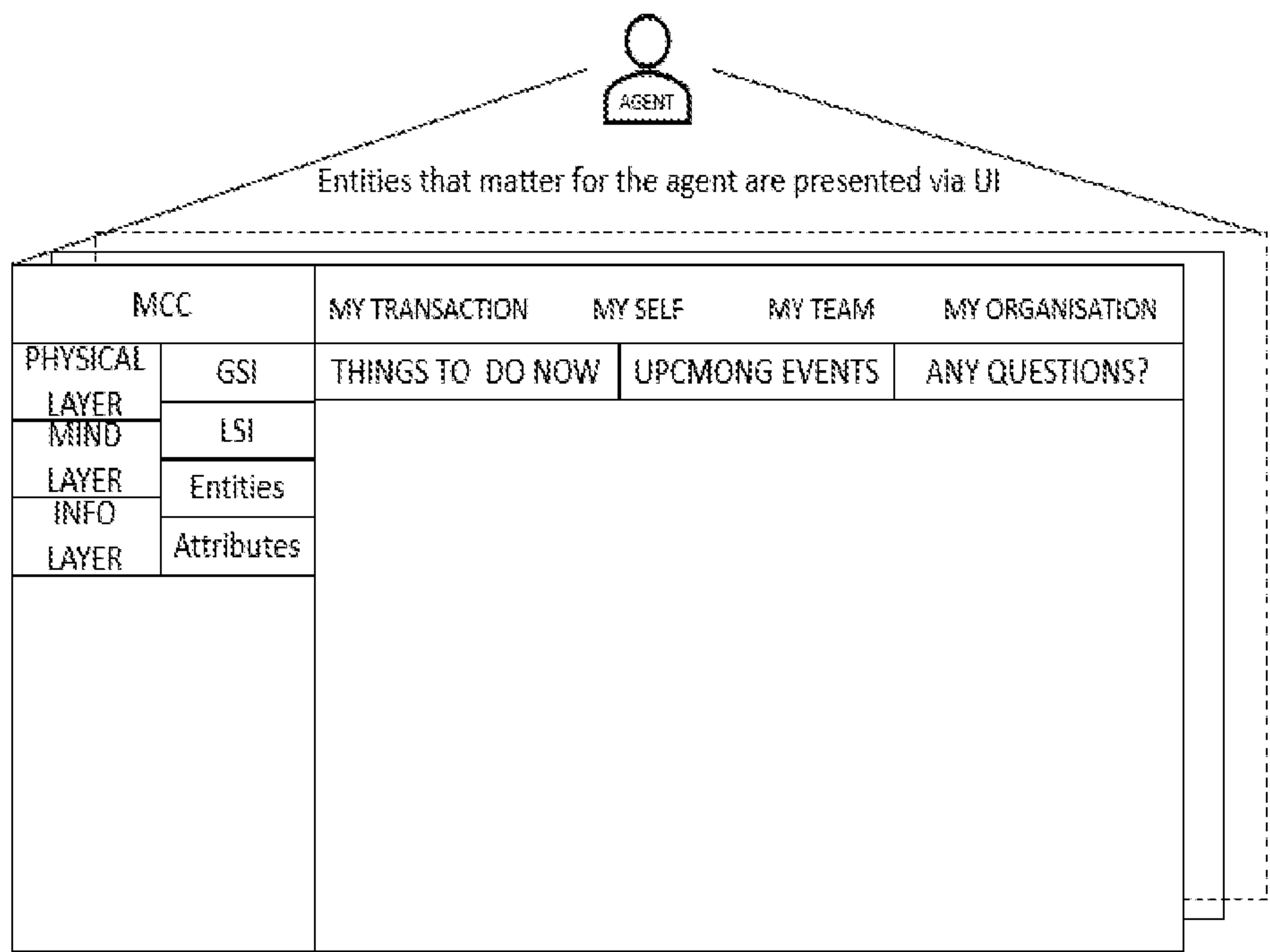
Paragraph-II:-

Check room availability. Get customer information. Get room stay information. Get transportation information. Get laundry information. Get place. Book hotel room.

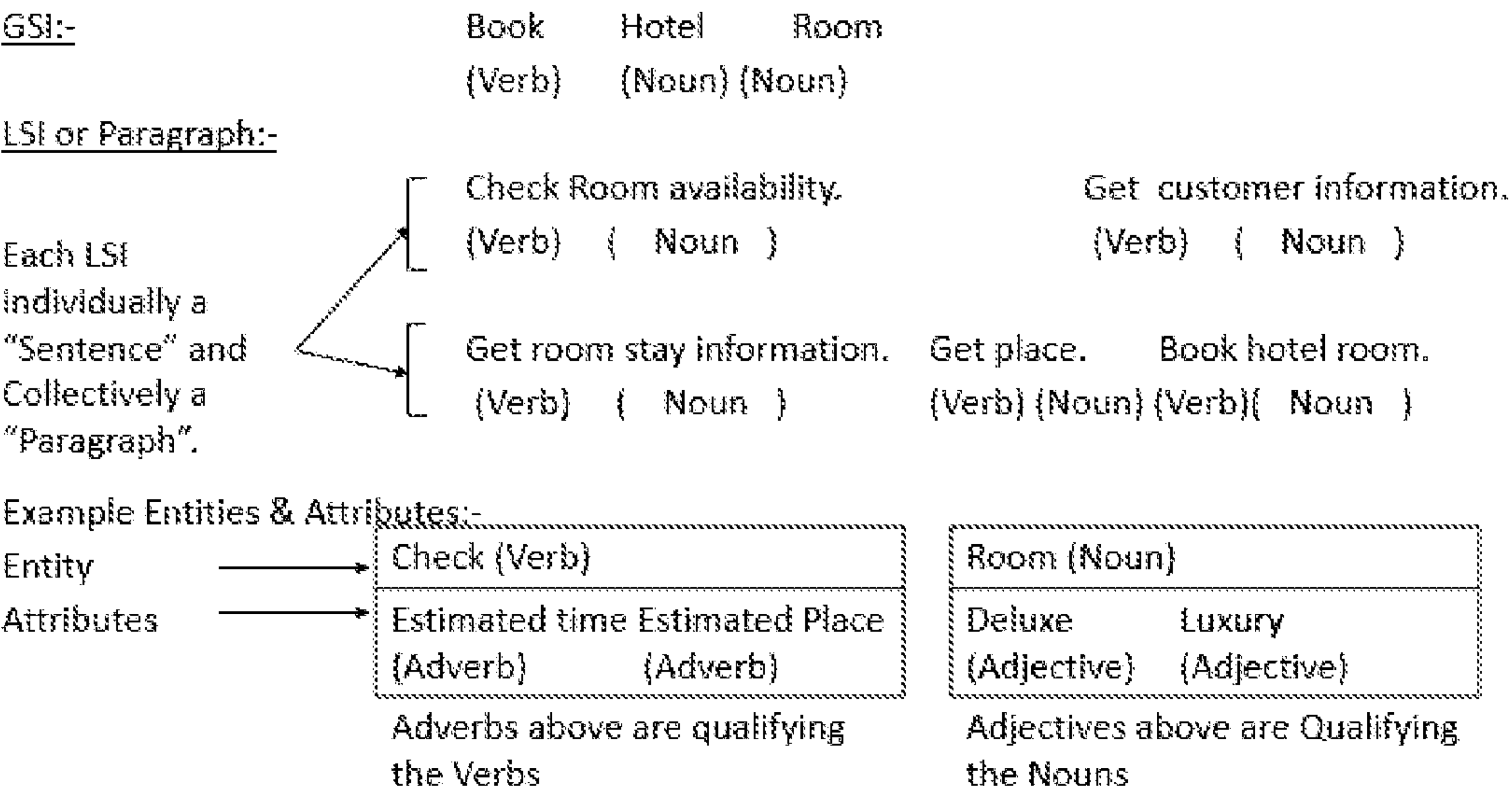
Drawing 50:



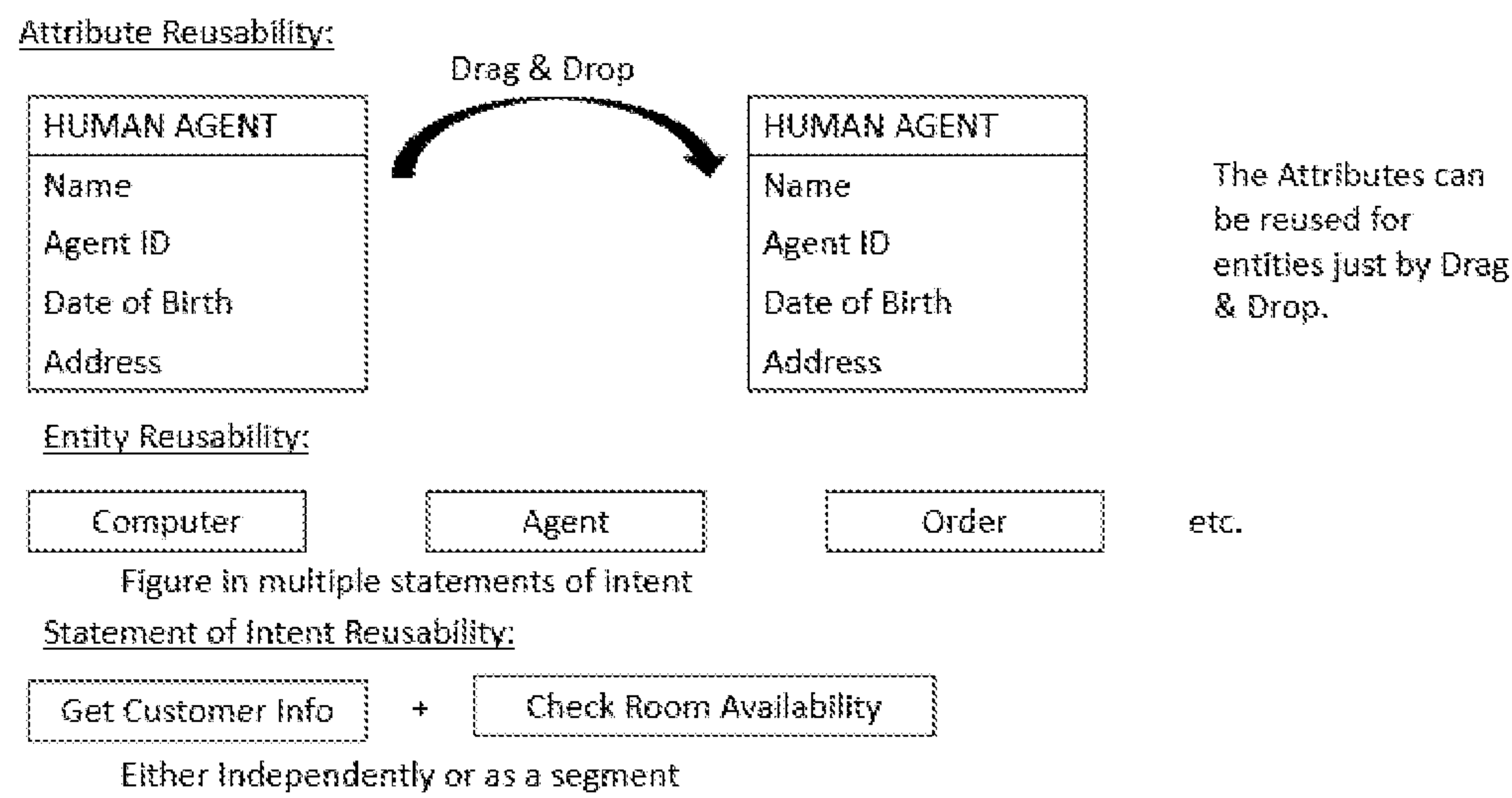
Drawing 51:



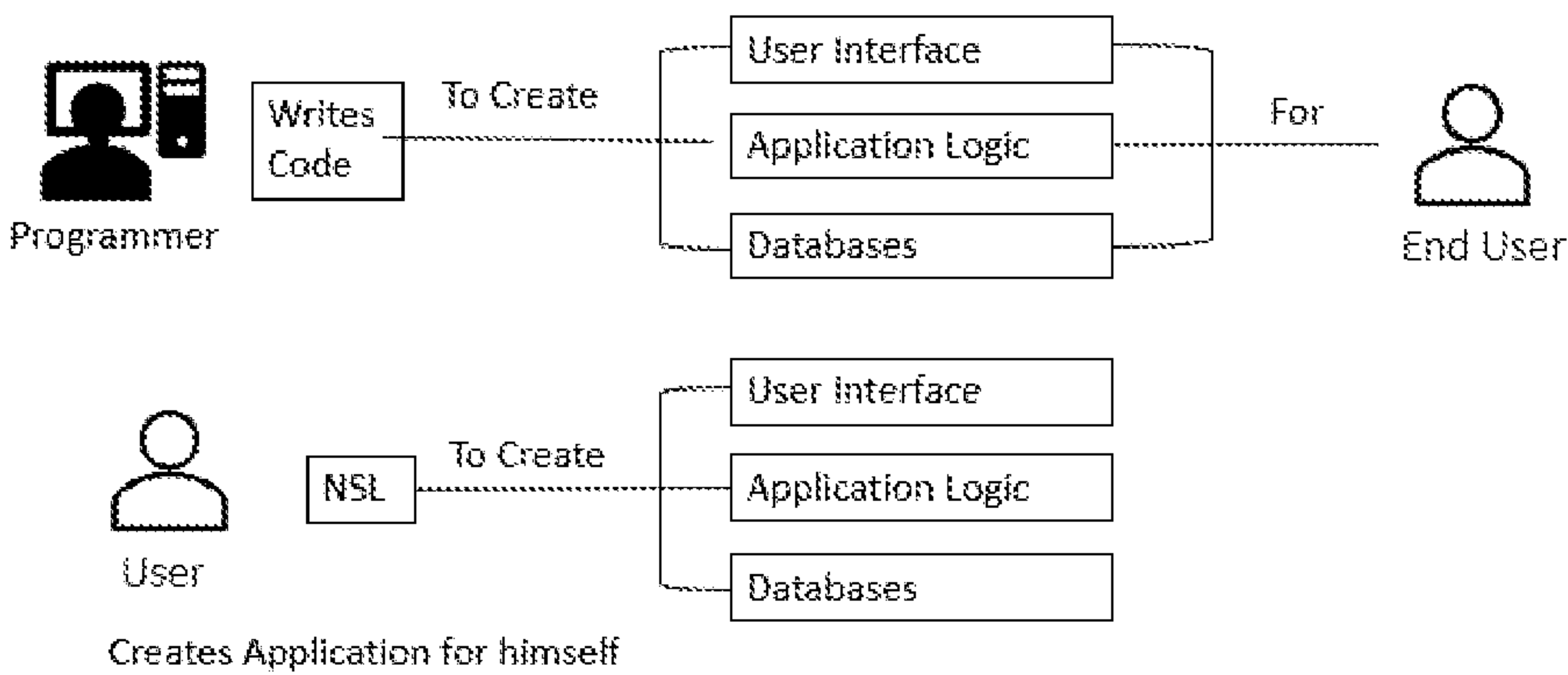
Drawing 52:



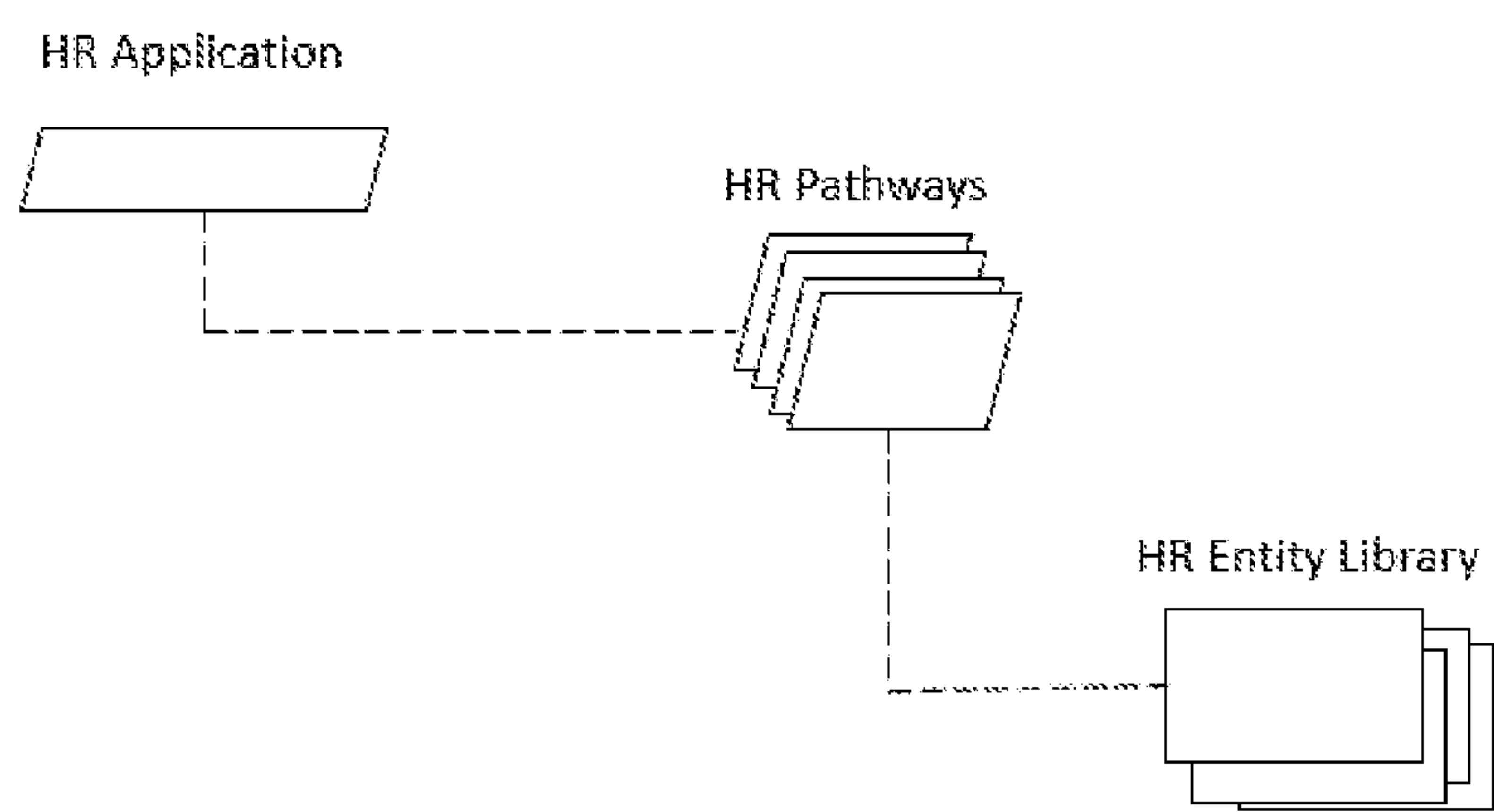
Drawing 53:



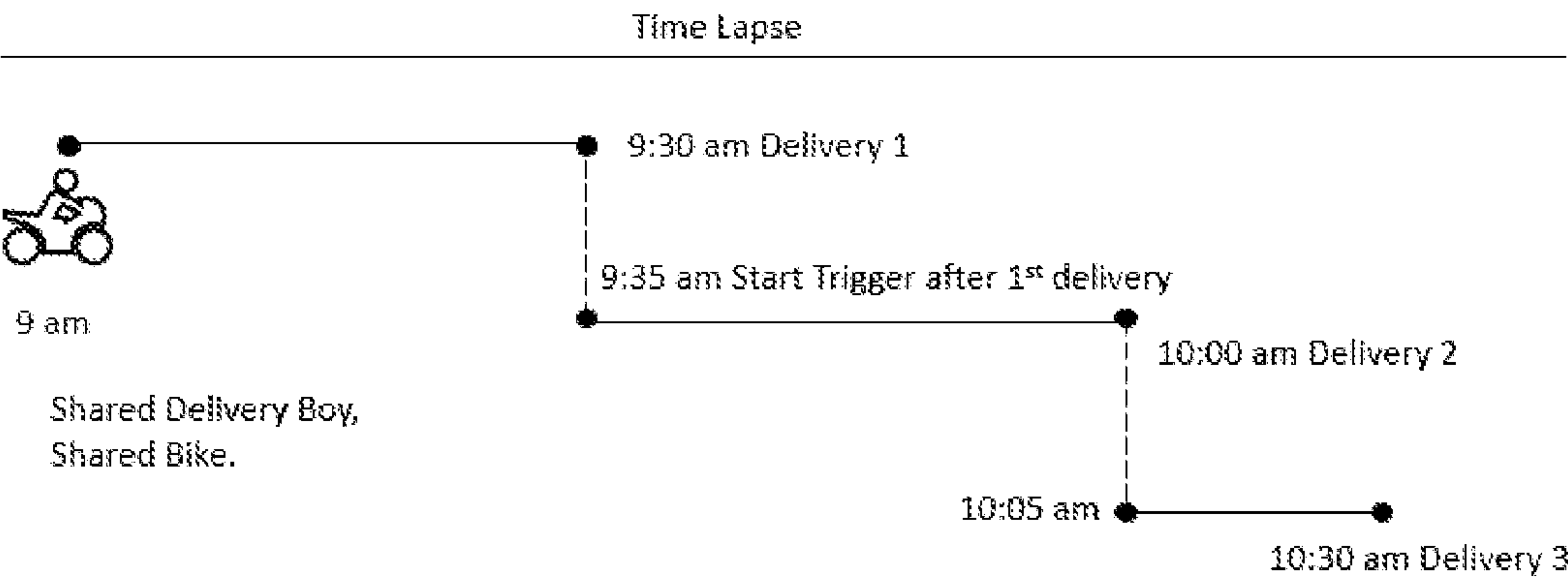
Drawing 54:



Drawing 55:



Drawing 56:



Drawing 57:

MCC		Stakeholder Engagement Center (SEC)			
PHYSICAL LAYER	GSI	MY TRANSACTION	MY SELF	MY TEAM	MY ORGANISATION
		THINGS TO DO NOW	UPCMONG EVENTS	ANY QUERIES	
MIND LAYER	LSI				
	Entities				
INFO LAYER	Attributes				

Drawing 58:

Java Code:

```
Package com.Hotel reservation
Public class fare{
    public int fare calculator
    (int rate, int days, int s)
    {
        if {s==1}
        {
            return days * rate;
        }
        if {s==2}
        {
            return days * rate *2;
        }
    }
}
```

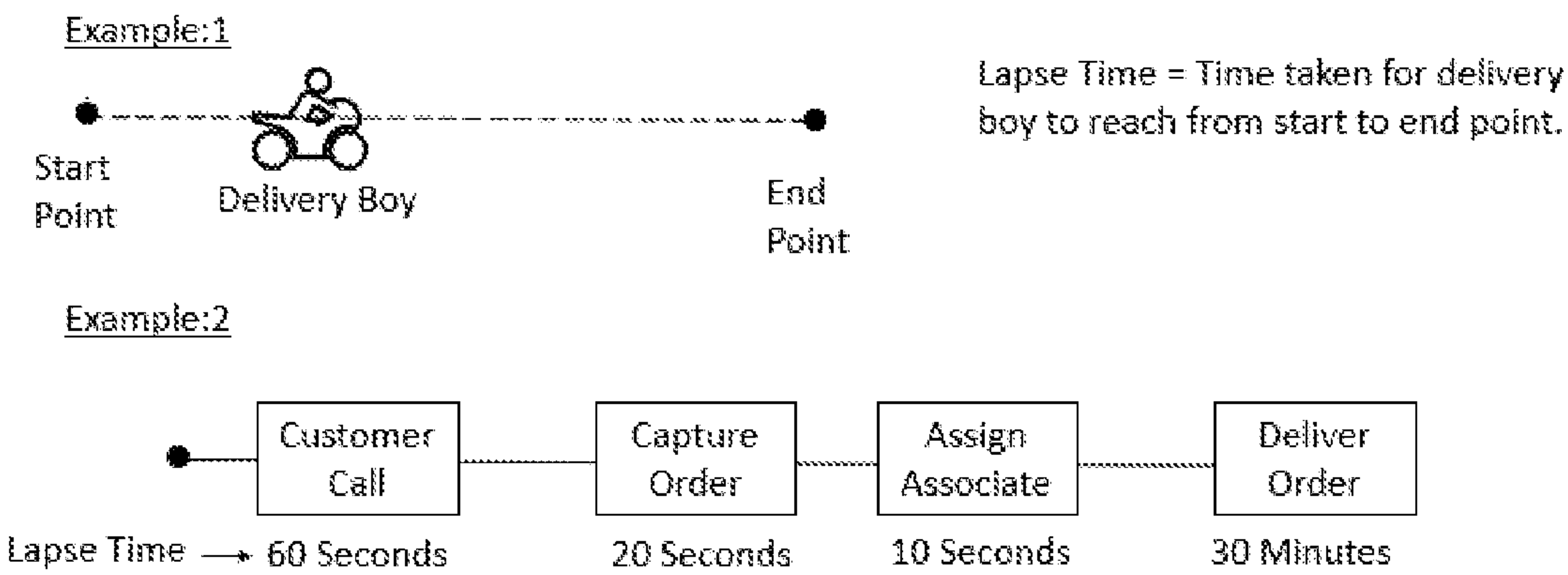
Natural language:

Fare calculator:
If occupancy = 1, fare is equal
To number of days* rate
If occupancy = 2, fare is equal
To number of days * rate *2

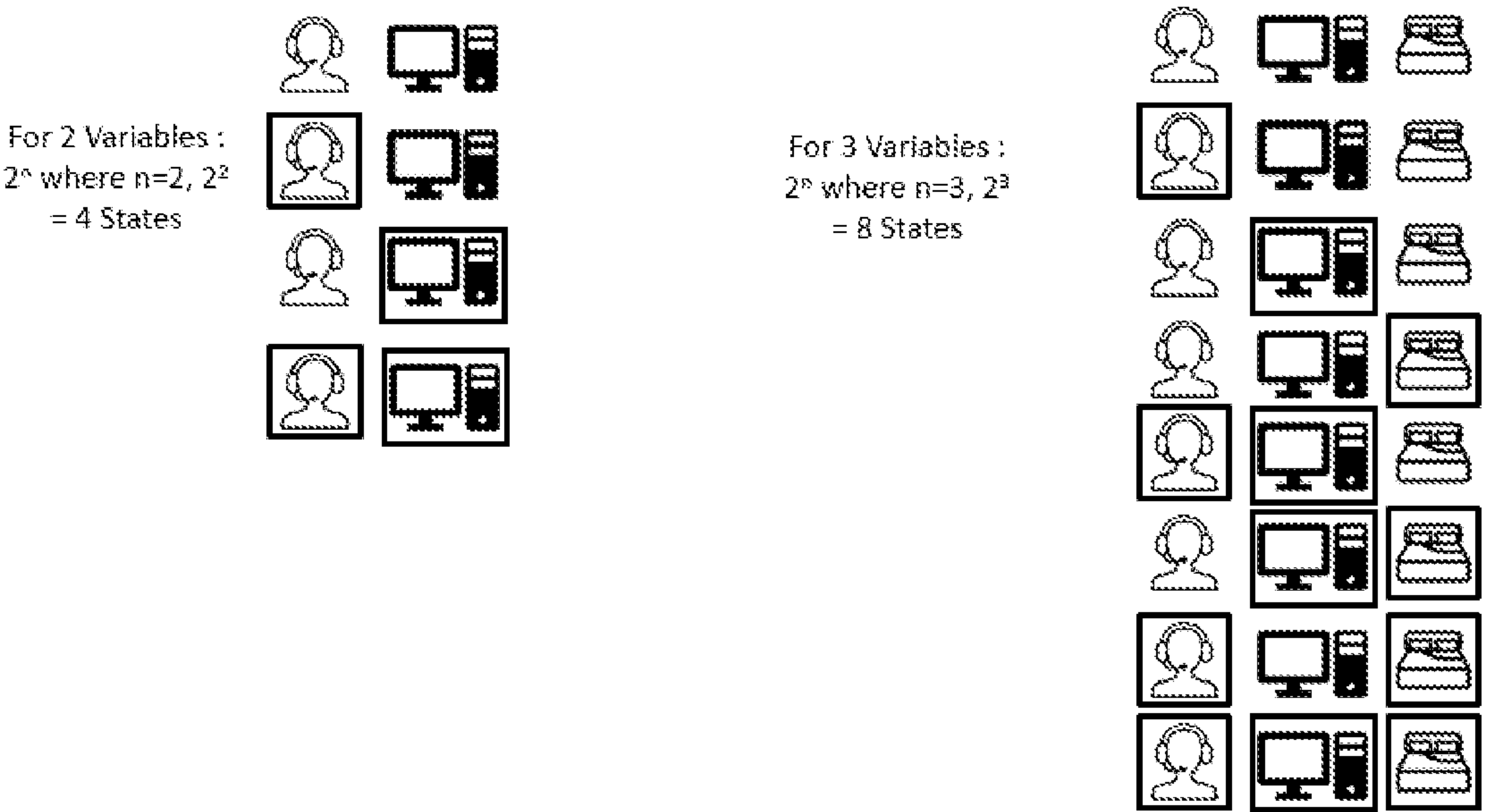
Drawing 59:

Information – Sachin had prepared dinner – Conveys meaning
– Sachin may have prepared dinner – Conveys meaning
Prescriptive information – Sachin prepares dinner
Value – Sachin prepares dinner

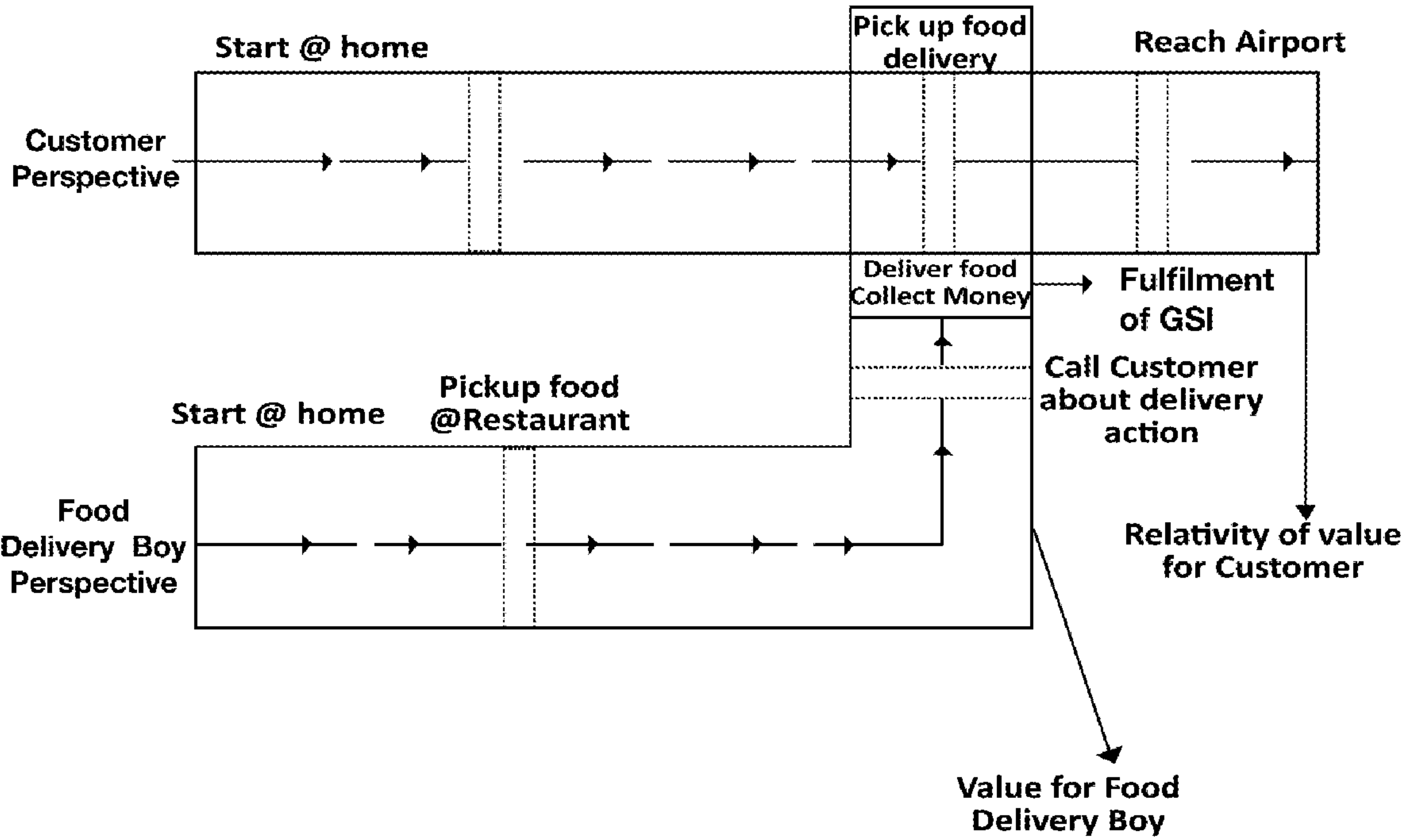
Drawing 60:



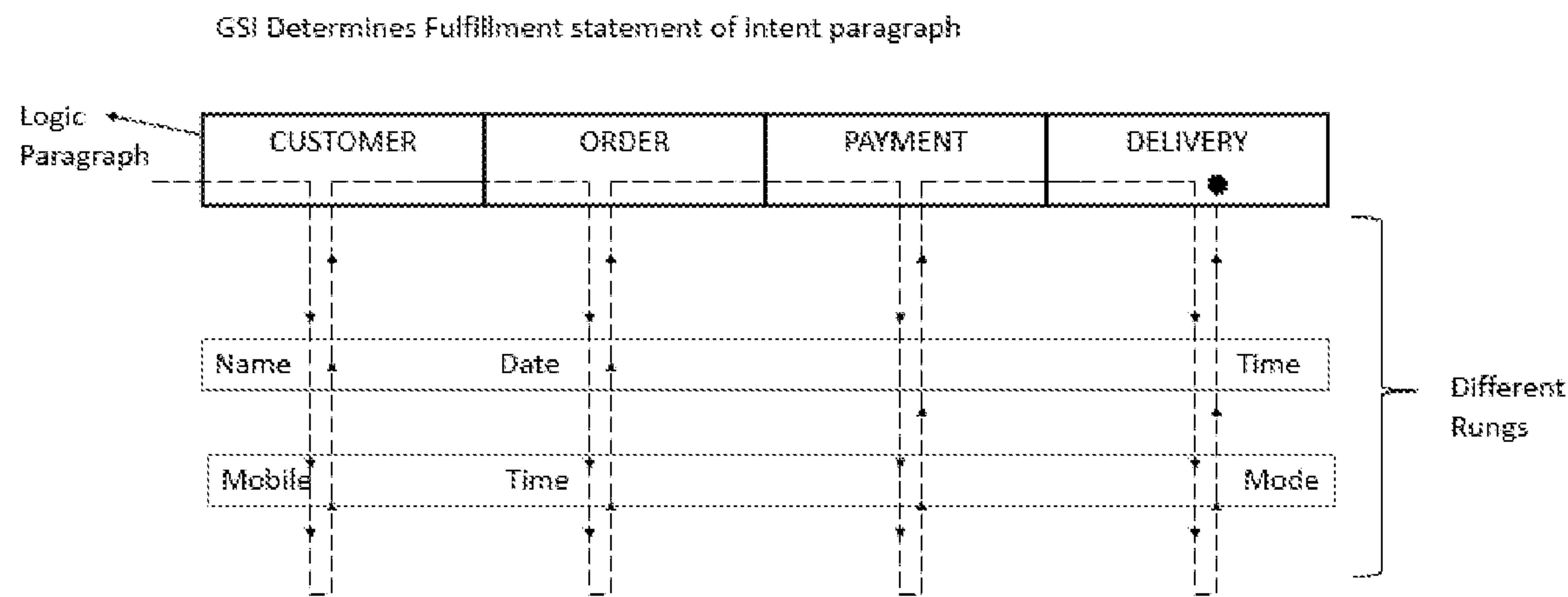
Drawing 61:



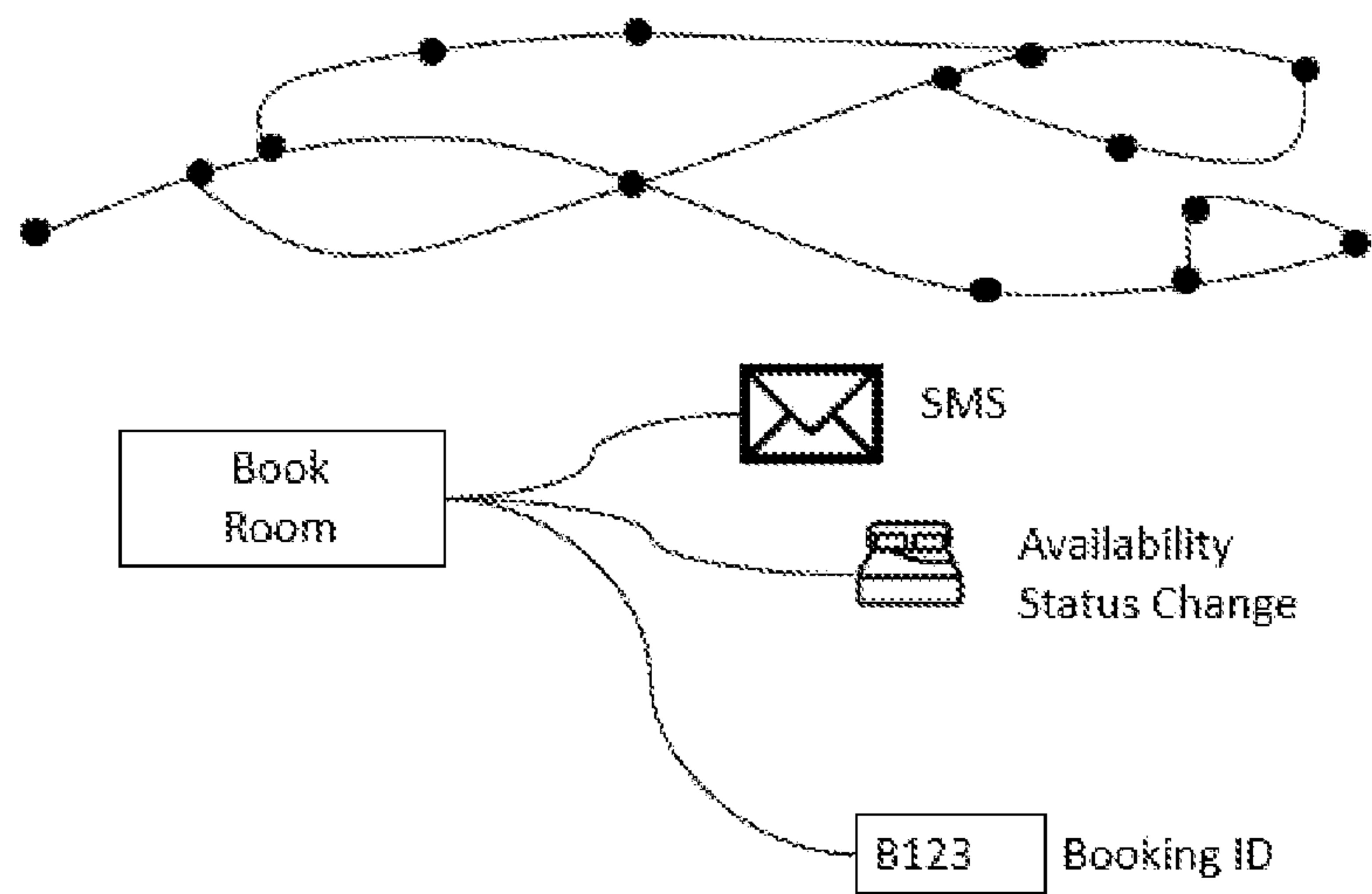
DRAWING 62:



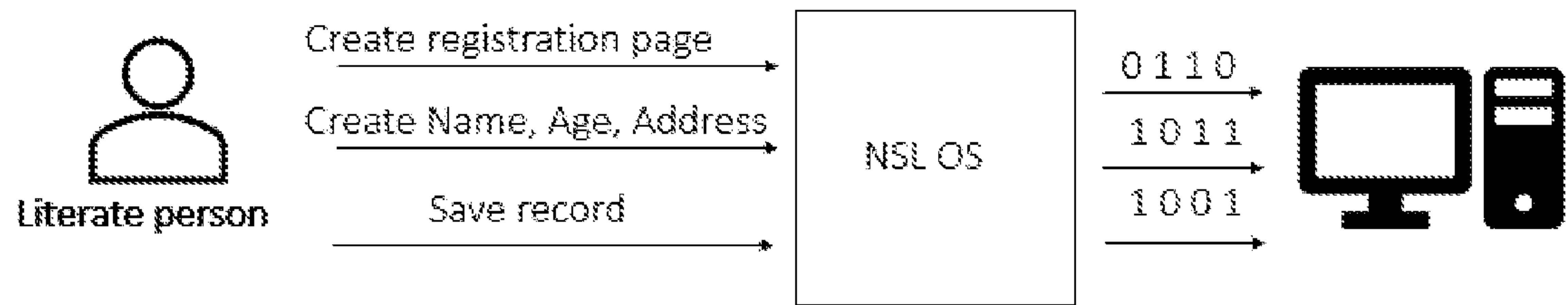
Drawing 63:



Drawing 64:



Drawing 65:



25/79

Drawing 68:

Book Hotel Room.

Get Customer Information. Select Room Type Luxury. Select Room Type Deluxe. Select Room Type Super Deluxe.

Get Customer Information. Select Service Transport Type 1. Select Service Transport Type 2.

Select Service Transport Type 3. Select Service Laundry Type 1. Select Service Laundry Type 2.

Select Service Laundry Type 1. Get Transportation Information. Get Laundry Information. Calculate Fare Room. Calculate Fare Transport. Calculate Fare Laundry. Cancel Room. Book Hotel Room.

Entity	Customer	Entity	Super deluxe room		Entity	Room stay information	
Primary Attributes	Name: Raju	Primary Attributes	Name: Raju	Room availability: Yes	Primary Attributes	Number of days: 2	Occupancy: 2
Entity	Room fare	Entity	Booking Confirmation				
Primary Attributes	Number of days * Rate * Occupancy: 10,000	Primary Attributes	ID No: Dsdx2	Booking name: Raju	Fare: 10,000	Number of days * Rate * Occupancy: 10,000	

होटल का कमरा बुक करें।

ग्राहक जानकारी प्राप्त करें. कमर प्रकार लक्जरी चुनें. कमर प्रकार डीलक्स का चयन करें. कमर प्रकार सुपर डीलक्स का चयन करें.

रूम स्टे की जानकारी लें. सेवा परिवहन प्रकार 1 चुनें. सेवा परिवहन प्रकार 2 चुनें.

सेवा परिवहन प्रकार 3 चुनें. सेवा लांड्री प्रकार 1 चुनें. सेवा धुलाई प्रकार 2 का चयन करें.

सेवा लांड्री प्रकार 3 का चयन करें. परिवहन जानकारी प्राप्त करें. कपड़े धोने की जानकारी प्राप्त करें. किराया कमरे की गणना करें.

किराया परिवहन की गणना. किराया कपड़े धोने की गणना करें. कमर रद्द करें. होटल का कमरा बुक करें.

नाम	ग्राहक	कमरा	सुपर डीलक्स कमरा		नाम	कमरे में रहने की जानकारी	
प्राथमिक विशेषताएँ	नाम: राजू	प्राथमिक विशेषताएँ	दर: 1,500	कमरे की उपलब्धता: हाँ	प्राथमिक विशेषताएँ	दिनों की संख्या: 3	व्यापक: 2

कमरा	कमरे का किराया	कमरा	बुकिंग की प्रुति			
प्राथमिक विशेषताएँ	दिनों की संख्या: 2 * व्यापक: 10,000	प्राथमिक विशेषताएँ	ID नंबर: 06322	बुकिंग नाम: राजू	किराया: 10,000	परिवहन की लागत: 0

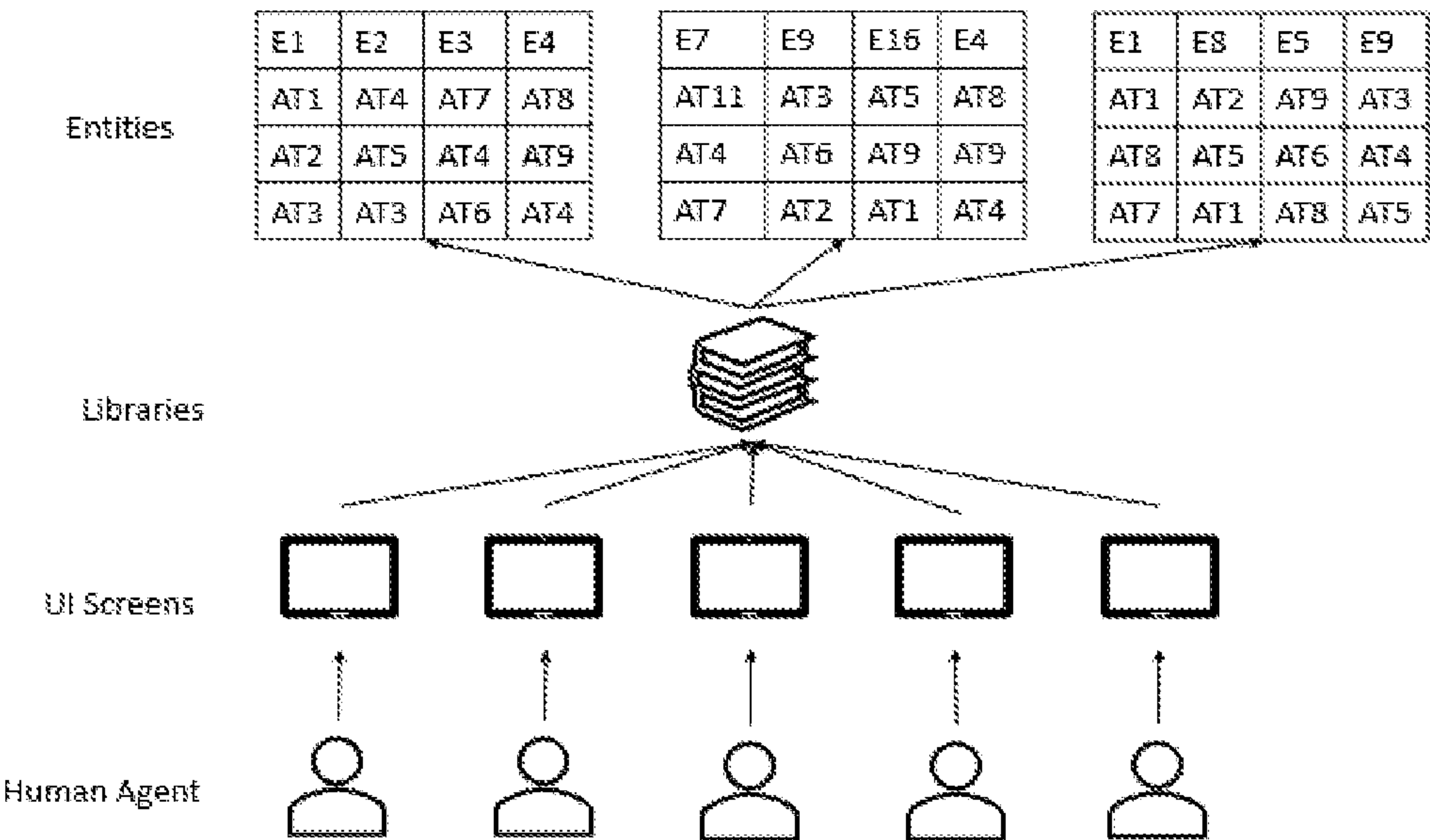
బుక్ హోటల్ రూం

కస్టమర్ సమాచారం పొందండి. గది రకం లగ్జరీ ఎంచుకోండి. గది రకం డిలక్స్ ఎంచుకోండి. గది రకం గది రకం సూపర్ డిలక్స్ ఎంచుకోండి.

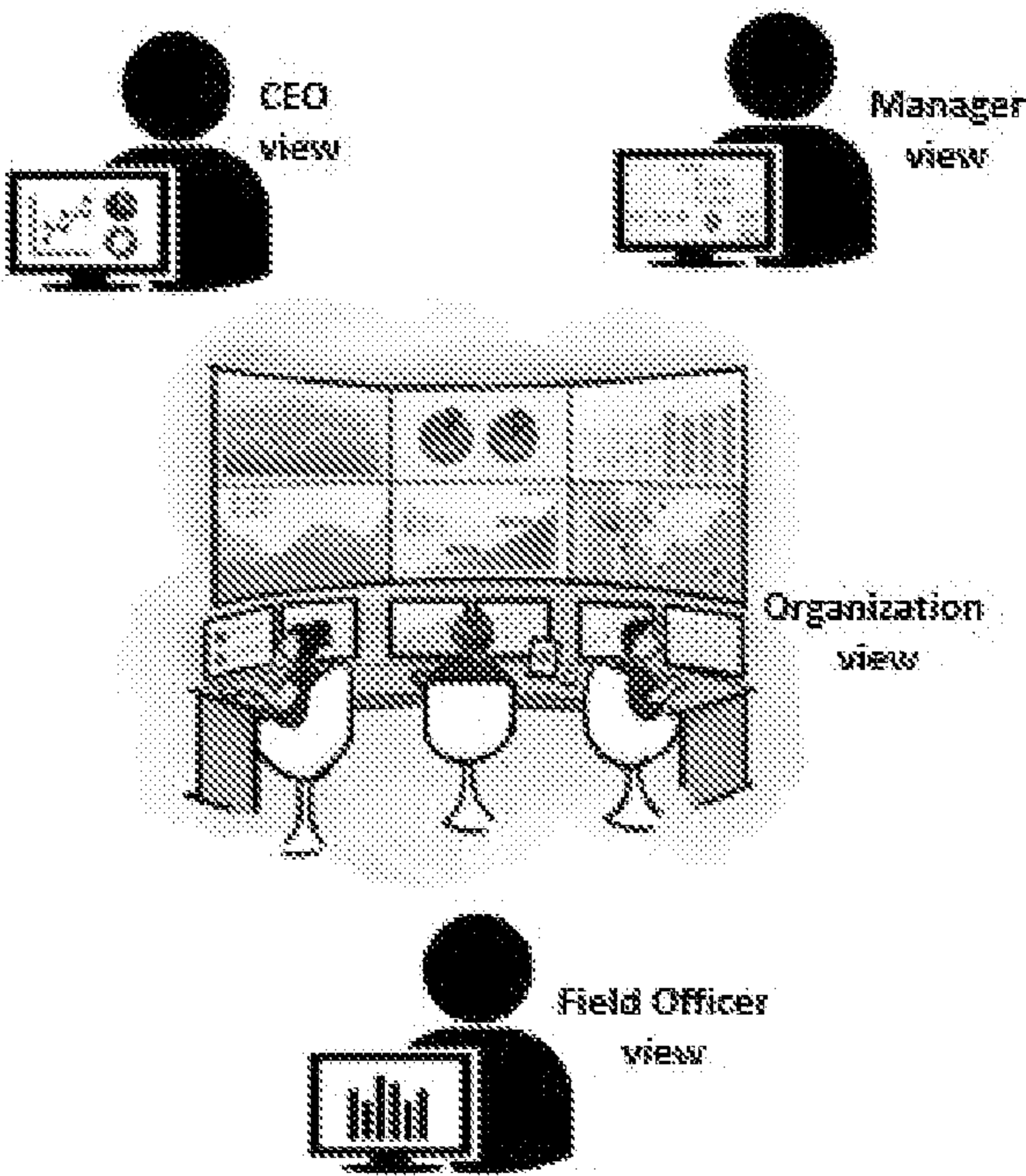
రూమ్ బుక్ సమాచారం సేకరించండి. సేవా రవాణా పద్ధతి 1 ఎంచుకోండి. సేవా రవాణా పద్ధతి 2 ఎంచుకోండి.
 సేవా రవాణా పద్ధతి 3 ఎంచుకోండి. సర్వీస్ లాండ్రీ టైప్ 1 ఎంచుకోండి. సర్వీస్ లాండ్రీ టైప్ 2 ఎంచుకోండి.
 సర్వీస్ లాండ్రీ టైప్ 3 ఎంచుకోండి. రవాణా సమాచారం పొందండి. లాండ్రీ సమాచారం పొందండి రూమ్ థ్రిప్పి తెలియపరచండి.
 రవాణా థ్రిప్పి తెలియపరచండి. లాండ్రీ థ్రిప్పి తెలియపరచండి. రూమ్ థ్రిప్పి తెలియపరచండి. బుక్ హోటల్ రూం.

ఎంటిటీ	కస్టమర్	ఎంటిటీ	సూపర్ డిలక్స్ గది		ఎంటిటీ	రూమ్ బుక్ సమాచారం	
ప్రాథమిక లక్షణాలు	పేరు: రాజు	ప్రాథమిక లక్షణాలు	ధర: 1500	రూమ్ లభ్యత: అవును	ప్రాథమిక లక్షణాలు	రోజుల సంఖ్య: 2	అక్యుపెన్సీ: 2
ఎంటిటీ	రూమ్ బుక్ సమాచారం	ఎంటిటీ	బుకింగ్ నిర్ధారణ				
ప్రాథమిక లక్షణాలు	రోజుల సంఖ్య * ధర * అక్యుపెన్సీ: 10,000	ప్రాథమిక లక్షణాలు	ID సంఖ్య: Dsdx2	బుకింగ్ పేరు: రాజు	ధర: 10,000	రవాణా ఖర్చు: 0	

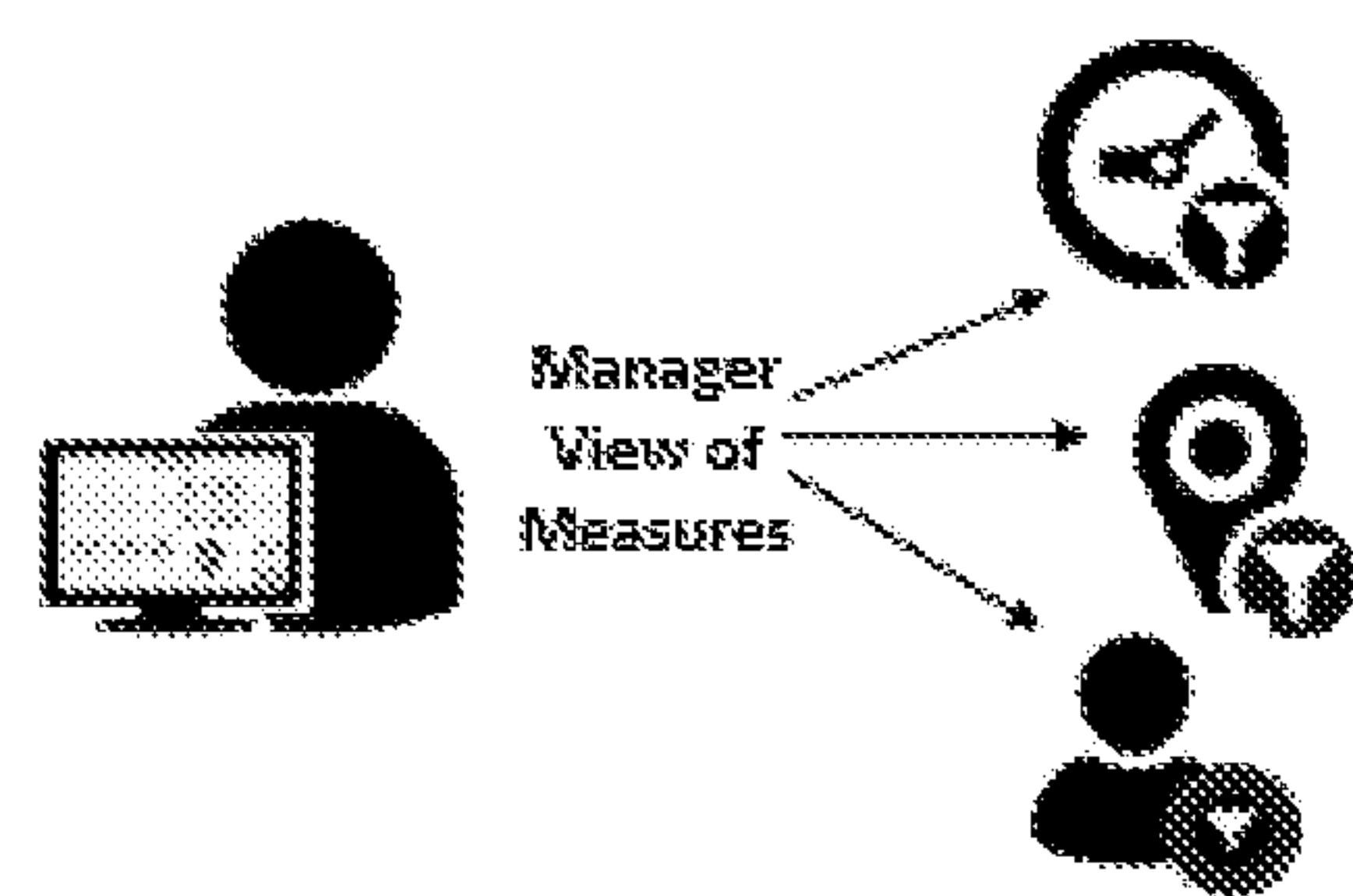
Drawing 69:



Drawing 70:

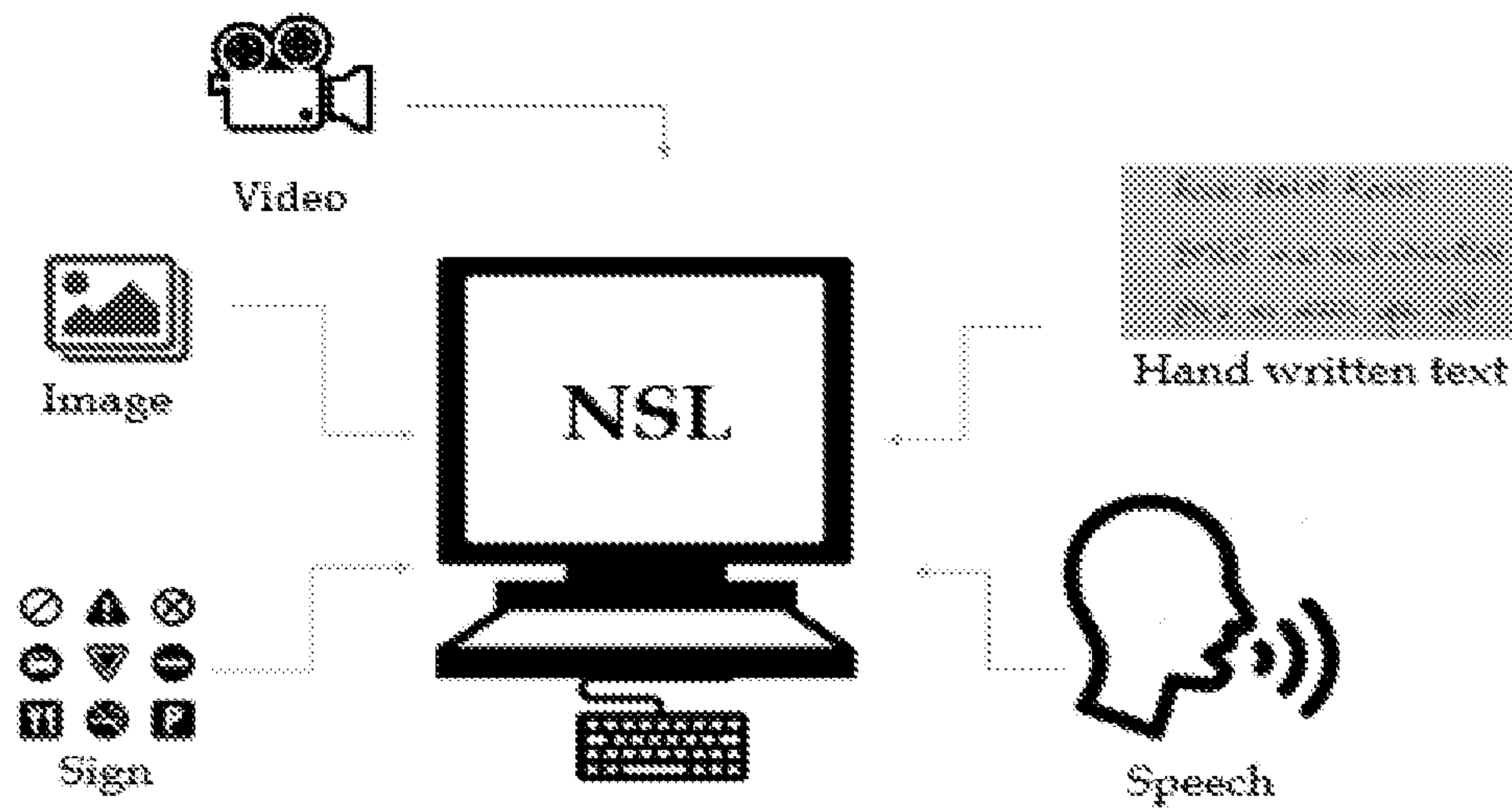


Drawing 71:

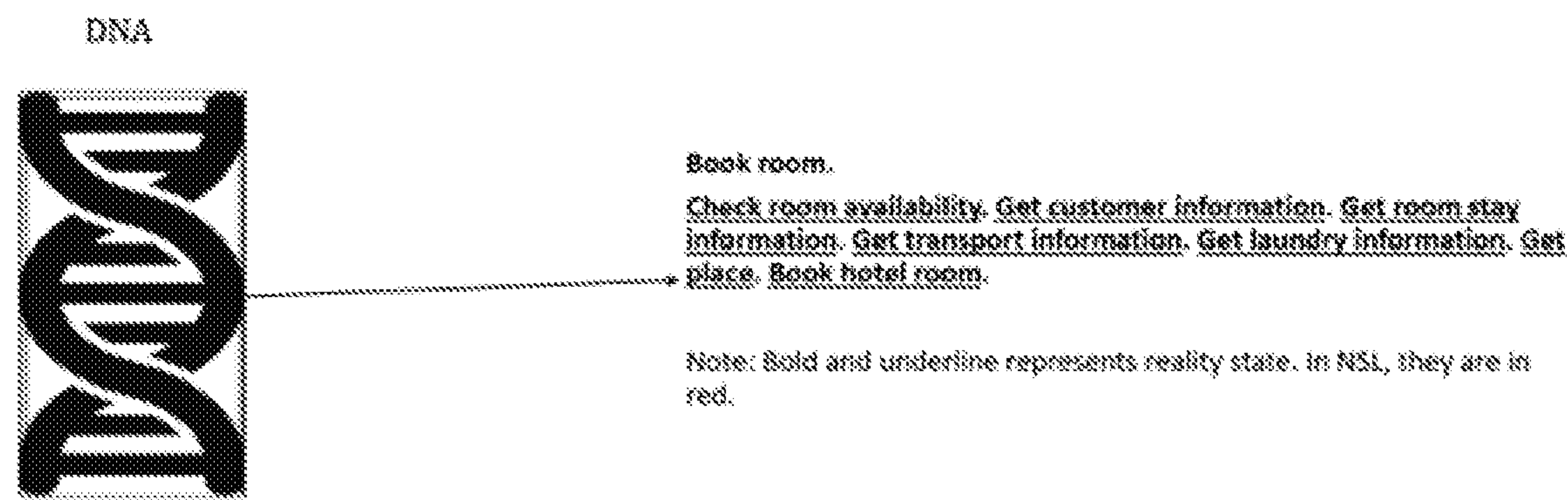


Every stakeholder can process only entities of importance dynamically using various filters – time-filter, space-filter, agent-filter, etc.,

Drawing 72:



Drawing 73:

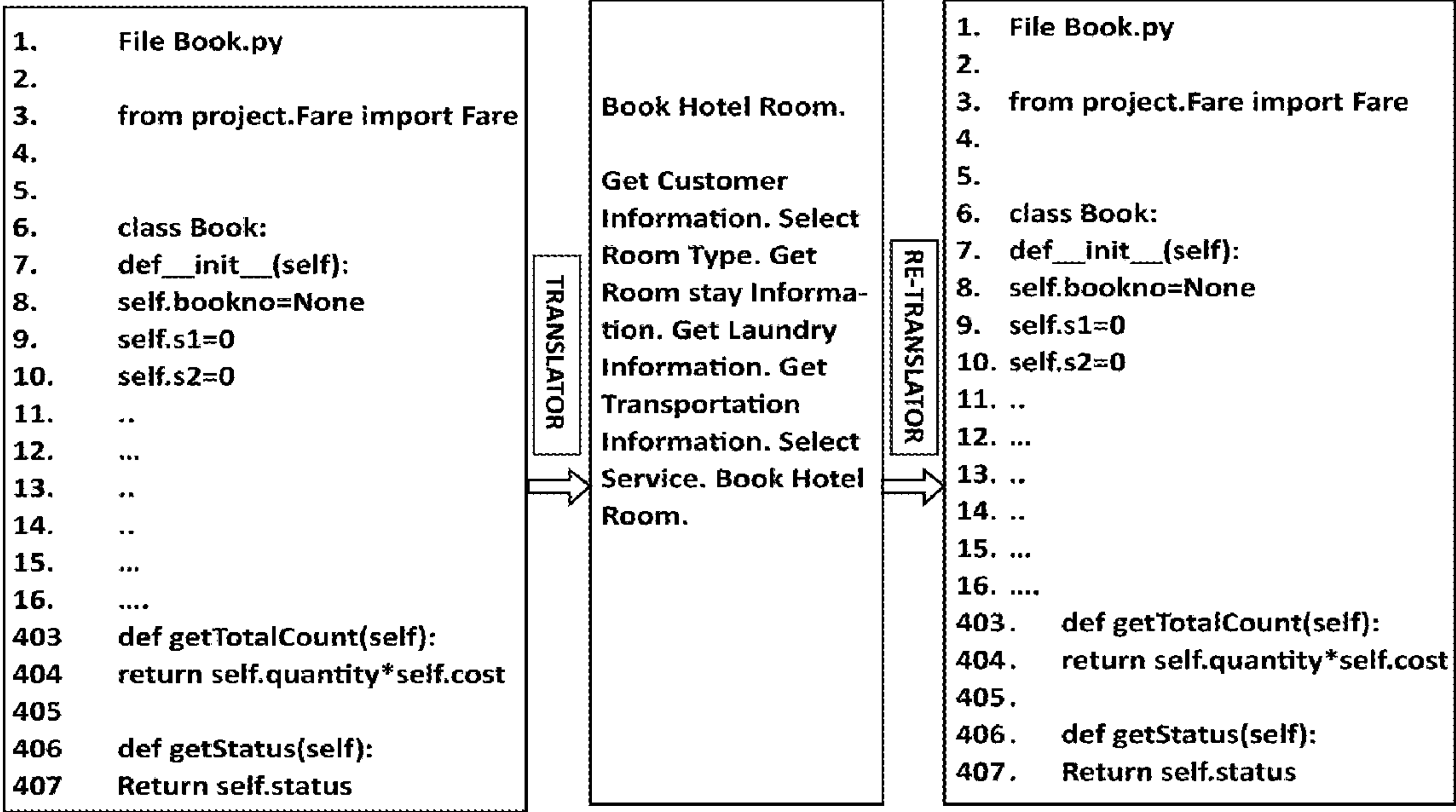


Drawing 74:

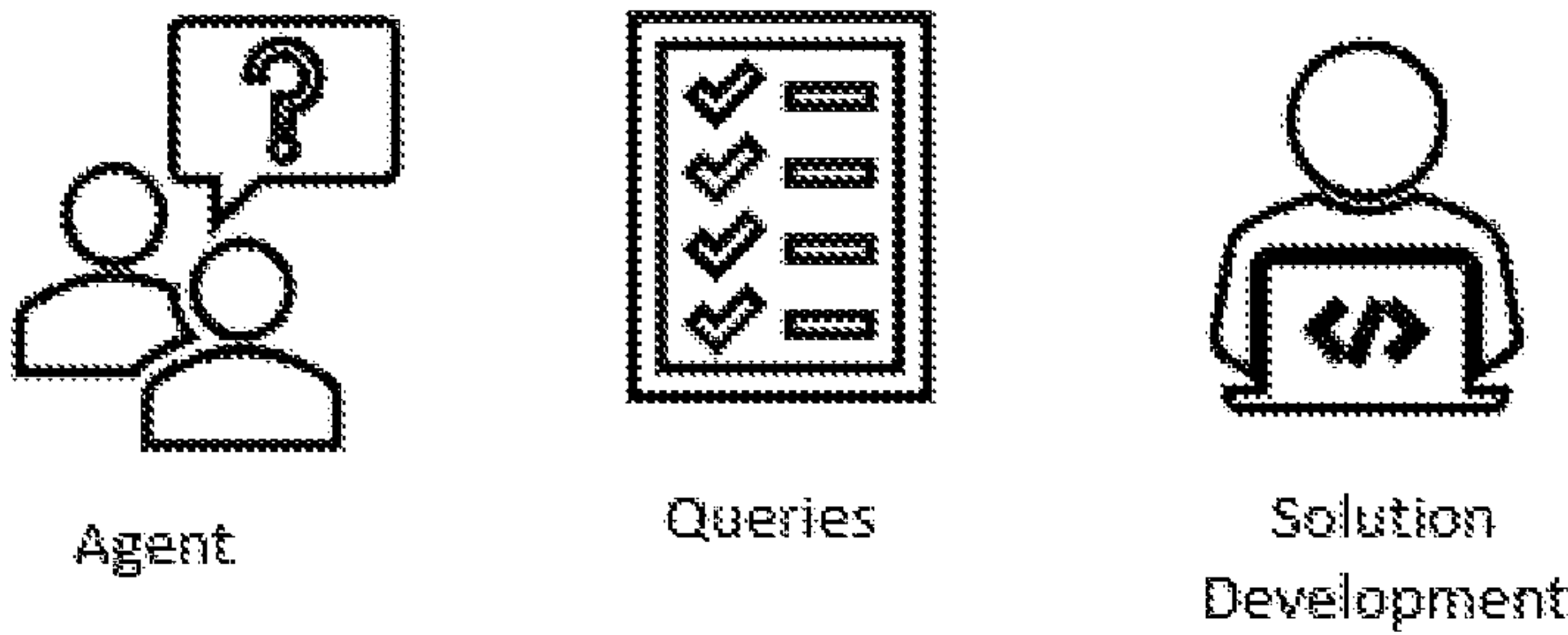
Book room

Check room availability. Get Customer information.
Get room stay information. Get transport information.
Get laundry information. Get price. Book room.

DRAWING 75:



Drawing 76:



Drawing 77:

Natural Solutions Language : Book room.

Natural Language : Book a Hotel Room.

Natural Solutions Language : Make Payment.

Natural Language: Make payment for the invoice.

Drawing 78:

LSI:

Call Centre Officer _____ Customer

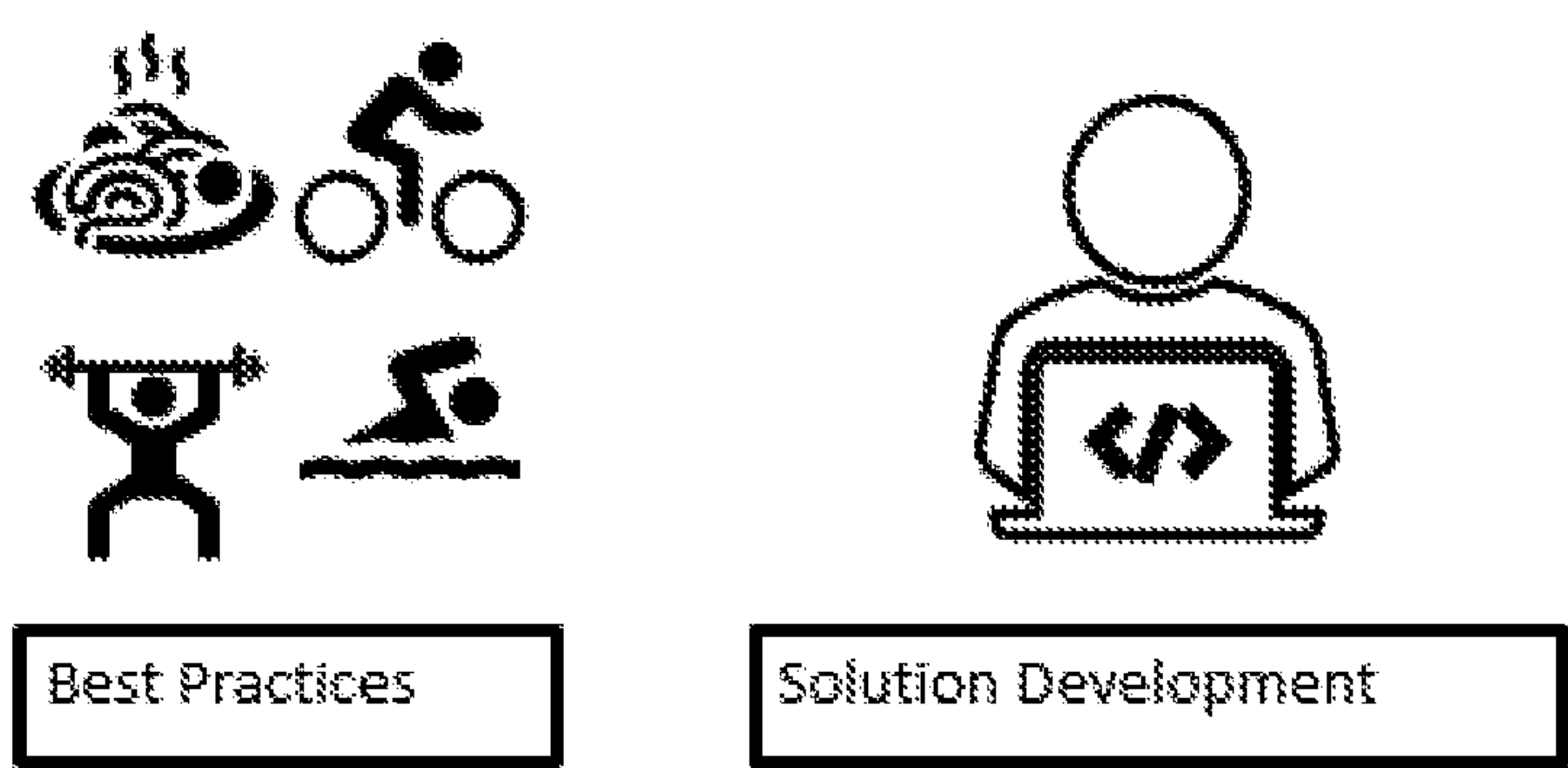
{ Suggested Verbs: Call, Mail, validate,...}

LSI:

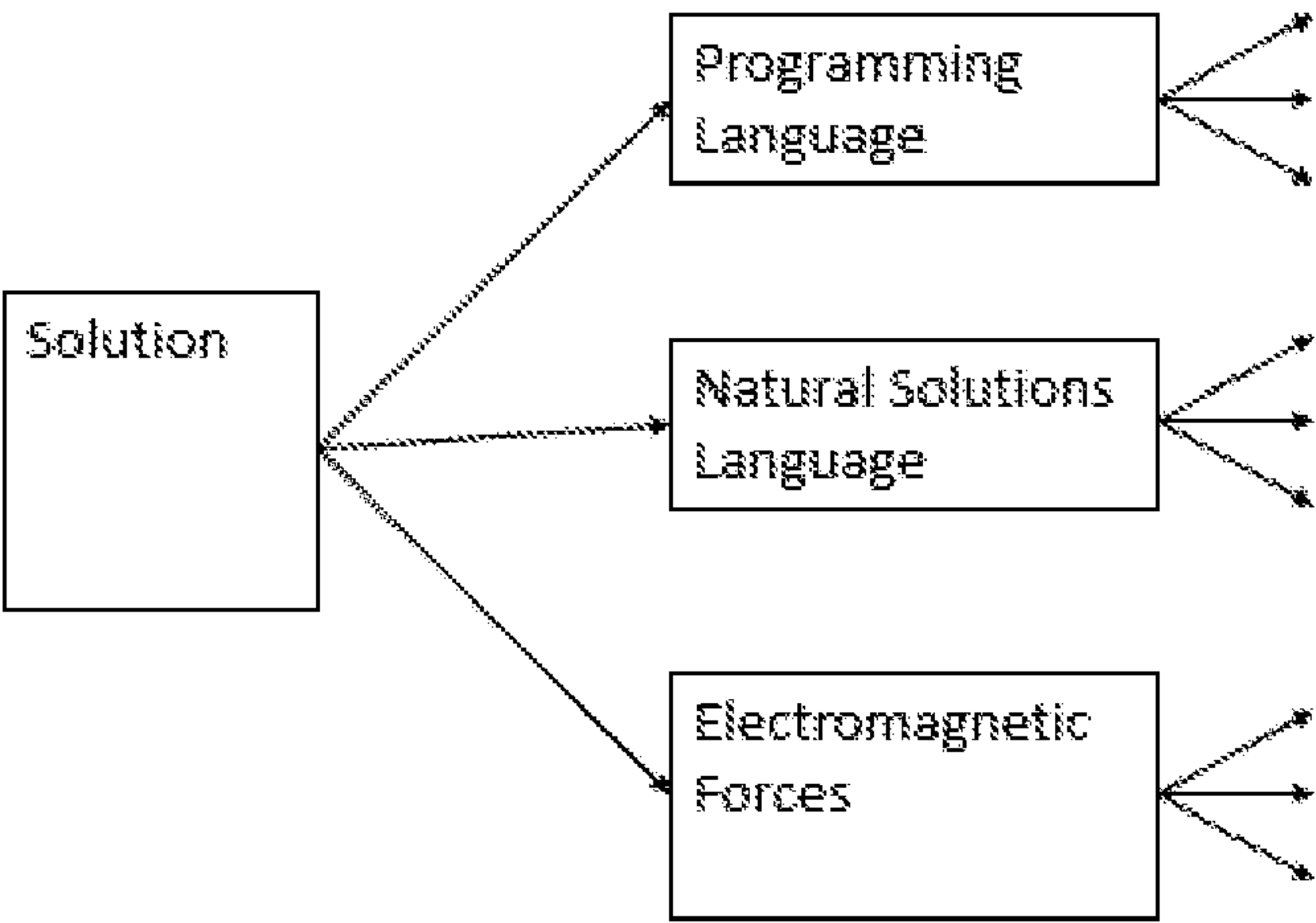
_____ Customer Information

{ Suggested Verbs: Get, Update, Check, validate,...}

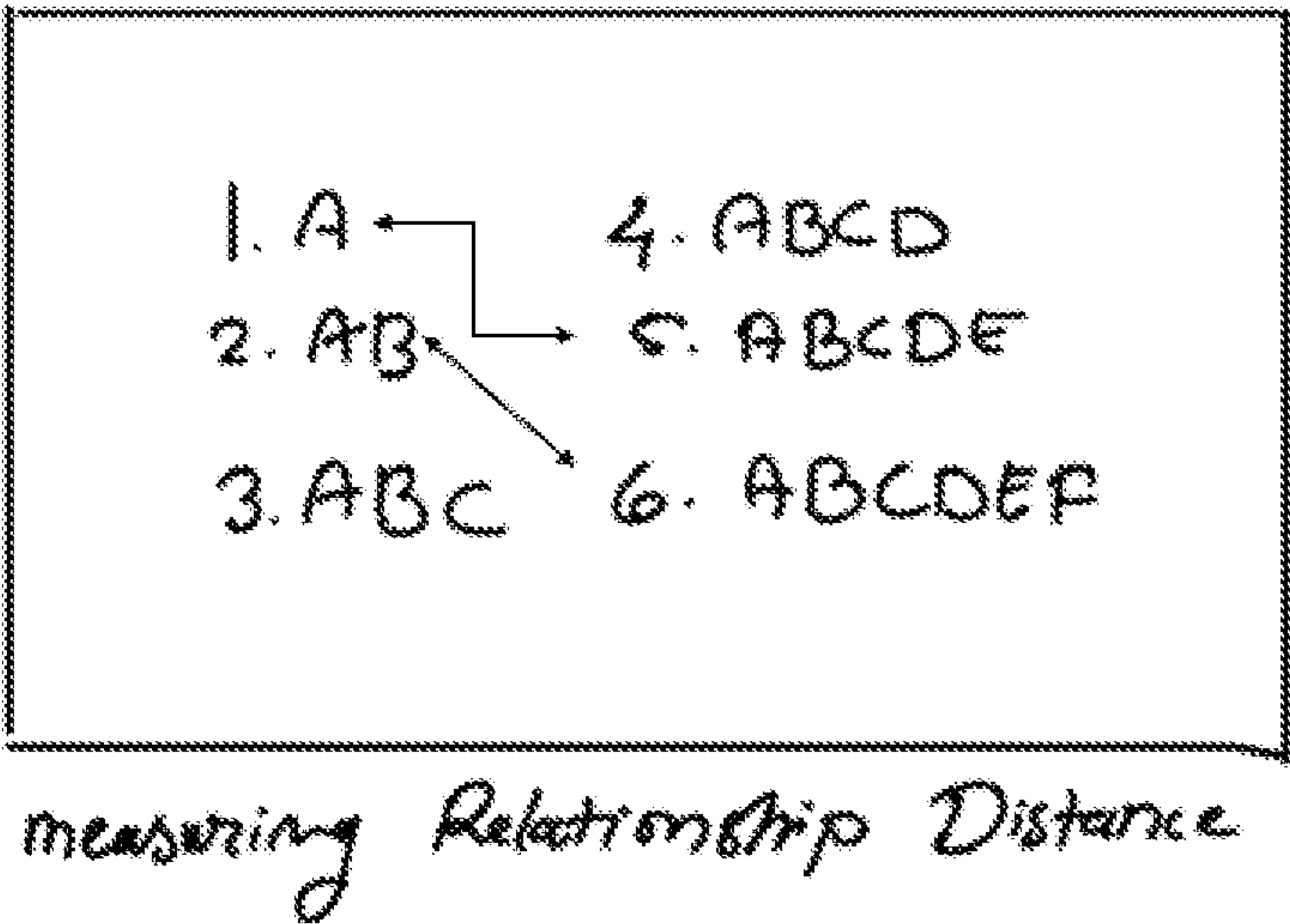
Drawing 79:



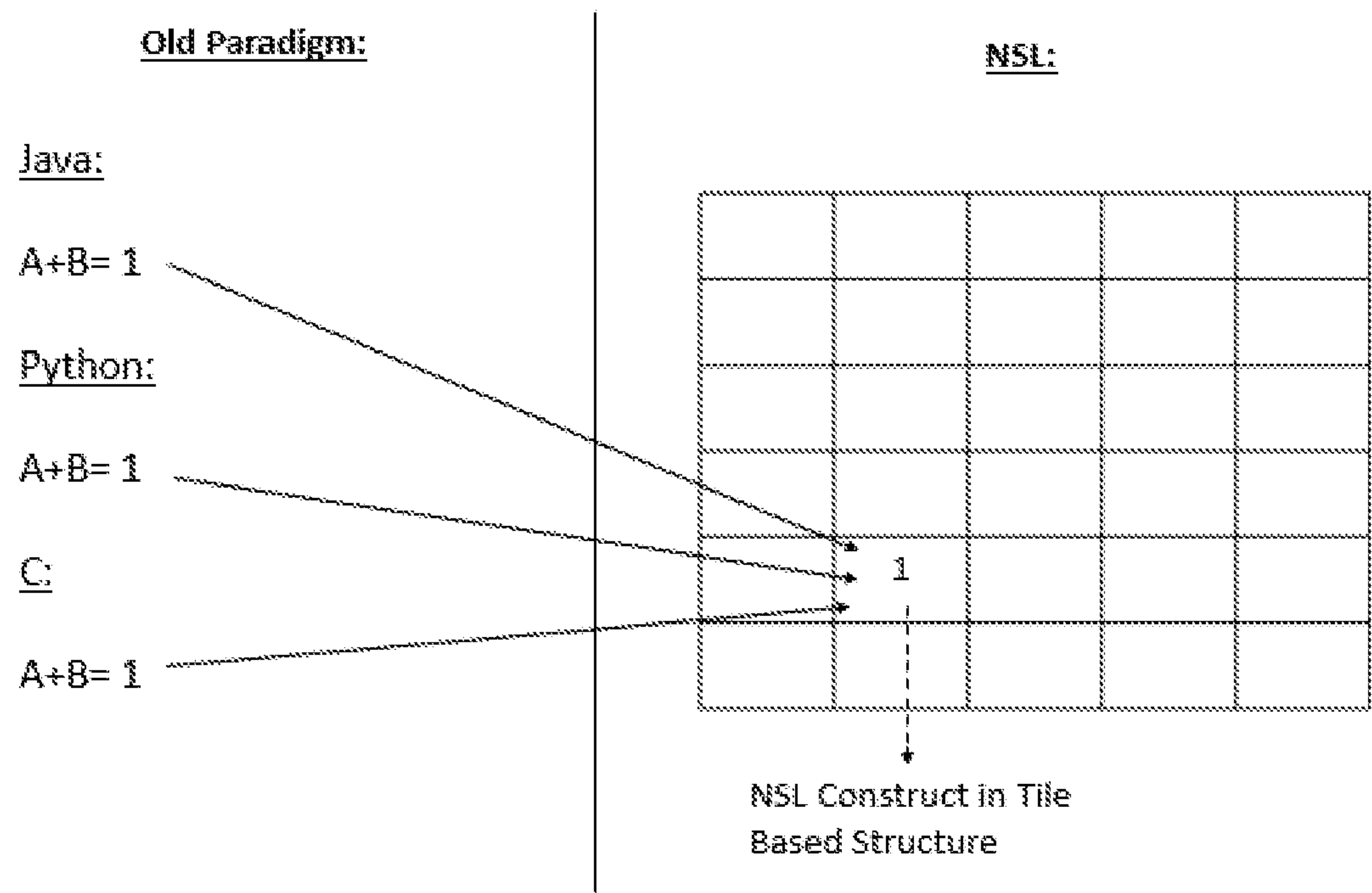
Drawing 80:



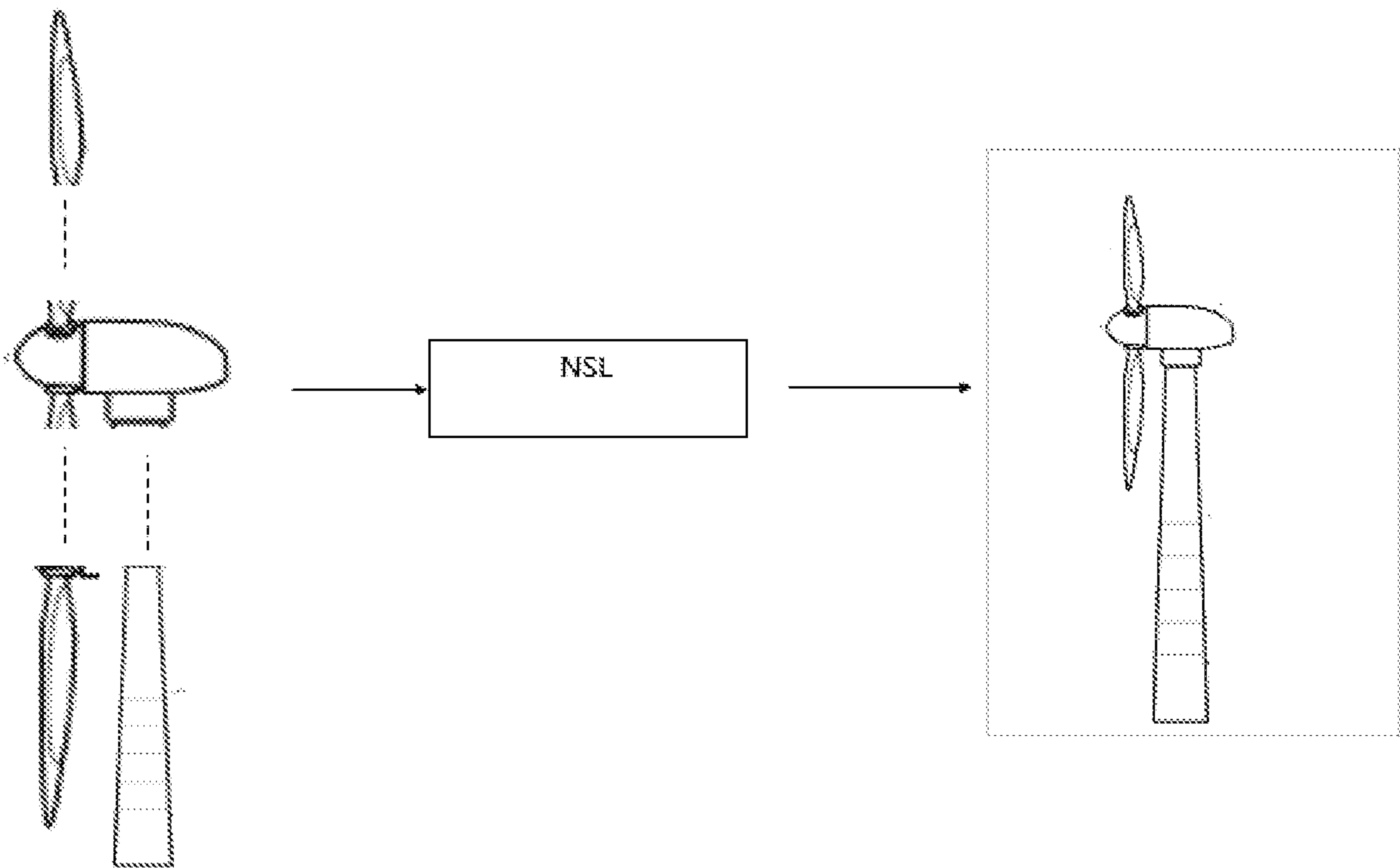
Drawing 81:



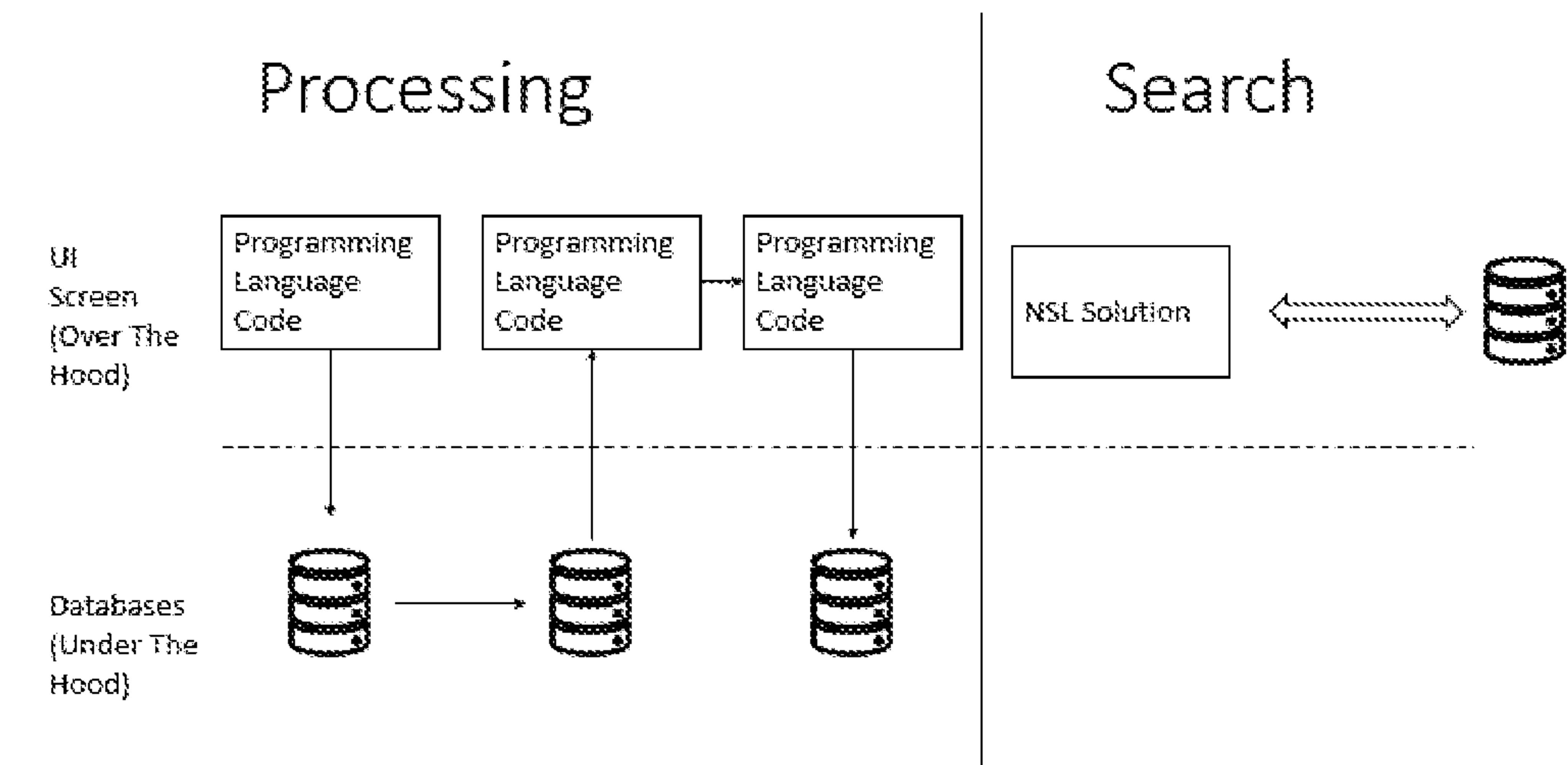
Drawing 82:



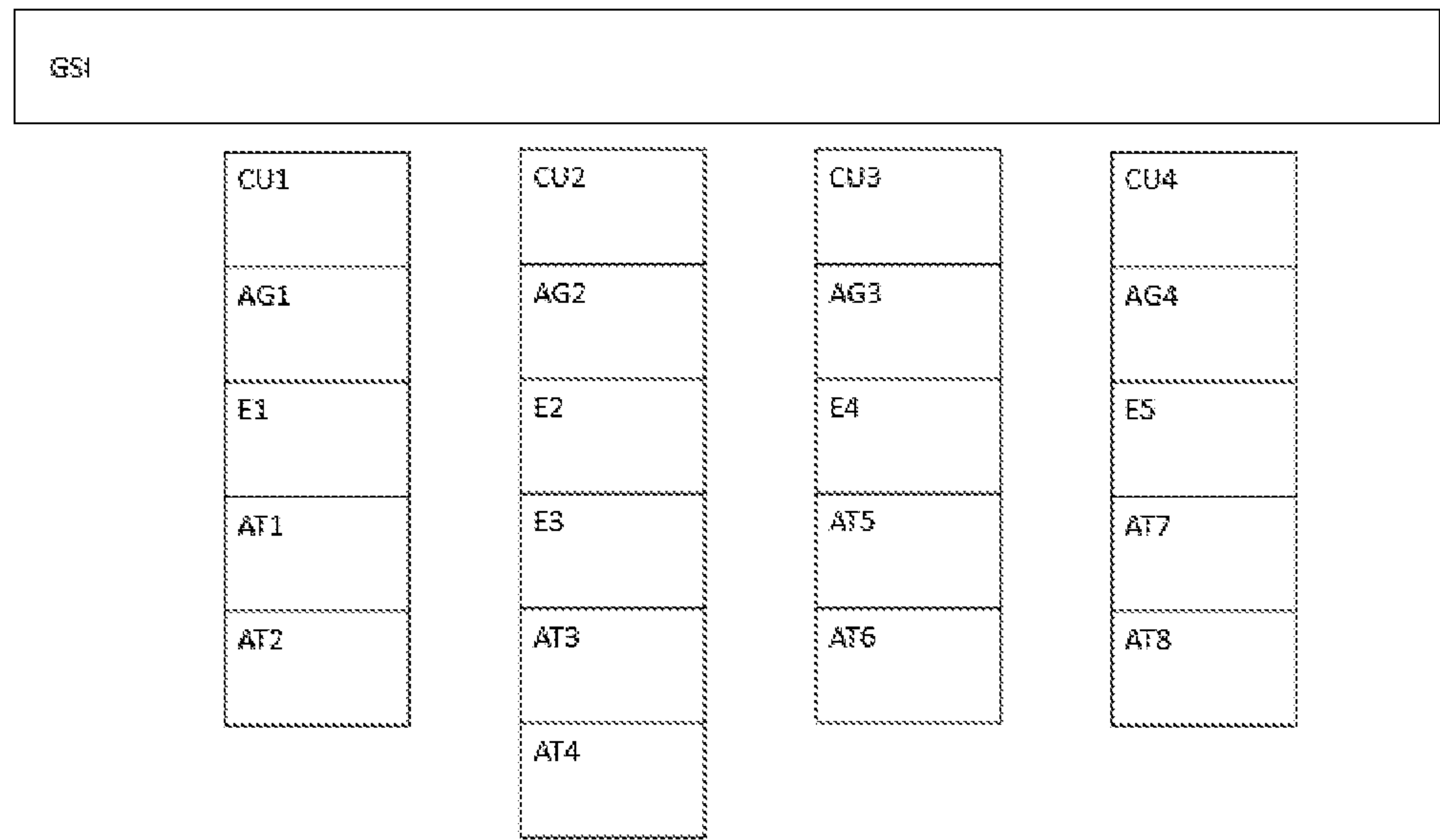
Drawing 83:



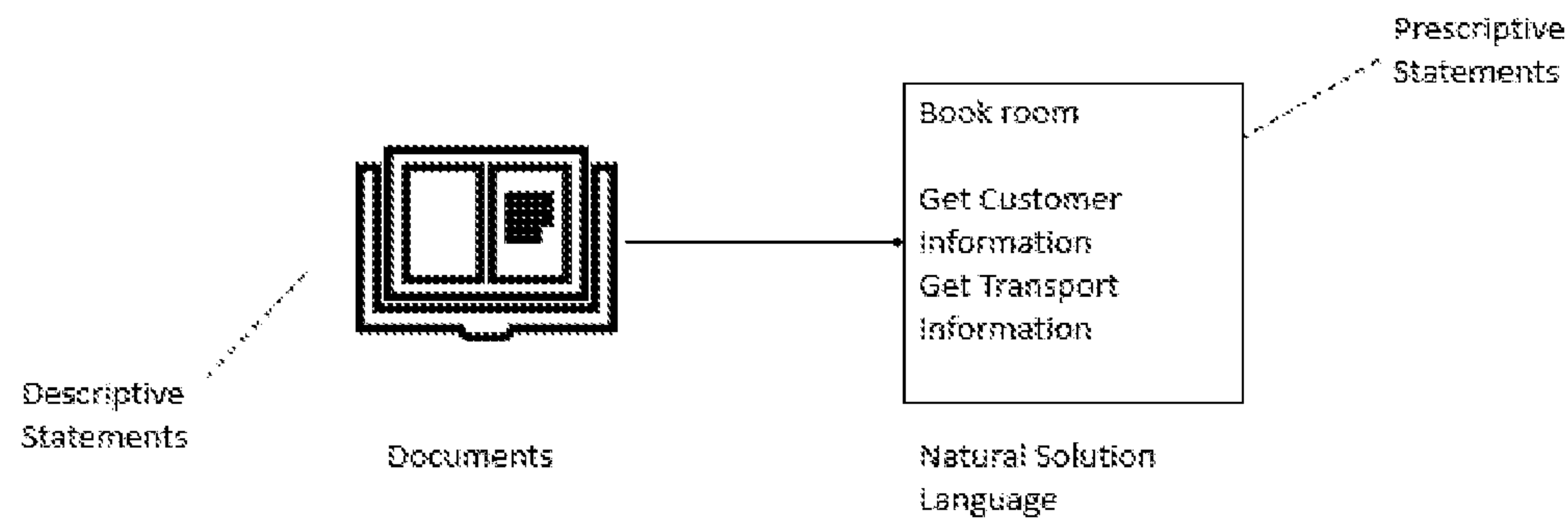
Drawing 84:



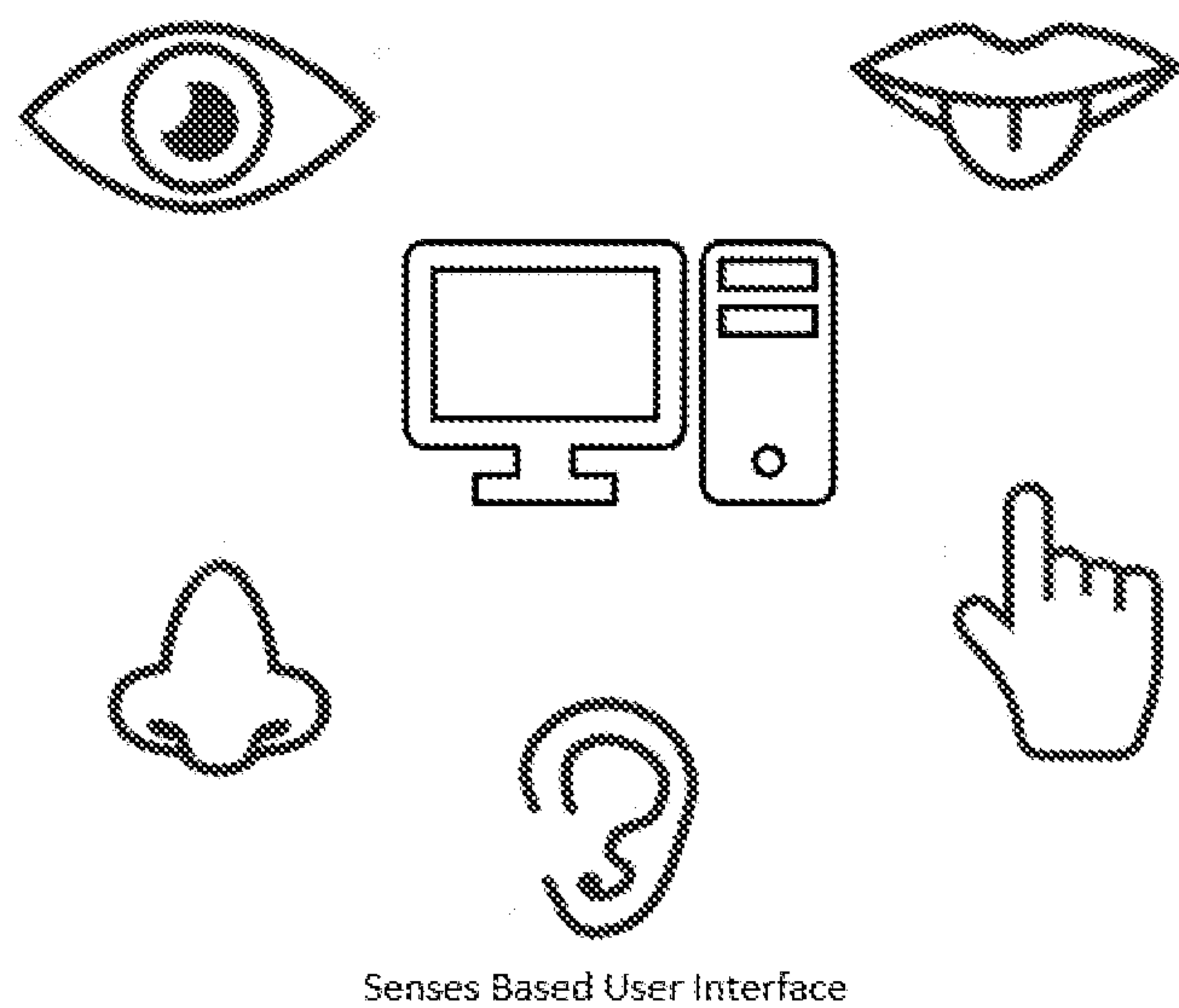
Drawing 85:



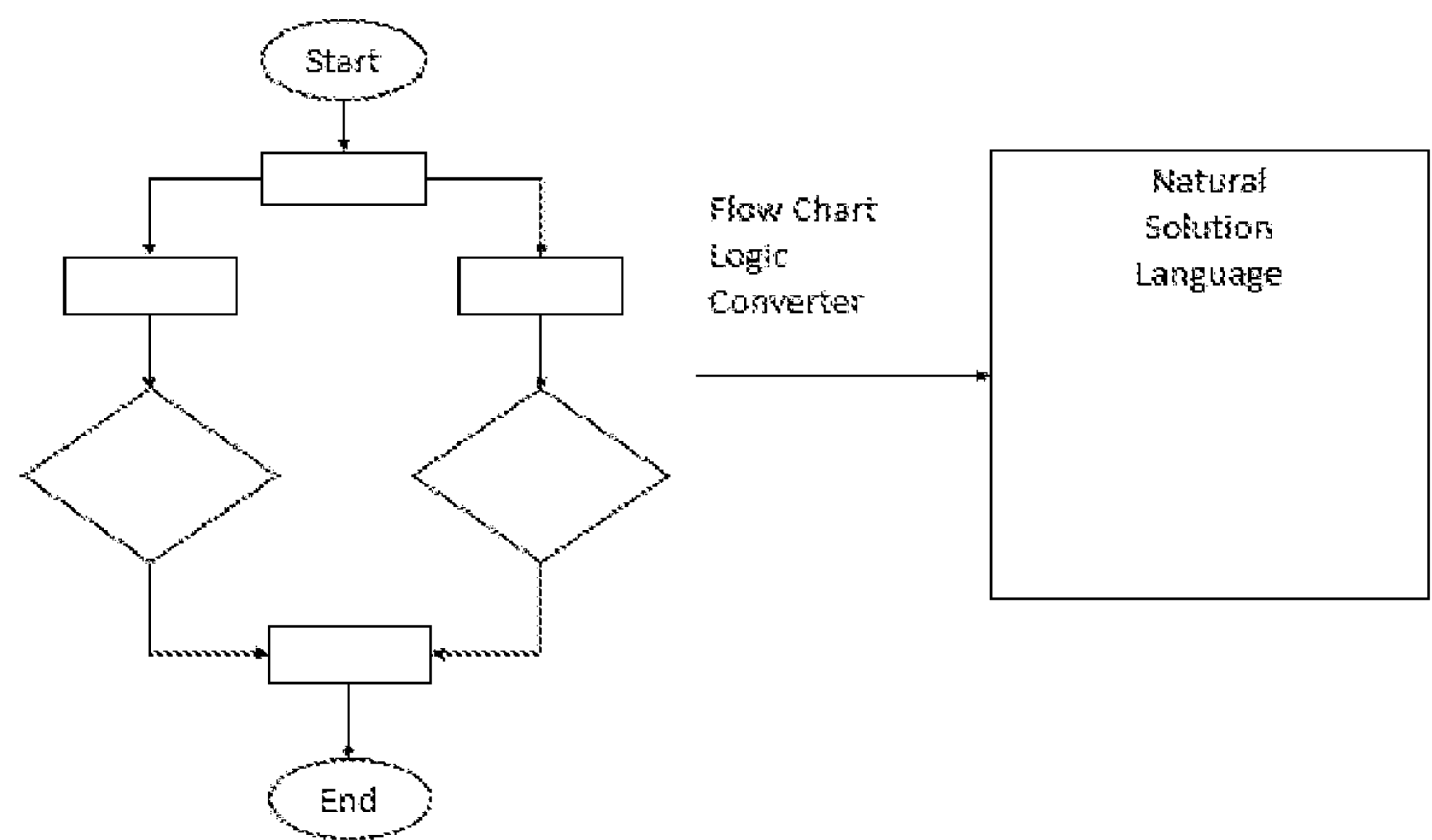
Drawing 86:



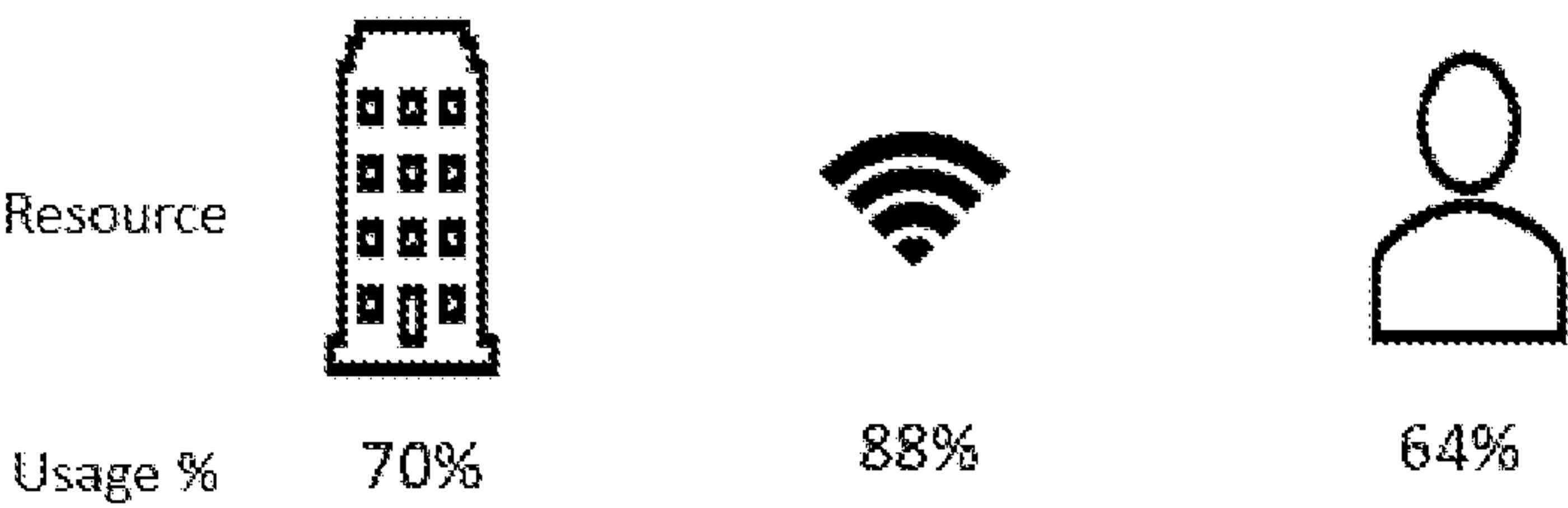
Drawing 87:



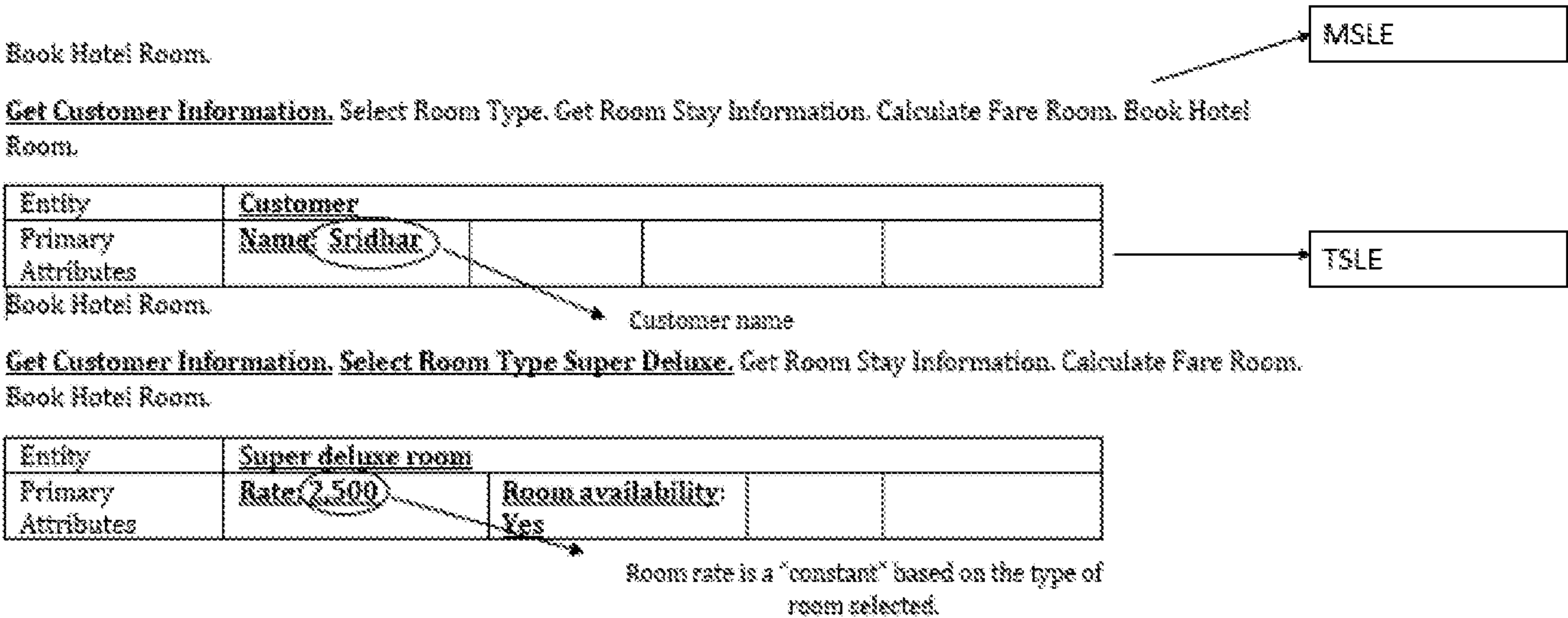
Drawing 88:



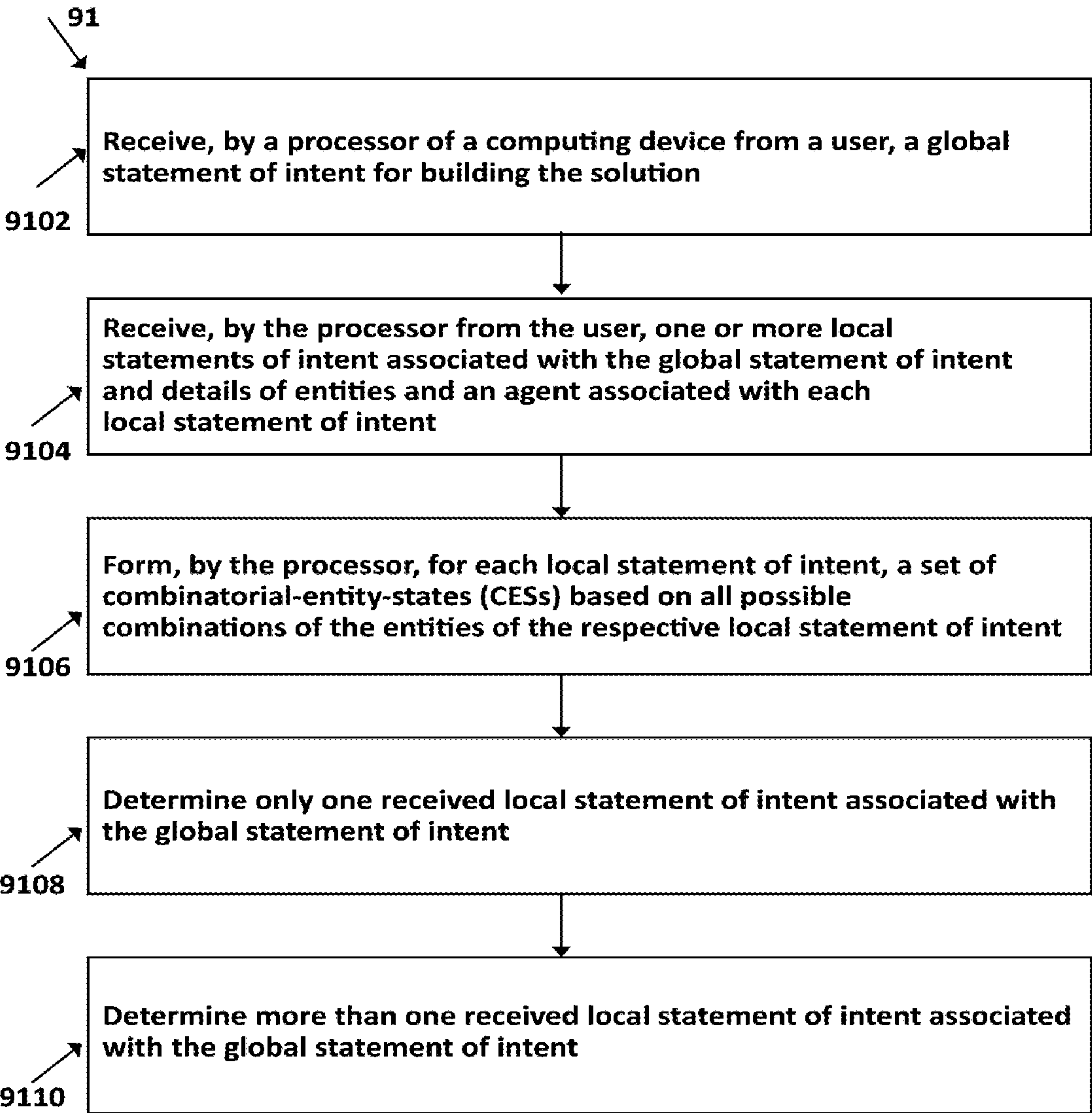
Drawing 89:



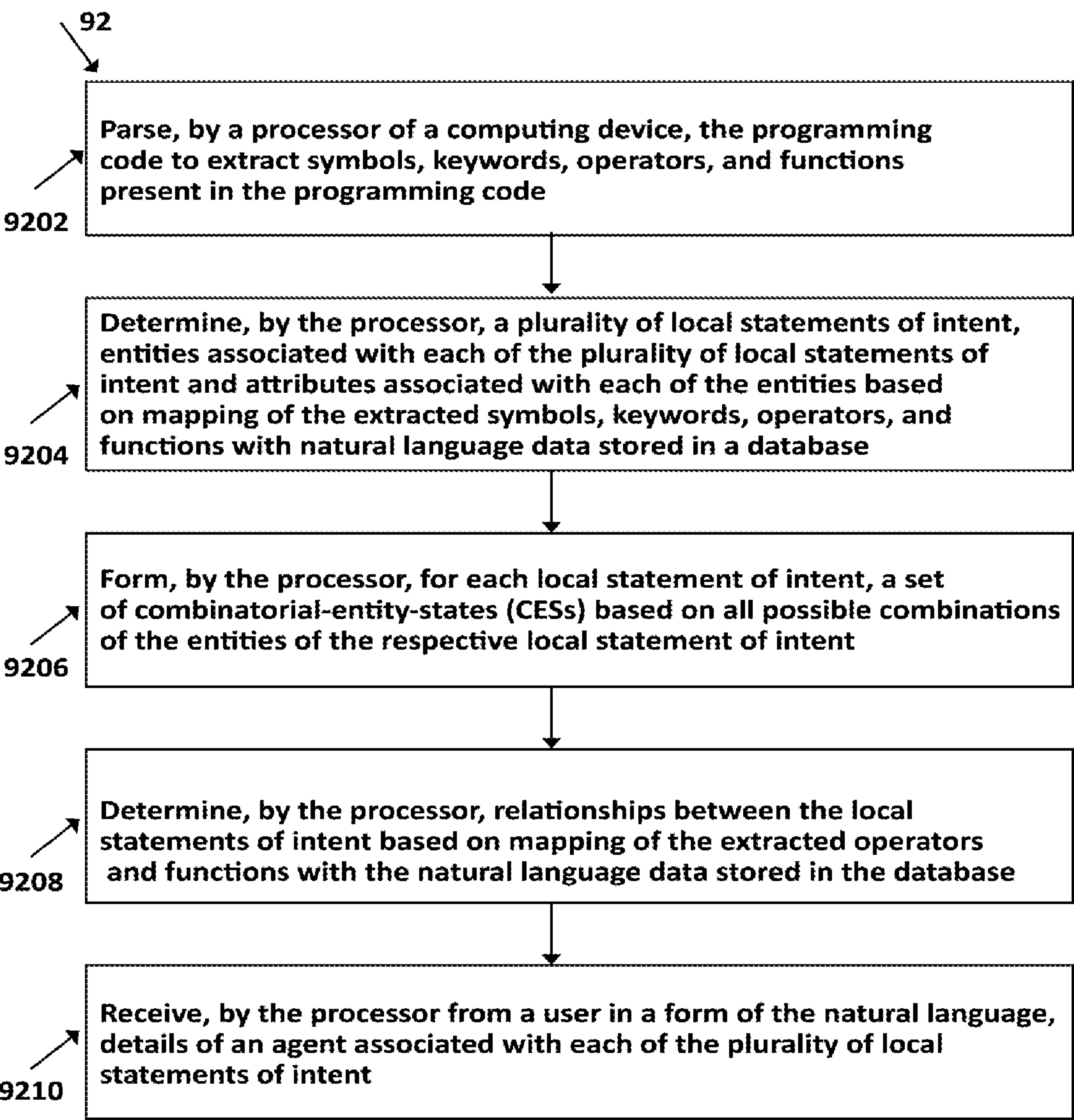
Drawing 90:



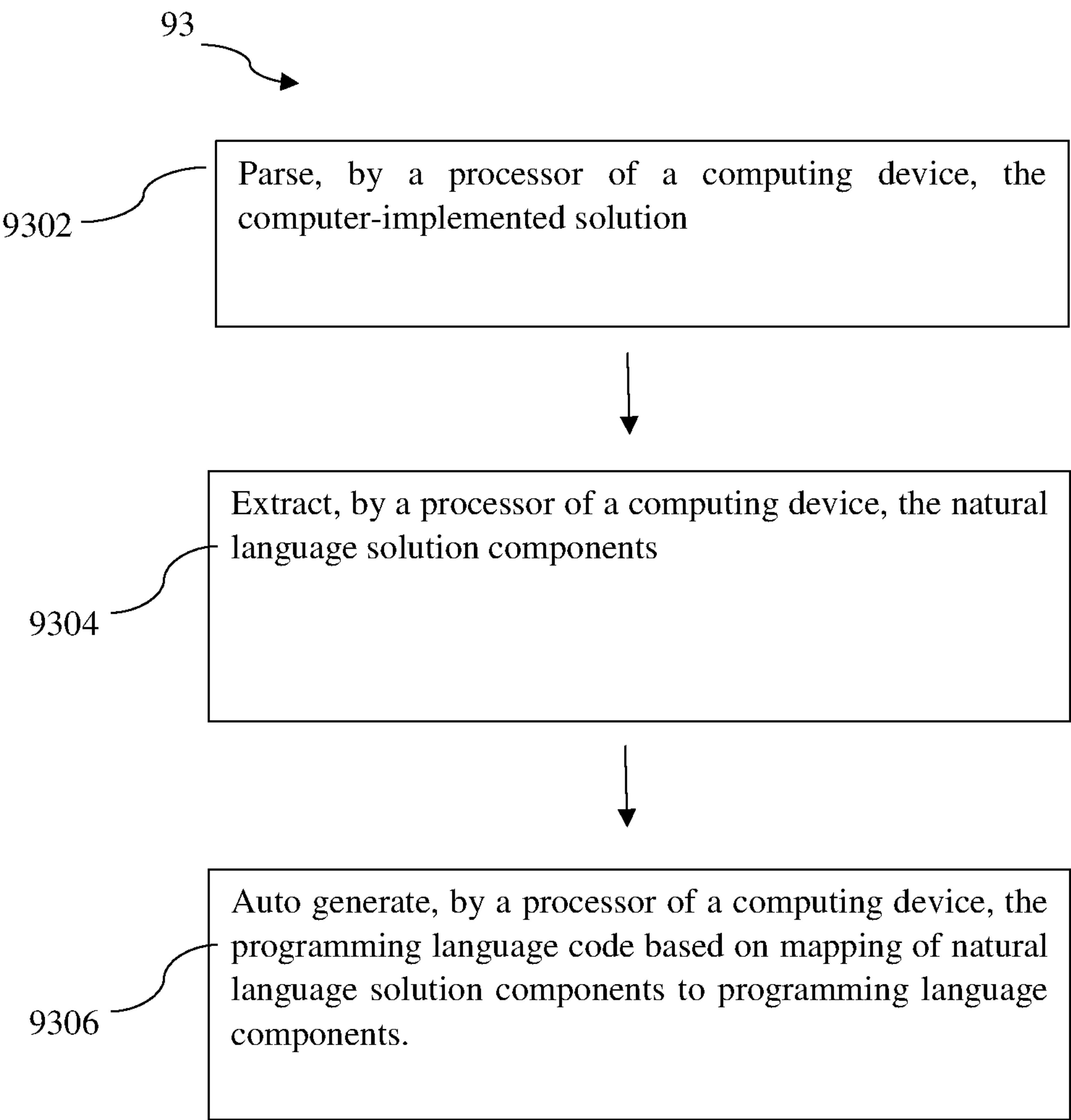
DRAWING 91:



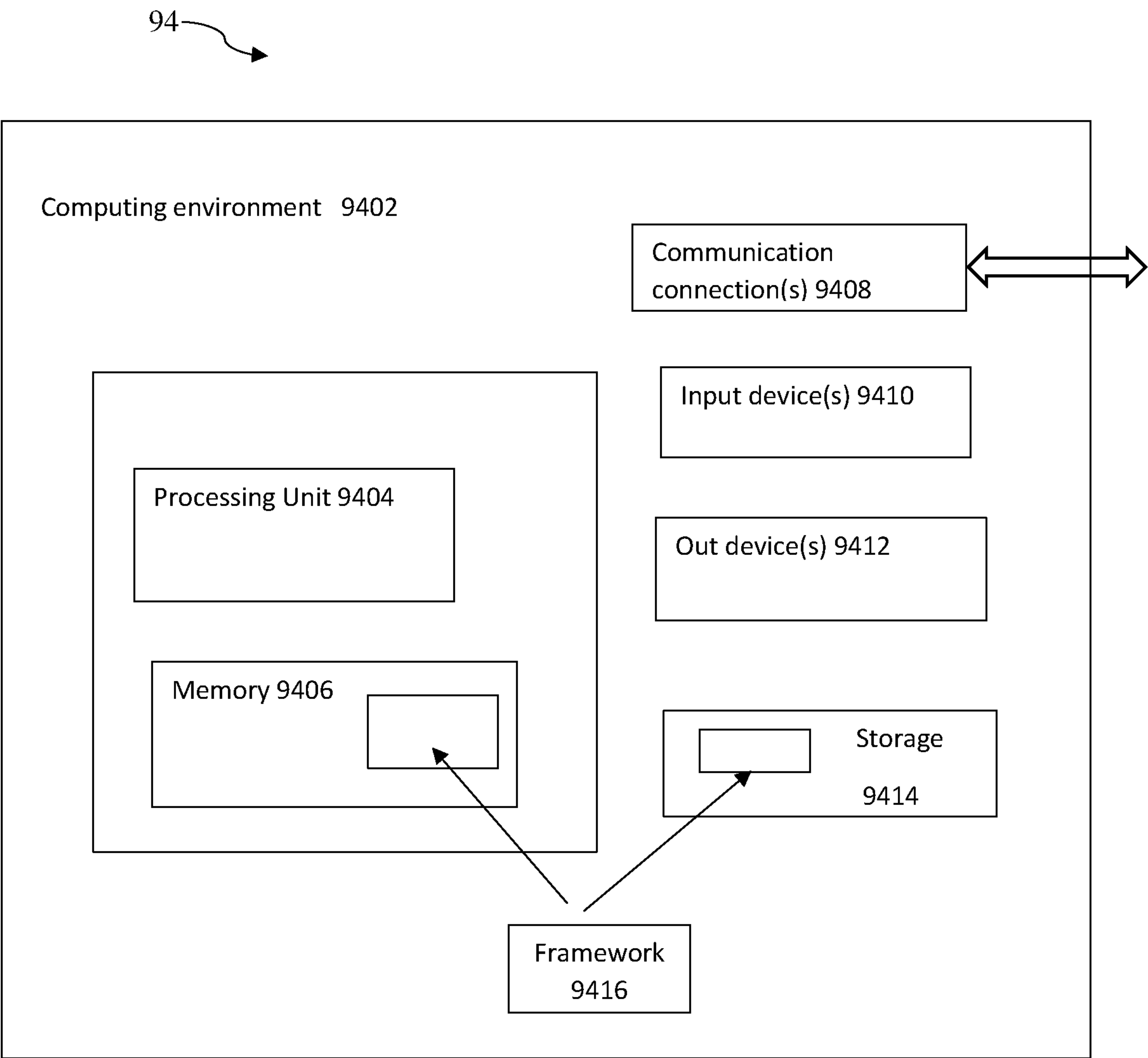
DRAWING 92:



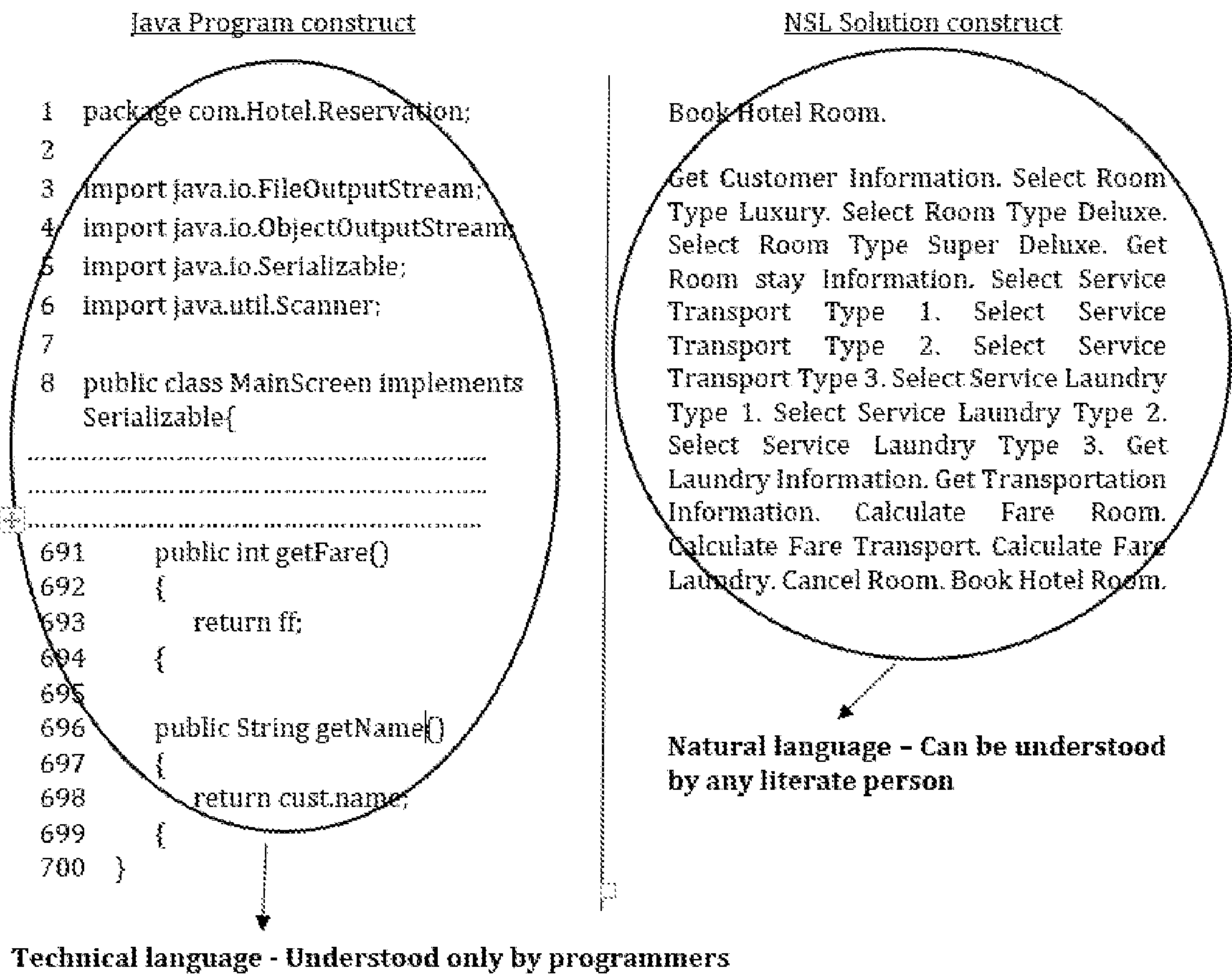
Drawing 93:



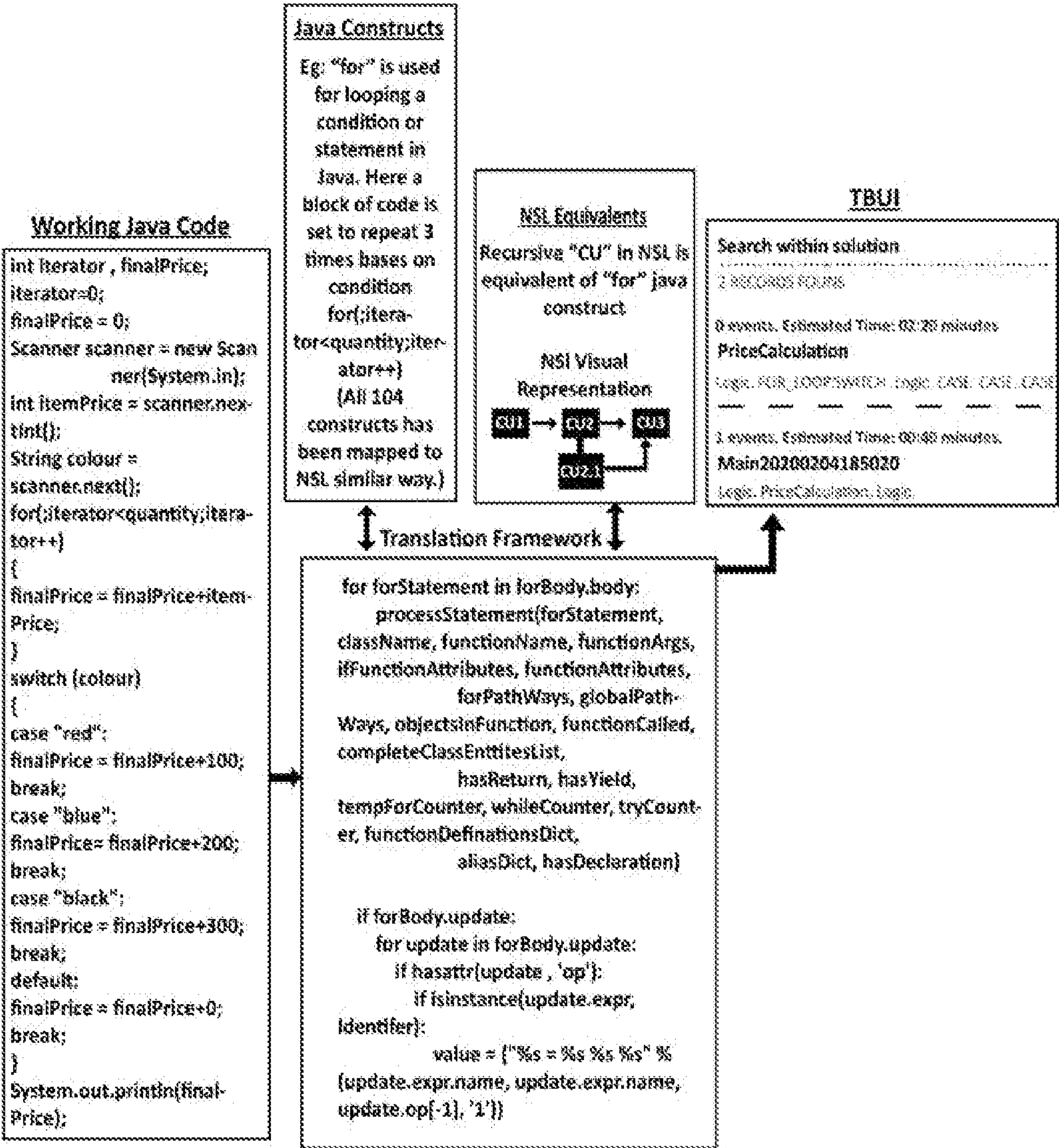
Drawing 94:



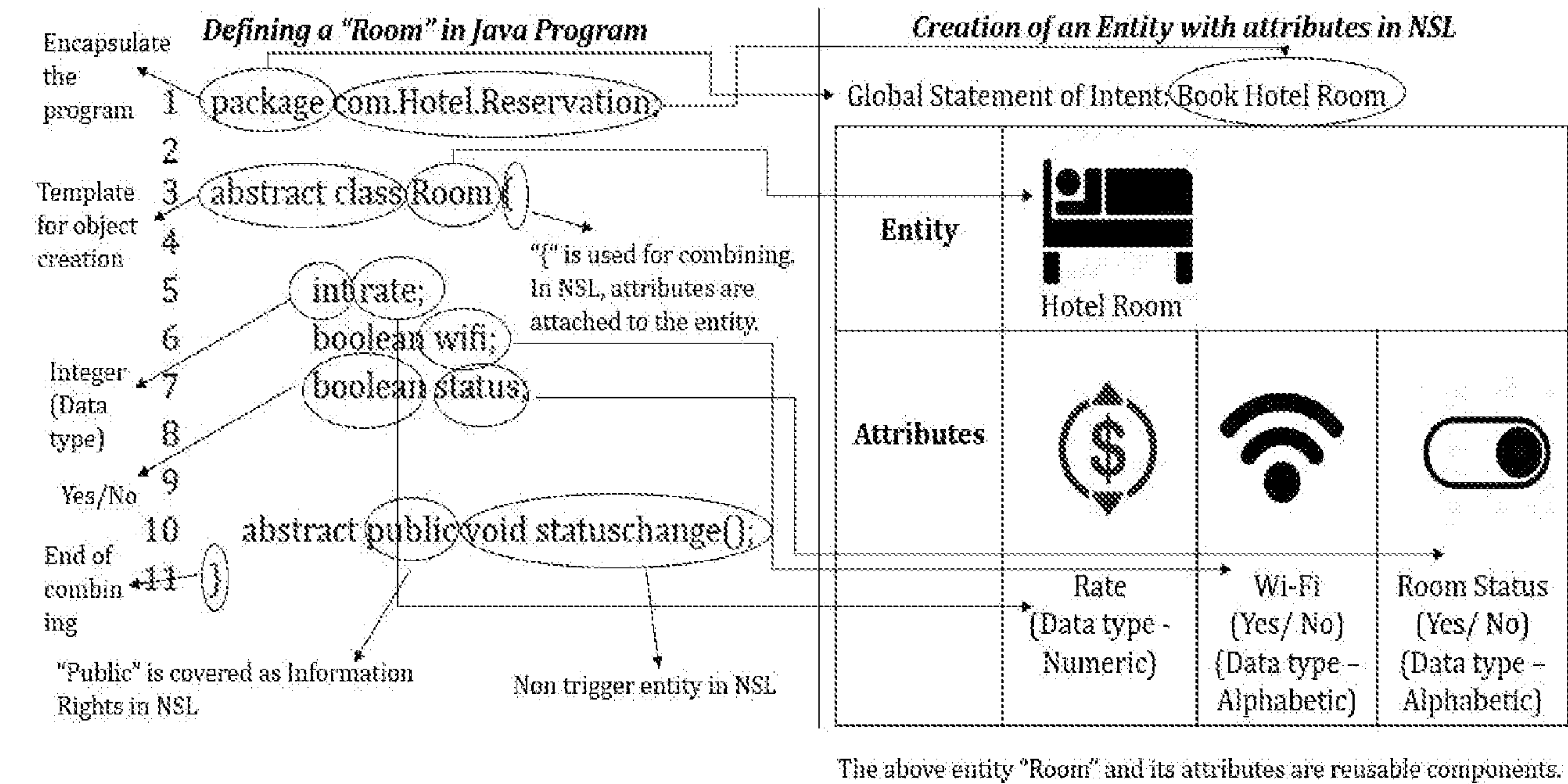
Drawing 95



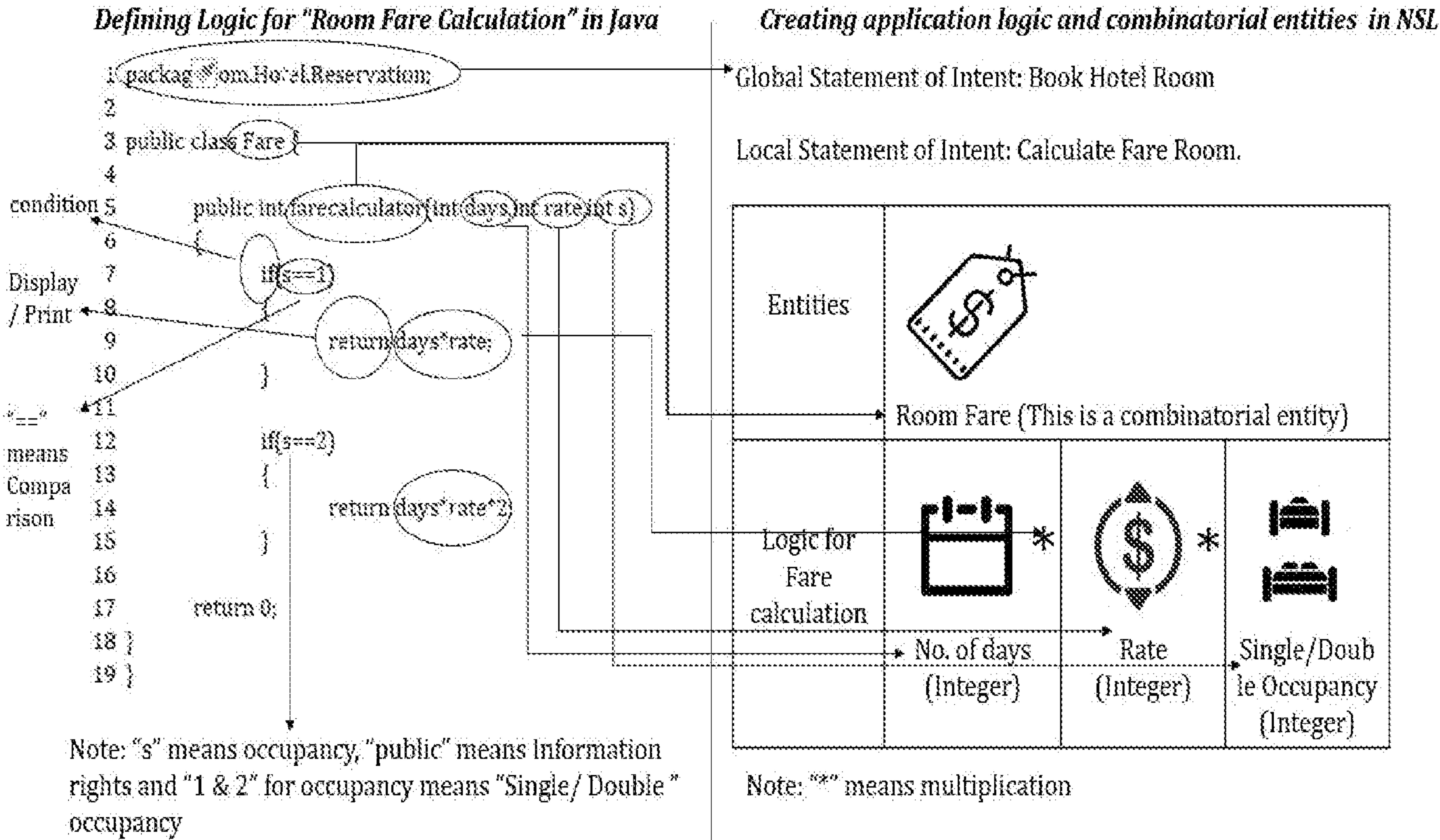
DRAWING 96:



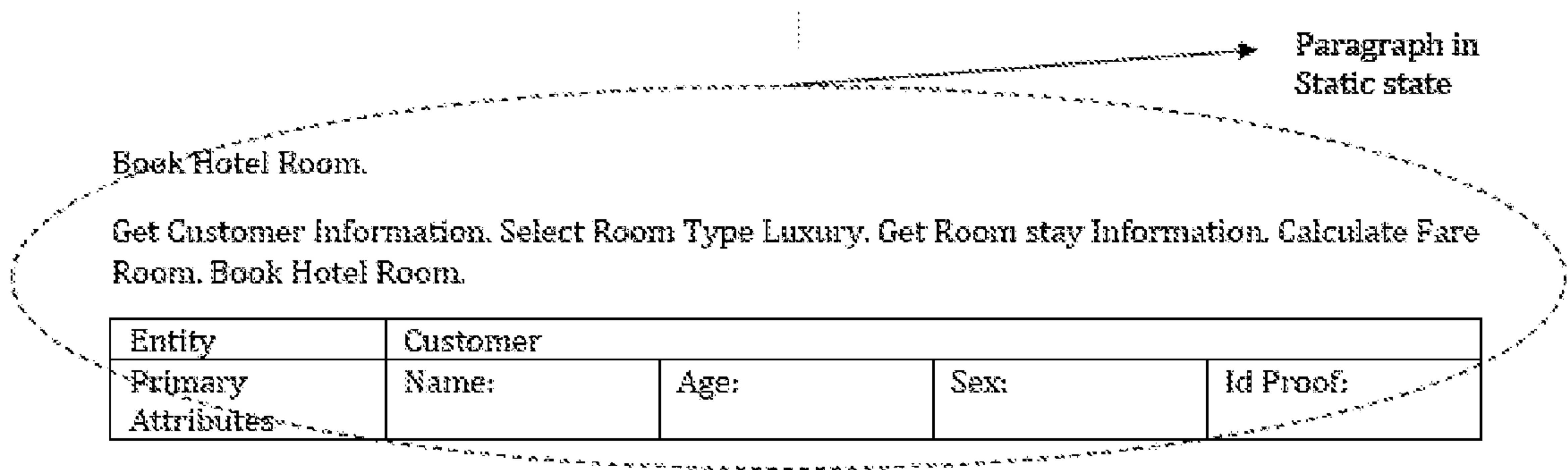
Drawing 97A



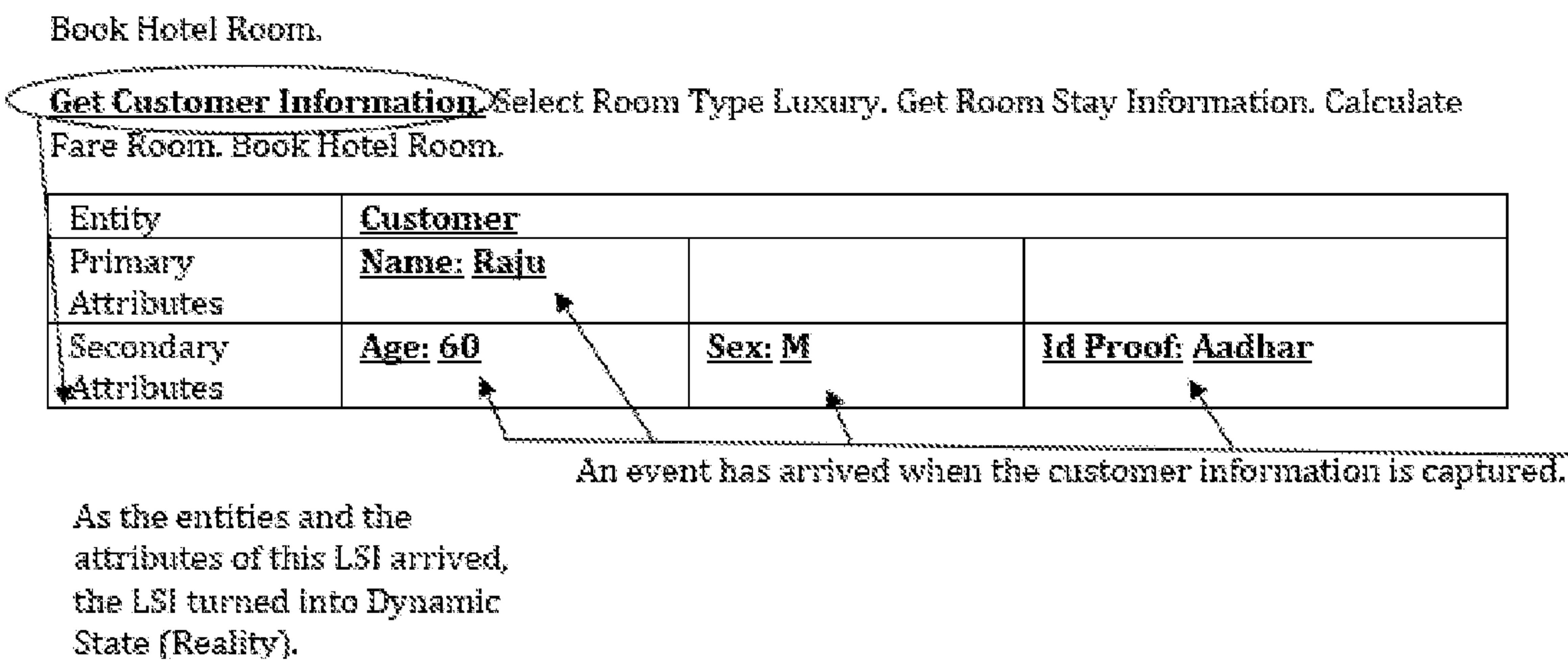
Drawing 97B



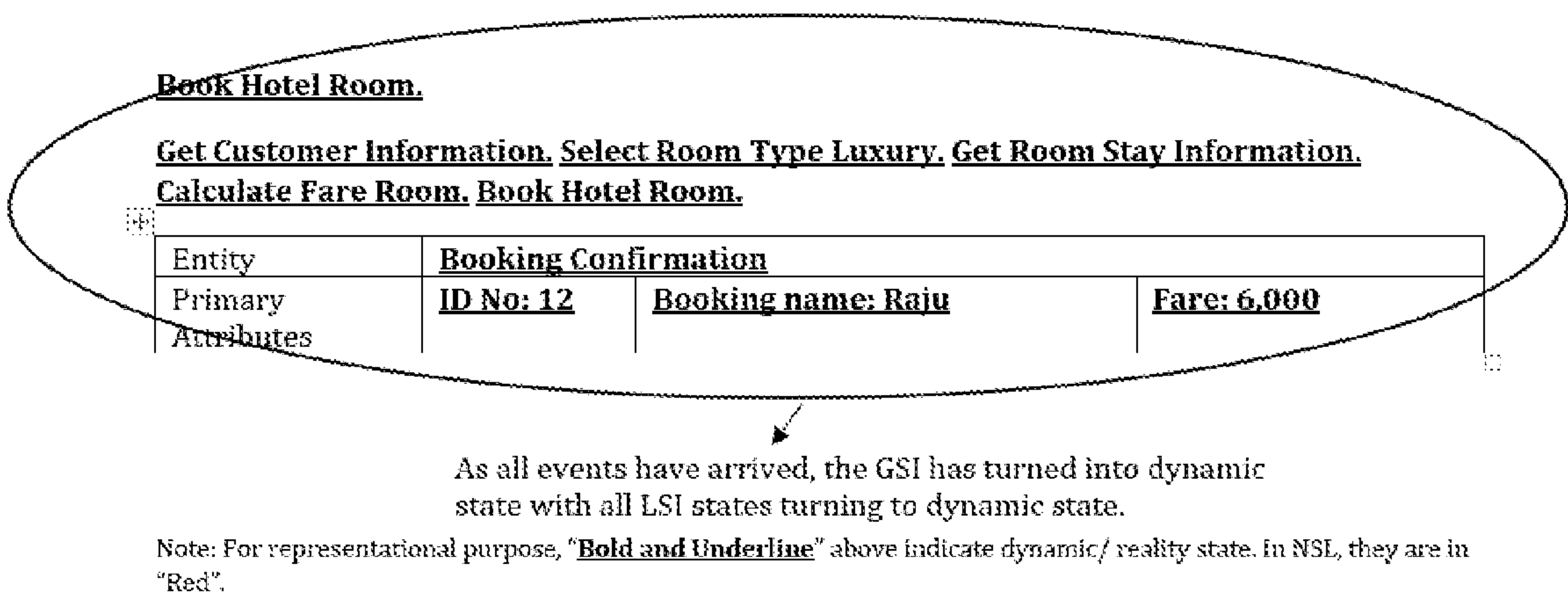
Drawing 98A



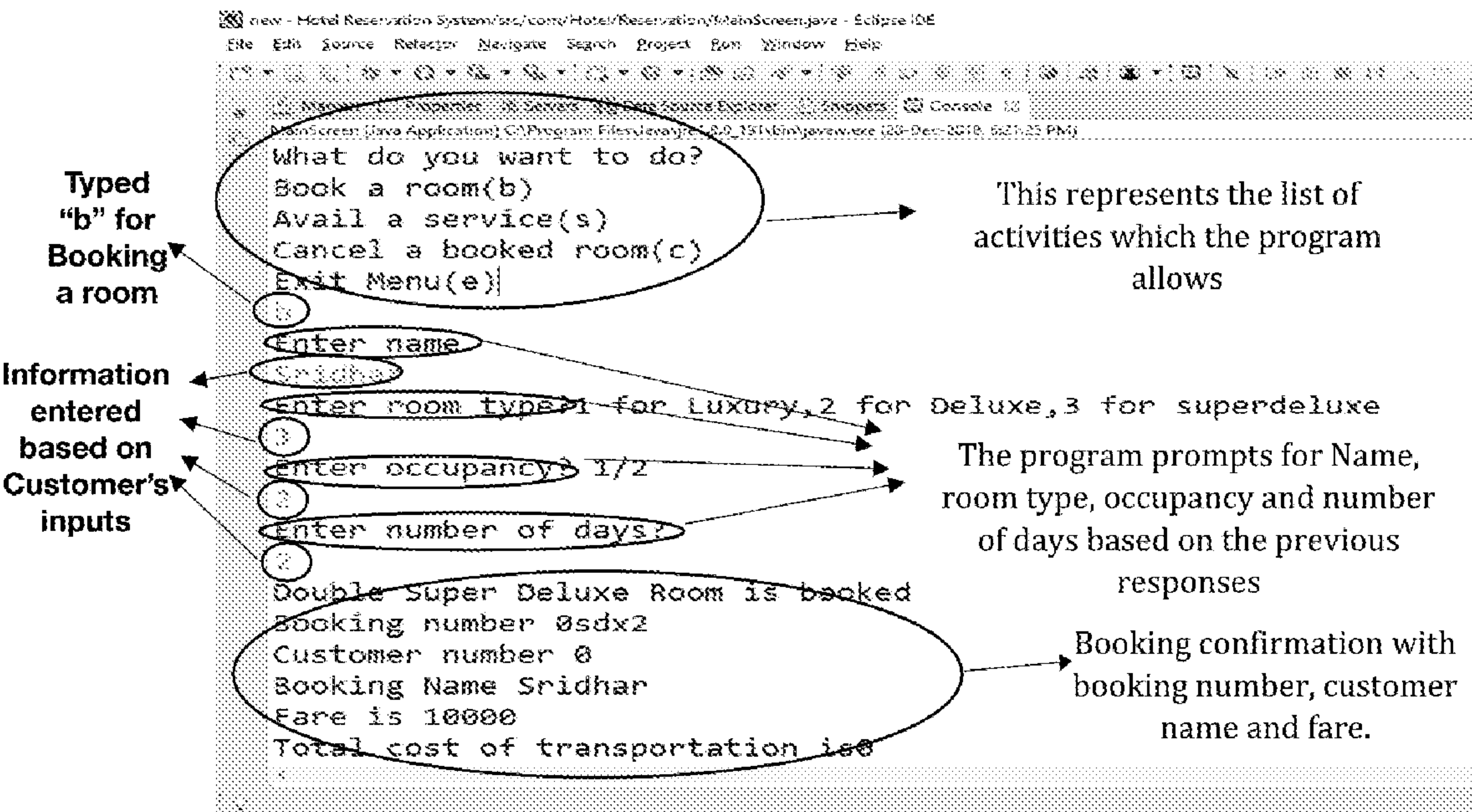
Drawing 98B



Drawing 98C



DRAWING 99A:



Drawing 99B

Book Hotel Room.

Get Customer Information. Select Room Type. Get Room Stay Information. Calculate Fare Room. Book Hotel Room.

Entity	<u>Customer</u>			
Primary Attributes	<u>Name:</u> <u>Sridhar</u>			

Book Hotel Room.

Customer name

Get Customer Information. Select Room Type Super Deluxe. Get Room Stay Information. Calculate Fare Room. Book Hotel Room.

Entity	<u>Super deluxe room</u>			
Primary Attributes	<u>Rate:</u> <u>2,500</u>	<u>Room availability:</u>		

Yes

Room rate is a "constant" based on the type of room selected.

Book Hotel Room.

Get Customer Information. Select Room Type Super Deluxe. Get Room Stay Information. Calculate Fare Room. Book Hotel Room.

Entity	<u>Room stay Information</u>		
Primary Attributes	<u>Number of days:</u> <u>2</u>	<u>Occupancy:</u> <u>2</u>	

Entered based on customer's inputs

Book Hotel Room.

Get Customer Information. Select Room Type Super Deluxe. Get Room Stay Information. Calculate Fare Room. Book Hotel Room.

Entity	<u>Room fare</u>
Primary Attributes	<u>Number of days * Rate *</u> <u>Occupancy:</u> <u>10,000</u>

Fare calculated by the machine agent

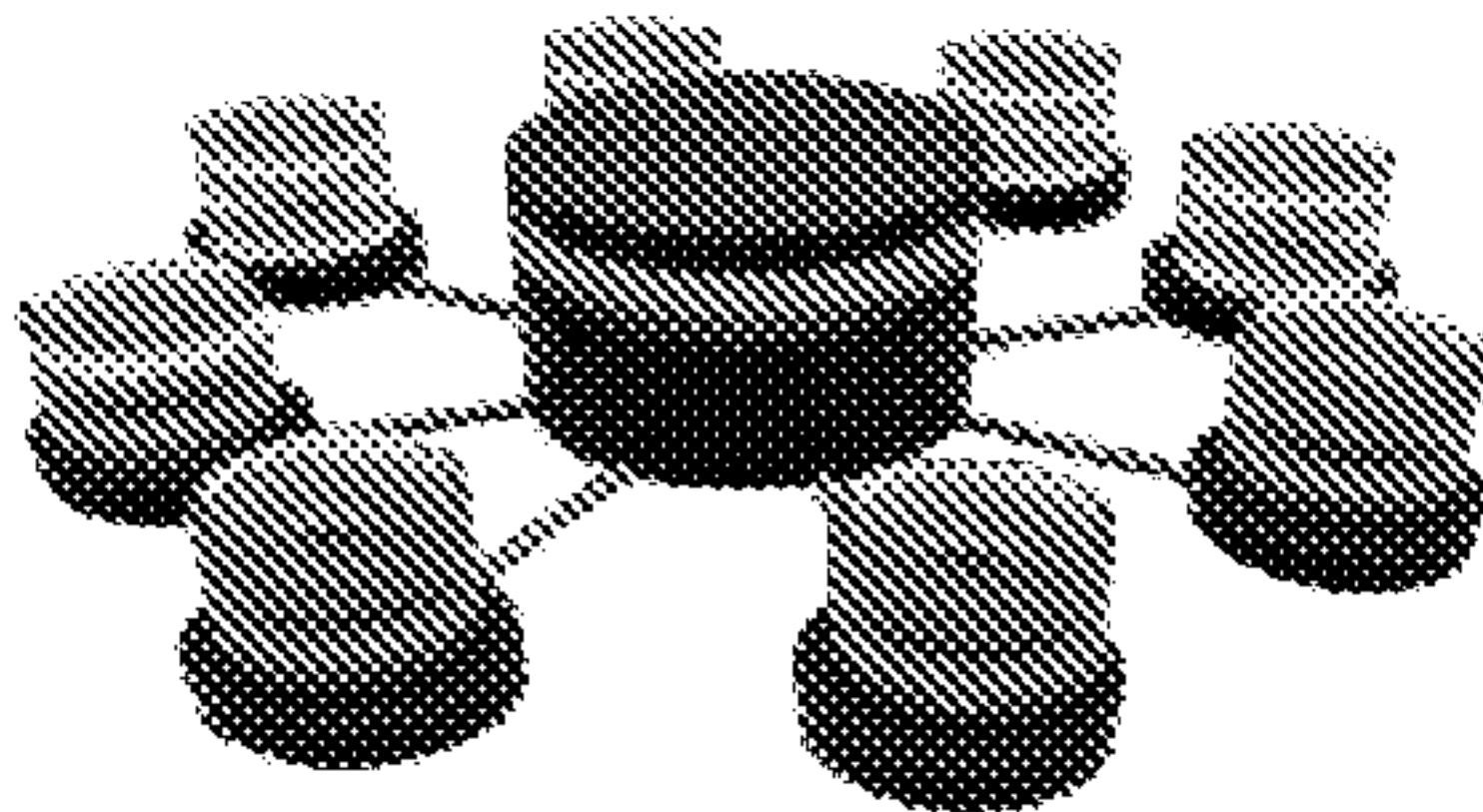
Book Hotel Room.

Get Customer Information. Select Room Type Super Deluxe. Get Room Stay Information. Book Hotel Room.

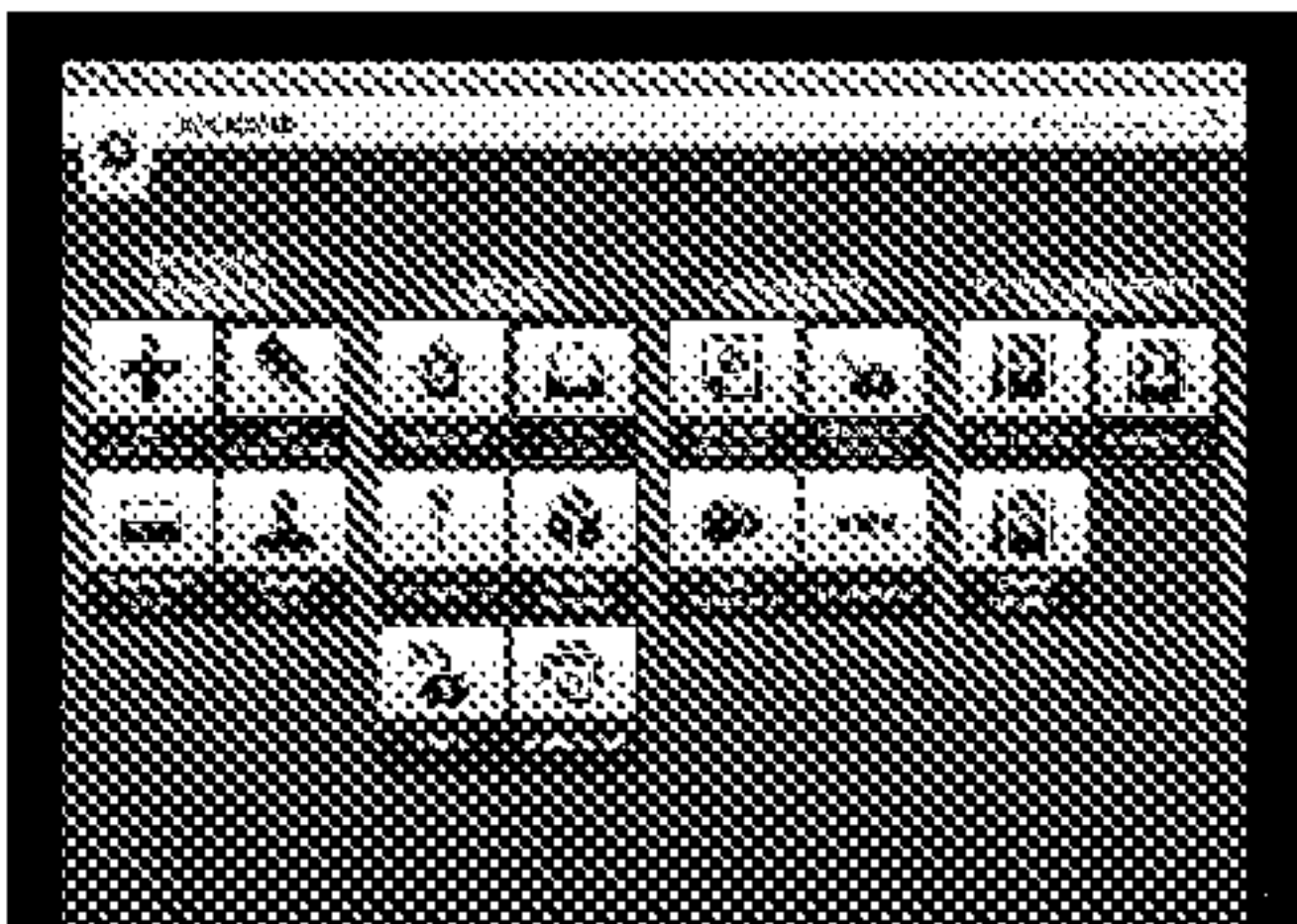
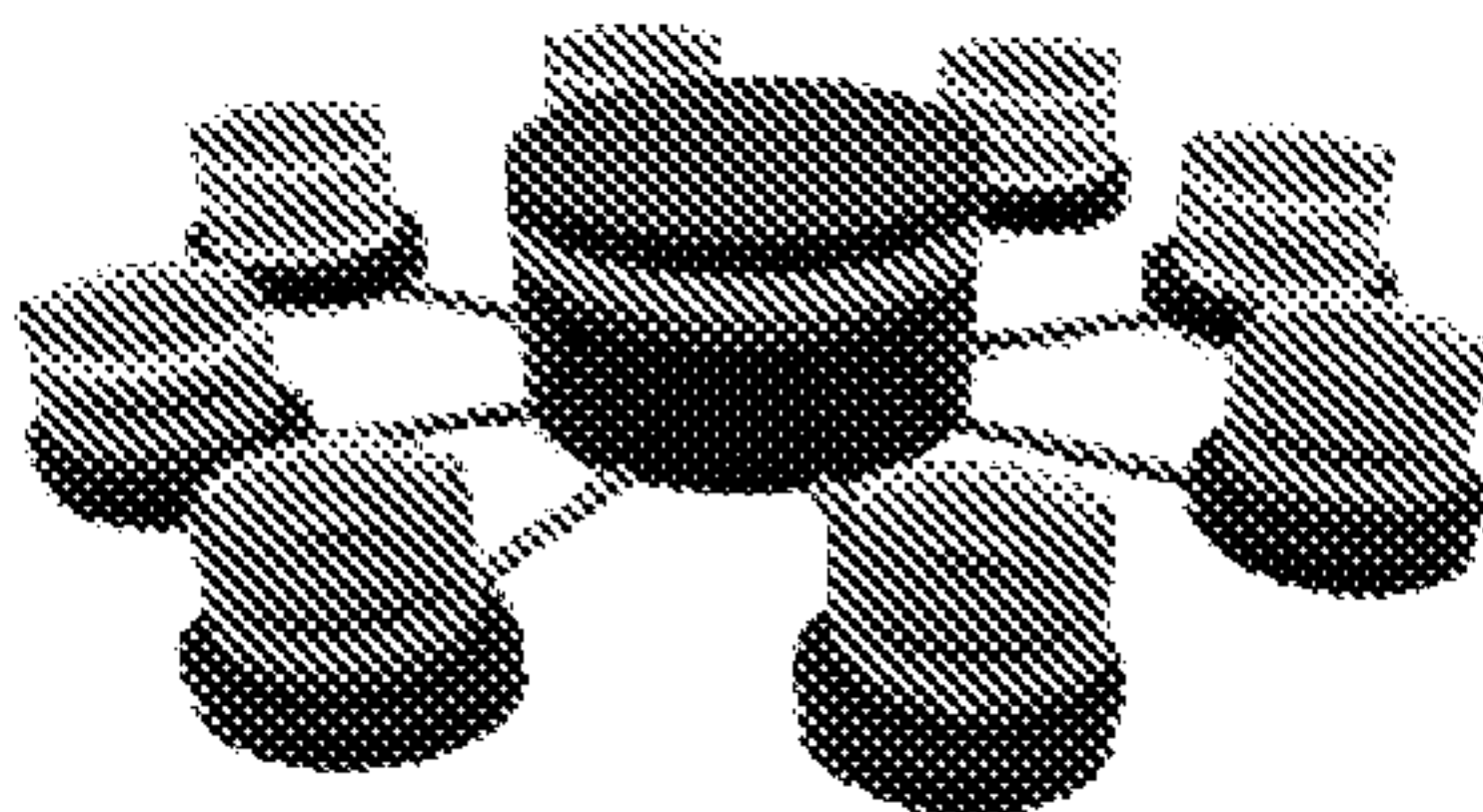
Entity	<u>Booking Confirmation</u>			
Primary Attributes	<u>ID No:</u> <u>0sdx2</u>	<u>Booking name:</u> <u>Sridhar</u>	<u>Fare:</u> <u>10,000</u>	<u>Cost of transportation:</u> <u>0</u>

Booking confirmation with booking number, customer name and fare.

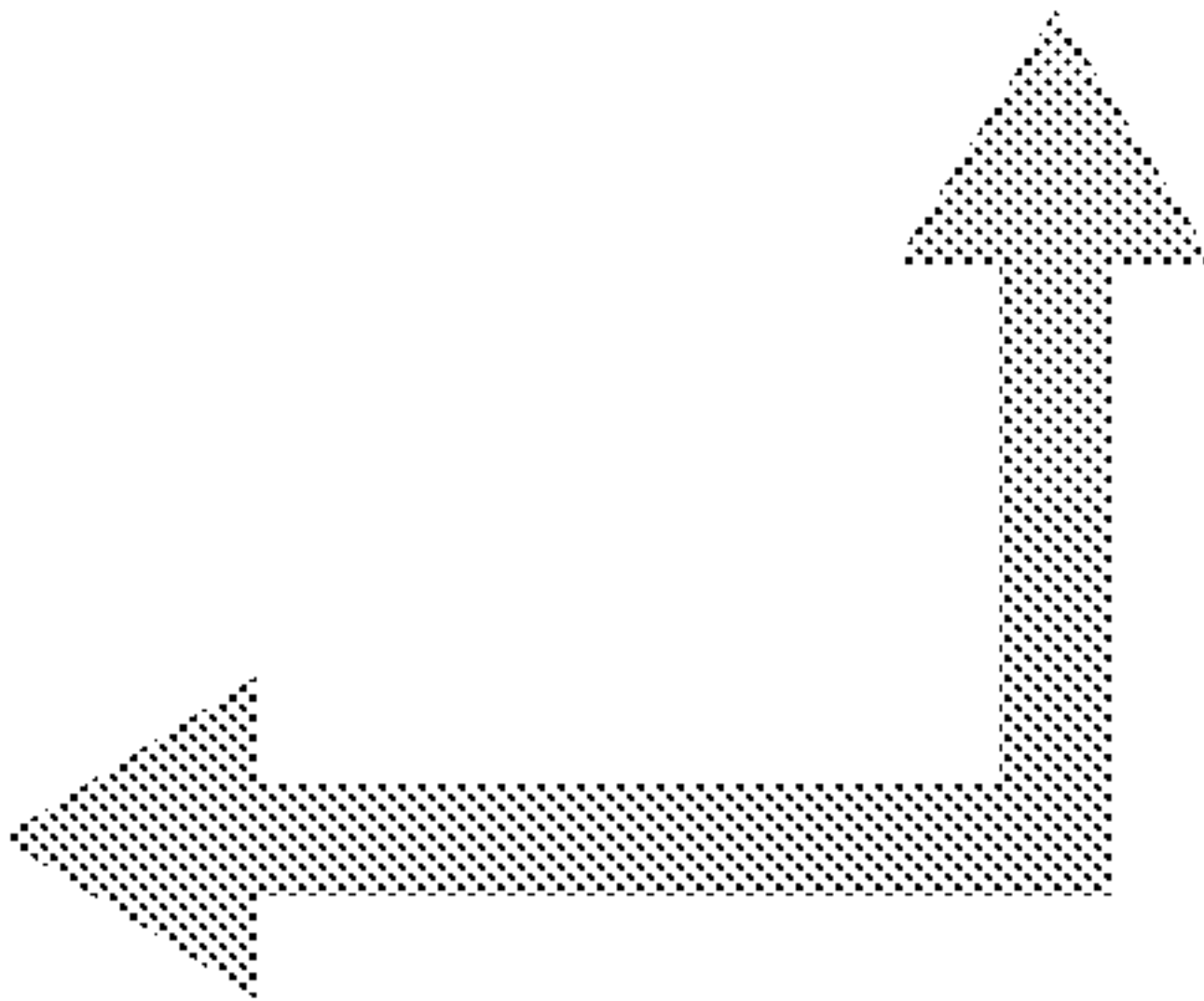
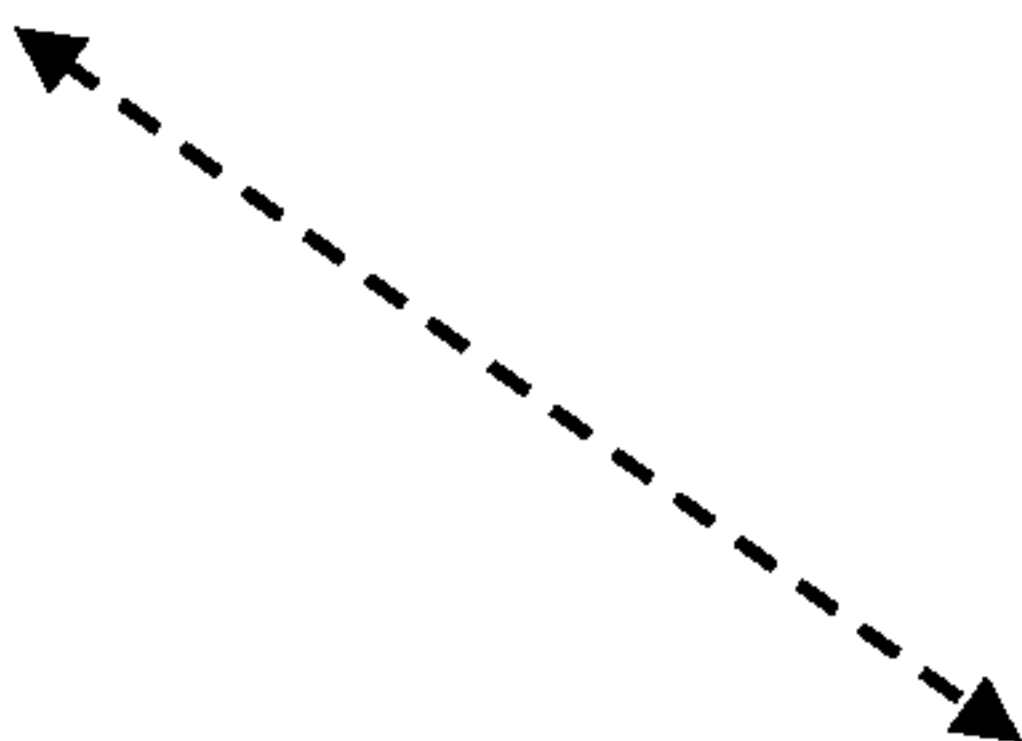
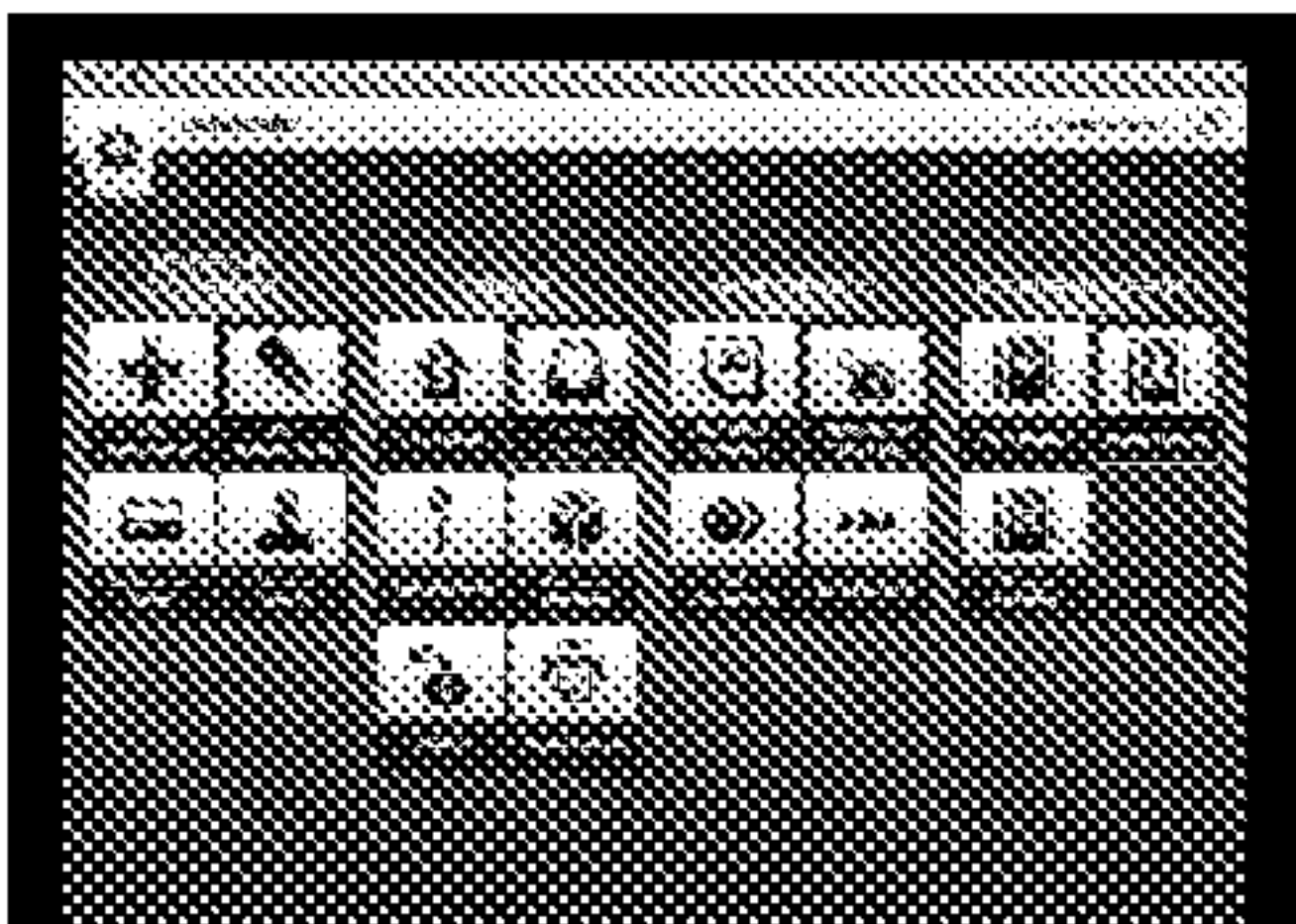
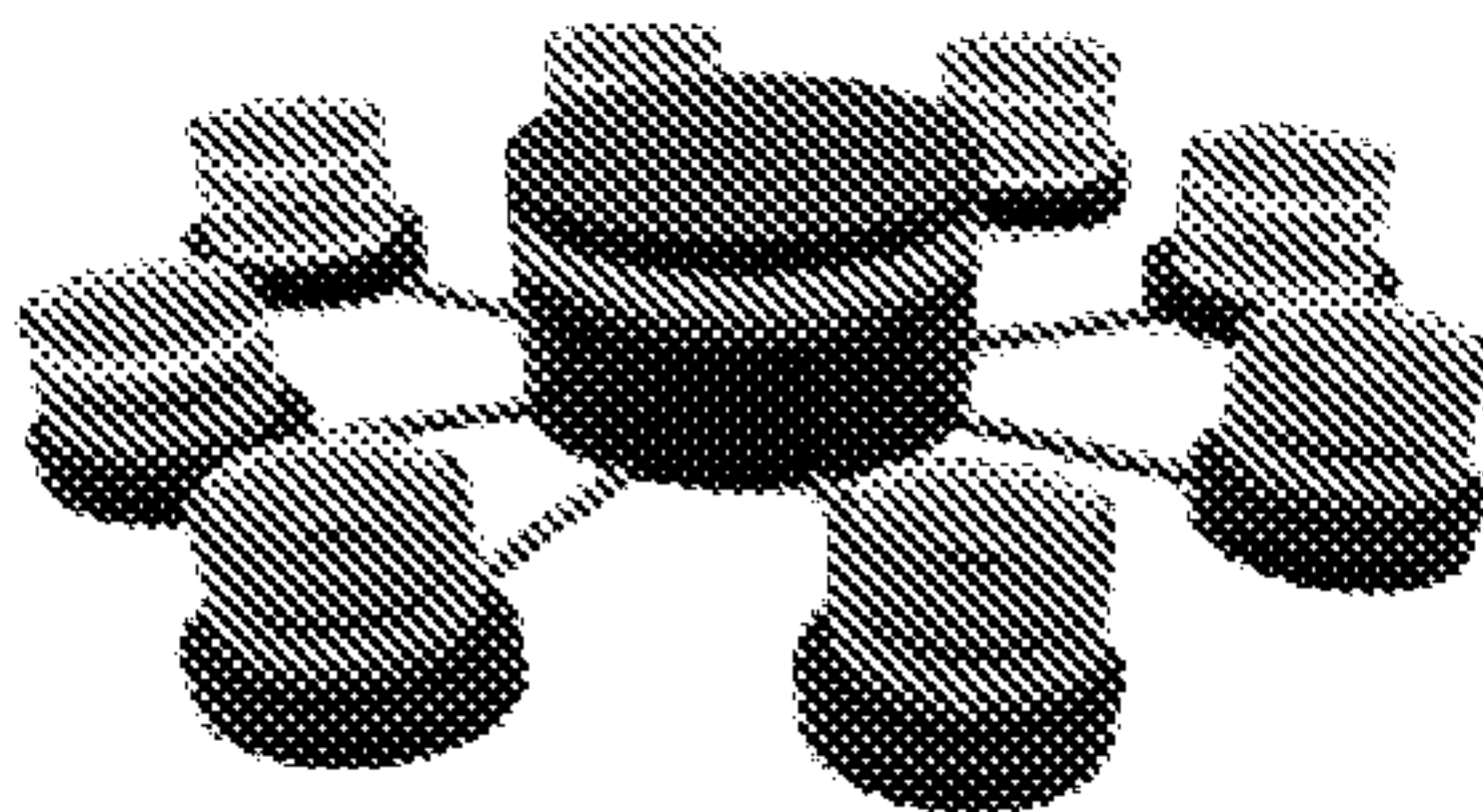
Drawing 100A



Drawing 100B



Drawing 100C

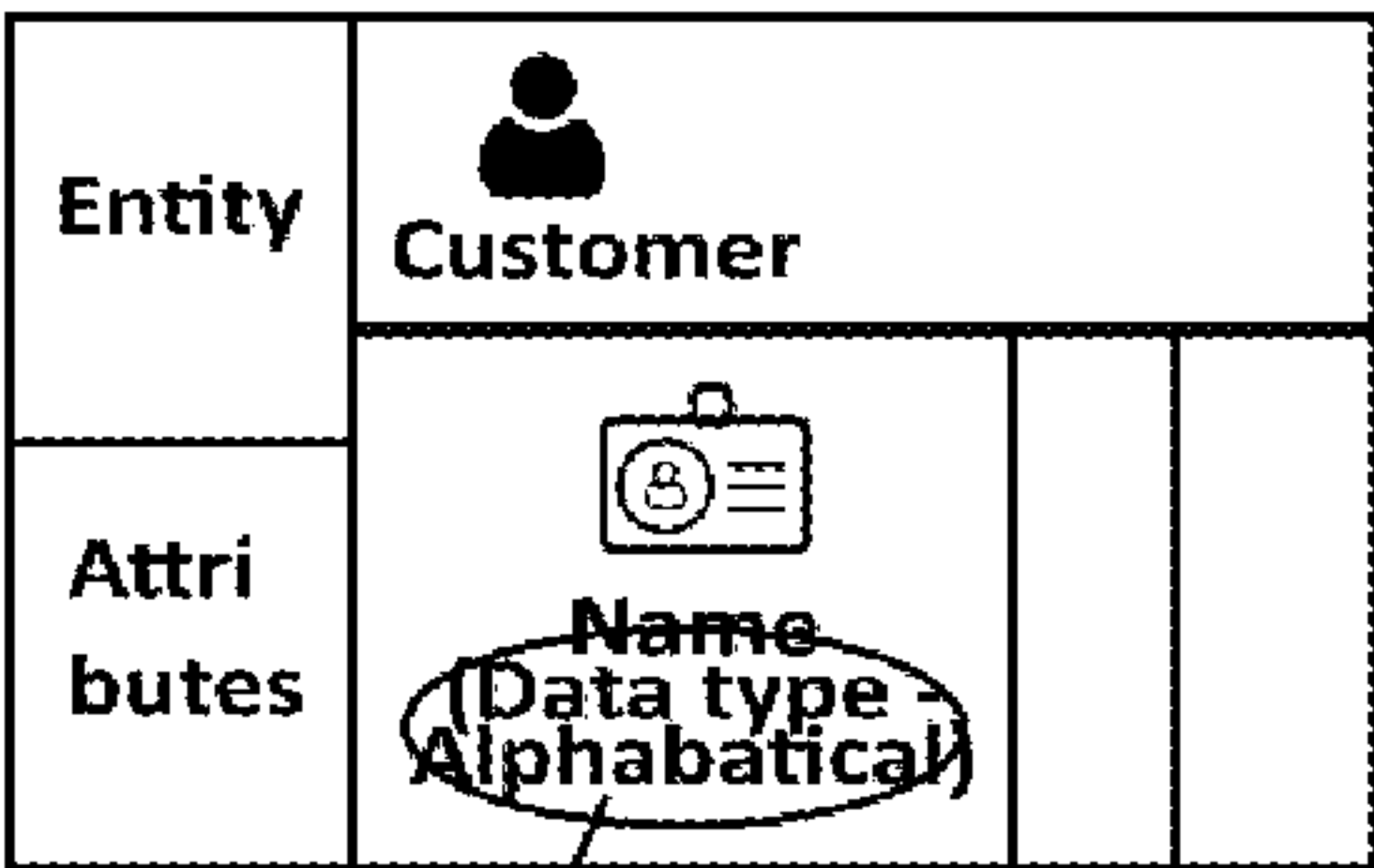


Drawing 101

Particulars	Java Program	NSL
Professionals required for developing the application	Java Programmer, UI/UX developer, Data base programmer	Any literate person
Application development time	160 hours approximately	2 hours
Solution training time	120 hours approximately	4 hours
Cost to build the application	USD 16,000 approximately	USD 200 approximately
Solution logic	Under the hood	Above the hood

DRAWING 102A:

```
55 System.out.println("What do you want to do?");
56 System.out.println("Book a room(b)");
57 Scanner in = new Scanner(System.in);
58     ch = in.next(".").charAt(0);
```

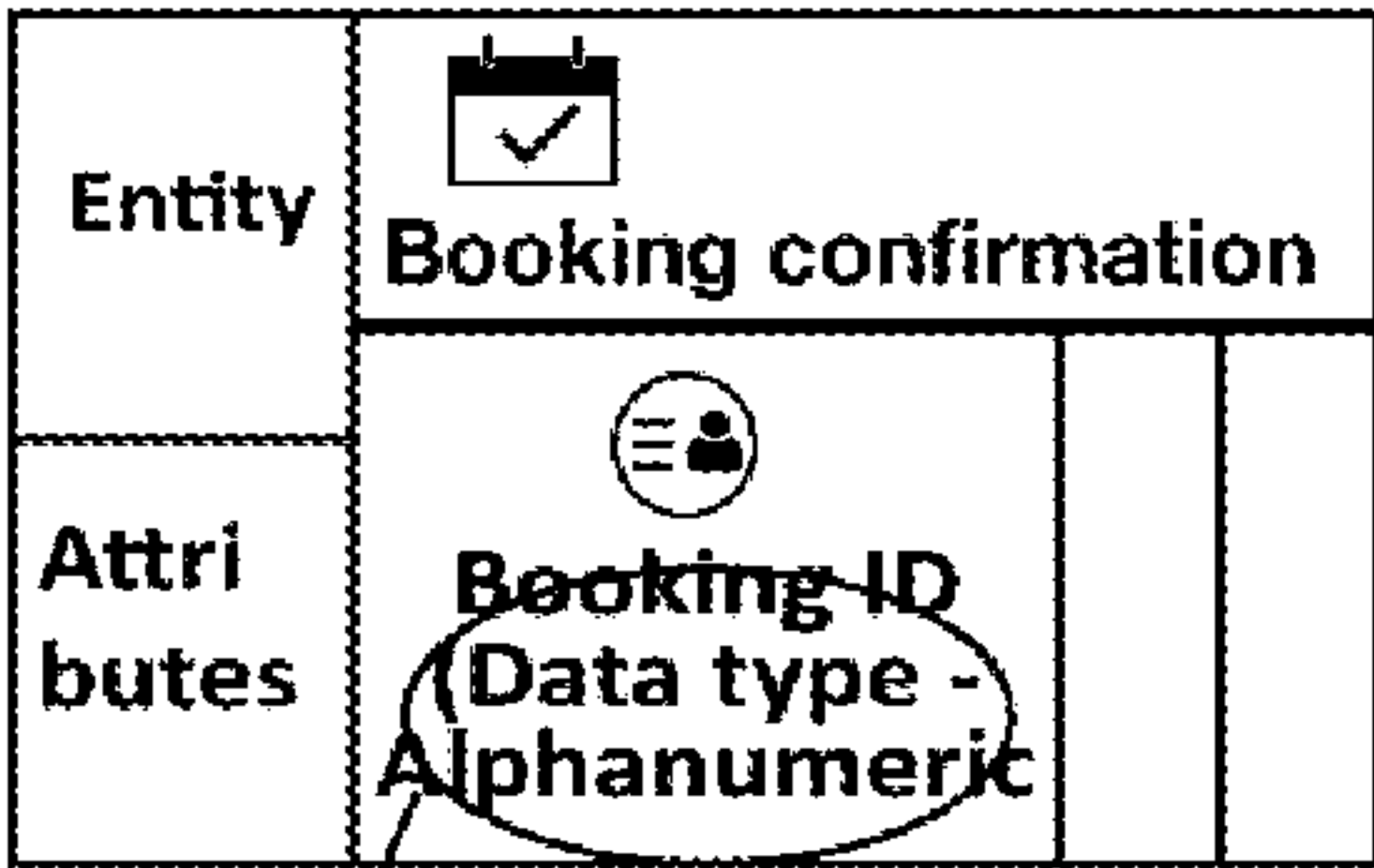


In Java, this is restricting the value of the entity to a single alphabet only.

In NSL, Char is restricted by the data type. It is attached as an attribute to an entity. If we restrict the length of the alphabet to "1", it is the NSL Equivalent of Char.

DRAWING 102B:

```
5 public class Customer {
6     int no;
7     String name;
8     String boookingno;
9     int type,id,d;
10    boolean status;
```

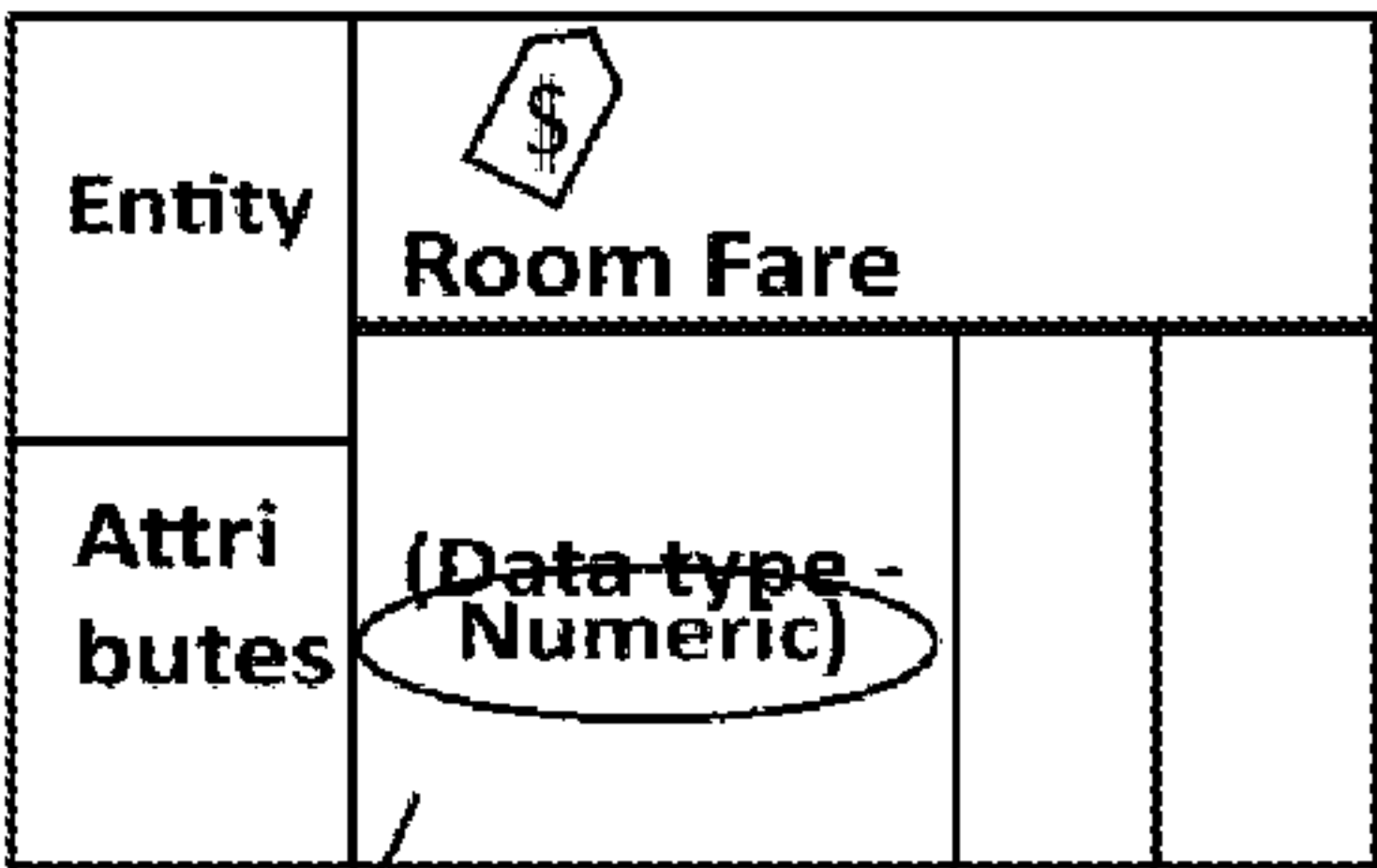


In Java, String is a data type that can be either numeric or alphabetic.

In NSL, String is represented by data type. Attaching the data type "alphanumeric" represents "string"

DRAWING 102C:

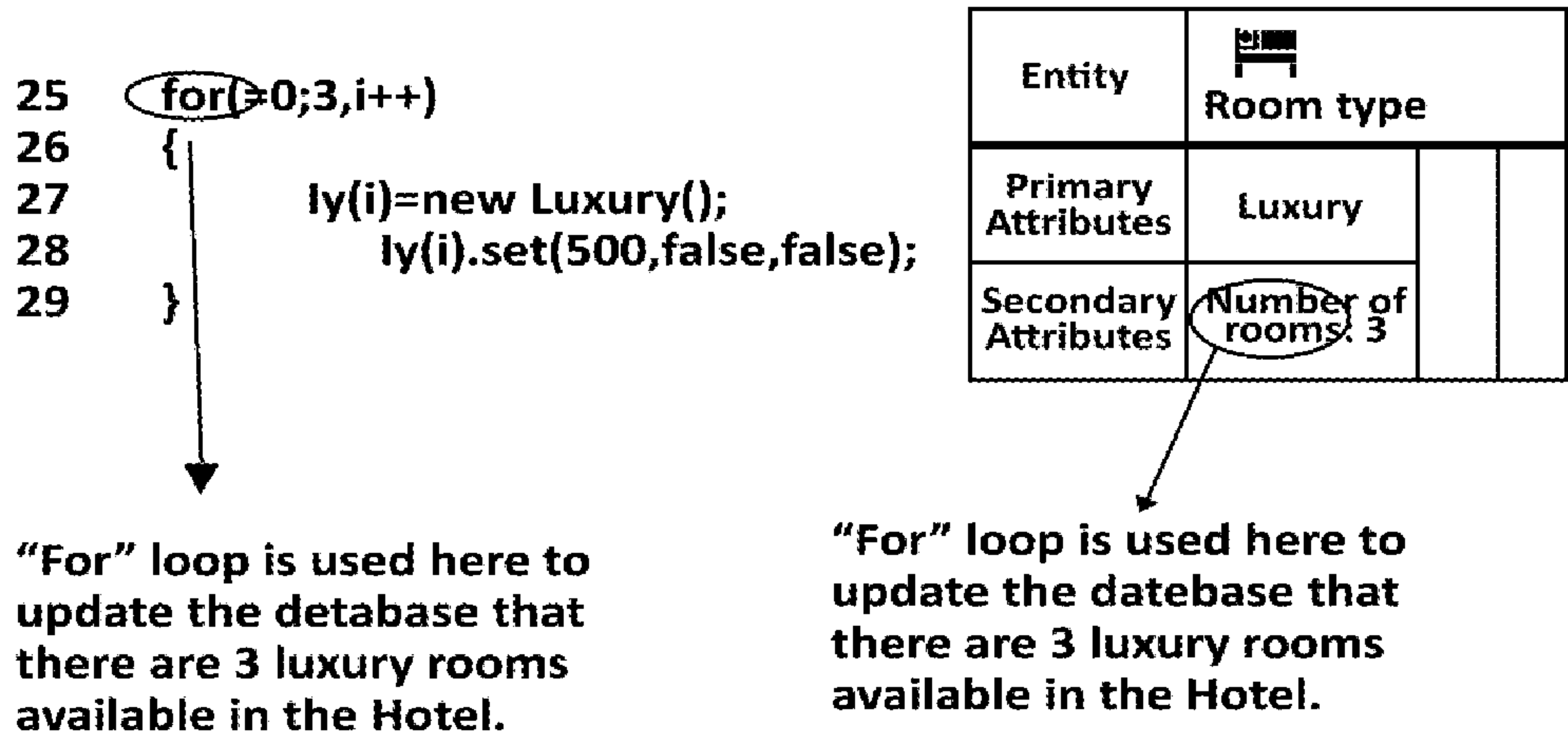
```
15 char ch='y';
16 double ff;
17 String name;
```



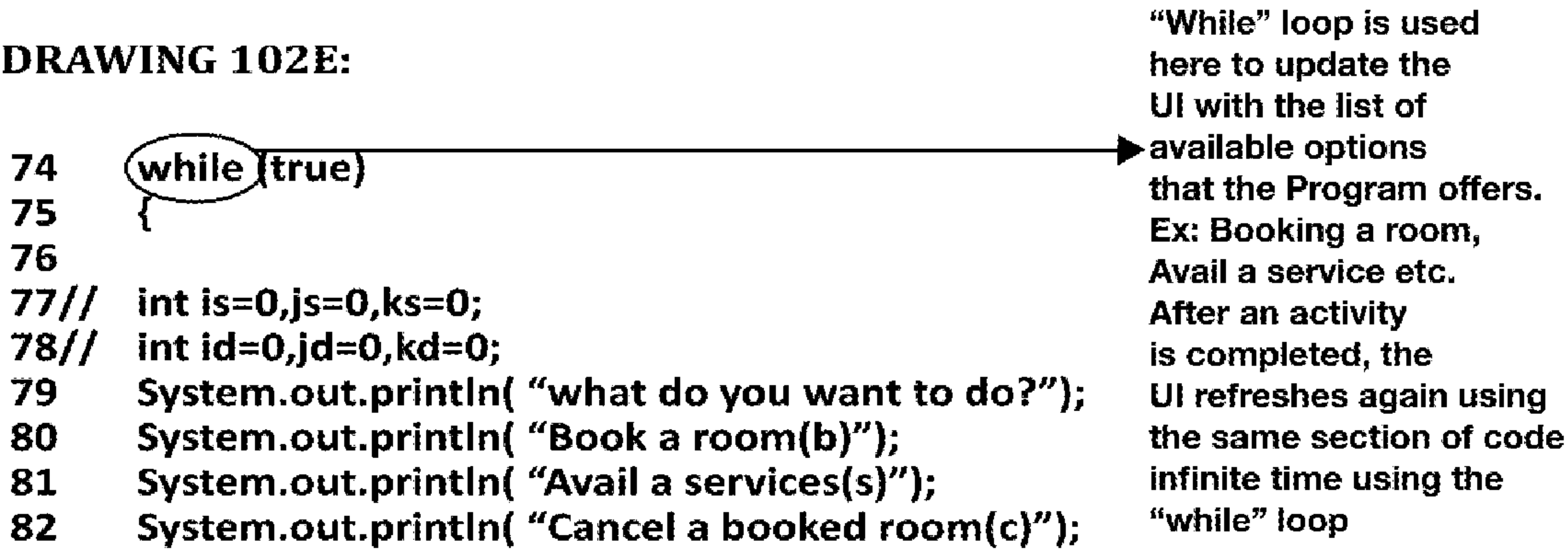
Double is used as data type allowing decimals for the entity "Fare calculator"

For the entity Room Fare, the data type is attached as an attribute, set to Numeric allowing decimals.

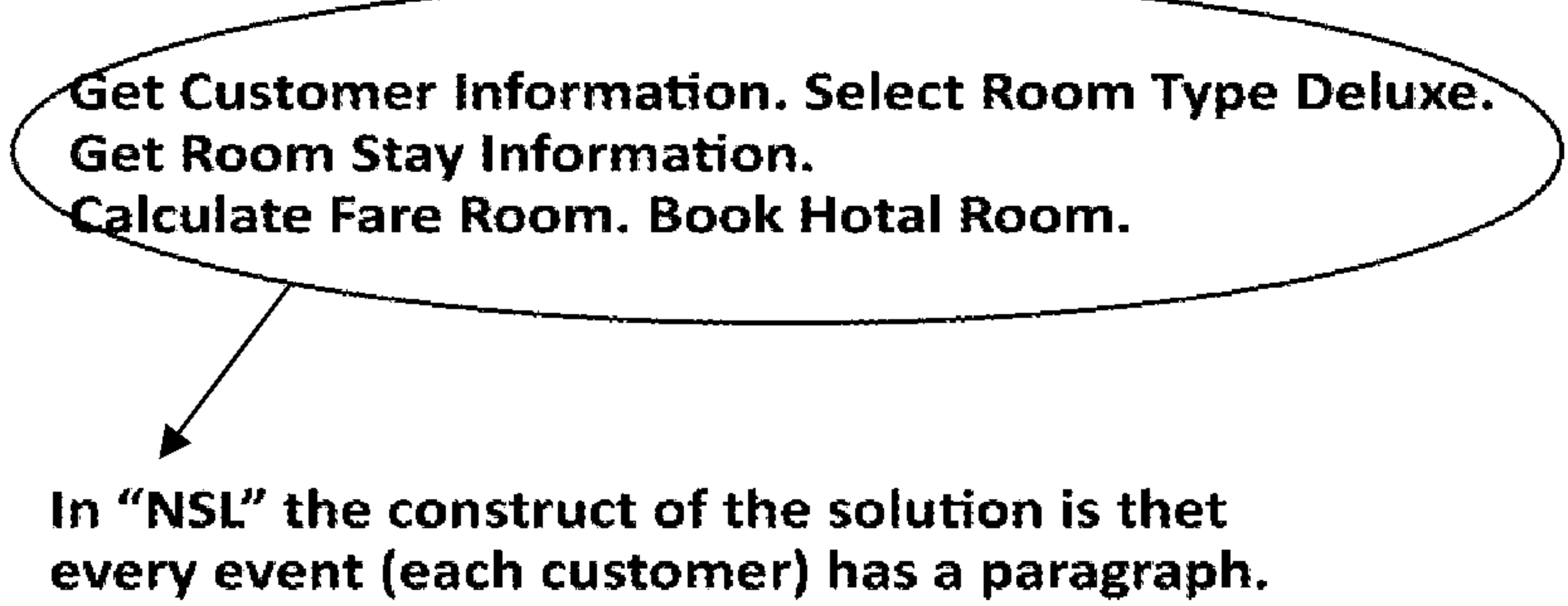
DRAWING 102D:



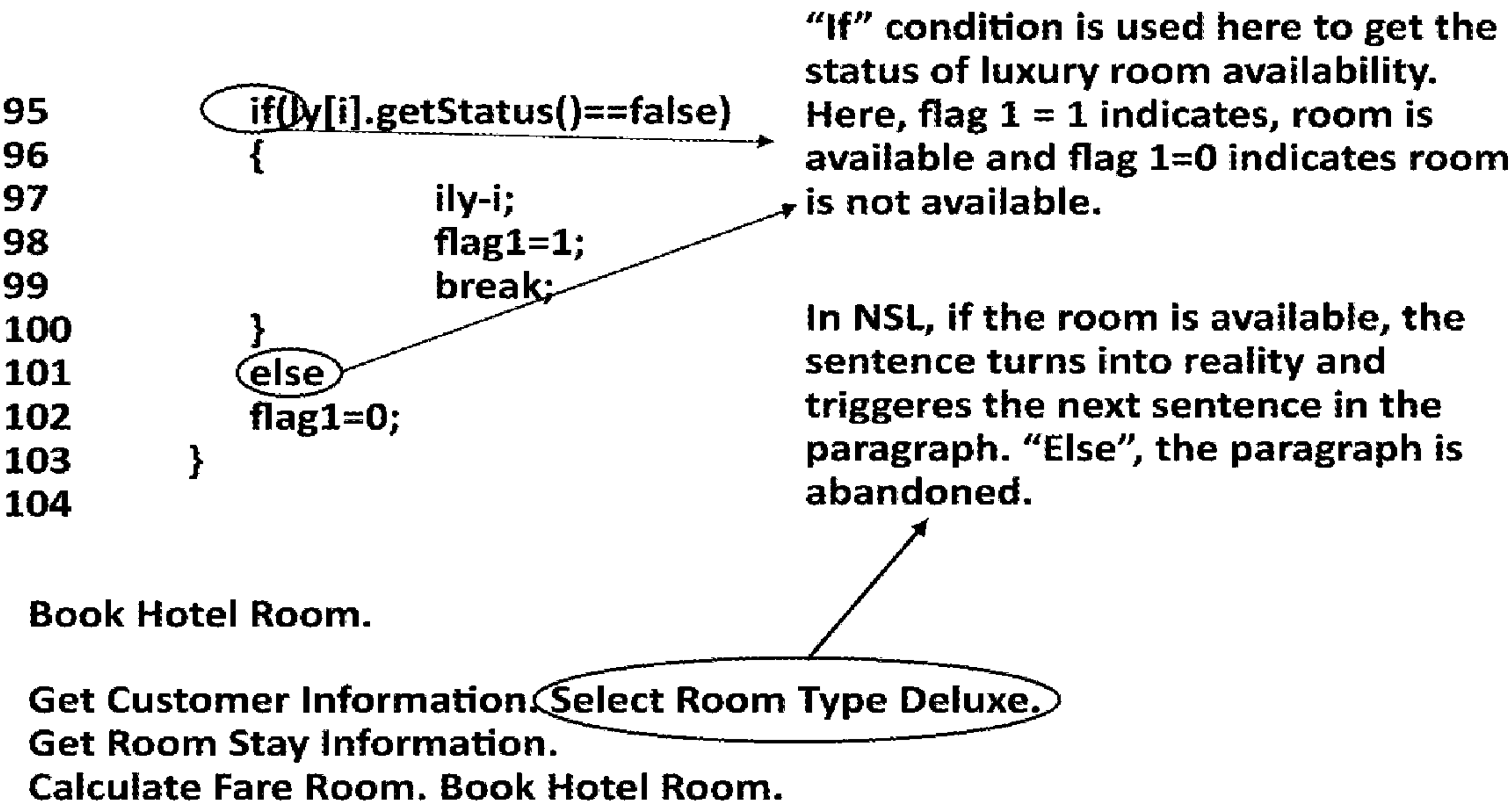
DRAWING 102E:



Book Hotel Room.



DRAWING 102F:



DRAWING 102G:

```
95      if(ly[i].getStatus()==false)
96      {
97          ily-i;
98          flag1=1;
99          break;
100     }
101     else
102     flag1=0;
103 }
104
```

“Break” here is used in Java to terminate the “else” loop if the “if” condition is true.

Book Hotel Room.

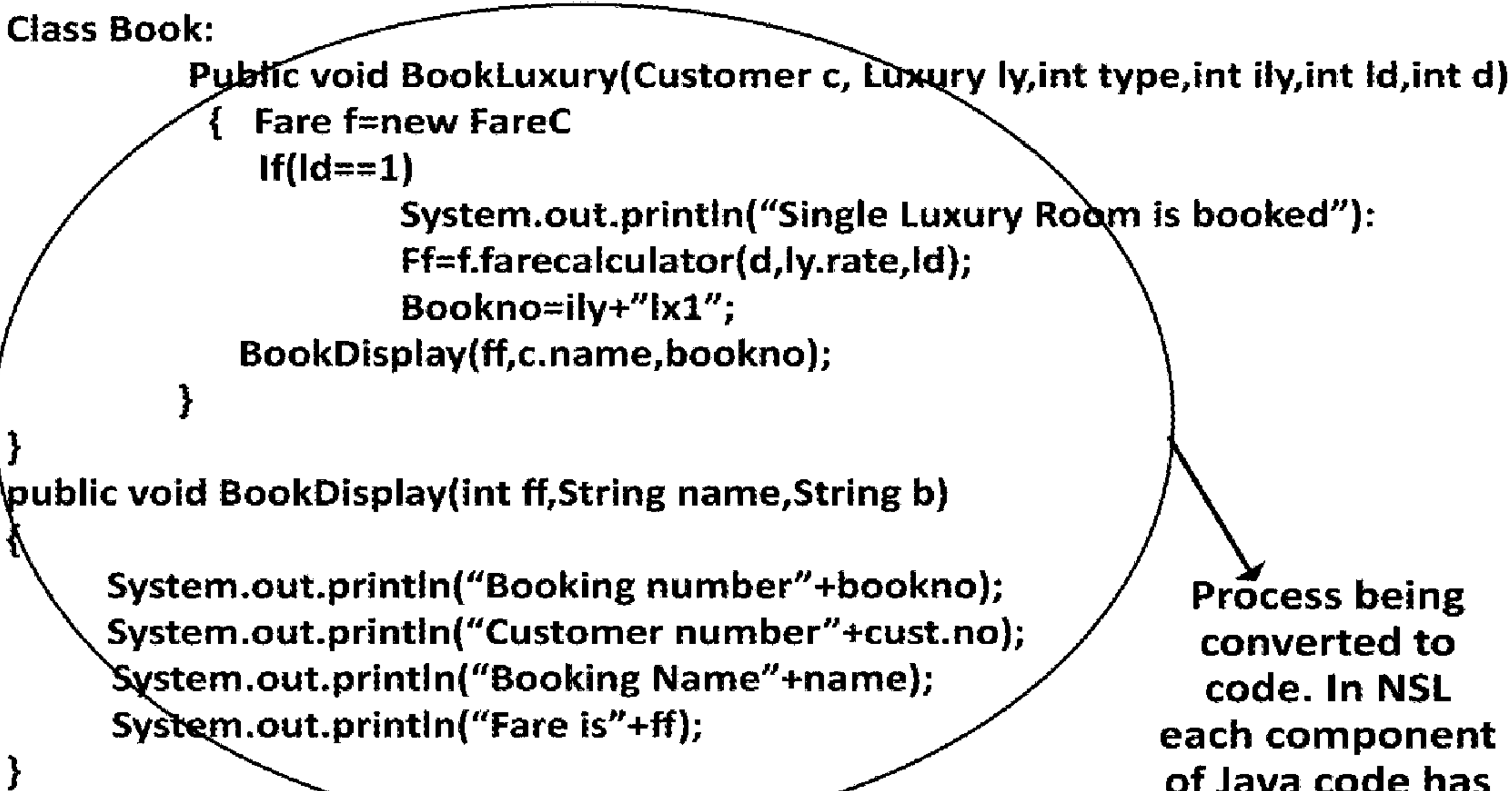
Get Customer Information. **Select Room Type Super Deluxe.**
Get Room Stay Information.
Calculate Fare Room. Book Hotel Room.

Book Hotel Room.

Get Customer Information. **Select Room Type Luxury.**
Get Room Stay Information. Calculate Fare Room.
Book Hotel Room.

In NSL, if the room type is selected as “Super Deluxe”, the paragraph for “Luxury room” does not get triggered.

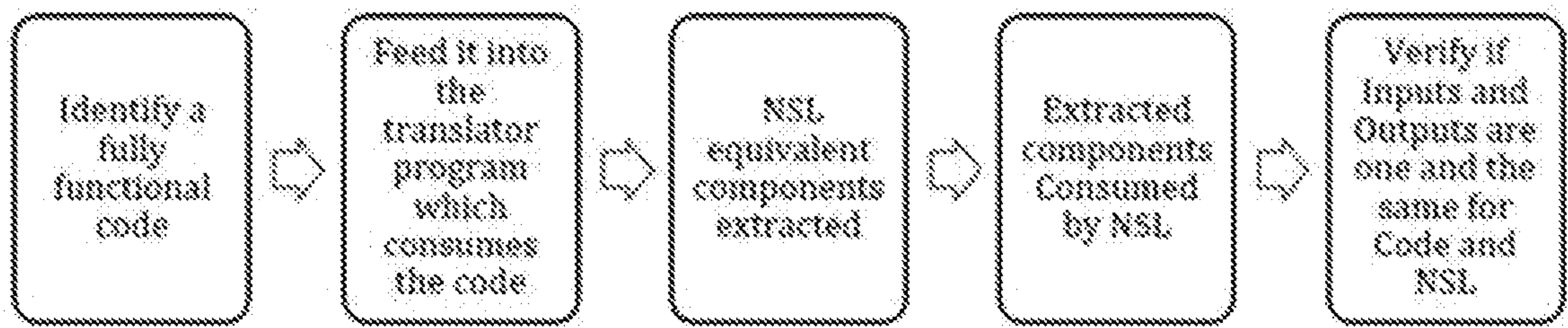
DRAWING 103:



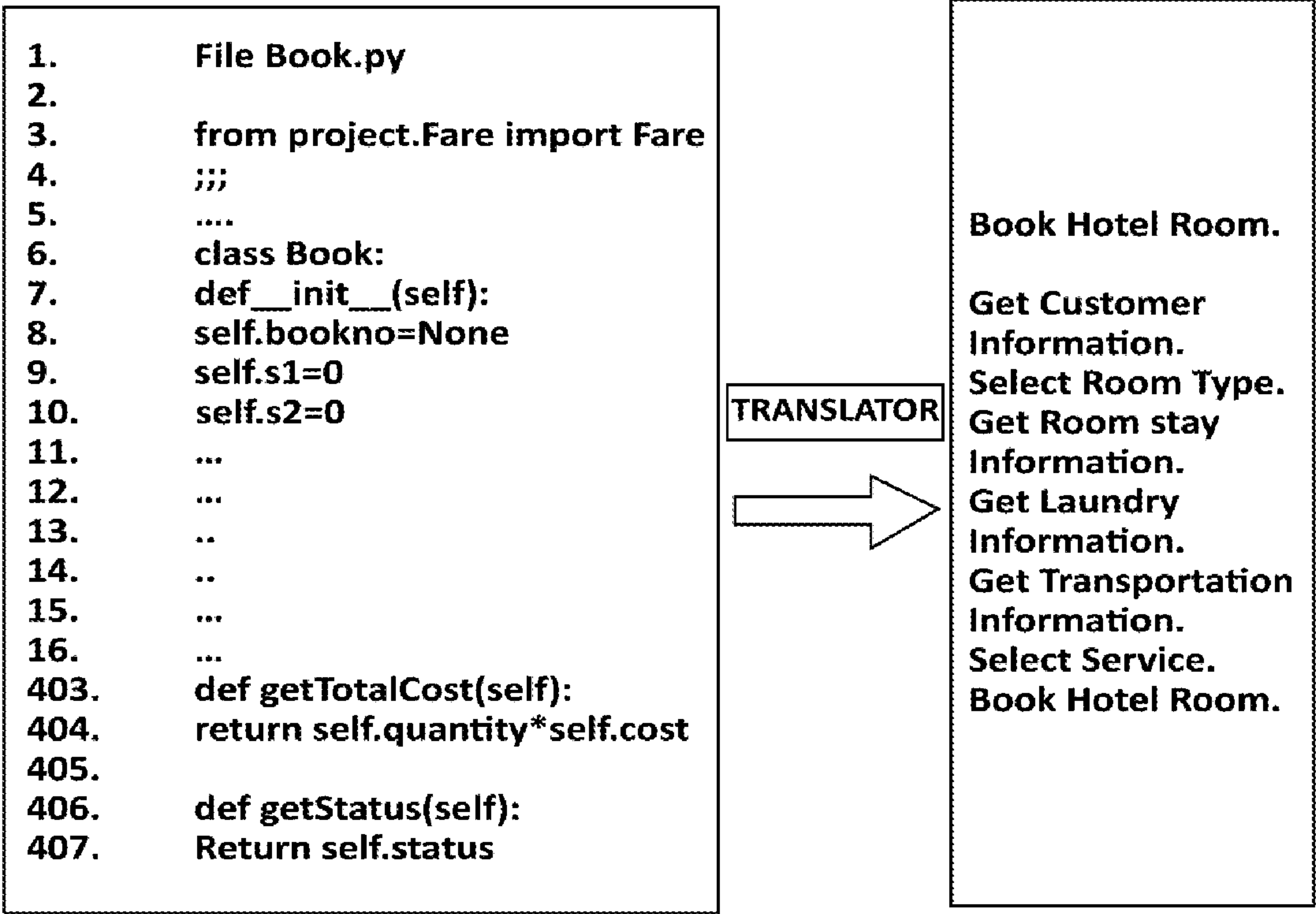
NSL representation
LSI: Book Luxury Room.

Entities	Room	Room	Customer	Room	Room Fare
Attibutes	Type: Rate:	Occupancy	No.of days:	Booking id:	Formula: days*occupancy* rate

Drawing 104



DRAWING 105:



Drawing 106

The below table shows a few Java constructs and their corresponding NSL equivalents:

#	JAVA CONSTRUCT	NSL EQUIVALENT
1	Java Classes and class constructs	Independent Entity
2	Class Members	Attribute
3	Java method invocation, execution	Drivers (CD, CES)
4	Development / Runtime	Potentiality/Reality
5	Java Constants	Constants
6	Java Variables at various scope levels	Variables
7	Java Numerals and numeric constants	Number
8	Java Keywords, Operators	Words, Operators
9	Simple java programming block	Basic CU
10	Program block execution	Transaction CU
11	Looping Constructs	Recursive CU
12	Conditional Constructs (if, switch, etc..)	Alternative CU
13	Java Tokens	ID Assignment
14	Java full lifecycle ID	Contextual ID
15	Frameworks, libraries and JDK	Block Boxes

Drawing 107A

```
public static volatile Scanner scanner;  
ArrayList<Teacher> teachers;  
ArrayList<Subjects> subjects;  
ArrayList<Student> students;  
ArrayList<Mark> marks;
```

Here, "Static" key word is used to freeze the entities such as teacher, Subject, Student, Marks and its values for further reuse.

Entities	Student		
Primary Attributes	Age	Gender	ID NO

In NSL, it is "constant".

Drawing 107B

```
switch (options)  
{  
    case 1:  
        Subjects subject = new Subjects();  
        subject.addSubject();  
        school.subjects.add(subject);  
        break;  
    case 2:  
        Teacher teacher = new Teacher();  
        teacher.addTeacher();  
        school.teachers.add(teacher);  
        break;  
    case 3:  
        school.assignSubjectToTeacher();  
        break;  
}
```

Case and Switch are used at UI level to move from one object to the other as the values are getting filled.

Get Subject Details. Get Teacher Info. Assign Subject to Teacher

In NSL, the second change unit is triggered only when the 1st Change unit is completed.

DRAWING 107C:

Case8:

```
school.calculatePercentageOfStudent();
break;
```

Case 9:

```
school.junkFunction();
break;
```

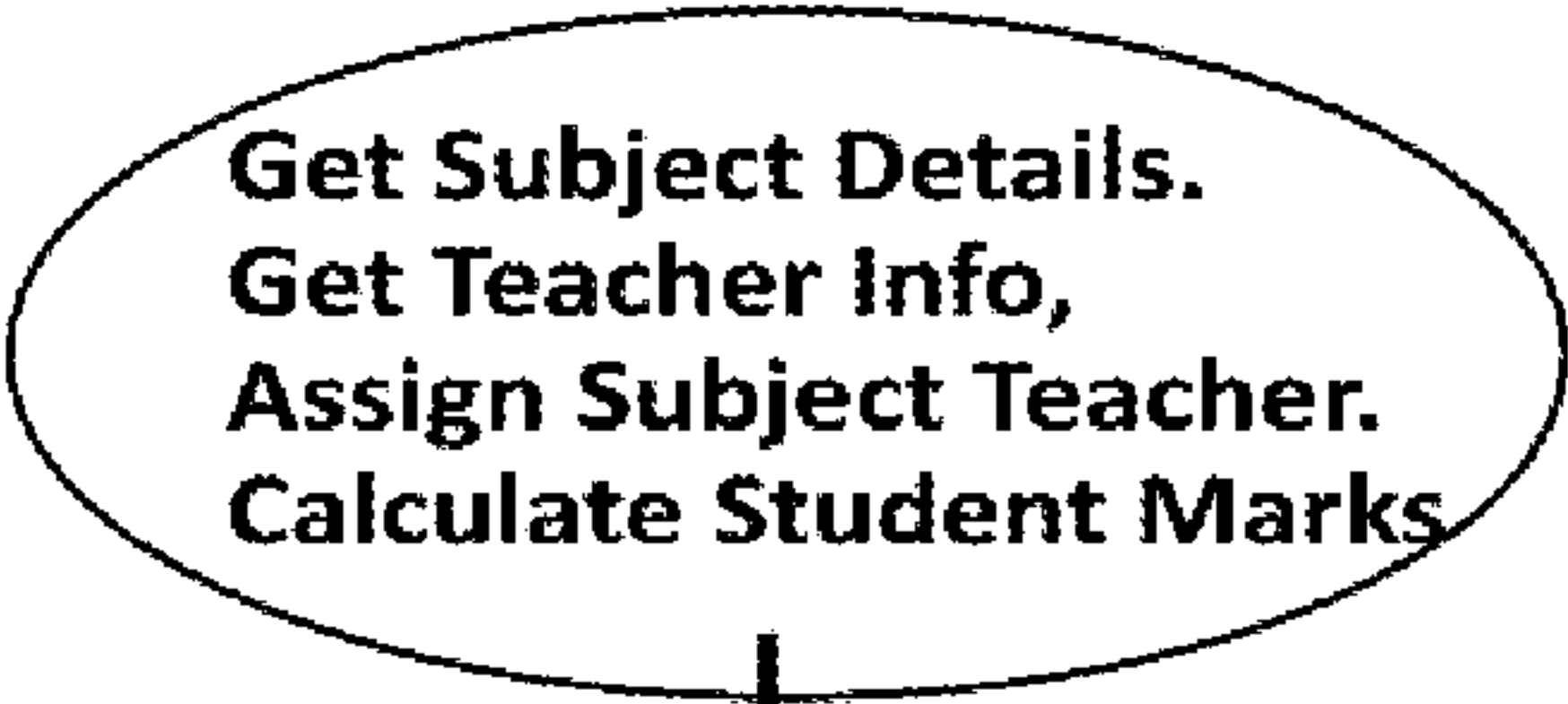
Case 10:

```
System.out.println("Exiting...\n Have a nice day.");

System.exit(0);
break;
```

default:

```
System.out.print("Invalid option!!!\n");
continue;
}
}
```



In NSL, the construct of the solution is such that every event (each customer) has a paragraph.

“Continue “is used here to update the UI with the list of available options that the program offers. Ex: Entering marks of students for each subject. After an activity is completed, the UI refreshes again using the same section of code for entering marks of next student.

Drawing 107D

```
class PersonalInfo
{
    String name;
    String mobile;
    String email;
    transient short age;
    Address address;
}
```

“Transient” Key word is used in cases where attributes can be selectively called for differentiation

In NSL, this is used at a “UI” level where the UI is attached as an attribute to an entity and can be switched on/off.

Drawing 107E

```
Class Subjects
{
    String name;
    int pass_mark;
    public long max_mark;
    int std;
    public void addSubject()
    {
```

The long keyword is used to declare a variable as a numeric type. A long value can hold a 64-bit integer number which ranges from -2⁶³ to 2⁶³ - 1.

Entities	Subject		
Primary Attribute	Name	Pass_Mark	Max_Mark
Secondary Attribute		Datatype: Integer Size = 50	

In NSL, a large numerical value of an attribute is captured by defining the size during Entity Creation.

Drawing 107F

```
class PersonalInfo
{
    String name;
    String mobile;
    String email;
    transient short age;
    Address address;
}
```

The **short** keyword is used to declare a variable as a numeric type. A **short** value can hold a 16-bit integer number

Entities	Student		
Primary Attribute	Name	Mobile	Age
Secondary Attribute			Datatype: Integer Size = 10

In NSL, a large numerical value of an attribute can be captured by defining the size during Entity Creation.

Drawing 107G

```
public void junkFunction()
{
    byte b;
    int i=10;
    do
    {
        i=i-5;
    }while(i>=0);
}
```

Is a data type. It will allocate one byte per variable.

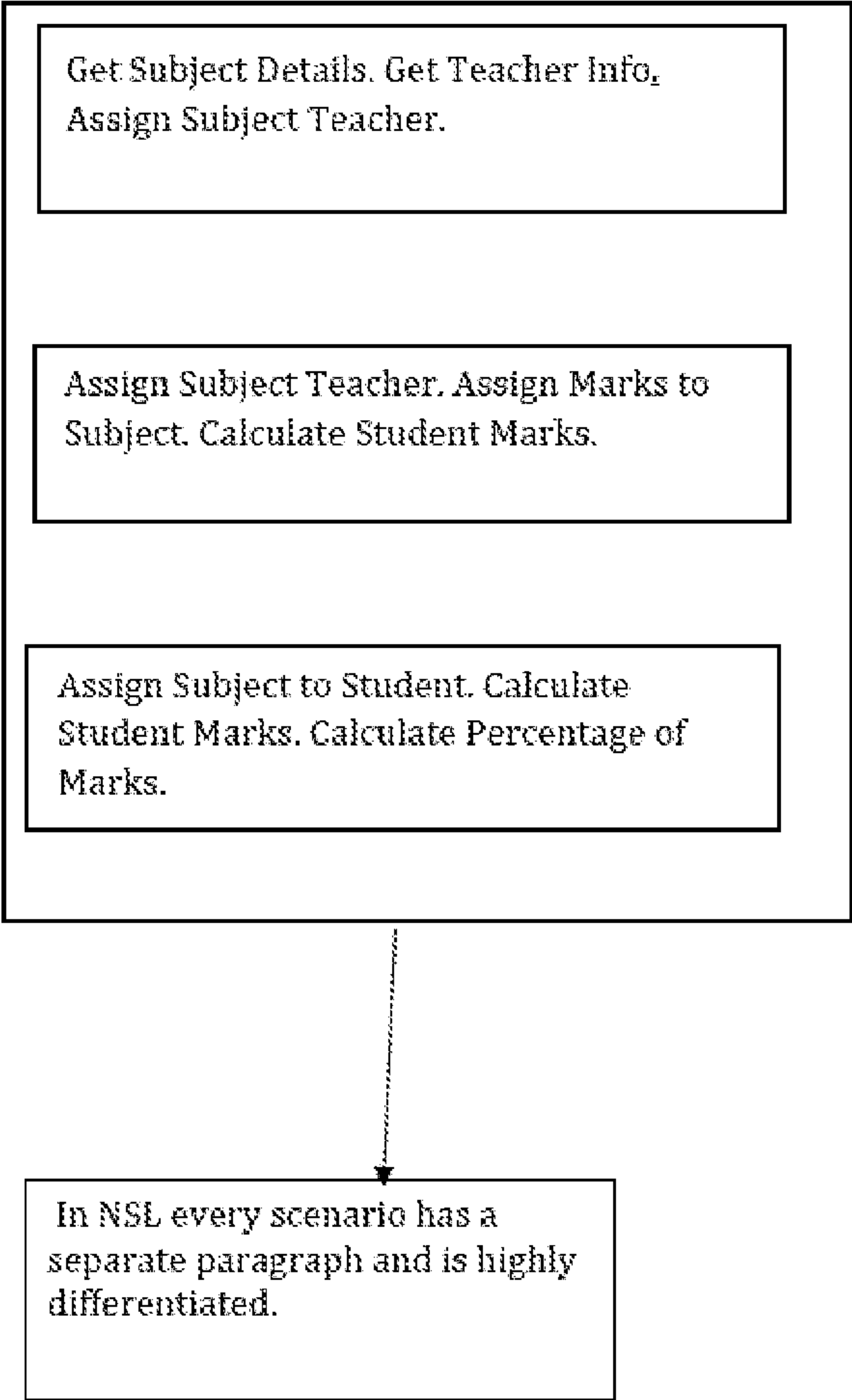
Entity	Student
Primary Attribute	Name
Secondary Attribute	Datatype: Integer Size = 10

In NSL, a numerical value of an attribute is captured by defining the size during Entity Creation.

Drawing 107H

```
While(true)
{
    school.showOptions();
    option = scanner.nextInt()
    switch (option)
    {
    case 2:
        Teacher teacher = new Teacher();
        teacher.addTeacher();
        school.teachers.add(teacher);
        break;
    default:
        System.out.print("Invalid option!!!\n");
        continue;
    }
}
```

Is used when a switch case is defined and when a user inputs a value that is not listed in any of the cases inside the switch, the program will execute the default segment.



Drawing 107I

```

}

class Student extends PersonalInfo
{
    public Student()
    {
        super();
    }
}

```

"Super" Key word is used call the properties of parent entity (Student Personal Info)

Entity	Attributes					
Student 1	Roll No	Name	Mobile	Address	Age	Email
Student 2	Roll No	Name	Mobile	Address	Age	Email

Drag all the attributes without values. Used for easy replication of standard attributes.

Drawing 107J

```

class Mark
{
    Subjects subject;
    Student student;
    private float mark_scored;

    protected void assignMark(Student student, Subjects subject, float mark_scored)
    {
        this.student = student;
        this.subject = subject;
        this.mark_scored = mark_scored;
    }

    protected float getMarkScored()
    {
        return mark_scored;
    }
}

```

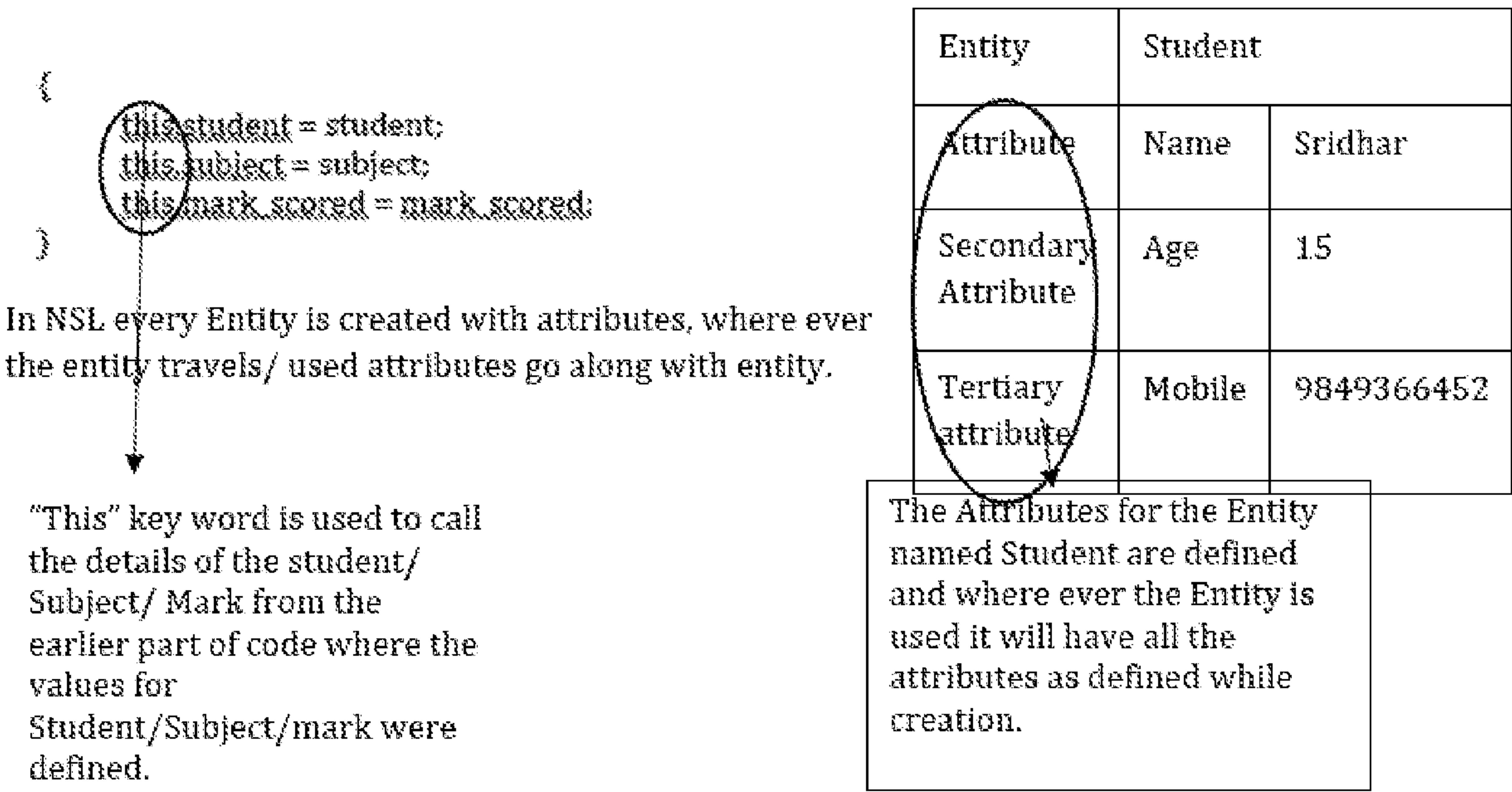
"Protected" Key word here is used to freeze the pass marks entered of a student in a subject by a teacher having the decision rights and where ever the student details are called in the program, for that subject of the student the same value would go.

Entity	English Subject	
Attribute	Pass Mark	45

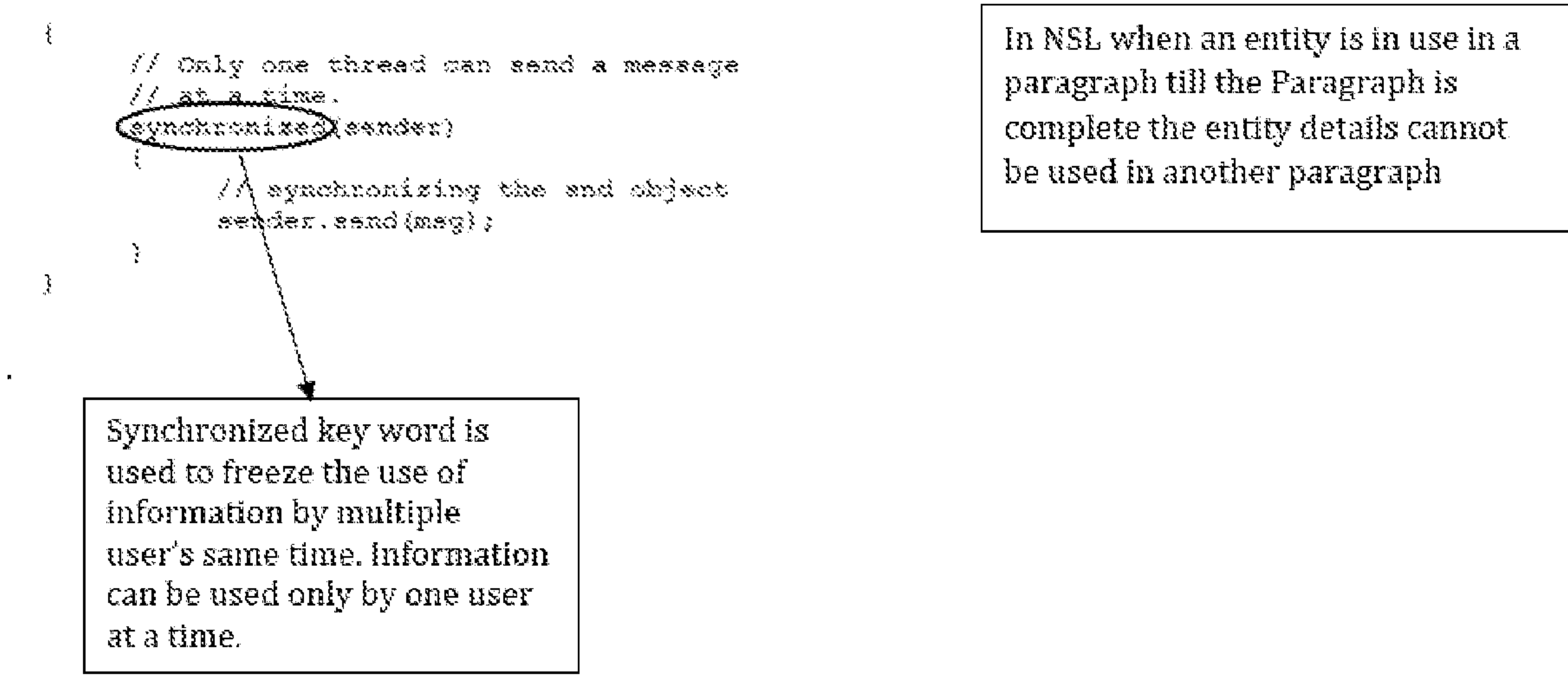
Here the Pass mark value is frozen for the entity named English through Decision Making Rights and passed on as information to the next level.

In NSL there are decision rights at different vantage levels and based on the decision right the value of an attribute is frozen for an entity.

Drawing 107K



Drawing 107L



DRAWING 107M:

```
}

public strictfp float calculatePercentageOfStudent ()
{
    System.out.println(" Enter role number of the student:");
    int role_number = scanner.nextInt();
    Student student = getStudent(role_number);
    float total = 0;
    float subject_total = 0;
    for (int i=0;i<marks.size();i++)
    {
        if(marks.get(i).student==student)
        {
            total = total + marks.get(i).getMarkScored();
            subject_total = subject_total+
marks.get(i).subject.max_mark;
        }
    }
    float percentage = (total/subject_total) * 100;
    System.out.println("Percentage of
"+student.name+"=")+percentage);
    return percentage;
}
```

Entity	Total Marks Percentage	
Attributes	Name	Sridhar
Secondary Attribute	Marks Attained	955
Tertiary Attribute	Total Marks	1000
4 th Attribute	% of Marks	95.5%

The attribute % of marks is
frozen at one value after
decimal for the entity
percentage of marks.

Strict floating point(fp) key word is used in this program to standardize the accuracy of the value after a decimal. Ex: Percentage of marks should have only 1 number after a decimal. Ex: 95.5%

DRAWING 107N:

final int total_marks=100;

System.out.println("Enter role number of the student:");

Int role_number=scanner.nextInt();

Student student= getStudent(role_number);

float total= 0;

float subject_total= 0;

for(int i=0;i<marks.size();i++)

{

If(marks.get(i).student==student)

{

total=total+marks.get(i).getMarkScored();

subject_total=subject_total+marks.get(i).subject.max_mark;

Entity	Total Marks Percentage in Mathematics	
Attributes	Name	Sridhar
Secondary Attribute	Marks Attained	85
Tertiary Attribute	Total Marks	100

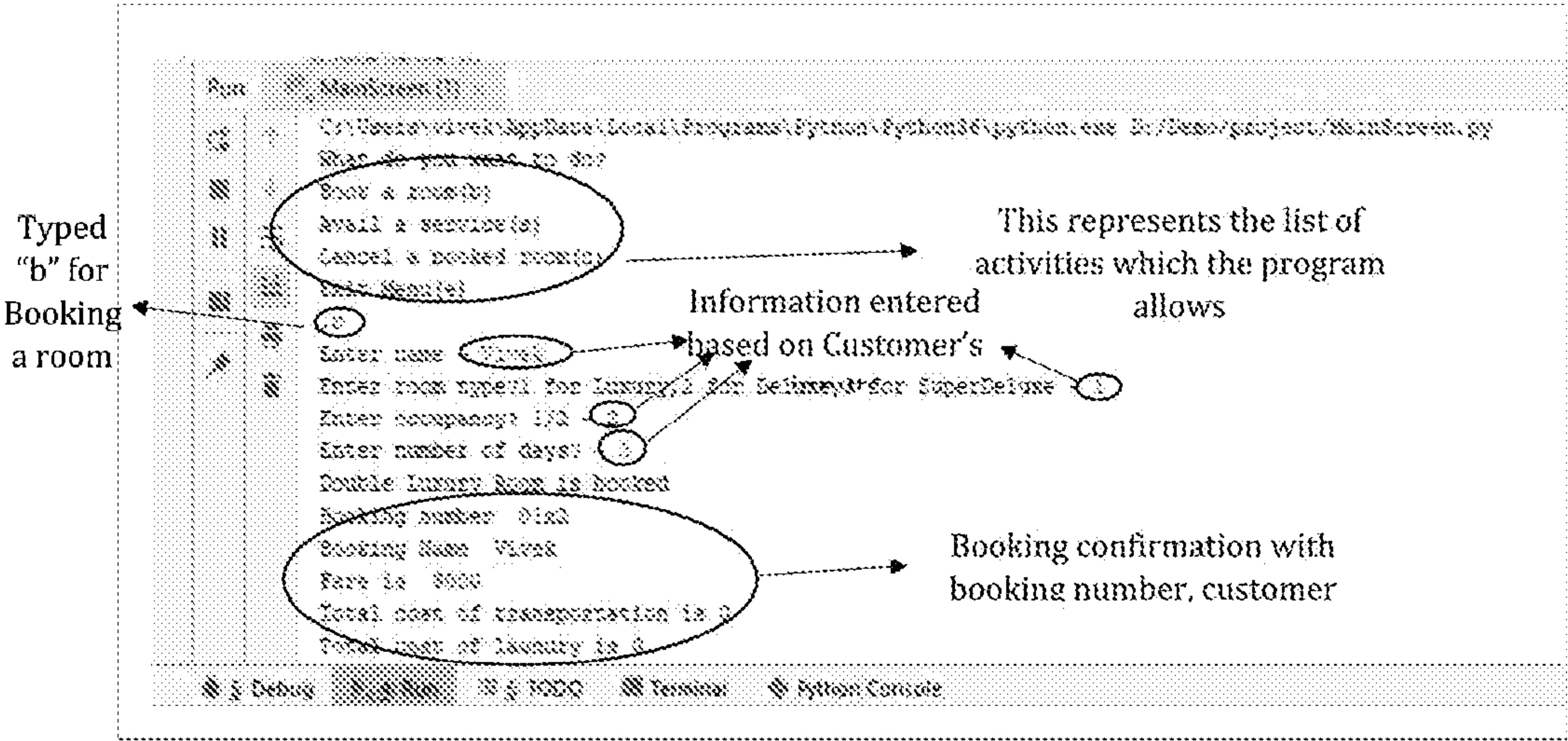


Final Key word is used in this program to fix the value of the Attribute -Total Marks

The Attribute Total Marks is fixed at 100 and cannot be changed.

In NSL, the value of an attribute is fixed in an entity and cannot be changed wherever it is reused. It represents the last level of differentiation.

DRAWING 108A:



Drawing 108B

Book Hotel Room.

Get Customer Information. Select Room Type. Get Room Stay Information. Calculate Fare Room.
Book Hotel Room.

Entity	<u>Customer</u>			
Primary Attributes	<u>Name:</u> Vivek			

Customer name

Book Hotel Room.

Get Customer Information. Select Room Type Super Deluxe. Get Room Stay Information.
Calculate Fare Room. Book Hotel Room.

Entity	<u>Luxury room</u>			
Primary Attributes	<u>Rate:</u> 1,000	<u>Room availability:</u> Yes		

Room rate is a "constant" based on the type of room selected.

Book Hotel Room.

Get Customer Information. Select Room Type Super Deluxe. Get Room Stay Information.
Calculate Fare Room. Book Hotel Room.

Entity	<u>Room stay information</u>		
Primary Attributes	<u>Number of days:</u> 3	<u>Occupancy:</u> 2	

Book Hotel Room.

Entered based on Customer's inputs

Get Customer Information. Select Room Type Super Deluxe. Get Room Stay Information.
Calculate Fare Room. Book Hotel Room.

Entity	<u>Room fare</u>
Primary Attributes	<u>Number of days * Rate * Occupancy:</u> 3,000

Fare calculated by the machine agent

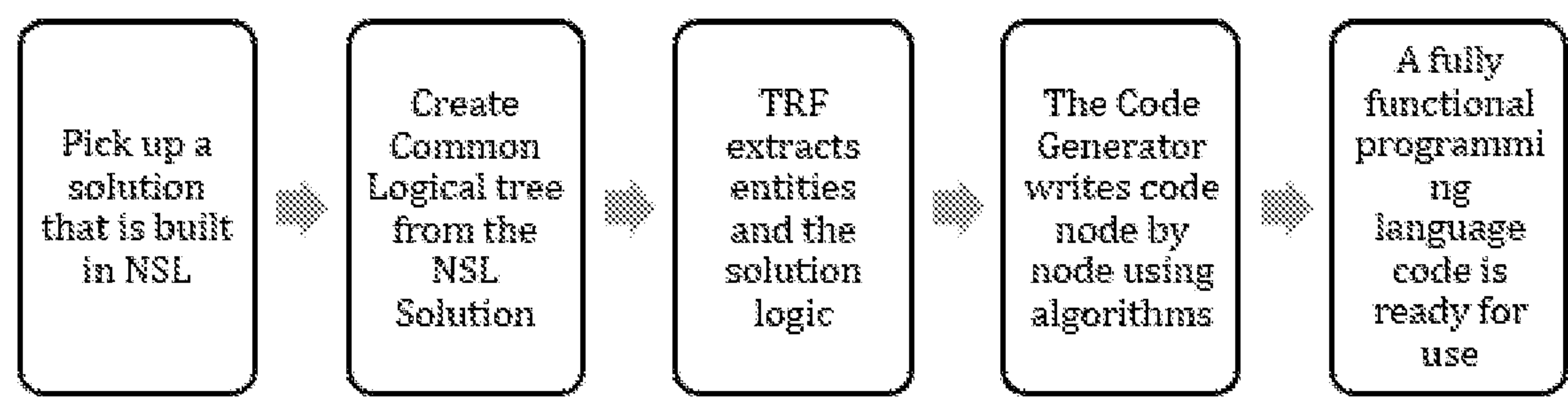
Book Hotel Room.

Get Customer Information. Select Room Type Super Deluxe. Get Room Stay Information.
Book Hotel Room.

Entity	<u>Booking Confirmation</u>			
Primary Attributes	<u>ID No:</u> 01x2	<u>Booking name:</u> Vivek	<u>Fare:</u> 3,000	<u>Cost of transportation:</u> 0

Booking confirmation with booking number, customer name and fare.

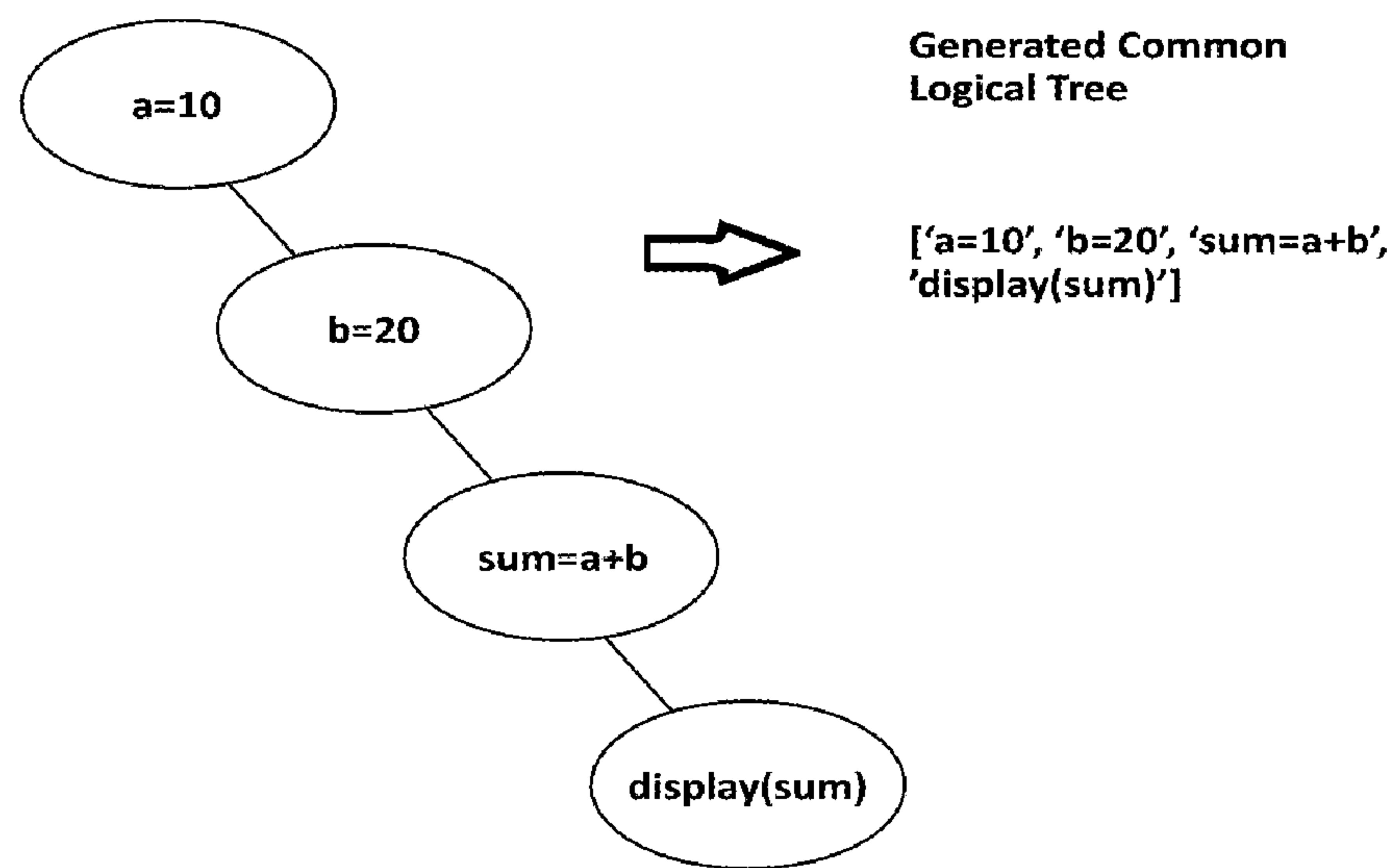
Drawing109:



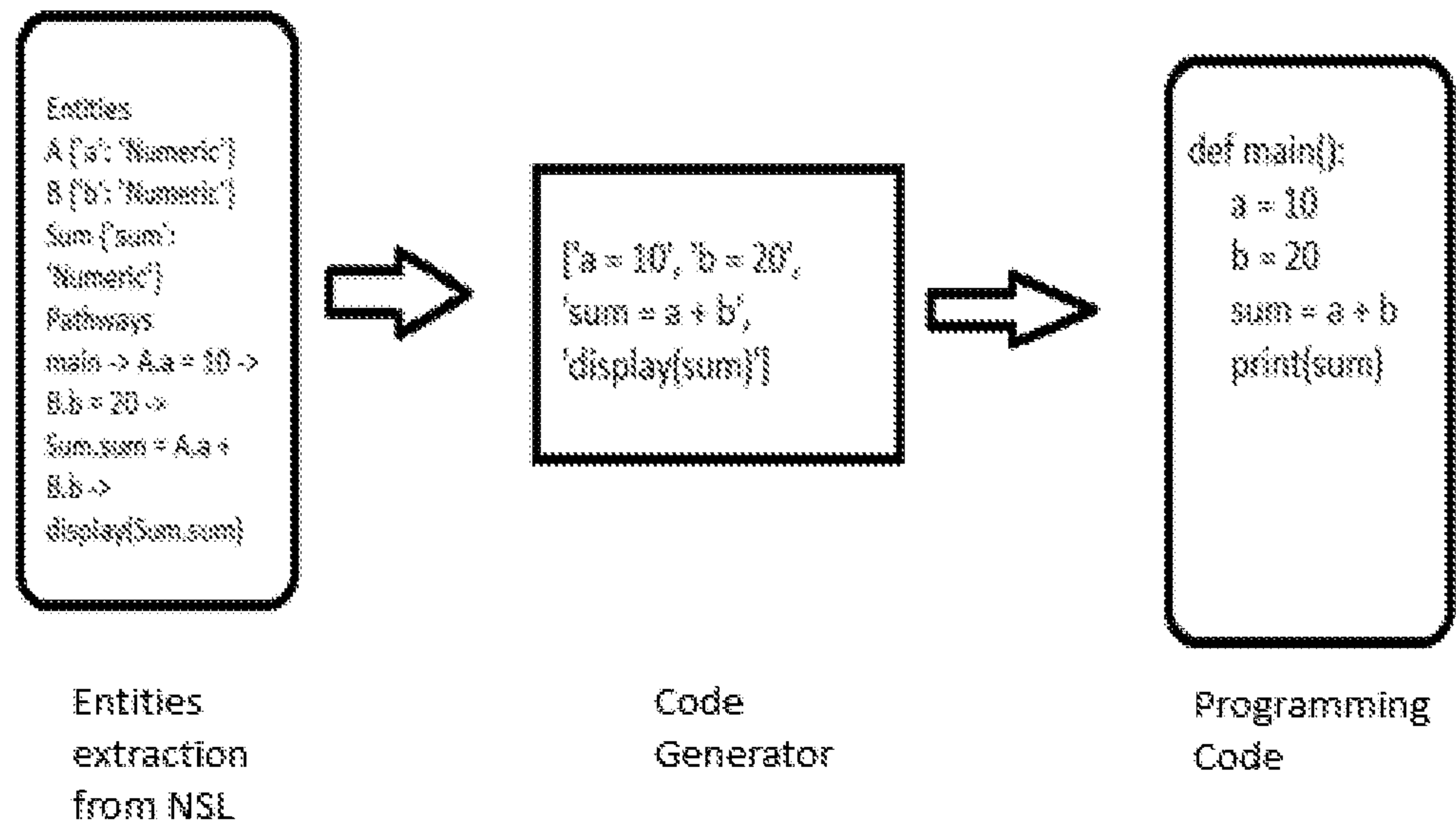
Drawing110:

Particulars	NSL	Python Program	C Program
Solution construct	Fill out the deposit slip. Validated by Teller. Verify Check. CU's: 3 Entities participating: 8	<pre>print(" ATM Banking Services ") print("1) Balance 2) WithDraw 3) Deposit 4)Quit") print("Enter options: ") option_select = input("Enter Here :") if option_select=="Balance": print(balance) else : if option_select=="WithDraw":</pre>Line 21	<pre>#include<stdio.h> int main() { printf("ATMBankingServices"); printf("1)Balance2)WithDraw3)Deposit4)Quit"); printf("Enteroptions:"); scanf("%d",&option_select); if (option_select=="Balance") printf("%d",new_balance)}</pre> Line 33
Inputs	Enter amount to deposit: 123 Username: ABC Password: XYZ	Enter amount to deposit: 123 Username: ABC Password: XYZ	Enter amount to deposit: 123 Username: ABC Password: XYZ
Outputs	Balance displayed 28,017	Balance displayed 28,017	Balance displayed 28,017

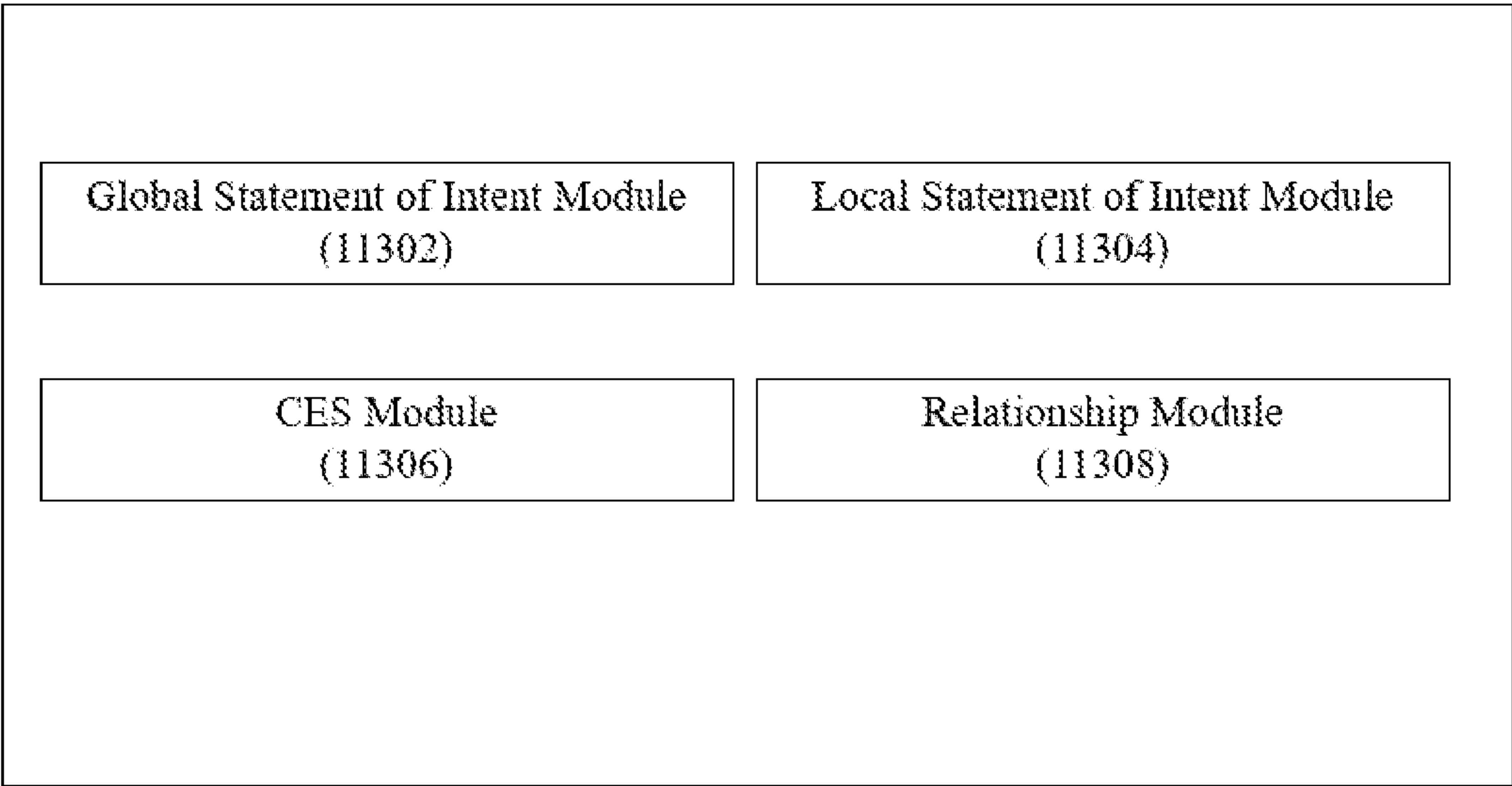
DRAWING 111:



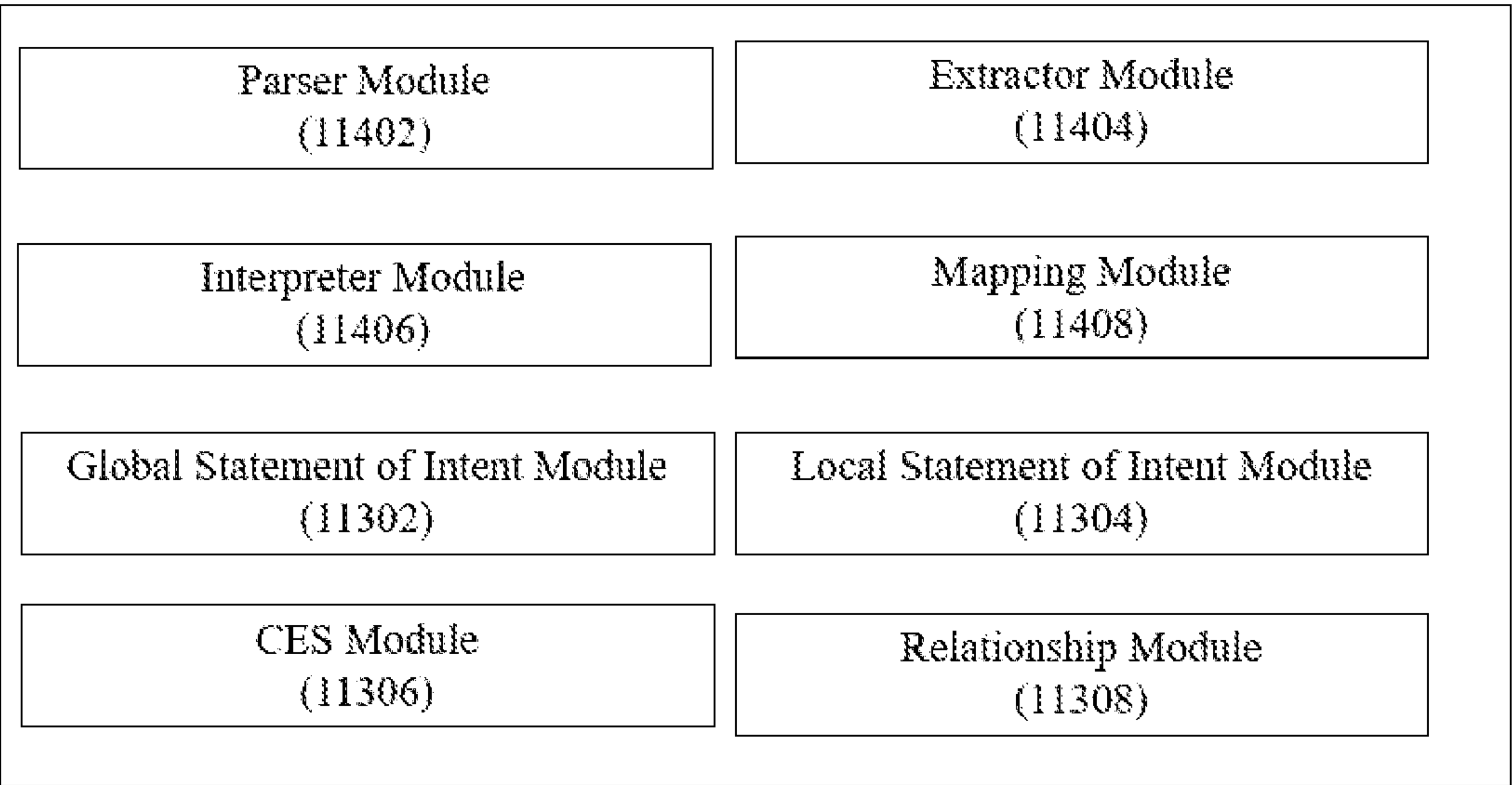
Drawing 12:



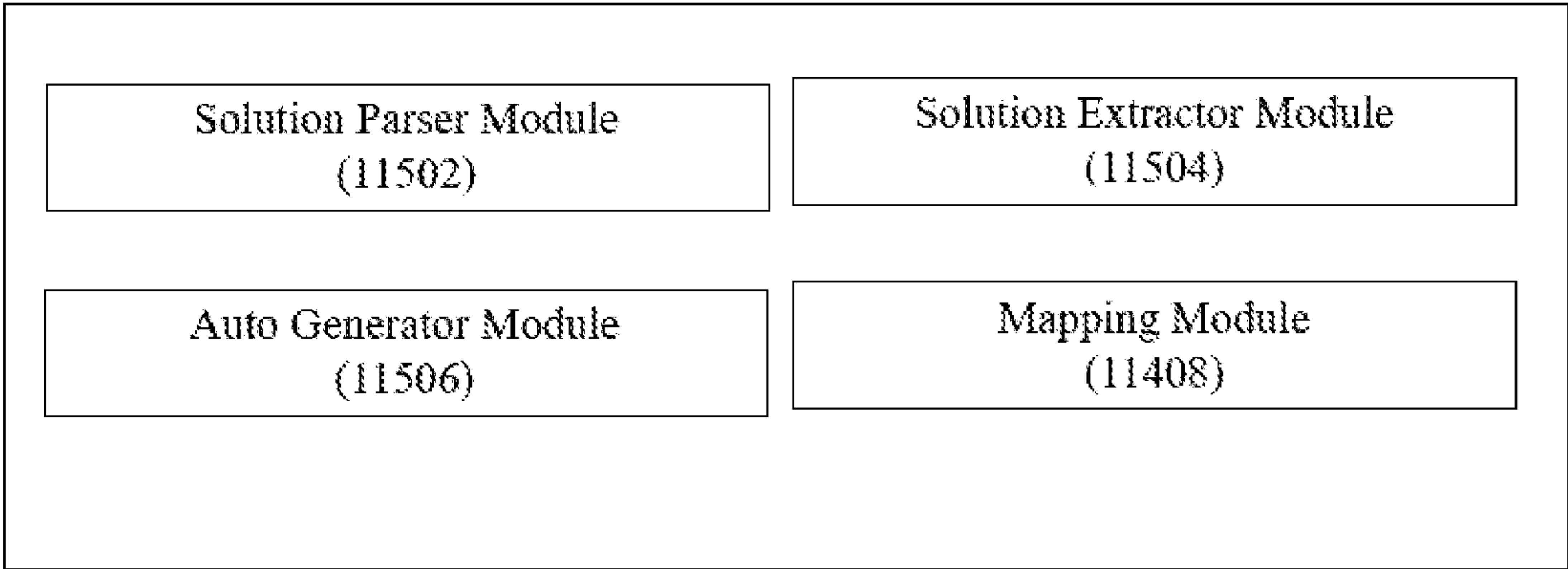
Drawing1 13:



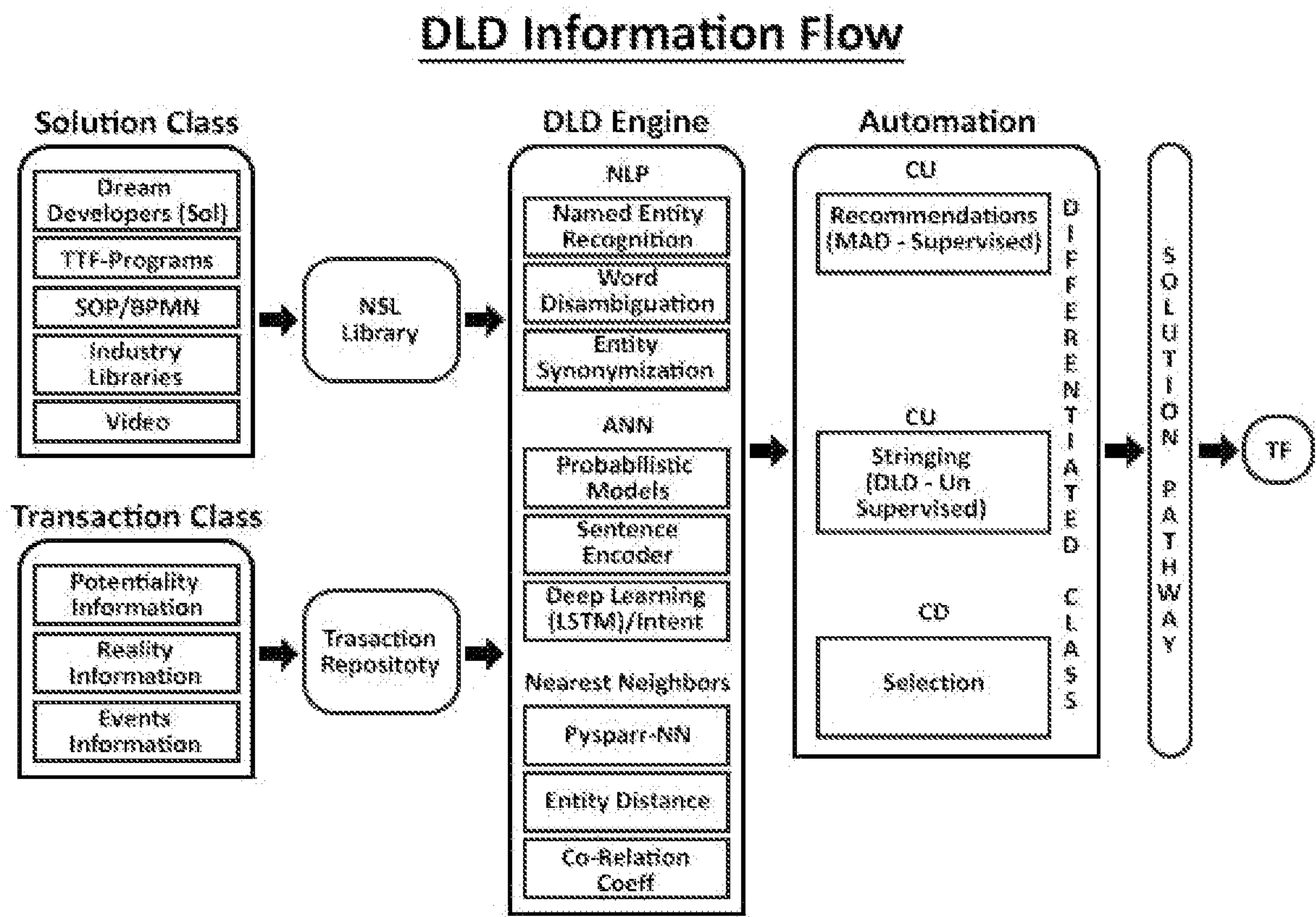
Drawing1 14:



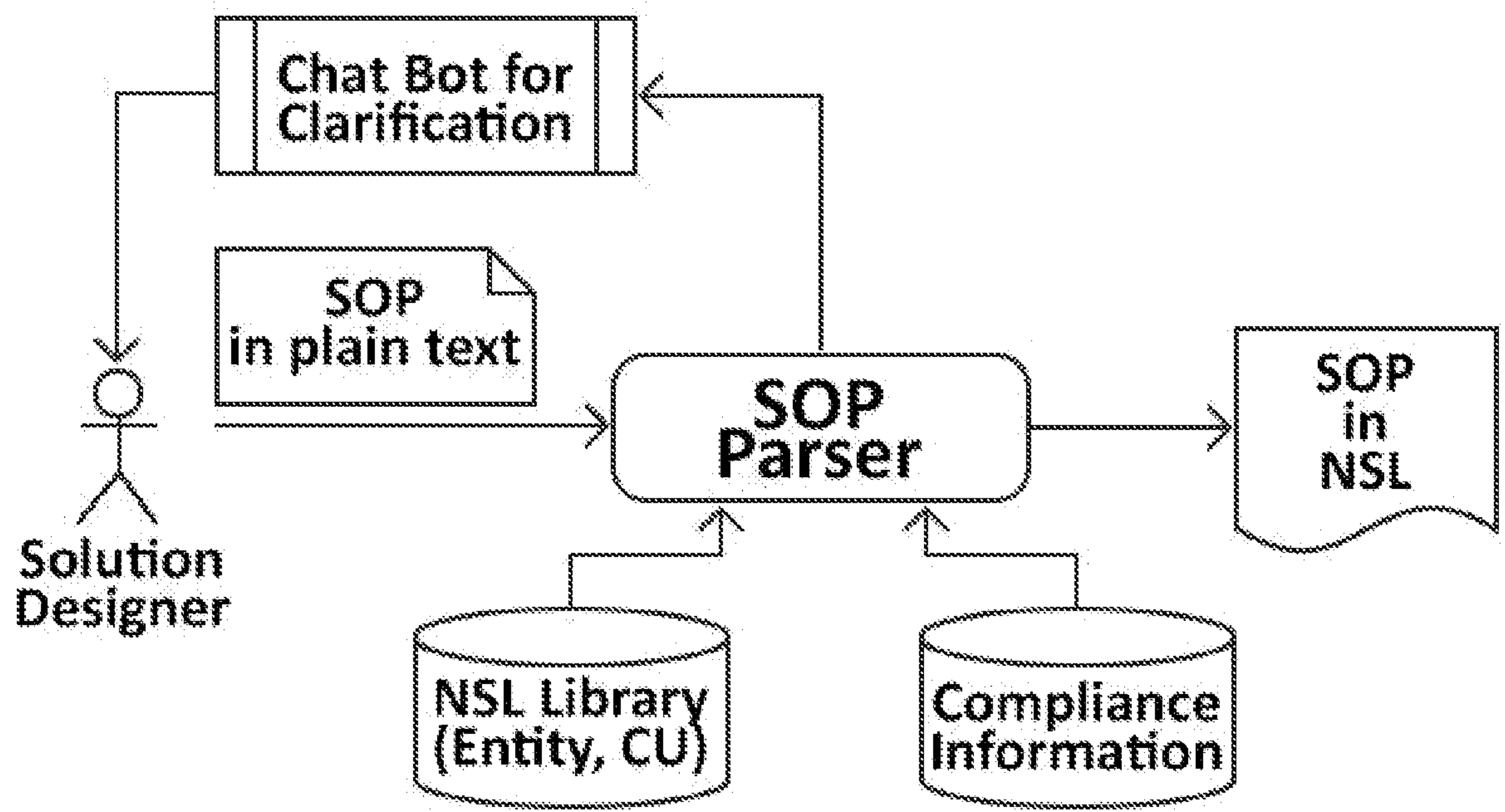
Drawing115:



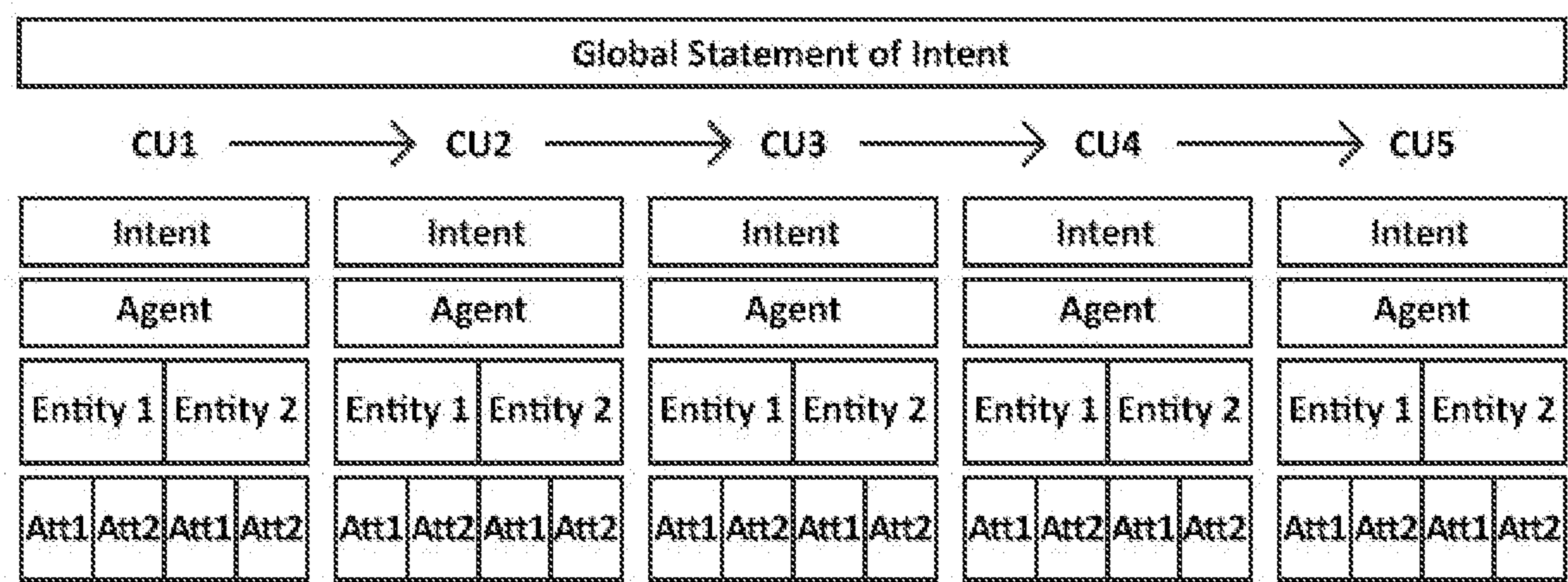
Drawing116:



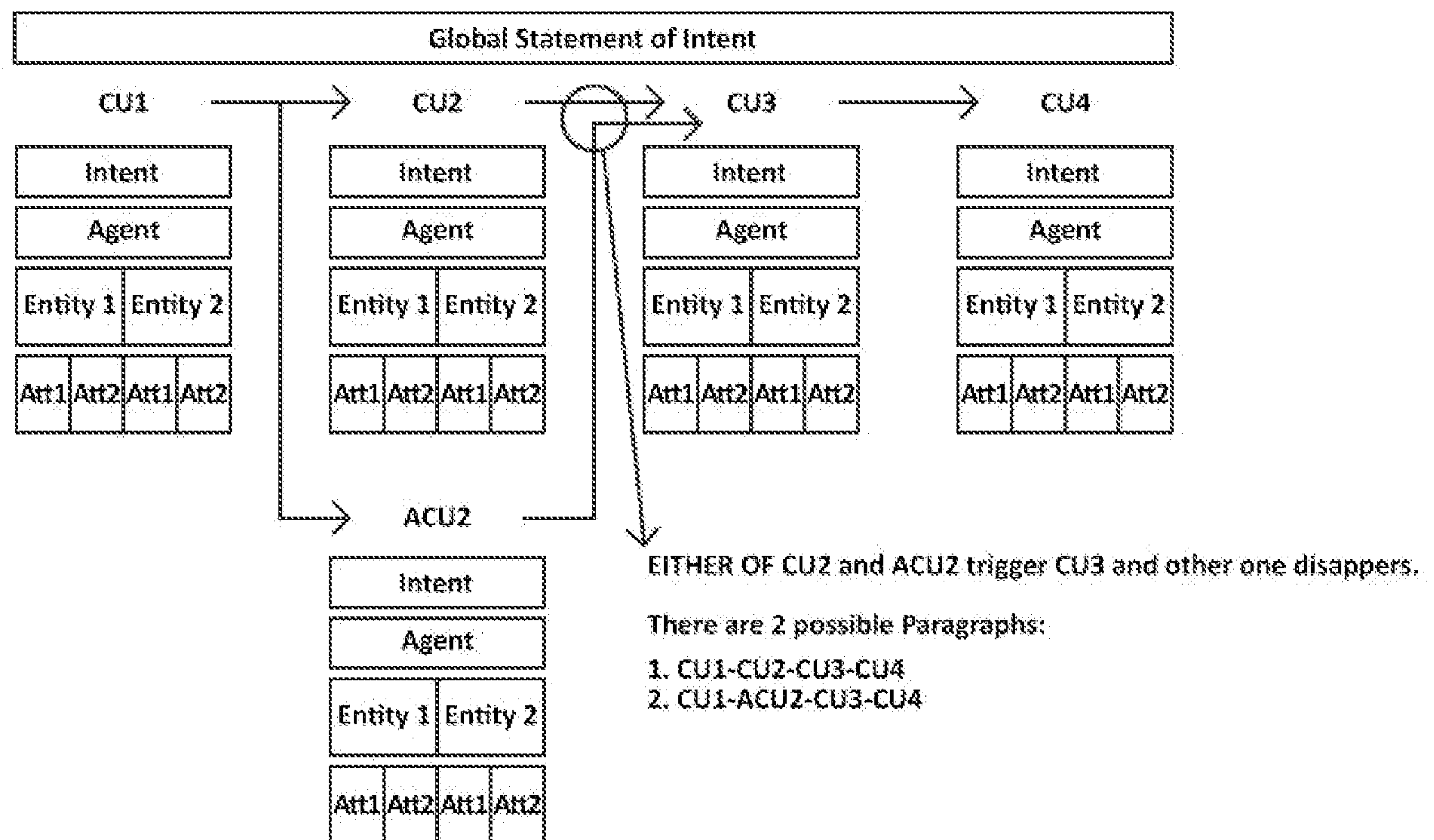
Drawing117:



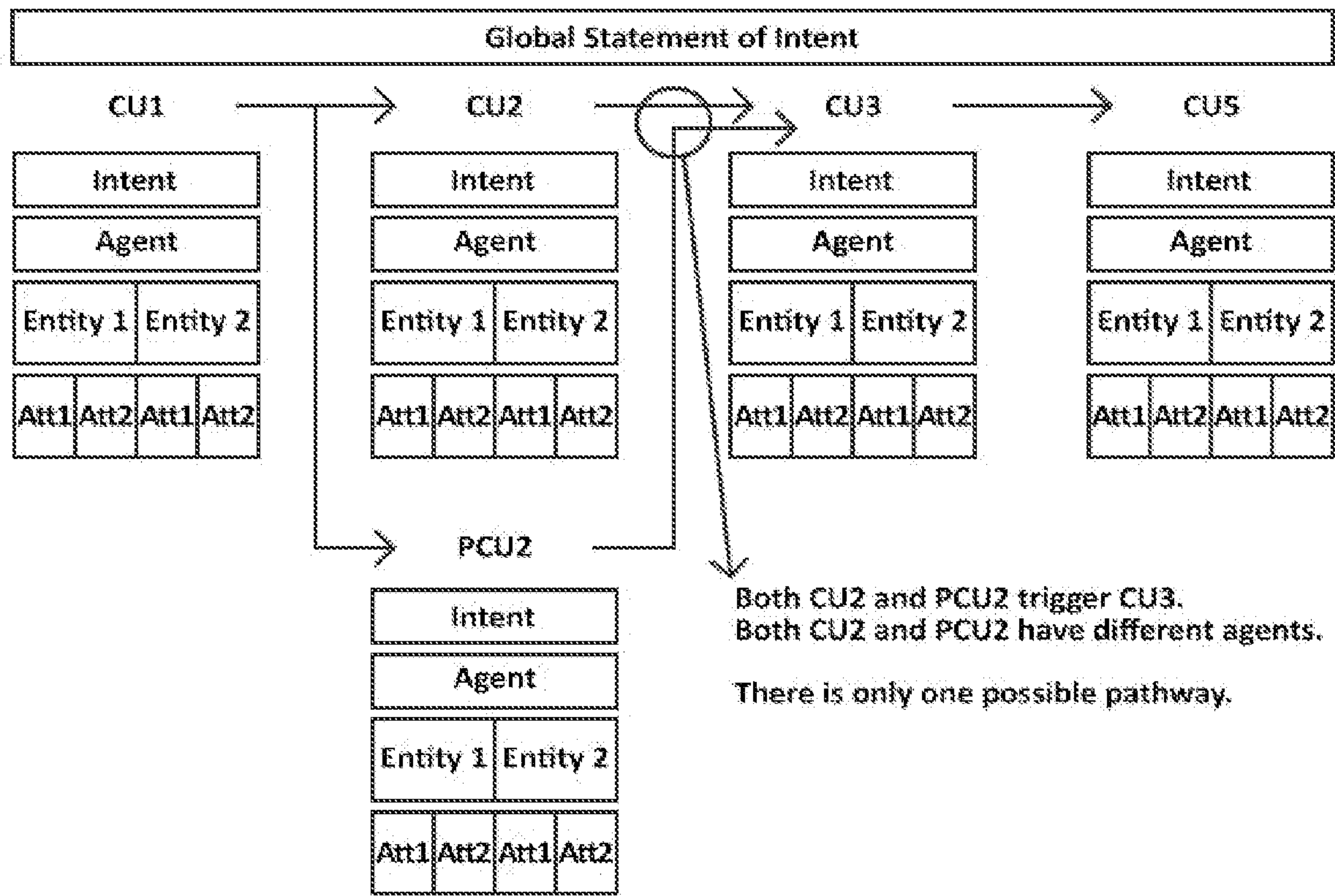
Drawing 118:



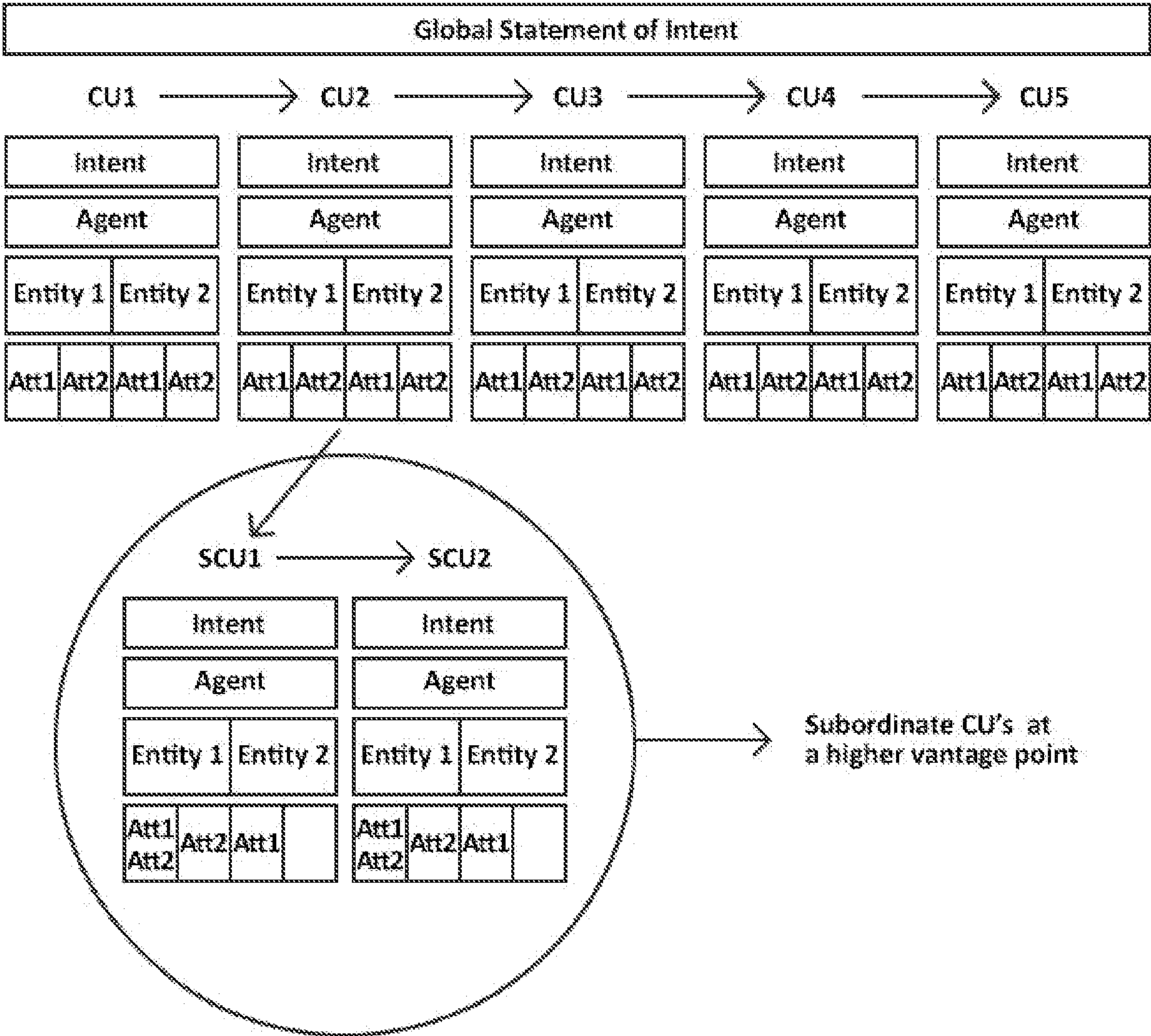
Drawing 119:



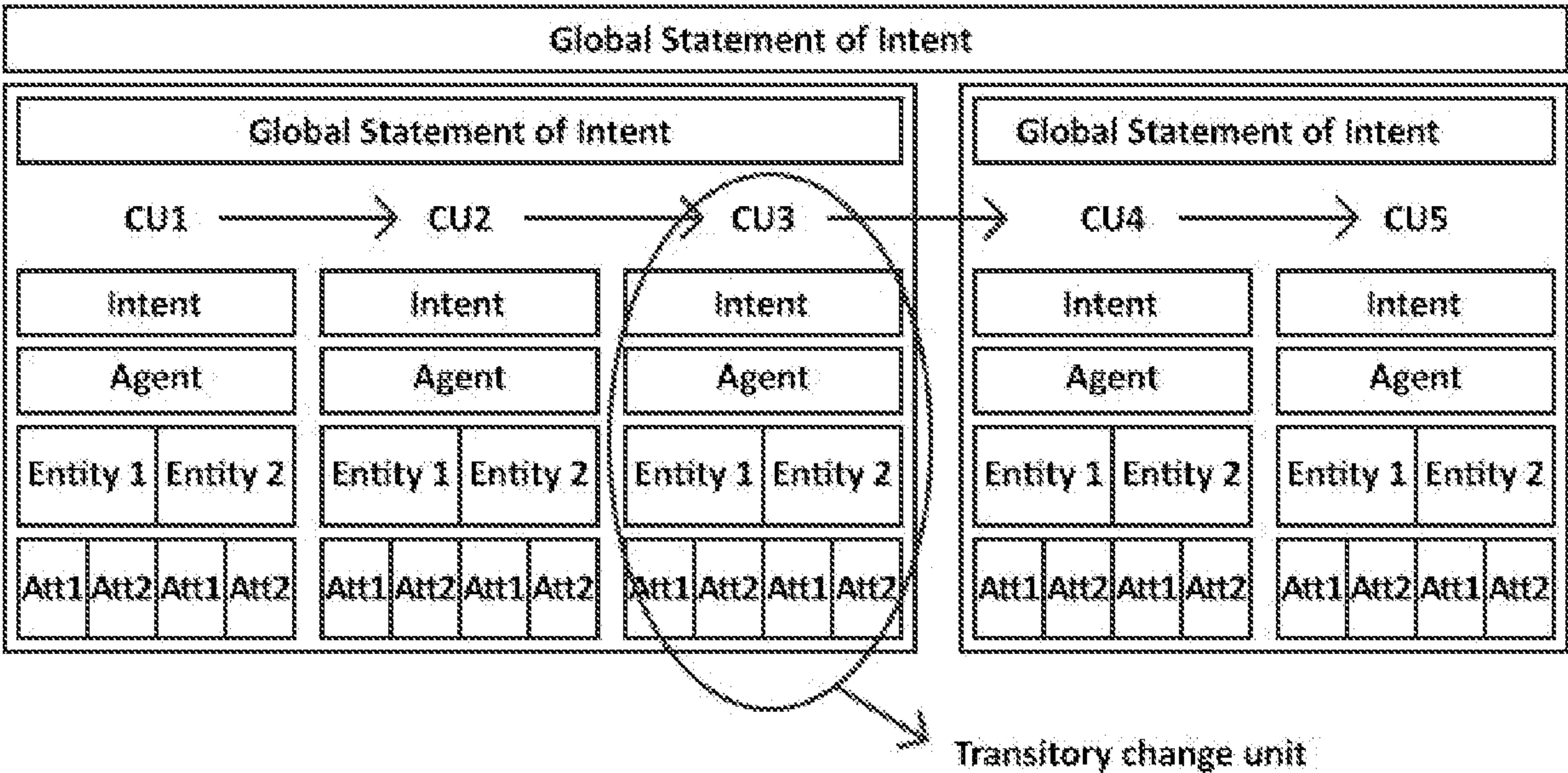
Drawing 120:



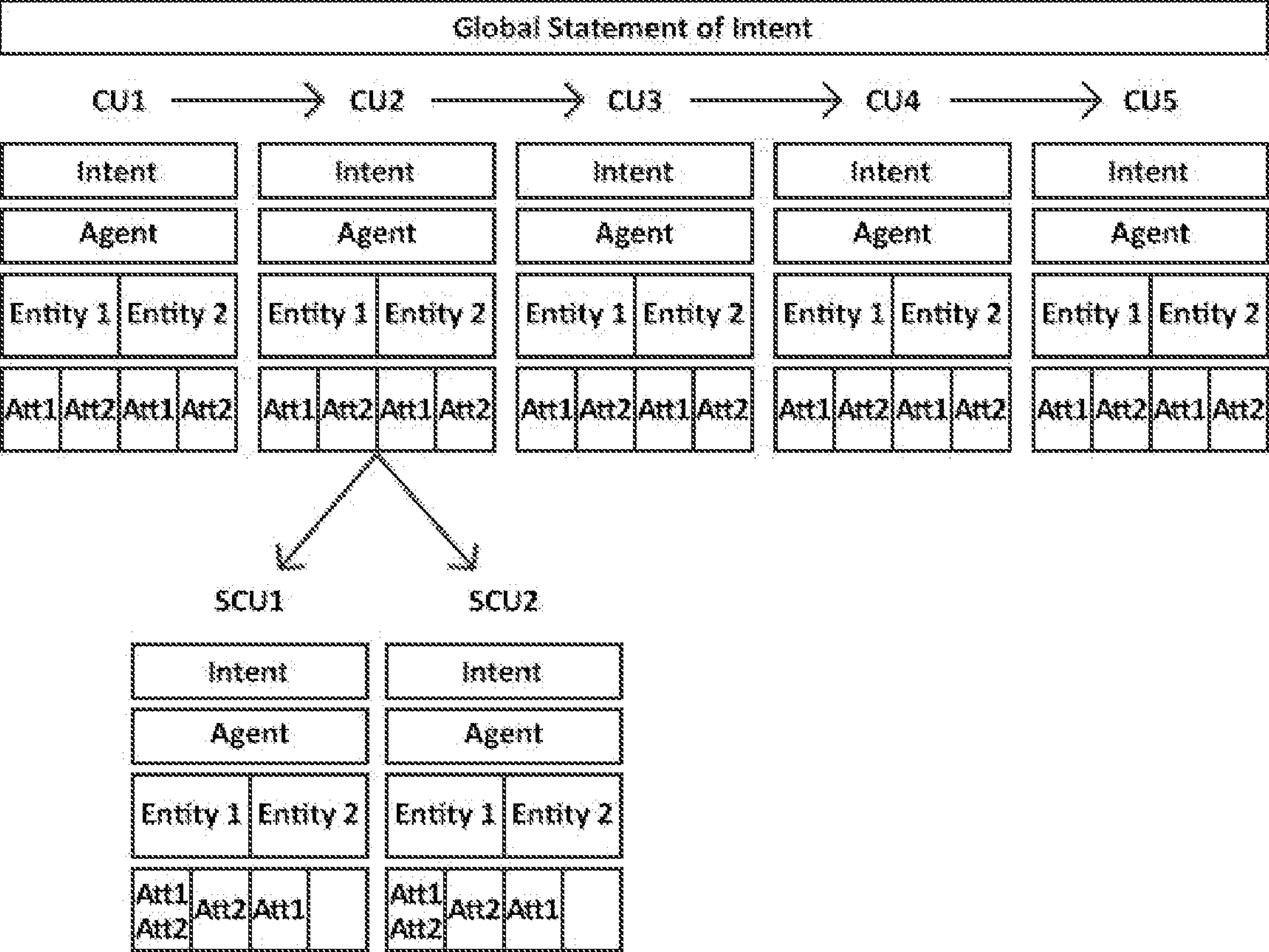
Drawing 121:



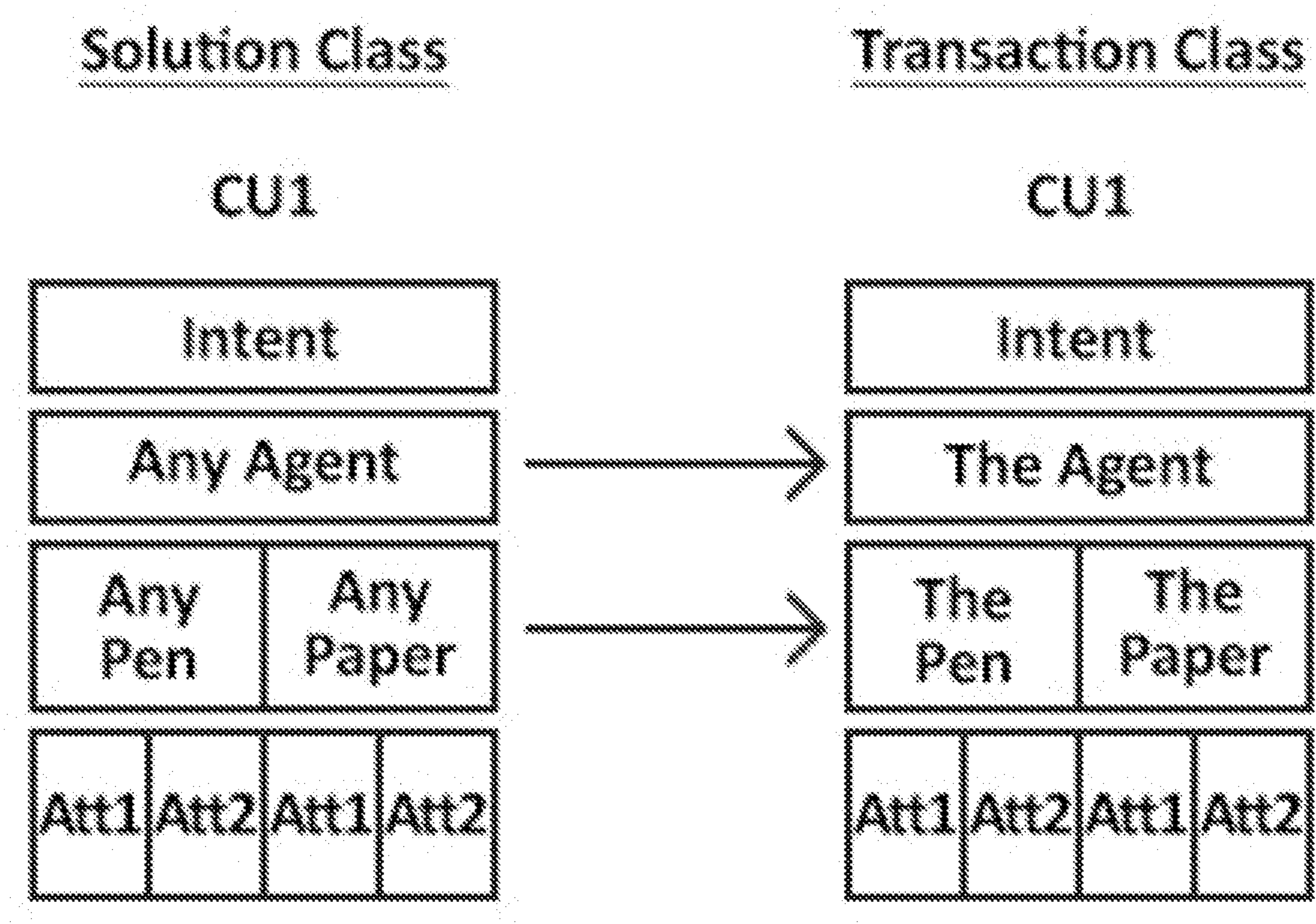
Drawing 122:



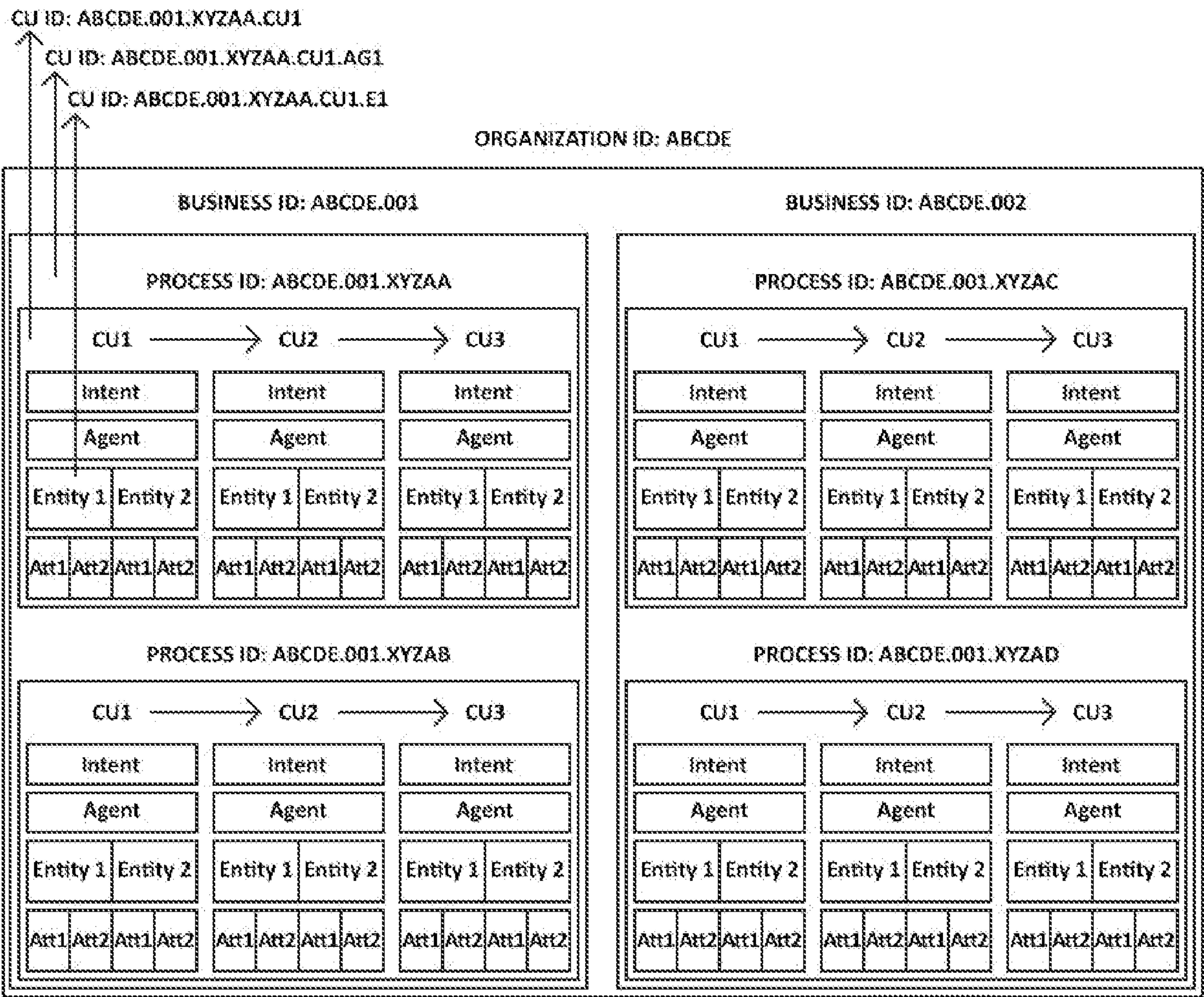
Drawing 123:



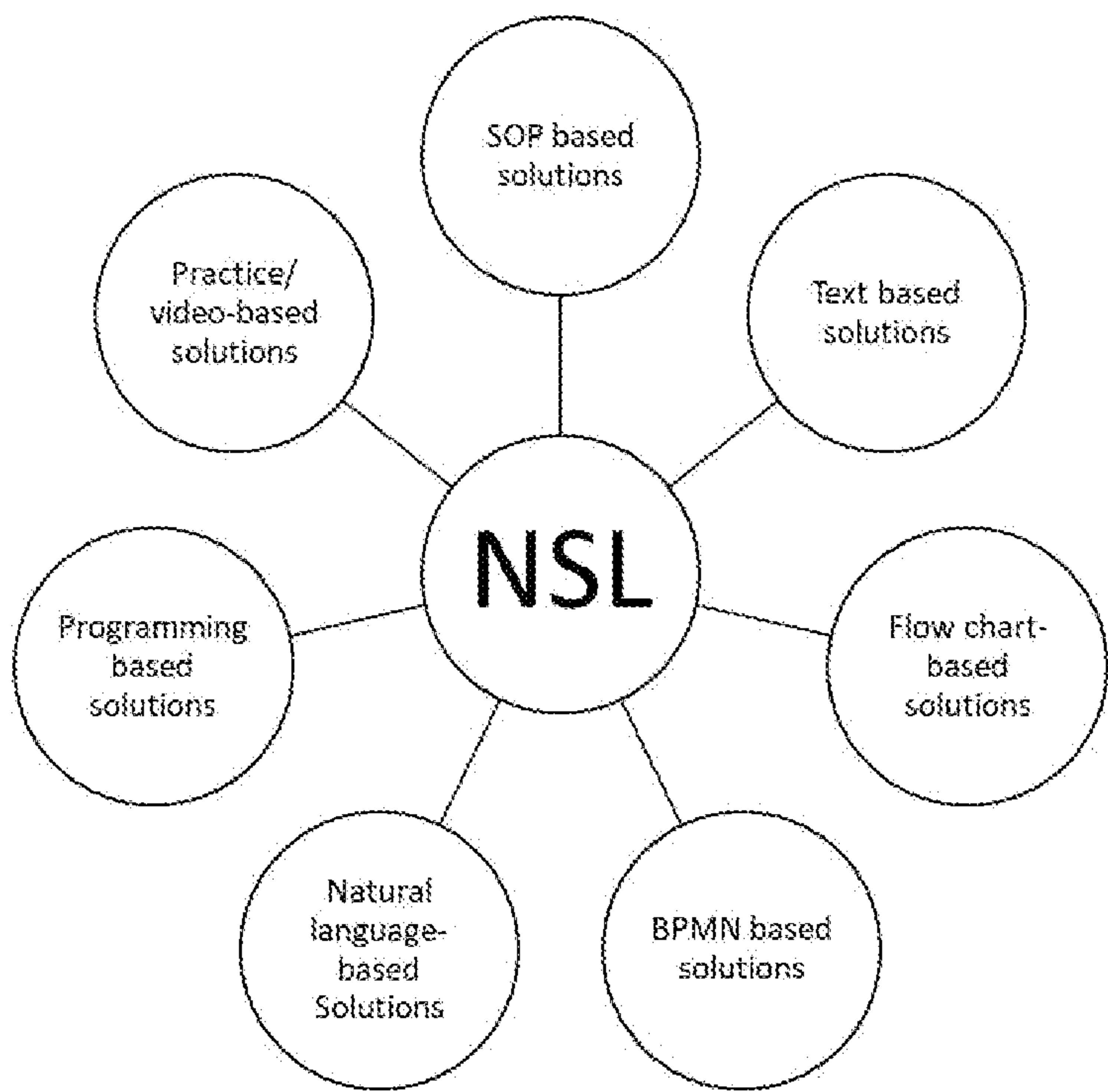
Drawing 124:



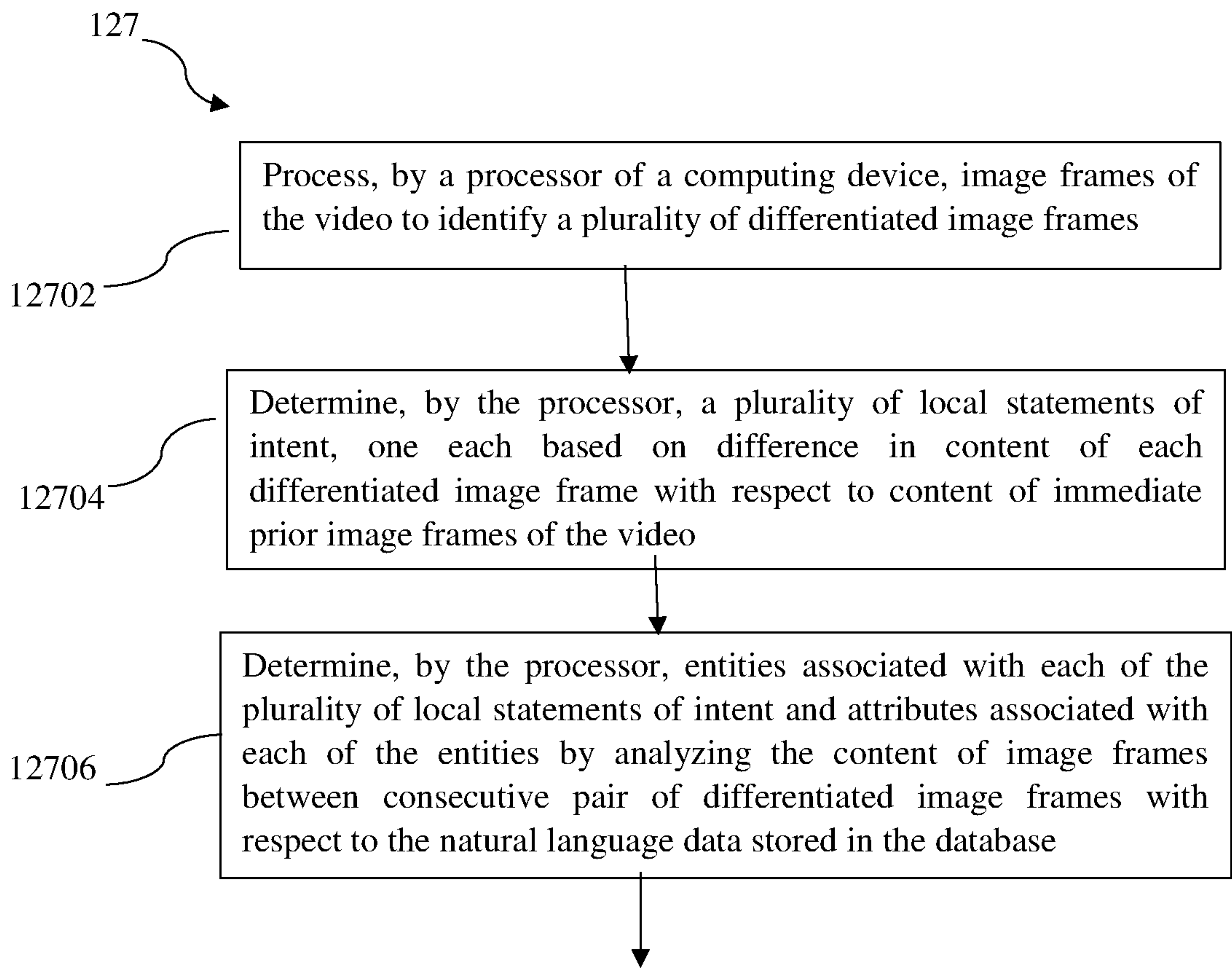
Drawing 125:



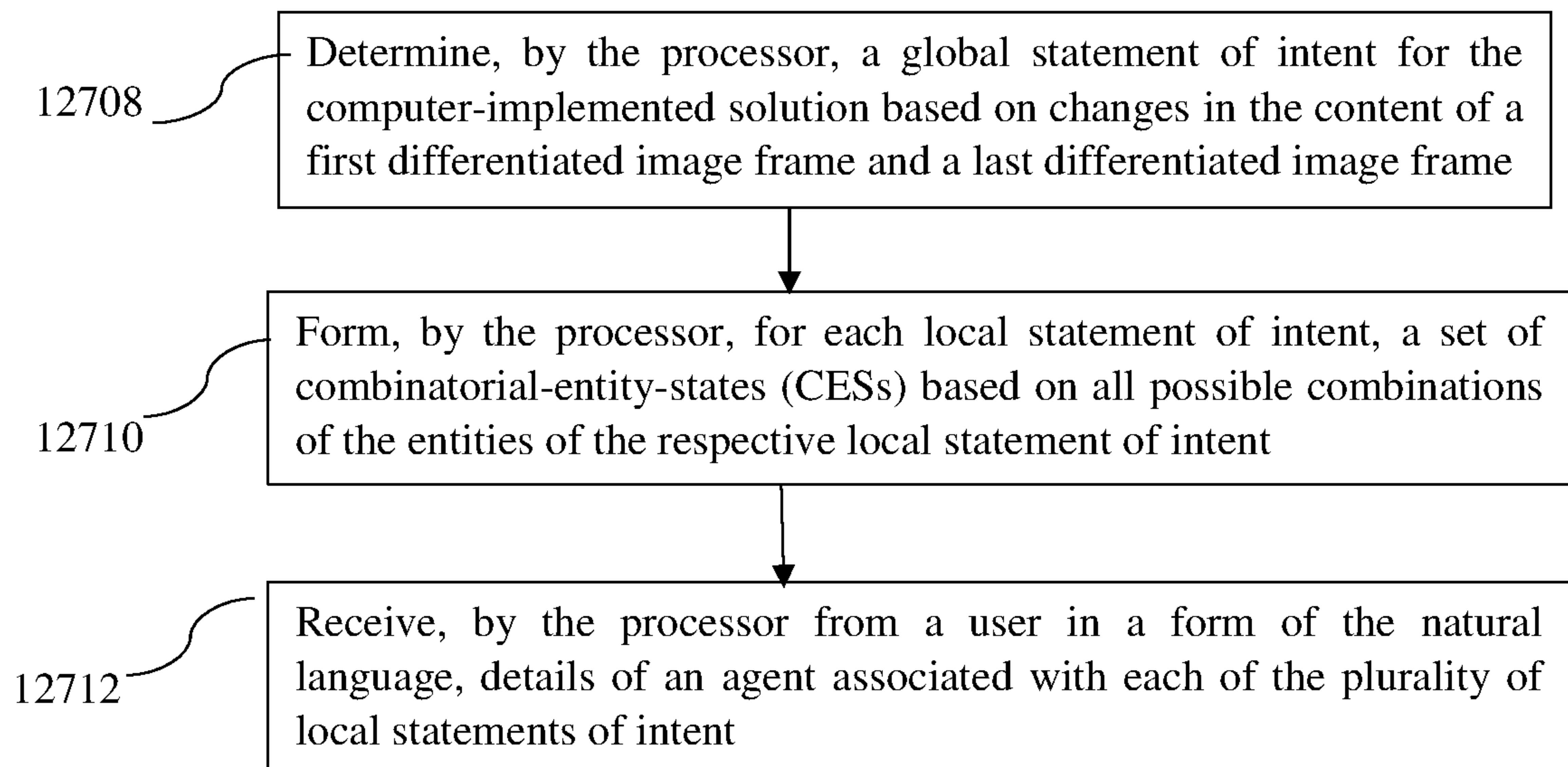
Drawing 126:



Drawing 127:



78/79



Drawing 128:

