



Robert C. Martin Series

Clean Architecture

A Craftsman's Guide to
Software Structure and Design

Robert C. Martin

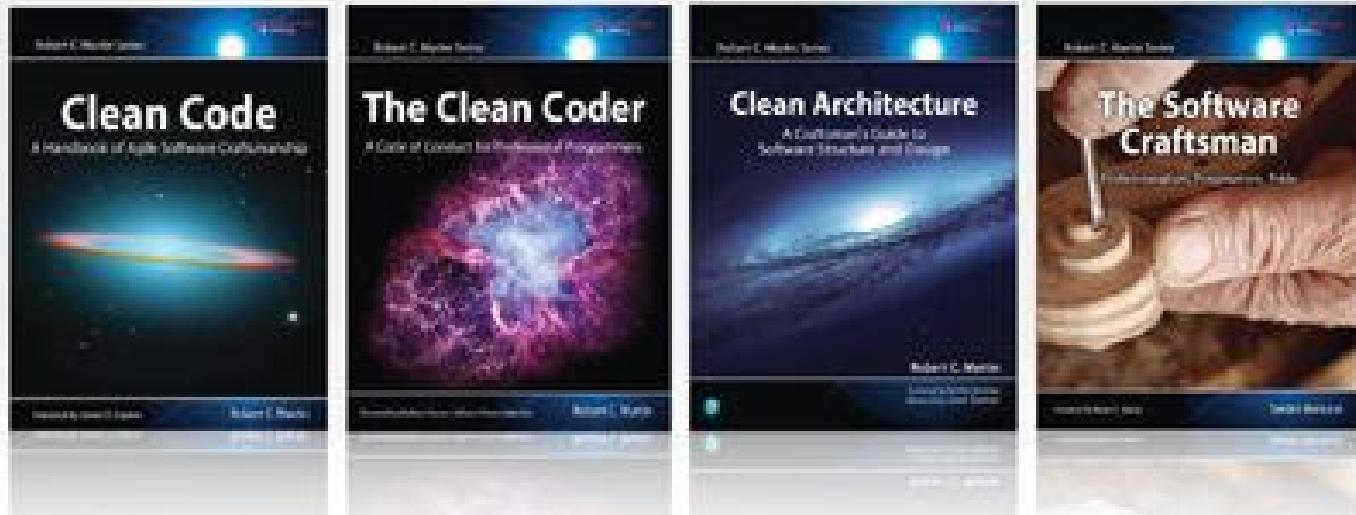
With contributions by James Grenning and Simon Brown

Foreword by Kevlin Henney

Afterword by Jason Gorman



Robert C. Martin Series



Visit informit.com/martinseries for a complete list of available publications.

The Robert C. Martin Series is directed at software developers, team-leaders, business analysts, and managers who want to increase their skills and proficiency to the level of a Master Craftsman. The series contains books that guide software professionals in the principles, patterns, and practices of programming, software project management, requirements gathering, design, analysis, testing, and others.



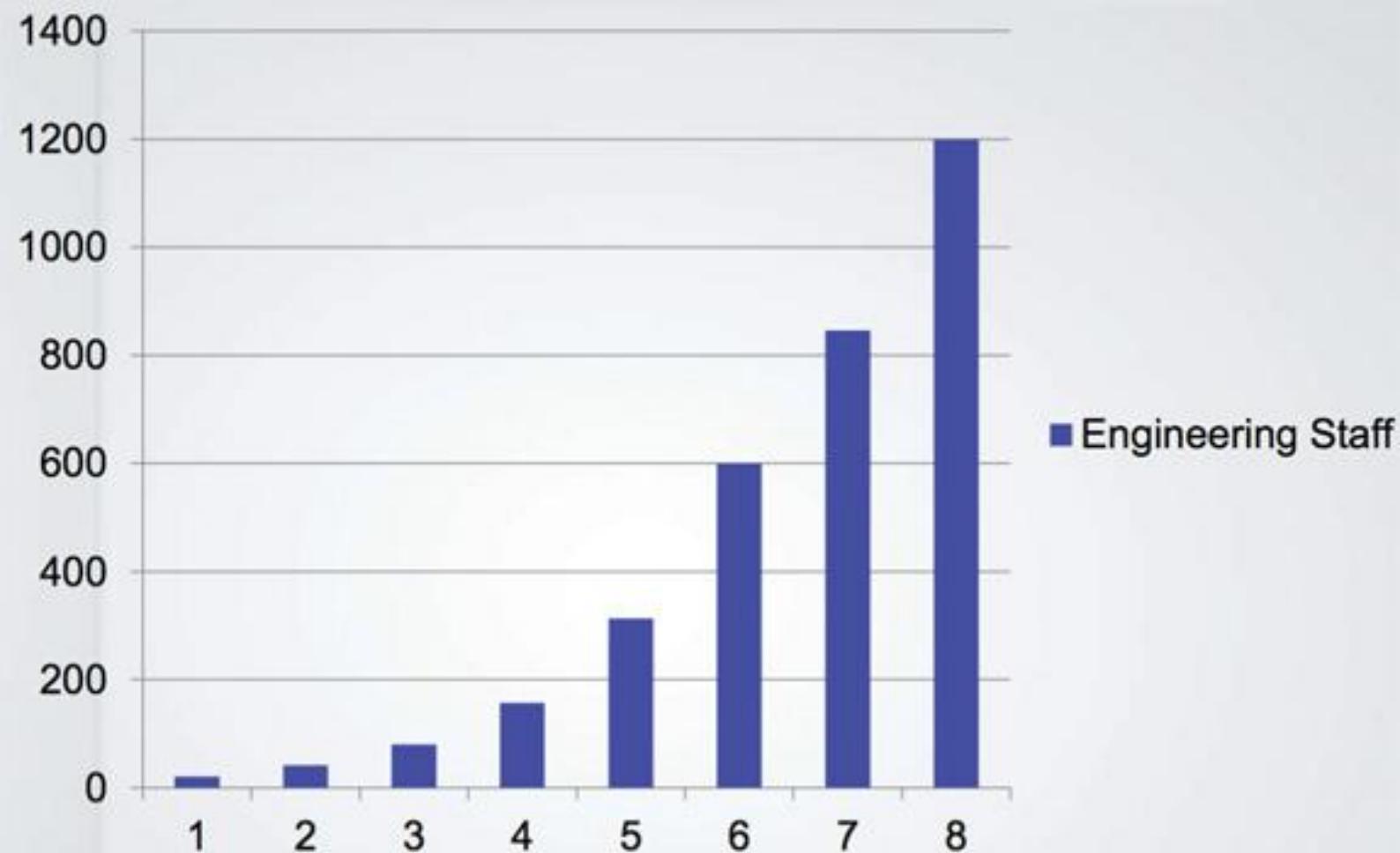
Make sure to connect with us!
informit.com/socialconnect



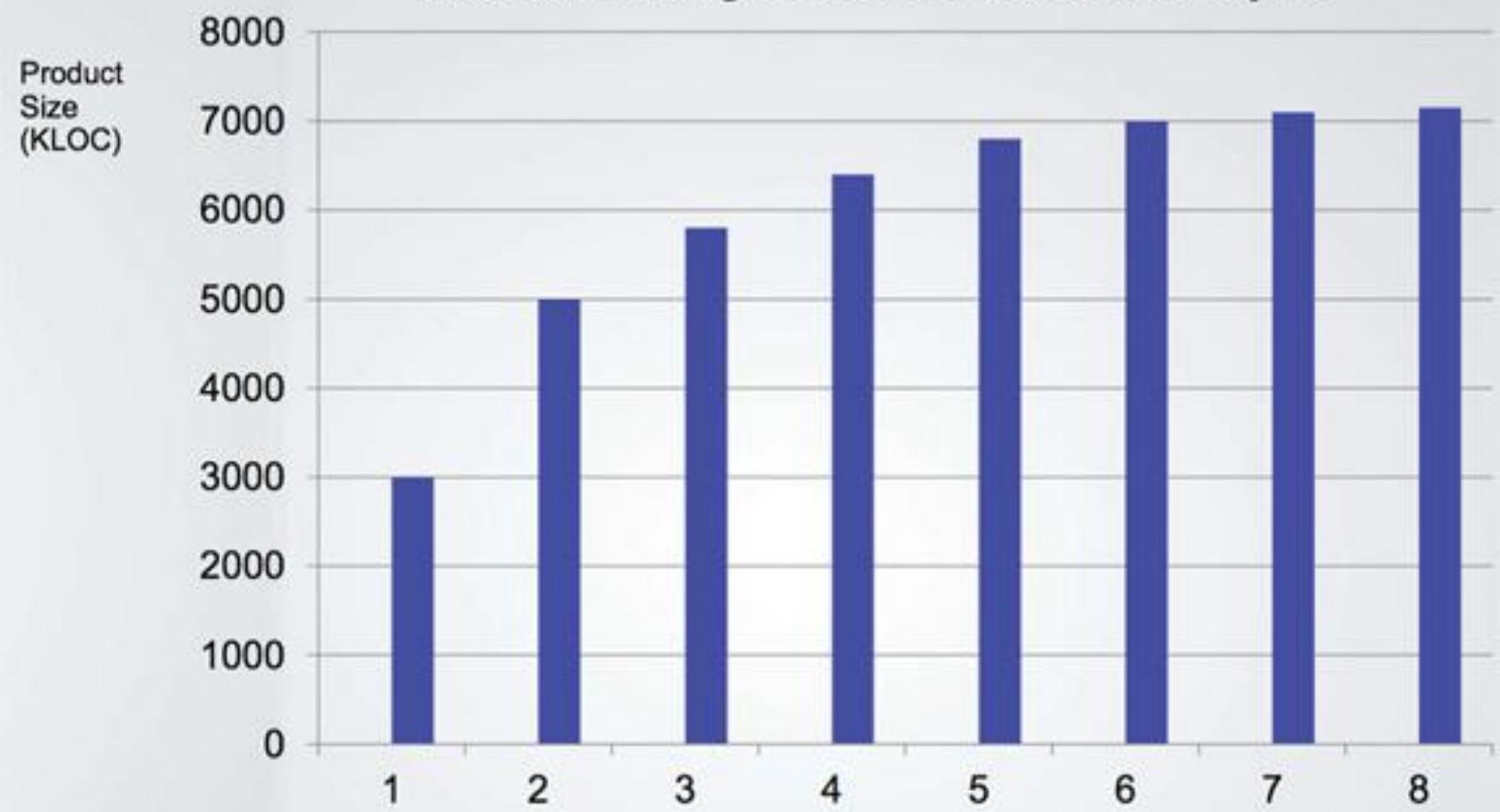




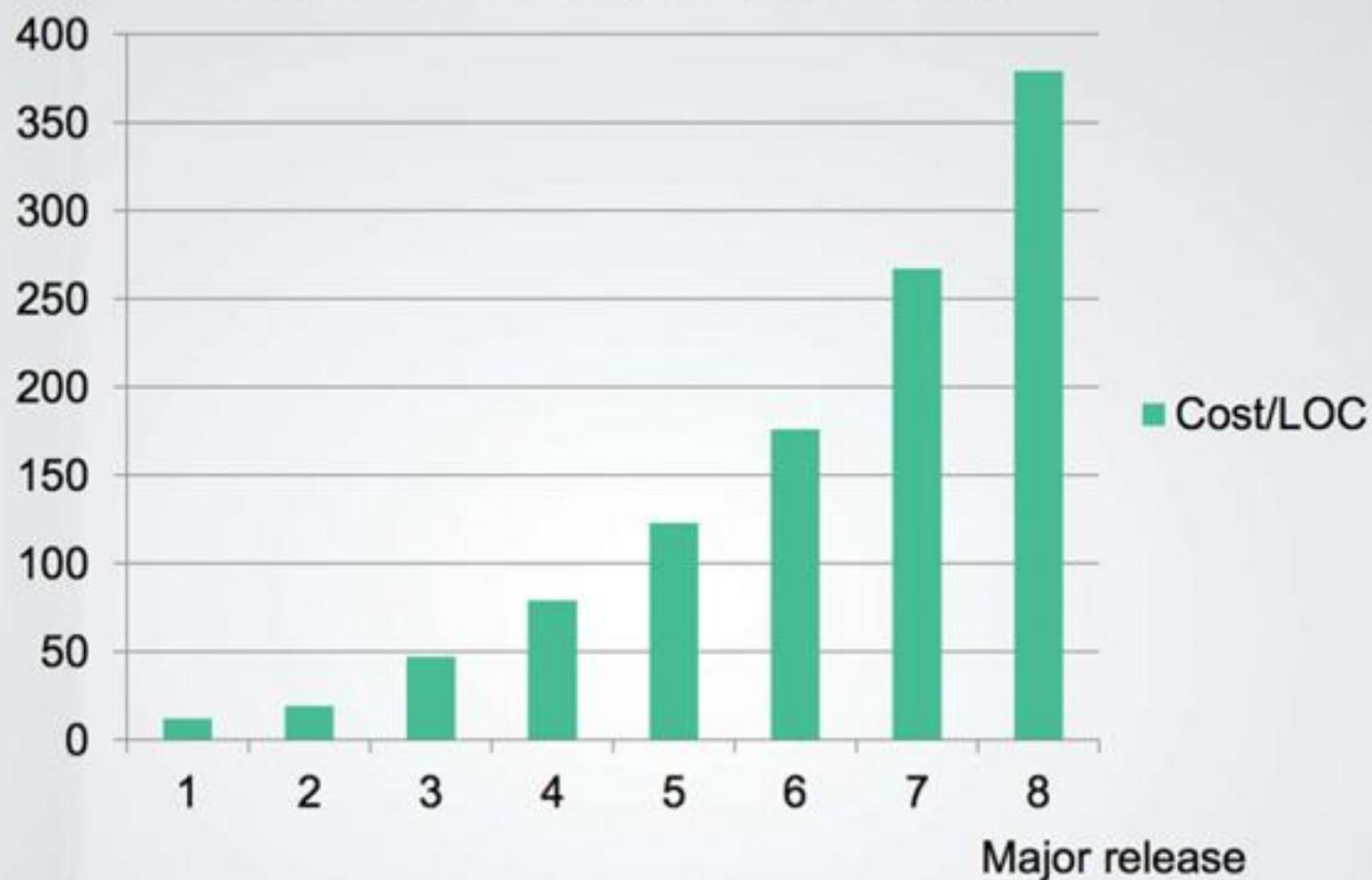
Market-Leading Software Product Life Cycle



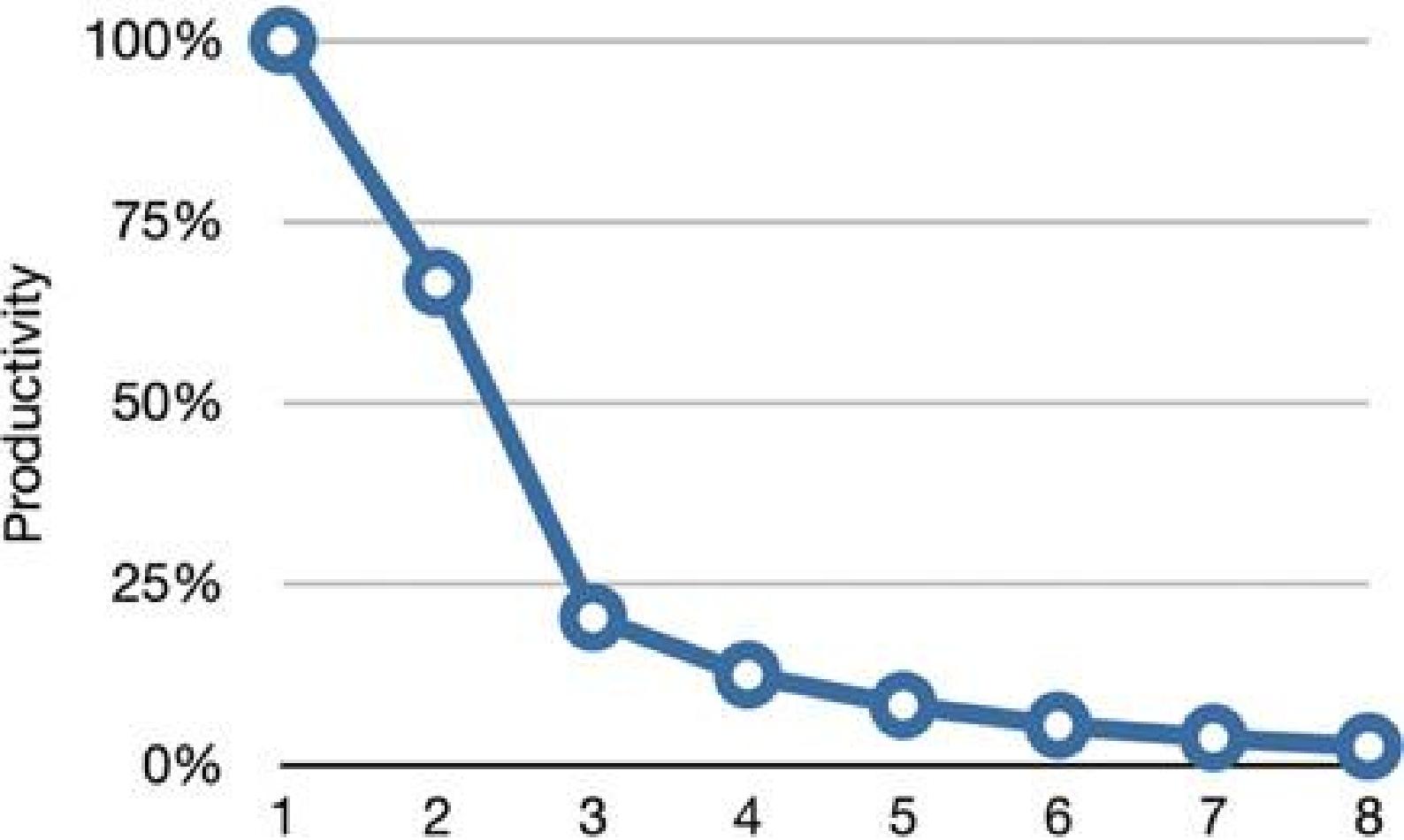
Market-Leading Software Product Life Cycle

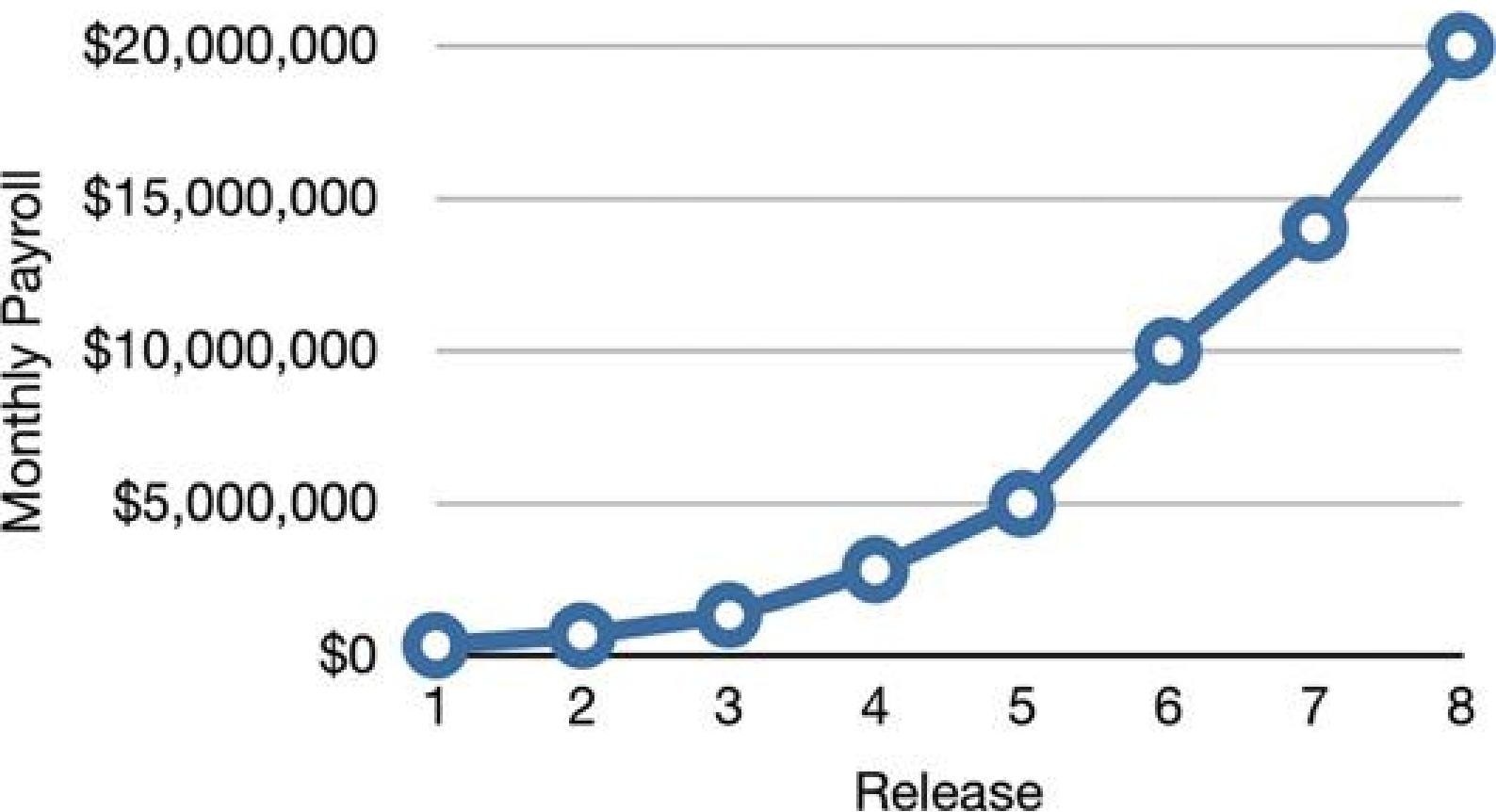


Market-Leading Software Product Life Cycle

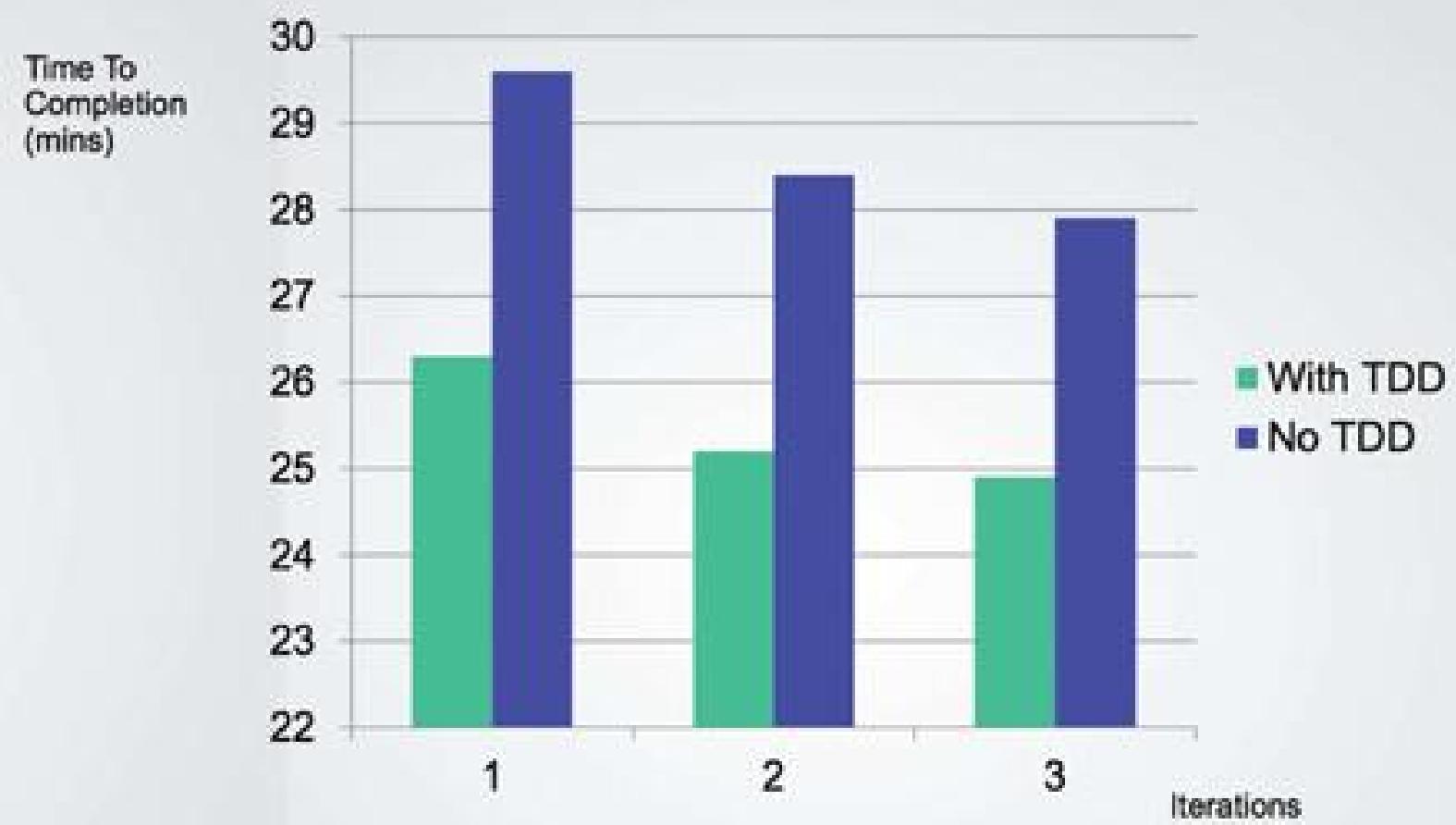


Productivity/Release





Roman Numerals Kata





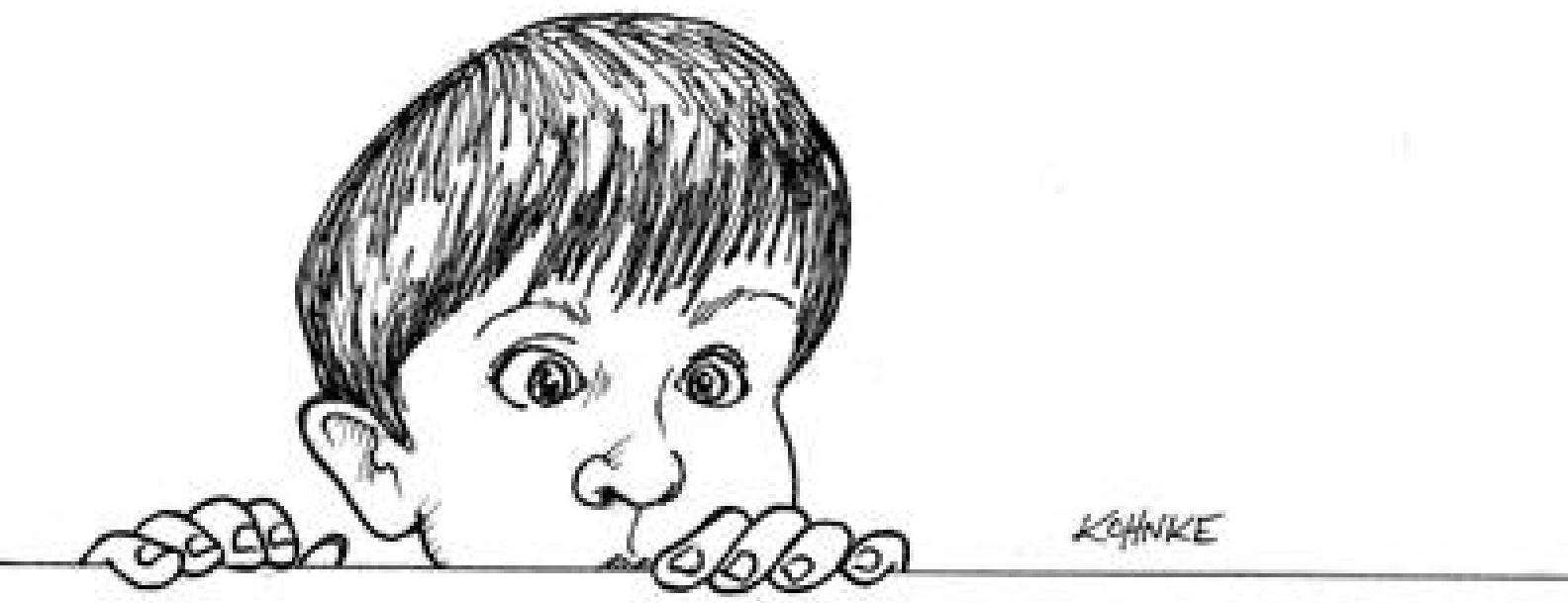
**IMPORTANT
URGENT**

**IMPORTANT
NOT URGENT**

**UNIMPORTANT
URGENT**

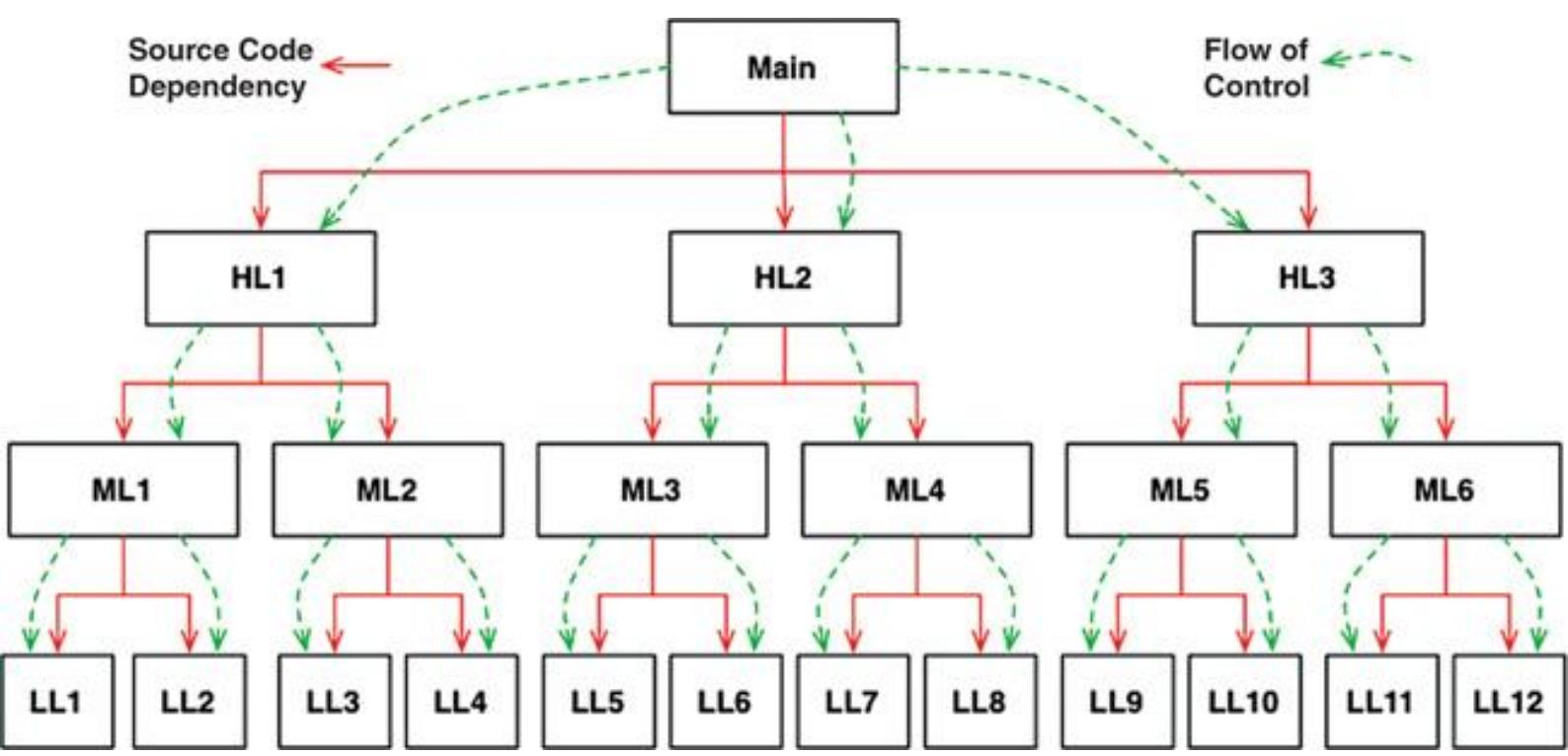
**UNIMPORTANT
NOT URGENT**

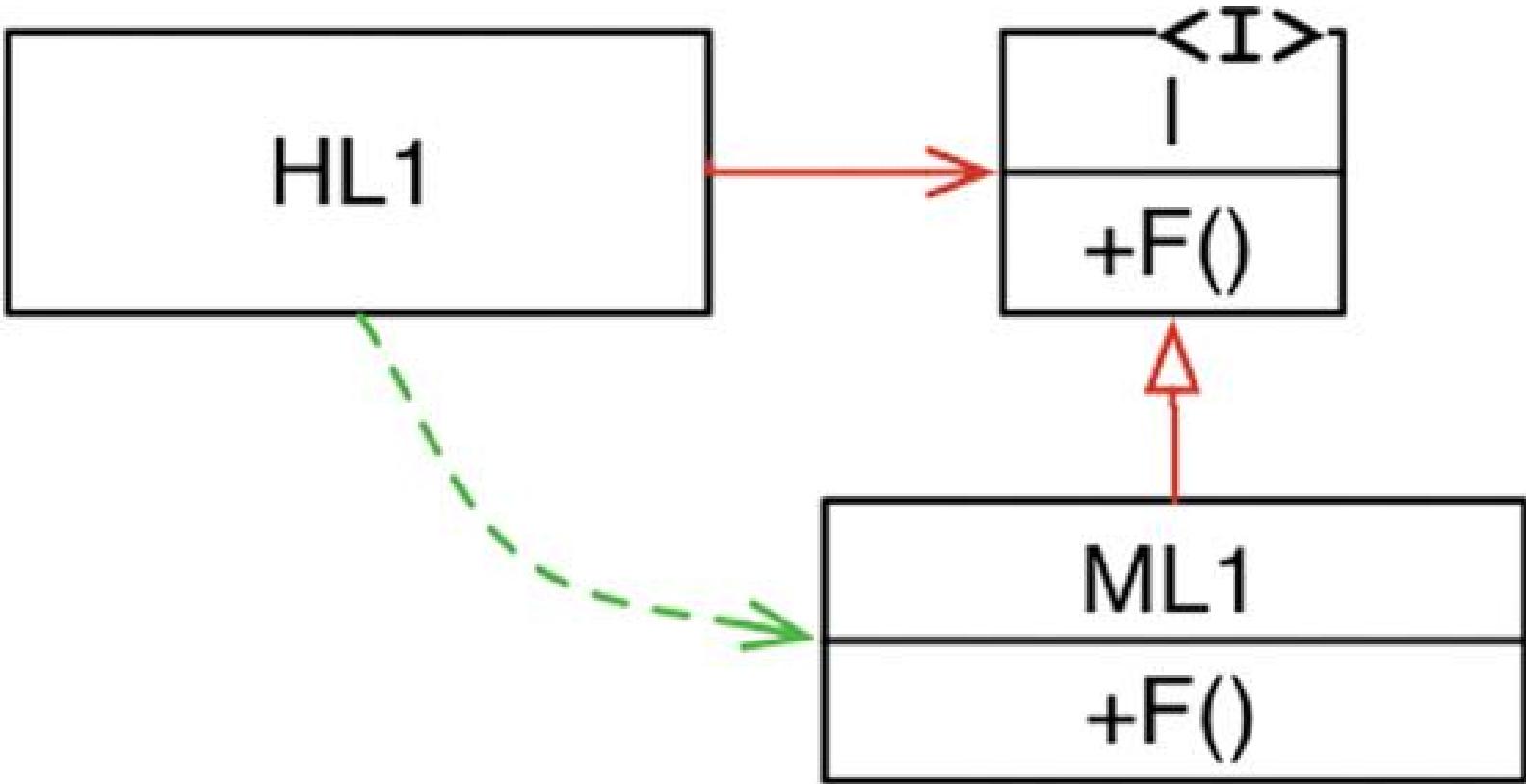


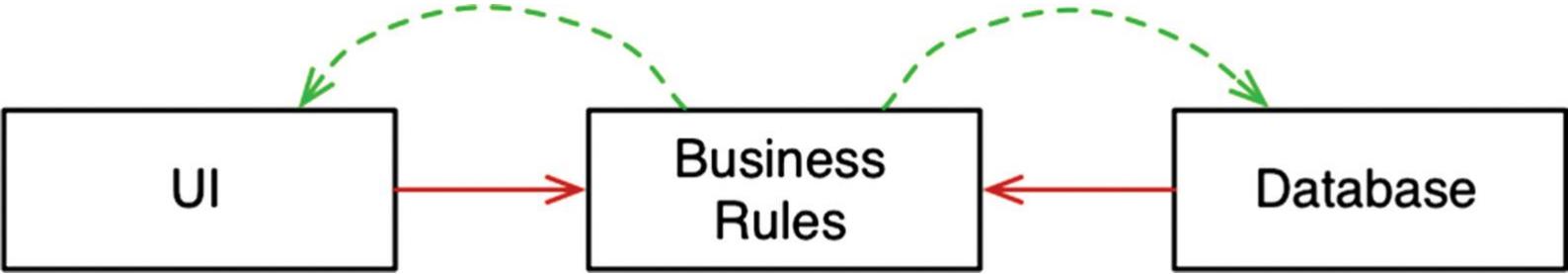






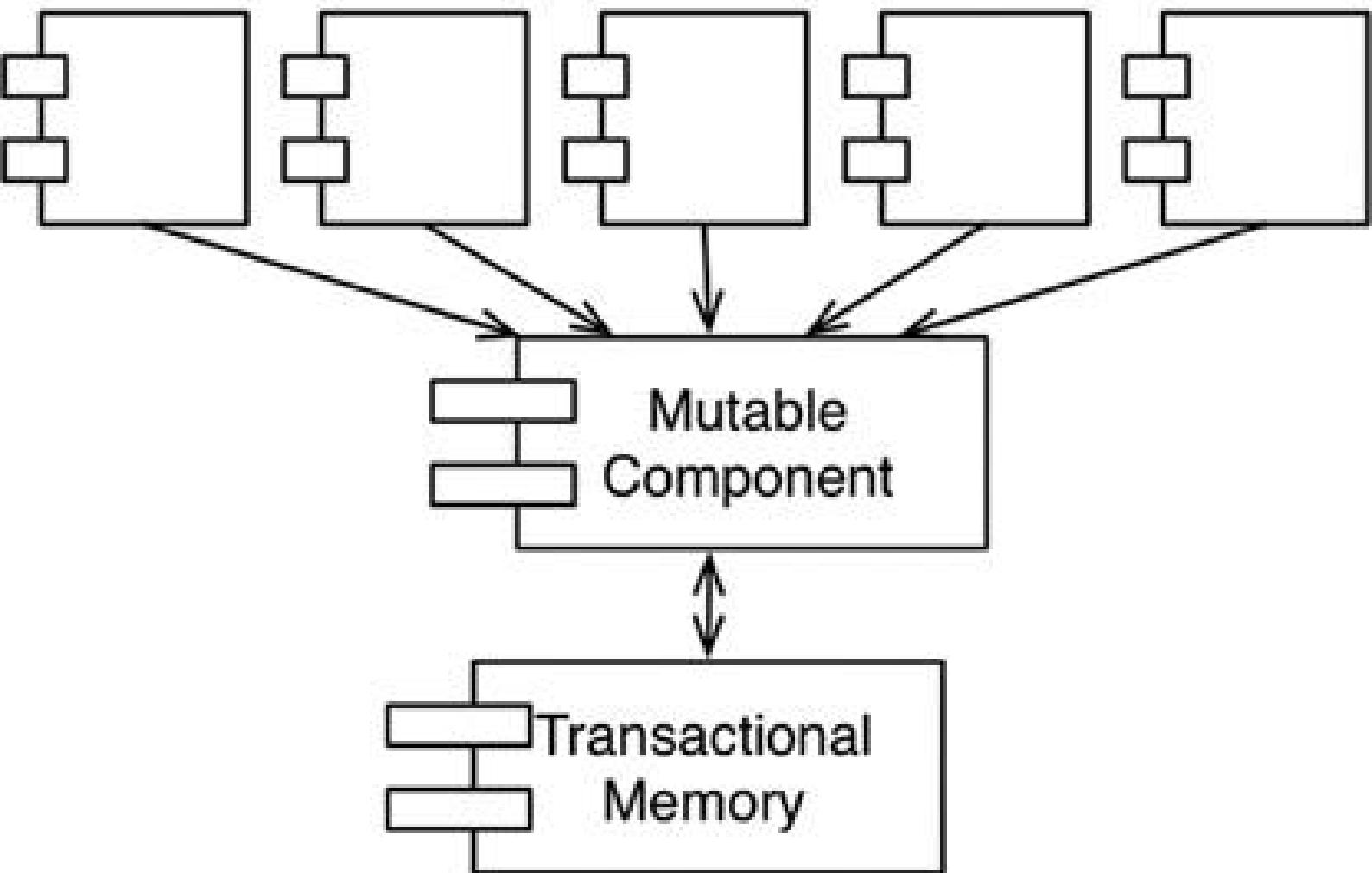


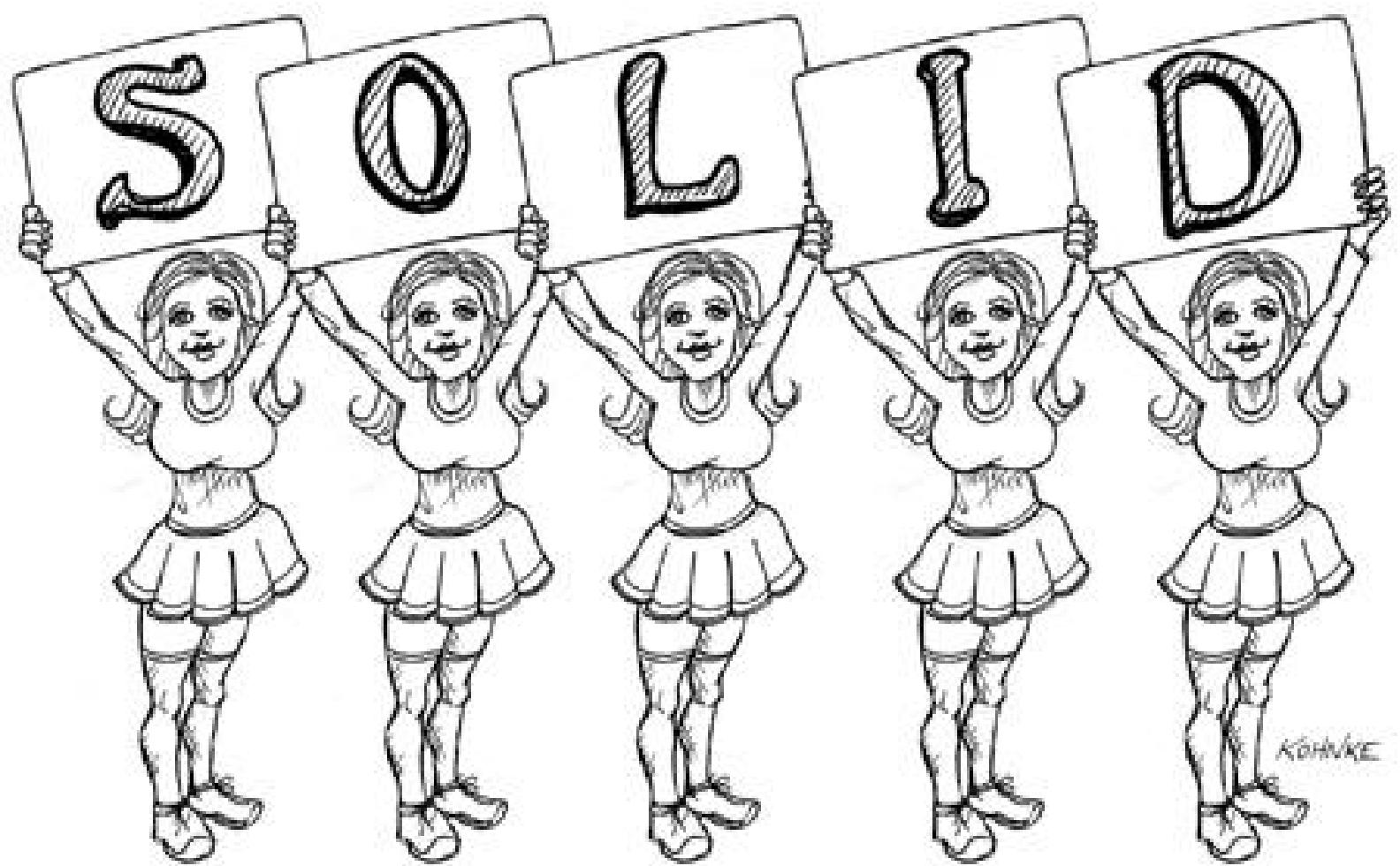




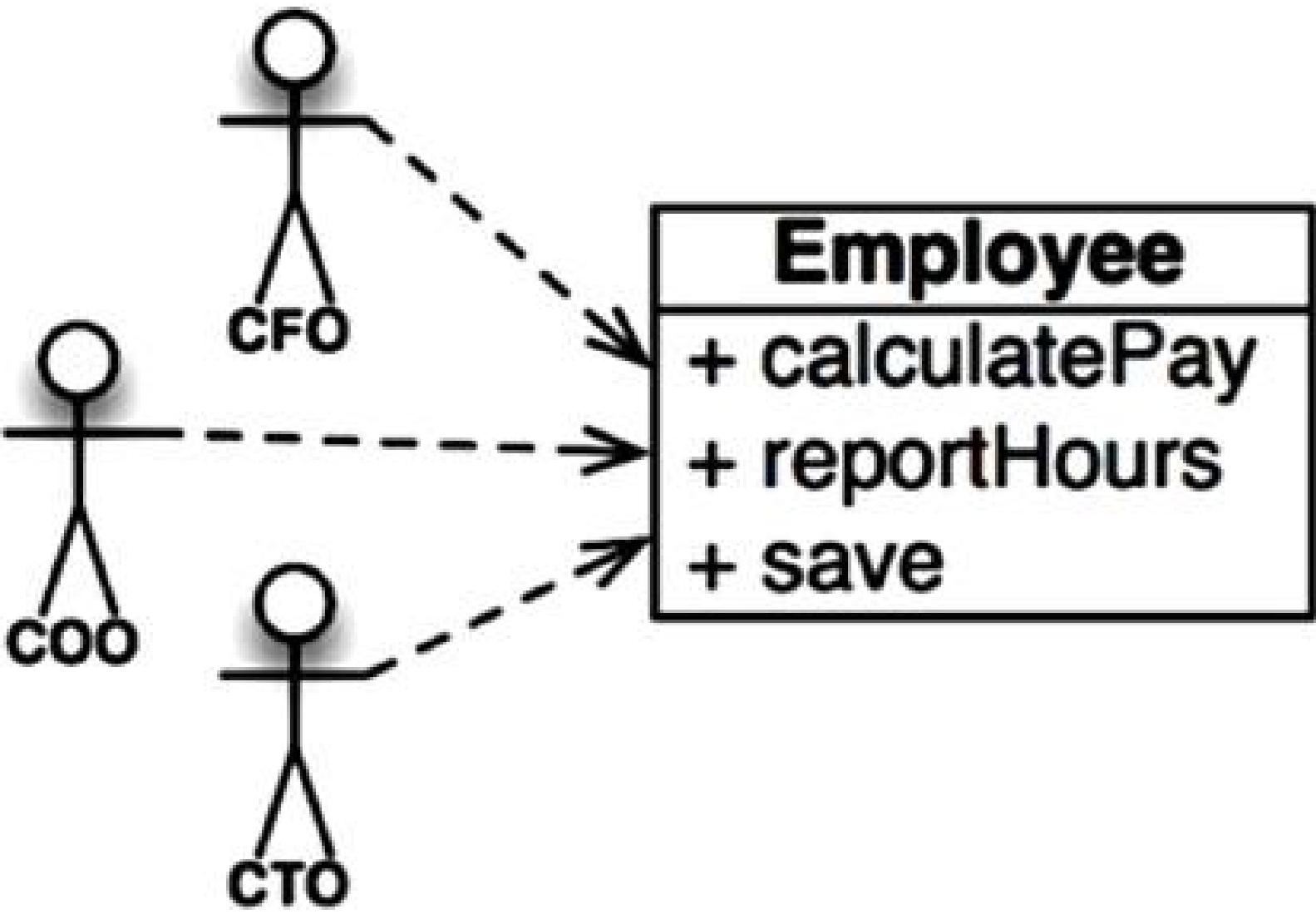


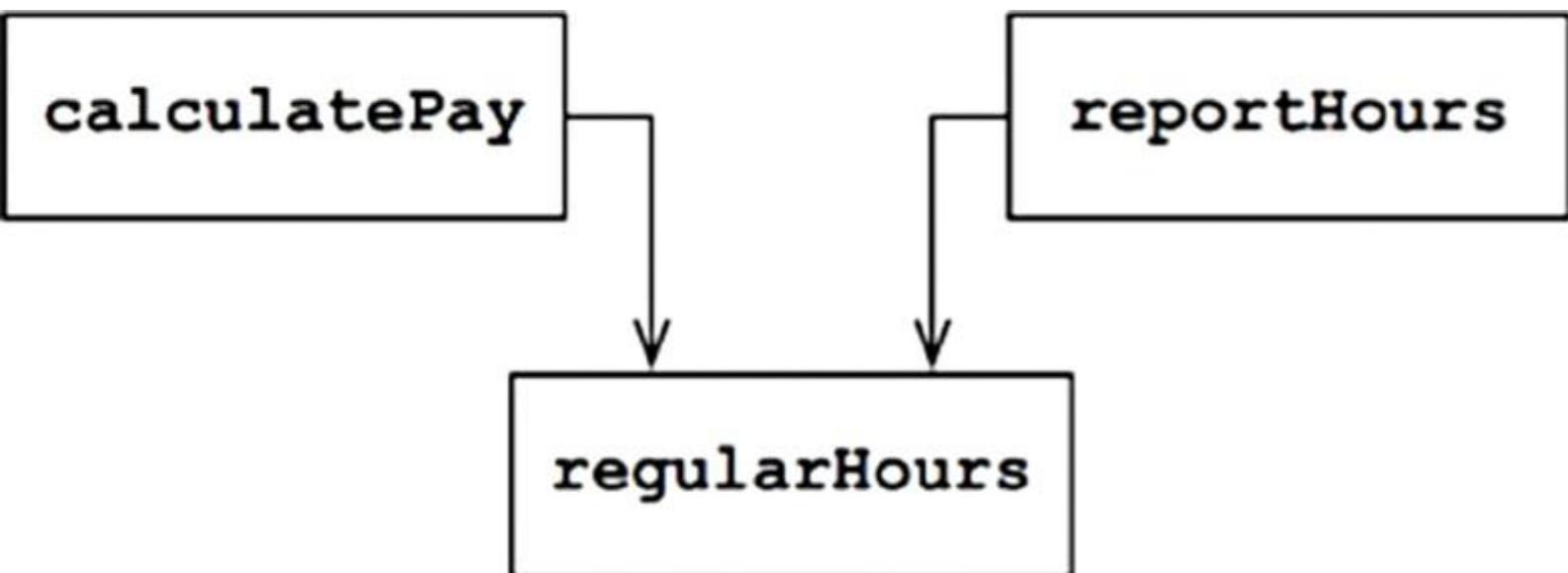
Immutable Components









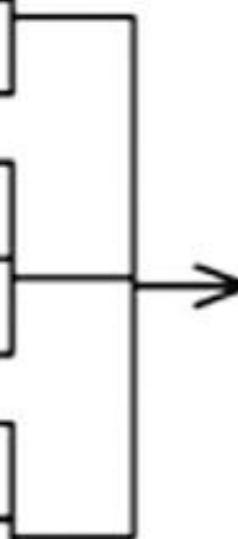


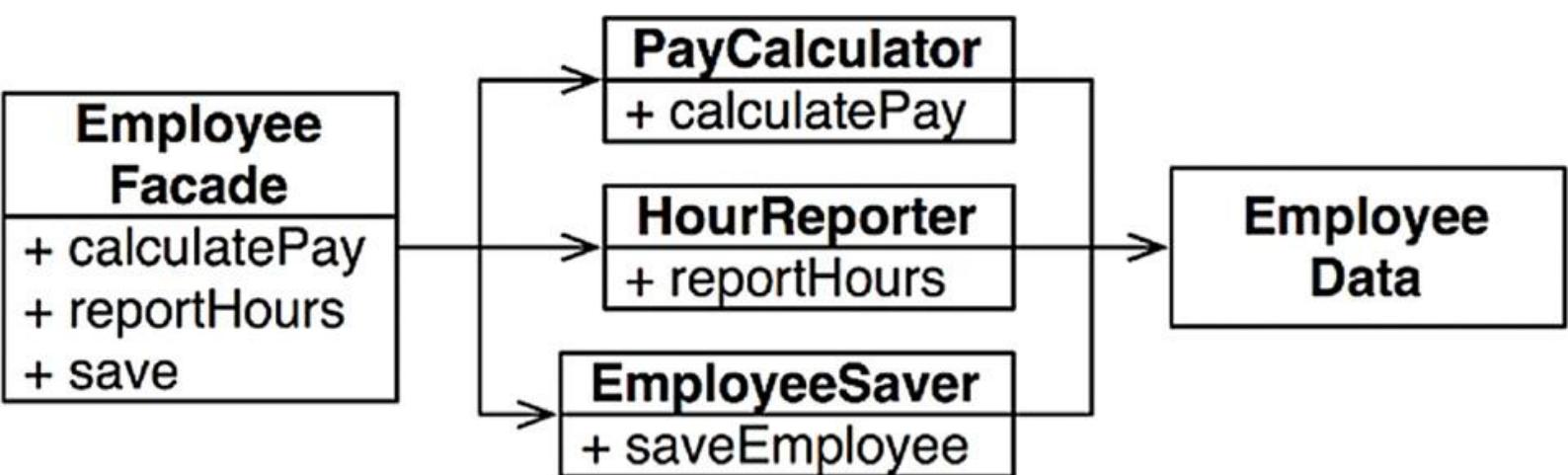
PayCalculator
+ calculatePay

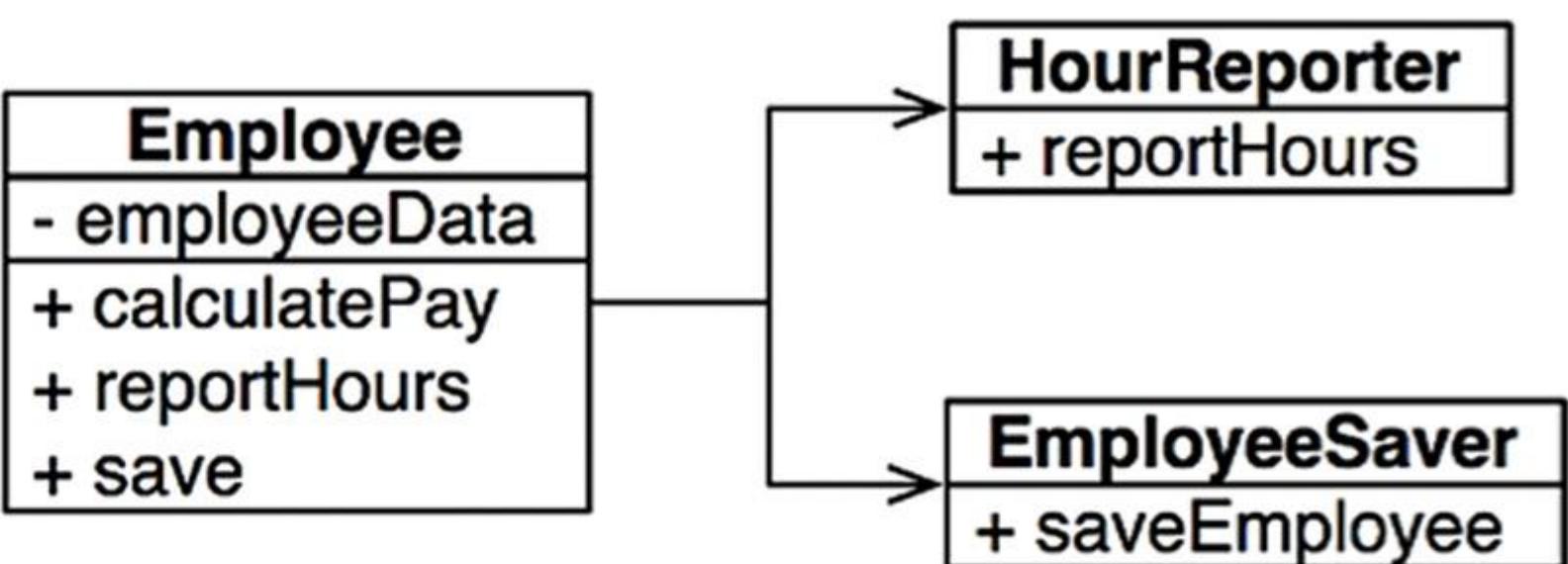
HourReporter
+ reportHours

EmployeeSaver
+ saveEmployee

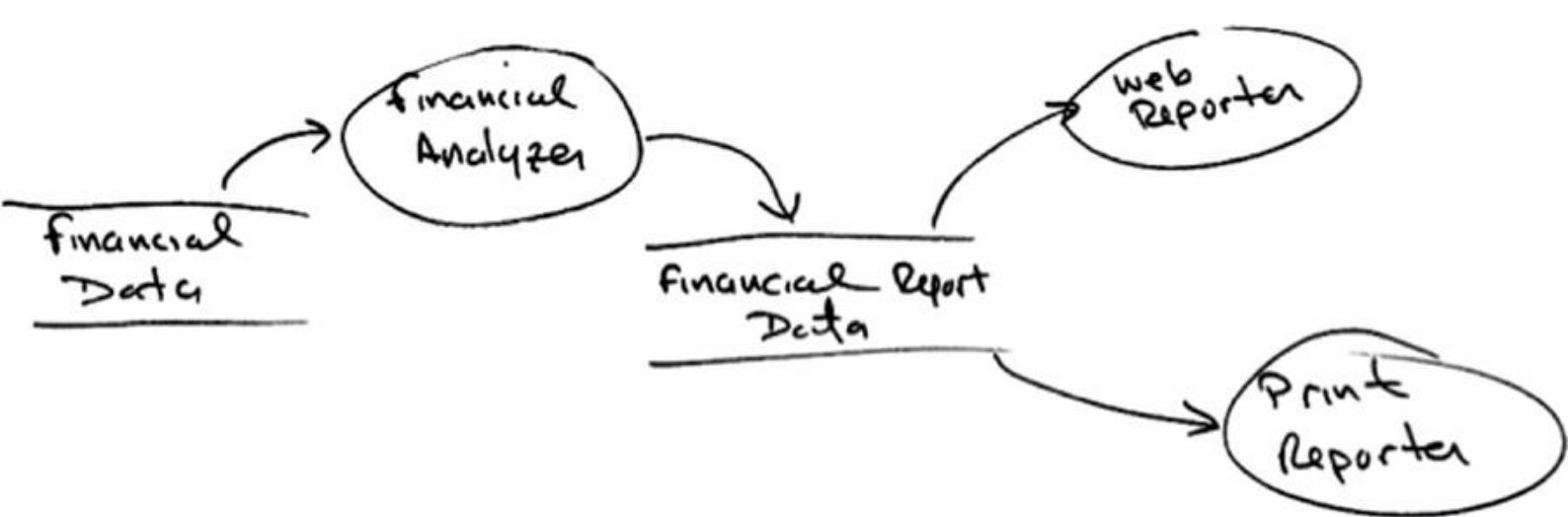
**Employee
Data**

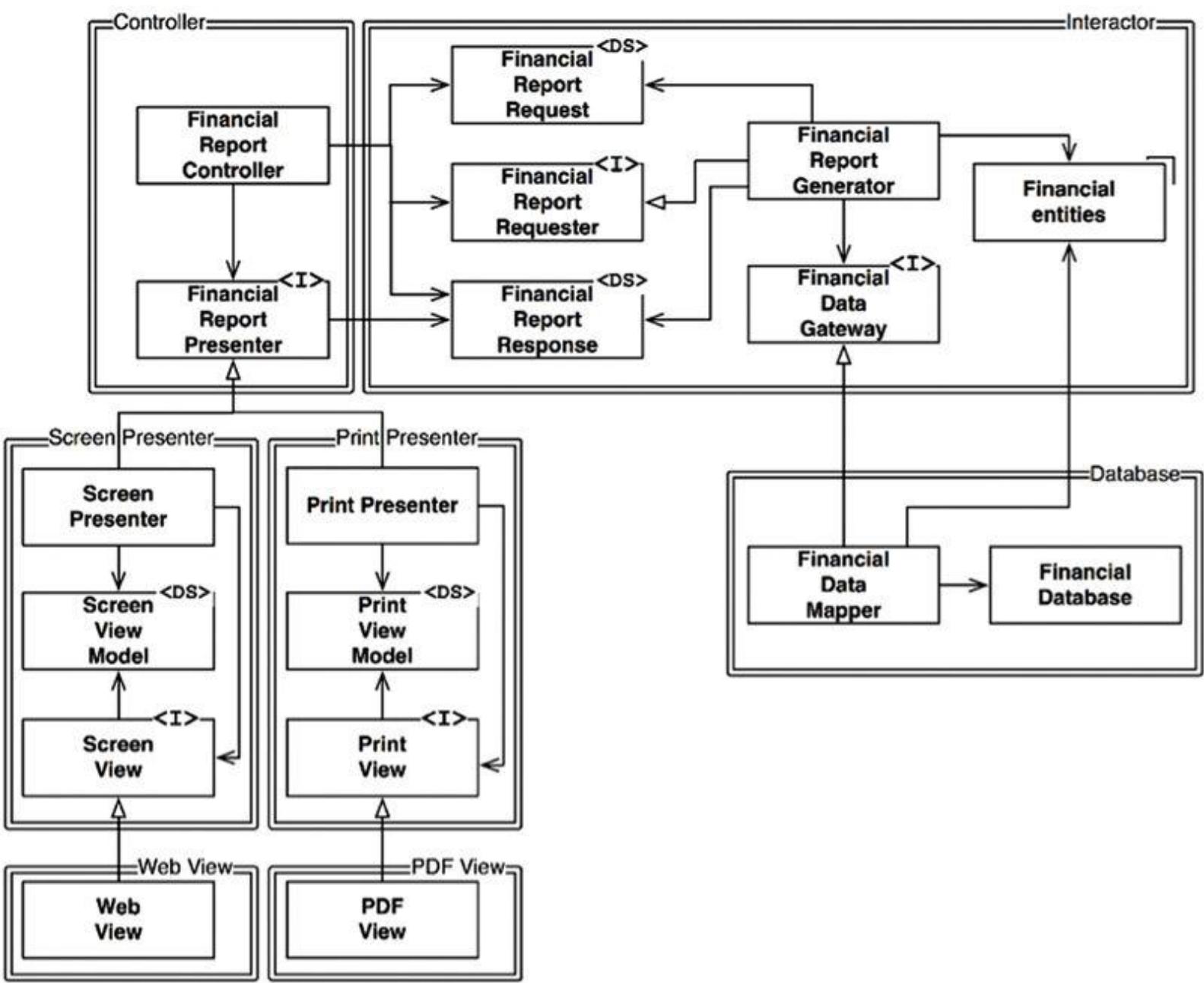


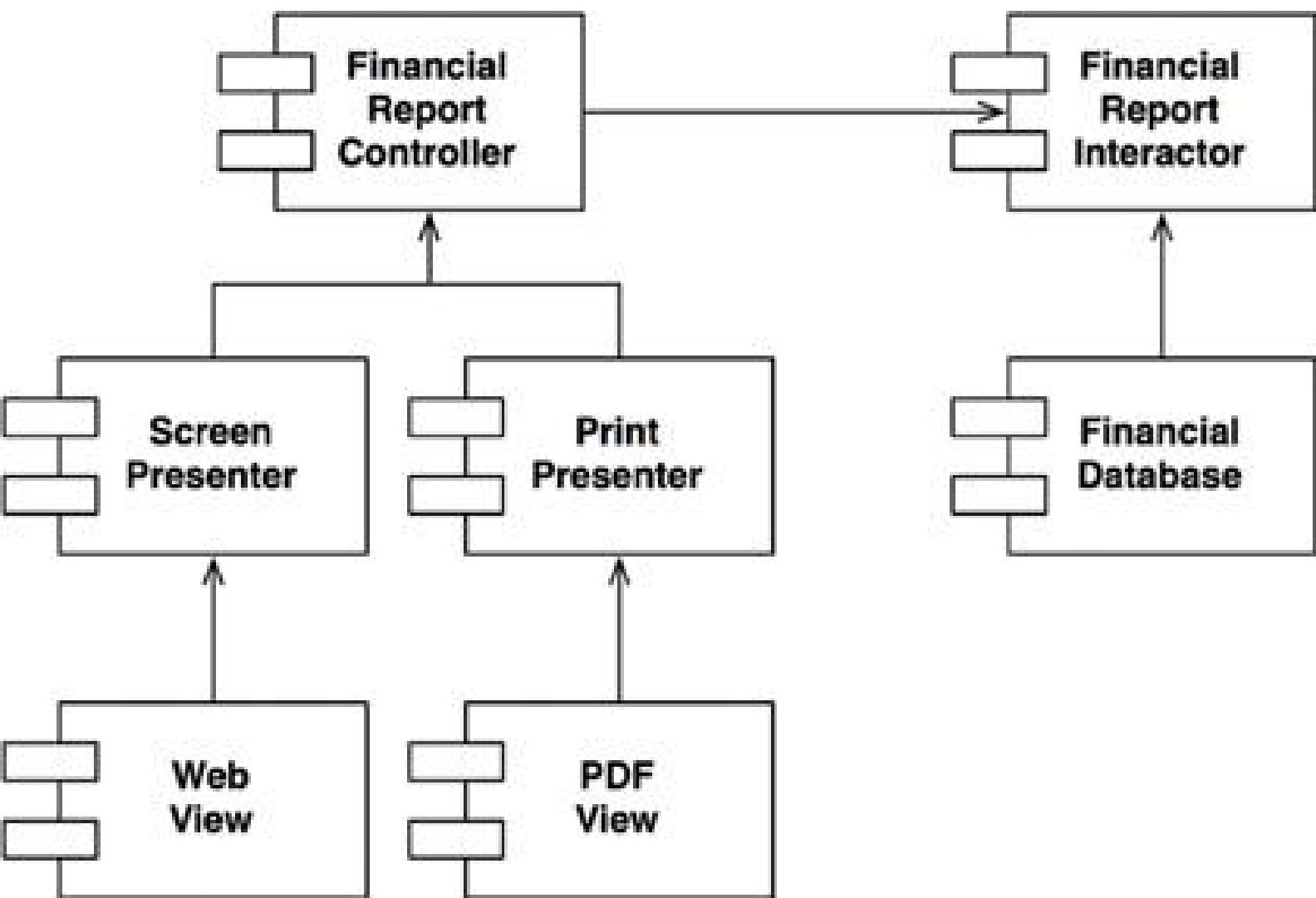




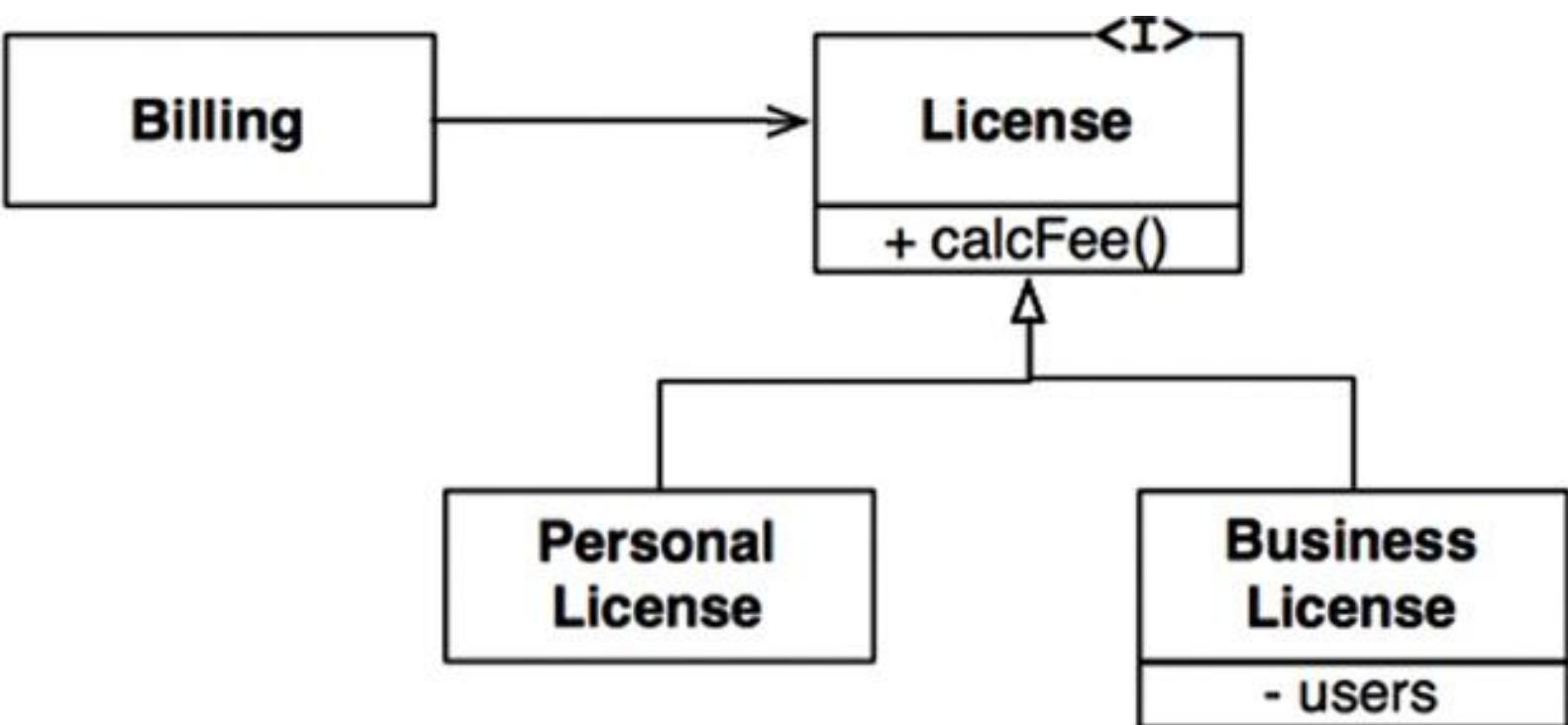


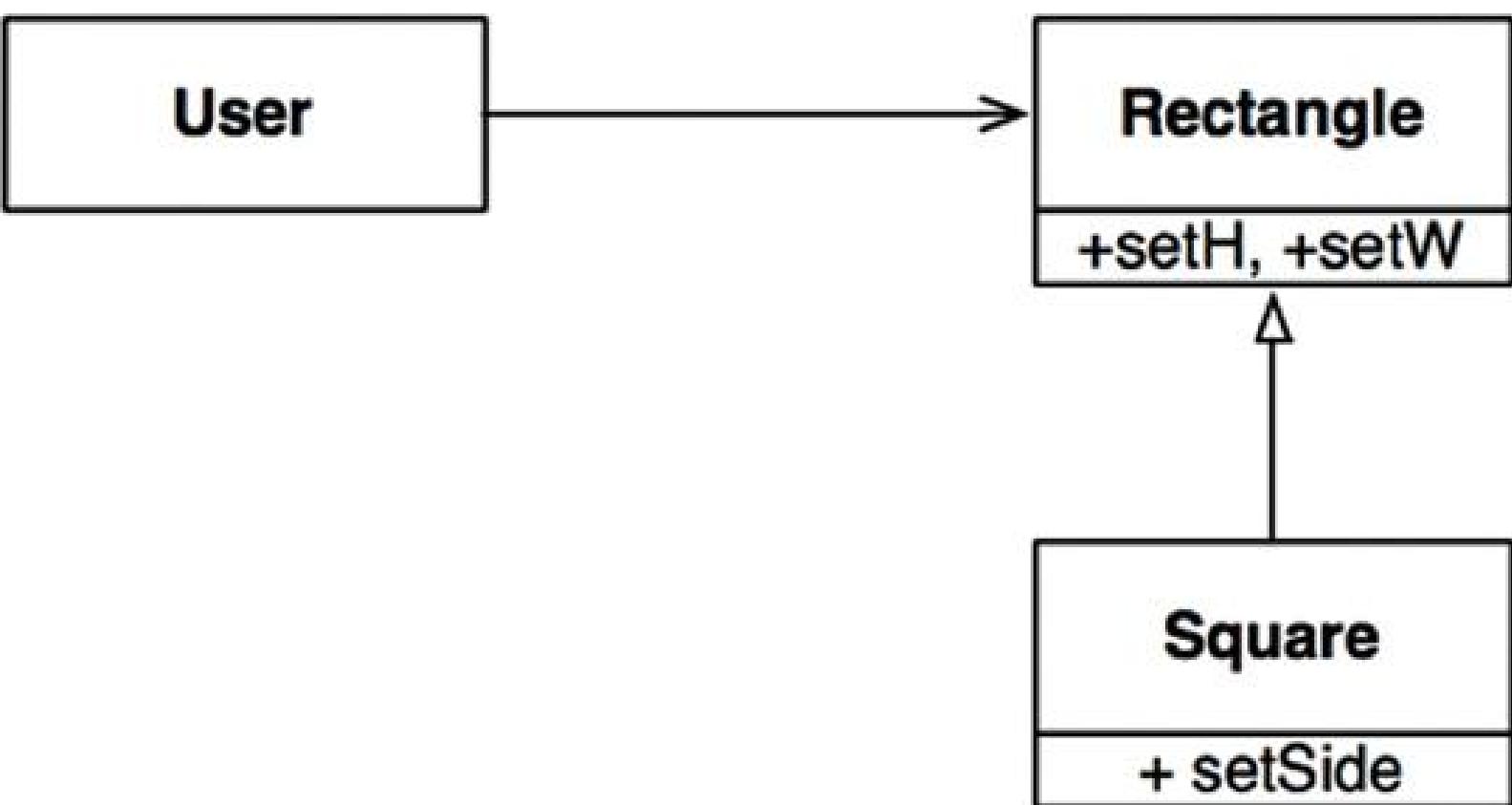














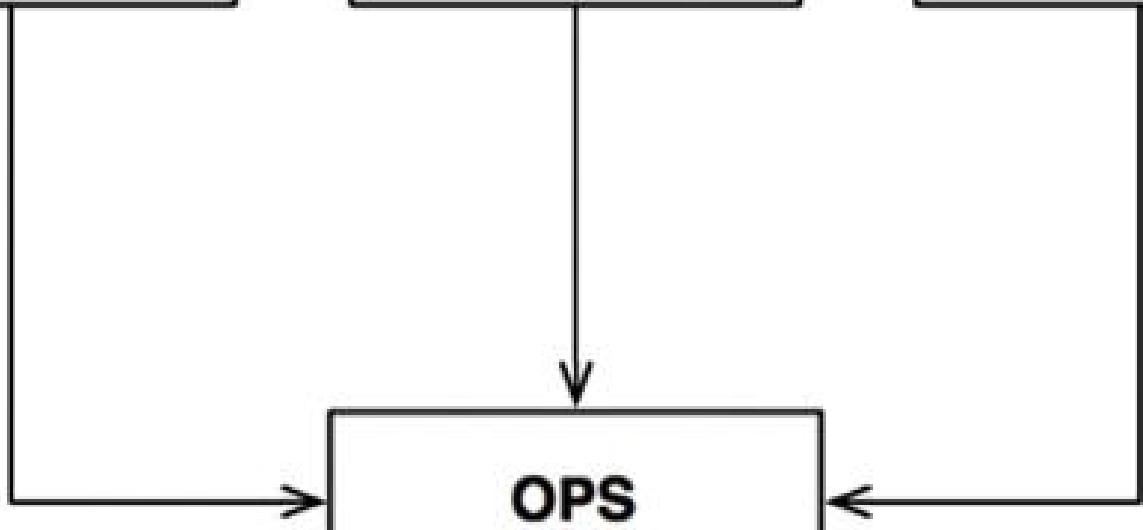
User1

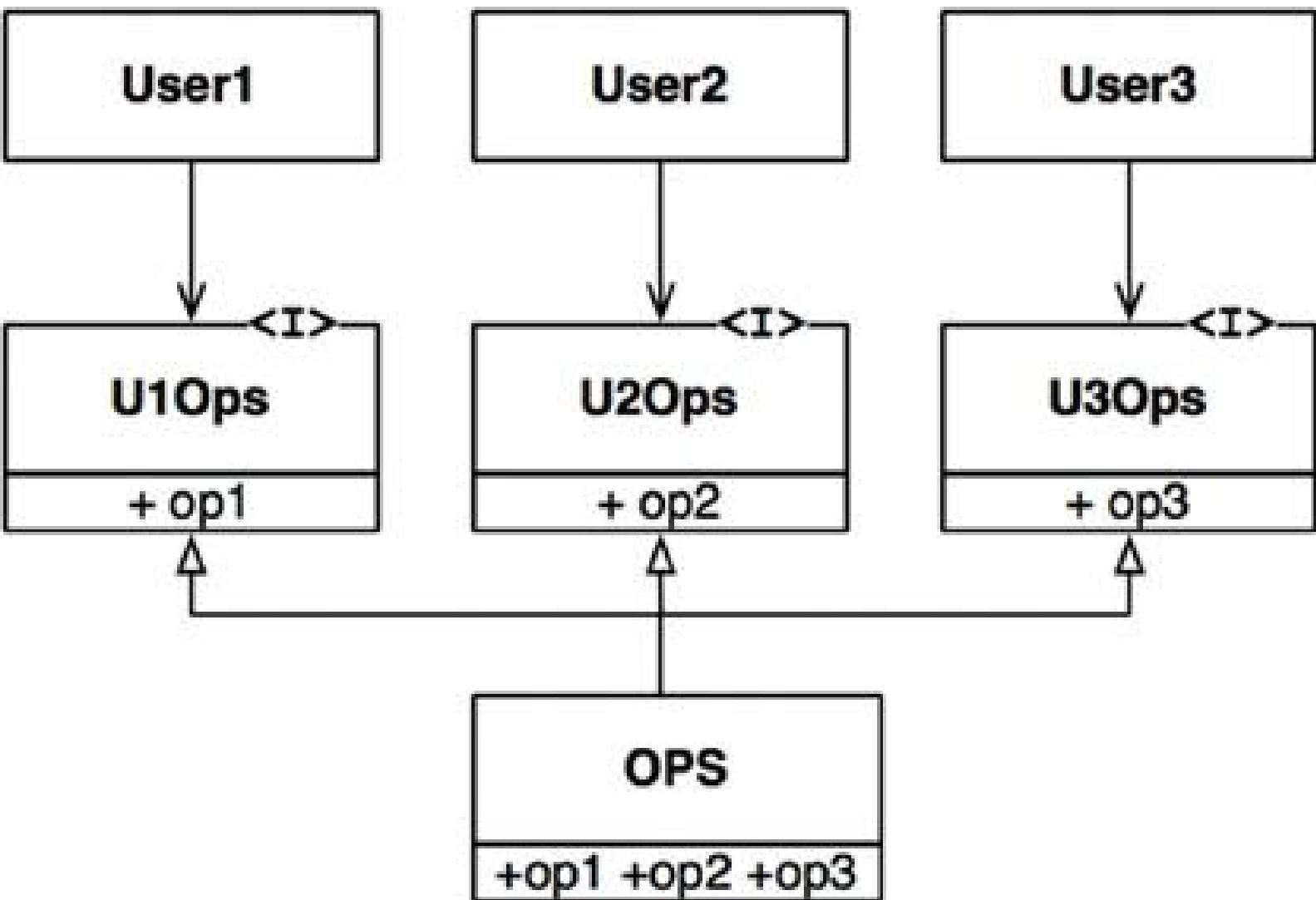
User2

User3

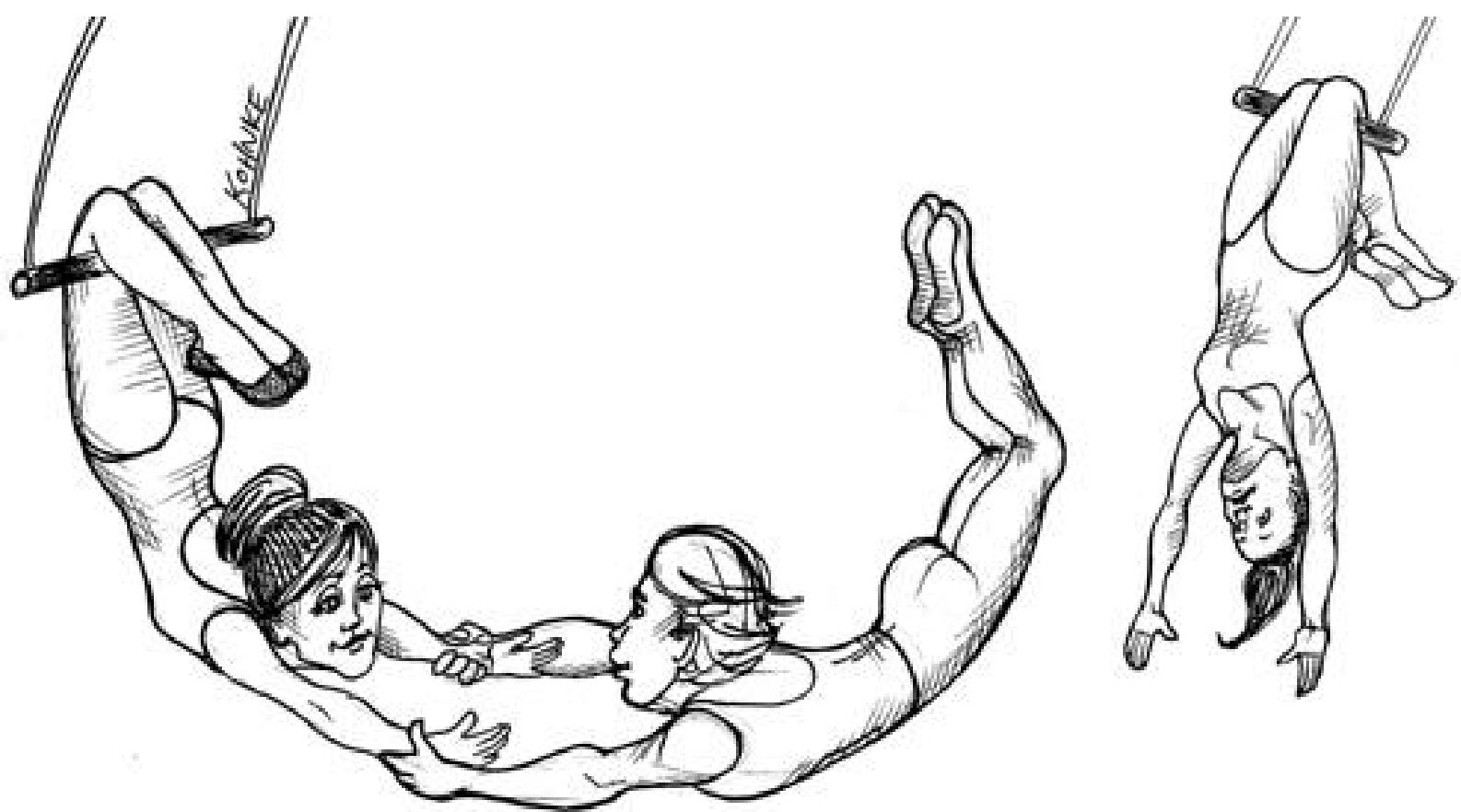
OPS

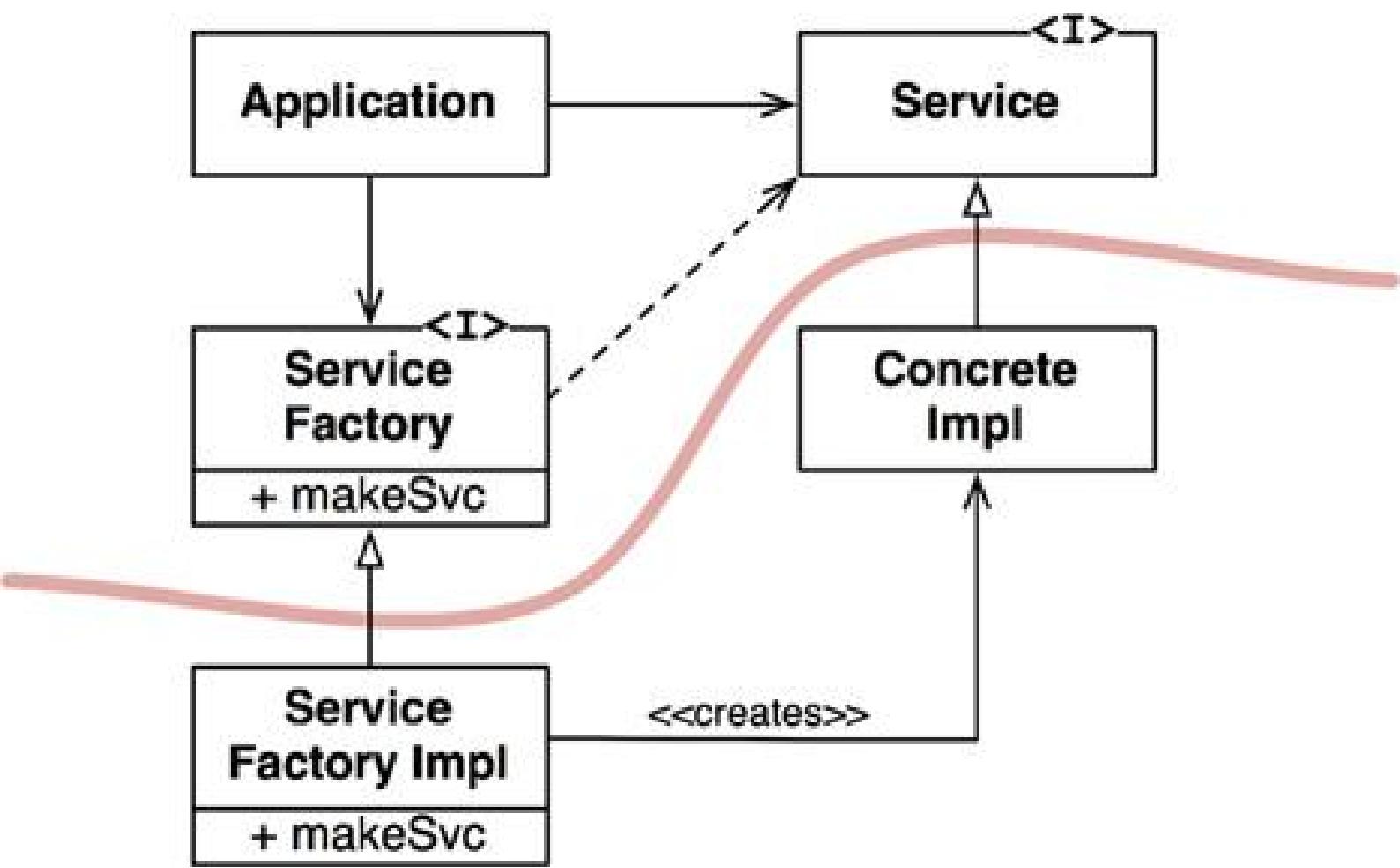
+op1 +op2 +op3

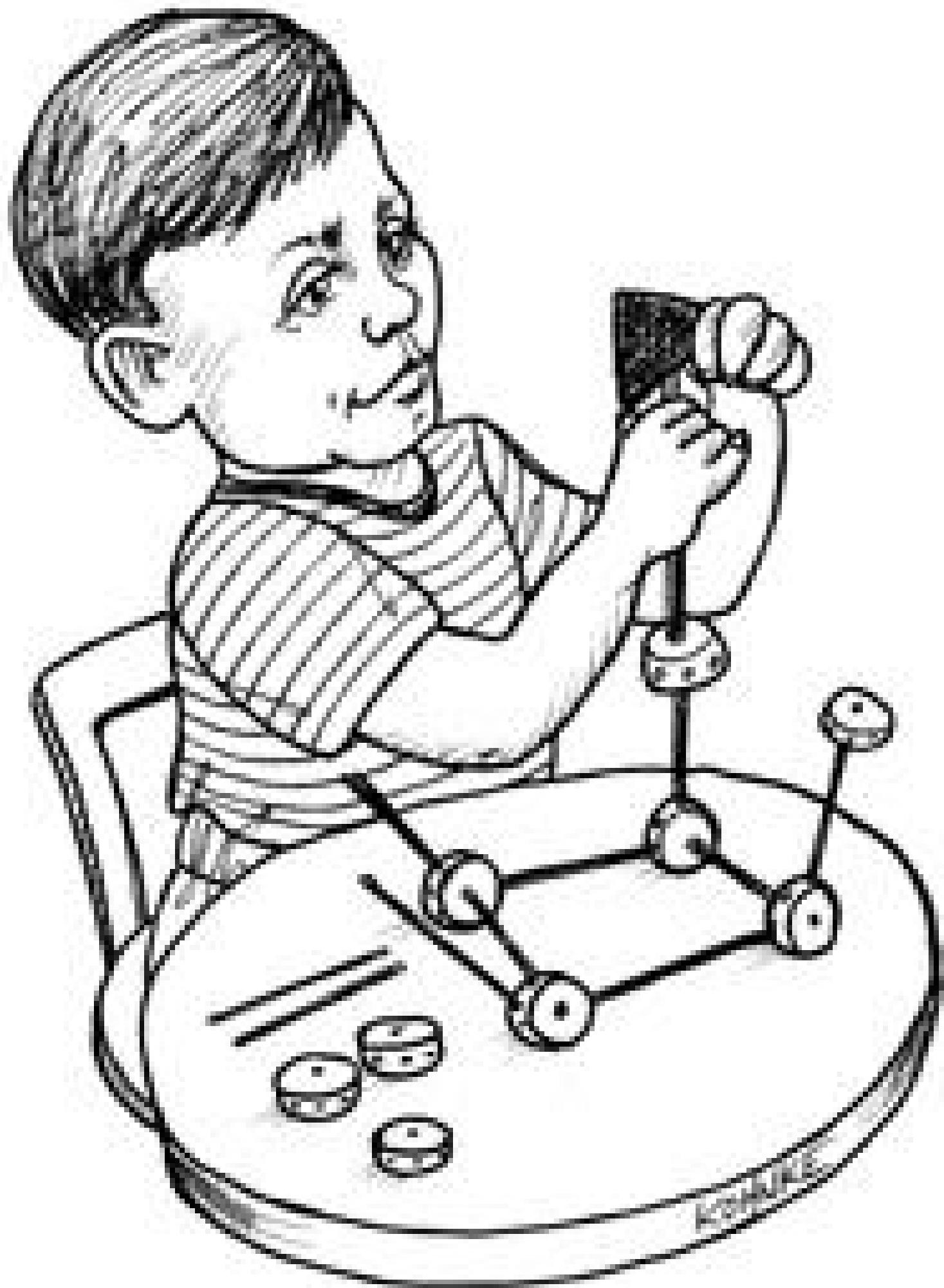








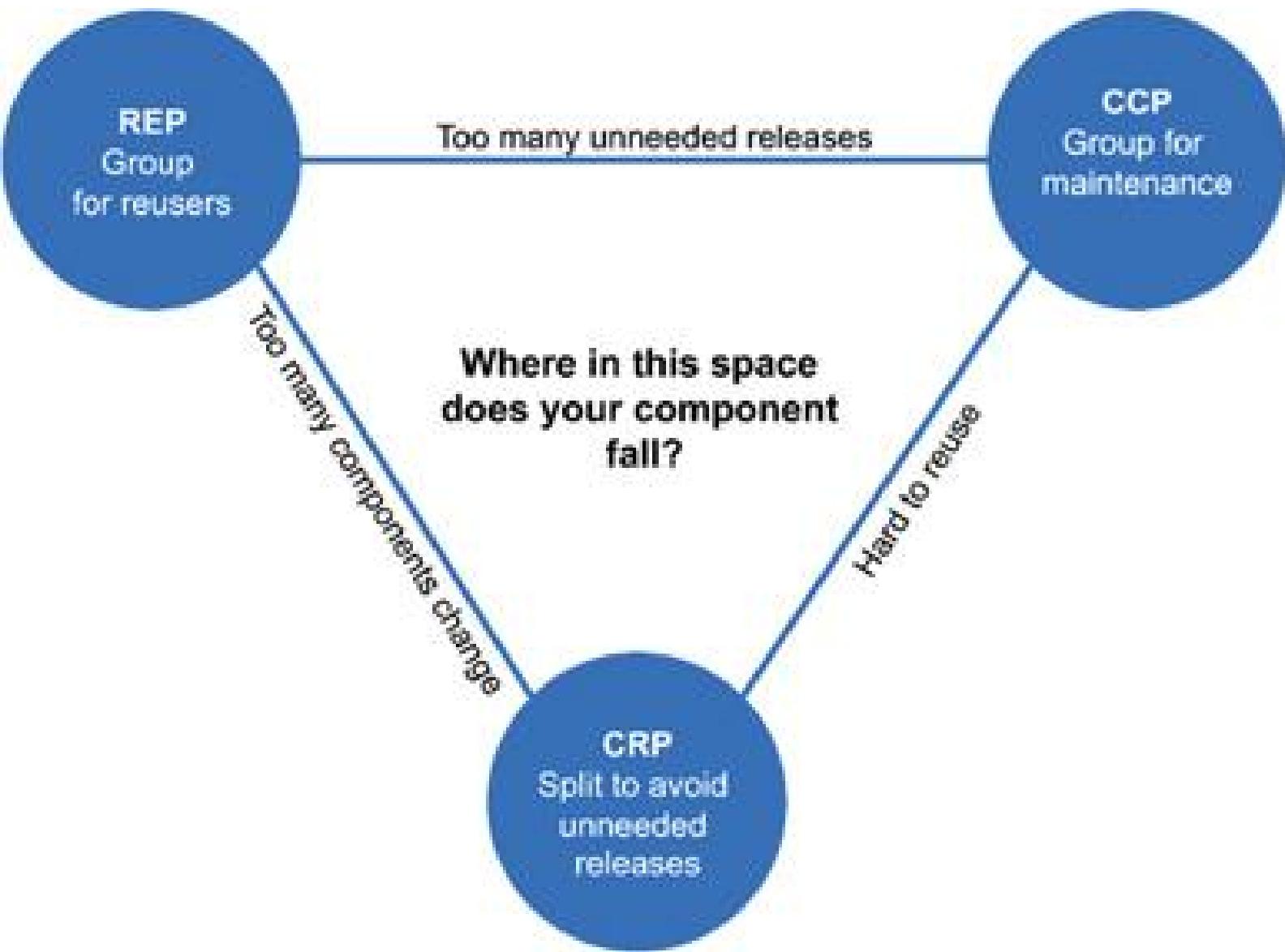


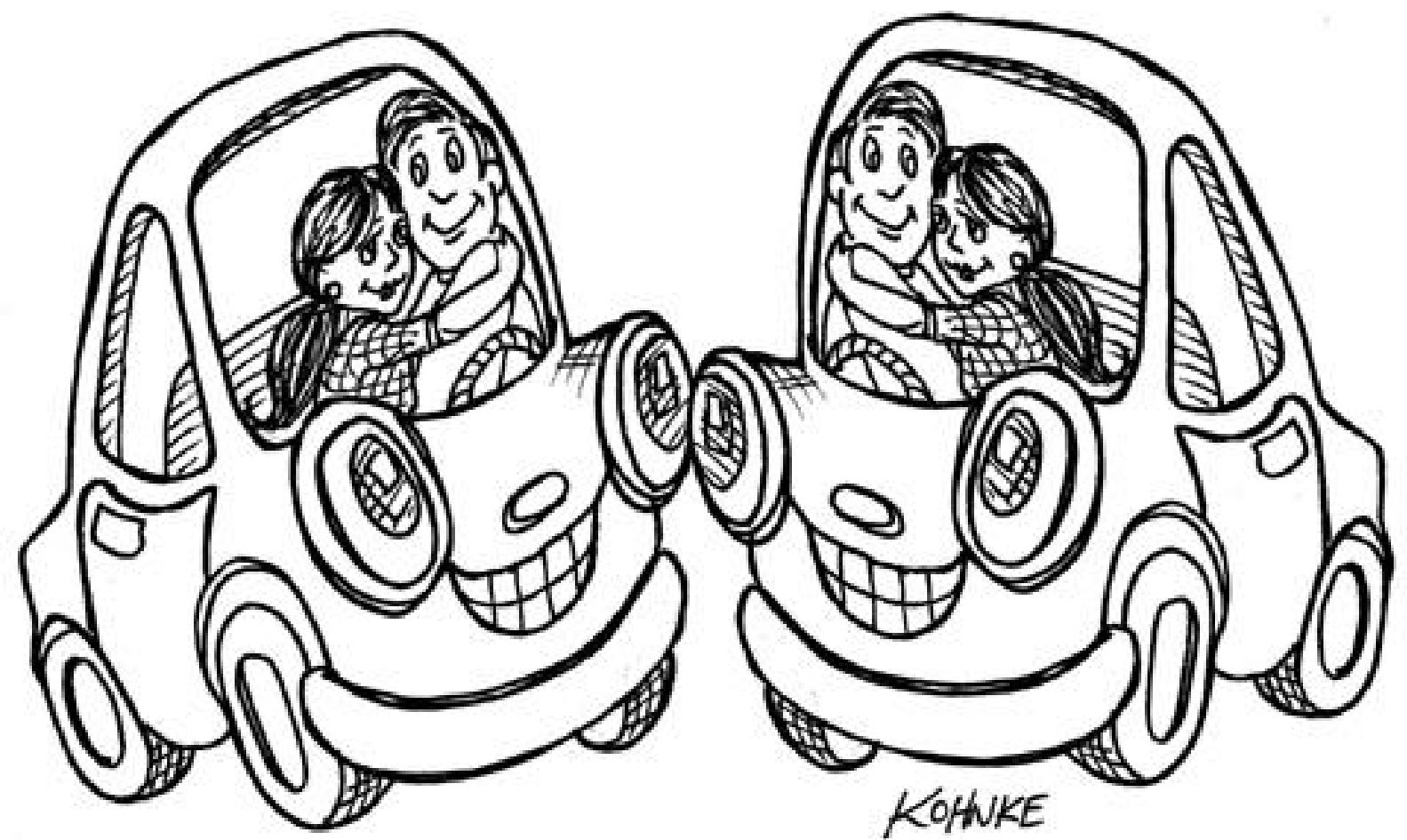


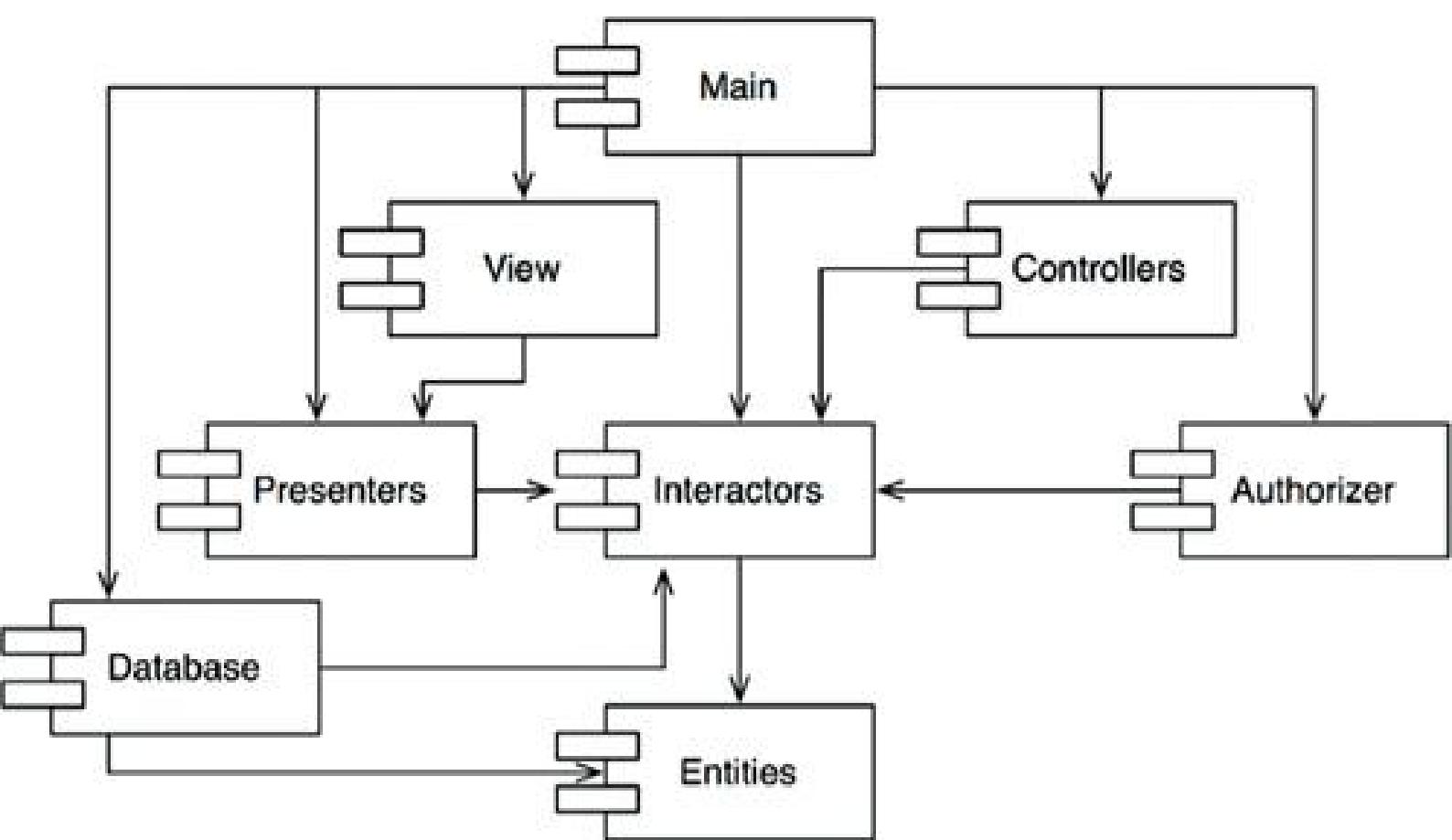
	x000-x177	x200-x377	x400-x577	x600-x777
0000-0777			Application	
1000-1777				
2000-2777				
3000-3777		Function Library		
4000-4777				
5000-5777				
6000-6777				
7000-7777				

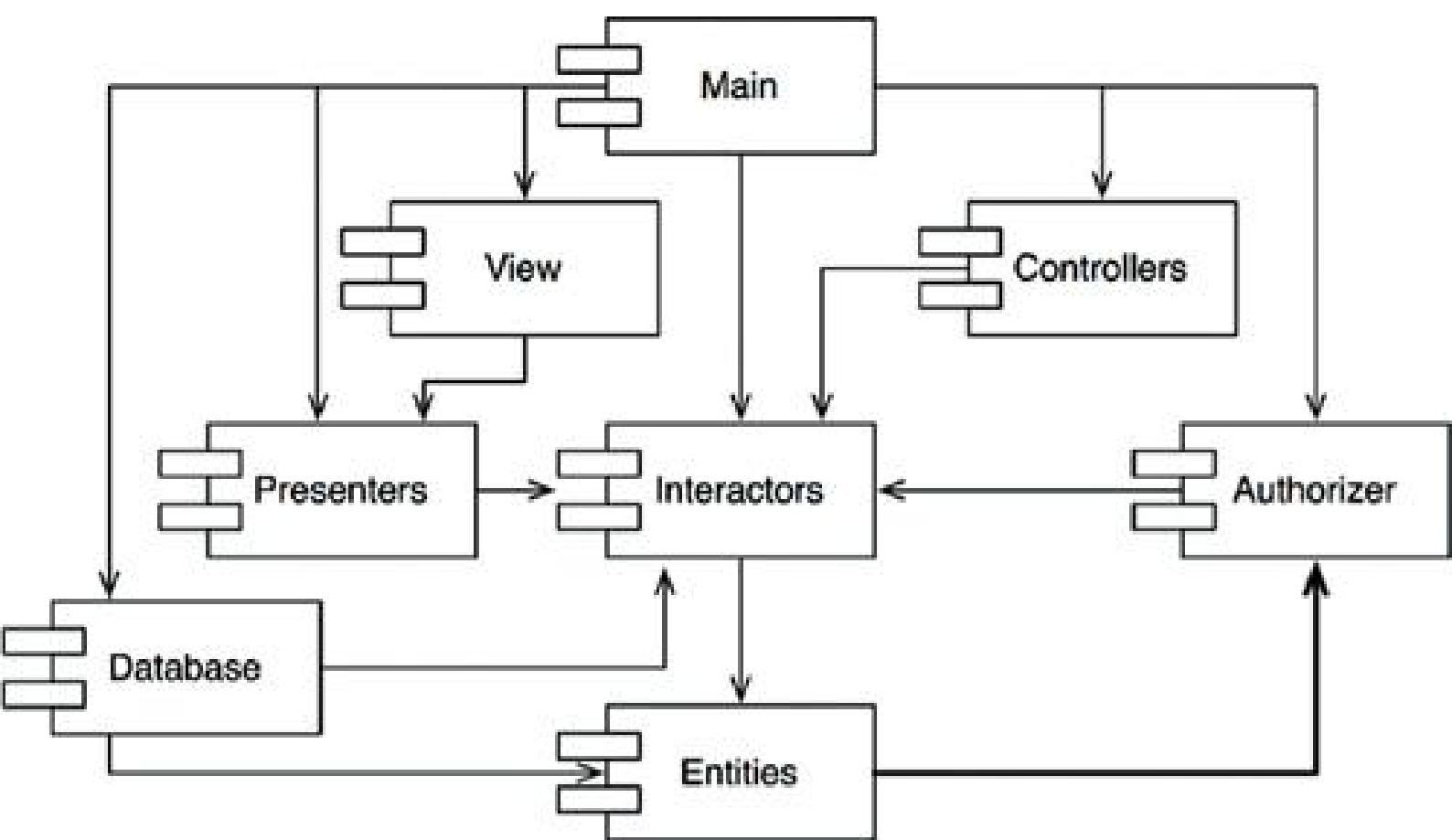
	x000-x177	x200-x377	x400-x577	x600-x777
0000-0777				
1000-1777		Application...		
2000-2777				
3000-3777		Function Library		
4000-4777				
5000-5777		...Application		
6000-6777				
7000-7777				

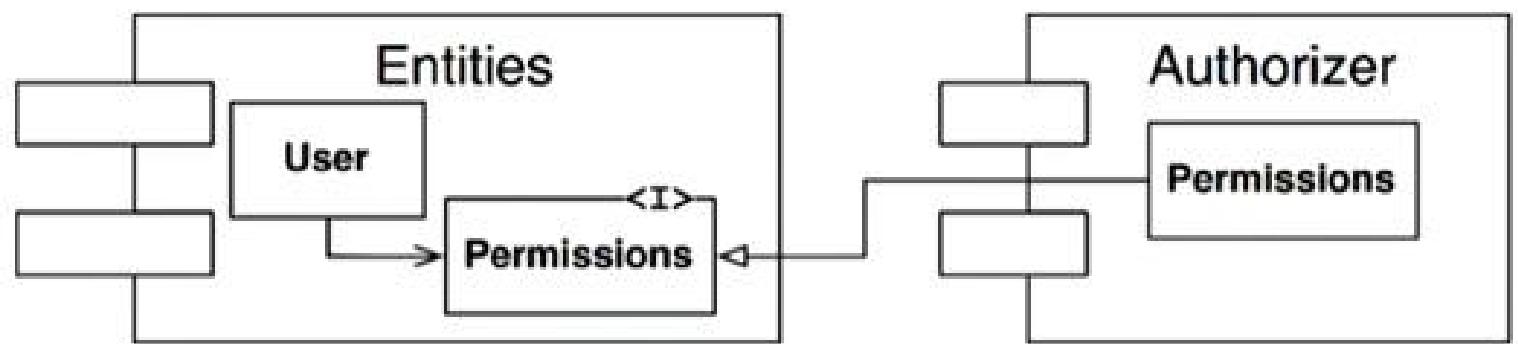
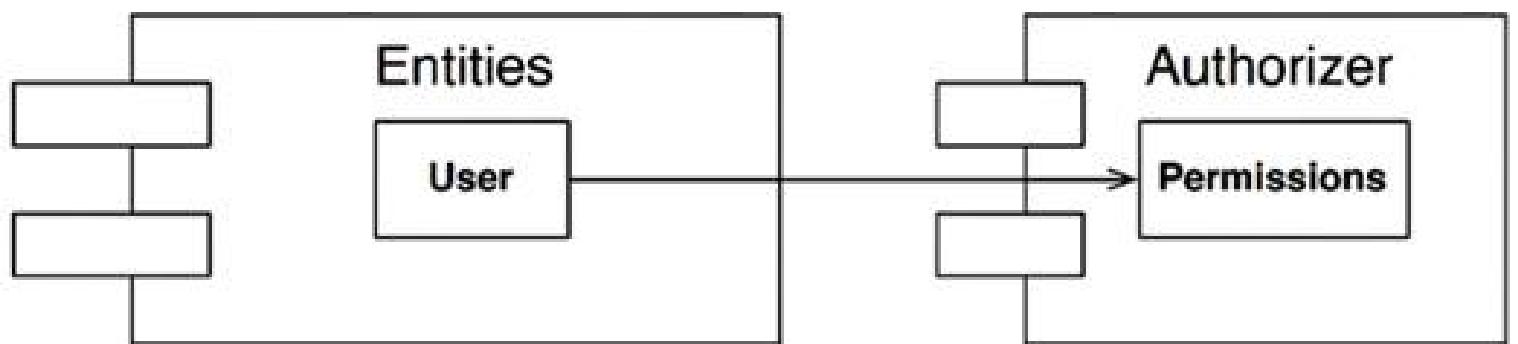


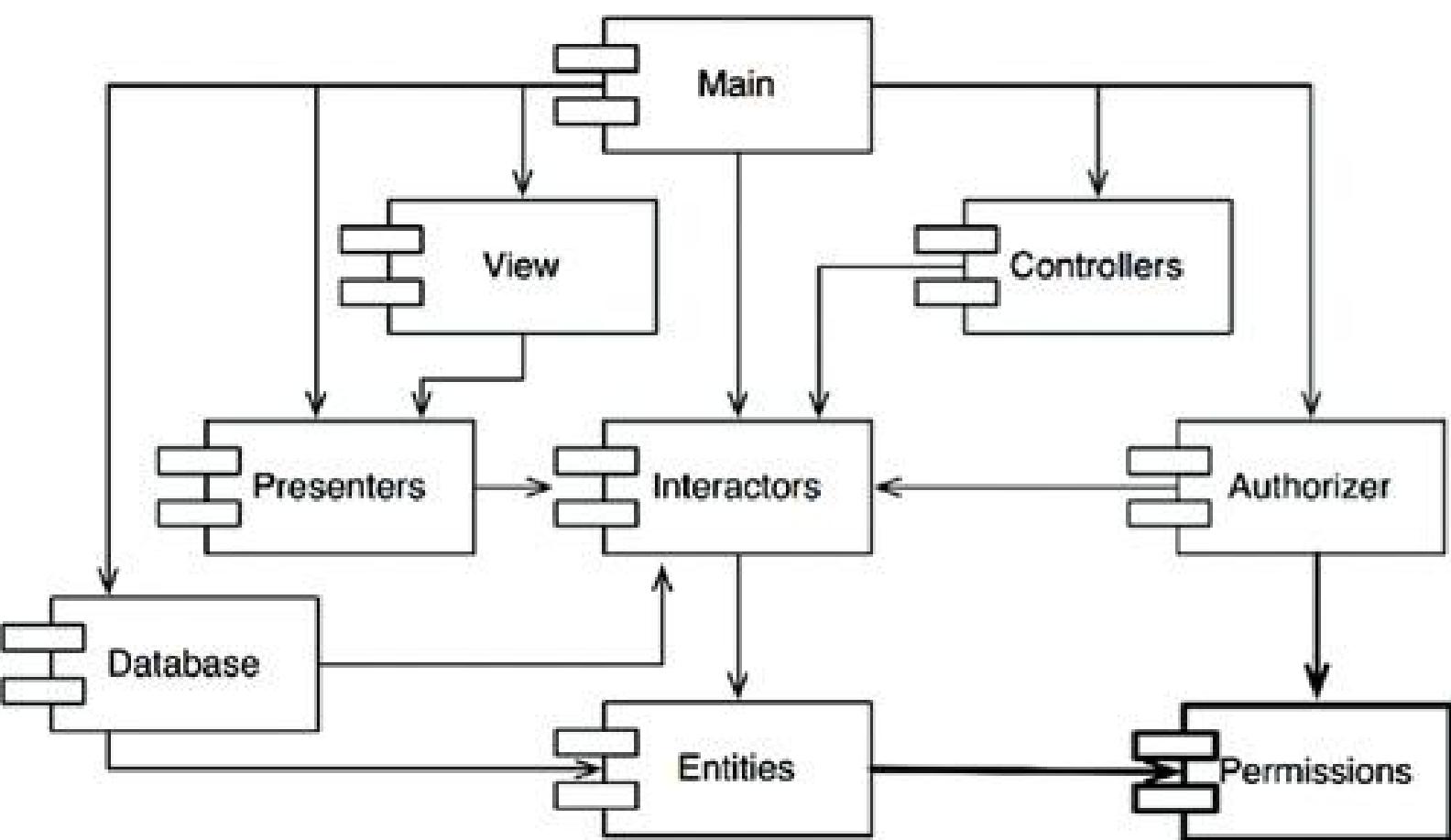


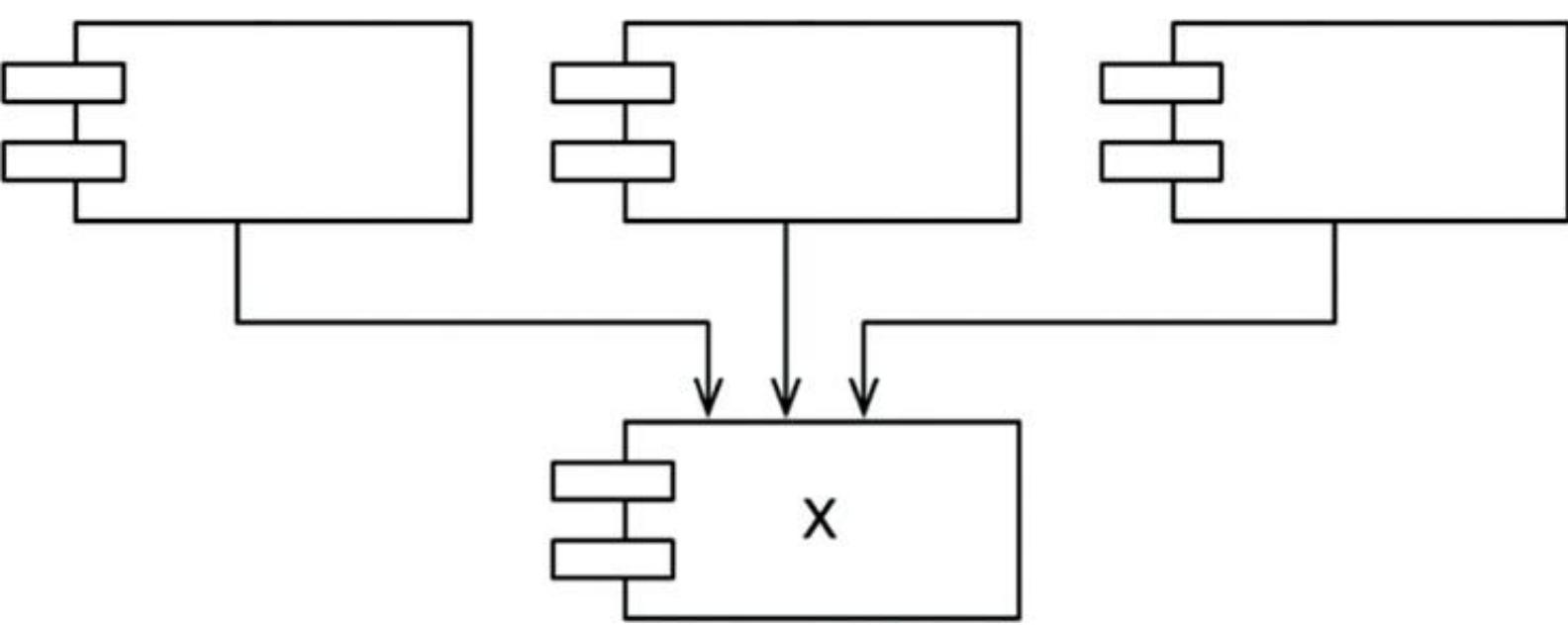


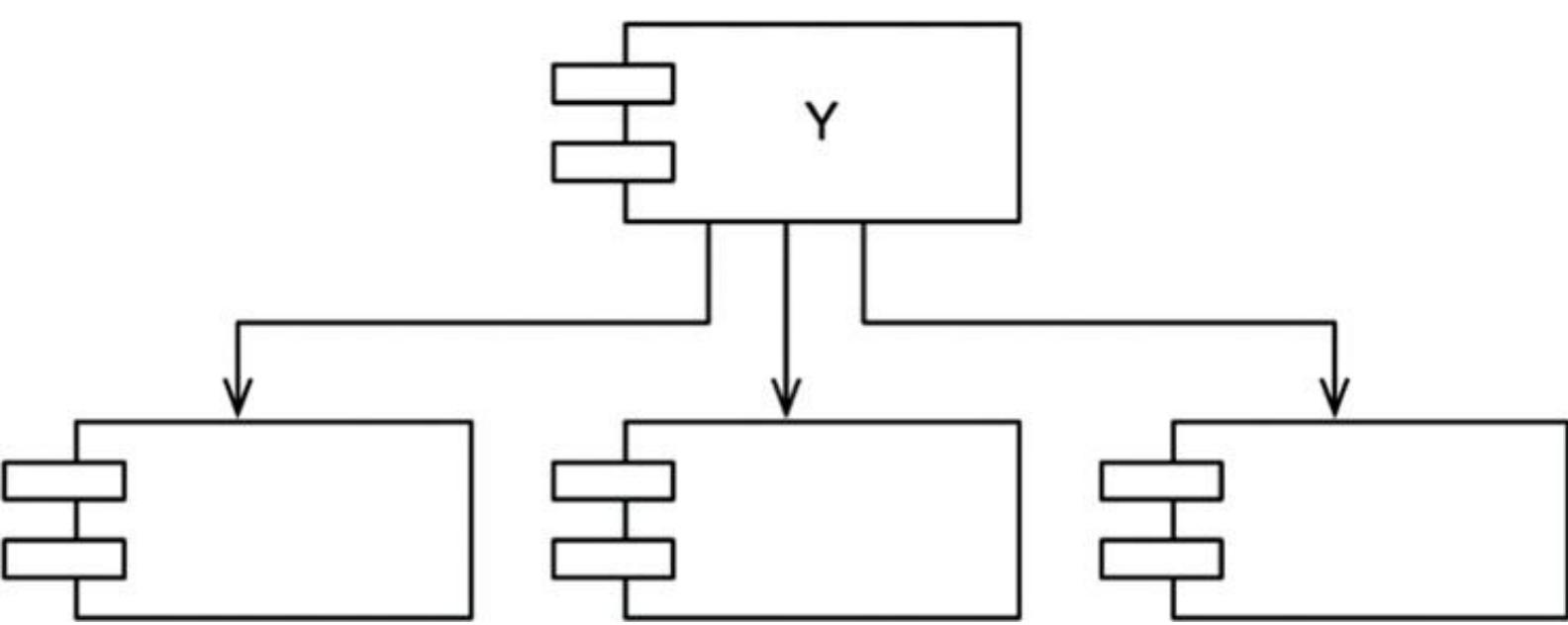


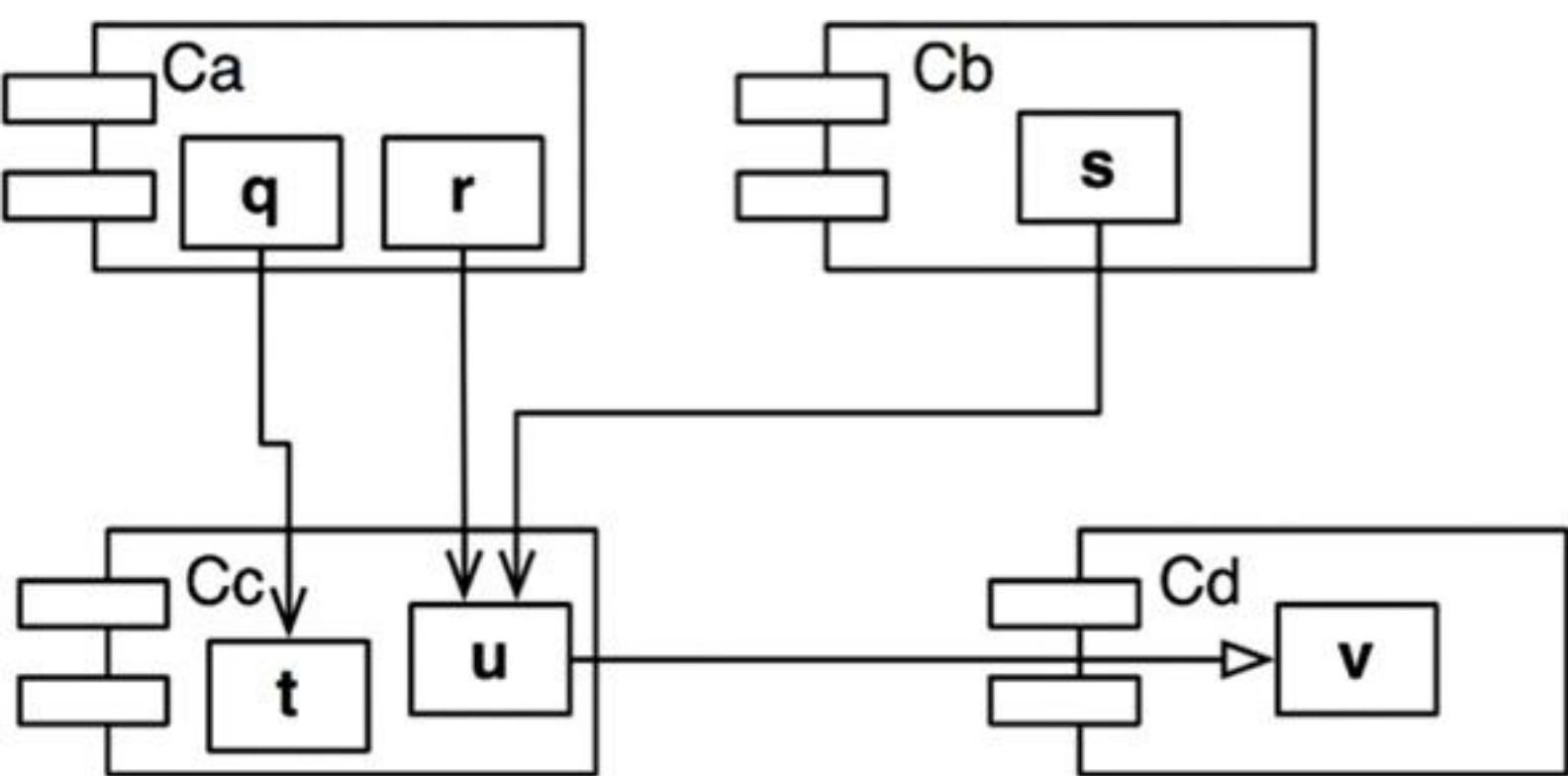


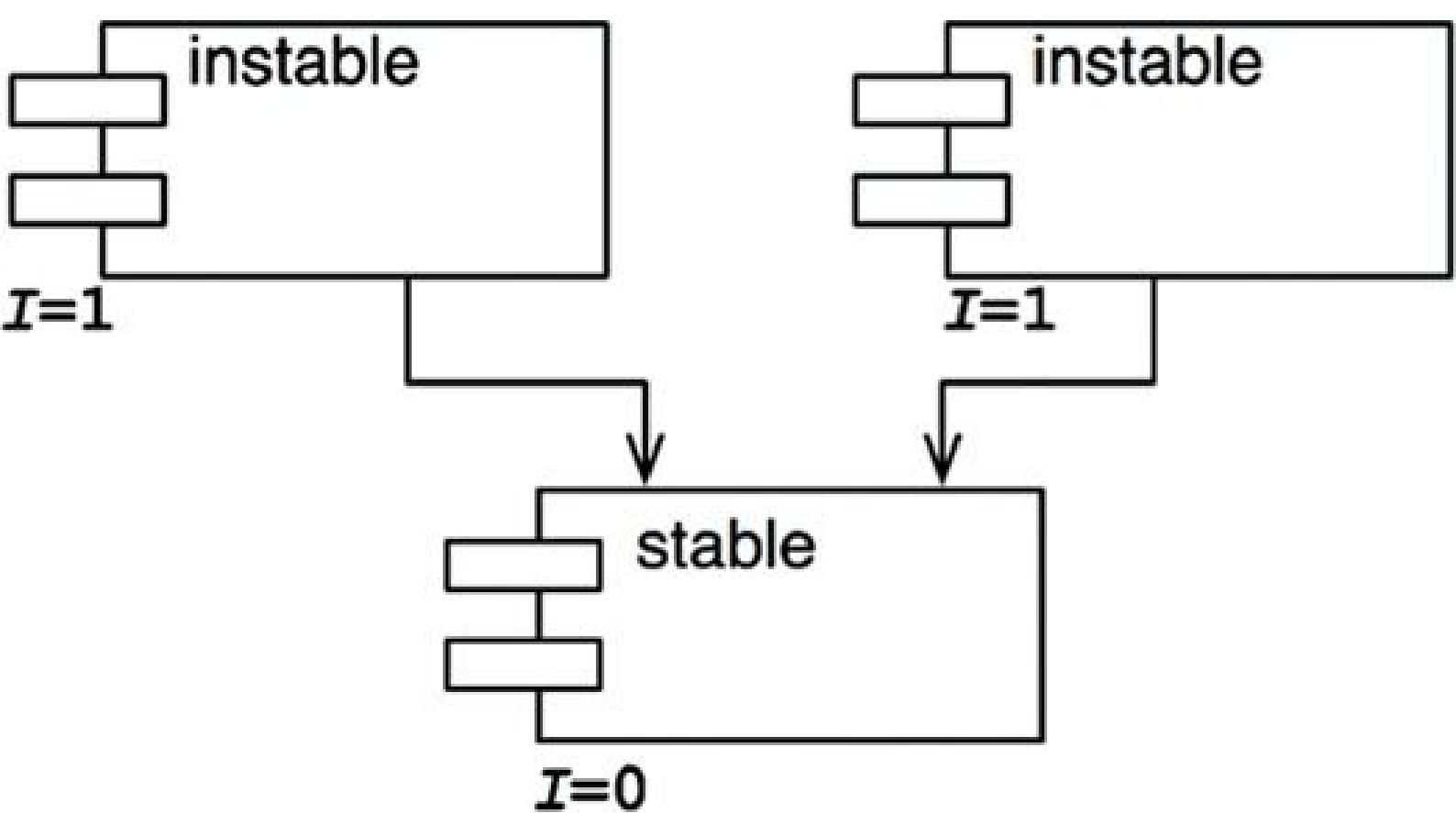


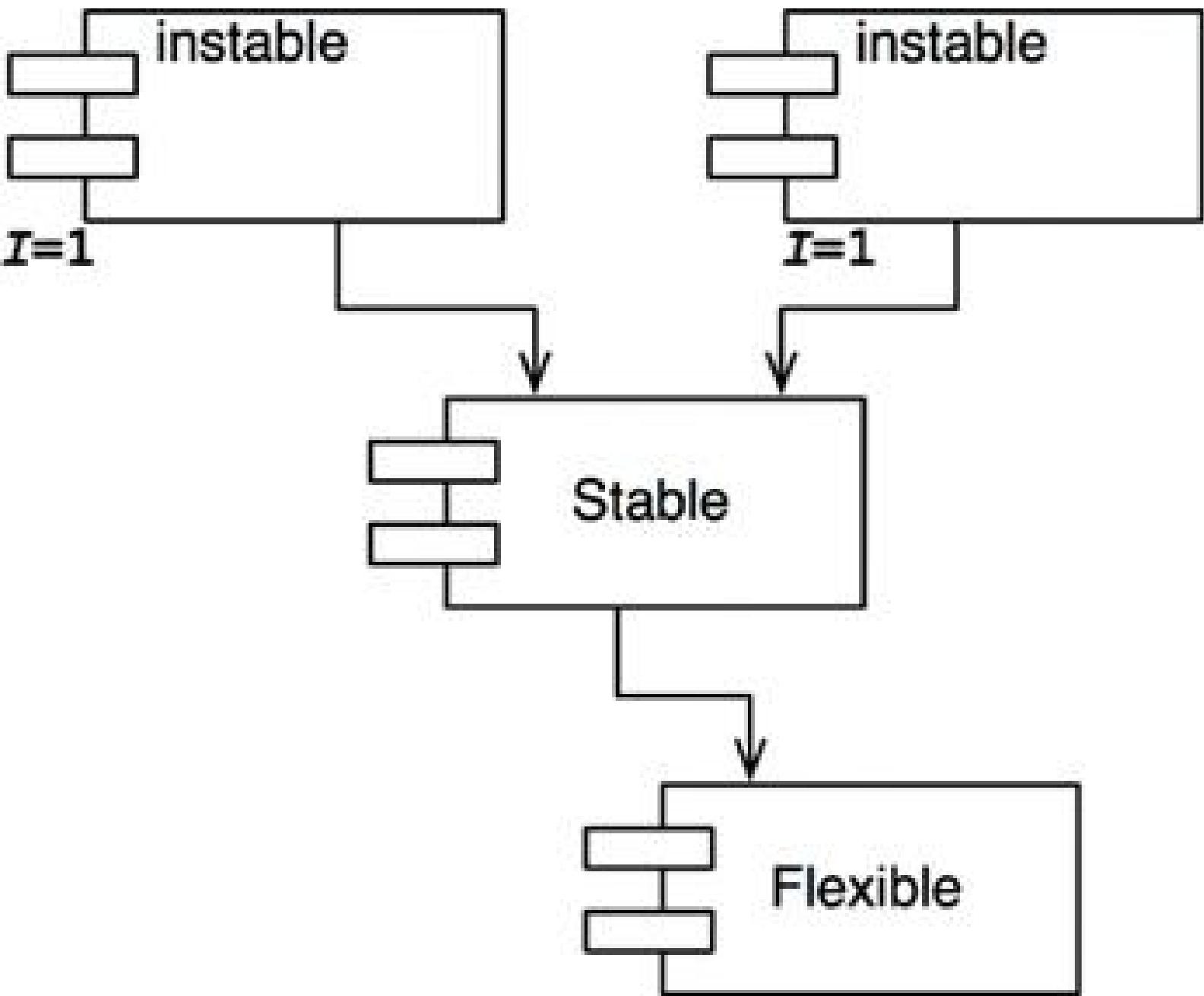


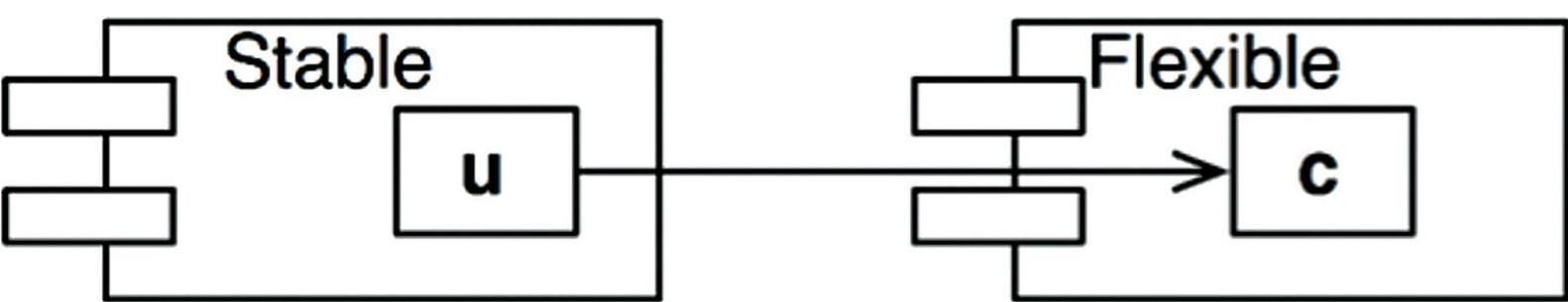


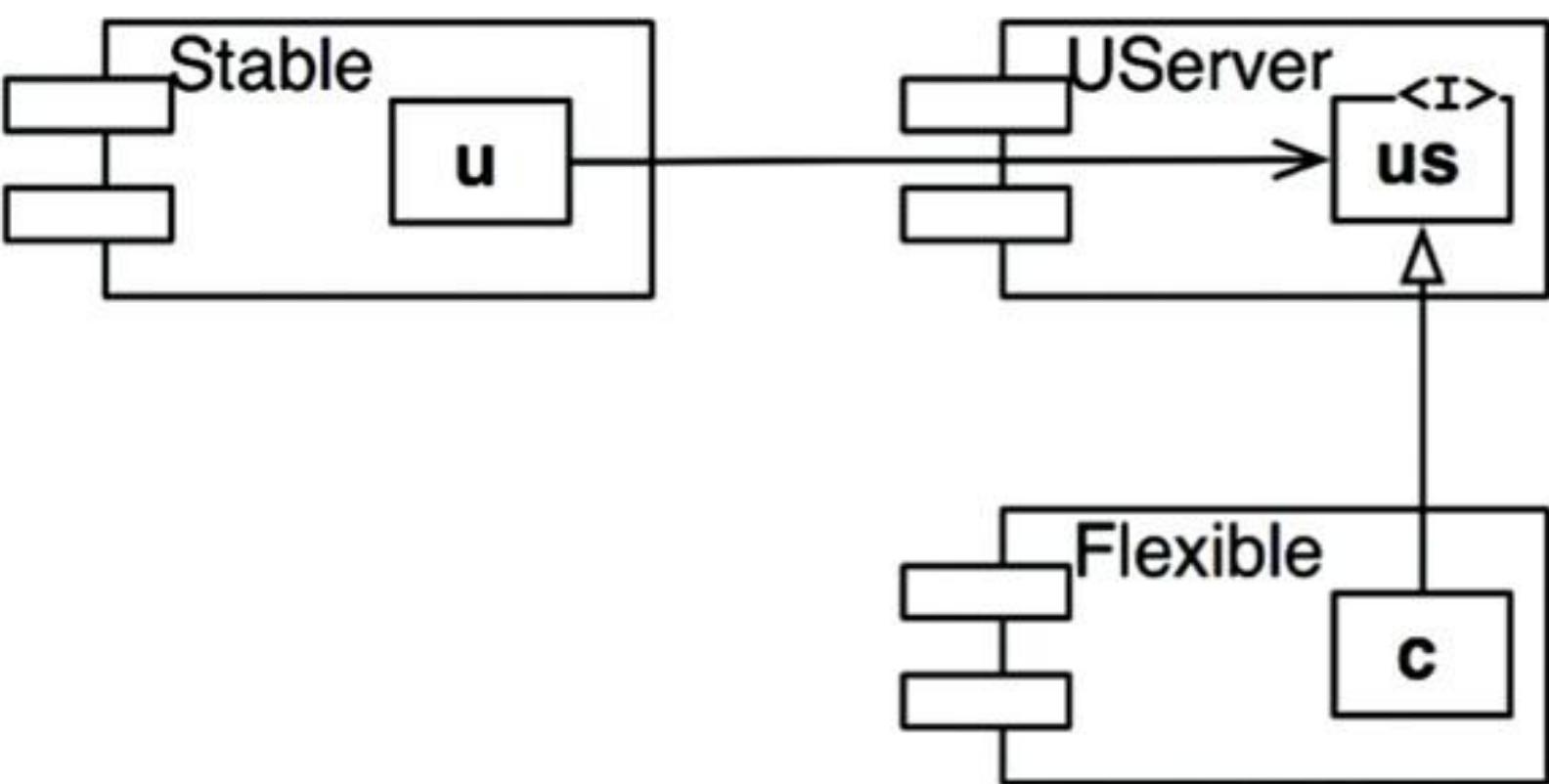








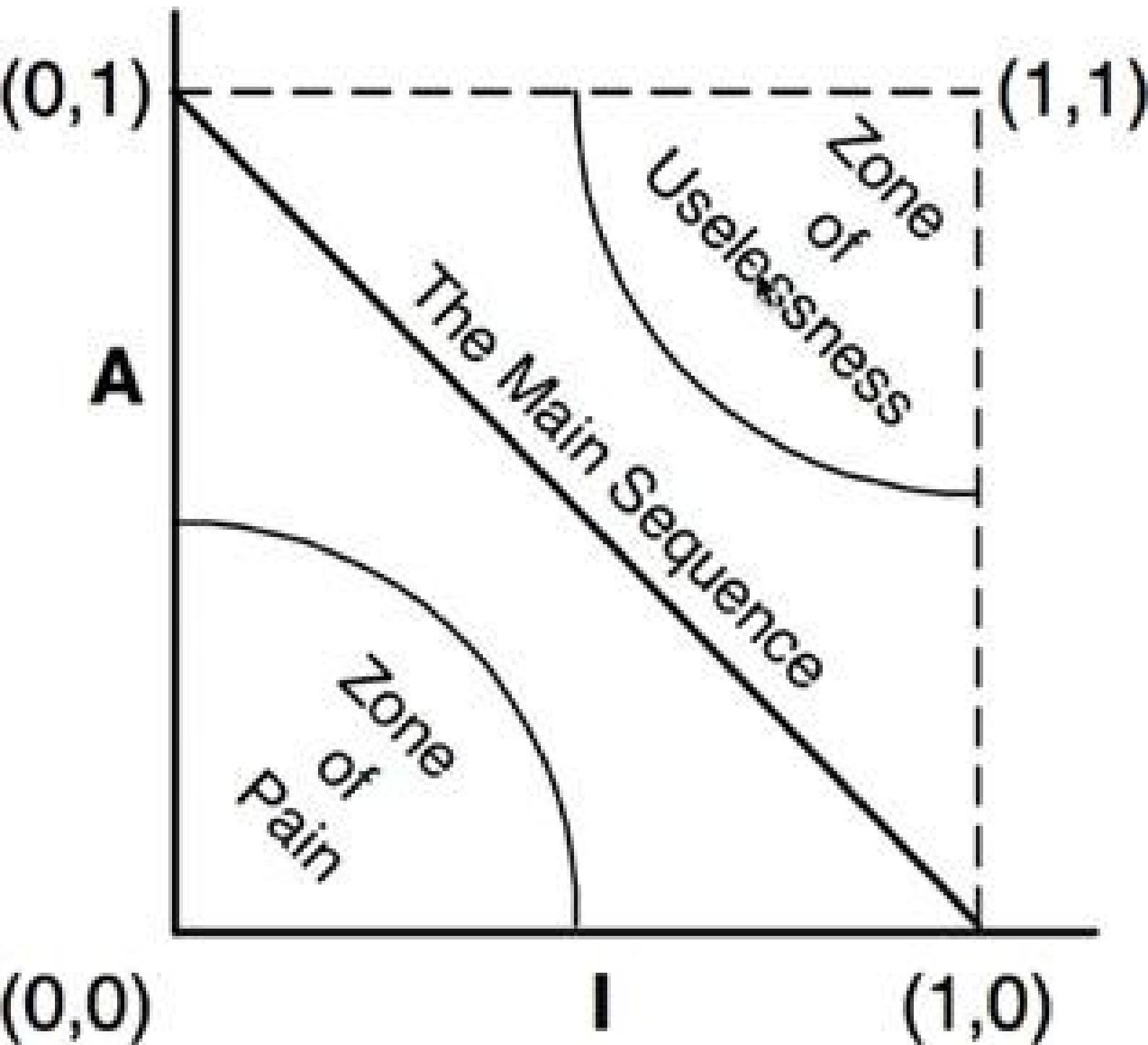




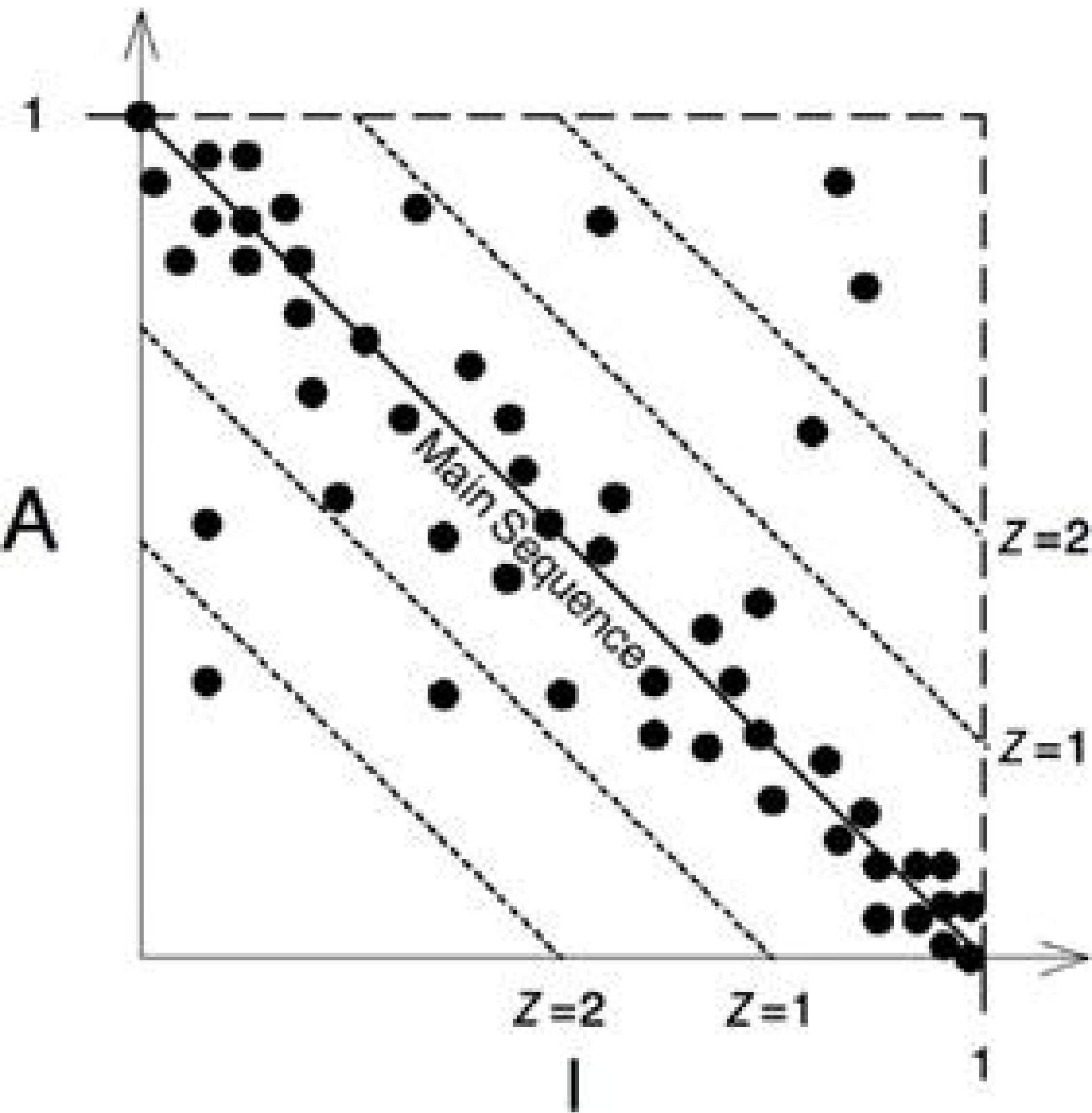
A

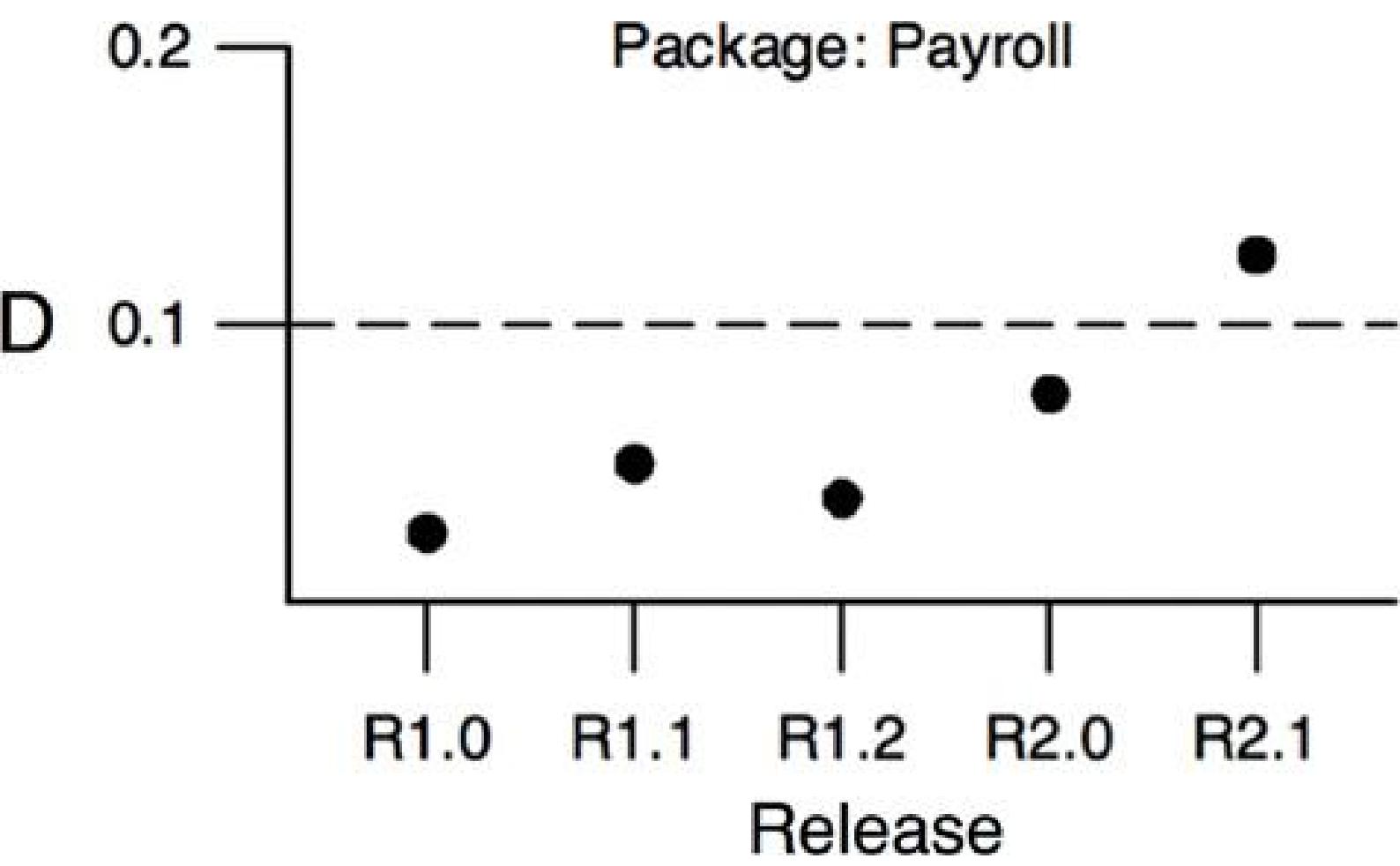
(0,1)

(1,0)



A









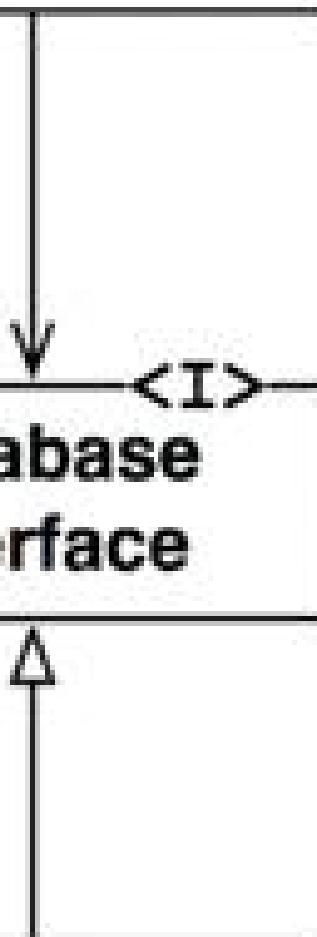


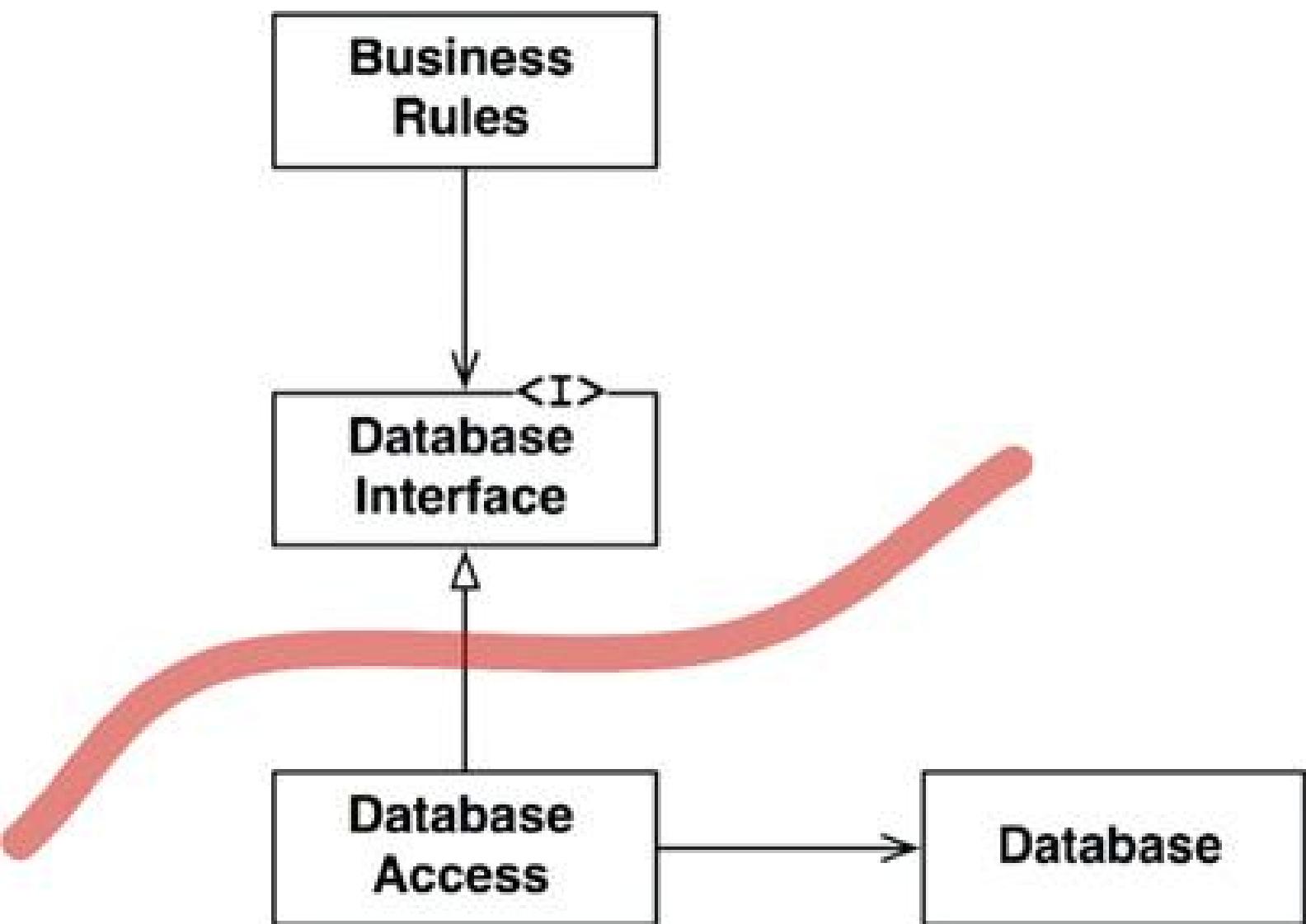
**Business
Rules**

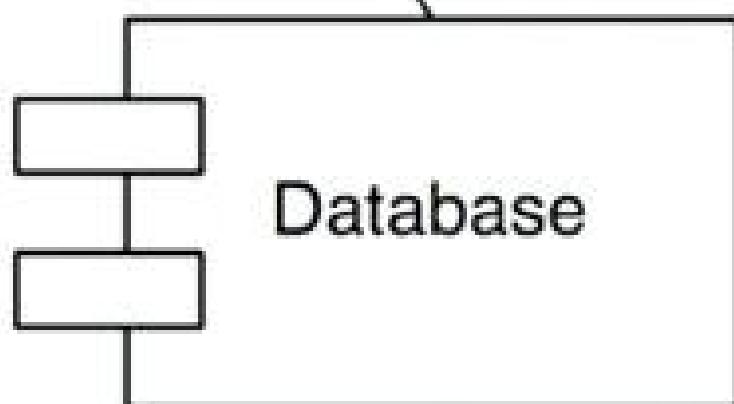
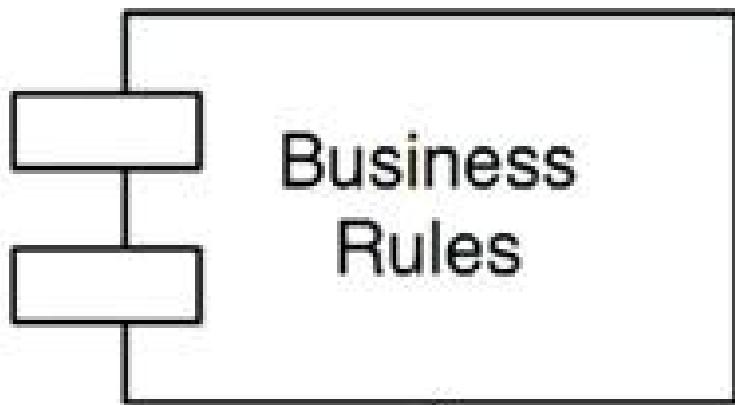
**Database
Interface**

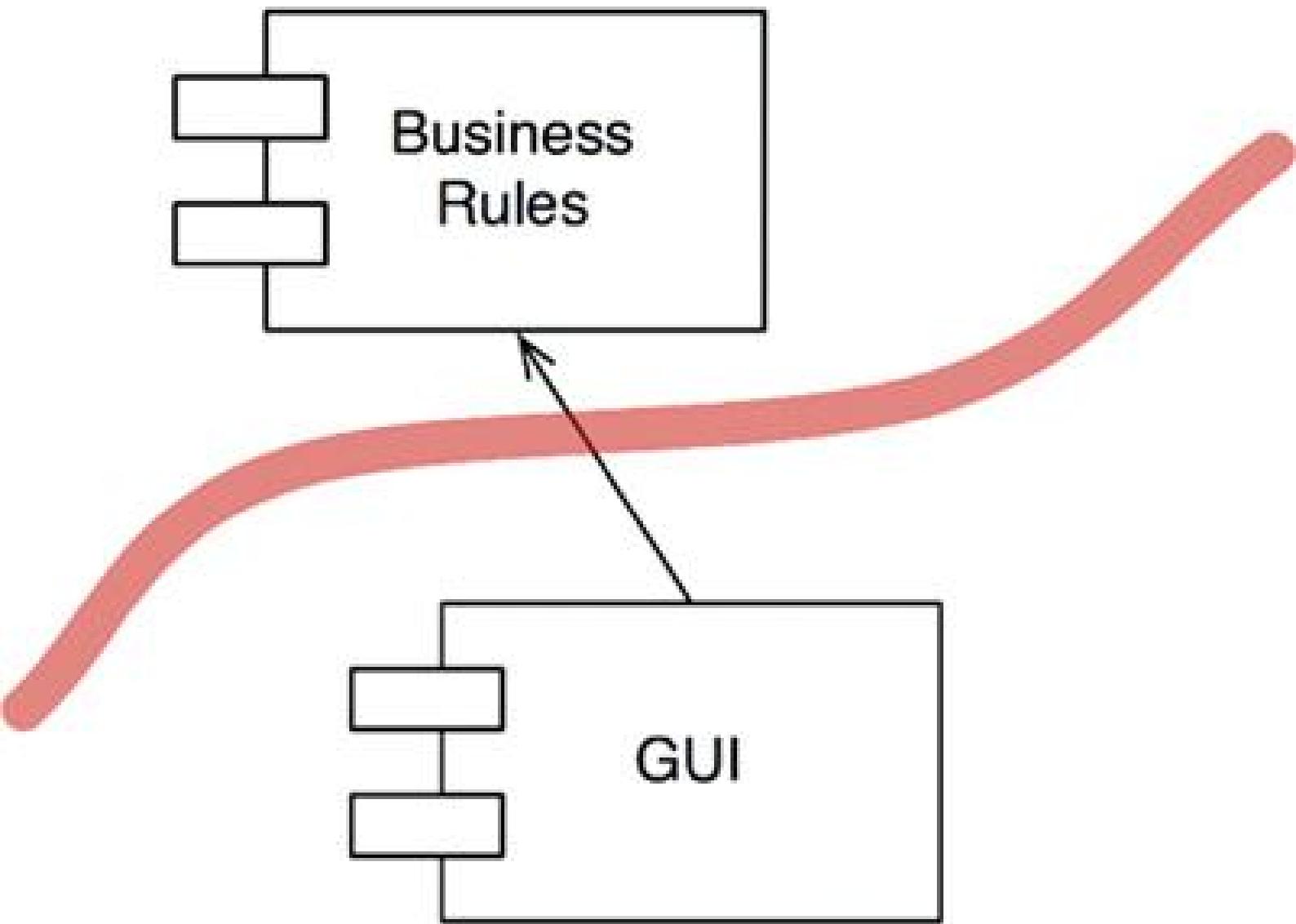
**Database
Access**

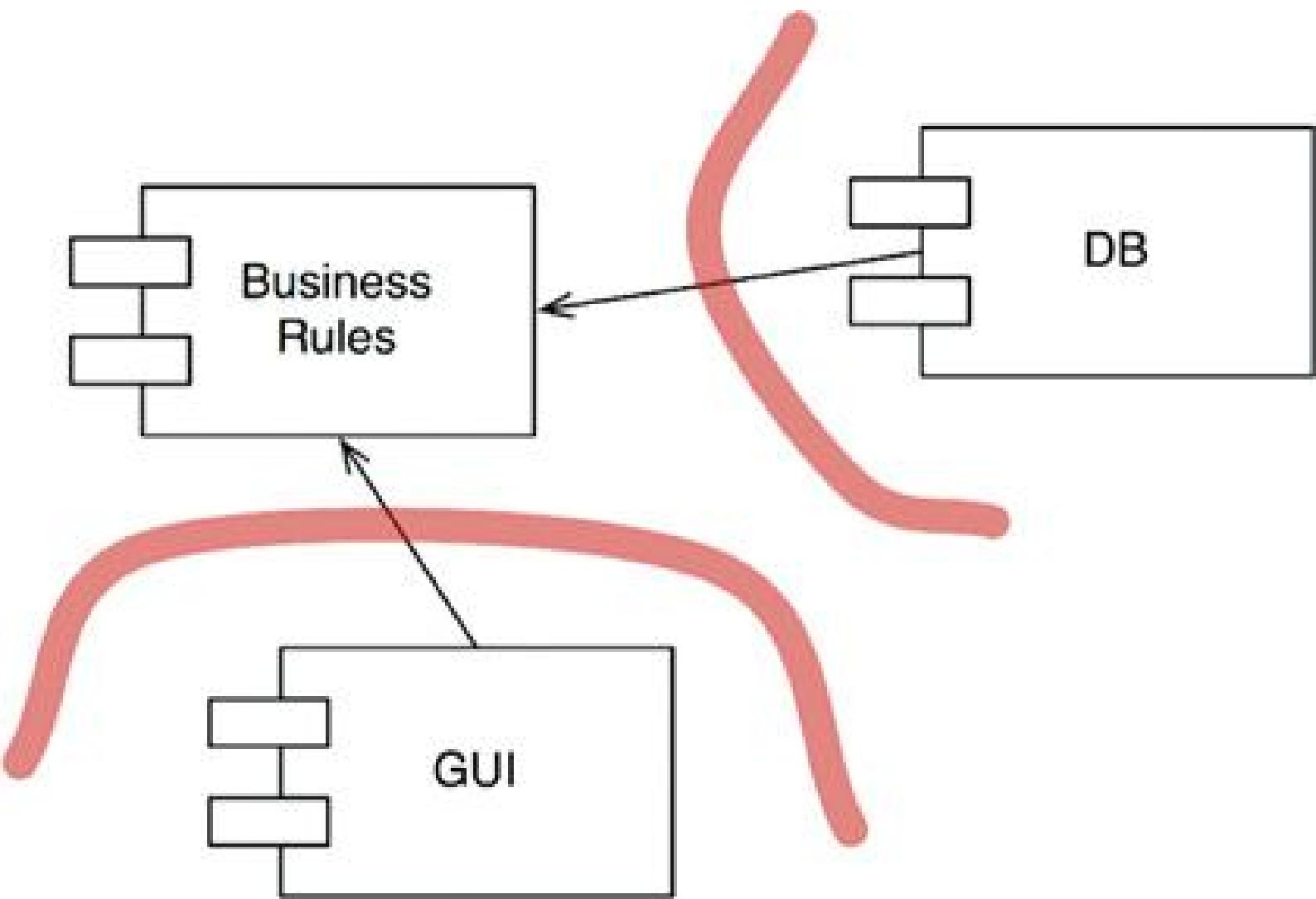
Database











vs

ReSharper

DONALD
WAS HERE

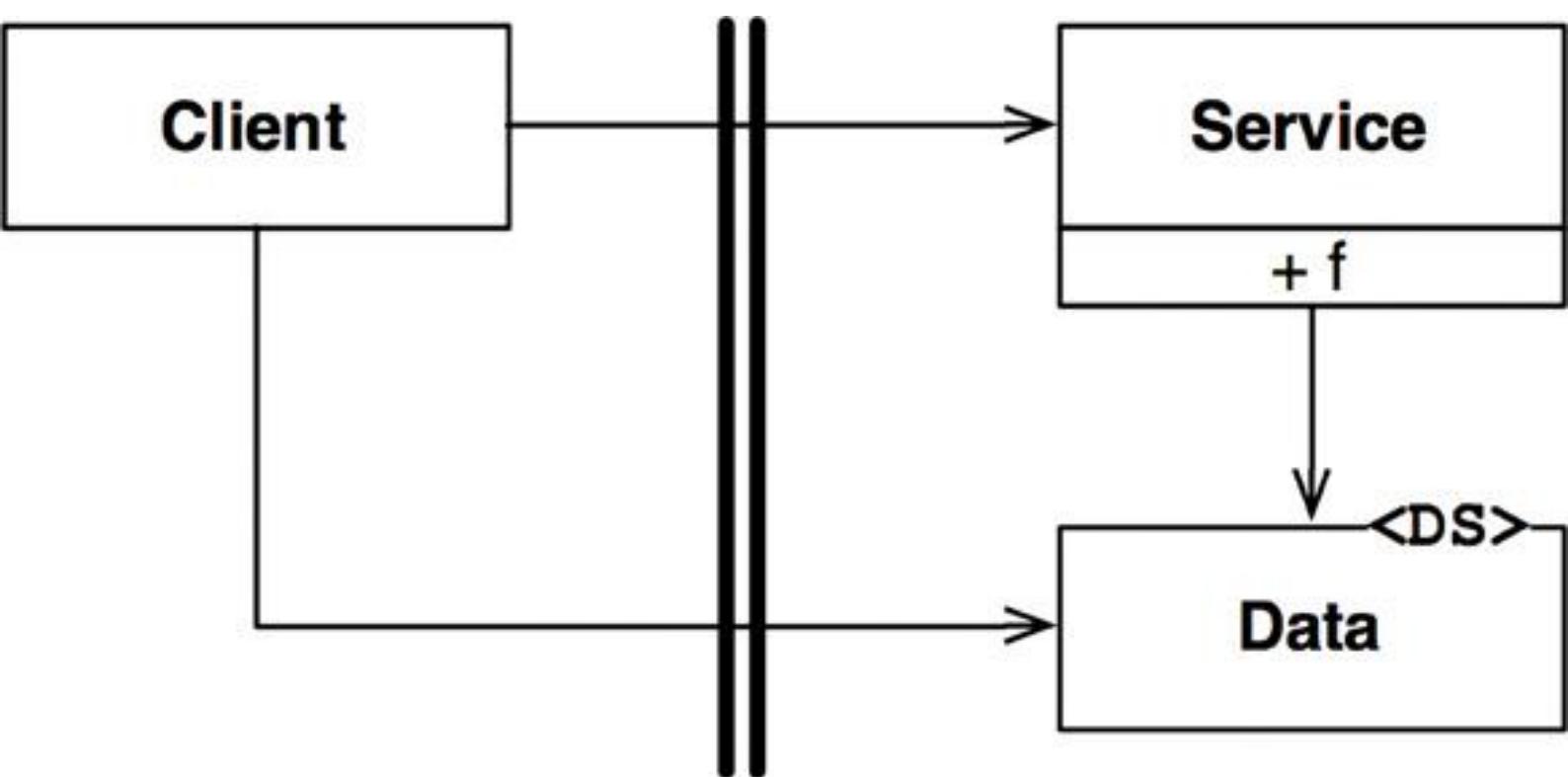
judge

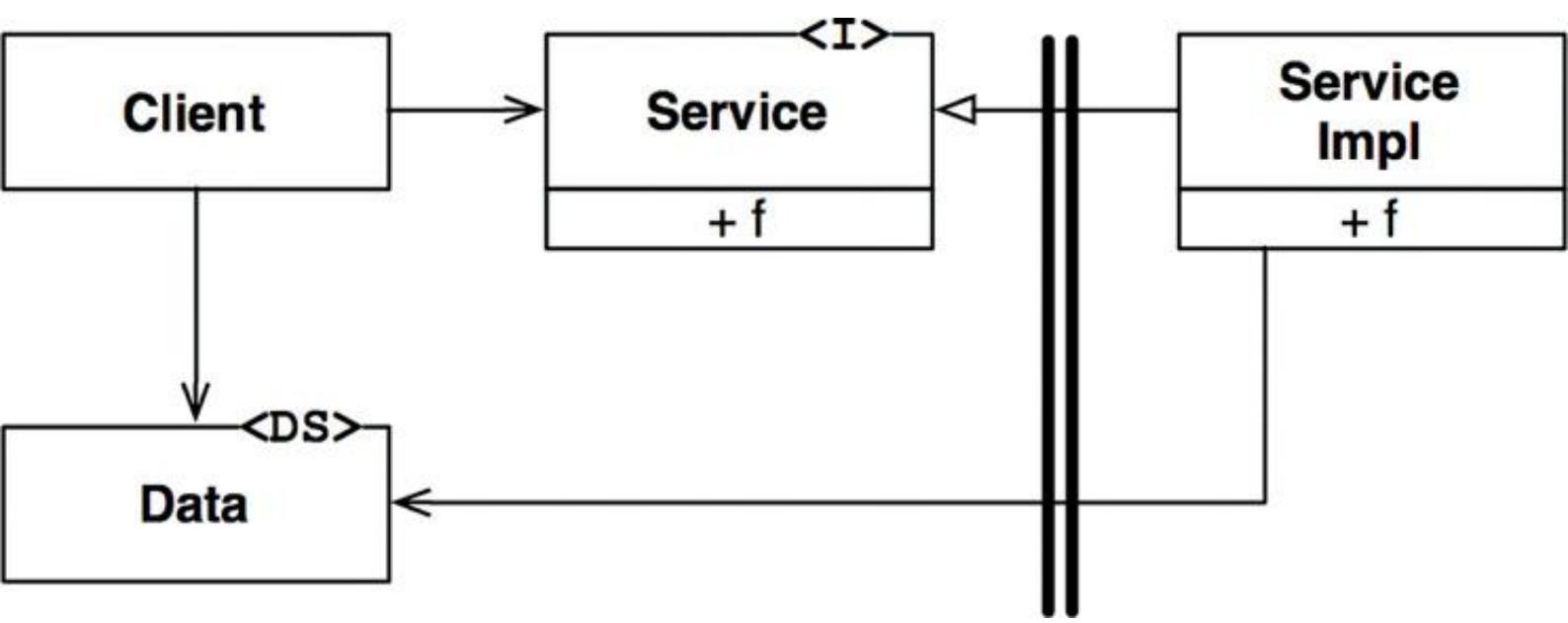
WERE?

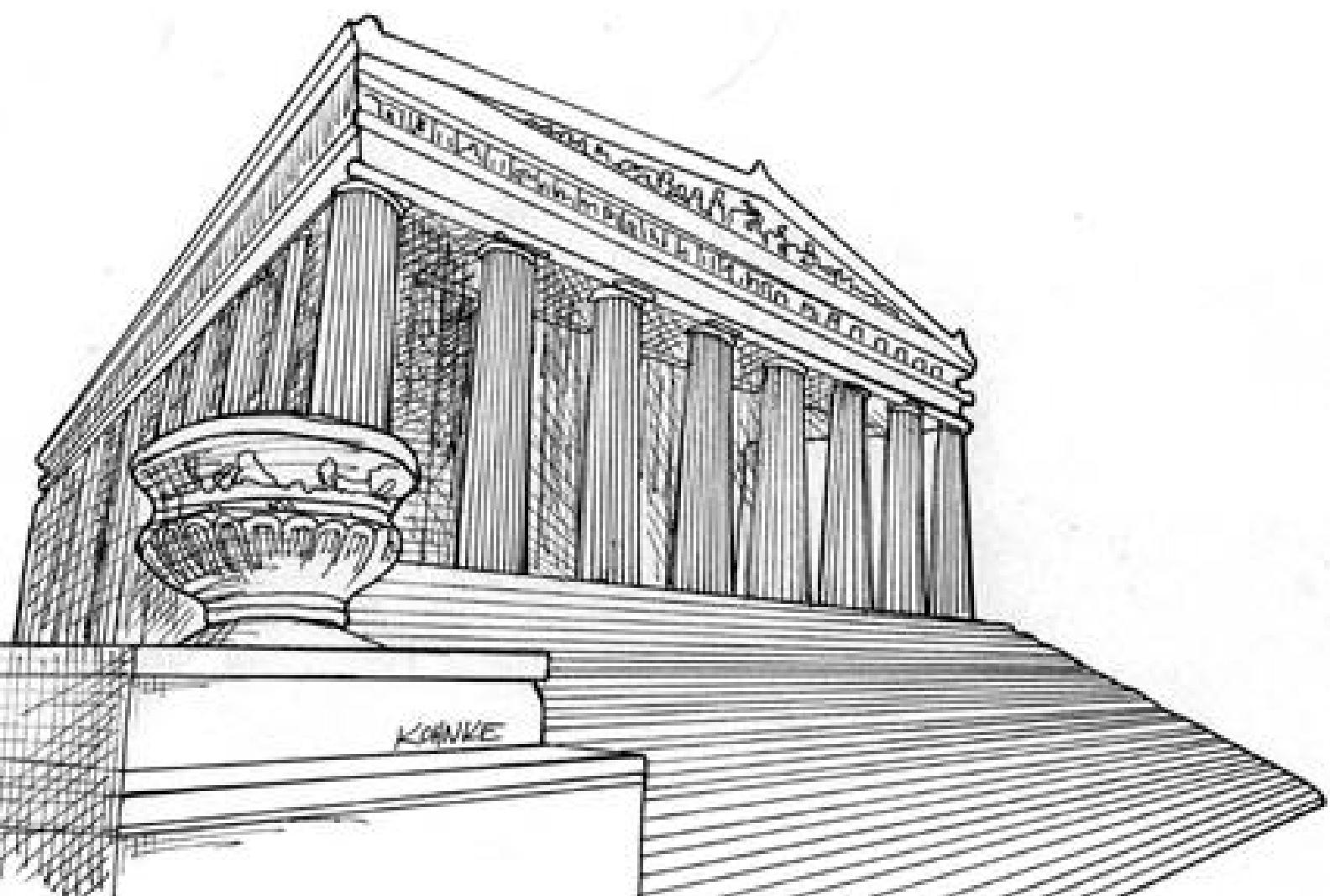
Popping? Hey Teacher!

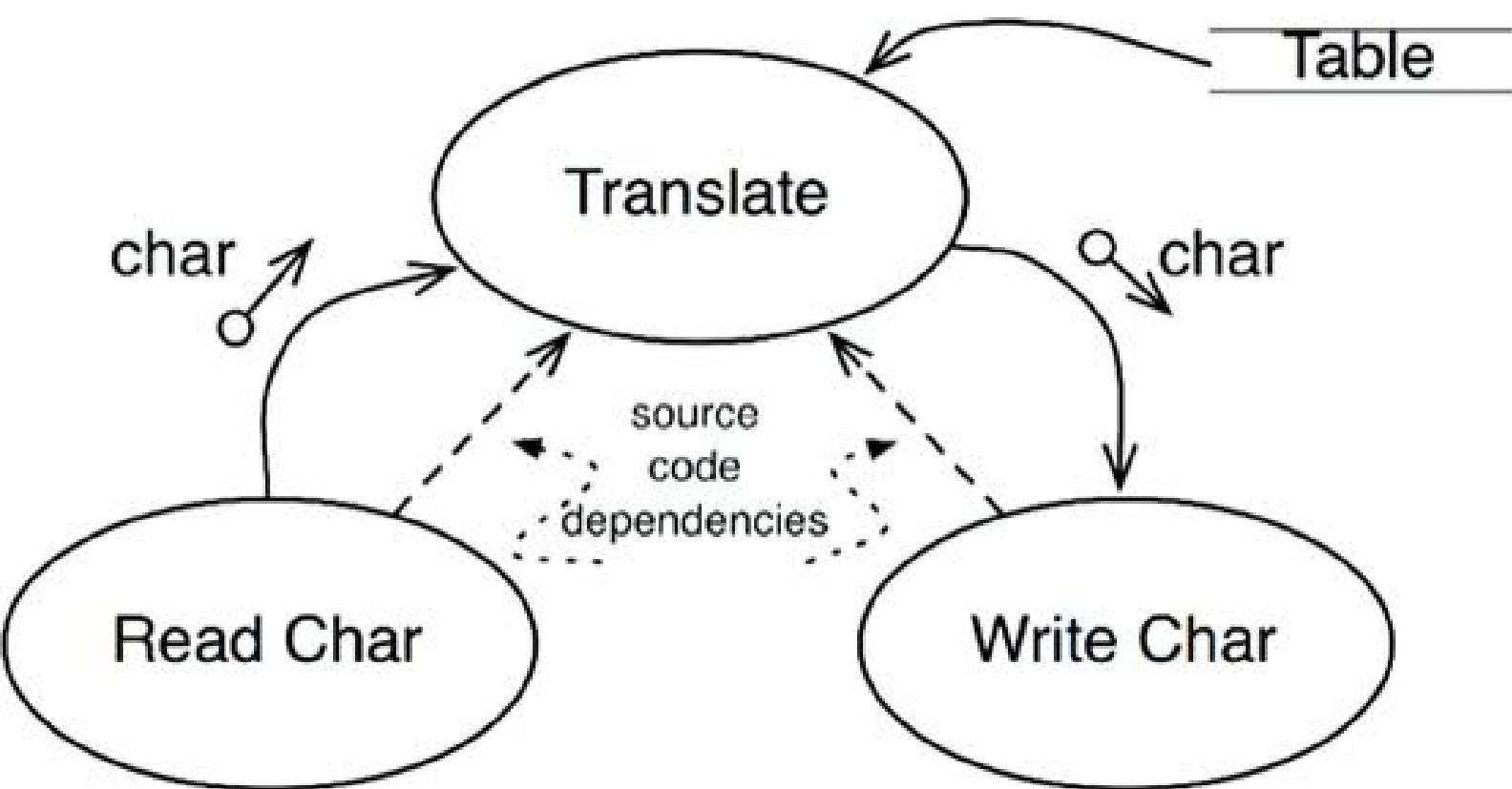
KOTONE

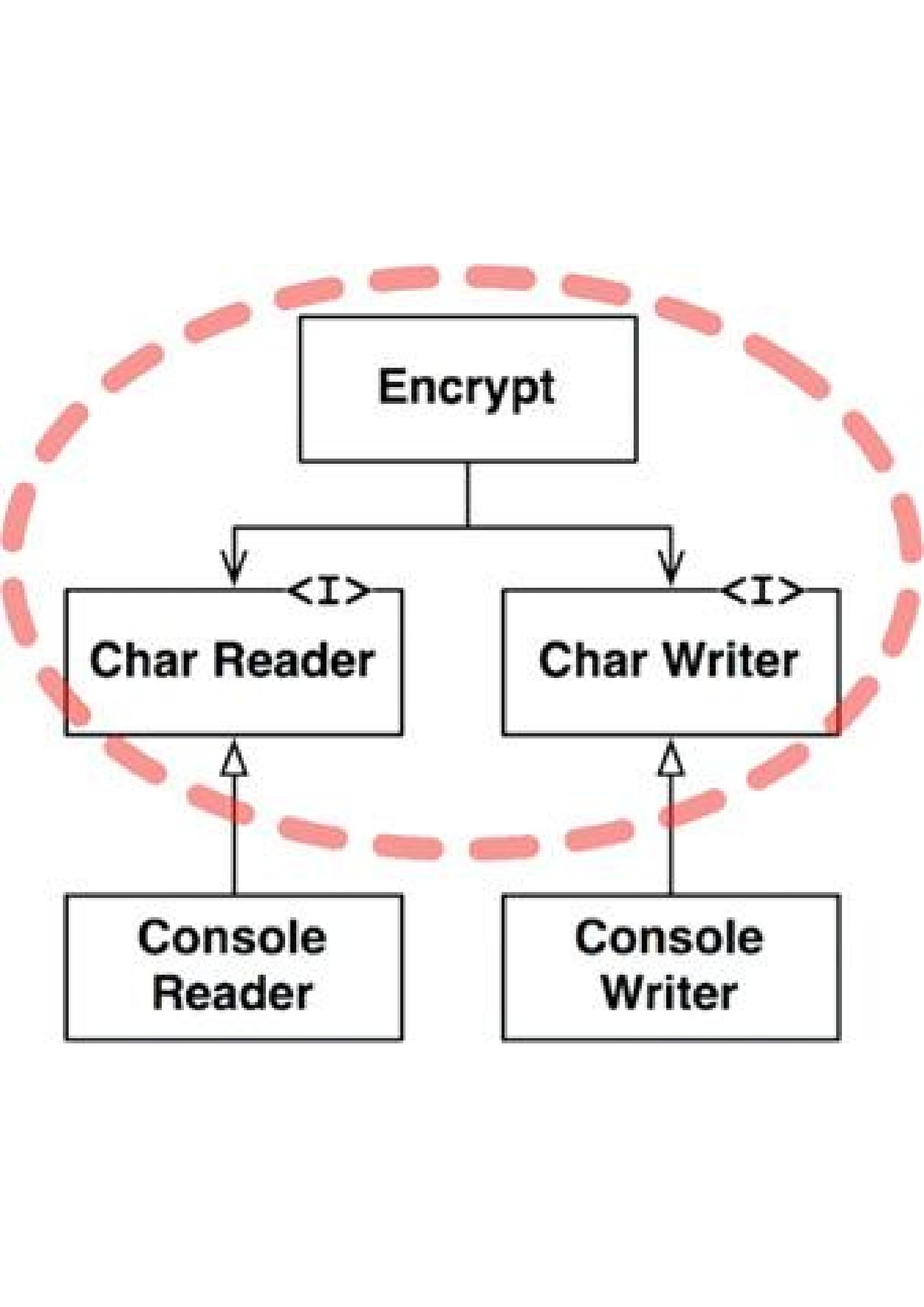




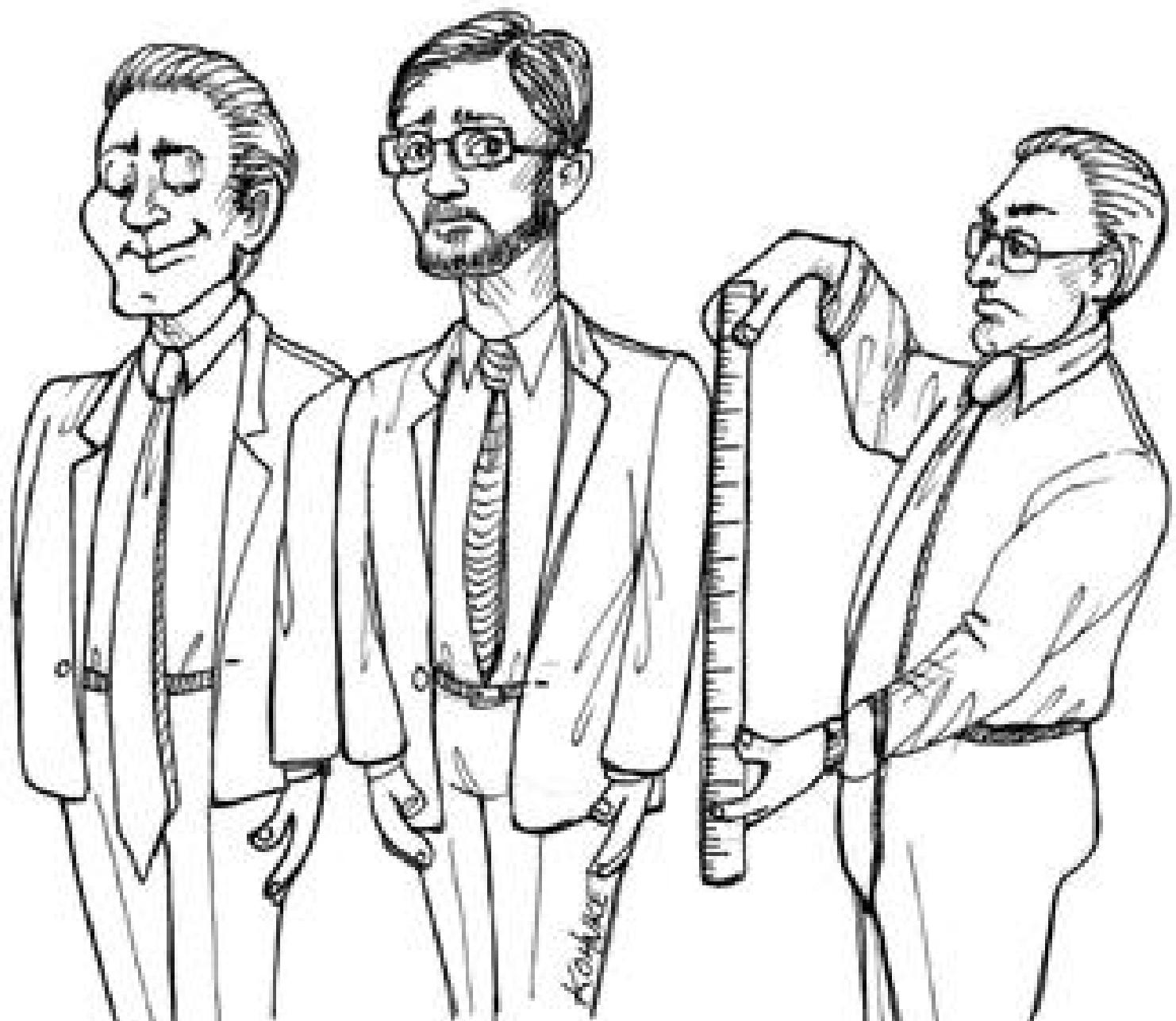












Loan

- principle
- rate
- period

- + makePayment()
- + applyInterest()
- + chargeLateFee()

Gather Contact Info for New Loan

Input: Name, Address, Birthdate, D.L. #, SSN, etc.
Output: Same info for readback + credit score.

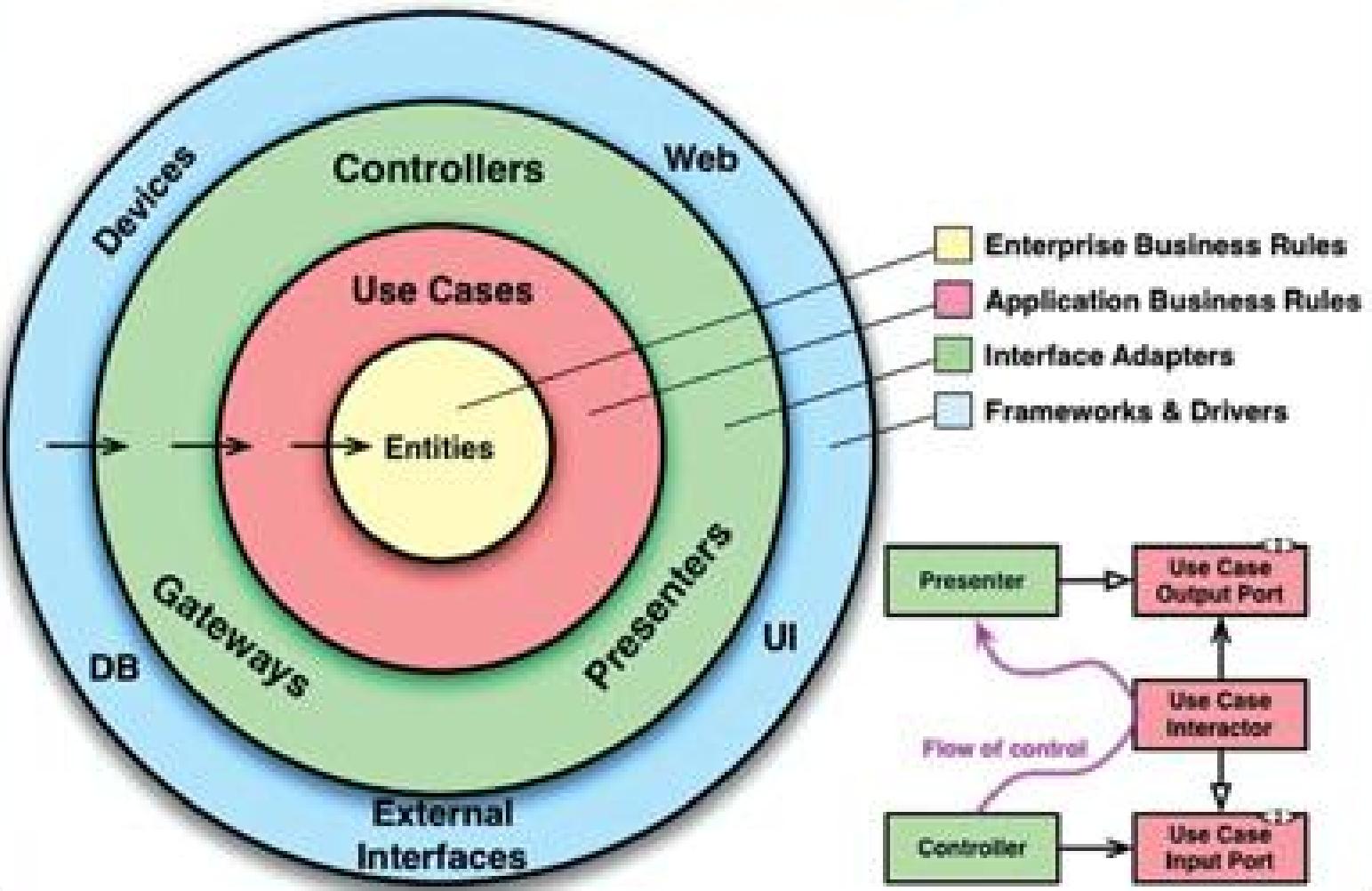
Primary Course:

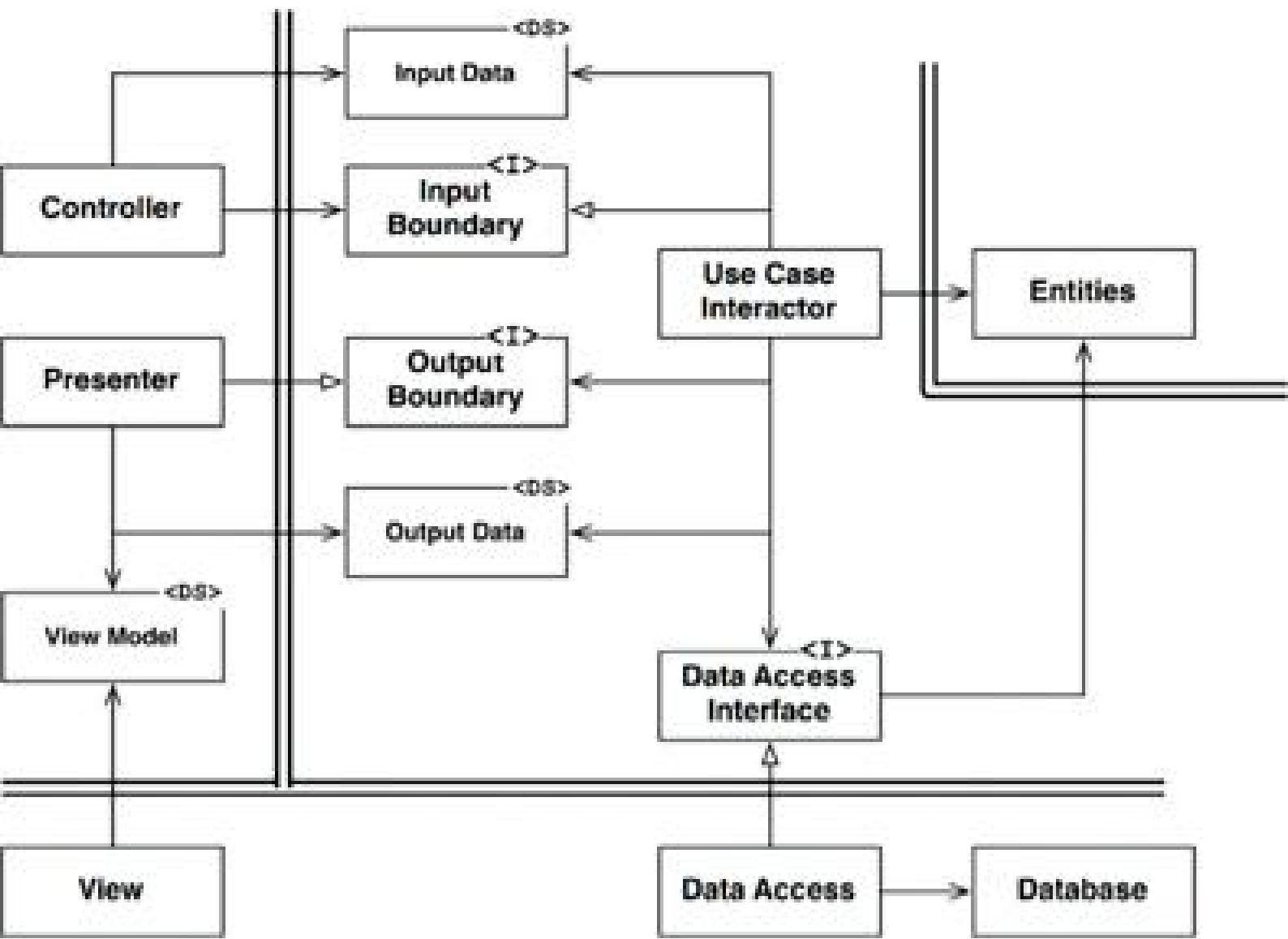
1. Accept and validate name.
2. Validate address, birthdate, D.L.#, SSN, etc.
3. Get credit score.
4. If credit score is < 500 activate Denial.
5. Else create Customer
and activate Loan Estimation.



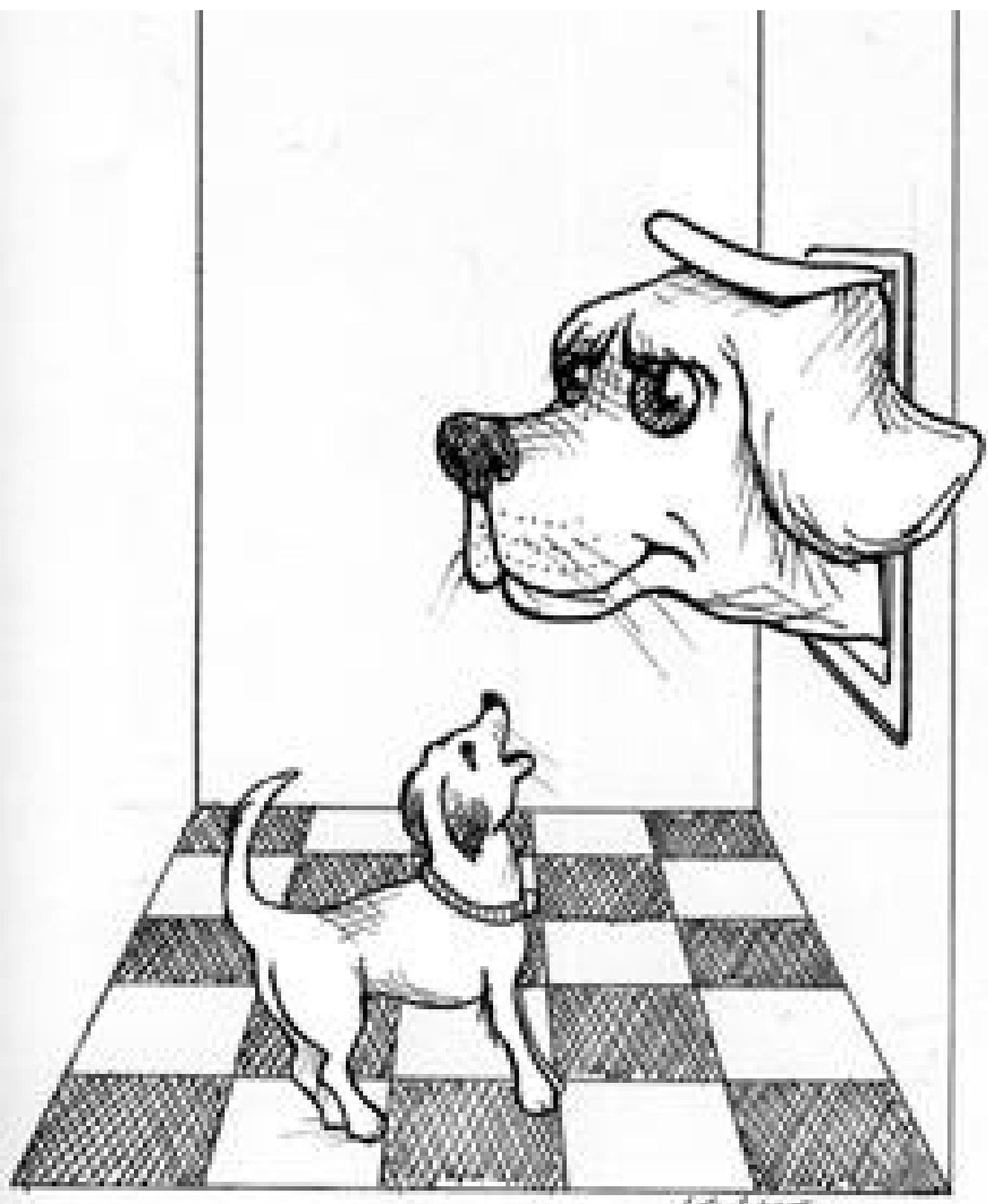


The Clean Architecture

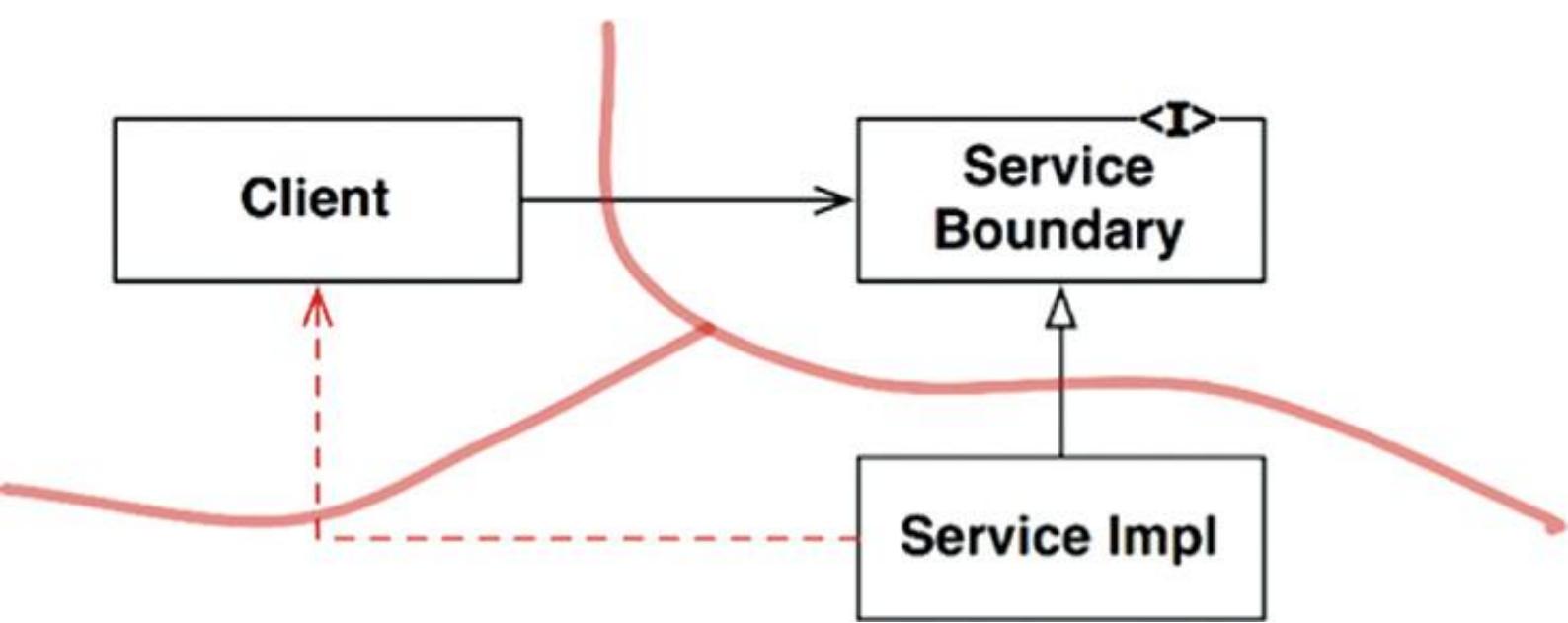


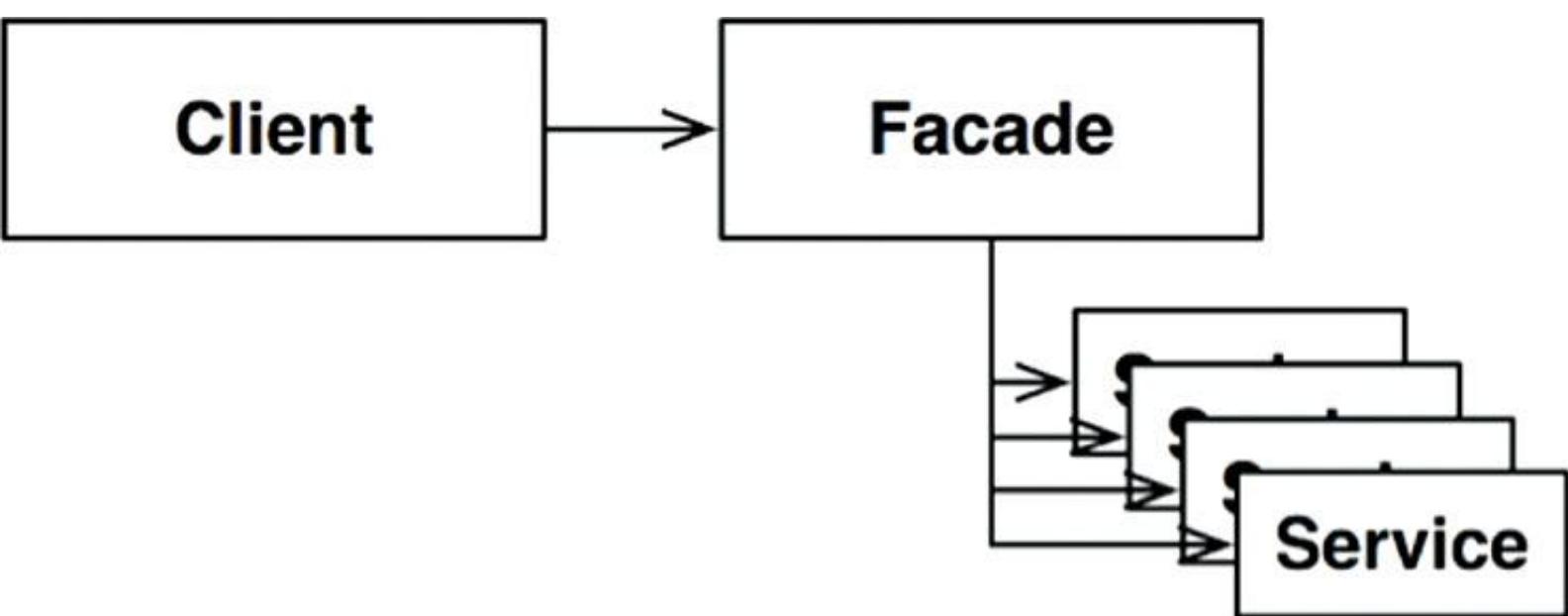






Kopflos



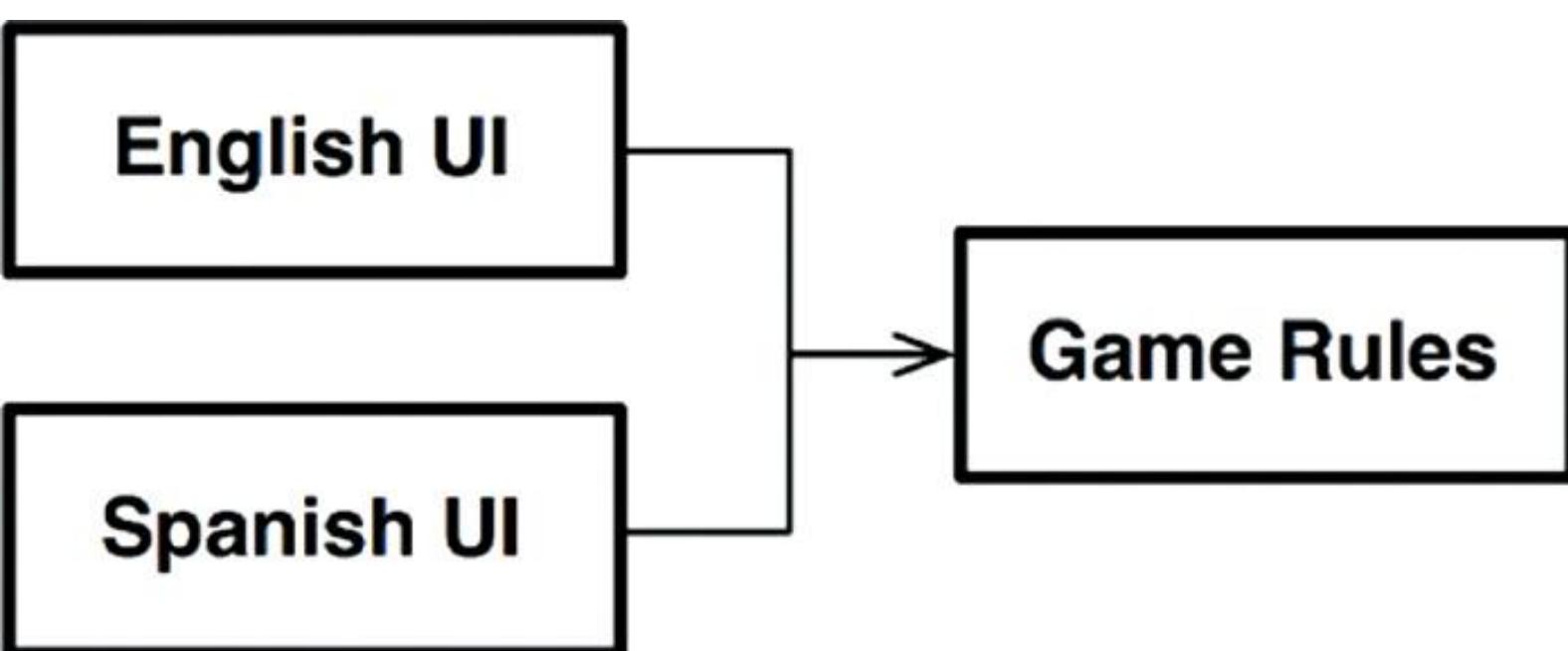


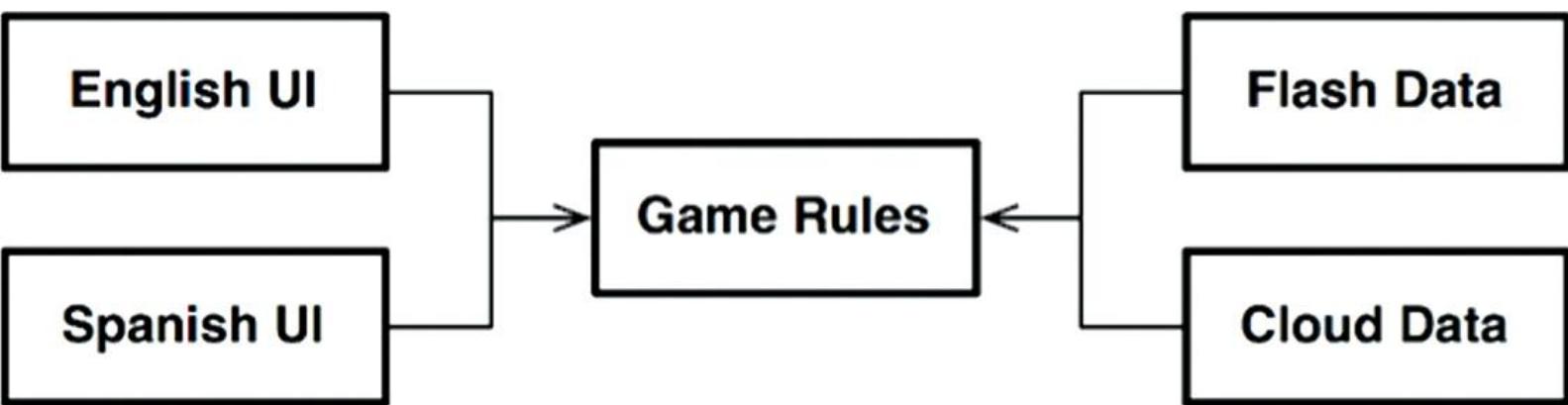


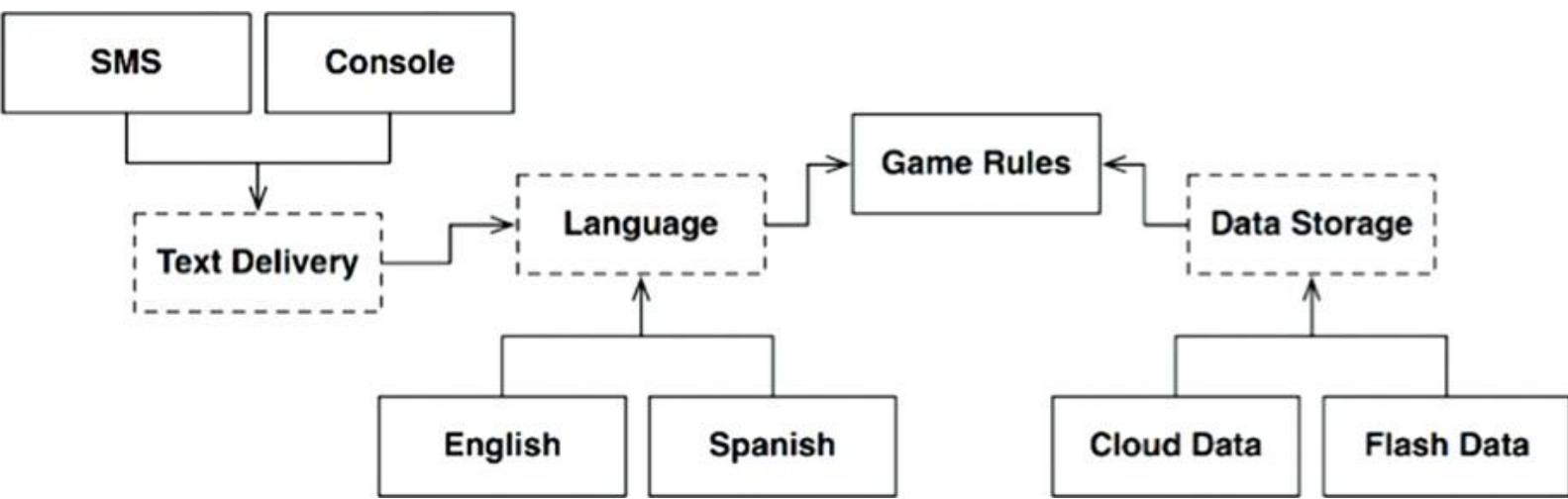
English UI

Spanish UI

Game Rules





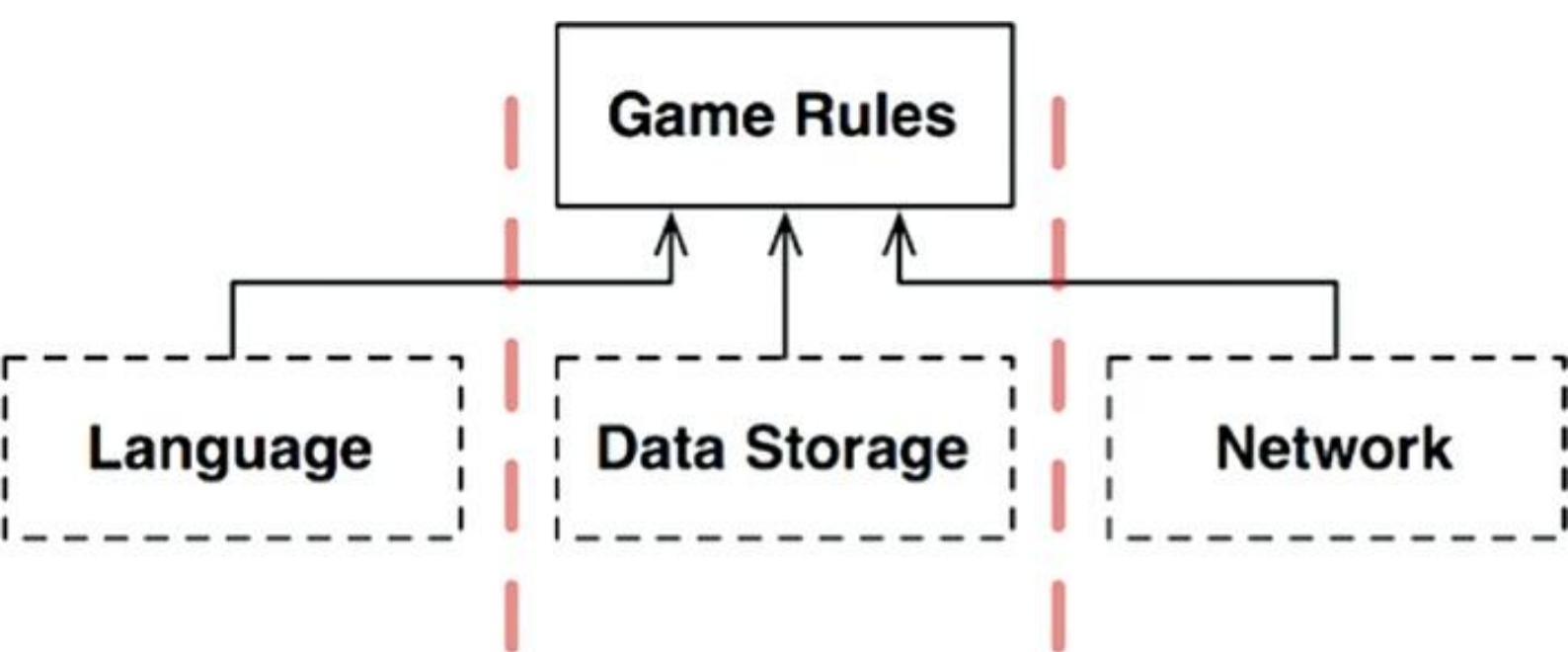


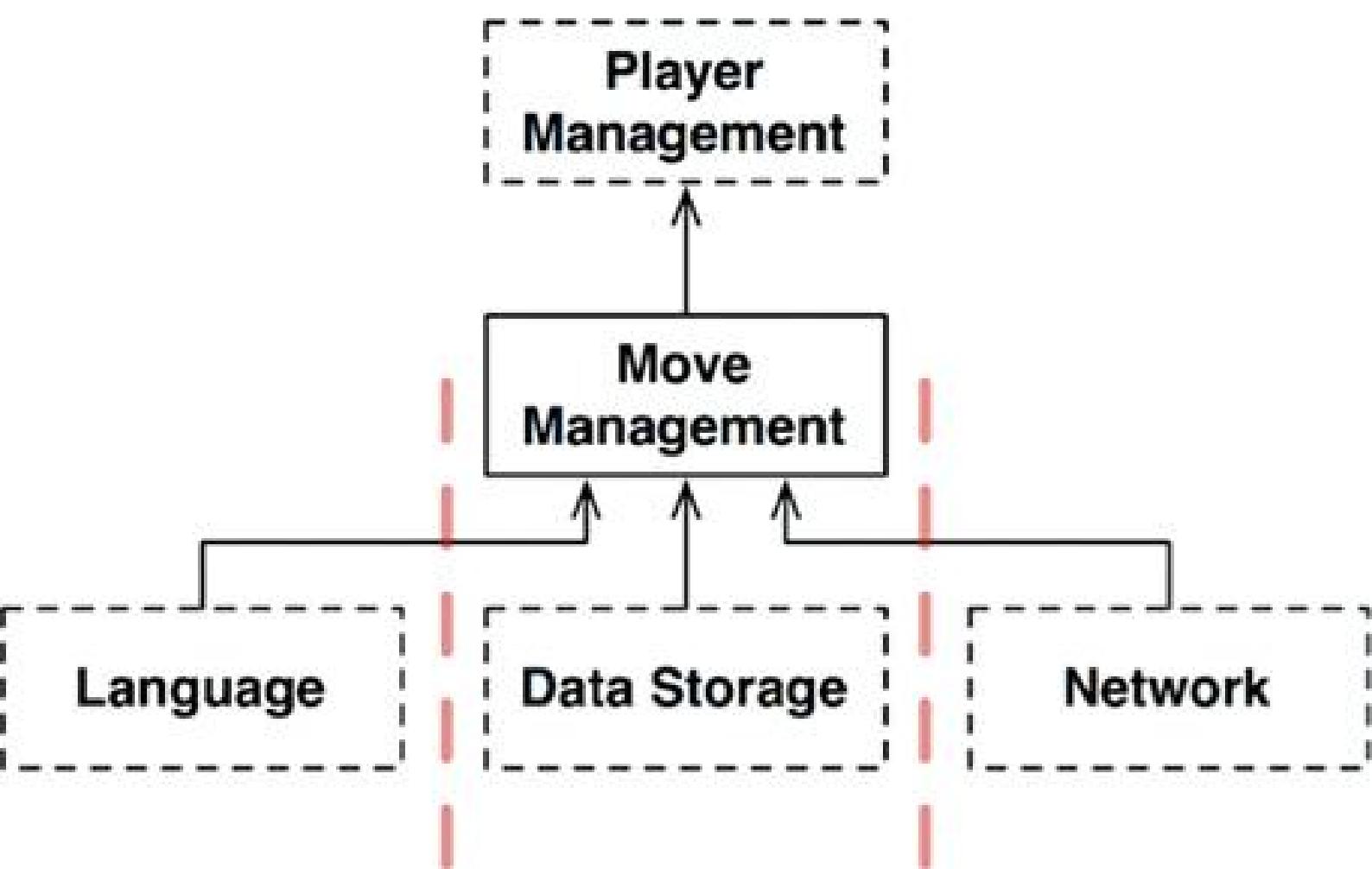
Game Rules

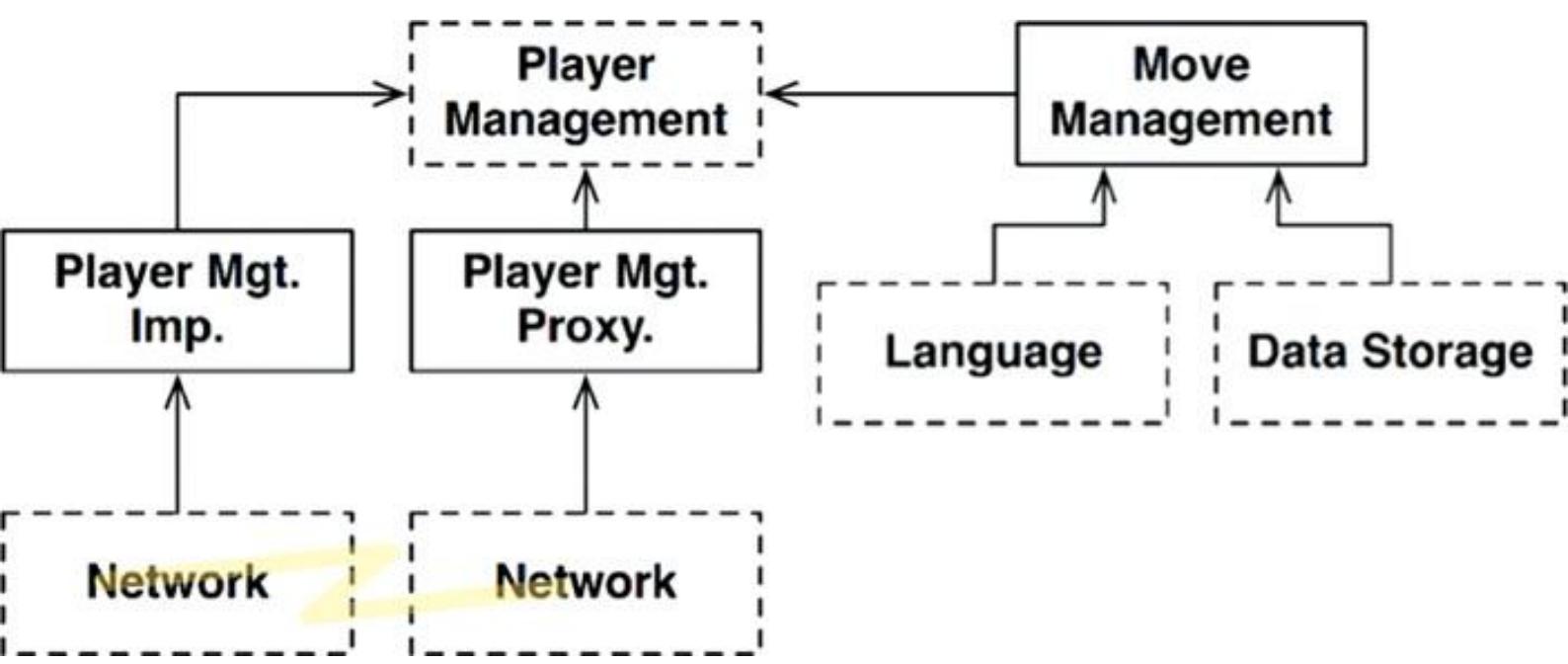
Language

Data Storage

Text Delivery

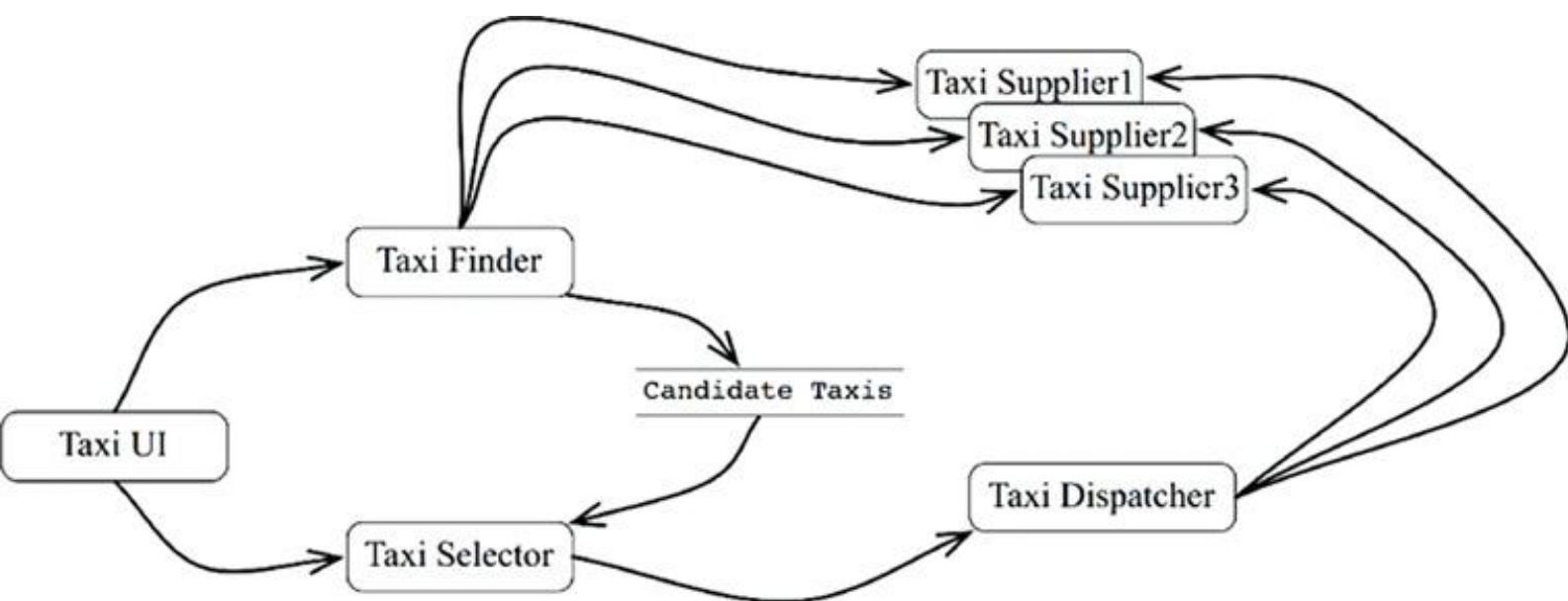


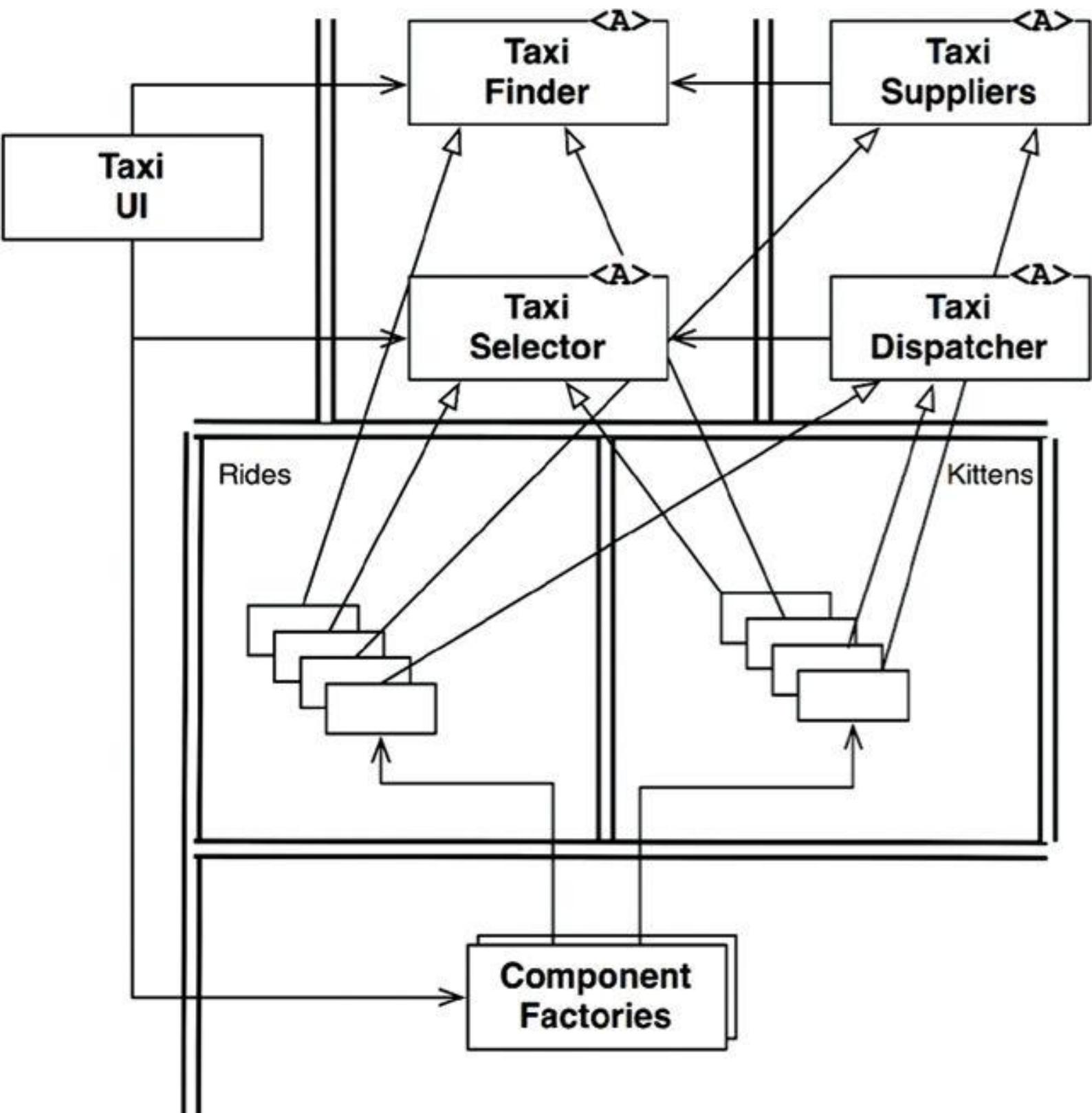


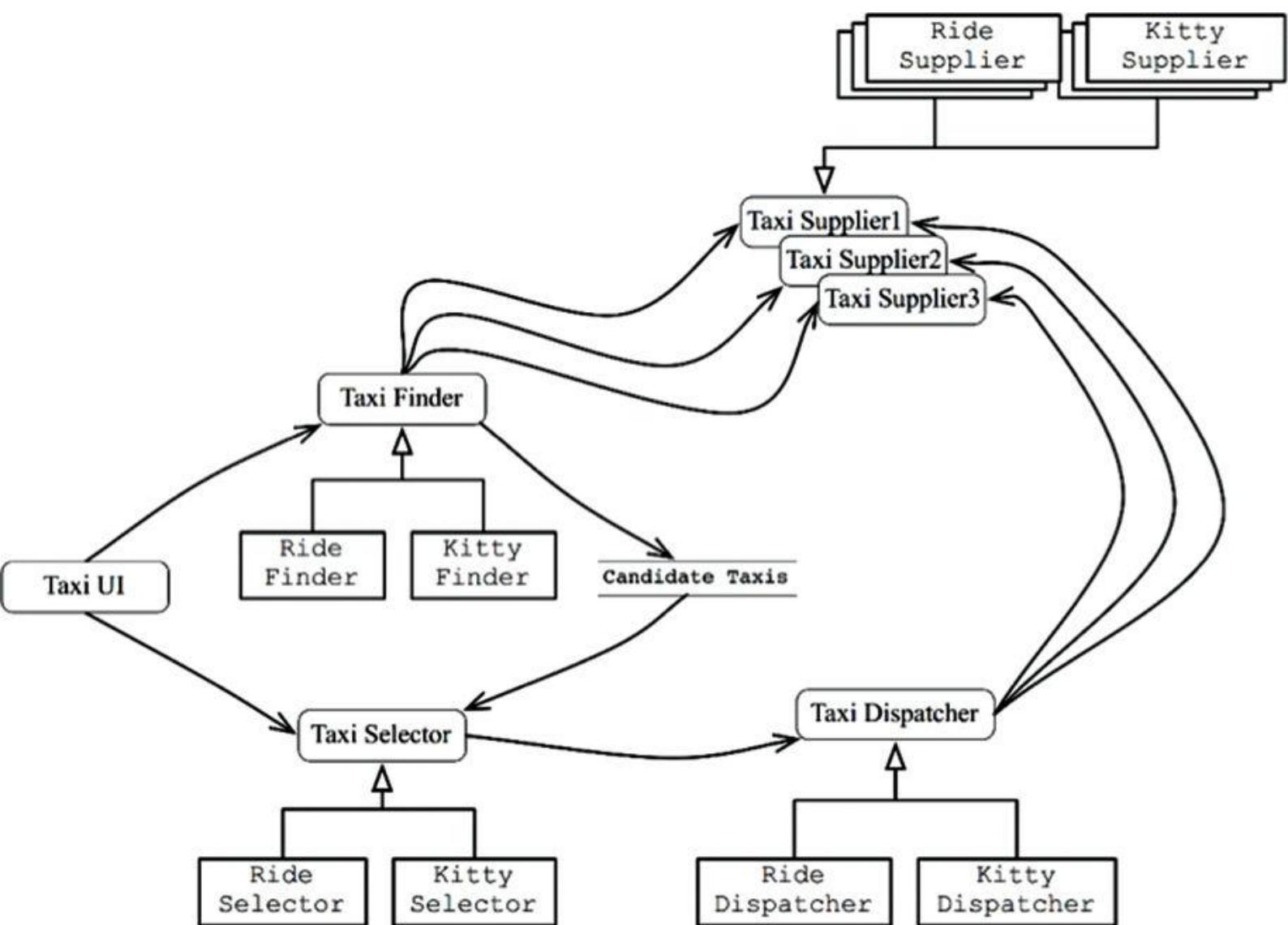


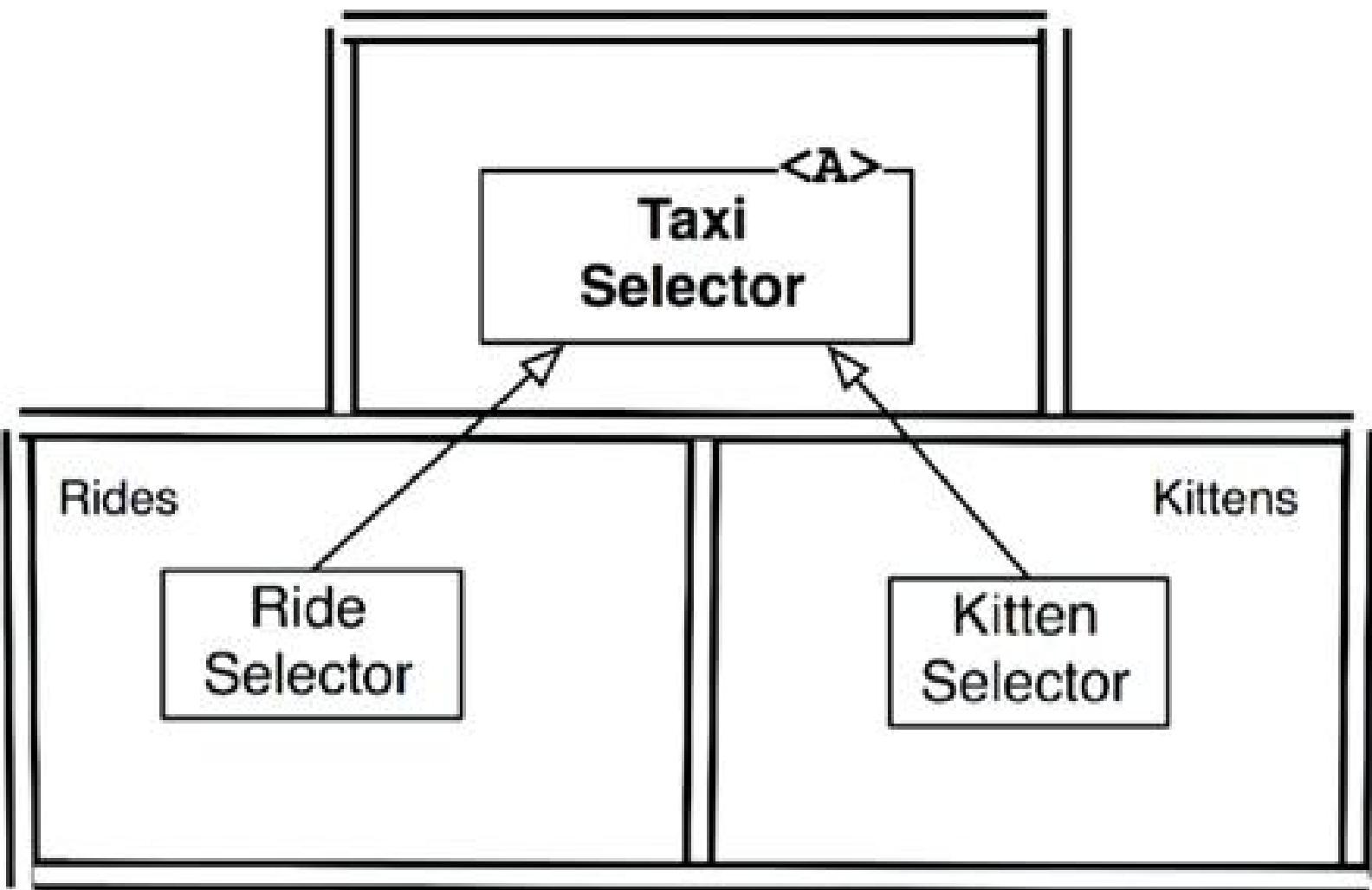
















Software

Firmware

Hardware



Firmware



Hardware

Software

Firmware

Hardware

HAL

```
graph LR; HAL((HAL)) --> Software[Software]; Software --- Firmware[Firmware]; Firmware --- Hardware[Hardware]
```

Software

Firmware

Hardware

Software

OS

Firmware

Hardware

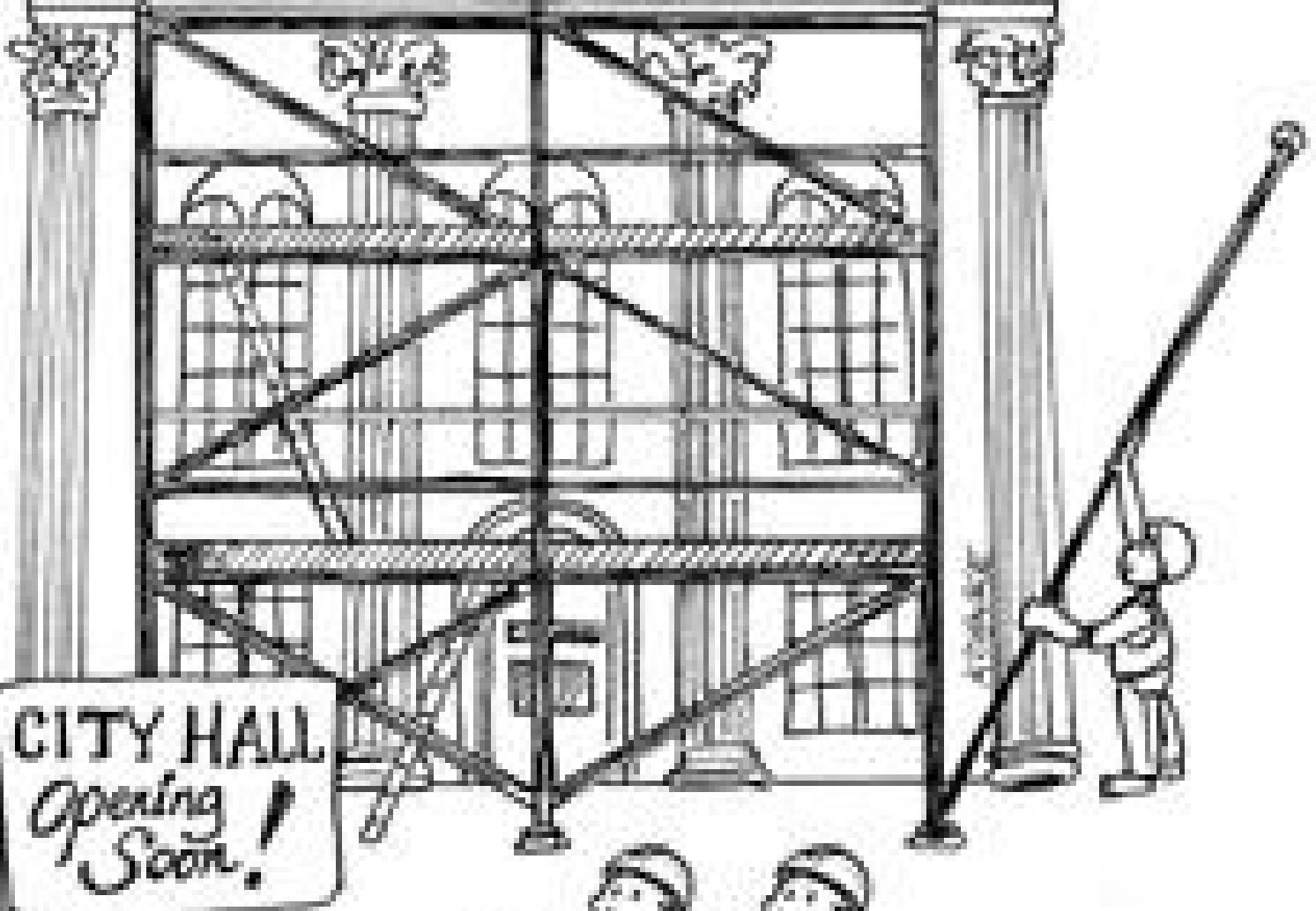
OSAL
→
HAL





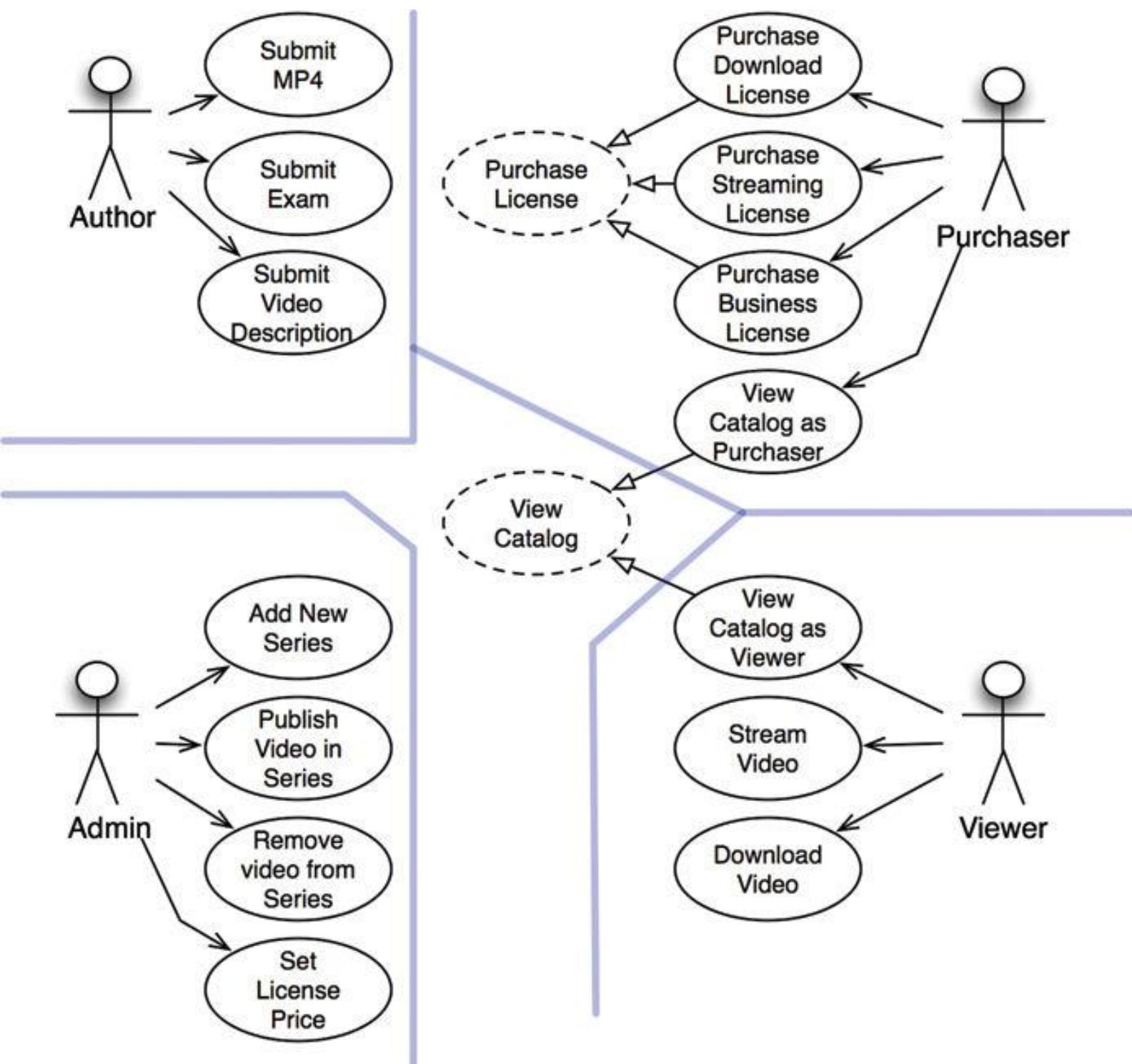


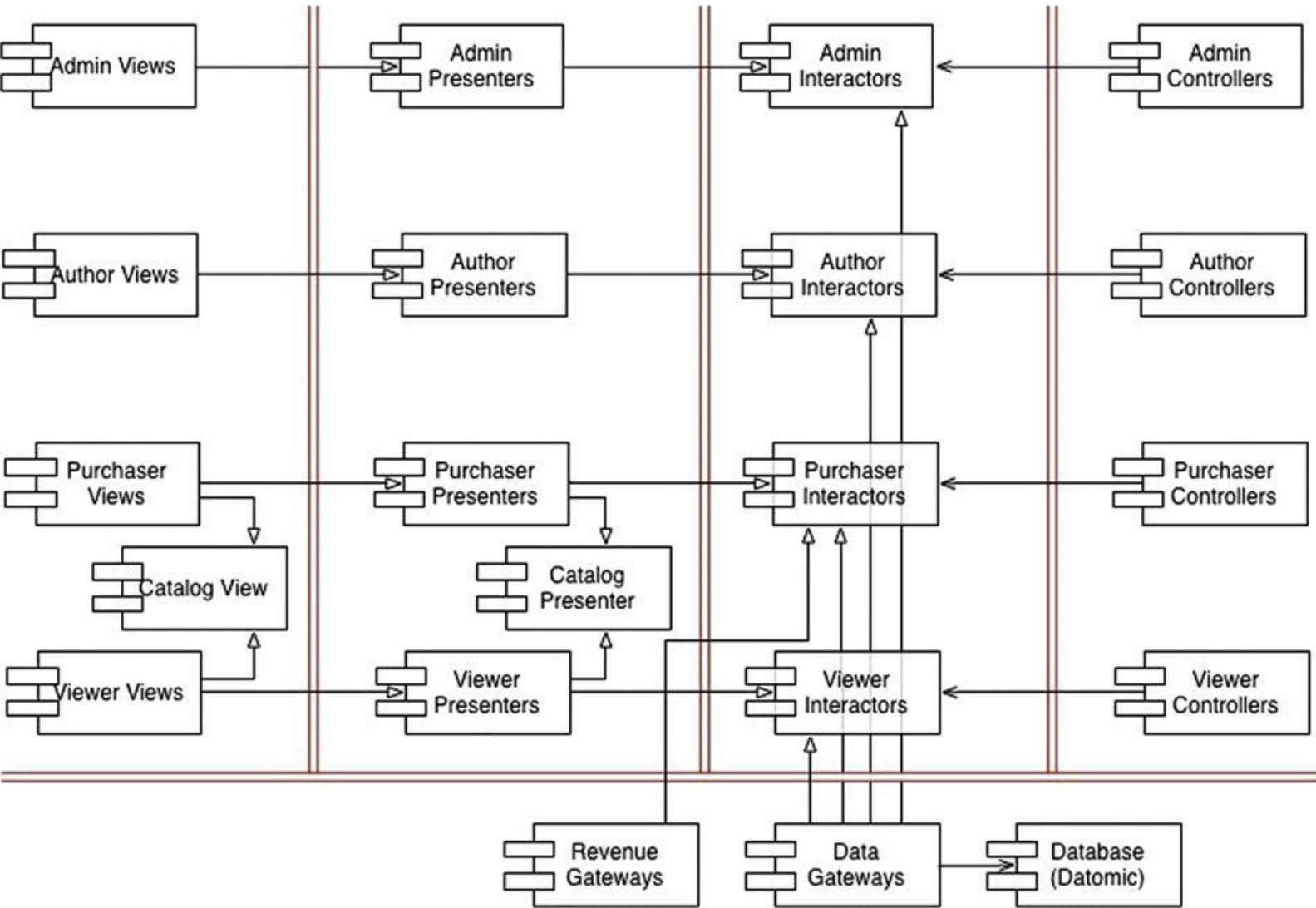
CITY HALL



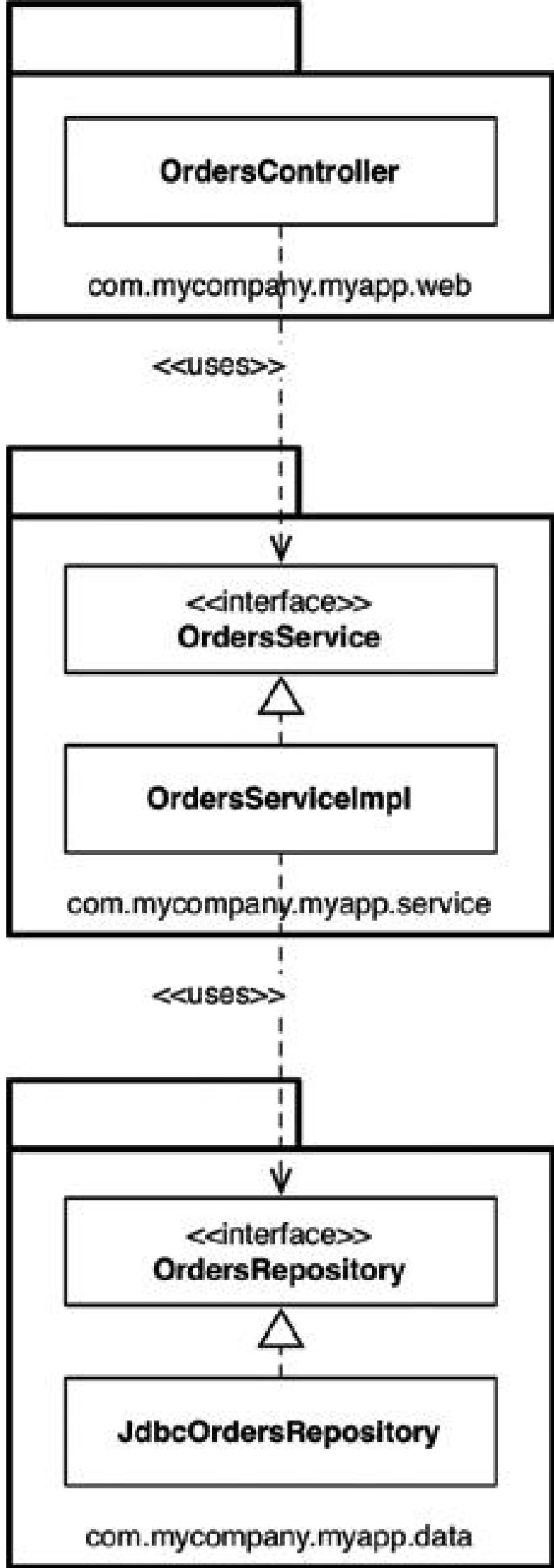


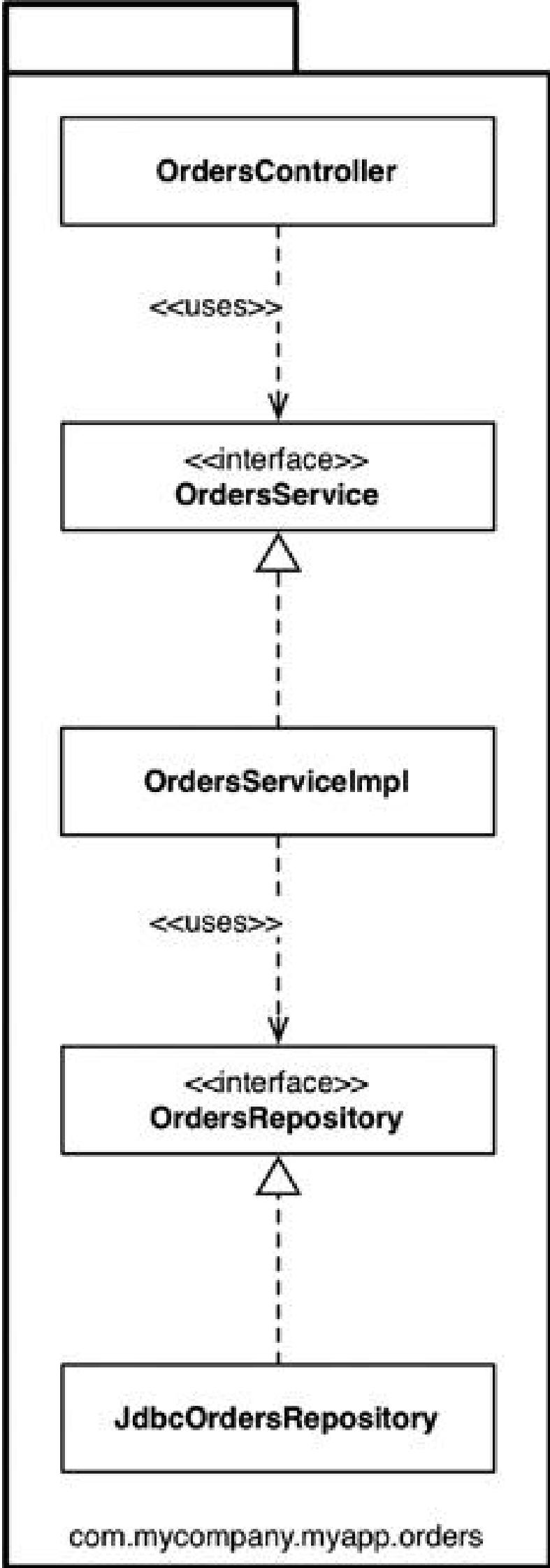
VCR TAPES





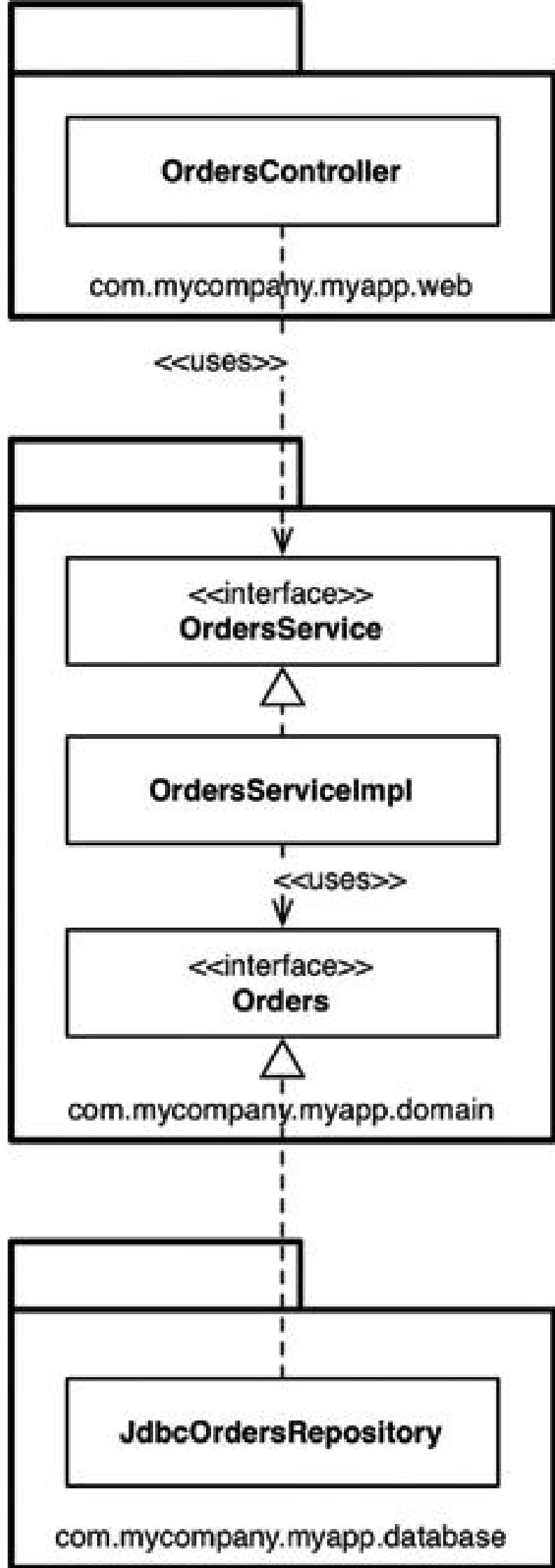


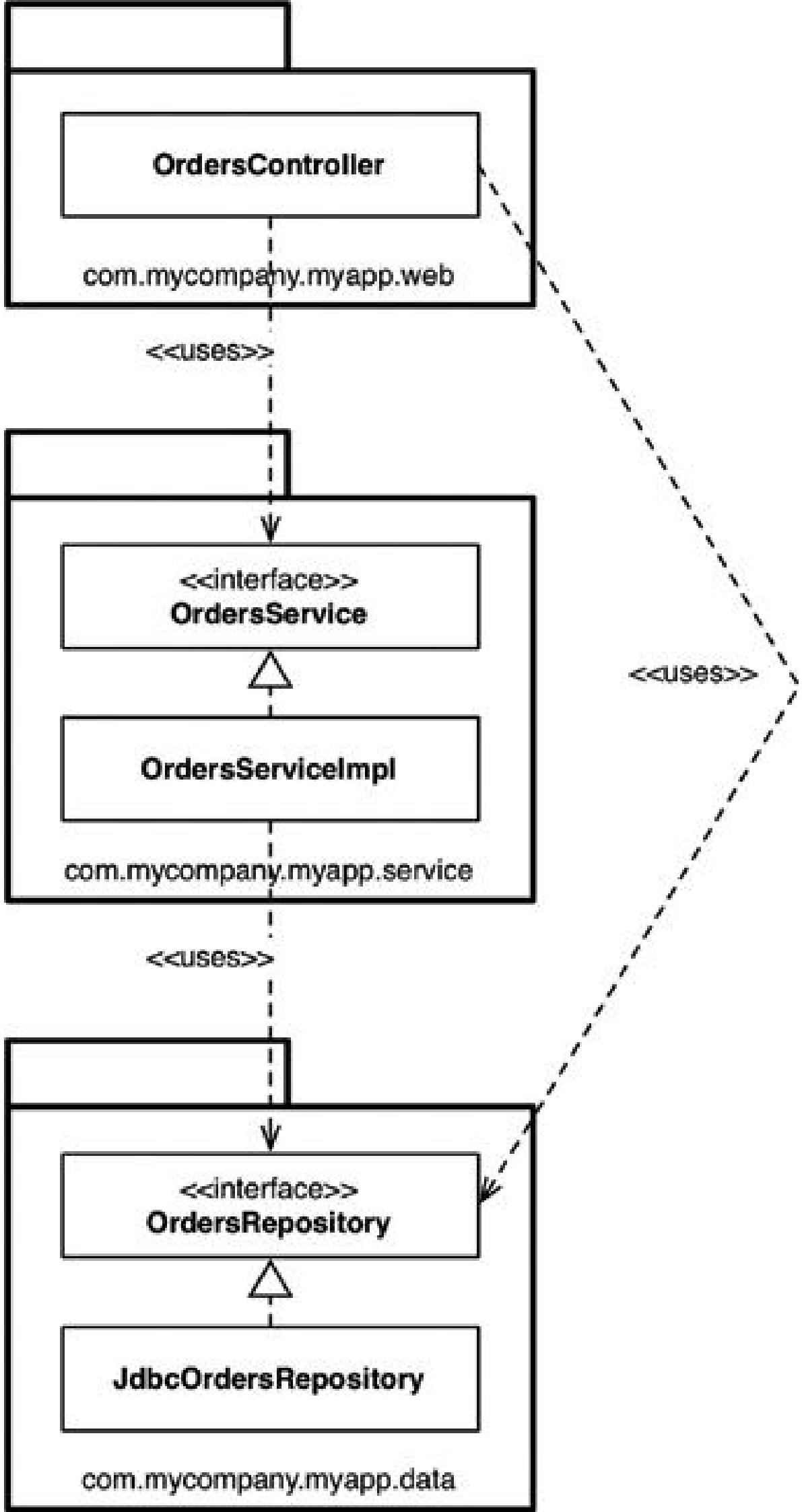


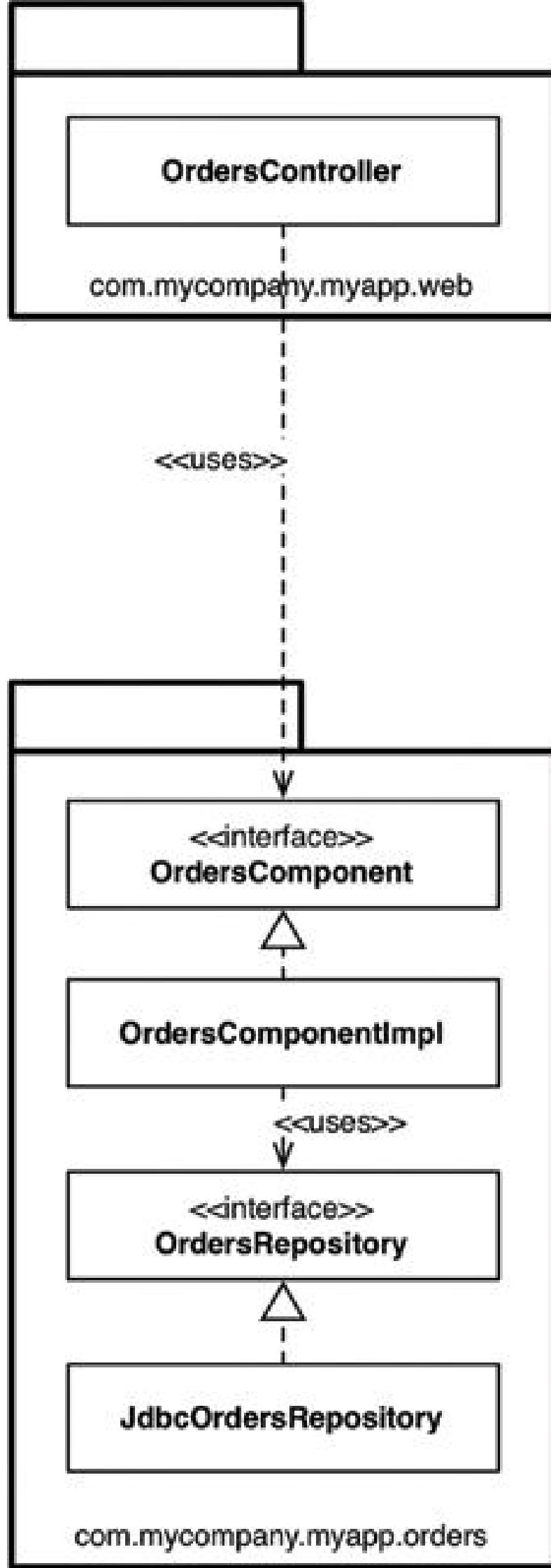


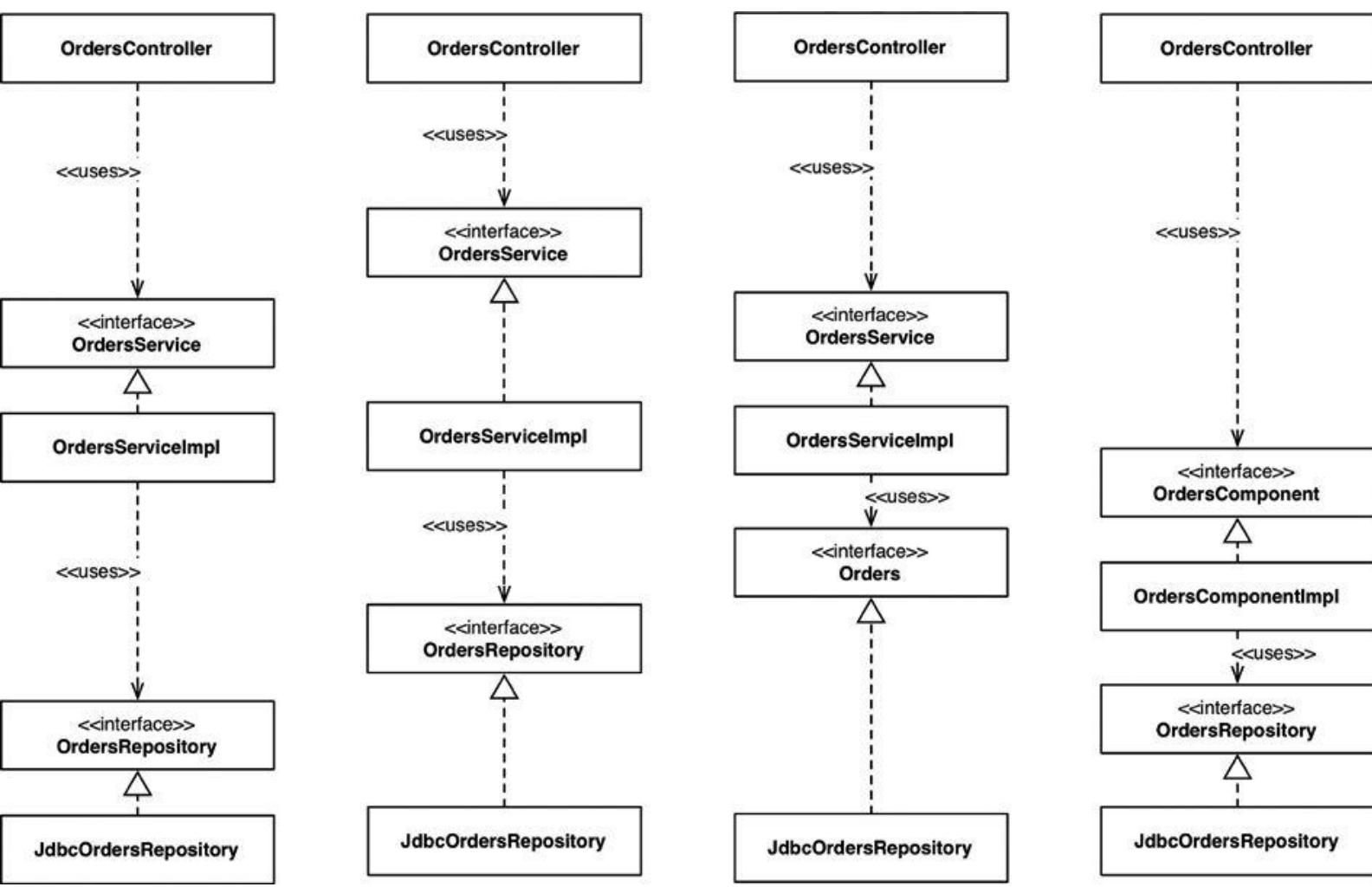
Infrastructure

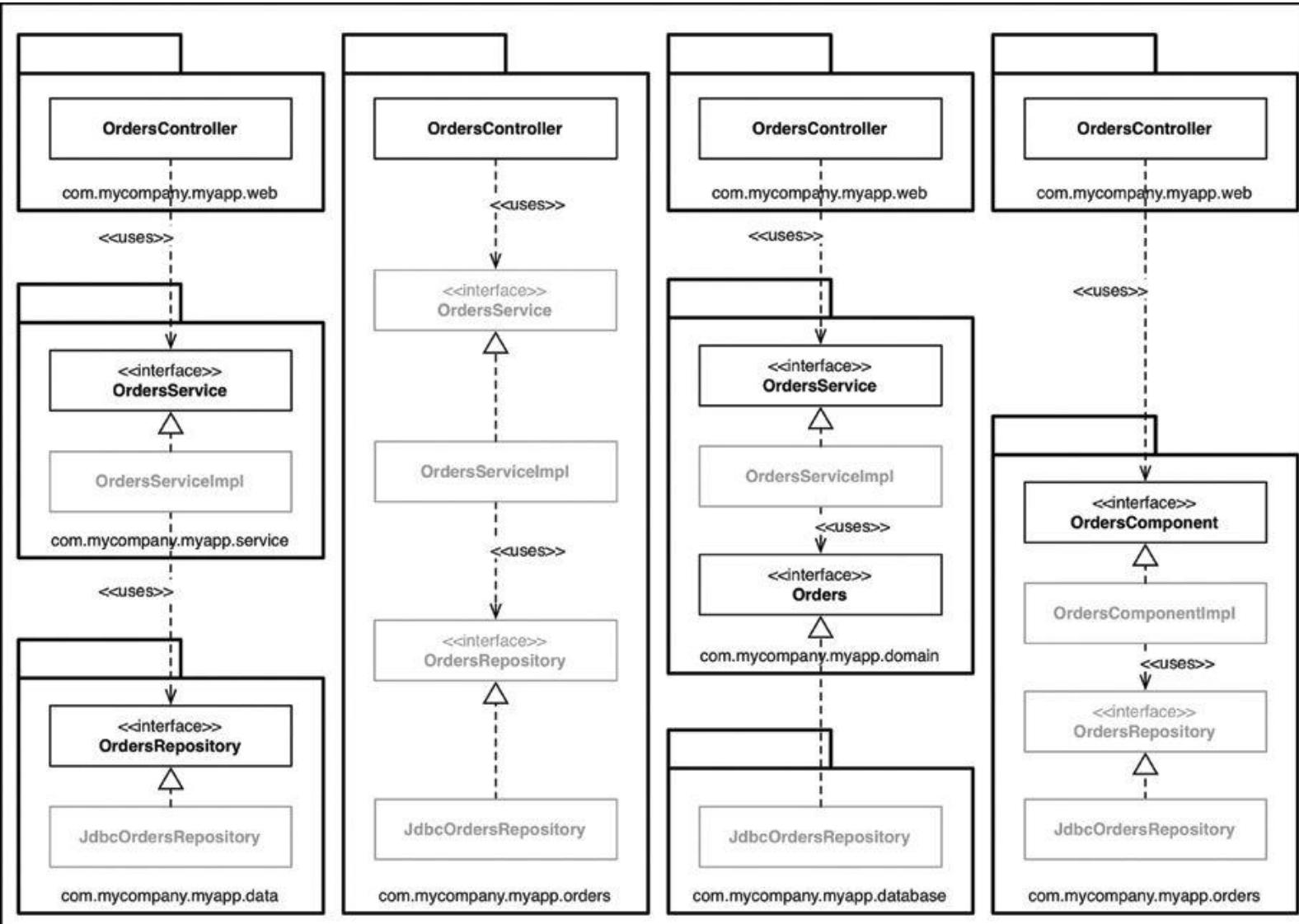
Domain









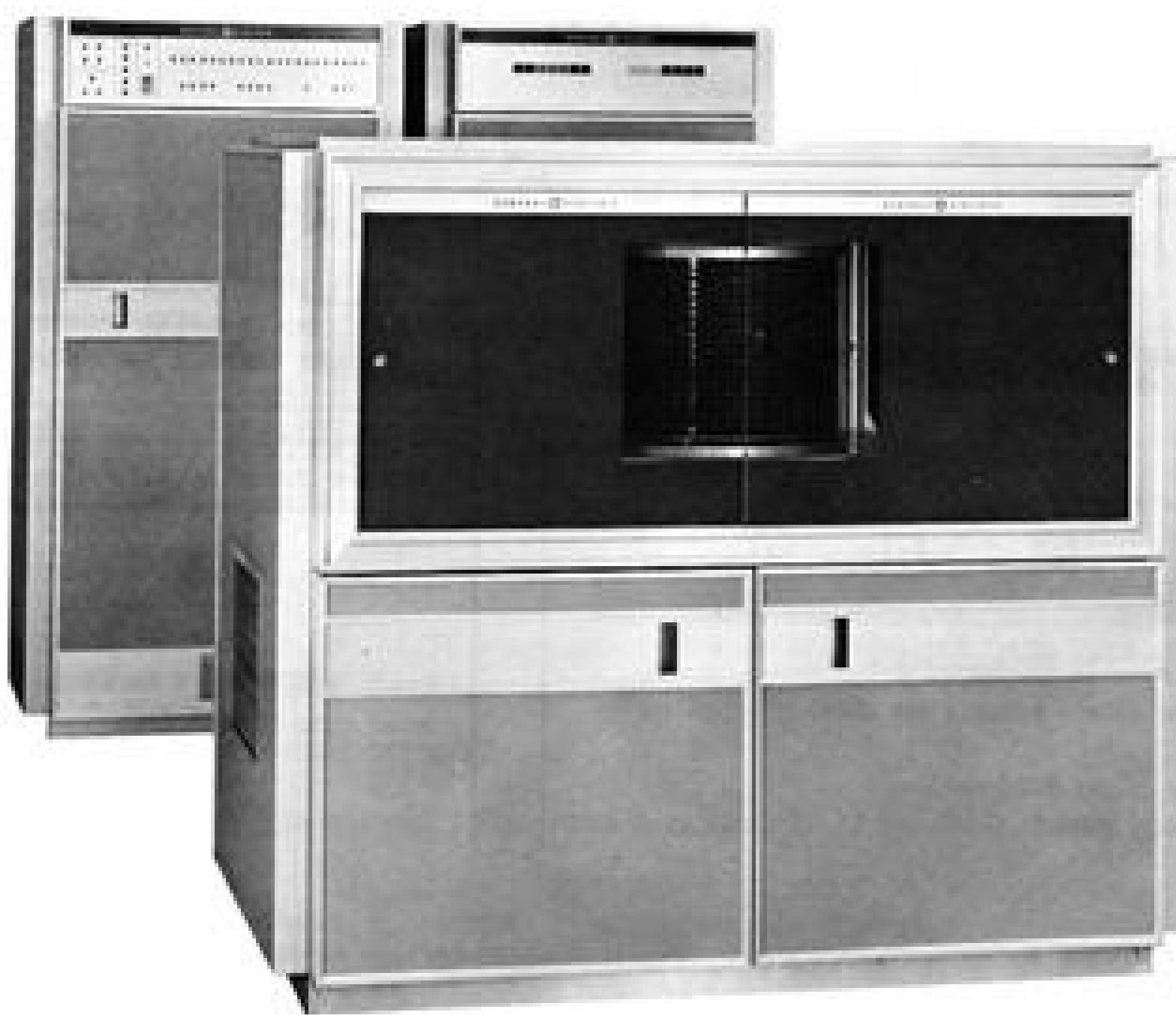


Infrastructure

Domain







MASS RANDOM ACCESS DATA STORAGE UNIT



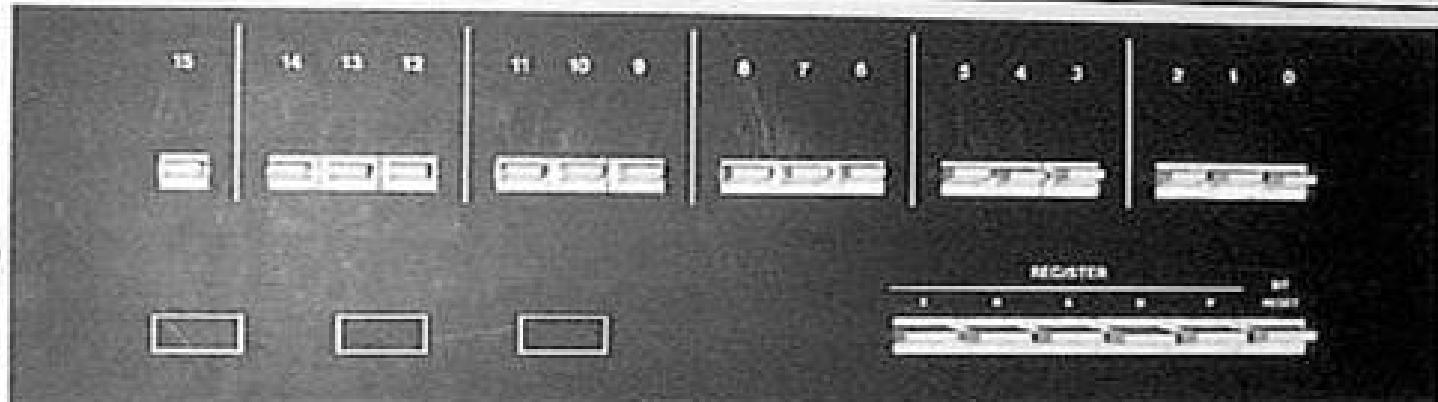


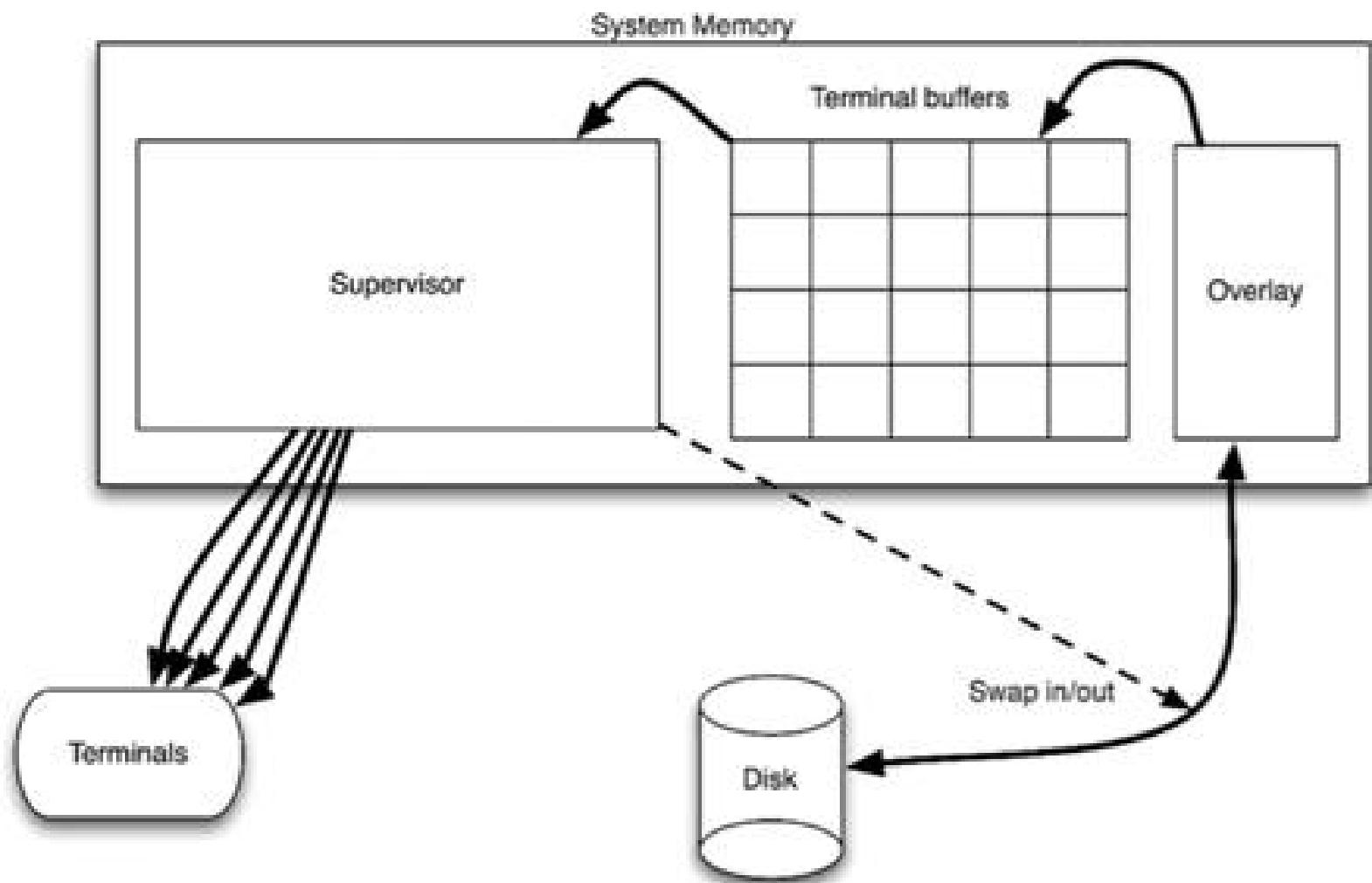




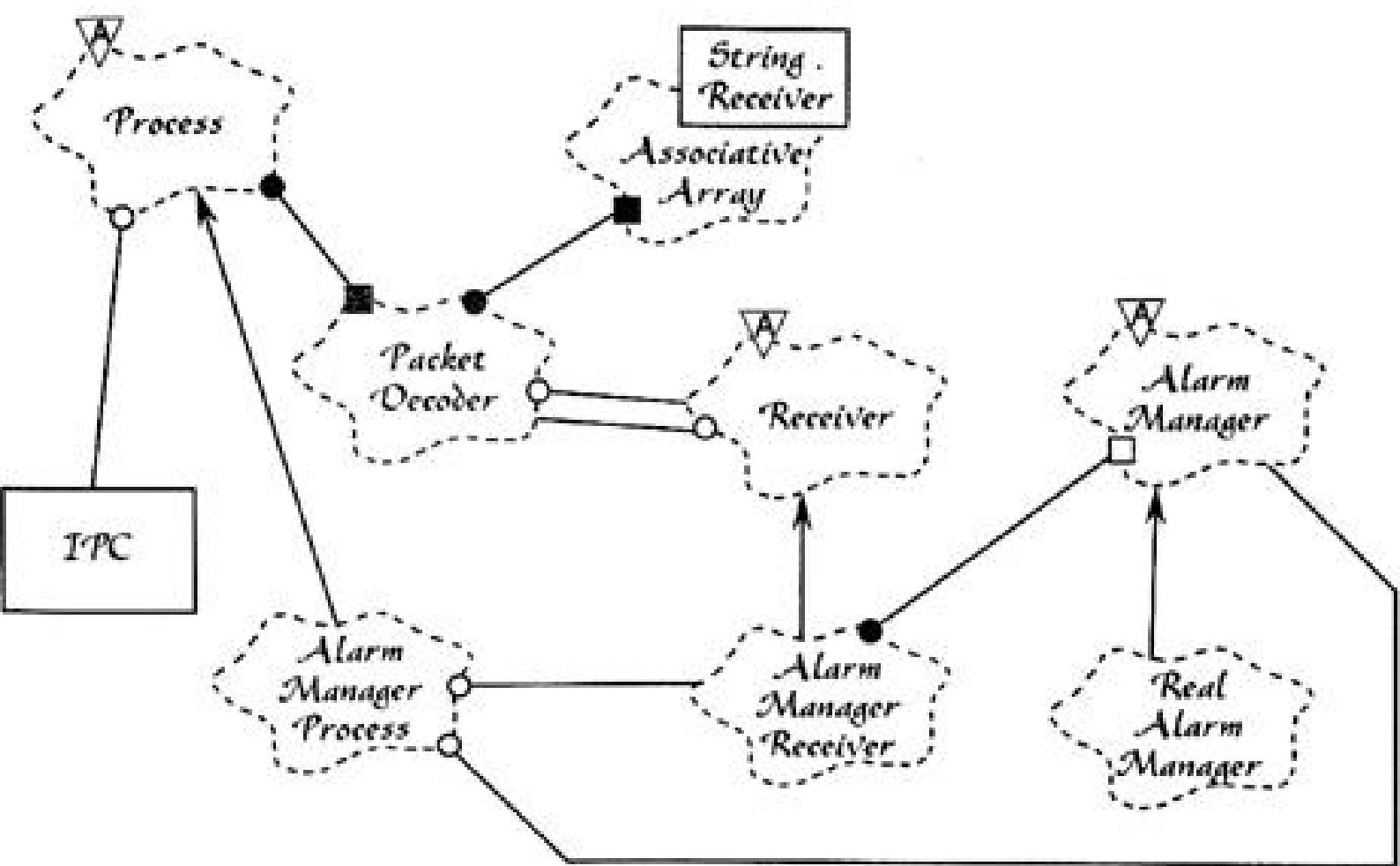
varian data machines

620/L-100











Register Your Product at informit.com/register Access additional benefits and **save 35%** on your next purchase

- Automatically receive a coupon for 35% off your next purchase, valid for 30 days. Look for your code in your InformIT cart or the Manage Codes section of your account page.
- Download available product updates.
- Access bonus material if available.
- Check the box to hear from us and receive exclusive offers on new editions and related products.

InformIT.com—The Trusted Technology Learning Source

InformIT is the online home of information technology brands at Pearson, the world's foremost education company. At InformIT.com, you can:

- Shop our books, eBooks, software, and video training
- Take advantage of our special offers and promotions (informit.com/promotions)
- Sign up for special offers and content newsletter (informit.com/newsletters)
- Access thousands of free chapters and video lessons

Connect with InformIT—Visit informit.com/community



Addison-Wesley · Adobe Press · Cisco Press · Microsoft Press · Pearson IT Certification · Prentice Hall · Que · Sams · Peachpit Press

point.h

```
struct Point;
struct Point* makePoint(double x, double y);
double distance (struct Point *p1, struct Point *p2);
```

point.c

```
#include "point.h"
#include <stdlib.h>
#include <math.h>

struct Point {
    double x,y;
};

struct Point* makepoint(double x, double y) {
    struct Point* p = malloc(sizeof(struct Point));
    p->x = x;
    p->y = y;
    return p;
}

double distance(struct Point* p1, struct Point* p2) {
    double dx = p1->x - p2->x;
    double dy = p1->y - p2->y;
    return sqrt(dx*dx+dy*dy);
}
```

point.h

```
class Point {  
public:  
    Point(double x, double y);  
    double distance(const Point& p) const;  
  
private:  
    double x;  
    double y;  
};
```

point.cc

```
#include "point.h"
#include <math.h>

Point::Point(double x, double y)
: x(x), y(y)
{ }

double Point::distance(const Point& p) const {
    double dx = x-p.x;
    double dy = y-p.y;
    return sqrt(dx*dx + dy*dy);
}
```

namedPoint.h

```
struct NamedPoint;  
  
struct NamedPoint* makeNamedPoint(double x, double y, char* name);  
void setName(struct NamedPoint* np, char* name);  
char* getName(struct NamedPoint* np);
```

namedPoint.c

```
#include "namedPoint.h"
#include <stdlib.h>

struct NamedPoint {
    double x,y;
    char* name;
};

struct NamedPoint* makeNamedPoint(double x, double y, char* name) {
    struct NamedPoint* p = malloc(sizeof(struct NamedPoint));
    p->x = x;
    p->y = y;
    p->name = name;
    return p;
}

void setName(struct NamedPoint* np, char* name) {
    np->name = name;
}

char* getName(struct NamedPoint* np) {
    return np->name;
}
```

main.c

```
#include "point.h"
#include "namedPoint.h"
#include <stdio.h>

int main(int ac, char** av) {
    struct NamedPoint* origin = makeNamedPoint(0.0, 0.0, "origin");
    struct NamedPoint* upperRight = makeNamedPoint
        (1.0, 1.0, "upperRight");
    printf("distance=%f\n",
        distance(
            (struct Point*) origin,
            (struct Point*) upperRight));
}
```

```
#include <stdio.h>

void copy() {
    int c;
    while ((c=getchar()) != EOF)
        putchar(c);
}
```

```
struct FILE {
    void (*open) (char* name, int mode);
    void (*close)();
    int (*read)();
    void (*write) (char);
    void (*seek) (long index, int mode);
};
```

```
#include "file.h"

void open(char* name, int mode) /*...*/
void close() /*...*/;
int read() {int c; /*...*/ return c;}
void write(char c) /*...*/
void seek(long index, int mode) /*...*/

struct FILE console = {open, close, read, write, seek};
```

```
extern struct FILE* STDIN;

int getchar() {
    return STDIN->read();
}
```

```
public class Squint {  
    public static void main(String args[]) {  
        for (int i=0; i<25; i++)  
            System.out.println(i*i);  
    }  
}
```

```
(println (take 25 (map (fn [x] (* x x)) (range))))
```

```
(println ; _____ Print
(take 25 ; _____ the first 25
(map (fn [x] (* x x)) ;__ squares
(range))) ; _____ of Integers
```

```
(def counter (atom 0)) ; initialize counter to 0  
(swap! counter inc)      ; safely increment counter.
```

```
Rectangle r = ...  
r.setWidth(5);  
r.setHeight(2);  
assert(r.area() == 10);
```

purplecab.com/driver/Bob

purplecab.com/driver/Bob

/pickupAddress/24 Maple St.

/pickupTime/153

/destination/ORD

```
if (driver.getDispatchUri().startsWith("acme.com")) ...
```

URI	Dispatch Format
Acme.com	/pickupAddress/%s/pickupTime/%s/dest/%s
.	/pickupAddress/%s/pickupTime/%s/destination/%s

*200
TLS
START, CLA
TAD BUFR
JMS GETSTR
CLA
TAD BUFR
JMS PUTSTR
JMP START
BUFR, 3000

GETSTR, 0
DCA PTR
NXTCH, KSF
JMP -1
KRB
DCA I PTR
TAD I PTR
AND K177
ISZ PTR
TAD MCR
SZA
JMP NXTCH

K177, 177
MCR, -15

*200
TLS
START, CLA
TAD BUFR
JMS GETSTR
CLA
TAD BUFR
JMS PUTSTR
JMP START
BUFR, 3000

GETSTR, 0
DCA PTR
NXTCH, KSF
JMP -1
KRB
DCA I PTR
TAD I PTR
AND K177
ISZ PTR
TAD MCR
SZA
JMP NXTCH

K177, 177
MCR, -15

PRTCHR, 0

TSF

JMP .-1

TLS

JMP I PRTCHR

```
function encrypt() {  
    while(true)  
        writeChar(translate(readChar()));  
}
```



```
"cold",
"drafty",
"dreadful"

};

private static final String[] shapes = new String[] {
    "round",
    "square",
    "oval",
    "irregular",
    "long",
    "craggy",
    "rough",
    "tall",
    "narrow"
};
```

```
private static final String[] cavernTypes = new String[] {  
    "cavern",  
    "room",  
    "chamber",  
    "catacomb",  
    "crevasse",  
    "cell",  
    "tunnel",  
    "passageway",  
    "hall",  
    "expanse"  
};
```

```
private static final String[] adornments = new String[] {
```

"smelling of sulfur",
"with engravings on the walls",
"with a bumpy floor",
",",
"littered with garbage",
"spattered with guano",
"with piles of Wumpus droppings",
"with bones scattered around",
"with a corpse on the floor",
"that seems to vibrate",
"that feels stuffy",
"that fills you with dread"
};

```
public static void main(String[] args) throws IOException {
    game = HtwFactory.makeGame("htw.game.HuntTheWumpusFacade",
                               new Main());
    createMap();
    BufferedReader br =
        new BufferedReader(new InputStreamReader(System.in));
    game.makeRestCommand().execute();
    while (true) {
        System.out.println(game.getPlayerCavern());
        System.out.println("Health: " + hitPoints + " arrows: " +
                           game.getQuiver());
        HuntTheWumpus.Command c = game.makeRestCommand();
```

```
System.out.println(">");

String command = br.readLine();
if (command.equalsIgnoreCase("e"))
    c = game.makeMoveCommand(EAST);
else if (command.equalsIgnoreCase("w"))
    c = game.makeMoveCommand(WEST);
else if (command.equalsIgnoreCase("n"))
    c = game.makeMoveCommand(NORTH);
else if (command.equalsIgnoreCase("s"))
    c = game.makeMoveCommand(SOUTH);
else if (command.equalsIgnoreCase("r"))
    c = game.makeRestCommand();
else if (command.equalsIgnoreCase("sw"))
    c = game.makeShootCommand(WEST);
else if (command.equalsIgnoreCase("se"))
    c = game.makeShootCommand(EAST);
else if (command.equalsIgnoreCase("sn"))
    c = game.makeShootCommand(NORTH);
else if (command.equalsIgnoreCase("ss"))
    c = game.makeShootCommand(SOUTH);
else if (command.equalsIgnoreCase("q"))
    return;

c.execute();

}
```

```
System.out.println(">");

String command = br.readLine();
if (command.equalsIgnoreCase("e"))
    c = game.makeMoveCommand(EAST);
else if (command.equalsIgnoreCase("w"))
    c = game.makeMoveCommand(WEST);
else if (command.equalsIgnoreCase("n"))
    c = game.makeMoveCommand(NORTH);
else if (command.equalsIgnoreCase("s"))
    c = game.makeMoveCommand(SOUTH);
else if (command.equalsIgnoreCase("r"))
    c = game.makeRestCommand();
else if (command.equalsIgnoreCase("sw"))
    c = game.makeShootCommand(WEST);
else if (command.equalsIgnoreCase("se"))
    c = game.makeShootCommand(EAST);
else if (command.equalsIgnoreCase("sn"))
    c = game.makeShootCommand(NORTH);
else if (command.equalsIgnoreCase("ss"))
    c = game.makeShootCommand(SOUTH);
else if (command.equalsIgnoreCase("q"))
    return;

c.execute();

}
```

```
private static void createMap() {  
    int nCaverns = (int) (Math.random() * 30.0 + 10.0);  
    while (nCaverns-- > 0)  
        caverns.add(makeName());  
  
    for (String cavern : caverns) {  
        maybeConnectCavern(cavern, NORTH);  
        maybeConnectCavern(cavern, SOUTH);  
        maybeConnectCavern(cavern, EAST);  
        maybeConnectCavern(cavern, WEST);  
    }  
  
    String playerCavern = anyCavern();  
    game.setPlayerCavern(playerCavern);  
    game.setWumpusCavern(anyOther(playerCavern));  
    game.addBatCavern(anyOther(playerCavern));  
    game.addBatCavern(anyOther(playerCavern));  
    game.addBatCavern(anyOther(playerCavern));  
  
    game.addPitCavern(anyOther(playerCavern));  
    game.addPitCavern(anyOther(playerCavern));  
    game.addPitCavern(anyOther(playerCavern));  
  
    game.setQuiver(5);  
}  
  
// much code removed...  
}
```

```
private static void createMap() {  
    int nCaverns = (int) (Math.random() * 30.0 + 10.0);  
    while (nCaverns-- > 0)  
        caverns.add(makeName());  
  
    for (String cavern : caverns) {  
        maybeConnectCavern(cavern, NORTH);  
        maybeConnectCavern(cavern, SOUTH);  
        maybeConnectCavern(cavern, EAST);  
        maybeConnectCavern(cavern, WEST);  
    }  
  
    String playerCavern = anyCavern();  
    game.setPlayerCavern(playerCavern);  
    game.setWumpusCavern(anyOther(playerCavern));  
    game.addBatCavern(anyOther(playerCavern));  
    game.addBatCavern(anyOther(playerCavern));  
    game.addBatCavern(anyOther(playerCavern));  
  
    game.addPitCavern(anyOther(playerCavern));  
    game.addPitCavern(anyOther(playerCavern));  
    game.addPitCavern(anyOther(playerCavern));  
  
    game.setQuiver(5);  
}  
  
// much code removed...  
}
```

```
ISR(TIMER1_vect) { ... }

ISR(INT2_vect) { ... }

void btn_Handler(void) { ... }

float calc_RPM(void) { ... }

static char Read_RawData(void) { ... }

void Do_Average(void) { ... }

void Get_Next_Measurement(void) { ... }

void Zero_Sensor_1(void) { ... }

void Zero_Sensor_2(void) { ... }

void Dev_Control(char Activation) { ... }

char Load_FLASH_Setup(void) { ... }

void Save_FLASH_Setup(void) { ... }

void Store_DataSet(void) { ... }

float bytes2float(char bytes[4]) { ... }

void Recall_DataSet(void) { ... }

void Sensor_init(void) { ... }

void uC_Sleep(void) { ... }
```

```
float calc_RPM(void) { ... }

void Do_Average(void) { ... }

void Get_Next_Measurement(void) { ... }

void Zero_Sensor_1(void) { ... }

void Zero_Sensor_2(void) { ... }
```

```
ISR(TIMER1_vect) { ... }*
```

```
ISR(INT2_vect) { ... }
```

```
void uC_Sleep(void) { ... }
```

Functions that react to the on off button press

```
void btn_Handler(void) { ... }
```

```
void Dev_Control(char Activation) { ... }
```

A Function that can get A/D input readings from the hardware

```
static char Read_RawData(void) { ... }
```

```
char Load_FLASH_Setup(void) { ... }
void Save_FLASH_Setup(void) { ... }
void Store_DataSet(void) { ... }
float bytes2float(char bytes[4]) { ... }
void Recall_DataSet(void) { ... }
```

```
void Sensor_init(void) { . . . }
```

```
#ifndef _ACME_STD_TYPES
```

```
#define _ACME_STD_TYPES
```

```
#if defined(_ACME_X42)
    typedef unsigned int          Uint_32;
    typedef unsigned short        Uint_16;
    typedef unsigned char         Uint_8;

    typedef int                  Int_32;
    typedef short                Int_16;
    typedef char                 Int_8;

#elif defined(_ACME_A42)
    typedef unsigned long         Uint_32;
    typedef unsigned int          Uint_16;
    typedef unsigned char         Uint_8;

    typedef long                 Int_32;
    typedef int                  Int_16;
    typedef char                 Int_8;

#else
    #error <acmetypes.h> is not supported for this environment
#endif

#endif
```

```
#ifndef _STDINT_H_
#define _STDINT_H_

#include <acmetypes.h>

typedef Uint_32 uint32_t;
typedef Uint_16 uint16_t;
typedef Uint_8  uint8_t;

typedef Int_32 int32_t;
typedef Int_16 int16_t;
typedef Int_8  int8_t;

#endif
```

```
#ifndef _STDINT_H_
#define _STDINT_H_

#include <acmetypes.h>

typedef Uint_32 uint32_t;
typedef Uint_16 uint16_t;
typedef Uint_8  uint8_t;

typedef Int_32 int32_t;
typedef Int_16 int16_t;
typedef Int_8  int8_t;

#endif
```

```
void say_hi()  
{  
    IE = 0b11000000;  
  
    SBUF0 = (0x68);  
  
    while(TI_0 == 0);  
  
    TI_0 = 0;  
  
    SBUF0 = (0x69);  
  
    while(TI_0 == 0);  
  
    TI_0 = 0;  
  
    SBUF0 = (0x0a);  
  
    while(TI_0 == 0);  
  
    TI_0 = 0;  
  
    SBUF0 = (0x0d);  
  
    while(TI_0 == 0);  
  
    TI_0 = 0;  
  
    IE = 0b11010000;  
}
```

```
void say_hi()  
{  
    IE = 0b11000000;  
  
    SBUF0 = (0x68);  
  
    while(TI_0 == 0);  
  
    TI_0 = 0;  
  
    SBUF0 = (0x69);  
  
    while(TI_0 == 0);  
  
    TI_0 = 0;  
  
    SBUF0 = (0x0a);  
  
    while(TI_0 == 0);  
  
    TI_0 = 0;  
  
    SBUF0 = (0x0d);  
  
    while(TI_0 == 0);  
  
    TI_0 = 0;  
  
    IE = 0b11010000;  
}
```

/pno 8475551212 /noise /dropped-calls