



Introduction to Software Engineering

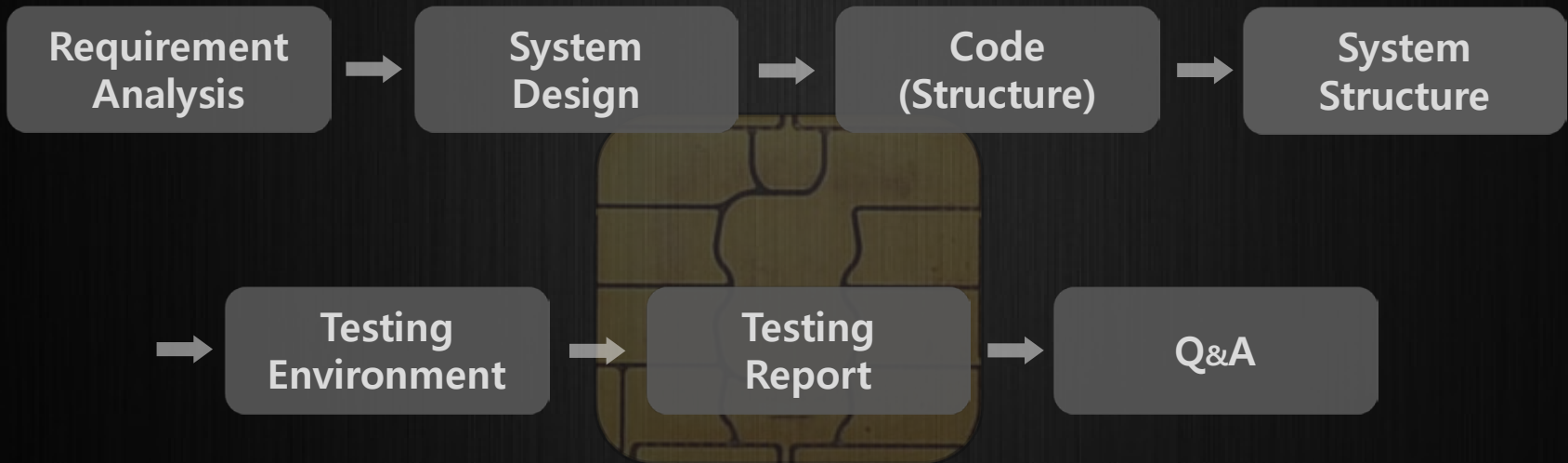


[2014SE_B][T1]

201111347 김태호
201111360 손준익

201111356 박준한
201111367 여승훈

Index



Reference	DS-2014SE-PTS-SRS_Ver1.0		[2014_SE_B][T1]T1_Answer_Report_Ver1.0
	[2014SE_B][T1]SRA Ver 4.0	[2014_SE_B][T1]T1_STP_Ver2.0	T7_STP_Ver1.0
	[2014SE_B][T1]SDS Ver 3.0	[2014_SE_B][T1]T1_STR_Ver2.0	T7_STR_Ver1.0

Requirement Analysis

System Design

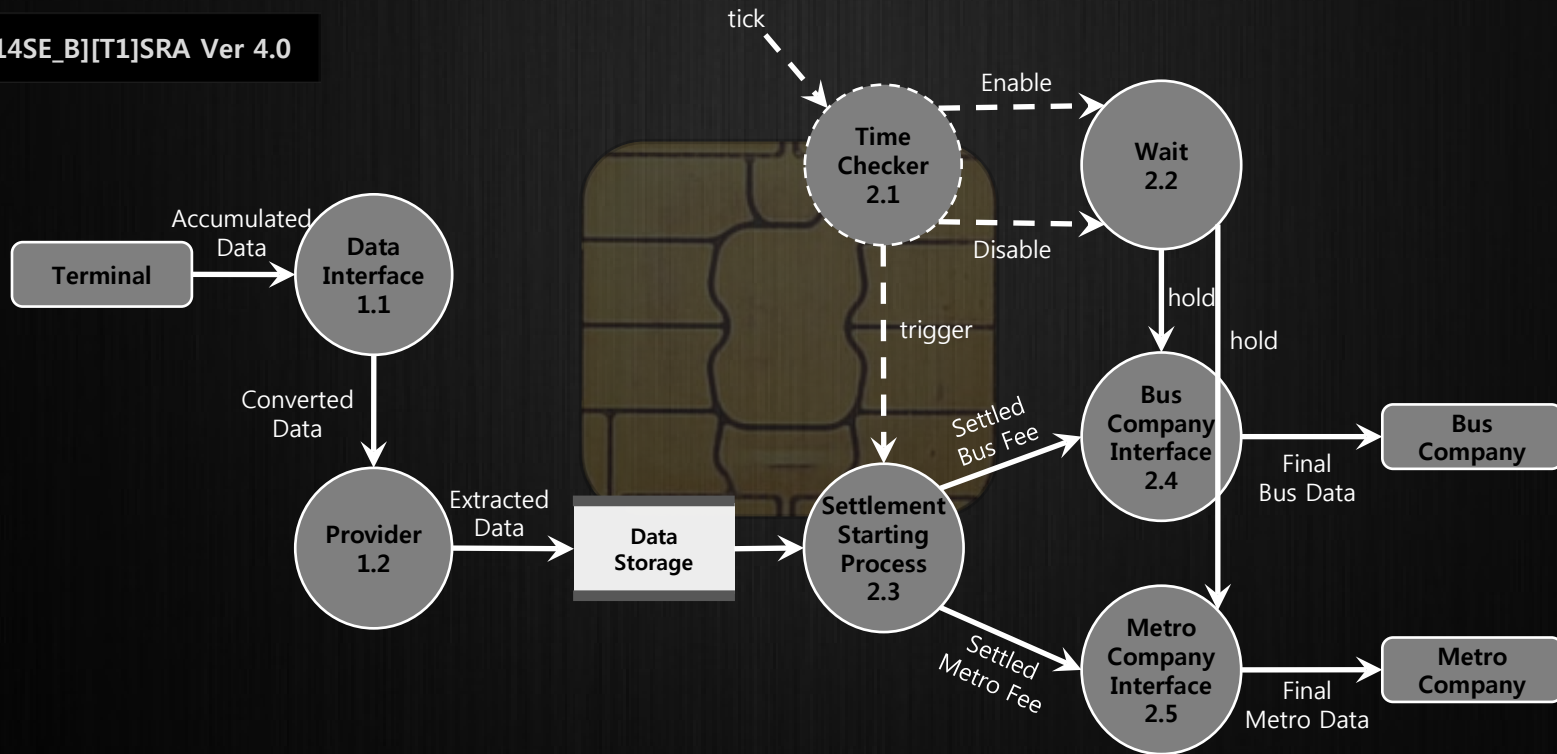
Code (Structure)

System Structure

Terminal System DFD

Settlement System DFD

[2014SE_B][T1]SRA Ver 4.0



Requirement Analysis

System Design

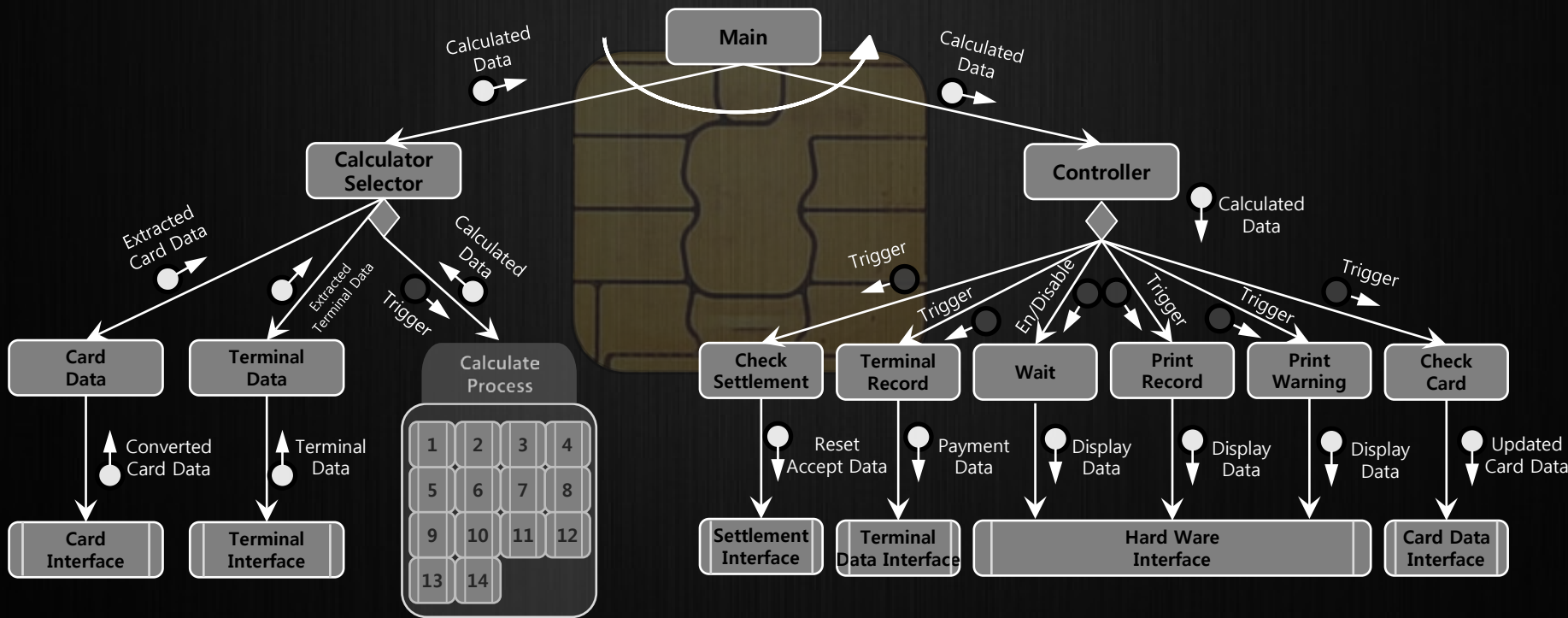
Code (Structure)

System Structure

Terminal System SD

Settlement System SD

[2014SE_B][T1]SDS Ver 3.0



Requirement Analysis

System Design

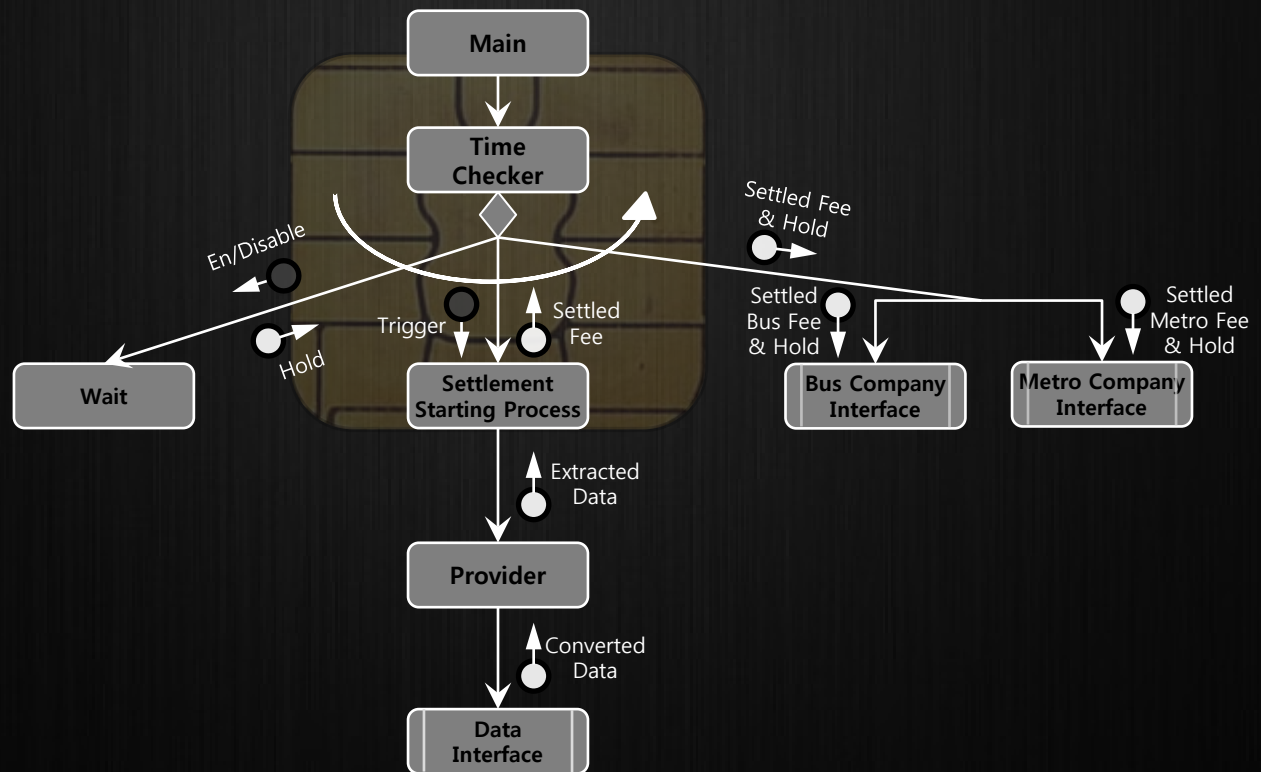
Code (Structure)

System Structure

Terminal System SD

Settlement System SD

[2014SE_B][T1]SDS Ver 3.0



Requirement
Analysis

System
Design

Code
(Structure)

System
Structure

Terminal System

솔루션 'Terminal' (1 프로젝트)

- Terminal
 - 리소스 파일
 - 소스 파일
 - Calculator_Selector.c
 - Controller.c
 - main.c
 - 외부 종속성
 - 헤더 파일
 - Calculator_Selector.h
 - Controller.h

Settlement System

솔루션 'Settlement' (1 프로젝트)

- Settlement
 - 리소스 파일
 - 소스 파일
 - main.c
 - Stack.c
 - TimeChecker.c
 - 외부 종속성
 - 헤더 파일
 - Stack.h
 - TimeChecker.h

Requirement
Analysis

System
Design

Code
(Structure)

System
Structure

Terminal
System

Settlement
System

Calculator Selector

```
void card_Data(ECD* result); //카드 txt파일을 읽어옴
void terminal_Data(ETD* result1, char current_date1[9], int select); //터미널 txt파일을 읽어옴

CD calculator_Selector(char current_time[13]);
CD Calculate_Bus_On_Unchecked(ECD ecd, ETD etd); //지하철->버스 미정산
CD Calculate_Bus_On_Standard(ECD ecd, ETD etd); //버스 기본 탑승
CD Calculate_Bus_On_Transfer(ECD ecd, ETD etd); //지하철->버스 환승

CD Calculate_Metro_On_Unchecked(ECD ecd, ETD etd); //버스->지하철 미정산
CD Calculate_Metro_On_Standard(ECD ecd, ETD etd); //지하철 기본
CD Calculate_Metro_On_Transfer(ECD ecd, ETD etd); //버스->지하철 환승

CD Calculate_Bus_Off_Transfered(ECD ecd, ETD etd); //버스 하차 환승시
CD Calculate_Bus_Off_Standard(ECD ecd, ETD etd); //버스 하차 기본

CD Calculate_Metro_Off_Transfered(ECD ecd, ETD etd); //지하철 하차 환승시
CD Calculate_Metro_Off_Standard(ECD ecd, ETD etd); //지하철 하차 기본
int board_Info();
```

Requirement
Analysis

System
Design

Code
(Structure)

System
Structure

Terminal
System

Settlement
System

Controller

```
void Wait(CD cd);  
void Print_Result(CD cd);  
void Print_Warning(CD cd);  
void Check_Card(CD cd);  
void Terminal_Record(CD cd);  
void Check_Settlement(CD cd);  
void Terminal_Data_Interface(PD pd);  
void Card_Data_Interface(UCD ucd);  
void Hardware_Interface(DD dd);  
void Settlement_Interface(RAD rad);  
void Controller(CD cd);  
  
//Card Data가 들어오지 않았을 경우 Wait 상태 표시  
//'Calculated_Balance>=0'일 경우 탑승가능 표시  
//'Calculated_Balance<0'일 경우 탑승불가 표시  
//'Calculated_Balance>=0'일 경우 카드 갱신 데이터 전송  
//'Calculated_Balance>=0'일 경우 결제 기록 데이터 전송  
//'현재시간%180==0'일 경우 Terminal 초기화 여부 데이터 전송  
//PD를 이용하여 TerminalLn.txt를 생성하는 Interface  
//UCD를 이용하여 Card.txt를 생성하는 Interface  
//Hardware에 Wait,Correct,Warning 을 선택하여 display하는 Interface  
//'현재시간%180==0'일 경우 Terminal 초기화 여부 데이터를 받는 Interface
```

Requirement
Analysis

System
Design

Code
(Structure)

System
Structure

Calculator Selector

Terminal
System

Settlement
System

```
typedef struct Converted_Card_Data{
    char time[14];
    char transportation[2];
    int on_board;
    int balance;
    char Terminal_ID[5];
}CCD;

typedef struct Termina_Data{
    char time[14];
    char transportation[2];
    int on_board;
    char Terminal_ID[5];
}TD;

typedef struct Extracted_Card_Data{
    int on_board;           //승하차 여부, boolean 형식으로 1이면 True, 0이면 False,
    //현재 내린상태면 0이고, 타고있는상태면 1이다.
    int balance;           //카드 잔액
    char Terminal_ID[2];   //터미널 ID
    //여기서 추출되는건 모두 마지막이라는데 중요
    char time1[13];        //바로전 태그된 시간
    char time2[13];        //바로전전 태그된시간
    char c_time[13];
    char transportation[2]; //교통수단 'B' or 'S'로 표시
    char transportation1[2]; //전교통수단 'B' or 'S'로 표시
    // char transportation2[2]; //전전교통수단 'B' or 'S'로 표시
    int count;
}ECD ;
```

Requirement
Analysis

System
Design

Code
(Structure)

System
Structure

Terminal
System

Settlement
System

Calculator Selector

```
typedef struct Extracted_Terminal_Data{
    char transportation[2]; //교통수단 'B' or 'S'로 표시
    char Terminal_ID[2];
    int count; //터미널이 여태 찍힌횟수 count
}ETD;
typedef struct Calculated_Data { //Calculaotr_Selector에서 받은 Data
    char time[13]; //태그된 시간
    char transportation[2]; //교통수단 'B' or 'S'로 표시
    int on_board; //승하차 여부, boolean 형식으로 1이면 ON, 0이면 OFF
    int balance; //카드 잔액
    int fee; //금액
    int Calculated_Balance; //카드잔액-금액'으로 '>=0'일때 탑승 가능
    char Terminal_ID[2]; //단말기 ID
    int count; //단말기 태그 횟수
}CD;
```

Requirement
Analysis

System
Design

Code
(Structure)

System
Structure

Terminal
System

Settlement
System

Controller

```
typedef struct Payment_Data {
    char time[13];
    char transportation[2];
    int on_board;
    int fee;
    char Terminal_ID[2];
    int count;
}PD;

typedef struct Updated_Card_Data {
    char time[13];
    char transportation[2];
    int on_board;
    int balance;
    char Terminal_ID[2];
    int count;
}UCD;

typedef struct Display_Data {
    char* Display;
    int fee;
    int calculated_Balance;
}DD;

typedef struct Reset_Accept_Data {
    char time[9];
    int reset;
}RAD;
```

//Terminal_Record에서 Terminal_Data_Interface로 넘어가는 Data
//태그된 시간
//교통수단 'B' or 'S'로 표시
//승하차 여부, boolean 형식으로 1이면 ON, 0이면 OFF
//금액
//단말기 ID
//단말기 태그 횟수

//Check_Card에서 Card_Data_Interface로 넘어가는 Data
//태그된 시간
//교통수단 'B' or 'S'로 표시
//승하차 여부, boolean 형식으로 1이면 ON, 0이면 OFF
//카드 잔액
//단말기 ID
//단말기 태그 횟수

//Hardware_Interface로 넘어가는 Data
//Display에 어떤 것을 출력할지 결정하는 Data
//금액
//'카드잔액-금액'

//Settlement_Inerface로 넘어가는 Data
//다음 날짜
//reset이 되었는지의 여부로 1이면 ON, 0이면 OFF

Requirement
Analysis

System
Design

Code
(Structure)

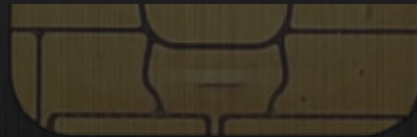
System
Structure

Terminal
System

Settlement
System

Settlement

```
void TimeChecker();  
int Wait(); // Hold Signal : 0/1  
SF* Settlement_Starting_Process();  
struct edstack* Provider();  
struct cdstack* Data_Interface(); // read Terminal file  
int Bus_Company_Interface(int settled_Fee,char* last_Tag,int hold_Signal); // make Bus Company file  
int Metro_Company_Interface(int settled_Fee,char* last_Tag,int hold_Signal); // make Metro Company file
```



Requirement
Analysis

System
Design

Code
(Structure)

System
Structure

Terminal
System

Settlement
System

Settlement

```
typedef struct Converted_Data{
    char time[13];
    char transportation;
    int on_Board;
    int fee;
    char Terminal_ID[2];
    int count;
}CD;

typedef struct Extracted_Data{
    char time[13];
    char transportation;
    int on_Board;
    int fee;
}ED;

typedef struct Settled_Fee{
    char time_Bus[13];
    char time_Metro[13];
    int settled_Bus_Fee;
    int settled_Metro_Fee;
}SF;
```

Requirement
Analysis

System
Design

Code
(Structure)

System
Structure

Clock

```
time:201412020624
```

Clock.txt

Terminal

```
=====
Choosing Tag
1.Metro      2.Bus
=====
2
<CLOCK: 201412020432 >>

TYPE: <<Calculate_Bus_Off_Standard>>
FEE : 0
CARD BALANCE : 954800

=====
Choosing Tag
1.Metro      2.Bus
=====
```

Settlement

```
Bus Company : get hold
Metro Company : get hold
time:201412020544
Bus Company : get hold
Metro Company : get hold
time:201412020552
Bus Company : get hold
Metro Company : get hold
time:201412020600
Bus Company : get hold
Metro Company : get hold
time:201412020608
Bus Company : get hold
Metro Company : get hold
time:201412020616
Bus Company : get hold
Metro Company : get hold
time:201412020624
```

Share
Time

Synchronize
Time

Requirement Analysis

System Design

Code (Structure)

System Structure

Terminal

```
=====
Choosing Tag
1.Metro      2.Bus
=====
2
<CLOCK: 201412020432 >>
TYPE: <<Calculate_Bus_Off_Standard>>
FEE : 0
CARD BALANCE : 954800

=====
Choosing Tag
1.Metro      2.Bus
=====
```

Terminal_Data_#DATE.txt

Settlement

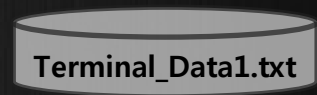
Settle Signal

OK.txt

Settled_(Bus or Metro)Fee_#DATE.txt

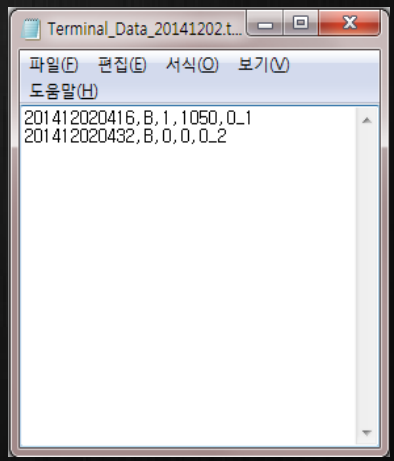
```
Bus Company : get hold
Metro Company : get hold
time:201412020544
Bus Company : get hold
Metro Company : get hold
time:201412020552
Bus Company : get hold
Metro Company : get hold
time:201412020600
Bus Company : get hold
Metro Company : get hold
time:201412020608
Bus Company : get hold
Metro Company : get hold
time:201412020616
Bus Company : get hold
Metro Company : get hold
time:201412020624
```

INITIAL STATE



Terminal

```
=====
Choosing Tag
1.Metro      2.Bus
=====
2
<CLOCK: 201412020432 >>
TYPE: <<Calculate_Bus_Off_Standard>>
FEE : 0
CARD BALANCE : 954800
=====
Choosing Tag
1.Metro      2.Bus
=====
```



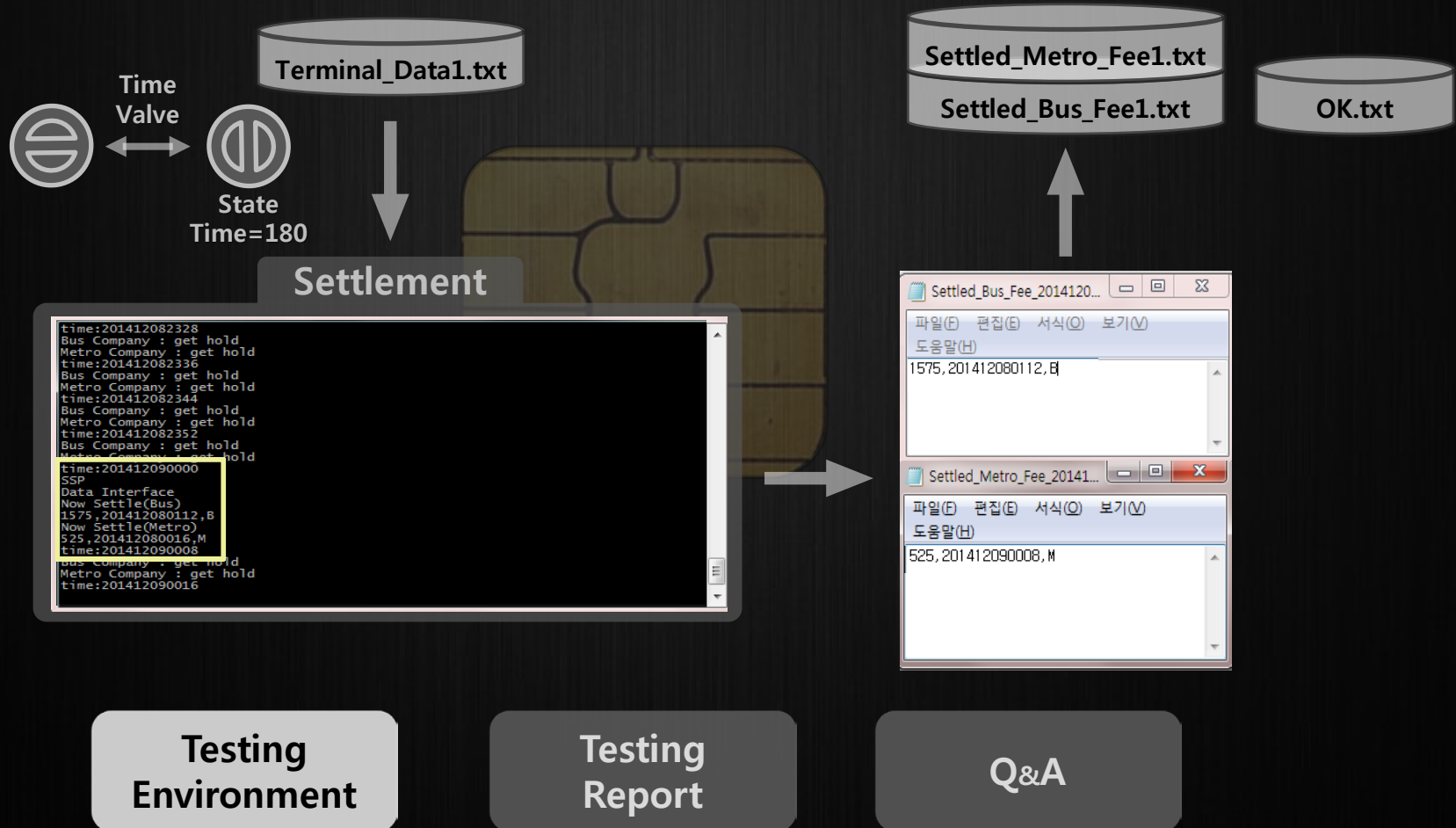
```
Terminal_Data_20141202.t...
파일(F) 편집(E) 서식(O) 보기(V)
도움말(H)
201412020416,B,1,1050,0_1
201412020432,B,0,0,0_2
```

Testing Environment

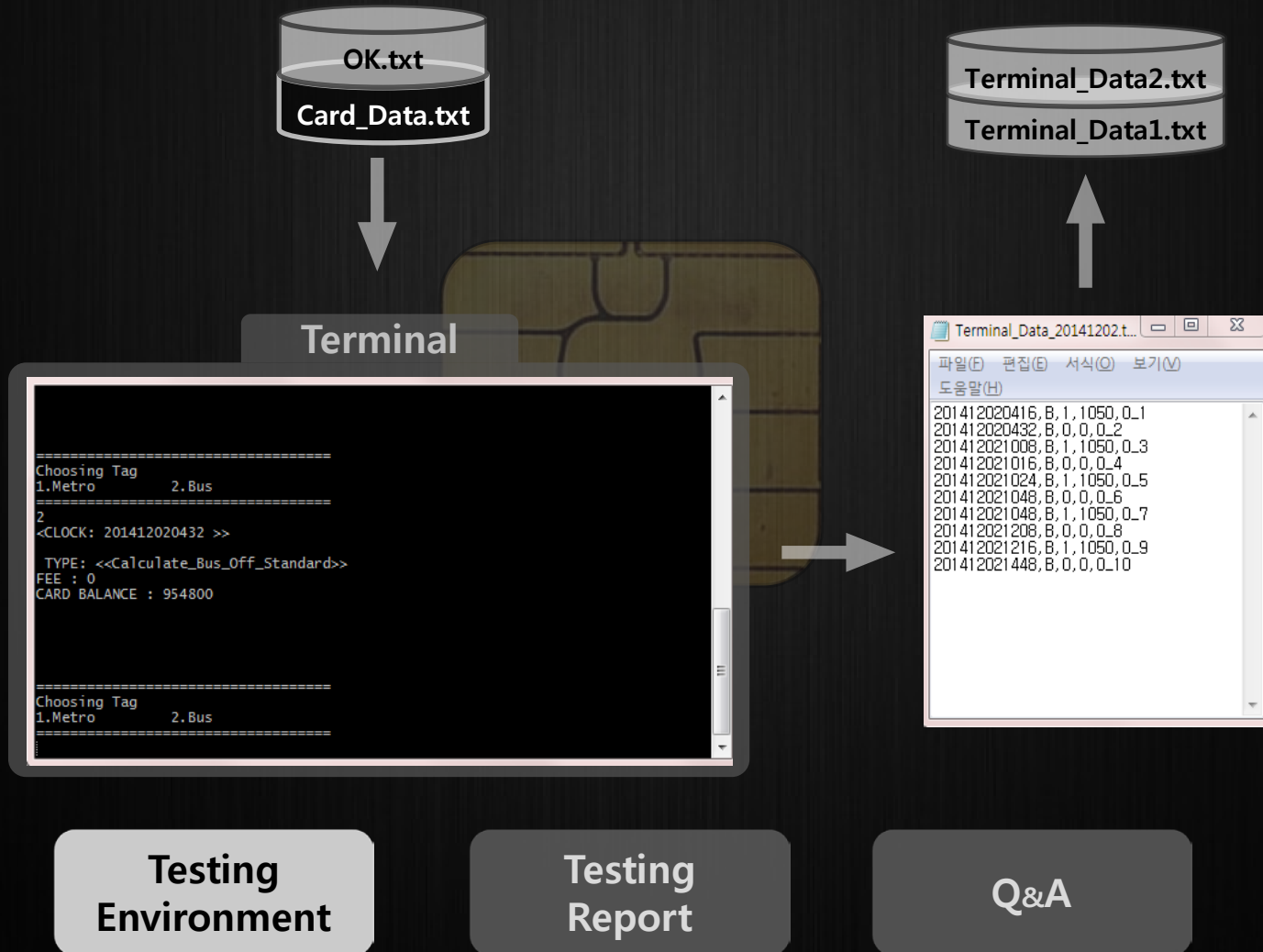
Testing Report

Q&A

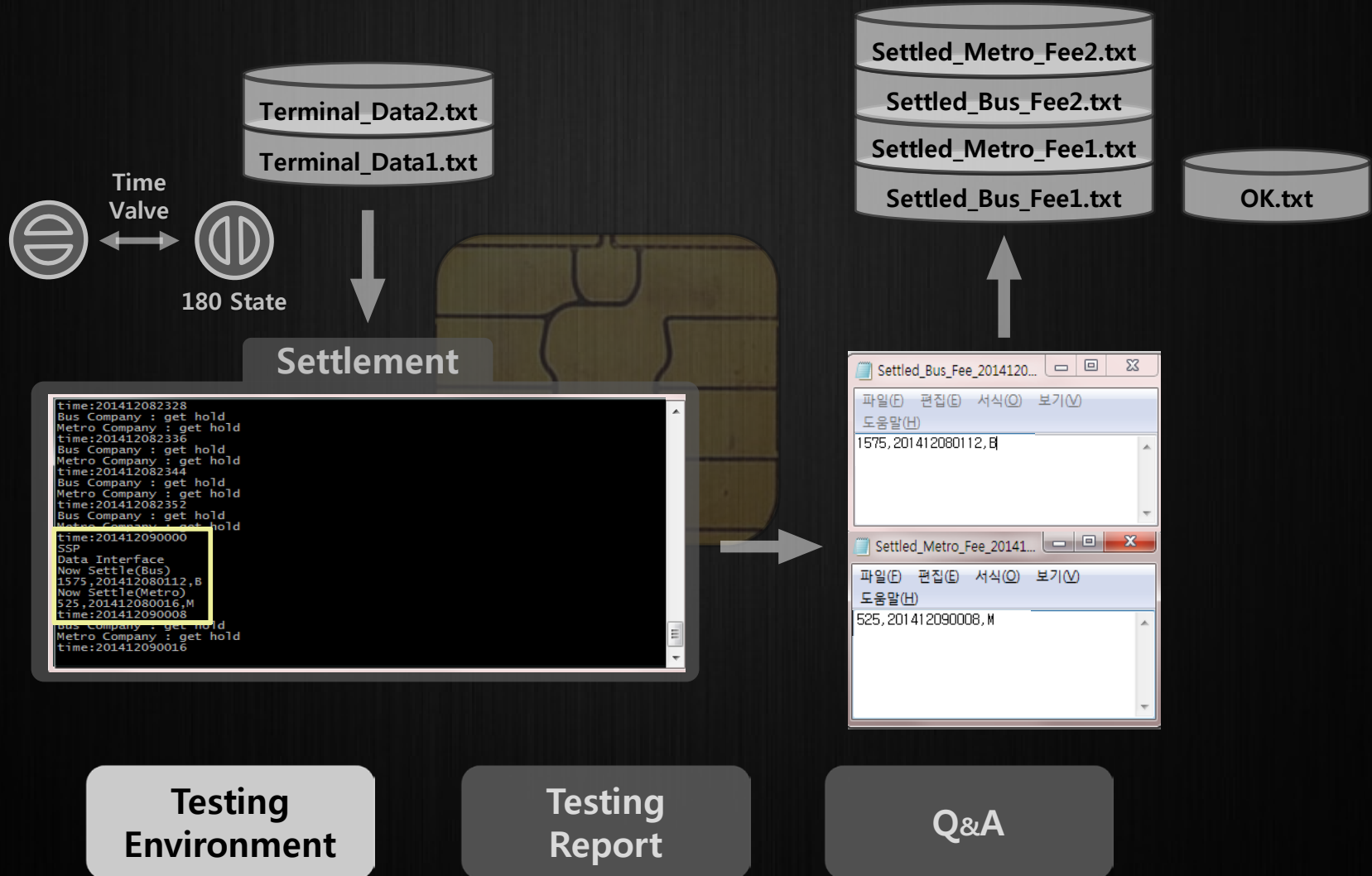
INITIAL STATE



OVER STATE



OVER STATE



Environment Needs : Cygwin

T7's report

Total Defect :18

Test case identifier	Description	Input specification	Output specification	Result
T7_STP_Ver1.0 PTS_STC_11	버스에서 하차 태그를 한 후 15초 이내에 지하철 승차 시 환승 처리가 되는가- 카드 잔액이 지하철 환승 최고 부과금액인 600원 미만인 경우	Card_Data.txt 내 잔액 < 1650인 상태, 2 -> 2 -> 1&1	NOT ENOUGH MONEY	Fixed
T7_STP_Ver1.0 PTS_STC_16	지하철에서 하차 태그를 한 후 15초 이내에 버스 승차 시 환승 처리가 되는가 카드 잔액이 버스 환승 최고 부과금액인 700원 미만인 경우	Card_Data.txt 내 잔액 < 1750인 상태, 1&1~5 중 하나 -> 1&1~5 중 하나 ->2	NOT ENOUGH MONEY	Fixed

Fix 2 Case

[2014_SE_B][T1]T1_STP_Ver2.0

T7_STP_Ver1.0

[2014_SE_B][T1]T1_STR_Ver2.0

T7_STR_Ver1.0

Testing
Environment

Testing
Report

Q&A

Environment Needs : Cygwin

T7's report

Total Defect :18

T7_STP_Ver1.0 PTS_STC_29	계산된 부과 요금과 기타 정보가 단말기 누적 결제 기록에 기록되는가	1&1~5 중 하나 or 2 (카드 태그 시)	태그시간,교통수단(B/M),승하차 (1/0),잔액,단말기ID_count	fixed
T7_STP_Ver1.0 PTS_STC_30	환승이 아닐 경우 해당 탑승 단말기 정보가 카드와 단말기 누적 결제 기록에 기록되는가	1&1~5 중 하나 or 2 (환승인 경우를 제외한 카드 태그 시)	태그시간,교통수단(B/M),승하차 (1/0),잔액,해당 탑승 단말기 ID_count	fixed
T7_STP_Ver1.0 PTS_STC_31	환승일 경우 카드에 기록되어 있는 탑승 단말기 정보가 카드와 단말기 누적 결제 기록에 기록되는가	2 -> 2 -> 1&1~5 중 하나 or 1&1~5 중 하나 -> 1&1~5 중 하나 -> 2	태그시간,교통수단(B/M),승하차 (1/0),잔액,카드에 기록되어 있는 탑승 단말기ID_count	fixed

Fix 3 Case

[2014_SE_B][T1]T1_STP_Ver2.0

T7_STP_Ver1.0

[2014_SE_B][T1]T1_STR_Ver2.0

T7_STR_Ver1.0

Testing
Environment

Testing
Report

Q&A

Environment Needs : Cygwin

T7's report

Total Defect :18

T7_STP_Ver1.0 PTS_STC_20	버스에서 지하철 환승 후 하차 태그를 안 하고... 지하철 승차 시 미 정산 요금 600원이 부과되는가, 일반 승차 처리가 되는가	not-in-case
T7_STP_Ver1.0 PTS_STC_21	지하철에서 버스 환승 후 하차 태그를 안 하고... 버스 승차 시 미 정산 요금 700원이 부과되고 일반 승차 처리가 되는가	not-in-case
T7_STP_Ver1.0 PTS_STC_24	지하철 승차 후 하차 태그를 안 하고... 지하철 승차 시 미 정산 요금 200원이 부과되는가, 일반 승차 처리가 되는가	not-in-case

Not-in-case : T1이 제작한 PTS시스템은 승차,하차 태그를 구분하지 않아 같은 교통수단에 대해 2번의 연속 승차가 불가능하다.

T7의 Test case 중 3가지는 이에 부합하므로 T1이 제작한 PTS에서는 존재할 수 없는 케이스이다. **Pass 3 Case**

[2014_SE_B][T1]T1_STP_Ver2.0

T7_STP_Ver1.0

[2014_SE_B][T1]T1_STR_Ver2.0

T7_STR_Ver1.0

Testing
Environment

Testing
Report

Q&A

Environment Needs : Cygwin

T7's report

Total Defect :18

T7_STP_Ver1.0 PTS_STC_37	환승의 경우를 제외한 버스/지하철에 대한 정산이 올바르게 이루어지는가- 누적 결제 기록에 bus와, 지하철 이용 기록이 같이 있는 경우	확인 할 수 없음		fixed
T7_STP_Ver1.0 PTS_STC_38	지하철->버스->지하철->버스... 순의 환승에 대한 정산이 올바르게 이루어 지는가- 누적 결제 기록에 지하철->버스->지하철->버스...	확인 할 수 없음		fixed

⋮

8 more

Fix 10 Case

[2014_SE_B][T1]T1_STP_Ver2.0

T7_STP_Ver1.0

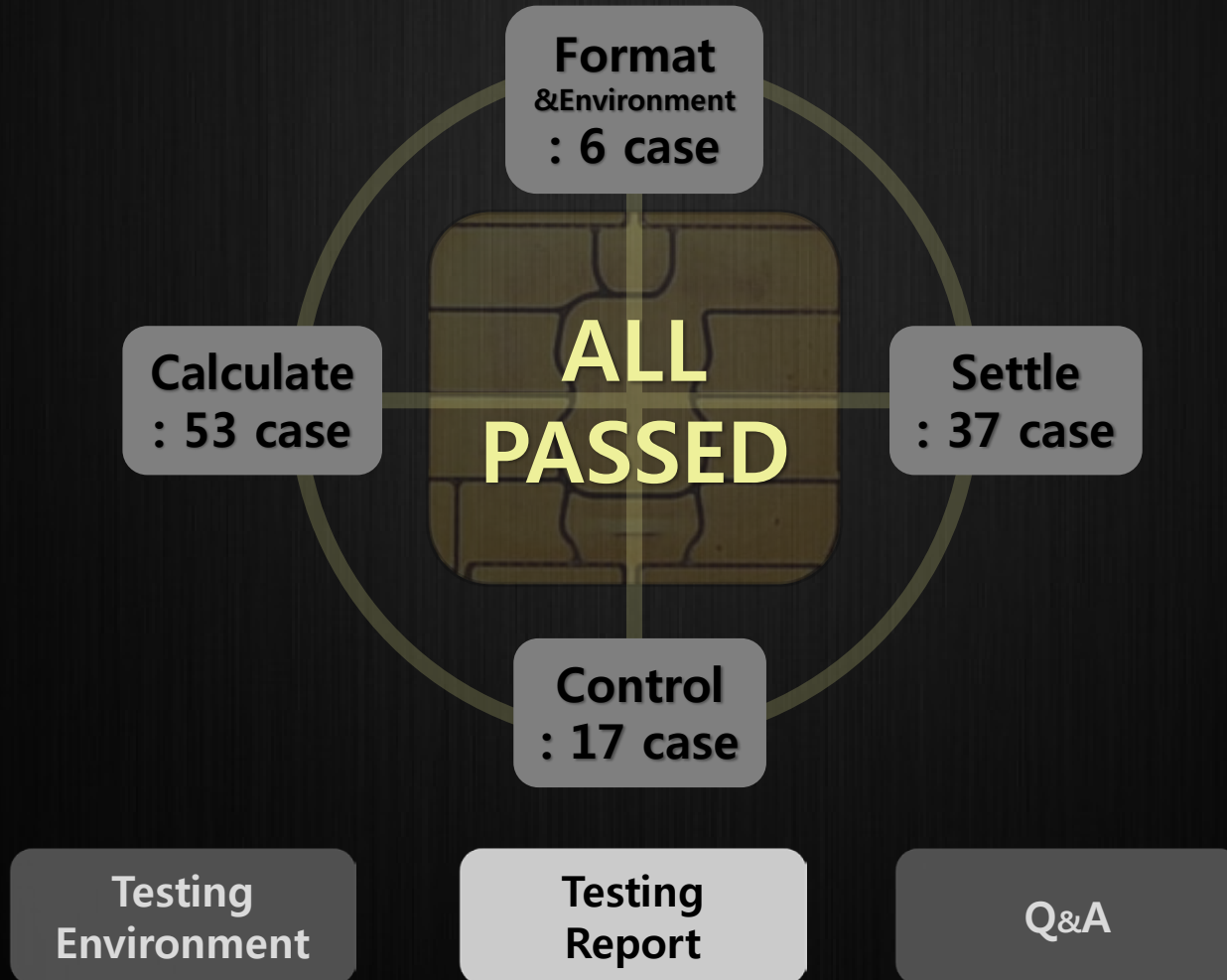
[2014_SE_B][T1]T1_STR_Ver2.0

T7_STR_Ver1.0

Testing
Environment

Testing
Report

Q&A





**Testing
Environment**

**Testing
Report**

Q&A