

# Structure & Protocol

작성자	윤희동
부서	EnjoyWorks - Planning

[illegible]

Common

테마 정보(Common\_RM\_Theme)

상품 정보(Common\_Goods\_info)

상품 주키	int
상품 명	Text
상품 썸네일	Text
상품 이미지	Image
상품 URL	Text
상품 유효 시작일	Date
상품 유효 종료일	Date

게시판 정보 (Common\_NoticeBoard\_info)

게시판 주키	int
게시 제목	Text
게시 내용	Text
게시 이미지	Text

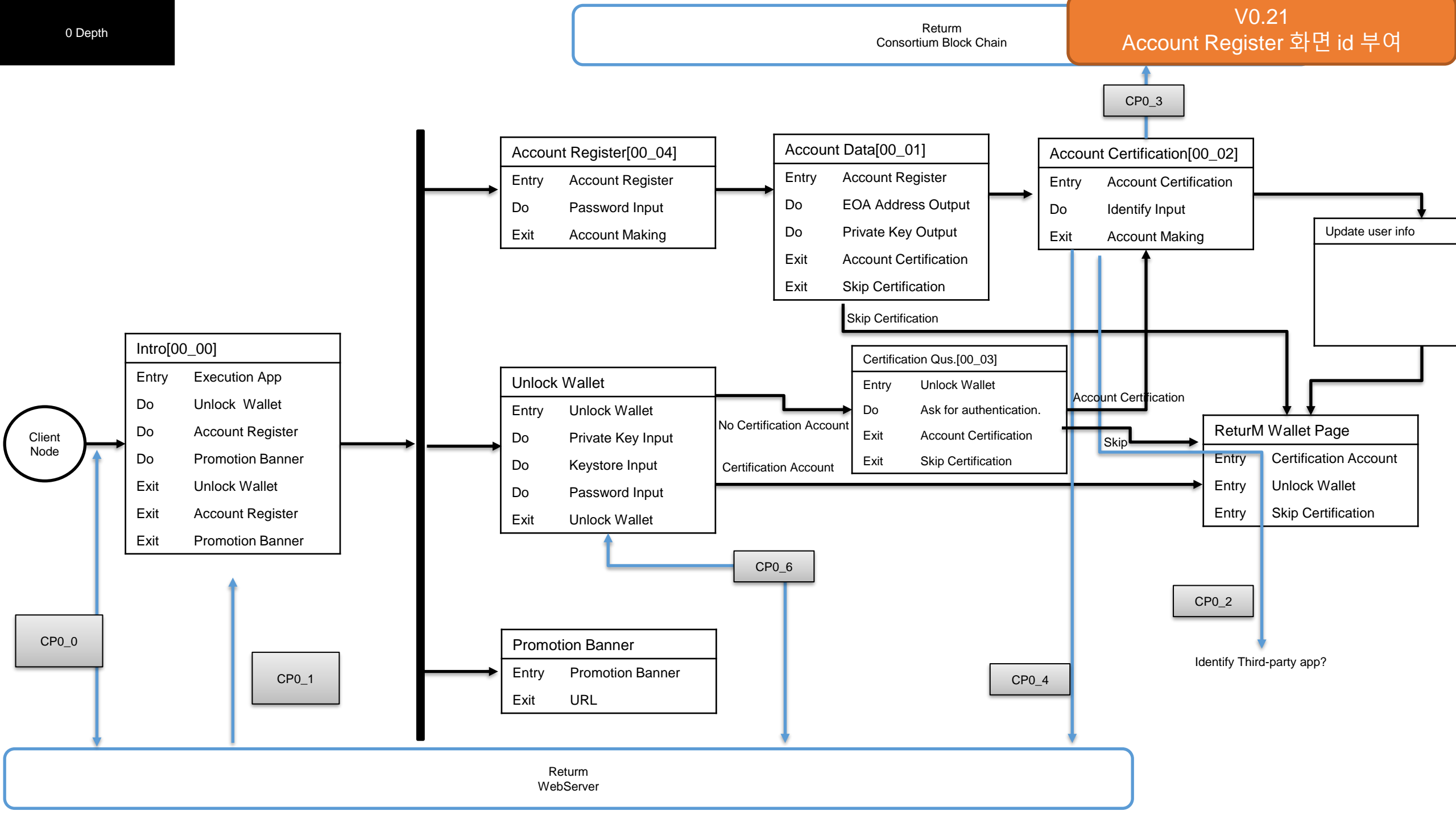
미션 정보(Common\_Mission\_info)

미션 주키	int
미션 명	Text
미션 종류	Text
미션 내용	Text
미션 이미지	Image
미션 URL	Text
미션 보상	Double
미션 최대 보상	double
미션 유효 시작일	Date
미션 유효 종료일	Date

```
'Common_RM_Theme' : [
  "theme_msg_code" : "success", //테마 설정 안 할 시 fail,설정 시 "success"
  "Common_Goods_info":[
    {Goods_no : "5", // 상품 주키
      Goods_URL : "" // 상품 정보
    },{
      ...
    }
  ],
  "Common_Mission_info":[
    {Mission_no : "5", // 상품 주키
      Mission_URL : "" // 상품 정보
    },{
      ...
    }
  ],
  "Common_NoticeBoard_info":[
    {NoticeBoard_info : "5", // 상품 주키
      NoticeBoard_info : "" // 상품 정보
    },{
      ...
    }
  ]
]
```

# Structure

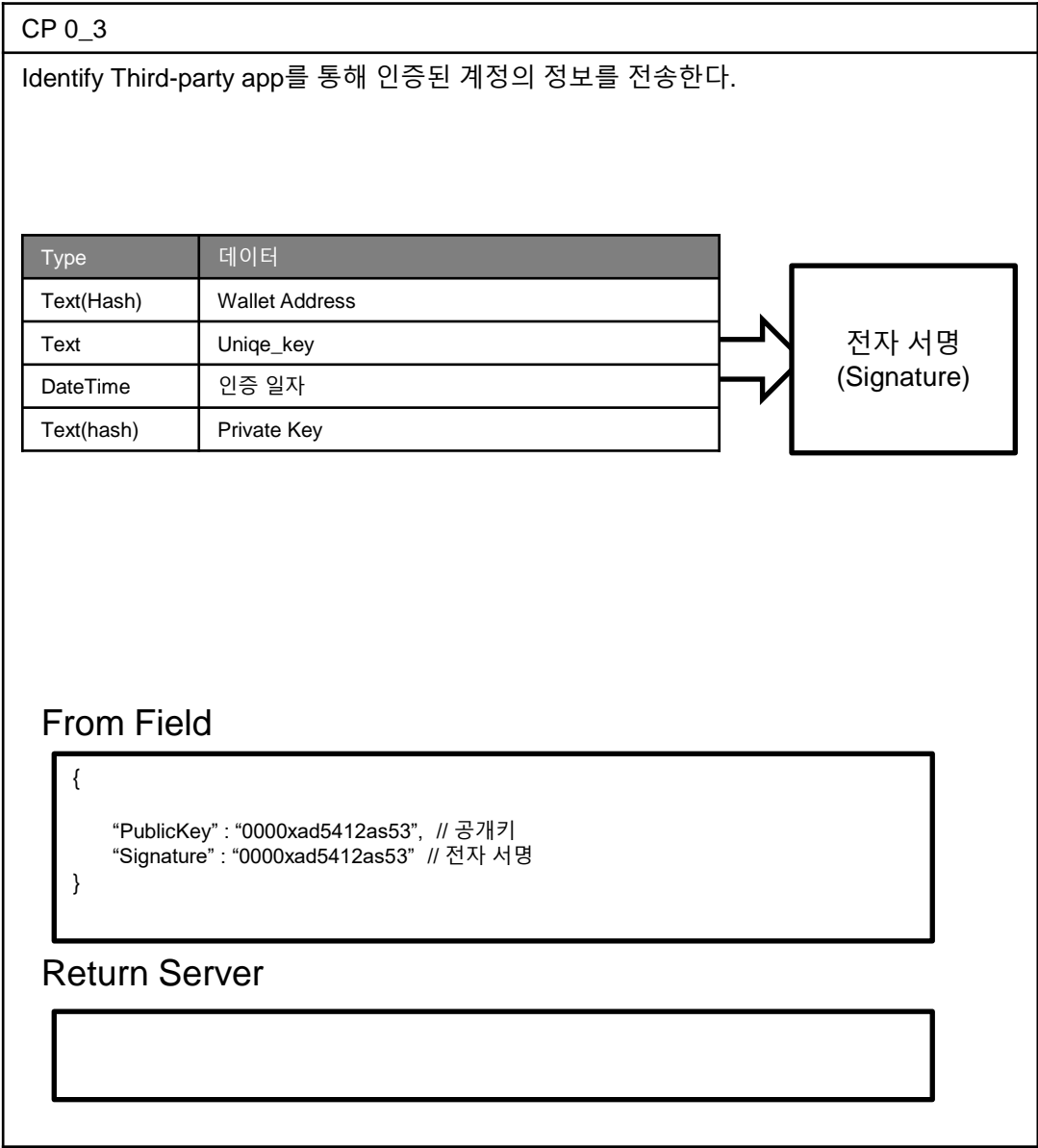
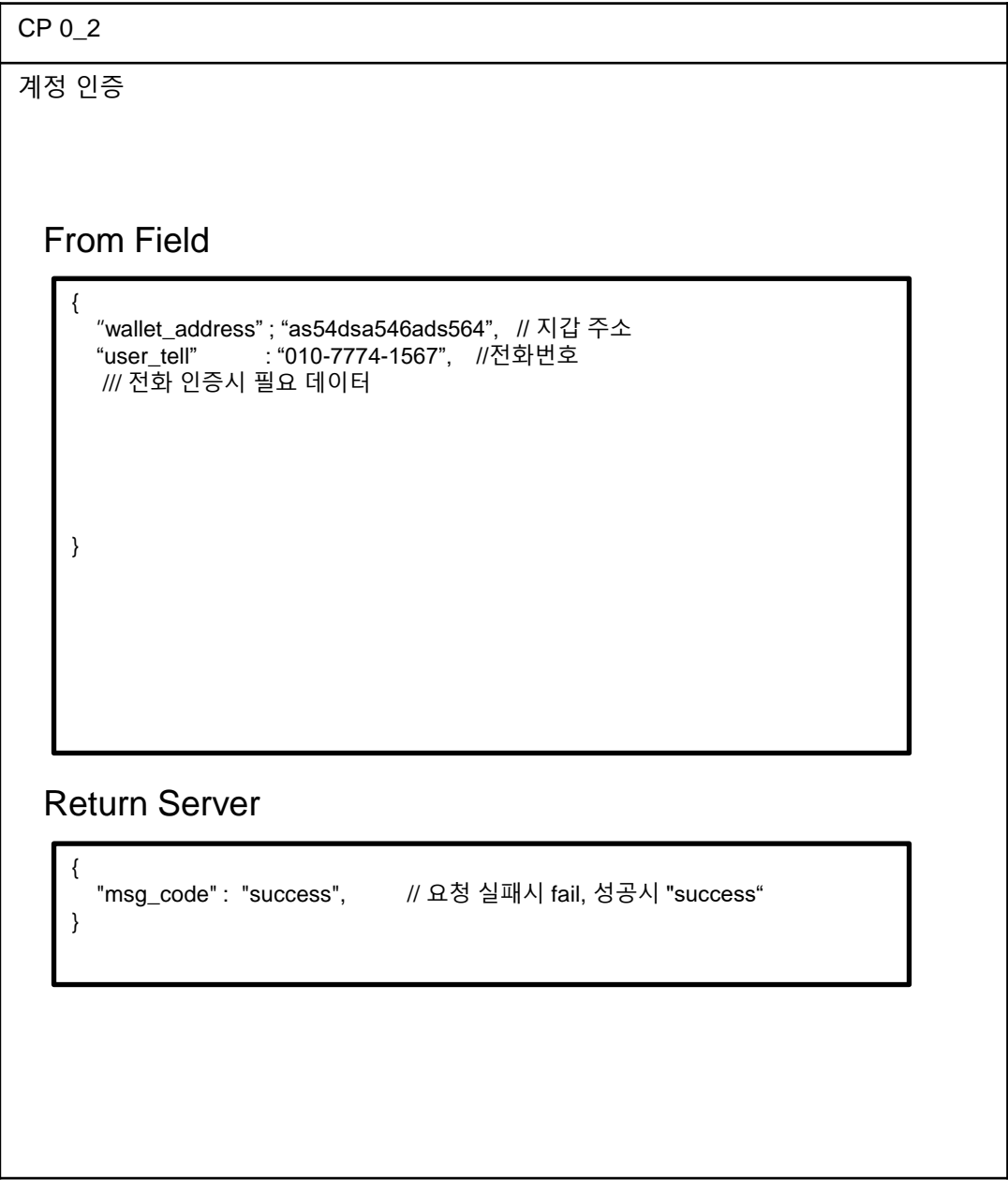
Depth 0



CP0_0
<p>App의 데이터 정보를 확인한다. Client는 Sever에 최신 정보를 Sever에 전송하면, Sever는 업데이트 정보를 확인한 후 msg_code를 전송한다. 이때 업데이트가 필요한 경우 msg_code와 파일의 FTP주소를 함께 전송하고, 업데이트가 필요 없으면 msg_code만을 전송한다.</p>
<div>From Field</div> <div><pre>{   "Update_info": "app 버전 정보"    //app의 버전 }</pre></div>
<div>Return Server</div> <div><div>&lt;업데이트 파일이 존재하지 않은 경우&gt; {   "msg_code": "success",    // 요청 실패시 fail, 성공시 "success" }</div><div>&lt;업데이트 파일이 존재하는 경우&gt; {   "msg_code": "success",    // 요청 실패시 fail, 성공시 "success"   "ftp_url": "업데이트 파일 URL" }</div></div>

CP 0_1
<p>프로 모션 정보를 확인한다. Client는 intro 페이지에 진입하여, Server부터 저장된 리턴 프로모션 정보를 요구한다. Server는 Client에서 전송 받은 date 파일을 기준으로 출력 정보를 선별하여 Client에 전송한다.</p>
<div>From Field</div> <div><pre>{   "Promotion_date": "2018-12-18"  //현재 일자 }</pre></div>
<div>Return Server</div> <div><pre>{   "msg_code": "success",    // 요청 실패시 fail, 성공시 "success"   "Promotion_list": [     {       "pl_kind": "일반 홍보",    // 리스트 종류       "pl_title": "리턴을 구매하세요", //프로모션 타이틀       "pl_image": "ftp/주소",    // 이미지 파일       'pl_detail' : "보여줄게 ~~~~~" // 상세 글     },     {       "pl_kind": "일반 홍보",    // 리스트 종류       "pl_title": "리턴을 구매하세요", //프로모션 타이틀       "pl_image": "ftp/주소",    // 이미지 파일       'pl_detail' : "보여줄게 ~~~~~" // 상세 글     }   ] }</pre></div>





CP 0\_4

계정 인증 정보를 저장한다. (전화 번호를 이용한 인증 생각중)  
Client는 유저의 Wallet Address와 유저의 계정 인증 정보를 Server에 전송하고, Server는 해당 정보를 저장한다.

From Field

```
{
  "wallet_address"; "as54dsa546ads564", // 지갑 주소
  "user_tell"      : "010-7774-1567", //전화번호
  "Certification_list" :{           // 계정 인증 API에 필요한 정보
    "Certification_0" : "",
    "Certification_1" : "",

    "Certification_n" : ""
  },
  "user_Certification_list" :{           // 계정 인증시 유저에게 받을 정보
    "세대" : "30대", // 연령대 정보
    "SEX" : "남",    // 성별
    "목적" : "투자", // 리템 사용 목적
    "가입 경로" : "친구 추천" //가입 경로, 마케팅 데이터
  }
}
```

Return Server

```
{
  "msg_code" : "success", // 요청 실패시 fail, 성공시 "success"
}
```

CP 0\_6

Wallet address의 계정 인증 여부를 확인한다. 유효성이 검증된 Wallet Address의 계정 정보를 확인 한다.  
이때 Client는 계정 인증이 되지 않은 Wallet Address에 대해 계정 인증 여부를 묻는다. 이때 Client는 계정 인증이 되어 있는 유저의 계정 주키를 보관한다.

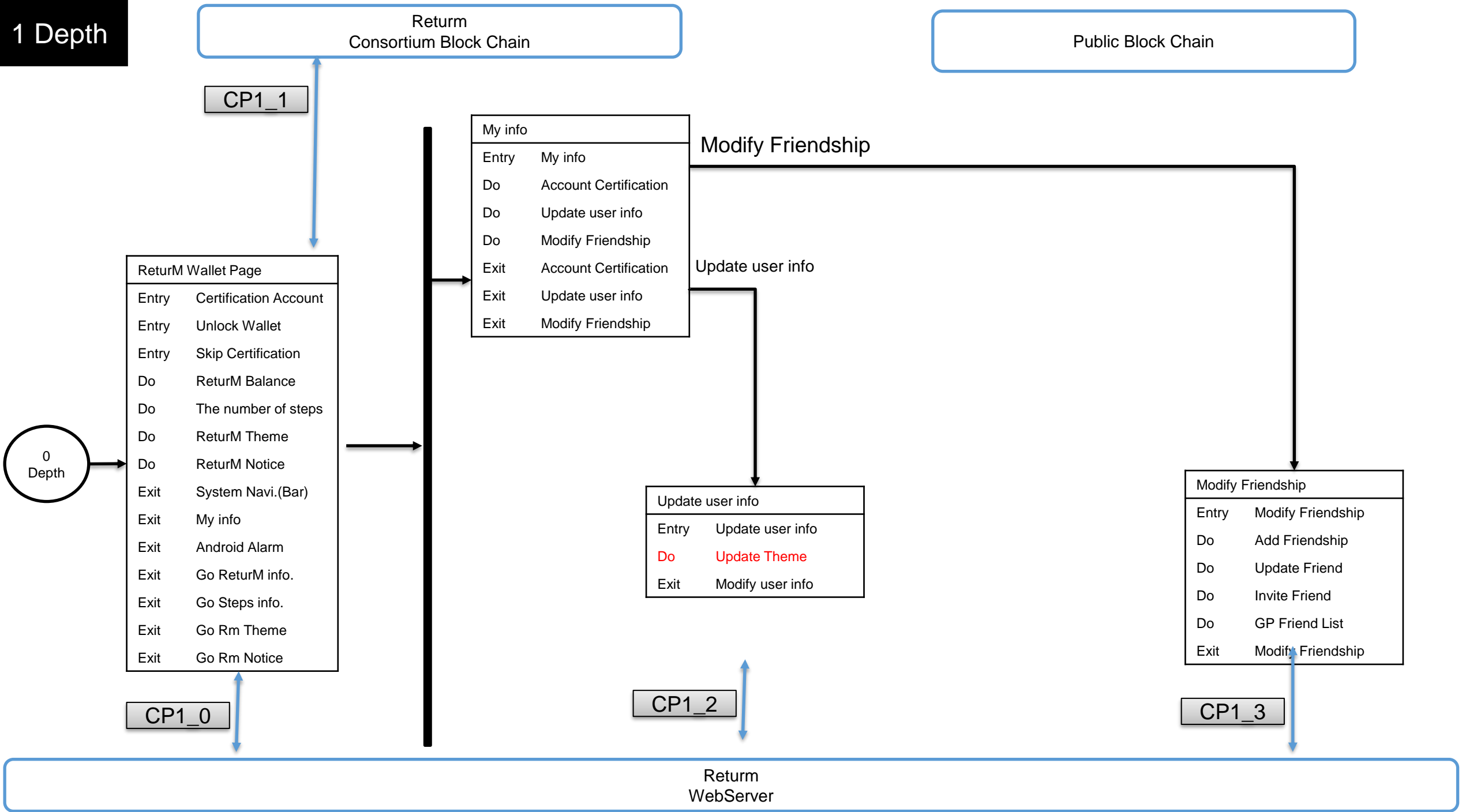
From Field

```
{
  "Wallet_Address" : "asdda456fa54" //지갑 키
}
```

Return Server

```
{
  "msg_code" : "success", // 요청 실패 시 fail, 성공 시 "success"
  "계정 주키" : "222"     //계정 주키
}
```

Depth 1



# 1 DEPTH - Communication protocol

CP 1\_0

메인 화면에 출력한 정보를 Server가 clien에게 반환한다. Server는 Client가 보낸 WalletAddress를 통해 계정 인증 여부를 확인하고, 이에 해당하는 Theme 정보를 반환한다. 이때 계정 인증을 하지 않은 계정에 대해서는 관리자 설정한 테마를 반환한다. 이때, client는 계정 인증 정보를 보관한다.

From Field

```
{
  "wallet_address"; "as54dsa546ads564", // 지갑 주소
}
```

Return Server

```
{
  "msg_code": "success",      // 요청 실패시 fail, 성공시 "success"
  "계정 주키" : "222",        //계정 주키. 미 인증시 Null 값
  "걸음 수"   : "7777",        //계정 주키
  'RM_Theme': [
    "theme_msg_code": "success", //테마 설정 안할시 fail,설정시 "success"
    "Goods_info":[
      {Goods_no: "5", // 상품 주키
        Goods_URL: "" // 상품 정보
      },...,{
      ...}
    ]
    "Mission_info":[
      {Mission_no: "5", // 상품 주키
        Mission_URL: "" // 상품 정보
      },...,{
      ...}
    ]
  ],
  RM_Notice :[
    {Notice_info: "노티스 정보"
    },...
  ]
}
```

CP 1\_1

유저의 Wallet 잔고를 확인한다.

From Field

```
{
  "wallet_address"; "as54dsa546ads564", // 지갑 주소
}
```

Return Consortium Block Chain

```
{
  "Wallet_balance"; "50000.0000", // RM 잔고
}
```

# 1 DEPTH - Communication protocol

## CP 1\_2

유저의 정보를 업데이트 한다. 정보 수정은 계정 인정이 끝난 유저만 수행 할 수 있다.

### From Field

```
{
  "wallet_address"; "as54dsa546ads564", // 지갑 주소
  "계정 주키" : "7777", // 계정 주키
  "user_Certification_list" : { // 계정 인증시 유저에게 받을 정보
    "세대" : "30대", // 연령대 정보
    "SEX" : "남", // 성별
    "목적" : "투자", // 리턴 사용 목적
    "가입 경로" : "친구 추천" // 가입 경로, 마케팅 데이터
  }
}
```

### Return Server

```
{
  "msg_code" : "success", // 요청 실패시 fail, 성공시 "success"
}
```

## CP 1\_3

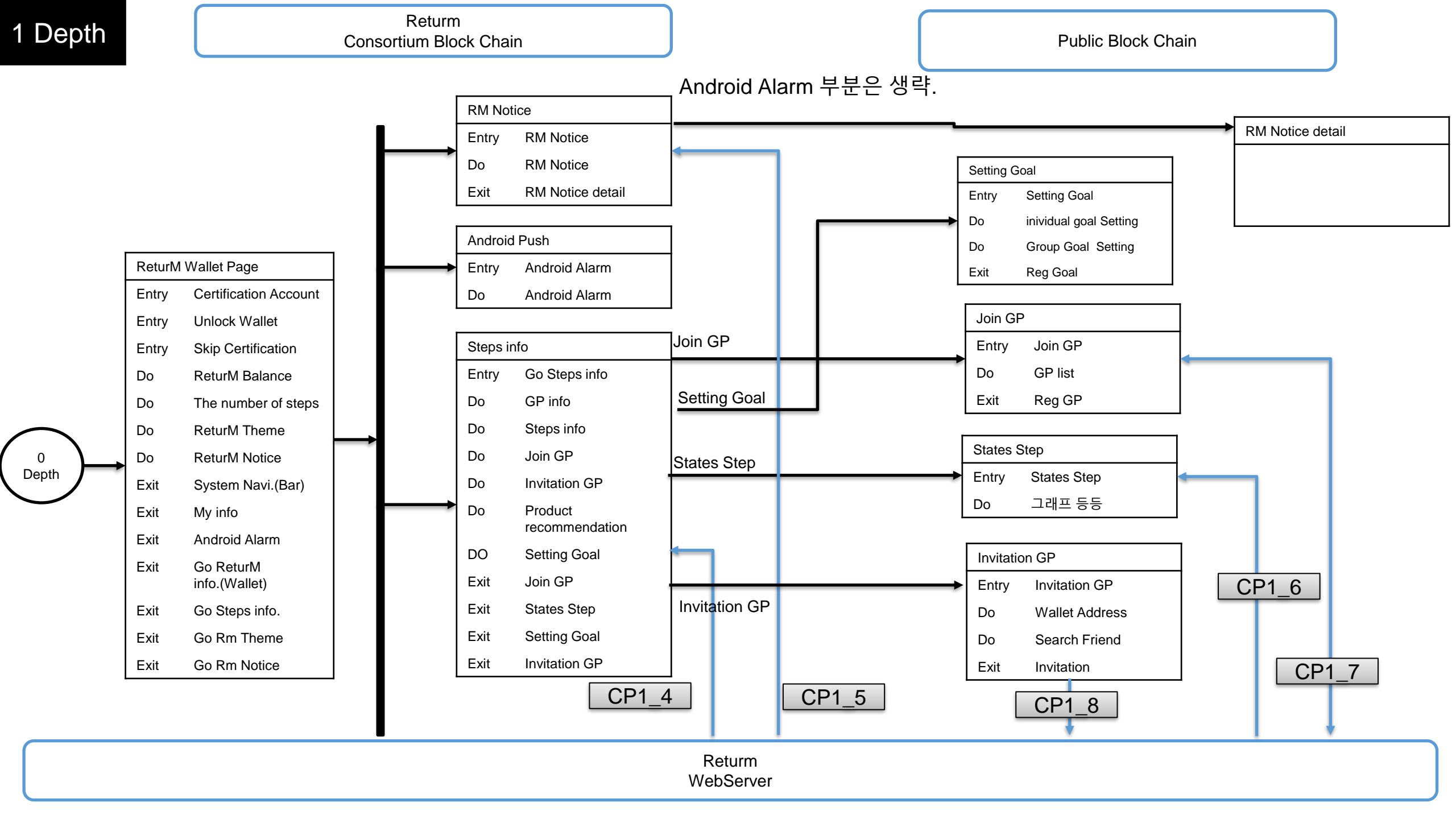
유저의 친구 목록을 저장한다.

### From Field

```
{
  "wallet_address"; "as54dsa546ads564", // 지갑 주소
  "계정 주키" : "7777", // 계정 주키
  "friend_list" : [
    { "WalletAddress" : "X0B045564a", // 친구 지갑 주소
      "NickName" : "가우스", // 별명
    },
    { "WalletAddress" : "X0B045564a", // 친구 지갑 주소
      "NickName" : "가우스", // 별명
    },
    { "WalletAddress" : "X0B045564a", // 친구 지갑 주소
      "NickName" : "가우스", // 별명
    },
    ....
  ]
}
```

### Return Server

```
{
  "msg_code" : "success", // 요청 실패시 fail, 성공시 "success"
}
```



# 1 DEPTH - Communication protocol

## CP 1\_4

만보기 정보를 출력한다. Client가 유저의 주키와 지갑 주소를 전송하면, Server는 유저의 걸을 수와 유저가 속한 그룹 모든 인원의 걸음 수를 전송한다. 이때 Server는 유저가 현재 받을 수 있는 RM 정보와 이로 구매 할 수 있는 상품 정보를 반환한다.

### From Field

```
{
  "wallet_address": "as54dsa546ads564", // 지갑 주소
  "계정 주키" : "7777",                // 계정 주키, 계정 인증 X → null
}
```

### Return Server

```
{
  "msg_code": "success",    // 요청 실패시 fail, 성공시 "success"
  "유저 걸음": "30",
  GP_list : [
    {
      "walletAddress": "54646asd56464", // 지갑 주소(?) 주키 ?
      "Nick": "가우스",                // 별명
      "걸음수": "50"                   // 걸음수
    }, {
      ....
    }
  ],
  "Common_Goods_info": [ ]
}
```

## CP 1\_5

리템 공지 사항 정보를 반환한다.

### From Field

```
{
}
```

### Return Server

```
{
  "msg_code": "success",    // 요청 실패시 fail, 성공시 "success"
  /// 공지 사항 정보
}
```



# 1 DEPTH - Communication protocol

## CP 1\_6

유저의 걸음을 통계 낸다. Client가 유저의 주키를 전송하며, Server는 유저의 걸음 정보를 반환한다. 통계는 역대 통계를 기본으로 하고,  
????????????????(통계 기간)

### From Field

```
{
  "wallet_address": "as54dsa546ads564", // 지갑 주소
  "계정 주키" : "7777", // 계정 주키, 계정 인증 X → null
  "오늘 일자" : "2018-12-18" // 오늘 일자 ← 기간 검색 있을 경우 필요
}
```

### Return Server

```
{
  "msg_code": "success", // 요청 실패시 fail, 성공시 "success"
  "유저 걸음" : "30",
  GP_list : [
    {
      "일자": "2018-11-11", // 걸음 걸이 정보
      "걸음수": "50" // 걸음수
    },
    {
      "일자": "2018-11-12", // 걸음 걸이 정보
      "걸음수": "55" // 걸음수
    },
    ....
  ]
}
```

## CP 1\_7

리턴 그룹 정보를 반환한다. Client는 유저가 그룹에 참여 하지 않은 경우 공개에 의한 그룹 참여 기능이 발동시킨다. 유저가 참여 가능한 그룹의 정보를 Server가 전송한다. 이때 그룹 공개 여부가 비밀이 아닌 그룹의 정보를 불러 온다.

### Return Server

```
{
  "msg_code": "success", // 요청 실패시 fail, 성공시 "success"
  "GP_List" : [
    {
      "GP_name": '트와이스', // 그룹명
      "GP_member": '9', // 그룹 인원수
      "GP_reg_date": '2018-07-16', // 그룹 생성일
      "GP_detaile": '트와이스는 힘들면 보기를 줄께', //그룹 설명
      "GP_Image": 'URL/.../1.png', // 그룹 상징이미지
      "GP_leader": '가우스', // 그룹장 별명
    },
    {
      "GP_name": '트와이스', // 그룹명
      "GP_member": '9', // 그룹 인원수
      "GP_reg_date": '2018-07-16', // 그룹 생성일
      "GP_detaile": '트와이스는 힘들면 보기를 줄께', //그룹 설명
      "GP_Image": 'URL/.../1.png', // 그룹 상징이미지
      "GP_leader": '가우스', // 그룹장 별명
    },
    ....
  ]
}
```

# 1 DEPTH - Communication protocol

CP 1\_8

유저가 친구를 그룹에 초대한다. 이때, Servedr는 해당 유저가 그룹에 속해 있는지 확인하고, 소속된 그룹이 존재하면, 초대 불가 메시지를 전송한다. (※ 1인 1 그룹이 원칙)

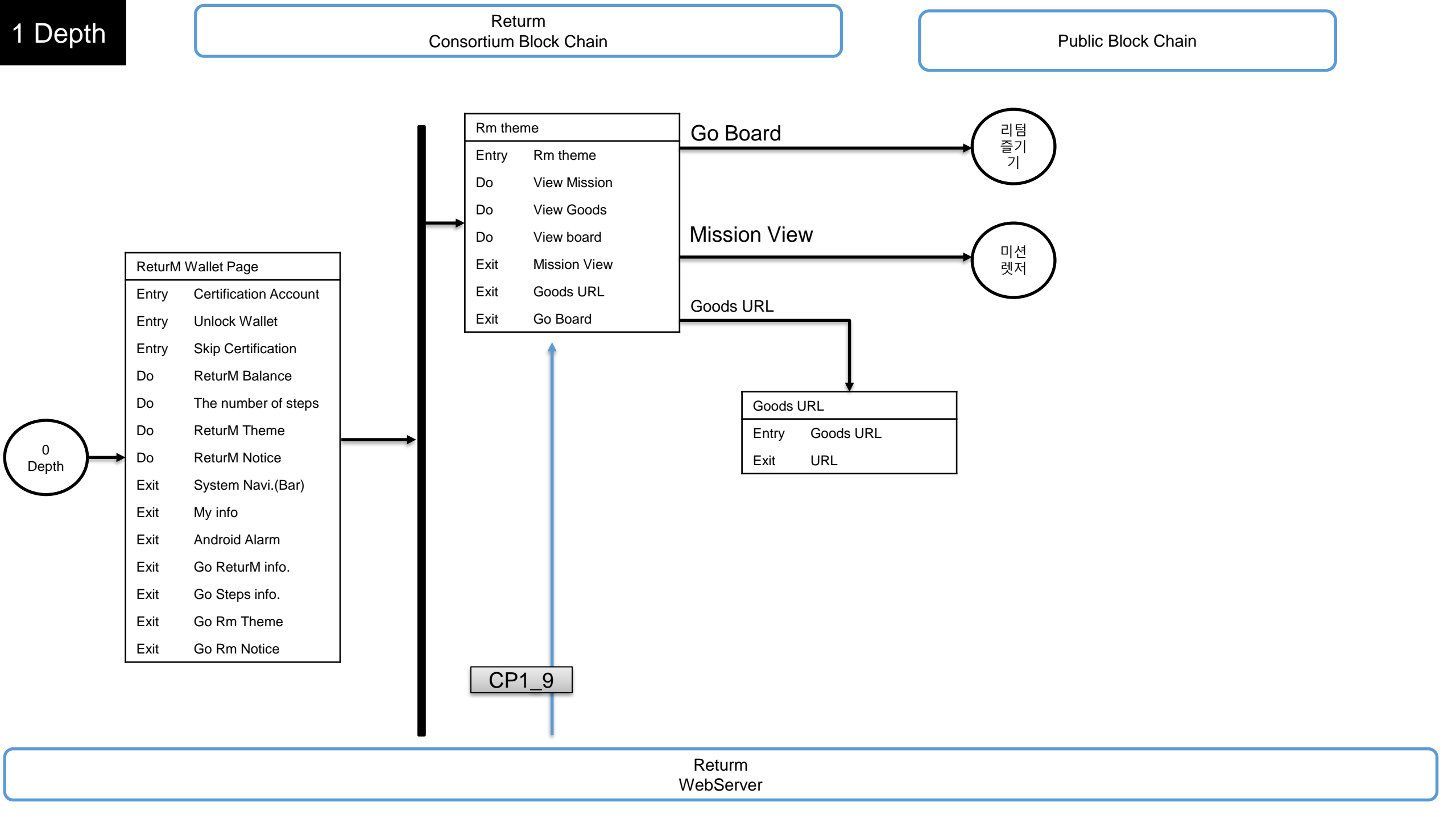
## From Field

```
{
  "wallet_address": "as54dsa546ads564", // 지갑 주소
  "계정 주키" : "7777", // 계정 주키
  "그룹 정보" : "45678" // 그룹 주키
  "오늘 일자" : "2018-12-18" // 오늘 일자
  "초대 리스트" : [
    {
      "walletAddress": "aas5654asd546" // 지갑 주소
    }, {
      "walletAddress": "aas5654asd546" // 지갑 주소
    }, {
      ....
    }
  ]
}
```

## Return Server

```
// 그룹 초대 신청이 완료 될 경우
{
  "msg_code": "success", // 요청 실패시 fail, 성공시 "success"
}

// 신청한 인원이 이미 그룹에 소속되어 있을 경우
{
  "msg_code": "success",
  "wallet_list": {"aas5654asd546", "aas5654asd546", ... "aas5654asd546"}
} // 초대가 불가했던 인원의 지갑 주소를 반환한다.
```



# 1 DEPTH - Communication protocol

CP 1\_9

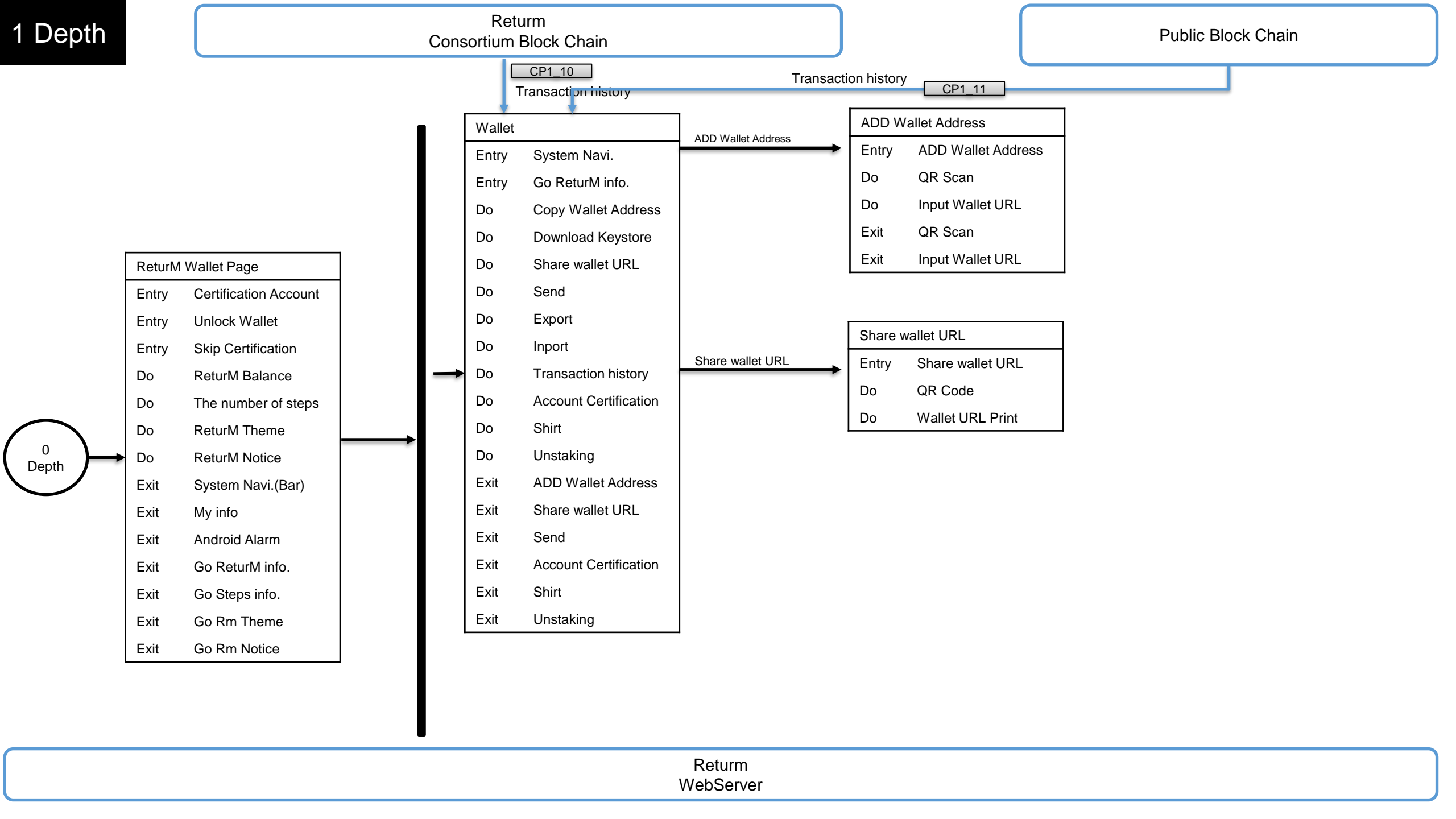
유저가 설정한 테마에 부합하는 미션 / 상품 정보를 반환한다. Client는 유저의 계정 정보를 전송하면, Server는 계정 인증 정보를 확인하고 인증된 계정 정보에 부합하는 테마를 찾아 유저에게 상품 및 미션을 추천한다. 유저가 인증을 하지 않은 경우 계정 주 키는 Null 값으로 전송되며, 출력 정보는 관리자가 선택한 정보를 출력한다.

## From Field

```
{  
  "계정 주키" : "21054" // 계정 주키  
}
```

## Return Server

```
{  
  "msg_code" : "success",      // 요청 실패시 fail, 성공시 "success"  
  "계정 주키" : "222",         // 계정 주키. 미 인증시 Null 값  
  "걸음 수"   : "7777",        // 계정 주키  
  'Common_RM_Theme': []  
}
```



CP 1\_10

private 지갑의 거래 내역을 확인 한다.

From Field

```
{  
  "wallet_address"; "as54dsa546ads564", // 지갑 주소  
}
```

Return Consortium Block Chain

```
{  
  "Wallet_balance"; "50000.0000", // RM 잔고  
}
```

CP 1\_11

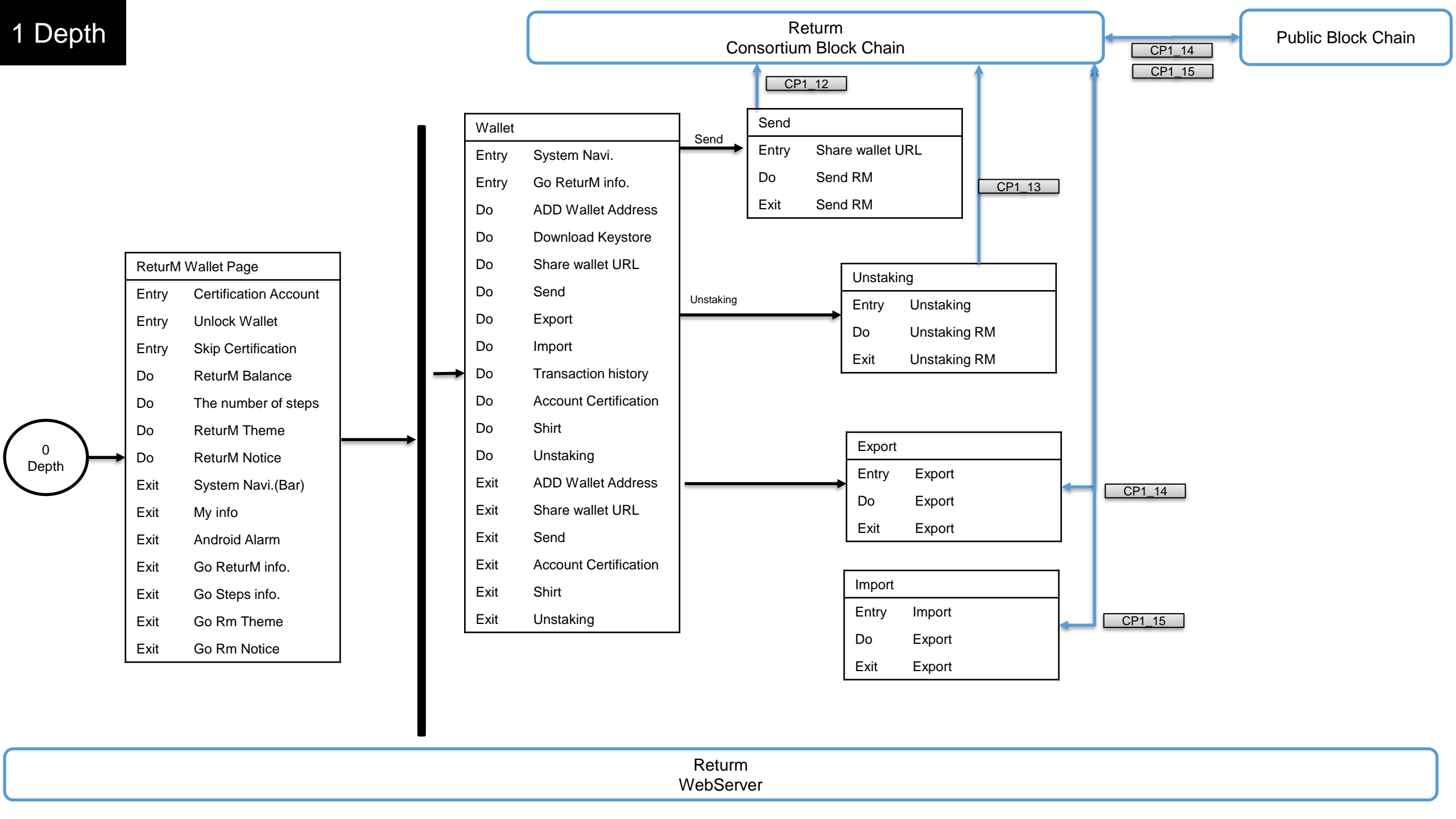
Public 지갑의 거래 내역을 확인 한다.

From Field

```
{  
  "wallet_address"; "as54dsa546ads564", // 지갑 주소  
}
```

Return Consortium Block Chain

```
{  
  "Wallet_balance"; "50000.0000", // RM 잔고  
}
```



Return Consortium Block Chain

CP1\_12

CP1\_13

CP1\_14

CP1\_15

Public Block Chain

CP1\_14

CP1\_15

Return WebServer

# 1 DEPTH - Communication protocol

CP 1\_12

유저의 RM을 송금한다. Client는 유저의 거래 내역과 개인키를 암호화하여 전자 서명(Signature)을 만든다. Client는 Signature와 유저의 Public Key를 Consortium Block Chain에 전달한다.

Type	데이터
Text(Hash)	Wallet Address(보내는 사람)
Text(Hash)	Wallet Address(받는 사람)
Double	RM Amount
Double	Staking RM
Double	UnStaking RM
Text	Nick(지갑에 표시된 받는 사람 닉네임)
Text	Memo
Text(hash)	Private Key
DateTime	송금 일자

전자 서명  
(Signature)

From Field

{  
    "PublicKey" : "0000xad5412as53", // 공개키  
    "Signature" : "0000xad5412as53" // 전자 서명  
}

Return Server

CP 1\_13

유저의 Staking 상태의 RM을 Unstaking한다. . Client는 유저의 거래 내역과 개인키를 암호화하여 전자 서명(Signature)을 만든다. Client는 Signature와 유저의 Public Key를 Consortium Block Chain에 전달한다.

Type	데이터
Text(Hash)	Wallet Address
Text(hash)	Private Key
Double	RM Amount

전자 서명  
(Signature)

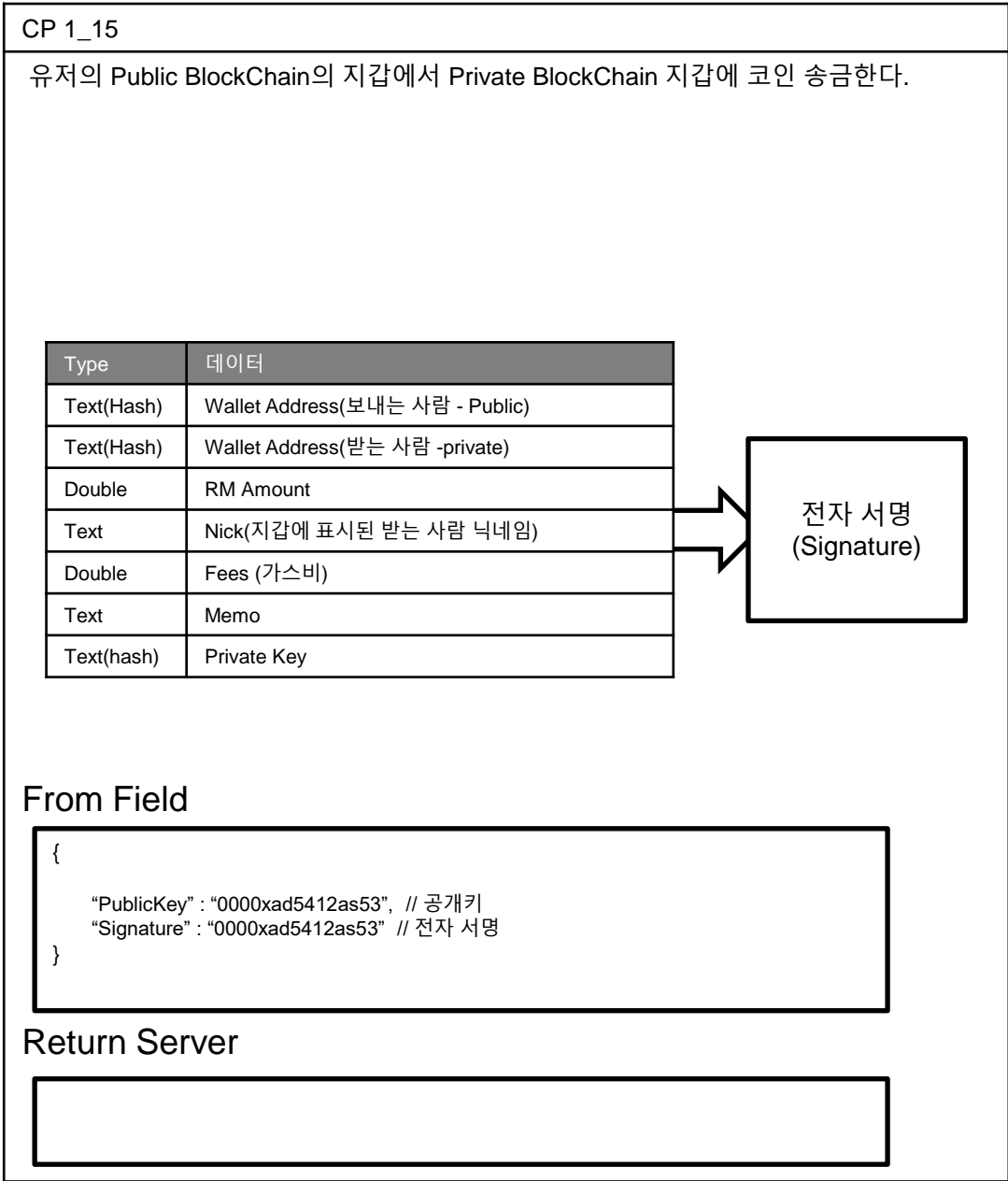
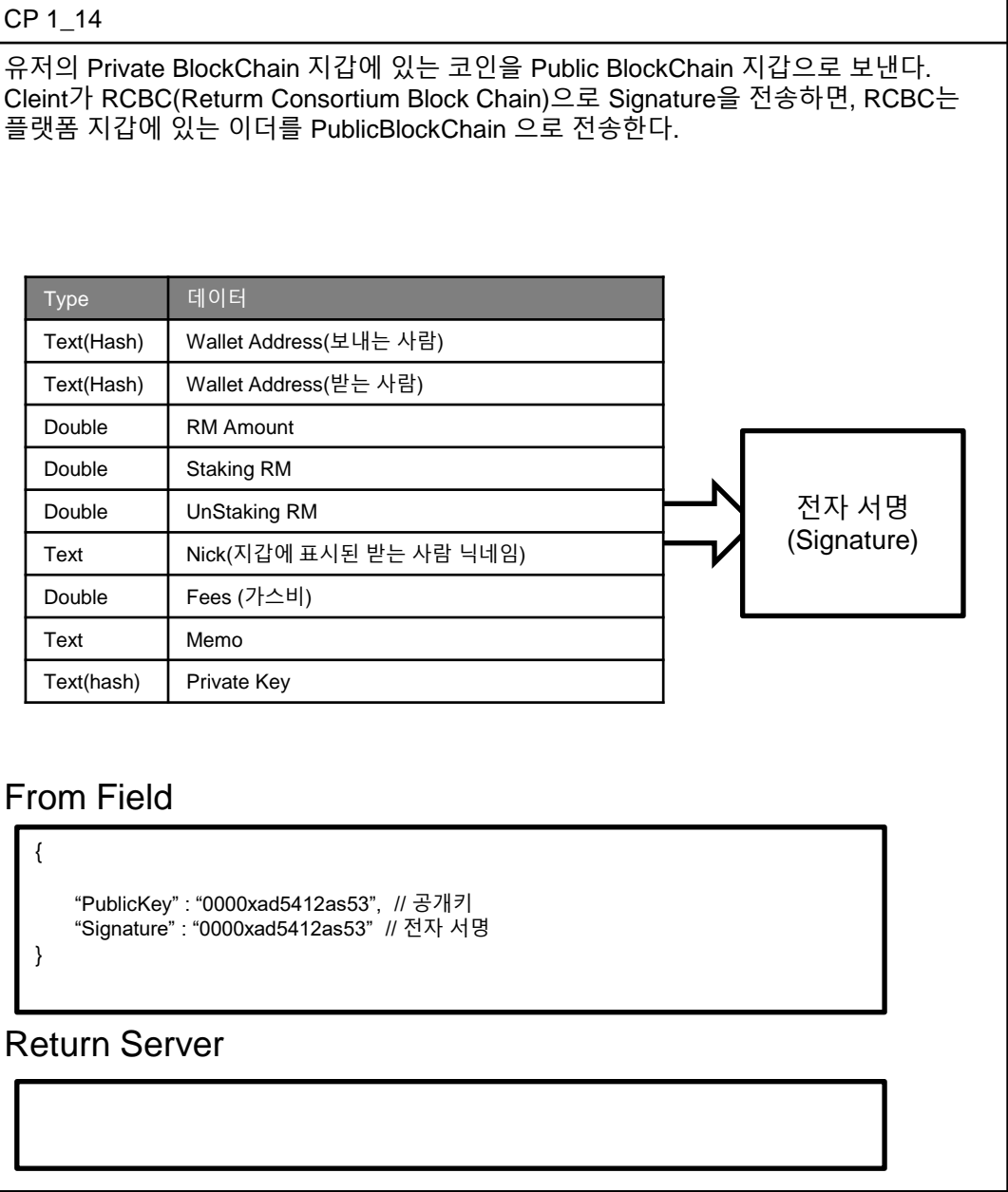
From Field

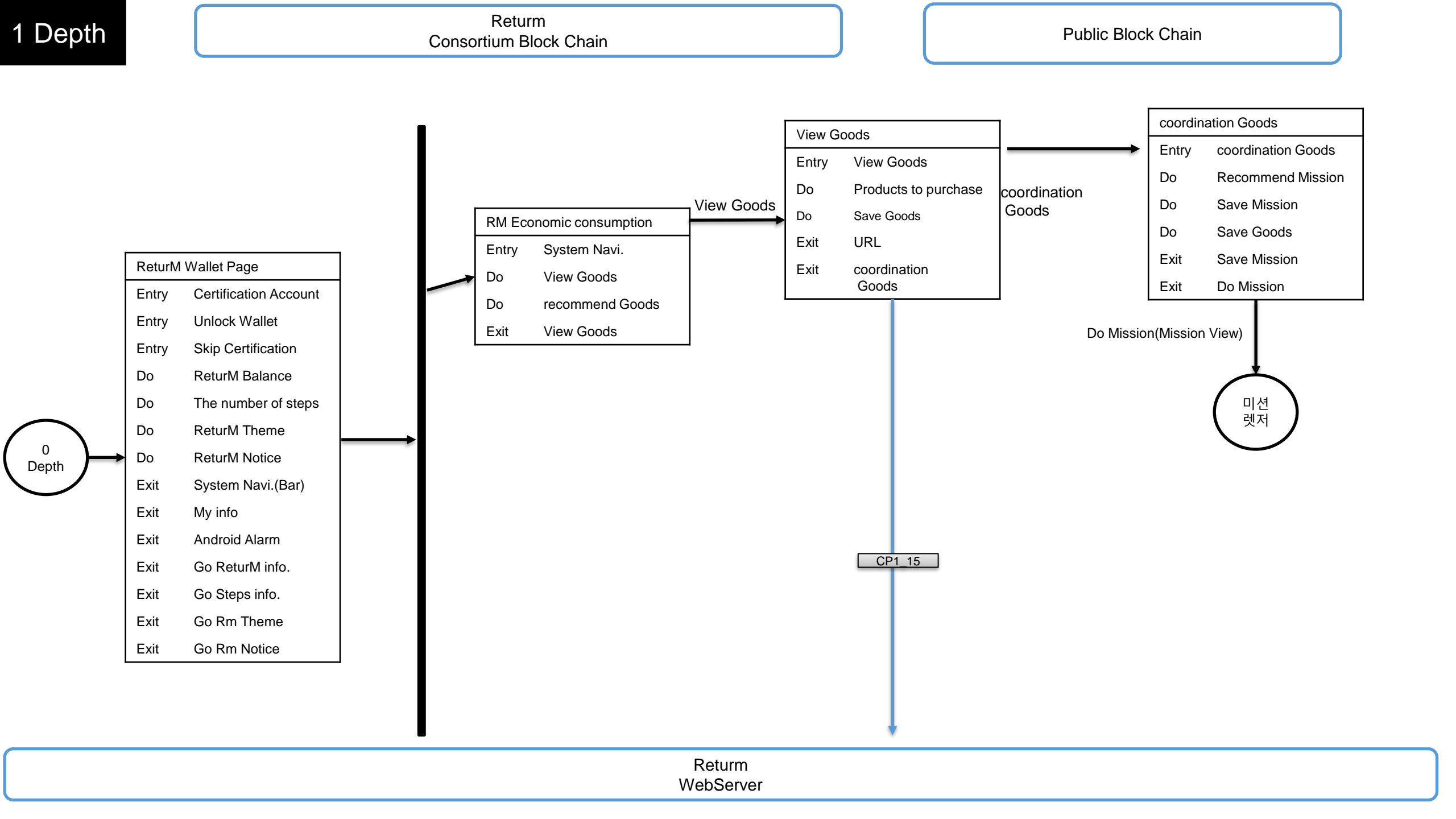
{  
    "PublicKey" : "0000xad5412as53", // 공개키  
    "Signature" : "0000xad5412as53" // 전자 서명  
}

Return Server

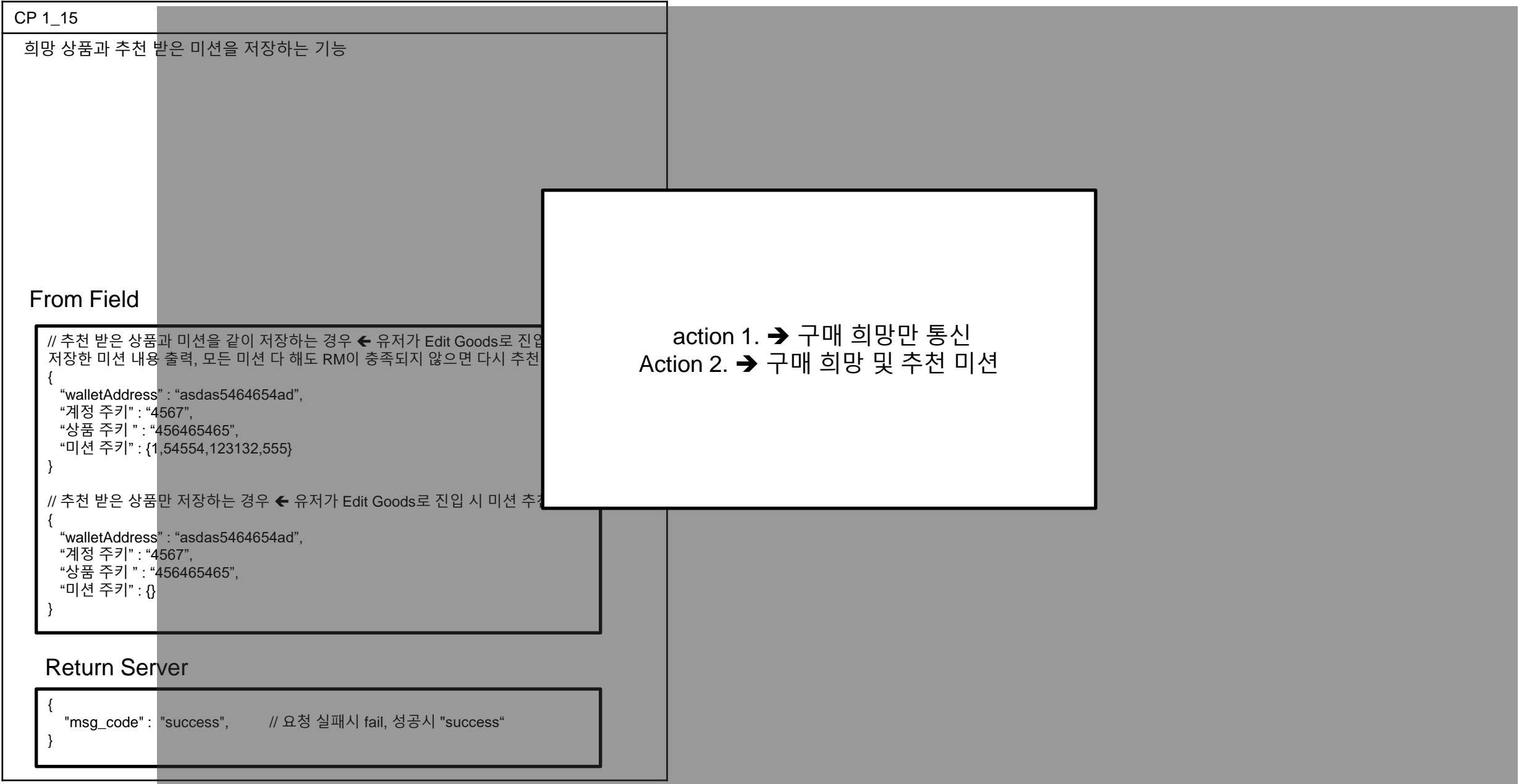


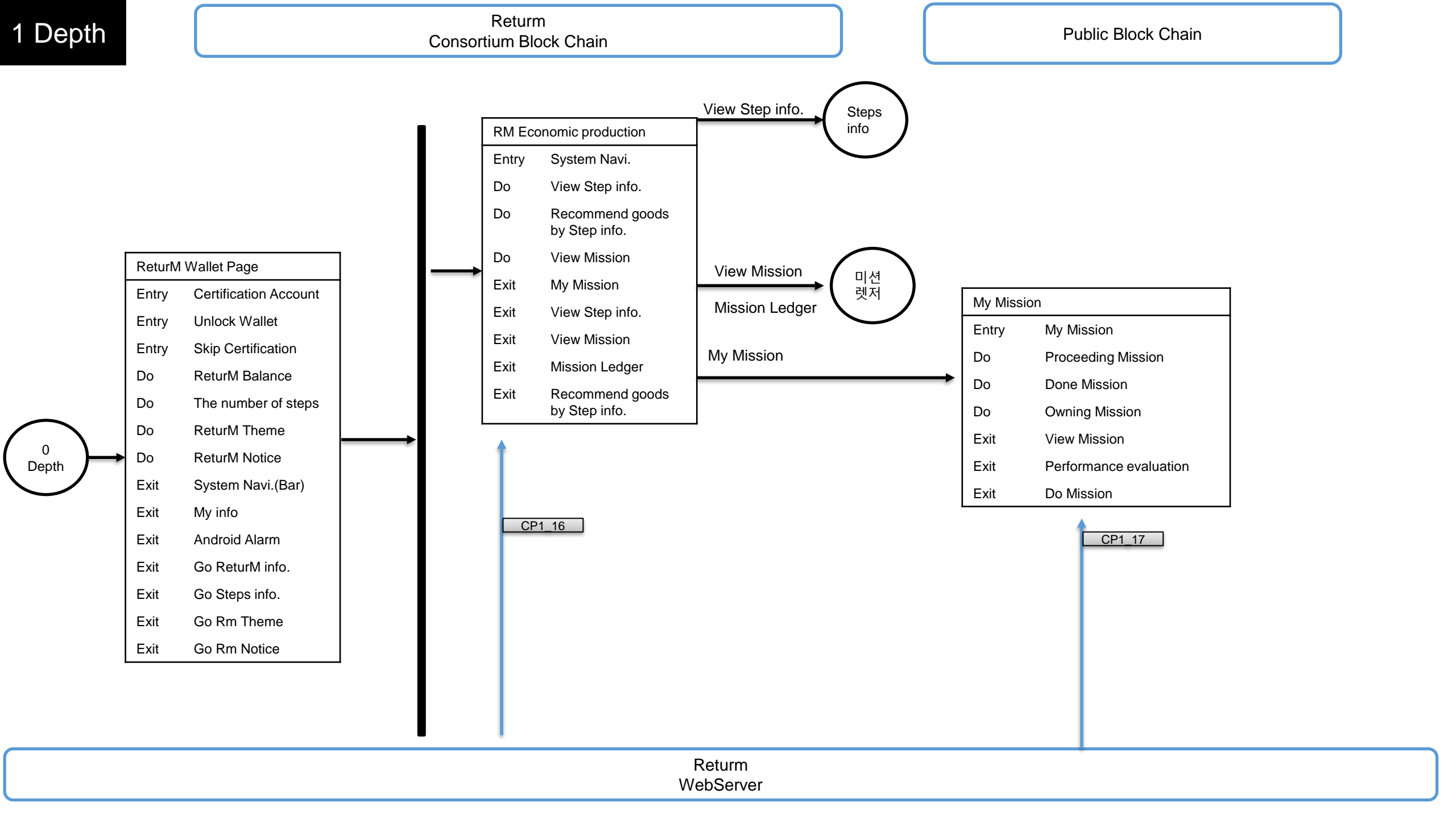
# 1 DEPTH - Communication protocol





# 1 DEPTH - Communication protocol





# 1 DEPTH - Communication protocol

## CP 1\_16

추천 미션 정보와 추천 상품 정보를 반환한다. Client가 유저의 계정 주키를 전송하면 Server는 유저가 설정한 테마와 관련된 추천 미션 / 추천 상품 정보를 반환 받는다. 이때 유저가 계정 인증을 진행하지 않은 경우, Null 값이 전송되고 관리자가 설정한 기본 테마의 추천 미션과 추천 상품 정보를 반환 받는다.

### From Field

```
{
  "계정 주키" : "5555" //계정 주키
}
```

### Return Server

```
{
  "msg_code" : "success",      // 요청 실패시 fail, 성공시 "success"
  'RM_Theme' : [
    "theme_msg_code" : "success", //테마 설정 안 할 시 fail, 설정 시 "success"
    "Common_Goods_info":[],
    "Common_Mission_info":[]
  ]
}
```

## CP 1\_17

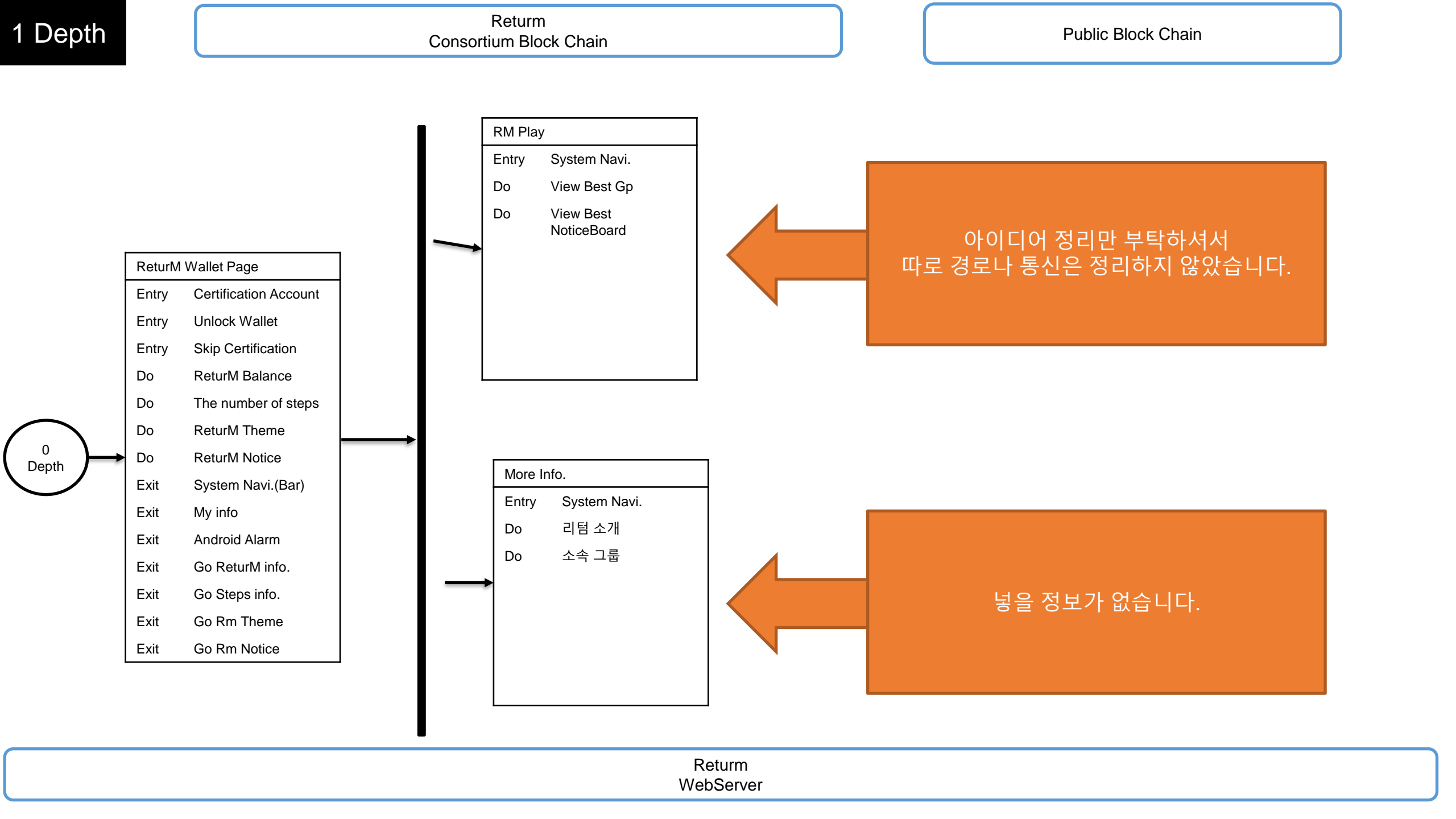
유저가 진행 한 / 진행 중인 / 미션을 게시한 미션에 대한 정보를 반환 받는다. 계정 인증을 하지 않은 유저는 접근할 수 없다.

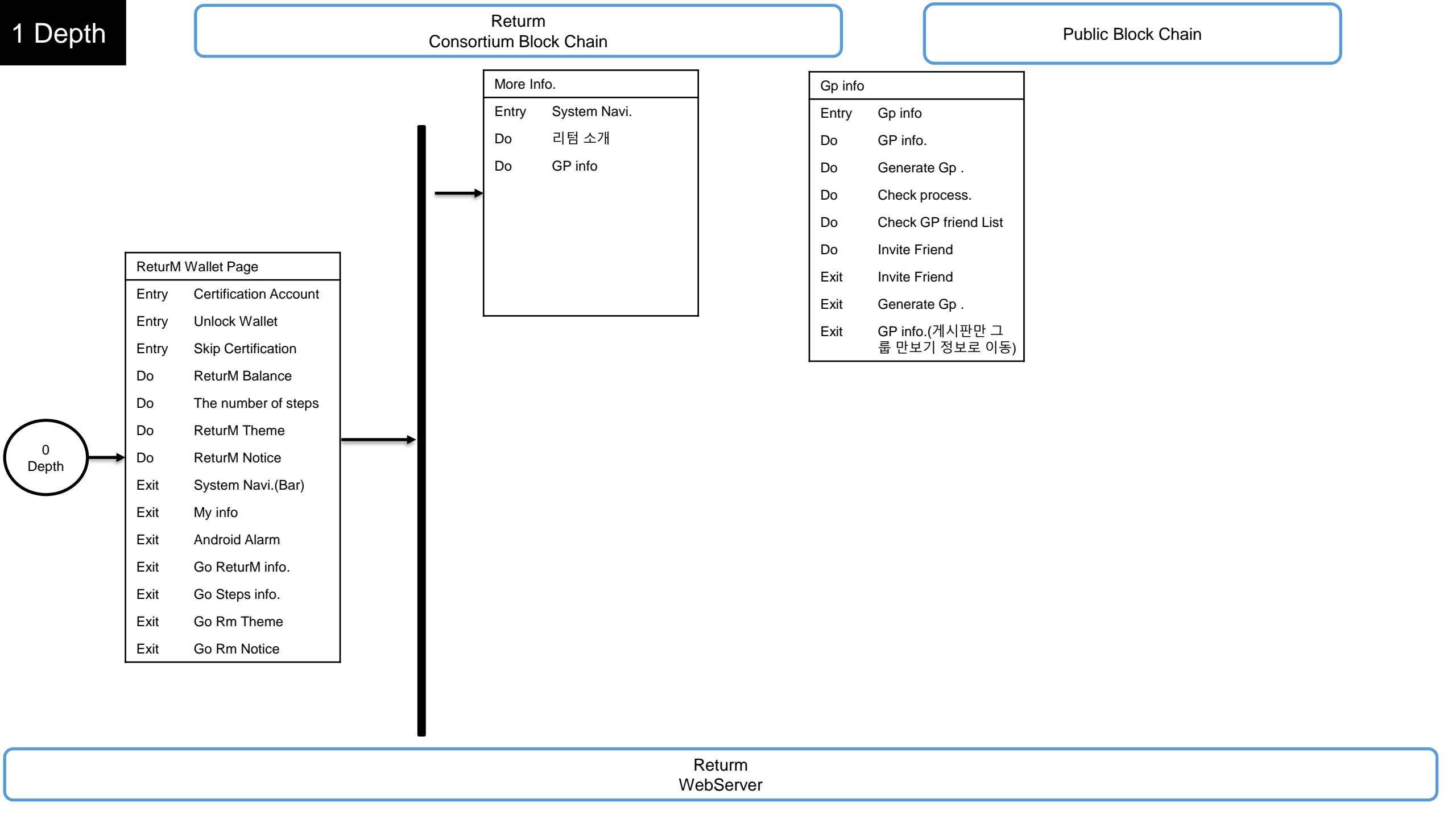
### From Field

```
{
  "계정 주키" : "5555" //계정 주키
}
```

### Return Server

```
{
  "msg_code" : "success",      // 요청 실패시 fail, 성공시 "success"
  "미션 리스트" : [
    {
      미션 주키 : '5555',
      미션 상태 : "완료" // 진행중, 완료, 게시한 미션
      미션 명 : "트와이스 페이스북 클릭",
      미션 종류 : "클릭 미션",
      미션 이미지 : "Ftp/.../aasda.png",
      미션 url : "http://.../index.html",
      미션 보상 : "3.1",
      미션 최대 보상 : "31",
      미션 유효 시작일 : "2018-11-31",
      미션 유효 시작일 : "2018-11-31"
    },{
      ...
    }
  ]
}
```

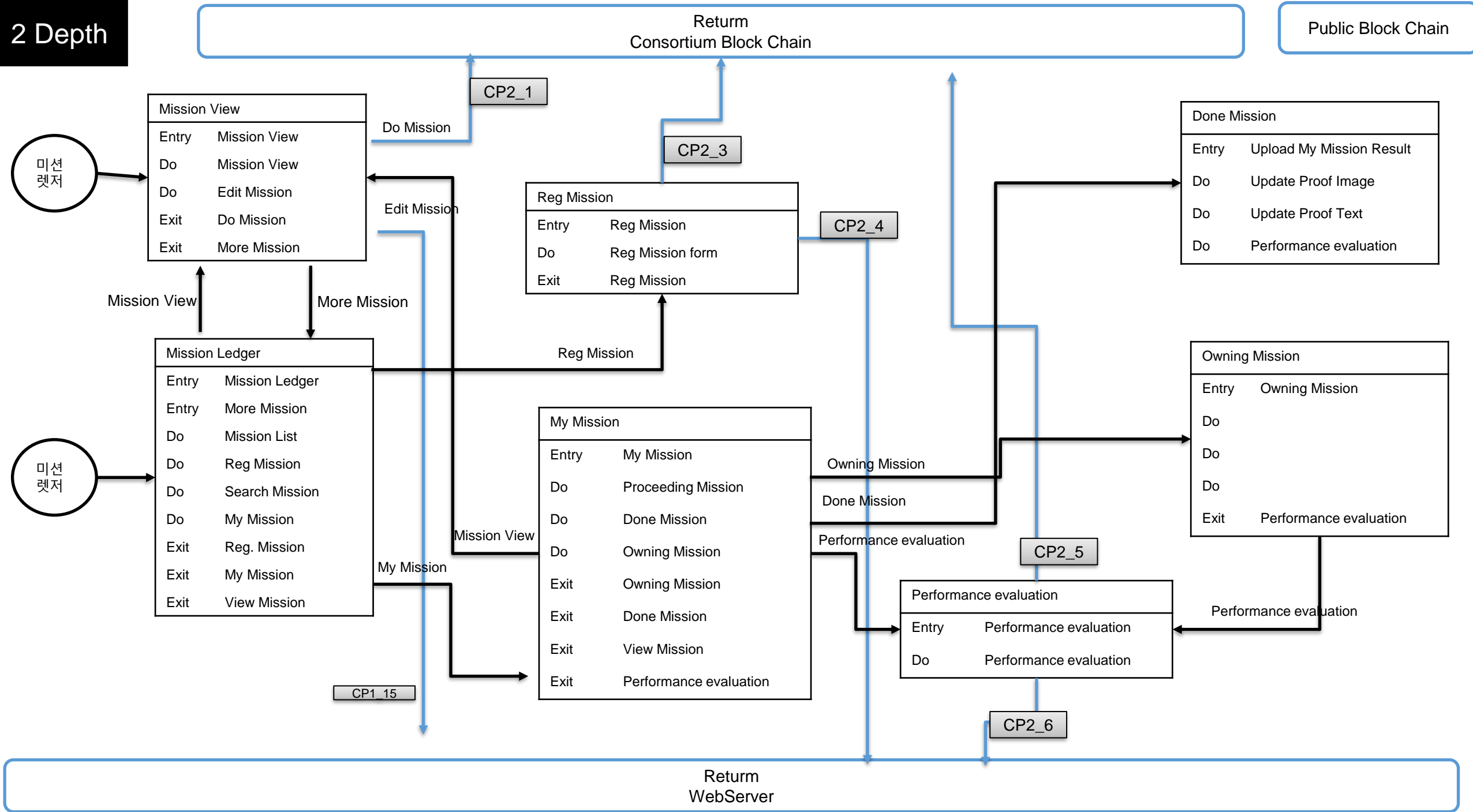




Depth 2



## 2 Depth



# 2 DEPTH - Communication protocol

## CP 2\_1

유저의 미션 수행 내역을 Private BlockChain에 전송한다. 이때 Private BlockChain은 유저의 미션 수행 내역을 확인 한 후, 조건을 충족하면 block를 생성하고 해당 유저에게 보상을 지급한다.

참고 사이트 : <https://steemit.com/coinkorea/@etainclub/smart-contract-6-dapp>

Type	데이터
Text(Hash)	Wallet Address
Double	보상 RM Amount
Text	수행 내역
Date	수행 일자
Text(hash)	Private Key
Text	미션 내용
Text	미션 종류

전자 서명  
(Signature)

### From Field

```
{
  "PublicKey": "0000xad5412as53", // 공개키
  "Signature": "0000xad5412as53" // 전자 서명
  "Smartcontractaddress": "0x6f9c338bb987f1baf619697784c9457b9afa119c",
}
```

### Return Server

```
// 스마트 콘트랙트 조건 미 충족시 보상 받는 RM 정보 없음.
{
  "msg_code": "success", // 요청 실패시 fail, 성공시 "success"
}

// 스마트 콘트랙트 조건 충족시 보상 받는 RM 정보 있음.
{
  "msg_code": "success", // 요청 실패시 fail, 성공시 "success"
  "rm_보상": "3.00"
}
```

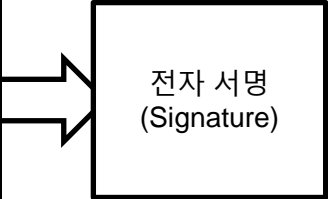
# 2 DEPTH - Communication protocol

## CP 2\_3

유저가 미션을 등록한다. 이때 private Blockchain은 최대 보상 Rm 만큼 유저의 Rm을 Staking 하고, 해당 내용을 이용해 Smart Contrack를 작성하고, 이를 블록에 저장한다. 이때 유저가 계정인증을 하지 않으면 미션을 등록할 수 없다.

참고 사이트 : <https://steemit.com/coinkorea/@etainclub/smart-contract-6-dapp>

Type	데이터
Text(Hash)	Wallet Address(미션 수행자, 보상 받을 지갑)
Double	보상 RM Amount
Double	최대 보상 RM Amount
Date	미션 유효 시작일
Date	미션 유효 종료일
Text	미션 충족 조건
Text(hash)	Private Key



### From Field

```
{
  "PublicKey" : "0000xad5412as53", // 공개키
  "Signature" : "0000xad5412as53" // 전자 서명
}
```

### Return Server

## CP 2\_4

유저가 설정한 미션을 서버에 저장한다. 이때 유저가 계정인증을 하지 않은 경우 미션을 등록 할 수 없다.

### From Field

```
{
}
```

### Return Server

# 2 DEPTH - Communication protocol

## CP 2\_5

유저의 미션 수행 내역을 평가하고 보상으로 리텀을 지급한다. Send 통신과 비슷하고 지급 받을 인원을 리스트 형태로 전송한다.

참고 사이트 : <https://steemit.com/coinkorea/@etainclub/smart-contract-6-dapp>

Type	데이터
Text(Hash)	Wallet Address(보내는 사람)
Text(Hash)	Wallet Address(받는 사람)
Double	RM Amount
Text	Memo
Text(hash)	Private Key

전자 서명  
(Signature)

### From Field

```
{
  {
    "PublicKey": "0000xad5412as53", // 공개키
    "Signature": "0000xad5412as53" // 전자 서명
  },{
    "PublicKey": "0000xad5412as53", // 공개키
    "Signature": "0000xad5412as53" // 전자 서명
  },{
    ..
  }
}
```

### Return Server

## CP 2\_6

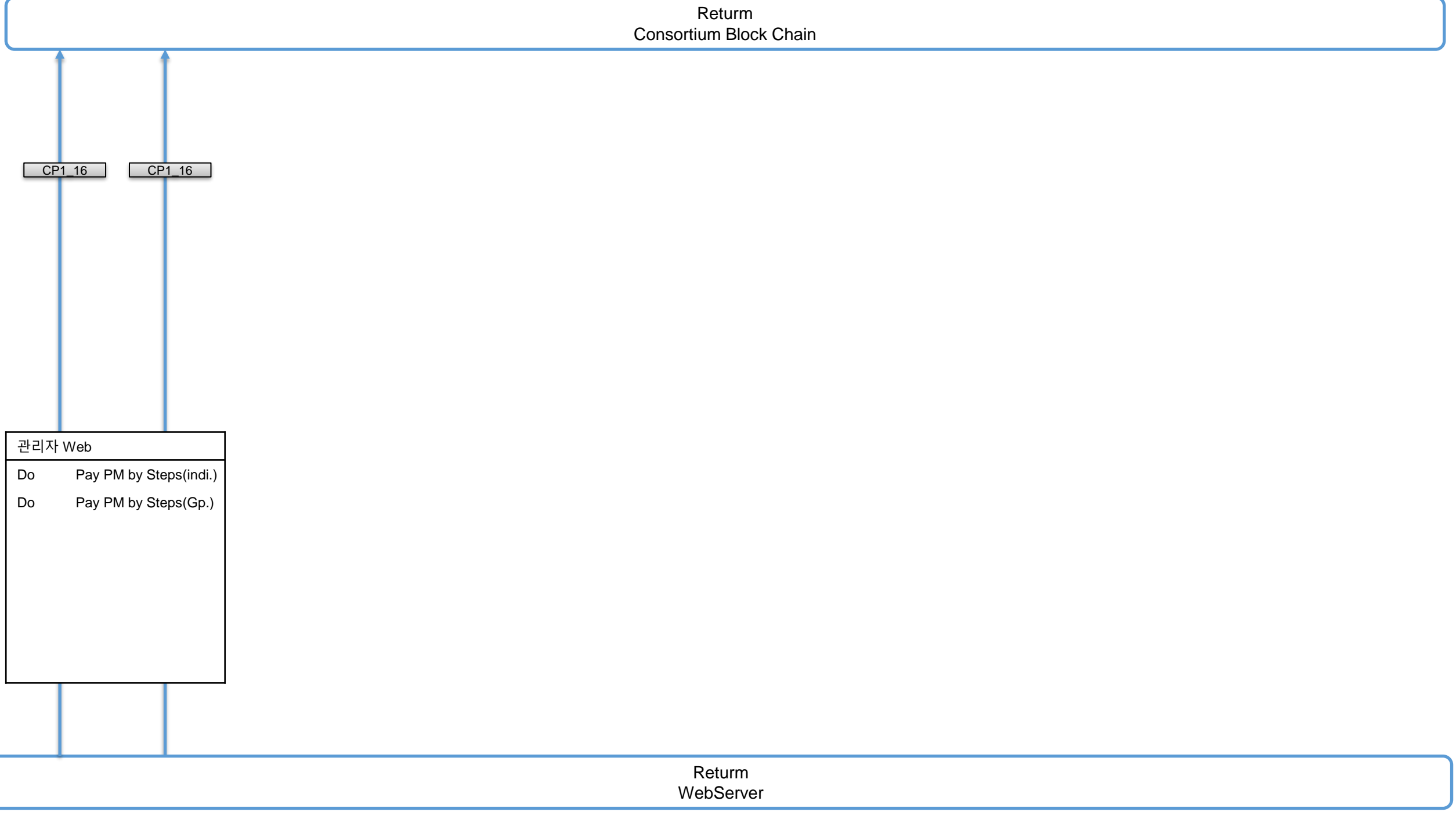
평가된 내용을 서버에 저장한다.

### From Field

```
{
}
```

### Return Server

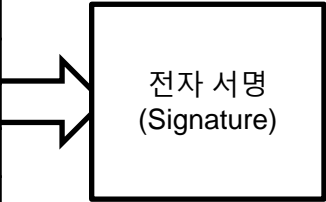
ETC



CP 3\_0

만보기를 통한 보상 RM 내역을 전송한다. 주1회 관리자가 만보기의 량을 보고 그룹 / 개인 보상 RM을 산출(시스템 산출)하여 그 내역을 블록체인에 기록한다. 이때, Private Key는 플랫폼 지갑의 Private Key로 한다.

Type	데이터
Text(Hash)	Wallet Address(, 보상 받을 지갑)
Double	보상 RM Amount
Int	한 주 동안 걸음 수
DateTime	만보기 시작일
DateTime	만보기 종료일
Vector	그 주에 걸은 걸음수의 일일 리스트
Text(hash)	Private Key



From Field

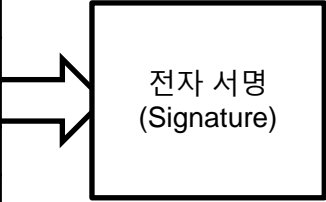
```
{
  "PublicKey": "0000xad5412as53", // 공개키
  "Signature": "0000xad5412as53" // 전자 서명
}
```

Return Server

CP 3\_1

만보기를 통한 보상 RM 내역을 전송한다. CP 3\_0과 같은 개념이나 이 보상은 그룹 참여에 의한 보상이다.

Type	데이터
Text(Hash)	Wallet Address(, 보상 받을 지갑)
Double	보상 RM Amount
Int	한 주 동안 걸음 수
DateTime	만보기 시작일
DateTime	만보기 종료일
Vector	그 주에 걸은 걸음수의 일일 리스트
Text	그룹 명
Int	그룹의 총 걸음 수
Text	그룹장 이름
DateTime	그룹 개설일
Text(hash)	Private Key



From Field

```
{
  "PublicKey": "0000xad5412as53", // 공개키
  "Signature": "0000xad5412as53" // 전자 서명
}
```

Return Server