<Insert OrganizationName>

<Distributed Vending Machine>
Software Architecture Document (SAD)

CONTENT OWNER: <송승현, 유혜리, 이동훈>

1 Introduction

1.1 Stakeholder Representation

Stakeholder	Description
유저	지불한 비용에 대한 물품보장이 되었으면 좋겠다. 자판기의 정확한 위치를 알 수 있었으면 좋겠다. 자판기 사용에 어려움이 없었으면 좋겠다. 자판기에서 제품 환불이 가능했으면 좋겠다.
관리자	자판기의 재고상황을 정확히 알 수 있으면 좋겠다. 재고가 얼마 남지 않았을 경우, 알람이 왔으면 좋겠다. 재고가 가장 없는 자판기를 알 수 있으면 좋겠다. 네트워크 동기화가 실시간으로 이루어지면 좋겠다.
개발자	추후 음료메뉴가 늘어날 시 간편히 메뉴에 추가되었으면 좋겠다.
대표(인수자)	자판기를 동시에 많은 인원이 사용해도 사용에 문제가 없었으면 좋겠다. 자판기 음료 종류를 최대한 다양화 하고 싶다.
디자이너	사용자가 시스템에 대한 학습시간이 최대한 적도록 디자인한다.

1.2 Viewpoint Definitions

Stakeholder	Viewpoint(s) that apply to that class of stakeholder’s concerns
유저	Module View
관리자	C&C View

개발자	Allocation View & C&C View
대표(인수자)	Module View
디자이너	Module View

1.2.1 Viewpoint Definition

1.2.1.1 Abstract

Module View(Layered View)

Layered View 를 위해 참조 아키텍처로 클라이언트는 Rich Client Application Architecture, 서버는 Service Application Architecture 를 사용하였다. Rich Client Application 은 자판기 내 소프트웨어 개발과 관련되며, QA-1 을 사전에 방지하는 터치식으로 하드웨어가 정해진다.

C&C View(Server-Client Style)

Server-Client 관계를 충족시키기 위하여 해당 스타일을 사용하였다. 참조 아키텍처인 Rich Client Application Architecture, Service Application Architecture 로 짜여진 두개의 구조 사이 통신을 Client-Server Style 의 C&C View 로 표현한다. DVM System 은 클라이언트-서버 스타일을 사용한다. 사용하는 서버로부터 DVM(클라이언트 어플리케이션)을 분리하는 시스템 뷰를 표현한다. 이 스타일은 공통 서비스를 분리해냄으로써 시스템의 이해와 재사용성을 지원한다

Allocation View(Deployment Style)

DVM System 의 소프트웨어 요소가 소프트웨어가 실행되는 컴퓨팅 플랫폼의 하드웨어에 할당되는 것을 Deployment Style 로 표현한다. 이 뷰는 성능과 가용성, 신뢰성, 보안을 분석하는데 유용하다.

1.2.1.2 Stakeholders and Their Concerns Addressed

Stakeholder	Concerns Addressed
유저	원하는 기능이 존재하는지 확인한다
대표	원하는 기능이 존재하는지 확인한다.
디자이너	원하는 기능을 최종 사용자가 불편함 없이 사용하는지 확인한다.

1.3 How a View is Documented

Name of view	Module View-Layer Style
Elements	• Layer: 계층은 포함하고 있는 모듈을 정의
Relations	• Allowed-to-use: 일반적인 종속 관계(Depends-on)의 특화 • 디자인 시 계층간 사용 규칙을 정의

Constraints	• 소프트웨어의 모든 부분은 정확히 한 계층에 할당 • 적어도 2 개 이상의 계층이 존재 • Allowed-to-use 관계가 순환되면 않됨
What it's for	• 변경용이성과 이식성 증대 • 재사용성 증진 • Concern 의 분리

Name of view	C&C View-ClientServer Style
Elements	• Client: 다른 컴포넌트에 서비스를 요청 • Server: 다른 컴포넌트에 서비스를 제공 • Request/Response 커넥터: 서버에서 제공하는 서비스에 대한 클라이언트의 비대칭적 호출
Relations	• Attachment: 클라이언트를 커넥터의 요청 역할에 묶어주고, 서버를 커넥터의 응답 역할에 묶어준다. 또한 어느 서비스가 어느 클라이언트로부터 요청을 받을 수 있는지도 결정한다.
Constraints	• 클라이언트는 Request/Response 커넥터를 통하여 서버와 연결 • 서버는 다른 서버에 대하여 클라이언트가 될 수 있다 • 특화는 다음과 같은 제한 사항을 내포: 포트에 대한 attachment 개수, 서버 사이의 허용된 관계 • 컴포넌트는 계층적으로 나뉠 수 있음
What it's for	• 변경용이성과 재사용성 증대, 확장성과 가용성 향상, 종속성과 보안 및 성능 분석

Name of view	Allowed view-deployments style
Elements	• 소프트웨어 요소: C&C 뷰의 요소 - 하드웨어로부터 요구되는 중요한 특성들을 포함한

	문서화를 위한 유용한 특성들 - 프로세싱, 메모리, 용량 요구사항, 결함 방지 등
Relations	• Allocated-to: 소프트웨어 요소가 어떤 물리적 장치에 탑재되는지 보여줌 - 특성은 할당이 실행 시간에 변경 가능한지 여부를 포함 • Migrates-to, copy-migrates-to, execution-migrates-to: 할당이 동적으로 이루어지는 경우에 사용
Constraints	• 할당 구성형태(Allocation Topology)에 대한 제약 없음 - 하드웨어에 의해 제공되는 특성에 의해 소프트웨어 요구되는 특성은 반드시 만족되어야함

1.4 Relationship to Other SADs

Not applicable

1.5 Process for Updating this SAD

gpfl111@konkuk.ac.kr 에 수정이 필요한 내용을 항목으로 정리하여 보내주시면 됩니다.

2 Architecture Background

2.1 Problem Background

2.1.1 System Overview

DVM 시스템은 재고가 없는 상품의 구매까지 지원하는 자판기 시스템이다. 자판기에 있는 Viewer 를 통해 상품을 선택, QR 결제하는 방식으로 상품을 구매할 수 있다. 재고가 없는 경우에는 해당 상품의 재고가 있는 다른 DVM 을 사용자가 선택하고 상품 수령코드를 받아, 선택한 DVM 에서 상품을 수령이 가능하다.

2.1.2 Goals and Context

2.1.2.1 Goal

- Stakeholder 의 요구사항을 반영한다.
- 시스템 유지보수를 용이하도록 돕는다.
- 효율적인 시스템 개발을 용이하도록 돕는다.

2.1.2.2 Context

- QAW 를 통해 산출된 Architecturally Significant Requirement 는 아래 내용과 같다. 설명한다.
- 본 시스템의 Primary functionality 는 2.1.3 절에서 설명한다.

ID	Use Case Description

UC-1	상품 선택 버튼을 누른다
UC-2	코드 입력 버튼을 누른다
UC-3	코드를 환불한다
UC-4	관리자 버튼을 누른다
UC-5	상품을 선택한다
UC-6	자판기에 재고가 있거나, 다른 재고가 있는 자판기 조회를 마친 경우 QR 결제를 요청한다
UC-7	재고가 없을 경우, 재고가 있는 다른 자판기를 조회한다
UC-8	자판기의 재고정보를 네트워크 상에 동기화 한다
UC-9	자판기의 네트워크 상태를 점검한다
UC-10	결제가 이루어진 자판기에서 선결제한 상품을 제공할 자판기로 코드를 전송한다
UC-11	관리자 상태에서 자판기의 재고를 변경한다

ID	Constraint
CON-1	최소 1000 명의 동시 사용자를 지원해야 한다.
CON-2	보안 관련한 문제가 발생할 경우 이와 관련한 정보가 관리자에게 빠르게 전달되어야 한다.
CON-3	네트워크 성능은 5 분 간격으로 점검할 수 있어야 한다.
CON-4	총 개발인원이 3 명을 넘지 않아야 한다.

ID	Concerns
CRN-1	최소 사양의 하드웨어에서도 원할히 작동해야 한다
CRN-2	자판기의 네트워크 연결을 관리하는 모듈이 필요하다
CRN-3	결제 완료 시, 실시간으로 재고상태가 동기화가 되어야 한다

2.1.3 Significant Driving Requirements

QAW 를 통해 산출된 Quality Attribute 는 다음과 같다.

ID	Quality Attribute	Scenario
QA-1	Robustness	사용자의 비정상적인 입력 값에 대하여 올바른 예외처리와 오류경고를 출력할 수 있어야한다.
QA-2	Security	외부에서 네트워크에 접근하려고 할 때, 30 초 내로 접근을 방지하고 관리자에게 관리자에게 외부 접근 알림을 전송한다.
QA-3	Availability	인증코드의 유효기간은 일주일이며 유효기간내에는 환불이 가능하다
QA-4	Usability	DVM 는 항상 동기화되어, 정확한 재고를 제공해야 한다.
QA-5	Scalability	DVM 수가 늘어날 경우 시스템 성능문제에 영향을 주지 않아야 한다.

2.2 Solution Background

2.2.1 Architectural Approaches

2.2.1.1 Module View

Layered View 를 위해 참조 아키텍처로 클라이언트는 Rich Client Application Architecture, 서버는 Service Application Architecture 를 사용하였다. Rich Client Application 은 자판기 내 소프트웨어 개발과 관련되며, QA-1 을 사전에 방지하는 터치식으로 하드웨어가 정해진다.

2.2.1.2 C&C View

Server-Client 관계를 충족시키기 위하여 해당 스타일을 사용하였다. 참조 아키텍처인 Rich Client Application Architecture, Service Application Architecture 로 짜여진 두개의 구조 사이 통신을 Client-Server Style 의 C&C View 로 표현한다. DVM System 은 클라이언트-서버 스타일을 사용한다. 사용하는 서버로부터 DVM(클라이언트 어플리케이션)을 분리하는 시스템 뷰를 표현한다. 이 스타일은 공통 서비스를 분리해냄으로써 시스템의 이해와 재사용성을 지원한다

2.2.1.3 Allocation View

DVM System 의 소프트웨어 요소가 소프트웨어가 실행되는 컴퓨팅 플랫폼의 하드웨어에 할당되는 것을 Deployment Style 로 표현한다. 이 뷰는 성능과 가용성, 신뢰성, 보안을 분석하는데 유용하다.

2.2.2 Analysis Results

해당 없음.

2.2.3 Requirements Coverage

ADD 3.0 의 1st iteration 과정에서 전체 시스템의 전반적인 구조 수립.

ADD 3.0 의 2nd iteration 과정에서 fuctionality 를 포함한 구현단위 추론 수립.

ADD 3.0 의 3rd iteration 과정에서 Quality Attributes 를 충족시키기 위한 Tactic 수립.

2.2.4 Summary of Background Changes Reflected in Current Version

해당 없음.

2.3 Product Line Reuse Considerations

시스템 재사용성을 위해 구현 수준에서 시스템이 모듈 단위로 분리되어 구현된다. 또한 각 모듈은 서로의 기능이 중첩되지 않도록 한다.

시스템은 increase resource 를 항상 고려하며, DVM 하드웨어 수가 늘어날 경우 추가 프로세서, 추가 메모리를 통해 시간을 감소시킬 수 있는 잠재력을 항상 지닌다.

3 Views

The views presented in this SAD are the following:

Name of view	Viewtype that defines this view	Types of elements and relations shown		Is this a module view?	Is this a component-and-connector view?	Is this an allocation view?
Layered View	Logical View	Module, Layer	'allowed-to-use'	O	X	X
C&C View - Server-Client Style	Process View	Client, Server, Request/Response Connector	attachment	X	O	X
Allocation View – Deployment Style	Development View	Elements of C&C View, Hardware of the computing platform	Allocate-to, Migrated-to, copy migrates-to, execution migrates-to	X	X	O

3.1 Layered View

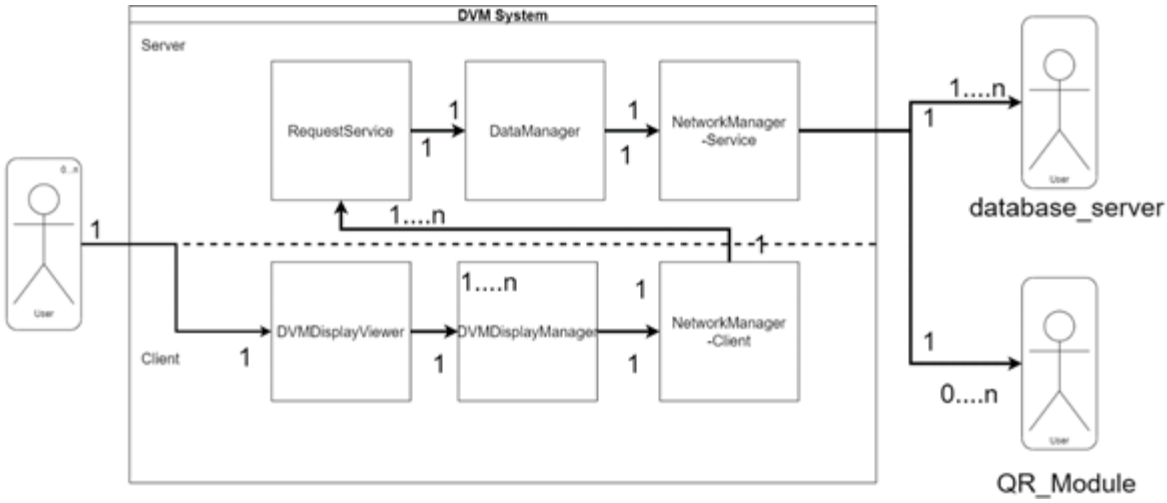
3.1.1 View Description

참조 아키텍처인 Rich Client Application Architecture, Service Application Architecture 에 맞도록 각 시스템을 3 개의 Layer 로 구분하고, 각 layer 내부에 모듈을 배치하였다.

3.1.2 Architecture Background

DVM System 은 크게 Client 와 Server 단으로 구분된다. Client 는 사용자와의 상호작용을, Server 는 데이터베이스, 외부 모듈과의 통신을 주로 담당하며, 자세한 구조는 3.1.3 에서 설명한다.

3.1.3 Variability Mechanisms



DVM system 은 모든 DVM 기기와 연결되어 있는 단일 database server 로 이루어져 있다. 각 DVM 마다 Client 구조를 포함하고 있으며 Client 는 DVMDisplayViewer, DVMDisplayManager 로 이루어져 있으며 NetworkManger-Client 가 Server 의 RequestService 와 통신하여 DataManager 와 NetworkManager-Server 를 통해 DB 와 외부모듈인 QR 모듈에 Client 의 request 를 전달한다. QR 모듈은 결제에 사용되는 모듈로 결제 성공-실패 여부를 전달해준다.

3.1.4 View Packets

3.1.4.1 View packet # 1 - Client packet

3.1.4.1.1 Primary Presentation

리치 클라이언트

Element		Relation
Layer	Module	
Presentation Layer	DVMDisplayViewer	DVMDisplayManager 와 'allowed-to-use' 관계
Business Layer	DVMDisplayManager	NetworkManager-Client 와 'allowed-to-use' 관계
Data Layer	NetworkManager-Client	RequestService 와 'allowed-to-use' 관계

3.1.4.1.2 Element Catalog

3.1.4.1.2.1 Elements

Element	Description
1. ViewerComponent	2. 사용자의 입력에 따라 화면을 보여준다. 3. 제공되는 interface: 4. DVMDisplayViewer
ManageComponent	사용자의 입력을 받은 Viewer 에서 받은 값에 따라 화면의 상태를 전환한다. 제공되는 interface: NetworkManager-Client
ClientNetwork-Component	RequestComponent 에 신호를 보내는 역할을 한다.

	제공되는 interface: NetworkManager-
RequestComponent	RequestComponent 는 ClientNetworkComponent 에서 받은 신호를 Data-Component 로 보내는 역할을 한다. 제공되는 interface: RequestService
DataComponent	DataComponent 는 RequestComponent 에서 받은 신호를 ServerNetwork-Component 로 보내는 역할을 한다. 제공되는 interface: Datamanager
ServerNetwork-Component	ServerNetworkComponent 는 Database 를 업데이트하고, 변경 사항을 Man-ageComponent 로 보내는 역할을 한다. 제공되는 interface: NetworkManager-Server

3.1.4.1.2.2 Relations

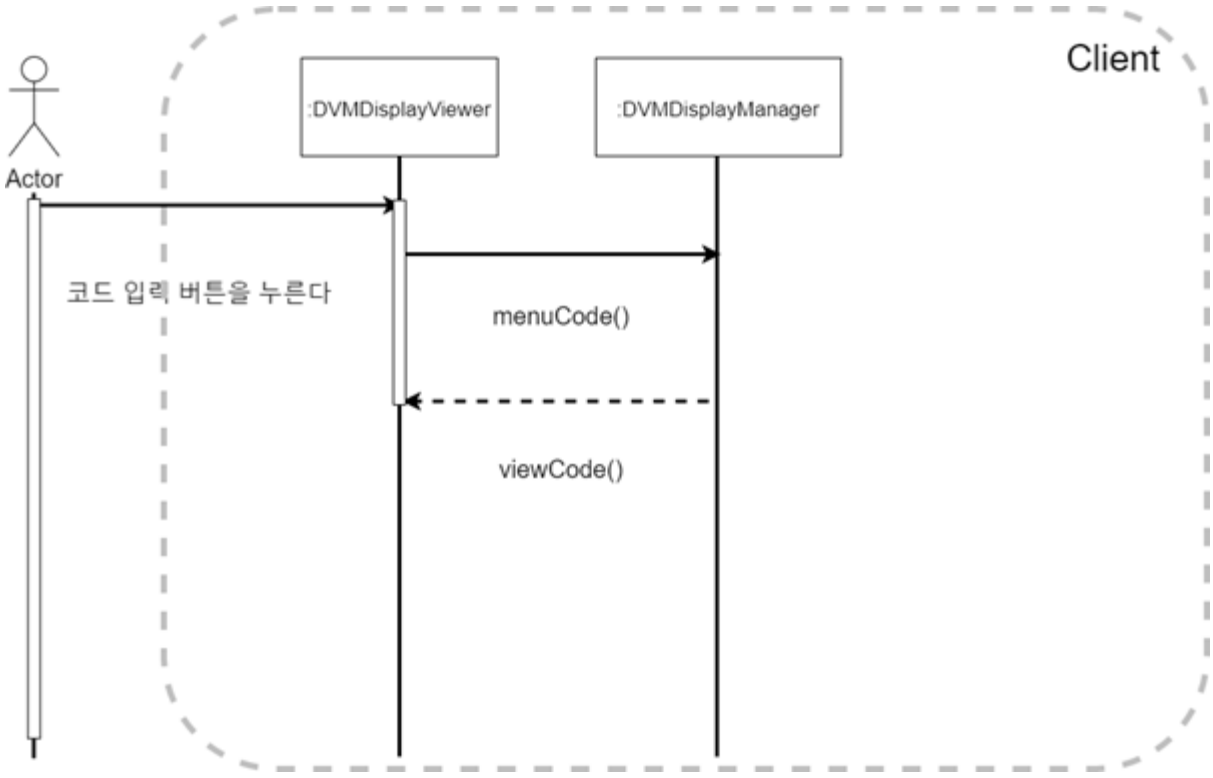
- 3.1.4.2.2.2 에서 설명한다.

3.1.4.1.2.3 Interfaces

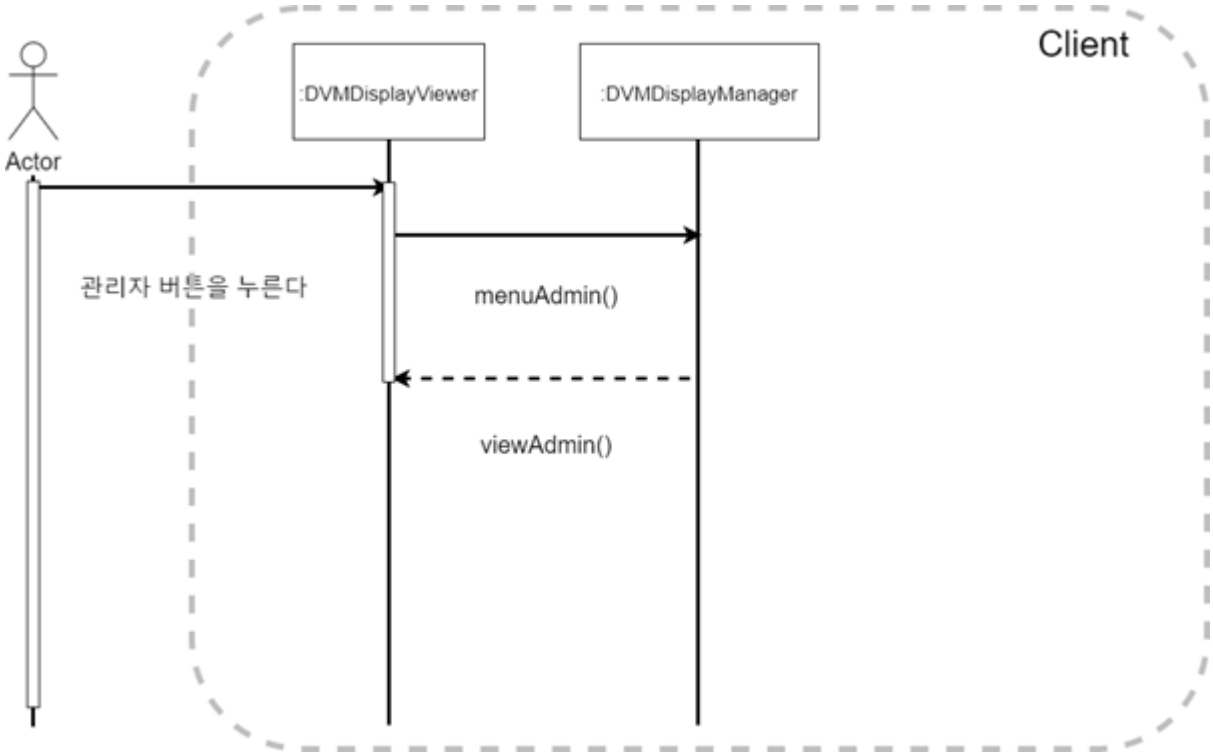
Interface	Description
DVMDisplayViewer	사용자와 터치방식으로 직접적인 interaction 을 통해 데이터 수신. 데이터 교류에 필요한 모든 DVM Display 화면 생성.

DVMDisplayMan-ager	DVM Display 화면 정보 관리
NetworkManager-Client	서버와의 데이터 통신
RequestService	클라이언트와의 데이터 교류
DataManager	데이터베이스 서버와 교류되는 데이터 관리
NetworkManager-Server	데이터베이스 서버와의 데이터 교류

3.1.4.1.2.4 Behavior



사용자가 메뉴에서 코드 입력을 선택 화면에 코드 입력 화면 출력



사용자가 메뉴에서 관리자 로그인을 선택 화면에 관리자 로그인 화면 출력

3.1.4.1.2.5 Constraints

- 최소 1000 명의 동시 사용자를 지원해야 한다.
- 총 개발인원이 3 명을 넘지 않아야 한다.

3.1.4.1.3 Context Diagram

- 3.1.4.2.3 에서 설명한다.

3.1.4.1.4 Variability Mechanisms

- 3.1.3 에서 설명한다.

3.1.4.1.5 Architecture Background

- 3.1.2 에서 설명한다.

3.1.4.1.6 Related View Packets

- 3.1.4.2.2.2 에서 설명한다

3.2

3.1.4.2 View packet # 2 - Server packet

3.1.4.2.1 Primary Presentation

Element		Relation
Layer	Module	
Service Layer	RequestService	DataManager 와 'allowed-to-use' 관계
Business Layer	DataManager	NetworkMager-Server 와 'allowed-to-use'관계
Data Layer	NetworkManager-Server	

3.1.4.2.2 Element Catalog

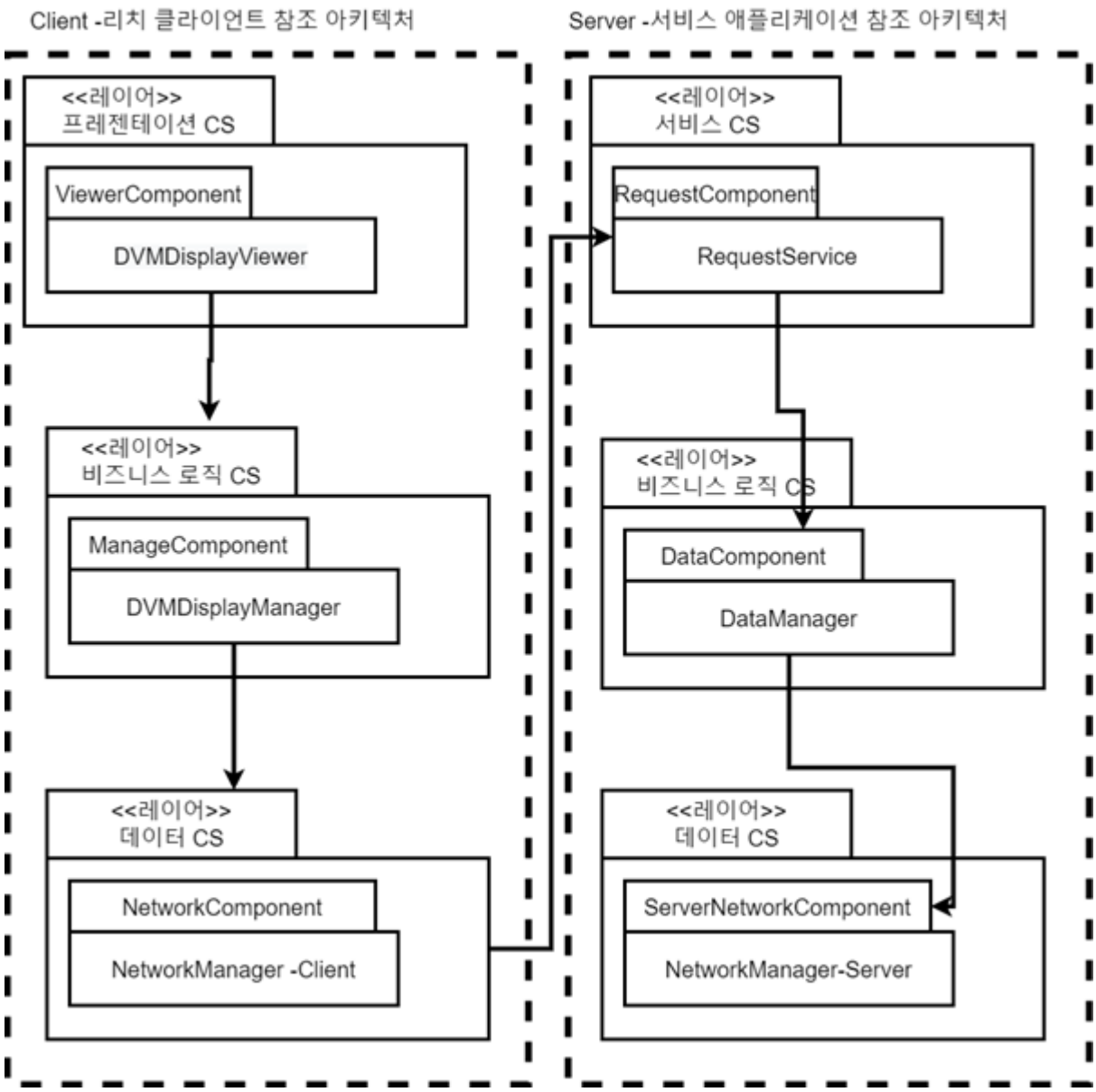
3.1.4.2.2.1 Elements

Element	Description
RequestComponent	RequestComponent 는 ClientNetworkComponent 에서 받은 신호를 DataComponent 로 보내는 역할을 한다.

	제공되는 interface: RequestService
DataComponent	DataComponent 는 RequestComponent 에서 받은 신호를 Server-NetworkComponent 로 보내는 역할을 한다. 제공되는 interface: Datamanager
ServerNetworkComponent	ServerNetworkComponent 는 Database 를 업데이트하고, 변경사항을 ManageComponent 로 보내는 역할을 한다. 제공되는 interface: NetworkManager-Server

3.1.4.2.2.2 Relations

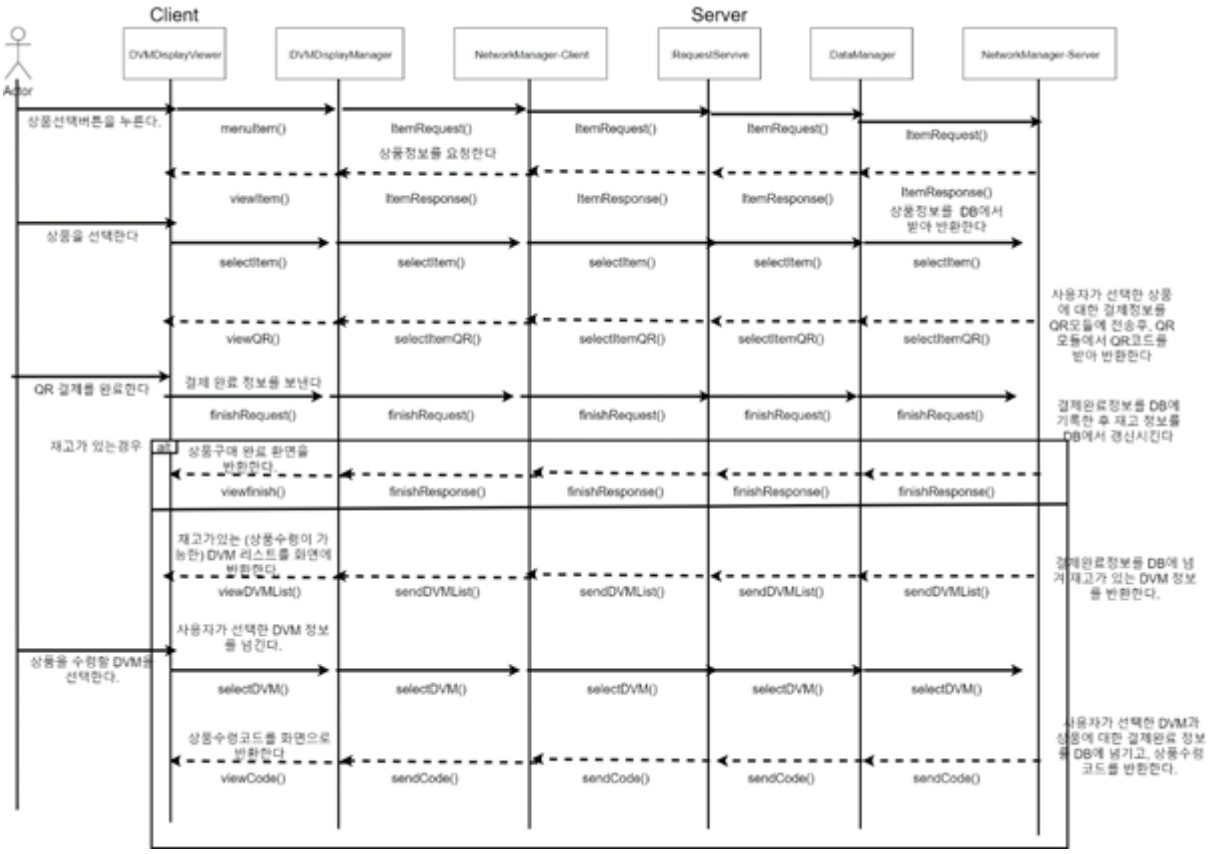
Layered View 는 DVM System 의 구현단위 그룹화를 보여준다. 다음절에서 나오는 C&C View 와 밀접한 연관이 있으며 다음절에서는 구현단위 그룹화 이외의 통신 연결과 관련한 다중 티어 스타일을 설명한다.



3.1.4.2.2.3 Interfaces

Interface	Description
RequestService	클라이언트와의 데이터 교류
DataManager	데이터베이스 서버와 교류되는 데이터 관리
NetworkManager-Server	데이터베이스 서버와의 데이터 교류

3.1.4.2.2.4 Behavior

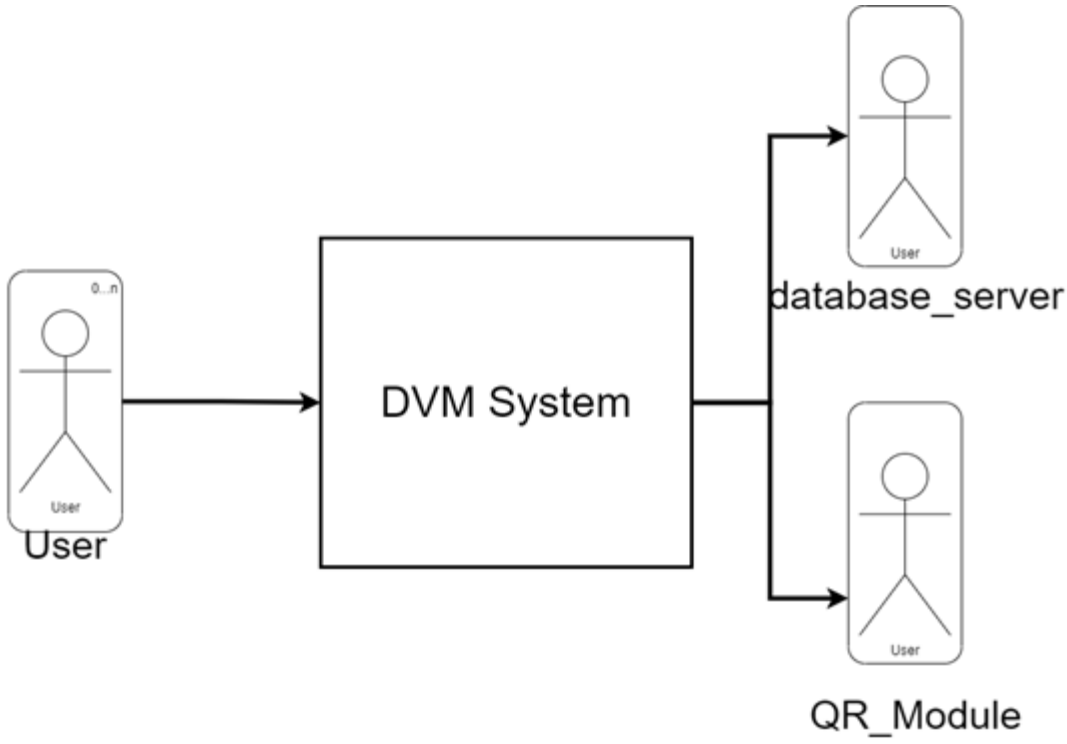


사용자가 상품을 선택하고 해당 DVM 에 재고가 있는지 확인 후 결제를 진행, 재고가 없을 경우 다른 DVM 의 재고를 조회하고 선결재를 진행한다.

3.1.4.2.2.5 Constraints

- 보안 관련한 문제가 발생할 경우 이와 관련한 정보가 관리자에게 빠르게 전달되어야 한다.
- 네트워크 성능은 5 분 간격으로 점검할 수 있어야 한다.

3.1.4.2.3 Context Diagram



3.1.4.2.4 Variability Mechanisms

- 3.1.3 에서 설명한다.

3.1.4.2.5 Architecture Background

- 3.1.2 에서 설명한다.

3.2 C&C View - Client-Server Style

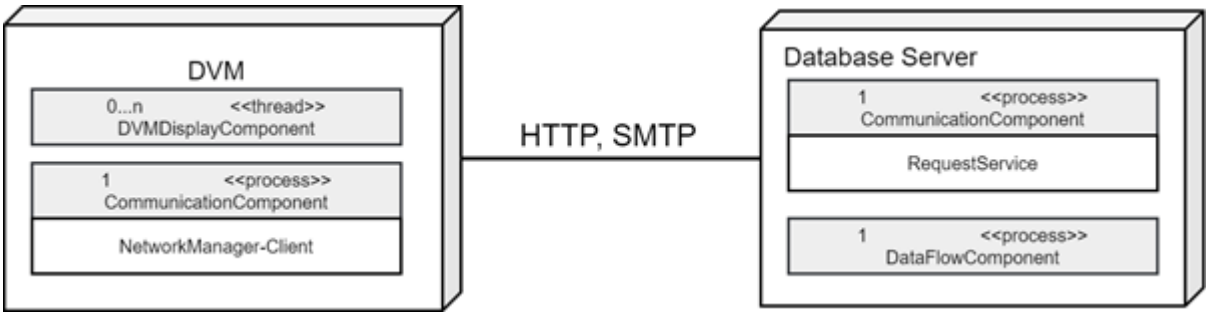
3.2.1 View Description

참조 아키텍처인 Rich Client Application Architecture, Service Application Architecture 로 짜여진 두개의 구조 사이 통신을 Client-Server Style 의 C&C View 로 표현한다. DVM System 은 클라이언트-서버 스타일을 사용한다. 사용하는 서버로부터 DVM(클라이언트 어플리케이션)을 분리하는 시스템 뷰를 표현한다. 이 스타일은 공통 서비스를 분리해냄으로써 시스템의 이해와 재사용성을 지원한다.

3.2.2 Architecture Background

DVM System 은 서버와 클라이언트 구조로 형성되며, 이는 NetworkManager-Client 가 클라이언트 통신을, RequestService 가 서버의 통신을 주관한다. 이 두 컴포넌트를 통해 사용자의 interaction 에 따른 데이터베이스 및 외부 모듈의 모든 데이터 통신이 발생한다. 이는 C&C View 의 Client-Server Style 관점에서 설명할 수 있다.

3.2.3 Variability Mechanisms



3.2.4 View Packets

3.2.4.1 View packet # 1

3.2.4.1.1 Primary Presentation

요소	클라이언트: 서버 컴포넌트 데이터를 호출하는 컴포넌트 서버: 클라이언트 컴포넌트에게 데이터를 제공하는 컴포넌트, 속성은 아키텍트의 관심사에 따라 다르지만 서버 포트의 특성에 대한 정보를 포함한다. RequestComponent: 클라이언트가 서버에 있는 데이터를 호출하는데 사용한다.

관계	결합관계는 클라이언트의 Service-Requset 포트와 커넥터의 요청 역할을 연관시킨다.
컴퓨팅 모델	클라이언트는 필요할 때 서버로부터 데이터를 호출하고 결과를 기다림으로써 상호작용을 시작한다.
제약사항	클라이언트 요청/응답 커넥터를 통해 서버와 연결된다.
사용	공통 서비스를 추출함으로써 재사용성과 변경용이성을 증진시킨다.

3.2.4.1.2 Element Catalog

3.2.4.1.2.1 Elements

Element	Description
ClientNetworkComponent	RequestComponent 에 신호를 보내는 역할을 한다. 제공되는 interface: NetworkManager-
RequestComponent	RequestComponent 는 ClientNetworkComponent 에서 받은 신호를 DataComponent 로 보내는 역할을 한다. 제공되는 interface: RequestService
ServerNetworkComponent	ServerNetworkComponent 는 Database 를 업데이트하고, 변경 사항을 ManageComponent 로 보내는 역할을 한다. 제공되는 interface: NetworkManager-Server

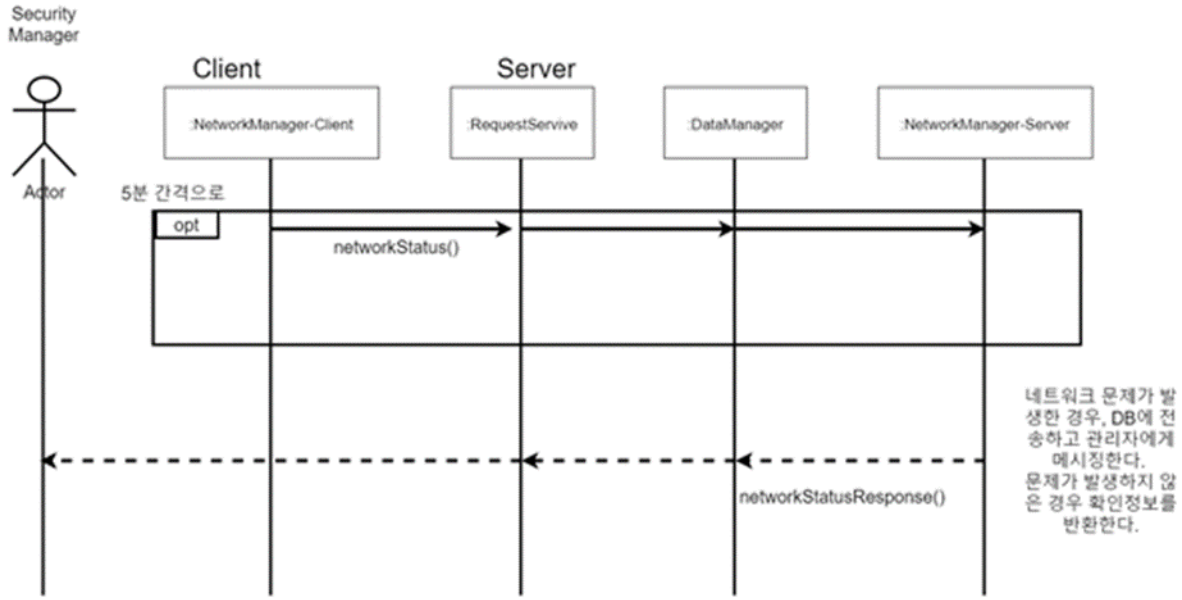
3.2.4.1.2.2 Relations

다른 C&C 스타일과 마찬가지로, 클라이언트-서버 스타일은 서비스와 데이터의 생산자를 해당 서비스와 데이터의 소비자로부터 분리한다. 현재 DVM System 의 클라이언트와 서버는 그룹화 되어 다중 티어 계층을 띄고 있으며 이는 분산환경 안의 다른 머신에 배포된다.

3.2.4.1.2.3 Interfaces

Interface	Description
NetworkManager-Client	서버와의 데이터 통신
RequestService	클라이언트와의 데이터 교류
NetworkManager-Server	데이터베이스 서버와의 데이터 교류

3.2.4.1.2.4 Behavior



네트워크 상태정보를 5 분마다 전송, 문제가 발생할 경우 보안관리자에게 메일 전송

3.2.4.1.2.5 Constraints

- 네트워크 성능은 5 분 간격으로 점검할 수 있어야 한다.

3.2.4.1.3 Context Diagram

- 3.2.3 참조.

3.3 Allocation View - Deployment Style

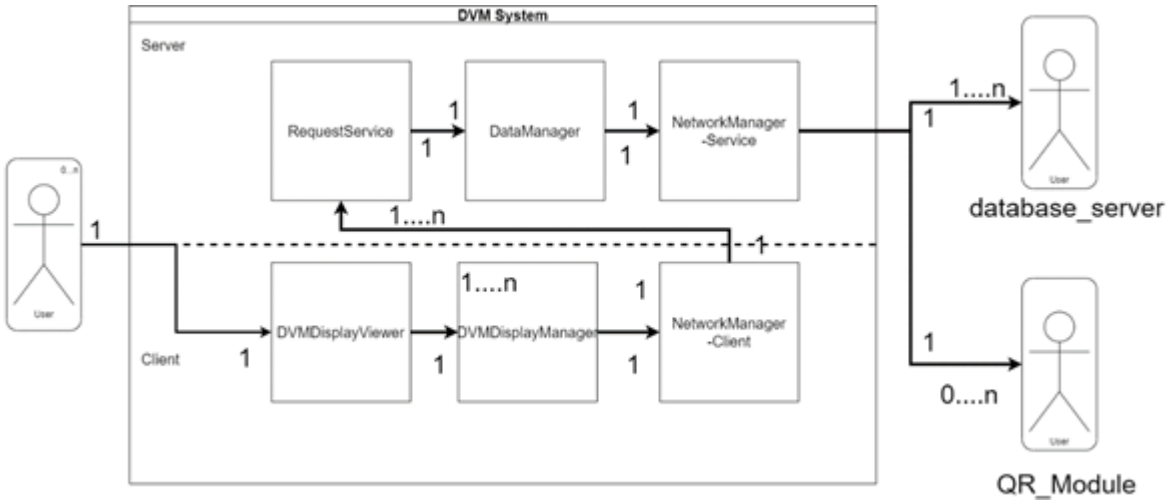
3.3.1 View Description

3.2 에서 설명한 DVM System 의 소프트웨어 요소가 소프트웨어가 실행되는 컴퓨팅 플랫폼의 하드웨어에 할당되는 것을 Deployment Style 로 표현한다. 이 뷰는 성능과 가용성, 신뢰성, 보안을 분석하는데 유용하다.

3.3.2 Architecture Background

DVM System 의 하드웨어 요소는 DVM 기기에 부착된 하드웨어이며, 각 하드웨어 내에 소프트웨어가 배포된다.

3.3.3 Variability Mechanisms



3.3.4 View Packets

3.3.4.1 View packet # 1

3.3.4.1.1 Primary Presentation

개요	Allocation View 는 소프트웨어 아키텍처와 환경 사이의 매핑을 서술한다. 그 중에서 Deployment Style 은 소프트웨어 아키텍처의 컴포넌트와 커넥터를 컴퓨팅 플랫폼의 하드웨어에 매핑을 서술한다.
요소	소프트웨어 요소와 환경 요소. 소프트웨어 요소는 환경에 요구되는 속성을 갖는다. 환경 요소는 소프트웨어에게 제공된 속성을 갖는다. - 소프트웨어 요소 : C&C View 의 요소와 동일.

	- 환경 요소 : 컴퓨팅 플랫폼의 하드웨어와 관련한 요소.
관계	실행하는 동안의 소프트웨어 요소가 놓여지는 물리적 단위.
제약사항	할당 토폴로지에 대한 제한은 없다. 소프트웨어 요구 속성은 하드웨어 제공 속성에 의해 충족된다.

3.3.4.1.2 Element Catalog

3.3.4.1.2.1 Elements

Element	Description
ViewerComponent	사용자의 입력에 따라 화면을 보여준다. 제공되는 interface: DVMDisplayViewer

3.3.4.1.2.2 Relations

앞서 표현된 모듈뷰(Layered View)와 C&C View(Server-Client Style)의 소프트웨어 컴포넌트와 커넥터와 소프트웨어가 실행되는 컴퓨팅 플랫폼의 하드웨어 사이의 매핑을 서술한다. 소프트웨어 컴포넌트는 3.1 과 3.2 에 서술되어 있다.

3.3.4.1.2.3 Interfaces

Interface	Description
DVMDisplayViewer	사용자와 터치방식으로 직접적인 interaction 을 통해 데이터 수신.

	데이터 교류에 필요한 모든 DVM Display 화면 생성.

3.3.4.1.2.4 Constraints

- 최소 1000 명의 동시 사용자를 지원해야 한다.

4 Relations Among Views

4.1 General Relations Among Views

Module View(Layered View)를 통해 시스템의 구조를 파악한다.
C&C View(Server-Client Style)를 통해 DVM과 데이터베이스 서버와의 통신을 정의한다.
Allocation View(Deployment Sytle 를 통해 SW 와 HW의 매핑을 정의한다.

4.2 View-to-View Relations

Module View(Layered View)를 통해 DVM System 의 요소와 소프트웨어 모듈을 정의하고, C&C View(Server-Client Style)를 통해 Module View 에서 정의한 모듈 사이(서버-클라이언트간)의 통신을 정의하고, Allocation View(Deployment Style)를 통해 Modue View 와 C&C View 에서 정의된 모듈, 통신의 하드웨어로 매핑하는 관계이다.

5 Referenced Materials

IEEE 1471	ANSI/IEEE-1471-2000, *IEEE Recommended Practice for Architectural Description of Software-Intensive Systems*, 21 September 2000.
Clements 2010	Clements, Bachmann, Bass, Garlan, Ivers, Little, Merson, Nord, Stafford, *Documenting Software Architectures: Views and Beyond 2nd Edition*, Addison Wesley Professional, 2010.