

System and Software Architecture Description (SSAD)

SnapValet

Team 03

Name	Role
Brian Vanover	Project Manager, Feasibility Analyst, Life Cycle Planner.
Abhinandan Patni	Operations Concept Engineer, Prototyper/Builder.
Xiaoting Bi	Feasibility Analyst, Prototyper/Builder.
Molly Karcher	IIV & V, Quality Focal Point, Prototyper/Builder.
Ditong Dong	System Architect, Prototyper/Builder.
Ridhima Manjrekar	Requirements Engineer, Prototyper/Builder.
Saikarthik Desiraju	Life Cycle Planner, Prototyper/Builder.

10/11/2014

Version History

Date	Author	Version	Changes made	Rationale
10/11/2014	Ditong Ding	1.0	<ul style="list-style-type: none">• Original for CSCI577 project SnapValet	<ul style="list-style-type: none">• Initial draft for use with SnapValet v1.0

Table of Contents

System and Software Architecture Description (SSAD)	i
Version History	ii
Table of Contents	iii
Table of Tables	iv
Table of Figures	vi
1. Introduction	1
1.1 Purpose of the SSAD	1
1.2 Status of the SSAD	1
2. System Analysis	2
2.1 System Analysis Overview	2
2.2 System Analysis Rationale	10
3. Technology-Independent Model	11
3.1 Design Overview	11
3.2 Design Rationale	13
4. Technology-Specific System Design	14
4.1 Design Overview	14
4.2 Design Rationale	15
5. Architectural Styles, Patterns and Frameworks	16

Table of Tables

<i>Table 1: Actors Summary.....</i>	<i>2</i>
<i>Table 2: Artifacts and Information Summary</i>	<i>3</i>
<i>Table 3: Process Description (Valet Operator Check in)</i>	<i>4</i>
<i>Table 4: Typical Course of Action (Valet Operator Check in).....</i>	<i>4</i>
<i>Table 5: Alternate Course of Action (Valet Operator Check in).....</i>	<i>5</i>
<i>Table 6: Process Description (Valet Process).....</i>	<i>5</i>
<i>Table 7: Typical Course of Action (Valet Process)</i>	<i>5</i>
<i>Table 8: Alternate Course of Action (Valet Process)</i>	<i>5</i>
<i>Table 9: Exceptional Course of Action (Valet Process)</i>	<i>6</i>
<i>Table 10: Exceptional Course of Action (Valet Process)</i>	<i>6</i>
<i>Table 11: Exceptional Course of Action (Valet Process)</i>	<i>6</i>
<i>Table 12: Process Description (Log in).....</i>	<i>7</i>
<i>Table 13: Typical Course of Action (Log in).....</i>	<i>7</i>
<i>Table 14: Exceptional Course of Action (Log in).....</i>	<i>7</i>
<i>Table 15: Process Description (Register).....</i>	<i>7</i>
<i>Table 16: Typical Course of Action (Register).....</i>	<i>8</i>
<i>Table 17: Exceptional Course of Action (Register).....</i>	<i>8</i>
<i>Table 18: Process Description (Hire Valet Operator)</i>	<i>8</i>
<i>Table 19: Typical Course of Action (Hire Valet Operator).....</i>	<i>8</i>
<i>Table 20: Process Description (Fire Valet Operator).....</i>	<i>9</i>
<i>Table 21: Typical Course of Action (Fire Valet Operator)</i>	<i>9</i>
<i>Table 22: Process Description (Return Tip).....</i>	<i>9</i>
<i>Table 23: Typical Course of Action (Return Tip).....</i>	<i>9</i>
<i>Table 7: Hardware Component Description</i>	<i>11</i>
<i>Table 8: Software Component Description.....</i>	<i>12</i>
<i>Table 9: Supporting Software Component Description.....</i>	<i>12</i>
<i>Table 10: Design Class Description</i>	<i>12</i>
<i>Table 11: Hardware Component Description</i>	<i>14</i>
<i>Table 12: Software Component Description.....</i>	<i>14</i>

<i>Table 13: Supporting Software Component Description.....</i>	<i>15</i>
<i>Table 14: Design Class Description.....</i>	<i>15</i>
<i>Table 15: Architectural Styles, Patterns, and Frameworks.....</i>	<i>16</i>

Table of Figures

<i>Figure 1: System Context Diagram</i>	<i>2</i>
<i>Figure 2: Artifacts and Information Diagram.....</i>	<i>3</i>
<i>Figure 3: Process Diagram</i>	<i>4</i>
<i>Figure 4: Hardware Component Class Diagram.....</i>	<i>11</i>
<i>Figure 5: Software Component Class Diagram</i>	<i>11</i>
<i>Figure 6: Deployment Diagram.....</i>	<i>11</i>
<i>Figure 7: Supporting Software Component Class Diagram.....</i>	<i>11</i>
<i>Figure 8: Design Class Diagram.....</i>	<i>12</i>
<i>Figure 9: Process Realization Diagram.....</i>	<i>13</i>
<i>Figure 10: Hardware Component Class Diagram.....</i>	<i>14</i>
<i>Figure 11: Software Component Class Diagram</i>	<i>14</i>
<i>Figure 12: Deployment Diagram.....</i>	<i>14</i>
<i>Figure 13: Supporting Software Component Class Diagram.....</i>	<i>14</i>
<i>Figure 14: Design Class Diagram.....</i>	<i>15</i>
<i>Figure 15: Process Realization Diagram.....</i>	<i>15</i>

1. Introduction

1.1 Purpose of the SSAD

The purpose of the SSAD is to record the results of SnapValet system analysis and design. This document is used by developers as reference to the system architecture, and the development of application must follow the statement in the SSAD. Furthermore, the SSAD is used to help the maintainer and clients to understand the architecture of the system once the application is delivered.

1.2 Status of the SSAD

The status of the SSAD is currently at the Foundations phase version number 1.0. This is the first version of this document.

2. System Analysis

2.1 System Analysis Overview

The primary purpose of SnapValet is to improve the experience of valet service. By providing cashless payment, car request feature, SnapValet will highly improve the speed of valet process, give drivers more choice when they want to use valet service. For valet operator and valet company, the system also provides some features in employee management and tip management, for supporting the cashless valet service.

2.1.1 System Context

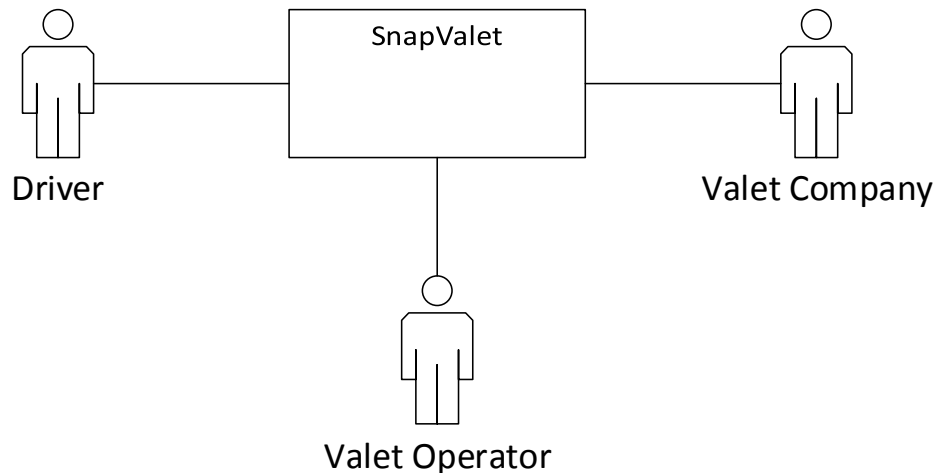


Figure 1: System Context Diagram

Table 1: Actors Summary

Actor	Description	Responsibilities
Driver	People who use valet service and use SnapValet app to notify or pay valet service	<ul style="list-style-type: none"> ● Request for valet service ● Request to retrieve car ● Pay valet service fee and tip
Valet Operator	People who provide valet service	<ul style="list-style-type: none"> ● Provide valet service ● Manage keys and cars ● Charge valet service fee
Valet Company	Entity which in charge of valet operators	<ul style="list-style-type: none"> ● Manage valet service fee and salary ● Manage employee

2.1.2 Artifacts & Information

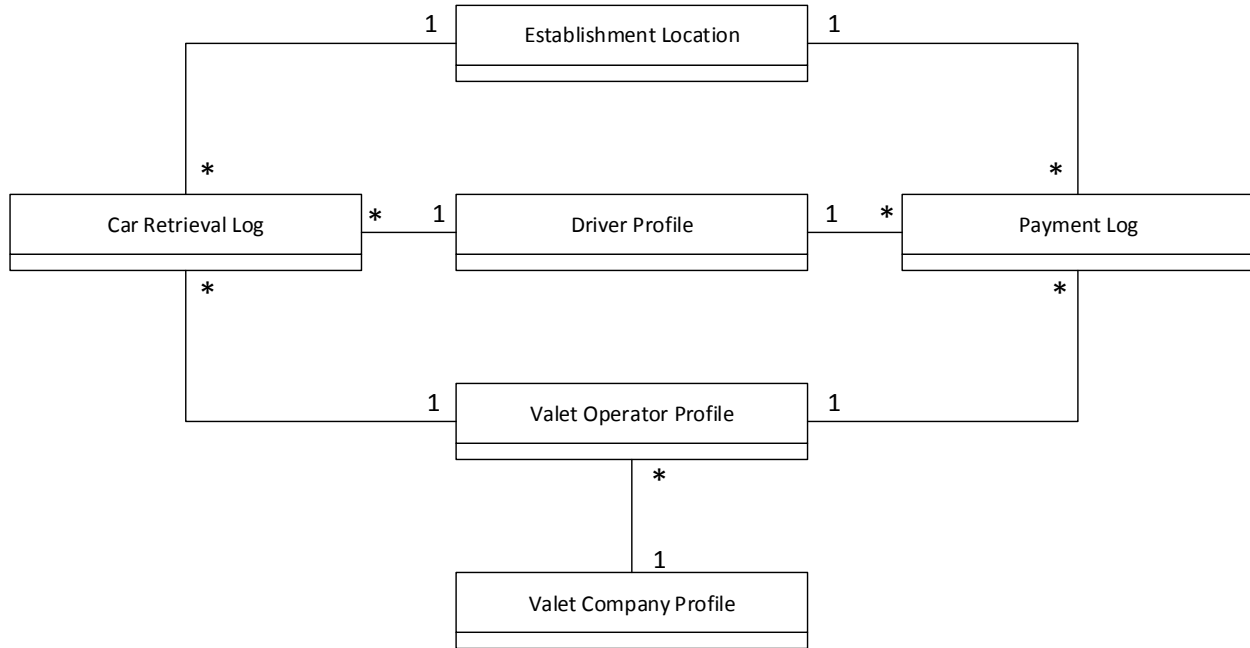


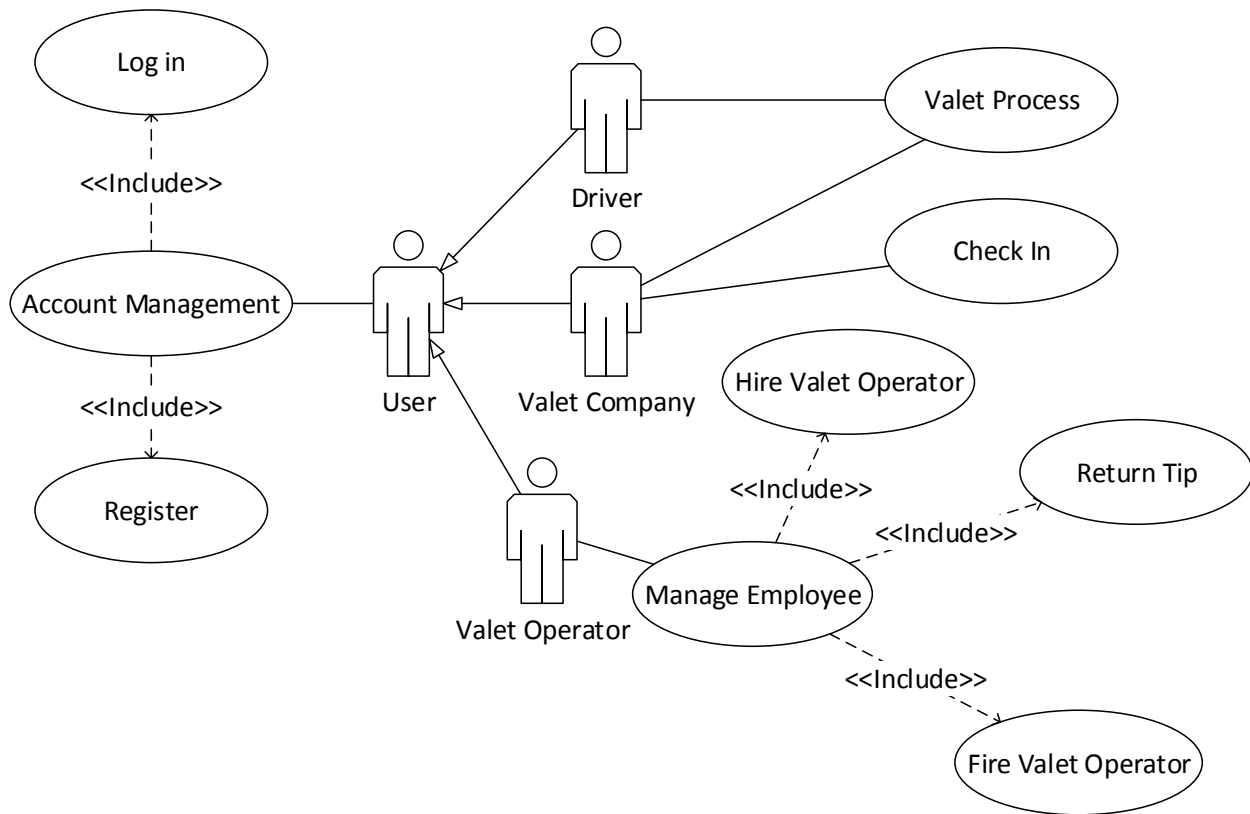
Figure 2: Artifacts and Information Diagram

Table 2: Artifacts and Information Summary

Artifact	Purpose
Valet Company Profile	Contains information of a valet company for cashless valet service.
Valet Operator Profile	Contains information of a valet operator for cashless valet service.
Driver Profile	Contains information of a driver for valet service.
Car Retrieval Log	Contains information of a car retrieval request from driver.
Payment Log	Contains information of a valet service fee and tip
Establishment Location	Contains information of the establishments which provide SnapValet service.

2.1.3 Behavior

Figure 3 illustrates the process diagram of SnapValet. It can be divided into three capabilities: valet service, employee management and account management.



S

Figure 3: Process Diagram

2.1.3.1 Valet Service

2.1.3.1.1 Valet Operator Check in

Table 3: Process Description (Valet Operator Check in)

Identifier	UC-1: Valet operator check in
Purpose	For valet operator to log in the public device in the establishment he works for.
Requirements	WC_3392, WC_3390
Development Risks	Account security problem
Pre-conditions	Valet operator has logged in.
Post-conditions	Valet operator checks in location.

Table 4: Typical Course of Action (Valet Operator Check in)

Seq#	Actor's Action	System's Response
1	Valet operator selects the location s/he working for.	Keep session in the public tablet for receiving notification in this

		establishment.
--	--	----------------

Table 5: Alternate Course of Action (Valet Operator Check in)

Seq#	Actor's Action	System's Response
1	Valet operator adds himself/herself into the operators list.	Keep session in the public tablet for receiving notification in this establishment.

2.1.3.1.2 Valet Process

Table 6: Process Description (Valet Process)

Identifier	UC-2: Valet Process
Purpose	For the whole process of valet service
Requirements	WC_3392, WC_3390
Development Risks	None
Pre-conditions	At least one valet operator has checked in this location via public tablet, driver has logged in.
Post-conditions	Driver get car and leave.

Table 7: Typical Course of Action (Valet Process)

Seq#	Actor's Action	System's Response
1	Driver requests for valet service and gives car and key to valet operator.	
2	Valet operator gives a ticket to driver, parks driver's car and manage key and car.	
3	Driver checks in location and request car, pays service fee and tip with application.	Send request to the public tablet to notify valet operator to retrieve car.
4	Valet operator gets notification from public tablet and assigns task to some operator.	Send notification to driver to notify him/her that the car is in retrieval.
5	Valet operator retrieves car and sends notification to driver.	Send notification to driver to notify him/her that the car is ready.
6	Driver gets his/her car and leaves	

Table 8: Alternate Course of Action (Valet Process)

Seq#	Actor's Action	System's Response
1	Driver requests for valet service	

	and gives car and key to valet operator.	
2	Valet operator gives a ticket to driver, parks driver's car and manage key and car.	
3	Driver checks in location and requests car without payment.	Send request to the public tablet to notify valet operator to retrieve car.
4	Valet operator gets notification from public tablet and assigns task to some operator.	Send notification to driver to notify him/her that the car is in retrieval.
5	Valet operator retrieves car and sends notification to driver.	Send notification to driver to notify him/her that the car is ready.
6	Driver gets his/her car, pay with cash and leave	

Table 9: Exceptional Course of Action (Valet Process)

Seq#	Actor's Action	System's Response
1	Driver requests for valet service but valet operator refuses the request as parking lot has been full.	

Table 10: Exceptional Course of Action (Valet Process)

Seq#	Actor's Action	System's Response
1	Driver requests for valet service and gives car and key to valet operator.	
2	Valet operator gives a ticket to driver, parks driver's car and manage key and car.	
3	Driver checks in wrong location and requests car.	Send request to the public tablet to notify valet operator to retrieve car.
4	Valet operator finds the mistake and sends notification to driver.	Send notification to driver to notify him/her about the mistake.

Table 11: Exceptional Course of Action (Valet Process)

Seq#	Actor's Action	System's Response
1	Driver requests for valet service and gives car and key to valet operator.	
2	Valet operator gives a ticket to driver, parks driver's car and manage key and car.	

3	Driver checks in location and requests car with/without payment.	Send request to the public tablet to notify valet operator to retrieve car.
4	Valet operator gets notification from public tablet and assigns task to some operator.	Send notification to driver to notify him/her that the car is in retrieval.
5	Valet operator retrieves car and sends notification to driver.	Send notification to driver to notify him/her that the car is ready.
6	As driver hasn't come after a period of time, valet operator re-parks car and sends notification to driver	Send notification to driver to notify him/her that the car has been re-parked.

2.1.3.2 Account Management

2.1.3.2.1 Log in

Table 12: Process Description (Log in)

Identifier	UC-3: Log in
Purpose	To check if a user has right to access system
Requirements	
Development Risks	Account security problem
Pre-conditions	SnapValet server has successfully started.
Post-conditions	User logs in system.

Table 13: Typical Course of Action (Log in)

Seq#	Actor's Action	System's Response
1	User provide username and password	The username and password is valid

Table 14: Exceptional Course of Action (Log in)

Seq#	Actor's Action	System's Response
1	User provide username and password	The username and password is invalid

2.1.3.2.2 Register

Table 15: Process Description (Register)

Identifier	UC-4: Register
-------------------	----------------

Purpose	For users to get access to system.
Requirements	
Development Risks	Account security problem
Pre-conditions	SnapValet server has successfully started.
Post-conditions	User gets an account.

Table 16: Typical Course of Action (Register)

Seq#	Actor's Action	System's Response
1	User provides personal information with username and password.	Add user information into system.

Table 17: Exceptional Course of Action (Register)

Seq#	Actor's Action	System's Response
1	User provide information with username and password.	As username duplicate/password does not meet constraint, refuse to add user information.

2.1.3.3 Employee Management

2.1.3.3.1 Hire Valet Operator

Table 18: Process Description (Hire Valet Operator)

Identifier	UC-5: Hire valet operator
Purpose	Add valet operator information linked to valet company
Requirements	
Development Risks	Account security problem
Pre-conditions	Valet company has logged in system.
Post-conditions	A new valet operator added into system.

Table 19: Typical Course of Action (Hire Valet Operator)

Seq#	Actor's Action	System's Response
1	Valet company adds a new valet operator into system with employee information.	Return a new generated employee ID.

2.1.3.3.2 Fire Valet Operator

Table 20: Process Description (Fire Valet Operator)

Identifier	UC-6: Fire valet operator
Purpose	Delete valet operator information linked to valet company
Requirements	
Development Risks	Account security problem
Pre-conditions	Valet company has logged in system.
Post-conditions	A previous valet operator deleted from system.

Table 21: Typical Course of Action (Fire Valet Operator)

Seq#	Actor's Action	System's Response
1	Valet company deletes a previous valet operator from system and confirm operation.	Return successfully delete notification to user.

2.1.3.3.3 Return Tip

Table 22: Process Description (Return Tip)

Identifier	UC-7: Return tip
Purpose	Return tip to specific valet operator.
Requirements	WC_3208
Development Risks	Account security problem
Pre-conditions	Valet company has logged in system.
Post-conditions	Valet company get a list of tip amount for employee.

Table 23: Typical Course of Action (Return Tip)

Seq#	Actor's Action	System's Response
1	Valet company queries for tips-valet operator list.	Search payment log and return tips-valet operator list.
2	Valet company returns tips with information in list.	

2.1.4 Modes of Operation

The SnapValet, as we envision implementing it, will operate in only one mode, so nothing further need be said of modes of operation.

2.2 System Analysis Rationale

Based on our analysis of how the users interact with the system, we have identified 3 classes of operational stakeholders.

Driver: They are the client of valet service users, and their responsibility for this system is to use request system and cashless payment to improve their valet experience.

Valet operator: They are the performer of valet service. Their responsibility for this system is to perform valet service, and receive salary and tips as return.

Valet Company: They are supervise of valet operator. Their responsibility is to manage the valet operators whether hired or fired, and to pay salary to them.

3. Technology-Independent Model

3.1 Design Overview

3.1.1 System Structure

<< This section should contain

- a UML hardware component class diagram
- a UML software component class diagram
- a UML deployment diagram
- If necessary, a class diagram for the system's supporting software infrastructure
- and descriptions of the hardware components, software components, and, if necessary, the supporting software infrastructure components of the technology/platform-independent system architecture

More information and example can be found in **ICM EPG> Task: Define Technology-Independent Architecture >>**

<<Hardware Component Class Diagram>>

Figure 4: Hardware Component Class Diagram

<<Software Component Class Diagram>>

Figure 5: Software Component Class Diagram

<<Deployment Diagram>>

Figure 6: Deployment Diagram

<<Optional: Supporting Software Infrastructure Diagram>>

Figure 7: Supporting Software Component Class Diagram

Table 24: Hardware Component Description

Hardware Component	Description
--------------------	-------------

Table 25: Software Component Description

Software Component	Description

Table 26: Supporting Software Component Description

Support Software Component	Description

3.1.2 Design Classes

This section should contain:

- UML class diagrams showing all the boundary, entity, and control classes in the design of the system being developed
- and a description of each class in the diagram

More information and example can be found in **ICM EPG> Task: Define Technology-Independent Architecture >>**

3.1.2.1 <Classes n>

<<Design Classes Class Diagram>>

Figure 8: Design Class Diagram**Table 27: Design Class Description**

Class	Type	Description

--	--	--

3.1.3 Process Realization

<< This section shows how the proposed architecture can be realized by constructing sequence diagrams. More information and example can be found in **ICM EPG> Task: Define Technology-Independent Architecture >>**

<<Process Realization Diagram>>

Figure 9: Process Realization Diagram

3.2 Design Rationale

<< This section should contain an explanation of how/why the architecture/design described in previous sections was chosen. More information and example can be found in **ICM EPG> Task: Define Technology-Independent Architecture >>**

4. Technology-Specific System Design

<< Once you know specific technology that you team is going to use, design the system and software architecture and document them in this section. >>

4.1 Design Overview

4.1.1 System Structure

<<Hardware Component Class Diagram>>

Figure 10: Hardware Component Class Diagram

<<Software Component Class Diagram>>

Figure 11: Software Component Class Diagram

<<Deployment Diagram>>

Figure 12: Deployment Diagram

<<Optional: Supporting Software Infrastructure Diagram>>

Figure 13: Supporting Software Component Class Diagram

Table 28: Hardware Component Description

Hardware Component	Description

Table 29: Software Component Description

Software Component	Description

Table 30: Supporting Software Component Description

Support Software Component	Description

4.1.2 Design Classes

4.1.2.1 <Classes n>

<<Design Classes Class Diagram>>

Figure 14: Design Class Diagram**Table 31: Design Class Description**

Class	Type	Description

4.1.3 Process Realization

<<Process Realization Diagram>>

Figure 15: Process Realization Diagram

4.2 Design Rationale

5. Architectural Styles, Patterns and Frameworks

<< Describe any implementation architecture styles (e.g. the Prism style and 3-tier architecture), patterns (e.g. pipe-and-filter and client-server), or frameworks (e.g. Java and CORBA) used to describe the system architecture. >>

Table 32: Architectural Styles, Patterns, and Frameworks

Name	Description	Benefits, Costs, and Limitations