Software Architecture Documentation

Paulo Gandra de Sousa

2010.08.03

pagsousa@gmail.com http://twitter.com/pagsousa http://linkedin.com/in/pagsousa

Goals

1. Present common views and SAD
2. Define stakeholders
3. Identify stakeholders’s concerns
4. Define what and how to document

Scope

IN

- Initial description of architecture

OUT

- Detailed architecture
- Data model

Introduction

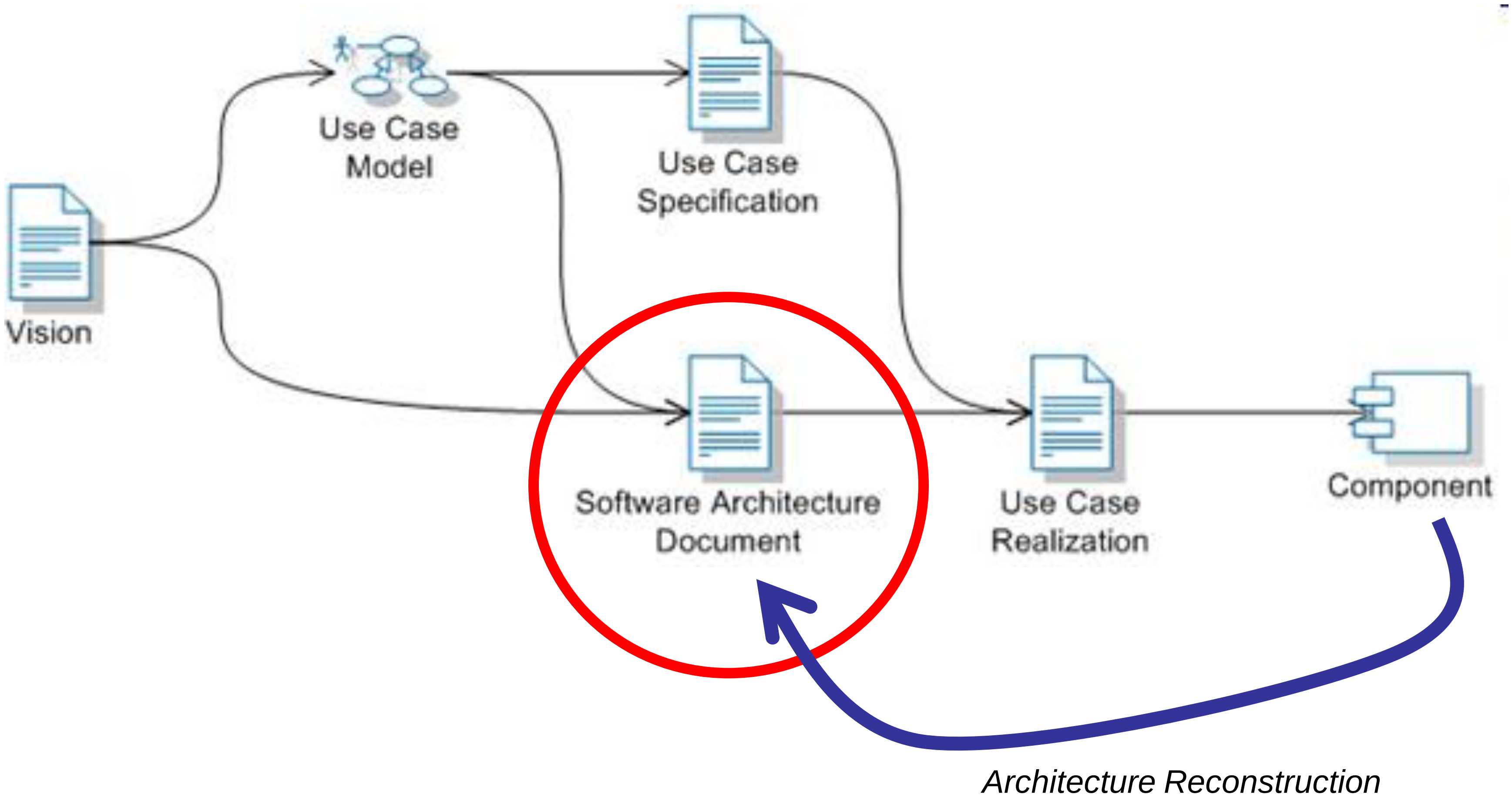
The software architecture of a program or computing system is the **structure** or structures of the system, which comprise software elements, the **externally visible properties** of those elements, and the **relationships** among them.

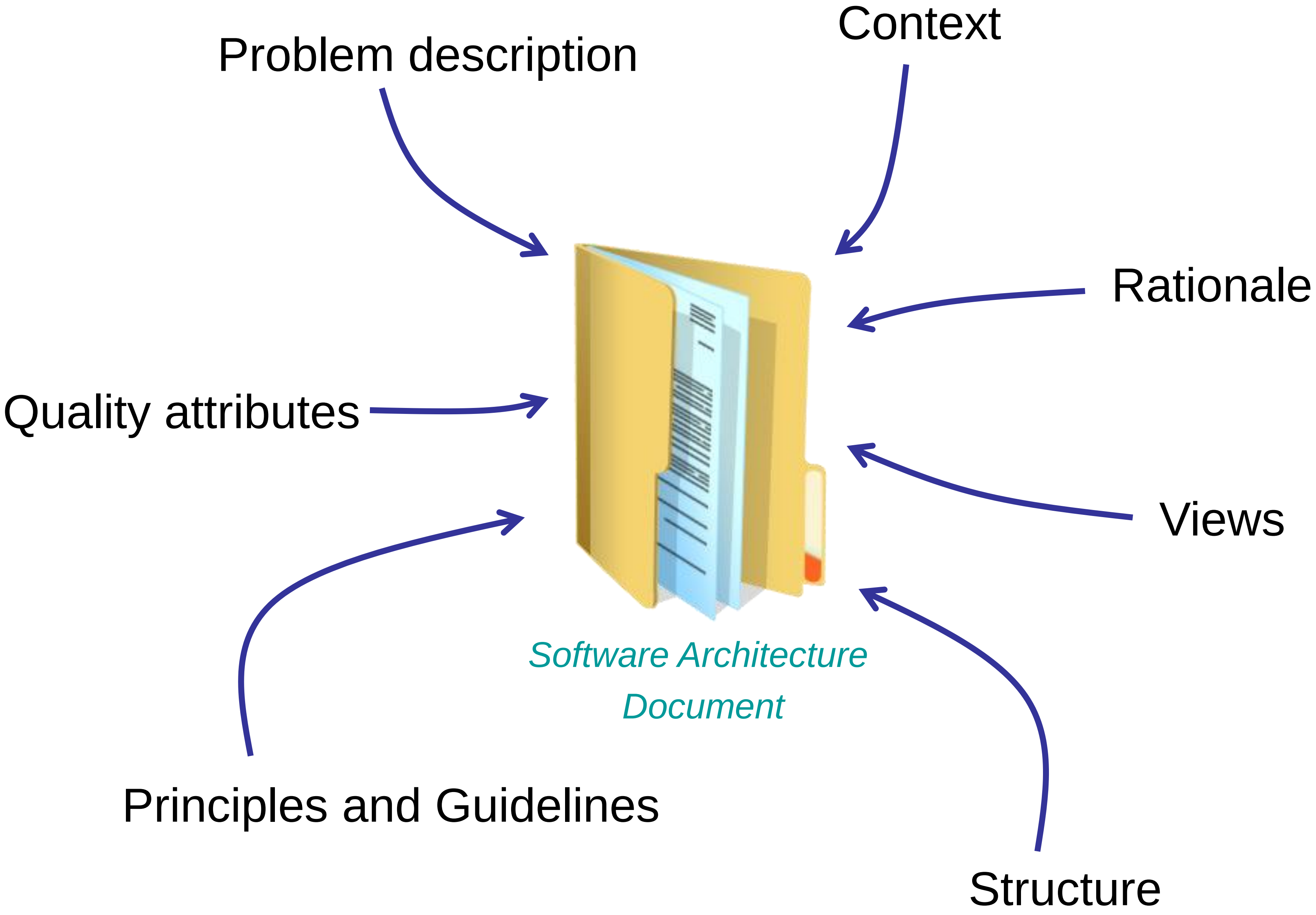
Bass, Clements, Kazman (2003)

[Software architecture is] The fundamental **organization** of a system embodied in its components, their **relationships** to each other, and to the environment, and the **principles** guiding its design and evolution.

ANSI/IEEE 1471 (2000)

Architecture in the software development process





Problem description

Context

Rationale

Quality attributes

Views

Software Architecture Document

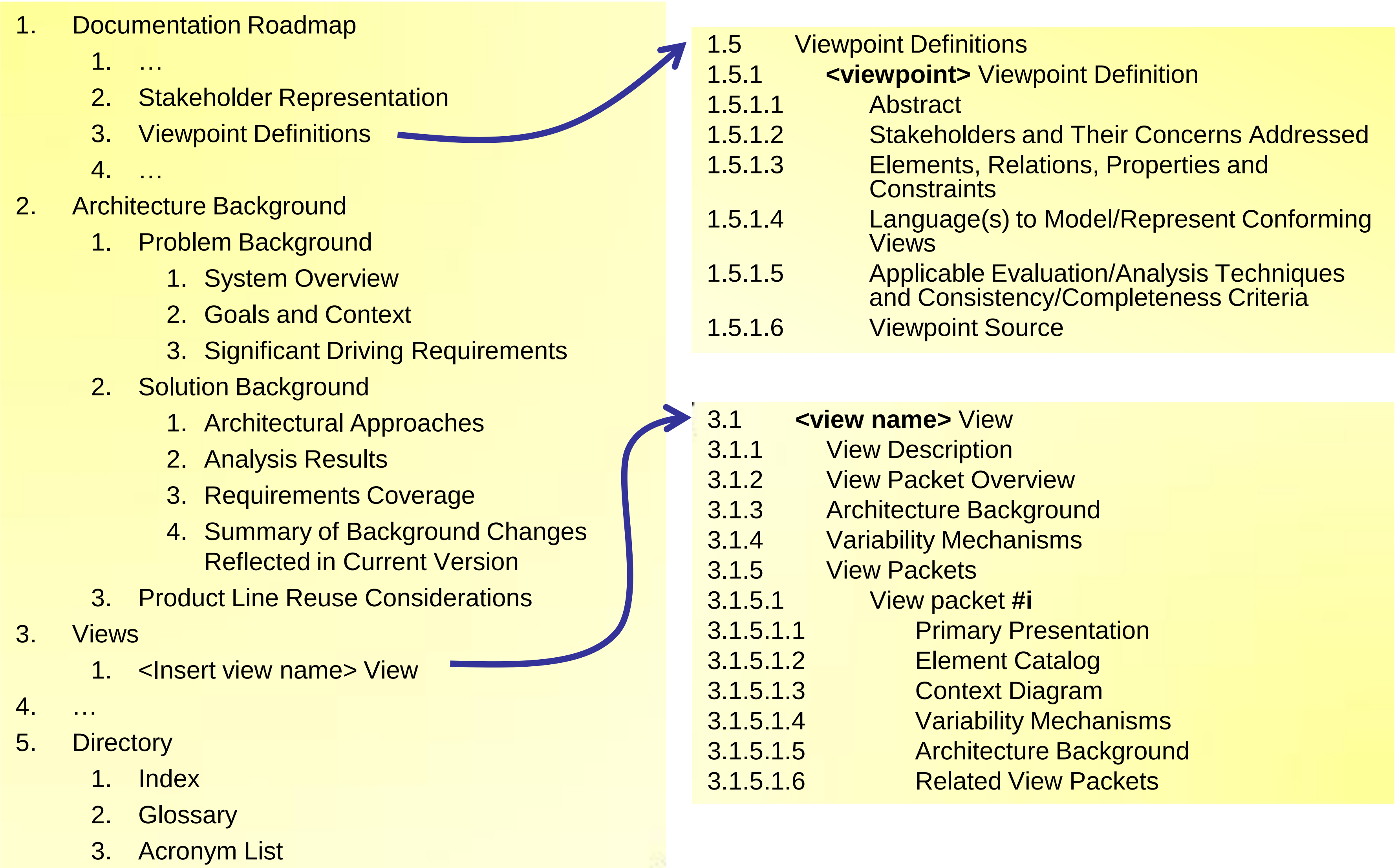
Principles and Guidelines

Structure

RUP’s SAD

1. Introduction
   1. Purpose
   2. Scope
   3. Definitions, Acronyms, and Abbreviations
   4. References
   5. Overview
2. Architectural Representation
3. Architectural Goals and Constraints
4. Use-Case View
   1. Use-Case Realizations
5. Logical View
   1. Overview
   2. Architecturally Significant Design Packages
6. Process View
7. Deployment View
8. Implementation View
   1. Overview
   2. Layers
9. Data View (optional)
10. Size and Performance
11. Quality

SEI's SAD



1. Documentation Roadmap
    1. …
    2. Stakeholder Representation
    3. Viewpoint Definitions
    4. …
2. Architecture Background
    1. Problem Background
        1. System Overview
        2. Goals and Context
        3. Significant Driving Requirements
    2. Solution Background
        1. Architectural Approaches
        2. Analysis Results
        3. Requirements Coverage
        4. Summary of Background Changes Reflected in Current Version
    3. Product Line Reuse Considerations
3. Views
    1. <Insert view name> View
4. …
5. Directory
    1. Index
    2. Glossary
    3. Acronym List

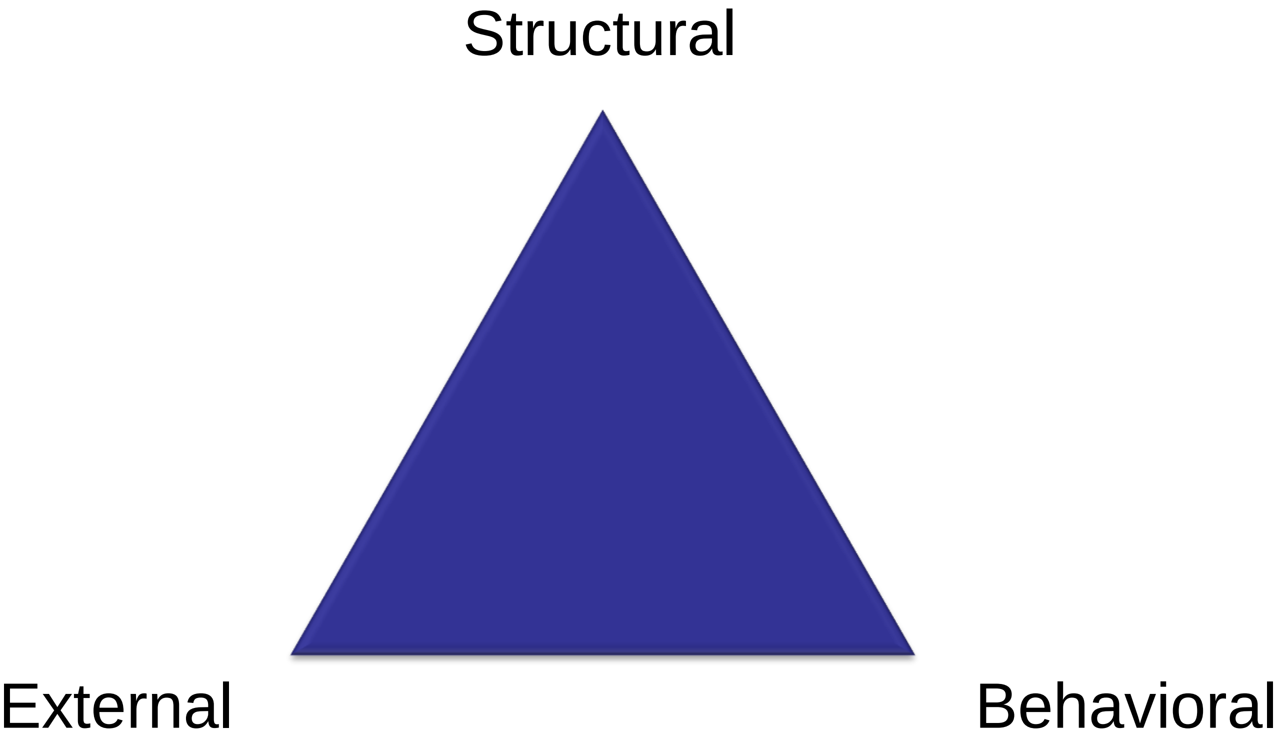
1.5 Viewpoint Definitions
1.5.1 **<viewpoint>** Viewpoint Definition
1.5.1.1 Abstract
1.5.1.2 Stakeholders and Their Concerns Addressed
1.5.1.3 Elements, Relations, Properties and Constraints
1.5.1.4 Language(s) to Model/Represent Conforming Views
1.5.1.5 Applicable Evaluation/Analysis Techniques and Consistency/Completeness Criteria
1.5.1.6 Viewpoint Source

3.1 **<view name>** View
3.1.1 View Description
3.1.2 View Packet Overview
3.1.3 Architecture Background
3.1.4 Variability Mechanisms
3.1.5 View Packets
3.1.5.1 View packet **#i**
3.1.5.1.1 Primary Presentation
3.1.5.1.2 Element Catalog
3.1.5.1.3 Context Diagram
3.1.5.1.4 Variability Mechanisms
3.1.5.1.5 Architecture Background
3.1.5.1.6 Related View Packets

Less is More*

** SAD should have the minimum information necessary to understand the system but no more than that*

Different kinds of views



Structural

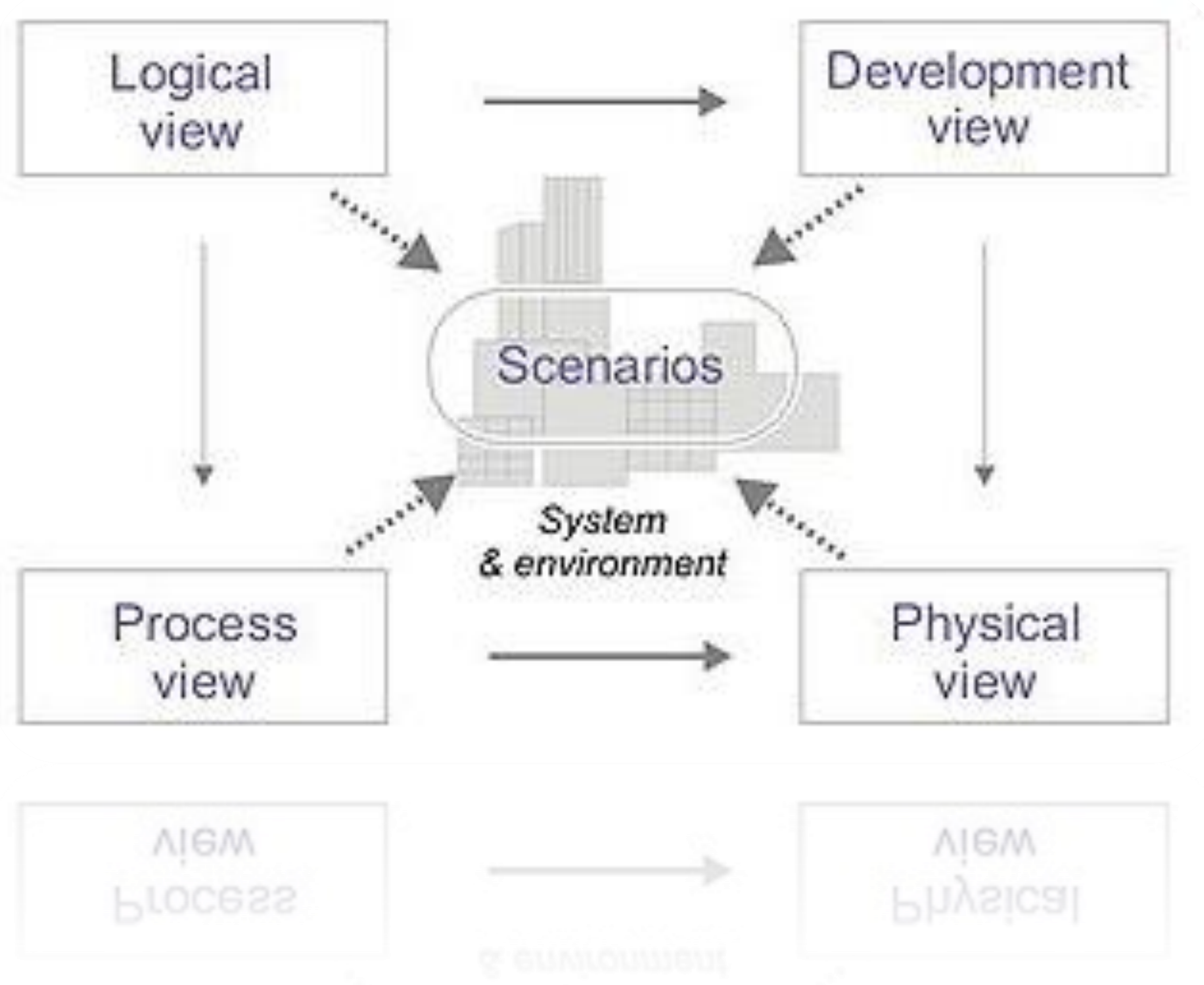
External

Behavioral

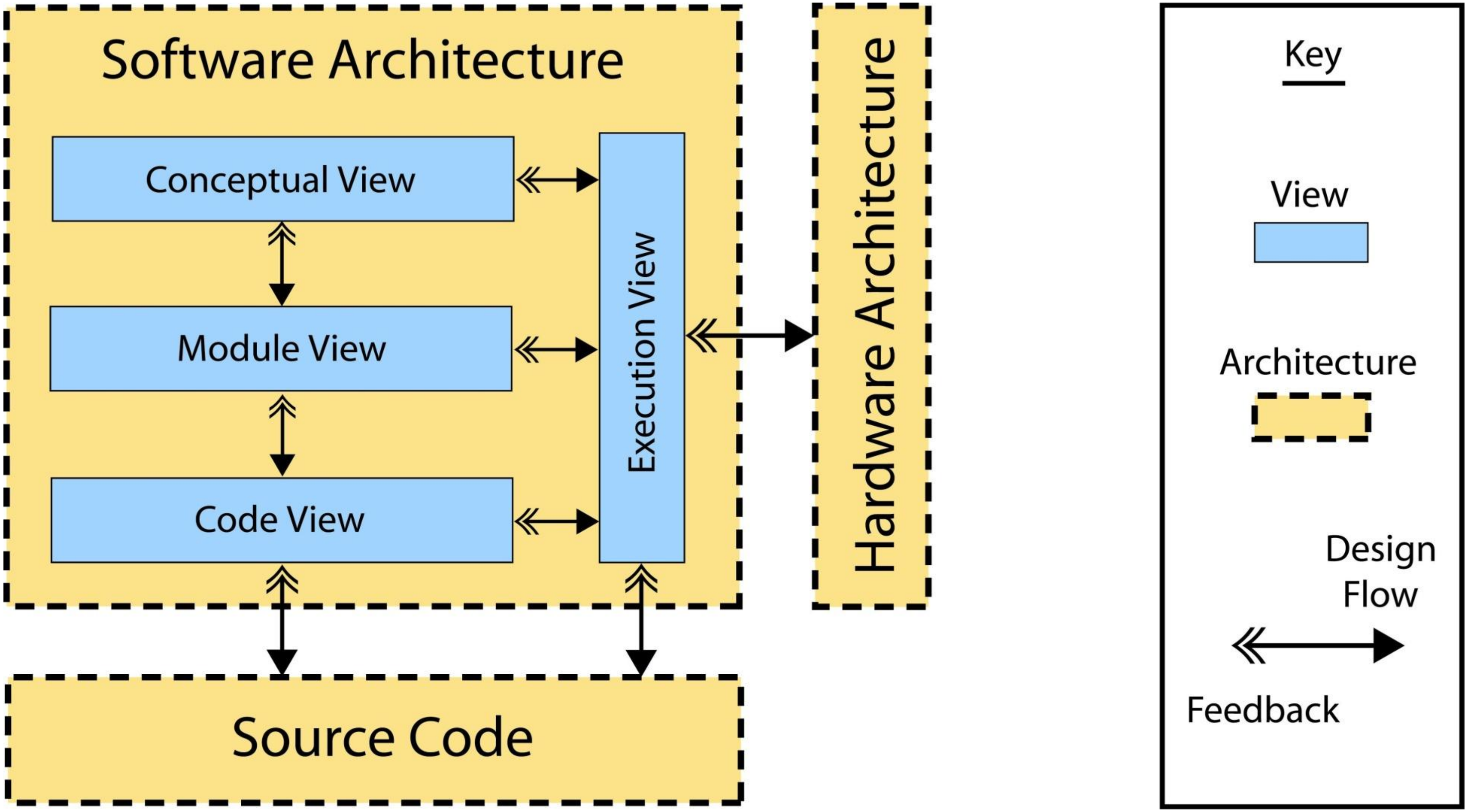
Different viewsets

- RUP 4+1
- Siemens
- Rozansky & Woods’ views and perspectives
- SEI’s views and beyond
- …

RUP 4+1

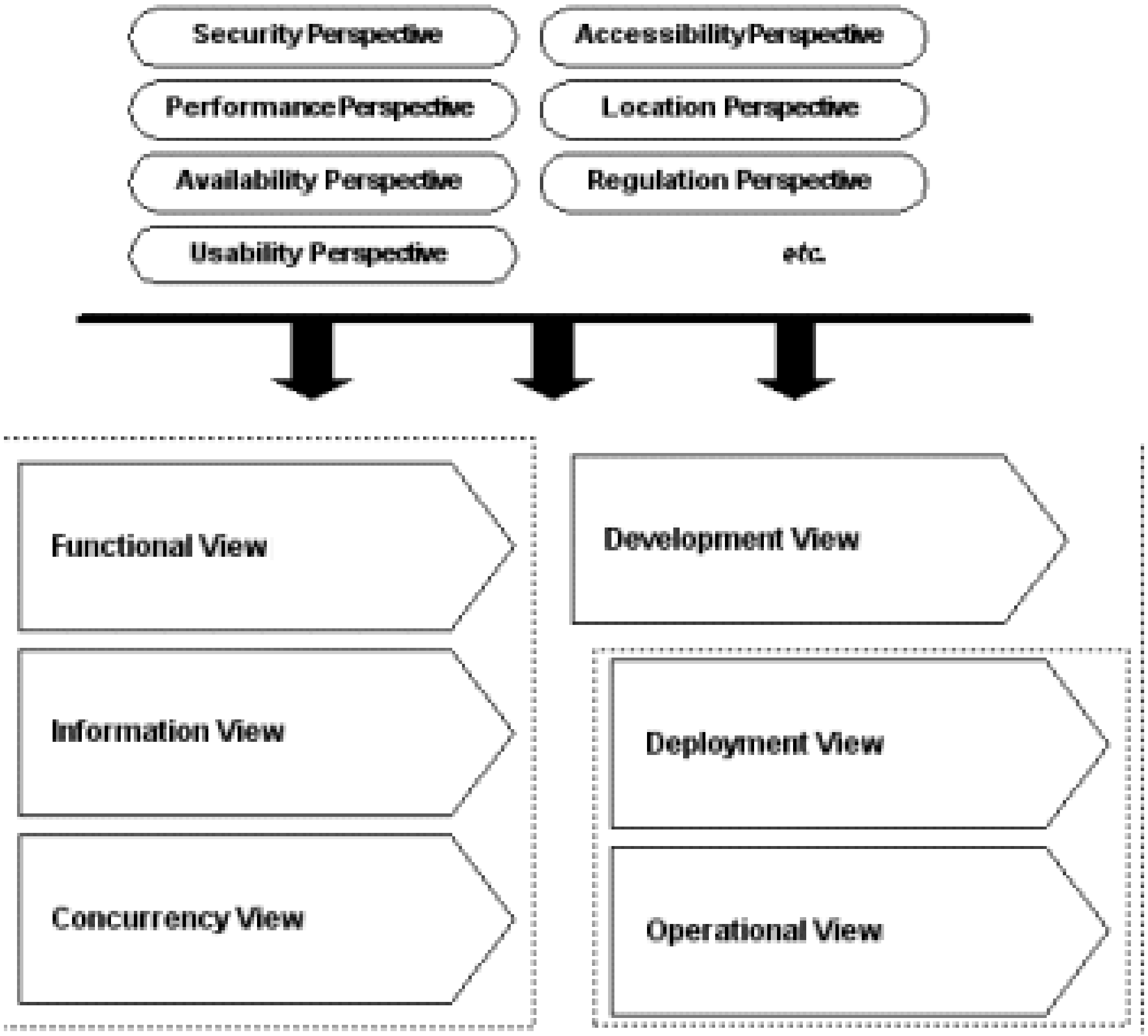

Logical view
Development view
Scenarios
System & environment
Process view
Physical view

Siemens

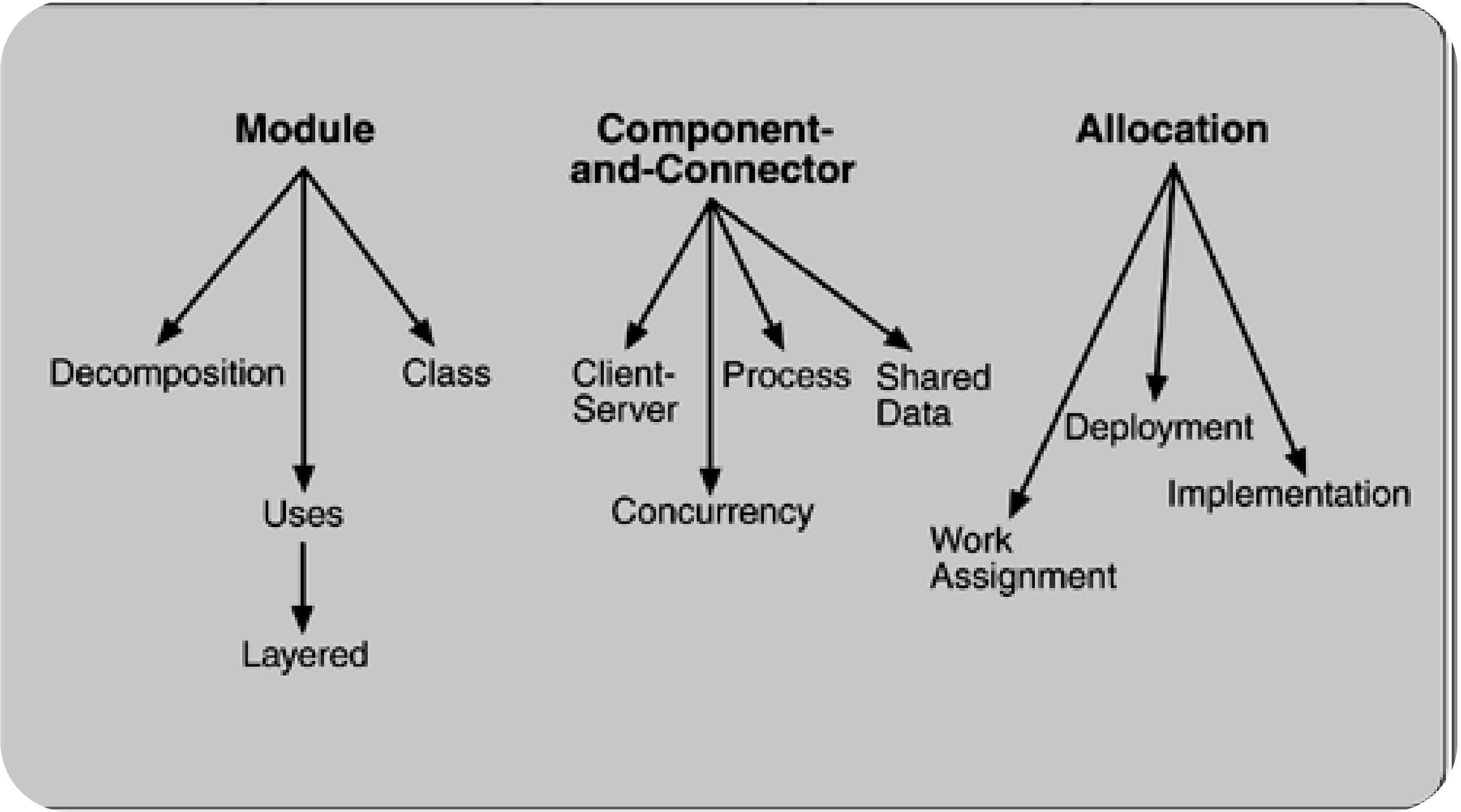


Adapted from “Applied Software Architecture”, Hofmeister, C. et al. (2000).

Rozansky & Woods



SEI's Views and Beyond



Module

Decomposition

Uses

Layered

Class

Component-and-Connector

Client-Server

Concurrency

Process

Shared Data

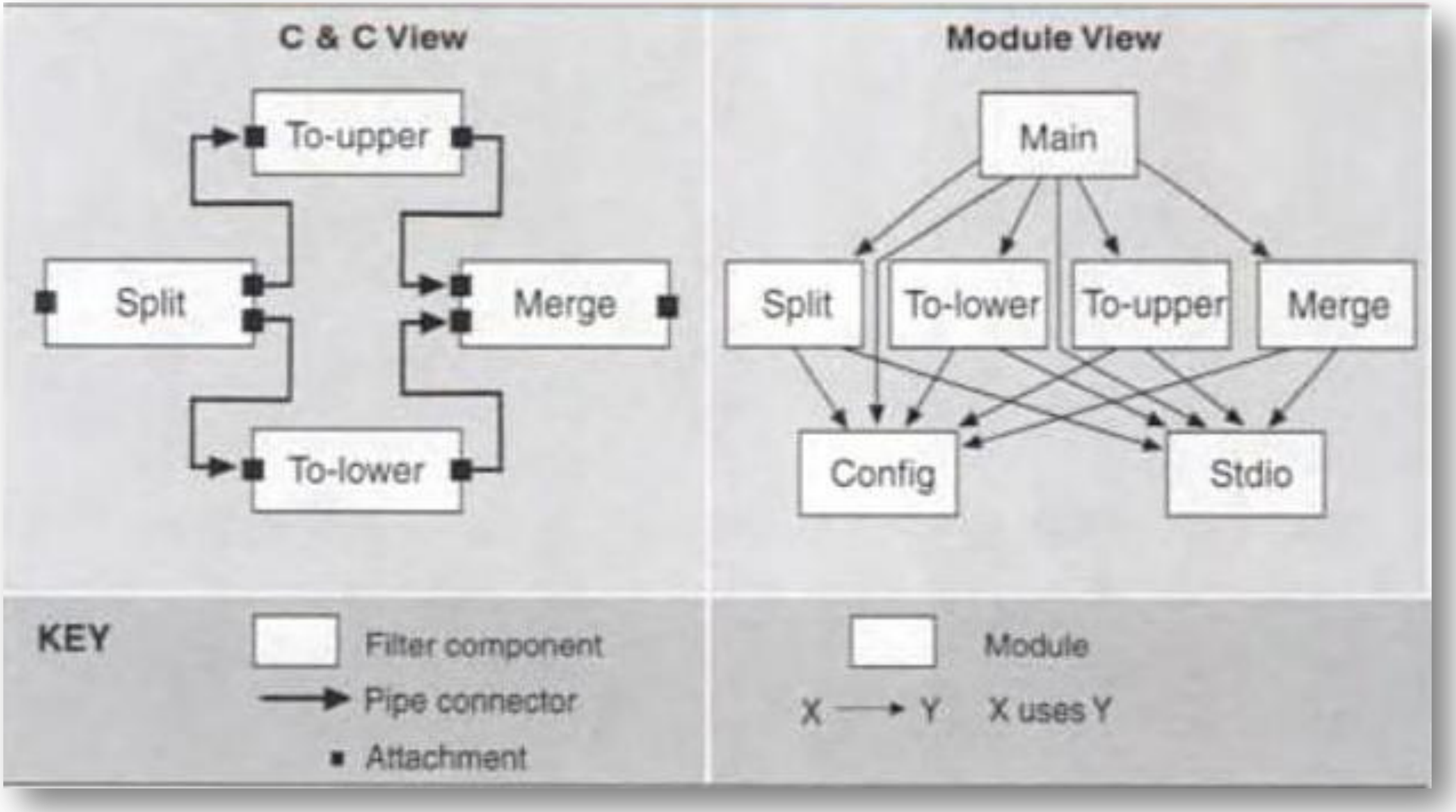
Allocation

Work Assignment

Deployment

Implementation

Module vs. C&C


C & C View
To-upper
Split
Merge
To-lower
Module View
Main
Split
To-lower
To-upper
Merge
Config
Stdio
KEY
Filter component
Pipe connector
Attachment
Module
X → Y
X uses Y

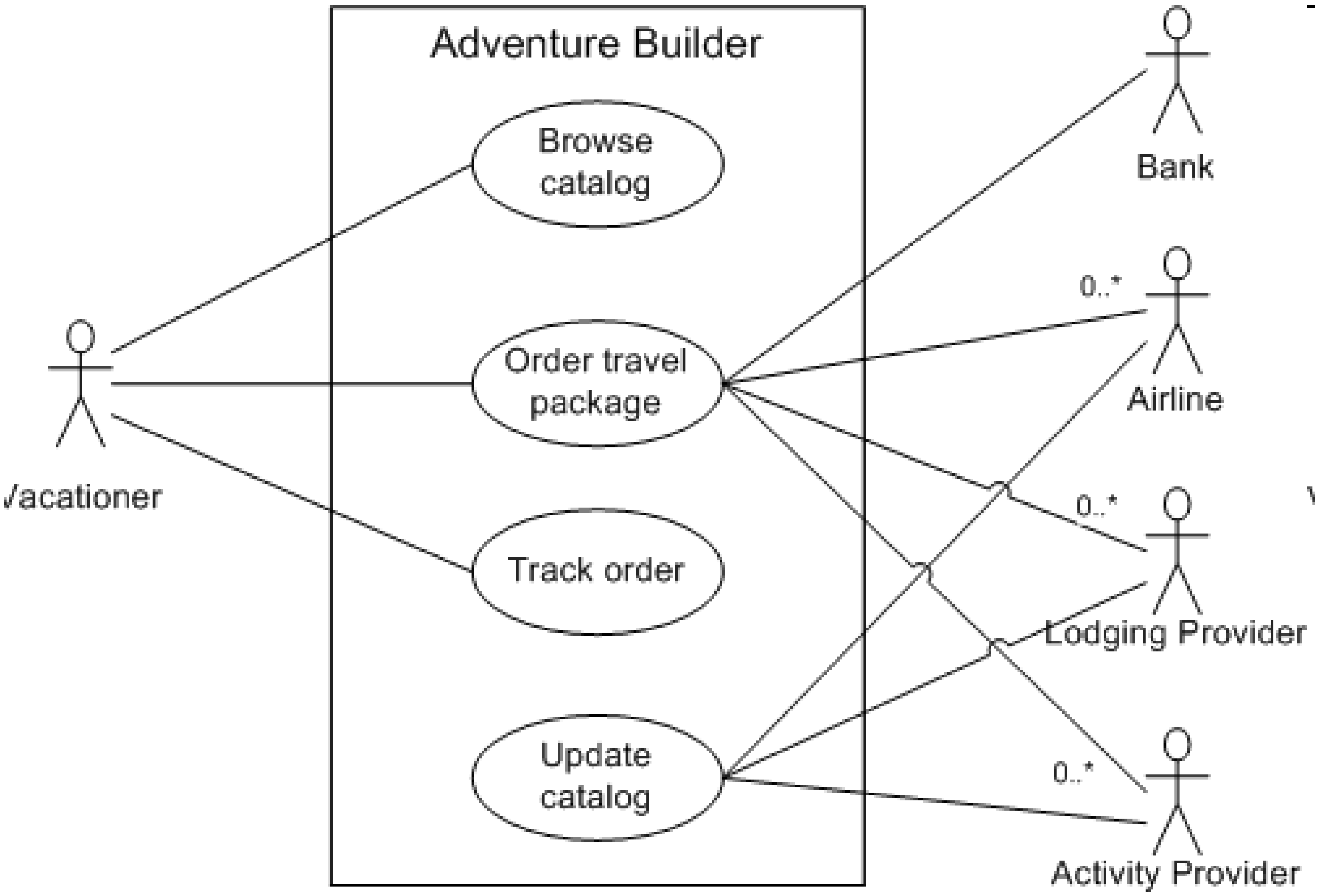
Usefullness of views

Structure	Relations	Useful for
Decomposition	Is a submodule of; shares secret with	Resource allocation and project structuring and planning; information hiding, encapsulation; configuration control
Uses	Requires the correct presence of	Engineering subsets; engineering extensions
Layered	Requires the correct presence of; uses the services of; provides abstraction to	Incremental development; implementing systems on top of "virtual machines" portability
Class	Is an instance of; shares access methods of	In OO design systems, producing rapid almost-alike implementations from a common template
Client-Server	Communicates with; depends on	Distributed operation; separation of concerns; performance analysis; load balancing
Process	Runs concurrently with; may run concurrently with; excludes; precedes; etc.	Scheduling analysis; performance analysis
Concurrency	Runs on the same logical thread	Identifying locations where resource contention exists, where threads may fork, join, be created or be killed
Shared Data	Produces data; consumes data	Performance; data integrity; modifiability
Deployment	Allocated to; migrates to	Performance, availability, security analysis
Implementation	Stored in	Configuration control, integration, test activities
Work Assignment	Assigned to	Project management, best use of expertise, management of commonality

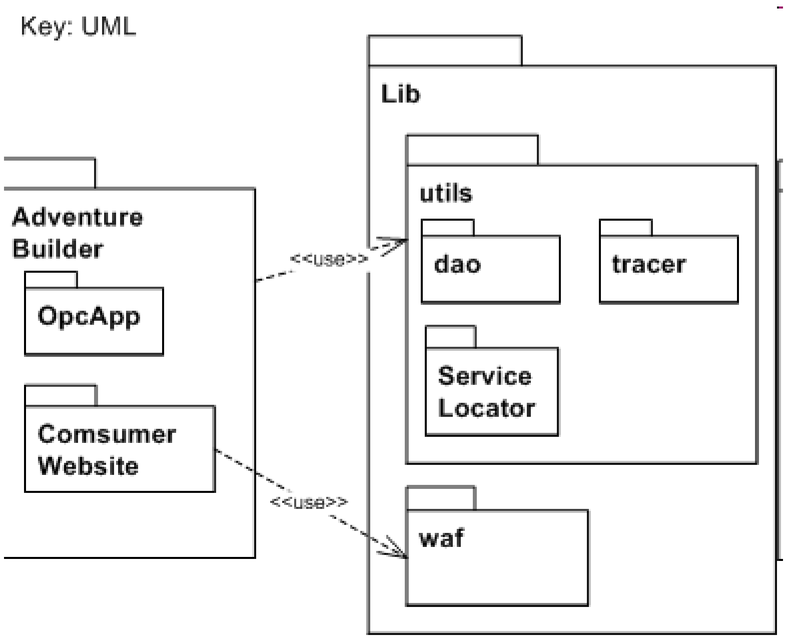
Sample Views

https://wiki.sei.cmu.edu/sad

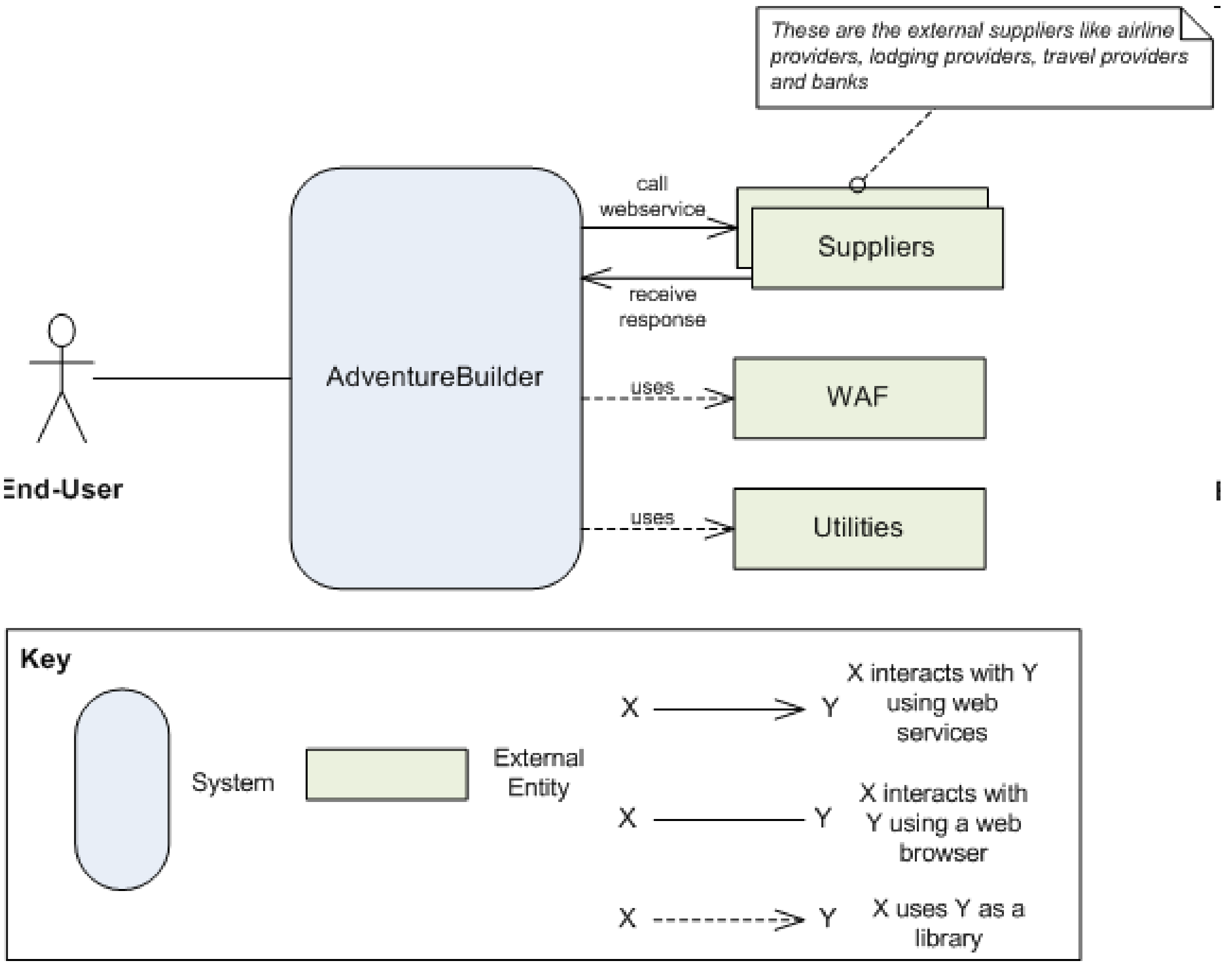
Functionality



High level **Module Uses** view

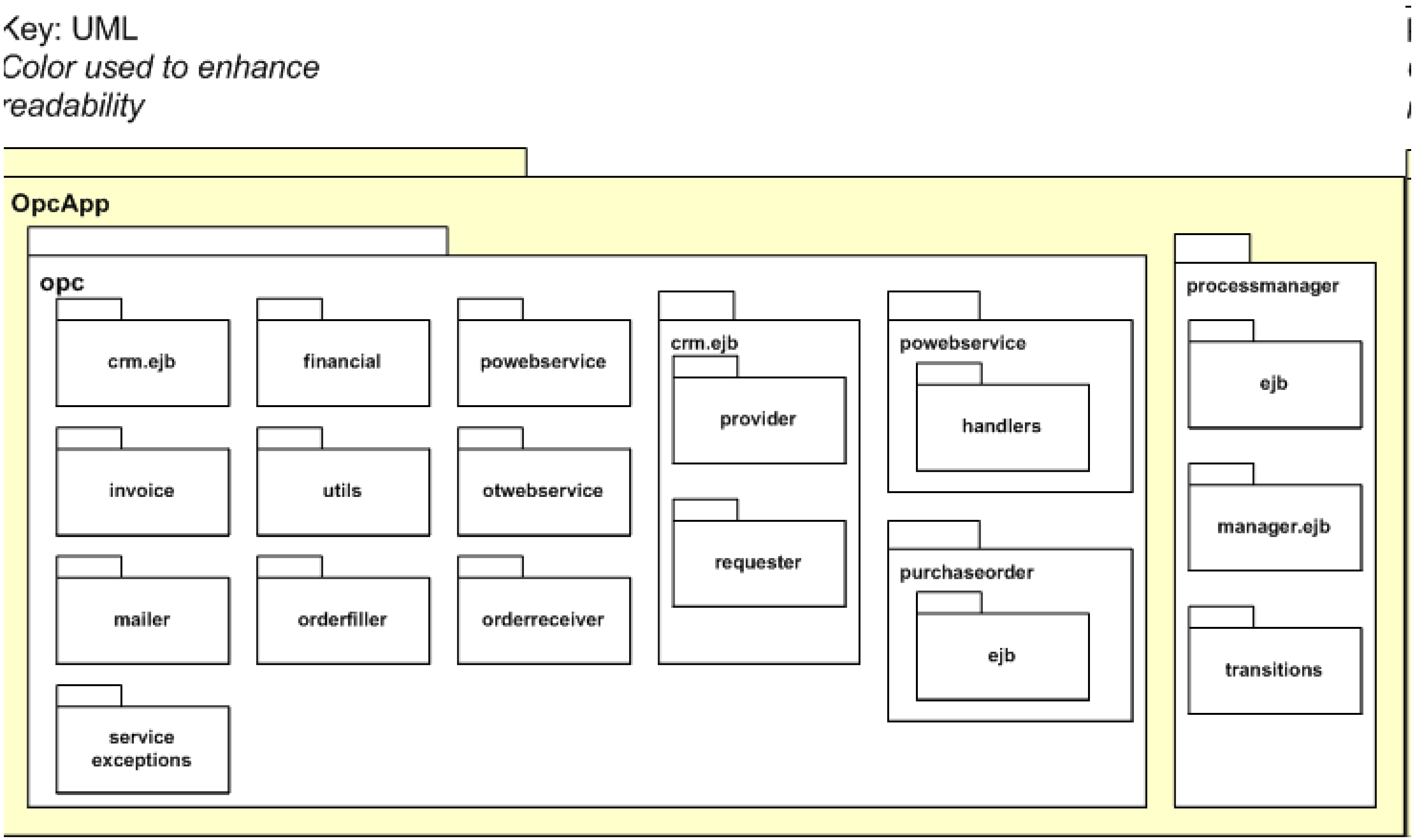


Context Diagram

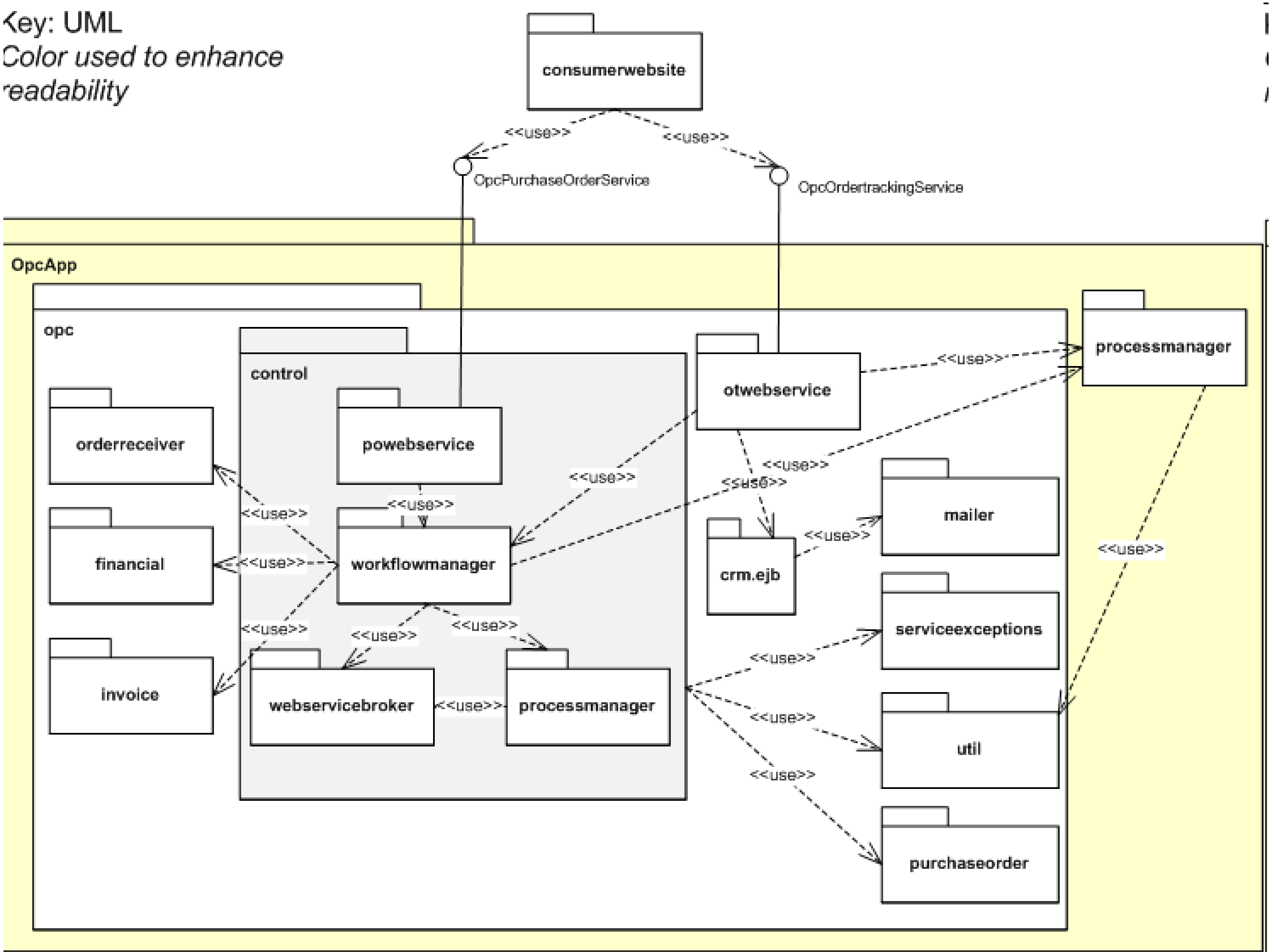


These are the external suppliers like airline providers, lodging providers, travel providers and banks

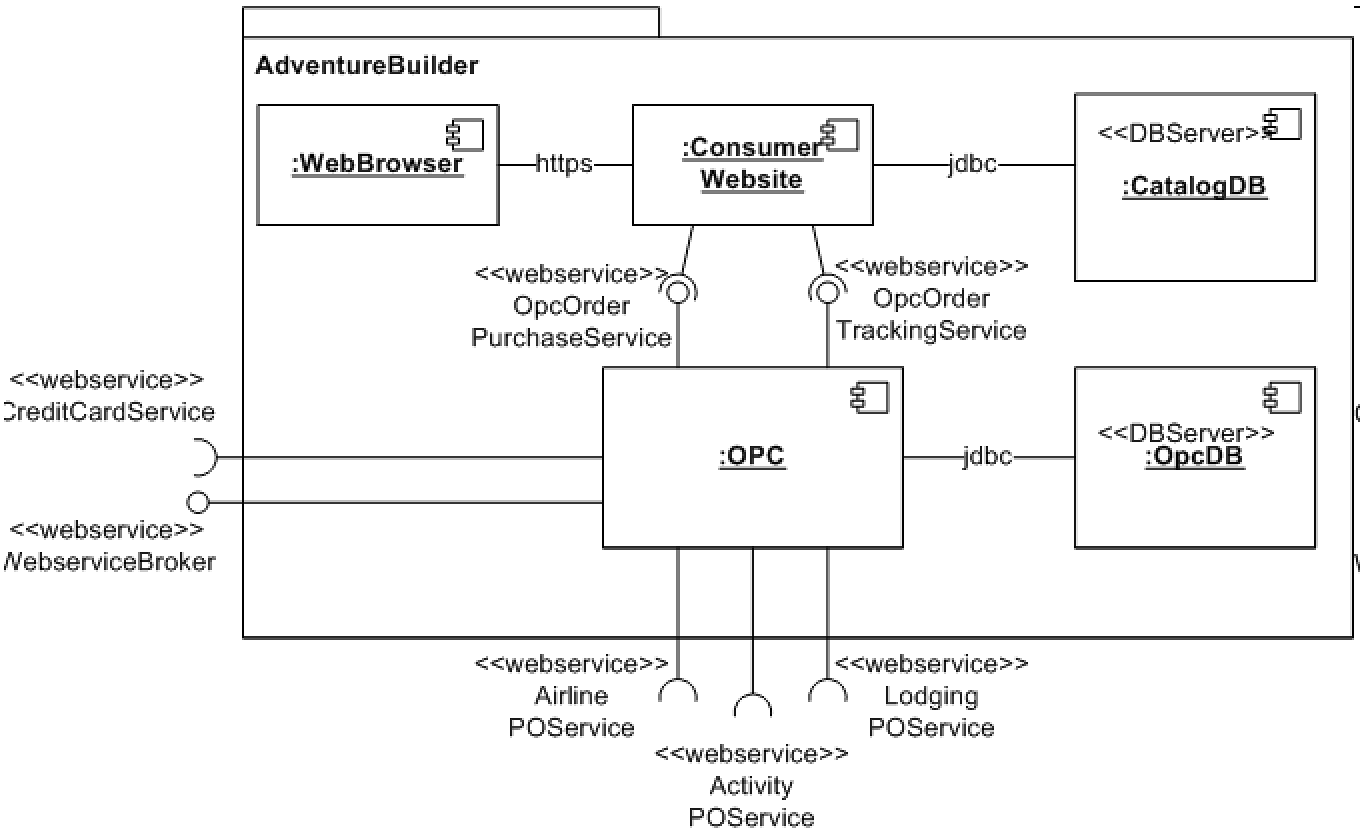
OPC **Module Decomposition** view



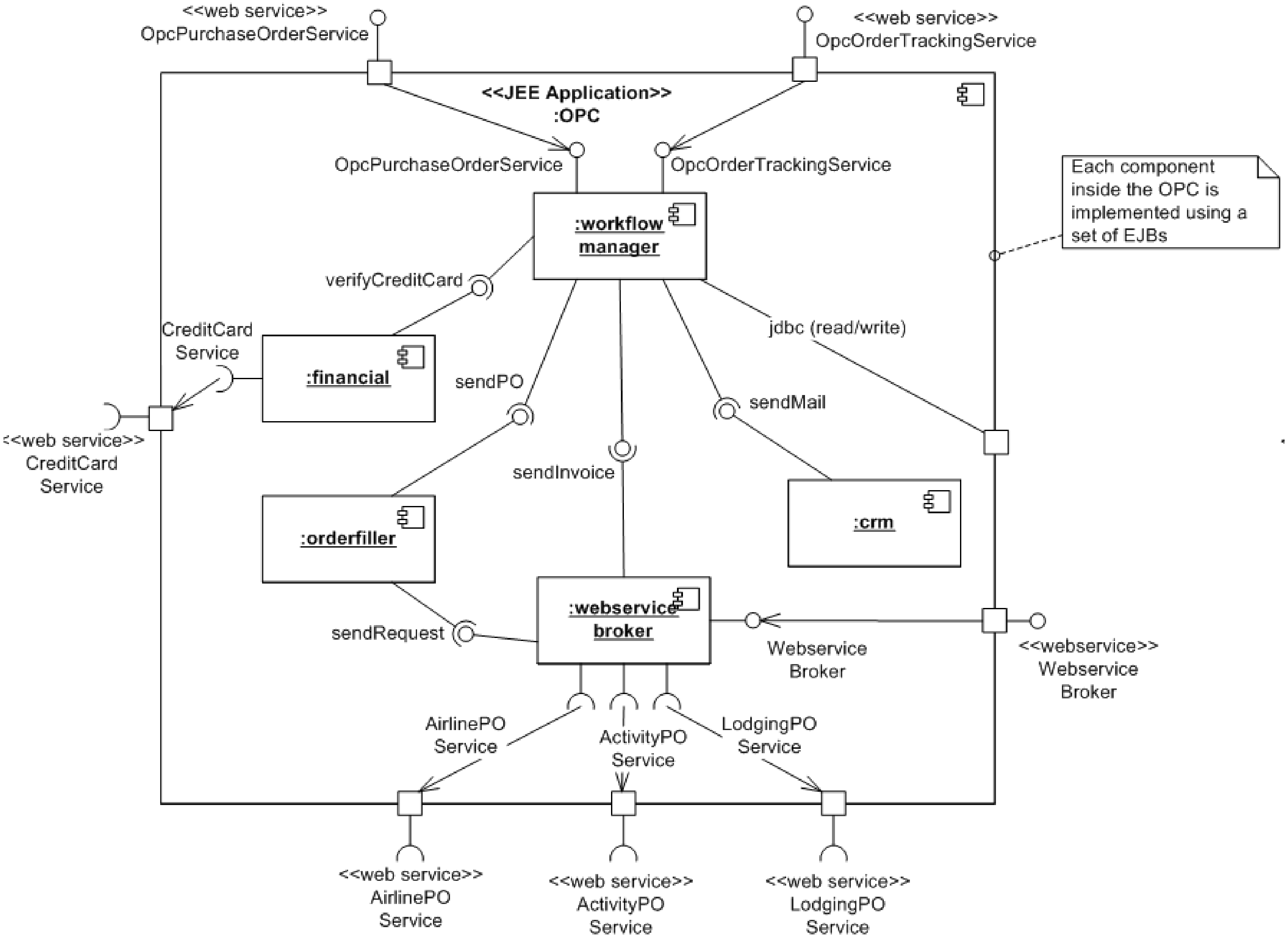
OPC **Module Uses** view



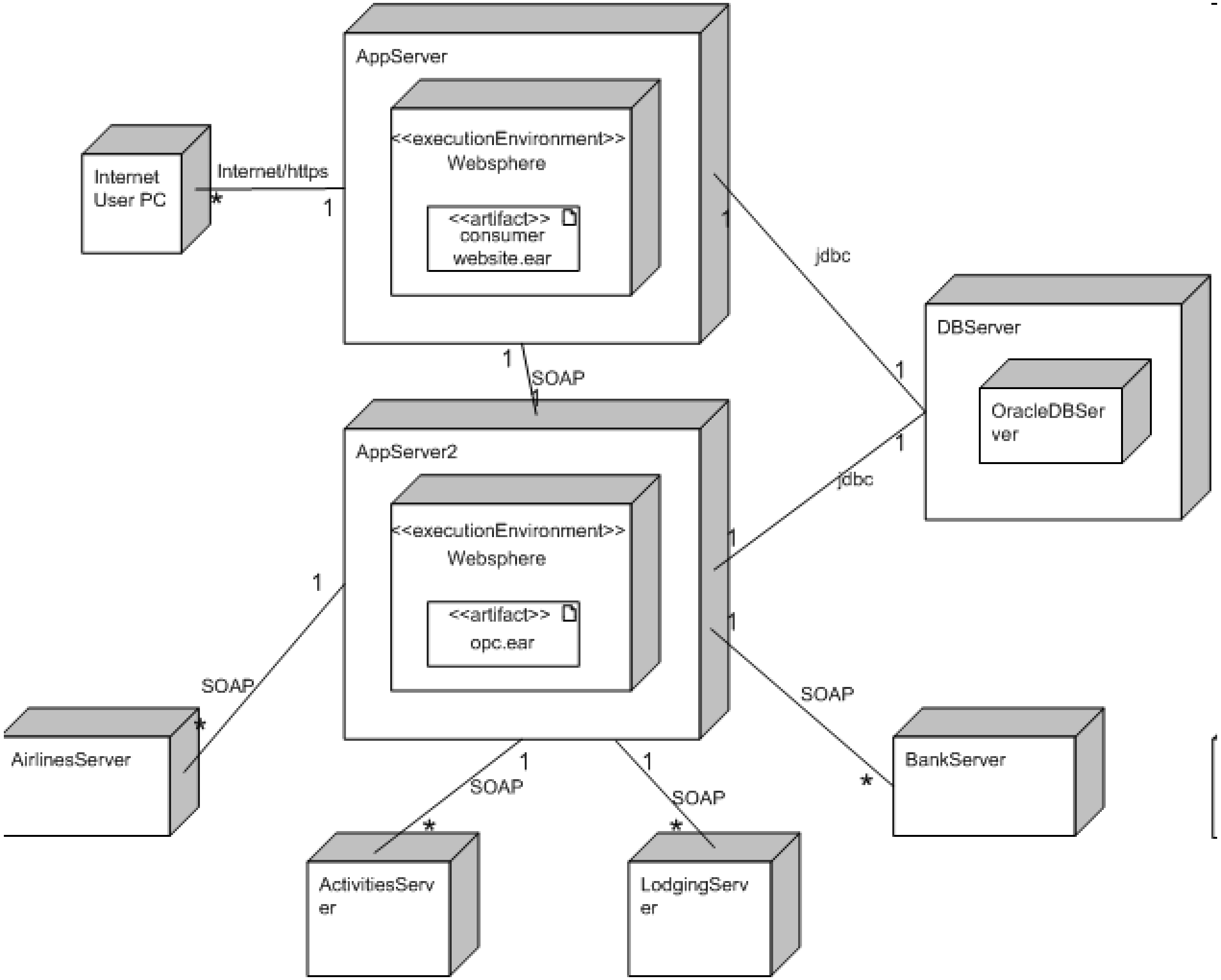
High level **C&C** view



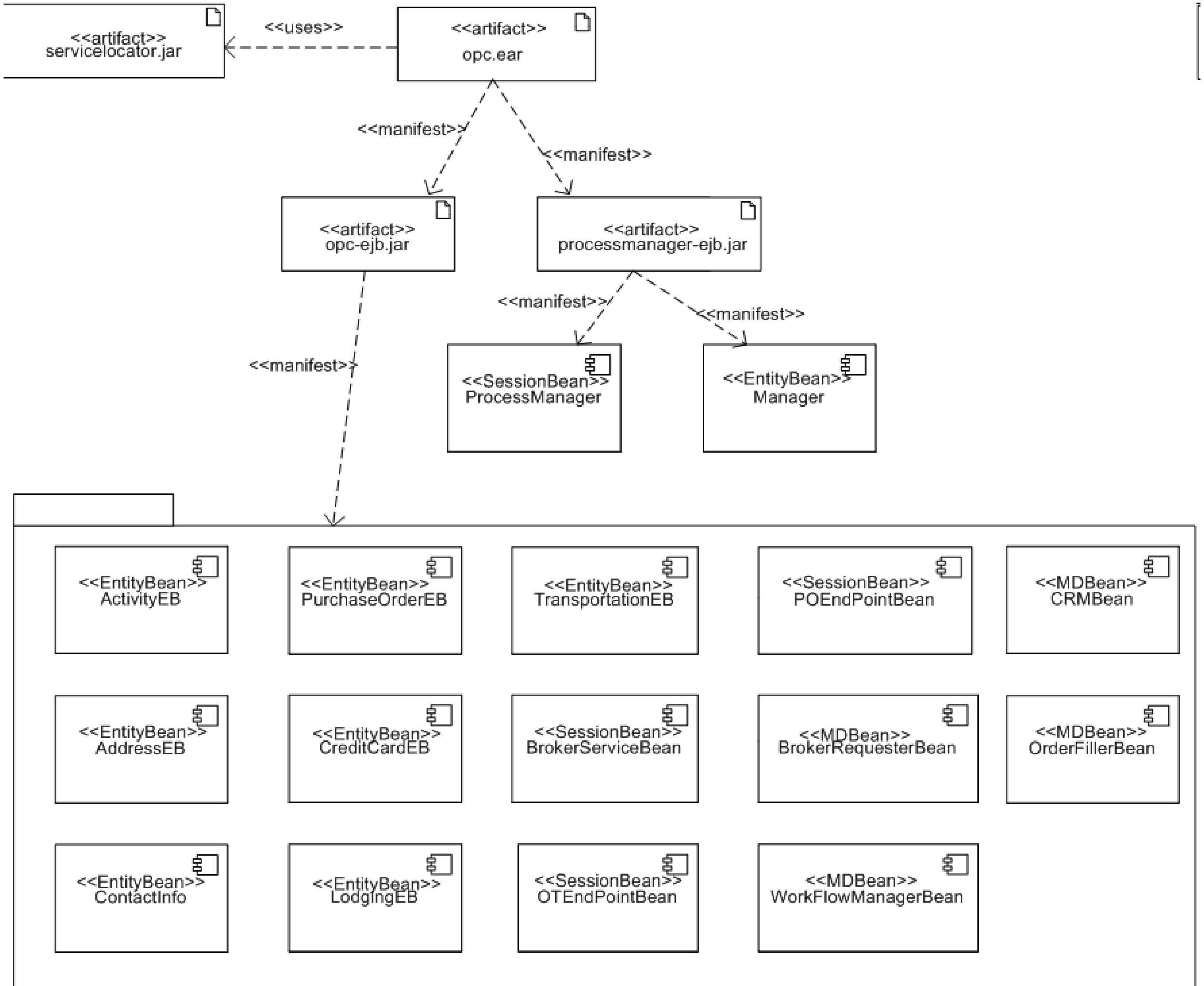
OPC **C&C** view


<<web service>>
OpcPurchaseOrderService
<<web service>>
OpcOrderTrackingService
<<JEE Application>>
:OPC
OpcPurchaseOrderService
OpcOrderTrackingService
:workflow manager
Each component inside the OPC is implemented using a set of EJBs
verifyCreditCard
CreditCard Service
:financial
jdbc (read/write)
sendPO
sendMail
<<web service>>
CreditCard
Service
sendInvoice
:orderfiller
:crm
:webservice broker
sendRequest
Webservice Broker
<<webservice>>
Webservice
Broker
AirlinePO Service
ActivityPO Service
LodgingPO Service
<<web service>>
AirlinePO
Service
<<web service>>
ActivityPO
Service
<<web service>>
LodgingPO
Service

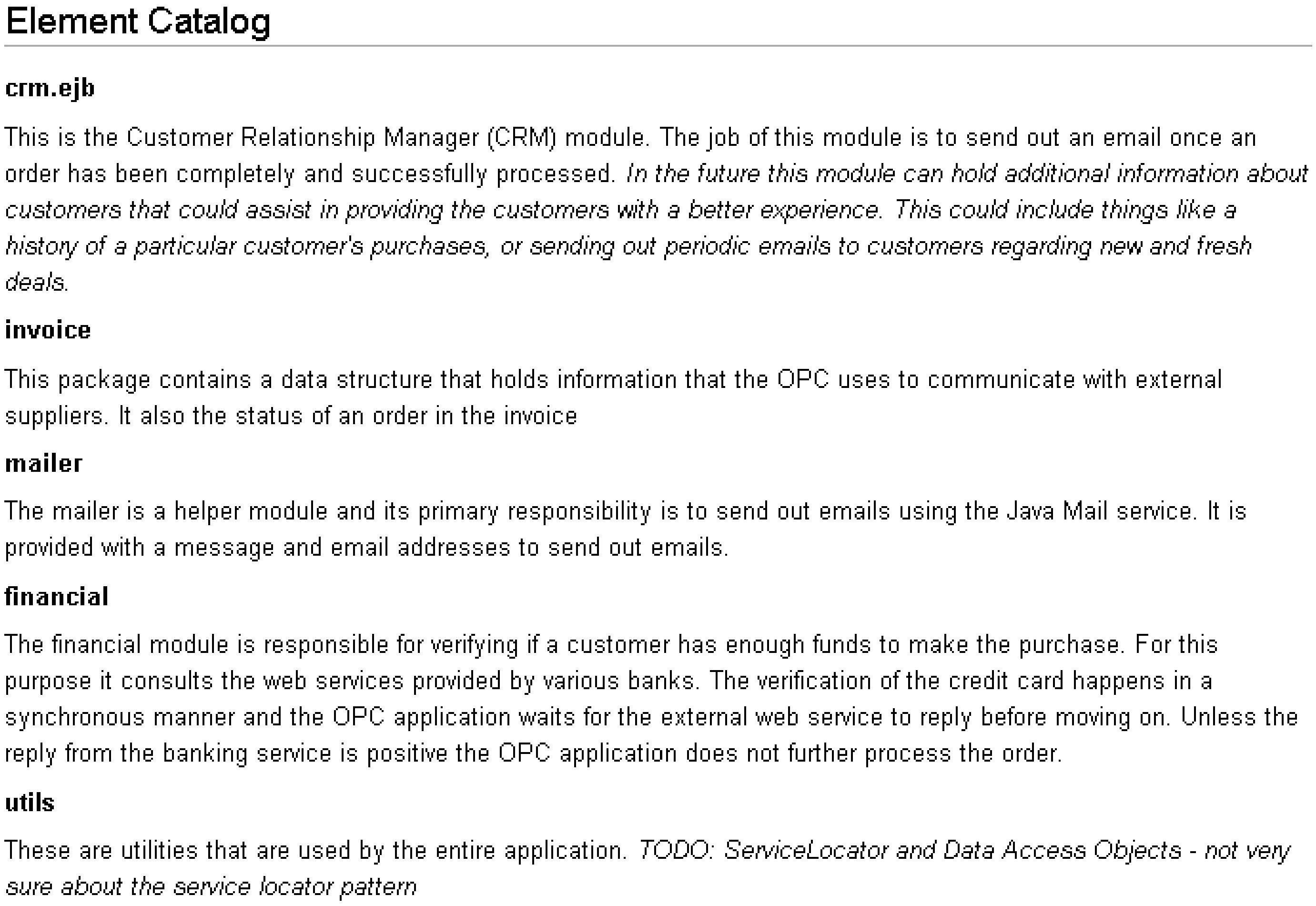
Deployment view



High level **Implementation** view



Element catalog



Element Catalog

crm.ejb

This is the Customer Relationship Manager (CRM) module. The job of this module is to send out an email once an order has been completely and successfully processed. *In the future this module can hold additional information about customers that could assist in providing the customers with a better experience. This could include things like a history of a particular customer's purchases, or sending out periodic emails to customers regarding new and fresh deals.*

invoice

This package contains a data structure that holds information that the OPC uses to communicate with external suppliers. It also the status of an order in the invoice

mailer

The mailer is a helper module and its primary responsibility is to send out emails using the Java Mail service. It is provided with a message and email addresses to send out emails.

financial

The financial module is responsible for verifying if a customer has enough funds to make the purchase. For this purpose it consults the web services provided by various banks. The verification of the credit card happens in a synchronous manner and the OPC application waits for the external web service to reply before moving on. Unless the reply from the banking service is positive the OPC application does not further process the order.

utils

These are utilities that are used by the entire application. *TODO: ServiceLocator and Data Access Objects - not very sure about the service locator pattern*

Discussion

SAD

- Intended audience (stakeholders)
- Stakeholders’ concerns to address
- Sections to include
- Views to include
- Depth of documentation

Stakeholders

- Sample stakeholders:
  - Project Manager
  - Member of Development Team
  - Testers and Integrators
  - Maintainers
  - Product Line Application Builder
  - Customer
  - End User
  - Analyst
  - Infrastructure Support
  - New Stakeholder
  - Current and Future Architect

Who are the stakeholders?

Which concerns do they want to address?

stakeholders

Stakeholder	Concerns to adddress

SEI example Stakeholders' documentation needs

	Module				C&C	Allocation	
Stakeholder	**Decomposition**	**Uses**	**Class**	**Layer**	***Various***	**Deployment**	**Implem.**
Project Manager	s	s		s		d	
Member of Development Team	d	d	d	d	d	s	s
Testers and Integrators		d	d		s	s	s
Maintainers	d	d	d	d	d	s	s
Product Line Application Builder		d	s	o	s	s	s
Customer					s	o	
End User					s	s	
Analyst	d	d	s	d	s	d	
Infrastructure Support	s	s		s		s	d
New Stakeholder	x	x	x	x	x	x	x
Current and Future Architect	d	d	d	d	d	d	s

d = detailed information, s = some details, o = overview information, x = anything

To fill in...

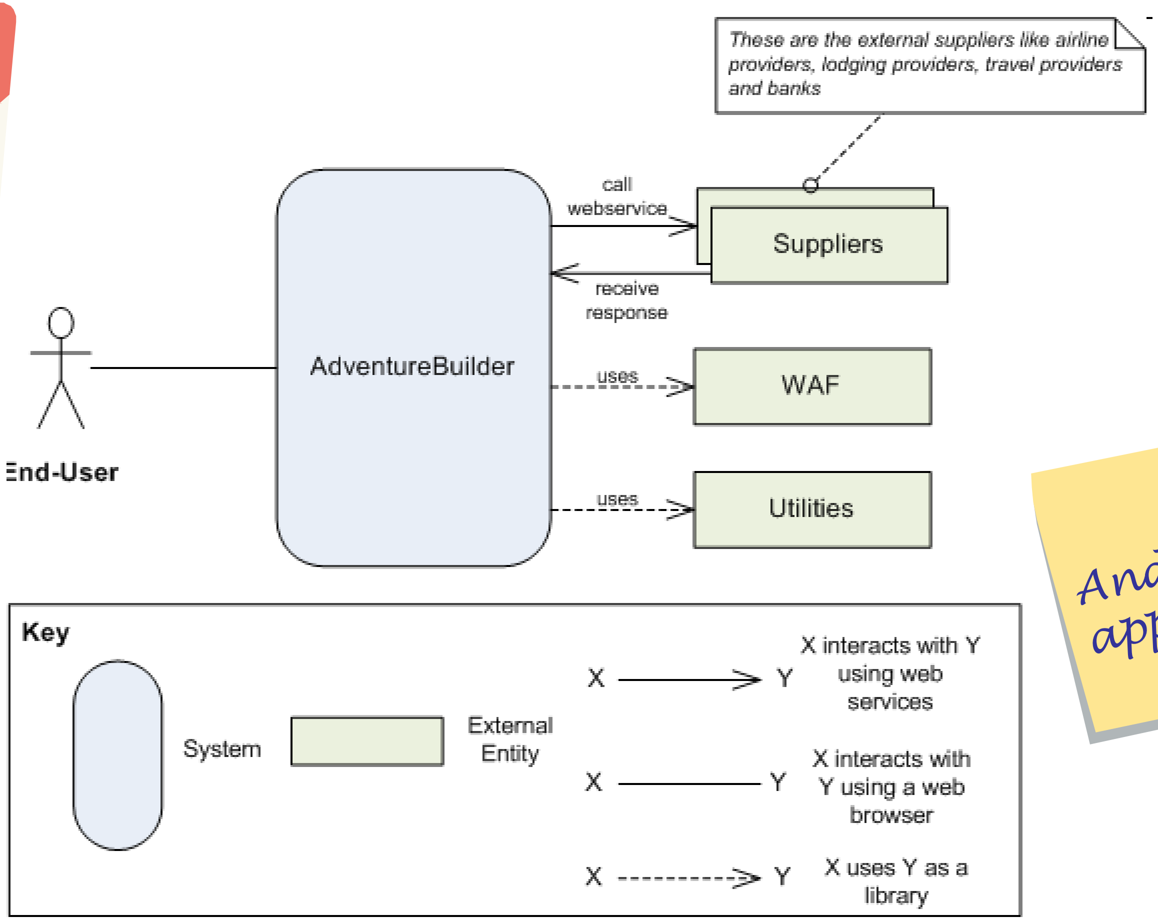
Stakeholders' documentation needs

Stakeholder	Module Views				C&C Views	Allocation Views	
	Decomposition	Uses	Class	Layer	*Various*	Deployment	Implementation

d = detailed information, s = some details, o = overview information, x = anything

Context

What is the context diagram for your project?


These are the external suppliers like airline providers, lodging providers, travel providers and banks
call webservice
receive response
Suppliers
AdventureBuilder
End-User
uses
WAF
uses
Utilities
Key
System
External Entity
X → Y X interacts with Y using web services
X — Y X interacts with Y using a web browser
X - - -> Y X uses Y as a library

And for each application?

RUP’s SAD

1. Introduction
    1. Purpose
    2. Scope
    3. Definitions, Acronyms, and Abbreviations
    4. References
    5. Overview
2. Architectural Representation
3. Architectural Goals and Constraints
4. Use-Case View
    1. Use-Case Realizations
5. Logical View
    1. Overview
    2. Architecturally Significant Design Packages
6. Process View
7. Deployment View
8. Implementation View
    1. Overview
    2. Layers
9. Data View (optional)
10. Size and Performance
11. Quality

Which sections to include?

SEI's SAD

Which sections to include?

1. Documentation Roadmap
   1. …
   2. Stakeholder Representation
   3. Viewpoint Definitions
   4. …
2. Architecture Background
   1. Problem Background
      1. System Overview
      2. Goals and Context
      3. Significant Driving Requirements
   2. Solution Background
      1. Architectural Approaches
      2. Analysis Results
      3. Requirements Coverage
      4. Summary of Background Changes Reflected in Current Version
   3. Product Line Reuse Considerations
3. Views
   1. <Insert view name> View
4. …
5. Directory
   1. Index
   2. Glossary
   3. Acronym List

1.5 Viewpoint Definitions
1.5.1 **<viewpoint>** Viewpoint Definition
1.5.1.1 Abstract
1.5.1.2 Stakeholders and Their Concerns Addressed
1.5.1.3 Elements, Relations, Properties and Constraints
1.5.1.4 Language(s) to Model/Represent Conforming Views
1.5.1.5 Applicable Evaluation/Analysis Techniques and Consistency/Completeness Criteria
1.5.1.6 Viewpoint Source

3.1 **<view name>** View
3.1.1 View Description
3.1.2 View Packet Overview
3.1.3 Architecture Background
3.1.4 Variability Mechanisms
3.1.5 View Packets
3.1.5.1 View packet **#i**
3.1.5.1.1 Primary Presentation
3.1.5.1.2 Element Catalog
3.1.5.1.3 Context Diagram
3.1.5.1.4 Variability Mechanisms
3.1.5.1.5 Architecture Background
3.1.5.1.6 Related View Packets

Proposed general content

- Architecture background
  - Problem Background
    - **System Overview**
    - **Goals and Context**
    - **Significant Driving Requirements**
  - Solution Background
    - *Architectural Approaches*
    - *Analysis Results*
    - *Requirements Coverage*

Bold - mandatory
Italic - desireable
Normal - optional

Proposed views to include in a SAD

- Functional view
- High-level module uses view
- High-level module decomposition view
- High-level C&C view
- Deployment view

Proposed content for each view

- View description
- **Primary presentation**
- Element catalog
  - **Elements**
  - *Relations*
  - *Interfaces*
  - Behavior
  - Constraints
- **Context diagram**
- Variability mechanisms
- *Architecture background*

Bold - mandatory
Italic - desireable
Normal - optional

Non-functional view, a.k.a., Quality attributes

- Performance
- Scalability
- Availability
- Security
- Extensibility
- Flexibility
- Monitoring and management
- Reliability
- Interoperability
- Legal and regulatory requirements
- Internationalisation and localisation
- Accessibility
- …

Which NFRs matter most for your project?

Other questions?

Thank you