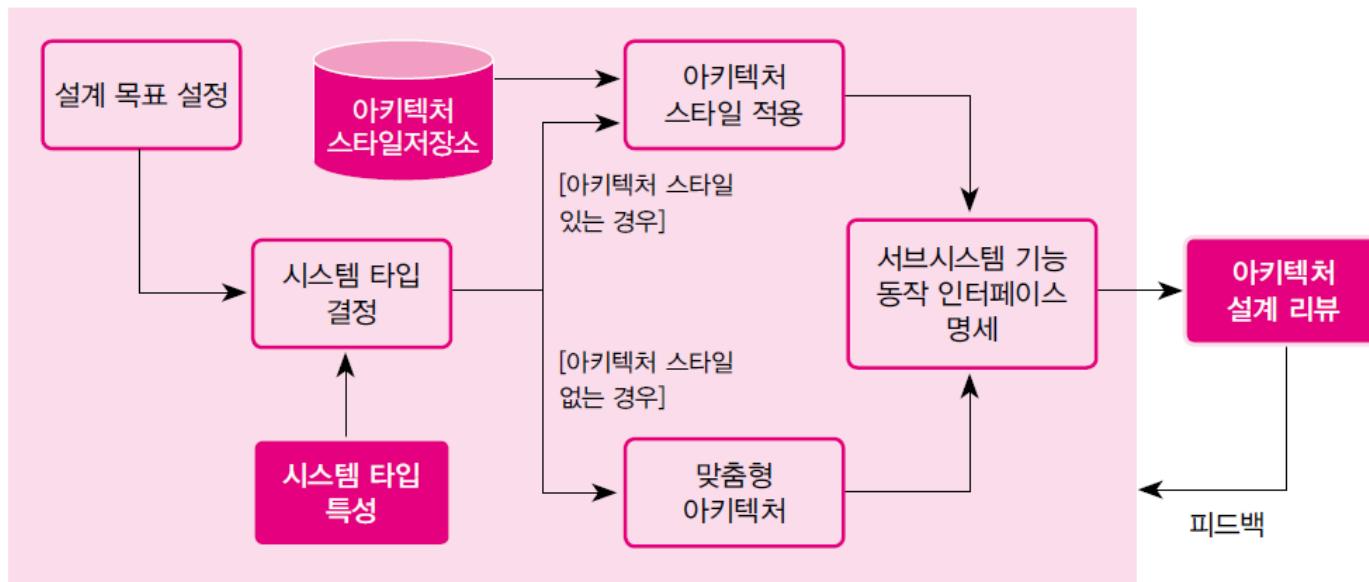


6.3 아키텍처 설계과정

● 아키텍처 설계 프로세스

- Step 1. 목표를 설정한다.
- Step 2. 시스템의 타입을 결정한다.
- Step 3. 아키텍처 스타일을 적용하거나 아키텍처 설계를 커스터마이징한다.
- Step 4. 서브시스템의 기능, 인터페이스 인터랙션 동작을 작성한다.
- Step 5. 아키텍처 설계를 검토한다.



아키텍처 설계 목표 설정

- 아키텍처 설계에서 고려해야 될 요구사항을 반영하게 됨 → 아키텍처 품질
 - 변경, 유지보수 용이성
 - 상용 컴포넌트의 사용
 - 시스템 성능
 - 신뢰성
 - 보안
 - 고장인내성
 - 복구
 - 확장성

시스템의 타입

● 대화형 시스템

액터와 시스템 사이의 대화를 통해 비즈니스 프로세스 실행
대화가 액터에서 시작해 끝남. 예) 온라인 쇼핑몰

● 이벤트 중심 시스템

상태에 의존적이며 반응 동작을 보인다.

외부 엔터티로부터 이벤트를 받고, 액션을 취하고 상태 변화
입력이벤트에는 순서가 정해져 있지 않음. 예) 주로 내장형
시스템

시스템의 타입

● 변환 시스템

입력을 출력으로 변환하는 정보처리 작업 수행.
순차적, 조건분기, 병렬 같은 제어 흐름
상태가 없고, 계산집약적인 알고리즘 요구

● 객체영속 시스템

- 저장미디어를 숨기고 데이터베이스나 파일 시스템에 객체를 저장하고 검색할 수 있는 능력을 가진 시스템



아키텍처 표현

● 아키텍처 관점

● 모듈

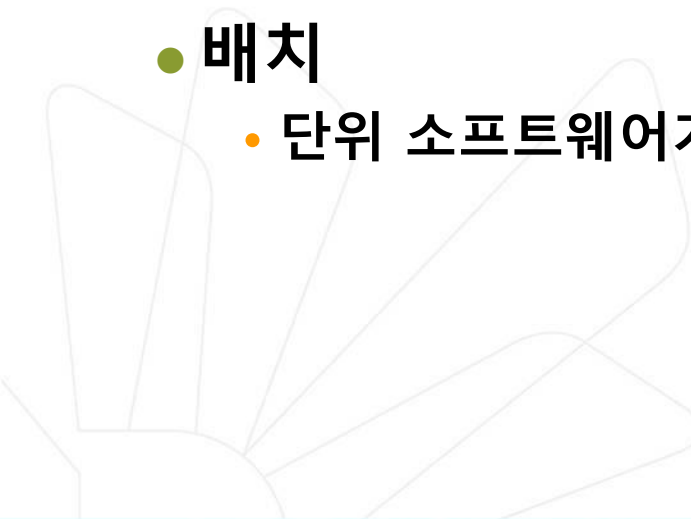
- 시스템을 단위코드의 집합으로 보는 관점

● 컴포넌트와 커넥션

- 시스템을 컴포넌트라 부르는 런타임 개체의 집합으로 보는 관점

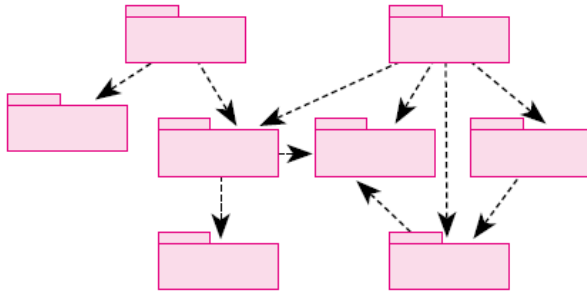
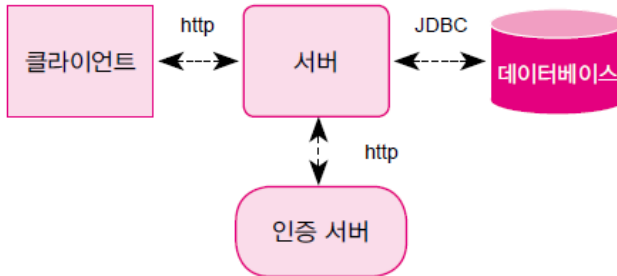
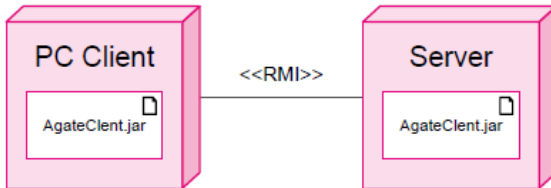
● 배치

- 단위 소프트웨어가 어떤 하드웨어 노드에 배치되는가에 관점



아키텍처 표현

● 아키텍처의 표현 관점

뷰	중심 관점	사례
모듈	코드 구조 - 의존, 상속, 부분 관계 등	
컴포넌트와 커넥터	런타임 구조 - 통신, 호출 관계	
배치	소프트웨어와 환경의 배치 구조	

Class
diagram,
package
diagram

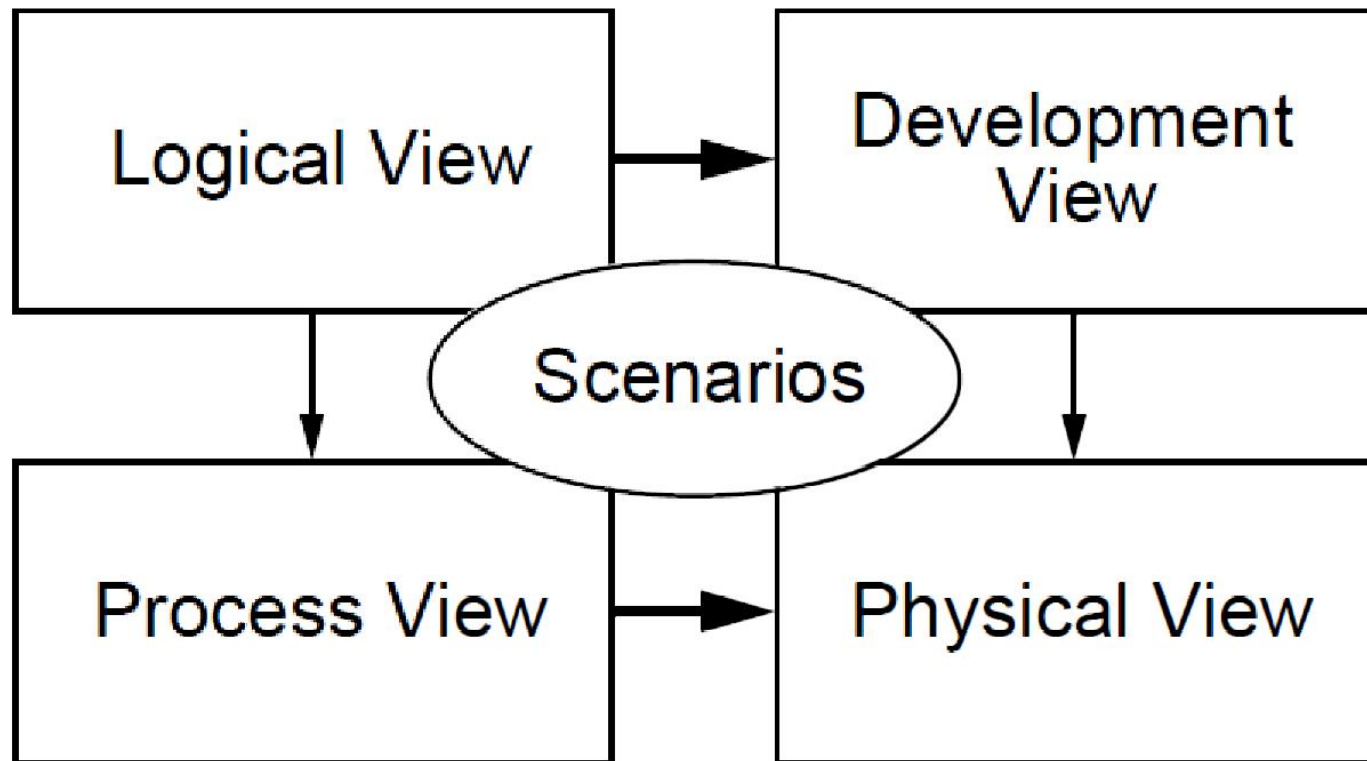
일반 process
구성도

deployment
diagram

4+1 Views

End-user
Functionality

Programmers
Software management

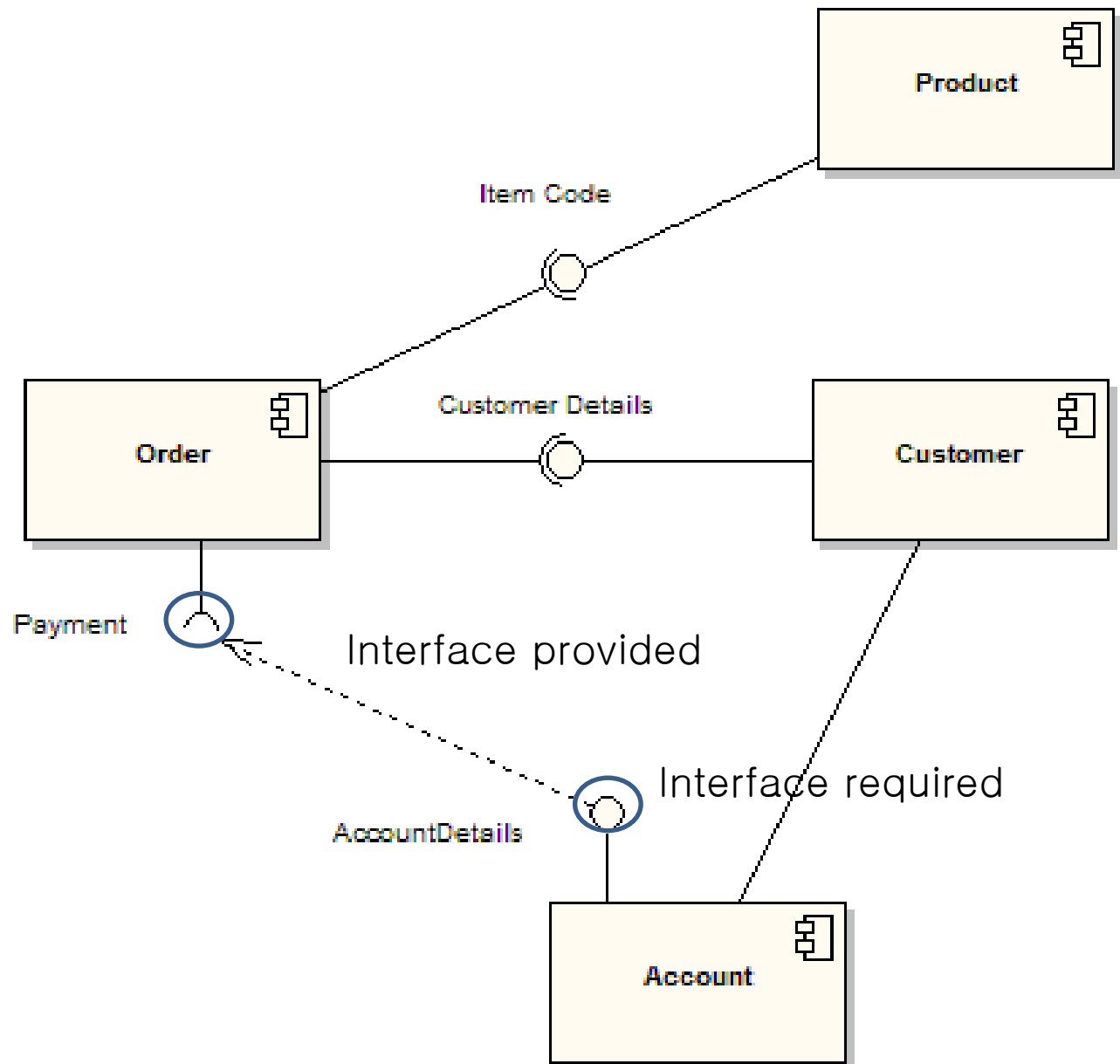


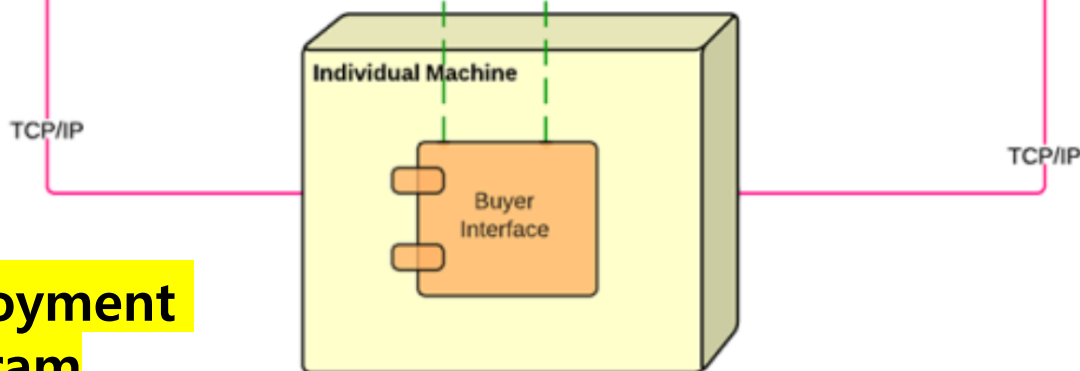
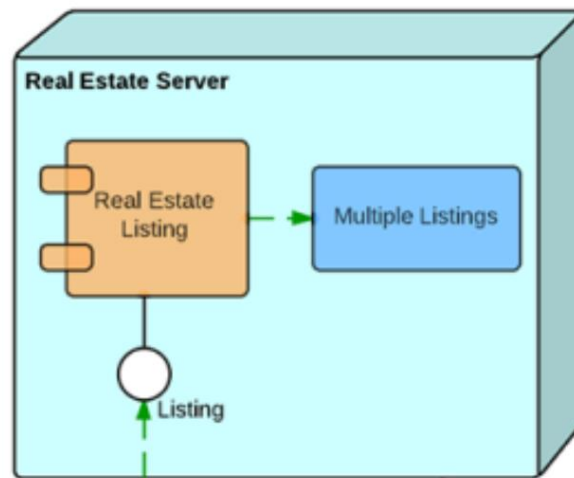
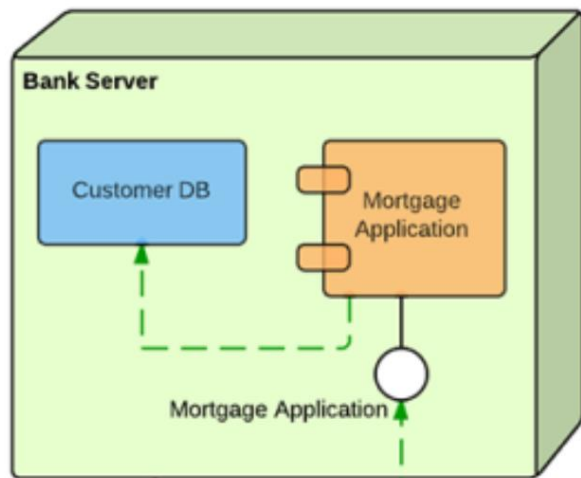
Integrators
Performance
Scalability

System engineers
Topology
Communications

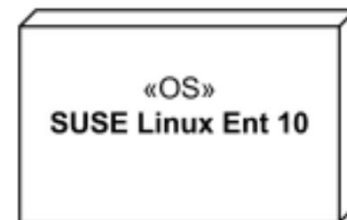
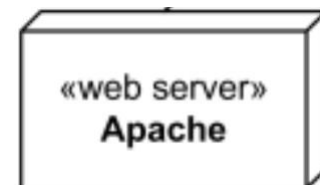
- **Logical view**: object-oriented decomposition. functionality that the system provides to end-users. class diagram, state diagram.
- **Development view**: Subsystem decomposition. component diagram, package diagram
- **Process view**: dynamic aspects (processes) of the system. sequence diagram, activity diagram.
- **Physical view**: deployment view. deployment diagram
- **Scenarios**: use case diagram

Component diagram Example.





deployment
diagram



Node 심볼 예제

