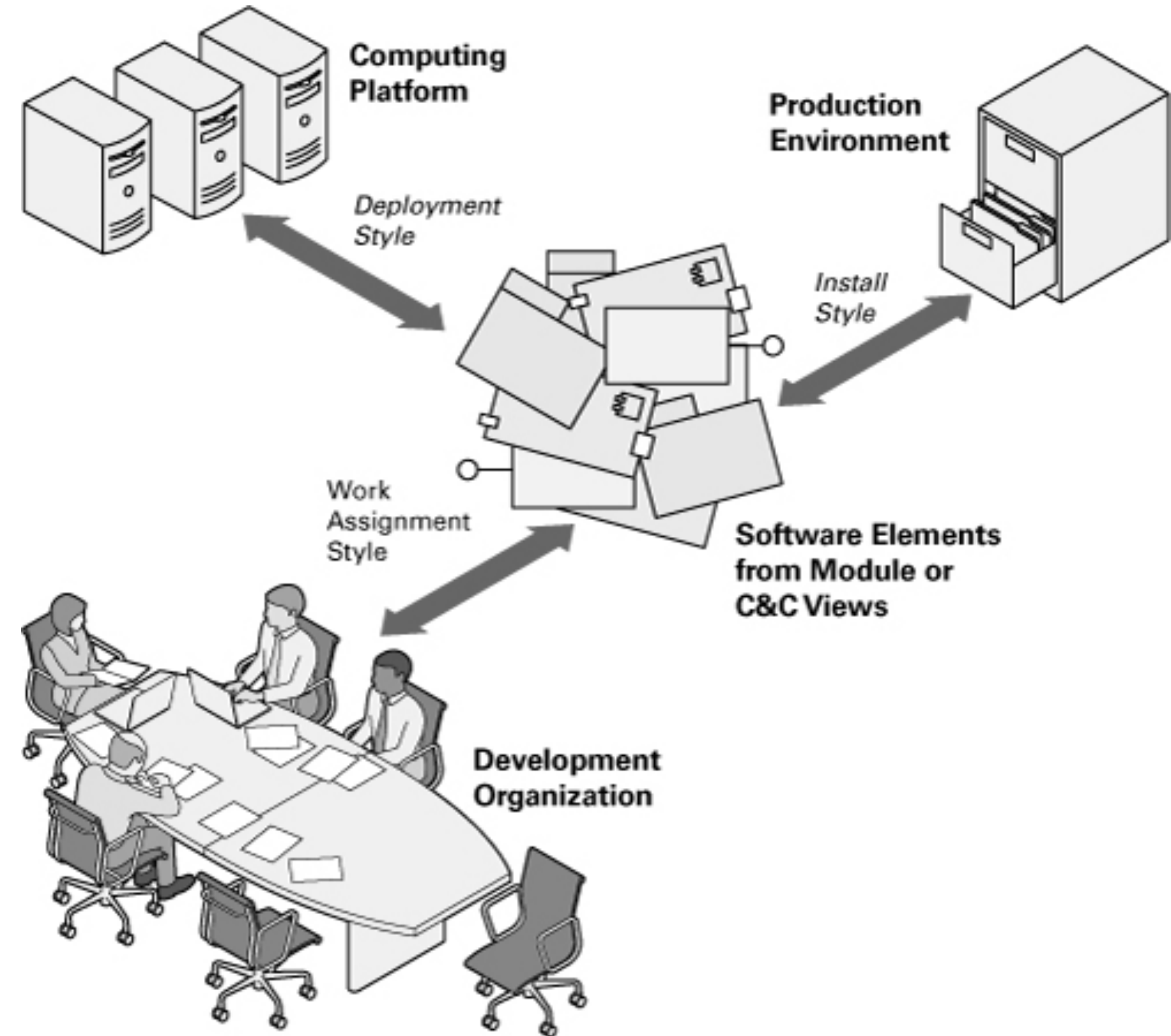


SENG 350

Architecture Styles: Allocation Views

Allocation views map software architecture onto "things" in the environment, e.g., :

- computing environment
- production environment
- development environment



Overview

Overview	Allocation styles describe the mapping between the software architecture and its environment.
Elements	<i>Software element</i> and <i>environmental element</i> . A software element has properties that are <i>required</i> of the environment. An environmental element has properties that are <i>provided</i> to the software.
Relations	<i>Allocated-to</i> . A software element is mapped (allocated to) an environmental element. Properties are dependent on the particular style.
Constraints	Varies by style

Deployment style

Overview

Overview	The deployment style describes the mapping of components and connectors in the software architecture to the hardware of the computing platform.
Elements	<ul style="list-style-type: none">• <i>Software element: elements from a C&C view.</i> Useful properties to document include the significant features required from hardware, such as processing, memory, capacity requirements, and fault tolerance.• <i>Environmental elements: hardware of the computing platform—processor, memory, disk, network (such as router, bandwidth, firewall, bridge), and so on.</i> Useful properties of an environmental element are the significant hardware aspects that influence the allocation decision.
Relations	<ul style="list-style-type: none">• <i>Allocated-to.</i> Physical units on which the software elements reside during execution. Properties include whether the allocation can change at execution time or not.• <i>Migrates-to, copy-migrates-to, and/or execution-migrates-to</i> if the allocation is dynamic. Properties include the trigger that causes the migration.
Constraints	The allocation topology is unrestricted. However, the required properties of the software must be satisfied by the provided properties of the hardware.

Properties to document (hardware)

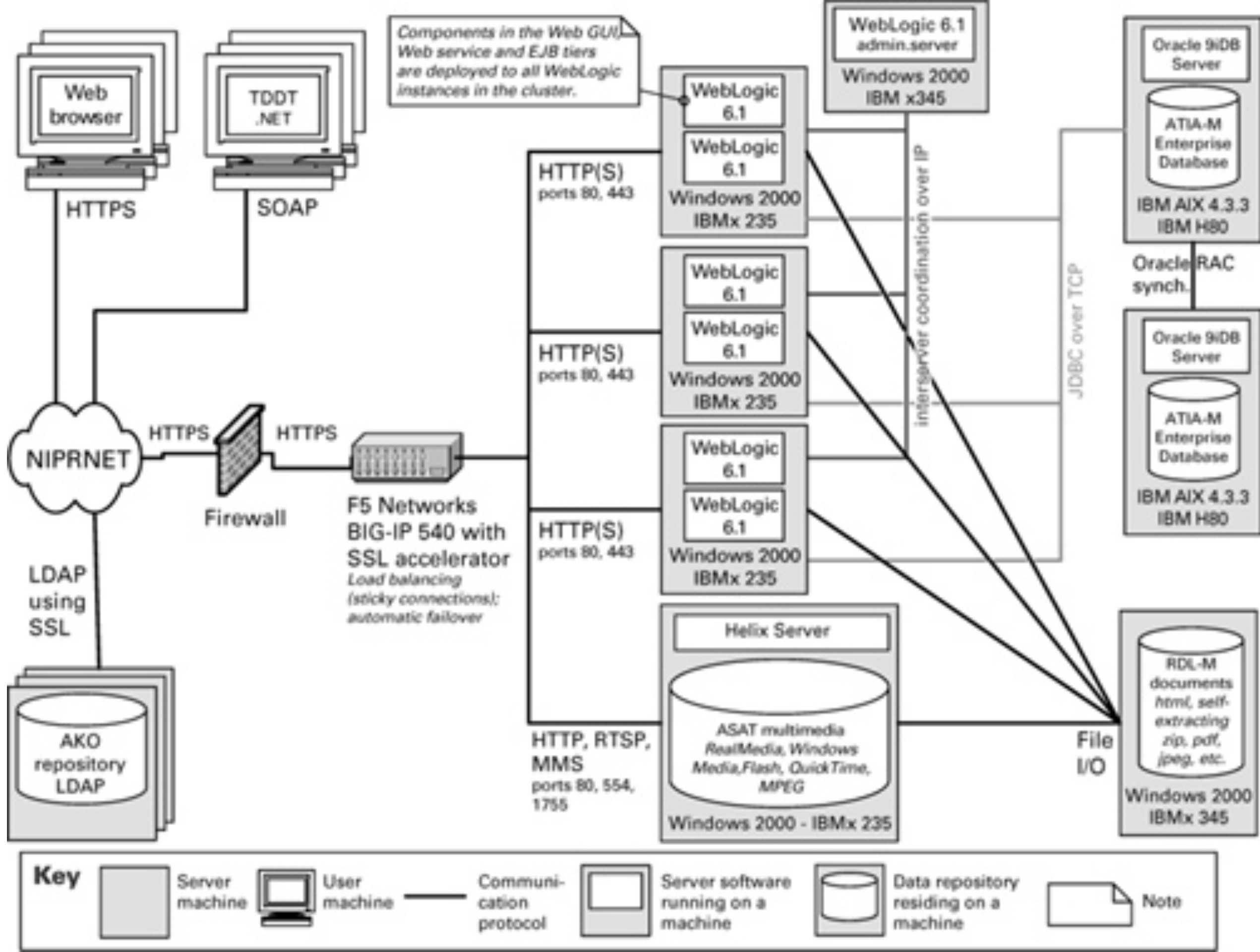
- CPU properties (e.g., clock speed, number of cores, bus speed, cache size, etc.)
- Memory properties (e.g., size and speed)
- Disk or other storage unit capacity (e.g., size, RAID)
- Bandwidth
- Fault tolerance

Properties to document (software)

- Resource consumption (avg, max, min)
- Resource requirements and constraints (max execution time)
- Safety critical (e.g., sw component must always be running)
- Migration trigger (e.g., condition triggering migration to another processor)

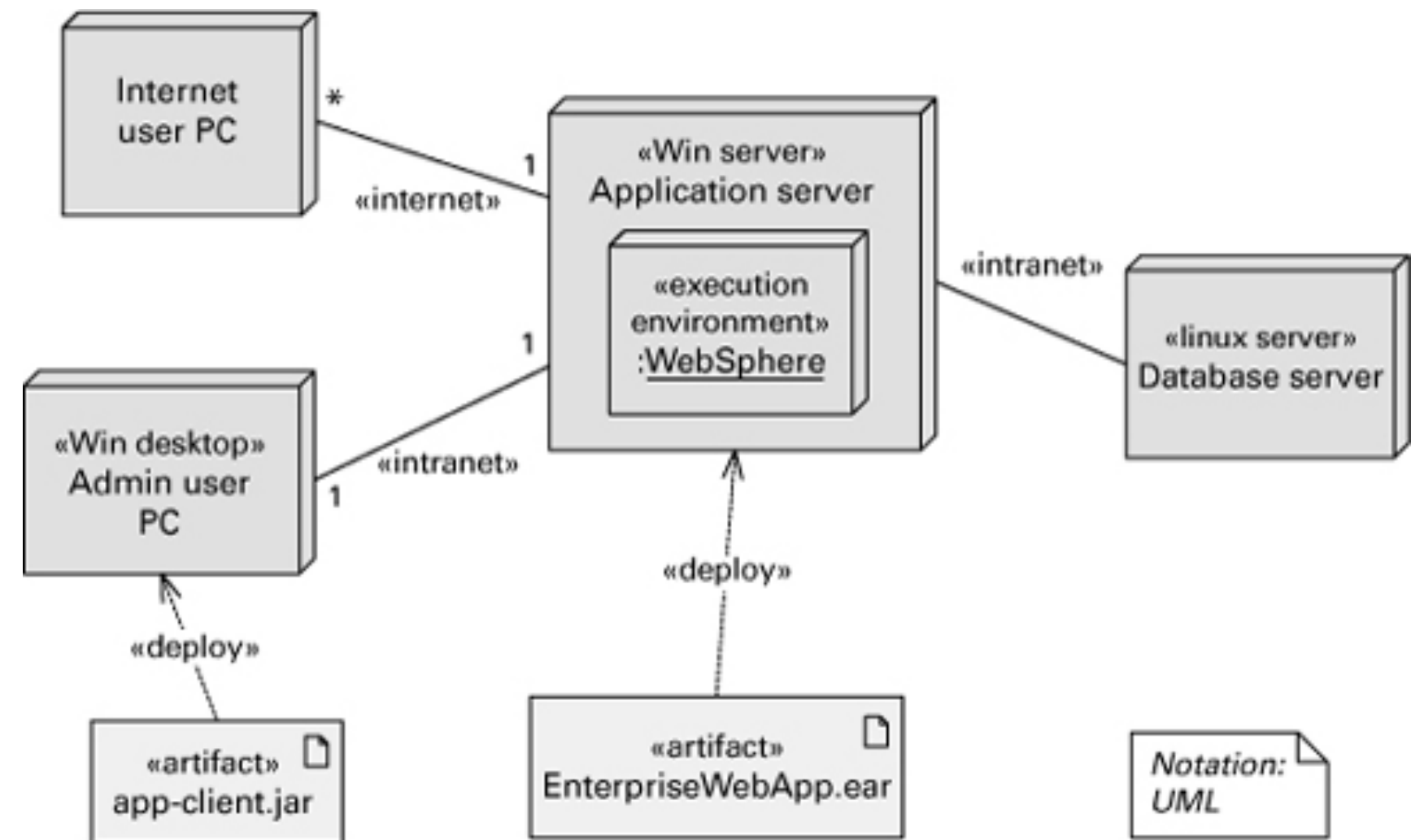
What the Deployment Style is used for

- analyze performance, availability, reliability, security, etc.
- Testing
- Cost estimation



Notation

- informal (e.g., previous slide)
- semi-formal (UML Deployment Diagrams)
- formal (e.g., AADL / SysML - later)



Install Style

Overview	The install style describes the mapping of components in the software architecture to a file system in the production environment.
Elements	<ul style="list-style-type: none">• <i>Software element: a C&C component. Required</i> properties of a software element, if any, usually include requirements on the production environments, such as a requirement to support Java or a database, or specific permissions on the file system.• <i>Environmental element: a configuration item, such as a file or a folder. Provided</i> properties of an environmental element include indications of the characteristics provided by the production environments.
Relations	<ul style="list-style-type: none">• <i>Allocated-to.</i> A component is allocated to a configuration item.• <i>Containment.</i> One configuration item is contained in another.
Constraints	Files and folders are organized in a tree structure, following an <i>is-contained-in</i> relation.

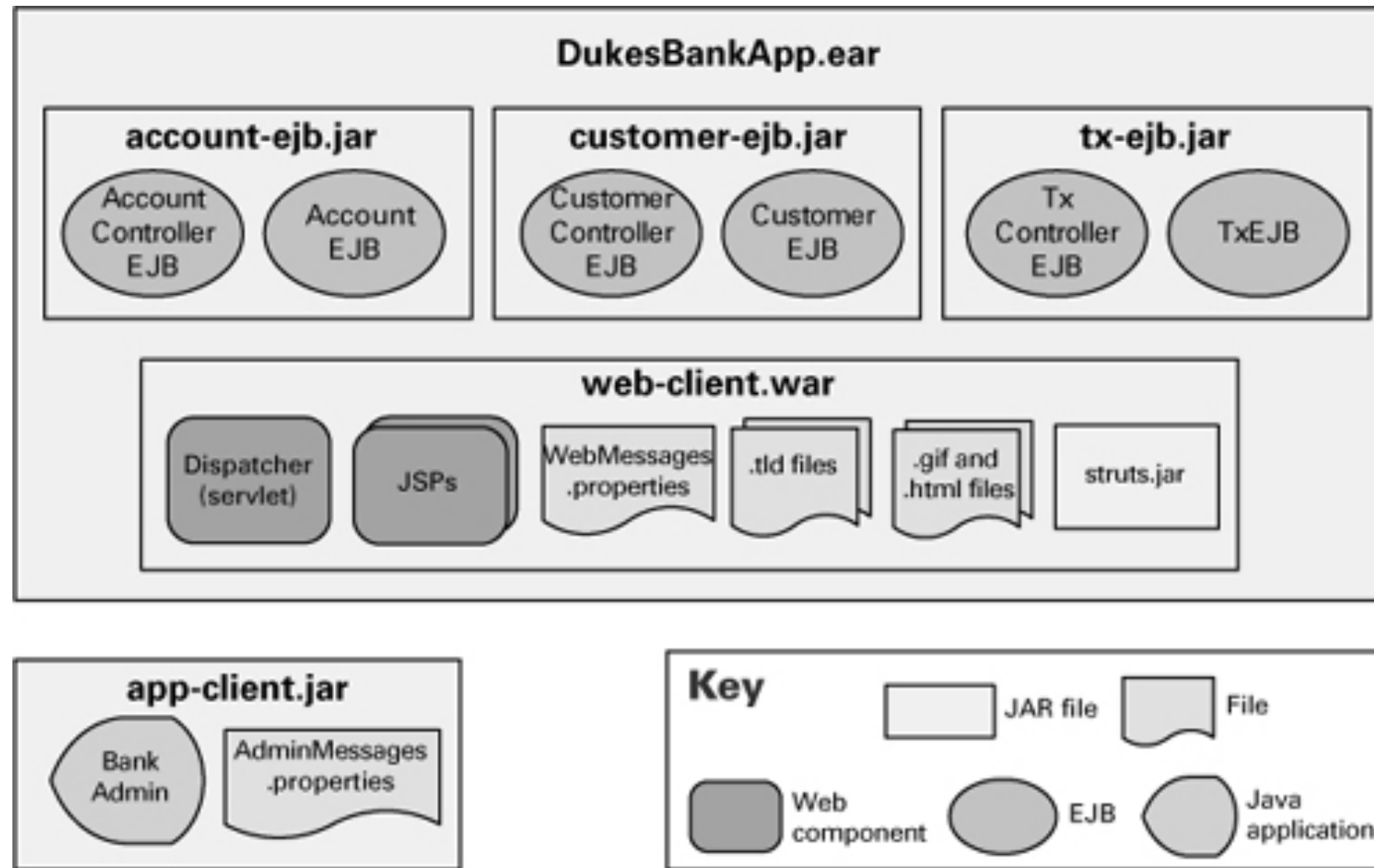
Properties to document

- configuration management system concerns (e.g., branches)
- required version of platform software (e.g., Java JDK)
- variations to installation

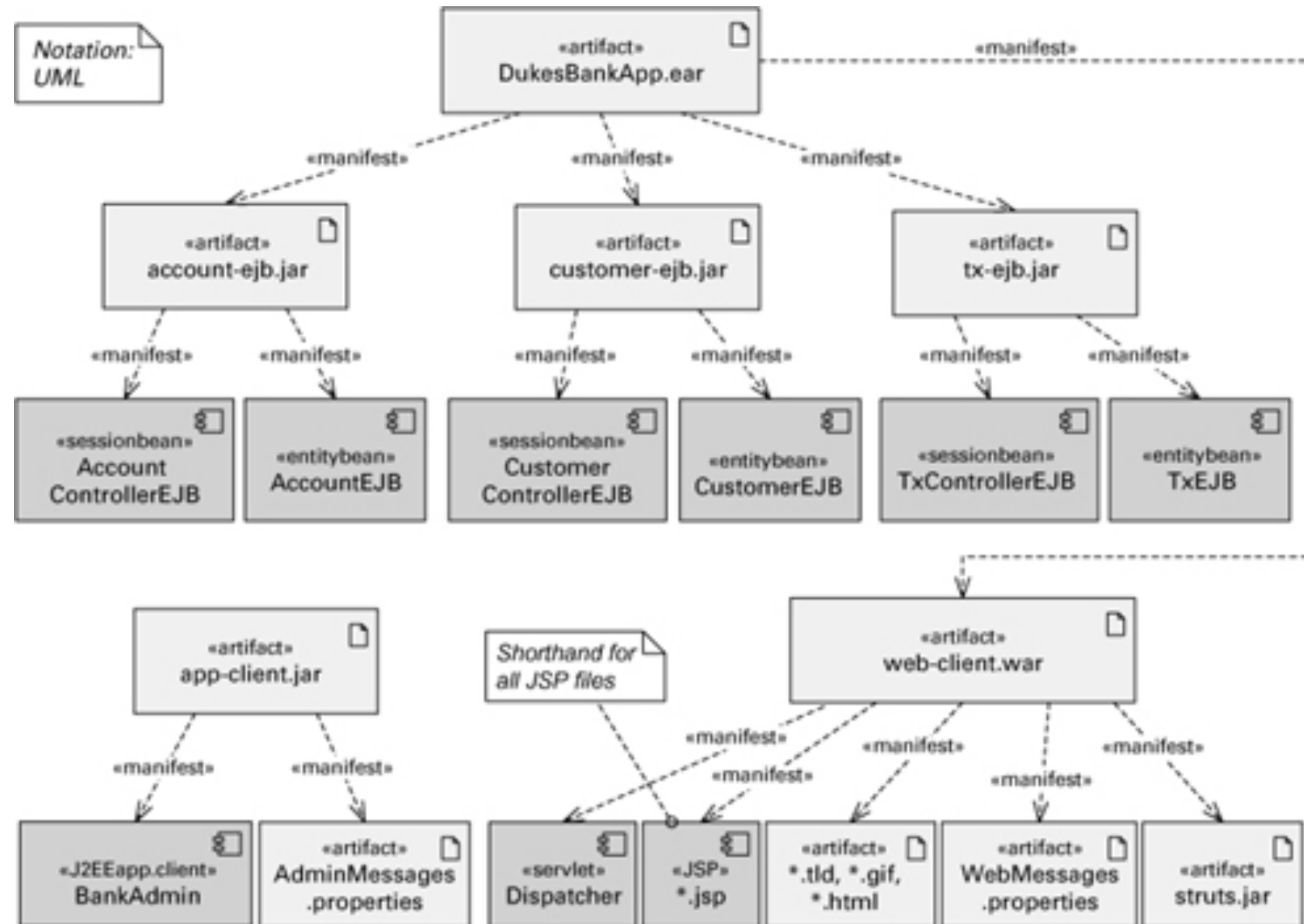
What the install style views are used for

- create build-and-deploy procedure
- navigate through large number of files and folders that constitute the installation
- select and configure installation variant
- update and configure files of multiple installed versions
- design and implement automatic update features

Notation - informal



Notation - semi-formal



Work Assignment Style

Overview	The work assignment style describes the mapping of the software architecture to the teams in the development organization.
Elements	<ul style="list-style-type: none">• <i>Software element: a module.</i> Properties include the required skill set and available capacity (effort, time) needed.• <i>Environmental element: an organizational unit, such as a person, a team, a department, a subcontractor, and so on.</i> Properties include the provided skill set and the capacity in terms of labor and calendar time available.
Relations	<i>Allocated-to.</i> A software element is allocated to an organizational unit.
Constraints	In general, the allocation is unrestricted; in practice, it is usually restricted so that one module is allocated to one organizational unit.

What the work assignment style is used for

- who will produce / procure which (major) module
- tools used in production
- environments for production

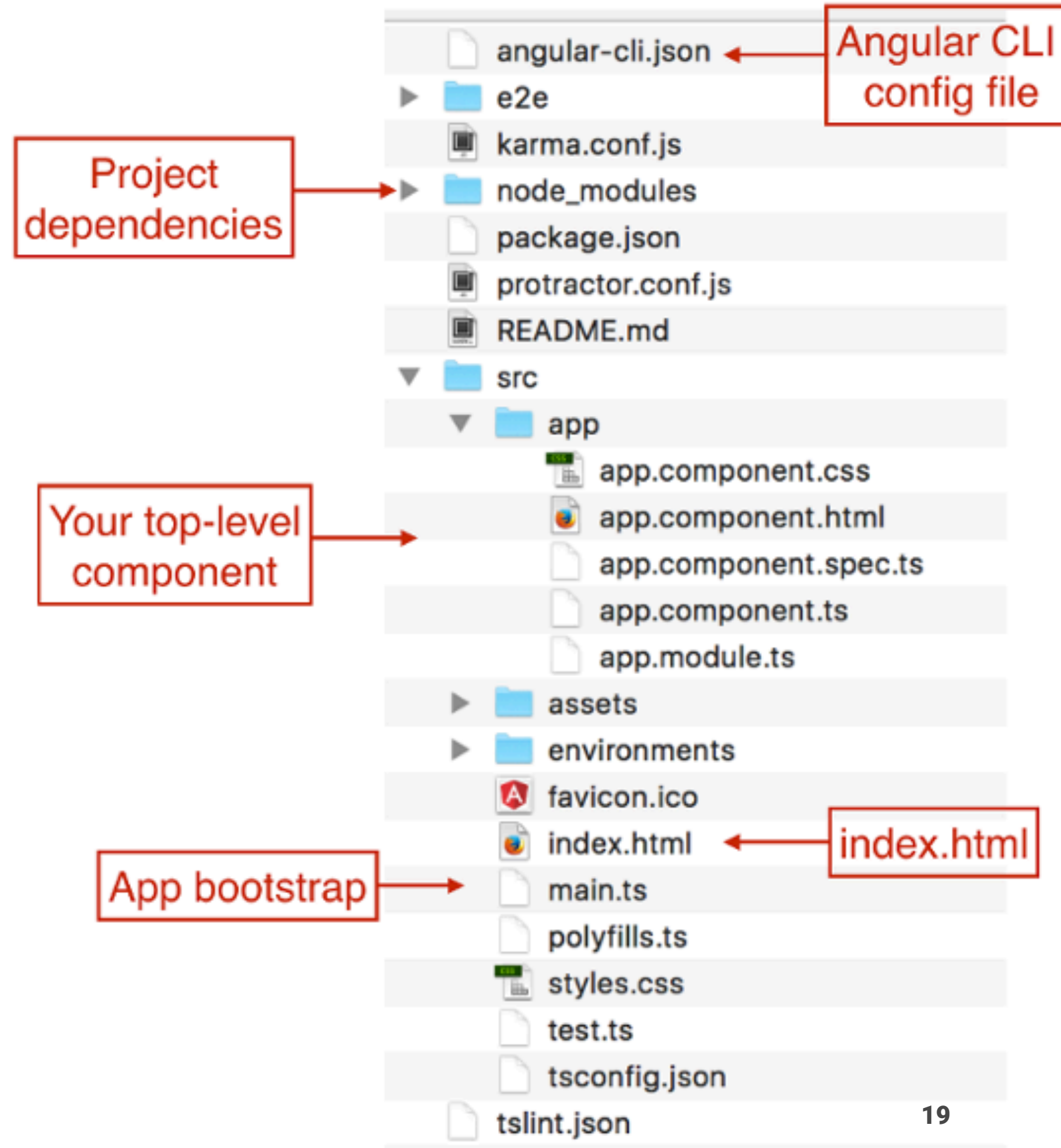
Notation

ECS Element (Module)		Organizational Unit
Segment	Subsystem	
Science Data Processing Segment (SDPS)	Client	Science team
	Interoperability	Prime contractor team 1
	Ingest	Prime contractor team 2
	Data Management	Data team
	Data Processing	Data team
	Data Server	Data team
	Planning	Orbital vehicle team
Flight Operations Segment (FOS)	Planning and Scheduling	Orbital vehicle team
	Data Management	Database team
	User Interface	User interface team
...

Other Allocation Styles

Implementation Style

- describes how development environment is organized (think "directory tree")



Other Allocation Styles

System Requirements Allocation

In embedded systems, system requirements may be allocated to software and hardware units.

<input type="checkbox"/> Use cases <input type="checkbox"/> Allocation <input checked="" type="checkbox"/> UC & Allocation		Use Cases			Allocation			
Req. ID	Req. Text	Use Case - Withdraw Cash	Use Case - Get Account Statement	Use Case - Authenticate Customer	MainframeInterface	TransactionController	CardReader	OutputDevice
Services								
SR-27	The system shall provide a cash withdrawal service.	X						
SR-55	The system shall, on request, display account statements on its screen.		X					
SR-57	The system shall, on request, provide account statements on paper printouts.		X					
Security								
SR-60	The system shall request a magnetic stripe card for user authentication.			X				
SR-61	The system shall capture the magnetic stripe card after three consecutive failed authentication attempts.			X				
SR-49	The system shall allow user to verify their identity using a 6 digit PIN code.			X				
Non-Functional Requirements								
SR-67	The system shall have a 640x480 pixel screen.							X
SR-68	The system shall enable replacement of any malfunctioning subsystem within 15 min.				X	X	X	X

Summary

- **Allocation Styles** map SW elements to elements in the environment
 - A **deployment view** maps runtime SW units to execution hardware
 - An **install view** maps SW artifacts to files and folders in the **production environment**
 - A **work assignment view** maps SW modules onto developers/teams/organizations
 - other allocation views and specializations