

## Architetture Software

# Descrizioni architettrurali, punti di vista e viste

**Dispensa ASW 140**  
ottobre 2014

*Se non è stato descritto,  
allora non esiste.*

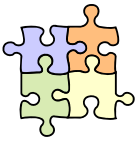
*Philippe Kruchten*



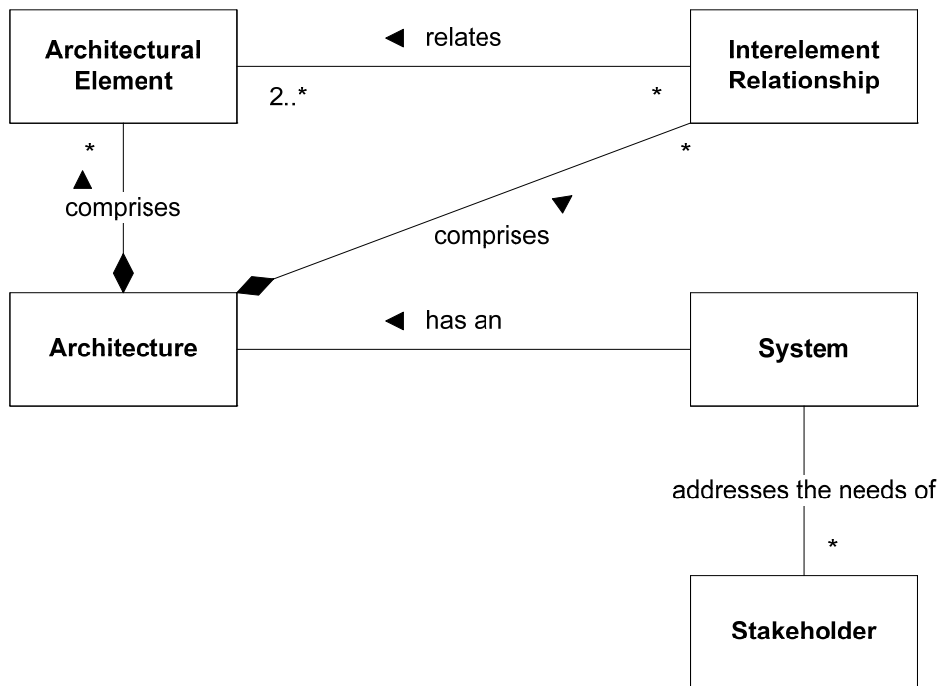
### - Fonti

- [SAP] Chapter 1, What is Software Architecture?
- [SAP] Chapter 18, Documenting Software Architectures
- [SSA] Chapter 2, Software Architecture Concepts
- [SSA] Chapter 3, Viewpoints and Views
  
- [Kruchten] Kruchten, Architectural Blueprints – The “4+1” View Model of Software Architecture, IEEE Software, 1995
  
- [PSA] Eeles & Cripps, The Process of Software Architecting
- [DSA] Clements et al., Documenting Software Architectures (SEI)
- [ISO-42010] Systems and Software Engineering – Architecture Description, ISO/IEC/IEEE, 2011





## \* Architetture sw: concetti fondamentali



## \* Descrizioni architetture

- Il processo di definizione dell'architettura di un sistema software ha, tra i suoi obiettivi, quello di realizzare un'opportuna "descrizione architetture" dell'architettura progettata
- Una **descrizione architetture (AD)** [ISO-42010] è un insieme di prodotti che documentano un'architettura
  - prodotti che costituiscono un'AD comprendono modelli architetture (viste) – ma anche definizione della portata, vincoli, principi, giustificazione logica, ...
- Una **descrizione architetture (AD)** [SSA] è un insieme di prodotti che documentano un'architettura
  - in un modo comprensibile alle parti interessate e
  - che dimostra che l'architettura soddisfa i loro interessi



## Descrizioni architetture

- Queste definizioni del termine AD sembrano enfatizzare la “descrizione” o “documentazione” di un’architettura
  - tuttavia, molto più rilevante è l’attività di “definizione” (ovvero, di progettazione) dell’architettura
    - durante la definizione di un’architettura è indispensabile utilizzare una buona descrizione – insieme a una buona “notazione” – poiché essa consente di concentrarsi e ragionare più facilmente su attività, problemi e decisioni di progetto significative
  - rilevante è anche il supporto che l’AD offre alla comunicazione con le parti interessate
  - in ogni caso, è spesso necessario (perché effettivamente utile) “documentare” l’architettura progettata



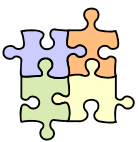
## Descrizioni architetture

- In pratica, una buona descrizione architetture
  - è la base per l’analisi delle decisioni iniziali di progetto
  - sostiene la comunicazione con le parti interessate
  - è una guida per lo sviluppo e l’evoluzione del sistema



## \* Organizzare una descrizione architeturale

- Alcune domande comuni nella progettazione dell'architettura software di un sistema
  - quali sono i principali elementi funzionali?
  - come interagiscono questi elementi tra loro e con il mondo esterno?
  - come vengono gestite, memorizzate e presentate le informazioni?
  - quali elementi hardware e software sono necessari a supportare questi elementi funzionali e queste informazioni?
  - come vengono allocati gli elementi software a quelli hardware?
  - quali caratteristiche e capacità operative devono essere fornite?
  - quali ambienti di sviluppo, test, supporto e formazione?
  - come organizzare il codice? chi sviluppa che cosa?



## \* Organizzare una descrizione architeturale

- Alcune domande comuni nella progettazione dell'architettura software di un sistema
  - quali sono i principali elementi funzionali?
  - come interagiscono questi elementi tra loro e con il mondo esterno?
  - come vengono gestite, memorizzate e presentate le informazioni?
  - quali elementi hardware e software sono necessari a supportare questi elementi funzionali e queste informazioni?
  - come vengono allocati gli elementi software a quelli hardware?
  - quali caratteristiche e capacità operative devono essere fornite?
  - quali ambienti di sviluppo, test, supporto e formazione?
  - come organizzare il codice? chi sviluppa che cosa?

Una tentazione a cui resistere:  
rispondere a tutte queste domande  
mediante un singolo modello,  
sovraccarico, che considera insieme tutti  
questi aspetti.



## Non un singolo modello ma tante viste

- Una descrizione architetturale deve cogliere le caratteristiche funzionali e le proprietà di qualità del sistema – e deve essere comprensibile e di valore a tutte le parti interessate
  - in pratica, l'esperienza dimostra che non è possibile ragionare su un'architettura software mediante un "singolo modello"
  - piuttosto, l'architettura di un sistema complesso può essere descritta in modo molto più efficace tramite un insieme di *viste*, separate ma correlate, ciascuna delle quali descrive un aspetto diverso dell'architettura
  - collettivamente, le viste descrivono l'intero sistema – le sue caratteristiche funzionali e proprietà di qualità – e dimostrano come il sistema può raggiungere i suoi obiettivi
  - questo è conforme con il fatto che un'architettura software è composta da un insieme di strutture – ciascuna vista ha lo scopo di descrivere una di queste strutture



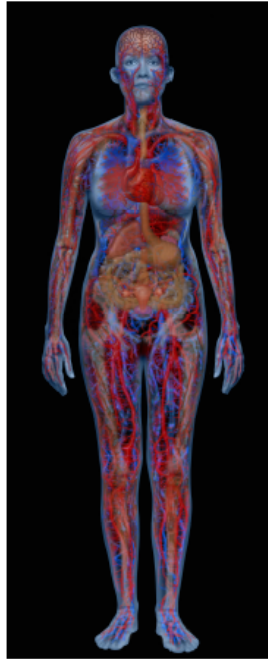
## Non un singolo modello ma tante viste

- Una descrizione architetturale deve cogliere le caratteristiche funzionali e le proprietà di qualità del sistema – e deve essere comprensibile e di valore a tutte le parti interessate
    - in pratica, l'esperienza dimostra che non è possibile ragionare su un'architettura software mediante un "singolo modello"
    - piuttosto, l'architettura di un sistema complesso può essere descritta in modo molto più efficace tramite un insieme di *viste*, separate ma correlate, ciascuna delle quali descrive un aspetto diverso dell'architettura
- Qui il suggerimento è di operare una decomposizione di un sistema complesso in un insieme di viste separate. Poiché bisogna gestire la complessità, vanno prese in considerazione in modo opportuno anche le correlazioni tra le parti – le viste devono essere correlate.



## Un'analogia

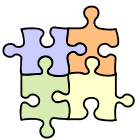
- Medici specialisti diversi sono interessati a viste differenti del corpo umano – che, pur distinte, sono inerentemente correlate



13

Descrizioni architettoniche, punti di vista e viste

Luca Cabibbo – ASw



## \* Viste architettoniche

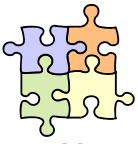
- Un'AD è composta da un insieme di *viste*, separate ma correlate
  - ciascuna *vista architettonica* ha lo scopo di descrivere *un insieme di elementi* del sistema e di *relazioni* tra di essi *che sono rilevanti rispetto a un certo interesse* (o insieme di interessi)
    - la definizione di ciascuna vista è centrata su uno specifico insieme di interessi – e non su uno specifico tipo di elementi
    - ciascuna vista descriverà (solo) gli elementi dell'architettura che sono rilevanti nei confronti di quegli interessi
  - in generale, ciascuna vista comprende uno o più modelli/diagrammi
- Ad es., una “vista funzionale”
  - può essere guidata da interessi funzionali – per questo, sarà composta da elementi che hanno responsabilità funzionali
  - non si occuperà, però, di disponibilità o di scalabilità

14

Descrizioni architettoniche, punti di vista e viste

Luca Cabibbo – ASw





## Viste architeturali



[SAP]

- una **vista** è una rappresentazione di un insieme coeso di elementi architeturali, che viene scritta e letta da parti interessate al sistema – una vista è la rappresentazione di un insieme di elementi e delle relazioni tra di essi



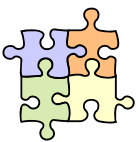
[ISO-42010]

- una **vista architettuale** è un prodotto che rappresenta l'architettura di un sistema secondo la prospettiva di specifici interessi del sistema



[SSA]

- una **vista** è una rappresentazione di uno o più aspetti strutturali di un'architettura, che illustra come l'architettura affronta uno o più interessi di una o più delle sue parti interessate

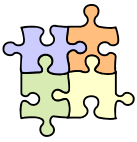


## Due questioni con le viste

### □ Due questioni (per ora) con le viste

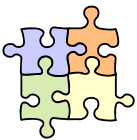
- quali viste creare in una descrizione architettuale?
  - questo dipende fortemente dal sistema – intuitivamente, quelle che consentono di descrivere tutti gli interessi rilevanti delle parti interessate a quello specifico sistema
  - ma quante? e quali? una per parte interessata? una per interesse? altro?
- come creare una vista architettuale?
  - intuitivamente, dipende da quali sono gli interessi rilevanti per la vista e dalle parti interessate a cui è destinata
  - ma quali elementi includere? a quale livello di dettaglio? come sostenere la comprensione da parti interessate diverse, ad es. che hanno un livello tecnico di comprensione differente? come sostenere l'analisi delle qualità?
- una risposta a queste domande è fornita dai punti di vista






## \* Punti di vista (e cataloghi di punti di vista)


- Un **punto di vista** è una tipologia standard di vista che può essere usata in un'AD
  - la scelta delle viste da produrre per un sistema viene spesso effettuata come una selezione da un **catalogo di punti di vista** – anziché decidere quali viste creare sulla base di “principi primi”
  - questa scelta viene fatta in base agli interessi rilevanti per un sistema e agli interessi che ciascun punto di vista è in grado di affrontare
  
- Si tratta di un'applicazione dei pattern alle AD
  - orientata al riuso di conoscenza relativa alla “soluzione esemplare di problemi significativi e ricorrenti”



## Punti di vista

 [ISO-42010]

- un **punto di vista architettuale** è una specifica delle convenzioni per la costruzione, l'interpretazione e l'uso di viste architetture per inquadrare/elaborare specifici interessi del sistema

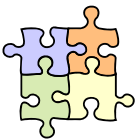
 [SSA] (adattata)

- un **punto di vista** è una collezione di pattern, template e convenzioni per costruire un tipo di vista
- un punto di vista definisce un insieme di interessi rilevanti per le parti interessate – nonché i principi, i modelli e le linee guida necessari per costruire viste di quel tipo



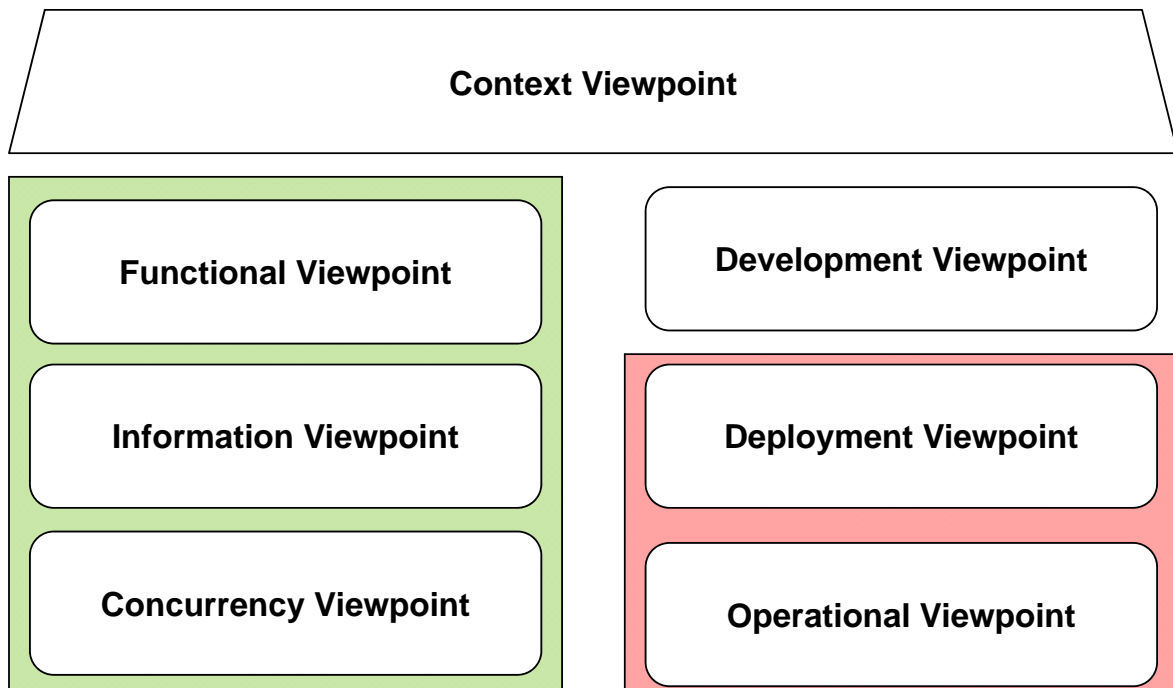
## Cataloghi di punti di vista

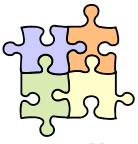
- Oggi esistono – e vengono adottati in pratica – diversi cataloghi punti di vista
  - da una parte, oggi non esiste nessun catalogo “standard” e “universale” di punti di vista
  - d'altra parte, alcuni punti di vista (e loro combinazioni) sono piuttosto diffusi
  - nel seguito saranno descritti alcuni cataloghi “moderni” di punti di vista: quelli proposti da [SSA], da [PSA] e da [SAP/DSA]



## \* Punti di vista di [SSA]

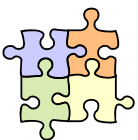
- [SSA] propone di organizzare un'AD sulla base di un catalogo di sette punti di vista





## Punti di vista di [SSA]

- Il catalogo di punti di vista di [SSA]
  - il punto di vista del Contesto descrive le relazioni, le dipendenze e le interazioni tra il sistema e il suo ambiente
  - i punti di vista Funzionale, delle Informazioni e della Concorrenza caratterizzano l'organizzazione fondamentale del sistema
  - il punto di vista dello Sviluppo ha lo scopo di sostenere la costruzione del sistema
  - i punti di vista di Deployment (rilascio) e Operazionale (di gestione) caratterizzano l'ambiente di utilizzo del sistema
- Caratteristiche e uso
  - (abbastanza) indipendenti
  - per descrivere un sistema, alcune viste saranno più importanti di altre – dipende dal sistema!
  - non tutti i sistemi richiedono tutte le viste



## Punti di vista di [SSA]

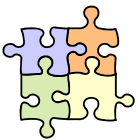
- In [SSA], ciascun punto di vista è definito soprattutto in termini di
  - quali sono i principali interessi affrontati da quel punto di vista
  - quali i modelli che possono essere utilizzati – e quali elementi e relazioni vi possono comparire
  - attività e linee guida per la realizzazione di tali modelli
  - alcune situazioni problematiche comuni
  - applicabilità e parti interessate
  - bibliografia



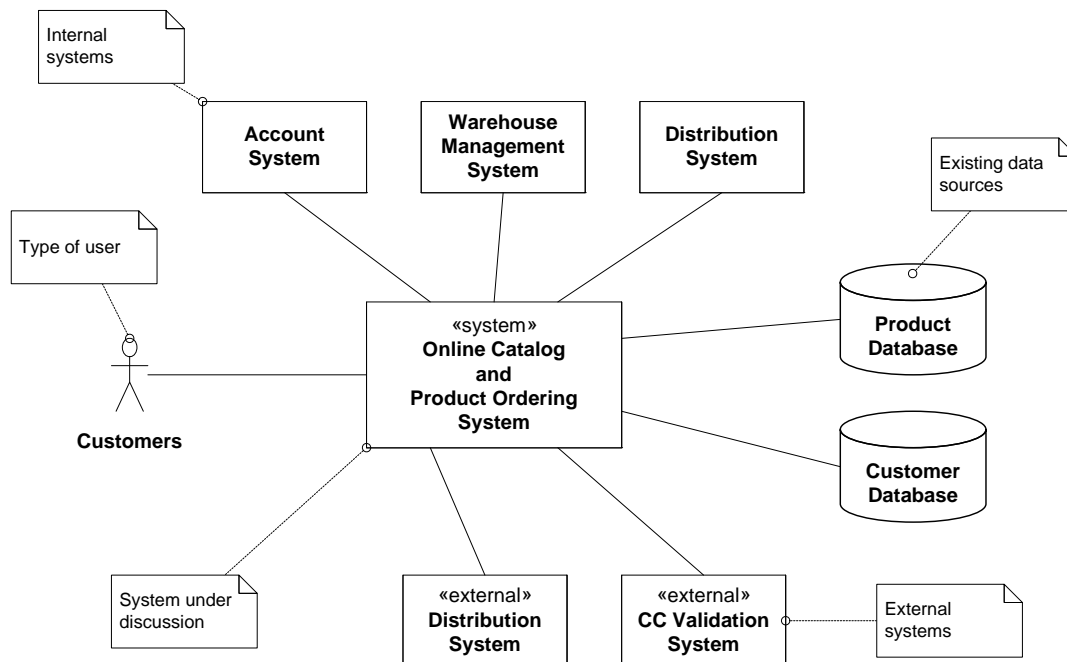
## I punti di vista di [SSA]

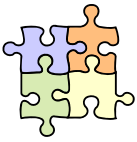
### ▣ Punto di vista del *Contesto*

- ▣ descrive le relazioni, le dipendenze e le interazioni tra il sistema e il suo ambiente (le persone, i sistemi e le entità esterne con cui interagisce)
- ▣ la vista del Contesto svolge un ruolo importante perché aiuta le parti interessate a comprendere la portata e le responsabilità del sistema, e come esso si relaziona con l'organizzazione
- ▣ i principali interessi affrontati da questo punto di vista sono proprio portata e responsabilità del sistema



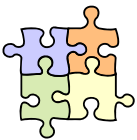
## Esempio (parziale) di diagramma di contesto



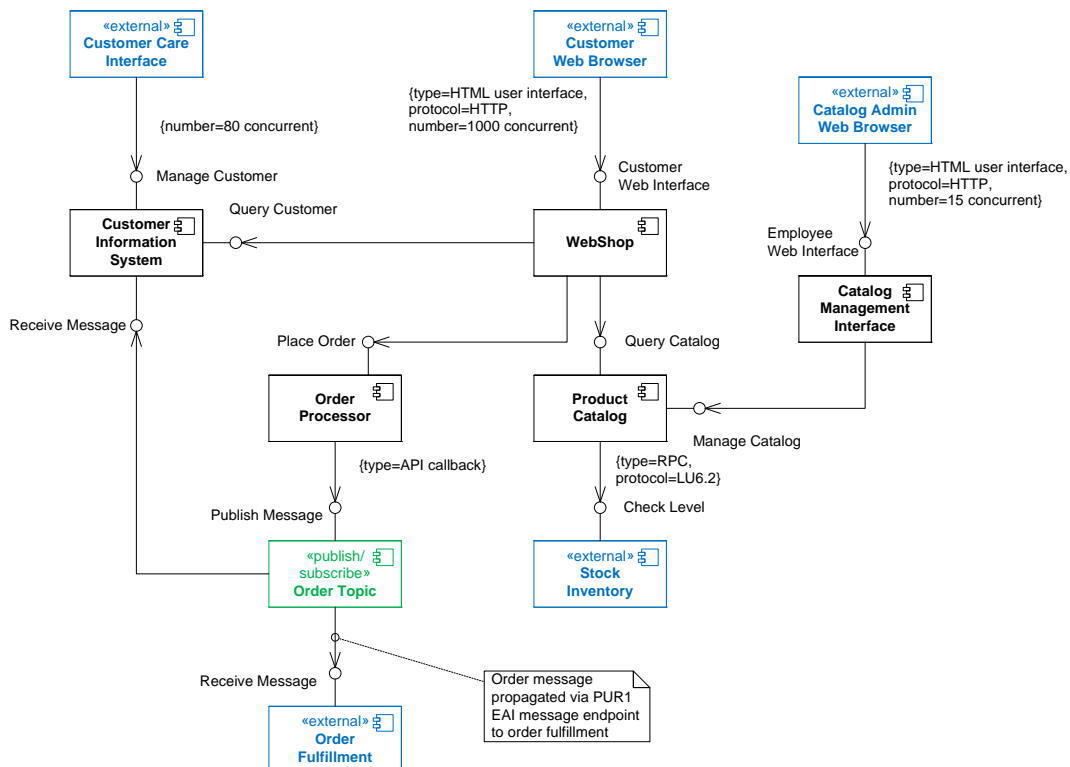


# I punti di vista di [SSA]

- Punto di vista *Funzionale*
  - descrive gli elementi funzionali del sistema, le loro responsabilità, interfacce e interazioni principali
    - pilastro di molti AD
    - guida la forma di altre strutture e viste
    - impatto significativo su alcune qualità del sistema – modificabilità, sicurezza, prestazioni, ...
- La vista funzionale è “un modello del progetto... ispirato dal dominio... che sostiene i requisiti funzionali”
  - quest’affermazione enfatizza rilevanza delle decomposizioni basate sul dominio del problema
  - la modellazione del dominio da effettuare – ad esempio, modellazione delle informazioni, delle attività e/o dei casi d’uso – dipende dal tipo di architettura che si intende realizzare – ad esempio, architetture a componenti o a servizi



# Esempio (parziale) di vista funzionale



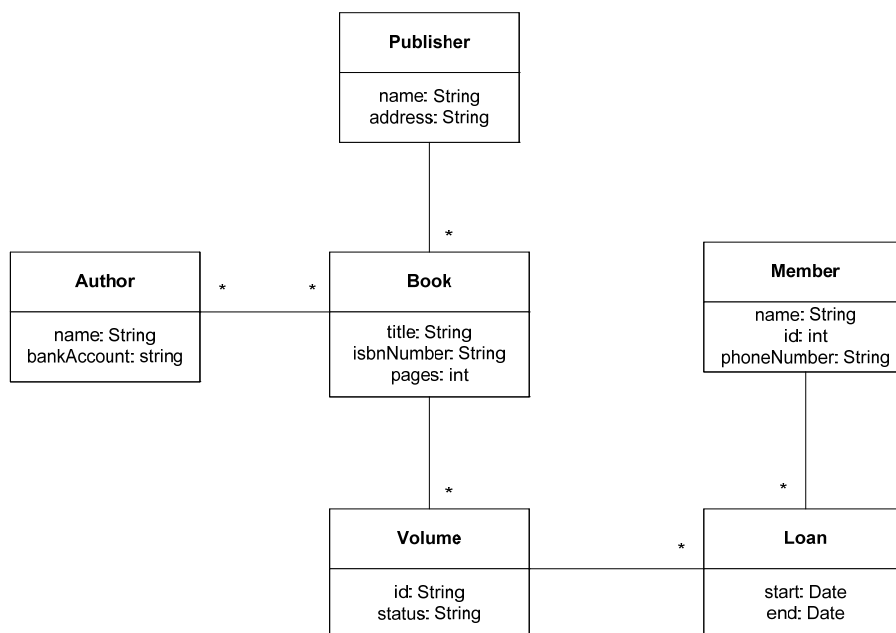


## I punti di vista di [SSA]

- Punto di vista delle **Informazioni**
  - descrive il modo in cui l'architettura memorizza, manipola, gestisce e distribuisce informazioni – in termini di strutture di dati statiche e di flussi di informazioni
  - importante perché lo scopo dei sistemi informatici è gestire informazioni



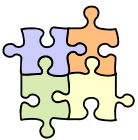
## Esempio (parziale) di vista delle informazioni



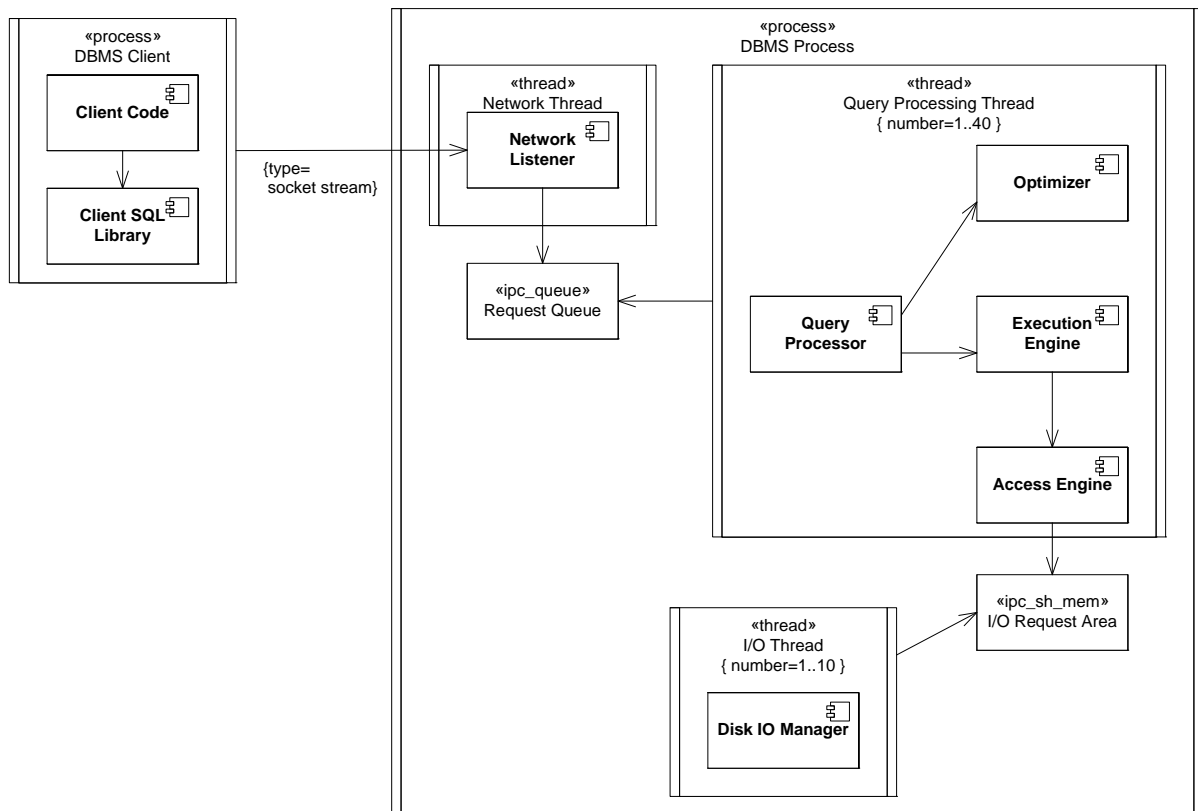


# I punti di vista di [SSA]

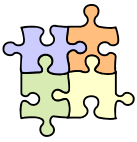
- ▢ Punto di vista della **Concorrenza**
  - descrive l'organizzazione della concorrenza e mappa gli elementi funzionali su unità di concorrenza, nonché le parti concorrenti del sistema e le loro necessità e modalità di sincronizzazione
  - struttura di processi e thread e meccanismi di comunicazione interprocesso



# Esempio (parziale) di vista della concorrenza

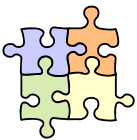




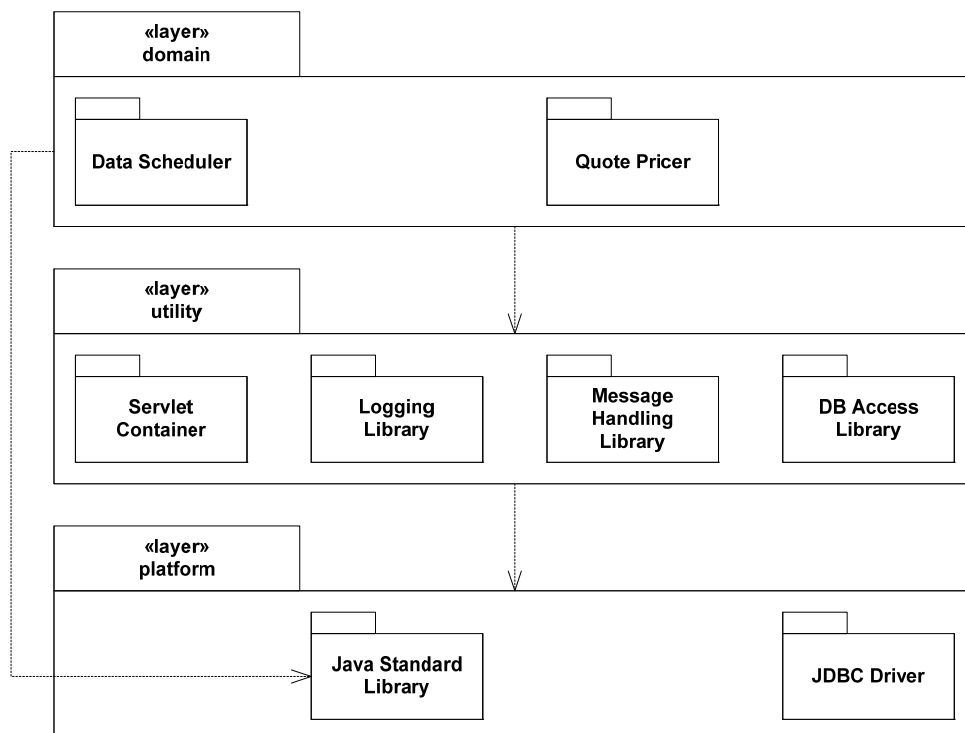


# I punti di vista di [SSA]

- ▣ Punto di vista dello *Sviluppo*
  - descrive l'architettura che supporta il processo di sviluppo
    - ad es., organizzazione del codice, dei moduli, dei test
    - di interesse per chi sviluppa, verifica, mantiene e migliora il sistema – e per chi deve gestire tali persone



# Esempio (parziale) di vista dello sviluppo

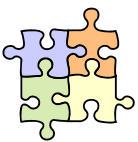




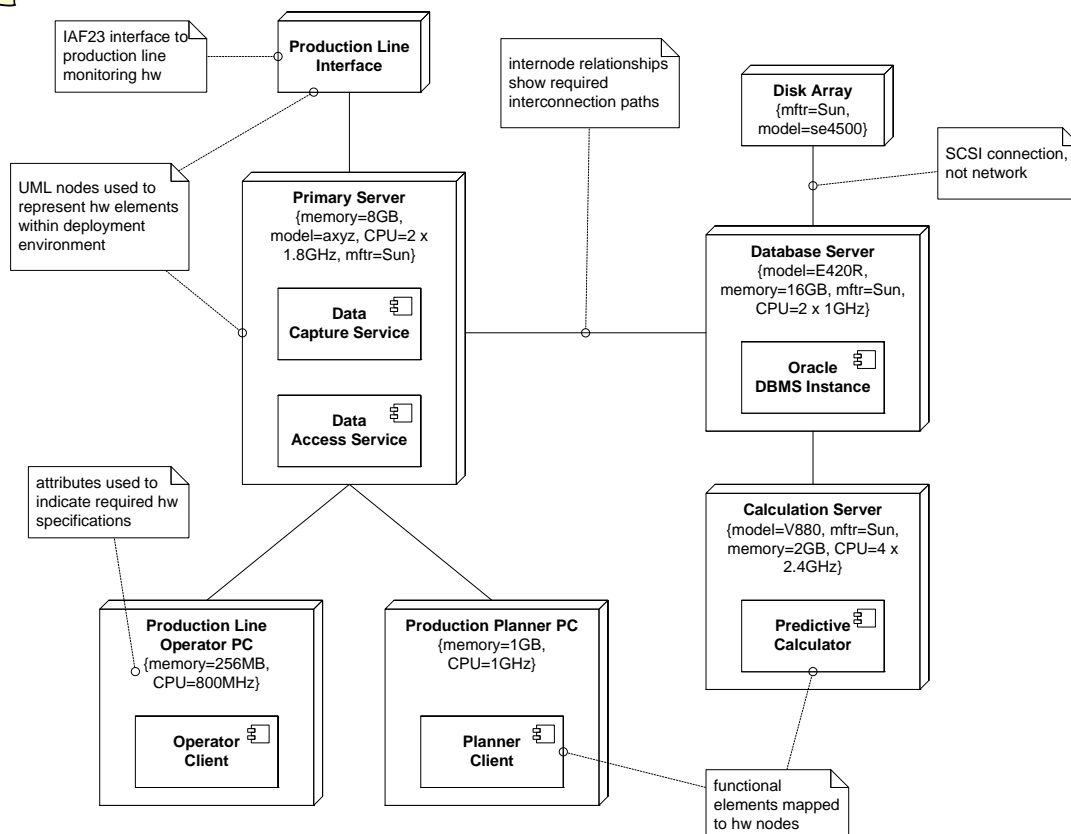
# I punti di vista di [SSA]

## ▣ Punto di vista del *Deployment*

- ▣ descrive l'ambiente in cui il sistema sarà rilasciato, comprese le dipendenze dall'ambiente runtime
  - ▣ elementi hardware (computer, dischi, reti, ...), requisiti per tali elementi, corrispondenze tra elementi software e l'ambiente di esecuzione



# Esempio (parziale) di vista di deployment





## I punti di vista di [SSA]

- Punto di vista *Operazionale*
  - descrive come il sistema sarà usato, amministrato e supportato quando sarà in esecuzione nell'ambiente di produzione
    - per affrontare interessi relativi alla gestione del sistema – ad es., installazione, aggiornamento, monitoraggio, gestione delle configurazioni, ...



## \* Benefici nell'uso di punti di vista e viste

- *Separazione degli interessi* – *separation of concerns*
  - la separazione degli interessi è un principio di progettazione fondamentale
    - separa interessi distinti in aree diverse, in modo che ciascuna area abbia uno scopo coeso
    - dividi il progetto in caratteristiche che si sovrappongono il meno possibile
  - la modellazione di un sistema con descrizioni separate ma correlate favorisce i processi di analisi, progettazione e comunicazione – poiché permette di concentrarsi separatamente su ciascun interesse



## Benefici nell'uso di punti di vista e viste

- Gestione della complessità
  - gli interessi possono essere gestiti separatamente
- Miglioramento dell'attenzione/focalizzazione dello sviluppatore
  - l'AD contiene non solo modelli per comunicare con gli utenti o l'acquirente, ma anche modelli focalizzati sugli interessi degli sviluppatori
- Comunicazione con gruppi di parti interessate
  - ciascuna parte interessata è probabilmente interessata solo a un sottoinsieme dell'AD
  - bisogna ricordare che anche i team di sviluppo sono parti interessate – e che la comunicazione tra architetto e team di sviluppo è di fondamentale importanza per il successo di un'architettura



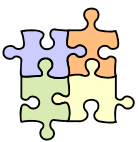
## \* Rischi legati alle viste

- I due problemi/rischi principali nell'uso delle viste
  - inconsistenza
    - l'uso di più viste solleva il problema della consistenza tra viste – che può essere mitigato applicando delle verifiche di mutua coerenza tra le viste – ad es., sulla base di una checklist di verifiche da effettuare
  - gestione di interessi “trasversali”
    - alcuni interessi possono essere effettivamente gestiti con riferimento (soprattutto) a una singola vista – ad es., la modificabilità
    - tuttavia, altre qualità possono corrispondere a interessi trasversali alle varie viste – ad es., la sicurezza
    - come è possibile controllare ed ottenere queste qualità, in modo efficace e coerente?



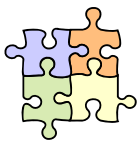
## Rischi legati alle viste

- Ulteriori possibili problemi/rischi nell'uso delle viste
  - frammentazione
    - l'uso di un numero eccessivo di viste può complicare la comprensione e l'analisi dell'AD
    - pertanto è meglio concentrarsi su viste che affrontano solo interessi veramente importanti
    - per ridurre il numero di vista, talvolta può essere accettabile avere viste "ibride"
  - selezione di un insieme sbagliato di viste
    - non sempre è ovvio capire quante e quali viste usare per descrivere un certo sistema



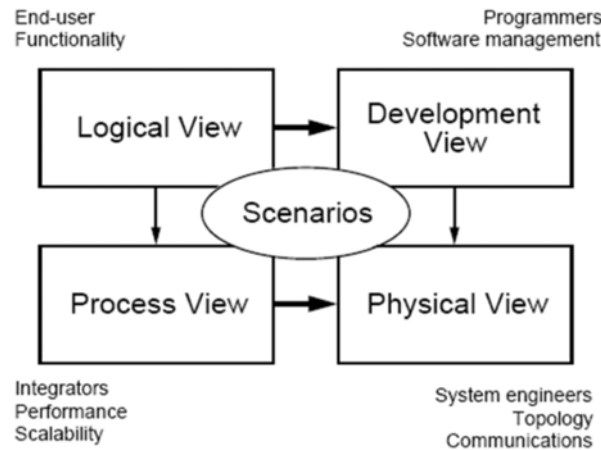
## Applicazione di scenari

- La definizione di un'architettura basata su viste multiple richiede l'adozione di metodi di progettazione compatibili con l'uso di più viste e con la presenza di qualità trasversali
  - un approccio molto valido è l'applicazione di scenari
  - in generale, per **scenario** si intende
    - una *situazione* che il sistema dovrà probabilmente affrontare nel suo ambiente di produzione,
    - insieme a una definizione della *risposta* richiesta dal sistema
  - intuitivamente, nell'applicazione di uno scenario vanno progettate/descritte le interazioni tra gli elementi architetture presenti nelle diverse viste
    - la "lettura" congiunta delle interazioni motivate da ciascuno scenario fornisce una visione complessiva sull'intero sistema
  - vedere le dispense asw160 su Ottenere qualità e asw250 su Interessi, requisiti e scenari



## \* Il modello a 4+1 viste

- Questo celebre articolo propone l'uso di viste e punti di vista nelle descrizioni architeturali
  - P. Kruchten – *Architectural Blueprints – The “4+1” View Model of Software Architecture* – IEEE Software, 1995
  - questo articolo definisce un primo catalogo di punti di vista – e propone l'applicazione di scenari (la vista “+1”) per correlare le altre viste



41

Descrizioni architeturali, punti di vista e viste

Luca Cabibbo – ASw



## \* Punti di vista di [PSA]



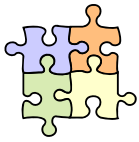
- [PSA] introduce un catalogo basato su due tipi di punti di vista
  - **basic viewpoints** – corrispondono, più o meno, ai punti di vista di [SSA] o a quelli del modello a 4+1 viste
  - **cross-cutting viewpoints** – casi specializzati della vista a scenari – corrispondono, più o meno, alle prospettive di [SSA]

		Basic Viewpoints			
		Requirements Viewpoint	Functional Viewpoint	Deployment Viewpoint	Validation Viewpoint
Cross-Cutting Viewpoints	Application Viewpoint				
	Infrastructure Viewpoint				
	System Mgmt Viewpoint				
	Availability Viewpoint				
	Performance Viewpoint				
	Security Viewpoint				

42

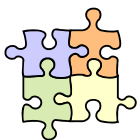
Descrizioni architeturali, punti di vista e viste

Luca Cabibbo – ASw



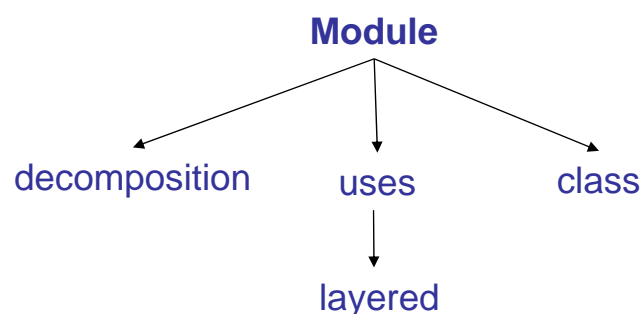
## \* Tipi di viste di [SAP/DSA]

- Secondo [DSA] – dagli stessi autori di [SAP] – le viste architettrali possono essere generalmente suddivise in quattro categorie (*viewtypes*, *tipi di vista*), sulla base della natura degli elementi che contengono oppure degli interessi che affrontano
  - viste a moduli
    - i moduli sono unità di implementazione
  - viste a componenti e connettori
    - con riferimento alle unità di esecuzione
  - viste di allocazione
    - mostrano le relazioni tra elementi software ed il loro ambiente esterno
  - viste per qualità
    - per affrontare interessi specifici
  - diversamente dagli altri cataloghi, i nomi dati alle viste sono prevalentemente “sintattici” e non “semantici”

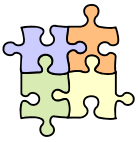


## Viste a moduli

- **Viste a moduli**
  - gli elementi sono *moduli* – unità di implementazione (codice)
    - ai moduli vengono assegnate responsabilità funzionali
    - meno interesse sul comportamento al tempo di esecuzione
  - possibili relazioni tra moduli – ad es., usa, estende, dipende da, strati, ...

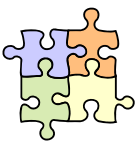






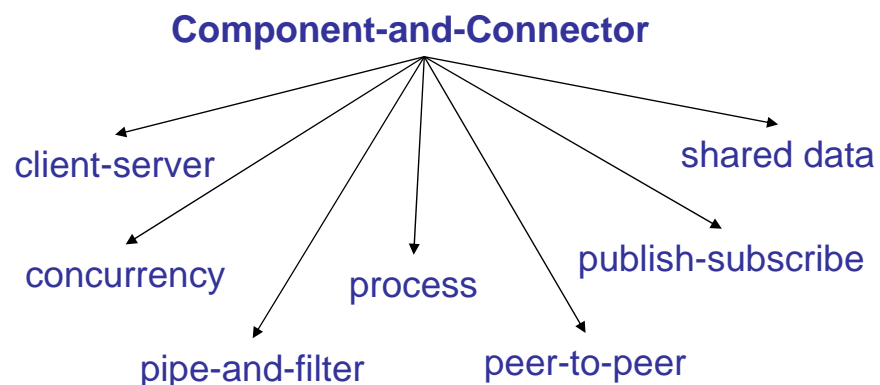
## Viste a moduli

- Le viste a moduli consentono di rispondere a domande come
  - quale è la responsabilità principale di un modulo?
  - quali altri moduli può usare un modulo?
  - quali altri moduli sono effettivamente usati da un modulo?
  - quali moduli specializzano altri moduli?



## Viste a componenti e connettori

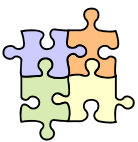
- **Viste a componenti e connettori**
  - gli elementi sono *componenti runtime* (unità di calcolo, processi) e *connettori* (meccanismi di interazione e comunicazione tra componenti)
  - punto di vista basato su processi/task/thread
  - relazioni – connessioni tra componenti e connettori – su porte





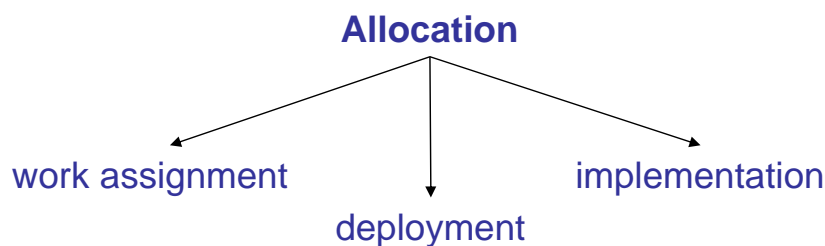
## Viste a componenti e connettori

- Le viste a componenti e connettori consentono di rispondere a domande come
  - quali sono i principali componenti in esecuzione? come interagiscono?
  - quali sono i dati condivisi?
  - quali parti del sistema sono replicate?
  - come si muovono i dati nel sistema?
  - quali parti del sistema sono eseguite in parallelo?
  - la struttura del sistema può cambiare durante l'esecuzione?



## Viste di allocazione

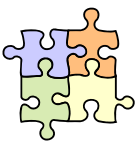
- **Viste di allocazione**
  - gli elementi sono sia *elementi software* (ad es., moduli) che *elementi in ambienti esterni* (in cui il software viene sviluppato o eseguito)
  - relazioni – di corrispondenza/allocazione





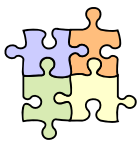
## Viste di allocazione

- Le viste di allocazione consentono di rispondere a domande come
  - su quale elemento hardware viene eseguito un componente software?
  - in quali file viene memorizzato un elemento – durante l'implementazione, il test e la costruzione del sistema?
  - quale è l'assegnazione di elementi software a team di sviluppo?



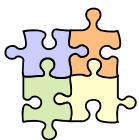
## Viste per qualità

- **Viste per qualità**
  - le viste a moduli, a componenti e connettori e di allocazione sono viste strutturali, e consentono di affrontare molti interessi
  - tuttavia, se certe proprietà di qualità sono pervasive o di particolare importanza in un sistema, le viste strutturali potrebbero essere inadeguate a ragionare in modo efficace su tali qualità – ad esempio, perché non è conveniente fare ragionamenti complessi su più viste
  - in tal caso, è possibile usare delle viste dedicate ad affrontare degli specifici interessi di qualità
    - ad esempio, una vista della sicurezza, delle prestazioni o della disponibilità
  - queste viste corrispondono, più o meno, ai **cross-cutting viewpoints** di [PSA] e alle prospettive di [SSA]



## \* Discussione

- Importanza di una buona descrizione architeturale esplicita
  - comunicazione con le parti interessate
  - base per l'analisi delle decisioni iniziali di progetto
  - guida allo sviluppo
- I sistemi software sono troppo complessi per poter essere descritti da un singolo diagramma
  - piuttosto, una descrizione architeturale è composta da più modelli o viste
  - ciascuna vista affronta alcuni interessi del sistema
  - la creazione delle viste può essere guidata da un opportuno catalogo di punti di vista



## Relazioni tra concetti fondamentali

