



Luca Cabibbo  
Architettura  
dei Sistemi  
Software

# Viste e punti di vista architettonici

**dispensa asw130**  
ottobre 2024

*These pictures  
are meant to entertain you.  
There is no significant meaning  
to the arrows between the boxes.*

*A speaker at a  
software architecture conference*

1

Viste e punti di vista architettonici

Luca Cabibbo ASW



## - Riferimenti

- ❑ Luca Cabibbo. **Architettura del Software: Strutture e Qualità**. Edizioni Efestò, 2021.
  - Capitolo 3, **Viste e punti di vista architettonici**
- ❑ Kruchten, P. **The “4+1” View Model of Software Architecture**. IEEE Software, 1995.
- ❑ Clements et al. **Documenting Software Architectures: Views and Beyond**, second edition. SEI, 2010.
- ❑ ISO/IEC/IEEE 42010:2011. **Systems and Software Engineering – Architecture Description**. 2011.

2

Viste e punti di vista architettonici

Luca Cabibbo ASW



## - Obiettivi e argomenti

### □ Obiettivi

- introdurre le descrizioni architettoniche
- discutere viste e punti di vista architettonici
- illustrare alcuni cataloghi rappresentativi di punti di vista

### □ Argomenti

- descrizioni architettoniche
- viste architettoniche
- punti di vista (e cataloghi di punti di vista)
- un catalogo di punti di vista [SSA]
- benefici e rischi legati alle viste
- scenari e applicazione di scenari
- tipi di viste [SAP]
- discussione



## \* Descrizioni architettoniche

### □ Alcune attività rilevanti nella definizione (e nel ciclo di vita) dell'architettura

- progettare l'architettura
- ragionare sull'architettura
- comunicare l'architettura

### □ Queste attività possono essere sostenute da un'opportuna *modellazione* dell'architettura

- una *descrizione architettonica* è un insieme di modelli usati per *descrivere* un'architettura – ma che sostiene anche *ragionamenti e comunicazione*



## Descrizioni architetture

- Una **descrizione architetture (AD)** [SSA, ISO-42010] è un insieme di prodotti/elaborati che descrivono/documentano un'architettura
  - una AD comprende un insieme di modelli architetture (viste architetture) – ma descrive anche i principali interessi, la portata del sistema, i vincoli, i principi rilevanti, le scelte di progetto (comprese quelle considerate ma scartate), e fornisce una giustificazione logica dell'architettura



## Descrizioni architetture

- Una **buona AD** non ha solo lo scopo di “descrivere” o “documentare” un'architettura – ma favorisce anche la sua **comprensione** e la sua **comunicazione**
  - deve supportare ragionamenti sull'architettura
  - deve sostenere la comunicazione con le parti interessate
  - in ogni caso, “documentare” l'architettura progettata è spesso necessario (ed utile)



## - Organizzare una descrizione architettuale

- Una descrizione architettuale deve cogliere le caratteristiche funzionali e le proprietà di qualità del sistema – e deve essere comprensibile e di valore a tutte le parti interessate
  - è possibile/utile organizzare l'AD di un sistema complesso mediante un "singolo modello"?
  - un modo efficace di descrivere l'architettura di un sistema software è mediante un insieme di modelli chiamate *viste*, separate ma correlate
    - ciascuna vista descrive un aspetto diverso dell'architettura
    - collettivamente, le viste descrivono l'intero sistema – e dimostrano come il sistema può raggiungere i suoi obiettivi
    - questo è conforme con il fatto che un'architettura software è composta da più strutture

7

Viste e punti di vista architeturali

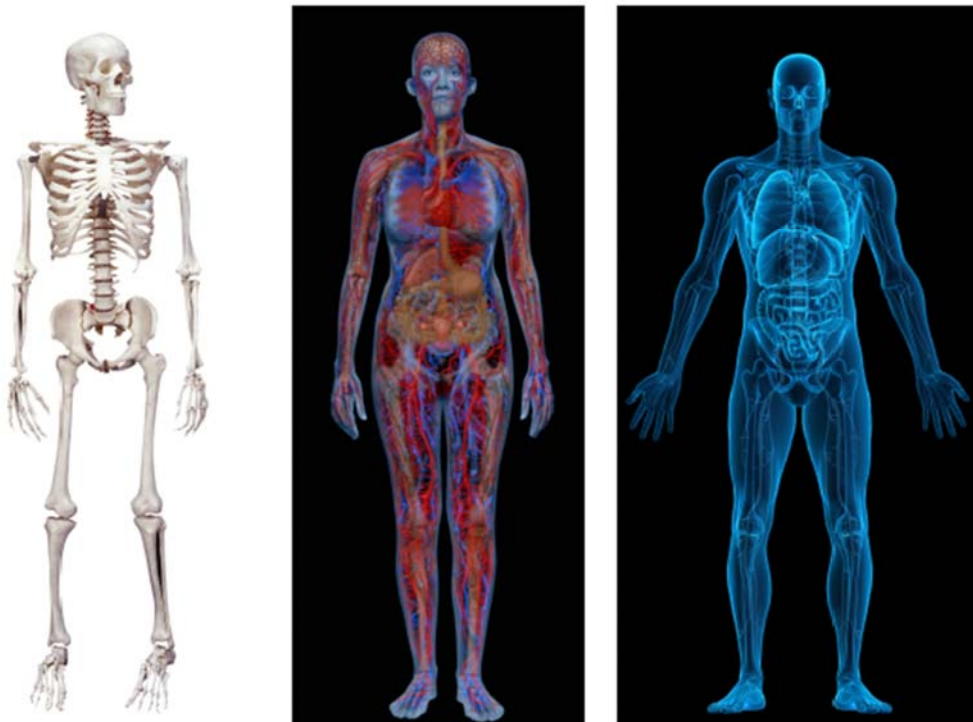
Luca Cabibbo ASW



## Un'analogia



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



8

Viste e punti di vista architeturali

Luca Cabibbo ASW



## \* Viste architettureali

- Una **vista architettureale** [ISO-42010, SSA, SAP] è un elaborato/modello che
  - rappresenta uno o più aspetti strutturali dell'architettura di un sistema
    - rappresenta un insieme coeso di elementi architettureali e le relazioni tra di essi
  - illustra come l'architettura affronta uno o più interessi di una o più delle sue parti interessate
    - rappresenta l'architettura secondo la prospettiva di specifici interessi del sistema



## Viste architettureali

- In pratica, ciascuna **vista** 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)
  - dunque, due aspetti principali nella definizione di ciascuna vista
    - un insieme di elementi e di relazioni tra elementi
    - uno specifico insieme di interessi
  - ciascuna vista descrive (tutti e soli) gli elementi dell'architettura che sono rilevanti nei confronti di quegli specifici interessi



## Due questioni con le viste

- Due questioni (per ora) con le viste architettoniche
  - quante e quali viste creare in una descrizione architettonica?
  - come creare ciascuna vista architettonica?
  - una risposta a queste domande è fornita dai punti di vista



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

- Un **punto di vista architettonico** [ISO-42010, SSA]
  - è una collezione di pattern, template e convenzioni per la costruzione, l'interpretazione e l'uso di viste architettoniche per inquadrare/elaborare specifici interessi del sistema
  - ciascun punto di vista fa riferimento a uno specifico insieme di interessi rilevanti per le parti interessate – e definisce i principi, i modelli e le linee guida necessari per costruire viste adatte ad affrontare tali interessi



# Cataloghi di punti di vista

- La scelta delle viste da produrre per descrivere un sistema viene di solito basata su un *catalogo di punti di vista* predefiniti
  - il primo catalogo di punti di viste è il modello a 4+1 viste, proposto nel 1995 da un celebre articolo di Kruchten – poi evoluto come modello a N+1 viste in RUP
  - oggi esistono diversi cataloghi di punti di vista – ma non c'è nessun catalogo “standard” e “universale” di punti di vista
    - in ogni caso, alcuni punti di vista sono piuttosto diffusi
  - descriviamo ora brevemente due cataloghi rappresentativi di punti di vista
    - i punti di vista di [SSA]
    - i tipi di vista di [SAP]

13

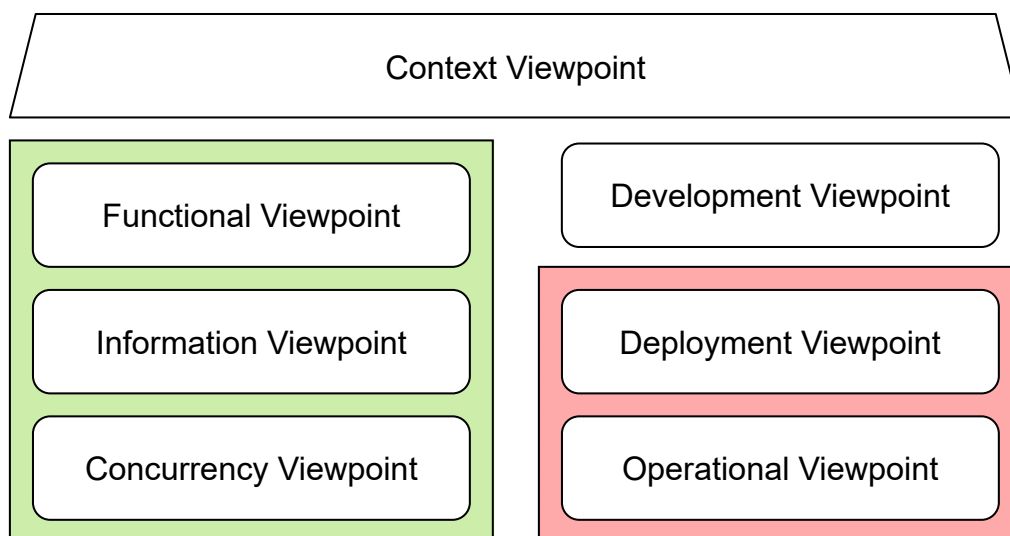
Viste e punti di vista architeturali

Luca Cabibbo ASW



## \* Un catalogo di punti di vista [SSA]

- [SSA] propone di organizzare una AD sulla base di un catalogo di sette punti di vista (nel seguito abbreviati in *pdv*)



14

Viste e punti di vista architeturali

Luca Cabibbo ASW



## Punti di vista di [SSA]

- Il catalogo di punti di vista di [SSA]
  - il pdv del Contesto descrive le relazioni, le dipendenze e le interazioni tra il sistema e il suo ambiente
  - i pdv Funzionale, delle Informazioni e della Concorrenza caratterizzano l'organizzazione fondamentale del sistema
  - il pdv dello Sviluppo ha lo scopo di sostenere la costruzione del sistema
  - i pdv di Deployment (rilascio) e Operational (“operations”) sono relativi all'ambiente di esecuzione e al rilascio del software



## Punti di vista di [SSA]

- Ciascun punto di vista di [SSA] è definito in termini di
  - i principali interessi affrontati da quel punto di vista
  - i modelli che possono essere utilizzati – 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





# Punto di vista del Contesto [SSA]

## □ Punto di vista del *Contesto*

- descrive le relazioni, le dipendenze e le interazioni tra il sistema (considerato a scatola nera) e il suo ambiente – le persone, i sistemi e le entità esterne con cui interagisce
- ha un ruolo importante nella comprensione della portata e delle responsabilità del sistema, e come esso si relaziona con il suo ambiente
- interessi principali: portata (scope) e responsabilità del sistema

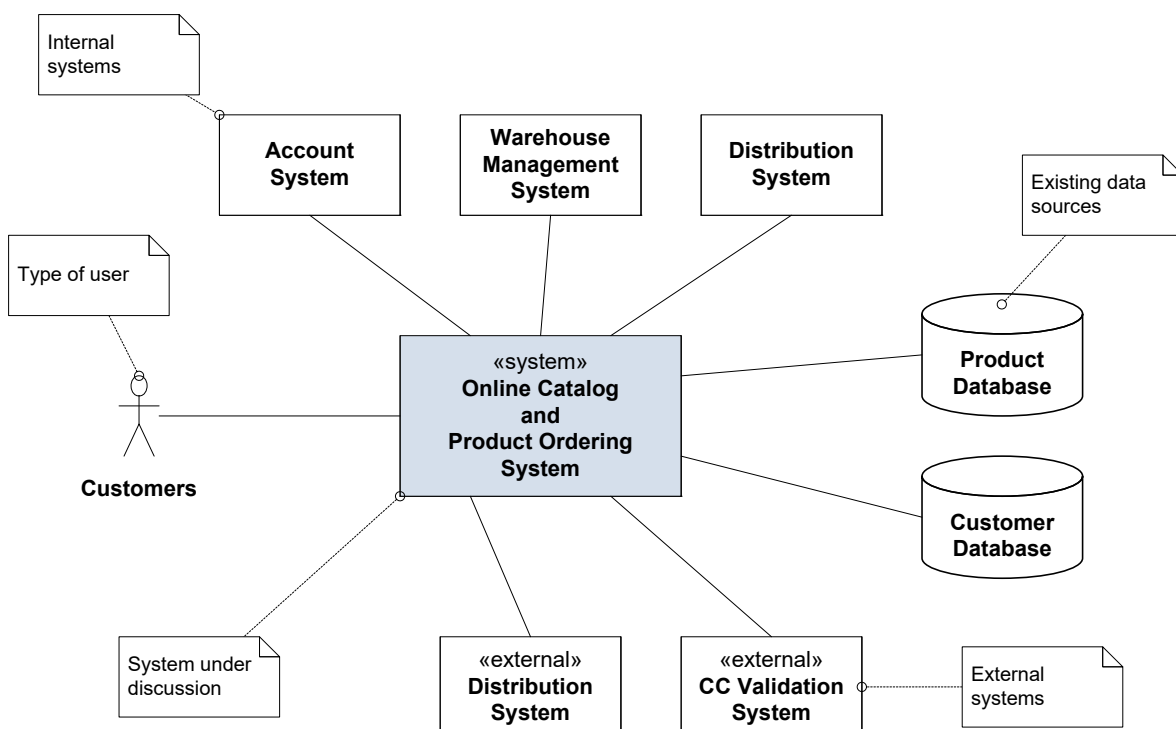
17

Viste e punti di vista architetturali

Luca Cabibbo ASW



# Esempio (parziale): diagramma di contesto



18

Viste e punti di vista architetturali

Luca Cabibbo ASW



# Punto di vista Funzionale [SSA]

## □ Punto di vista *Funzionale*

- descrive gli elementi funzionali del sistema, con le loro responsabilità, interfacce e interazioni principali
- complessivamente, descrive come il sistema può eseguire le funzionalità richieste
- la vista funzionale è la “pietra angolare” di molte AD – e può guidare la forma di altre strutture e viste
- interessi principali: capacità funzionali (funzionalità) del sistema e struttura funzionale interna del sistema
- mostra elementi software (componenti e connettori) a cui sono assegnate responsabilità funzionali – che sono spesso “ispirati” al dominio del sistema

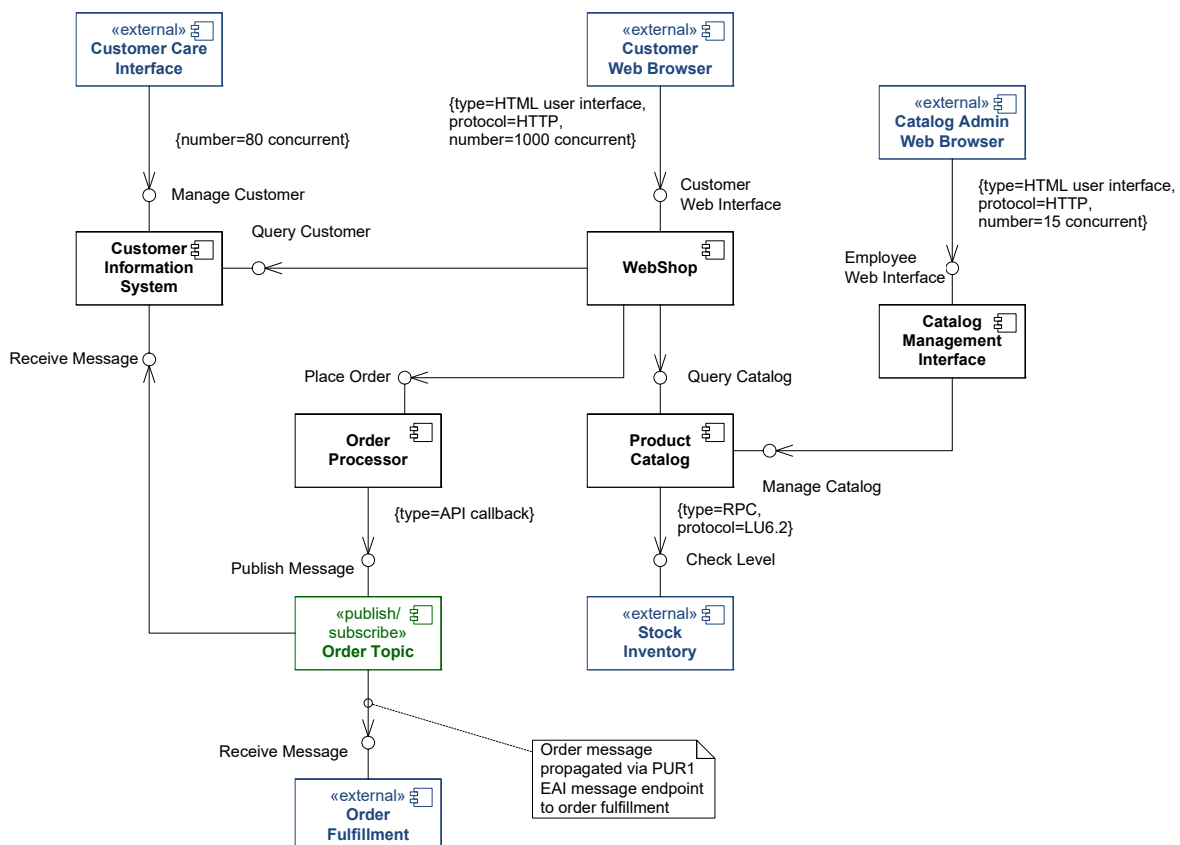
19

Viste e punti di vista architetturali

Luca Cabibbo ASW



# Esempio (parziale): vista funzionale



20

Viste e punti di vista architetturali

Luca Cabibbo ASW

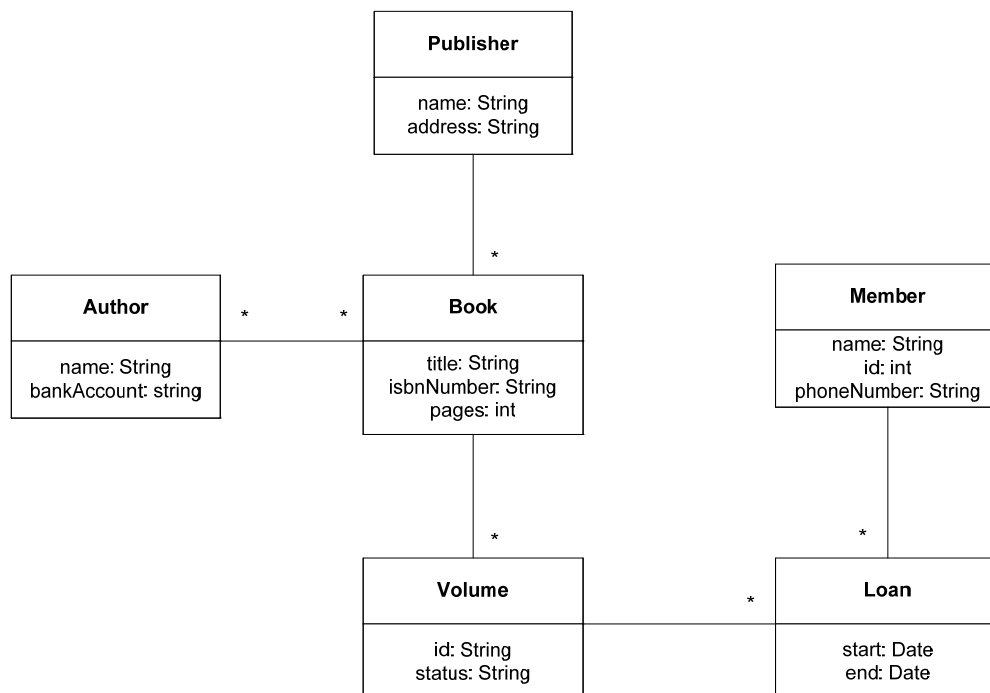


# Punto di vista delle Informazioni [SSA]

- 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
  - interesse principale: le informazioni e la loro gestione
  - importante perché lo scopo dei sistemi informatici è gestire informazioni



## Esempio (parziale): vista delle informazioni





# Punto di vista della Concorrenza [SSA]

- Punto di vista della **Concorrenza**
  - descrive l'organizzazione della concorrenza e mappa gli elementi funzionali su unità di concorrenza (processi e thread), nonché le parti concorrenti del sistema e le loro necessità e modalità di comunicazione e sincronizzazione
  - sostiene prestazioni, disponibilità e scalabilità

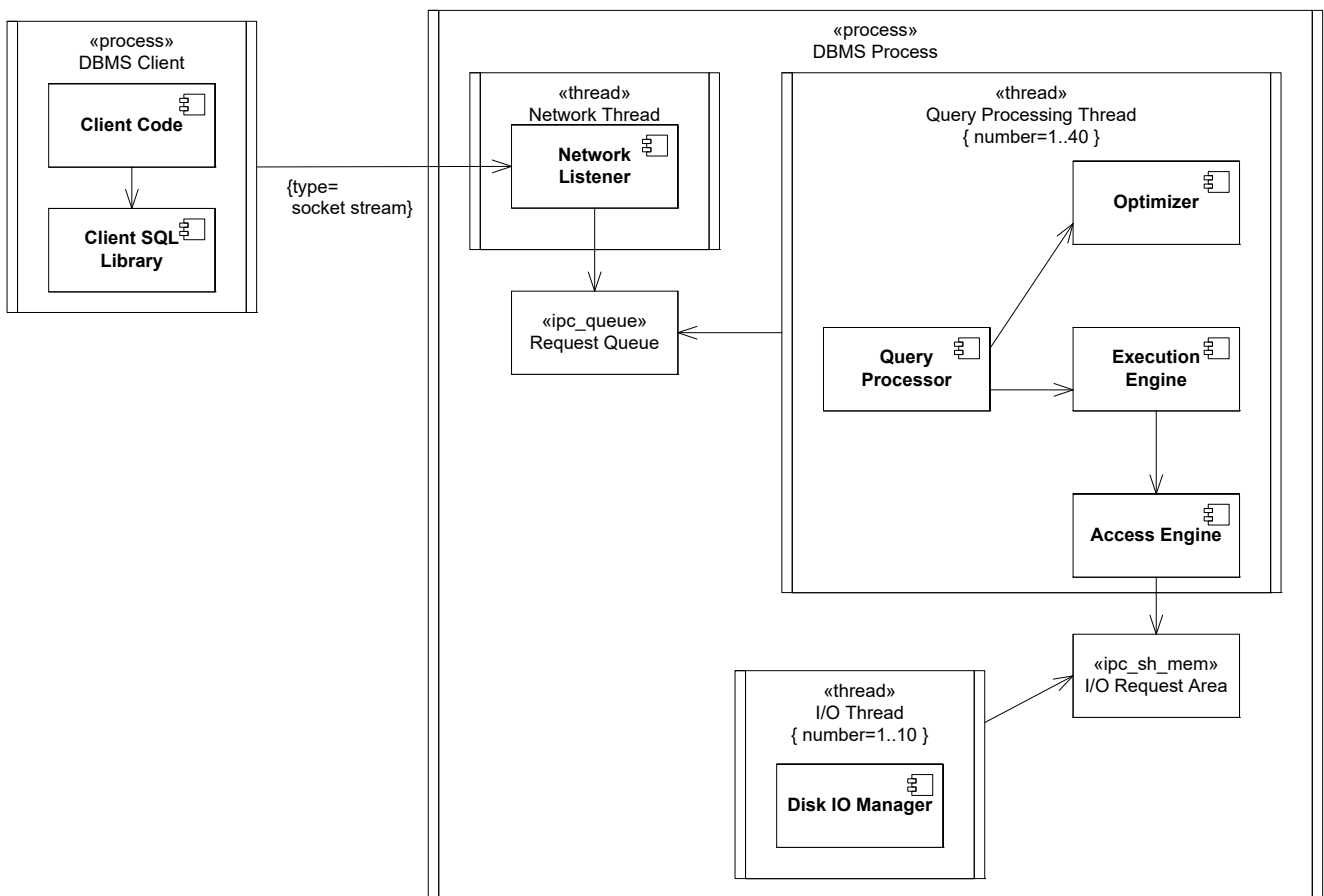
23

Viste e punti di vista architetturali

Luca Cabibbo ASW



## Esempio (parziale): vista della concorrenza



24

Viste e punti di vista architetturali

Luca Cabibbo ASW

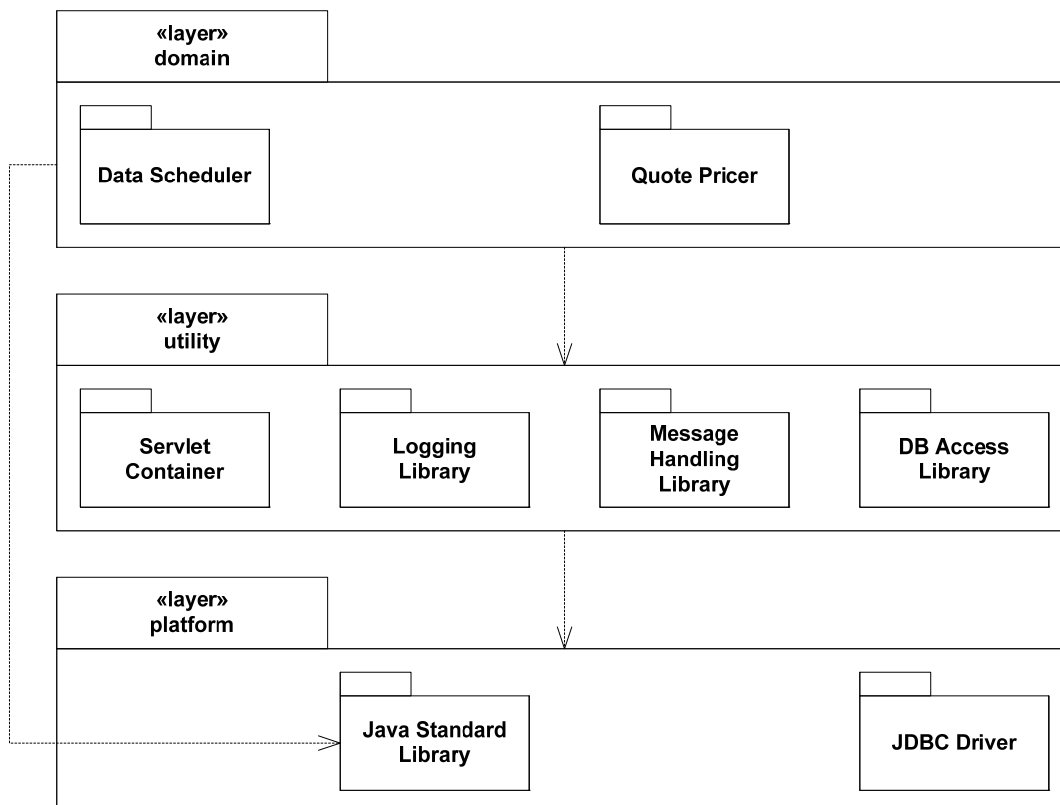


# Punto di vista dello Sviluppo [SSA]

- Punto di vista dello *Sviluppo*
  - descrive l'architettura che supporta il processo di sviluppo
  - sostiene modificabilità e verificabilità



## Esempio (parziale): vista dello sviluppo





# Punto di vista del Deployment [SSA]

- Punto di vista del *Deployment*
  - descrive l'ambiente di esecuzione in cui sarà rilasciato il sistema, comprese le dipendenze dall'ambiente runtime
  - sostiene disponibilità e scalabilità

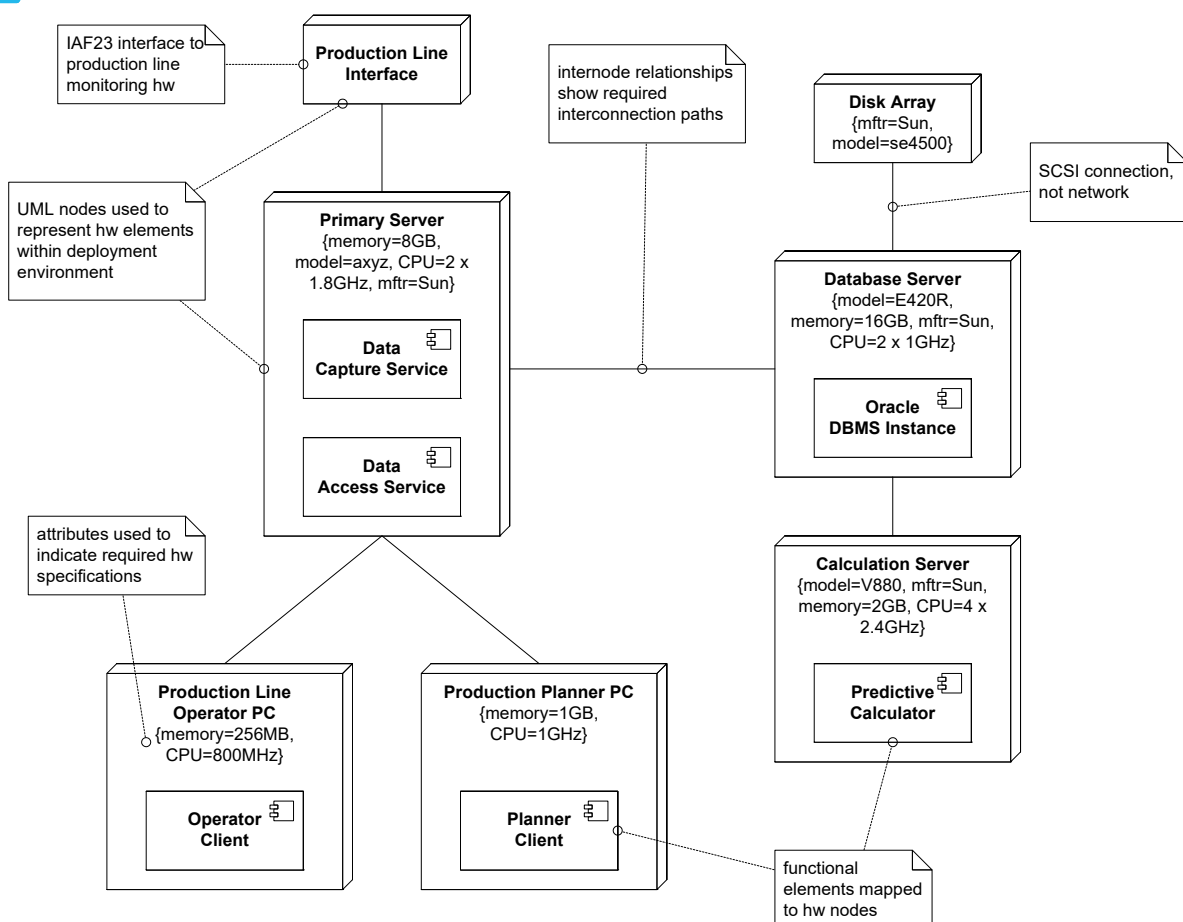
27

Viste e punti di vista architetturali

Luca Cabibbo ASW



## Esempio (parziale): vista di deployment



28

Viste e punti di vista architetturali

Luca Cabibbo ASW



# Punto di vista Operational [SSA]

## □ Punto di vista *Operational*

- ha a che fare con l'*operations* del sistema – con la gestione e l'amministrazione dell'ambiente di esecuzione e del rilascio del software in questo ambiente – per fare in modo che il sistema possa essere fruito dai suoi utenti
- descrive come il sistema sarà "operato", amministrato e supportato quando sarà in esecuzione nell'ambiente di produzione
- interessi principali: rilascio del software, monitoraggio, gestione delle configurazioni, backup, ...
- sostiene disponibilità, scalabilità e modificabilità

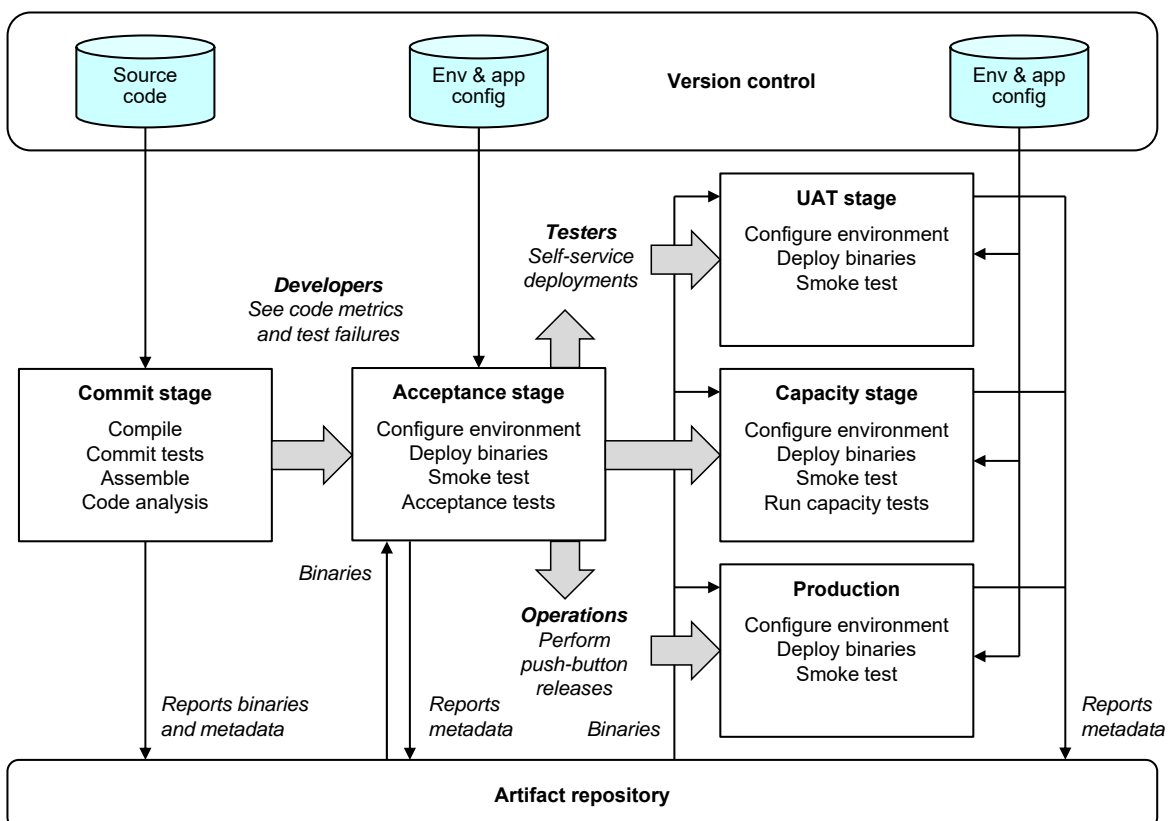
29

Viste e punti di vista architetturali

Luca Cabibbo ASW



## Esempio (parziale): vista operational



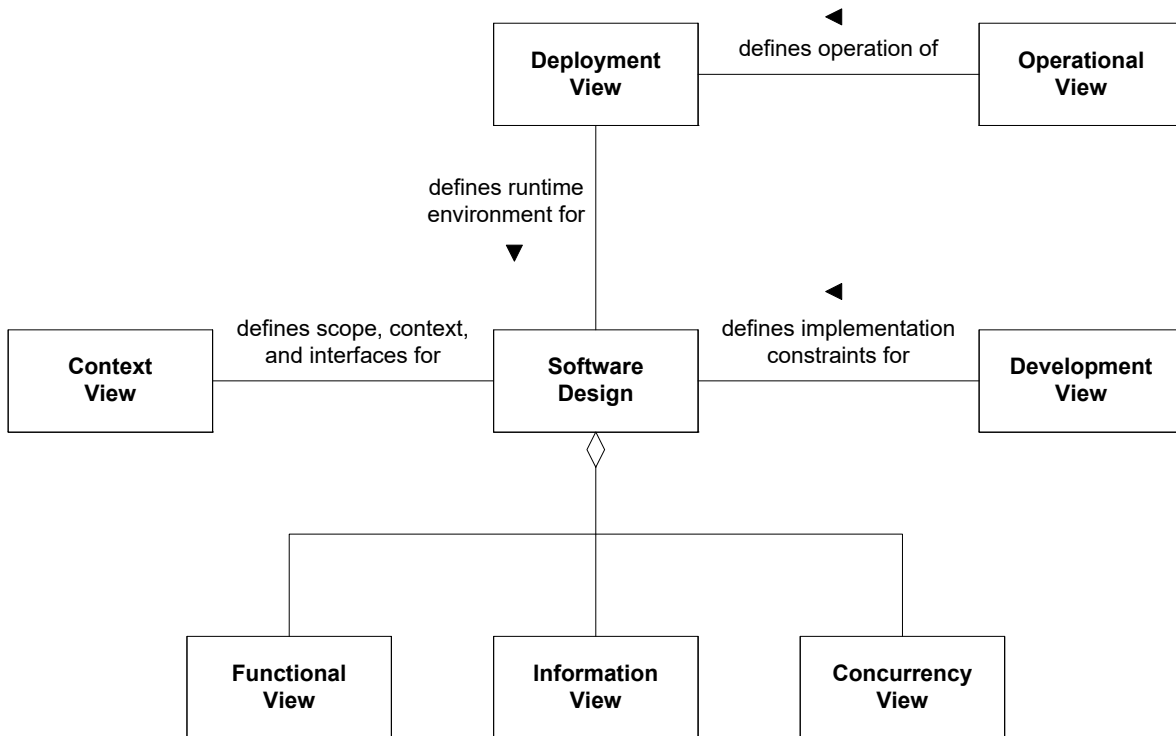
30

Viste e punti di vista architetturali

Luca Cabibbo ASW



# Punti di vista di [SSA]



31

Viste e punti di vista architetturali

Luca Cabibbo ASW



## \* Benefici e rischi legati alle viste

- L'uso di viste e punti di vista presenta dei vantaggi – ma anche alcuni possibili inconvenienti e rischi, che è necessario conoscere e sapere come evitare o mitigare

32

Viste e punti di vista architetturali

Luca Cabibbo ASW





## - Benefici legati alle viste

- Separazione degli interessi
  - la *separazione degli interessi* (*separation of concerns*) è un principio di progettazione fondamentale
    - separare interessi distinti in aree diverse, in modo che ciascuna area abbia uno scopo coeso
    - dividere il progetto in caratteristiche che si sovrappongono il meno possibile
  - la modellazione di un sistema con viste separate ma correlate favorisce le attività di analisi, progettazione e comunicazione – perché consente di concentrarsi separatamente su ciascun interesse



## Benefici legati alle viste



- Ulteriori benefici legati alle viste
  - gestione della complessità
  - miglioramento dell'attenzione/focalizzazione
  - comunicazione con le parti interessate



## - Rischi legati alle viste

- I due problemi/rischi principali legati all'uso delle viste
  - incoerenza
    - l'uso di più viste solleva il problema della coerenza tra viste
  - gestione di interessi "trasversali"
    - alcuni interessi di qualità richiedono di effettuare considerazioni nell'ambito di più viste



## Rischi legati alle viste



- Ulteriori possibili problemi/rischi legati all'uso delle viste
  - frammentazione
  - selezione di un insieme sbagliato di viste



## \* Scenari e applicazione di scenari

- L'applicazione di scenari è un approccio fondamentale alla gestione di interessi o qualità trasversali a più viste
  - le tecniche basate sugli scenari e sulla loro applicazione possono essere utilizzate in diversi ambiti
    - per identificare, descrivere e organizzare i requisiti architetaturalmente significativi
    - per guidare la progettazione dell'architettura
    - per descrivere, comunicare e comprendere l'architettura
    - per valutare (validare e verificare) l'architettura



## Scenari architeturali

- Uno **scenario architeturale** (o semplicemente **scenario**) [SSA] è
  - una *situazione* che il sistema dovrà probabilmente affrontare nel suo ambiente di produzione
  - insieme a una definizione della *risposta* richiesta dal sistema
- Due tipologie di scenari architeturali
  - *scenari funzionali*
    - uno scenario funzionale descrive una sequenza di eventi esterni a cui il sistema deve rispondere
  - *scenari di qualità*
    - uno scenario di qualità definisce come il sistema deve reagire a una situazione scatenata da un evento esterno, per esibire una certa proprietà di qualità



## Applicazione di scenari

- L'*applicazione di uno scenario* richiede di progettare, descrivere o valutare le interazioni tra un insieme di elementi architettonici – in una o più viste architettonici – che sono richieste o previste per la gestione di quello scenario
  - l'applicazione di scenari considera non solo gli aspetti **statici** delle **strutture** architettonici – ma anche gli aspetti **dinamici** delle **interazioni** tra elementi architettonici
  - la progettazione (o la lettura) congiunta delle interazioni motivate dai diversi scenari sostiene una visione complessiva dell'intero sistema
    - consente di gestire interessi trasversali e di assicurare la coerenza tra le diverse viste architettonici
  - l'applicazione di scenari rimpiazza la vista "+1" del modello a 4+1 viste di Kruchten



## \* Tipi di viste [SAP]

- [SAP] classifica le viste architettonici in tre categorie principali (più una) – chiamati *viewtypes* (*tipi di vista*) o *view styles* – di solito sulla base della natura degli elementi che contengono
  - viste a moduli
  - viste a componenti e connettori
  - viste di allocazione
  
  - viste per qualità



## Viste a moduli

### □ Viste a moduli

- gli elementi sono *moduli*
  - ai moduli vengono assegnate responsabilità funzionali
  - meno interesse sul comportamento al tempo di esecuzione
- possibili relazioni tra moduli sono, ad es., *usa*, *dipende da*, *estende*, *strati*, ...

### □ Per 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 certo modulo?
- quali moduli specializzano altri moduli?



## Viste a componenti e connettori

### □ Viste a componenti e connettori

- gli elementi sono *componenti runtime* e *connettori*
- punto di vista basato su processi/task/thread
- le relazioni sono connessioni tra componenti e connettori – collegando “porte” con “ruoli”

### □ Per 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 in parte *elementi software* e in parte *elementi in ambienti esterni*
  - le relazioni sono di corrispondenza o di allocazione
- Per 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 lavoro 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
- potrebbero però essere poco efficaci per ragionare su alcune proprietà di qualità, pervasive o di particolare importanza in un sistema
- in questi casi, [SAP] suggerisce di usare anche delle *viste per qualità* – per affrontare degli specifici interessi di qualità o rivolti a delle specifiche parti interessate

