



Università degli Studi di Salerno

Corso di Ingegneria del Software
Classe 1 Resto 0
Corso di Laurea in Informatica
A.A. 2022/23

Quiad Rationale Management Document

Versione 3.0
15/12/2022



QUIAD
Family Tree

Partecipanti al progetto e scriventi

Nome	Matricola
Di Pasquale Valerio	0512110638
Troisi Vito	0512109807

Revision History

Data	Versione	Descrizione	Autore
11/11/2022	1.0	Prima stesura RMD, indice, razionale dietro il trade-off "Doppio caching"	D.P.V. T.V.
18/11/2022	2.0	Razionale per issue "Preprocessing?"	D.P.V. T.V.
15/12/2022	3.0	Razionale per issue "Single-page application?"	D.P.V. T.V.

Indice

1. Introduzione	
1.1 Overview.....	p.4
1.2 Riferimenti.....	p.4
2. Architettura del sistema proposto	
2.1 Trade-off "Doppio caching".....	p.5
2.2 REST API vs Preprocessing.....	p.7
2.3 Single-page vs Multi-page.....	p.9
3. Glossario.....	p.11

Overview

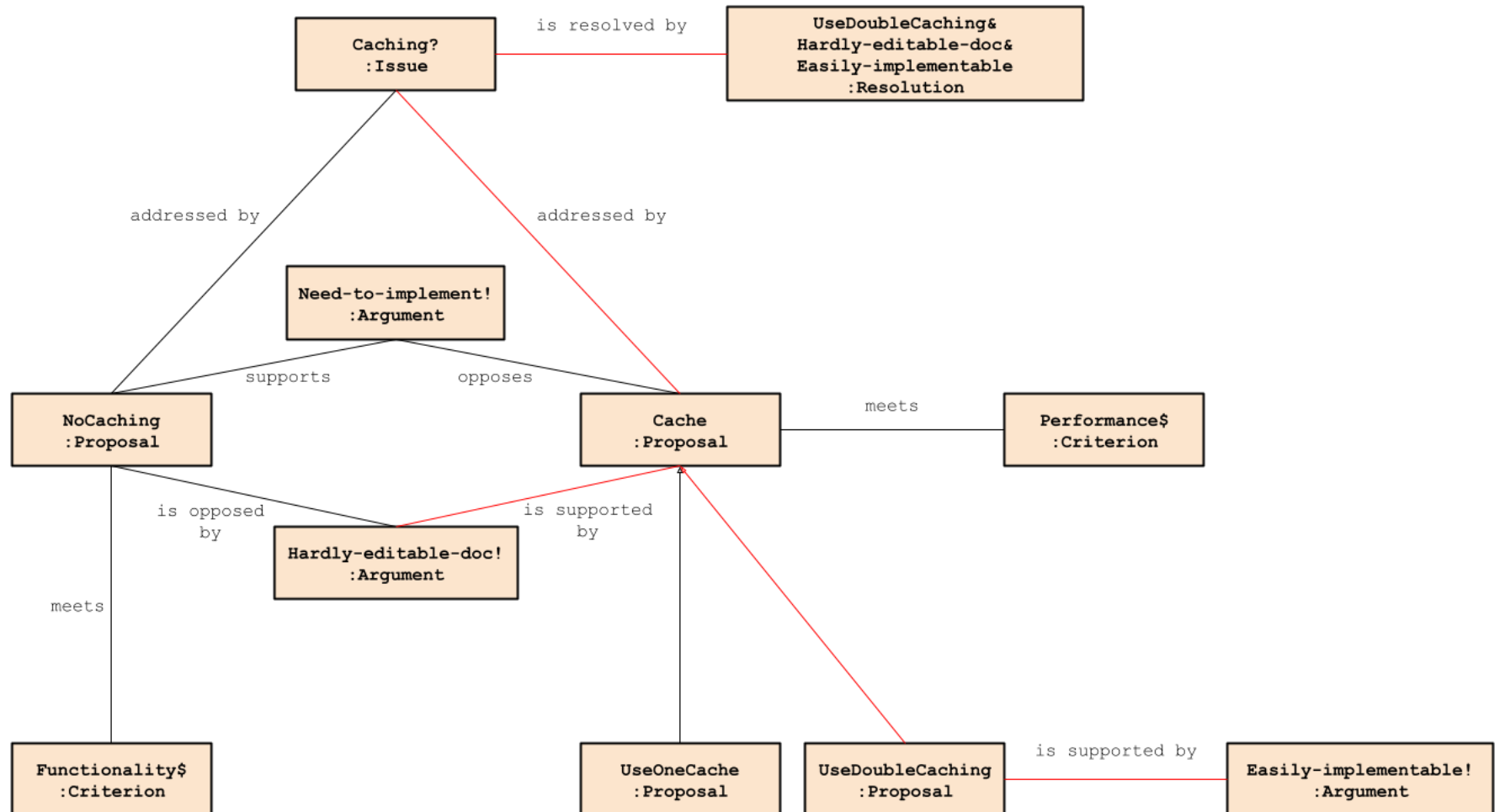
Il presente documento è atto a presentare il razionale dietro le principali scelte affrontate in fase di System Design. Saranno dunque evidenziate le issues affrontate, le alternative proposte in merito ad esse, i criteri che hanno guidato le scelte e le argomentazioni che hanno supportato la decisione finale.

Il razionale sarà espresso mediante gli opportuni diagrammi, i quali esibiranno anche la resolution proposta. Le argomentazioni saranno riportate in seguito a ciascun diagramma.

Prima di procedere nella lettura del documento in questione si raccomanda di leggere con attenzione la sezione Riferimenti ed il Glossario dei termini rilevanti.

Riferimenti

Il presente documento riguarda il razionale dietro le scelte fatte in fase di progettazione del sistema Quiad ed esplicitate nel System Design Document (SDD). Si raccomanda dunque di osservare in questi ultimi l'applicazione concreta delle risoluzioni alle issues proposte.

Trade-off "Doppio caching" (riferimento SDD p.4)

Argomentazione "Need-to-implement"

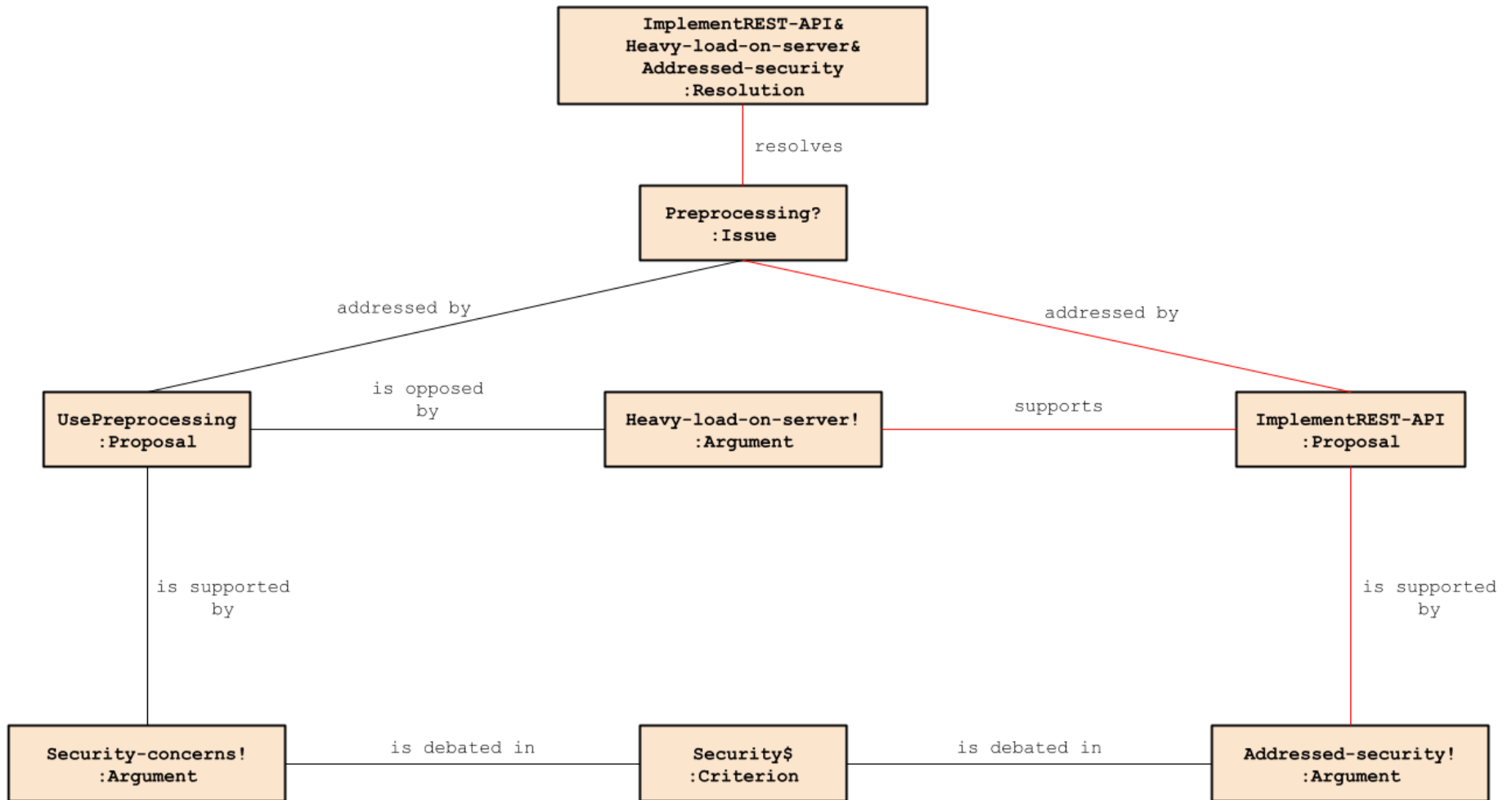
Come sarà probabilmente apparente, l'utilizzo di una tecnica di caching, o doppio caching, necessita dell'implementazione di tale tecnica, sul client e sul server.

Argomentazione "Hardly editable-doc"

Si osservi che un documento, se richiesto dall'utente, dev'essere stato approvato da un utente supervisore in quanto è stato reso disponibile alla visualizzazione. Osservando che i documenti sono di carattere storico, una ragionevole ipotesi potrebbe osservare che poco probabilmente un documento sarà soggetto a modifiche.

Argomentazione "Easily-implementable"

L'implementazione del meccanismo, sia esso lato client o server, consiste in una struttura dati che attui un'associazione chiave-valore di elementi (i.e. i documenti) a cui è associata una data scadenza (e.g. un'ora). Molte delle tecnologie utilizzate per l'implementazione di sistemi client/server sono provvisti di API che consentono l'uso del caching. La difficoltà di implementazione, è pertanto meno significativa del guadagno in termini di performance.

REST API vs Preprocessing (riferimento SDD p.10)

Argomentazione "Heavy-load-on-server"

Impiegando il preprocessing, il server dovrà generare una pagina da zero per ogni richiesta del client. Il preprocessing va dunque ad appesantire il carico che grava sul server.

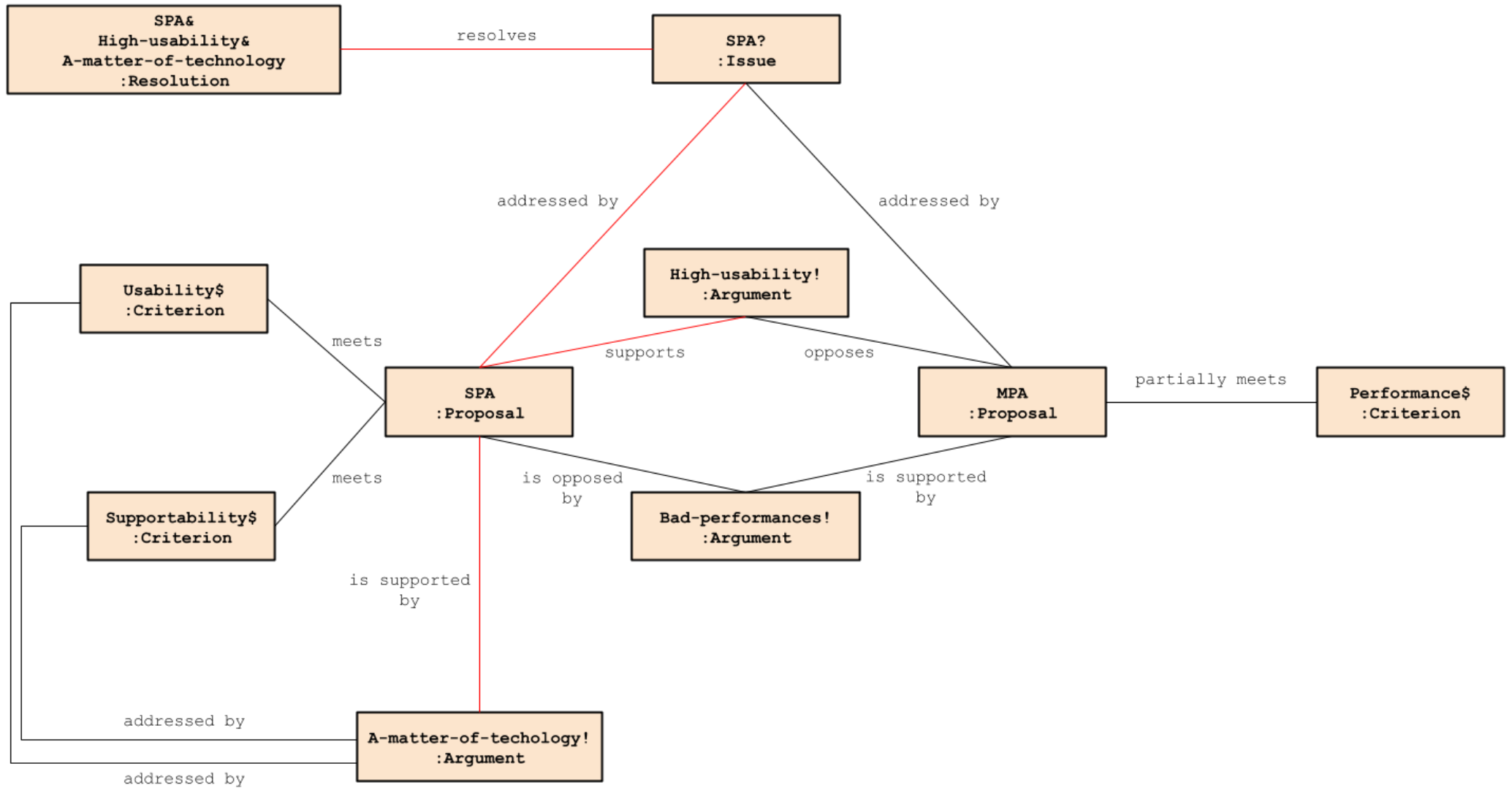
Argomentazione "Security-concerns"

L'uso di una REST API produce dei significativi problemi di sicurezza, nel caso in cui vengano inviati dati sensibili, specie su connessione non sicura.

Argomentazione "Addressed-security"

I client utilizzeranno il protocollo *https* per le richieste da inoltrare verso il server. L'access control sarà inoltre verificato per via programmatica mediante la access table (di cui al SDD).

Single-page application vs Multi-page application (riferimento SDD p.8)



Argomentazione "High-usability"

L'usabilità di una SPA deriva dal fatto che, mediante tale paradigma, non è più necessario, anche da parte dell'utente, ricaricare la pagina né assistere ai cosiddetti "full page reloads".

Argomentazione "Bad-performances"

Per quanto non sia un criterio considerato in termini di requisiti non funzionali, è questione ben nota che con le SPA siano soggette a cali di performance più considerevoli, dovuti alla continua interazione con il server al fine di ottenere le sezioni mancanti delle pagine.

Argomentazione "A-matter-of-technology"

Angular in particolar modo, effettua il pre-rendering delle pagine lato back-end, così da alleggerire il carico client-side. Ciò, unito al fatto che tale tecnologia consente di rispondere agevolmente ad alcuni requisiti di usabilità, quali RNFU-1, RNFU-3, RNFU-4 oltre che di supportabilità, quale RNFS-1 (a tal proposito, si osservi il RAD sezione 3.3.1). Ciò agevola l'uso di Angular e consente l'attuazione del paradigma Single-page application senza violare requisito alcuno.

Glossario

Quiad	<p>Il nome del progetto e della piattaforma che ne concretizza i servizi. Consente la gestione del proprio albero genealogico e la ricerca di associata documentazione storica.</p> <p>E' latino, e significa "A chi"!</p>
Issue/ Problematica	I problemi decisionali affrontati in fase di System Design. Richiedono l'analisi della alternative proposte.
Alternativa/ Proposal	Le differenti proposte di soluzione per un'assegnata issue. Possono, a loro volta, sollevare problematiche.
Criterion/ Criterio	Una metrica di giudizio della "bontà" di un'alternativa. Per comprendere al meglio i criteri, si raccomanda la lettura degli obiettivi di progettazione esposti nel System Design Document.
Argument/ Argomentazione	Un'arringa che supporta o contrasta un'alternativa. Le argomentazioni costituiscono l'approccio principale al razionale ivi presentato.
Resolution/ Risoluzione	La decisione di intraprendere una data alternativa proposta, così da chiudere (i.e. risolvere) una issue.