

# Il Manifesto dei Sistemi Reattivi

---

*Publicato il 16 Settembre 2014 (v2.0)*

Enti ed aziende operanti in vari settori stanno sperimentando in modo indipendente dei pattern architetturali simili per forgiare sistemi software. Questi sistemi si dimostrano robusti, resilienti e flessibili, e sono in grado di soddisfare al meglio le esigenze delle applicazioni software moderne.

Questo movimento si verifica perchè i requisiti delle applicazioni software sono cambiati drasticamente negli ultimi anni. Fino a pochi anni fa le applicazioni di grandi dimensioni erano ancora caratterizzate dall'impiego di decine di server, dall'aver tempi di risposta nell'ordine dei secondi, dal richiedere ore di manutenzione offline e dal gestire pochi gigabyte di dati. Le applicazioni di oggi sono deployate su ogni genere di device, da quelli mobili ai cluster cloud-based che utilizzano migliaia di processori multi-core; gli utenti di queste applicazioni si aspettano tempi di risposta nell'ordine dei millisecondi e un uptime pari al 100%. L'unità di utilizzo dei dati è il Petabyte. Molto semplicemente, le esigenze delle applicazioni di oggi non possono essere soddisfatte dalle architetture software del passato.

Crediamo che sia necessario un approccio coerente all'architettura dei sistemi software, e crediamo che i suoi elementi chiave siano già individuabili singolarmente: siamo a favore di sistemi che siano Responsivi, Elastici e Orientati ai Messaggi. Li chiamiamo Sistemi Reattivi.

I Sistemi Reattivi sono flessibili, a basso accoppiamento e [scalabili](#): ciò li rende più semplici da sviluppare e malleabili al cambiamento. Tali sistemi sono di gran lunga più tolleranti ai [guasti](#) e reagiscono ad essi in modo elegante e non brusco. I Sistemi Reattivi sono altamente responsivi, offrendo agli [utenti](#) un feedback concreto e interattivo.

## I Sistemi Reattivi sono:

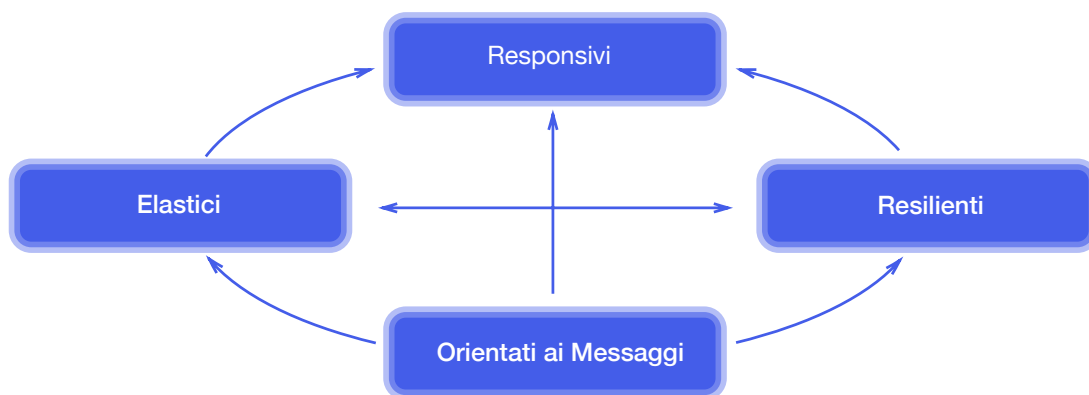
**Responsivi:** Il [sistema](#), se è in generale possibile dare una risposta ai client, la dà in maniera tempestiva. La responsività è la pietra miliare dell'usabilità e dell'utilità del sistema; essa presuppone che i problemi vengano identificati velocemente e gestiti in modo efficace. I sistemi responsivi sono focalizzati a minimizzare il tempo di risposta, individuando per esso un limite massimo prestabilito di modo da garantire una qualità del servizio consistente nel tempo. Il comportamento risultante è quindi predicibile, il che semplifica la gestione delle situazioni di errore, genera fiducia negli utenti finali e predispone ad ulteriori interazioni con il

sistema.

**Resilienti:** Il sistema resta responsivo anche in caso di [guasti](#). Ciò riguarda non solo i sistemi ad alta disponibilità o mission-critical: infatti, accade che ogni sistema che non è resiliente si dimostrerà anche non responsivo in seguito ad un guasto. La resilienza si acquisisce tramite [replica](#), contenimento, [isolamento](#) e [delega](#). I guasti sono relegati all'interno di ogni [componente](#), isolando così ogni componente dagli altri e quindi garantendo che il guasto delle singole porzioni del sistema non comprometta il sistema intero. Il recupero di ogni componente viene delegato ad un altro componente (esterno) e l'alta disponibilità viene assicurata tramite replica laddove necessario. I client di un componente vengono dunque alleviati dal compito di gestirne i guasti.

**Elastici:** Il sistema rimane responsivo sotto carichi di lavoro variabili nel tempo. I Sistemi Reattivi possono adattarsi alle variazioni nella frequenza temporale degli input incrementando o decrementando le [risorse](#) allocate al processamento degli stessi. Questo porta ad architetture che non hanno né sezioni contese né colli di bottiglia, favorendo così la distribuibilità o la replica dei componenti e la ripartizione degli input su di essi. I Sistemi Reattivi permettono l'implementazione predittiva, oltre che Reattiva, di algoritmi scalabili perché fondati sulla misurazione real-time della performance. Tali sistemi, raggiungono l'[elasticità](#) in maniera cost-effective su commodity hardware e piattaforme software a basso costo.

**Orientati ai Messaggi:** I Sistemi Reattivi si basano sullo [scambio di messaggi asincrono](#) per delineare per ogni componente il giusto confine che possa garantirne il basso accoppiamento con gli altri, l'isolamento e la [trasparenza sul dislocamento](#) e permetta di esprimere i [guasti](#) del componente sotto forma di messaggi al fine di delegarne la gestione. L'utilizzo di uno scambio esplicito di messaggi permette migliore gestibilità del carico di lavoro, elasticità e controllo dei flussi di messaggi mediante setup e monitoraggio di code di messaggi all'interno del sistema e mediante l'applicazione di [feedback di pressione](#) laddove necessari. Uno scambio di messaggi trasparente rispetto al dislocamento rende possibile, ai fini della gestione dei guasti, l'utilizzo degli stessi costrutti e semantiche sia su cluster che su singoli host. Uno stile di comunicazione [non bloccante](#) fa sì che l'entità ricevente possa solo consumare le [risorse](#), il che porta ad un minor sovraccarico sul sistema.



I sistemi software di grandi dimensioni si compongono di sistemi più piccoli e dunque dipendono dalle proprietà Reattive dei loro costituenti. Ciò significa che i Sistemi Reattivi si fondano su principi di design che considerano tali proprietà ad ogni livello della scala architeturale, rendendo così componibili i sottosistemi Reattivi. I sistemi software più grandi al mondo hanno architetture basate sui principi Reattivi e riescono ad operare servendo miliardi di utenti ogni giorno: è giunto il momento di utilizzare questi principi con consapevolezza e dall'inizio del processo di design, evitando di riscoprirli nuovamente ogni volta.