
# Project:

## Optimizing Hardware usage in multi-cloud environment

Teacher: Henry Muccini
Project Partner: Daniele Spinosi, Micron Technology

## Deliverable Template

| | |
|---|---|
| **Date** | **13/12/2020** |
| **Team ID** | **qwertyasdf** |

| Team Members | | |
|---|---|---|
| **Name and Surname** | **Matriculation number** | **E-mail address** |
| **Leonardo Serilli** | | **leonardo.serilli@student.univaq.it** |
| **Gabriele Colapelle** | | **gabriele.colapelle@student.univaq.it** |
| **Alessando d’Orazio** | | **alessandro.dorazio2@student.univaq.it** |
| **Pietro Ciammaricone** | | **pietro.ciammaricone@student.univaq.it** |


# Table of Contents

# Challenges/Risk Analysis

*In this section, you should describe, using the table below, the most challenging or discussed or risky design tasks, architecture requirements, or design decisions related to this project. Please describe when the risk arised, when and how it has been solved.*

| Risk | Date the risk is identified | Date the risk is identified | Explanation on how the risk has been managed |
|---|---|---|---|
| **T1 downtime** I servizi T1 devono avere il **99.999% di disponibilità** | **5/12/2020** | **7/12/2020** | **In ogni factory, almeno due data center distanti tra loro** in modo che, anche con la perdità di uno dei due, la factory continuerà ad operare grazie alla ridondanza. Maggiore è il numero di data center per factory maggiore è la **fault tolerance** del sistema. |
| **Gestione Fault Tolerance** | **9/12/2020** | **10/12/2020** | **La fault tolerance** è gestita principalmente a livello **hardware,** ottimizzando il numero di data center per ogni factory**,** ma anche a **livello software** |
| **Bilanciamento Workload CPU** | **9/12/2020** | | |
| | | | |
| | | | |

# Spec Refinement

*In this section, you should revise/improve/specialize the project specification, by reporting any reasoning, extra requirements, requirements specialization, you made to* better understand the project.

*DO NOT SPEND TIME IN OVER-DOCUMENTING REQUIREMENTS. THIS ASSIGNMENT IS ABOUT THE SYSTEM ARCHITECTURE AND ARCHITECTURE DECISIONS.*

1. I servizi si dividono in due fasce di criticità, T1 e T2, in questi rientrano le seguenti categorie:
    a. Servizi T1: incudono I servizi safety critical, buisness critical e mission critical;
    b. Serivizi T2: includono I servizi che non impattano sulla produzione, ad esempio quelli che collezionano statistiche di produzione .
2. Ogni servizio/applicazione si deve occupare di **una funzionalità.** Questo implica che ogni servizio deve svolgere solamente le operazioni che riguardano l’ambito di quella funzionalità.
3. I servizi T1 devono avere il **99.999% di disponibilità** ; ogni servizio identificato come T1 deve essere eseguito localmente e non deve mai essere inattivo, poichè ne risentirebbe la produzione.
4. La fault tolerance deve essere garantità dai data center.
5. I data center interni alla factory devono ospitare i servizi T1 e inoltre devono anche essere collegati tramite un servizio LAN dedicato a bassa latenza.
6. I servizi T2 devono essere gestiti in un **cloud pubblico**
7. Un servizio identificato come T2 può essere inattivo per un massimo di due ore senza impattare sulla produzione, quindi Il contratto WAN per tali servizi deve garantire un MTTR minore di due ore.
8. In caso di “grandi fallimenti”, i servizi **T2** devono poter operare anche con **prestazioni degradate**
9. I **servizi MES** devono essere inseriti in containerizzati con **Docker** e istanziati in un **cloud privato open shift.**
10. Tra data center dislocati in factory ci deve essere una WAN privata a bassa latenza
11. I **servizi MES** devono essere inseriti in container e devono essere istanziati in un **cloud privato.**
12. L’utilizzo della **CPU** deve essere **massimizzato**
    a. Ogni server ha un **50% di utilizzo di CPU medio**
    b. I servizi **T1** lavorano con il **40%** di cpu workload totale
    c. I servizi **T2** lavorano con il **60%** di cpu workload totale
13. I servizi devono essere scalati e bilanciati con **kubernetes**.
14. Una Struttura ha **3 Factory**
15. Ogni **Factory** ha **3 data center**
    a. 2 operativi
    b. 1 ridondante
16. Ogni data center ha **10 server**

    a. I server possono avere più di una cpu, al fine di bilanciare in modo corretto il workload totale
17. Nel caso di **crash** di un data center, deve essere garantita abbastanza potenza di calcolo

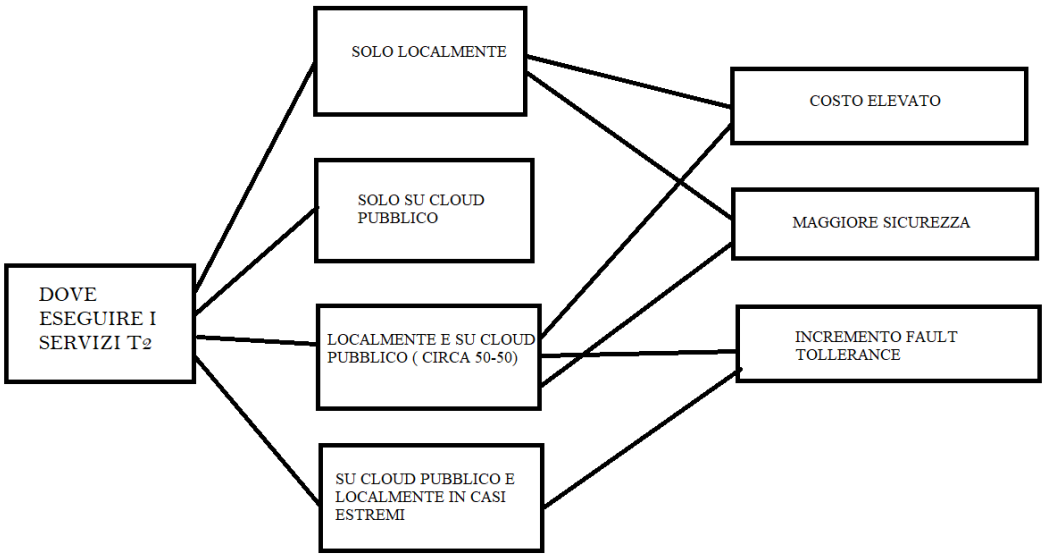
# Design Decisions

*In this section, you are required to document up to five (5) most important architecture design decisions using the QOC Template. For “most important” we mean those that may have a bigger impact on your architecture, those that you discussed more, or you think are the most relevant. The QOC Template is available in Schoology. You are required to provide both the tabular representation and the graphical one inside this document.*

- Per i servizi T2 devono essere presenti delle istanze on demand, qualora si verifichino grandi fallimenti sui data center in cui generalmente operano. In questo modo tali servizi, nonstante costretti a lavorare con prestazioni degradate, avrebbero una disponibilità elevata.
  Le istanze on demand avranno capacità minori, in modo da ammortizzare i costi; tale decisione è resa possibile dal fatto che i servizi T2 non sono critici per il sistema.

| **Concern** | | Dove far girare I servizi T2 |
|---|---|---|
| **Alternative(s)** | | 1. Solo localmente<br>2. Solo su cloud pubblico<br>3. Localmente e su cloud pubblico<br>4. Principlamente su cloud pubblico ma anche localmente in caso di necessità |
| **Ranking criteria** | | 1. Comprare hardware per far girare I servizi T2 localmente, gestire i servizi localmente ( + costi iniziali)<br>2. Affitto del cloud pubblico con minore sicurezza ( - costi inziali, - sicurezza)<br>3. Aggiungere hardware e affittare cloud pubblico ( + costi, + sicurezza)<br>4. Affittare cloud pubblico e lasciare un margine di CPU nei server privati per gestirli localmente in casi estremi ( - costi + fault tolerance) |
| **Architectural decision** | **Identifier** | 4 |

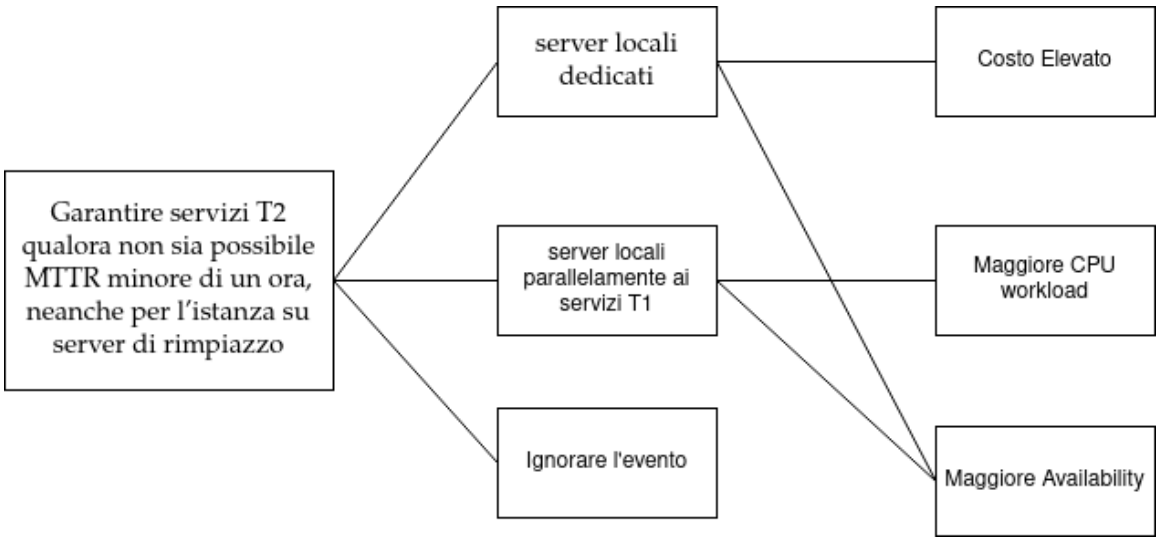
| | | |
|---|---|---|
| | **Description** | I servizi T2 verranno eseguiti su un cloud pubblico, in caso di grandi fallimenti del cloud, il lavoro verrà spostato temporaneamente su data center locali |
| | **Status** | |
| | **Relationship(s)** | |
| | **Rationale** | Ogni servizio T2 deve occuparsi di una funzionalità |




- Nel caso non possa essere garantito MTTR minore di un ora per i servizi T2, anche per l’istanza dei servizi eseguita sul data center di rimpiazzo, un istanza deve temporaneamente essere eseguita sui data center locali.

| | | |
|---|---|---|
| **Concern** | | Garantire servizi T2 qualora non sia possibile MTTR minore di un ora, neanche per l’istanza su server di rimpiazzo |

| | | |
|---|---|---|
| **Alternative(s)** | | 1. Spostarli temporaneamente su server locali dedicati.<br>2. Spostarli temporaneamente su server locali parallelamente ai servizi T1<br>3. Ignorare l’evento descritto per via della sua bassa probabilità |
| **Ranking criteria** | | 1. Acquistare hardware dedicato (+ costi, + availability)<br>2. Utilizzo CPU distribuito tra T1 e T2 ( - costi, + CPU workload, + availability) |
| **Architectural decision** | **Identifier** | 2 |
| | **Description** | Nel caso non possa essere garantito l’MTTR indicato, neache sul data center di rimpiazzo, un istanza dei servizi T2 verrà temporaneamente eseguita sui data center locali in parallelo con I servizi T1 |
| | **Status** | |
| | **Relationship(s)** | |
| | **Rationale** | Ogni server ha il 50% di utilizzo di CPU medio, 40% dedicato ai servizi T1, e il 10% restante può essere utilizzato per I servizi T2 |

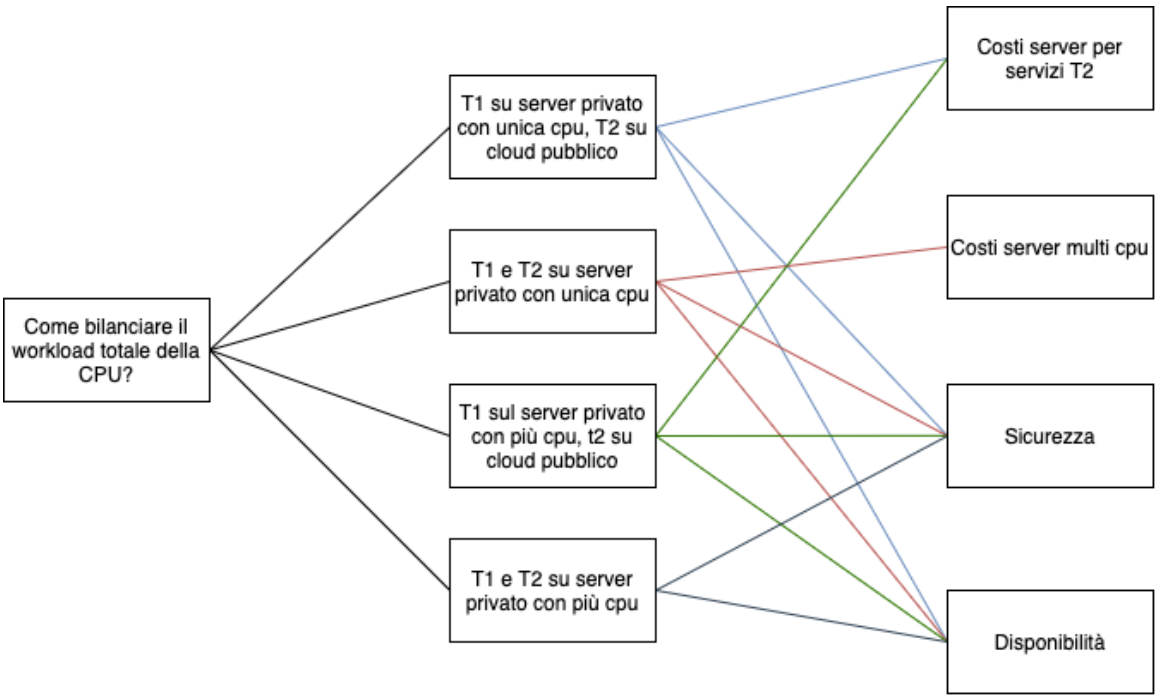


- La Fault tolerance verrà garantita principalmente a livello hardware, prevedendo che ogni factory abbia 3 data center di cui 2 operativi ed uno ridondante, in modo tale che nel caso in cui un data center (o entrambi) smetta di essere operativo, il terzo data center può operare al posto suo.

| | | |
|---|---|---|
| **Concern** | | Garantire operabilità dei servizi T1 al 99.9999% |
| **Alternative(s)** | | 1) Solo due datacenter con il 50% del carico |
| **Ranking criteria** | | Lievitazione dei costi hardware<br>Costi di manutenzione più elevati<br>Costo in spazio fisico nella factory |
| **Architectural decision** | **Identifier** | 3 |
| | **Description** | I data center principali opereranno sotto il 50% di carico in modo tale da garantire che nel caso due di essi falliscano, il terzo possa prendersi in carico il workload di entrambi. (Si assume come da specifica che non ci siano prestazione degradate con utilizzo CPU > 80%) |
| | **Status** | |
| | **Relationship(s)** | |
| | **Rationale** | Nonostante di data center si trovano in luoghi fisicamente separati (Seppur nella stessa factory), questo non garantisce una fault tolerance totale, inoltre reputiamo meglio avere due datacenter al 50% (nel caso uno fallisca) che uno al 100% |

4. Ogni server di un data center della factory è provvisto di più cpu, in modo tale che la cpu abbia il 50% di workload medio, e che a un servizio T1 possa esserne dedicato il 40%. Inoltre il 10% rimanente può essere dato ai servizi T2 nel caso straordinario descritto al punto 2 delle Design Decision.

| | | |
|---|---|---|
| **Concern** | | Come bilanciare il workload totale della CPU |
| **Alternative(s)** | | 1. Servizi T1 sul server privato con unica cpu, T2 su cloud pubblico<br>2. T1 e T2 su server privato con unica cpu<br>3. T1 sul server privato con più CPU, T2 su cloud pubblico<br>4. T1 e T2 su server privato con più cpu |
| **Ranking criteria** | | 1. Costi relativi all’acquisto dei server che facciano girare anche i servizi T2<br>2. Costi relativi all’acquisto di server multi cpu<br>3. Sicurezza. I dati manipolati dai servizi non devono essere accessibili da terzi<br>4. Disponibilità. I servizi T1 devono avere il 99,999% di disponibilità |
| **Architectural decision** | **Identifier** | 16a |
| | **Description** | I servizi T1 saranno eseguiti sui server privati, mentre i servizi T2 saranno eseguiti sul cloud pubblico. In alcuni casi straordinari, i servizi T2 potranno essere eseguiti sui server privati. Per ottimizzare il workload dei server, si utilizzano server multicpu |
| | **Status** | |
| | **Relationship(s)** | |
| | **Rationale** | Un server può avere più di una CPU |


Come bilanciare il workload totale della CPU?
T1 su server privato con unica cpu, T2 su cloud pubblico
T1 e T2 su server privato con unica cpu
T1 sul server privato con più cpu, t2 su cloud pubblico
T1 e T2 su server privato con più cpu
Costi server per servizi T2
Costi server multi cpu
Sicurezza
Disponibilità